Modelling the transfer prices of football players

BY

IVO HENDRIKS (195549)

A thesis submitted in partial fulfillment of the requirements for the degree of Master in Econometrics and Mathematical Economics

Tilburg School of Economics and Management Tilburg University

Supervised by:

dr. P. Cizek

September 2017

Contents

1 Management Summary

The growing popularity of professional football as well as its economic significance have caused investors worldwide to devote their attention towards this market. This is accompanied with an increase in information availability regarding among other things salaries and transfers. A higher transfer fee is often accompanied with a higher expectation on that players' ability. However, these transfers cause debates each year, because players may or may not perform up to expectations during the transfer. The factors that determine whether or not the players will be successful and therefore what they are worth are complex. Determining a fair price for a player therefore is a challenge. A scientific grasp on how certain characteristics impact prices, can be beneficial in deciding upon transfer prices and verifying whether asking prices of competitors are valid. This makes it an interesting topic for football clubs and all that are associated with football. As a result, the research question that this research attempts to answer is as follows: *'What is the predicted transfer price of a football player, given its characteristics?'*

In an attempt to answer this research question, multiple sets of data are combined, to include many variables that potentially influence transfer prices. Next, multiple models are constructed and repeatedly tested. The results show that a regression analysis that performs both a selection procedure as well as regularization performs best for predicting transfer prices of football players. Furthermore, the remaining time on a contract, the strengths of the clubs involved in the transfer, the performance of the player on the field, the age of the player, the year in which the transfer took place and the amount of games played by the player, play important roles when determining the transfer price of a football player. Furthermore, this research finds light evidence that the Premier League clubs pay more for similar players than clubs from other leagues. It argues that this is due to the giant football economy England benefits from, and that this is reflected in the money spent on football players. It furthermore finds that clubs influenced by large investors generally pay larger transfer sums, and that three of the four best team in Europe are able to attract players for a smaller transfer sum. Finally, results show that clubs just below the very best, generally receive larger transfer sums. Nevertheless, the models lack valuable features, such as interests from multiple clubs, and the reason for the transfer to take place. The models are also vulnerable to selection bias, as the models are built upon publicly available data. Future research may improve results, by making the necessary adjustments to deal with these issues.

2 Introduction

Football has become one of the most popular sports, and likely the most popular sport throughout the world [31]. The amount of money that is involved in top professional football clubs is over hundreds of millions of dollars [19]. The Premier League alone for example, England's top football league, enjoyed revenues exceeding 2.40 billion in the season of 2016-17 [37]. All teams in the league received giant fees for participating, as well as, among others, multiple payments for broadcasting rights and commercial rights [16] [36]. The financial income of a football club is largely dependent on the sportive successes this club acquires, since broadcasting money and marketing campaigns depend on your clubs position nationally, as well as internationally [40]. Furthermore, its sportive success depends on the team performing well in different leagues each year. Many questions arise throughout such a year on how to accomplish these successes.

In recent years, an increase in data driven decision making in almost all businesses has occurred, including sports. Baseball, basketball, rugby and tennis, all use many statistics and data sources to optimize decisions, with large successes. On this topic, however, football is far behind. Although the football world is starting to make use of the increased availability and knowledge surrounding data, key decisions are still often based solely on what can be perceived. As a result, successful coaches nowadays rely on their tactical opinion, while remaining distant from data analysis and empirical evidence.

One of the topics for which data could be used is the buying and selling of players. There are many possible characteristics that could influence the transfer price of a player, such as the age of a player, whether or not he is able to play on a certain position and his foot of preference. However, buying a player also brings risks to the table, as it is hard to foresee what the precise effect of a transfer is. Moreover, differences between image of players and clubs might influence transfer prices. All together, this creates interesting questions which data analysis might be able to answer. For example, how do clubs come to an agreement on what a player is worth? And why does one club stop bidding when a certain price is asked, while another club remains interested?

The goal of this research is to predict the transfer price of individual football players. This task requires solid data on players, as well as information regarding the transfers of these players. In doing so, a fundamental grasp of the impact of characteristics on the transfer prices in the football world can be managed and used to validate and construct transfer prices. This will be beneficial for all football teams and the firms associated with these teams. If the model returns precise transfer predictions, it can be applied in the upcoming transfer seasons and a thorough understanding of transfer prices can be achieved.

In this research, transfer prices are explained by different individual characteristics such as height and foot of preference, the clubs involved in the transfer, the remaining contract duration and a score representing their skills on the field in different seasons [49]. As a result, this research will discuss the influence of different variables such as a player's age, the number of goals scored and the strength of the clubs involved. Therefore, the corresponding research question of this research is: *'What is the predicted transfer price of a football player, given its characteristics?'*

When attempting to answer this research question, difficulties arise surrounding the possible heterogeneity between players and between football clubs. For example, it is unlikely that all football players are identical in their motivation and talent. Next to that, football clubs might have different reasons regarding the buying and selling of players. Moreover, both players and clubs could differ in their image or brand identity, which might influence transfer prices. In order to tackle this difficulty, this research tries to capture the heterogeneity between players by including a score that captures the performance of each player, such as the number of passes, shots and crosses on the field. This way, we can distinguish players having the same physical attributes, based on their level of skill. With regards to clubs, we try to capture heterogeneity by including a value that corresponds to the sportive

strength of the involved clubs in the transfer [46]. Furthermore we include zero-one variables that capture the positive or negative fluctuation in the buying or selling of a player for each club. Reasons such as a strong brand identity or having many investors can potentially be measured this way.

To try and come up with a reliable data set for prediction, we have gathered four sets of data from three different sources. After gathering the data from the three different sources, we need to structure and combine them, in order to apply the data in prediction. The first unstructured data set contains roughly 300.000 transfers, of which over 43000 unique players. The second data set contains many personal characteristics of a player, ever being transferred. Both these sets are gathered from the same site, which is a highly respected and well-known football information site [50]. The third data set contains in-field information on all players playing in the top leagues throughout the world, from the years 2012 and onwards [49]. The fourth data set contains an index that represents the sportive strength of the corresponding club, for all dates after 2007 [46].

This research uses different methods to predict transfer prices, finding which model is most appropriate to predict these transfer prices is expected. Furthermore we expect that the level of skill of the player on the field and the strength of the clubs involved in the transfer have great influence on the transfer price, and are key in predicting the transfer price. The results coincide with these expectations.

The remainder of this report is structured as follows: The next chapter will explain the most important work regarding this topic. Furthermore, it will point out some possible improvements and discusses how this research tries to incorporate these improvements. The fourth chapter introduces the data set that is used for modelling, the way in which it is gathered, and what has been done to structure it. It will thoroughly take you through the process of web scraping and explain its potential fall-backs. Furthermore, in chapter five the variables are introduced and defined, emphasize is placed on the Squawka Scores and the Euro Club Index, which are both included in an attempt to control for player- and club performance. The sixth chapter given an overview of econometric techniques used in this research and introduces the different models. It will also explain the statistical techniques that are applied to account for overfitting and to measure prediction performance. The seventh chapter discusses heterogeneity issues and how it attempts to solve the problem. It will continue by discussin possible improvements to the model fit by transforming, removing and adding certain variables. The eighth chapter show the results of this research. After which chapter nine summarizes and concludes the main accomplishments of this research, and gives recommendations for future research.

3 Previous work

So far, academic research has mainly focused its attention on sports such as baseball, basketball and American football, whereas the academic research on football is still behind [19]. However, some research is done regarding the pricing of football players. For example, the influence of club characteristics on transfer fees show significant results, among others, regarding the rank of the football club and the influence this has on the buying as well as the selling of players [13]. Furthermore, research regarding the influence of player characteristics on transfer fees, found that age negatively influences the transfer price and that counter-intuitively, scoring a goal does not increase the worth of a player [38]. However, contradicting these findings, evidence is found for a positive effect of age, as well as positive effects of the amount of games played and goals scored [11]. This is once more verified in another research, showing positive significance in age, amount of games played and goal ratio per game [12]. Next to this, the Bundesliga, which is the highest football competition level in Germany, showed an increase in total transfer fees of football players, by a factor 10 in 20 years [19]. Moreover, another research tried to capture the importance of certain characteristics of football players on their salaries. Evidence was found that being left footed or equally good in both feet, has a significant influence on your salary [10]. Finally, the duration of the contract is found to have a significant positive effect on the salary of the players, and also its average duration is increased by half a year [24] [17]. All together, these findings do not make up for an answer, on whether or not certain transfer prices are valid. Furthermore, it contains contradictions regarding the impact of certain characteristics.

In the current literature, the data sets used are often small, having less than 200 observations. Also more often than not, research focused on one country and different competition levels, which causes problems if the variables are expected to have different impacts on different competition levels. Furthermore, performance characteristics, such as performance on the pitch, are often not taken into account. When researchers do try to capture performance characteristics, they include the amount of goals scored and assists provided. These are only attacking qualities of a player however, and are far more likely to occur for attacking players. Furthermore, this way defending qualities are neglected, as well as possessive qualities, such as interceptions, passes or crosses. Secondly goals and assists should be weighted according to the competition the player is in. Scoring a goal in a top England league is likely to be more valuable than the leagues below, which is not taken into account in these researches. Recommendations for future researches involve capturing as much infield performance characteristics per individual player as possible and to rate players based on these characteristics accordingly [19]. Furthermore, the performance between seasons might have different impact on transfer worth of players, whereas most researches only take current season statistics into account [19]. This research attempts to involve statistics, that deal with the above mentioned issues. Finally this research models towards the prediction of football players. Whereas researches so far have mainly focused on modelling towards explaining the impact of variables on the transfer price.

4 Data

This chapter discusses how the final data set is created from the different data sources. The first section introduces the different data sources. The second section explains the method to gather the data from these different sources. The final section explains the way in which the data is structured.

4.1 Data Sources

This section will explain the data sets used, how they are configured and what has been done to transform the different sets into one large data set, usable for statistical analysis.
The data comes from three different internet websites: Transfermarkt.com, Squawka.com and Euroclubindex.com [50] [49] [46]. The first being a site gathering all information on upcoming and previous transfers of football players, the second being a site which captures and monitors everything that happens during a game, such that it can summarize the statistics per team and player. Think of statistics such as shots on target, goals, assists, passes, interceptions and many more for an individual player. From now on these will be referred to as in-field statistics. The final website ranks all European football clubs on their performance, and does so each by distributing points for every match.

Transfermarkt.com has information on almost all known transfers, referring to players being swapped from one team to another. This includes transfers on loan, for free, or bought, for most countries on different professional competition levels. Furthermore, it has available for all these transfers, information regarding the player being transferred, such as nationality, date of birth and gender, we will refer to these as player statistics. Squawka.com has available many in-field statistics for individual players for many leagues and recent years. As mentioned before, this research does not consider statistics on the second professional level of a country. The statistics per league and per season that were gathered are displayed in table 1. Euroclubindex.com has available the sportive strength expressed as a value, called the Euro club index (ECI) for all European clubs since 2007. Together these form four data sets from three sources, as described in table 2.

League	**Oldest Season**	**Up to: Season**
English Premier League	2012/2013	2016/2017
Spanish La Liga	2012/2013	2016/2017
Italian Serie A	2012/2013	2016/2017
German Bundesliga	2012/2013	2016/2017
Dutch Eredivisie	2013/2014	2016/2017
French Ligue 1	2012/2013	2016/2017
Turkish Super Lig	2014/2015	2016/2017
Russian Premier League	2013/2014	2014/2015
Australian A-League	2013/2014	2016/2017
Brazilian Serie A	2013/2014	2016/2017
Portuguese Primeira Liga	2012/2013	2016/2017

Table 1: In-field statistics and Leagues

4.2 Data gathering

The data from Euroclubindex was obtained in a structured way via the company responsible for the algorithms behind the ranking: Hypercube Business Innovation [47]. Their interest in this research gave access to the set of data regarding the Euroclubindex. Unfortunately this was not the

Transfermarkt.com		Squawka.com	Euroclubindex.com
Transfer Information	Transferred Player Information	In-Field player statistics	Club Strength
$TransferSum$, $Clubsinvolved$	$Nationality$, $Dateofbirth$	$StatisticalScore$	ECI

Table 2: Different data sources

case for the other websites. The data from these websites were gathered through a written script in Python, a web-scraper, of which the code can be found in the appendix in section 10.11. In short the script does the following. The script first structures the HTML code of a given site. It then searches for certain keywords in the HTML code specified by the user and returns information that corresponds to that keyword. This way, one is able to make the script search for, example given, all links to players profiles. By specifying the right keywords to look for, the scraper is able to find the information requested. Finally, it stores these captured elements into a SQL database. This way, four tables in SQL are created containing the different data from the three sites. The next step is forming one data set out of these four SQL data tables.

4.3 Data structuring

We have four data sets containing different information. Because the goal is to predict transfer prices, we need information from the transfer data set, as well as the player-characteristic data set as well as the in-field statistics data set, as well as the ECI corresponding to the clubs to coincide. For example, we wish to know the nationality of a player being transferred, as well as the amount of goals scored in the previous season, as well as the strength of the club selling the player. To have this information we need the transfer information(data set 1), the nationality belonging to that respective player(data set 2), the goals scored in the previous season by that player(data set 3), and finally the ECI of the selling club (data set 4). To run estimations, we need players to appear in all four data sets. The first two data sets can easily be matched to one another, because players in both data sets from transfermarkt.com are given an unique ID. This ID allows us to easily find the intersection of these two data sets, forming them into one. Then we are left with 3 data sets to combine, of which not all information in either sets, correspond to information from the others. This is explained visually in figure 1. The intersection with the Euroclubindex data set is not that hard to find too, because clubs are unique and both the club and corresponding year are captured by Transfermarkt, after which we only have to match the strength of the club in that particular year.

Problems arise when trying to find the intersection with the Squawka data set, since none of the elements contained in both of the two particular data sets are guaranteed to be unique. Think of a player being named 'John Doe', having this name in data set 2 as well as data set 3 does not guarantee that its the same individual.

The final data set which is used in the analysis is therefore configured via an algorithm that checks the following criteria: First the names in the two remaining data sets must be exactly the same. Second the football position of the player has to be the same in both data sets, when looking at four different positions (Forward, Midfield, Defense, Goalkeeper).

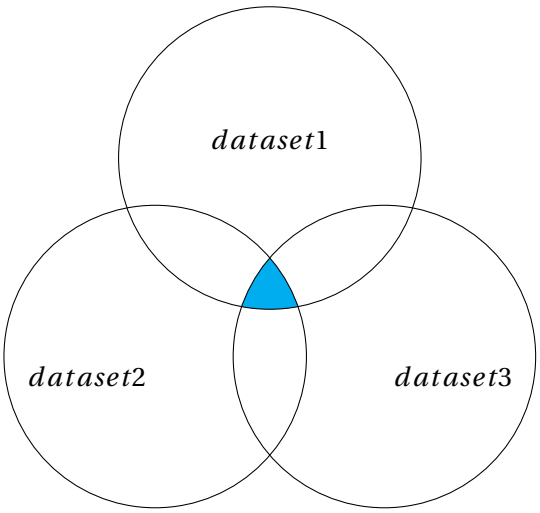


Figure 1: Intersection data sets

The third criterion checks whether or not the club at which the in-field statistics gathered (data set 3) coincided with the most previous or second most previous club owner of the player (combined data sets 1, 2 and 4). This is done to include players who gather in-field statistics while being on loan for a different club and being sold directly when returning, without playing with the team which holds their contract. The observations meeting all criteria are kept and stored, after which one final manual check was done to secure there were no errors. More information on why this final check was needed, will be explained in the introduction of the Squawka variable in chapter 5. The matching of the three data sets is thus done as depicted in figure 2:

After matching the different data sets based on their individual name, position and clubs played for, for the 11 mentioned leagues, and years 2012 and on-wards, we are left with 1737 unique transfers. The next chapter explains all variables that are given for each unique transfers. Note that even though we are left with 1737 transfers from combining the four data sets, that this does not mean that for all observations, all variables are available for econometric analysis. This is due to the fact that for some transfers, some variables captured are unknown even though the rest of the data is in tact. Examples are the amount of goals scored by a player four seasons ago, or classified information such as the amount of months that were left on the broken contract of a transferred player. This implies that we could lose observations by adding certain variables for the analyses.

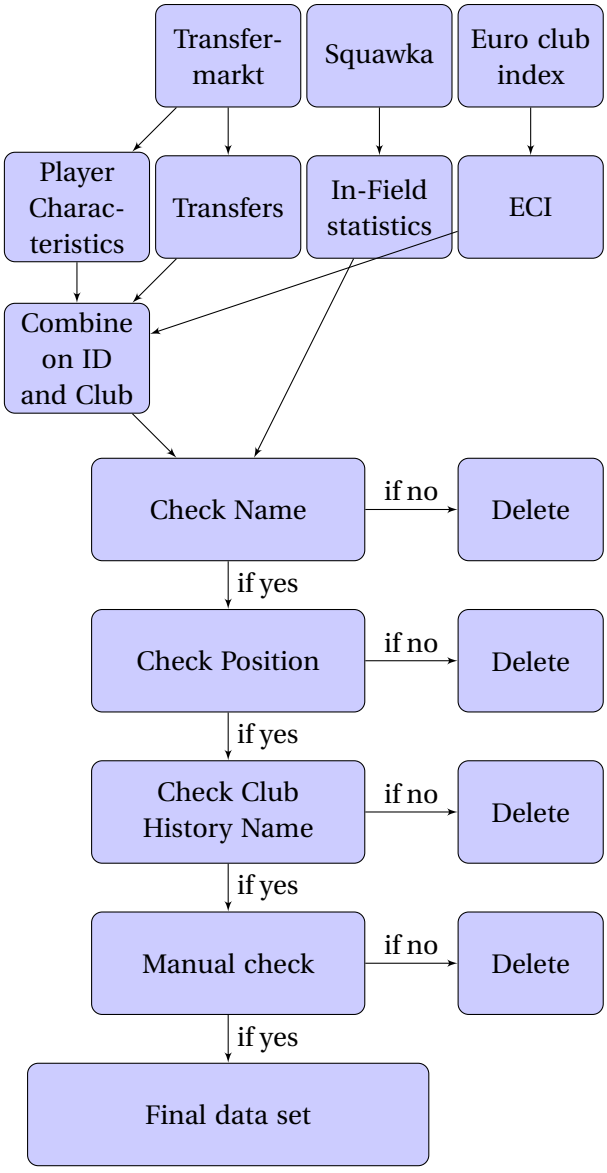


Figure 2: Combining the different data sources

5 Variables and definitions

Lots of variables are used to potentially explain the transfer price of a player. In tables 3 and 4 an overview is given, as well as brief explanations regarding these variables. More in-depth explanations regarding the Euro Club Index and Squawka Scores can be found in the up-following sections. Note that these tables do not list all the variables that are available. The reason as described in the previous chapter, is that the addition of some variables results in fewer observations. If the sample size reduction becomes significant, the variable is only included if researches in the past showed a significant relation between the particular variable and the transfer price, or if the particular variable has an intuitive relation with the transfer price. This will be discusses more thoroughly in chapter 7.

Summary statistics regarding the variables can be found in the appendix 10.1. Furthermore the correlation matrix, showing the correlation between the variables is shown in the appendix 10.1 as well.

5.1 Euro Club Index

One must include a variable which distinguishes a players worth between a top-level team and a low-level team. Otherwise, problems might occur with players having similar statistics on different competitive levels, whereas one player would sell for much more or less depending on the statistics corresponding to the level of play. Furthermore, top-level teams often have more money and are willing to go to greater lengths to buy a certain player. For these reasons we somehow need to measure the strength of the clubs involved and take these into consideration when modelling the transfer prices.

In this research the teams are scored according to the Euro Club Index (ECI) [46]. The ECI is a ranking of football teams throughout Europe.The ECI-value of a team represents the expected level of sportive success, it can be used to predict the outcome of a football match. These ECI-values are calculated based on the historical and actual sporting results, with the most recent matches having more impact on the ECI-value.

It works as follows: The result of a football match depends on the playing strength of the teams, a home advantage and the match performance of the teams. The playing strength is given by the ECI-values. The home advantage is updated each year for each league. The match performance will vary according to a normal distribution. This way it is possible to calculate the probabilities of the different match results (win, draw, loss). The ECI is calculated in such a way that its predictive force is maximized. The ECI-values will be included for both teams involved in the transfer, that way, the strength of the buying as well as the selling team will be captured.

Table 5 displays the top 10 football teams and their ECI values throughout Europe at the end of the 2016-17 season.

5.2 Squawka

One must include variables that capture the performance of players on the field. Past researches have shown the importance of statistics such as goals scored and games played on the value of a football player. However, it is expected that statistics such as chances created or successful crosses to also be of importance. Therefore the Squawka Scores are considered in this research [49]. Squawka.com captures all on ball actions and distributes three different scores for each player each match. These are the following scores:

- Squawka Attack Score (SQ_Att)

Variable	Short Name	Definition
Transfer Sum	Transfer_Sum	The realized transfer sum
Contract Duration	MonthsRemaining	The amount of months that were left on the contract of the bought player
Year of Transfer	Year	The year in which the player was bought
Age of player	Age	Corresponds to the age of the player when the transfer took place
Height of player	Height	The height of the player in cm
Preferred Foot Left	dummy_left	If the players' foot of preference is left or he is equally strong in both feet
Euro Club Index value of the buying team	ECI_New_Team	Corresponds to the score that is given according to the Euro Club Index
Euro Club Index value of the selling team	ECI_Old_Team	Corresponds to the score that is given according to the Euro Club Index
Squawka attack score	SQ_Att_t	Corresponds to the attacking score a player was given in period t
Squawka defense score	SQ_Def_t	Corresponds to the defending score a player was given in season t
Squawka possession score	SQ_Poss_t	Corresponds to the possessive score a player was given in season t
Amount of games played	SQ_Games_t	Corresponds to the amount of games played in season t
Amount of minutes played	SQ_Min_t	Corresponds to the amount of minutes a player was on the pitch in season t
Ranked in league	SQ_Rank_t	Corresponds to the rank the player has based on the sum of all three Squawka scores
Striker	strikerdummy	Equals 1 if the players' position in field corresponds to a left forward, right forward, striker, or secondary striker, 0 otherwise.

Table 3: Variables and definitions

Variable	Short Name	Definition
Midfielder	midfielddummy	Equals 1 if the players' position in field corresponds to a left-, right- or central-midfielder, 0 otherwise
Defender	defenddummy	Equals 1 if the players' position in field corresponds to a left back, right back or central defender, 0 otherwise
Summer or Winter	dummy_winter	Equals 1 if the player was bought during the winter transfer period
Buying club Dummy	dummy_club	Corresponds to the club that buys the player
Selling club Dummy	dummy_club01	Corresponds to the club that sells the player

Table 4: Variables and definitions

Rank	Team	ECI
1	Real Madrid	4427
2	FC Barcelona	4348
3	Bayern München	4053
4	Atletico Madrid	3922
5	Juventus	3905
6	Paris Saint-German	3853
7	Chelsea	3582
8	Manchester City	3495
9	Borussia Dortmund	3482
10	Napoli	3459

Table 5: Top 10 Football teams in May 2017 according to Euro Club Index

- Squawka Defense Score (SQ_Def)
- Squawka Possession Score (SQ_Poss)

The scores indicate the ability of the respective player to positively influence a game of football. The higher the score, the more positive influence this player has had on the ball. The scores are calculated by taking every recorded on-ball action while playing and evaluating the outcome.

The attack score considers all events leading up to chances or goals, these being shots, crosses, take-ons, goals, assists, key-passes and chances created. The defense score takes all events into consideration that potentially stop the opponent from scoring a goal, these being tackles, interceptions, blocks, clearances, fouls, cards and saves. The possession score takes the passes, and through balls in consideration. Each component of each score has a base score, this score is then multiplied by an amount depending on the success of the event, the area in which the event takes place, the position of the player involved and how the event is executed (left footed, right footed, header, volley). After each game the scores are summed up per player per club, representing the scores of a player. This is done until the season reaches an end, after which all players start at 0 again for their respective club. In this research two moments per season are taken into account for when the scores are gathered. One just before the winter transfer period opens, and after the last played game of that first season half. The second is just before the summer transfer period opens, and after the last played game of the season. Note that therefore the summer score contains the score the player had during winter break, regarding he did not make a transfer during the winter. The index t in SQ_Att_t indicates the season corresponding to a score. Zero being the score in the summer corresponding to the season of the transfer, one being the score during the winter of the season of the transfer, two being the score during the summer of the second-most recent season, three being the score during the winter of the second-most recent season, and so on wards.

Note that if a player transfers during a season to a different club, he starts with a score of 0 with the new club, because scores are kept per player per club. This is important to realize, because of this fact there could potentially be multiple scores per player for the same date, because of a transfer. An example would be a player transferring during the winter break, as well as during the summer the up-following transfer period. This would mean this player has a winter Squawka score for the buying club in the winter, SQ_Att_1, and two SQ_Att_0 scores for the buying club in the summer, one equal to SQ_Att_1, the old club from which he transferred, and one new score playing the second season half for the new club. This issue was dealt with by manual checking the players having double scores per date and deleting the score that was not relevant for the transfer.

6 Econometric techniques

Modelling, evaluating and predicting the transfer prices, require econometric methods to formally do so. This chapter will define and introduce three model classes used in this research: Least squares estimation, ridge regression and the lasso. Furthermore, it will introduce the knowledge to construct and interpret the models.

6.1 Defining the data

We are interested in predicting transfer prices. We must therefore find a suitable way to model transfer prices in terms of the available data. Simply said, we must study how y varies with changes of x, where y corresponds to the transfer price and x corresponds to variables of interest [41]. This section will introduce the basic definitions needed in the remaining sections, which discuss different approaches into studying the relationship between y and x.

The vector of transfer prices will be denoted by y. We will denote each variable by x_i, where subscript i represents the ith variable. The jth element of the vector y and x_i will be denoted by y_j and x_{ij} respectively. The amount of variables considered is denoted by P and the amount of observations denoted by N. β_i corresponds to the true effect of x_i on Y. A hat defines an estimate. Therefore $\widehat{\beta}_i$ is our estimate of the true value of β_i. Estimates regarding the transfer price will be denoted by $\widehat{y}_j$, which thus corresponds to the estimate of the true value y_j. We are considering linear regression models in this research, and therefore assume the following equation to hold [21]:

$$y = \beta_0 + \sum_{i=1}^{p} x_i \cdot \beta_i + \epsilon, \tag{1}$$

in which β_0 is the intercept and ϵ is a Gaussian random variable with mean zero and variance σ^2:

$$\epsilon \sim N(0, \sigma^2). \tag{2}$$

The parameter of interest is β_i, which describes the direction and strength of the relationship between y and x_i [41]. The input variables x_i come in different forms [21]:

- Quantitative forms
- Transformed Quantitative forms
- Zero-one or dummy forms
- Interaction forms

Quantitative data can be transformed when necessary or when it improves the predictive accuracy, often these transformations are logarithmic, square or square root transformations [41]. Zero-one or dummy variables are defined as a variable that represent one if a certain criteria is met, and zero otherwise. An example could be a variable that equals 1 if the individual observed is a man and 0 if not. Finally, interaction forms come into play when there is a belief that the effect of x_i and x_k together are also of importance, such that we include $x_i \cdot x_k$ as an additional variable. More about the use of these and why are discussed in chapter 7.

After having defined the various types of data, we must now decide on which estimation procedure to follow, in order to model the transfer prices of football players.

6.2 Least Squares

In this section we discuss a popular method for linear models: Least Squares. Briefly stated, given data on y and on x, least squares tries to fit x in such a way, that minimizes the sum of squared errors of its prediction to its real value. It defines the fitted value for y via the model [41]:

$$\widehat{y}_j = \widehat{\beta}_0 + \sum_{i=1}^{p} x_i \cdot \widehat{\beta}_i, \tag{3}$$

In order to find the appropriate values for β, we define the residual sum of squares (RSS):

$$RSS = \sum_{j=1}^{n} (y_j - \widehat{y}_j)^2, \tag{4}$$

which we can rewrite via (3) to:

$$RSS = \sum_{j=1}^{n} (y_j - \widehat{\beta}_0 - \sum_{i=1}^{p} x_{ij}\widehat{\beta}_i)^2. \tag{5}$$

Formally, we wish to minimize the RSS and do so by characterizing the solutions β_0 and β_i to the minimization problem: [41]:

$$argmin_b \sum_{j=1}^{N} (y_j - b_0 - \sum_{i=1}^{p} x_{ij} b_i)^2, \tag{6}$$

where b_0 and b_i are the arguments for the minimization problem, resulting in our Least Squares estimate $\widehat{\beta}^{LS}$. To solve equation (6), linear independence is necessary, which will be introduced in the next section, as well as several other properties and assumptions regarding the least squares estimation.

6.3 Properties and Assumptions

In this section, five assumptions, and the statistical properties which least squares exhibits when these assumptions hold, are discussed. The assumptions are as follows [41]:

- Linear in Parameters
- Random Sampling
- No Perfect Collinearity
- Zero Conditional Mean
- Homoskedasticity

The first assumption states that the model is linear in its parameters β_i. It is reasonably flexible, because we can transform y and the x_i such that this assumption holds [41]. Transformations regarding the data will be the topic of discussion in chapter 7. The second assumptions states that the sample we are modelling is random and representative for the true population [41]. In this research, this requires that the data regarding transfer prices follows the true distribution. This will be assumed throughout this report and discussed in the final chapter. The third assumption states that none of the independent variables is constant, as well as that there exist no perfect linear relationships between any of the independent variables [41]. This latter is referred to as linear independence. Instances such as: $x_1 = 3 \cdot x_2$, or $x_1 = 0.5 \cdot x_2 + x_3$, are therefore linear dependent. Linear dependency issues are present in this research and are dealt with accordingly in chapter 7. The fourth assumption

states that the error ϵ has an expected value equal to zero given any of the independent variables, formally [41]:

$$\mathbb{E}[\epsilon|x_i] = 0. \tag{7}$$

The fourth assumption will be discussed in the final chapter. These four assumptions together make the least squares estimate unbiased [41]. The fifth assumption states that the error ϵ has the same variance for any of the independent variables, formally [41]:

$$Var[\epsilon|x_i] = \sigma^2. \tag{8}$$

If the latter assumptions fails, the model exhibits heteroskedasticity. A formal test will be performed to test for heteroskedasticity and dealt with accordingly. These five assumptions make that the least squares estimate are the best linear unbiased estimator [41].

6.4 Stepwise selection

Although least squares enables us to form a basic model for the relationship between the transfer prices and explanatory variables, it does not answer the question about which variables should be included in the model. Often the amount of variables or predictors to choose from is very large, and we would like to determine a smaller subset of variables with the strongest explanatory power [21]. There are two main reasons why we do not want to include all variables in a least squares estimation. The first regards interpretation, because a large amount of variables can distort and complicate interpretation. The second is the danger of overfitting; the more variables, the more likely we will find an effect by randomness [21]. We need a method to select the variables for prediction, and then estimate these via Least Squares. There are a number of different strategies in picking the subset of variables, one of which is best-subset selection.

Best-subset selection finds for each possible subset size of variables the best model and performs least squares on this subset. The large drawback to this approach is that as soon as we consider many variables, this becomes infeasible, since there are 2^p subsets to consider. This will not be doable with the amount of variables considered in this research, which brings us to the core of this section: Stepwise selection. Rather than search through all possible subsets, we wish to find an appropriate way to select some of them. Methods that do so are forward stepwise selection and backward stepwise selection. The difference is that forward stepwise selection starts with a model with an intercept, i.e. the null model with no variables, after which it repeatedly adds variables to the model that best improve the fit, whereas backward stepwise selection, starts with all variables, and repeatedly removes variables that least impact the fit [21]. The performance of both methods is very similar, and therefore we only implement forward stepwise selection in this research [21].

The idea is that we start with a model containing only an intercept, and then add variables sequentially to the model. The order in which the variables are added depend on which give the best improvement to the model, where the measure for best improvement is prespecified [21]. After each addition, the procedure re-evaluates the new model before adding the next best variable to the model. This is repeated until the addition of a variable no longer satisfies some predetermined criterion. Different criteria may be used to force the algorithm to stop at a certain point. One could think of having a certain p-value the additional regressor must satisfy. This research however, does not consider any stopping criteria and forces the model to repeat its procedure until all steps are finished, the reasoning is explained after discussing the model fit criteria.

In stepwise selection procedures, the criteria of model fit can be one of several fitting measures, the one considered in this research is: The residual sum of squares. Recall that the RSS is simply a measure of the discrepancy between the model and the output data. For its formal definition we refer back to (4). Other popular criteria for model fitting are the Akaike Information Criterion and the Bayesian Information Criterion. Both methods penalize addition of variables in a different way and

its interesting to see which performs better with prediction purposes. However, both methods rely on having a sufficiently large sample size [29]. The sufficiency of the sample size is deemed by the complexity of the underlying data generating process. With the sample size at hand and corresponding variables, its likely that this is not the case. Next, cross validation well approximates the true out of sample predictive performance, even when dealing with a relatively small sample size [29]. This research therefore incorporates cross validation, to validate each step in the forward selection procedure. Thus, the selection procedure chooses the variables that minimized the RSS in each step. Later on, all steps are evaluated and the step with the lowest out of sample prediction error is chosen. This is also the reason why this research does not consider a stopping criterion, forcing the algorithm to eventually pick all variables, does not mean that the final model will be picked. It does however, allow a comparison of out-of-sample prediction performance, with every step in the procedure.

We note that this by no means guarantees to return the best subset, because the variables chosen in previous steps, might no longer be best when considering a combination of other variables. This does not mean however that it is inferior to the best-subset selection procedure. Besides having computational benefits, such that is applicable even for large p, it is also a more constrained search, resulting in a lower variance and (possibly) a larger bias, which as explained before, might be preferred [21]. Other model fitting measures, such as the R-squared and adjusted R-squared will be mentioned further on in this report and therefore defined below.

The R-squared is a number that indicates what proportion of the variance in the dependent variable is explained by the explanatory variables. The coefficient therefore ranges from 0 to 1, or 0 to 100%. It is closely related to the previously defined RSS, namely as follows:

$$R^2 = 1 - \frac{RSS}{TSS}, \tag{9}$$

where TSS represents the Total sum of squares and is defined as:

$$TSS = \sum_{j=1}^{n} (y_j - \overline{y})^2, \tag{10}$$

minimizing the RSS therefore yields the same results as maximizing the R-squared. The potential flaw of the R-squared is that it can increase by addition of a variable by randomness, even if the added variable has no explanatory power on the dependent variable. The adjusted R-squared tries to take this into account by only increasing when the addition of a variable causes a larger increase in R-squared, than one would expect by pure chance. For this reason the adjusted R-squared can be negative and is always lower than or equal to the R-squared value.

After the stepwise selection procedures are done, least squares estimation is performed on the remaining set of variables.

6.5 Ridge Regression

One of the mentioned properties of the least squares estimator is that it is the best linear unbiased estimator, under the right assumptions. We will explain in this section why this is not necessarily optimal for prediction.

Consider the accuracy of an estimator following from a model $\widehat{\theta}$ compared the true value θ. This accuracy is often measured by means of the Mean Squared Error (MSE) [21]:

$$MSE(\widehat{\theta}) = \mathbb{E}[(\widehat{\theta} - \theta)^2], \tag{11}$$

with some computational effort this can be rewritten to:

$$MSE(\widehat{\theta}) = Variance(\widehat{\theta}) + Bias(\widehat{\theta},\theta)^2. \tag{12}$$

We can see that the error depends on the bias as well as the variance. Choosing an estimator with no bias, such as least squares, does therefore not necessarily have the smallest MSE, there could be biased estimators with smaller MSE and better prediction accuracy. All steps regarding how equation 13 can be rewritten to equation 14, can be found in the appendix section 10.3.

The models we use optimize the parameters to fit the data as well as possible. If we then use these parameters on an independent sample, it generally fits much worse. This is the problem of overfitting. So when the goal is to predict out of sample data, we need to pick the right balance between variance and bias of our estimators [21]. This brings us to the ridge regression model. The remainder of this section will define the ridge regression model, why it could be beneficial, and in which cases the model is likely to perform well.

The ridge model imposes a penalty on the size of the estimates. Depending on the shrinkage parameter λ, the coefficients are shrunk to zero, and therefore reduce the variance of the estimates. Note that this creates a bias in the estimates [21]. As just discussed, this could potentially lead to better prediction results. The coefficients for $\widehat{\beta}^{ridge}$ are characterized to the minimization problem [21]:

$$argmin_b \sum_{j=1}^{n} (y_j - b_0 - \sum_{i=1}^{p} x_{ij} b_i)^2 + \lambda \sum_{i=1}^{p} b_i^2, \tag{13}$$

in which b_0 and b_i represent the arguments for the minimization problem again, and in which $\lambda \geq 0$, and represents the parameter that controls for the penalizing of the coefficients: the larger its value, the higher the penalty and therefore the more shrinkage of the coefficients. Note that when we set λ equal to zero, we simply get the least squares estimate defined in (6). The value of λ must be picked in such a way that it minimizes an estimate of the expected prediction error [21]. Discussions related to the value of λ and how to estimate it will be done in chapter 7.

Ridge is generally used when their are many variables to consider, as well many being correlated. Including such highly correlated variables causes the coefficients to exhibit high variance. A potential coefficient on one variable can be canceled out by a number of negatively correlated variables with positive coefficients as well. Ridge solves this potential problem via the penalization, allowing the model to shrink the coefficients [21].

Ridge typically selects all variables, which might not be preferred if a large set of variables have no true effect on the dependent variable [21]. For this reason, another method belonging to the class of penalized regressions is introduced in the next section.

6.6 Lasso

Least Absolute Shrinkage and Selection Operator (lasso), will be discussed in this section. The lasso method, just like in ridge regression, penalizes and thus shrinks the coefficients. While conceptually ridge regression and the lasso are both regularization techniques, they use different penalty functions, causing the methods to return very different results. Its estimate $\widehat{\beta}^{Lasso}$ is characterized as follows [21]:

$$argmin_b \sum_{j=1}^{n} (y_j - b_0 - \sum_{i=1}^{p} x_{ij} b_i)^2 + \lambda \sum_{i=1}^{p} |b_i|, \tag{14}$$

in which b_0 and b_i represent the arguments for minimization problem again, and in which λ satisfies $\lambda \geq 0$. Note the main difference between the ridge regression and the lasso models. The ridge regression penalty term: $\sum_{i=1}^{p} \beta_i^2$ is replaced by the lasso penalty term $\sum_{i=1}^{p} |\beta_i|$. Ridge regression limits the size of the coefficient vector, while the lasso also introduces sparsity in the estimates. That is, with the lasso we will have many β_j for $j \in \{1, \ldots, p\}$ equal to 0 [21]. Note that such sparsity is useful, because it performs variable selection and therefore increases interpretation of the remaining variables. The lasso tends to perform worse when dealing with a large amount of highly correlated variables, in which case Ridge usually performs better as mentioned in the previous section. However, because the lasso also performs variable selection, it may perform better with variables having little to no correlation with the dependent variable [21].

Note that even though the lasso does variable selection, its selected variables are typically biased [21]. In order to get a better interpretation regarding the variables selected by the lasso, one can perform Post-lasso estimation. A least squares estimation is performed on the variables with non-zero coefficients from the lasso estimation. Empirical results show that this technique results in a smaller coefficient bias, and a quicker convergence rate towards the true parameter values [5] [2]. Note that this procedure is introduced to increase interpretation, not to increase prediction performance, as the whole reason for the lasso regression to return biased estimates, is to reduce variance and increase prediction performance. Therefore, for interpretation purposes, a post-lasso estimation will be performed as well.

6.7 Model selection

Now we mentioned before that the Gauss-Markov theorem states that the unbiased coefficients given via the Least Squares method, are best out of all unbiased linear estimators, under the right circumstances. We have also mentioned why it could be wise to choose an estimator that is not necessarily unbiased. There may however, very well exist biased estimators with smaller variance, and better prediction performance. By constraining and shrinking coefficients, the variance of the estimates can be reduced at the cost of a small increase in bias, which sometimes leads to overall improved prediction accuracy. The second reason is the interpretation of our model. Having a large number of predictors, we would like to have a smaller subset that explain the strongest effects, giving a clear insight on the explanatory variables. Some of the smaller details would have to be sacrificed [21].

The models considered for prediction are:

- Forward stepwise selection
- The lasso
- Ridge regression

For each of these methods, arguments can be given for why this method would be best for prediction. Forward stepwise selection benefits from the use of least squares estimation, but cannot pick more than n covariates and does not guarantee the best subset. Another risk of this method, is that the addition of a variable does not necessarily improve the predictive power of the model, it just improves the RSS. For this reason, each step taken by the algorithm is saved and its predictive power is measured. This cannot be done on a set containing the same observations, as it must be independent from the data used to construct the model. This means for the forward stepwise selection, that we evaluate the performance of the model in each step, on a sample of the data not used for modelling, called the validation set. Afterwards, we can pick the model corresponding to the step with the best prediction performance. If this model is different from the model in the last step of the stepwise selection, we will have picked a model with a smaller R-squared, but likely to have better predictive power. Note that the prediction error of the validation set, which determines how many variables to

include, depends on the observations in the validation set. Furthermore, we must also devise a measure of prediction performance, such that we can compare the different models. How to deal with these issues will be discusses in section 8.

Ridge regression is a shrinkage method that shrinks the coefficients by applying a penalty on their size. It is a useful method for prediction when one is dealing with lots of variables to choose from and many of them exhibit high correlations. The lasso is a shrinkage method which potentially sets some coefficients to zero by forcing the total sum of the absolute values of these coefficients, to be less than a specified value λ. This way the lasso effectively chooses a simpler model.

Figure 3 shows this difference between the Lasso and the Ridge. The Lasso considers the absolute coefficient value, and is therefore inclined to set variables to zero and thus perform variable selection. The Ridge squares the errors, and therefore prefers multiple variables to shrink, instead of choosing one and setting the other to zero. Suppose you have the following output y and predictors x_1 and x_2, with the following relationship: $y = x_1$. Also, x_1 and x_2 are almost perfectly correlated, such that y could be predicted with x_2 as well. The penalty in the lasso will be $|\beta_1| + |\beta_2|$, whereas the penalty in ridge regression will be $\beta_1^2 + \beta_2^2$. The lasso will be indifferent between choosing any of the following estimates for y, $0.2 \cdot x_1 + 0.8 \cdot x_2$, $0.5 \cdot x_1 + 0.5 \cdot x_2$ and $0 \cdot x_1 + 1 \cdot x_2$, all the penalties will equal 1. For Ridge its different, these predictors will be penalized as: 0.68, 0.50 and 1 respectively. This is why ridge regression tends to have multiple predictors shrink towards each other and the lasso does not [21]. This is one reason why ridge regression perhaps performs better with many co-linear predictors: If highly correlated variables give little information to choose between linear combinations of these predictors, the lasso almost randomly picks one, whereas ridge regression chooses the one with the most equal weights [22]. Equal weights might be of more use in out of sample data, where less information goes lost. This is very much true with the variables we are considering as predictors, if we recall the correlation matrix in the appendix section 10.1, we see many highly correlated, and therefore a strong motivation to use ridge regression. The lasso has the benefit of variable selection, because of the large amount of variables we consider, a model with less variables is much more interpretable and perhaps better in predicting than a model with many variables, giving reason why the lasso might be preferred. These two characteristics of the ridge regression and the lasso make up for interesting results considering the data set we have available.

The outcomes of both the ridge regression and the lasso are highly dependent upon the value of λ, it is therefore important that this value is chosen with care. Furthermore, the ridge regression and the prediction performance are like the stepwise selection procedure vulnerable to the randomness of the training and test set chosen. One has to deal with this randomness, which will be the topic of discussion in the next section.

6.8 Prediction Performance

This section first introduces the method of k-fold cross validation, after which the measures to define the prediction performance will be introduced.

The goal is that we fit our model in such a way, that we are able to predict as precisely as possible independent data regarding transfer prices. Or in a more practical setting, choose the model that has the lowest error when potentially predicting future transfer prices of football players. To achieve this goal, one must test the prediction performance of the different models, as if they predict independent data from which they were build upon. There are many techniques to do this with, the more popular ones are various bootstrapping (BS) techniques and various cross validation (CV) procedures. The latter one is what will be used in this research. Both techniques are great to avoid overfitting problems, which as previously mentioned, may hurt prediction performance.

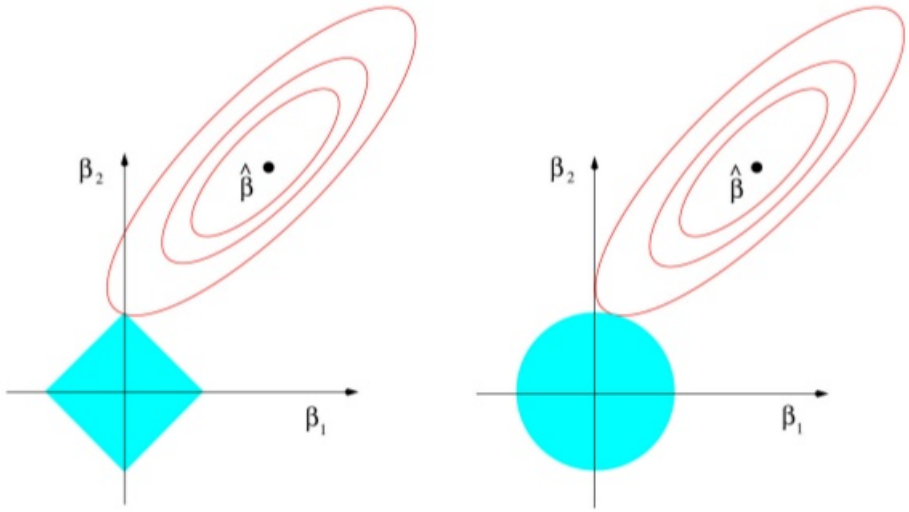


Figure 3: The lasso(left) and the ridge regression(right), constraint area's in blue, and the least squares estimate and its corresponding error function in red.

The choice on which procedure to use comes mainly down to a variance and bias trade off. In general cross validation tends to have smaller bias but somewhat high variance in its prediction errors, whereas bootstrapping reduces the variance but tends to have a bias [26] [15]. Moreover, leave one-out cross validation, while being near unbiased, exhibits such high variance that k-fold CV is the more superior choice [8] [20]. Furthermore, evidence was found that the .632+ BS method suffers from a bias problem regardless of the sample size used and that repeated k-fold CV shows stable performance and is preferred [25]. This research therefore considers the k-fold CV procedure [32].

The k-fold CV procedure works as follows: First, the data set is divided into k non-overlapping folds. For a single fold, the remaining k-1 folds are used to train the model, called the training set, and tested on the single fold, called the testing set. This is repeated for all k folds. The error corresponds to the average prediction error the model has on each single fold. This way, all observations will be part of a test set once.

There are now a few things to consider, before implementing the k-fold CV procedure. How many folds do we choose (the choice of k), and do we use repeated or non-repeated CV. In repeated CV, we do multiple k-fold CV's in which we randomly choose the k folds each time. The repeated CV therefore has worse computation time. However, the repeated CV procedure outperforms the non-repeated one and shows much less variance [29]. Furthermore, tests of k-fold CV, in comparison with various BS techniques, revealed that the heavier computation time required for repeated k-fold CV is worth carrying out [25].

Then it comes down to how many folds one chooses when performing k-fold CV. Again, this comes down to bias and variance trade off. For repeated k-fold CV lower choices of k results in higher bias but lower variance and vice versa [29]. However, the variance is already partly reduced after performing repeated CV and if the aim is to minimize prediction error k=10 outperforms smaller values such as k=5 and k=2 [29] [32]. This research therefore continues with a ten-fold cross validation procedure to measure its prediction errors.

In the stepwise selection procedure, this allows for different picks regarding the number of variables, because the observations in the sets change for each fold, and therefore does the model. This reduces some of the randomness in relying on a certain amount of variables, because all observations are represented in the training set as well as the test set in one of the folds. Considering a single

division of training and testing sets, this is not the case. Theoretically, the validation set, which is constructed from the training set, could be the same for different folds, because a sub sample is taken from the k-1 folds. This issue can be resolved by applying k-fold CV once more and creating k-folds within the training set. This however, worsens computation time, because for each fold in the first k-fold CV, k folds have to be created and used in the second k-fold CV. This research therefore takes a random validation set for each fold, and takes the randomness in the validation set for granted. The remaining concern is to split the training set into two divisions, of which one the validation set. The larger the validation set, and therefore the smaller the remaining training sample, increases the variance of the parameter estimates. However, with a smaller validation set, the performance measure will have greater variance. There are two competing concerns: with less training data, your parameter estimates have greater variance. With less testing data, your performance statistic will have greater variance [18]. Furthermore, there is no obvious choice yet, and most researches therefore choose an arbitrary division [30]. In practice, the percentage for the validation set is often chosen as 10% or 20%, depending on the amount of observations [30]. This research chooses 20%, to allow for less variance in the performance measure, while expecting to maintain proper parameter estimations. The validation set therefore consists of 20% of the observation, in the training set consisting of 80% of the observations, giving the validation set 16% of the total observations.

For the ridge regression and the lasso, the k-fold CV does not only allow for a more precise measure of the errors of these models, it also allows for a more rigorous choice for the values of λ. By performing a ten-fold cross validation, the training set is divided in ten parts (folds), of which every combination of nine parts is used to predict the corresponding remaining tenth part. This is done for all combinations, for a grid of λ values ranging from 0.01 to 10^{10}. We then consider two values of λ. The first λ yields the smallest mean squared prediction error. The second λ picked is the one which yields the largest mean squared prediction error, within one standard deviation of the smallest. The argument for the first pick is that we would like to have precise predictions and therefore a λ that has the smallest prediction error in the cross validation would be a suitable pick. The second λ does not necessarily yield the lowest prediction error, but is included for the following reason: using the maximum value of λ that lies within one standard deviation of the minimum, will result in a model with less variables [27]. This is due to the fact that the λ is responsible for the shrinkage and possibly elimination of coefficients, therefore the larger the λ, the more shrinkage or elimination. Even though it returns a model with less prediction power, its relative prediction power cannot be distinguished from the minimum λ in terms of prediction error, given the uncertainty of the 10 fold cross validation estimates of the errors [27]. This "one-standard-error" rule, is often applied when selection the best model [21].

The minimum λ could be slightly overfitted, but returns the best error in the cross validation. The maximum λ within one standard deviation returns slightly worse error, but the model is simpler and still comparable with the model resulting from picking the minimum λ.
Figure 4 and 5 show the results of a ten-fold cross-validation and the corresponding λ values for both ridge and the lasso. Note that the dashed lines represent the λ with the minimum error and the λ with the maximum error within one standard deviation from the minimum.

Note that the top numbers represent the amount of variables that are picked by the estimation, which also shows the selective power discussed before by the lasso.

Next, we must define a measure for prediction performance. Recall that we split our sample into a training set and a test set, the training set is to estimate the coefficients, the test set to test whether or not our model is able to predict out of sample values. By comparing the error between the true values in the test set and the predicted values, we can compare the models. To do so, we make use of ten-fold cross-validation as introduced just now.

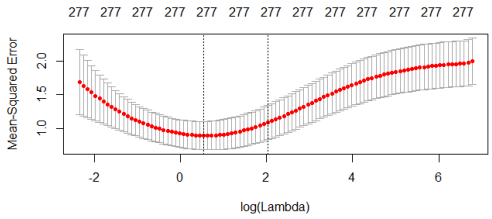


Figure 4: Lambda's in ten-fold cross validation by Ridge

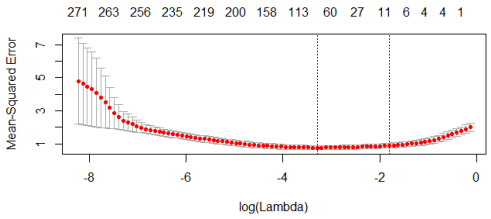


Figure 5: Lambda's in ten-fold cross validation by lasso

Once we gather all information regarding the prediction errors per model, we must decide upon an error-measure. There are many methods to distinguish from, here are a few:

- Mean Squared Error (MSE).
- Root Mean Squared Error (RMSE).
- Mean Absolute Error (MAE).
- Mean Percentage Error.

Having the following mathematical definitions:

$$MSE = \frac{1}{n} \cdot \sum_{j=1}^{n} (\widehat{y}_j - y_j)^2. \tag{15}$$

$$RMSE = \sqrt{\frac{1}{n} \cdot \sum_{j=1}^{n} (\widehat{y}_j - y_j)^2}. \tag{16}$$

$$MAE = \frac{1}{n} \cdot \sum_{j=1}^{n} \left| \widehat{y}_j - y_j \right|. \tag{17}$$

$$MPE = \frac{100\%}{n} \cdot \sum_{j=1}^{n} \frac{y_j - \widehat{y}_j}{y_j}. \tag{18}$$

Since our models consist of a log transformed dependent variable, we are considering percentage changes and therefore relative errors. This makes the use of the MPE redundant. Furthermore, an error of 1 million on a transfer price of 10 million is the same as an error of 10 million on a transfer price of 100 million. The difference between RMSE and MSE with MAE is that RMSE and MSE square the errors before they are averaged, for this reason a larger weight is given to errors that are relatively far off. Making this measure more useful if larger errors are undesirable compared to having multiple small errors. On the other hand, MAE can be useful if one is worried about the effect of a few large mispredictions on the evaluation of the model.

In this research we focus on the MSE as well as the MAE, while we prefer a model that has multiple small errors to a model with a few large errors, we still want to distinguish if a certain model might be harmed from predicting outliers badly. If a certain model performs very well in MAE terms, but badly in MSE terms, we can conclude that this is due a small amount of very large errors on the predicted transfers.

7 Finalizing the method

In this chapter we will discuss the problems regarding heterogeneity in modelling the transfer prices. Econometric theory often relies on the assumption that the statistical properties of variables in the data set are the same for all observations of this variable. In the following sections we will discuss why this is likely to be not the case and we consider player heterogeneity as well as club heterogeneity. The remainder of the chapter discusses final adjustments to the data set and its variables, in an attempt to improve the accuracy of the model.

7.1 Player heterogeneity

When modelling the transfer prices of players, one wants to distinguish between players having similar physical attributes, similar positions and similar clubs. This is because one player might be much better than the other, which could be hard to measure considering the variables above. Difficult exogenous variables such as motivation, skill and talent may largely influence the players' performance on field and therefore have an impact on the transfer price of this player. Measuring precisely someones talent or motivation is difficult if not impossible. However, one still needs to distinguish between players performing better, while having the same remainder of attributes. For this reason this research includes the Squawka scores.

The Squawka scores show the performance of a player on the field, which we believe is largely caused by its talent, skill and motivation, after accounting for physical attributes and club strength. If a player with similar attributes as one other, scores much more frequently, this will be observed via Squawka yielding a higher score. This way, we are able to distinguish between players having similar attributes and playing for a similar club.

Now when comparing players with similar attributes playing for different clubs. It could be the case that one player scores more goals than the other player, because the league competition is much worse, or because its club is much better and therefore more goals are scored. For this reason we also include the rank of the Squawka score achieved compared to the rest of the league. This way we can distinguish whether or not this player performed very well within the league in which the player performed. Furthermore, as mentioned before, we include the sportive strengths of the clubs, which indicates the strength of the rest of the team and therefore potential help from teammates. This is done via the ECI ranking as defined in Chapter 5. Finally, we include an interaction effect of the Squawka scores with the level of play of the club (ECI). This allows for a measurement of the score achieved while playing for a team with a certain strength.

Note that these do not necessarily capture the above described situations fully, it simply gives more information regarding the Squawka ratings scored. Furthermore, note that the ECI rating serves multiple purposes, as the ECI is also used to capture some of the club heterogeneity, as will be explained in the upcoming section. The player heterogeneity not captured by the above included variables will be accepted as random effects.

7.2 Club heterogeneity

The modelling of transfer prices needs to account for the differences between clubs. The revenue differences between clubs, as well as their historical and current successes and finally its brand identity, could all largely influence the transfer price. Arguments for price fluctuations caused by these are e.g. top-players in their current club wanting to go to a better club, better clubs often having more money and are therefore able to withstand certain offers or bid higher themselves. Including the revenue of clubs could potentially capture some of these effects, but these are unavailable or difficult to obtain for many clubs. Therefore in this research the ECI, as defined in chapter 5, is introduced

as a way of capturing these effects on the transfer price. The ECI measures the sportive strength and therefore gives a good indication of recent successes. Now because, clubs which perform better receive more money, it is expected that the strength of a club also capture some of its wealth. It is however hard to measure the brand identity of a club via this variable. With brand identity being the imago of the club, whether it be via historical achievements, successes or other factors. Having a positive imago might give the club more bargaining power, furthermore, it could be a reason for the player wanting to go to a club with a positive imago. For these reasons involving variables to capture these effects might be useful. In this research we try and capture these effects, if any, by including binary variables (dummies) for each club. We include two more variables in each transfer, corresponding to a dummy for the selling club, and a dummy for the buying club. Note that the binary variables do not only capture the potential effects of the imago of a club, they capture all other fluctuations as well. The question then remains on which club dummies to include and which not, choosing the relevant dummies will be discussed in the next chapter.

Another possible method to deal with the club heterogeneity, is to demean the variables for each club. What this does is, it subtracts the average value of each variable corresponding to the club. This way, the remaining values of the variables will be centered around zero for each club. This method imposes however, that the variation caused by different clubs does not vary over time, which might not be true. Furthermore, this method does not grant the ability to measure the effect of each club, whereas involving a club dummy allows for this. For these reasons this method is not chosen as a way of dealing with the club heterogeneity.

7.3 Additional Variables

Besides adding the variables separately into the regression formula, it could be useful to include interaction effects, as briefly stated in chapter 6. These are useful when we believe there is an explanatory effect of the variables simultaneously. Think of it this way: being taller might increase or decrease the transfer price, but what if the effect is different per position of a football player? Goalkeepers tend to be larger players, a positive influence of height in this position is expected. Figure 6 to 9 show scatter plots of the log transformed transfer price and the corresponding height for each position. From the figures it is not immediately obvious whether or not there could be effects of heights specific to a certain position. It seems that defenders and goalkeepers tend to be larger than strikers and midfielders and perhaps a positive effect of height on these positions exists. By including the new variables: $Height \cdot dummy\_position$ where position $\in Striker, Midfielder, Defender, Goalkeeper$. The models can select these variables if needed. Another potentially interesting interaction term would be the that of the Squawka Scores and the position. Perhaps scoring a goal as a striker has a different effect on your transfer price then it would have if your position on the field was elsewhere. Thirdly, the importance of player age. It could be the case that aging makes a player be worth more up to a certain age, say 20, after which age makes your transfer price drop. This might be due to physical growth and experience in the first place, whereafter talent and remaining years take over. Including the square of the age of the player is able to capture these possible effects.

Finally, as mentioned in the first section of this chapter, the Squawka scores are multiplied by the level of play of the club in which they were gathered. This is to emphasize on the ability of the player to perform, while playing on a certain skill-level. All interactive variables that are included are defined in table 6.

Together these make up for an additional 48 variables. Recall from chapter 6.3 and assumption 3, we require linear independent variables, i.e. that a linear combination of variables is not equal to another variable. For this reason, we cannot include all dummy variables for all positions. Because we would get:

$$StrikerDummy = 1 - (MidfieldDummy + DefendDummy + GoalkeepDummy), \tag{19}$$

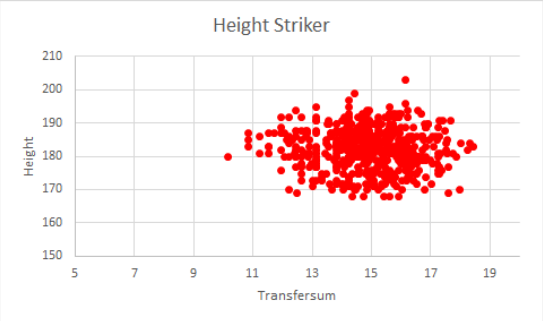


Figure 6: Height Striker

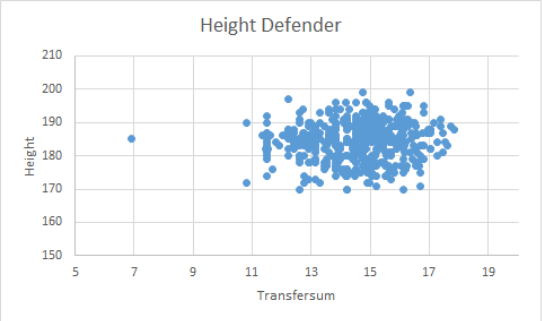


Figure 7: Height Defender

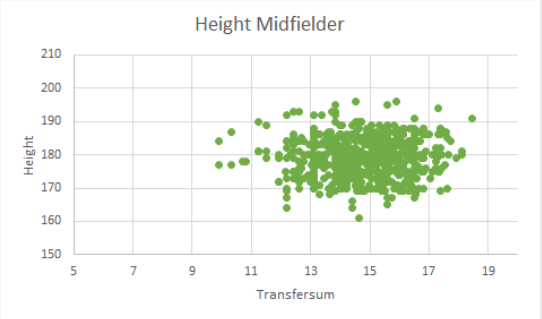


Figure 8: Height Midfielder

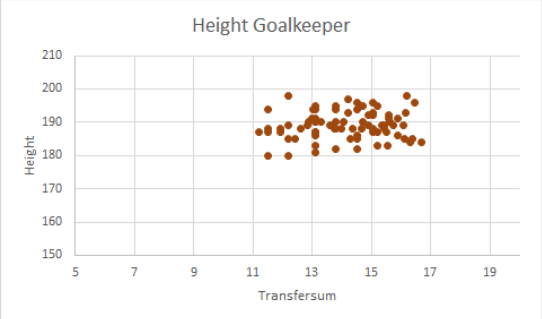


Figure 9: Height Goalkeeper

Variable	**Analysis Name**	**Definition**
Height · position	Height_position	The simultaneous influence of height and a certain position on transfer prices
Squawka · position	SQ_Att_position_t SQ_Def_position_t SQ_Poss_position_t	The simultaneous influence of the Squawka score and a certain position on transfer prices
Squawka / SQ_Games_t	SQ_Att_games_t SQ_Def_games_t SQ_Poss_games_t	The influence of the Squawka score per game on transfer prices
Squawka / SQ_Min_t	SQ_Att_min_t SQ_Def_games_t SQ_Poss_games_t	The influence of the Squawka score per minute on transfer prices
Age $\cdot Age$	Age Squared	The age of the player squared, to potentially capture non-linear effects.
Squawka Scores $\cdot ECI$	SQ_Att_t_ECI SQ_Def_t_ECI SQ_Poss_t_ECI	The Squawka score in period t multiplied by its clubs' level of play in the given period
Squawka Stats $\cdot ECI$	SQ_Games_t_ECI SQ_Min_t_ECI SQ_Rank_t_ECI	The Squawka statistic in period t multiplied by its clubs' level of play in the given period

Table 6: Interactive Variables and definitions
Where the position is either Striker, Defender or Goalkeeper and t is either 0, 1 or 2.

resulting in linear dependence. To solve this problem, we will not use any variables corresponding to the position midfielder. This way, the effect of these dummies are relative to the position midfielder, which means that for instance a positive sign on the $StrikerDummy$ is interpreted as a striker being generally sold for a higher price than a midfielder.

Next, we will remove all club dummies in buying and selling positions which are only represented in the sample once. This is again due to linear dependence issues. First, if a transfer between clubs occurs, where both clubs are not represented anywhere else in the sample, the model can not distinguish between potential effects of the buying club or the selling club, because they are both in this particular observation and only in this particular observation. These dummies will both consist of a single one for the specific observation, while for the rest of the observations they contain a zero. This makes both vectors equal to each other, causing linear dependence. Secondly, having multiple vectors consisting of zeros and a single one, are additive to a vector with two ones, perhaps on the same position as club represented twice in the sample, causing again linear dependence. These two phenomena are mathematically displayed below:

$$\begin{pmatrix} 1 \\ \vdots \\ 0 \\ \vdots \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ \vdots \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \tag{20}$$

with the dots representing all zeros for the remaining observations. The second situation in mathematics:

$$\begin{pmatrix} 1 \\ \vdots \\ 0 \\ \vdots \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{pmatrix}. \tag{21}$$

For these two reasons we do not include any club dummies for clubs represented once in the buying or once in the selling position. Note, that the second argument may still hold for clubs represented more than once in the sample, although chances are greatly reduced, it might occur that a club buys only two player in the whole sample, from a club who sells only these two players in the whole sample, causing again the zeros and ones to align. Fortunately, these do not exist in our data set. Giving a total of 277 remaining explanatory variables.

7.4 Excluded Variables

As mentioned in previous chapters, the total amount of observations left for analyses depend upon the variables included. For this reason, some variables part of the data sets are not taken into consideration in this research. The variables that are neglected have the following properties:

- They reduce the sample size significantly
- They have no intuitive explanatory power
- They have shown no significant effects in past research

This resulted in removal of variables such as Player Weight and City of birth. This line of thought is also the reason why there is no dummy for a player being right- and double-footed. Past research

showed a positive effect of being left-footed on being more valuable. Including a variable that captures this effect for left- and double-footed therefore makes sense. The other way around does not.

Furthermore, sample size reduces by a significant amount the more variables regarding historical statistics are included, because they are not available for all players. For this reason, the Squawka scores, games played, and minutes played, are included up to two complete seasons in the past and all historic achievements before are not taken into account. The reason that this research considers two is, because it wants to distinguish between the importance of the season statistics leading up to the transfer period, and the past. Literature in the past did not take these into account, while expectations are that they do matter [19].

7.5 Transformations

In this research we consider different models for prediction, all belonging to the class of linear models. Recall that in chapter 6.3, the first assumption states that the model is linear in its parameters β_i. This assumption holds for any linear regression model, which includes ridge regression and the lasso [21]. Note that this assumption is flexible, due to the fact we can transform the data beforehand. This section will discuss such transformations, to allow for a as close to linear relationship between y and x_i. Furthermore, data transformations can possibly improve the interpretation of the model [41]. The remaining part of this section will discuss such transformations performed in modelling the transfer prices.

One often used transformation is the logarithmic transformation of the dependent variable. One reason is that, if we leave the dependent variable non transformed, we are considering the variables to have an additive influence on the dependent variable, whereas a logarithmic transformation changes the interpretation to having a relative influence [41]. Intuitively, additivity in variables for transfer prices does not make that much sense, being one year older does not necessarily fluctuate the transfer worth of a player with a constant number. Instead, it might make its previous worth be a certain percentage more or less. With a logarithmic transformation of the dependent variable, we allow for a multiplicative model, and this relationship is exactly what is achieved. Furthermore, when looking at the histogram of our dependent variable $Transfersum$ in figure 10, we spot a very large tail to the right. Giving reason to believe that a transformation would improve model accuracy.

The histogram of the logarithmic transformed dependent variable as well as its corresponding normal distribution is displayed in figure 11. The normal distribution is constructed via the average (14.92) and the standard deviation (1.44) of the transfer prices in the data. The figure strengthens our belief in the log transformation of our dependent variable transfer price. Not only does this this transformation increase the interpretation of the model, it is also expected to increases the fit.

For our independent variables, x_i, we would like the relationship to be as close as possible to linear with y. Creating scatter plots of y for each x_i, we are able to see if this might be violated and whether or not transformations of this particular x_i are in order. In figure 12 through 14 scatter plots for the logarithmic transformed transfer sum and the Squawka scores are displayed. It is not clear whether or not a transformation for these variables is in order, to increase the linearity. Furthermore, the scores contain negative values. Transformations such as logarithms are therefore not defined. If one wants to apply a transformation such as the natural logarithm, one needs to adjust the negative values in some way, or remove them from the sample. These adjustments make the results regarding this variable harder to interpret, and therefore this research does not consider a transformation regarding the Squawka scores. All remaining scatter plots of the dependent variable transfer prices on its independent variables, are displayed in the appendix section 10.2. All together, the set of variables is complete and the models are defined. The results are presented in the next chapter.

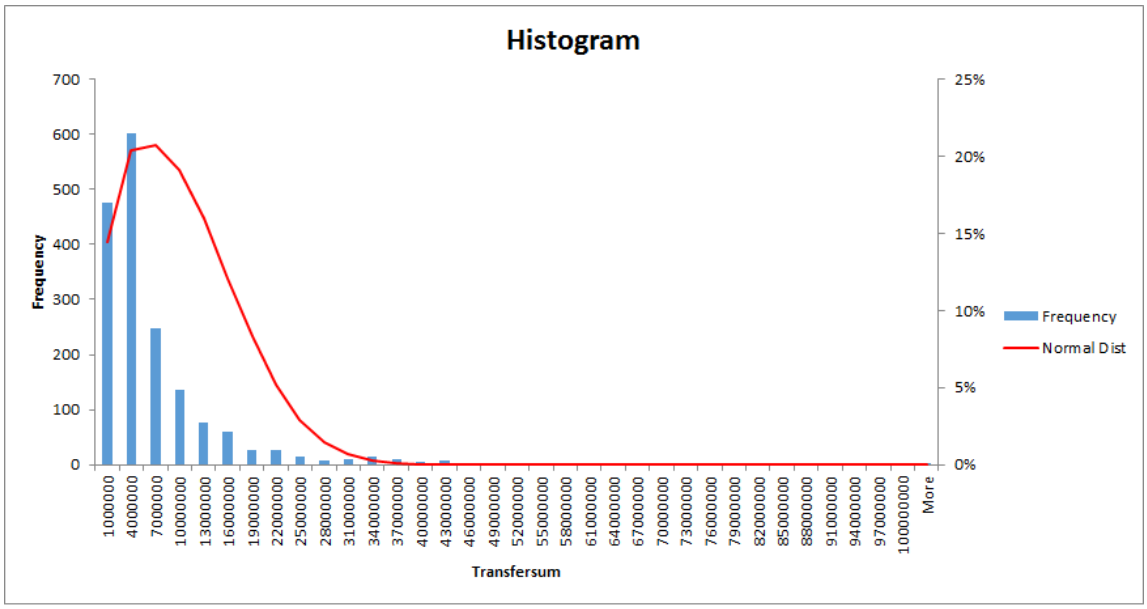


Figure 10: Histogram transfer prices

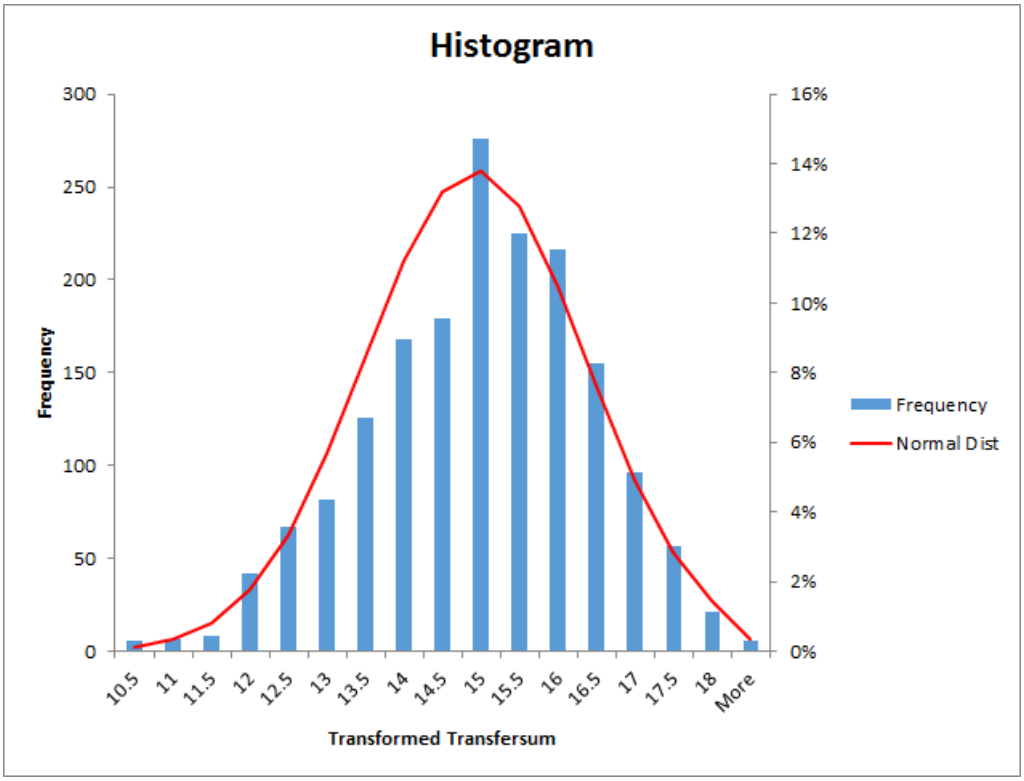


Figure 11: Histogram transformed transfer prices

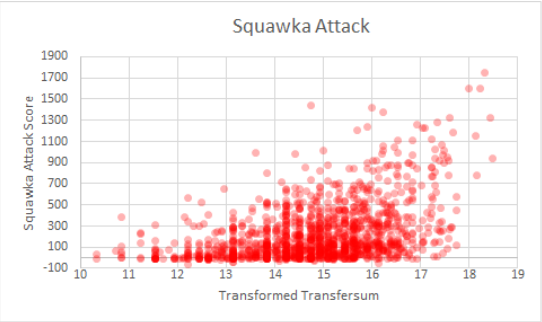


Figure 12: Squawka Attacking

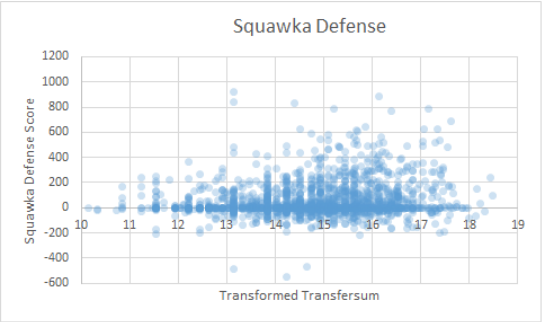


Figure 13: Squawka Defending

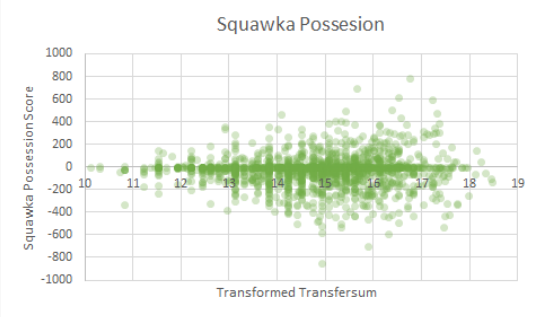


Figure 14: Squawka Possession

8 Results

In this chapter the results of this research are presented. First the results regarding the penalization parameter λ are presented. Next, the empirical results of all five of the models are presented, after which they are discussed. Post-lasso estimation results will then follow, and finally a robustness check on certain parameters of interest is performed. From here on we will refer to the ridge regression and the lasso with the minimum prediction value of λ as respectively Ridge1 and Lasso1. The Ridge regression and the lasso with a maximum λ within one standard deviation of its minimum prediction value for λ will be referred to as Ridge2 and Lasso2 respectively.

8.1 Lambda

This section shows the findings for the penalization parameter λ in both ridge regression and both of the lasso models. The Lasso1 model returns for each fold a value of λ for all hundred runs. The minimum average value found in a run corresponds to 0.035, with its maximum 0.062. An average of 0.046 and a median of 0.045 per run were found. The Lasso2 model returns higher values of λ, with an average of 0.176 and a median of 0.177 per run. The minimum value is 0.122 and the maximum value for λ is 0.215. The Ridge1 model returns an average value of λ of 1.612 per run, with its median equal to its average. The minimum corresponds to 1.21 and the maximum to 2.30. The Ridge2 model gives an average λ per run of 6.00, its median is 5.95. Minimum and maximum are respectively 4.45 and 8.38. All these statistics are summed up table 7.

Model	**Mean**	**Median**	**Maximum**	**Minimum**
Lasso1	0.046	0.045	0.062	0.035
Lasso2	0.176	0.177	0.215	0.122
Ridge1	1.612	1.612	1.21	2.30
Ridge2	6.00	5.95	8.38	4.45

Table 7: Overview of lambda values for averages of a run for each model.

Plots regarding all runs and all folds of the ridge regression and the lasso models, can be found in the appendix in sections 10.6 through 10.9. Note that the value of λ potentially influences the selected variables as well as their coefficient magnitudes. The higher λ values corresponding to the Lasso2 model will therefore result in a different model. A robustness check will be performed to see how these values of λ influence the non-zero coefficients. This will be done in section 8.5.

8.2 The five models

Table 8 shows the average MSE and the average MAE of the forward selection, the Ridge1, Ridge2, Lasso1 and Lasso2 models for 100 runs. We can see that the Lasso1 performs best on average when considering any of the error measures MSE or MAE. The MAE is also lower on average for all considered models, even though its value is below one. Indicating that there are some badly predicted runs causing the error to be larger than one, and therefore potentially blow up the MSE because of the square.

Table 9 shows for each model, the amount of runs in which they got the smallest MSE and MAE, averaged over all folds.

Tables 8 and 9 gives stronger belief as to why the Lasso1 should be the preferred model of choice for prediction. The Lasso1 performed the best in 90 out of a hundred runs when comparing mean

Model	Mean Squared Error	Mean Absolute Error
Forward stepwise selection	0.853	0.676
Ridge1	0.863	0.642
Ridge2	1.000	0.732
Lasso1	0.765	0.622
Lasso2	0.872	0.685

Table 8: MSE and MAE for all considered prediction models

Model	Runs with smallest average MSE	Runs with smallest average MAE
Forward stepwise selection	5	0
Ridge1	3	9
Ridge2	0	0
Lasso1	90	91
Lasso2	2	0

Table 9: Runs in which considered model ranks with the lowest MSE and MAE.

squared error, and 91 out of a hundred runs when comparing mean absolute error. The mean squared error and mean absolute errors returned values of 0.765 and 0.622 respectively. Considering that the dependent variable, transfer prices, are log-transformed, these values can not be interpreted directly. The errors correspond to the differences in the prediction in logarithm and the true value in logarithmic form. By transforming these values back to its original form, we see the relative performance of the models. In the case of the Mean Absolute Error of the Lasso1 model, this is as follows:

$$e^{0.622} = 1.86. \tag{22}$$

It seems that our absolute errors are wrong by a factor of nearly 2. Giving reason to believe that there are more factors influencing the transfer price of football players or that there exists a lot of randomness. This will be discussed more thoroughly in the next chapter.

All models choose different amounts of variables. Their average number of variables with non-zero coefficients over all runs is shown in table 10. We can see that both ridge models perform relatively poor by not having the benefit of variable selection, which the Lasso has. The forward stepwise selection clearly picks less variables for predicting, than it would if we were to maximize R-squared or even Adjusted R-squared. Furthermore, interesting to note is that the forward stepwise selection is only able to beat all models when considering the MSE, this might be an indication that this model is more likely to generate multiple small errors than it is to predict a large errors. This way, the MAE is still worse, while some bad prediction of the other models are penalized more in the MSE criterion

Model	Average amount of variables with non-zero coefficients
Forward stepwise selection	15
Ridge1	277
Ridge2	277
Lasso1	69
Lasso2	12

Table 10: Amount of variables picked per model. Averaged over all runs and folds.

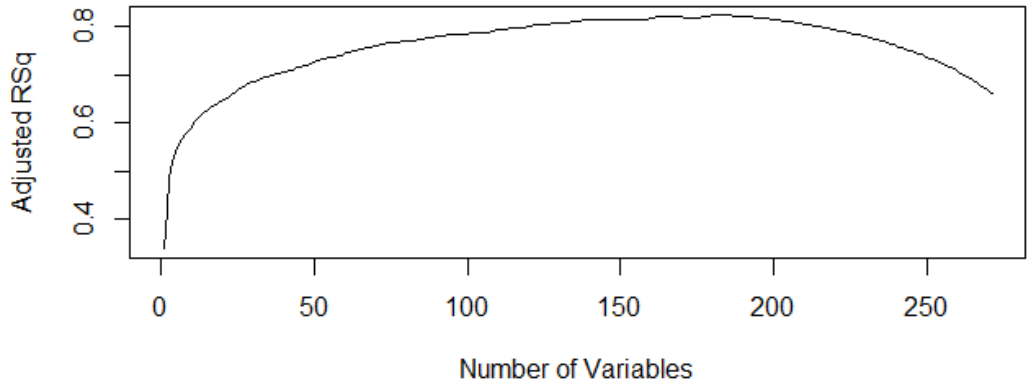


Figure 15: Adjusted R-squared for each step in selection procedure.

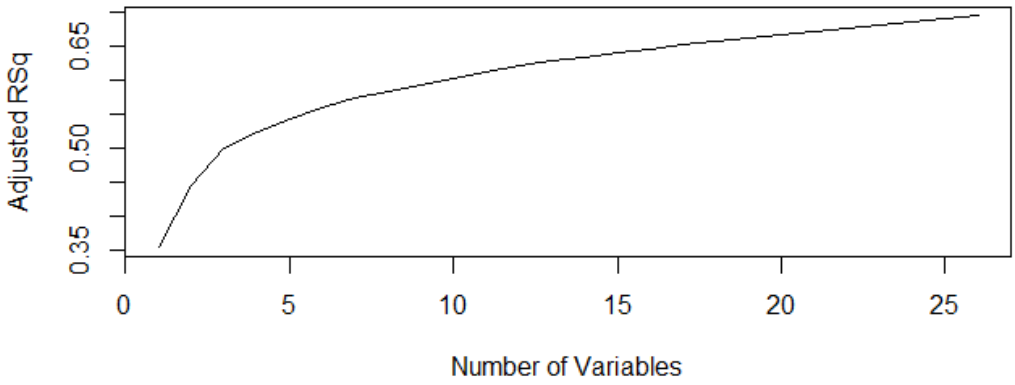


Figure 16: Adjusted R-squared for the first 25 steps in selection procedure.

and therefore considered worse. This line of thought can also be considered for the Ridge1 model. Its ability to outperform the other models is larger when considering MAE instead of MSE. Perhaps the ridge predicts some transfer prices very far off, while maintaining a small error in most of the others, allowing for a relatively better MAE than MSE.

In the previous chapter, we stated that by validating each step in the forward stepwise selection model and picking the model of which the MSE is the lowest, potentially results in a model with less variables and not the model that has highest explanatory power over the input data. Figure 15 shows a plot of each step in the selection procedure (and therefore the amount of variables picked) corresponding to one run and its corresponding adjusted R-squared. Recall that the adjusted R-squared already takes into account that a variable can have explanatory power over our explained variable by chance, and therefore penalizes the explanatory power of the model by each additional variable. Next, figure 16 shows a plot of the first twenty-five steps in the selection procedure corresponding to one of the 100 runs. The variable amount picked via the prediction error in the validation test concluded, that 6 variables picked gave the minimal prediction error in this particular run. The figure further strengthens the argument the highest adjusted R-squared, is not necessarily useful in prediction purposes.

Table 11 shows a list of variables, that were picked more than 50% of all runs in the forward stepwise selection model, how often they were picked, and their corresponding average coefficient. We expect these variables to play a decisive role in the transfer price of football players, or to be highly correlated with one. Table 12 shows similar information, but then with regards to the Lasso2 model. Note

Variables	Amount picked	Average coefficient
$Intercept$	1000	-97.31
$MonthsRemaining$	1000	0.029
$AgeSquared$	798	-0.0018
ECI_New_Team	1000	0.00096
$SQ\_Att\_0 * ECI$	905	$3.68 \cdot 10^{-7}$
$SQ\_Games\_0 * ECI$	545	$5.38 \cdot 10^{-6}$

Table 11: Decisive variables for the forward stepwise selection model.

Variables	Amount picked	Average coefficient
$Intercept$	1000	9.93
$MonthsRemaining$	1000	0.020
ECI_New_Team	1000	0.00074
SQ_Rank_0	630	-0.00011
$SQ\_Att\_0 * ECI$	992	$2.14 \cdot 10^{-7}$
$SQ\_Att\_2 * ECI$	721	$3.08 \cdot 10^{-8}$
$SQ\_Games\_0 * ECI$	1000	$7.09 \cdot 10^{-6}$
$SQ\_Games\_1 * ECI$	586	$8.56 \cdot 10^{-7}$

Table 12: Decisive variables for the Lasso2 model.

that the forward stepwise selection procedure and Lasso2 have strong preference towards some of the same variables displayed in the tables. What is interesting is that the SQ_Att_0 and SQ_Games_0 seems to be of great importance, given the level of play. This implies that the total amount of attacking statistics a player gathers through a season, are worth more the higher the strength of the team for which the player plays. The same argument holds for the amount of games played. The Lasso1 model and its corresponding selection, amounts and coefficients are displayed in table 13. These consist of 17 non-club dummies, and 43 club dummies. Only the non club dummy variables are shown in table 13. The club dummies will be presented in the next section. Note that the Lasso1 model selects far more variables on average, and therefore deviations in variables picked is likely. A further note is that we do not include any of the ridge regression models, because these models do not have a selection procedure. A final note regarding the tables: the magnitudes of the coefficients differ very much, because the methods use different penalization procedures. A better understanding of the impact of the variables can be concluded when looking at its magnitude as well as the average size of the corresponding variable in the summary statistics section in the appendix 10.1. Furthermore, section 8.4 will attempt to interpret the true values corresponding to the non-zero coefficients in the Lasso1 model.

We see that the Lasso1 includes the $SQ\_Att\_0\_Games$ variable, implying that the how many goals per games you score matters for your transfer price in this model. Also the year is an important variable in this model, with a positive sign, indicating that there is some sort of trend going on in the transfer prices of football players, i.e. prices increase over the years, everything else equal. This is in line with the fact that the largest transfer sum record gets broken almost every year. Furthermore, it seems goalkeepers are worth less than midfielders. Interesting to note is that it seems that goalkeepers are worth less when their height increases. This contradicts the intuition that goalkeepers need to have a very large height. However if we look at the mean of the $GoalkeeperDummy$ in the summary statistics section in the appendix 10.1. We note that only 5% of the sample regards goalkeepers. Calculating the minimum, maximum and mean height of a goalkeeper in the sample size

Variables	Amount picked	Average coefficient
$Intercept$	1000	-158
$Year$	987	0.084
$MonthsRemaining$	1000	0.024
$AgeSquared$	1000	-0.0012
$GoalkeeperDummy$	888	-0.14
$Height * Goalkeeper$	584	-0.00044
ECI_Old_Team	998	0.00012
ECI_New_Team	1000	0.00086
$SQ\_Att\_0\_Games$	989	0.017
$SQ\_Def\_1 * Defender$	688	0.00017
$SQ\_Att\_0 * ECI$	745	$5.41 \cdot 10^{-8}$
$SQ\_Att\_2 * ECI$	995	$1.43 \cdot 10^{-7}$
$SQ\_Def\_0 * ECI$	981	$1.47 \cdot 10^{-7}$
$SQ\_Games\_0 * ECI$	1000	$8.00 \cdot 10^{-6}$
$SQ\_Games\_1 * ECI$	835	$1.31 \cdot 10^{-6}$
$SQ\_Games\_2 * ECI$	558	$4.35 \cdot 10^{-7}$
SQ_Rank_2	558	-0.00024

Table 13: Decisive variables for the Lasso1 model.

yields: 181, 195 and 188 respectively. The sample already contains relatively large goalkeepers, when comparing to any other position, and perhaps the difference in transfer prices once one is already relatively large, is simply randomness. Finally, because the average coefficient size is measured, it could very well be that in most of the runs the variable has a positive sign, and in some of the runs a relatively large negative sign, causing its average to be negative. The reason the sign might change is due to inclusion of other variables. Perhaps in the runs where the $GoalkeeperDummy$ is not included, the $Height \cdot Goalkeeper$ returns a relatively large negative weight. Finally, note that the variables $MonthsRemaining$, ECI_New_Team, $SQ\_Att\_0\_ECI$ and $SQ\_Games\_0\_ECI$ are chosen over 50% of the time, for all models with a selection procedure. These variables are therefore likely to have the most significant effect on the transfer price.

Some variables are not picked by any models with selection procedures, and these are expected to have low to no effect on the transfer price. For the complete list, one can compare the variables mentioned, with the complete variable list in the appendix section 10.1. Some of the interesting variables with no result will be discussed here. Whether or not the player is left- or right-footed does not matter, which contradicts evidence from previous research. Furthermore, possessive in-field statistics, such as successful passing and through balls shows no impact on the transfer price. Finally, whether or not the transfer occurred during the summer or winter transfer window, does not seem to matter.

The coefficients included in the tables corresponding to the lasso models in this section should not be interpreted for true effects, reasons being that with the shrinkage caused by the lasso, all non-zero coefficients are typically biased towards zero [21]. Furthermore, without standard errors, coefficients are less meaningful. The reason standard errors are not included is due to the nature of the lasso, it returns biased coefficients. Methods to calculate standard errors that take into account potential biased estimates are not yet statistically valid [28]. An approach to reduce this bias is to use the Lasso model to identify the set of variables to include, and then apply a least squares model on this set of variables [21] [4]. This will be discussed further in section 8.4, in which we interpret the variables.

Buying club PS	Amount picked	Coefficient	**Buying Club NS**	Amount picked	Coefficient
Watford	996	0.76	Real Madrid	649	-0.13
Newcastle Utd	934	0.30	FC Barcelona	966	-0.45
Middlesbrough	970	0.48	Atlético Madrid	896	-0.39
Aston Villa	991	0.67	Hertha BSC	851	-0.18
AC Milan	961	0.25	SM Caen	935	-0.32
Crystal Palace	967	0.72	Greuther Fürth	986	-1.14
Manchester Utd	985	0.30	Fortuna Düsseldorf	840	-0.15
Redbull Leipzig	951	0.57	Standard Liege	771	0.07
Arsenal	873	0.17	Olympiacos	512	-0.05
Bologna	904	0.17	Sassuolo	987	-0.49
Manchester City	922	0.18	Besiktas	893	-0.27
Internazionale	942	0.30			
Leicester City	985	0.70			
Sunderland	988	0.83			
Swansea City	971	0.35			
Hamburger SV	859	0.15			
PSG	547	0.04			
Stoke City	795	0.18			

Table 14: Influence of clubs in buying positions on the transfer price.

8.3 Club dummies

This section will show the results of all the variables corresponding to the buying- and selling-clubs, the club-dummies. The forward stepwise selection procedure as well as the Lasso2 model, both only select dummies in rare occasions. This has to do with the fact that both models consider a relative small amount of variables, and they prefer some other variables for prediction purposes as indicated in the previous section.

Interesting is which of the club dummies receive which signs, and which of them are picked often many times by a model such as Lasso1. There are four possible outcomes for a club regarding his dummy:

- Buying Position: Positive sign (PS)
- Buying Position: Negative sign (NS)
- Selling Position: Positive sign (PS)
- Selling Position: Negative sign (NS)

The first would indicate that if this club is the buying club, the corresponding bought player would transfer for a higher price, everything else equal. If it has a negative sign, the club is able to purchase players for a relatively lower price. For the selling positions, a positive sign indicates that the club is selling their players for a higher transfer price, a negative sign that the club is selling their players for a lower transfer price.

The Lasso1 model returns an interesting list for clubs with signs on buying positions and selling positions. They are displayed in table 14 and 15. Note that again we only show the clubs which were chosen at least 50% of the time.

Note that a positive sign for a buying club dummy, indicate that this particular club, increases the transfer price of the player, indicating that this club pays more for the same player, everything

Selling club PS	Amount picked	Coefficient	**Selling Club NS**	Amount picked	Coefficient
AS Roma	628	0.05	AFC Ajax	868	-0.21
Granada	678	0.07	Napoli	696	-0.10
AS Monaco	521	0.04	SC Heerenveen	546	-0.07
Borussia Dortmund	953	0.25	SC Cambuur	954	-0.78
Lazio	507	0.20	Heracles Almelo	711	-0.10
Atalanta	756	0.08	FC Nantes	583	-0.10
Udinese	722	0.09	Levante	873	-0.32

Table 15: Influence of clubs in selling positions on the transfer price.

else constant. Furthermore note, that the buying club list (PS), contains 18 clubs, of which 12 play in the Premier League. Recall from our introduction, mentioning the billions of dollars that the are distributed among the clubs in the Premier League, making the Premier League by far the league with the highest revenue as well as having the most fans throughout the world. Taking this into account, it is not surprising that these teams with so much money are paying more than other clubs. More about this and why this is intuitive will be discussed with the Post-lasso estimation, in the next section. This section will also interpret the remainder of the club-dummies.

8.4 Lasso and Post-lasso estimation

This section will show results of two single estimations. First the Lasso1 model will be reran on all observations with λ equal to its mean optimal λ. Conclusions will be drawn regarding the estimates chosen and magnitudes. Secondly, a least squares estimation will be performed on the variables selected by this final lasso model. The latter procedure is referred to as Post-lasso estimation.

The Lasso1 model performed best in almost all runs, as well as in both error measures. Further more, the average optimal value for λ over all runs and folds is 0.046. This Lasso model is therefore chosen as the optimal model for prediction. The model parameters are estimated with all observations, and a λ equal to 0.046. It returned a selection of 66 chosen variables. Eighteen of these variables correspond to non club dummies, the other 48 correspond to club dummies. All 277 variables and their corresponding coefficients from this Lasso model are shown in the appendix section 10.10.

Assuming this final lasso model is the optimal prediction model, the least squares estimation could give more insight regarding the explanatory power of these variables. As mentioned earlier post-lasso estimation results in a smaller bias and would be able to better justify which variables fluctuate the transfer price and by how much [21] [5]. Note that this model is not considered for prediction, and mainly used to measure unbiased effects of the variables. A few remarks have to be discussed before realizing the output. Recall that the fifth assumption from chapter 6.3 in least squares estimation assumes homoskedasticity. Testing whether or not this is the case, must be done, before interpreting the results. A Breusch-Pagan heteroskedasticity test shows that heteroskedasticity is indeed present, and we must therefore deal with the problem accordingly [9]. Therefore, in the post-lasso estimation, we use white-heteroskedastic consistent standard errors [42]. The appendix section 10.4 shows the results of the heteroskedasticity test.

Finally, the standard errors reported by least squares estimation are incorrect and often too optimistic, when any kind of model selection has been done [14]. For this reason, one must adjust the standard errors such that they take into account the selection procedure foregoing the least squares estimation. Furthermore, different kind of model selection criteria, require different kinds of adjustments to the standard errors [6]. Cross validation is a prediction criteria, which influences model selection, lasso in itself performs model selection as well. In order to account for these different model

selection procedures, one must devise a statistical inference that is valid under any type of selection procedure followed [6]. A method has been devised to deal with these problems. However, one of the assumptions in order for the method to be accurate is that there is no presence of heteroskedasticity. This is unfortunately not the case in this sample and therefore we cannot use this procedure. This makes statistical inference after variable selection regarding the standard errors unrealistic [6]. We therefore do not adjust the heteroskedastic consistent standard errors. Bear in mind that these are not necessarily accurate, and they are likely to be too optimistic. The true significance and confidence intervals of the parameters in question are therefore likely to be lower and larger respectively.

Results for the least squares estimation with heteroskedastic-consistent standard errors, on the 66 variables chosen by the lasso model, yields the results as depicted in table 16 and 17.

The coefficients regarding the variables are perhaps difficult to interpret, because many of the variables have very different scales and therefore coefficient sizes. For this reason, the sixth column contains standardized coefficients. These are calculated by first subtracting the mean of all variables and dividing by their respective standard errors, after which a least squares regression is performed.

$$Standardized(x_{ij}) = \frac{x_{ij} - \mu_i}{\sigma_i}, \tag{23}$$

with μ_i defined as:

$$\mu_i = \sum_{j=1}^{n} \frac{x_{ij}}{n}, \tag{24}$$

and σ_i defined as:

$$\sigma_i = \sqrt{\frac{\sum_{j=1}^{n}(x_{ij} - \mu_i)}{n-1}} \tag{25}$$

The standardized coefficients are more easily comparable, because they all have mean zero and standard error one. This way, we can assess the relative strength of the variable as follows: One standard error increase of variable x_i yields a β_i^* increase in the log transfer price [1]. Where β_i^* refers to the coefficient of the standardized variable. A $\beta_i^* < 0.2$ is considered weak, whereas $0.2 < \beta_i^* < 0.5$ is considered moderate, and $\beta_i^* > 0.5$ is considered to have a strong effect [1]. Recall that the t-Statistics are likely to be too optimistic and cannot be used to test for significance. Evaluating both the standardized coefficient as well as its t-Statistic allows for a reasonable expectation on whether or not this variable is truly important with respect to the transfer price. Variables with a relatively low standardized coefficient, as well as a insignificant t-statistic, are expected to not have a significant impact on the transfer price. *

Lots of interesting results are present in the estimation output. Recall that we are dealing with log transformed transfer prices, and therefore the interpretation regarding the coefficients is relative. The variable $Year$ reveals that there is a trend going on where transfer prices increase over the years, by approximately 14%. Recent years have shown an increase in capital flow in the football market, which is reflected in the coefficient on $Year$. Secondly, age has a negative effect on the transfer price, the effect measured is not linear. This causes the drop in transfer price to be larger for each year one gets older. Take for example a player aged 19, turning 20 will cause his transfer price to drop by: $(20^2 \cdot -0.001389) - (19^2 \cdot -0.001389) = 0.0542$, so by 5.42%, everything else constant. This effect increases for an older person, example given aged 29, $(30^2 \cdot -0.001389) - (29^2 \cdot -0.001389) = 0.0720$, causing the transfer price to decrease by 7.20%, everything else constant. This is likely to be due to a few reasons. For one, being older makes you more injure prone, and your physique generally worsens. Second, this player's learning curve is generally less steep than for a younger player. Furthermore, someones professional football career lifespan, becomes smaller the older one gets, resulting in a lower worth and thus transfer price. More arguments can be thought of, for why this effect is found. Note that we

*To allow for a reasonably loose interpretation, considering its ill-defined standard errors, a confidence level of 90% is considered, and therefore a t-statistic of 1.645

Dependent Variable: TRANSFERSOM
Method: Least Squares
Included observations: 424 after adjustments
White heteroskedasticity-consistent standard errors & covariance

Variable	Coefficient	Std. Error	t-Statistic	Prob.	Std. Coef.
C	−268.0635	111.0915	−2.412997	0.0163	$1.91E-05$
YEAR	0.138798	0.055109	2.518621	0.0122	0.003011
AGE_SQUARED	−0.001389	0.000244	−5.688948	0.0000	−0.154250
GOALKEEPDUMMY	−1.197360	7.492696	−0.159804	0.8731	−0.346754
MONTHSREMAINING	0.021271	0.002781	7.648962	0.0000	0.196282
ECI_OLD_TEAM	$4.31E-05$	$7.88E-05$	0.546644	0.5850	0.014533
ECI_NEW_TEAM	0.001082	$9.41E-05$	11.49556	0.0000	0.466676
SQ_RANK_2	$-7.44E-05$	0.000340	−0.218805	0.8269	−0.014077
SQ_ATT_1_SQ_MIN_1	0.702951	0.429032	1.638459	0.1022	0.059596
HEIGHT_GOALKEEP	0.004226	0.039375	0.107316	0.9146	0.278956
SQ_DEF_1_DEFENDER	0.000250	0.000482	0.518720	0.6043	0.015487
SQ_ATT_0_ECI	$1.59E-07$	$8.82E-08$	1.807873	0.0715	0.101064
SQ_ATT_2_ECI	$2.46E-07$	$1.01E-07$	2.440508	0.0151	0.089994
SQ_DEF_0_ECI	$3.08E-07$	$1.08E-07$	2.855144	0.0046	0.105972
SQ_GAMES_0_ECI	$6.55E-06$	$1.94E-06$	3.369501	0.0008	0.127139
SQ_GAMES_1_ECI	$8.07E-07$	$2.58E-06$	0.313162	0.7543	0.006652
SQ_GAMES_2_ECI	$3.93E-07$	$1.79E-06$	0.219698	0.8262	0.019704
DUMMY_ATALANTA	0.216212	0.297042	0.727884	0.4672	0.018374
DUMMY_AS_ROMA	0.231369	0.193892	1.193288	0.2335	0.021873
DUMMY_GRANADA_CF	0.312446	0.221192	1.412553	0.1587	0.023368
DUMMY_LEVANTE_UD	−0.559841	0.518062	−1.080646	0.2806	−0.032693
DUMMY_FC_NANTES	−0.697912	0.276891	−2.520531	0.0121	−0.047317
DUMMY_SSC_NAPOLI	−0.482843	0.725475	−0.665554	0.5061	−0.052284
DUMMY_SC_CAMBUUR	−1.066569	0.544696	−1.958100	0.0510	−0.063330
DUMMY_BOR__DORTMUND	0.549891	0.242830	2.264510	0.0241	0.054789
DUMMY_HERACLES_ALMELO	−0.401923	0.218748	−1.837378	0.0670	−0.023097
DUMMY_AFC_AJAX	−0.658257	0.307898	−2.137902	0.0332	−0.056297
DUMMY_OGC_NICE	−0.898309	0.349204	−2.572448	0.0105	−0.060100
DUMMY_LAZIO	1.030286	1.322586	0.778994	0.4365	0.045960
DUMMY_SC_HEERENVEEN	−0.626955	0.086155	−7.277028	0.0000	−0.032401
DUMMY_FC_AUGSBURG	−0.374612	0.253179	−1.479634	0.1398	−0.033397
DUMMY_GETAFE_CF	−1.860932	0.446123	−4.171341	0.0000	−0.089989
DUMMY_UDINESE_CALCIO	0.401864	0.141155	2.846970	0.0047	0.023942

R-squared	0.794764	Mean dependent var	15.14923
Adjusted R-squared	0.758848	S.D. dependent var	1.397380
S.E. of regression	0.686215	Akaike info criterion	2.223005
Sum squared resid	169.5206	Schwarz criterion	2.834285
Log likelihood	−407.2770	Hannan-Quinn criter.	2.464519
F-statistic	22.12825	Durbin-Watson stat	1.986709
Prob(F-statistic)	0.000000	Wald F-statistic	136.2098
Prob(Wald F-statistic)	0.000000		

Table 16: Post-Lasso estimation results

Dependent Variable: TRANSFERSOM
Method: Least Squares
Included observations: 424 after adjustments
White heteroskedasticity-consistent standard errors & covariance

DUMMY_HAMBURGER_SV01	0.646235	0.257323	2.511382	0.0125	0.061908
DUMMY_WATFORD01	1.344084	0.185730	7.236766	0.0000	0.099066
DUMMY_NEWCASTLE01	0.901396	0.215644	4.180030	0.0000	0.059610
DUMMY_MIDDLESBROUGH01	1.137837	0.154944	7.343558	0.0000	0.082488
DUMMY_ASTON_VILLA01	1.374760	0.196881	6.982708	0.0000	0.129370
DUMMY_AC_MILAN01	0.571159	0.187719	3.042624	0.0025	0.044909
DUMMY_BESIKTAS01	−0.590863	0.292664	−2.018914	0.0442	−0.042527
DUMMY_FC_BARCELONA01	−1.042497	0.407404	−2.558879	0.0109	−0.089367
DUMMY_SASSUOLO01	−0.786558	1.194421	−0.658527	0.5106	−0.064800
DUMMY_MAN_UTD01	0.524838	0.190871	2.749704	0.0063	0.049642
DUMMY_PARIS_SG01	0.168172	0.198593	0.846818	0.3977	0.015749
DUMMY_STANDARD_LIEGE01	−0.407491	0.198504	−2.052810	0.0408	−0.021782
DUMMY_BOLOGNA01	0.668059	0.171492	3.895578	0.0001	0.065239
DUMMY_SAINT_ETIENNE01	−0.240022	0.255231	−0.940413	0.3476	−0.014620
DUMMY_FC_K_LAUTERN01	−0.896052	0.116873	−7.666865	0.0000	−0.039789
DUMMY_MAN_CITY01	0.557598	0.203859	2.735215	0.0065	0.042389
DUMMY_SUNDERLAND01	1.514490	0.169291	8.946086	0.0000	0.096155
DUMMY_ATLETICO_MADRID01	−1.157180	0.452441	−2.557636	0.0109	−0.070143
DUMMY_SWANSEA01	0.933534	0.251930	3.705534	0.0002	0.079908
DUMMY_INTER01	0.668563	0.363950	1.836962	0.0670	0.052974
DUMMY_LEICESTER01	1.474507	0.386672	3.813333	0.0002	0.115370
DUMMY_ARSENAL01	0.449146	0.205371	2.186998	0.0294	0.033294
DUMMY_HERTHA_BSC01	−0.726104	0.311417	−2.331616	0.0203	−0.053408
DUMMY_SM_CAEN01	−0.710411	0.193328	−3.674648	0.0003	−0.045544
DUMMY_F__DUSSELDORF01	−0.601016	0.202141	−2.973257	0.0031	−0.038390
DUMMY_REAL_MADRID01	−0.776981	0.357411	−2.173912	0.0304	−0.053715
DUMMY_SAMPDORIA01	0.564006	0.184613	3.055072	0.0024	0.035443
DUMMY_CRYSTAL_PALACE01	1.500934	0.177727	8.445145	0.0000	0.073383
DUMMY_STOKE_CITY01	0.778467	0.309933	2.511724	0.0125	0.044119
DUMMY_GREUTHER_FURTH01	−1.648314	0.474125	−3.476543	0.0006	−0.086832
DUMMY_RB_LEIPZIG01	1.490190	0.159054	9.369066	0.0000	0.074326

R-squared	0.794764	Mean dependent var	15.14923
Adjusted R-squared	0.758848	S.D. dependent var	1.397380
S.E. of regression	0.686215	Akaike info criterion	2.223005
Sum squared resid	169.5206	Schwarz criterion	2.834285
Log likelihood	−407.2770	Hannan-Quinn criter.	2.464519
F-statistic	22.12825	Durbin-Watson stat	1.986709
Prob(F-statistic)	0.000000	Wald F-statistic	136.2098
Prob(Wald F-statistic)	0.000000		

Table 17: Post-Lasso estimation results

only consider professional leagues, and that we have no players in our sample aged 17 or younger. It may very well be that a player of a younger age might be worth less, and the price will increase up to a certain age, after which it drops again. This research cannot answer this question, because there is no data on all age groups. Note that the amount of months that are left on the contract before the player is sold is also of importance. Each extra month causes the price to increase by 2%. For each extra month a player has a on a contract the selling club could potentially benefit from him, it requires some reimbursement to take away this potential benefit, and therefore the measured effect is no surprise. Next, we note that the ECI_OLD_TEAM variable has a positive coefficient, although due to its relatively large standard error its true confidence interval is likely to be huge and it is therefore difficult to interpret this variable. Note that the interactive variables with ECI, correspond to the ECI of the selling club and therefore ECI_OLD_TEAM, which means the strength of the selling club is important none the less. Next, the strength of the buying team increases the transfer price by 0.1% per ECI point.This can be explained through several arguments. For one, stronger clubs are generally more successful, win more prizes and therefore attract more marketing campaigns and sponsorships, which yields more money and thus more financial power to buy these players. Secondly, the selling clubs might know of the financial strength and be inclined to go to greater lengths to get this higher transfer price from these clubs. Probably more arguments can be thought of, for why this effect is present. SQ_Rank_2 returning a negative sign contradicts intuition, as this means that the better the player performs relatively in his league, the lower his transfer price. However, this variable is likely to be insignificant, due to its relatively large standard error, and therefore not too much emphasize should be put on this result. The height of the goalkeeper shows a positive sign, indicating that the height of the goalkeeper has a positive impact on his transfer price. The variable is again likely to be insignificant.

The Squawka attacking score per minute $SQ\_ATT\_1\_SQ\_MIN\_1$ indicates that the attacking performance per minute is important during the first half of the season. It is unexpected that this variable is included but the attacking performance per minute during the second half is not. However, we note that the variables $SQ\_ATT\_0\_ECI$ and $SQ\_ATT\_2\_ECI$, are included in the estimation, and these variables do capture attacking performance during the second half of the season. The Squawka score for a defender $SQ\_DEF\_1\_DEFENDER$, shows a relatively small standardized coefficient, as well as likely being statistically insignificant. The Squawka scores regarding attacking qualities and games however, multiplied by the level of play (ECI) are important. For instance, an increase in the amount of games played in the most previous season times the ECI of the club for which the games were played, causes the transfer price to increase by 0.000655%. Note that the interactive variables in this research are generally very large, and therefore coefficients which represent the effect of a variable increase of 1, have very small numbers. The standardized coefficients give a better insight regarding the relative effect of these variables and their coefficients. The standardized coefficients regarding the interactive variables with ECI, are relatively high, showing that they are important relative to the other variables. This effect is expected, as the 'better' the club, the more money they have and the more money they want to let go of a strong player who performs well. Next, a player becomes more expensive when more games have been played on a high level. This has probably to do with consistency and not being injure-prone, as well experience playing on a high level, giving clubs more willingness to pay higher prices.

When considering the club dummies, we can see that club differences are present. Some relatively large standardized coefficients are displayed. Recall that an "01" on the end of the club dummy implicates that the club is the buying club. There are four different club effects to distinguish from:

- Positive sign on buying position
- Negative sign on buying position
- Positive sign on selling position
- Negative sign on selling position

The first indicates that this club generally pays a higher transfer price, than other clubs would, given equal parameter values. The second effect indicates that this club pays a lower transfer price in general. The third that this club is able to sell their players for more money. The fourth that a club generally sells their players for a lower transfer price. Note that it is difficult to interpret the standard errors and therefore the statiscal significance regarding the coefficients. Not all effects mentioned are therefore expected to be present for all the corresponding clubs.

The list of clubs for which a positive dummy was included on the buying position is very interesting. The list is dominated by Premier League clubs, having 12 clubs from the Premier League, as well as 6 non-premier league clubs. One would expect that the sportive success captures the wealth of the clubs for a great deal, but the Premier League clubs have more much wealth than their sportive successes does believe, and are therefore perhaps inclined to pay more. The remaining clubs with a positive sign on the buying position are AC Milan, Redbull Leipzig, Bologna, Internazionale, PSG and Hamburger SV. AC Milan, has enjoyed many financial injections from Chinese investors, wanting to bring the club back to its formal glory. The same holds for Internazionale of which many stocks were bought by Chinese investors back in 2012. Both these clubs used to be European Giants, and have won many trophies in the past, but haven't been able to reach their level of sportive successes in years. The Chinese investors injected lots of capital into these clubs, to invest in new and better players, and potentially become one of Europe's best again. Therefore their capital is far bigger than their ECI does believe, and is most likely the reason for the dummy inclusion by the models. Redbull Leipzig is slightly different, but has a similar story. The club was founded in 2009, and started many leagues below the Bundesliga, the top league in Germany, after which they received many giant fundings from their main sponsor, Redbull. Again, this club has much more money than their ECI does believe, and is not worried about possible bankruptcy, giving us a likely explanation for the inclusion of this club dummy. Paris Saint German (PSG), was bought by Qatar Sports Investments back in 2011, after which the club made giant investments to be one of the worlds best, again giving a possible explanation for the inclusion of the dummy. The latter are Bologna and Hamburger SV, for which there is no specific information giving reason to believe that these club are likely to pay higher transfer prices. It could be random variation of the data, it could also be bad scouting policies of the club.

Negative signs on buying club dummies indicate that these clubs are able to buy players for less money. Real Madrid, FC Barcelona and Atlético Madrid, have spanned the top of the football world in the most recent years, recall table 5 in chapter 5 with the highest ranked football teams according to their ECI. The above mentioned teams are respectively first, second and fourth on this list. Many players are likely to want to go to these clubs, making it easier for these clubs to attract players at lower prices. The remaining clubs, Besiktas, Sassuolo, Standard Liege, Saint Etienne, FC K. Lautern, Hertha BSC, SM Caen, F. Dusseldorf and Greuther Fürth, are not immediately clear as to why they would be able to buy good players for a relatively small transfer sum. It could simply be good scouting policies.

The selling club dummies consist for one of AS Roma, Borussia Dortmund, AS Monaco, Lazio and Atalanta, indicating that these clubs are able to receive more money for the same player as would any other club, everything else equal. There is no clear reason as to why this would be the case. These clubs however, are considered among the sub-top of Europe, often a shopping place for the top tier clubs of Europe. This should be captured by the ECI of the buying club, but perhaps their is some sort of bidding war going on between these clubs, causing the transfer price to inflate. This will be discussed more thoroughly in the Recommendation chapter. The remaining clubs Udinese and Granada, perhaps have been lucky, have strong negotiating skills within the clubs or are included due to pure randomness.

Negative signs on the positions imply that these clubs sell their players for a smaller amount than would any other club. Possibly financial instability or bad judgment could cause this to happen.

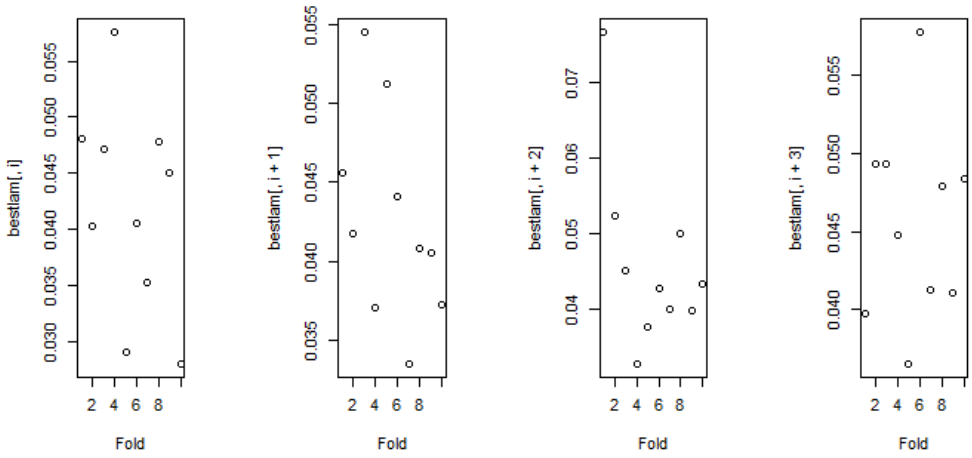


Figure 17: Lasso1 different lambda values per fold

Interesting to note is that no dutch teams are presented on any other position but are presented four times in this particularly position, namely SC Cambuur, Heracles Almelo, AFC Ajax and SC Heerenveen. These clubs are very different within the Dutch league and therefore no immediate conclusion can be drawn based on this result. The remaining clubs on this list are Levante UD, FC Nantes, SSC Napoli, OGC Nice, FC Augsburg and Getafe CF.

8.5 Robustness

This section will perform certain robustness checks. First the chosen values of λ will be compared. Second, the effect of club dummies on the remaining coefficients will be discussed.

In the first section, some summary statistics regarding the values of λ were shown. Figures 17 through 20 show the spread in folds for four runs for all four models. The remainder of the plots for all runs and all four models can be found in the appendix section 10.6 through 10.9. Note that the spread in these plots, results in different empirical results for each fold. The effect of having a larger or smaller λ on the variables chosen, will be investigated next. Investigations regarding the magnitudes of the coefficients when λ fluctuates will be done after.

The Lasso1 and Lasso2 models use different values of λ for penalization of the variables. The average optimal λ value for Lasso1 equals 0.046, where the value for λ in Lasso2 equals 0.176. In the appendix section 10.10, results regarding the Lasso1 model with a 0.046 λ are displayed. In table 18 we can find the results regarding the Lasso2 model with the λ value of 0.176. Note that only the variables with a non-zero coefficient are displayed. All variables picked by the Lasso2 model are also picked by the Lasso1 model, except for the SQ_Rank_0 variable.

Next, we will try and determine the order in which variables are added once the penalty becomes smaller. By decreasing the value of λ in small steps and noting all variables, we are able to determine the importance of each variable and the propensity of the lasso model to choose a certain variable. Starting at 0.176, we take small steps of 0.05 and report the differences compared to the previous model. This will yield a total of 26 steps. Table 19 and 20 shows all steps and the results.

The club dummies added in steps with higher values of λ, are likely to have a stronger effect. This would then be the case for Gruether Fürth as a buying club, SC Cambuur as a selling club, Watford

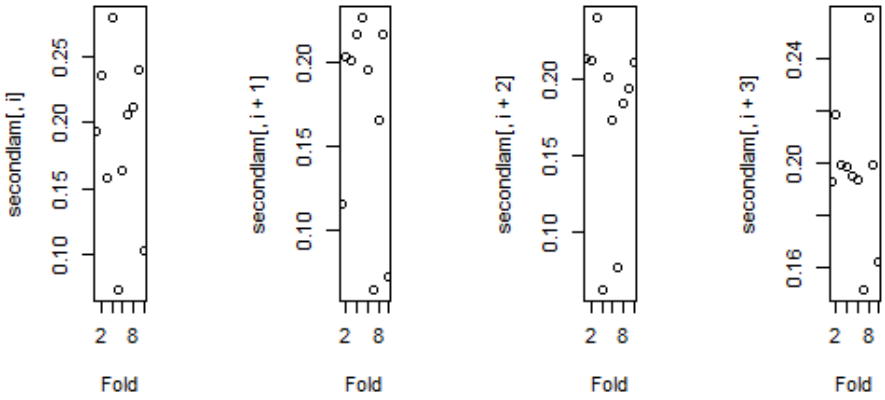


Figure 18: Lasso2 different lambda values per fold

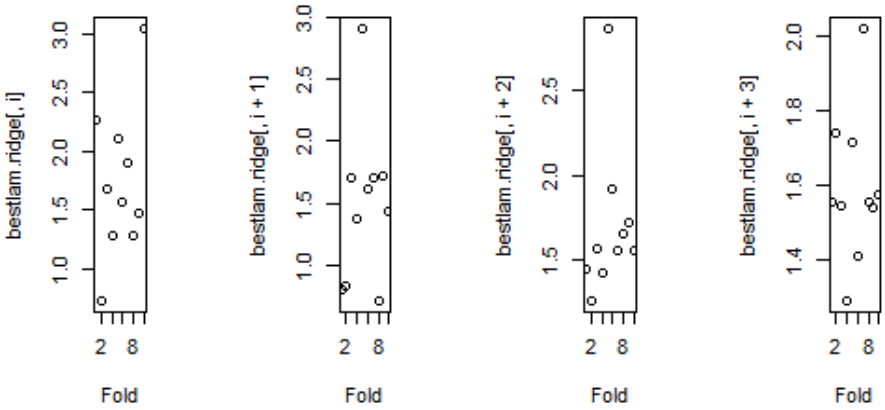


Figure 19: Ridge1 different lambda values per fold

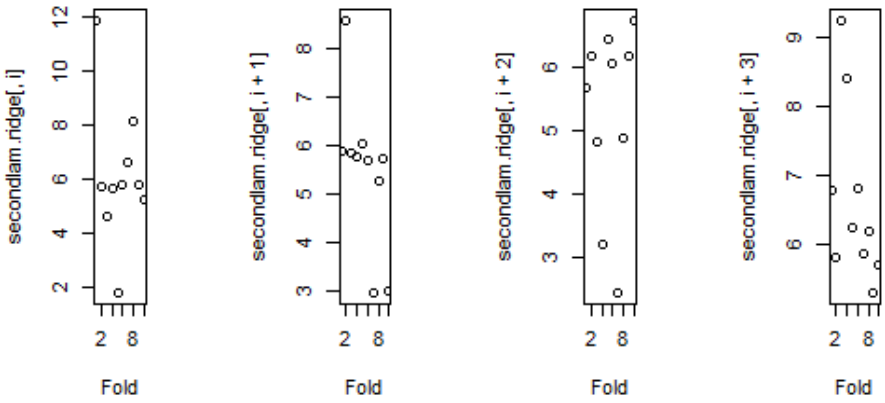


Figure 20: Ridge2 different lambda values per fold

Variables	**Coefficient**
Intercept	12.4
Age_Squared	$-5.73 \cdot 10^{-4}$
MonthsRemaining	$2.09 \cdot 10^{-2}$
ECI_Old_Team	$5.05 \cdot 10^{-6}$
ECI_New_Team	$7.44 \cdot 10^{-4}$
SQ_Rank_0	$-1.38 \cdot 10^{-4}$
SQ_Att_0*ECI	$2.25 \cdot 10^{-7}$
SQ_Att_2*ECI	$4.16 \cdot 10^{-8}$
SQ_GAMES_0*ECI	$7.32 \cdot 10^{-6}$
SQ_GAMES_1*ECI	$8.46 \cdot 10^{-7}$

Table 18: Results for Lasso2 model with $\lambda = 0.176$.

λ	**Added variable**	**Removed variable**
0.171	-	-
0.166	-	-
0.161	SQ_Rank_2	-
0.156	-	-
0.151	Dummy_Greuther_Fürth_1	-
0.146	-	-
0.141	-	-
0.136	-	-
0.131	-	-
0.126	Dummy_SC_Cambuur	-
0.121	Dummy_Watford_1	-
0.116	Dummy_Aston_Villa_1	-
0.111	Dummy_Sunderland_1 Dummy_Sassuolo_1 Dummy_Getafe_1	-
0.106	SQ_Att_0/SQ_Games_0 Year Dummy_Man_Utd_1 Dummy_Leicester_1	-
0.101	GoalkeepDummy	-
0.096	SQ_Att_0 $\cdot ECI$ SQ_Att_2 $\cdot ECI$ SQ_Def_0 $\cdot ECI$	-
0.091	SQ_Games_0 $\cdot ECI$ SQ_Games_1 $\cdot ECI$ Dummy_Crystal_Palace_1 Height*Goalkeeper	-
0.086	Dummy_Middlesbrough_1	-
0.081	Dummy_FC_Barcelona_1 Dummy_Bor._Dortmund	-
0.076	Dummy_RB_Leipzig_1 Dummy_SM_Caen_1 Dummy_Internazionale_1 Dummy_Swansea_1 Dummy_Besiktas_1 Dummy_Levante	-
0.071	SQ_Games_2 $\cdot ECI$ Dummy_Man_City_1 Dummy_AC_Milan_1 Dummy_Newcastle_Utd_1	-

Table 19: Comparison with Lasso2 model.

Lambda Value	Added variable	Removed variable
0.066	SQ_Defender_1*Defender SQ_Rank_1 Dummy_Arsenal_1 Dummy_Atlético_Madrid_1 Dummy_FC_K'Lautern_1	-
0.061	Dummy_Düsseldorf_1 Dummy_Bologna_1 Dummy_Standard_Liege_1 Dummy_AFC_Ajax	SQ_Rank_0
0.056	Dummy_Stoke_City_1 Dummy_Hertha_BSC_1 Dummy_Saint_Etienne_1 Dummy_Hamburger_SV_1 Dummy_Atalanta	-
0.051	Dummy_Real_Madrid_1 Dummy_Udinese_Calcio Dummy_Heracles_Almelo Dummy_AS_Roma	SQ_Rank_1
0.046	Dummy_Sampdoria_1 Dummy_Augsburg Dummy_SC_Heerenveen Dummy_Lazio Dummy_OGC_Nice Dummy_Inter Dummy_FC_Nantes Dummy_SSC_Napoli	-

Table 20: Comparison with Lasso2 model.

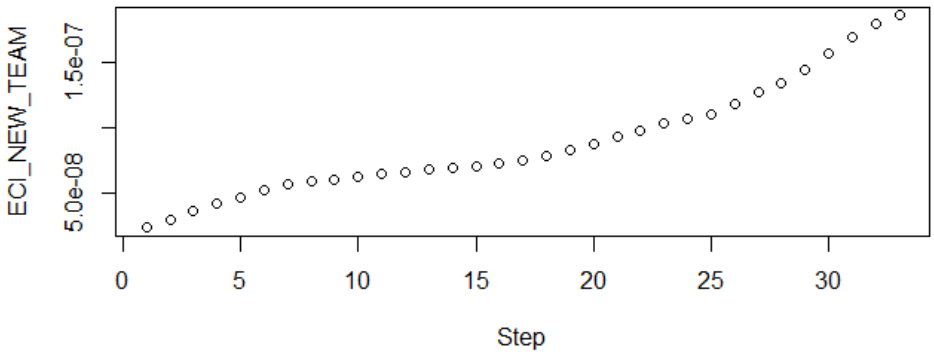


Figure 21: Sensitivity of coefficient size to λ

and Aston Villa as buying clubs. Clubs added in the final step are likely to be of less importance. Furthermore, note that the difference in variables returned between all values of λ is very small. Only the SQ_Rank_0 and SQ_Rank_1 variable get added and removed for some value of λ.

Next, we will determine the effect of the λ value on the magnitudes of the non-club dummies. We will include all variables from the post-lasso estimation from table 16 17, except for the club dummies. We expect these variables to be of high importance, considering they were picked by the lasso model with the best performance value λ. Bear in mind that the magnitudes of these variables depend upon the variables included and therefore sudden jumps might be due to a change in the set of non-zero coefficients. A plot with the ECI_NEW_TEAM coefficient size for different λ values is given in figure 21. The steps refer to steps of 0.005 for λ, starting at value λ = 0.191 to λ = 0.031. Which captures both the Lasso1 λ = 0.046 average optimal value, as well as the Lasso2 λ = 0.176 average optimal value. Appendix section 10.5 shows plots regarding all variables from the post-lasso estimation.

Table 21, gives an overview of coefficient sizes for four values of λ. Generally, we expect variable magnitudes to increase for a decrease in λ. This does not hold for all the variables represented in the table. This is verified by the plots in appendix 10.5. Note that this is likely due to inclusion of other variables, with decreases in λ, causing the other variables to be of less importance. Note next, the huge drop in the intercept. This is likely caused by the inclusion of the variable $Year$ in this step, explaining the trend in transfer prices going on. A final note, the intercept is the only variable which switches signs during the decrease in λ values.

Next, we will determine the effect of the club dummies on the remaining variables. For interpretation purposes, we will use all variables from the post-lasso estimation. Recall from section 7.3, that no club dummies in the total data set are involved, for which only one transfer on the corresponding buying or selling position took place. However, there do exist club dummies for which there are for instance, two, three or four observations total. Its interesting to determine how much the inclusion of these club dummies affect the coefficients of the remainder of the variables, and if perhaps only clubs with at least a certain amount of variables should be included. Table 22 gives an overview of the coefficients from multiple least squares estimations. The left column corresponds to the non club dummy variables from the post-lasso estimation. The second column corresponds to the post-lasso estimation on the variables picked by Lasso1. Columns three through five correspond to post-lasso estimations on the variables picked by Lasso1, with only club dummies involved with at least 4, 6 and

Main Variables	λ **=0.186**	λ **=0.136**	λ **= 0.086**	λ **= 0.036**
C	12.41089	12.42475	-51.40548	-201.1783
Year	0	0	0.03161635	0.1056585
AGE_SQUARED	-0.0005148356	-0.0008270806	-0.001073515	-0.00121842
GOALKEEPDUMMY	0	0	-0.05610533	-0.05904031
MONTHSREMAINING	0.02044825	0.02286362	0.02470517	0.02342014
ECI_OLD_TEAM	0	5.380054e-05	0.0001045379	0.000110202
ECI_NEW_TEAM	0.0007418317	0.0007423513	0.0007567188	0.0009138812
SQ_RANK_2	0	-0.0001395736	-0.0003368257	-0.0002293991
SQ_ATT_1_SQ_MIN_1	0	0	0	0.07731674
HEIGHT_GOALKEEPER	0	0	-0.0001175971	-0.001212933
SQ_DEF_1_DEFENDER	0	0	0	0.0003090044
SQ_ATT_0_ECI	2.28031e-07	2.268661e-07	1.729566e-07	3.386226e-11
SQ_ATT_2_ECI	2.976342e-08	6.613519e-08	9.786862e-08	1.796422e-07
SQ_DEF_0_ECI	0	0	5.133557e-08	1.553099e-07
SQ_GAMES_0_ECI	7.334855e-06	7.088491e-06	7.51569e-06	8.424169e-06
SQ_GAMES_1_ECI	5.024856e-07	1.96279e-06	2.174379e-06	9.268025e-07
SQ_GAMES_2_ECI	0	0	0	1.684332e-07

Table 21: Coefficient sensitivity for different values of λ

8 observation on the relevant position. The sixth column corresponds to the coefficients without any club dummies.

Most of the variables show an increase in magnitude, when less club dummies are included. None of the coefficients show a variation in their sign. The next chapter will summarize the important results, as well as conclude the findings of the research.

Dependent Variable: TRANSFERSOM
Method: Least Squares
Included observations: 424 after adjustments

Variable	Coefficient	4+	6+	8+	None
C	−268.0635	−338.7268	−409.3910	−425.9837	−348.1711
YEAR	0.138798	0.173938	0.209091	0.217432	0.178856
AGE_SQUARED	−0.001389	−0.001635	−0.001738	−0.001784	−0.001799
GOALKEEPDUMMY	−1.197360	−2.866335	−2.807241	−5.922878	−7.141999
MONTHSREMAINING	0.021271	0.022264	0.025926	0.025710	0.027313
ECI_OLD_TEAM	$4.31E-05$	$9.69E-05$	0.000135	0.000187	0.000193
ECI_NEW_TEAM	0.001082	0.000997	0.000875	0.000782	0.000723
SQ_RANK_2	$-7.44E-05$	−0.000479	−0.000680	−0.000643	−0.000594
SQ_ATT_1_SQ_MIN_1	0.702951	0.815981	0.906040	0.851787	0.690924
HEIGHT_GOALKEEP	0.004226	0.013234	0.013208	0.029896	0.036312
SQ_DEF_1_DEFENDER	0.000250	0.000801	0.000577	0.000501	0.000584
SQ_ATT_0_ECI	$1.59E-07$	$1.21E-07$	$1.57E-07$	$2.11E-07$	$2.32E-07$
SQ_ATT_2_ECI	$2.46E-07$	$1.95E-07$	$1.15E-07$	$1.69E-07$	$1.61E-07$
SQ_DEF_0_ECI	$3.08E-07$	$1.78E-07$	$1.58E-07$	$2.54E-07$	$2.57E-07$
SQ_GAMES_0_ECI	$6.55E-06$	$7.63E-06$	$7.88E-06$	$6.61E-06$	$6.41E-06$
SQ_GAMES_1_ECI	$8.07E-07$	$3.41E-06$	$4.75E-06$	$4.42E-06$	$5.74E-06$
SQ_GAMES_2_ECI	$3.93E-07$	$8.76E-07$	$2.10E-06$	$1.28E-06$	$1.83E-06$

Table 22: Robustness Check on Post-Lasso estimation results.

9 Summary Conclusions and Recommendations

This research examined the transfer prices of football players in an attempt to predict them. It considers five different models to do so, which are tested repeatedly, with different error measures. This chapter summarizes the results, concludes the findings and discusses potential improvements for future research.

9.1 Summary

This research finds that the Least Absolute Shrinkage and Selection Operator (lasso) where its penalization parameter λ is chosen by minimizing its prediction error, performs best for predicting transfer prices of football players. Its average mean squared error and mean absolute errors in predicting the logarithmic transfer price are respectively 0.765 and 0.622. Furthermore, it is able to perform best considering both error measures, for respectively 90 and 91 out of a hundred runs. In addition, this model chooses on average 69 non-zero coefficients out of the total set of 277 explanatory variables. Its average value for the penalization parameter λ, is 0.046. The final prediction model is therefore the lasso with a λ equal to 0.046, which returns 66 variables with non-zero coefficients. The complete prediction model and its coefficients can be found in the appendix 10.10.

The lasso typically returns biased estimates, and therefore the coefficients cannot be interpreted as true effects. To interpret the 66 variables chosen by the prediction model, a least squares estimation is performed on the non-zero coefficients of the lasso model with a λ value of 0.046. The output as well as arguments for this procedure are discussed in section 8.4. By comparing standardized coefficients and t-statistics, the following variables are expected to play important roles:[†]

- $AgeSquared$
- $MonthsRemaining$
- ECI_New_Team
- $SQ\_Att\_0\_ECI$
- $SQ\_Att\_2\_ECI$
- $SQ\_Def\_0\_ECI$
- $SQ\_Games\_0\_ECI$

Taking into account the corresponding signs, they imply the following: First, the age of a player negatively influences transfer prices, and this effect becomes stronger the older one becomes. Secondly, for each extra month on the contract that must be broken for the transfer to proceed, the transfer price increases. Thirdly, the higher the strength of the team interested in buying the player, measured by the Euro Club Index, the larger the transfer sum. Fourth and fifthly, the in-field performance of players regarding plays on offense in the most recent summer window, as well as the year before, multiplied by the strength of the team in which the player plays, positively influences the transfer price. Sixthly, the amount of defending plays made in the most recent summer transfer window, multiplied by the strength of the team in which the player played, positively influences the transfer price. Finally, the amount of games played in the most recent summer window, multiplied by the strength of the team in which the player played, positively influences the transfer price. The latter results imply that the performance on the field and games played, increases the transfer sum, and that this increase becomes greater with a higher level of play. Out of these variables, the following variables are picked in more than 50% of runs by all models with a selection procedure, and therefore considered of greatest

[†]The t-statistics are generally too optimistic, because of the selection procedure foregoing the least squares estimation. More detailed explanation can be found in section 8.4.

importance in modelling transfer prices: $MonthsRemaining$, ECI_New_Team, $SQ\_Att\_0\_ECI$, $SQ\_Games\_0\_ECI$. Characteristics that showed no clear effect on the transfer prices are the following: whether or not the player is left-footed, possessive in-field statistics, whether or not the transfer took place during the winter or summer transfer window

Next, this research finds that Premier League clubs typically pay larger transfer prices, as well as clubs that benefit from large investors. Furthermore, it finds that three out of four of the top four best teams throughout Europe, typically pay less, than their characteristics do believe. Next, it finds that sub-top teams often receive larger transfer sums, and finally no obvious results are found for clubs that sell players for smaller transfer sums.

9.2 Conclusions

This research attempts to answer the question: *"What is the predicted transfer price of a football player, given its characteristics."* It concludes that the Least Absolute Shrinkage and Selection Operator, with its penalization parameter λ equal to 0.046, is able to best predict transfer prices for football players, out of all considered models. It should not be used as an measure of the true correct transfer price, as its errors are significant. The model can, however, be used to give an indication about the validity of a transfer price.

We conclude that the characteristics of greatest importance in determining the transfer price are the following: The amount of months that remain on the contract to break, players are worth more if bound longer to a club. Next, better teams generally pay more money for players. Finally, the amount of games someones plays or goal scoring opportunities someone creates increase the transfer worth, this worth increases with the strength of the club for which this player plays.

Finally regarding clubs, this research concludes that it is likely that the Premier League clubs, as well as clubs influenced by large investors, pay larger transfer sums, because they are generally wealthier. Also, Real Madrid, FC Barcelona and Atlético Madrid, the top tier clubs in Spain and respectively the first, second and fourth strongest team according to the Euro Club Index, attract players for a lower transfer price, likely due to their popularity. Next, sub-top teams, such as AS Roma, Borussia Dortmund, AS Monaco, Atalanta and Lazio, receive higher transfer prices for players than expected, likely because they sell players to the strongest teams throughout Europe and multiple teams are interested in the same player. No conclusions are drawn regarding clubs selling players for a lower amount than expected. The results showed a variety of clubs with different characteristics.

9.3 Recommendations

In this chapter we focus on future research on the subject of transfers in football. In the previous chapter we mentioned that the prediction errors are still relatively high, implying that there might be some other decisive variables that are not included in the models. We will name some of which we believe might be important.

There might be some merit in the amount of clubs interested in the player. The current models take into account the clubs that buy and sell the player, but not if there were multiple clubs involved. Examples are young players in mid tier clubs who suddenly break through in a giant tournament, such as the World Cup. The top clubs in Europe all want to grab this talent before he becomes too expensive, or signs for a competitor. This drives the price up madly, because the selling club can discuss prices with multiple clubs. The increase in popularity of the player by playing in a giant tournament and succeeding, has given the player even more value, which brings us to our next improvement for further research: Popularity. Buying a player solely to increase sportive successes would yield different prices as they would, when marketing finances are taken into account. Not only can this player increase sportive successes, and thus gaining money for the club. Popularity means sponsoring and

shirt sales, giving a direct way of gaining money on the purchase of a player. This would imply that even though a player does not improve his skill, becoming more popular increases his worth. Furthermore, it also works the other way around. The media channels tend to focus more on transfers worth millions. Players that sell for huge amounts, receive even more media attention afterwards, increasing their popularity and potentially the shirt sale and media incomes. Hypothetically this mean that in some occasions paying more for the same player, would yield more money in the long run.

An interesting development in the football world is the quick advance of the Chinese football leagues. Xi Jinping, president of China, has made a policy plan that would make Chinas sport economy be twice as big in 2025 as the whole worlds sport economy together is right now. Furthermore, he wants China to win ánd organize the worldcup of football within fifteen years. All these plans do not come with no funding, China is finishing building 20.000 football schools by December 2017, and is investing hugely in European stars, to make the sport more popular in their own country [48]. Many stars in Europa do not want to transfer to a Chinese league, where the overall level of play is still far worse. For this reason, transfer bids by Chinese clubs have sky-rocketed, making record-breaking bids on European players, of which many have accepted. This research does not take into account these Chinese developments and the potential effects it may have on the transfer prices in Europa.

Our final recommendation for further research, regards information on the terms in which the transfer took place. Very often a player leaves when he feels undervalued or not appreciated, causing the player to lose motivation and therefore value for the club. Other clubs are still highly interested and can come to an agreement fairly easy with the player. The club that owns the player faces the following dilemma, not selling the player, will likely mean that the player does not sign a new contract, and walks out the door for free. Selling him, means that clubs are likely to bid less, causing the transfer price to deflate. The same story can be told when there is controversy between the coach and the player or between a player and another player, the club has no benefit in keeping this player, because of the unease between the two, causing the transfer price to deflate.

A last note regards the assumptions in 6.3. The second assumption states that the sample studied is representative for the true model. Transfers are often not publicly available, and the transfers that are, might have a reason to be made available. Either these are huge money transfers and therefore there is more incentive to leak them, or these are transfers within big leagues or between big clubs. If this would be the case, the data observed is not representative for the true population, because there is selection bias in which transfers are studied. We expect that within the next years, this potential selection bias will decrease, as more and more transfers are becoming publicly available. The fourth assumption states that the error has an expected value of zero, given any of the independent variables. This assumption might fail, if we expect there is an important variable omitted, which is correlated with any of the included variables [41]. The above recommendations for future research might involve one of these variables, which could greatly impact the transfer price, as well as being correlated with any of the other variables, creating this omitted variable problem. This would cause the assumption to be violated. If we expect this to be the case, future research should involve as many of these variables as possible.

So far we have focused on possible improvements for the fit of the model by including more variables. An other improvement would be a larger data set.The current data set, even though it started off with many observations, dropped to a much smaller amount when all variables were included. This total set then had to be divided into a training set and a test set, of which the training set in some cases had to be divided again into a validation set. These divisions are mandatory for a good modelling and prediction procedure, but damage the amount of observations on which we can fit the model. Recent years, information regarding transfers have become more publicly available, giving reason to believe that applying the same method a year or perhaps two later, would yield many more observations.

References

Acock, A. C. (2014). A Gentle Introduction to Stata (4th ed.). Texas: Stata Press.

Ahrens, A., Bhattacharjee, A. (2015). Two-step lasso estimation of the spatial weights matrix. Econometrics, 3(1), 128-155.

Bailey, T. L., Elkan, C. (1993). Estimating the accuracy of learned concepts.". In Proc. International Joint Conference on Artificial Intelligence.

Belloni, A., Chernozhukov, V. (2009). Least squares after model selection in high-dimensional sparse models.

Belloni, A. and Chernozhukov, V. (2012). Supplement to “Least squares after model selection in high-dimensional sparse models.”

Berk, R., Brown, L., Buja, A., Zhang, K., Zhao, L. (2013). Valid post-selection inference. The Annals of Statistics, 41(2), 802-837.

Breiman, L. (1995). Better subset regression using the nonnegative garrote. Technometrics, 37(4), 373-384.

Breiman, L., Spector, P. (1992). Submodel selection and evaluation in regression. The X-random case. International statistical review/revue internationale de Statistique, 291-319.

Breusch, T. S., Pagan, A. R. (1979). A simple test for heteroscedasticity and random coefficient variation. Econometrica: Journal of the Econometric Society, 1287-1294.

Bryson, A., Frick, B., Simmons, R. (2013). The returns to scarce talent: footedness and player remuneration in European soccer. Journal of Sports Economics, 14(6), 606-628.

Carmichael, F., Thomas, D., Ward, R. (2000). Team performance: the case of English premiership football. Managerial and decision Economics, 31-45.

Dobson, S., Gerrard, B. (1999). The determination of player transfer fees in English professional soccer. Journal of Sport Management, 13(4), 259-279.

Dobson, S., Gerrard, B., Howe, S. (2000). The determination of transfer fees in English nonleague football. Applied Economics, 32(9), 1145-1152.

Efron, B. (2014). Estimation and accuracy after model selection. Journal of the American Statistical Association, 109(507), 991-1007.

Efron, B., Tibshirani, R. (1997). Improvements on cross-validation: the 632+ bootstrap method. Journal of the American Statistical Association, 92(438), 548-560.

"English Premier League generates highest revenue, German Bundesliga most profitable". The Observer. Guardian News and Media. 10 June 2010. Retrieved 20 September 2010.

Feess, E., Tibitanzl, F. (2004). Mikroökonomie.

Flood, I., Kartam, N. (1994). Neural networks in civil engineering. I: Principles and understanding. Journal of computing in civil engineering, 8(2), 131-148.

Frick, B. (2007). THE FOOTBALL PLAYERS'LABOR MARKET: EMPIRICAL EVIDENCE FROM THE MAJOR EUROPEAN LEAGUES. Scottish Journal of Political Economy, 54(3), 422-446.

Jain, A. K., Dubes, R. C., Chen, C. C. (1987). Bootstrap techniques for error estimation. IEEE transactions on pattern analysis and machine intelligence, (5), 628-633.

Hastie, T., Tibshirani, R., Friedman, J. (2002). The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Biometrics.

Hastie, T., Tibshirani, R., Wainwright, M. (2015). Statistical learning with sparsity: the lasso and generalizations. CRC press.

Hoerl, A. E., Kennard, R. W. (1970). Ridge regression: Biased estimation for nonorthogonal problems. Technometrics, 12(1), 55-67.

Hübl, L., Swieter, D. (2002). Der Spielemarkt in der Fußball-Bundesliga. In Sportökonomie (pp. 105-126). Gabler Verlag.

Kim, J. H. (2009). Estimating classification error rate: Repeated cross-validation, repeated hold-out and bootstrap. Computational statistics data analysis, 53(11), 3735-3745.

Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. International Joint Conference on Artificial Intelligence, 14, 1137–1145.

Krstajic, D., Buturovic, L. J., Leahy, D. E., Thomas, S. (2014). Cross-validation pitfalls when selecting and assessing regression and classification models. Journal of cheminformatics, 6(1), 10.

Kyung, M., Gill, J., Ghosh, M., Casella, G. (2010). Penalized regression, standard errors, and Bayesian lassos. Bayesian Analysis, 5(2), 369-411.

Leeb, H. (2008). Evaluation and selection of models for out-of-sample prediction when the sample size is small relative to the complexity of the data-generating process. Bernoulli, 661-690.

Maier, H. R., Dandy, G. C. (2000). Neural networks for the prediction and forecasting of water resources variables: a review of modelling issues and applications. Environmental modelling software, 15(1), 101-124.

Matheson, V. (2003). European football: a survey of the literature. Mimeo, Department of Economics, Williams College

M. Stone, "Cross-Validatory Choice and Assessment of Statistical Predictions," J. Royal Statistical Soc. Series B, vol. 36, pp. 111-147, 1974.

Osborne, M. R., Presnell, B., Turlach, B. A. (2000). A new approach to variable selection in least squares problems. IMA journal of numerical analysis, 20(3), 389-403.

Osborne, M. R., Presnell, B., Turlach, B. A. (2000). On the lasso and its dual. Journal of Computational and Graphical statistics, 9(2), 319-337.

Park, T., Casella, G. (2008). The bayesian lasso. Journal of the American Statistical Association, 103(482), 681-686.

"Premier League value of central payments to Clubs". The Premier League. Retrieved 6 June 2017.

"Premier League wages keep on rising, Deloitte says". BBC News. British Broadcasting Corporation. 9 June 2011. Retrieved 13 August 2012.

Ruijg, J., van Ophem, H. (2015). Determinants of football transfers. Applied Economics Letters, 22(1), 12-19.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society. Series B (Methodological), 267-288.

"Top Soccer Leagues Get 25% Rise in TV Rights Sales, Report Says". Bloomberg. Retrieved 4 August 2014

Wooldridge, J. M. (2015). Introductory econometrics: A modern approach. Nelson Education.

White, Halbert (1980), "A Heteroskedasticity-Consistent Covariance Matrix Estimator and a Direct Test for Heteroskedasticity", Econometrica, 48 (4): 817–838

Yuan, M., Lin, Y. (2007). On the non-negative garrotte estimator. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 69(2), 143-161.

Zhang, P. (1993). Model selection via multifold cross validation. The Annals of Statistics, 299-313.

Zou, H., Hastie, T. (2005). Regularization and variable selection via the elastic net. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 67(2), 301-320.

http(1). http://www.euroclubindex.com/

http(2). http://www.hypercube.nl/welcome

http(3). http://www.independent.co.uk/news/world/asia/chinas-xi-jinping-loves-football-so-much-hes-put-it-on-the-national-curriculum-but-can-he-secure-the-10071110.html

http(4). http://www.squawka.com/football-player-rankings

http(5). https://www.transfermarkt.com/

10 Appendix

10.1 Summary Statistics Correlation diagram

Variable	Mean	Standard Deviation	Kurtosis	Skewness
Natural log of Transfer sum	15,15	1,40	2,47	-0,79
Transfer sum	8253564	11675268	19,40	3,61
Month of transfer	6,84	1,66	7,43	-2,86
Year of transfer	2015	0,73	-1,00	-0,47
Winter dummy	0,07	0,26	9,33	3,36
Age of player	25,61	3,15	-0,25	0,31
Age Squared	666	165	0,13	0,61
Striker Dummy	0,28	0,45	-0,99	1,01
Goalkeep Dummy	0,05	0,22	14,51	4,06
Defend Dummy	0,27	0,44	-0,88	1,06
Months Remaining	22,61	12,17	-0,62	0,50
Left foot dummy	0,28	0,45	-1,07	0,97
Height of player	182	6,14	-0,38	-0,17
ECI Selling Club	2686	582	0,95	0,94
ECI Buying Club	2659	610	0,58	0,76
SQ_Minutes_0	1791	901	-1,10	-0,28
SQ_Minutes_1	958	566	1,04	0,61
SQ_Minutes_2	1626	904	-1,07	0,01
SQ_Games_0	24,04	9,42	-0,65	-0,67
SQ_Games_1	12,89	6,05	2,17	0,70
SQ_Games_2	22,38	9,79	-0,87	-0,44
SQ_Defence_0	90,22	171	3,05	1,29
SQ_Defence_1	49,43	103	3,72	1,44
SQ_Defence_2	82,41	172	5,37	1,74
SQ_Attack_0	256	257	4,27	1,73
SQ_Attack_1	142	164	17,23	2,90
SQ_Attack_2	227	225	2,09	1,45
SQ_Poss_0	-41,83	155	1,40	0,14
SQ_Poss_1	-24,04	90	2,35	-0,28
SQ_Poss_2	-30,34	164	2,40	0,43
SQ_Rank_0	177	139	-0,07	0,84
SQ_Rank_1	177	130	-0,82	0,51
SQ_Rank_2	208	152	-0,79	0,58

Interaction Variable	Mean	Standard Deviation	Kurtosis	Skewness
SQ_Att_0/min	0,15	0,13	19,69	2,73
SQ_Att_1/min	0,14	0,12	0,91	1,04
SQ_Att_2/min	0,15	0,14	18,12	2,96
SQ_Def_0/min	0,05	0,10	22,73	2,66
SQ_Def_1/min	0,04	0,11	13,09	1,14
SQ_Def_2/min	0,03	0,13	19,39	-1,95
SQ_Poss_0/min	-0,02	0,09	0,73	0,29
SQ_Poss_1/min	-0,03	0,10	0,55	0,08
SQ_Poss_2/min	-0,03	0,10	0,41	0,17
SQ_Att_0/games	9,80	8,07	2,53	1,34
SQ_Att_1/games	9,64	8,71	2,60	1,41
SQ_Att_2/games	9,40	8,16	4,68	1,65
SQ_Def_0/games	3,44	6,32	1,96	0,41
SQ_Def_1/games	3,25	7,16	1,53	0,29
SQ_Def_2/games	3,13	7,15	2,58	0,43
SQ_Poss_0/games	-1,45	5,83	0,49	0,46
SQ_Poss_1/games	-1,47	6,33	0,47	0,40
SQ_Poss_2/games	-1,38	6,48	0,93	0,48
Height*Striker	50	82	-0,97	1,01
Height*Goalkeep	9,79	42	14,55	4,06
Height*Defend	49	82	-0,87	1,06
Sq_Att_0*Striker	114	239	8,51	2,62
Sq_Att_1*Striker	61	135	7,14	2,63
Sq_Att_2*Striker	96	207	6,13	2,49
Sq_Def_0*Striker	-2,23	42	41	-3,21
Sq_Def_1*Striker	-0,25	26	16,26	-0,03
Sq_Def_2*Striker	-4,45	41	12,05	-1,78
Sq_Poss_0*Striker	-41	99	5,83	-2,43
Sq_Poss_1*Striker	-24	60	11,36	-3,12
Sq_Poss_2*Striker	-40	98	8,76	-2,41
Sq_Att_0*defender	30	65	5,73	2,49
Sq_Att_1*defender	16	40	7,05	2,74
Sq_Att_2*defender	27	63	8,28	2,83
Sq_Def_0*defender	55	133	7,09	2,70
Sq_Def_1*defender	31	84	11,50	3,22
Sq_Def_2*defender	49	129	9,11	2,77
Sq_Poss_0*defender	6,17	74	13,10	0,90
Sq_Poss_1*defender	3,96	43	17,08	1,40
Sq_Poss_2*defender	8,91	83	15,41	2,18
Sq_Att_0*Goalie	0,30	4,30	410	20,11
Sq_Att_1*Goalie	0,29	4,34	415	20,27
Sq_Att_2*Goalie	0,26	1,86	114	9,99
Sq_Def_0*Goalie	8,07	80	68	6,61
Sq_Def_1*Goalie	3,19	36	54	4,52
Sq_Def_2*Goalie	9,86	88	92	8,95
Sq_Poss_0*Goalie	0,59	21	94	6,39
Sq_Poss_1*Goalie	0,49	9,36	99	7,57
Sq_Poss_2*Goalie	1,78	28	48	3,62

Interaction Variable	Mean	Standard Deviation	Kurtosis	Skewness
SQ_Att_0*ECI Selling club	683080	730894	7,47	2,20
SQ_Att_1*ECI Selling club	374618	451112	22	3,34
SQ_Att_2*ECI Selling club	618009	665160	4,38	1,89
SQ_DEF_0*ECI Selling club	243973	482510	4,26	1,67
SQ_DEF_1*ECI Selling club	129732	279075	3,66	1,49
SQ_DEF_2*ECI Selling club	214968	462118	5,87	1,80
SQ_POSS_0*ECI Selling club	-86352	437022	4,37	1,02
SQ_POSS_1*ECI Selling club	-48701	246033	3,83	0,47
SQ_POSS_2*ECI Selling club	-56294	478419	7,11	1,28
SQ_RANK_0*ECI Selling club	466152	369399	-0,10	0,83
SQ_RANK_1*ECI Selling club	469252	350831	-0,43	0,60
SQ_RANK_2*ECI Selling club	549441	414706	-0,21	0,76
SQ_GAMES_0*ECI Selling club	64177	28652	-0,07	0,10
SQ_GAMES_1*ECI Selling club	34033	16600	2,22	0,82
SQ_GAMES_2*ECI Selling club	59515	28705	-0,16	0,14
SQ_MIN_0*ECI Selling club	4719705	2511583	-0,33	0,17
SQ_MIN_1*ECI Selling club	2501582	1491635	1,43	0,76
SQ_MIN_2*ECI Selling club	4290187	2483829	-0,20	0,32

Selling Club Dummies	Buying Club Dummies
1.FSV Mainz 05	1. FC Köln
AC Milan	1.FC K'lautern
ADO Den Haag	1.FSV Mainz 05
AFC Ajax	AC Milan
AS Roma	AFC Ajax
Aston Villa	Arsenal
Atalanta	AS Roma
Atlético Madrid	Aston Villa
AZ Alkmaar	Atalanta
Bay. Leverkusen	Atlético Madrid
Bayern Munich	AZ Alkmaar
Benfica	Bay. Leverkusen
Besiktas	Bayern Munich
Bor. Dortmund	Besiktas
Bor. M'gladbach	Bologna
Braga	Bor. Dortmund
Bursaspor	Bor. M'gladbach
Cagliari Calcio	Cagliari Calcio
Chelsea	Celta de Vigo
Chievo Verona	Chelsea
Dinamo Moscow	Chievo Verona
E. Frankfurt	Club Brugge
Évian	Crystal Palace
FC Augsburg	Dep. La Coruña
FC Barcelona	E. Frankfurt
FC Groningen	F. Düsseldorf
FC Lorient	FC Augsburg
FC Nantes	FC Barcelona
FC Schalke 04	FC Basel
FC Twente	FC Groningen
FC Utrecht	FC Ingolstadt
Feyenoord	FC Lorient
Fiorentina	FC Porto
G. Bordeaux	FC Schalke 04
Galatasaray	Feyenoord
Genoa	Fiorentina
Getafe CF	Galatasaray
Granada CF	Genoa
Guingamp	Granada CF
Hamburger SV	Greuther Fürth
Hannover 96	Hamburger SV
Hellas Verona	Hannover 96
Heracles Almelo	Hellas Verona
Hertha BSC	Hertha BSC
Inter	Inter
Juventus	Juventus
Lazio	Lazio

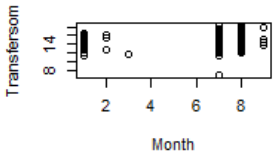
Selling Club Dummies	Buying Club Dummies
Levante UD	Leicester
LOSC Lille	Liverpool
Málaga CF	LOSC Lille
Man City	Man City
Marseille	Man Utd
Monaco	Marseille
Montpellier	Middlesbrough
OGC Nice	Monaco
Olympique Lyon	Newcastle
Paris SG	OGC Nice
Parma	Olympiacos
PEC Zwolle	Olympique Lyon
PSV Eindhoven	Paris SG
Real Madrid	Pescara
Real Sociedad	PSV Eindhoven
Saint-Étienne	RB Leipzig
Sampdoria	Real Betis
Sassuolo	Real Madrid
SC Bastia	Real Sociedad
SC Cambuur	RSC Anderlecht
SC Freiburg	Rubin Kazan
SC Heerenveen	Saint-Étienne
Sevilla FC	Sampdoria
Spurs	Sassuolo
SSC Napoli	SD Eibar
Stade Rennais	Sevilla FC
SV Darmstadt 98	SM Caen
Swansea	Southampton
Torino	SSC Napoli
Toulouse	Stade Rennais
Trabzonspor	Standard Liège
TSG Hoffenheim	Stoke City
Udinese Calcio	Sunderland
US Palermo	Swansea
Valencia CF	Torino
VfB Stuttgart	Trabzonspor
Villarreal CF	TSG Hoffenheim
Vitesse	Udinese Calcio
Werder Bremen	Union Berlin
	US Palermo
	Valencia CF
	VfB Stuttgart
	VfL Wolfsburg
	Villarreal CF
	Watford
	Werder Bremen
	West Ham

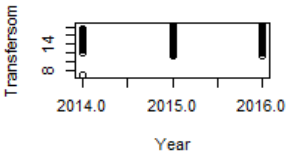
Table

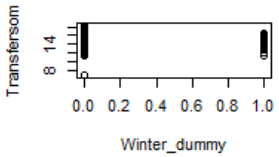
	AGE	HEIGHT	REMAINING...	TRANSFE...	ECI_NEW_...	ECI_OLD_...	SQ_ATT_0	SQ_DEF_0	SQ_POSS_0
AGE	1.000000	0.049132	-0.144802	-0.041682	-0.024551	0.035631	-0.011822	0.099519	0.145615
HEIGHT	0.049132	1.000000	-0.008626	-0.034653	-0.042495	-0.030163	-0.092455	0.049117	0.076042
REMAINING_MONTHS	-0.144802	-0.008626	1.000000	-0.007538	0.224911	0.090537	0.008233	0.055730	-0.045095
TRANSFER_YEAR	-0.041682	-0.034653	-0.007538	1.000000	0.004373	-0.038957	-0.052382	0.039928	-0.034154
ECI_NEW_TEAM	-0.024551	-0.042495	0.224911	0.004373	1.000000	0.354945	0.047026	0.085309	0.049919
ECI_OLD_TEAM	0.035631	-0.030163	0.090537	-0.038957	0.354945	1.000000	0.047687	-0.030946	0.054430
SQ_ATT_0	-0.011822	-0.092455	0.008233	-0.052382	0.047026	0.047687	1.000000	-0.053211	-0.372941
SQ_DEF_0	0.099519	0.049117	0.055730	0.039928	0.085309	-0.030946	-0.053211	1.000000	0.257151
SQ_POSS_0	0.145615	0.076042	-0.045095	-0.034154	0.049919	0.054430	-0.372941	0.257151	1.000000
SQ_GAMES_0	0.070219	-0.028736	0.003159	0.018244	0.082046	-0.020008	0.548690	0.337487	-0.191893
SQ_MIN_0	0.091363	-0.013742	0.021468	0.045154	0.109362	-0.045494	0.498608	0.441336	-0.117219
SQ_RANK_0	-0.035464	0.018886	-0.031659	-0.025581	-0.347532	-0.109735	-0.167572	-0.294898	-0.217069
STRIKERDUMMY	0.006143	0.038500	0.019692	-0.062062	-0.069364	0.013541	0.298980	-0.282621	-0.414967
DEFENDDUMMY	0.036529	0.226963	-0.085122	-0.009186	0.054735	-0.021889	-0.259513	0.229286	0.232711
MIDFIELDDUMMY	-0.112588	-0.228379	0.017656	0.069337	0.028667	0.019525	0.132560	-0.103217	0.061032
GOALKEEPDUMMY	0.083477	0.276347	-0.030256	0.011498	0.077403	-0.018065	-0.147081	0.088436	0.100297
WINTERDUMMY	0.073446	-0.023793	-0.010600	-0.006436	0.012056	0.013655	-0.140715	-0.110079	0.046671
DUMMYLEFT	-0.043934	-0.140630	0.017217	-0.021183	0.053540	0.044738	0.002023	0.047519	-0.000437

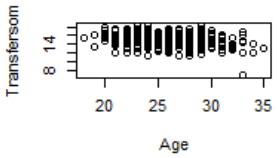
	SQ_GAME...	SQ_MIN_0	SQ_RANK_0	STRIKERD...	DEFENDD...	MIDFIELDD...	GOALKEE...	WINTERD...	DUMMYLEFT
AGE	0.070219	0.091363	-0.035464	0.006143	0.036529	-0.112588	0.083477	0.073446	-0.043934
HEIGHT	-0.028736	-0.013742	0.018886	0.038500	0.226963	-0.228379	0.276347	-0.023793	-0.140630
REMAINING_MONTHS	0.003159	0.021468	-0.031659	0.019692	-0.085122	0.017656	-0.030256	-0.010600	0.017217
TRANSFER_YEAR	0.018244	0.045154	-0.025581	-0.062062	-0.009186	0.069337	0.011498	-0.006436	-0.021183
ECI_NEW_TEAM	0.082046	0.109362	-0.347532	-0.069364	0.054735	0.028667	0.077403	0.012056	0.053540
ECI_OLD_TEAM	-0.020008	-0.045494	-0.109735	0.013541	-0.021889	0.019525	-0.018065	0.013655	0.044738
SQ_ATT_0	0.548690	0.498608	-0.167572	0.298980	-0.259513	0.132560	-0.147081	-0.140715	0.002023
SQ_DEF_0	0.337487	0.441336	-0.294898	-0.282621	0.229286	-0.103217	0.088436	-0.110079	0.047519
SQ_POSS_0	-0.191893	-0.117219	-0.217069	-0.414967	0.232711	0.061032	0.100297	0.046671	-0.000437
SQ_GAMES_0	1.000000	0.943851	-0.173795	-0.008161	-0.017757	0.046756	0.006324	-0.358187	0.019168
SQ_MIN_0	0.943851	1.000000	-0.199167	-0.128303	0.080603	0.001824	0.050969	-0.309588	0.024374
SQ_RANK_0	-0.173795	-0.199167	1.000000	0.103875	-0.088839	-0.088477	0.116176	-0.017776	-0.136944
STRIKERDUMMY	-0.008161	-0.128303	0.103875	1.000000	-0.364819	-0.461840	-0.130451	0.042965	-0.034373
DEFENDDUMMY	-0.017757	0.080603	-0.088839	-0.364819	1.000000	-0.331332	-0.156771	-0.052821	0.081662
MIDFIELDDUMMY	0.046756	0.001824	-0.088477	-0.461840	-0.331332	1.000000	-0.093711	0.000981	-0.043725
GOALKEEPDUMMY	0.006324	0.050969	0.116176	-0.130451	-0.156771	-0.093711	1.000000	-0.017969	-0.108418
WINTERDUMMY	-0.358187	-0.309588	-0.017776	0.042965	-0.052821	0.000981	-0.017969	1.000000	0.058058
DUMMYLEFT	0.019168	0.024374	-0.136944	-0.034373	0.081662	-0.043725	-0.108418	0.058058	1.000000

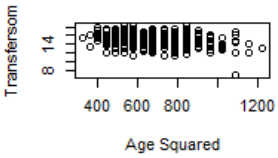
10.2 Scatter Plots

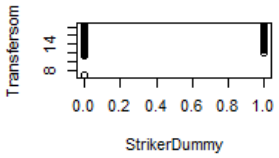


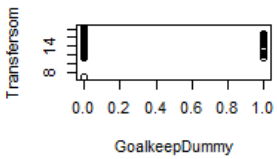


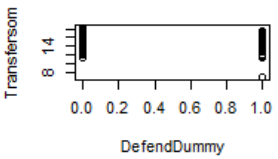


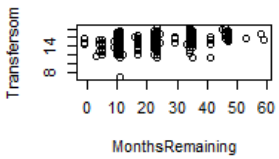


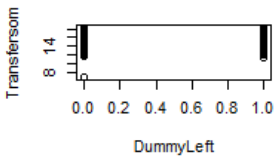


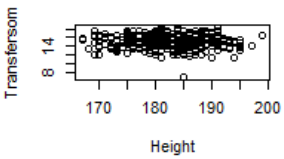


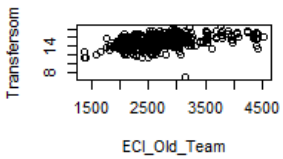


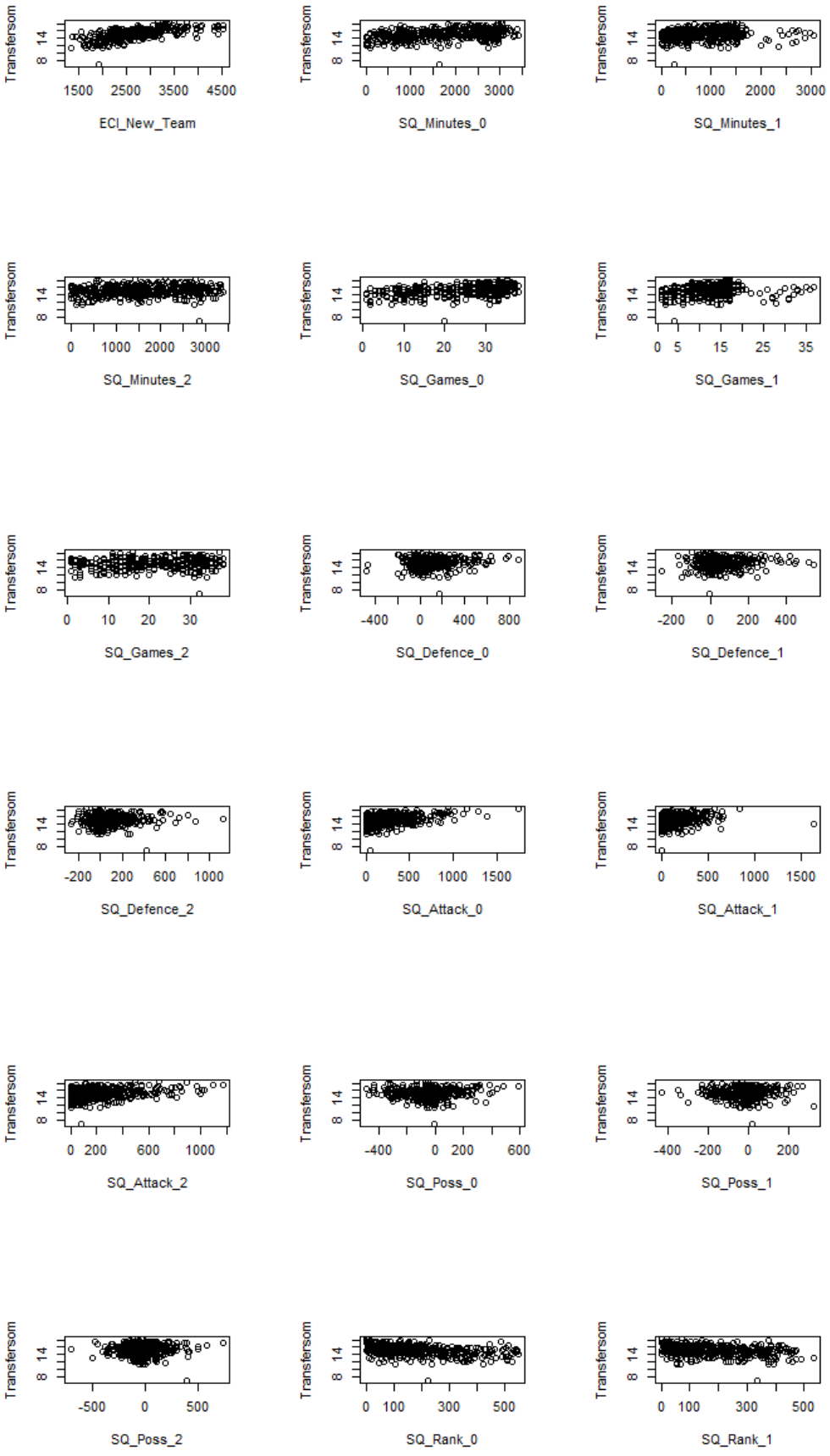


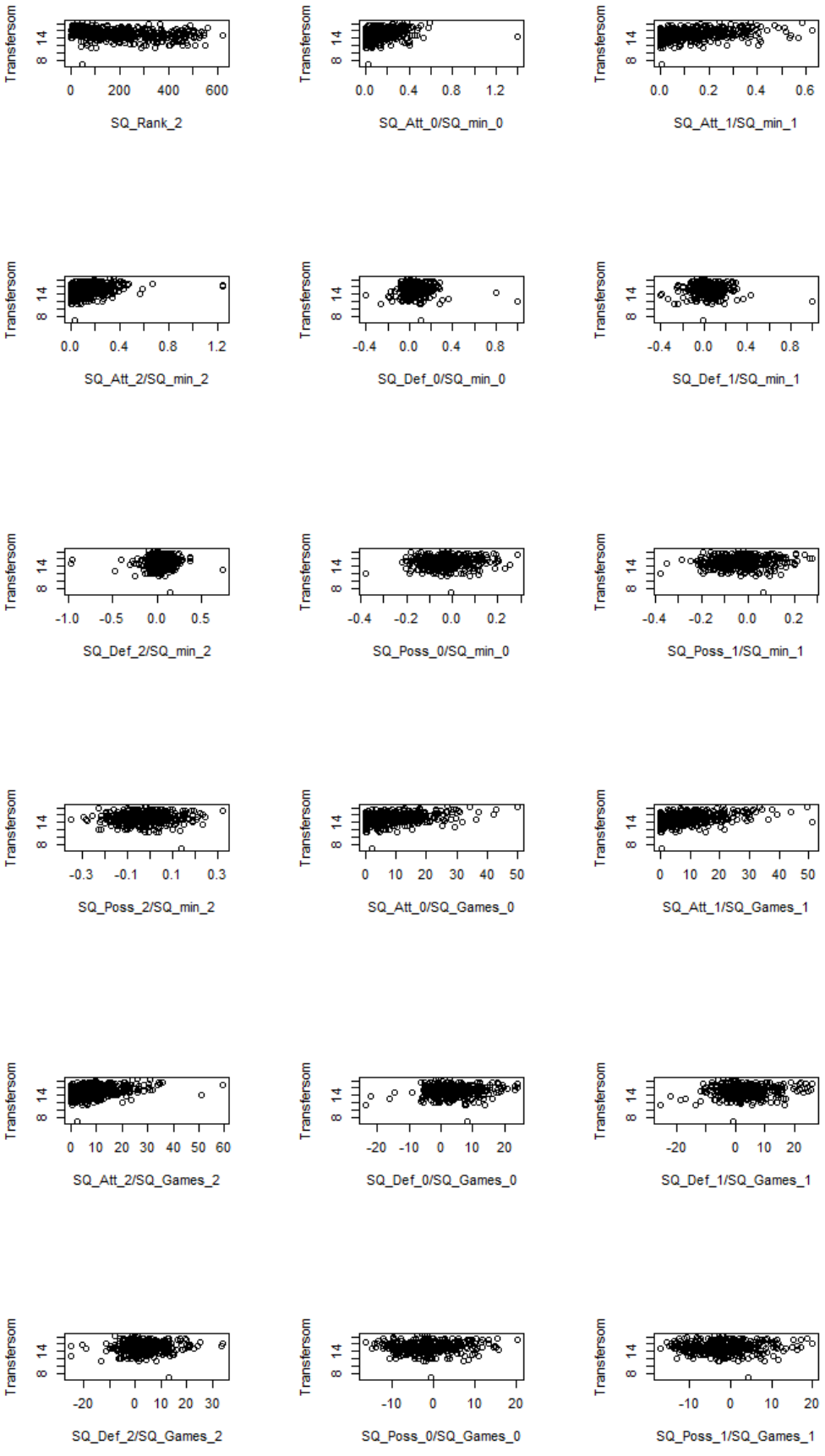


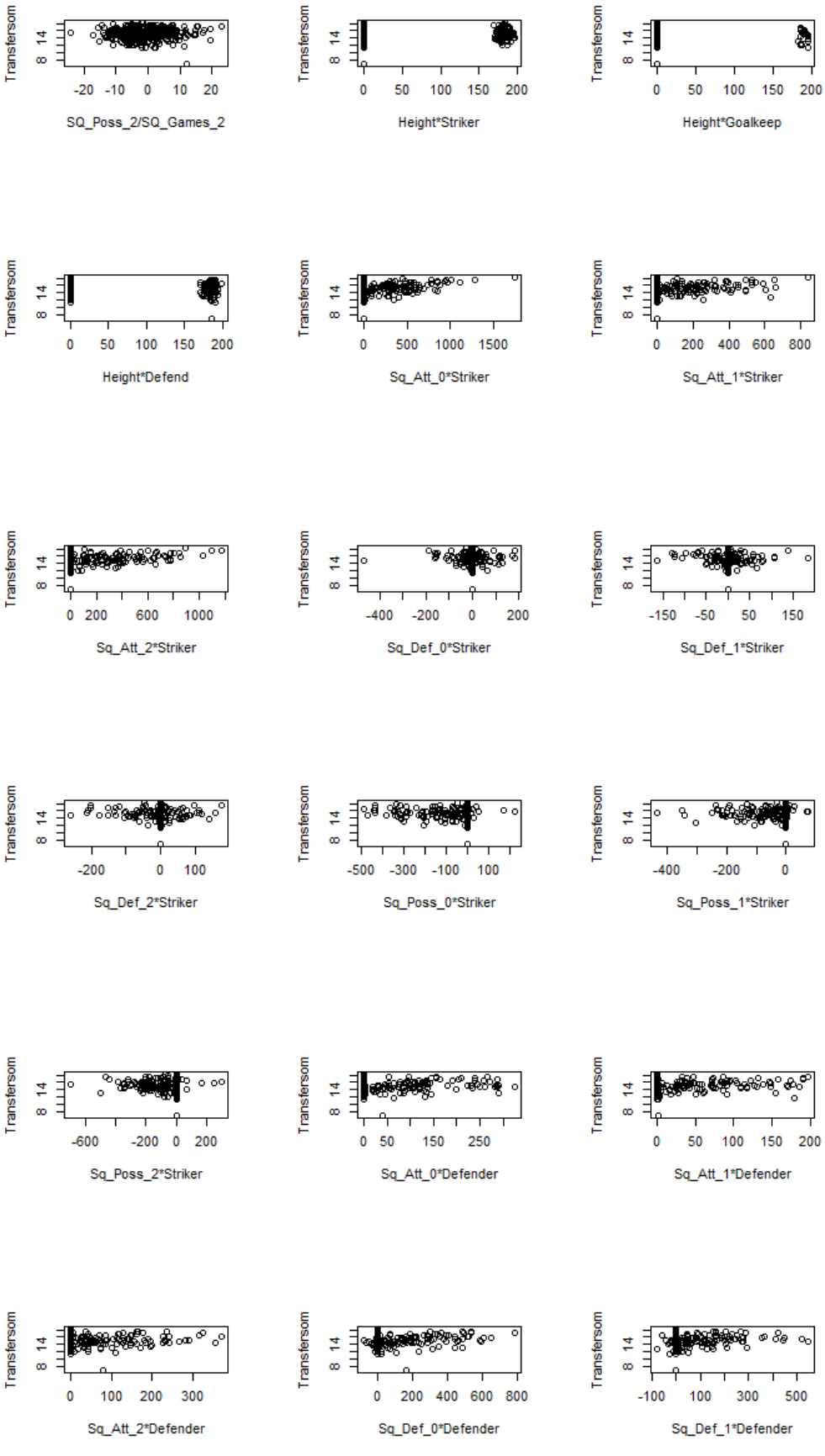


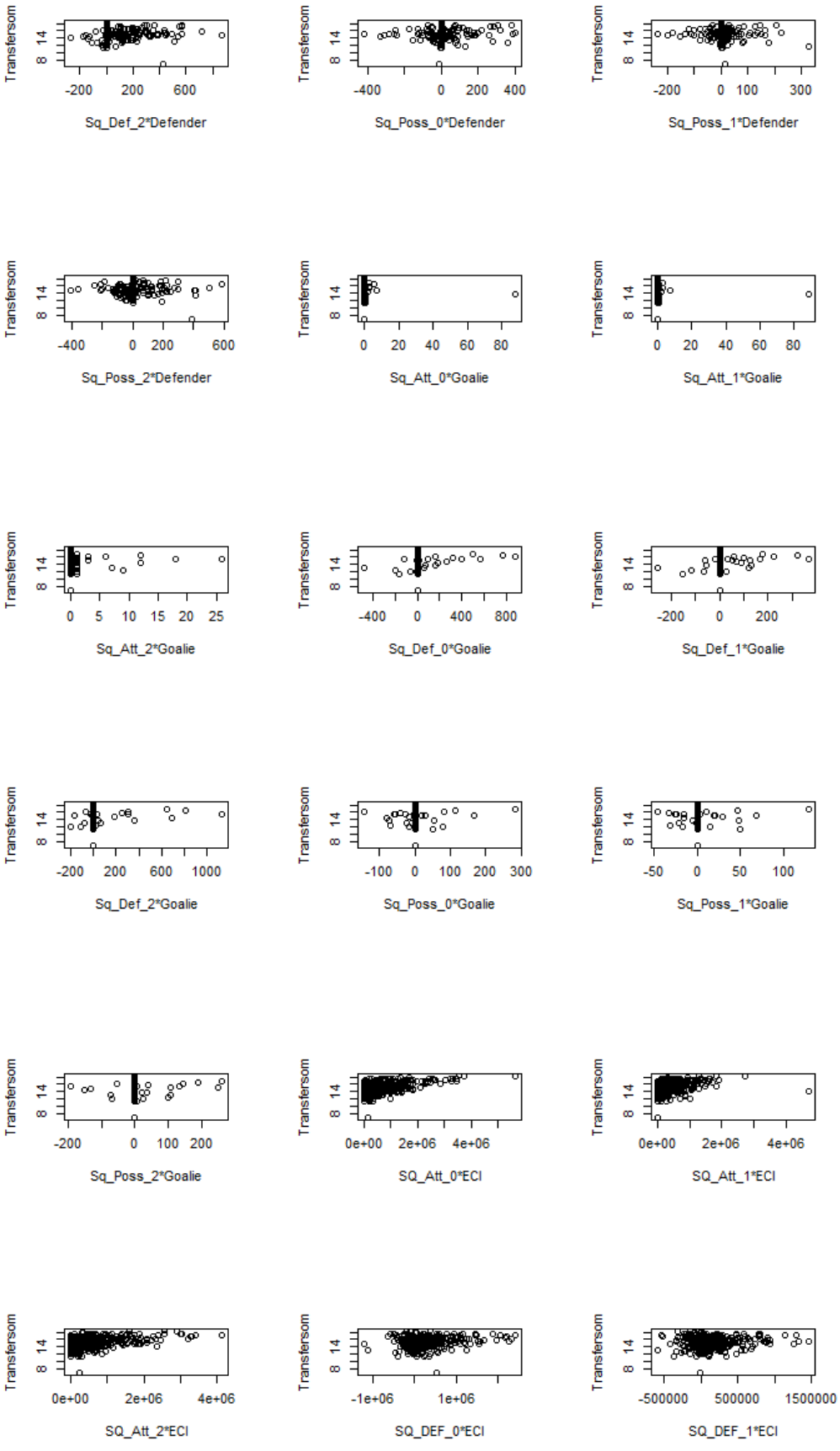


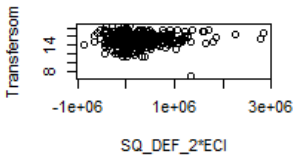


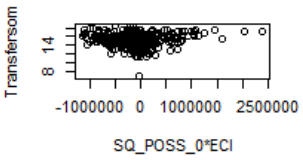

Transfersom
8 14
1500 2500 3500 4500
ECI_New_Team
0 1000 2000 3000
SQ_Minutes_0
0 1000 2000 3000
SQ_Minutes_1
0 1000 2000 3000
SQ_Minutes_2
0 10 20 30
SQ_Games_0
0 5 15 25 35
SQ_Games_1
0 10 20 30
SQ_Games_2
-400 0 400 800
SQ_Defence_0
-200 0 200 400
SQ_Defence_1
-200 200 600 1000
SQ_Defence_2
0 500 1000 1500
SQ_Attack_0
0 500 1000 1500
SQ_Attack_1
0 200 600 1000
SQ_Attack_2
-400 0 200 600
SQ_Poss_0
-400 -200 0 200
SQ_Poss_1
-500 0 500
SQ_Poss_2
0 100 300 500
SQ_Rank_0
0 100 300 500
SQ_Rank_1

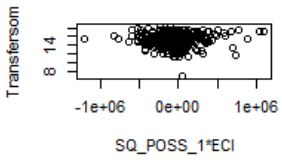

Transfersom
8 14
0 200 400 600
SQ_Rank_2
0.0 0.4 0.8 1.2
SQ_Att_0/SQ_min_0
0.0 0.2 0.4 0.6
SQ_Att_1/SQ_min_1
0.0 0.4 0.8 1.2
SQ_Att_2/SQ_min_2
-0.4 0.0 0.4 0.8
SQ_Def_0/SQ_min_0
-0.4 0.0 0.4 0.8
SQ_Def_1/SQ_min_1
-1.0 -0.5 0.0 0.5
SQ_Def_2/SQ_min_2
-0.4 -0.2 0.0 0.2
SQ_Poss_0/SQ_min_0
-0.4 -0.2 0.0 0.2
SQ_Poss_1/SQ_min_1
-0.3 -0.1 0.1 0.3
SQ_Poss_2/SQ_min_2
0 10 20 30 40 50
SQ_Att_0/SQ_Games_0
0 10 20 30 40 50
SQ_Att_1/SQ_Games_1
0 10 20 30 40 50 60
SQ_Att_2/SQ_Games_2
-20 -10 0 10 20
SQ_Def_0/SQ_Games_0
-20 0 10 20
SQ_Def_1/SQ_Games_1
-20 0 10 20 30
SQ_Def_2/SQ_Games_2
-10 0 10 20
SQ_Poss_0/SQ_Games_0
-10 0 10 20
SQ_Poss_1/SQ_Games_1

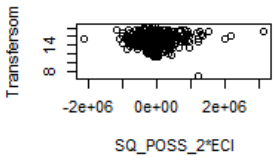

Transfersom
8 14
-20 -10 0 10 20
SQ_Poss_2/SQ_Games_2
0 50 100 150 200
Height*Striker
0 50 100 150 200
Height*Goalkeep
0 50 100 150 200
Height*Defend
0 500 1000 1500
Sq_Att_0*Striker
0 200 400 600 800
Sq_Att_1*Striker
0 200 600 1000
Sq_Att_2*Striker
-400 -200 0 200
Sq_Def_0*Striker
-150 -50 50 150
Sq_Def_1*Striker
-200 0 100
Sq_Def_2*Striker
-500 -300 -100 100
Sq_Poss_0*Striker
-400 -200 0
Sq_Poss_1*Striker
-600 -200 0 200
Sq_Poss_2*Striker
0 50 150 250
Sq_Att_0*Defender
0 50 100 150 200
Sq_Att_1*Defender
0 100 200 300
Sq_Att_2*Defender
0 200 400 600 800
Sq_Def_0*Defender
-100 100 300 500
Sq_Def_1*Defender

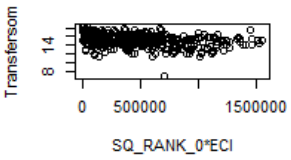

Transfersom
8 14
-200 200 600
Sq_Def_2*Defender
-400 0 200 400
Sq_Poss_0*Defender
-200 0 100 300
Sq_Poss_1*Defender
-400 0 200 600
Sq_Poss_2*Defender
0 20 40 60 80
Sq_Att_0*Goalie
0 20 40 60 80
Sq_Att_1*Goalie
0 5 10 15 20 25
Sq_Att_2*Goalie
-400 0 400 800
Sq_Def_0*Goalie
-200 0 200
Sq_Def_1*Goalie
-200 200 600 1000
Sq_Def_2*Goalie
-100 0 100 200 300
Sq_Poss_0*Goalie
-50 0 50 100
Sq_Poss_1*Goalie
-200 0 100 200
Sq_Poss_2*Goalie
0e+00 2e+06 4e+06
SQ_Att_0*ECI
0e+00 2e+06 4e+06
SQ_Att_1*ECI
0e+00 2e+06 4e+06
SQ_Att_2*ECI
-1e+06 1e+06
SQ_DEF_0*ECI
-500000 500000 1500000
SQ_DEF_1*ECI

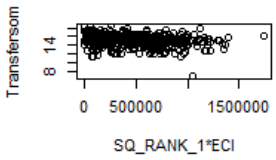

Transfersom
8 14
-1e+06
1e+06
3e+06
SQ_DEF_2*ECI

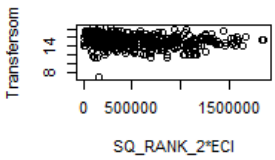

Transfersom
8 14
-1000000
0
1000000
2500000
SQ_POSS_0*ECI

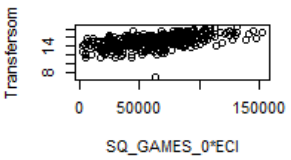

Transfersom
8 14
-1e+06
0e+00
1e+06
SQ_POSS_1*ECI

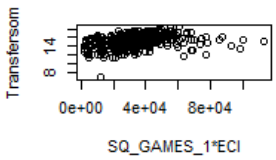

Transfersom
8 14
-2e+06
0e+00
2e+06
SQ_POSS_2*ECI

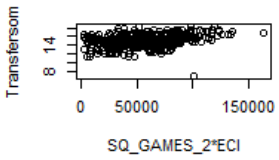

Transfersom
8 14
0
500000
1500000
SQ_RANK_0*ECI

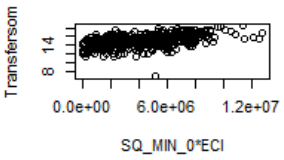

Transfersom
8 14
0
500000
1500000
SQ_RANK_1*ECI

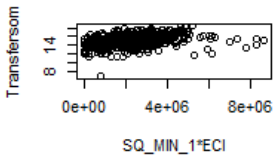

Transfersom
8 14
0
500000
1500000
SQ_RANK_2*ECI

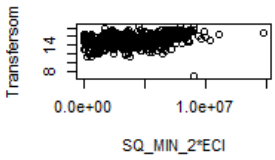

Transfersom
8 14
0
50000
150000
SQ_GAMES_0*ECI


Transfersom
8 14
0e+00
4e+04
8e+04
SQ_GAMES_1*ECI


Transfersom
8 14
0
50000
150000
SQ_GAMES_2*ECI


Transfersom
8 14
0.0e+00
6.0e+06
1.2e+07
SQ_MIN_0*ECI


Transfersom
8 14
0e+00
4e+06
8e+06
SQ_MIN_1*ECI


Transfersom
8 14
0.0e+00
1.0e+07
SQ_MIN_2*ECI

10.3 Bias Variance trade-off

$$
\begin{aligned}
&= \mathrm{E}\,[(\widehat{\theta}\text{ - }\theta)^2] = \mathbb{E}[(\widehat{\theta} - \mathbb{E}[\widehat{\theta}] + \mathbb{E}[\widehat{\theta}] - \theta)^2] \\
&= \mathrm{E}\,[(\widehat{\theta}\text{ - E }[\widehat{\theta}]\,)^2 + 2\cdot(\widehat{\theta} - \mathbb{E}[\widehat{\theta}])\cdot(\mathbb{E}[\widehat{\theta}] - \theta + (\mathbb{E}[\widehat{\theta}] - \theta)^2] \\
&= \mathrm{E}\,[(\widehat{\theta}\text{ - E }[\widehat{\theta}]\,)^2] + \mathbb{E}[2\cdot(\widehat{\theta} - \mathbb{E}[\widehat{\theta}])\cdot(\mathbb{E}[\widehat{\theta}] - \theta] + E[(\mathbb{E}[\widehat{\theta}] - \theta)^2] \\
&= \mathrm{E}\,[(\widehat{\theta}\text{ - E }[\widehat{\theta}]\,)^2] + 2\cdot\mathbb{E}[(\widehat{\theta}] - \theta)\cdot\mathbb{E}[\widehat{\theta} - \mathbb{E}[\widehat{\theta}]] + (\mathbb{E}[\widehat{\theta}] - \theta)^2 \\
&= \mathrm{E}\,[(\widehat{\theta}\text{ - E }[\widehat{\theta}]\,)^2] + 2\cdot\mathbb{E}[(\widehat{\theta}] - \theta)\cdot(\mathbb{E}[\widehat{\theta}] - \mathbb{E}[\widehat{\theta}]) + (\mathbb{E}[\widehat{\theta}] - \theta)^2 \\
&= \mathrm{E}\,[(\widehat{\theta}\text{ - E }[\widehat{\theta}]\,)^2] + (\mathbb{E}[\widehat{\theta}] - \theta)^2 \\
&= \mathrm{Var}(\widehat{\theta}) + \mathrm{Bias}(\widehat{\theta},\theta)^2
\end{aligned}
$$

(26)

10.4 Heteroskedasticity

Heteroskedasticity Test: Breusch-Pagan-Godfrey

F-statistic	3.554886	Prob. F(63,360)	0.0000
Obs*R-squared	162.6113	Prob. Chi-Square(63)	0.0000
Scaled explained SS	631.9624	Prob. Chi-Square(63)	0.0000

Test Equation:
Dependent Variable: RESID^2
Method: Least Squares
Date: 09/13/17 Time: 12:46
Sample: 1 424
Included observations: 424

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	322.0703	172.0953	1.871466	0.0621
YEAR	−0.159263	0.085396	−1.864991	0.0630
AGE_SQUARED	0.000227	0.000424	0.536146	0.5922
GOALKEEPDUMMY	2.657505	12.21959	0.217479	0.8280
MONTHSREMAINING	−0.000767	0.005334	−0.143735	0.8858
ECI_OLD_TEAM	$7.40E-05$	0.000130	0.568489	0.5701
ECI_NEW_TEAM	−0.000361	0.000150	−2.404846	0.0167
SQ_RANK_2	−0.000558	0.000525	−1.063943	0.2881
SQ_ATT_1_SQ_MIN_1	0.051670	0.736014	0.070202	0.9441
HEIGHT_GOALKEEP	−0.013882	0.064613	−0.214855	0.8300
SQ_DEF_1_DEFENDER	−0.000620	0.000898	−0.690338	0.4904
SQ_ATT_0_ECI	$-2.44E-08$	$1.67E-07$	−0.146218	0.8838
SQ_ATT_2_ECI	$-2.14E-07$	$1.67E-07$	−1.277163	0.2024
SQ_DEF_0_ECI	$4.73E-08$	$1.75E-07$	0.270763	0.7867
SQ_GAMES_0_ECI	$1.51E-06$	$3.60E-06$	0.420917	0.6741
SQ_GAMES_1_ECI	$-3.38E-06$	$4.26E-06$	−0.792733	0.4285
SQ_GAMES_2_ECI	$2.12E-06$	$3.09E-06$	0.688340	0.4917
DUMMY_ATALANTA	0.235145	0.562705	0.417882	0.6763
DUMMY_AS_ROMA	0.004557	0.432915	0.010527	0.9916
DUMMY_GRANADA_CF	−0.197777	0.535181	−0.369552	0.7119
DUMMY_LEVANTE_UD	0.064496	0.724745	0.088991	0.9291
DUMMY_FC_NANTES	0.096139	0.635022	0.151394	0.8797
DUMMY_SSC_NAPOLI	3.122411	0.474054	6.586613	0.0000
DUMMY_SC_CAMBUUR	0.124691	0.678602	0.183747	0.8543
DUMMY_BOR__DORTMUND	0.145096	0.467377	0.310447	0.7564
DUMMY_HERACLES_ALMELO	−0.125807	0.598891	−0.210066	0.8337
DUMMY_AFC_AJAX	0.287724	0.437224	0.658071	0.5109
DUMMY_OGC_NICE	0.273387	0.620717	0.440438	0.6599
DUMMY_LAZIO	−0.956614	0.849856	−1.125619	0.2611
DUMMY_SC_HEERENVEEN	−0.408061	0.801701	−0.508995	0.6111
DUMMY_FC_AUGSBURG	−0.143883	0.472203	−0.304706	0.7608

Variable	Coefficient	Std. Error	t-Statistic	Prob.
DUMMY_GETAFE_CF	0.497742	0.822543	0.605125	0.5455
DUMMY_UDINESE_CALCIO	0.033663	0.707873	0.047555	0.9621
DUMMY_HAMBURGER_SV01	−0.474913	0.392121	−1.211140	0.2266
DUMMY_WATFORD01	−0.185915	0.598173	−0.310804	0.7561
DUMMY_NEWCASTLE01	−0.289377	0.576013	−0.502379	0.6157
DUMMY_MIDDLESBROUGH01	−0.594244	0.597259	−0.994952	0.3204
DUMMY_ASTON_VILLA01	−0.359765	0.436115	−0.824932	0.4100
DUMMY_AC_MILAN01	−0.271447	0.490818	−0.553051	0.5806
DUMMY_BESIKTAS01	−0.244279	0.583115	−0.418922	0.6755
DUMMY_FC_BARCELONA01	1.054671	0.507284	2.079057	0.0383
DUMMY_SASSUOLO01	5.271321	0.550037	9.583568	0.0000
DUMMY_MAN_UTD01	0.083523	0.455454	0.183384	0.8546
DUMMY_PARIS_SG01	0.025351	0.498065	0.050899	0.9594
DUMMY_STANDARD_LIEGE01	−0.475391	0.862028	−0.551479	0.5816
DUMMY_BOLOGNA01	−0.308802	0.443699	−0.695971	0.4869
DUMMY_SAINT_ETIENNE01	−0.028051	0.607223	−0.046196	0.9632
DUMMY_FC_K_LAUTERN01	−0.544254	0.812427	−0.669911	0.5033
DUMMY_MAN_CITY01	−0.087907	0.610729	−0.143938	0.8856
DUMMY_SUNDERLAND01	−0.094730	0.674444	−0.140457	0.8884
DUMMY_ATLETICO_MADRID01	0.794040	0.690128	1.150568	0.2507
DUMMY_SWANSEA01	−0.678132	0.476430	−1.423361	0.1555
DUMMY_INTER01	0.260365	0.519318	0.501360	0.6164
DUMMY_LEICESTER01	−0.760180	0.551420	−1.378585	0.1689
DUMMY_ARSENAL01	0.066731	0.581855	0.114686	0.9088
DUMMY_HERTHA_BSC01	−0.454621	0.683823	−0.664822	0.5066
DUMMY_SM_CAEN01	−0.538733	0.662556	−0.813114	0.4167
DUMMY_F__DUSSELDORF01	−0.507956	0.666003	−0.762693	0.4461
DUMMY_REAL_MADRID01	0.629010	0.642128	0.979570	0.3280
DUMMY_SAMPDORIA01	−0.473253	0.732390	−0.646176	0.5186
DUMMY_CRYSTAL_PALACE01	−0.302290	0.834603	−0.362196	0.7174
DUMMY_STOKE_CITY01	−0.175661	0.661973	−0.265360	0.7909
DUMMY_GREUTHER_FURTH01	−0.307818	0.816462	−0.377014	0.7064
DUMMY_RB_LEIPZIG01	−0.502168	0.819001	−0.613147	0.5402

R-squared	0.383517	Mean dependent var	0.399813
Adjusted R-squared	0.275633	S.D. dependent var	1.314371
S.E. of regression	1.118658	Akaike info criterion	3.200395
Sum squared resid	450.5028	Schwarz criterion	3.811675
Log likelihood	−614.4837	Hannan-Quinn criter.	3.441908
F-statistic	3.554886	Durbin-Watson stat	2.018240
Prob(F-statistic)	0.000000		

10.5 Coefficient sensitivy to λ

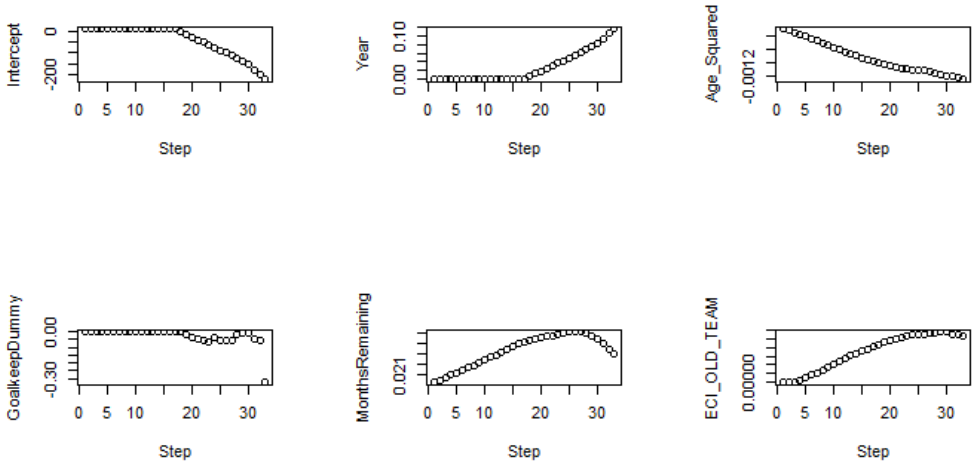


Figure 22: Sensitivity of coefficient sizes to λ

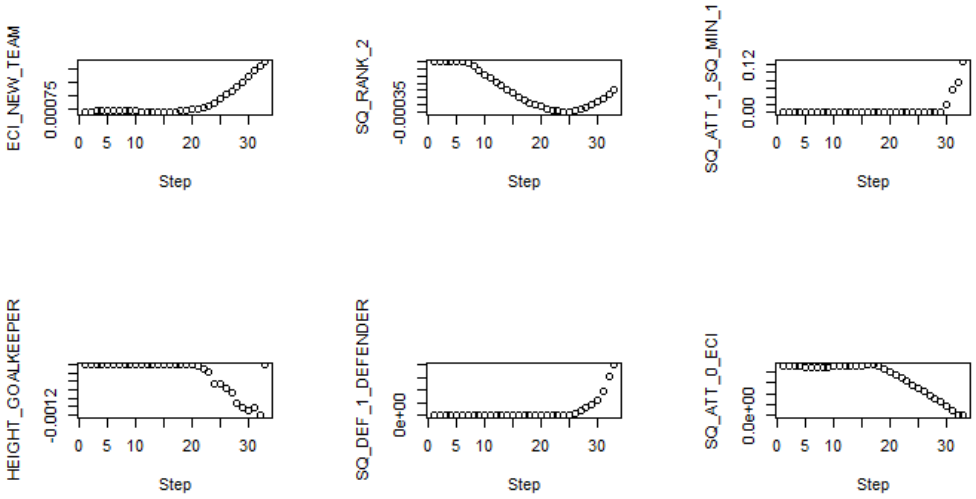


Figure 23: Sensitivity of coefficient sizes to λ

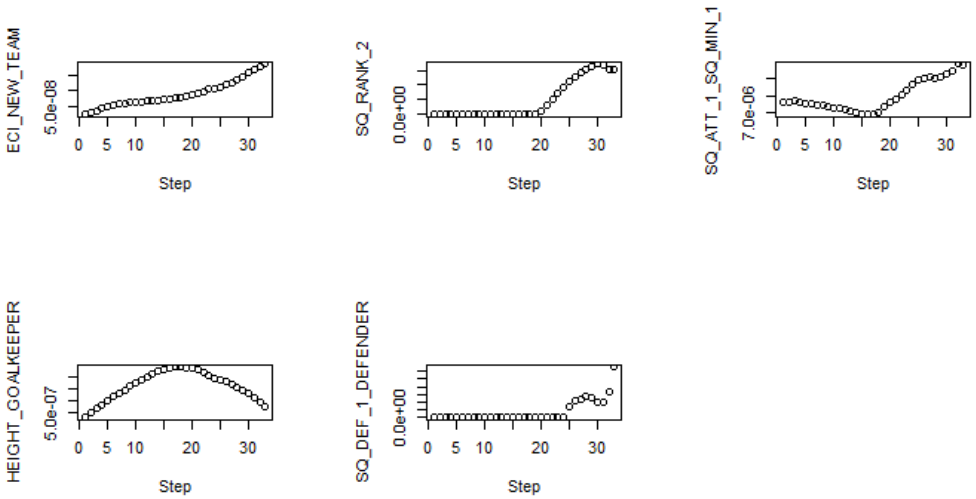
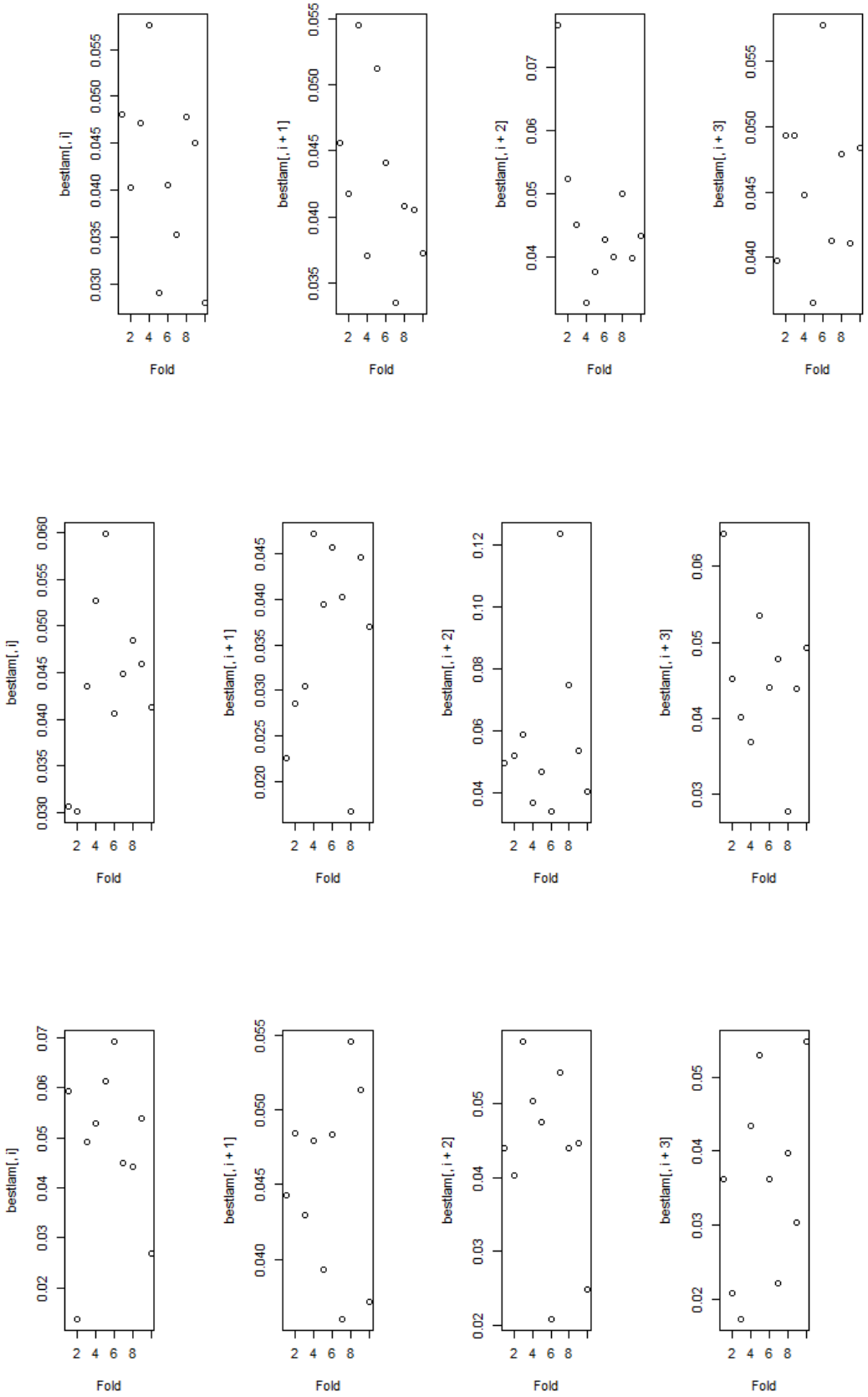


Figure 24: Sensitivity of coefficient sizes to λ

10.6 Lambda Lasso1



bestlam[, i]
bestlam[, i + 1]
bestlam[, i + 2]
bestlam[, i + 3]
Fold

bestlam[, i]
bestlam[, i + 1]
bestlam[, i + 2]
bestlam[, i + 3]
Fold

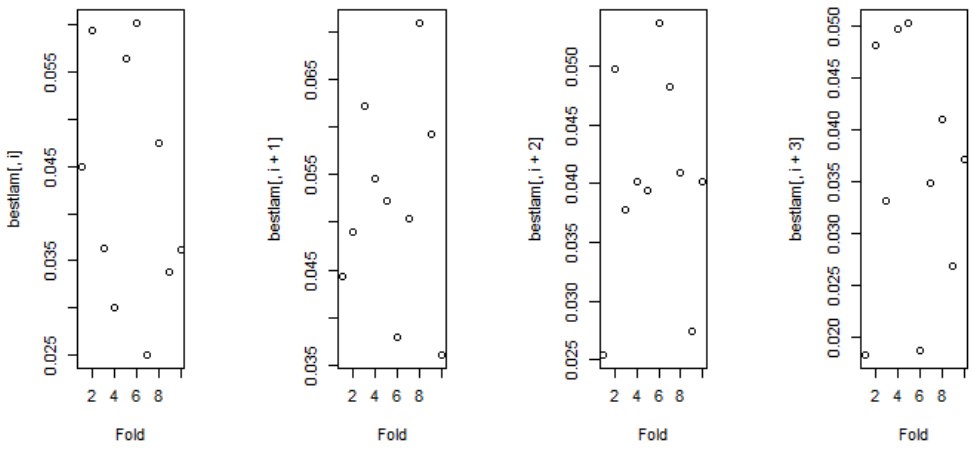
bestlam[, i]
Fold
bestlam[, i + 1]
Fold
bestlam[, i + 2]
Fold
bestlam[, i + 3]
Fold
bestlam[, i]
Fold
bestlam[, i + 1]
Fold
bestlam[, i + 2]
Fold
bestlam[, i + 3]
Fold
bestlam[, i]
Fold
bestlam[, i + 1]
Fold
bestlam[, i + 2]
Fold
bestlam[, i + 3]
Fold

bestlam[, i]
bestlam[, i + 1]
bestlam[, i + 2]
bestlam[, i + 3]
Fold

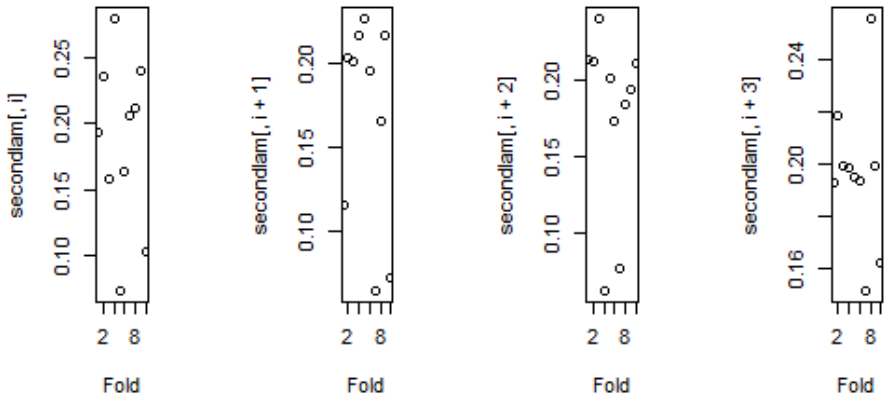
bestlam[, i]
bestlam[, i + 1]
bestlam[, i + 2]
bestlam[, i + 3]
Fold

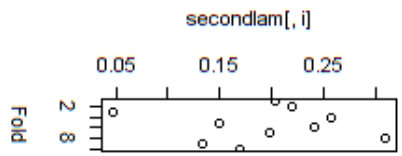
bestlam[, i]
bestlam[, i + 1]
bestlam[, i + 2]
bestlam[, i + 3]
Fold

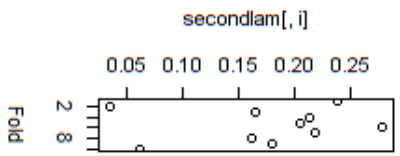
bestlam[, i]
bestlam[, i + 1]
bestlam[, i + 2]
bestlam[, i + 3]
Fold

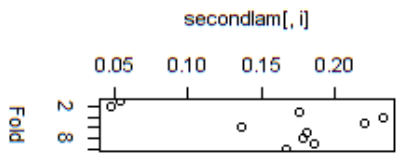


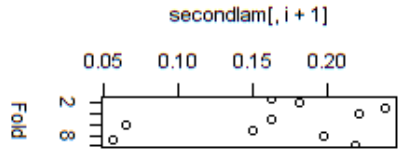
10.7 Lambda Lasso2

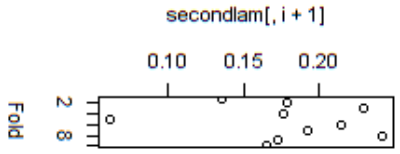


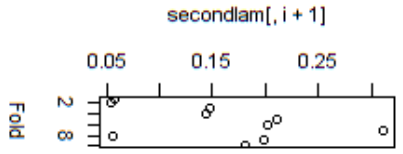

secondlam[, i]
0.05
0.15
0.25
Fold
2
8

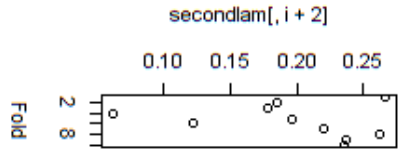

secondlam[, i]
0.05
0.10
0.15
0.20
0.25
Fold
2
8

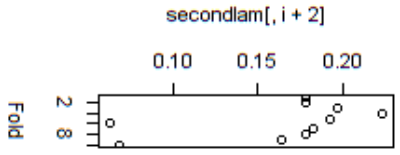

secondlam[, i]
0.05
0.10
0.15
0.20
Fold
2
8

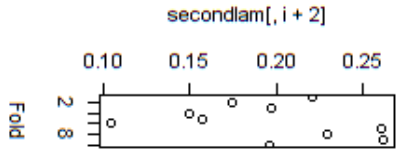

secondlam[, i + 1]
0.05
0.10
0.15
0.20
Fold
2
8

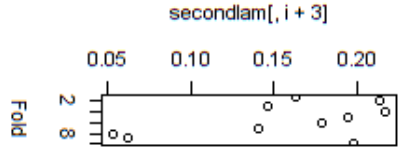

secondlam[, i + 1]
0.10
0.15
0.20
Fold
2
8

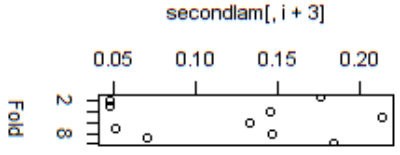

secondlam[, i + 1]
0.05
0.15
0.25
Fold
2
8

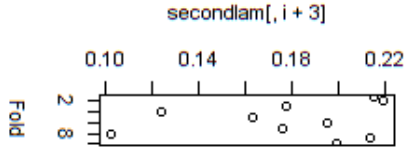

secondlam[, i + 2]
0.10
0.15
0.20
0.25
Fold
2
8

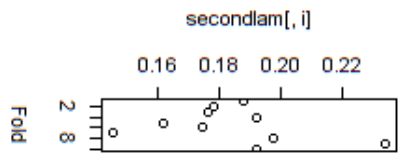

secondlam[, i + 2]
0.10
0.15
0.20
Fold
2
8

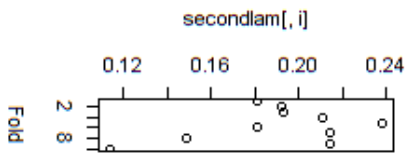

secondlam[, i + 2]
0.10
0.15
0.20
0.25
Fold
2
8

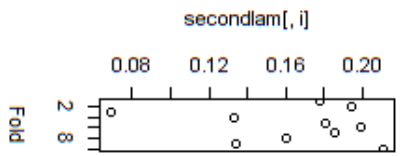

secondlam[, i + 3]
0.05
0.10
0.15
0.20
Fold
2
8

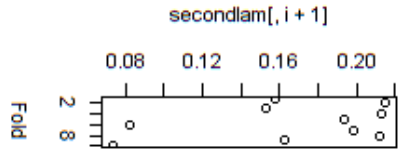

secondlam[, i + 3]
0.05
0.10
0.15
0.20
Fold
2
8

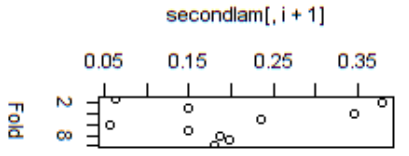

secondlam[, i + 3]
0.10
0.14
0.18
0.22
Fold
2
8

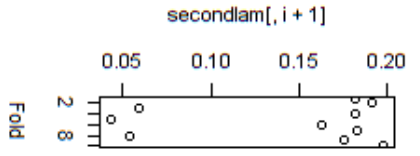

secondlam[, i]
0.16 0.18 0.20 0.22
Fold
2 8

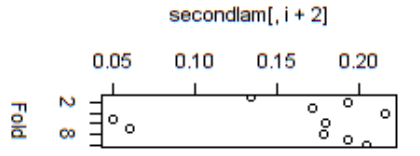

secondlam[, i]
0.12 0.16 0.20 0.24
Fold
2 8

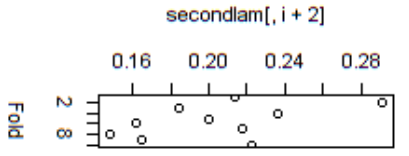

secondlam[, i]
0.08 0.12 0.16 0.20
Fold
2 8

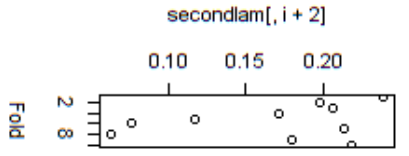

secondlam[, i + 1]
0.08 0.12 0.16 0.20
Fold
2 8

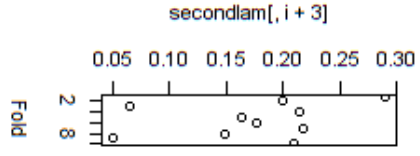

secondlam[, i + 1]
0.05 0.15 0.25 0.35
Fold
2 8

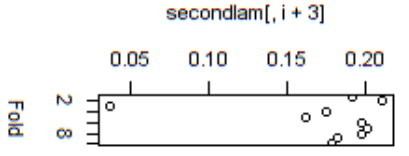

secondlam[, i + 1]
0.05 0.10 0.15 0.20
Fold
2 8

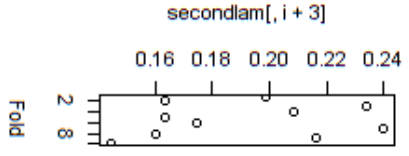

secondlam[, i + 2]
0.05 0.10 0.15 0.20
Fold
2 8

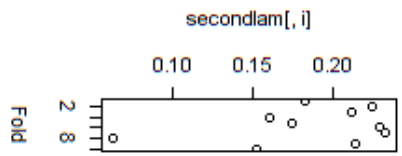

secondlam[, i + 2]
0.16 0.20 0.24 0.28
Fold
2 8

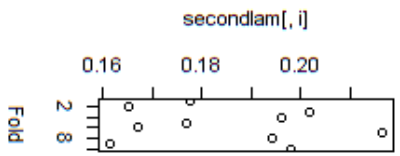

secondlam[, i + 2]
0.10 0.15 0.20
Fold
2 8

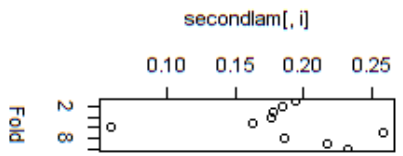

secondlam[, i + 3]
0.05 0.10 0.15 0.20 0.25 0.30
Fold
2 8

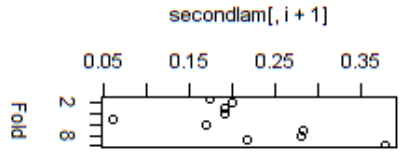

secondlam[, i + 3]
0.05 0.10 0.15 0.20
Fold
2 8

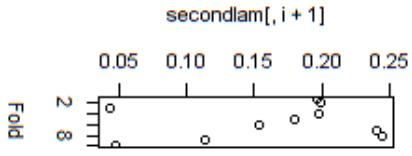

secondlam[, i + 3]
0.16 0.18 0.20 0.22 0.24
Fold
2 8

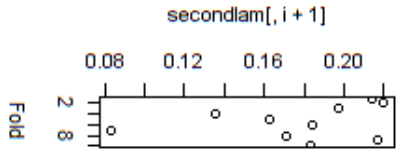

secondlam[, i]
0.10
0.15
0.20
Fold
2
8

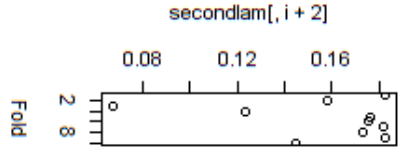

secondlam[, i]
0.16
0.18
0.20
Fold
2
8

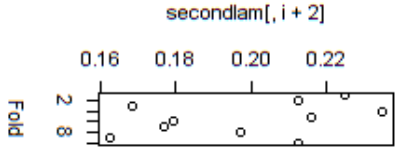

secondlam[, i]
0.10
0.15
0.20
0.25
Fold
2
8

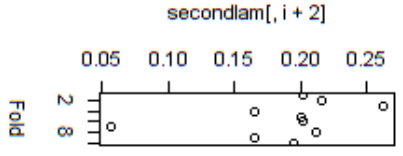

secondlam[, i + 1]
0.05
0.15
0.25
0.35
Fold
2
8

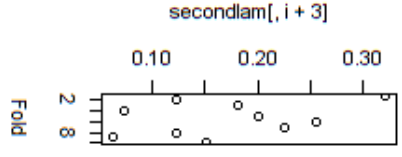

secondlam[, i + 1]
0.05
0.10
0.15
0.20
0.25
Fold
2
8

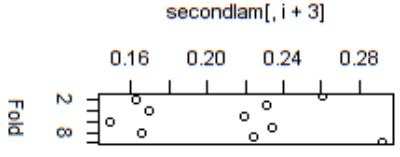

secondlam[, i + 1]
0.08
0.12
0.16
0.20
Fold
2
8

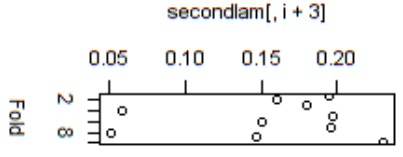

secondlam[, i + 2]
0.08
0.12
0.16
Fold
2
8

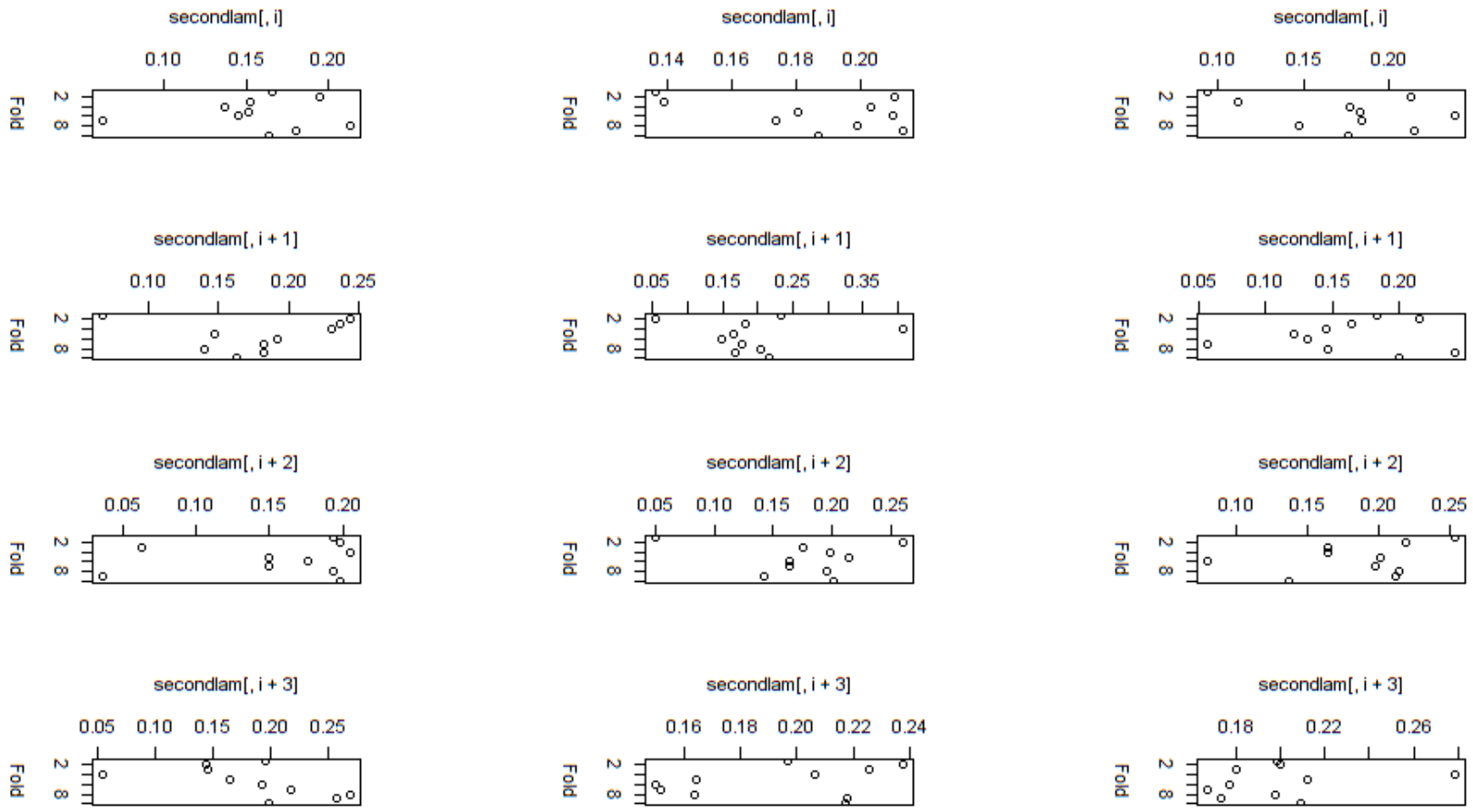

secondlam[, i + 2]
0.16
0.18
0.20
0.22
Fold
2
8

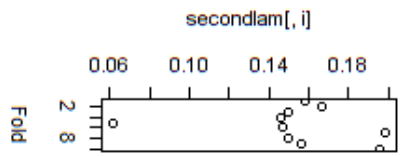

secondlam[, i + 2]
0.05
0.10
0.15
0.20
0.25
Fold
2
8

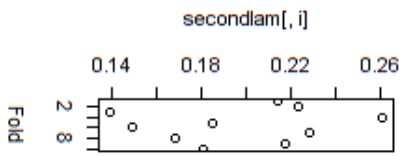

secondlam[, i + 3]
0.10
0.20
0.30
Fold
2
8

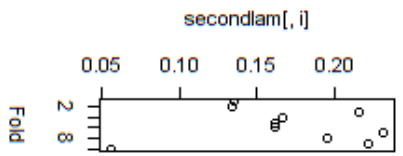

secondlam[, i + 3]
0.16
0.20
0.24
0.28
Fold
2
8

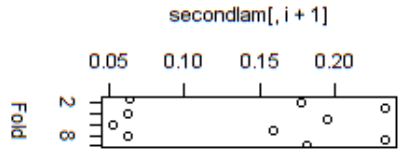

secondlam[, i + 3]
0.05
0.10
0.15
0.20
Fold
2
8

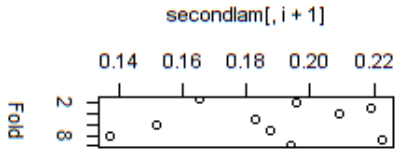

secondlam[, i]
0.10 0.15 0.20
Fold
2 8
secondlam[, i]
0.14 0.16 0.18 0.20
Fold
2 8
secondlam[, i]
0.10 0.15 0.20
Fold
2 8
secondlam[, i + 1]
0.10 0.15 0.20 0.25
Fold
2 8
secondlam[, i + 1]
0.05 0.15 0.25 0.35
Fold
2 8
secondlam[, i + 1]
0.05 0.10 0.15 0.20
Fold
2 8
secondlam[, i + 2]
0.05 0.10 0.15 0.20
Fold
2 8
secondlam[, i + 2]
0.05 0.10 0.15 0.20 0.25
Fold
2 8
secondlam[, i + 2]
0.10 0.15 0.20 0.25
Fold
2 8
secondlam[, i + 3]
0.05 0.10 0.15 0.20 0.25
Fold
2 8
secondlam[, i + 3]
0.16 0.18 0.20 0.22 0.24
Fold
2 8
secondlam[, i + 3]
0.18 0.22 0.26
Fold
2 8

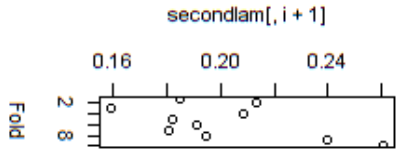

secondlam[, i]
0.06
0.10
0.14
0.18
Fold
2
8


secondlam[, i]
0.14
0.18
0.22
0.26
Fold
2
8

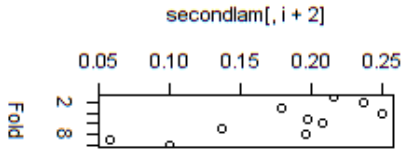

secondlam[, i]
0.05
0.10
0.15
0.20
Fold
2
8

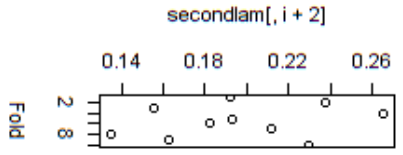

secondlam[, i + 1]
0.05
0.10
0.15
0.20
Fold
2
8

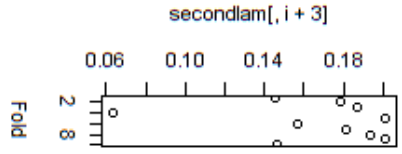

secondlam[, i + 1]
0.14
0.16
0.18
0.20
0.22
Fold
2
8

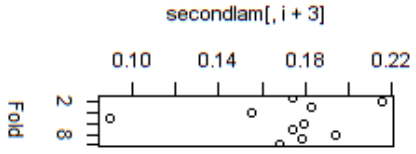

secondlam[, i + 1]
0.16
0.20
0.24
Fold
2
8

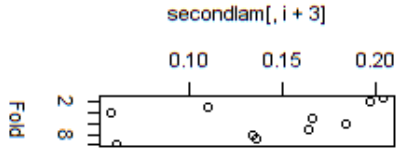

secondlam[, i + 2]
0.14
0.16
0.18
0.20
0.22
Fold
2
8

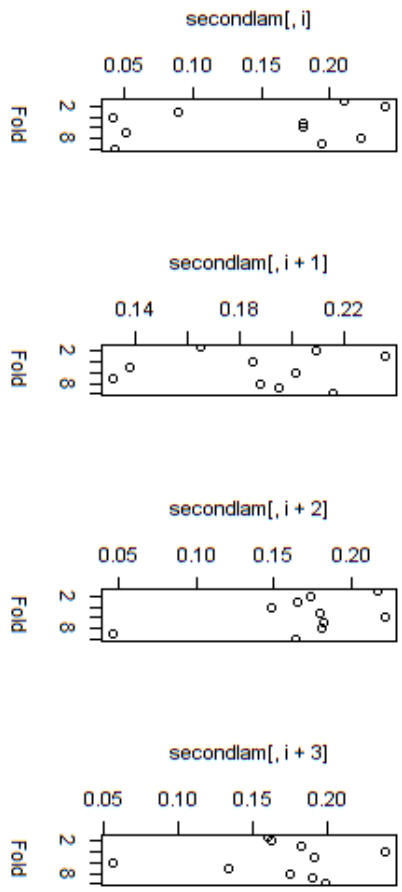

secondlam[, i + 2]
0.05
0.10
0.15
0.20
0.25
Fold
2
8

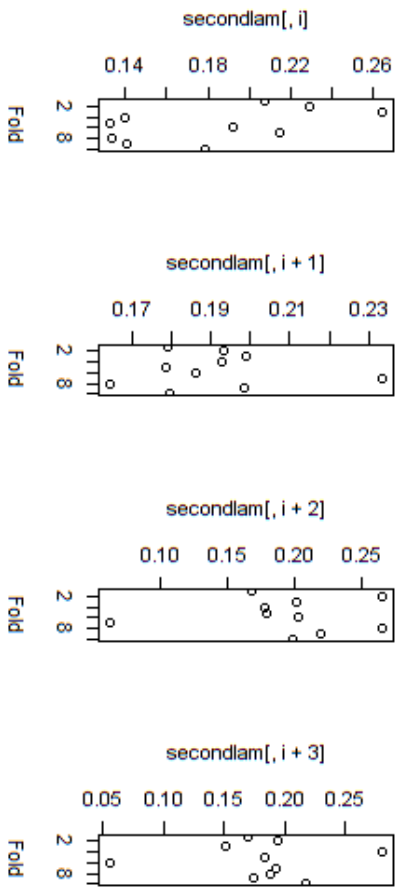

secondlam[, i + 2]
0.14
0.18
0.22
0.26
Fold
2
8

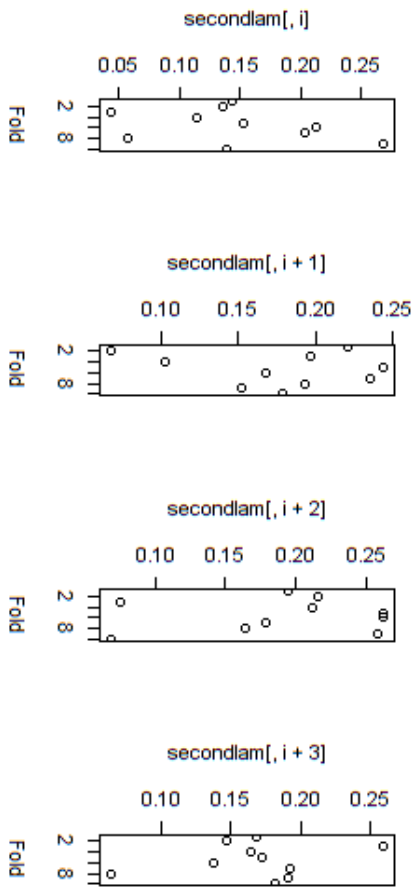

secondlam[, i + 3]
0.06
0.10
0.14
0.18
Fold
2
8


secondlam[, i + 3]
0.10
0.14
0.18
0.22
Fold
2
8


secondlam[, i + 3]
0.10
0.15
0.20
Fold
2
8


secondlam[, i]
0.05 0.10 0.15 0.20
Fold
2 8


secondlam[, i]
0.14 0.18 0.22 0.26
Fold
2 8


secondlam[, i]
0.05 0.10 0.15 0.20 0.25
Fold
2 8

secondlam[, i + 1]
0.14 0.18 0.22
Fold
2 8

secondlam[, i + 1]
0.17 0.19 0.21 0.23
Fold
2 8

secondlam[, i + 1]
0.10 0.15 0.20 0.25
Fold
2 8

secondlam[, i + 2]
0.05 0.10 0.15 0.20
Fold
2 8

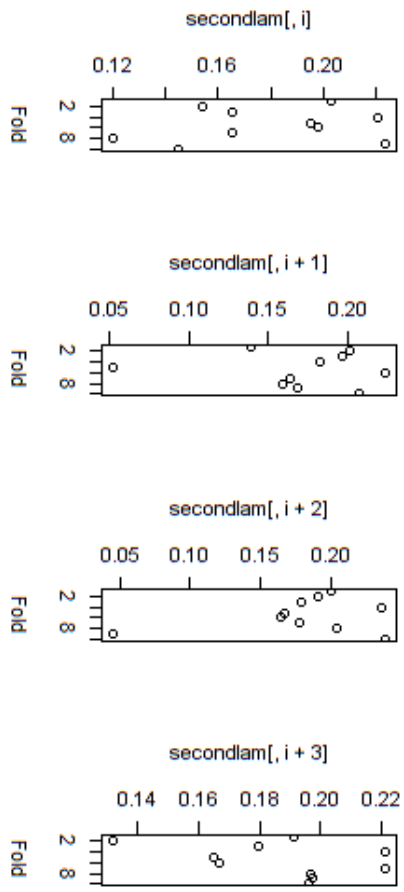
secondlam[, i + 2]
0.10 0.15 0.20 0.25
Fold
2 8

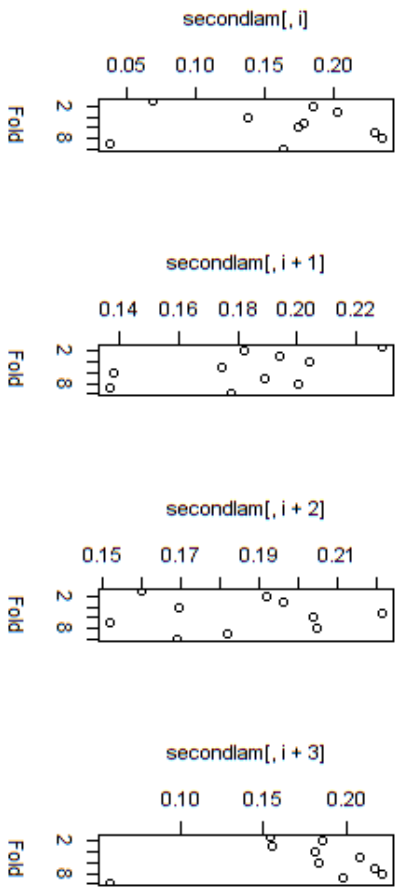
secondlam[, i + 2]
0.10 0.15 0.20 0.25
Fold
2 8

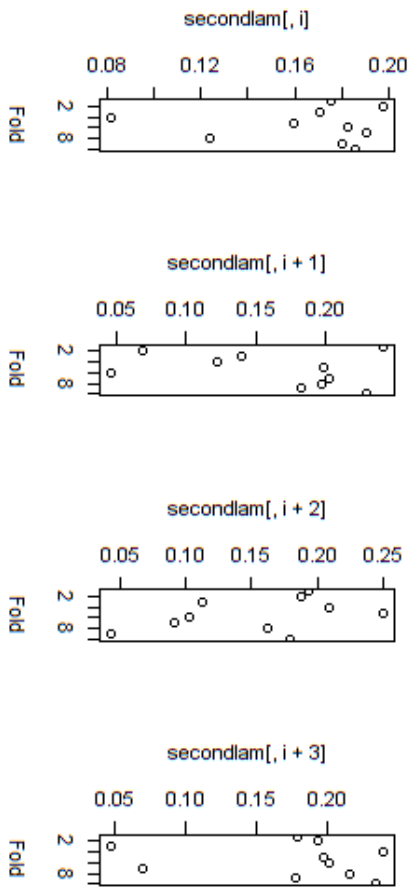
secondlam[, i + 3]
0.05 0.10 0.15 0.20
Fold
2 8

secondlam[, i + 3]
0.05 0.10 0.15 0.20 0.25
Fold
2 8

secondlam[, i + 3]
0.10 0.15 0.20 0.25
Fold
2 8


secondlam[, i]
0.12
0.16
0.20
Fold
2
8


secondlam[, i]
0.05
0.10
0.15
0.20
Fold
2
8


secondlam[, i]
0.08
0.12
0.16
0.20
Fold
2
8

secondlam[, i + 1]
0.05
0.10
0.15
0.20
Fold
2
8

secondlam[, i + 1]
0.14
0.16
0.18
0.20
0.22
Fold
2
8

secondlam[, i + 1]
0.05
0.10
0.15
0.20
Fold
2
8

secondlam[, i + 2]
0.05
0.10
0.15
0.20
Fold
2
8

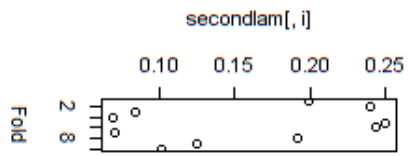
secondlam[, i + 2]
0.15
0.17
0.19
0.21
Fold
2
8

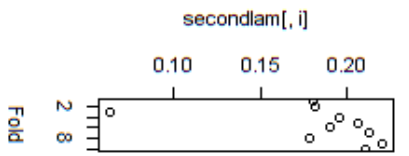
secondlam[, i + 2]
0.05
0.10
0.15
0.20
0.25
Fold
2
8

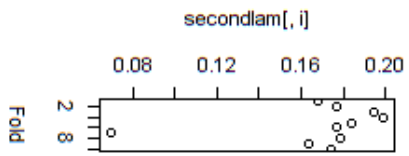
secondlam[, i + 3]
0.14
0.16
0.18
0.20
0.22
Fold
2
8

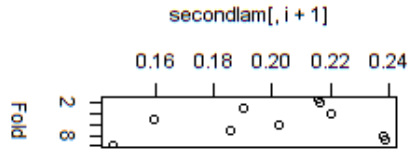
secondlam[, i + 3]
0.10
0.15
0.20
Fold
2
8

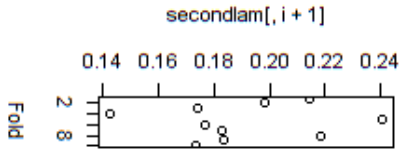
secondlam[, i + 3]
0.05
0.10
0.15
0.20
Fold
2
8

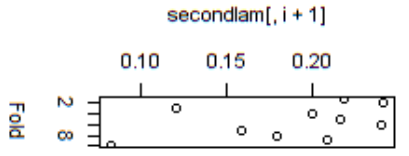

secondlam[, i]
0.10
0.15
0.20
0.25
Fold
2
8

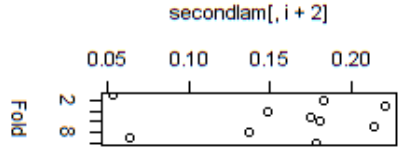

secondlam[, i]
0.10
0.15
0.20
Fold
2
8

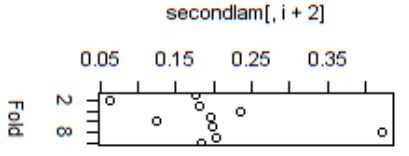

secondlam[, i]
0.08
0.12
0.16
0.20
Fold
2
8

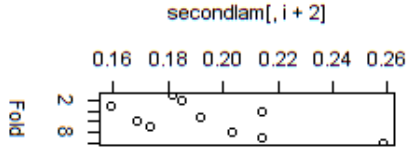

secondlam[, i + 1]
0.16
0.18
0.20
0.22
0.24
Fold
2
8

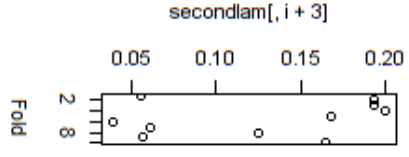

secondlam[, i + 1]
0.14
0.16
0.18
0.20
0.22
0.24
Fold
2
8

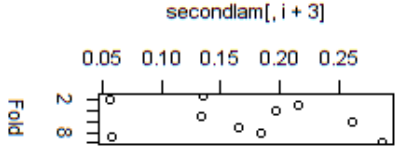

secondlam[, i + 1]
0.10
0.15
0.20
Fold
2
8

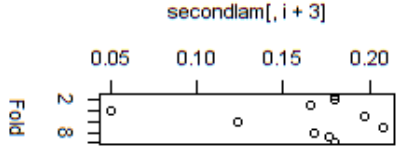

secondlam[, i + 2]
0.05
0.10
0.15
0.20
Fold
2
8


secondlam[, i + 2]
0.05
0.15
0.25
0.35
Fold
2
8

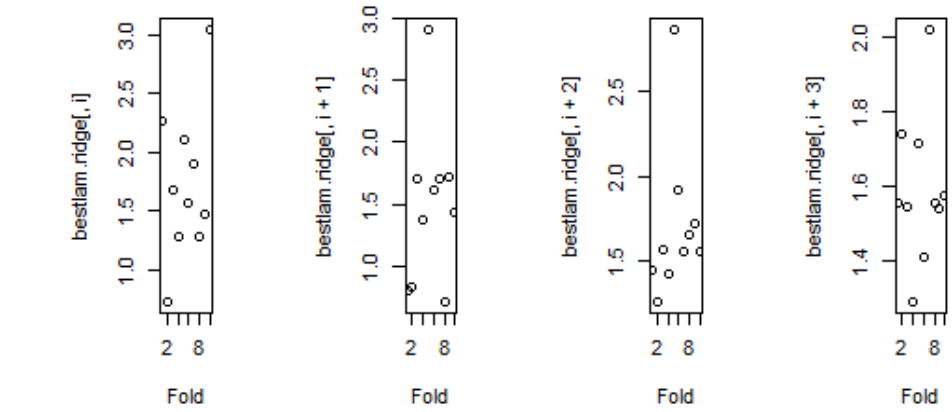

secondlam[, i + 2]
0.16
0.18
0.20
0.22
0.24
0.26
Fold
2
8

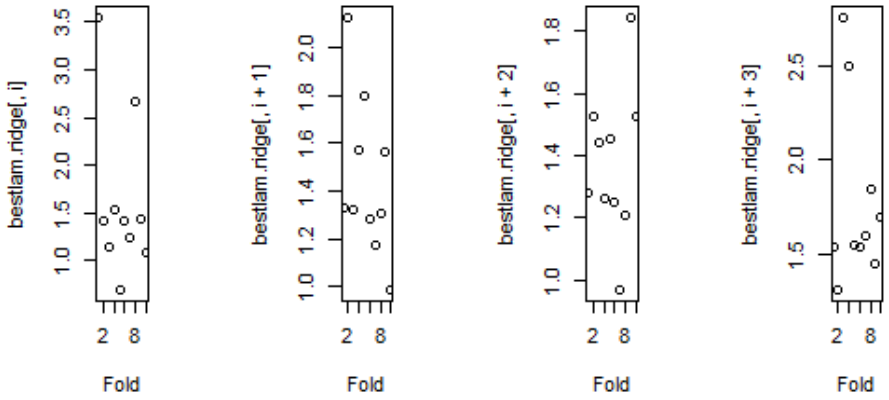

secondlam[, i + 3]
0.05
0.10
0.15
0.20
Fold
2
8

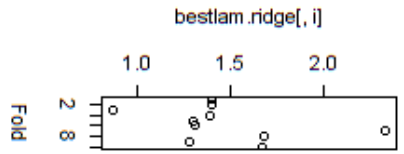

secondlam[, i + 3]
0.05
0.10
0.15
0.20
0.25
Fold
2
8


secondlam[, i + 3]
0.05
0.10
0.15
0.20
Fold
2
8

10.8 Lambda Ridge1

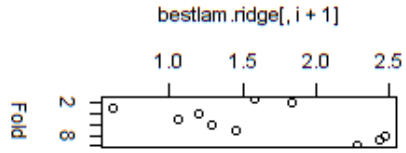


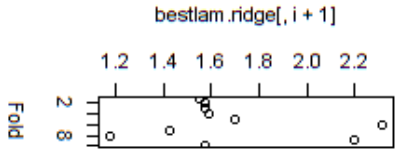



bestlam.ridge[, i]
1.0
1.5
2.0
Fold
2
8

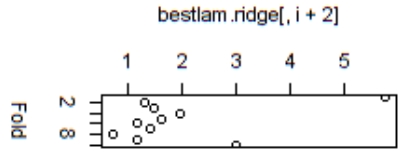

bestlam.ridge[, i]
1.2
1.4
1.6
1.8
2.0
2.2
Fold
2
8

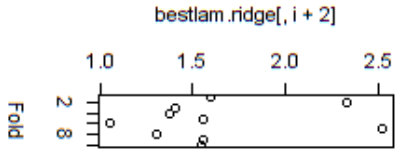

bestlam.ridge[, i]
1.5
2.0
2.5
Fold
2
8


bestlam.ridge[, i + 1]
1.0
1.5
2.0
2.5
Fold
2
8


bestlam.ridge[, i + 1]
1.2
1.4
1.6
1.8
2.0
2.2
Fold
2
8

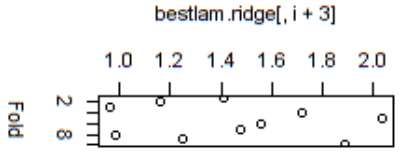

bestlam.ridge[, i + 1]
1.4
1.8
2.2
2.6
Fold
2
8


bestlam.ridge[, i + 2]
1
2
3
4
5
Fold
2
8

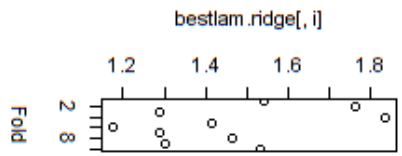

bestlam.ridge[, i + 2]
1.0
1.5
2.0
2.5
Fold
2
8

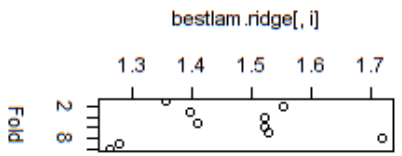

bestlam.ridge[, i + 2]
1.0
1.5
2.0
2.5
Fold
2
8

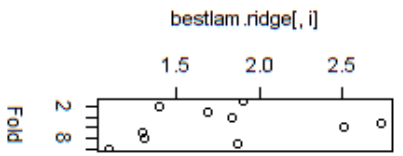

bestlam.ridge[, i + 3]
1.5
2.0
2.5
3.0
Fold
2
8


bestlam.ridge[, i + 3]
1.0
1.2
1.4
1.6
1.8
2.0
Fold
2
8

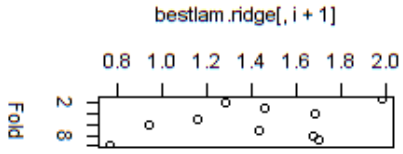

bestlam.ridge[, i + 3]
1.0
1.5
2.0
Fold
2
8


bestlam.ridge[, i]
1.2
1.4
1.6
1.8
Fold
2
8

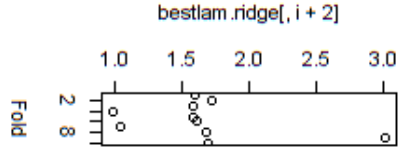

bestlam.ridge[, i]
1.3
1.4
1.5
1.6
1.7
Fold
2
8

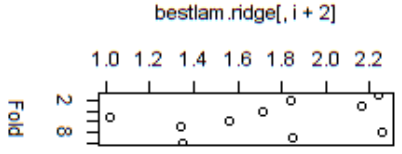

bestlam.ridge[, i]
1.5
2.0
2.5
Fold
2
8


bestlam.ridge[, i + 1]
1.1
1.2
1.3
1.4
1.5
1.6
1.7
Fold
2
8


bestlam.ridge[, i + 1]
0.8
1.0
1.2
1.4
1.6
1.8
2.0
Fold
2
8


bestlam.ridge[, i + 1]
1.0
1.2
1.4
1.6
1.8
2.0
Fold
2
8

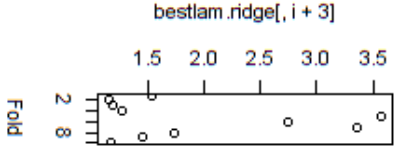

bestlam.ridge[, i + 2]
1.0
1.5
2.0
2.5
3.0
Fold
2
8

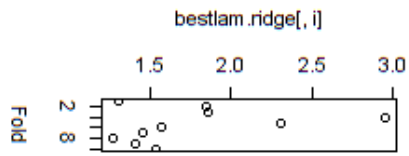

bestlam.ridge[, i + 2]
1.0
1.2
1.4
1.6
1.8
2.0
2.2
Fold
2
8

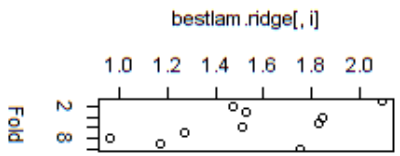

bestlam.ridge[, i + 2]
1.5
2.0
2.5
3.0
3.5
4.0
Fold
2
8

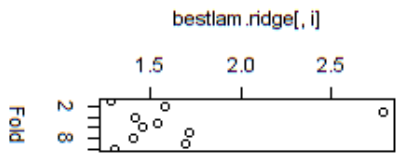

bestlam.ridge[, i + 3]
1.0
1.5
2.0
2.5
Fold
2
8


bestlam.ridge[, i + 3]
1.5
2.0
2.5
3.0
Fold
2
8


bestlam.ridge[, i + 3]
1.5
2.0
2.5
3.0
3.5
Fold
2
8

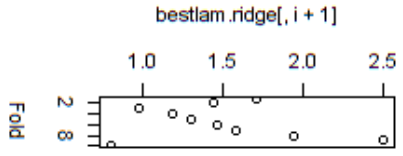

bestlam.ridge[, i]
1.5 2.0 2.5 3.0
Fold
2 8

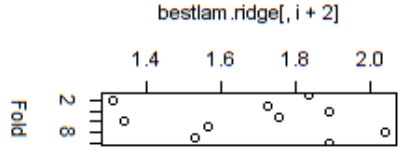

bestlam.ridge[, i]
1.0 1.2 1.4 1.6 1.8 2.0
Fold
2 8

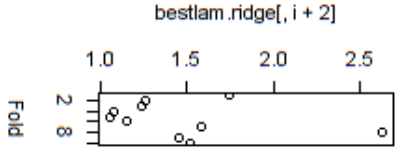

bestlam.ridge[, i]
1.5 2.0 2.5
Fold
2 8


bestlam.ridge[, i + 1]
1.5 2.0 2.5 3.0 3.5
Fold
2 8

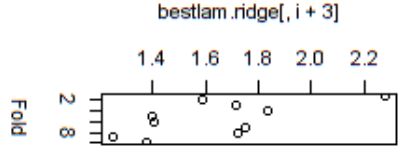

bestlam.ridge[, i + 1]
1.0 1.5 2.0 2.5 3.0
Fold
2 8


bestlam.ridge[, i + 1]
1.0 1.5 2.0 2.5
Fold
2 8


bestlam.ridge[, i + 2]
1.4 1.6 1.8 2.0
Fold
2 8

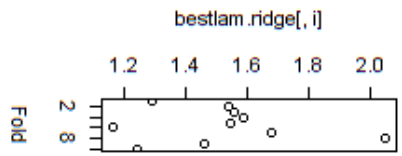

bestlam.ridge[, i + 2]
1.0 1.5 2.0 2.5
Fold
2 8

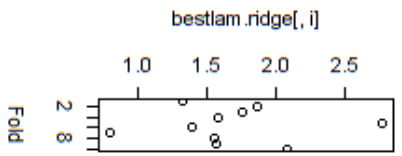

bestlam.ridge[, i + 2]
1.2 1.4 1.6 1.8
Fold
2 8

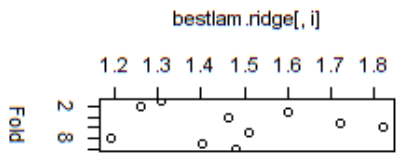

bestlam.ridge[, i + 3]
1.4 1.6 1.8 2.0 2.2
Fold
2 8


bestlam.ridge[, i + 3]
1.0 1.2 1.4 1.6 1.8 2.0
Fold
2 8

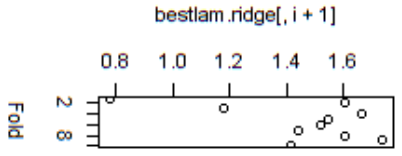

bestlam.ridge[, i + 3]
1.5 2.0 2.5
Fold
2 8


bestlam.ridge[, i]
1.2 1.4 1.6 1.8 2.0
Fold
2 8

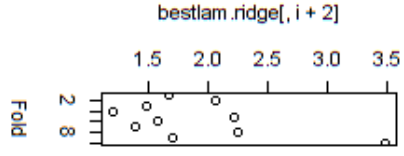

bestlam.ridge[, i]
1.0 1.5 2.0 2.5
Fold
2 8

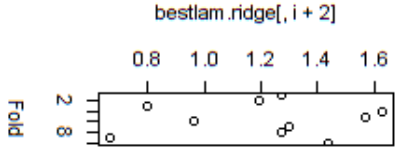

bestlam.ridge[, i]
1.2 1.3 1.4 1.5 1.6 1.7 1.8
Fold
2 8


bestlam.ridge[, i + 1]
1.5 2.0 2.5 3.0 3.5
Fold
2 8

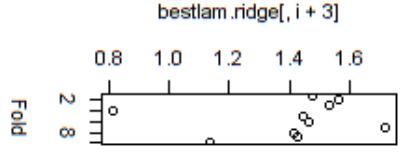

bestlam.ridge[, i + 1]
0.8 1.0 1.2 1.4 1.6
Fold
2 8

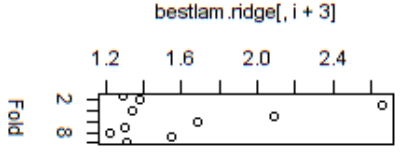

bestlam.ridge[, i + 1]
1.2 1.4 1.6 1.8 2.0 2.2
Fold
2 8

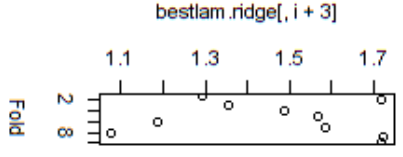

bestlam.ridge[, i + 2]
1.5 2.0 2.5 3.0 3.5
Fold
2 8

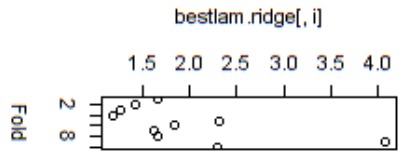

bestlam.ridge[, i + 2]
0.8 1.0 1.2 1.4 1.6
Fold
2 8


bestlam.ridge[, i + 2]
1.4 1.6 1.8 2.0 2.2
Fold
2 8


bestlam.ridge[, i + 3]
0.8 1.0 1.2 1.4 1.6
Fold
2 8

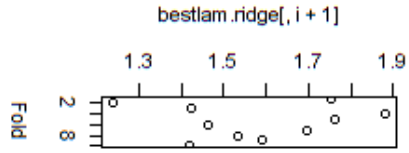

bestlam.ridge[, i + 3]
1.2 1.6 2.0 2.4
Fold
2 8


bestlam.ridge[, i + 3]
1.1 1.3 1.5 1.7
Fold
2 8


bestlam.ridge[, i]
1.5 2.0 2.5 3.0 3.5 4.0
Fold
2
8


bestlam.ridge[, i]
1.5 2.0 2.5 3.0
Fold
2
8


bestlam.ridge[, i]
1.6 2.0 2.4 2.8
Fold
2
8

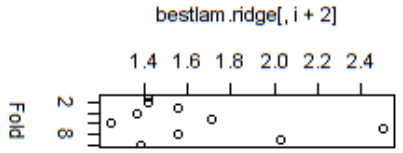

bestlam.ridge[, i + 1]
1.3 1.5 1.7 1.9
Fold
2
8

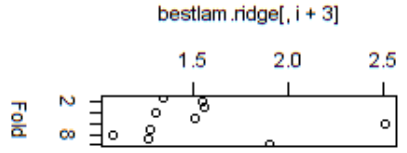

bestlam.ridge[, i + 1]
1.0 2.0 3.0 4.0
Fold
2
8

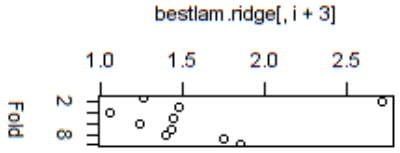

bestlam.ridge[, i + 1]
1.4 1.6 1.8 2.0 2.2 2.4
Fold
2
8


bestlam.ridge[, i + 2]
1.0 1.5 2.0 2.5 3.0 3.5
Fold
2
8


bestlam.ridge[, i + 2]
1.2 1.4 1.6 1.8 2.0
Fold
2
8


bestlam.ridge[, i + 2]
1.4 1.6 1.8 2.0 2.2 2.4
Fold
2
8

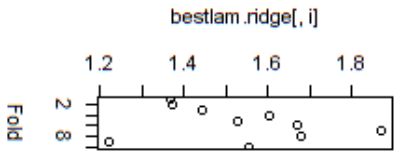

bestlam.ridge[, i + 3]
1.5 2.0 2.5
Fold
2
8


bestlam.ridge[, i + 3]
1.0 1.5 2.0 2.5
Fold
2
8


bestlam.ridge[, i + 3]
1.0 1.5 2.0 2.5 3.0 3.5
Fold
2
8

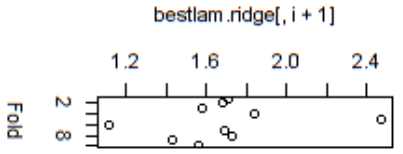

bestlam.ridge[, i]
1.2
1.6
2.0
2.4
Fold
2
8


bestlam.ridge[, i]
1.0
1.5
2.0
2.5
Fold
2
8

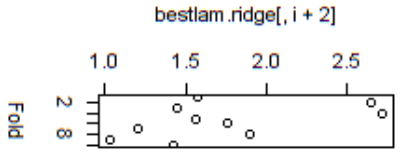

bestlam.ridge[, i]
1.2
1.4
1.6
1.8
Fold
2
8

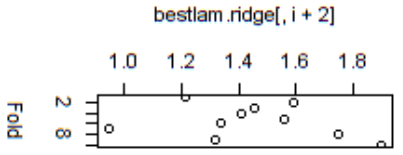

bestlam.ridge[, i + 1]
1.0
1.2
1.4
1.6
1.8
Fold
2
8

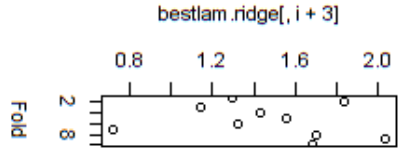

bestlam.ridge[, i + 1]
1.5
2.0
2.5
3.0
3.5
Fold
2
8

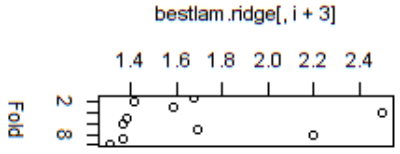

bestlam.ridge[, i + 1]
1.2
1.6
2.0
2.4
Fold
2
8

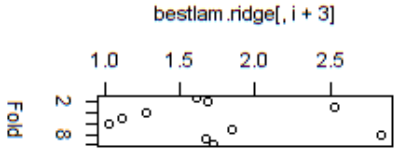

bestlam.ridge[, i + 2]
1.0
1.2
1.4
1.6
1.8
Fold
2
8

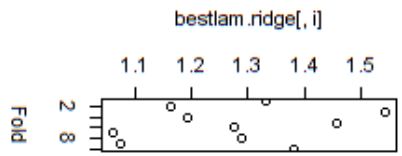

bestlam.ridge[, i + 2]
1.0
1.5
2.0
2.5
Fold
2
8

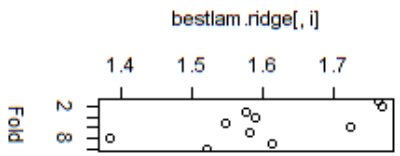

bestlam.ridge[, i + 2]
1.0
1.2
1.4
1.6
1.8
Fold
2
8

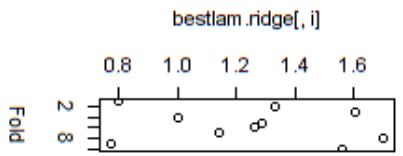

bestlam.ridge[, i + 3]
0.8
1.2
1.6
2.0
Fold
2
8

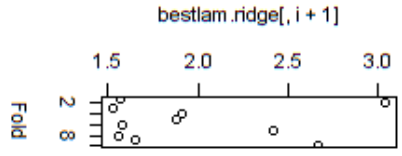

bestlam.ridge[, i + 3]
1.4
1.6
1.8
2.0
2.2
2.4
Fold
2
8


bestlam.ridge[, i + 3]
1.0
1.5
2.0
2.5
Fold
2
8

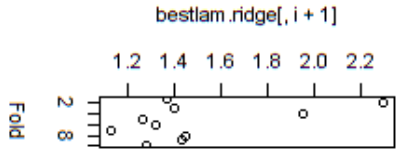

bestlam.ridge[, i]
1.1 1.2 1.3 1.4 1.5
Fold
2 8


bestlam.ridge[, i]
1.4 1.5 1.6 1.7
Fold
2 8

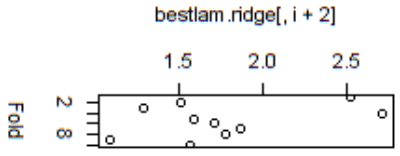

bestlam.ridge[, i]
0.8 1.0 1.2 1.4 1.6
Fold
2 8


bestlam.ridge[, i + 1]
1.5 2.0 2.5 3.0
Fold
2 8

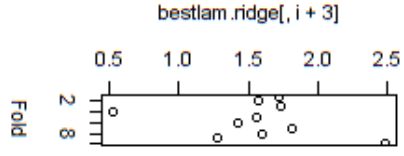

bestlam.ridge[, i + 1]
1.0 1.5 2.0 2.5 3.0
Fold
2 8


bestlam.ridge[, i + 1]
1.2 1.4 1.6 1.8 2.0 2.2
Fold
2 8

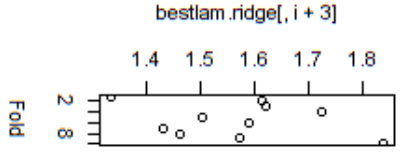

bestlam.ridge[, i + 2]
1.2 1.4 1.6 1.8 2.0
Fold
2 8


bestlam.ridge[, i + 2]
1.5 2.0 2.5
Fold
2 8


bestlam.ridge[, i + 2]
1.5 2.0 2.5
Fold
2 8


bestlam.ridge[, i + 3]
0.5 1.0 1.5 2.0 2.5
Fold
2 8


bestlam.ridge[, i + 3]
1.4 1.5 1.6 1.7 1.8 1.9
Fold
2 8


bestlam.ridge[, i + 3]
1.4 1.5 1.6 1.7 1.8
Fold
2 8

bestlam.ridge[, i]
0.8 1.2 1.6 2.0
2 8
Fold

bestlam.ridge[, i + 1]
1.4 1.5 1.6 1.7 1.8
2 8
Fold

bestlam.ridge[, i + 2]
1.1 1.2 1.3 1.4 1.5 1.6 1.7
2 8
Fold

bestlam.ridge[, i + 3]
1.0 1.5 2.0 2.5 3.0
2 8
Fold

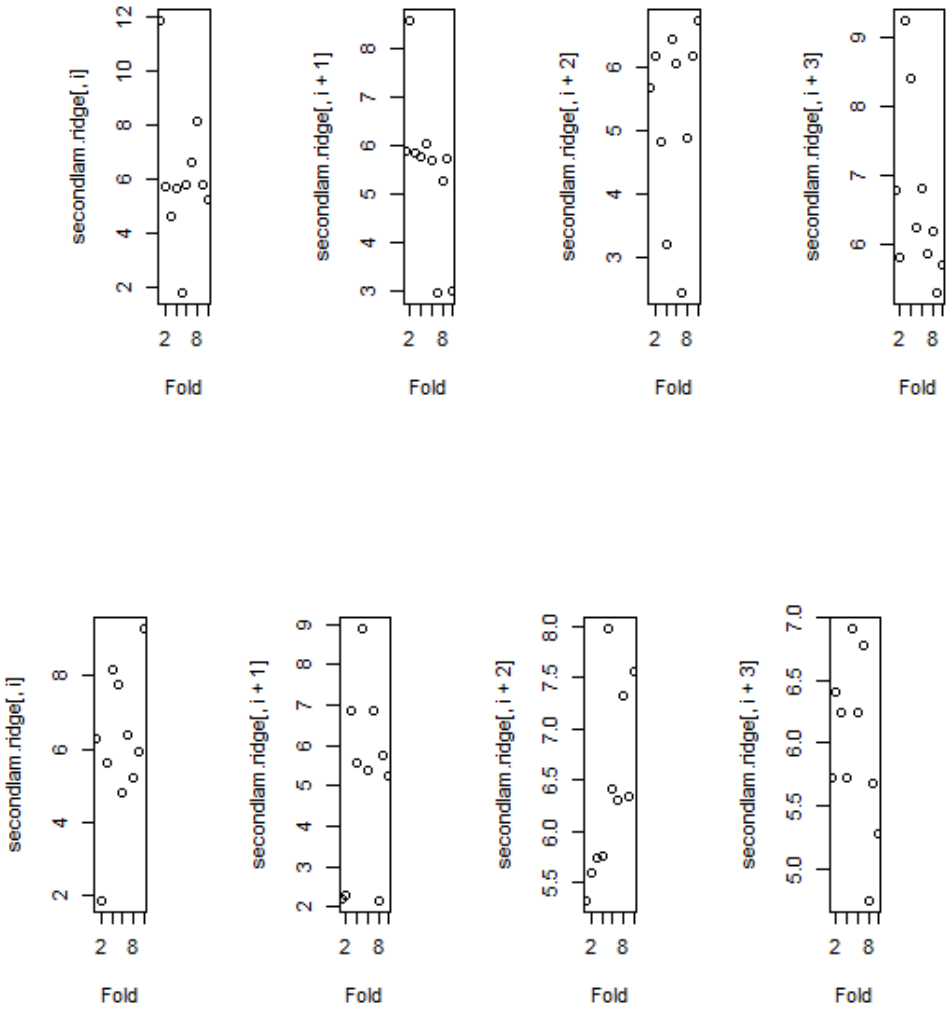
bestlam.ridge[, i]
1.2 1.4 1.6 1.8 2.0 2.2
2 8
Fold

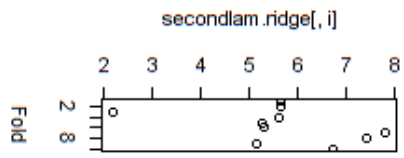
bestlam.ridge[, i + 1]
1.4 1.6 1.8 2.0 2.2 2.4
2 8
Fold

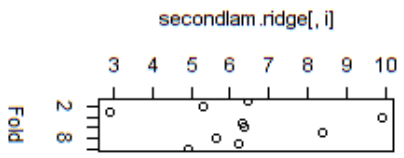
bestlam.ridge[, i + 2]
0.5 1.0 1.5 2.0 2.5 3.0
2 8
Fold

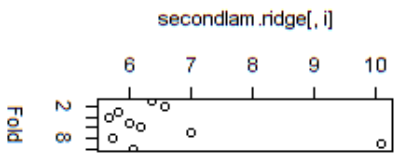
bestlam.ridge[, i + 3]
0.8 1.0 1.2 1.4 1.6 1.8
2 8
Fold

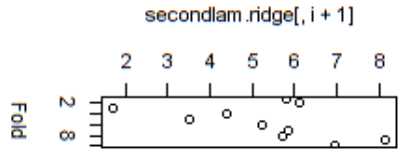
10.9 Lambda Ridge2

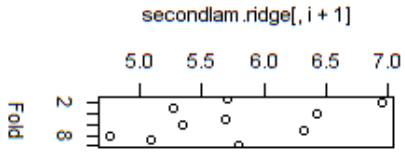


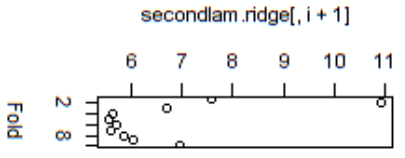

secondlam.ridge[, i]
2 3 4 5 6 7 8
Fold
2
8

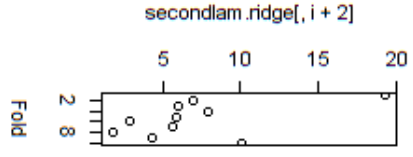

secondlam.ridge[, i]
3 4 5 6 7 8 9 10
Fold
2
8

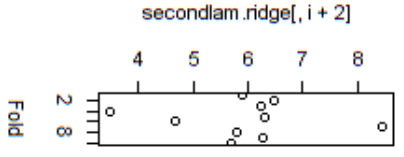

secondlam.ridge[, i]
6 7 8 9 10
Fold
2
8

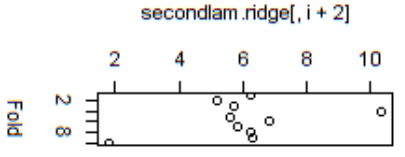

secondlam.ridge[, i + 1]
2 3 4 5 6 7 8
Fold
2
8

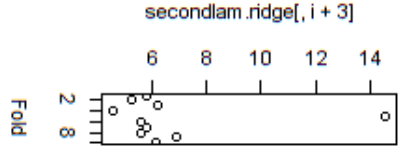

secondlam.ridge[, i + 1]
5.0 5.5 6.0 6.5 7.0
Fold
2
8

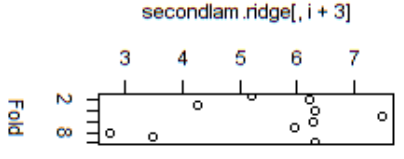

secondlam.ridge[, i + 1]
6 7 8 9 10 11
Fold
2
8

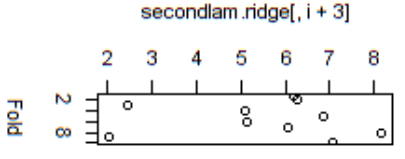

secondlam.ridge[, i + 2]
5 10 15 20
Fold
2
8

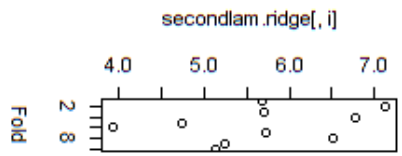

secondlam.ridge[, i + 2]
4 5 6 7 8
Fold
2
8

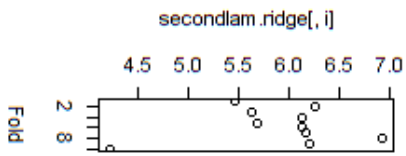

secondlam.ridge[, i + 2]
2 4 6 8 10
Fold
2
8

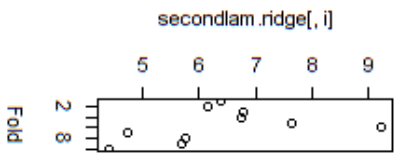

secondlam.ridge[, i + 3]
6 8 10 12 14
Fold
2
8

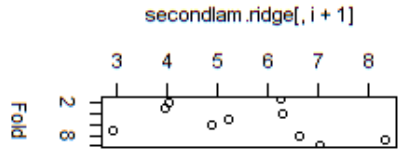

secondlam.ridge[, i + 3]
3 4 5 6 7
Fold
2
8

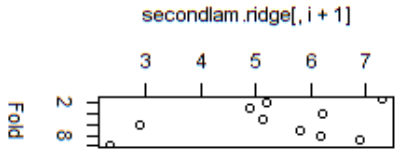

secondlam.ridge[, i + 3]
2 3 4 5 6 7 8
Fold
2
8

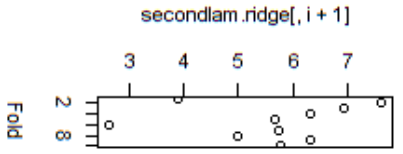

secondlam.ridge[, i]
4.0
5.0
6.0
7.0
Fold
2
8

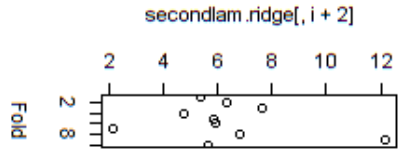

secondlam.ridge[, i]
4.5
5.0
5.5
6.0
6.5
7.0
Fold
2
8

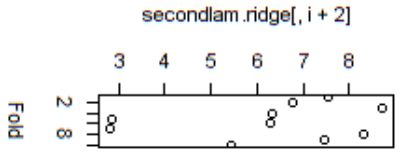

secondlam.ridge[, i]
5
6
7
8
9
Fold
2
8

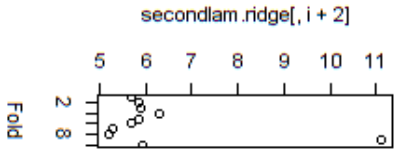

secondlam.ridge[, i + 1]
3
4
5
6
7
8
Fold
2
8

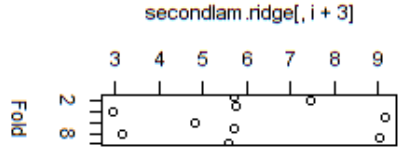

secondlam.ridge[, i + 1]
3
4
5
6
7
Fold
2
8

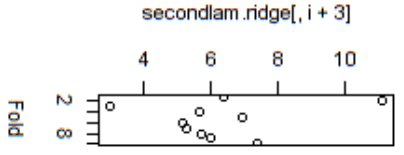

secondlam.ridge[, i + 1]
3
4
5
6
7
Fold
2
8

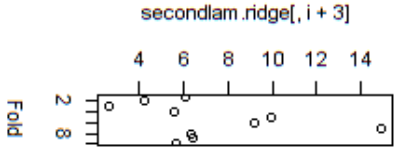

secondlam.ridge[, i + 2]
2
4
6
8
10
12
Fold
2
8

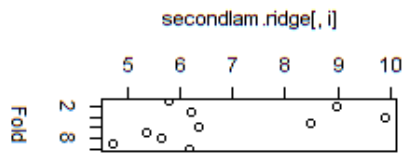

secondlam.ridge[, i + 2]
3
4
5
6
7
8
Fold
2
8

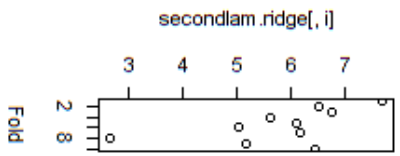

secondlam.ridge[, i + 2]
5
6
7
8
9
10
11
Fold
2
8

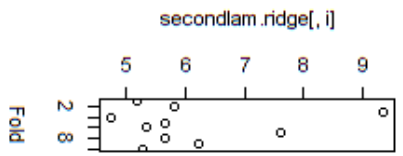

secondlam.ridge[, i + 3]
3
4
5
6
7
8
9
Fold
2
8

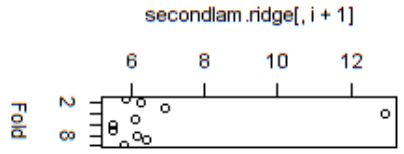

secondlam.ridge[, i + 3]
4
6
8
10
Fold
2
8

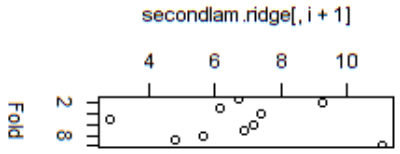

secondlam.ridge[, i + 3]
4
6
8
10
12
14
Fold
2
8

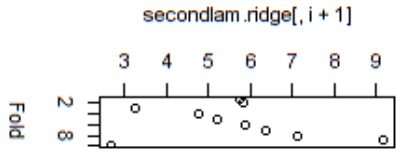

secondlam.ridge[, i]
5 6 7 8 9 10
Fold
2 8

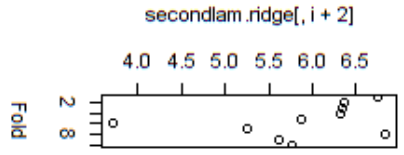

secondlam.ridge[, i]
3 4 5 6 7
Fold
2 8

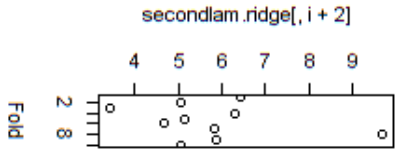

secondlam.ridge[, i]
5 6 7 8 9
Fold
2 8

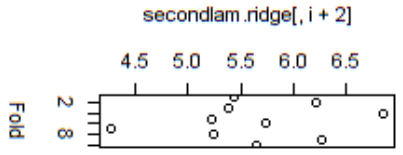

secondlam.ridge[, i + 1]
6 8 10 12
Fold
2 8

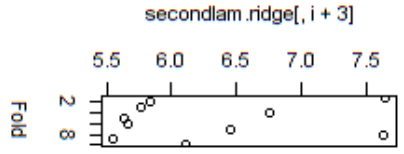

secondlam.ridge[, i + 1]
4 6 8 10
Fold
2 8

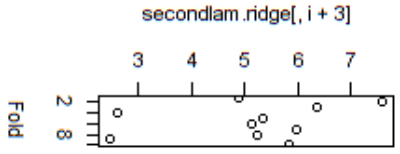

secondlam.ridge[, i + 1]
3 4 5 6 7 8 9
Fold
2 8

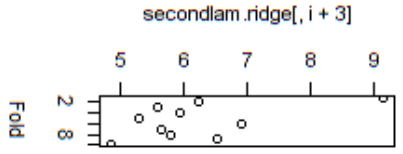

secondlam.ridge[, i + 2]
4.0 4.5 5.0 5.5 6.0 6.5
Fold
2 8

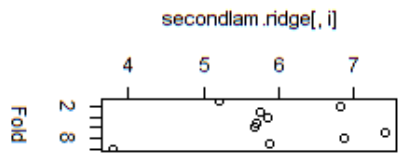

secondlam.ridge[, i + 2]
4 5 6 7 8 9
Fold
2 8

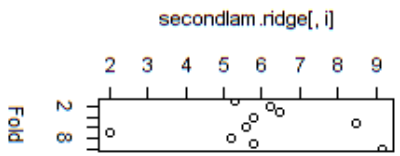

secondlam.ridge[, i + 2]
4.5 5.0 5.5 6.0 6.5
Fold
2 8

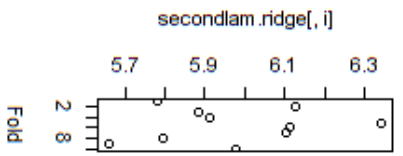

secondlam.ridge[, i + 3]
5.5 6.0 6.5 7.0 7.5
Fold
2 8

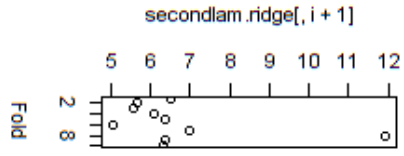

secondlam.ridge[, i + 3]
3 4 5 6 7
Fold
2 8

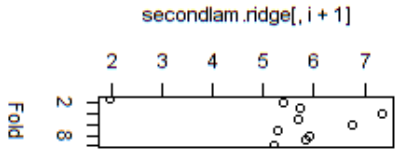

secondlam.ridge[, i + 3]
5 6 7 8 9
Fold
2 8

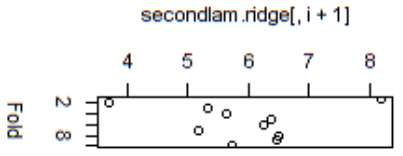

secondlam.ridge[, i]
4 5 6 7
Fold
2 8

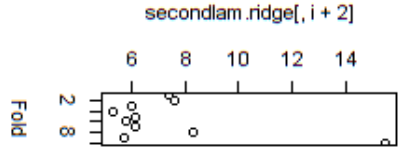

secondlam.ridge[, i]
2 3 4 5 6 7 8 9
Fold
2 8

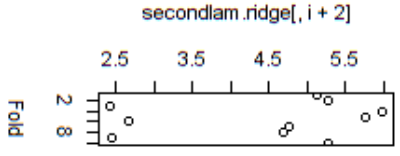

secondlam.ridge[, i]
5.7 5.9 6.1 6.3
Fold
2 8

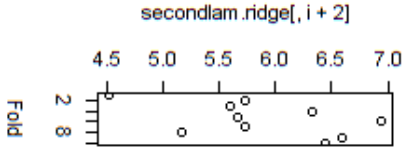

secondlam.ridge[, i + 1]
5 6 7 8 9 10 11 12
Fold
2 8

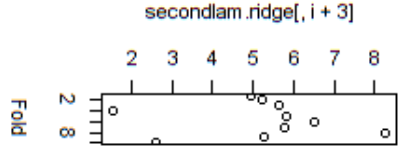

secondlam.ridge[, i + 1]
2 3 4 5 6 7
Fold
2 8

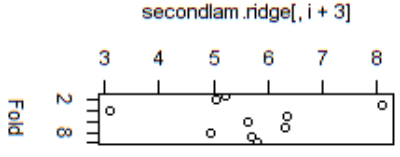

secondlam.ridge[, i + 1]
4 5 6 7 8
Fold
2 8

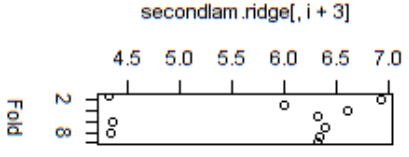

secondlam.ridge[, i + 2]
6 8 10 12 14
Fold
2 8

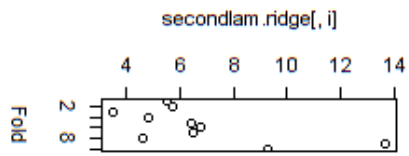

secondlam.ridge[, i + 2]
2.5 3.5 4.5 5.5
Fold
2 8

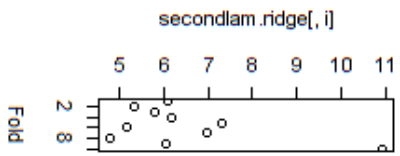

secondlam.ridge[, i + 2]
4.5 5.0 5.5 6.0 6.5 7.0
Fold
2 8

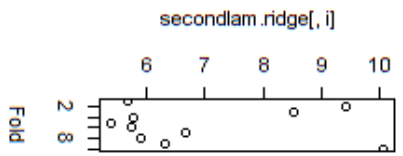

secondlam.ridge[, i + 3]
2 3 4 5 6 7 8
Fold
2 8

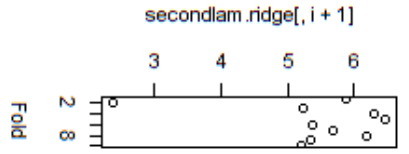

secondlam.ridge[, i + 3]
3 4 5 6 7 8
Fold
2 8

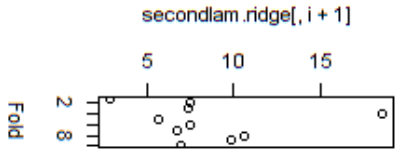

secondlam.ridge[, i + 3]
4.5 5.0 5.5 6.0 6.5 7.0
Fold
2 8

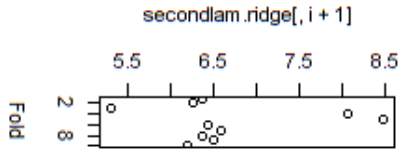

secondlam.ridge[, i]
4 6 8 10 12 14
Fold
2 8

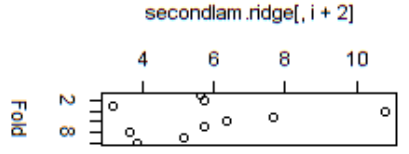

secondlam.ridge[, i]
5 6 7 8 9 10 11
Fold
2 8

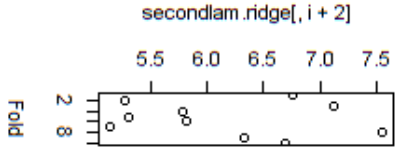

secondlam.ridge[, i]
6 7 8 9 10
Fold
2 8

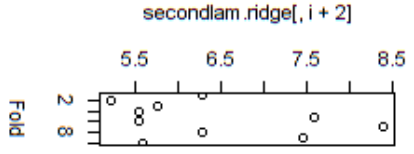

secondlam.ridge[, i + 1]
3 4 5 6
Fold
2 8

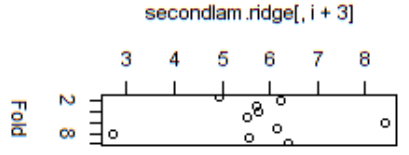

secondlam.ridge[, i + 1]
5 10 15
Fold
2 8

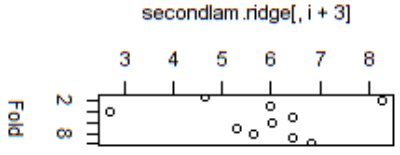

secondlam.ridge[, i + 1]
5.5 6.5 7.5 8.5
Fold
2 8

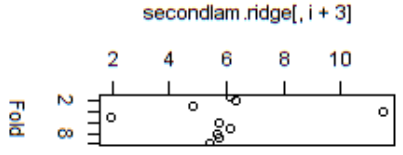

secondlam.ridge[, i + 2]
4 6 8 10
Fold
2 8

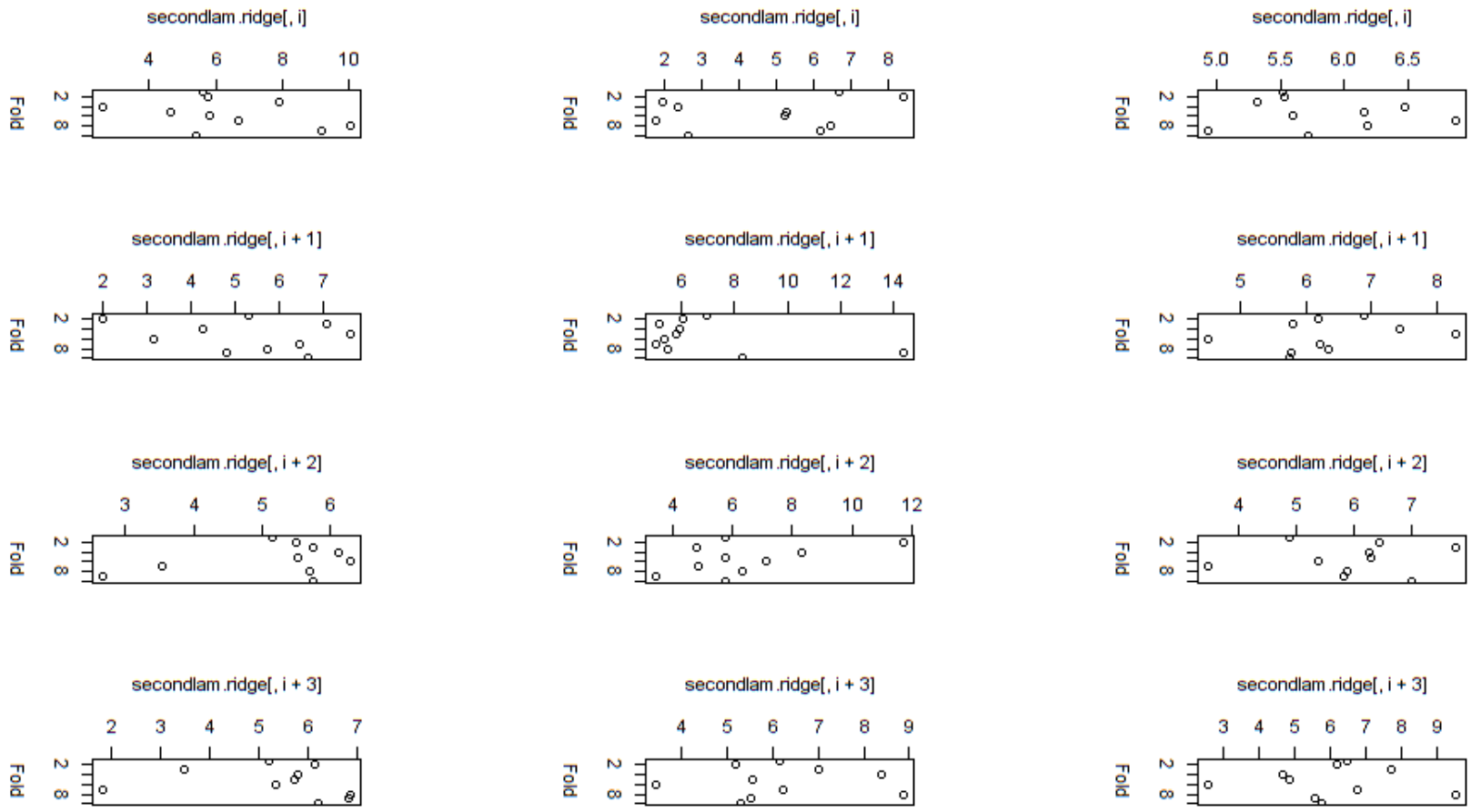

secondlam.ridge[, i + 2]
5.5 6.0 6.5 7.0 7.5
Fold
2 8

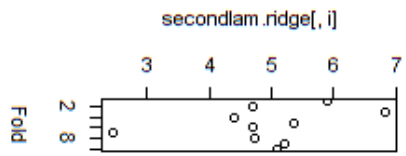

secondlam.ridge[, i + 2]
5.5 6.5 7.5 8.5
Fold
2 8

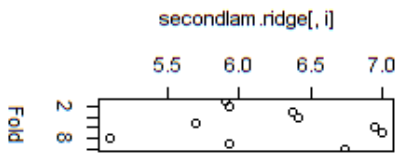

secondlam.ridge[, i + 3]
3 4 5 6 7 8
Fold
2 8

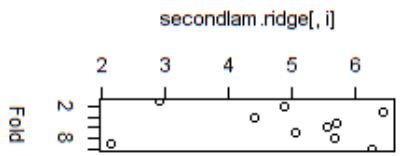

secondlam.ridge[, i + 3]
3 4 5 6 7 8
Fold
2 8

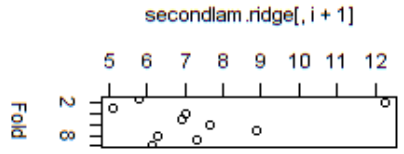

secondlam.ridge[, i + 3]
2 4 6 8 10
Fold
2 8

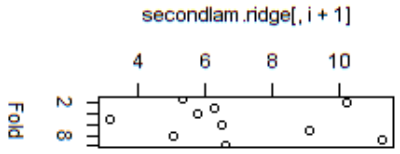

secondlam.ridge[, i]
secondlam.ridge[, i + 1]
secondlam.ridge[, i + 2]
secondlam.ridge[, i + 3]
Fold
2
8

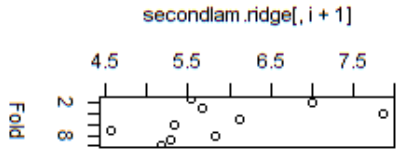

secondlam.ridge[, i]
3 4 5 6 7
Fold
2 8

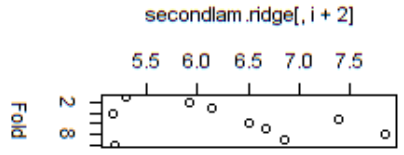

secondlam.ridge[, i]
5.5 6.0 6.5 7.0
Fold
2 8

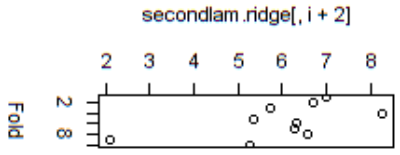

secondlam.ridge[, i]
2 3 4 5 6
Fold
2 8

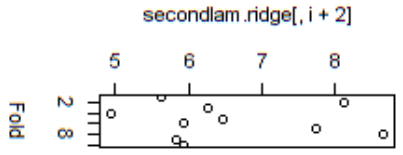

secondlam.ridge[, i + 1]
5 6 7 8 9 10 11 12
Fold
2 8

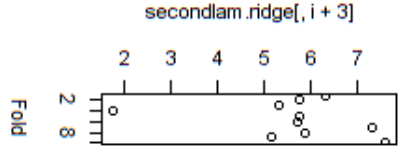

secondlam.ridge[, i + 1]
4 6 8 10
Fold
2 8

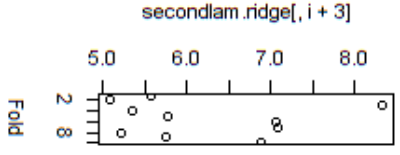

secondlam.ridge[, i + 1]
4.5 5.5 6.5 7.5
Fold
2 8

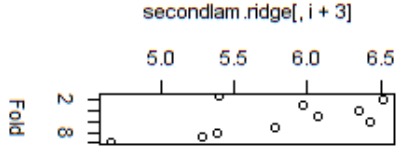

secondlam.ridge[, i + 2]
5.5 6.0 6.5 7.0 7.5
Fold
2 8


secondlam.ridge[, i + 2]
2 3 4 5 6 7 8
Fold
2 8


secondlam.ridge[, i + 2]
5 6 7 8
Fold
2 8


secondlam.ridge[, i + 3]
2 3 4 5 6 7
Fold
2 8


secondlam.ridge[, i + 3]
5.0 6.0 7.0 8.0
Fold
2 8


secondlam.ridge[, i + 3]
5.0 5.5 6.0 6.5
Fold
2 8

secondlam.ridge[, i]
Fold
secondlam.ridge[, i + 1]
Fold
secondlam.ridge[, i + 2]
Fold
secondlam.ridge[, i + 3]
Fold
secondlam.ridge[, i]
Fold
secondlam.ridge[, i + 1]
Fold
secondlam.ridge[, i + 2]
Fold
secondlam.ridge[, i + 3]
Fold

10.10 Lasso with average optimal Lambda results

Variable	Coefficient
(Intercept)	-154.3228000
Month	.
Year	0.0824557
Winter_dummy	.
Age	.
Age Squared	-0.0011799
StrikerDummy	.
GoalkeepDummy	-0.0071867
DefendDummy	.
MonthsRemaining	0.0244010
DummyLeft	.
Height	.
ECI_Old_Team	0.0001159
ECI_New_Team	0.0008737
SQ_Minutes_0	.
SQ_Minutes_1	.
SQ_Minutes_2	.
SQ_Games_0	.
SQ_Games_1	.
SQ_Games_2	.
SQ_Defence_0	.
SQ_Defence_1	.
SQ_Defence_2	.
SQ_Attack_0	.
SQ_Attack_1	.
SQ_Attack_2	.
SQ_Poss_0	.
SQ_Poss_1	.
SQ_Poss_2	.
SQ_Rank_0	.
SQ_Rank_1	.
SQ_Rank_2	-0.0002808
SQ_Att_0/SQ_min_0	.
SQ_Att_1/SQ_min_1	0.0194032
SQ_Att_2/SQ_min_2	.
SQ_Def_0/SQ_min_0	.
SQ_Def_1/SQ_min_1	.
SQ_Def_2/SQ_min_2	.
SQ_Poss_0/SQ_min_0	.
SQ_Poss_1/SQ_min_1	.
SQ_Poss_2/SQ_min_2	.
SQ_Att_0/SQ_Games_0	0.0194442
SQ_Att_1/SQ_Games_1	.
SQ_Att_2/SQ_Games_2	.
SQ_Def_0/SQ_Games_0	.
SQ_Def_1/SQ_Games_1	.
SQ_Def_2/SQ_Games_2	.
SQ_Poss_0/SQ_Games_0	.
SQ_Poss_1/SQ_Games_1	.
SQ_Poss_2/SQ_Games_2	.

Variable	Coefficient
Height*Striker	.
Height*Goalkeep	-0.0011074
Height*Defend	.
Sq_Att_0*Striker	.
Sq_Att_1*Striker	.
Sq_Att_2*Striker	.
Sq_Def_0*Striker	.
Sq_Def_1*Striker	.
Sq_Def_2*Striker	.
Sq_Poss_0*Striker	.
Sq_Poss_1*Striker	.
Sq_Poss_2*Striker	.
Sq_Att_0*Defender	.
Sq_Att_1*Defender	.
Sq_Att_2*Defender	.
Sq_Def_0*Defender	.
Sq_Def_1*Defender	0.0001168
Sq_Def_2*Defender	.
Sq_Poss_0*Defender	.
Sq_Poss_1*Defender	.
Sq_Poss_2*Defender	.
Sq_Att_0*Goalie	.
Sq_Att_1*Goalie	.
Sq_Att_2*Goalie	.
Sq_Def_0*Goalie	.
Sq_Def_1*Goalie	.
Sq_Def_2*Goalie	.
Sq_Poss_0*Goalie	.
Sq_Poss_1*Goalie	.
Sq_Poss_2*Goalie	.
SQ_Att_0*ECI	0.00000004
SQ_Att_1*ECI	.
SQ_Att_2*ECI	0.0000002
SQ_DEF_0*ECI	0.0000002
SQ_DEF_1*ECI	.
SQ_DEF_2*ECI	.
SQ_POSS_0*ECI	.
SQ_POSS_1*ECI	.
SQ_POSS_2*ECI	.
SQ_RANK_0*ECI	.
SQ_RANK_1*ECI	.
SQ_RANK_2*ECI	.
SQ_GAMES_0*ECI	0.0000081
SQ_GAMES_1*ECI	0.0000013
SQ_GAMES_2*ECI	0.0000001
SQ_MIN_0*ECI	.
SQ_MIN_1*ECI	.
SQ_MIN_2*ECI	.

Variable	Coefficient
Dummy_Stade_Rennais	.
Dummy_US_Palermo	.
Dummy_LOSC_Lille	.
Dummy_Aston_Villa	.
Dummy_FC_Barcelona	.
Dummy_AC_Milan	.
Dummy_Valencia_CF	.
Dummy_TSG_Hoffenheim	.
Dummy_Sevilla_FC	.
Dummy_Atalanta	0.0754434
Dummy_Cagliari_Calcio	.
Dummy_Parma	.
Dummy_Werder_Bremen	.
Dummy_AS_Roma	0.0215024
Dummy_Galatasaray	.
Dummy_Besiktas	.
Dummy_Sampdoria	.
Dummy_Granada_CF	0.0447371
Dummy_Saint-Étienne	.
Dummy_Real_Madrid	.
Dummy_FC_Twente	.
Dummy_Levante_UD	-0.2876349
Dummy_Juventus	.
Dummy_VfB_Stuttgart	.
Dummy_Atlético_Madrid	.
Dummy_Hellas_Verona	.
Dummy_Monaco	.
Dummy_Toulouse	.
Dummy_Olympique_Lyon	.
Dummy_Dinamo_Moscow	.
Dummy_FC_Utrecht	.
Dummy_Bayern_Munich	.
Dummy_PEC_Zwolle	.
Dummy_Marseille	.
Dummy_Paris	.
Dummy_Spurs	.
Dummy_FC_Nantes	-0.0220556
Dummy_SSC_Napoli	-0.0272664
Dummy_Montpellier	.
Dummy_FC_Lorient	.
Dummy_E.Frankfurt	.
Dummy_FC_Schalke_04	.
Dummy_Bay.Leverkussen	.
Dummy_1.FSV_Mainz_05	.
Dummy_Inter	0.0084447
Dummy_Évian	.
Dummy_Genoa	.
Dummy_Vitesse	.

Variable	Coefficient
Dummy_Villarreal_CF	.
Dummy_Hamburger_SV	.
Dummy_FC_Groningen	.
Dummy_Trabzonspor	.
Dummy_SC_Cambuur	-0.7903891
Dummy_PSV_Eindhoven	.
Dummy_Bor.M gladbach	.
Dummy_Hertha_BSC	.
Dummy_Bor.Dortmund	0.2413802
Dummy_Hannover_96	.
Dummy_Heracles_Almelo	-0.0803710
Dummy_AFC_Ajax	-0.1760478
Dummy_AZ_Alkmaar	.
Dummy_Fiorentina	.
Dummy_Guingamp	.
Dummy_SC_Freiburg	.
Dummy_OGC_Nice	-0.0275809
Dummy_Feyenoord	.
Dummy_Man_City	.
Dummy_Lazio	0.0256686
Dummy_G.Bordeaux	.
Dummy_SC_Heerenveen	-0.0071448
Dummy_ADO_Den_Haag	.
Dummy_Torino	.
Dummy_FC_Augsburg	-0.0060967
Dummy_Chelsea	.
Dummy_Getafe_CF	-1.0623800
Dummy_Udinese_Calcio	0.0820570
Dummy_SC_Bastia	.
Dummy_Bursaspor	.
Dummy_Málaga_CF	.
Dummy_Benfica	.
Dummy_Sassuolo	.
Dummy_SV_Darmstadt_98	.
Dummy_Braga	.
Dummy_Swansea	.
Dummy_Chievo_Verona	.
Dummy_Real_Sociedad	.
Dummy_Hamburger_SV_1	0.1384958
Dummy_Watford_1	0.7889550
Dummy_Newcastle_1	0.3284419
Dummy_Monaco__1	.
Dummy_Middlesbrough_1	0.5023871
Dummy_Aston_Villa_1	0.6595365
Dummy_Sevilla_FC_1	.
Dummy_AC_Milan_1	0.2517595
Dummy_Besiktas__1	-0.2658967
Dummy_Torino__1	.

Variable	Coefficient
Dummy_Olympiacos_1	.
Dummy_SSC_Napoli_1	.
Dummy_Liverpool_1	.
Dummy_Genoa__1	.
Dummy_FC_Barcelona_1	-0.4518155
Dummy_SD_Eibar_1	.
Dummy_FC_Porto_1	.
Dummy_Sassuolo__1	-0.4208464
Dummy_Udinese_Calcio_1	.
Dummy_Valencia_CF_1	.
Dummy_Pescara_1	.
Dummy_PSV_Eindhoven_1	.
Dummy_West_Ham_1	.
Dummy_AS_Roma_1	.
Dummy_Cagliari_Calcio_1	.
Dummy_Bayern_Munich_1	.
Dummy_Real_Sociedad_1	.
Dummy_Man_Utd_1	0.3141788
Dummy_AZ_Alkmaar_1	.
Dummy_FC_Schalke_04_1	.
Dummy_Paris_SG_1	0.0214212
Dummy_Standard_Liege_1	-0.1893487
Dummy_Bologna_1	0.1759546
Dummy_Galatasaray__1	.
Dummy_Saint-Étienne__1	-0.1161338
Dummy_Rubin_Kazan_1	.
Dummy_Chelsea__1	.
Dummy_1.FSV_Mainz_05_1	.
Dummy_Bor.M gladbach_1	.
Dummy_1.FC_K lautern_1	-0.2745314
Dummy_Man_City_1	0.1989723
Dummy_Southampton_1	.
Dummy_Celta_de_Vigo_1	.
Dummy_OGC_Nice_1	.
Dummy_VfL_Wolfsburg_1	.
Dummy_E.Frankfurt_1	.
Dummy_Fiorentina__1	.
Dummy_Villarreal_CF_1	.
Dummy_RSC_Anderlecht_1	.
Dummy_Sunderland_1	0.8601276
Dummy_Atlético_Madrid_1	-0.3862837
Dummy_Granada_CF_1	.
Dummy_FC_Augsburg_1	.
Dummy_Trabzonspor__1	.
Dummy_Feyenoord__1	.
Dummy_FC_Groningen_1	.
Dummy_Werder_Bremen_1	.
Dummy_Swansea__1	0.3387403

Variable	Coefficient
Dummy_Real_Betis_1	.
Dummy_Union_Berlin_1	.
Dummy_Inter__1	0.3093181
Dummy_FC_Basel_1	.
Dummy_Leicester_1	0.6972173
Dummy_Bor.Dortmund_1	.
Dummy_Arsenal_1	0.1732935
Dummy_VfB_Stuttgart_1	.
Dummy_AFC_Ajax_1	.
Dummy_Juventus__1	.
Dummy_Hertha_BSC_1	-0.1365643
Dummy_SM_Caen_1	-0.3060485
Dummy_TSG_Hoffenheim_1	.
Dummy_F.Düsseldorf_1	-0.1508591
Dummy_Marseille__1	.
Dummy_Real_Madrid_1	-0.0749099
Dummy_Bay.Leverkussen_1	.
Dummy_Hellas_Verona_1	.
Dummy_FC_Lorient_1	.
Dummy_Sampdoria__1	0.0501097
Dummy_FC_Ingolstadt_1	.
Dummy_1.FC_Köln_1	.
Dummy_Atalanta__1	.
Dummy_Olympique_Lyon_1	.
Dummy_Stade_Rennais_1	.
Dummy_Hannover_96_1	.
Dummy_US_Palermo_1	.
Dummy_Crystal_Palace_1	0.7732245
Dummy_Stoke_City_1	0.1668322
Dummy_Dep.la_Coruna_1	.
Dummy_Club_Brugge_1	.
Dummy_LOSC_Lille_1	.
Dummy_Greuther_Fürth_1	-1.1850040
Dummy_Chievo_Verona_1	.
Dummy_Lazio__1	.
Dummy_RB_Leipzig_1	0.6239777

Lasso estimates with lambda = 0.046.

10.11 Web-Scrapers

```
from selenium import webdriver
from bs4 import BeautifulSoup
import pyodbc

################################
#### HIER AANPASSEN
################################
filename = 'Games.txt'
with open(filename) as f:
    datums = [line.rstrip() for line in f]
```

```
for datum in datums:
    try:
        print(datum)
        data = []

##        profile = webdriver.FirefoxProfile()
##        profile.set_preference('startup.homepage_welcome_url.additional', 'about:blank')

        driver = webdriver.Chrome()
        #link = "http://www.squawka.com/football-player-rankings#performance-score#player-stats#dutch-↩
            eredivisie|season-2013/2014#all-teams#all-player-positions#16#41#0#0#90#02/08/2013#"+datum+"#↩
            season#1#all-matches#total#desc#total"
        #link = "http://www.squawka.com/football-player-rankings#performance-score#player-stats#dutch-↩
            eredivisie|season-2014/2015#all-teams#all-player-positions#16#39#0#0#90#08/08/2014#"+datum+"#↩
            season#1#all-matches#total#desc#total"
        #link = "http://www.squawka.com/football-player-rankings#performance-score#player-stats#dutch-↩
            eredivisie|season-2015/2016#all-teams#all-player-positions#16#39#0#0#90#08/08/2015#"+datum+"#↩
            season#1#all-matches#total#desc#total"

        #link = "http://www.squawka.com/football-player-rankings#performance-score#player-stats#dutch-↩
            eredivisie|season-2016/2017#all-teams#all-player-positions#16#40#0#0#90#05/08/2016#"+datum+"#↩
            season#1#all-matches#total#desc#total"
##############################
#### CHOOSE LINK
##############################

##        link = "http://www.squawka.com/football-player-rankings#performance-score#player-stats#italian-↩
    serie-a|season-2015/2016#all-teams#all-player-positions#16#45#0#0#90#01/08/2015#"+datum+"#season#1#↩
    all-matches#total"
        ##link = "http://www.squawka.com/football-player-rankings#performance-score#player-stats#english-↩
            premier-league|season-2011/2012#all-teams#all-player-positions#16#45#0#0#90#01/08/2011#"+↩
            datum+"#season#1#all-matches#total"
##        link = "http://www.squawka.com/football-player-rankings#performance-score#player-stats#french-↩
    ligue-1|season-2012/2013#all-teams#all-player-positions#16#45#0#0#90#01/08/2012#"+datum+"#season#1#↩
    all-matches#total"
##        link = "http://www.squawka.com/football-player-rankings#performance-score#player-stats#spanish-↩
    la-liga|season-2012/2013#all-teams#all-player-positions#16#45#0#0#90#01/08/2012#"+datum+"#season#1#↩
    all-matches#total"
##        link = "http://www.squawka.com/football-player-rankings#performance-score#player-stats#russian-↩
    premier-league|season-2014/2015#all-teams#all-player-positions#16#45#0#0#90#01/08/2014#"+datum+"#↩
    season#1#all-matches#total"
##        link = "http://www.squawka.com/football-player-rankings#performance-score#player-stats#german-↩
    bundesliga|season-2015/2016#all-teams#all-player-positions#16#45#0#0#90#01/08/2015#"+datum+"#season↩
    #1#all-matches#total"
##        link = "http://www.squawka.com/football-player-rankings#performance-score#player-stats#turkish-↩
    super-lig|season-2015/2016#all-teams#all-player-positions#16#45#0#0#90#01/08/2015#"+datum+"#season#1#↩
    all-matches#total"
    ## link = "http://www.squawka.com/football-player-rankings#performance-score#player-stats#portuguese-↩
        primeira-liga|season-2009/2010#all-teams#all-player-positions#16#45#0#0#90#01/08/2014#"+datum+"#↩
        season#1#all-matches#total"
##        link = "http://www.squawka.com/football-player-rankings#performance-score#player-stats#↩
    brazilian-serie-a|season-2015/2016#all-teams#all-player-positions#16#45#0#0#90#01/08/2015#"+datum+"#↩
    season#1#all-matches#total"
##        link = "http://www.squawka.com/football-player-rankings#performance-score#player-stats#↩
    australian-a-league|season-2013/2014#all-teams#all-player-positions#16#45#0#0#90#01/08/2013#"+datum↩
    +"#season#1#all-matches#total"
        link = "http://www.squawka.com/football-player-rankings#performance-score#player-stats#dutch-↩
            eredivisie|season-2014/2015#all-teams#all-player-positions#16#40#0#0#90#01/08/2014#"+datum+"#↩
            season#1#all-matches#total"



        driver.get(link)

        x = 1

        while True:
            driver.execute_script("window.scrollTo(0, document.body.scrollHeight);")
            x += 1
            if x == 100:
                break

        pageSource = driver.page_source
        soup = BeautifulSoup(pageSource, "html.parser")
```

```
        driver.quit()

        test = soup.find_all('td')

        test2 = soup.find_all('div',{'class':'stats-player-name'})

        test3 = soup.find_all('div',{'class':'stats-player-team'})

        rank = []
        naam = []
        club = []
        position = []
        games = []
        mins = []
        defence = []
        attack = []
        poss = []
        total = []

        for i in range(0, len(test), 8):
            a = test[i].get_text()
            rank.append(a)

        for i in range(len(test2)):
            b = test2[i].get_text()
            naam.append(b)

        for i in range(len(test3)):
            team = test3[i].get_text()
            c = team.split(' - ')[1]
            club.append(c)
            d = team.split(' - ')[0]
            position.append(d)

        for i in range(2, len(test), 8):
            e = test[i].get_text()
            games.append(e)

        for i in range(3, len(test), 8):
            f = test[i].get_text()
            mins.append(f)

        for i in range(4, len(test), 8):
            g = test[i].get_text()
            defence.append(g)

        for i in range(5, len(test), 8):
            h = test[i].get_text()
            attack.append(h)

        for i in range(6, len(test), 8):
            i = test[i].get_text()
            poss.append(i)

        for i in range(7, len(test), 8):
            j = test[i].get_text()
            total.append(j)

        for i in range(len(rank)):
            data.append(datum)
            data.append('Eredivisie')

################################
#### HIER AANPASSEN
################################
##            data.append('2009/2010')
##            data.append('2010/2011')
##            data.append('2011/2012')
##            data.append('2012/2013')
##            data.append('2013/2014')
            data.append('2014/2015')
##            data.append('2015/2016')
##            data.append('2016/2017')
```

```
            data.append(rank[i])
            data.append(naam[i].replace("'","''+''''+'"))
            data.append(club[i].replace("'","''+''''+'"))
            data.append(position[i])
            data.append(games[i])
            data.append(mins[i])
            data.append(defence[i].replace(',',''))
            data.append(attack[i].replace(',',''))
            data.append(poss[i].replace(',',''))
            data.append(total[i].replace(',',''))

        conn = pyodbc.connect(r'DRIVER={SQL Server};SERVER=xx;DATABASE=xxx;UID=xx;PWD=xxx))%')
        dbCursor = conn.cursor()
        for k in range(0, len(data), 13):
            sql = "INSERT INTO SQ_PlayerRanking VALUES (CONVERT(DATE, '"+data[k]+"', 105),'"+data[k+1]+"'↩
                 ,'"+data[k+2]+"',"+data[k+3]+",'"+data[k+4]+"','"+data[k+5]+"','"+data[k+6]+"',"+data[k↩
                 +7]+","+data[k+8]+","+data[k+9]+","+data[k+10]+","+data[k+11]+","+data[k+12]+")"
            try:
                dbCursor.execute(sql)
                conn.commit()
            except:
                print(sql)
                continue

    except:
        print('Deze '+datum+' gaat fout')
        continue
```

```
from bs4 import BeautifulSoup
import requests, csv, itertools
from datetime import datetime

startTime = datetime.now()

string = "EUROPA_"

filename = string+'spelers_links.txt'
with open(filename) as f:
    links = [line.rstrip() for line in f]

open(string+"transfers_links.txt", 'w').close()

print(len(links))
w = 0

for link in links:
    w = w+1
    print(w)
    r = requests.get('http://www.transfermarkt.com/'+str(link))
    html = r.text
    soup = BeautifulSoup(html, "html.parser")
    test = soup.find_all('td',{'class':'zentriert hide-for-small'})

    links = []

    for i in range(len(test)):
        test2 = test[i].find_all('a')
        try:
            a = test2[0].get('href')
            with open(string+"transfers_links.txt", "a") as text_file:
                print(a, file=text_file)
        except:
            continue

print(datetime.now() - startTime)
```

```
from bs4 import BeautifulSoup
import requests, csv, itertools
from datetime import datetime
```

```
startTime = datetime.now()



string = "EUROPA_"

filename = string+'competitie_links.txt'
with open(filename) as f:
    links = [line.rstrip() for line in f]

open(string+"spelers_links.txt", 'w').close()

for link in links:
    for z in range(2000, 2017):
        r = requests.get('http://www.transfermarkt.com'+str(link)+'/saison_id/'+str(z))
        html = r.text
        soup = BeautifulSoup(html, "html.parser")
        test = soup.find_all('a',{'class':'spielprofil_tooltip'})

        links = []

        for i in range(len(test)):
            try:
                a = test[i].get('href')
                with open(string+"spelers_links.txt", "a") as text_file:
                    if a.replace('profil', 'transfers') not in open(string+"spelers_links.txt").read():
                        print(a.replace('profil', 'transfers'), file=text_file)
            except:
                print('Er mislukt iets')
                continue

print(datetime.now() - startTime)
```

```
from bs4 import BeautifulSoup
import requests, pyodbc, time

string = "Try_"

open(string+"mislukt_spelerinfo.txt", 'w').close()

filename = 'TM_PlayerInfo_20170527.txt'
with open(filename) as f:
    links = [line.rstrip() for line in f]

w = 0
print(len(links))

for link in links:
    try:
        w = w + 1
        print(w)
        r = requests.get('http://www.transfermarkt.com/xxx/profil/spieler/'+str(link), headers=header)
        html = r.text
        soup = BeautifulSoup(html, "html.parser")

        name1 = soup.find_all('h1',{'itemprop':'name'})
        name2 = name1[0].get_text()
        naam = name2.replace('\r','').replace('\t','').replace('\n','')

        playerid = link

    ##    try:
    ##        waarde1 = soup.find_all('div',{'class':'marktwert'})
    ##        waarde2 = waarde1[0].find_all('a')
    ##        waarde = waarde2[0].get_text().replace('\r','').replace('\t','').replace('\n','').replace('↩
      dzd.     ','000').replace(' mln.     ', '0000').replace(',','')
    ##    except:
    ##        waarde = 'NULL'

        test = soup.find_all('div',{'class':'row collapse'})
        test2 = test[0].find_all('table',{'class':'auflistung'})
```

```
        data = []
        data2 = []

        nationaliteiten = []

        for i in range(len(test2)):
            test3 = test2[0].find_all('tr')
            for j in range(len(test3)):
                data.append(test3[j].get_text().replace('\n', '').replace('\r', '').replace('\xa0\xa0', '↩
                    ').replace('  ', ''))
                if 'Nationality' in test3[j].get_text():
                    test4 = test3[j].find_all('img',{'class':'flaggenrahmen'})
                    for k in range(len(test4)):
                        nationaliteiten.append(test4[k].get('title'))

        nationaliteit1 = 'NULL'
        nationaliteit2 = 'NULL'

        try:
            nationaliteit1 = nationaliteiten[0]
        except:
            nationaliteit1 = 'NULL'
        try:
            nationaliteit2 = nationaliteiten[1]
        except:
            nationaliteit2 = 'NULL'

        datum = time.strftime("%x")
        plaats = 'NULL'
        lengte = 'NULL'
        positie = 'NULL'
        voet = 'NULL'
        spelersmakelaar = 'NULL'
        contract = 'NULL'
        club = 'NULL'

        for z in range(len(data)):
            if 'Date of birth' in data[z]:
                datum = data[z].replace('Date of birth:','').replace(' Happy Birthday', '').strip()
            if 'Place of birth' in data[z]:
                plaats = data[z].replace('Place of birth:','').strip()
            if 'Height' in data[z]:
                lengte = data[z].replace('Height:','').replace('m', '').replace(',','').replace('-','NULL↩
                    ').replace('N/A','NULL').strip()
                if len(lengte) == 0:
                    lengte = 'NULL'
            if 'Position' in data[z]:
                positie = data[z].replace('Position:','').strip()
            if 'Foot' in data[z]:
                voet = data[z].replace('Foot:','').strip()
            if "Player's agent" in data[z]:
                spelersmakelaar = data[z].replace("Player's agent:",'').strip()
            if 'Contract until' in data[z]:
                contract = data[z].replace('Contract until:','').strip()
            if 'Current club' in data[z]:
                club = data[z].replace('Current club:','').strip()

        data2.append(playerid)
        data2.append(naam.replace("'","''"))
    ##        data2.append(club.replace("'","''"))
    ##        data2.append(waarde.replace("'","''"))
        data2.append(datum.replace("'","''"))
        data2.append(plaats.replace("'","''"))
        data2.append(lengte.replace("'","''"))
        data2.append(nationaliteit1.replace("'","''"))
        data2.append(nationaliteit2.replace("'","''"))
        data2.append(positie.replace("'","''"))
        data2.append(voet.replace("'","''"))
        data2.append(spelersmakelaar.replace("'","''"))
    ##        data2.append(contract.replace("'","''"))
    ##        print(data2)

        conn = pyodbc.connect(r'DRIVER={SQL Server};SERVER=xxx;DATABASE=xxx;UID=xxx;PWD=xxx))%')
        dbCursor = conn.cursor()
```

```
        for k in range(0, len(data2), 10):
            sql = "INSERT INTO [TM_PlayerInfo2] VALUES ("+str(data2[k])+",'"+data2[k+1]+"',CONVERT(DATE,'↩
                "+data2[k+2]+"'),'"+data2[k+3]+"',"+str(data2[k+4])+",'"+data2[k+5]+"','"+data2[k+6]+"','↩
                "+data2[k+7]+"','"+data2[k+8]+"','"+data2[k+9]+"')"
##            print(sql)
            dbCursor.execute(sql)
        conn.commit()
        update1 = "UPDATE [TM_PlayerInfo2] SET c_BirthPlace = NULL WHERE c_BirthPlace = 'NULL'"
        update2 = "UPDATE [TM_PlayerInfo2] SET c_Nationality1 = NULL WHERE c_Nationality1 = 'NULL'"
        update3 = "UPDATE [TM_PlayerInfo2] SET c_Nationality2 = NULL WHERE c_Nationality2 = 'NULL'"
        update4 = "UPDATE [TM_PlayerInfo2] SET c_Position = NULL WHERE c_Position = 'NULL'"
        update5 = "UPDATE [TM_PlayerInfo2] SET c_PreferredFoot = NULL WHERE c_PreferredFoot = 'NULL'"
        update6 = "UPDATE [TM_PlayerInfo2] SET d_BirthDate = NULL WHERE d_BirthDate = CONVERT(DATE, ↩
            GETDATE())"
        update7 = "UPDATE [TM_PlayerInfo2] SET c_Agent = NULL WHERE c_Agent = 'NULL'"
        dbCursor.execute(update1)
        dbCursor.execute(update2)
        dbCursor.execute(update3)
        dbCursor.execute(update4)
        dbCursor.execute(update5)
        dbCursor.execute(update6)
        dbCursor.execute(update7)
        conn.close()
    except:
        with open(string+"mislukt_spelerinfo.txt", "a") as text_file:
            print(link, file=text_file)
        continue
```