Examining New Age Analytics of European Soccer

STAT 420: Methods of Applied Statistics

Spring 2021

Group 51

Jefferson Mathews, jrm10

David Lin, yiyangl7

Vidushi Somani, vsomani3

Benny Zhao, bzhao22

Contents

Introduction

European soccer is increasingly popular in the recent few years, which inspired us to use it as the topic for this statistical project. This project is focused on the analysis of statistical data about European soccer, specifically, the statistics on four European Leagues, EPL, La Liga, Serie A, and Ligue 1, during the 17-18, 18-19, and 19-20 seasons.

The data set we used in this report is an aggregation of multiple separate data sets that can be found at the following sources:

- France: https://fbref.com/en/comps/13/history/Ligue-1-Seasons
- England: https://fbref.com/en/comps/9/history/Premier-League-Seasons
- Italy: https://fbref.com/en/comps/11/history/Serie-A-Seasons
- Spain: https://fbref.com/en/comps/12/history/La-Liga-Seasons

We are mainly interested in predicting success with various predictors, including expected goals (xG), expected goals allowed (xGA), number of possessions as a proportion of attempted passes (Poss), end of season rankings (data set uses column name `Notes`), and average player age of a team (Age). We use the proportion of wins as our response variable, which represents how successful a team is objectively and accurately. To find the best statistical model, we utilize a series of methods, including collinearity analysis, simple relationship analysis, Akaike Information Criterion (AIC), Bayesian Information Criterion (BIC), residual diagnostics, and outlier diagnostics. Through the exploration, the optimal model is found.

Methods

Before proceeding to the process we went through to arrive at our final model, the following is the `R` code we used to import the data set and the libraries used during our study, as well as fix the typos within the data set file.

```r
library(readr)
library(knitr)
library(faraway)
library(lmtest)
library(ggplot2)
library(ggthemes)
european_soccer <- read_csv("europeanSoccer.csv")
european_soccer$Notes[which(european_soccer$Notes == "Relagtion Spot")] = "Relegation Spot"
```

The libraries we used will be listed again in the appendix along with their purposes.

Response variable & predictor variable selection

First, we must pick a metric that best represents the success of a team for any given season; the metric we chose would be the proportion of games won by a team out of all the games played by the team in the season.

```r
W_prop <- european_soccer$W / (european_soccer$W + european_soccer$L + european_soccer$D)
european_soccer <- cbind(european_soccer, W_prop)
```

Next, we picked a selection of predictor variables from the data set that we think would have prediction power for predicting the proportion of the games won by a team. We decided on the following predictor variables to consider:

- Expected goals (xG), which measures the quality of the shots made by the team throughout the season, taking into account factors such as assist type, shot angle, distance from the goal, whether it was a headed shot, and whether it was defined as a big chance.
  - Essentially, it is the offensive power of a team, measured as a probability out of 100 of scoring, so a higher value means a more offensively capable team.
- Expected goals allowed (xGA), which measures a team's ability to prevent scoring chances
  - Essentially, it is the defensive power of a team, measured as a probability out of 100 for failing to prevent the opponent from scoring, so a lower value is a more defensively capable team.
- Amount of possession (Poss), which measures the proportion of passes attempted
- The average player ages of a team (Age)
- Ranking at the end of the season (Notes)

  - Three tiers in total, listed in descending order of distinction:
    - European spot
    - Neither
    - Relegation spot

Collinearity analysis

First, we chose to examine whether there was collinearity within the full model that included all the aforementioned predictors by looking at the values of the variance inflation factors of each of the predictors.

```r
model_full <- lm(W_prop ~ xG + xGA + Poss + Age + Notes, data = european_soccer)
vif(model_full)
```

Predictor variable	VIF value
xG	3.205039
xGA	1.966579
Poss	3.030101
Age	1.030270
End of season ranking: Neither	2.068080
End of season ranking: Relegation Spot	2.266972

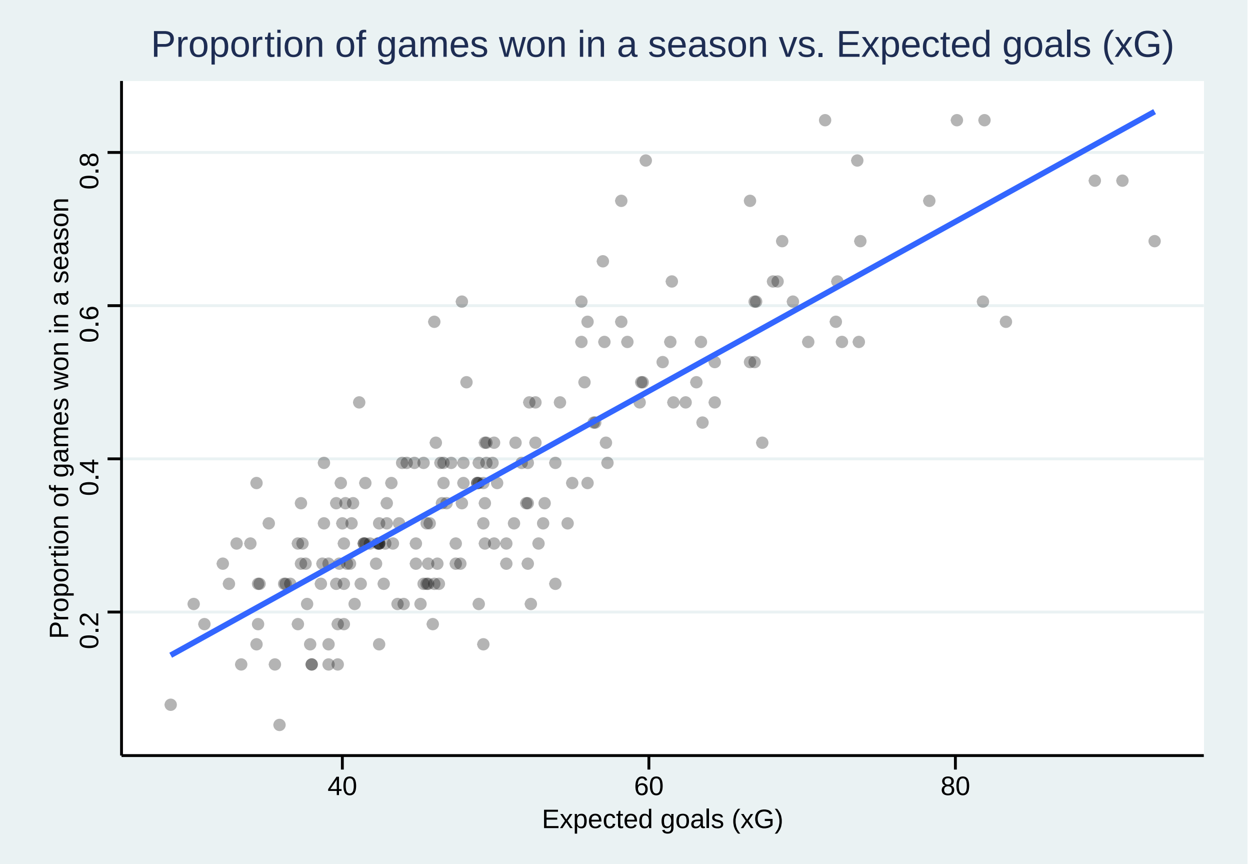
Since none of the VIF values of the predictors were greater than 5, we concluded that there was no need to exclude any variables from our final model.

Simple relationship analysis

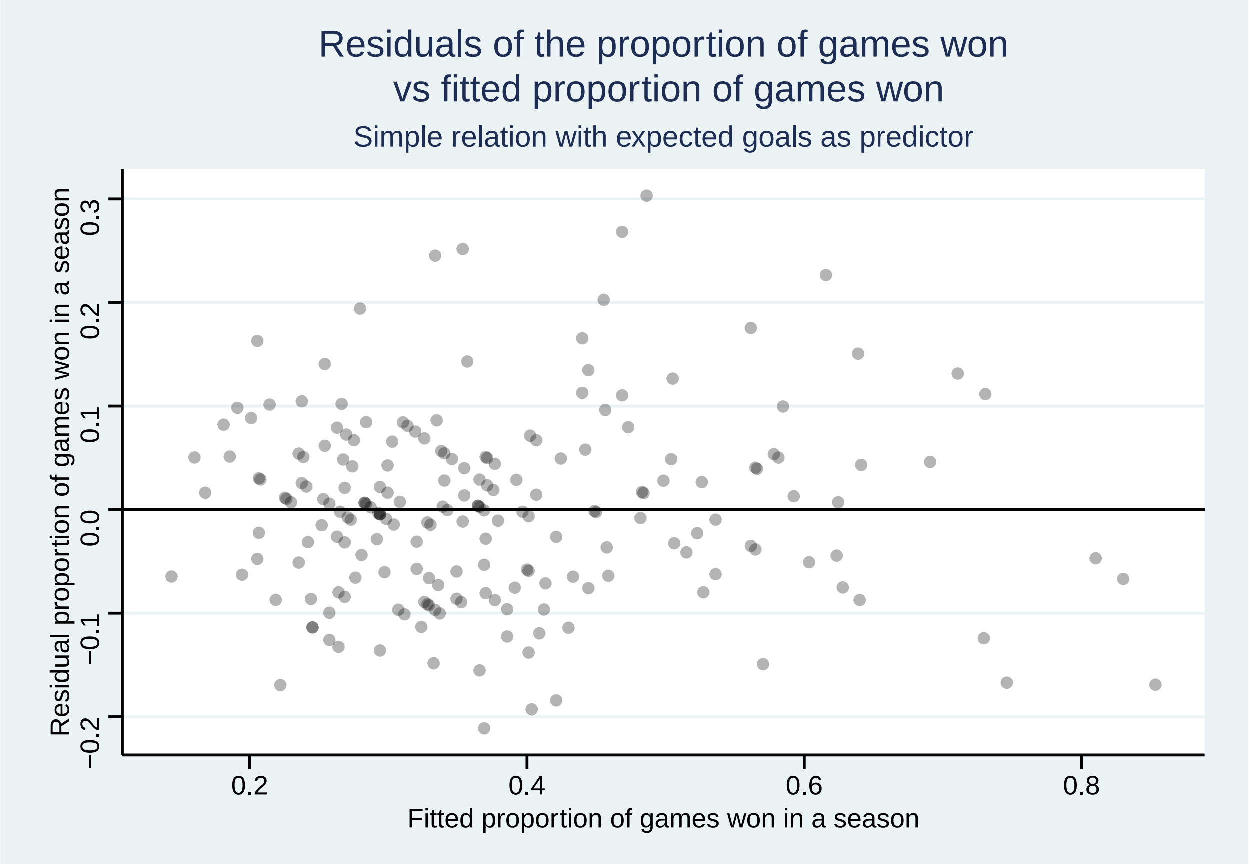
Next, we decided to look for any evidence of a relationship, linear or otherwise, between the proportion of games won by a team in a season and each of those predictor variables, individually.

The first simple relationship we examined was the proportion of games won by a team in a season and the expected goals.

```
model_xG <- lm(W_prop ~ xG, data = european_soccer)
ggplot(data = european_soccer, aes(x = xG, y = W_prop)) +
  geom_point(alpha = 0.3) +
  geom_smooth(method = "lm", se = FALSE) +
  theme_stata() +
  labs(title = "Proportion of games won in a season vs. Expected goals (xG)",
       x = "Expected goals (xG)",
       y = "Proportion of games won in a season")
```

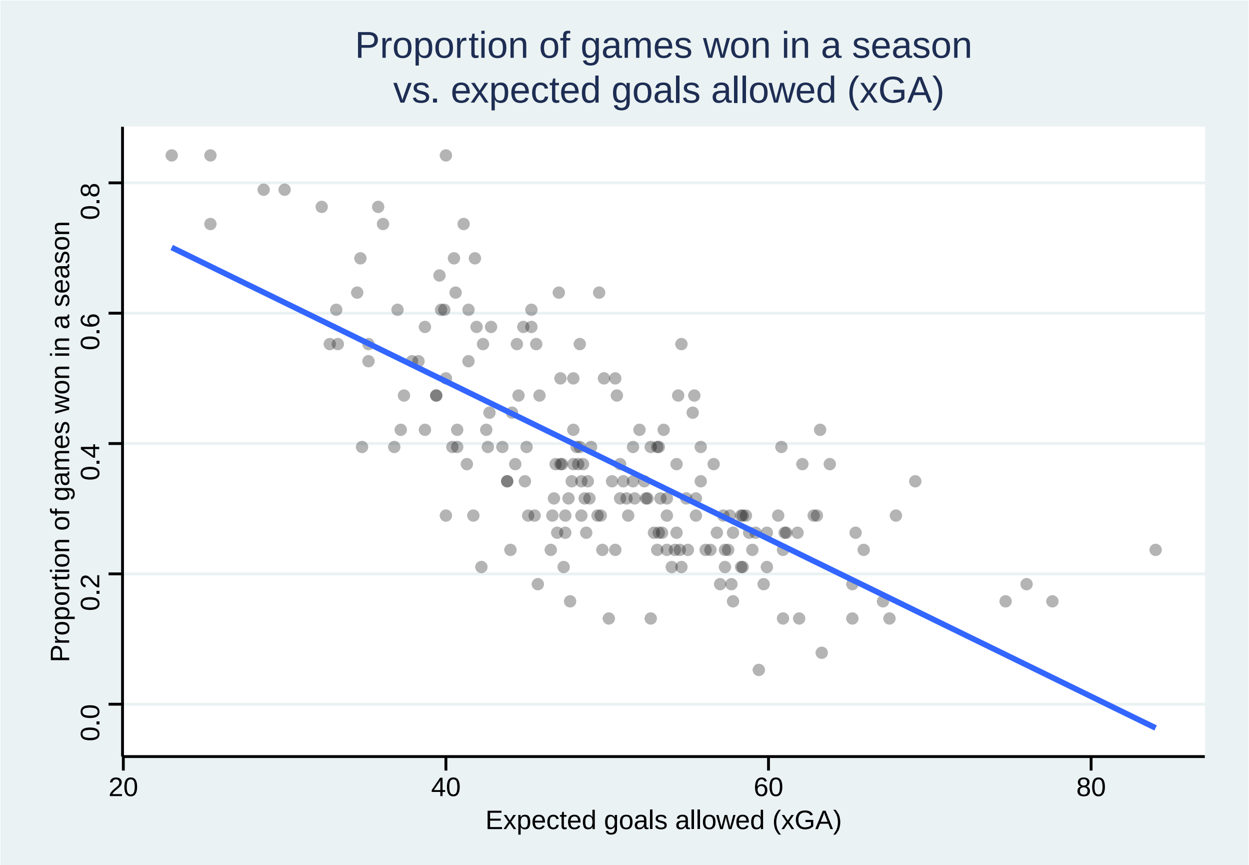


```
model_xG_fitted <- fitted(model_xG)
model_xG_resid <- resid(model_xG)
ggplot(data = data.frame(model_xG_fitted, model_xG_resid),
       aes(x = model_xG_fitted, y = model_xG_resid)) +
  geom_point(alpha = 0.3) +
  geom_abline(intercept = 0, slope = 0) +
  theme_stata() +
  labs(title = "Residuals of the proportion of games won\n vs fitted proportion of games won",
       subtitle = "Simple relation with expected goals as predictor",
       x = "Fitted proportion of games won in a season",
       y = "Residual proportion of games won in a season")
```

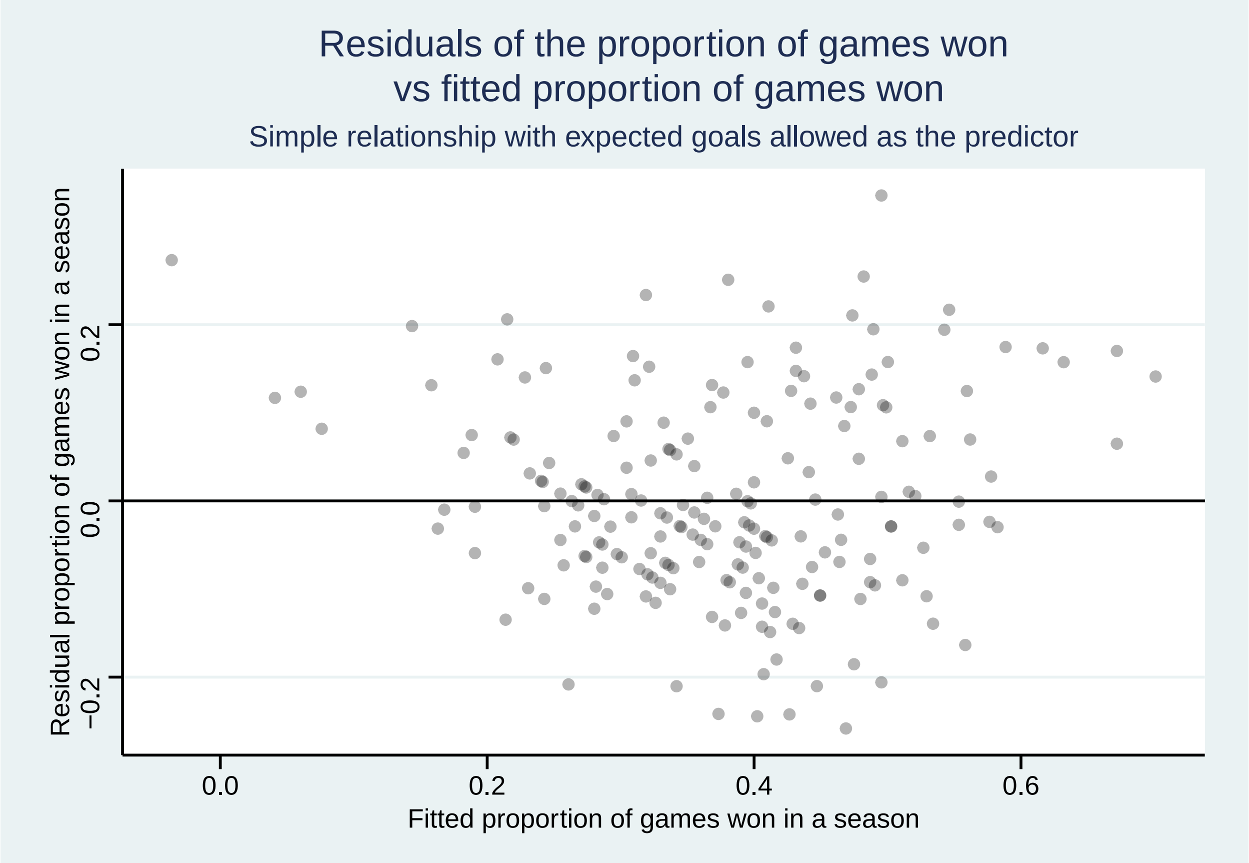


Next, we performed residual analysis by plotting the residual plot of the model between the proportion of games won and the expected goals allowed.

```r
model_xGA <- lm(W_prop ~ xGA, data = european_soccer)
ggplot(data = european_soccer, aes(x = xGA, y = W_prop)) +
  geom_point(alpha = 0.3) +
  geom_smooth(method = "lm", se = FALSE) +
  theme_stata() +
  labs(title = "Proportion of games won in a season\n vs. expected goals allowed (xGA)",
       x = "Expected goals allowed (xGA)",
       y = "Proportion of games won in a season")
```

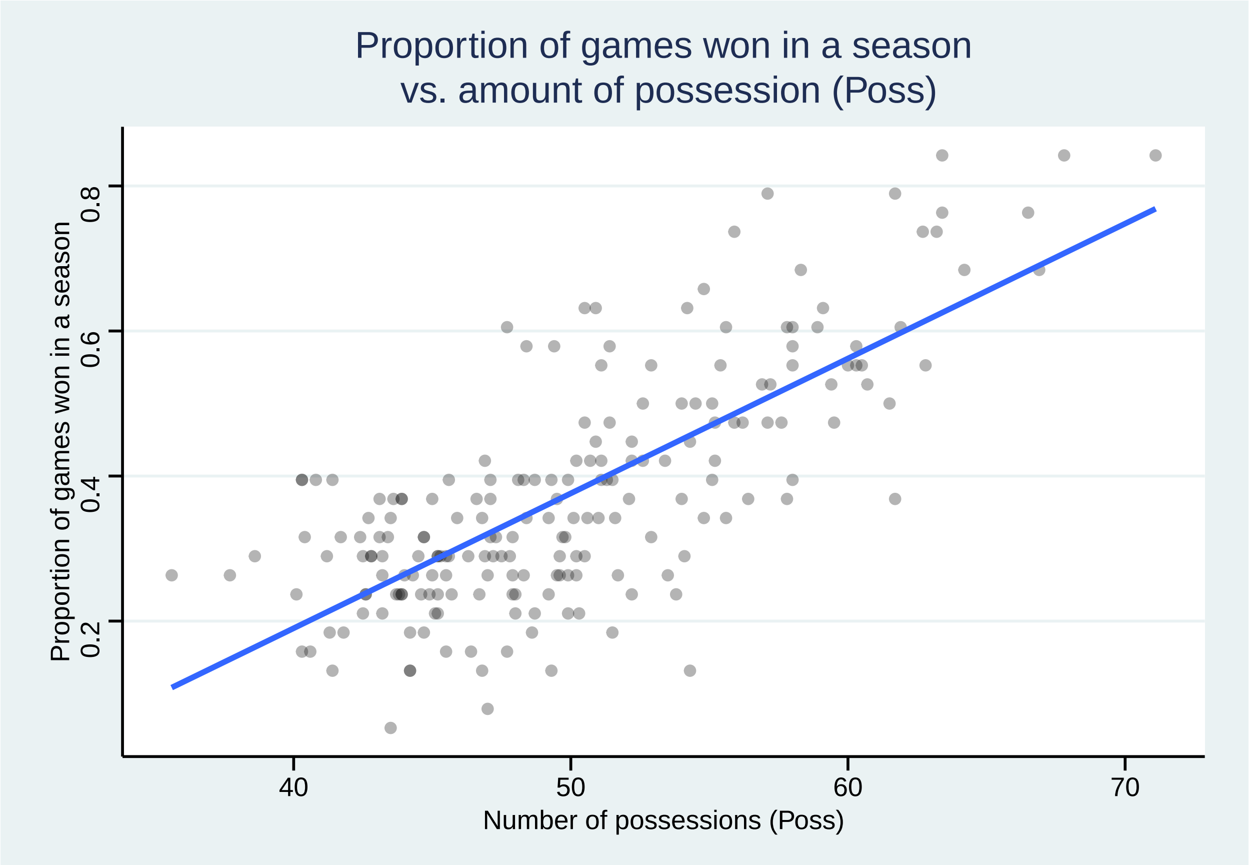


```
model_xGA_fitted <- fitted(model_xGA)
model_xGA_resid <- resid(model_xGA)
ggplot(data = data.frame(model_xGA_fitted, model_xG_resid),
       aes(x = model_xGA_fitted, y = model_xGA_resid)) +
  geom_point(alpha = 0.3) +
  geom_abline(intercept = 0, slope = 0) +
  theme_stata() +
  labs(title = "Residuals of the proportion of games won\n vs fitted proportion of games won",
       subtitle = "Simple relationship with expected goals allowed as the predictor",
       x = "Fitted proportion of games won in a season",
       y = "Residual proportion of games won in a season")
```

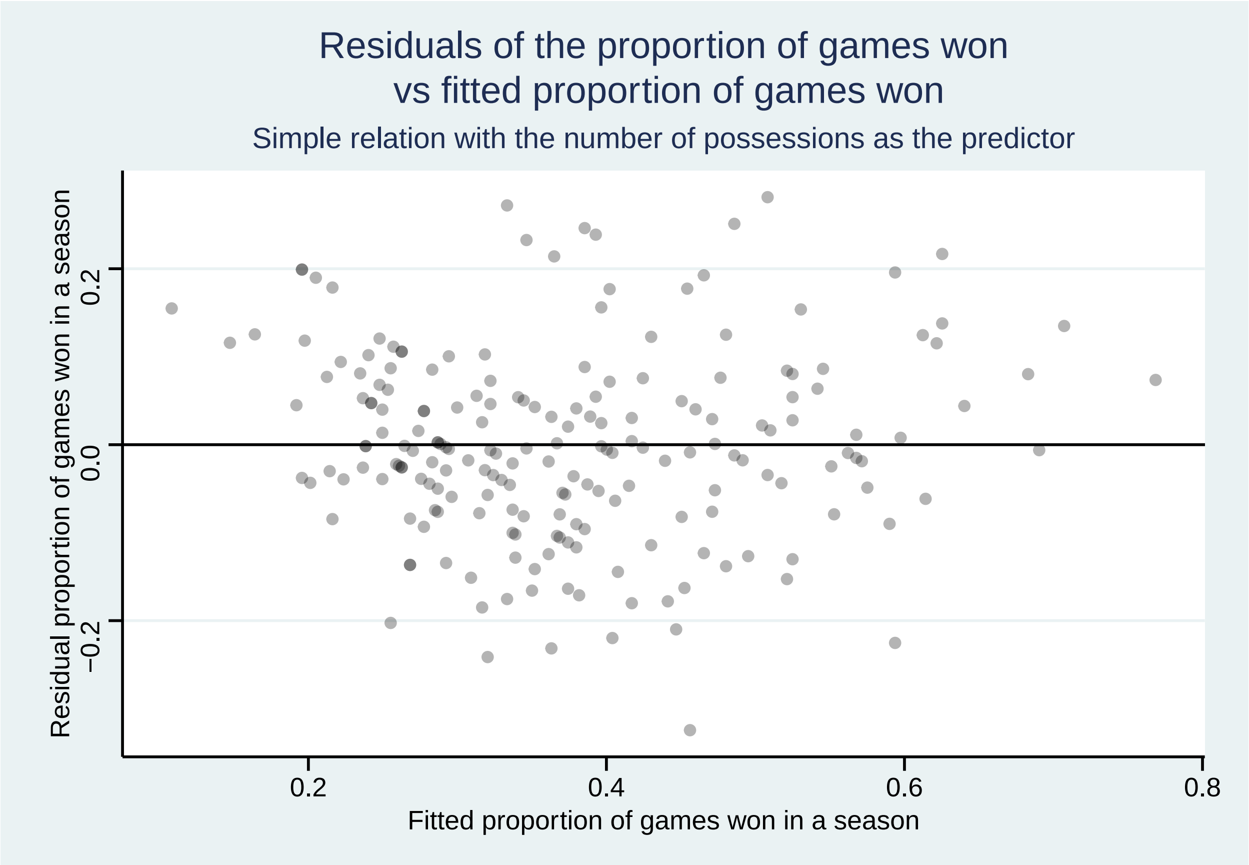


The next variable to consider in a simple relationship with the proportion of games won in a season was the number of possessions.

```r
model_Poss <- lm(W_prop ~ Poss, data = european_soccer)
ggplot(data = european_soccer, aes(x = Poss, y = W_prop)) +
  geom_point(alpha = 0.3) +
  geom_smooth(method = "lm", se = FALSE) +
  theme_stata() +
  labs(title = "Proportion of games won in a season\n vs. amount of possession (Poss)",
       x = "Number of possessions (Poss)",
       y = "Proportion of games won in a season")
```



```
model_Poss_fitted <- fitted(model_Poss)
model_Poss_resid <- resid(model_Poss)
ggplot(data = data.frame(model_Poss_fitted, model_Poss_resid),
       aes(x = model_Poss_fitted, y = model_Poss_resid)) +
  geom_point(alpha = 0.3) +
  geom_abline(intercept = 0, slope = 0) +
  theme_stata() +
  labs(title = "Residuals of the proportion of games won\n vs fitted proportion of games won",
       subtitle = "Simple relation with the number of possessions as the predictor",
       x = "Fitted proportion of games won in a season",
       y = "Residual proportion of games won in a season")
```



The observations we made on each of the three previous simple relationships were similar, in that none of the relationships look curved, so a polynomial transformation will not be necessary. However, there are some issues regarding the distribution of the residuals for the predicted proportion of games won in a season, where the variance is non-constant and not-normally distributed about the line of best fit in the proportion of games won in a different season for different values of the expected goals, expected goals allowed, and the number of possessions as a proportion of attempted passes; therefore, a log transformation would be the way to remedy this.

We summarize the issues with the distribution of the residuals in the following table, which displays the p-values that were produced from running the Shapiro-Wilk and the Breusch-Pagan tests on the distributions of the residuals from the three simple models.

We were able to quickly run the Shapiro-Wilk and Breusch-Pagan tests on each of the models using the `diagnostics` function we have included in the appendix.

```r
diagnostics(model = model_xG, plotit = FALSE, testit = TRUE)
diagnostics(model = model_xGA, plotit = FALSE, testit = TRUE)
diagnostics(model = model_Poss, plotit = FALSE, testit = TRUE)
```

Predictor variable	Shapiro-Wilk test p-value	Breusch-Pagan test p-value
Expected goals	0.0127497	0.0217262
Expected goals allowed	0.0496668	0.1040044
Proportion of possessions	0.5846183	0.2331615

However, we believed it was more important for our model to be interpretable, and since the flaws of the distribution of the variance of the residuals were not too severe, we decided not to transform those three variables.

Additionally, we looked at the individual p-values of the t-tests that were run on the coefficient of the predictor, with the null hypothesis that there was no linear relationship between the predictor (expected goals, expected goals allowed, number of possessions as a proportion of attempted passes) and the proportion of games won in a season, which is summarized in the following:

```r
summary(model_xG)$coefficients[2, 4]
summary(model_xGA)$coefficients[2, 4]
summary(model_Poss)$coefficients[2, 4]
```

Predictor variable	t-test p-value
Expected goals	0
Expected goals allowed	0
Proportion of possessions	0

Since the p-values were very low, we concluded that there was strong evidence that indicates the statement of there not being a relationship between the proportion of games won, and the expected goals, expected goals allowed, number of possessions as a proportion of attempted passes, individually, is false. Hence, these predictor variables are viable to include in our final model.

We also created prediction intervals to extrapolate values for certain response variables in order to see how big of a difference in win proportion to expect when comparing teams that had small expected goals values and possessions compared to teams that had large expected goals values and possessions.

We generated the following 99% prediction intervals for hypothetical teams that have an expected goal value of 20 and 100.

```r
new_goals = data.frame(xG = c(20, 100))
predict(model_xG, newdata = new_goals, interval = "prediction", level = 0.99)
```

Hypothetical expected goals	Midpoint	Lower bound	Upper bound
20	0.0463222	-0.1902233	0.2828677
100	0.9306827	0.6881729	1.1731924

In context, this would mean that…

- For teams that have an expected goal value of 20, we would expect to observe that these teams' win proportion of a season to fall between -0.1902233 and 0.2828677 for 99% of the seasons that these teams participated in, assuming that their expected goal value stays the same.
  - Since it does not make sense for a team to have a negative win proportion, it is better to say that we would approximate 99% of all seasons participated for the teams that have an expected goal value of 20 to have a win proportion of at most 0.2828677, assuming that their expected goal value stays the same.
- For a teams that have a perfect expected goal value of 100, we would expect to observe that these teams have a win proportion of a season to fall between 0.6881729 and 1.1731924 for 99% of the seasons that these teams participated in, assuming that their expected goal value stays the same.
  - Since it does not make sense for a team to have a win proportion over 1, it is better to say that we would approximate that in 99% of all seasons played by teams with an expected goal value of 100, these teams would have a win proportion of at least 0.6881729, assuming that their expected goal value stays the same.

We also generated 99% prediction intervals for hypothetical teams that have possessions as a proportion of attempted passes of 30 and 80.

```r
new_poss = data.frame(Poss = c(30, 80))
predict(model_Poss, newdata = new_poss, interval = "prediction", level = 0.99)
```

Hypothetical possession proportion	Midpoint	Lower bound	Upper bound
30	0.0041203	-0.2863685	0.2946092
80	0.9340742	0.6357252	1.2324233

In context, this would mean that…

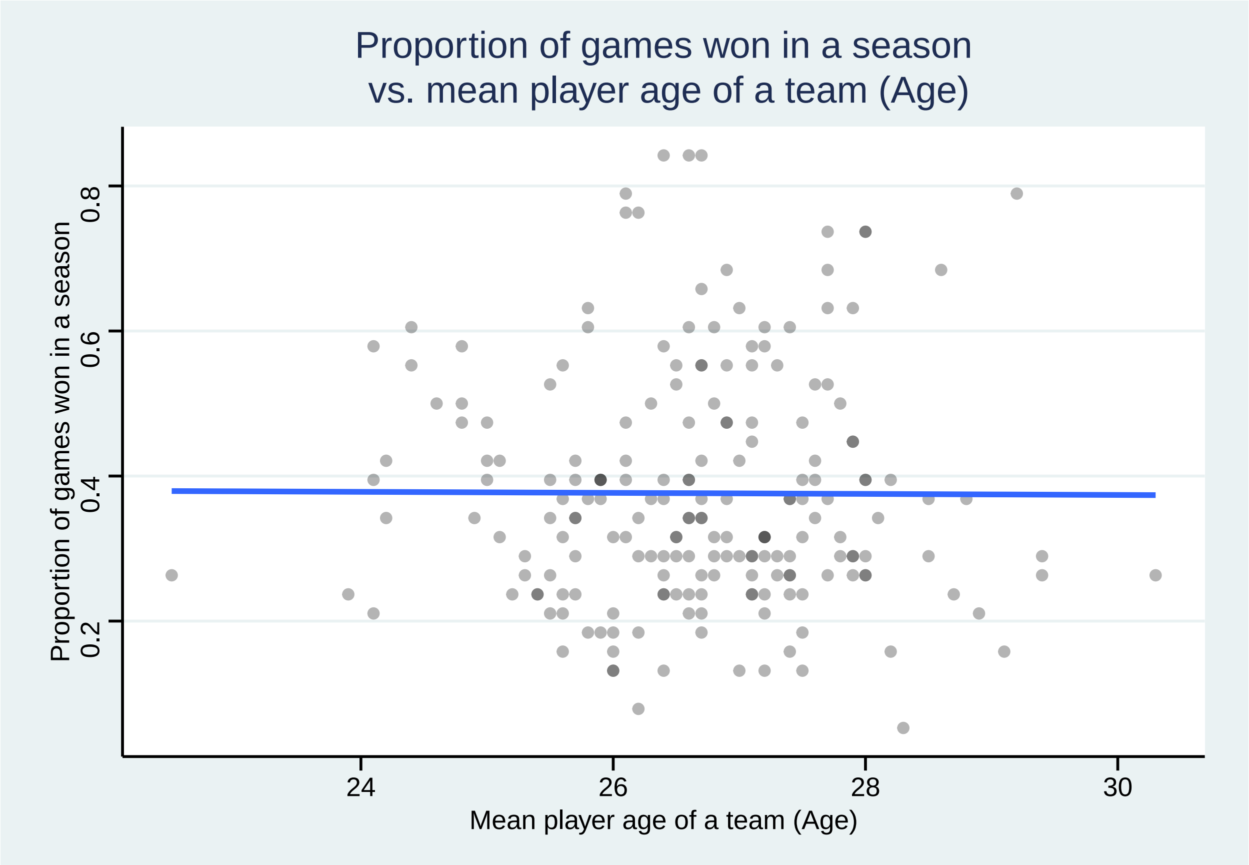
- For teams that have possessions as a proportion of attempted passes of 30, we would observe that for 99% of the seasons these teams participated in, they would have a win proportion that is between -0.2863685 and 0.2946092, assuming their proportion of possessions remains the same in each season.

  - Since it does not make sense to have a negative win proportion, it is better to say that for teams that have possessions as a proportion of attempted passes of 30, we would observe that for 99% of the seasons that these teams participated in, they would have a win proportion of at most 0.2946092, assuming their proportion of possessions remains the same in each season.

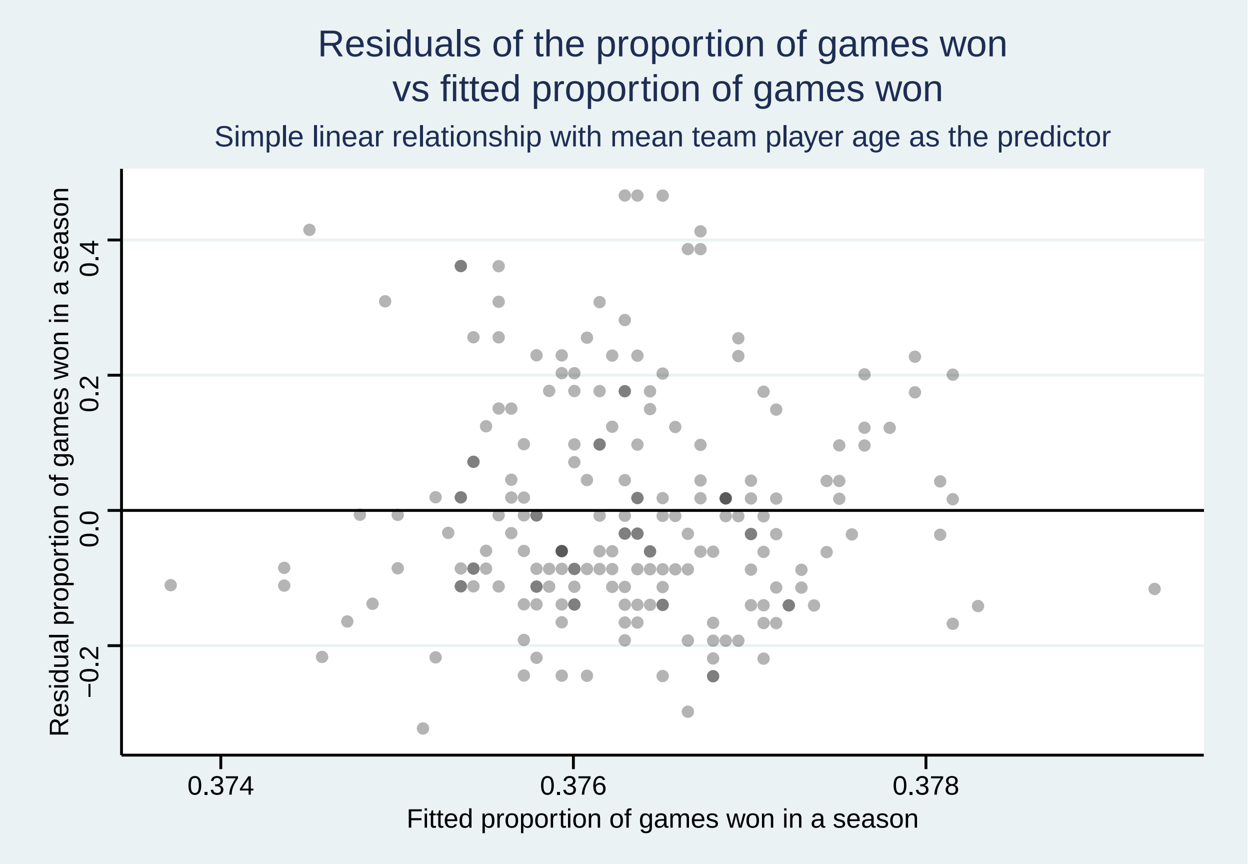
- For teams that have possessions as a proportion of attempted passes of 80, we would observe that for 99% of the seasons these teams participated in, they would have a win proportion that is between 0.6357252 and 1.2324233, assuming their proportion of possessions remains the same in each season.
  - Since it does not make sense to have a win proportion over 1, it is better to say that for teams that have possessions as a proportion of attempted passes of 80, we would observe that for 99% of the seasons that these teams participated in, they would have a win proportion of at least 0.6357252, assuming their proportion of possessions remains the same in each season.

The next variable to consider in a simple relationship with the proportion of games won in a season was the average age of the players on a team.

```
model_Age <- lm(W_prop ~ Age, data = european_soccer)
ggplot(data = european_soccer, aes(x = Age, y = W_prop)) +
  geom_point(alpha = 0.3) +
  geom_smooth(method = "lm", se = FALSE) +
  theme_stata() +
  labs(title = "Proportion of games won in a season\n vs. mean player age of a team (Age)",
       x = "Mean player age of a team (Age)",
       y = "Proportion of games won in a season")
```



```
model_Age_fitted <- fitted(model_Age)
model_Age_resid <- resid(model_Age)
ggplot(data = data.frame(model_Age_fitted, model_Age_resid),
       aes(x = model_Age_fitted, y = model_Age_resid)) +
  geom_point(alpha = 0.3) +
  geom_abline(intercept = 0, slope = 0) +
  theme_stata() +
  labs(title = "Residuals of the proportion of games won\n vs fitted proportion of games won",
       subtitle = "Simple linear relationship with mean team player age as the predictor",
       x = "Fitted proportion of games won in a season",
       y = "Residual proportion of games won in a season")
```



Given the small value of the coefficient for the mean player age of a team in the simple model, we suspected that there would be, at best, a very weak relationship between the proportion of games won and the mean player age of a team.

```
summary(model_Age)
```

We viewed the p-value of the t-test for the coefficient of the predictor, and given the large p-value of 0.943597, we concluded that using the mean player age as an additive predictor would not add any predictive power into our final model, but we may consider it for an interaction term.

In order to quantify how much predictive power would increase by adding the mean player age as a predictor, we decided to add the mean player age as predictor to the model that had the highest R^2 value out of the simple models that had expected goals, expected goals allowed, and number of possessions as a proportion of attempted passes, which are summarized in the following:

```
summary(model_xG)$r.squared
summary(model_xGA)$r.squared
summary(model_Poss)$r.squared
```

Predictor variable	R^2
Expected goals	0.7034973
Expected goals allowed	0.5239362
Possessions as a propotion of attempted passes	0.5602088

We ran an F-test to determine the additional predictive power by adding the mean player age as a predictor to the expected goals model.

```
anova(model_xG, lm(W_prop ~ xG + Age, data = european_soccer))
```

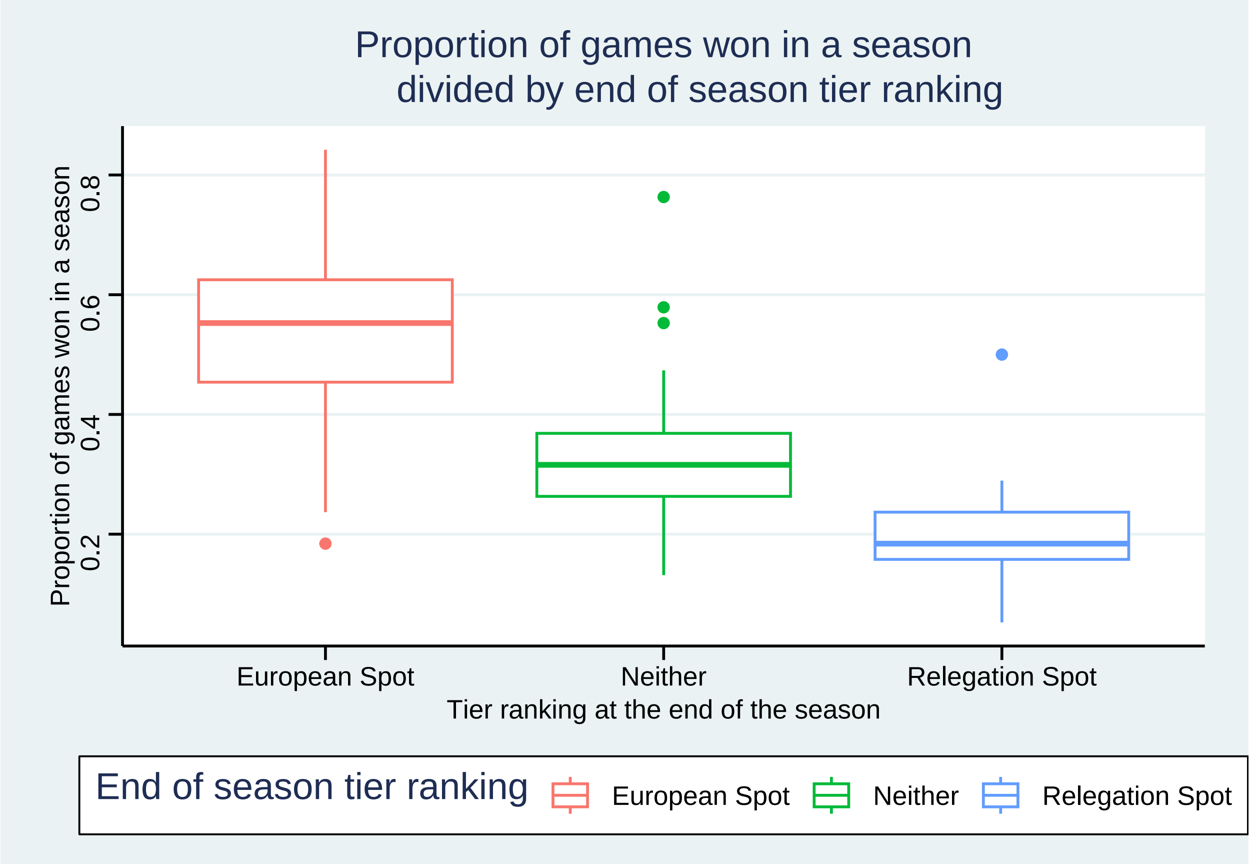
```
## Analysis of Variance Table
##
## Model 1: W_prop ~ xG
## Model 2: W_prop ~ xG + Age
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1    198 1.5839
## 2    197 1.5704  1  0.013539 1.6984  0.194
```

Given p-value being as large as 0.19402, we had more evidence to not include the mean player age as an additive predictor into our model.

The final variable to consider in a relationship with the proportion of games won in a season was the tier ranking of the team at the end of the season, which was the only categorical variable out of the predictors we chose to potentially include in our final model.

By plotting the proportion of games won and stratifying by the tier of the ranking at the end of the season, we can see there is a visible difference in the distribution of the proportion of games won.

```r
ggplot(data = european_soccer, aes(x = Notes, y = W_prop, color = Notes)) +
  geom_boxplot() +
  scale_color_brewer(palette = "Pastel2") +
  theme_stata() +
  labs(x = "Tier ranking at the end of the season",
       y = "Proportion of games won in a season",
       title = "Proportion of games won in a season
       divided by end of season tier ranking") +
  scale_colour_discrete(name = "End of season tier ranking")
```



In order to quantify how much adding the end of season tier ranking as a predictor would be, we decided to add tier ranking as predictor to the model that had the highest R^2 value out of the simple models, which was the expected goals model.

We thrn ran an F-test to compare the simple model using only the expected games as a predictor and using expected games and the tier ranking as predictors.

```r
anova(model_xG, lm(W_prop ~ xG + Notes, data = european_soccer))
```

```
## Analysis of Variance Table
##
```

```
## Model 1: W_prop ~ xG
## Model 2: W_prop ~ xG + Notes
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)    
## 1    198 1.5839                                  
## 2    196 1.1051  2   0.47887 42.467 4.764e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Given the small p-value of $4.7635797 \times 10^{-16}$, we concluded that adding the tier ranking achieved by the team as the end of the season as a categorical dummy variable would greatly improve the predictive power of our model.

We decided on one of the candidate models to use expected goals (xG), expected goals allowed (xGA), number of possessions as a proportion of attempted passes (Poss), and the tier ranking of the team at the end of the season (data set uses column name `Notes`) as predictors for the proportion of games won in a season ($\mathrm{W}_{\mathrm{prop}}$), which can be written as the following:

```
model_final_i <- lm(W_prop ~ xG + xGA + Poss + Notes, data = european_soccer)
summary(model_final_i)
```

$$y_{\mathrm{W}_{\mathrm{prop}}} = 0.2120 + 0.006140x_{\mathrm{xG}} - 0.003698x_{\mathrm{xGA}} + 0.002031x_{\mathrm{Poss}} - 0.07553x_{\mathrm{Neither}} - 0.1309x_{\mathrm{Relegation\ Spot}},$$

where

- $x_{\mathrm{Neither}} = 1$ if the tier ranking of the team at the end of the season is neither, or equal to 0 otherwise, and
- $x_{\mathrm{Relegation\ Spot}} = 1$ if the tier ranking of the team at the end of the season is Relegation spot, or equal to 0 otherwise.

Interaction terms

We also chose to examine two more models produced from backwards variable selection with both AIC and BIC as the criterion and using two-way interaction terms between the expected goals (xG), expected goals allowed (xGA), number of possessions as a proportion of attempted passes (Poss), average player age in a team (Age), and tier ranking at the end of the season (data set uses column name `Notes`). We began with the full model using the following variables:

- Expected goals
- Expected goals allowed
- Amount of possession
- Tier ranking at the end of the season
- All possible two-way interactions between the above four variables along with the average player age

```r
model_int_full <- lm(W_prop ~ xG + xGA + Poss + Notes +
                     (xG + xGA + Poss + Notes + Age) ^ 2, data = european_soccer)
model_aic <- step(model_int_full, direction = "backward", trace = 0)
summary(model_aic)
```

The model produced by backwards variable selection using AIC is the following:

$$\begin{aligned}y_{\text{W}_\text{prop}} = &-1.4737 + 0.005644x_{\text{xG}} + 0.02699x_{\text{xGA}} + 0.01318x_{\text{Poss}} - 0.07311x_{\text{Neither}} + \\ &-0.1448x_{\text{Relegation Spot}} + 0.04263x_{\text{Age}} - 0.0002303x_{\text{xGA}}x_{\text{Poss}} - 0.0007121x_{\text{xGA}}x_{\text{Age}},\end{aligned}$$

where

- $x_{\text{Neither}} = 1$ if the tier ranking of the team at the end of the season is neither, or equal to 0 otherwise, and
- $x_{\text{Relegation Spot}} = 1$ if the tier ranking of the team at the end of the season is Relegation spot, or equal to 0 otherwise.

```r
model_int_full <- lm(W_prop ~ xG + xGA + Poss + Notes +
                     (xG + xGA + Poss + Notes + Age) ^ 2, data = european_soccer)
model_bic_n <- length(resid(model_int_full))
model_bic <- step(model_int_full, direction = "backward", k = log(model_bic_n), trace = 0)
summary(model_bic)
```

The model produced by backwards variable section using BIC is the following:

$$\begin{aligned}y_{\text{W}_\text{prop}} = &-0.3313 + 0.005501x_{\text{xG}} + 0.008182x_{\text{xGA}} + 0.01336x_{\text{Poss}} - 0.07613x_{\text{Neither}} + \\ &-0.1466x_{\text{Relegation Spot}} + 0.04263x_{\text{Age}} - 0.0002367x_{\text{xGA}}x_{\text{Poss}},\end{aligned}$$

where

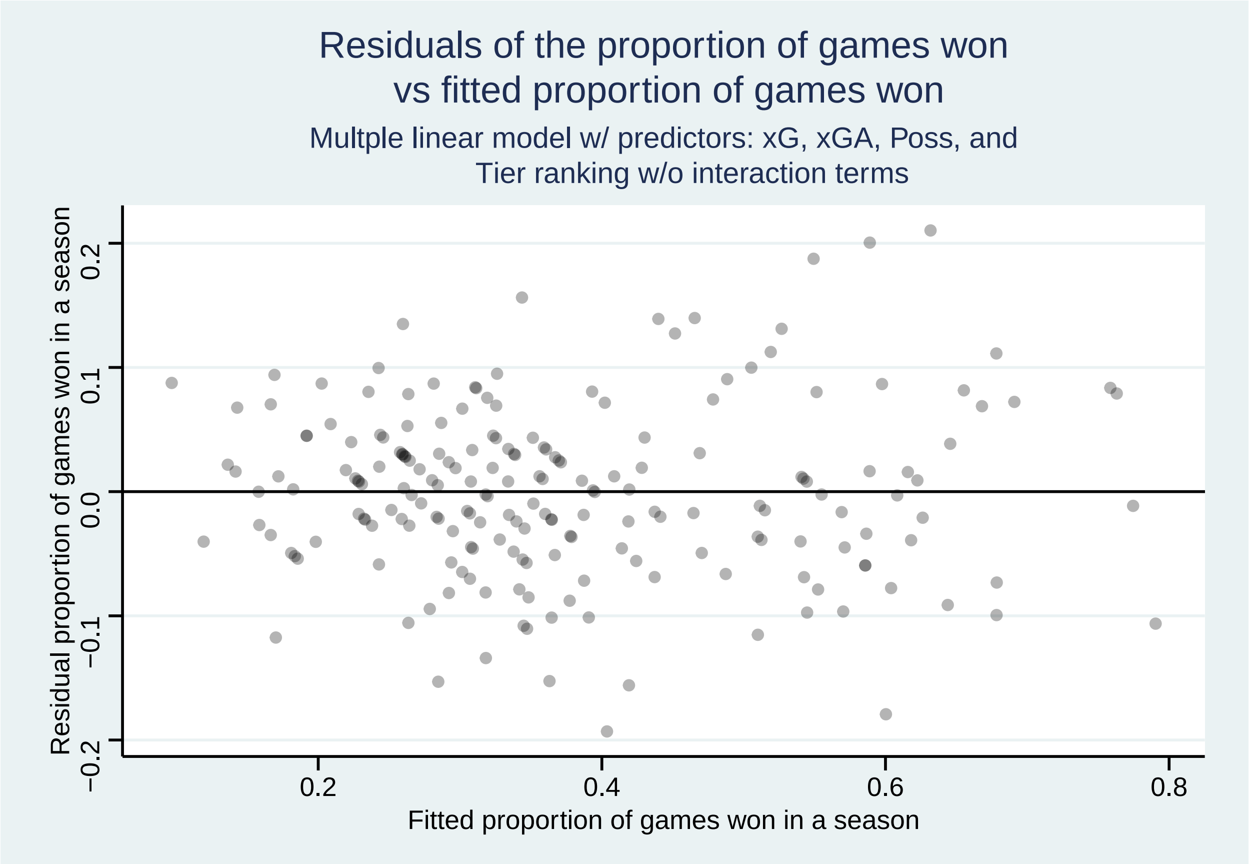
- $x_{\text{Neither}} = 1$ if the end of season tier ranking of the team is neither or equal to 0 otherwise, and
- $x_{\text{Relegation Spot}} = 1$ if the tier ranking of the team is Relegation spot or equal to 0 otherwise.

Residual diagnostics

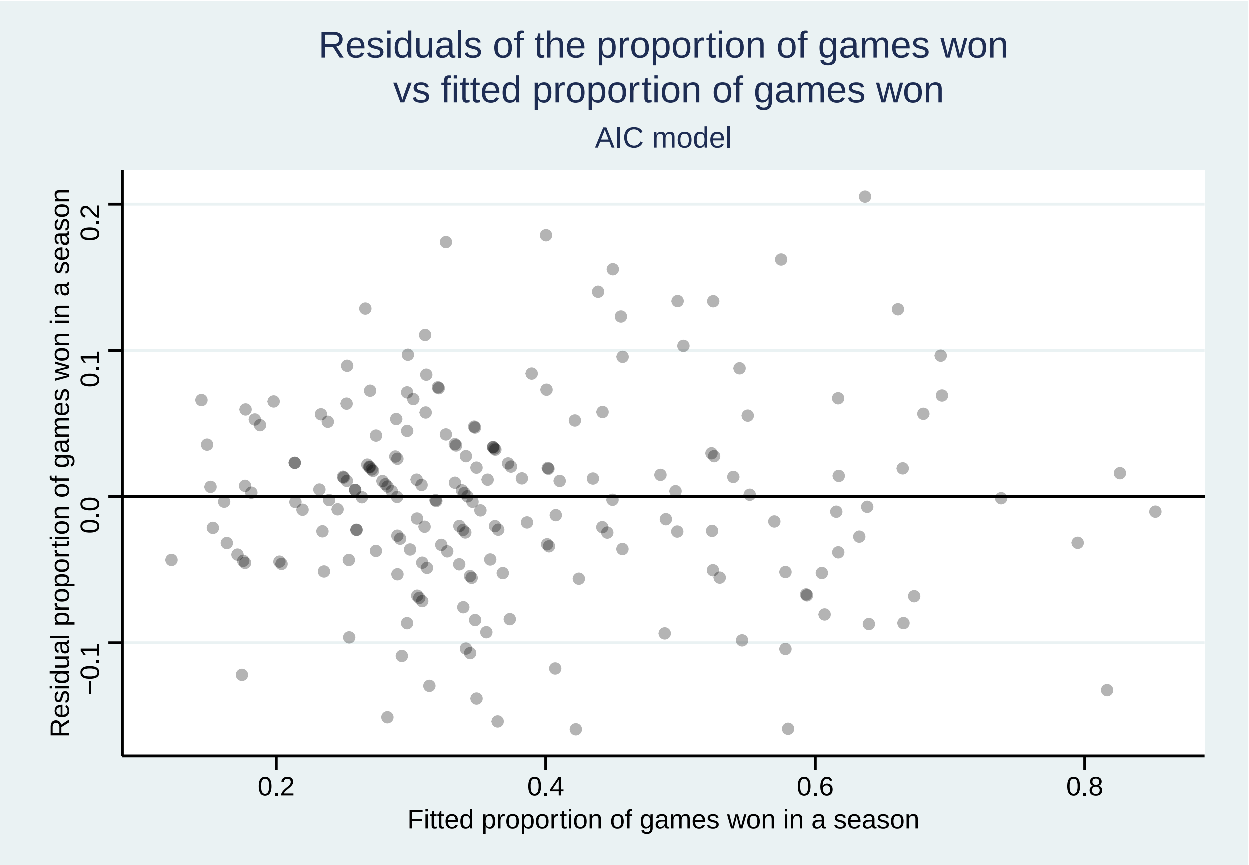
Next, we examined the distribution of residuals from the three aforementioned models, which were:

- The multiple linear additive model that did not have interaction terms and only used expected goals, expected goals allowed, possessions as a proportion of attempted passes, and tier ranking at the end of the season
- Backwards variable selection model that used AIC as the criterion
- Backwards variable selection model that used BIC as the criterion

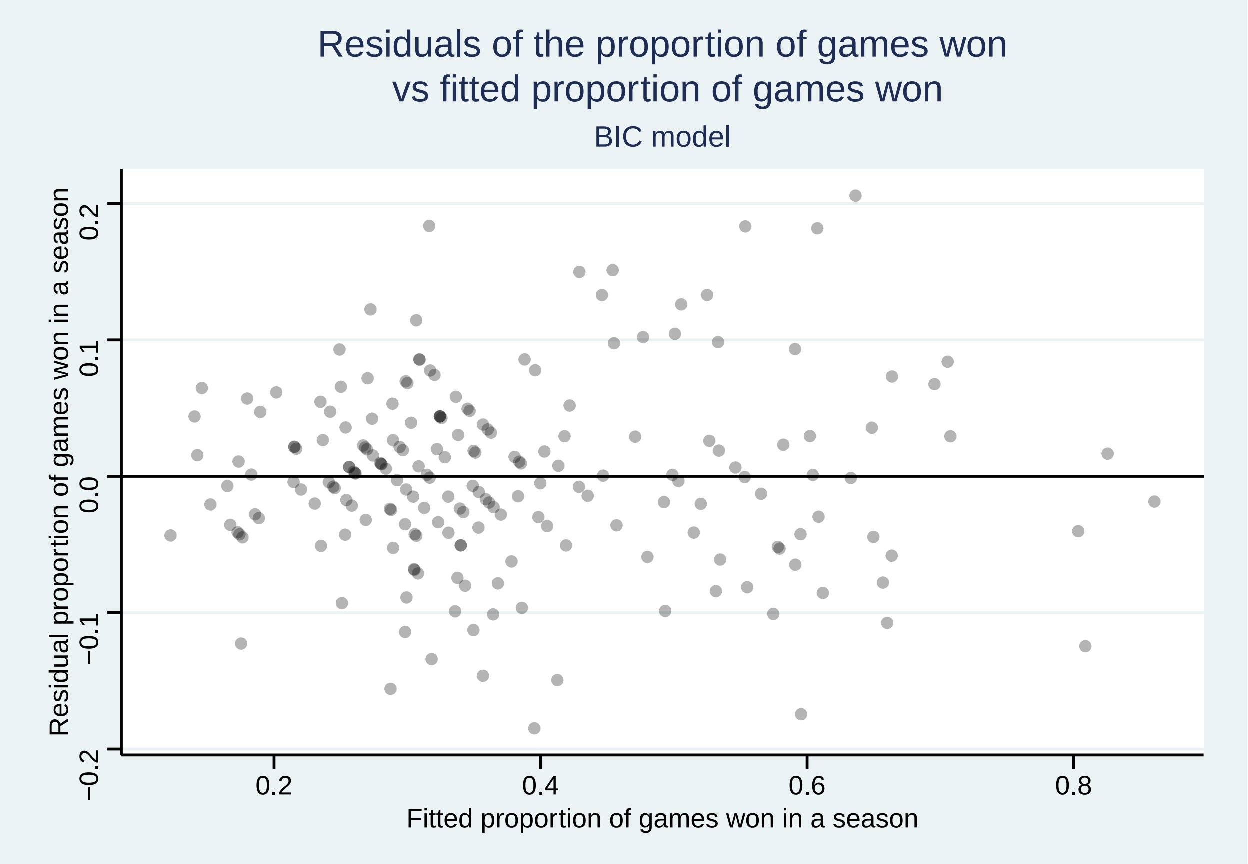
```r
model_final_fitted <- fitted(model_final_i)
model_final_resid <- resid(model_final_i)
ggplot(data = data.frame(model_final_fitted, model_final_resid),
       aes(x = model_final_fitted, y = model_final_resid)) +
  geom_point(alpha = 0.3) +
  geom_abline(intercept = 0, slope = 0) +
  theme_stata() +
  labs(title = "Residuals of the proportion of games won\n vs fitted proportion of games won",
       subtitle = "Multple linear model w/ predictors: xG, xGA, Poss, and
       Tier ranking w/o interaction terms",
       x = "Fitted proportion of games won in a season",
       y = "Residual proportion of games won in a season")
```



```
model_aic_fitted <- fitted(model_aic)
model_aic_resid <- resid(model_aic)
ggplot(data = data.frame(model_aic_fitted, model_aic_resid),
       aes(x = model_aic_fitted, y = model_aic_resid)) +
  geom_point(alpha = 0.3) +
  geom_abline(intercept = 0, slope = 0) +
  theme_stata() +
  labs(title = "Residuals of the proportion of games won\n vs fitted proportion of games won",
       subtitle = "AIC model",
       x = "Fitted proportion of games won in a season",
       y = "Residual proportion of games won in a season")
```



```
model_bic_fitted <- fitted(model_bic)
model_bic_resid <- resid(model_bic)
ggplot(data = data.frame(model_bic_fitted, model_bic_resid),
       aes(x = model_bic_fitted, y = model_bic_resid)) +
  geom_point(alpha = 0.3) +
  geom_abline(intercept = 0, slope = 0) +
  theme_stata() +
  labs(title = "Residuals of the proportion of games won\n vs fitted proportion of games won",
       subtitle = "BIC model",
       x = "Fitted proportion of games won in a season",
       y = "Residual proportion of games won in a season")
```



We observed that from all three distributions of residuals from the three models, the variance of the residuals tends to increase in the range of fitted proportion of games won in a season between 0.3 and 0.6. The first multiple linear model and the BIC model appears to have residuals that look normally distributed along the $y = 0$ line, but the AIC model's residuals tend to have positive skew.

To quantify how large of an issue the non-constant residual variance and skewed residual distribution, we performed Shapiro-Wilk and Breusch-Pagan tests on the residuals for all three models. The p-values from the tests performed is summarized in the following:

```
diagnostics(model = model_final_i, plotit = FALSE, testit = TRUE)
diagnostics(model = model_aic, plotit = FALSE, testit = TRUE)
diagnostics(model = model_bic, plotit = FALSE, testit = TRUE)
```

Model	Shapiro-Wilk test p-value	Breusch-Pagan test p-value
Multiple linear model (w/o interaction terms)	0.4227991	0.0033629
AIC model	0.0780570	0.0317978
BIC model	0.0721157	0.0064687

Given the small p-values from the Breusch-Pagan test for all three models and the small p-values from the Shapiro-Wilk test for the models produced by backwards variable selection using AIC and BIC as criterion, we concluded that there were definite issues with the three models where the multiple linear model had residuals that had non-constant variance, the AIC model had residuals that were not normally distributed about the line of best fit and did not have constant variance, and the BIC model also had residuals that were not normally distributed about the line of best fit, that also did not have constant variance.

We decided to perform outlier diagnostics with hopes of eliminating high influence points from the data set so that models that had a smaller degree of issues with the distribution of their residuals could be constructed using data sets that included only low influence teams.

Outlier diagnostics

In order to determine which teams were considered outliers, we used the Cook's distance to determine the influence that each team had on the resulting model. We decided that teams that had a Cook's distance greater than $\frac{4}{n}$, where n is the number of teams in the data set.

```
sum(cooks.distance(model_final_i) > 4 / length(cooks.distance(model_final_i)))
sum(cooks.distance(model_aic) > 4 / length(cooks.distance(model_aic)))
sum(cooks.distance(model_bic) > 4 / length(cooks.distance(model_bic)))
```

From the multiple linear model without interaction terms, there were 15 outliers; from the AIC model, there were 14 outliers. From the BIC model, there were 13 outliers.

From the prior residual diagnostics, we hope that by removing the outliers, the resulting data set that has only lower influence points will produce models where the distribution of the residuals resembles more of a normal distribution about the line of best fit and that the variance of the residuals becomes more consistent.

After removing the respective outliers from the data set, (15 for the multiple linear model without interaction terms, 14 for the AIC model, and 13 for the BIC model), we constructed the three candidate models again, but using the data sets that contained low influence points for each respective model.

```
model_final_low_inf_idx <- cooks.distance(model_final_i) <= 4 / length(cooks.distance(model_final_i))
model_final_ii <- lm(W_prop ~ xG + xGA + Poss + Notes, data = european_soccer,
                     subset = model_final_low_inf_idx)
```

```
model_aic_low_inf <- cooks.distance(model_aic) <= 4 / length(cooks.distance(model_aic))
european_soccer_aic_low_inf <- european_soccer[model_aic_low_inf, ]
model_int_full_li_aic <- lm(W_prop ~ xG + xGA + Poss + Notes +
                              (xG + xGA + Age + Notes + Poss)^2,
                            data = european_soccer_aic_low_inf)
model_aic_li <- step(model_int_full_li_aic,
                     direction = "backward",
                     trace = 0)

model_bic_low_inf <- cooks.distance(model_bic) <= 4 / length(cooks.distance(model_bic))
european_soccer_bic_low_inf <- european_soccer[model_bic_low_inf, ]
model_int_full_li_bic <- lm(W_prop ~ xG + xGA + Poss + Notes +
                              (xG + xGA + Age + Notes + Poss)^2,
                            data = european_soccer_bic_low_inf)

model_bic_li_n <- length(resid(model_int_full_li_bic))
model_bic_li <- step(model_int_full_li_bic,
                     direction = "backward",
                     k = log(model_bic_li_n),
                     trace = 0)
```

Then, we compared the adjusted R^2 value before and after removing the outliers and summarized our findings:

```
summary(model_final_i)$adj.r.squared
summary(model_final_ii)$adj.r.squared

summary(model_aic)$adj.r.squared
summary(model_aic_li)$adj.r.squared

summary(model_bic)$adj.r.squared
summary(model_bic_li)$adj.r.squared

summary(model_final_ii)$adj.r.squared - summary(model_final_i)$adj.r.squared
summary(model_aic_li)$adj.r.squared - summary(model_aic)$adj.r.squared
summary(model_bic_li)$adj.r.squared - summary(model_bic)$adj.r.squared
```

Model	Adj. R-Squared from Model w/ High Inf.	Adj. R-Squared from Model w/o High Inf.	Difference
Multiple linear model (w/o interaction terms)	0.8188286	0.8554364	0.0366078
AIC model	0.8311125	0.8699546	0.0388421

	Adj. R-Squared from Model	Adj. R-Squared from Model	
Model	w/ High Inf.	w/o High Inf.	Difference
BIC model	0.8280759	0.8659728	0.0378970

Given the increase in the value of the adjusted R^2 for all three models after removing the outliers for each individual model, we decided to replace the former three models that included the high influence teams in the data set with the three new models that used data sets that included only the low influence teams.

We also decided to perform residual diagnostics once again on the new models that only accounts for low influence teams:

```r
diagnostics(model = model_final_ii, plotit = FALSE, testit = TRUE)
diagnostics(model = model_aic_li, plotit = FALSE, testit = TRUE)
diagnostics(model = model_bic_li, plotit = FALSE, testit = TRUE)
```

Model	Shapiro-Wilk test p-value	Breusch-Pagan test p-value
Multiple linear model (w/o interaction terms)	0.8038321	0.2043908
AIC model	0.9609045	0.1037089
BIC model	0.9825724	0.3566308

We observed that the p-values of the Shapiro-Wilk test and the Breusch-Pagan test performed on the distributions of the residuals are much larger than when they were performed on the residuals of the models that accounted for high influence teams. Hence, we decided to keep these three models as our candidate models.

Comparison of final candidate models.

We arrived with three candidate models to best predict the proportion of games won by a team in a season:

- Multiple linear model that has the following predictors: expected goals (xG), expected goals allowed (xGA), number of possessions as a proportion of attempted passes (Poss), and the tier ranking of the team at the end of the season (data set uses column name `Notes`).
- Backwards variable selection model using AIC as the criterion, with the same additive predictors as the above multiple linear model along with average team player age (Age) and two interaction terms:
  - The interaction between expected goals and number of possessions as a proportion of attempted passes
  - The interaction between the expected goals allowed and the number of possessions as a proportion of attempted passes.
- Backwards variable selection model using BIC as the criterion, with the same additive predictor as the first multiple linear model above, along with a single interaction term:

  - The interaction between expected goals allowed and the number of possessions as a proportion of attempted passes

In order to compare the models, we summarized the relevant details of all three models as a table, using the `loocv_rmse` and `rmse` functions we have included in the appendix to compute the LOOCV RMSE and the RMSE of each model, along with the `diagnostics` function to retrieve the p-values from running the Shapiro-Wilk and Breusch-Pagan tests on each model.

```r
summary(model_final_ii)
summary(model_aic_li)
summary(model_bic_li)

rmse(model_final_ii)
rmse(model_aic_li)
rmse(model_bic_li)

loocv_rmse(model_final_ii)
loocv_rmse(model_aic_li)
loocv_rmse(model_bic_li)

diagnostics(model = model_final_ii, plotit = FALSE, testit = TRUE)
diagnostics(model = model_aic_li, plotit = FALSE, testit = TRUE)
diagnostics(model = model_bic_li, plotit = FALSE, testit = TRUE)
```

Selection criteria	Multiple linear model	AIC model	BIC model
No. of parameters (inc. intercept)	6.0000000	8.0000000	7.0000000
Adj. R-squared	0.8554364	0.8699546	0.8659728
RMSE	0.0571525	0.0541925	0.0549465
LOOCV-RMSE	0.0590768	0.0565712	0.0568372
Residual Shapiro-Wilk test p-value	0.8038321	0.9609045	0.9825724
Residual Breusch-Pagan test p-value	0.2043908	0.1037089	0.3566308

Results

Most favorable model

When examining the table displaying the selection criterion of the three candidate models, we decided on the BIC model to be the most effective model out of the three. The coefficients of the parameters, the standard error, along with the t-value and the p-value of the t-test performed on each individual predictor are summarized below:

```
summary(model_bic_li)$coefficients
```

Parameter	Coefficient	Standard error	t-value	p-value (two-sided)
Intercept	- 0.3565199	0.1496288	- 2.382696	0.0182282
Expected Goals (xG)	0.0054742	0.0006564	8.340061	0.0000000
Expected goals allowed (xGA)	0.0096279	0.0030237	3.184132	0.0017111
Amount of possession (Poss)	0.0133974	0.0030235	4.431088	0.0000163
Tier ranking: Neither	- 0.0835819	0.0130401	- 6.409602	0.0000000
Tier ranking: Relegation Spot	- 0.1702663	0.0190380	- 8.943484	0.0000000
Interaction between expected goals allowed & amt. of possession	- 0.0002535	0.0000590	- 4.299144	0.0000280

Predictor analysis

Since the intercept begins at a negative number, and the response variable is a proportion, any of the predictors that have positive coefficients are those that help a team in being more capable of winning games in the season. This is obvious with expected goals and number of possessions as a proportion of attempted passes, since a higher expected goal value means that a team is stronger offensively and capable of scoring goals, and the greater the number of possessions means that a team has more opportunities to score a goal.

However, what was surprising was the value of the expected goals allowed, which measures the number of goals the team allowed the enemy to score. Intuitively, there should be a negative relationship between the expected goals allowed and the proportion of games won, since a smaller value for the expected goals allowed would mean that the team is stronger defensively and capable of preventing their opponent from scoring.

Furthermore, for the categorical dummy variable of the tier ranking of the team at the end of the season, the model predicts that teams that have a tier ranking of neither or relegation spot instead of European spot will win a lesser proportion of games in a season on average. This is not a surprise as a team having a high tier ranking at the end of the season implies that a team has won a large proportion of the games that they have played.

As for the interaction term between the expected goals allowed and the number of possessions as a proportion of attempted passes, we conjectured that a team that had many possessions would in theory perform well; however, if they had poor defense and keep allowing the opponent to steal the ball from, and allowing them to score, then the team would likely perform worse.

We can represent our final model using mathematical symbols as follows:

$$\begin{gathered}y_{\mathrm{W_{Prop}}} = -0.3565 + 0.005474x_{\mathrm{xG}} + 0.009628x_{\mathrm{xGA}} + 0.01340x_{\mathrm{Poss}} \\ -0.08358x_{\mathrm{Neither}} - 0.1703x_{\mathrm{Relegation\ Spot}} - 0.0002535x_{\mathrm{xGA}}x_{\mathrm{Poss}}\end{gathered}$$

Where the variables are defined as follows:

Variable	Meaning	Additional notes
$y_{\mathrm{W_{Prop}}}$	Proportion of games won in a season	Measured out of 1
x_{xG}	Expected goals	Measured out of 1
x_{xGA}	Expected goals allowed	Measured out of 1
x_{Poss}	Possessions as a proportion of attempted passes	
x_{Neither}	Tier ranking is neither	Equal to 1 if tier ranking is neither; otherwise, it is 0
$x_{\mathrm{Relegation\ Spot}}$	Tier ranking is Relegation spot	Equal to 1 if tier ranking is Relegation spot; otherwise, it is 0

Reasoning behind the choice of final model

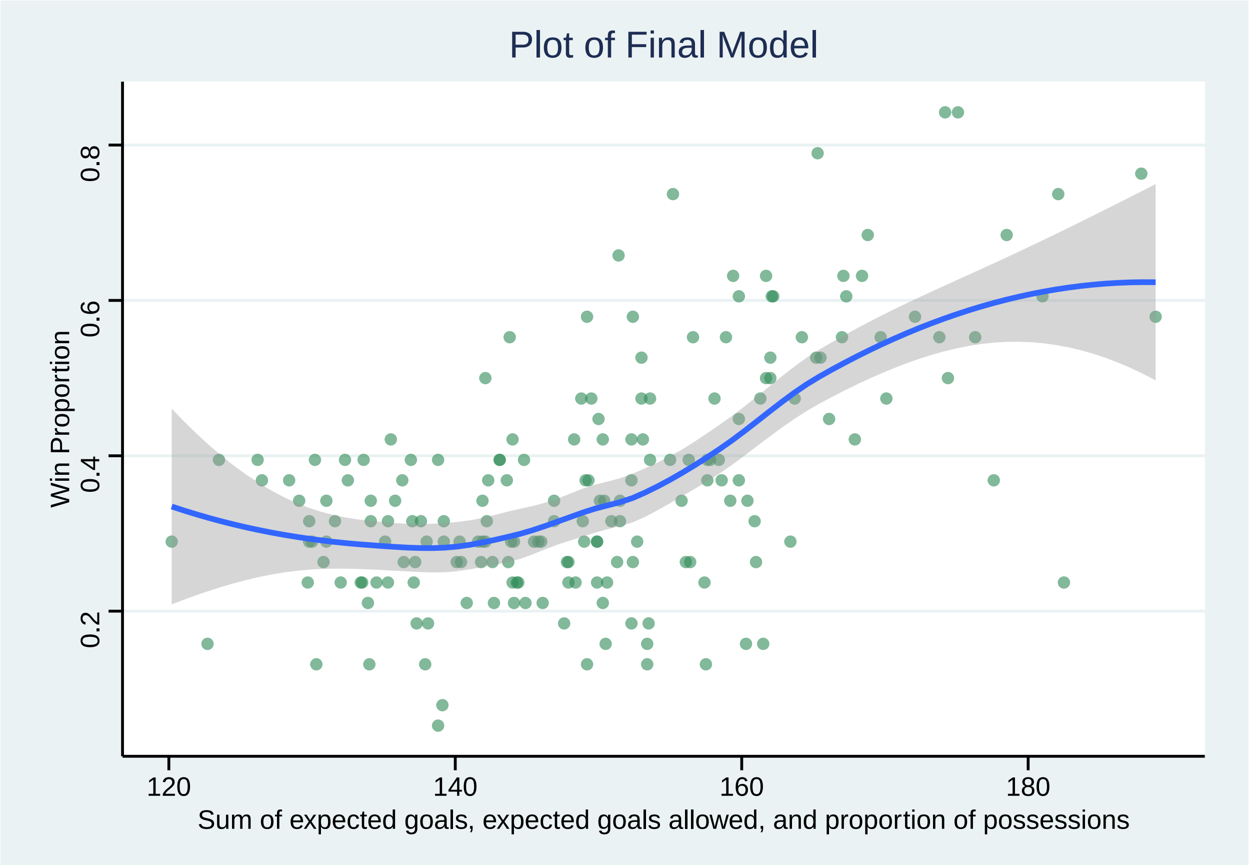
Compared to the other two candidate models that we ended up with, differences in the RMSE, LOOCV-RMSE, adjusted R^2 were marginal. Hence, we looked at the remaining criteria, which were the number of parameters and the p-values produced from running the Shapiro-Wilk and Breusch-Pagan tests on the distribution of the residuals from each of the models. We concluded that the differences in the number of parameters were also marginal, with the least number of parameters being 6, and the most being 8.

However, the differences in the p-values from the Shapiro-Wilk and the Breusch-Pagan were quite substantial. None of the p-values produced were small enough to completely disqualify using one of the models to predict the proportion of games won in a season as the smallest p-value was the Breusch-Pagan test on the AIC model with a magnitude of 0.1037089. However, compared to the other two models, the p-values of those tests on the BIC model's residuals were the biggest, with the largest difference of 0.1787403 from the smallest p-value of the Shapiro-Wilk test on the multiple linear model that did not include interaction terms. Since we did not use any transformations of the variables, we picked the BIC model to ensure that any flaws in the distribution of the residuals is minimized and that we are able to use prediction intervals since prediction intervals require the distribution of the response (the proportion of games won in a season in this case) given a predictor to be normally distributed.

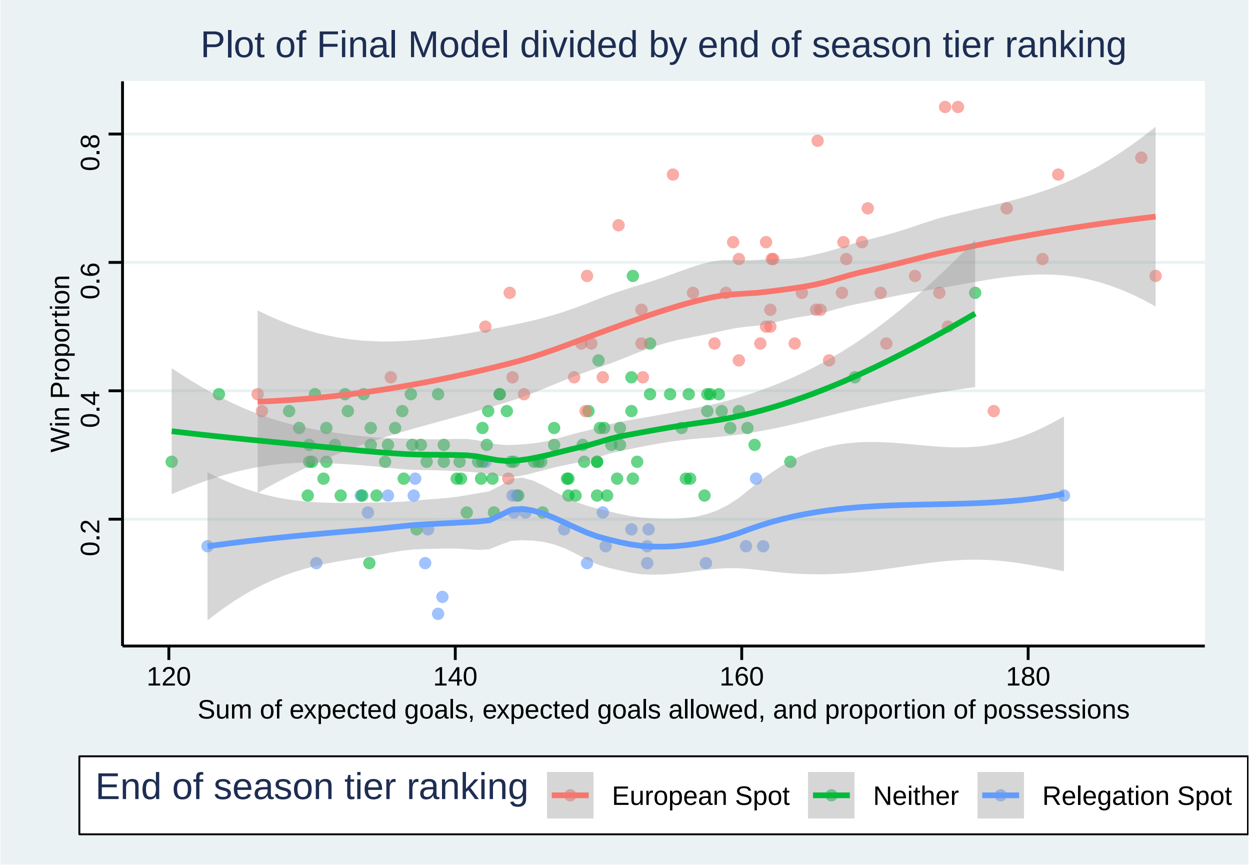
Given the context of the data set and what we hope to achieve with this study, being able to create prediction intervals for the success of a team in a season would be very beneficial.

Graphical visualizations of our model

```
ggplot(data = european_soccer_bic_low_inf,
       mapping = aes(x = xG + xGA + Poss, y = W_prop)) +
  geom_point(col = "Seagreen", alpha = 0.6)+
  geom_smooth(model = model_bic_li) +
  theme_stata()+
  labs(title = "Plot of Final Model",
       x = "Sum of expected goals, expected goals allowed, and proportion of possessions",
       y = "Win Proportion")
```



```r
ggplot(data = european_soccer_bic_low_inf,
       mapping = aes(x = xG + xGA + Poss, y = W_prop, color = Notes)) +
  geom_point(alpha = 0.6) +
  geom_smooth(model = model_bic_li) +
  theme_stata()+
  labs(title = "Plot of Final Model divided by end of season tier ranking",
       x = "Sum of expected goals, expected goals allowed, and proportion of possessions",
       y = "Win Proportion") +
  scale_colour_discrete(name = "End of season tier ranking")
```



Discussion

```
model_bic_li$coefficients
```

Based on the coefficients of the model, we would predict that the single action that would increase the proportion of games won would be for a team to increase the number of possessions as a proportion of attempted passes they obtain during a game; specifically, by we predict that the proportion of games won to increase by 0.0133974 on average for a single increase in the number of possessions as a proportion of passes in a game, assuming that all other factors are kept the same. Additionally, the single factor that decreases the proportion of games won would be the tier ranking achieved by the team at the end of the season, where being a Relegation Spot instead of European spot decreases the proportion of games on average by 0.1702663, assuming that all other factors like expected goals are kept the same.

Additionally, from during our method of finding candidate models, we concluded that the average player age has little impact on the proportion of games won in a season. We conjecture that this may be due to younger players having greater fluid intelligence, where they are in their physical and cognitive peak, but they have less experience compared to the older players that have the greater crystallized intelligence that have the intuition of the game.

```
summary(model_bic_li)$adj.r.squared
```

When assessing the viability of our model we concluded that after adjusting for correlation due to random change, approximately 86.5972832% of the variance in the proportion of games in a season is explained by our model.

One of the limitations that we had discussed was the usefulness of this model to predict the proportion of games won in the middle of a season because we become unable to use the tier ranking of the team as a predictor since that is only known once the season has ended.

Moving on, during our method of finding candidate models, we had dropped about 13 to 15 outliers depending on the candidate model. We are interested in which variables that these outliers deviated so far from the herd that resulted in them being considered high influence. For example, they could be potential powerhouse teams that dwarf the other teams.

One of the flaws would be how the teams were recorded multiple times in the data set as we aggregated data sets from different seasons, so the same teams would be multiple samples within our data set, but just be from different years. This brings up the query of conducting a study on the aggregated data of a team, so that we would be able to predict the success of a team by comparing the change in the variables like expected goals from season to season as a way of determining what a team should focus their efforts on improving in order to maximize the proportion of the games won.

In general, we need to acknowledge that the data collection may not be entirely accurate and could be incomplete. Its usability is not quantifiable and there may be some bias in the logging of the data.

Continuing with this data set, some areas that we would have liked to explore include creating a training and testing model to use the attendance of teams to predict wins, losses, and goals. There are also many different areas we could explore given this data and the results we were able to produce in this study.

Appendix

List of `R` libraries used

- `ggplot2`: visualizing the data
- `ggthemes`: adding extra geoms, scales, and themes for `ggplot2`
- `lmtest`: providing the function for the Breusch-Pagan test
- `faraway`: providing the `vif` function for calculating variance inflation factor
- `knitr`: enabling the knitting of the `.rmd` file as a pdf file, with support for Markdown and LaTeX, along with formatting the output of certain code chunks as a table rather than a console output
- `readr`: reading and loading the `.csv` file

Functions used throughout report

We used the following function when performing the residual diagnostics, with `plotit` set to `FALSE`, as we used `ggplot2` to display the residual plots.

```r
diagnostics = function(model, pcol = "grey", lcol = "dodgerblue",
                       plotit = TRUE, testit = TRUE) {
  if (plotit == TRUE) {
    par(mfrow = c(1, 2))
    plot(x = fitted(model), y = resid(model), col = pcol,
         xlab = "Fitted", ylab = "Residuals", main = "Residual plot")
    abline(h = 0,  col = lcol)

    qqnorm(resid(model), col = pcol,
           main = c("Normal Q-Q Plot of Inputted Model"))
    qqline(resid(model), col = lcol)
  }
  if (testit == TRUE) {
    library(lmtest)
    p_sw <- shapiro.test(resid(model))$p.value
    p_bp <- bptest(model)$p.value[[1]]
    list(p_sw = p_sw, p_bp = p_bp)
  }
}
```

```r
loocv_rmse = function(model) {
  sqrt(mean((resid(model) / (1 - hatvalues(model))) ^ 2))
}
```

```r
rmse = function(model) {
  sqrt(mean(resid(model) ^ 2))
}
```