

sta210 project

Cole Walker, Madison Griffin

loading packages & dataset

```
library(tidyverse)
library(tidymodels)
library(readxl)
library(MASS)
library(leaps)
library(caret)
library(glmnet)
library(Stat2Data)
#library(statnnet)
library(lme4)
library(UpSetR)
library(nlme)
library(sjstats)
set.seed(8)
soccer <- read_excel("AllTimeRankingByClub.xlsx")
```

Introduction and data

Data Cleaning

```
soccer = soccer %>%
  rename(goals_for = `Goals For`,
         goals_against = `Goals Against`,
         goal_diff = `Goal Diff`)

soccer = soccer %>%
  mutate(winspermatch = Win/Played,
```

```

    pointspermatch = Pts/Played,
    goalspermatch = goals_for/Played,
    goalsagainstpermatch = goals_against/Played,
    goalmarginpermatch = goal_diff/Played)

soccer = soccer %>%
  mutate(topfiveleague = ifelse(Country == 'ESP' | Country == 'ENG' | Country == 'GER' | C

```

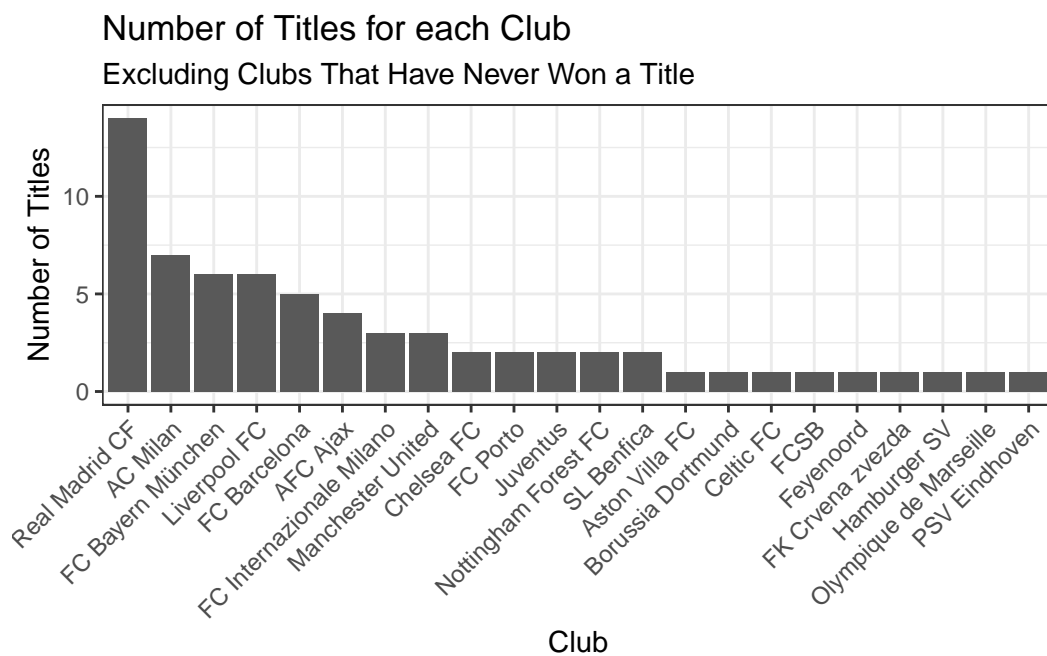
EDA

Plot 1:

```

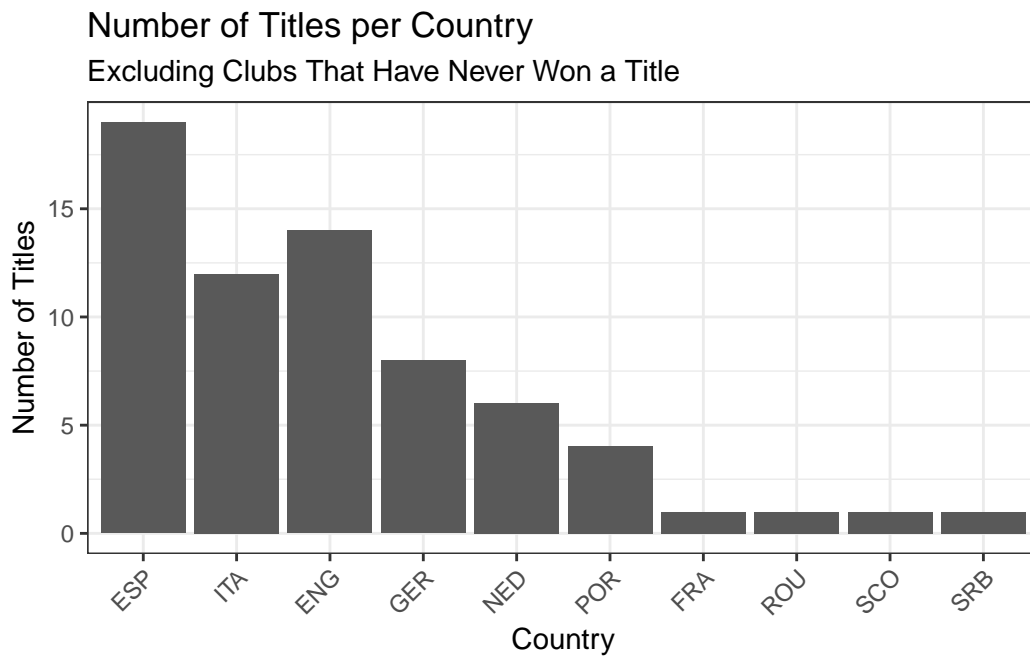
soccer %>%
  filter(Titles > 0) %>%
  ggplot(aes(x = reorder(Club, (-Titles)), y = Titles)) +
  geom_bar(stat = 'identity') +
  labs(x = 'Club', y = 'Number of Titles', title = 'Number of Titles for each Club',
       subtitle = 'Excluding Clubs That Have Never Won a Title') +
  theme_bw() +
  scale_x_discrete(guide = guide_axis(angle = 45))

```



Plot 2:

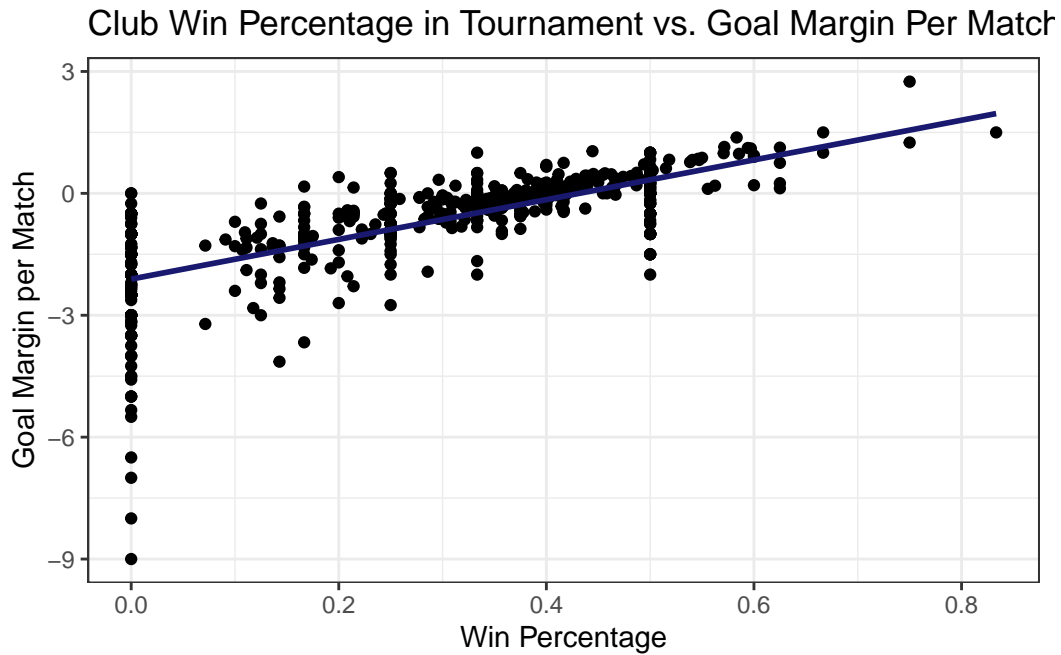
```
soccer %>%  
  filter(Titles > 0) %>%  
  ggplot(aes(x = reorder(Country, (-Titles)), y = Titles)) +  
  geom_bar(stat = 'identity') +  
  labs(x = 'Country', y = 'Number of Titles', title = 'Number of Titles per Country',  
       subtitle = 'Excluding Clubs That Have Never Won a Title') +  
  theme_bw() +  
  scale_x_discrete(guide = guide_axis(angle = 45))
```



Plot 3:

```
ggplot(data = soccer, aes(x = winspermatch, y = goalmarginpermatch)) +  
  geom_point() +  
  geom_smooth(method = 'lm', se = F, color = 'midnightblue') +  
  labs(x = "Win Percentage", y = 'Goal Margin per Match',  
       title = 'Club Win Percentage in Tournament vs. Goal Margin Per Match') +  
  theme_bw()
```

`geom_smooth()` using formula = 'y ~ x'



Methods

we can only do linear or linear mixed effects...expand later

Checking Assumptions

Model 1: Linear Regression

Outcome:

- points per match

Predictors:

- wins per match
- goal margin per match
- top five league

```
linear = lm(pointspermatch ~ winspermatch + goalmarginpermatch + topfiveleague,  
            data = soccer)  
summary(linear)
```

Call:

```
lm(formula = pointspermatch ~ winspermatch + goalmarginpermatch +  
    topfiveleague, data = soccer)
```

Residuals:

Min	1Q	Median	3Q	Max
-0.33632	-0.06813	0.00603	0.06892	0.56598

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	0.434020	0.016755	25.904	<2e-16 ***
winspermatch	1.379068	0.043462	31.731	<2e-16 ***
goalmarginpermatch	0.097704	0.006335	15.423	<2e-16 ***
topfiveleaguetop five	0.025563	0.016412	1.558	0.12

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1354 on 526 degrees of freedom

Multiple R-squared: 0.8845, Adjusted R-squared: 0.8838

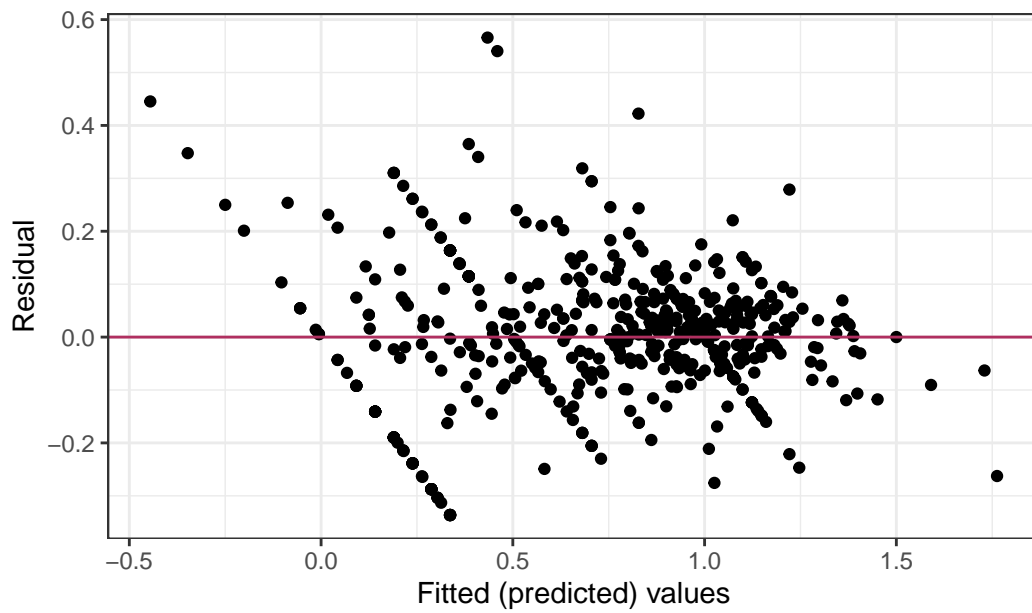
F-statistic: 1343 on 3 and 526 DF, p-value: < 2.2e-16

Conditions for Model 1: Violated linearity, constant variance, and normality

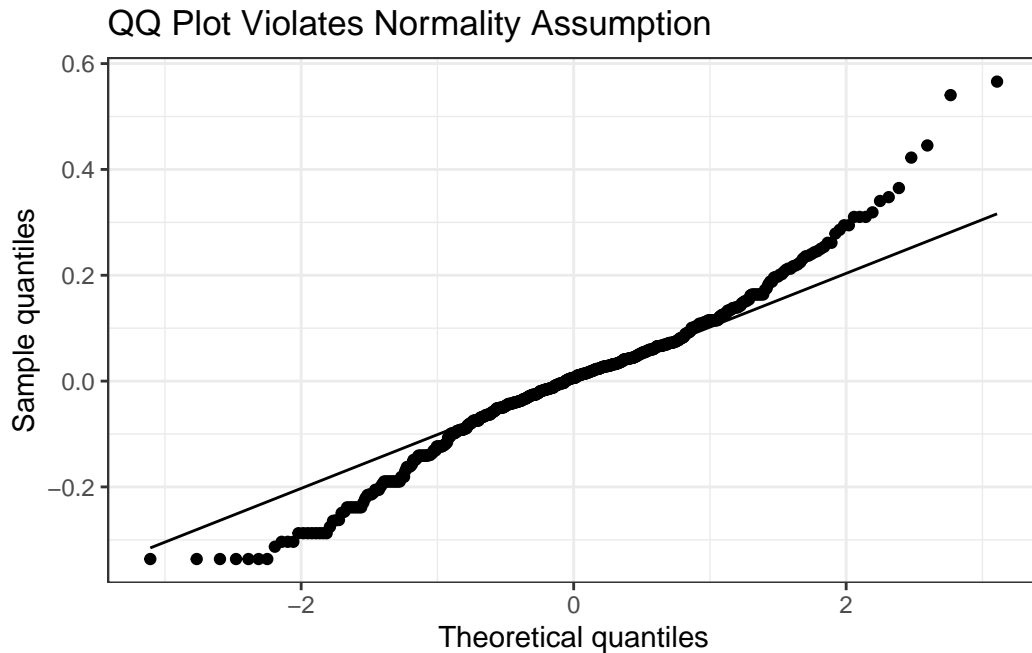
```
linearaug = augment(linear)
```

```
ggplot(linearaug, aes(x = .fitted, y = .resid)) +  
  geom_point() +  
  geom_hline(yintercept = 0, color = 'maroon') +  
  labs(x = "Fitted (predicted) values", y = 'Residual') +  
  ggtitle('Residual Plot Violates Linearity & Constant Variance Assumptions') +  
  theme_bw()
```

Residual Plot Violates Linearity & Constant Variance Assumpt



```
ggplot(linearaug, aes(sample = .resid)) +  
  stat_qq() +  
  stat_qq_line() +  
  theme_bw() +  
  labs(x = 'Theoretical quantiles',  
       y = 'Sample quantiles',  
       title = 'QQ Plot Violates Normality Assumption')
```



Model 2: Linear Mixed Effects Model

Outcome:

- points per match

Predictors:

- wins per match
- goal margin per match
- random intercept for top five league

```
linearmixed = lmer(pointspermatch ~ 1 + winspermatch + goalmarginpermatch +
  (1|topfiveleague), data = soccer)
summary(linearmixed)
```

Linear mixed model fit by REML ['lmerMod']

Formula: `pointspermatch ~ 1 + winspermatch + goalmarginpermatch + (1 | topfiveleague)`

Data: soccer

REML criterion at convergence: -595.8

Scaled residuals:

Min	1Q	Median	3Q	Max
-2.4907	-0.5174	0.0556	0.5114	4.1696

Random effects:

Groups	Name	Variance	Std.Dev.
topfiveleague	(Intercept)	0.0001921	0.01386
Residual		0.0183274	0.13538

Number of obs: 530, groups: topfiveleague, 2

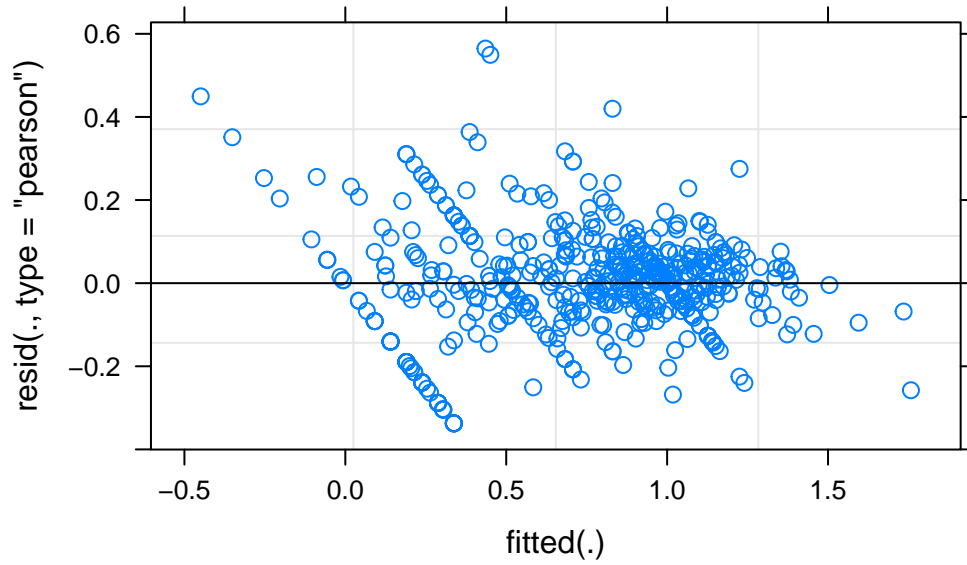
Fixed effects:

	Estimate	Std. Error	t value
(Intercept)	0.443040	0.019785	22.39
winspermatch	1.382252	0.043345	31.89
goalmarginpermatch	0.098334	0.006304	15.60

Correlation of Fixed Effects:

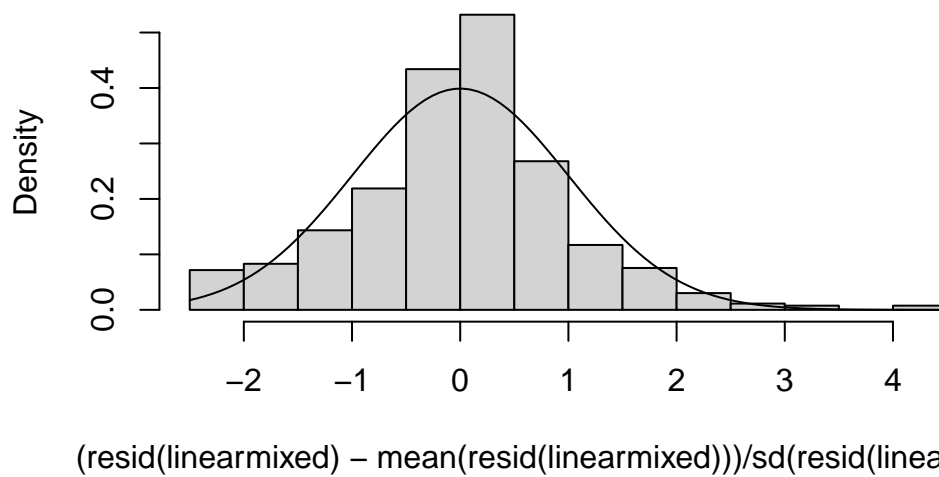
	(Intr)	wnsprm
winspermtch	-0.780	
glmrnprmtc	0.636	-0.690

```
plot(linearmixed)
```

```
hist((resid(linearmixed) - mean(resid(linearmixed))) / sd(resid(linearmixed)), freq = FALSE)
```

Plot of $(\text{resid}(\text{linearmixed}) - \text{mean}(\text{resid}(\text{linearmixed}))) / \text{sd}(\text{resid}(\text{linearmixed}))$



Assumptions are worse for mixed effects...moving forward with linear regression

Variable Selection - LASSO

```
y = soccer$pointspermatch
x = model.matrix(pointspermatch ~ winspermatch + goalspermatch + goalsagainstpermatch +
                 goalmarginpermatch + topfiveleague, data = soccer)
m_lasso_cv = cv.glmnet(x, y, alpha = 1)

best_lambda = m_lasso_cv$lambda.min
best_lambda
```

```
[1] 0.003448411
```

```
m_best = glmnet(x, y, alpha = 1, lambda = best_lambda)
m_best$beta
```

```
6 x 1 sparse Matrix of class "dgCMatrix"
s0
```

```
(Intercept)      .
winspermatch      1.37038572
goalspermatch     .
goalsagainstpermatch .
goalmarginpermatch 0.09659559
topfiveleague     0.01905888
```

```
bestlasso = lm(pointspermatch ~ winspermatch + goalmarginpermatch + topfiveleague,
               data = soccer)
```

Variable Selection - Stepwise Selection

```
m_none = lm(pointspermatch ~ 1, data = soccer)
m_all = lm(pointspermatch ~ winspermatch + goalspermatch + goalsagainstpermatch +
           goalmarginpermatch + topfiveleague, data = soccer)
```

Forward Selection

```

stepAIC(m_none,
        scope = list(lower = m_none, upper = m_all),
        data = soccer, direction = 'forward')

```

Start: AIC=-977.7
pointspermatch ~ 1

	Df	Sum of Sq	RSS	AIC
+ winspermatch	1	69.170	14.292	-1911.0
+ goalmarginpermatch	1	54.871	28.590	-1543.5
+ goalspermatch	1	41.898	41.564	-1345.2
+ goalsagainstpermatch	1	33.999	49.463	-1253.0
+ topfiveleague	1	9.140	74.322	-1037.2
<none>			83.462	-977.7

Step: AIC=-1910.99
pointspermatch ~ winspermatch

	Df	Sum of Sq	RSS	AIC
+ goalmarginpermatch	1	4.6072	9.6847	-2115.2
+ goalsagainstpermatch	1	3.7403	10.5516	-2069.8
+ goalspermatch	1	0.4669	13.8250	-1926.6
+ topfiveleague	1	0.2923	13.9996	-1919.9
<none>			14.2919	-1911.0

Step: AIC=-2115.24
pointspermatch ~ winspermatch + goalmarginpermatch

	Df	Sum of Sq	RSS	AIC
+ topfiveleague	1	0.044462	9.6402	-2115.7
<none>			9.6847	-2115.2
+ goalspermatch	1	0.000047	9.6846	-2113.2
+ goalsagainstpermatch	1	0.000047	9.6846	-2113.2

Step: AIC=-2115.68
pointspermatch ~ winspermatch + goalmarginpermatch + topfiveleague

	Df	Sum of Sq	RSS	AIC
<none>			9.6402	-2115.7
+ goalspermatch	1	0.00015982	9.6400	-2113.7
+ goalsagainstpermatch	1	0.00015982	9.6400	-2113.7

Call:

```
lm(formula = pointspermatch ~ winspermatch + goalmarginpermatch +  
    topfiveleague, data = soccer)
```

Coefficients:

	(Intercept)	winspermatch	goalmarginpermatch
	0.43402	1.37907	0.09770
topfiveleague	0.02556		

```
bestforward = lm(pointspermatch ~ winspermatch + goalmarginpermatch +  
    topfiveleague, data = soccer)
```

Backward Selection

```
stepAIC(m_all,  
        scope = list(lower = m_none, upper = m_all),  
        data = soccer, direction = 'backward')
```

Start: AIC=-2113.69

```
pointspermatch ~ winspermatch + goalspermatch + goalsagainstpermatch +  
    goalmarginpermatch + topfiveleague
```

Step: AIC=-2113.69

```
pointspermatch ~ winspermatch + goalspermatch + goalsagainstpermatch +  
    topfiveleague
```

	Df	Sum of Sq	RSS	AIC
<none>			9.6400	-2113.7
- topfiveleague	1	0.0446	9.6846	-2113.2
- goalspermatch	1	0.7904	10.4305	-2073.9
- goalsagainstpermatch	1	3.9802	13.6203	-1932.5
- winspermatch	1	14.4821	24.1221	-1629.6

Call:

```
lm(formula = pointspermatch ~ winspermatch + goalspermatch +  
    goalsagainstpermatch + topfiveleague, data = soccer)
```

Coefficients:

(Intercept)	winspermatch	goalspermatch
0.43510	1.38121	0.09647
goalsagainstpermatch	topfiveleague	top five
-0.09789	0.02570	

```
bestbackward = lm(pointspermatch ~ winspermatch + goalspermatch + goalsagainstpermatch +  
topfiveleague, data = soccer)
```

Both Selection

```
stepAIC(m_none,  
scope = list(lower = m_none, upper = m_all),  
data = soccer, direction = 'both')
```

Start: AIC=-977.7

pointspermatch ~ 1

	Df	Sum of Sq	RSS	AIC
+ winspermatch	1	69.170	14.292	-1911.0
+ goalmarginpermatch	1	54.871	28.590	-1543.5
+ goalspermatch	1	41.898	41.564	-1345.2
+ goalsagainstpermatch	1	33.999	49.463	-1253.0
+ topfiveleague	1	9.140	74.322	-1037.2
<none>			83.462	-977.7

Step: AIC=-1910.99

pointspermatch ~ winspermatch

	Df	Sum of Sq	RSS	AIC
+ goalmarginpermatch	1	4.607	9.685	-2115.2
+ goalsagainstpermatch	1	3.740	10.552	-2069.8
+ goalspermatch	1	0.467	13.825	-1926.6
+ topfiveleague	1	0.292	14.000	-1919.9
<none>			14.292	-1911.0
- winspermatch	1	69.170	83.462	-977.7

Step: AIC=-2115.24

pointspermatch ~ winspermatch + goalmarginpermatch

Df	Sum of Sq	RSS	AIC
----	-----------	-----	-----

```

+ topfiveleague      1      0.0445  9.6402 -2115.7
<none>                                9.6847 -2115.2
+ goalspermatch      1      0.0000  9.6846 -2113.2
+ goalsagainstpermatch 1      0.0000  9.6846 -2113.2
- goalmarginpermatch 1      4.6072 14.2919 -1911.0
- winspermatch       1     18.9058 28.5905 -1543.5

```

Step: AIC=-2115.68

pointspermatch ~ winspermatch + goalmarginpermatch + topfiveleague

	Df	Sum of Sq	RSS	AIC
<none>			9.6402	-2115.7
- topfiveleague	1	0.0445	9.6847	-2115.2
+ goalspermatch	1	0.0002	9.6400	-2113.7
+ goalsagainstpermatch	1	0.0002	9.6400	-2113.7
- goalmarginpermatch	1	4.3594	13.9996	-1919.9
- winspermatch	1	18.4525	28.0927	-1550.8

Call:

```
lm(formula = pointspermatch ~ winspermatch + goalmarginpermatch +
    topfiveleague, data = soccer)
```

Coefficients:

(Intercept)	winspermatch	goalmarginpermatch
0.43402	1.37907	0.09770
topfiveleague	top five	
0.02556		

```
bestboth = lm(pointspermatch ~ winspermatch + goalmarginpermatch + topfiveleague, data = s
```

Variable Selection - All Subset

```

soccer_allsub = regsubsets(pointspermatch ~ winspermatch + goalspermatch +
                           goalsagainstpermatch + goalmarginpermatch + topfiveleague,
                           data = soccer,
                           nbest = 1, nvmax = 5)

```

Warning in leaps.setup(x, y, wt = wt, nbest = nbest, nvmax = nvmax, force.in = force.in, : 1 linear dependencies found

Reordering variables and trying again:

```
soccer_allsub
```

Subset selection object

```
Call: regsubsets.formula(pointspermatch ~ winspermatch + goalspermatch +
  goalsagainstpermatch + goalmarginpermatch + topfiveleague,
  data = soccer, nbest = 1, nvmax = 5)
```

5 Variables (and intercept)

	Forced in	Forced out
winspermatch	FALSE	FALSE
goalspermatch	FALSE	FALSE
goalsagainstpermatch	FALSE	FALSE
topfiveleaguetop five	FALSE	FALSE
goalmarginpermatch	FALSE	FALSE

1 subsets of each size up to 4

Selection Algorithm: exhaustive

```
summary(soccer_allsub)$rsq
```

```
[1] 0.8287614 0.8839632 0.8844959 0.8844979
```

```
summary(soccer_allsub)$which
```

	(Intercept)	winspermatch	goalspermatch	goalsagainstpermatch
1	TRUE	TRUE	FALSE	FALSE
2	TRUE	TRUE	FALSE	FALSE
3	TRUE	TRUE	FALSE	FALSE
4	TRUE	TRUE	TRUE	TRUE

	goalmarginpermatch	topfiveleaguetop five
1	FALSE	FALSE
2	TRUE	FALSE
3	TRUE	TRUE
4	FALSE	TRUE

```
bestallsubset = lm(pointspermatch ~ winspermatch, data = soccer)
```

Comparing RMSE after variable selection

RMSE All Subset: 0.1642128

RMSE Best Backward: 0.1348656 - LOWEST

RMSE Best Both, Forward, Lasso: 0.1348667

(they had the same predictors in it)

CONCLUSION:

- best backward has lowest rmse so better
- predictors: wins per match, top five league, goals per match, goals against per match

```
rmse(bestallsubset)
```

```
[1] 0.1642128
```

```
rmse(bestbackward)
```

```
[1] 0.1348656
```

```
rmse(bestboth)
```

```
[1] 0.1348667
```

```
rmse(bestforward)
```

```
[1] 0.1348667
```

```
rmse(bestlasso)
```

```
[1] 0.1348667
```

##possibly delete – Comparing Models: RMSE

RMSE Model 1: 0.1348656 RMSE Model 2: 0.1348656 RMSE Model 3: 0.1348667 RMSE Model 4: 0.1349185


```
rmse(model1)
```

```
object 'model1' not found
```

```
[1] NA
```

```
rmse(model2)
```

```
object 'model2' not found
```

```
[1] NA
```

```
rmse(model3)
```

```
object 'model3' not found
```

```
[1] NA
```

```
rmse(model4)
```

```
object 'model4' not found
```

```
[1] NA
```

Results

Conclusion