Final\_Project\_Clean

Yuka, Adela, Jack

2022-04-30

# 1. The statement of the problem

NBA players are on average the highest-paid athletes in the world, according to Statista.com. The NBA players get paid an average salary of around 7.5 million. The median salary is about 3.8 million. The highest salary in the NBA for the 2016-2017 season is about 25 million, including superstar LeBron James from Cleveland Cavaliers.

Oftentimes sports players would seem to have major contracts with really high annual salaries (some people would even think they should not get paid so much).

Since of one our group members is a super fan of the NBA, he believes that those basketball players are paid by their season total performance. However, other members in our group think otherwise.

Through this project, we want to find out whether the NBA players and their season total performance have a strong positive correlation.

For this project, we would use the 2016-2017 season total performance and actual salaries to create a prediction model. Then, we fit the 2017-2018 season total performance to the prediction model, see the difference between the salaries we expected during 2017-2018 season and the actual salaries in 2017-2018.

### Purpose:

1. Discover which predictors variables are critical to the salaries of the NBA players
2. Use a multiple regression model to predict NBA players’ salaries
3. Examine the difference between the predicted salaries and actual salaries

# 2. Data Section

### Data source

Our season total performance and salary data sets were collected from Basketball Reference (<https://www.basketball-reference.com/>)

### Processing the data

1. Data set:

Combines NBA player performance and salary data by using player ID and team. During the regular season, some of the players will change their team, so they have two different performances and salaries.

1. Data cleaning:

We combined three datasets (NBA player salaries summary 1985-2018, season performance for both 2016-2017 and 2017-2018) from Basketball Reference. We joined the data sets based on the playerID and their team names. Originally, the data set has a total of 34 variables, including 6 categorical variables, and 28 continuous variables.

In the first step of data cleaning, we removed 20 variables that seem to be either duplicated or are a combination of other variables. (i.e. trb = orb + drb). Later, we dropped the person who has the total performances as they transferred the teams during the season in order to focus on their performance.

# 3. Model Building Process

NBA <- read\_csv("NBA.csv")

## Rows: 418 Columns: 35  
## ── Column specification ────────────────────────────────────────────────────────  
## Delimiter: ","  
## chr (6): player, player\_id, trans\_team, pos, tm, name  
## dbl (29): rk, age, g, gs, mp, fg, fga, fg%, 3p, 3pa, 3p%, 2p, 2pa, 2p%, efg%...  
##   
## ℹ Use `spec()` to retrieve the full column specification for this data.  
## ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

head(NBA, 5)

## # A tibble: 5 × 35  
## rk player player\_id trans\_team pos age tm g gs mp fg  
## <dbl> <chr> <chr> <chr> <chr> <dbl> <chr> <dbl> <dbl> <dbl> <dbl>  
## 1 1 Alex Abr… abrinal01 none SG 23 OKC 68 6 1055 134  
## 2 2 Quincy A… acyqu01 trans PF 26 DAL 6 0 48 5  
## 3 2 Quincy A… acyqu01 trans PF 26 BRK 32 1 510 65  
## 4 3 Steven A… adamsst01 none C 23 OKC 80 80 2389 374  
## 5 4 Arron Af… afflaar01 none SG 31 SAC 61 45 1580 185  
## # … with 24 more variables: fga <dbl>, `fg%` <dbl>, `3p` <dbl>, `3pa` <dbl>,  
## # `3p%` <dbl>, `2p` <dbl>, `2pa` <dbl>, `2p%` <dbl>, `efg%` <dbl>, ft <dbl>,  
## # fta <dbl>, `ft%` <dbl>, orb <dbl>, drb <dbl>, trb <dbl>, ast <dbl>,  
## # stl <dbl>, blk <dbl>, tov <dbl>, pf <dbl>, pts <dbl>, name <chr>,  
## # `16\_17\_salary` <dbl>, `17\_18salary` <dbl>

### Data Set - salary with 14 predictors variables

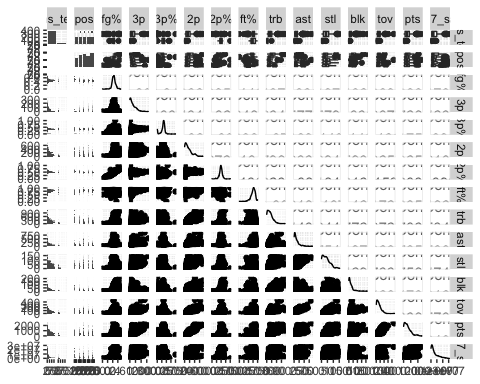
The predictor variables include 2 categorical variables and 12 continuous variables.

original\_data <- NBA %>%   
 select(-rk, -player, -player\_id, -`17\_18salary`,   
 -name, -tm, -`fg`, -`fga`, -`3pa`, -`2pa`, -ft, -fta, -g, -gs,   
 -`efg%`, -mp, -orb, - drb, -pf, -age)  
  
head(original\_data, 5)

## # A tibble: 5 × 15  
## trans\_team pos `fg%` `3p` `3p%` `2p` `2p%` `ft%` trb ast stl blk  
## <chr> <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 none SG 0.393 94 0.381 40 0.426 0.898 86 40 37 8  
## 2 trans PF 0.294 1 0.143 4 0.4 0.667 8 0 0 0  
## 3 trans PF 0.425 36 0.434 29 0.414 0.754 107 18 14 15  
## 4 none C 0.571 0 0 374 0.572 0.611 613 86 89 78  
## 5 none SG 0.44 62 0.411 123 0.457 0.892 125 78 21 6  
## # … with 3 more variables: tov <dbl>, pts <dbl>, `16\_17\_salary` <dbl>

### Correlation Analysis

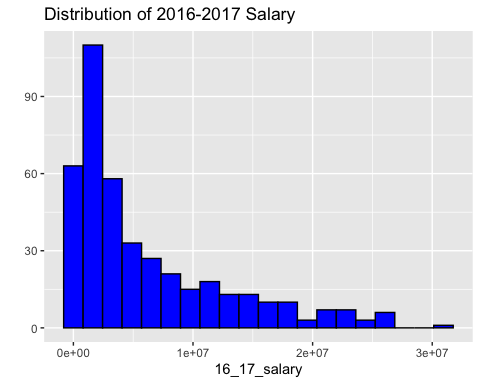
original\_data %>%  
 ggpairs()



Based on the correlation plot, we can see the strongest linear relationship occurs between salary and points, although there could be a bit of a curvi-linear relationship. 3p, 2p, trb, ast, stl, tov have strong relationships as well.

### Checking Y Predictable

qplot(data = original\_data, x = `16\_17\_salary`,   
 geom = "histogram",   
 main = "Distribution of 2016-2017 Salary",   
 bins = 20, color = I("black"), fill = I("blue"))

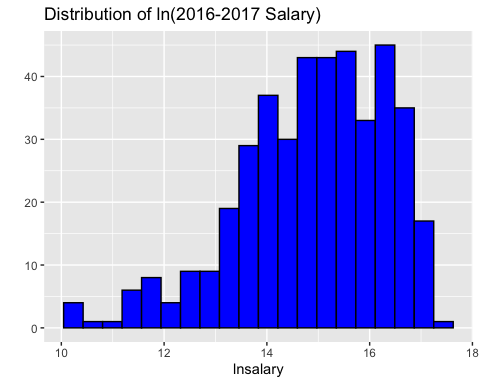


According to the histogram plot, we can see that the plot is right-skewed. To remedy the issue, we transformed the salary by taking logs.

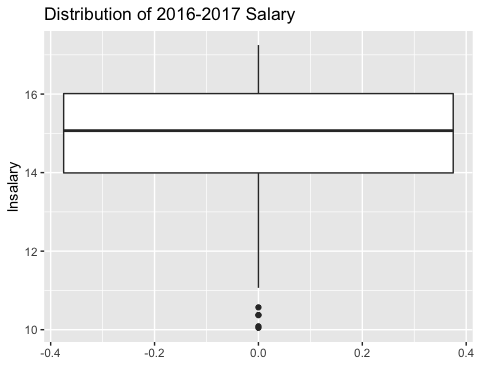
### Log Salary for Better Prediction

original\_data %>%  
 mutate(lnsalary = log(`16\_17\_salary`)) ->   
 original\_data

qplot(data = original\_data, x = lnsalary, geom = "histogram",   
 bins = 20, color = I("black"), fill = I("blue"),   
 main = "Distribution of ln(2016-2017 Salary)")



qplot(data = original\_data, y = lnsalary, geom = "boxplot",  
 main = "Distribution of 2016-2017 Salary")



Examining the histogram of log salary, we see possibly a very slight left skew, but it is closer to symmetric than without transform salary. According to the box plt, there are some of the outlines.

### Original Model(only continuous variables)

original\_data %>%   
 select(-`16\_17\_salary`, -pos , -trans\_team) ->   
 new\_og\_data  
  
og\_model <- lm(lnsalary ~ ., data = new\_og\_data)  
  
mult\_og <- tidy(og\_model)  
mult\_og

## # A tibble: 13 × 5  
## term estimate std.error statistic p.value  
## <chr> <dbl> <dbl> <dbl> <dbl>  
## 1 (Intercept) 13.0 0.509 25.6 9.64e-87  
## 2 `fg%` 6.46 1.50 4.29 2.19e- 5  
## 3 `3p` 0.0111 0.00406 2.74 6.46e- 3  
## 4 `3p%` -0.0287 0.479 -0.0600 9.52e- 1  
## 5 `2p` 0.00425 0.00292 1.46 1.46e- 1  
## 6 `2p%` -5.02 1.18 -4.24 2.79e- 5  
## 7 `ft%` 0.400 0.504 0.793 4.28e- 1  
## 8 trb 0.00241 0.000644 3.75 2.00e- 4  
## 9 ast 0.00133 0.00105 1.27 2.03e- 1  
## 10 stl 0.00191 0.00318 0.602 5.48e- 1  
## 11 blk -0.00397 0.00314 -1.27 2.07e- 1  
## 12 tov -0.00240 0.00309 -0.776 4.38e- 1  
## 13 pts -0.00109 0.00116 -0.940 3.48e- 1

* Fitted model: = 13.05 + 6.5 + 0.01 - 0.028 + 0.004 - 5.02 + 0.39 + 0.0024 + 0.0013 + 0.0019

g\_mult\_og <- broom::glance(og\_model)  
g\_mult\_og

## # A tibble: 1 × 12  
## r.squared adj.r.squared sigma statistic p.value df logLik AIC BIC  
## <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 0.427 0.410 1.10 25.1 5.40e-42 12 -627. 1283. 1339.  
## # … with 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>

og\_anova <- lb\_anovat\_lm(og\_model, reg\_collapse = TRUE)  
og\_anova

## Source Df SS MS F P  
## 1 Regression 12 366.3515 30.529290 25.11287 5.396828e-42  
## 2 Error 405 492.3517 1.215683 NA NA  
## 3 Total 417 858.7032 2.059240 NA NA

vif(og\_model)

## `fg%` `3p` `3p%` `2p` `2p%` `ft%` trb   
## 5.212855 17.304773 1.377049 58.726716 4.390578 1.318019 5.222122   
## ast stl blk tov pts   
## 6.799010 3.645980 2.771145 12.541091 107.912304

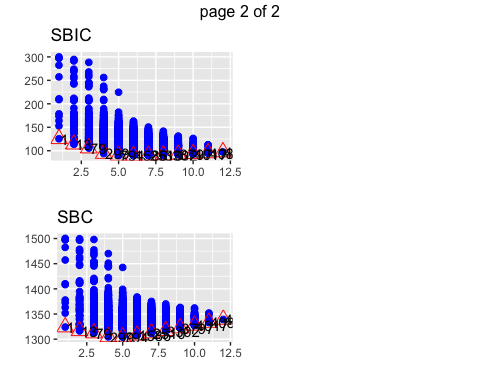
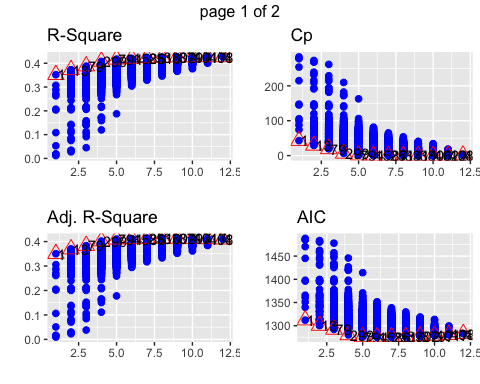
According to the VIF, it shows our original model isn’t good enought to be our final model. Some of the varables are more than 5.

### All Subset Models

all\_subsets\_model <- ols\_step\_all\_possible(og\_model)

plot(all\_subsets\_model)

## Warning: It is deprecated to specify `guide = FALSE` to remove a guide. Please use `guide = "none"` instead.  
## It is deprecated to specify `guide = FALSE` to remove a guide. Please use `guide = "none"` instead.  
## It is deprecated to specify `guide = FALSE` to remove a guide. Please use `guide = "none"` instead.  
## It is deprecated to specify `guide = FALSE` to remove a guide. Please use `guide = "none"` instead.  
## It is deprecated to specify `guide = FALSE` to remove a guide. Please use `guide = "none"` instead.  
## It is deprecated to specify `guide = FALSE` to remove a guide. Please use `guide = "none"` instead.  
## It is deprecated to specify `guide = FALSE` to remove a guide. Please use `guide = "none"` instead.  
## It is deprecated to specify `guide = FALSE` to remove a guide. Please use `guide = "none"` instead.  
## It is deprecated to specify `guide = FALSE` to remove a guide. Please use `guide = "none"` instead.  
## It is deprecated to specify `guide = FALSE` to remove a guide. Please use `guide = "none"` instead.  
## It is deprecated to specify `guide = FALSE` to remove a guide. Please use `guide = "none"` instead.  
## It is deprecated to specify `guide = FALSE` to remove a guide. Please use `guide = "none"` instead.  
## It is deprecated to specify `guide = FALSE` to remove a guide. Please use `guide = "none"` instead.  
## It is deprecated to specify `guide = FALSE` to remove a guide. Please use `guide = "none"` instead.  
## It is deprecated to specify `guide = FALSE` to remove a guide. Please use `guide = "none"` instead.  
## It is deprecated to specify `guide = FALSE` to remove a guide. Please use `guide = "none"` instead.  
## It is deprecated to specify `guide = FALSE` to remove a guide. Please use `guide = "none"` instead.  
## It is deprecated to specify `guide = FALSE` to remove a guide. Please use `guide = "none"` instead.



* Based on the , Mallow’s cp, and AIC criteria, we would choose the model that contains all 5 variables (Model 793). Model 793 has 5 variables, Cp = 6.09, AIC = 1275.92, $R^2{adj} = 0.41.

### Reduce Model based on All Subset Models Method

reduce <- NBA %>%   
 mutate(lnsalary = log(`16\_17\_salary`)) %>%   
 select(lnsalary,`fg%`,`3p`,`2p`,`2p%`,trb, pos, trans\_team)

reducemodel <- lm(lnsalary ~`fg%` + `3p` + `2p`+`2p%`+ trb , data = new\_og\_data)  
  
reduce\_model\_t <- tidy(reducemodel)   
reduce\_model\_t

## # A tibble: 6 × 5  
## term estimate std.error statistic p.value  
## <chr> <dbl> <dbl> <dbl> <dbl>  
## 1 (Intercept) 13.4 0.320 42.0 7.68e-151  
## 2 `fg%` 6.31 1.46 4.32 1.92e- 5  
## 3 `3p` 0.00857 0.00114 7.51 3.76e- 13  
## 4 `2p` 0.00172 0.000687 2.50 1.29e- 2  
## 5 `2p%` -5.04 1.16 -4.34 1.76e- 5  
## 6 trb 0.00183 0.000479 3.81 1.58e- 4

\*Reduce model: = 13.441 + 6.308 + 0.009 + 0.002 - 5.039 + 0.002

reduce\_model\_g <- broom::glance(reducemodel)  
reduce\_model\_g

## # A tibble: 1 × 12  
## r.squared adj.r.squared sigma statistic p.value df logLik AIC BIC  
## <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 0.417 0.410 1.10 58.8 3.76e-46 5 -631. 1276. 1304.  
## # … with 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>

reduce\_model\_a <- lb\_anovat\_lm(reducemodel)  
reduce\_model\_a

## Source Df SS MS F P  
## 1 Regression 5 357.7327 71.546533 58.84013 3.755897e-46  
## 2 Error 412 500.9705 1.215948 NA NA  
## 3 Total 417 858.7032 2.059240 NA NA

tidy(reducemodel, conf.int = "TRUE", conf.level = 0.98)

## # A tibble: 6 × 7  
## term estimate std.error statistic p.value conf.low conf.high  
## <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 (Intercept) 13.4 0.320 42.0 7.68e-151 12.7 14.2   
## 2 `fg%` 6.31 1.46 4.32 1.92e- 5 2.90 9.71   
## 3 `3p` 0.00857 0.00114 7.51 3.76e- 13 0.00590 0.0112   
## 4 `2p` 0.00172 0.000687 2.50 1.29e- 2 0.000112 0.00332  
## 5 `2p%` -5.04 1.16 -4.34 1.76e- 5 -7.75 -2.33   
## 6 trb 0.00183 0.000479 3.81 1.58e- 4 0.000707 0.00294

vif(reducemodel)

## `fg%` `3p` `2p` `2p%` trb   
## 4.896551 1.364878 3.252252 4.211965 2.888815

According to the VIF, it shows our original model is good enough to be our final model, all of the variables are less than 5.

### Adding Dummy Variables - lnsalary with 5 continuous predictors variables

We want to figure out will the position they play and transfer to different teams during the regular season influence their salaries?

Since some players had transferred teams during the season, we decided to create a dummy variable for transfer team or not (0=did not transfer, 1=transferred). We also created dummy variables for their positions to predict salary based on the position they played (c = center, pf = power forward, sf = small forward, pg = point guard, and sg = shooting guard; 0 = did not play in position, 1 = played in that position).

results <- dummy\_cols(.data = reduce, select\_columns = c("pos","trans\_team"))  
  
results %>%  
 select(pos, pos\_C, pos\_PF, pos\_PG, pos\_SF, pos\_SG, trans\_team, trans\_team\_none,   
 trans\_team\_trans) %>%  
 head(6)

## # A tibble: 6 × 9  
## pos pos\_C pos\_PF pos\_PG pos\_SF pos\_SG trans\_team trans\_team\_none  
## <chr> <int> <int> <int> <int> <int> <chr> <int>  
## 1 SG 0 0 0 0 1 none 1  
## 2 PF 0 1 0 0 0 trans 0  
## 3 PF 0 1 0 0 0 trans 0  
## 4 C 1 0 0 0 0 none 1  
## 5 SG 0 0 0 0 1 none 1  
## 6 C 1 0 0 0 0 none 1  
## # … with 1 more variable: trans\_team\_trans <int>

newresult <- dummy\_cols(.data = reduce, select\_columns = c("pos","trans\_team"),   
 remove\_selected\_columns = TRUE)  
  
rename(.data = newresult, trans = trans\_team\_trans) -> newdummy  
  
dummy\_model <- lm(lnsalary ~ ., data = newdummy)

dumtidyout <- tidy(dummy\_model)  
  
dumglout <- glance(dummy\_model)  
  
dumtidyout

## # A tibble: 13 × 5  
## term estimate std.error statistic p.value  
## <chr> <dbl> <dbl> <dbl> <dbl>  
## 1 (Intercept) 12.3 0.399 30.7 3.76e-108  
## 2 `fg%` 6.36 1.50 4.24 2.72e- 5  
## 3 `3p` 0.00867 0.00119 7.26 1.99e- 12  
## 4 `2p` 0.00178 0.000696 2.56 1.07e- 2  
## 5 `2p%` -4.95 1.16 -4.26 2.56e- 5  
## 6 trb 0.00132 0.000525 2.51 1.26e- 2  
## 7 pos\_C 0.274 0.226 1.21 2.27e- 1  
## 8 pos\_PF 0.451 0.173 2.60 9.56e- 3  
## 9 pos\_PG 0.313 0.157 1.99 4.71e- 2  
## 10 pos\_SF 0.516 0.168 3.07 2.27e- 3  
## 11 pos\_SG NA NA NA NA   
## 12 trans\_team\_none 0.955 0.225 4.25 2.65e- 5  
## 13 trans NA NA NA NA

dumglout

## # A tibble: 1 × 12  
## r.squared adj.r.squared sigma statistic p.value df logLik AIC BIC  
## <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 0.459 0.446 1.07 34.5 1.86e-48 10 -615. 1255. 1303.  
## # … with 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>

dummy\_a <- lb\_anovat\_lm(dummy\_model, reg\_collapse = FALSE)  
dummy\_a

## Source Df SS MS F P  
## 1 `fg%` 1 45.7115354 45.7115354 40.0336807 6.589589e-10  
## 2 `3p` 1 181.4834296 181.4834296 158.9412740 5.413203e-31  
## 3 `2p` 1 89.2304456 89.2304456 78.1470834 2.918848e-17  
## 4 `2p%` 1 23.6276402 23.6276402 20.6928382 7.125296e-06  
## 5 trb 1 17.6796146 17.6796146 15.4836201 9.784910e-05  
## 6 pos\_C 1 0.1990211 0.1990211 0.1743006 6.765379e-01  
## 7 pos\_PF 1 1.8985145 1.8985145 1.6626990 1.979716e-01  
## 8 pos\_PG 1 0.3603853 0.3603853 0.3156217 5.745600e-01  
## 9 pos\_SF 1 13.1593822 13.1593822 11.5248482 7.541662e-04  
## 10 trans\_team\_none 1 20.6296784 20.6296784 18.0672548 2.646649e-05  
## 11 Error 407 464.7235673 1.1418269 NA NA  
## 12 Total 417 858.7032143 2.0592403 NA NA

* Reduce model: = 12.277 + 6.357 + 0.009 + 0.002 -4.949 + 0.001 + 0.274 + 0.451 + 0.313 + 0.516 + 0.955

### Comparing models

dumglout #dummy models

## # A tibble: 1 × 12  
## r.squared adj.r.squared sigma statistic p.value df logLik AIC BIC  
## <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 0.459 0.446 1.07 34.5 1.86e-48 10 -615. 1255. 1303.  
## # … with 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>

g\_mult\_og #full model wihtout dummy

## # A tibble: 1 × 12  
## r.squared adj.r.squared sigma statistic p.value df logLik AIC BIC  
## <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 0.427 0.410 1.10 25.1 5.40e-42 12 -627. 1283. 1339.  
## # … with 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>

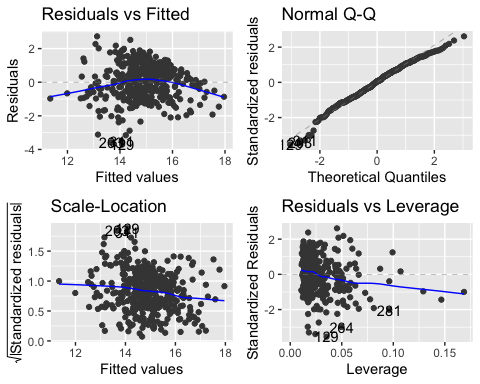
reduce\_model\_g # reduce model

## # A tibble: 1 × 12  
## r.squared adj.r.squared sigma statistic p.value df logLik AIC BIC  
## <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 0.417 0.410 1.10 58.8 3.76e-46 5 -631. 1276. 1304.  
## # … with 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>

For the full model without dummy variables (12 variables) = 0.4096448 For the reduced model (5 variables) = 0.4095163  
For the reduce model including dummy variables (10 variables) = 0.4455106  
In terms of , the reduce model including dummy variables does a better job fitting the data as it has the higher

### Model Testing

# Assumtion  
autoplot(dummy\_model)



Based on the fitted residual plot, it seems some multi-linear regression assumptions are violated.

# Residual normality test  
shapiro.test(dummy\_model$residuals)

##   
## Shapiro-Wilk normality test  
##   
## data: dummy\_model$residuals  
## W = 0.98822, p-value = 0.00187

According to the Shapiro-Wilk normality test with a test statistic of 0.99 and an associated p-value = 0.00187, since the p-value is below 0.05 which indicates the NBA Dummy data significantly deviate from a normal distribution.

# Residual independence test  
durbinWatsonTest(dummy\_model)

## lag Autocorrelation D-W Statistic p-value  
## 1 0.09443852 1.80766 0.052  
## Alternative hypothesis: rho != 0

* = residual from the regression are not auto-correlated (autocorrelation coefficient, p=0)
* : Alter = residuals from the regression are auto-correlated (AC, p > 0)

According to the Durbin-Watson test with a test statistic of 1.81 and an associated p-value = 0.024, since the test statistic fell into the range of 0 to 2, which indicates that that there is a positive autocorrelation.

# Residual variance homogeneity test  
ncvTest(dummy\_model)

## Non-constant Variance Score Test   
## Variance formula: ~ fitted.values   
## Chisquare = 16.55129, Df = 1, p = 4.7352e-05

According to the non-constant variance score test with a chisquare 16.5513 and an associated p-value = 0.000047, which indicates that there has a heteroskedasticity issue.

# Testing for Non-Constant Variance Residual by using Breusch-Pagan  
library(lmtest)

## Loading required package: zoo

##   
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':  
##   
## as.Date, as.Date.numeric

bptest(dummy\_model)

##   
## studentized Breusch-Pagan test  
##   
## data: dummy\_model  
## BP = 28.039, df = 10, p-value = 0.001779

Test: \* \*

According to the Breusch-Pagan Test with a test statistic = 28.039 and an associated p-value = 0.0018, under the assumption that is true (variance of the disturbance terms is constant), it would be quite unlikely that we would observe a test statistic of our magnitude or larger.

Consequently, we can reject and conclude there is a sufficient statistical evidence to indicate that the variance is changing relative to the magnitude of lnsalary, which means we need to take further procedure to fix the issues.

As the results from several model testing, we can confirm there is some heteroskedascity issues with our residual and fitted value, as well as the residual normality.

### Fixing Heteroskedasticity by Using WLS

Since the residual plot show that the error look like uneven distribution, it violates the assumption of homogeneity of variance. As the result, it has heterodasticity issue, so we solve this violation by using WLS.

refit <- lm(abs(residuals(dummy\_model)) ~ fitted(dummy\_model))  
refit

##   
## Call:  
## lm(formula = abs(residuals(dummy\_model)) ~ fitted(dummy\_model))  
##   
## Coefficients:  
## (Intercept) fitted(dummy\_model)   
## 2.5359 -0.1131

wts <- 1 / fitted(refit)^2

### Final Model

Test: for some

lm\_wls <- lm(lnsalary ~ ., data = newdummy, weights = wts)  
tidy(lm\_wls)

## # A tibble: 13 × 5  
## term estimate std.error statistic p.value  
## <chr> <dbl> <dbl> <dbl> <dbl>  
## 1 (Intercept) 12.5 0.442 28.3 4.65e-98  
## 2 `fg%` 5.73 1.60 3.58 3.89e- 4  
## 3 `3p` 0.00743 0.00102 7.30 1.55e-12  
## 4 `2p` 0.00168 0.000588 2.86 4.52e- 3  
## 5 `2p%` -4.65 1.29 -3.62 3.38e- 4  
## 6 trb 0.00122 0.000445 2.74 6.44e- 3  
## 7 pos\_C 0.198 0.219 0.906 3.66e- 1  
## 8 pos\_PF 0.412 0.170 2.42 1.61e- 2  
## 9 pos\_PG 0.241 0.154 1.57 1.18e- 1  
## 10 pos\_SF 0.424 0.163 2.60 9.77e- 3  
## 11 pos\_SG NA NA NA NA   
## 12 trans\_team\_none 1.03 0.266 3.88 1.21e- 4  
## 13 trans NA NA NA NA

glance(lm\_wls)

## # A tibble: 1 × 12  
## r.squared adj.r.squared sigma statistic p.value df logLik AIC BIC  
## <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 0.450 0.437 1.24 33.3 4.20e-47 10 -606. 1236. 1284.  
## # … with 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>

final\_model\_a <- lb\_anovat\_lm(lm\_wls)  
final\_model\_a

## Source Df SS MS F P  
## 1 Regression 10 509.6673 50.966732 33.3337 4.198082e-47  
## 2 Error 407 622.2970 1.528985 NA NA  
## 3 Total 417 1131.9643 2.714543 NA NA

Test Statistic: \* \* p-value < 0.00001

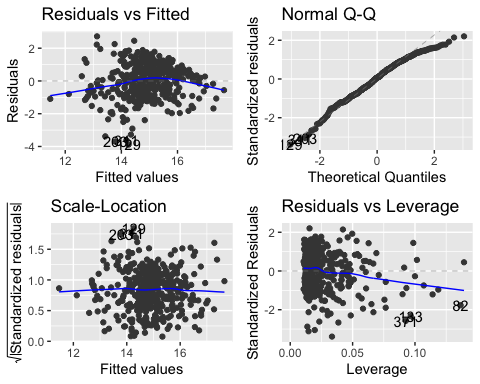
* Reduce model: = 12.496 + 5.729 + 0.007 + 0.002 -4.648 + 0.001 + 0.198 + 0.412 + 0.241 + 0.424 + 1.033

For the final model = 0.436743.

Conclusion. Given the test statistic F = 33.334 and its corresponding p-value < 0.00001. If none of the predictor variables were useful in explaining the variation we see in log salary, it would be almost impossible to observe a test statistic of our magnitude or greater.

Consequently, we will reject and conclude that there is overwhelming statistical evidence to indicate that at least one the predictor variables is useful in explaining the variation in log salary.

# Assumtion  
autoplot(lm\_wls)



# Residual normality test  
shapiro.test(lm\_wls$residuals)

##   
## Shapiro-Wilk normality test  
##   
## data: lm\_wls$residuals  
## W = 0.98346, p-value = 0.0001044

# Residual independence test  
durbinWatsonTest(lm\_wls)

## lag Autocorrelation D-W Statistic p-value  
## 1 0.09488015 1.80693 0.032  
## Alternative hypothesis: rho != 0

# Residual variance homogeneity test  
ncvTest(lm\_wls)

## Non-constant Variance Score Test   
## Variance formula: ~ fitted.values   
## Chisquare = 0.8191605, Df = 1, p = 0.36543

The model has followed all assumptions except residual normality test.

### Evaluate Forecast Model

test <- read\_csv("player17\_18.csv")

## Rows: 605 Columns: 31  
## ── Column specification ────────────────────────────────────────────────────────  
## Delimiter: ","  
## chr (5): Player, player\_id, trans\_team, Pos, Tm  
## dbl (26): Age, G, GS, MP, FG, FGA, FG%, 3P, 3PA, 3P%, 2P, 2PA, 2P%, eFG%, FT...  
##   
## ℹ Use `spec()` to retrieve the full column specification for this data.  
## ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

names(test)[1:31] <- tolower(names(test)[1:31])  
  
test\_result <- dummy\_cols(.data = test,   
 select\_columns = c("pos","trans\_team"), remove\_selected\_columns = TRUE)  
rename(.data = test\_result, trans = trans\_team\_trans) -> test\_dummy  
  
# final reduce model  
full\_predict <- predict(lm\_wls, newdata = test\_dummy, interval = "confidence", level = 0.95)

## Warning in predict.lm(lm\_wls, newdata = test\_dummy, interval = "confidence", :  
## prediction from a rank-deficient fit may be misleading

full\_predict <- cbind(test\_dummy, full\_predict)  
  
full\_predict1 <- full\_predict %>%   
 left\_join(NBA, by = c("player\_id" = "player\_id", "tm" = "tm")) %>%   
 select(fit, `17\_18salary`)  
   
diff <- log(full\_predict1$`17\_18salary`)-full\_predict1$fit  
  
MAD <- mean(abs(diff),na.rm = TRUE)  
MSE <- mean(diff^2,na.rm = TRUE)

We use a forecasting model to determine how well it does in producing accurate forecasts, not how well it fits the historical model. Measuring forecast accuracy, MAD=0.864, MSE=1.124.

From the result, both MAD and MSE are small and close to 0, actual values are very close to the predicted values. It means that the prediction model we done is working well.

# 4. Inferences Based on the Model

After we build the multiple regression model, we can predict NBA players’ salaries. The model has followed all assumptions except residual normality test.

Furthermore, the difference between predicted and actual salaries is small, which means that our model is great for applying.

# 5. Further Directions

Since the data only offer the data that indicate players trans team during the regular season, isn’t include off the season data that most of the palyers trans team time. For the further study, we recommend that add the resign the contrast or not, because the longer time interval model contrast effect maybe improve the results.

# 6. Group Work

Project Concept Contribution: Jack

Data collection: Adela

Data cleaning: Yuka

Model Building Process: Yuka, Adela

Analysis result: Yuka, Adela, Jack

PPT: Jack