Final Project

Arnav, Casey, Harry, Maxx, Mohit

Introduction

Loading and Cleaning the Data

First we load the data we are using and remove NA's. Then we also set aside a portion of the data for later training/testing data.

```
fifa <- read.csv('fifa_players.csv')
fifa <- na.omit(fifa)

# Generate training/test data:
set.seed(123)


# 80% Training 20% test data:
sample <- sample(c(TRUE, FALSE), nrow(fifa), replace=TRUE, prob=c(0.8,0.2))

training_data  <- fifa[sample, ]
test_data   <- fifa[!sample, ]
```

The total amount of observations in our FIFA dataset is: 789, and the total variables/columns is: 51

Removing Outliers:

```
ggplot(data = fifa, aes(x=overall_rating))+
  geom_histogram(binwidth = 1)
```

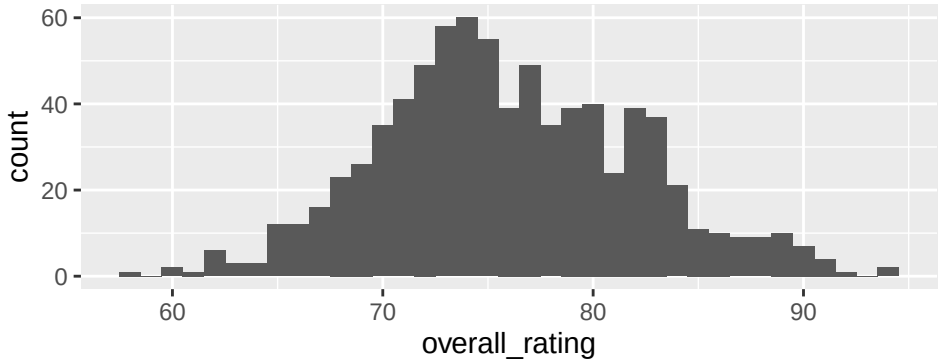


Fig. 1

As we can see, our main response variable, overall_rating, is approximately normally distributed and has no outliers.

```
ggplot(data = fifa, aes(x=height_cm, y=weight_kgs))+
  geom_point()
```

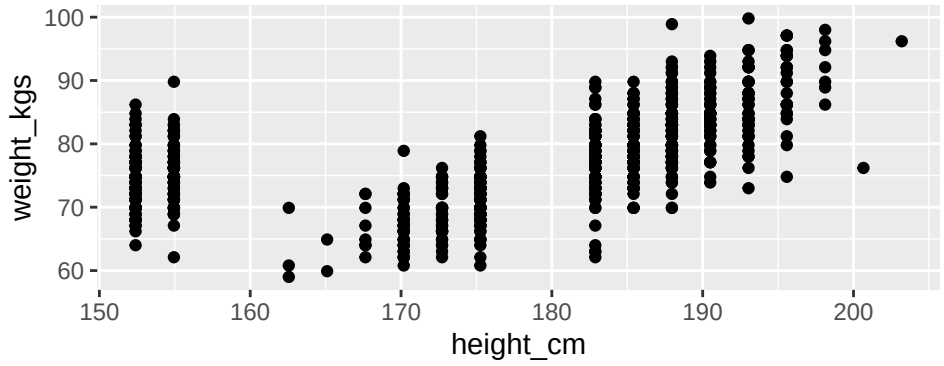
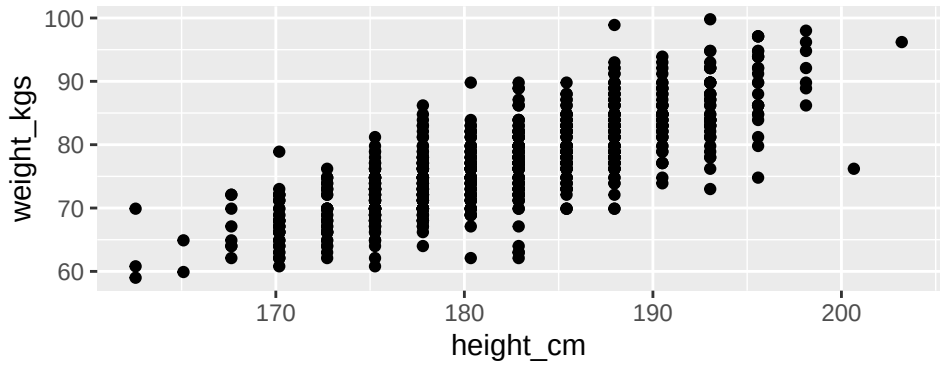


Fig. 2

We know from background information that height and weight should be positively correlated, and we expect to see that in this visualization. We mostly see this, but something is clearly off. We googled the heights to convert centimeters to inches and found that the outliers occur at heights 5'0" and 5'1". We also noticed that the data has an empty patch in the middle of the x axis, indicating that some heights are not included. We hypothesized that there was an error by the creators of the dataset in their data scraping process, in which 5'10" players were recorded as 5'0" and 5'11" was recorded as 5'1". The next code chunk shows us testing this hypothesis and seeing that 5'10" and 5'11" were indeed the missing heights, so we decided to rewrite the data set so that all 5'0" entries were corrected to 5'10" and all 5'1" entries were corrected to 5'11". However, the issue with this is that there could be a few players who really are 5'0" and 5'1" in the dataset and they need to be recorded with their accurate height. To solve this problem, we looked through the original data source (https://sofifa.com/players?col=hi&sort=asc&r=190043&set=true) and searched for 5'0" and 5'1" players. There were only 3, so we manually filtered by name to get their true height reflected in the dataset. After the following chunk the height column will have accurate data.

```
#wondering if 5'11 was scraped at 5'1 and 5'10 was scraped as 5'0
fifa_test <- fifa %>%
  mutate(height_cm = ifelse(height_cm == 152.4, 177.8, height_cm) ) %>%
  mutate(height_cm = ifelse(height_cm == 154.94, 180.34, height_cm) )  %>%
  mutate(height_cm = ifelse(full_name == "Kazuki Yamaguchi" |name == "H. Nakagawa" | full_name == "Cris

ggplot(data = fifa_test, aes(x=height_cm, y=weight_kgs))+
  geom_point()
```




```
fifa <- fifa_test
```

Fig. 3

Response to Teacher Question 3:

"With such a large sample size, you will find a lot of significant predictors. Discuss why."

Our sample size is over 17000 (17898). This can result in many of our predictors being statistically significant because a large sample size leads to an even more precise discovery of relationships between variables, reducing variance. Thus, with a very large sample size, even if the predictor's impact is smaller, it could still be considered statistically significant. If we think in terms of confidence intervals, they become extremely narrow because the margin of error decreases, and this causes more predictor values to be outside of the interval and thus considered significant. While a predictor could be statistically significant, it may not be an impactful predictor of the output in reality.

One way we can tackle this problem is by using corrections such as Bonferroni, Holm, or Benjamini Hochberg to lower false positive rates. Also, we used training and testing split in predictions, which can reduce the number of samples during training and prevent overfitting to just the training data (and performing poorly on new data), and could use AIC to choose models as well.

Research Question 1:

Which factors are the biggest contributors to overall FIFA ratings?

Here is how we set up the model

```
new_fifa <- fifa[c("overall_rating", "weak_foot.1.5.", "skill_moves.1.5.", "sprint_speed", "dribbling",
                   "strength", "stamina", "value_euro", "wage_euro")]
basic <- lm(overall_rating ~ weak_foot.1.5. + skill_moves.1.5. + sprint_speed + dribbling + strength +
              stamina + log(value_euro) + log(wage_euro), data=new_fifa)
summary <- summary(basic)
summary
```

```
##
## Call:
## lm(formula = overall_rating ~ weak_foot.1.5. + skill_moves.1.5. +
##     sprint_speed + dribbling + strength + stamina + log(value_euro) +
##     log(wage_euro), data = new_fifa)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.8913 -1.1483 -0.2056  0.8665 10.0303
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)        5.7201386  0.8789562   6.508 1.36e-10 ***
## weak_foot.1.5.     0.0574574  0.0849759   0.676   0.4991
## skill_moves.1.5. -0.1743704  0.1386981  -1.257   0.2091
## sprint_speed     -0.0135744  0.0067985  -1.997   0.0462 *
## dribbling        -0.0178645  0.0084239  -2.121   0.0343 *
## strength          0.0028848  0.0061699   0.468   0.6402
## stamina           0.0008879  0.0067918   0.131   0.8960
## log(value_euro)   4.2350991  0.0745139  56.836  < 2e-16 ***
## log(wage_euro)    0.6026149  0.0618216   9.748  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 1.657 on 780 degrees of freedom
## Multiple R-squared:  0.9267, Adjusted R-squared:  0.926
## F-statistic:  1233 on 8 and 780 DF,  p-value: < 2.2e-16
```

```
cor_matrix <- cor(new_fifa[, sapply(new_fifa, is.numeric)])
heatmap(cor_matrix, symm = TRUE, col = colorRampPalette(c("blue", "white", "red"))(100))
```

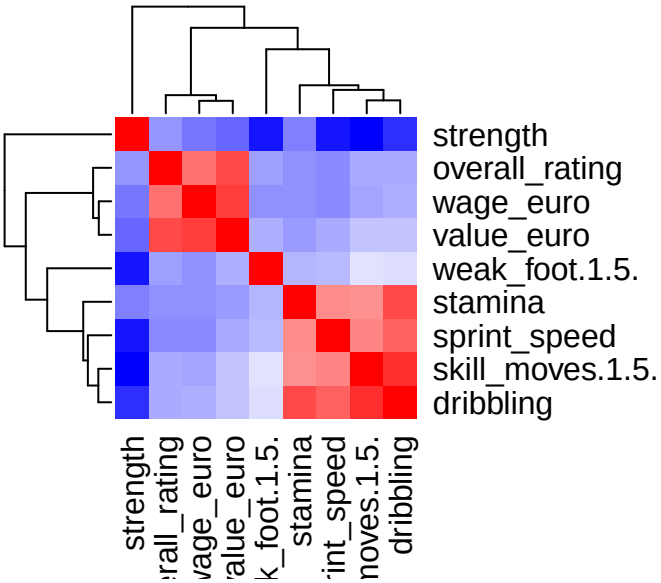


Fig. 4

So we started by looking at the relationship between each of our chosen predictors and the output variable of overall_rating

```
par(mfrow = c(1, 2))
plot(new_fifa$value_euro, new_fifa$overall_rating)
plot(new_fifa$wage_euro, new_fifa$overall_rating)
```

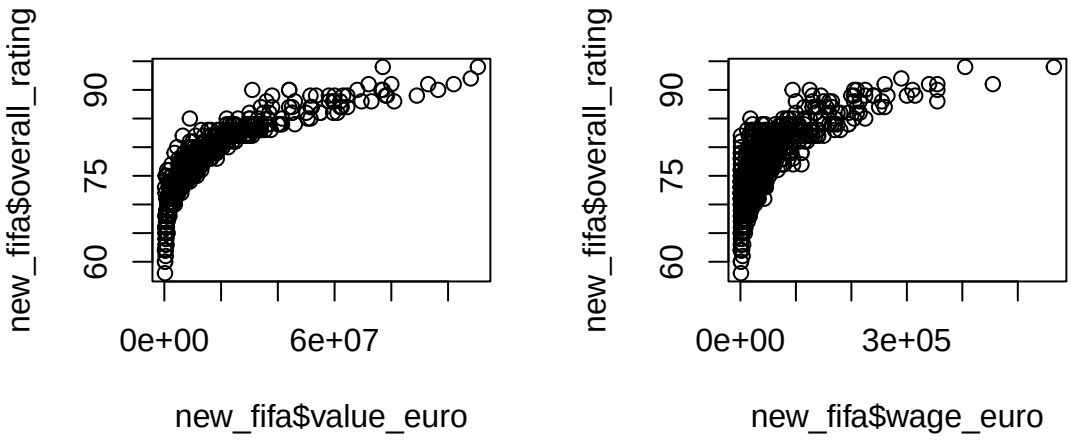


Fig. 5

We saw roughly linear relationships for most cases. However, for value_euro and wage_euro, our scatter plots look very similar to a logarithmic curve. Thus, we need to apply a transformation to our data and instead use ln(value_euro) and ln(wage_euro)

```r
par(mfrow = c(1, 2))
plot(log(new_fifa$value_euro), new_fifa$overall_rating)
plot(log(new_fifa$wage_euro), new_fifa$overall_rating)
```

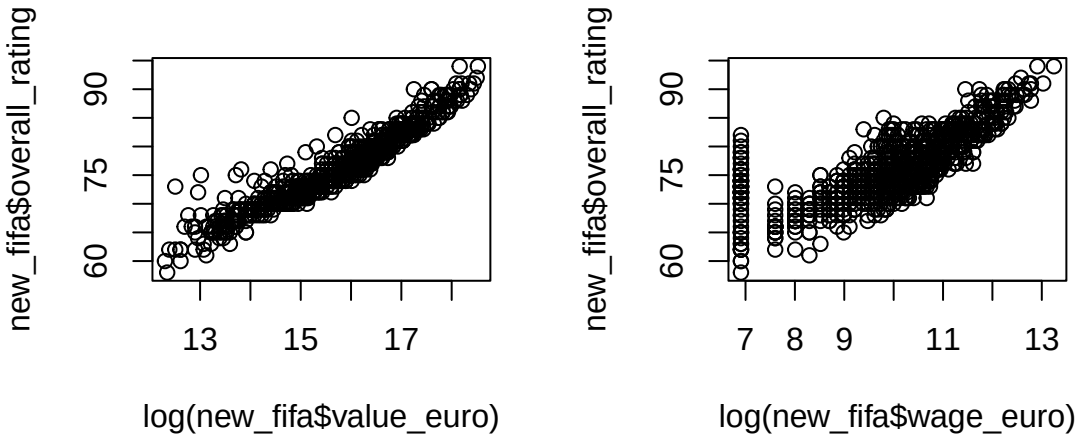


Fig. 6

This shows a much better linear relationship.

Now let's look at the R value or correlation coefficient of each variable to overall_rating:

```r
correlations <- data.frame(
  Variable = c("Weak Foot", "Skill Moves", "Sprint Speed", "Dribbling", "Strength", "Stamina", "Log(Valu
  Correlation = c(
    cor(new_fifa$weak_foot.1.5., new_fifa$overall_rating),
    cor(new_fifa$skill_moves.1.5., new_fifa$overall_rating),
    cor(new_fifa$sprint_speed, new_fifa$overall_rating),
    cor(new_fifa$dribbling, new_fifa$overall_rating),
    cor(new_fifa$strength, new_fifa$overall_rating),
    cor(new_fifa$stamina, new_fifa$overall_rating),
    cor(log(new_fifa$value_euro), new_fifa$overall_rating),
    cor(log(new_fifa$wage_euro), new_fifa$overall_rating)
  )
)
print(correlations)
```

```
##             Variable Correlation
## 1          Weak Foot   0.1951030
## 2        Skill Moves   0.2204819
## 3       Sprint Speed   0.1478155
## 4          Dribbling   0.2179792
## 5           Strength   0.1769668
## 6            Stamina   0.1607639
## 7 Log(Value in Euro)   0.9517758
## 8  Log(Wage in Euro)   0.7784407
```

Table 1

We can see we found particularly high values for log(value_euro), log(wage_euro).

Now, let us form our linear regression model and utilize an F test to see our results.

```
model1 <- lm(overall_rating ~ weak_foot.1.5. + skill_moves.1.5. + sprint_speed + dribbling + strength +


summary <- summary(model1)
summary
```

```
##
## Call:
## lm(formula = overall_rating ~ weak_foot.1.5. + skill_moves.1.5. +
##     sprint_speed + dribbling + strength + stamina + log(value_euro) +
##     log(wage_euro), data = new_fifa)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.8913 -1.1483 -0.2056  0.8665 10.0303
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)       5.7201386  0.8789562   6.508 1.36e-10 ***
## weak_foot.1.5.    0.0574574  0.0849759   0.676   0.4991
## skill_moves.1.5. -0.1743704  0.1386981  -1.257   0.2091
## sprint_speed     -0.0135744  0.0067985  -1.997   0.0462 *
## dribbling        -0.0178645  0.0084239  -2.121   0.0343 *
## strength          0.0028848  0.0061699   0.468   0.6402
## stamina           0.0008879  0.0067918   0.131   0.8960
## log(value_euro)   4.2350991  0.0745139  56.836  < 2e-16 ***
## log(wage_euro)    0.6026149  0.0618216   9.748  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.657 on 780 degrees of freedom
## Multiple R-squared:  0.9267, Adjusted R-squared:  0.926
## F-statistic:  1233 on 8 and 780 DF,  p-value: < 2.2e-16
```

So, we found p-values < 0.05 (our alpha/significance level) for the following predictors: weak_foot, sprint_speed, dribbling, strength, stamina, log(value_euro), log(wage_euro) meaning that these predictors were found to have a statistically significant impact on overall_rating. Because skill_moves had a p-value > 0.05, it means it is not statistically significant based on our model, so now we will create a new model without this predictor and compare results. This likely means the other variables also account for the impact made by skill moves. Also, in this model we had a very high Adjusted R^2 of 0.906 which is much higher than our base model's R^2 of 0.5495 and means our model does a great job in explaining the variance in overall_rating.

```
model2 <- lm(overall_rating ~ weak_foot.1.5. + sprint_speed + dribbling + strength + stamina + log(valu

summary2 <- summary(model2)
summary2
```

```
##
## Call:
## lm(formula = overall_rating ~ weak_foot.1.5. + sprint_speed +
##     dribbling + strength + stamina + log(value_euro) + log(wage_euro),
##     data = new_fifa)
##
## Residuals:
```

```
##     Min      1Q  Median      3Q     Max
## -3.9281 -1.1439 -0.1818  0.8657 10.0378
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)      5.662417   0.878082   6.449 1.98e-10 ***
## weak_foot.1.5.   0.052768   0.084926   0.621   0.5346
## sprint_speed    -0.013678   0.006801  -2.011   0.0446 *
## dribbling       -0.025300   0.006000  -4.216 2.77e-05 ***
## strength         0.004562   0.006026   0.757   0.4493
## stamina          0.002088   0.006727   0.310   0.7563
## log(value_euro)  4.224651   0.074077  57.031  < 2e-16 ***
## log(wage_euro)   0.605831   0.061792   9.804  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.658 on 781 degrees of freedom
## Multiple R-squared:  0.9266, Adjusted R-squared:  0.9259
## F-statistic:  1408 on 7 and 781 DF,  p-value: < 2.2e-16
```

Here we can see we only have significant predictors and increased accuracy in our model.

Running an ANOVA to compare our two models:

```
anova(model1, model2)
```

```
## Analysis of Variance Table
##
## Model 1: overall_rating ~ weak_foot.1.5. + skill_moves.1.5. + sprint_speed +
##     dribbling + strength + stamina + log(value_euro) + log(wage_euro)
## Model 2: overall_rating ~ weak_foot.1.5. + sprint_speed + dribbling +
##     strength + stamina + log(value_euro) + log(wage_euro)
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1    780 2142.7
## 2    781 2147.1 -1   -4.3419 1.5805 0.2091
```

As we can see from the ANOVA, we had a p-value of 0.2091. Remember, in the ANOVA test, our null hypothesis is that there is no difference between our models. Our p-value being $>$ alpha 0.05 proves that we fail to reject the null and do not have significantly significant evidence there is a difference between the two models. We can interpret this as meaning that our smaller model is as effective as the larger one and is a better choice for us to use since it is more condensed.

Research Question 2:

Do categorical ratings such as weak foot and skill moves affect overall FIFA rating?

The columns weak foot and skill moves represent those respective ratings in FIFA. The ratings are a number 1-5 out of 5 stars. So this is a discrete numeric variable. First, we will do a visuaization of the data to get an idea of what our data suggests. Because we only have 5 categories for rating, we can do a boxplot of overall rating for each category.

```
par(mfrow=c(1,2))
boxplot(fifa$overall_rating ~ fifa$weak_foot.1.5.,
        col='orange',
        main='Overall Rating by Weak Foot',
```

```
        xlab='Weak Foot(1-5)',
        ylab='FIFA Ovr. Rating')

boxplot(fifa$overall_rating ~ fifa$skill_moves.1.5.,
        col='steelblue',
        main='Overall Rating by Skill Move',
        xlab='Skill Moves(1-5)',
        ylab='FIFA Ovr. Rating')
```

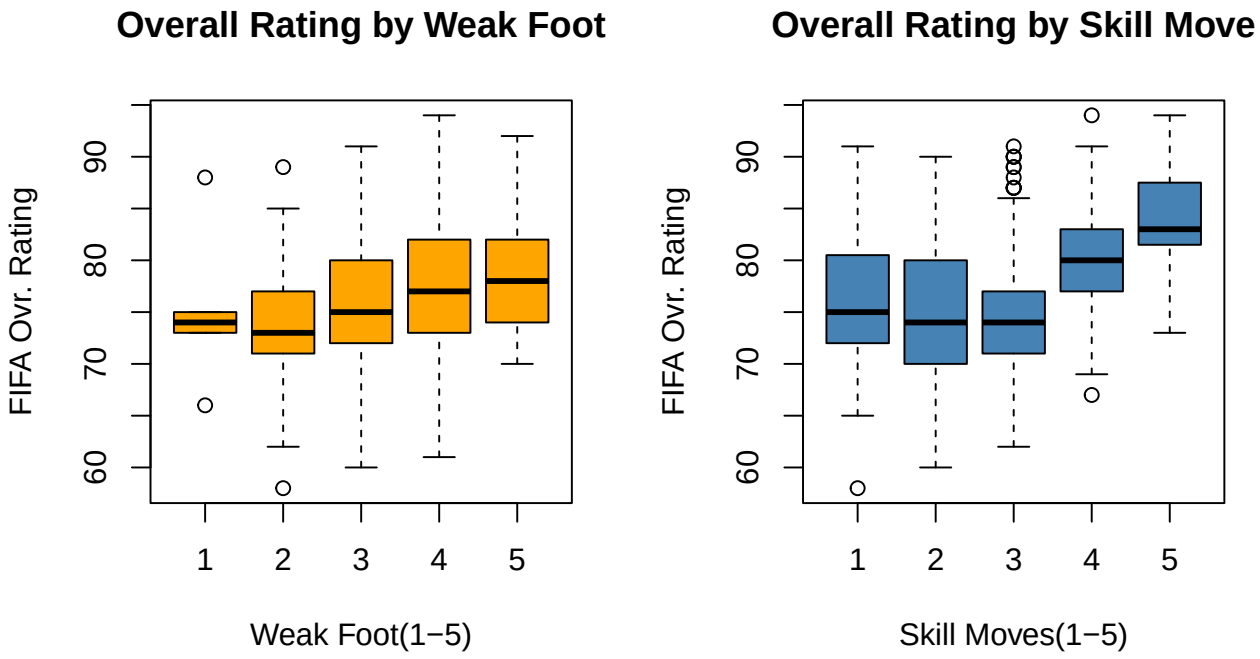


Fig. 7

Both these plots indicate that there is some correlation between these discrete predictors. For the most part, the quartiles and medians gradually increase as we move up by 1 star, and we can see a noticeable difference between 1 star and 5 stars, particularly in skill moves. However, the boxplots are labeling many points as outliers on almost every boxplot. We already examined the overall rating column for outliers, so we know these points belong in our data. It makes me wonder if this is a case of correlation not causation, because these points seem unaffected by the respective discrete predictor. Additionally, it could be that these points have some other factor that is causing these discrete ratings to not matter.

We will run a linear regression on just these two predictors to expand on these comments:

```
discrete_lm <- lm(data=fifa, overall_rating ~ weak_foot.1.5. + skill_moves.1.5. )
summary(discrete_lm)
```

```
##
## Call:
## lm(formula = overall_rating ~ weak_foot.1.5. + skill_moves.1.5.,
##     data = fifa)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -15.1701  -3.9824  -0.7947   3.9191  17.1161
##
## Coefficients:
```

```
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)        69.5177     0.9553  72.772  < 2e-16 ***
## weak_foot.1.5.      1.0892     0.2964   3.675 0.000254 ***
## skill_moves.1.5.    1.0985     0.2322   4.730 2.66e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.899 on 786 degrees of freedom
## Multiple R-squared:  0.06469,    Adjusted R-squared:  0.06231
## F-statistic: 27.18 on 2 and 786 DF,  p-value: 3.856e-12
```

This confirms what we found looking at the boxplots. This linear model found that both predictors are definitely statistically signficiant, as evidenced by the p-values. Based on the estimate for $\hat{\beta}_i$, it is indicated that skill moves has more correlation with overall rating, which is consistent with what we saw in the plots. Both are positively correlated. The model states that, if skill moves rating is held constant, for each star increase of weak foot rating, we can expect overall rating to increase by 0.8555. Also, if weak foot rating is held constant, for each star increase of skill moves rating, we can expect overall rating to increase by 3.5636, which is quite a significant increase.

However, we know just including these two variables is not going to be our best model. A more practical question is does including these variables in our base model making it significantly better vs without them. To test this, we will use an f-test using the anova test function.

```
basic <- lm(overall_rating ~ weak_foot.1.5. + skill_moves.1.5. + sprint_speed + dribbling + strength + s
nondiscrete <- lm(overall_rating ~ sprint_speed + dribbling + strength + stamina + value_euro + wage_eu
anova(nondiscrete, basic)
```

```
## Analysis of Variance Table
##
## Model 1: overall_rating ~ sprint_speed + dribbling + strength + stamina +
##     value_euro + wage_euro
## Model 2: overall_rating ~ weak_foot.1.5. + skill_moves.1.5. + sprint_speed +
##     dribbling + strength + stamina + value_euro + wage_euro
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1    782 8646.5
## 2    780 8608.2  2    38.335 1.7368 0.1768
```

```
###
# This basic is meant to be same as original basic? If so it is missing log transform or euro variables
###
```

The f-test indicated the base model was significantly better than the model removing discrete predictors, with the pvalue approximately 0. This indicates beyond a reasonable doubt that we should include these predictors in our model, they can be used to predict overall rating.

I'm still curious about the outliers we saw in the boxplots. I wonder if there are some positions for which these predictors don't matter.

Let's look at the same plot for only goalkeepers:

```
gk <- fifa %>% filter(positions == "GK")
par(mfrow=c(1,2))
boxplot(gk$overall_rating ~ gk$weak_foot.1.5.,
        col='orange',
        main='GK Overall Rating by Weak Foot',
        xlab='Weak Foot(1-5)',
        ylab='FIFA Ovr. Rating')
```

```
boxplot(gk$overall_rating ~ gk$skill_moves.1.5.,
        col='steelblue',
        main='GK Overall Rating by Skill Move',
        xlab='Skill Moves(1-5)',
        ylab='FIFA Ovr. Rating')
```

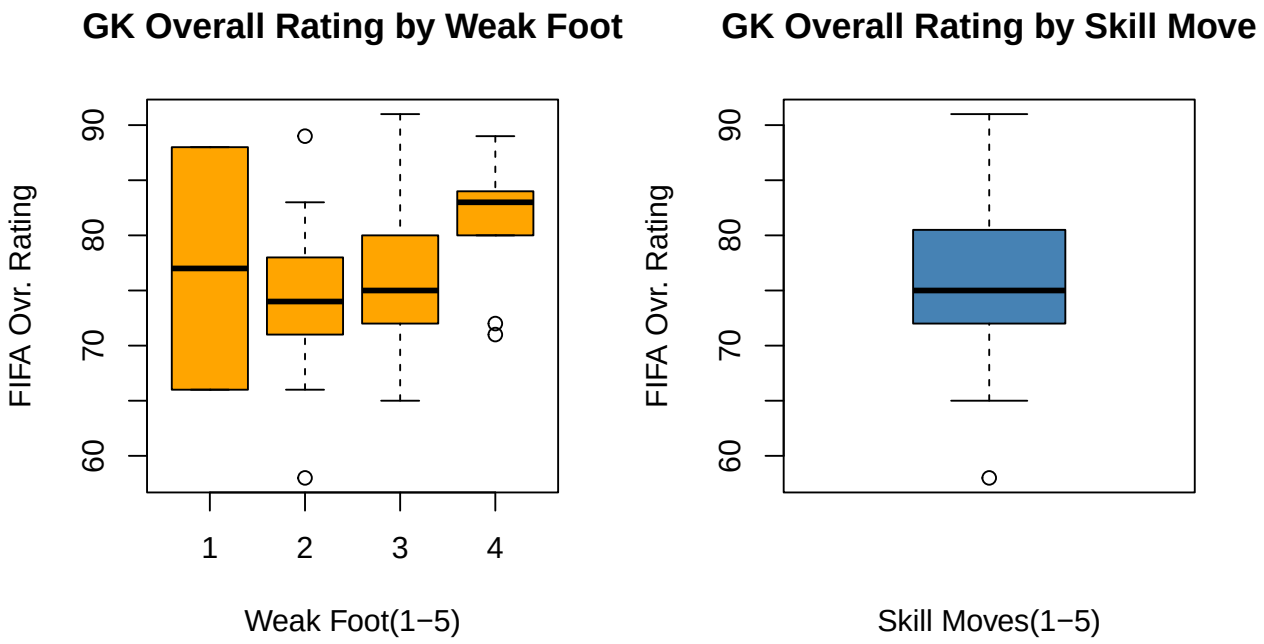


Fig. 8

It appears that the correlation for weak foot is weaker than for all positions. More significantly though, we found that every goalkeeper only has one star skill moves. This means that skill moves rating will have no predictive value for goalkeepers, and this is also effecting the signficance of this variable on our base model.

Let's also explore this for centerbacks*, another position for which on the ball skills are less important.

```
cb <- fifa %>% filter(positions == "CB")
par(mfrow=c(1,2))
boxplot(cb$overall_rating ~ cb$weak_foot.1.5.,
        col='orange',
        main='CB Overall Rating by Weak Foot',
        xlab='Weak Foot(1-5)',
        ylab='FIFA Ovr. Rating')

boxplot(cb$overall_rating ~ cb$skill_moves.1.5.,
        col='steelblue',
        main='CB Overall Rating by Skill Move',
        xlab='Skill Moves(1-5)',
        ylab='FIFA Ovr. Rating')
```

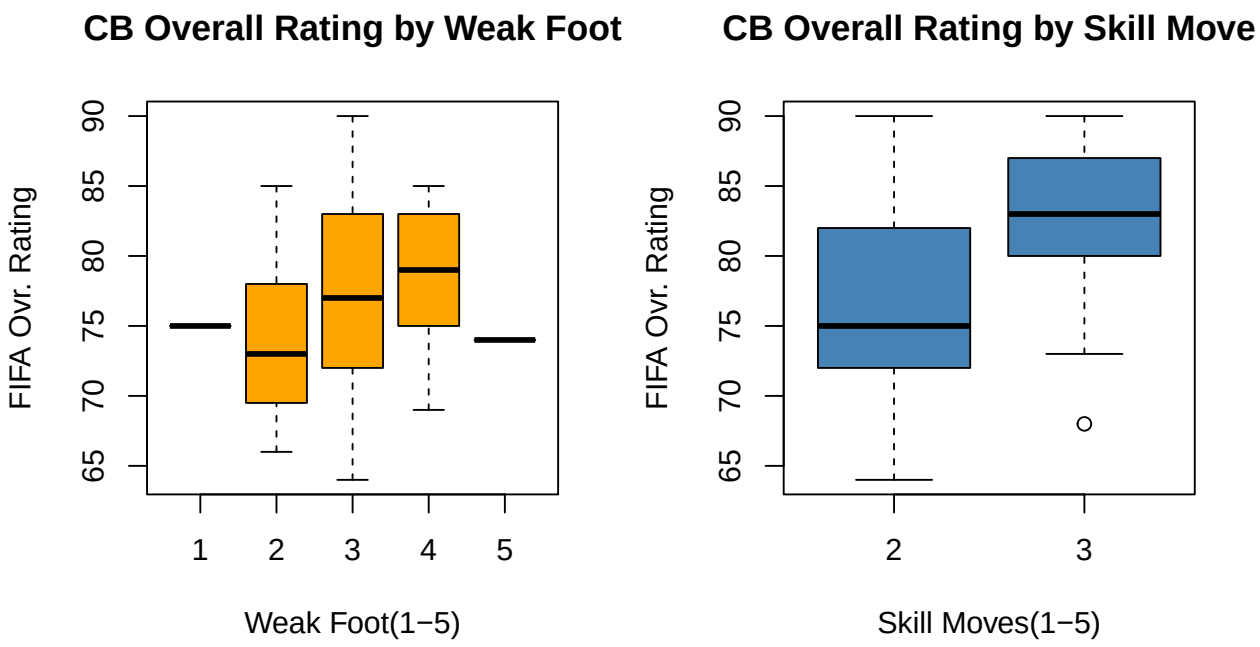


Fig. 9

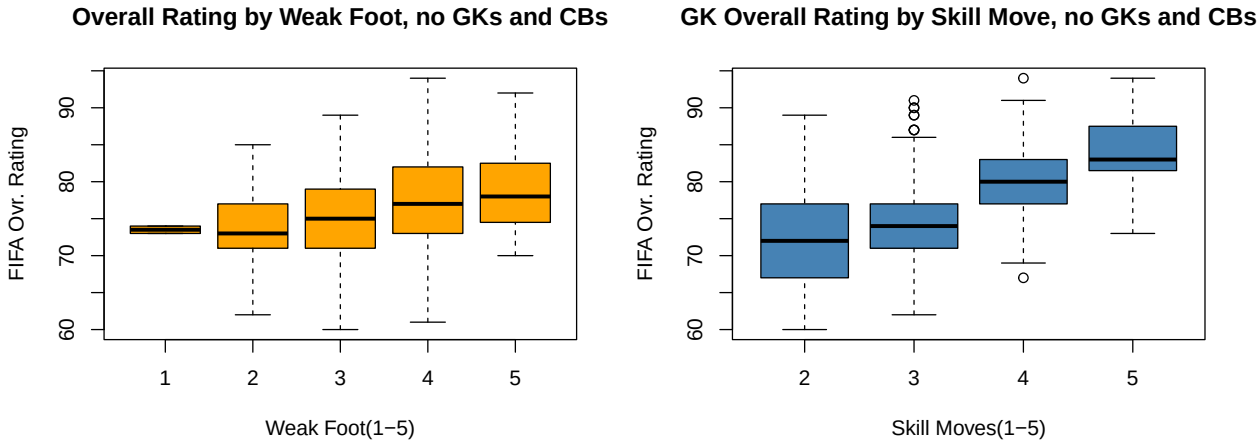
We see some similar issues in the data for centerbacks(CBs). Weak foot seems to have very little correlation with overall rating for CBs, and every CB in the game only have a skill move rating of 2 or 3.

*It is worth noting that the positions column lists all the positions a player can play in one string. So by filtering `positions == CB`, we are selecting players who play CB as their only position. Domagoj Vida, for example, has positions "CB,RB", so he is not included in this filter even though his primary position is CB. We played around with using the stringr library to include these players as well, but the results weren't as meaningful, so we chose to filter players who play exclusively CB, which is the majority of CBs. This wasn't an issue for goalkeepers as they all only play goalkeeper; outfield positions are more fluid creating this issue.

Repeating our analysis without GKs and CBs:

```
filtered <- fifa %>% filter(positions != "GK" & positions != "CB")
par(mfrow=c(1,2))
boxplot(filtered$overall_rating ~ filtered$weak_foot.1.5.,
        col='orange',
        main='Overall Rating by Weak Foot, no GKs and CBs',
        xlab='Weak Foot(1-5)',
        ylab='FIFA Ovr. Rating')

boxplot(filtered$overall_rating ~ filtered$skill_moves.1.5.,
        col='steelblue',
        main='GK Overall Rating by Skill Move, no GKs and CBs',
        xlab='Skill Moves(1-5)',
        ylab='FIFA Ovr. Rating')
```




```
discrete_lm <- lm(data=filtered, overall_rating ~ weak_foot.1.5. + skill_moves.1.5. )
summary(discrete_lm)
```

```
##
## Call:
## lm(formula = overall_rating ~ weak_foot.1.5. + skill_moves.1.5.,
##     data = filtered)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -13.734  -3.734  -0.750   3.132  18.148
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)       60.2229     1.2734  47.292  < 2e-16 ***
## weak_foot.1.5.     0.8656     0.2984   2.901  0.00386 **
## skill_moves.1.5.   4.0162     0.3317  12.108  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.249 on 584 degrees of freedom
## Multiple R-squared:  0.237,  Adjusted R-squared:  0.2344
## F-statistic: 90.69 on 2 and 584 DF,  p-value: < 2.2e-16
```

Fig. 10

Based on the new boxplots, filtering by position didn't affect weak foot that much, but the skill moves plot looks very strongly correlated now. These observations are backed up by the updated linear model, which has a slightly higher $\hat{\beta}$ for weak foot but much higher for skill moves.

In conclusion, there is only a small amount of positive correlation between weak foot and overall rating. It is unclear if weak foot really has an effect on overall rating or if the two variables are just correlated. On the other hand, there is a strong positive correlation between skill moves and overall rating. Skill moves is a strong predictor of overall rating. It is either having a strong effect on overall rating, or it is very strongly correlated with a variable that is, such as dribbling or ball_control. However, all goalkeepers in FIFA have one star skill moves, and they are the only players in the game with this. Thus, skill moves is only a useful predictor for non goalkeepers, and using it in a dataset with goalkeepers will make it less effective of a predictor. A similar effect also applies to centerbacks, although less extreme.

Research Question 3:

Does a player's overall rating contribute to their market value?

```
correlation <- cor(fifa$overall_rating, fifa$value_euro)
print(paste("Correlation: ", round(correlation, 4)))
```

```
## [1] "Correlation:  0.8283"
```

```
# Fit basic SLR
model <- lm(data = training_data, value_euro ~ overall_rating)

# Extract and print R-squared value
r_squared <- summary(model)$r.squared
print(paste0("R-squared: ", round(r_squared, 4)))
```

```
## [1] "R-squared: 0.6833"
```

```
ggplot(fifa, aes(x = overall_rating, y = value_euro)) +
  geom_point() +
  geom_smooth(method = "lm") +
  labs(title = "Overall Rating vs Market Value", x = "Overall Rating", y = "Market Value (€)")
```

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

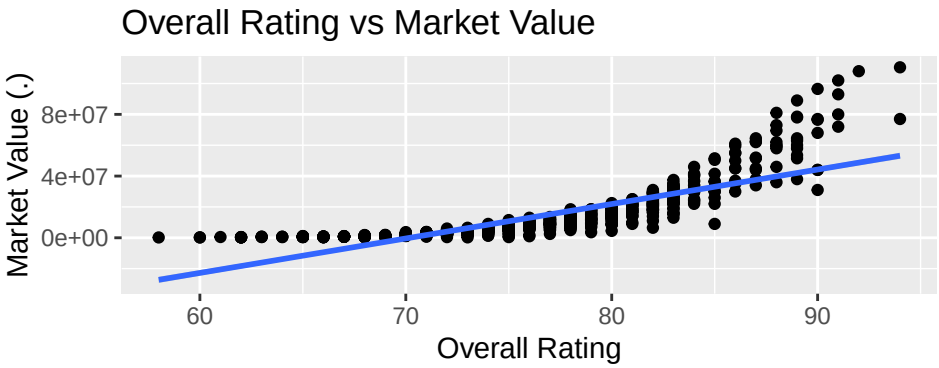


Fig. 11

We have a moderately strong positive relationship between a player's overall rating and their market value from a correlation of 0.6309, As we can see, this is horrendous. Let's look into a transformation!

```
boxcox(model, plotit = T)
```

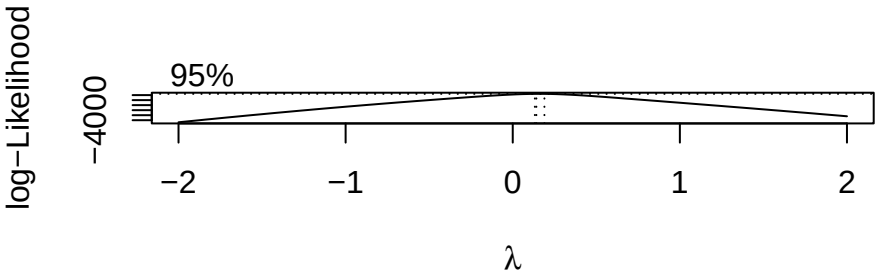


Fig. 12

We can see $\lambda = 0$ so let's try a log transformation!

```r
log_correlation <- cor(fifa$overall_rating, log(fifa$value_euro), use = "complete.obs")
print(paste("Log-transformation Correlation: ", round(log_correlation, 4)))
```

```
## [1] "Log-transformation Correlation:  0.9518"
```

```r
# Fit basic SLR
log_transformed_model <- lm(data = training_data, log(value_euro) ~ overall_rating)

# Extract and print R-squared value
r_squared_log <- summary(log_transformed_model)$r.squared
print(paste0("Log Transformed R-squared: ", round(r_squared_log , 4)))
```

```
## [1] "Log Transformed R-squared: 0.903"
```

```r
ggplot(fifa, aes(x = overall_rating, y = log(value_euro))) +
  geom_point() +
  geom_smooth(method = "lm") +
  labs(title = "Overall Rating vs Market Value", x = "Overall Rating", y = "Log Market Value (€)")
```

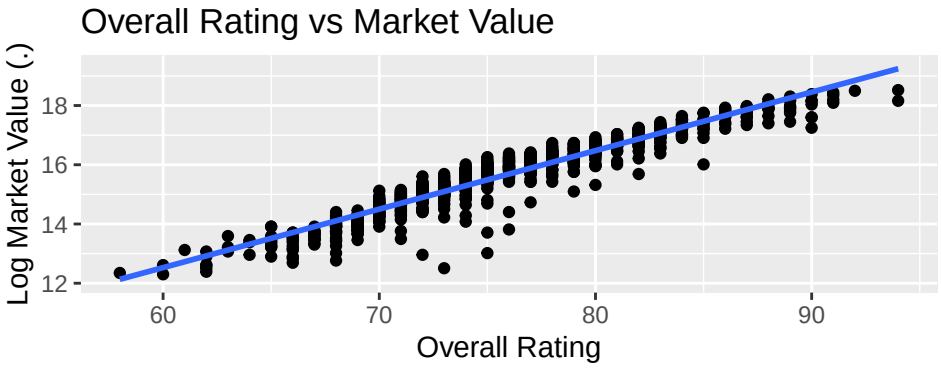


Fig. 13

Our correlation suggests a strong relationship between player overall rating and market value after taking the log-transform of market value. Our r squared is much higher now too.

Research Question 4:

What is the best model?

We use adjusted R squared and AIC to find best model because a model that has more predictors obviously will have a higher R squared (proportional of variability explained by the model)

Due to the nature that many predictors are significant and they have a lot of multicolinearity.

So, for the best model, we might not use agility and stamina, but instead use reactions.

```r
best_guess_model <- lm(data=fifa, overall_rating ~   sprint_speed +
                         dribbling + height_cm  + log(value_euro) + wage_euro +
                         national_rating + crossing +dribbling  + balance+jumping +
                         reactions + penalties)


summary(best_guess_model)
```

```
##
## Call:
```

```
## lm(formula = overall_rating ~ sprint_speed + dribbling + height_cm +
##     log(value_euro) + wage_euro + national_rating + crossing +
##     dribbling + balance + jumping + reactions + penalties, data = fifa)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.5097 -0.7755 -0.1288  0.5854  7.0677
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      1.559e+01  2.447e+00   6.369 3.25e-10 ***
## sprint_speed    -7.842e-03  5.398e-03  -1.453 0.146703
## dribbling       -4.340e-02  6.230e-03  -6.966 6.93e-12 ***
## height_cm       -3.469e-02  1.144e-02  -3.033 0.002505 **
## log(value_euro)  3.200e+00  7.596e-02  42.134  < 2e-16 ***
## wage_euro        1.367e-05  9.944e-07  13.750  < 2e-16 ***
## national_rating  7.652e-02  1.376e-02   5.561 3.70e-08 ***
## crossing         1.393e-02  4.754e-03   2.930 0.003492 **
## balance         -1.828e-02  6.600e-03  -2.769 0.005752 **
## jumping          1.342e-02  3.976e-03   3.376 0.000773 ***
## reactions        1.683e-01  1.111e-02  15.150  < 2e-16 ***
## penalties        1.133e-02  4.336e-03   2.614 0.009124 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.247 on 777 degrees of freedom
## Multiple R-squared:  0.9587, Adjusted R-squared:  0.9581
## F-statistic:  1638 on 11 and 777 DF,  p-value: < 2.2e-16
```

```
## All subsets
library(leaps) # this is the library that contains the regsubsets function

regsubsets.out <- regsubsets(overall_rating ~  age + height_cm +weight_kgs +
                          log(value_euro) + wage_euro + international_reputation.1.5. +
                          national_rating + crossing + finishing + heading_accuracy +
                          short_passing + volleys + dribbling + curve + freekick_accuracy +
                          long_passing + ball_control + acceleration + sprint_speed+agility +
                          reactions + balance +shot_power +jumping + stamina +strength + long_shots +
                          aggression + interceptions +vision + positioning +penalties + composure +
                          marking + standing_tackle +sliding_tackle   ,data=fifa, nvmax=10)
                           # nvmax indicates the max number of predictors




summary(regsubsets.out)$cp
```

```
##  [1] 2833.81252  799.05330  473.61495  259.76458  153.98098   90.66128
##  [7]   50.19510   34.16193   29.47486   23.34656
```

```
summary(regsubsets.out)$bic
```

```
##  [1] -1851.187 -2497.353 -2673.647 -2815.937 -2895.700 -2946.425 -2979.441
##  [8] -2990.247 -2990.156 -2991.558
```

```r
summary(regsubsets.out)$adjr2
```

```
##  [1] 0.9057575 0.9587472 0.9672436 0.9728446 0.9756312 0.9773120 0.9783974
##  [8] 0.9788435 0.9789923 0.9791794
```

```r
fit1 <- lm(overall_rating ~ age + height_cm +weight_kgs + log(value_euro) + wage_euro + international_r
```

From our output from above, we can get some really good models with adjusted R squared near 0.9789. Additionally, we use the step function for AIC below.

```r
# Most comprehensive is dir='both'
best_model <- step(fit1, dir="both", trace=0) # k=2 by default (AIC), trace=0 to not show each step
summary(best_model)
```

```
##
## Call:
## lm(formula = overall_rating ~ age + height_cm + weight_kgs +
##     log(value_euro) + wage_euro + international_reputation.1.5. +
##     national_rating + short_passing + long_passing + sprint_speed +
##     reactions + balance + shot_power + aggression + positioning +
##     composure + marking, data = fifa)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.2672 -0.5611 -0.0581  0.4672  5.4933
##
## Coefficients:
##                                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)                    4.878e+00  1.809e+00   2.696 0.007165 **
## age                            2.619e-01  1.107e-02  23.652  < 2e-16 ***
## height_cm                     -2.973e-02  9.023e-03  -3.295 0.001028 **
## weight_kgs                     1.533e-02  7.056e-03   2.172 0.030132 *
## log(value_euro)                3.682e+00  6.040e-02  60.962  < 2e-16 ***
## wage_euro                      7.982e-06  8.037e-07   9.931  < 2e-16 ***
## international_reputation.1.5.  4.694e-01  5.987e-02   7.839 1.51e-14 ***
## national_rating                6.154e-02  9.627e-03   6.393 2.82e-10 ***
## short_passing                 -2.011e-02  7.109e-03  -2.829 0.004797 **
## long_passing                   8.072e-03  5.270e-03   1.532 0.125987
## sprint_speed                   1.588e-02  3.624e-03   4.380 1.35e-05 ***
## reactions                      7.674e-02  8.634e-03   8.888  < 2e-16 ***
## balance                       -7.848e-03  4.607e-03  -1.704 0.088854 .
## shot_power                    -8.491e-03  3.865e-03  -2.197 0.028351 *
## aggression                    -7.676e-03  3.109e-03  -2.469 0.013774 *
## positioning                   -1.718e-02  3.606e-03  -4.764 2.27e-06 ***
## composure                      1.410e-02  5.458e-03   2.584 0.009961 **
## marking                        9.307e-03  2.600e-03   3.579 0.000366 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8654 on 771 degrees of freedom
## Multiple R-squared:  0.9803, Adjusted R-squared:  0.9798
## F-statistic:  2251 on 17 and 771 DF,  p-value: < 2.2e-16
```

Response to Teacher Question 4:

"Is your goal prediction or interpretation? I think prediction is suitable for your dataset. Compute prediction intervals and you could also use a training/test approach."

The best model for prediction isn't necessarily the best model found for analysis via stepwise regression or regsubsets(), but in this case, using our best analysis model works really well. Now that we have our best model:

```
# Note this is from research question 4 (our best model)
prediction_model <- lm(formula = overall_rating ~ age + height_cm + weight_kgs +
    log(value_euro) + wage_euro + international_reputation.1.5. +
    national_rating + short_passing + long_passing + sprint_speed +
    reactions + balance + long_shots + aggression + positioning +
    composure + marking, data = training_data)

#predicted_overall_rating <- predict(prediction_model, newdata = test_data)
#test_data$overall_rating - predicted_overall_rating


# Get prediction intervals for test data
pred_intervals <- predict(prediction_model, newdata = test_data, interval = "prediction", level = 0.95)

# Combine actual vs predicted values with intervals
results <- cbind(test_data$overall_rating, pred_intervals)
colnames(results) <- c("Actual", "Predicted", "Lower_Bound", "Upper_Bound")

# print first 5 rows
print(head(results))
```

```
##    Actual Predicted Lower_Bound Upper_Bound
## 4      88  86.93408    85.17880    88.68937
## 6      88  88.16662    86.40732    89.92592
## 9      89  88.54906    86.76143    90.33669
## 12     89  88.82839    87.06246    90.59432
## 21     87  86.24289    84.50502    87.98077
## 26     87  86.21659    84.47465    87.95853
```

```
# Compute RMSE to evaluate prediction accuracy
# (----->look into other accuracy tests[like why root of MSE and not just MSE]? I just looked this one
rmse <- sqrt(mean((results[, "Actual"] - results[, "Predicted"])^2))
cat("Root Mean Squared Error (RMSE):", rmse)
```

```
## Root Mean Squared Error (RMSE): 0.8487813
```

```
# Combine actual vs predicted with lower and upper bounds
results <- cbind(test_data$overall_rating, pred_intervals)
colnames(results) <- c("Actual", "Predicted", "Lower_Bound","Upper_Bound")
# Plot Actual vs Predicted with prediction intervals
ggplot(results, aes(x = Actual, y = Predicted)) +
  geom_point(aes(color = "Actual vs Predicted"), size = 2, shape = 21, color = "black", fill="pink") +
  geom_errorbar(aes(ymin = Lower_Bound, ymax = Upper_Bound), width = 0.2) +
  geom_abline( color = "blue", linetype ="dashed") +
  labs(title = "Actual vs Predicted Overall Rating with Prediction Intervals",
       x = "Actual Overall Rating",y = "Predicted Overall Rating")
```

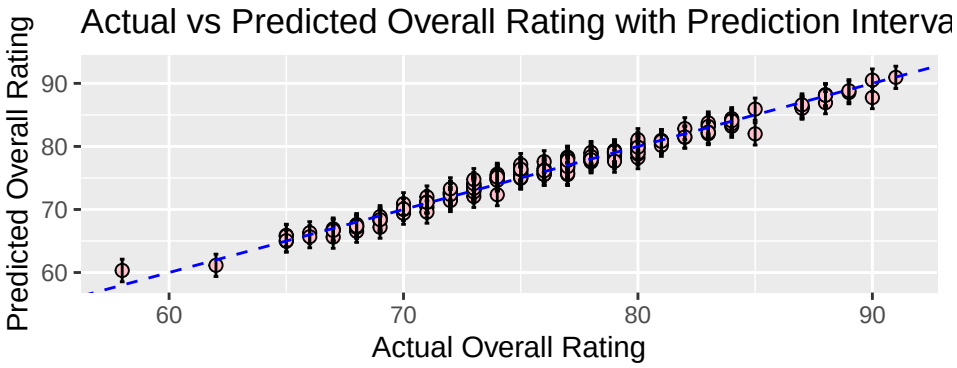


Fig. 14

Contributions

Itroduction:

```
- Loading and Cleaning the Data: Maxx
- Removing Outliers: Harry
- Response to Teacher Question 3: Arnav
```

Research Question 1: Arnav

Research Question 2: Harry

Research Question 3: Casey

Research Question 4: Casey

Teacher Question 4: Casey and Mohit

Organization and formatting: Maxx