Final_Project_Clean

Yuka, Adela, Jack

2022-04-29

```
NBA <- read_csv("NBA.csv")
```

Loading Data

```
## 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.
```

```
original_data <- NBA %>%
  select(-rk, -player, -player_id, -`17_18salary`, -name, -tm, -`fg`, -`fga`, -`3pa`, -`2pa`, -ft, -fta

glimpse(original_data)
```

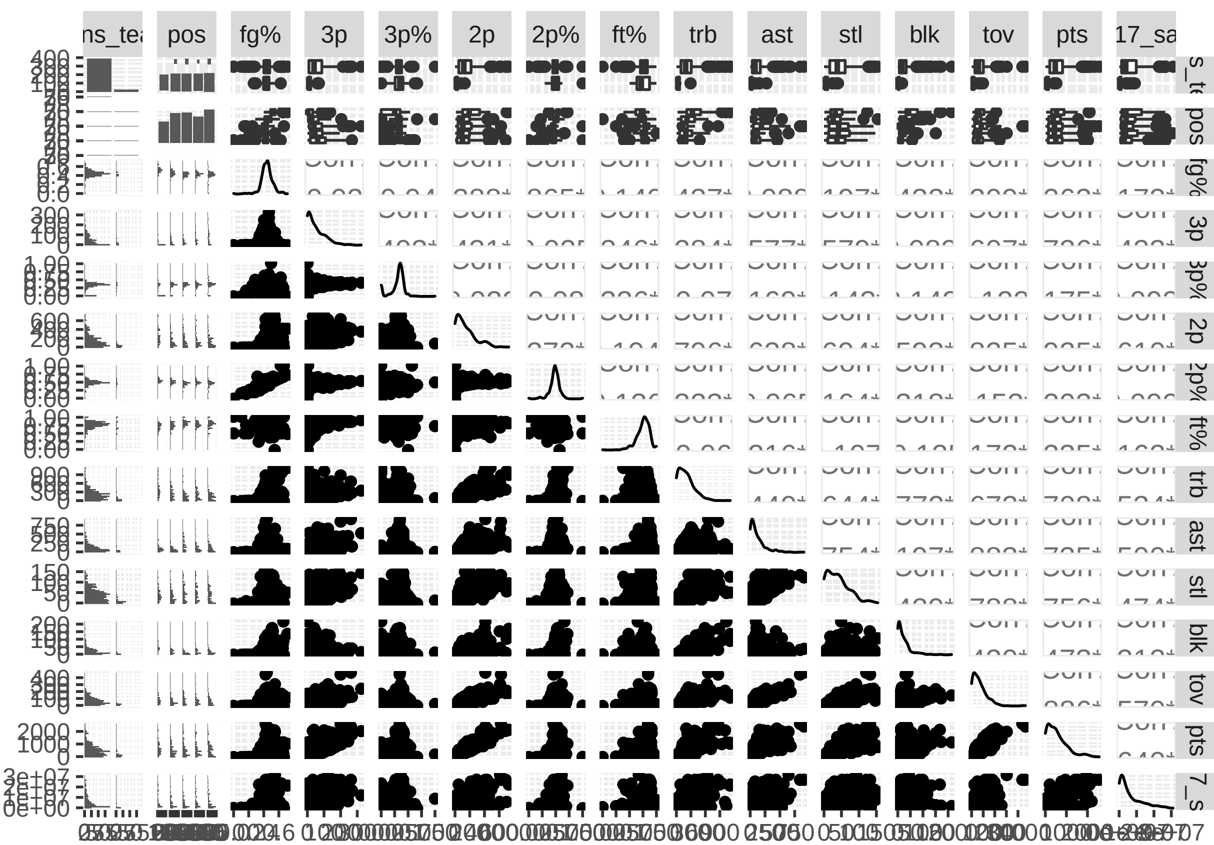
Original Model - lnsalary with 15 variables

```
## Rows: 418
## Columns: 15
## $ trans_team     <chr> "none", "trans", "trans", "none", "none", "none", "none~
## $ pos            <chr> "SG", "PF", "PF", "C", "SG", "C", "PF", "PF", "SG", "PF~
## $ `fg%`          <dbl> 0.393, 0.294, 0.425, 0.571, 0.440, 0.500, 0.477, 0.458,~
## $ `3p`           <dbl> 94, 1, 36, 0, 62, 0, 23, 0, 15, 70, 14, 21, 15, 204, 49~
## $ `3p%`          <dbl> 0.381, 0.143, 0.434, 0.000, 0.411, 0.000, 0.411, 0.000,~
## $ `2p`           <dbl> 40, 4, 29, 374, 123, 89, 477, 77, 259, 113, 16, 54, 78,~
## $ `2p%`          <dbl> 0.426, 0.400, 0.414, 0.572, 0.457, 0.511, 0.480, 0.461,~
## $ `ft%`          <dbl> 0.898, 0.667, 0.754, 0.611, 0.892, 0.725, 0.812, 0.697,~
## $ trb            <dbl> 86, 8, 107, 613, 125, 177, 523, 219, 391, 451, 24, 96, ~
## $ ast            <dbl> 40, 0, 18, 86, 78, 12, 139, 57, 98, 99, 11, 34, 91, 68,~
## $ stl            <dbl> 37, 0, 14, 89, 21, 20, 46, 18, 115, 60, 3, 13, 51, 31, ~
## $ blk            <dbl> 8, 0, 15, 78, 6, 22, 88, 24, 29, 44, 0, 7, 26, 14, 151,~
## $ tov            <dbl> 33, 2, 19, 146, 42, 31, 98, 29, 100, 94, 7, 26, 39, 55,~
## $ pts            <dbl> 406, 13, 209, 905, 515, 207, 1243, 177, 643, 532, 86, 2~
## $ `16_17_salary` <dbl> 5994764, 1050961, 1914544, 3140517, 12500000, 4600000, ~
```

Corralation Analysis

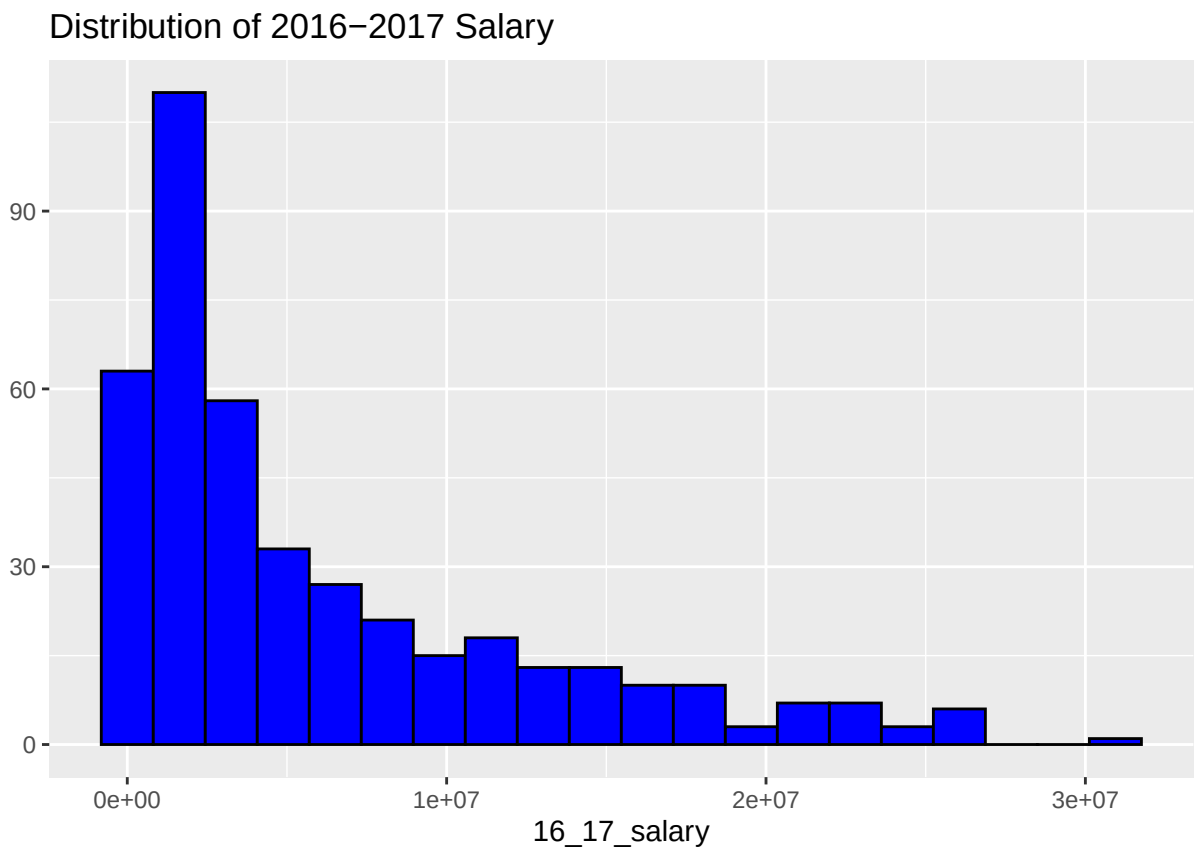
```
original_data %>%
    ggpairs()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

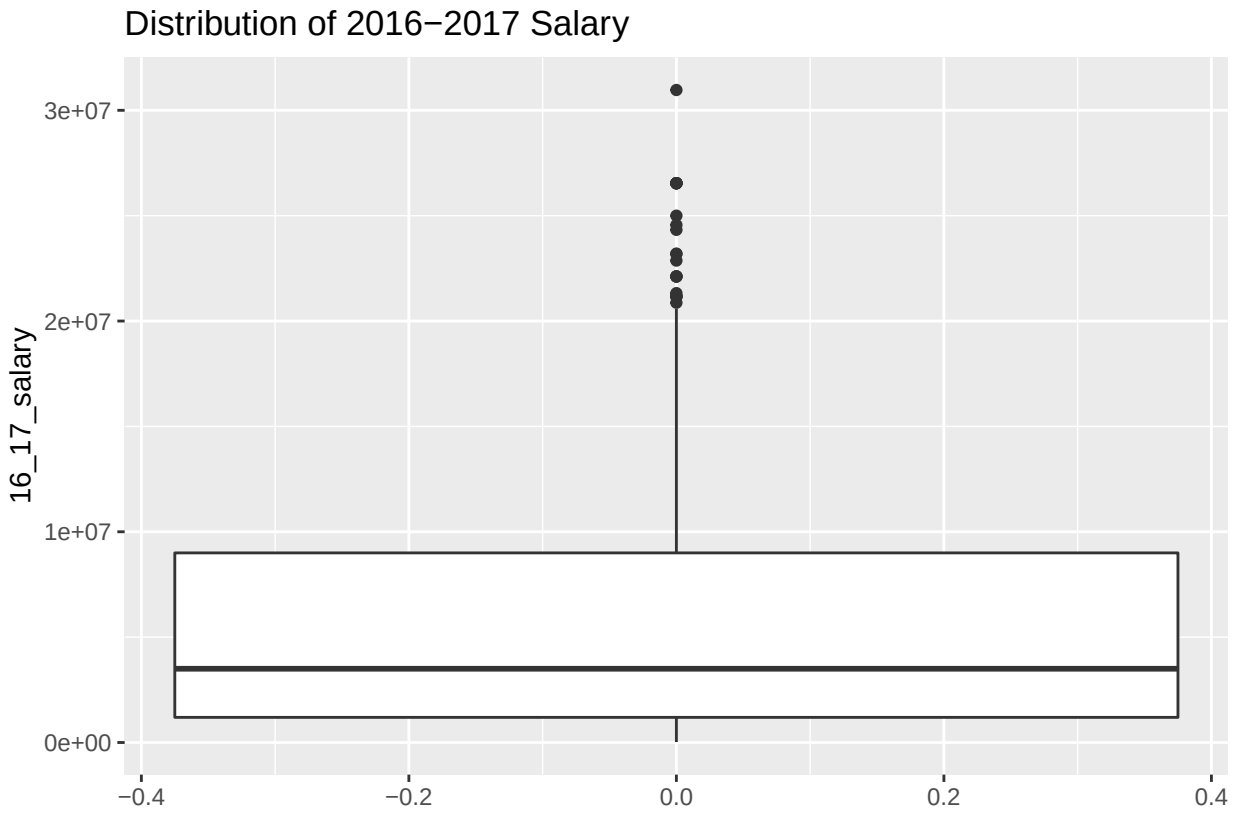


Checking Y Variable

```
qplot(data = original_data, x = `16_17_salary`, geom = "histogram", main = "Distribution of 2016-2017 Sa
```



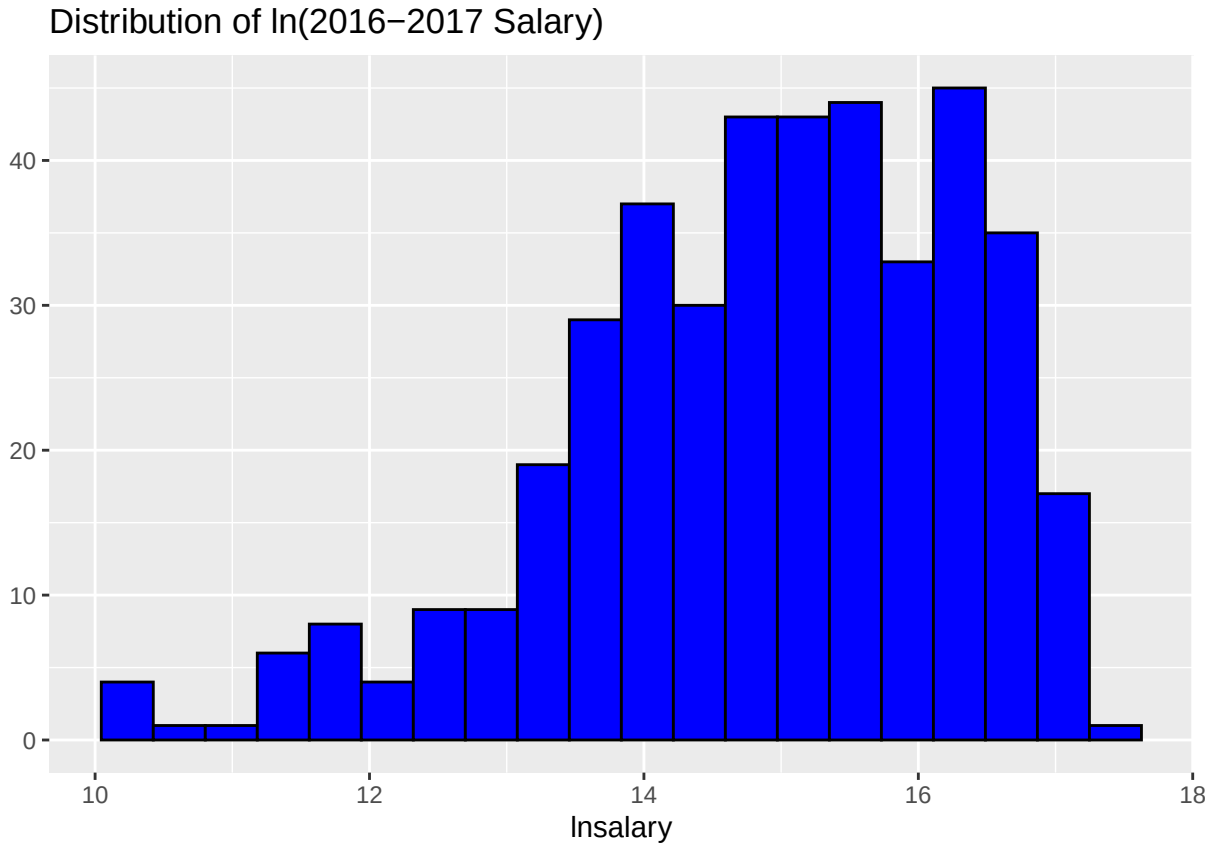

```
qplot(data = original_data, y = `16_17_salary`, geom = "boxplot", main = "Distribution of 2016-2017 Sala
```



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("
```



Original Model

```
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 x 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: $\widehat{salary} = 13.02 - 1.023 \cdot trans.team + 5.82 \cdot fg\% + 0.0094 \cdot 3p + 0.25 \cdot 3p\% + 0.0034 \cdot 2p - 4.18 \cdot 2p\% + 0.394 \cdot ft\% + 0.0023 \cdot trb + 0.0015 \cdot ast + 0.00037 \cdot stl - 0.0043 \cdot blk + -0.0025 \cdot tov - 0.00078 \cdot pts$

```
g_mult_og <- broom::glance(og_model)
g_mult_og
```

```
## # A tibble: 1 x 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
```

```
## shows our original model isn't good enought to be our final model
```

Reduce Model based on All Subset Models Method

```
reduce <- NBA %>%
  mutate(lnsalary = log(`16_17_salary`)) %>%
  select(lnsalary,`fg%`,`3p`,`3p%`,`2p%`,trb,blk, pos, trans_team)
```

```
reducemodel <- lm(lnsalary ~`fg%` + `3p` + `3p%`+ `2p%`+ trb + blk, data = new_og_data)

reduce_model_t <- tidy(reducemodel)
reduce_model_t
```

```
## # A tibble: 7 x 5
##   term        estimate std.error statistic   p.value
##   <chr>          <dbl>     <dbl>     <dbl>     <dbl>
## 1 (Intercept) 13.3      0.340      39.1    2.00e-140
## 2 `fg%`        7.21     1.47        4.89   1.44e-  6
## 3 `3p`         0.00935  0.00121     7.72   8.77e- 14
## 4 `3p%`        0.0142   0.469       0.0304 9.76e-  1
## 5 `2p%`       -5.48     1.18       -4.66   4.28e-  6
## 6 trb          0.00322  0.000495    6.49   2.41e- 10
## 7 blk         -0.00469  0.00309    -1.52   1.30e-  1
```

```
reduce_model_g <- broom::glance(reducemodel)
reduce_model_g
```

```
## # A tibble: 1 x 12
##   r.squared adj.r.squared sigma statistic  p.value    df logLik   AIC   BIC
##       <dbl>         <dbl> <dbl>     <dbl>    <dbl> <dbl>  <dbl> <dbl> <dbl>
```

```
## 1     0.411         0.402  1.11      47.8 2.05e-44     6  -633. 1282. 1314.
## # ... with 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>
```

```
reduce_model_a <- lb_anovat_lm(reducemodel, reg_collapse = FALSE)
reduce_model_a
```

```
##   Source  Df         SS         MS          F            P
## 1  `fg%`   1  45.711535  45.711535  37.150549 2.520088e-09
## 2   `3p`   1 181.483430 181.483430 147.494698 3.258910e-29
## 3  `3p%`   1   1.349973   1.349973   1.097146 2.955089e-01
## 4  `2p%`   1  49.400315  49.400315  40.148484 6.190796e-10
## 5    trb   1  72.212975  72.212975  58.688724 1.339115e-13
## 6    blk   1   2.833997   2.833997   2.303238 1.298734e-01
## 7  Error 411 505.710989   1.230440         NA           NA
## 8  Total 417 858.703214   2.059240         NA           NA
```

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

```
## # A tibble: 7 x 7
##   term        estimate std.error statistic   p.value conf.low conf.high
##   <chr>          <dbl>     <dbl>     <dbl>     <dbl>    <dbl>     <dbl>
## 1 (Intercept) 13.3      0.340      39.1    2.00e-140 12.5      14.1
## 2 `fg%`        7.21     1.47        4.89   1.44e-  6  3.77     10.6
## 3 `3p`         0.00935  0.00121     7.72   8.77e- 14  0.00652   0.0122
## 4 `3p%`        0.0142   0.469       0.0304 9.76e-  1 -1.08      1.11
## 5 `2p%`       -5.48     1.18       -4.66   4.28e-  6 -8.22     -2.73
## 6 trb          0.00322  0.000495    6.49   2.41e- 10  0.00206   0.00437
## 7 blk         -0.00469  0.00309    -1.52   1.30e-  1 -0.0119    0.00253
```

```
vif(reducemodel)
```

```
##    `fg%`     `3p`    `3p%`    `2p%`      trb      blk
## 4.934932 1.517883 1.306680 4.273724 3.054989 2.659630
```

Plotting Reduce Model

Adding Dummy Variables

```
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 x 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
```

```
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: 14 x 5
##    term            estimate std.error statistic    p.value
##    <chr>              <dbl>     <dbl>     <dbl>      <dbl>
##  1 (Intercept)     12.0      0.413       29.1    1.98e-101
##  2 `fg%`            7.09     1.53         4.64   4.79e-  6
##  3 `3p`             0.00918  0.00127      7.25   2.12e- 12
##  4 `3p%`            0.167    0.476        0.351  7.26e-  1
##  5 `2p%`           -5.21     1.20        -4.35   1.71e-  5
##  6 trb              0.00280  0.000490     5.72   2.12e-  8
##  7 blk             -0.00482  0.00312     -1.54   1.23e-  1
##  8 pos_C            0.278    0.235        1.18   2.37e-  1
##  9 pos_PF           0.381    0.172        2.21   2.74e-  2
## 10 pos_PG           0.344    0.158        2.19   2.93e-  2
## 11 pos_SF           0.496    0.169        2.93   3.58e-  3
## 12 pos_SG          NA       NA           NA     NA
## 13 trans_team_none  1.00     0.228        4.40   1.41e-  5
## 14 trans           NA       NA           NA     NA
```

```
dumglout
```

```
## # A tibble: 1 x 12
##   r.squared adj.r.squared sigma statistic  p.value    df logLik   AIC   BIC
##       <dbl>         <dbl> <dbl>     <dbl>    <dbl> <dbl>  <dbl> <dbl> <dbl>
## 1     0.453         0.439  1.08      30.6 7.96e-47    11  -617. 1261. 1313.
## # ... with 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>
```

```
dumglout #dummy models
```

Comparing models

```
## # A tibble: 1 x 12
##   r.squared adj.r.squared sigma statistic  p.value    df logLik   AIC   BIC
##       <dbl>         <dbl> <dbl>     <dbl>    <dbl> <dbl>  <dbl> <dbl> <dbl>
## 1     0.453         0.439  1.08      30.6 7.96e-47    11  -617. 1261. 1313.
## # ... with 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>
```

```
g_mult_og #full model wihtout dummy
```

```
## # A tibble: 1 x 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>
```

Evaluate Forecase Model

```
test <- read_csv("player17_18.csv")
```

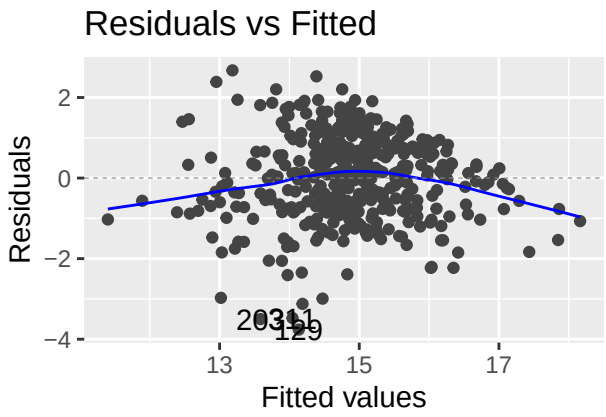
```
## Rows: 605 Columns: 31
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (5): Player, player_id, trans_team, Pos, Tm
## dbl (26): Age, G, GS, MP, FG, FGA, FG%, 3P, 3PA, 3P%, 2P, 2PA, 2P%, eFG%, FT...
##
## i Use `spec()` to retrieve the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

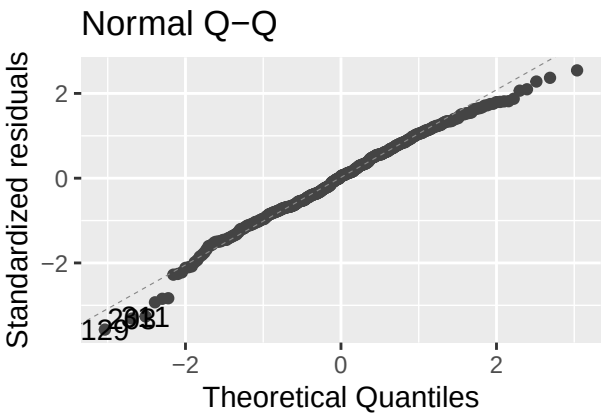
```
names(test)[1:31] <- tolower(names(test)[1:31])

test_result <- dummy_cols(.data = test, select_columns = c("pos","trans_team"), remove_selected_columns
rename(.data = test_result, trans = trans_team_trans) -> test_dummy

# final reduce model
full_predict <- predict(dummy_model, newdata = test_dummy, interval = "confidence", level = 0.95)
```

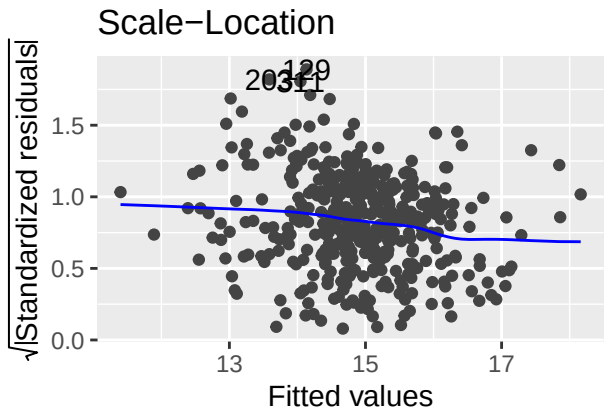
```
## Warning in predict.lm(dummy_model, newdata = test_dummy, interval =
## "confidence", : prediction from a rank-deficient fit may be misleading
```

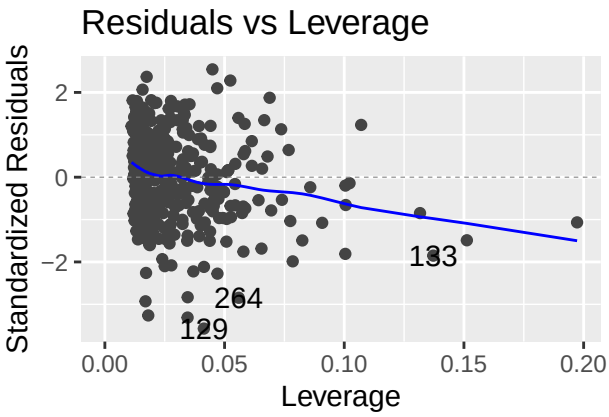
```
full_predict <- cbind(test_dummy, full_predict)

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)
```

```
# Assumtion
autoplot(dummy_model)
```









Model Testing

```
# Residual normality test
shapiro.test(dummy_model$residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  dummy_model$residuals
## W = 0.98952, p-value = 0.0044
```

```
# Residual independence test
durbinWatsonTest(dummy_model)
```

```
##  lag Autocorrelation D-W Statistic p-value
##    1      0.09668787       1.80392   0.042
##  Alternative hypothesis: rho != 0
```

```
# Residual variance homogeneity test
ncvTest(dummy_model)
```

```
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 16.58006, Df = 1, p = 4.6639e-05
```

###Fixing Heteroskedasticity 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)
##              2.5736              -0.1155
```

```
wts <- 1 / fitted(refit)^2
wts
```

```
##         1         2         3         4         5         6         7         8
## 1.3139321 0.7784160 1.0669035 1.7415935 1.3145586 1.2649231 1.6867334 1.3236409
##         9        10        11        12        13        14        15        16
## 1.3926923 1.6943956 1.1259437 1.1369445 1.3883922 2.5416157 1.9106950 2.3891206
##        17        18        19        20        21        22        23        24
## 2.2657706 1.4479385 1.4765432 1.7206249 1.0568075 1.1917826 1.3086838 1.8842766
##        25        26        27        28        29        30        31        32
## 1.1818393 0.9117170 1.4567945 1.3559995 1.9197232 1.1474760 1.5844776 1.0047336
##        33        34        35        36        37        38        39        40
## 1.4104334 1.4074747 1.1185804 1.0650642 1.0300057 1.2286621 1.7057966 1.3416980
##        41        42        43        44        45        46        47        48
## 1.4579988 1.7447894 1.7355692 1.8846292 1.6937650 1.1598112 1.4511626 1.2966590
##        49        50        51        52        53        54        55        56
## 1.2154760 1.4592878 0.9919133 1.2512884 1.1526795 2.1151108 0.9059705 0.7918780
##        57        58        59        60        61        62        63        64
## 1.7091294 0.9756593 1.6522444 1.7293278 1.1996887 1.0654812 1.4273550 2.0685052
##        65        66        67        68        69        70        71        72
## 1.5117123 1.1087807 1.2416718 1.5587992 1.0726732 1.5358259 1.2362786 2.0750991
##        73        74        75        76        77        78        79        80
## 1.2012403 1.3669209 1.9230787 1.6989952 1.4420860 1.1087368 2.2879854 1.4921915
##        81        82        83        84        85        86        87        88
## 1.8028371 3.8151365 1.6426611 2.0610191 1.4086580 1.2117462 1.5098700 1.5161729
##        89        90        91        92        93        94        95        96
## 1.4897048 1.2481009 1.8256875 1.3373414 1.0236113 1.9034234 2.7473320 1.4307470
##        97        98        99       100       101       102       103       104
## 1.3443138 1.2307202 1.9941132 0.7664052 1.0603413 1.4583350 1.2528227 1.3188107
##       105       106       107       108       109       110       111       112
## 1.5075980 1.1908640 1.3238050 1.2270897 1.6315322 1.4322175 1.3913883 0.8259454
##       113       114       115       116       117       118       119       120
## 1.0881764 1.2768424 1.0262945 1.5774401 1.2345455 1.3340941 1.6066782 1.7170903
##       121       122       123       124       125       126       127       128
## 2.0264625 1.3101201 1.8449022 1.8798215 1.4503341 0.8478217 1.0083204 3.0100842
##       129       130       131       132       133       134       135       136
## 1.1294951 1.1355101 1.3233132 1.4241260 2.1799690 0.9130305 0.8467721 1.7269897
##       137       138       139       140       141       142       143       144
## 2.0001748 1.2557549 1.2929776 1.3072973 1.4628801 1.7655692 1.2816387 1.9158607
##       145       146       147       148       149       150       151       152
## 1.3718867 1.8533756 1.2484788 1.2054154 1.6033851 3.8391738 1.5901304 1.3847470
##       153       154       155       156       157       158       159       160
## 1.2821151 1.4937305 1.3096743 0.9233109 1.9379409 0.8775385 1.2412200 1.0748547
##       161       162       163       164       165       166       167       168
## 1.0795338 2.3835416 1.3130832 1.3126169 1.4091470 1.6655368 1.3224118 1.2679402
##       169       170       171       172       173       174       175       176
## 1.4615582 1.5735393 1.6997649 1.6331641 1.3612466 1.5931849 1.4227624 1.5180070
##       177       178       179       180       181       182       183       184
## 1.7158989 2.3900224 1.2112691 1.3367983 1.3444203 1.4915133 1.3040325 2.0099937
```

```
##       185       186       187       188       189       190       191       192
## 1.5539091 2.1097463 0.8511981 1.0865854 1.4161030 2.6955238 1.3928604 1.3442711
##       193       194       195       196       197       198       199       200
## 1.1558241 1.4124318 1.3845271 1.5608485 1.6057130 1.8349325 1.3510189 1.6667469
##       201       202       203       204       205       206       207       208
## 1.0898318 2.1315328 0.9916642 1.3147136 1.3765563 1.2093115 2.8067907 1.4915312
##       209       210       211       212       213       214       215       216
## 1.7131758 1.2107161 1.6958721 1.5437199 1.2108876 1.4622620 1.5115789 1.3105043
##       217       218       219       220       221       222       223       224
## 1.3656159 1.2551600 1.0848151 1.1508483 1.4823525 1.3439476 1.1011079 1.5743718
##       225       226       227       228       229       230       231       232
## 1.5117738 2.2930216 1.4716386 1.6188164 1.3490651 0.9657107 2.6110182 1.3708939
##       233       234       235       236       237       238       239       240
## 1.3324929 1.2303336 1.2282102 1.7301278 1.4638773 2.8412396 2.2574214 1.3333960
##       241       242       243       244       245       246       247       248
## 1.4564530 1.2733210 1.2072132 1.2910230 1.8291477 1.4139862 1.0839182 2.0109467
##       249       250       251       252       253       254       255       256
## 1.3651468 1.1079094 1.1937122 1.3633927 1.3300821 1.2928899 1.0880282 1.2636703
##       257       258       259       260       261       262       263       264
## 1.2714755 1.4597522 1.1578137 1.4629533 2.0904518 1.3433057 1.7991030 0.8739172
##       265       266       267       268       269       270       271       272
## 1.8698304 1.7225018 1.6814928 1.3051219 1.9665814 1.0970945 1.1144090 1.3605642
##       273       274       275       276       277       278       279       280
## 1.3857782 1.5328273 1.1980446 1.4392345 1.4164242 1.5118953 1.2397609 1.1744380
##       281       282       283       284       285       286       287       288
## 0.8764755 1.6370828 1.1161890 1.0311028 1.4810144 1.3038371 1.7270943 1.2861188
##       289       290       291       292       293       294       295       296
## 1.1415522 1.0194745 0.9173025 1.4017567 1.6536704 1.5506894 1.5307493 1.2205789
##       297       298       299       300       301       302       303       304
## 1.6708275 1.3154889 1.2031511 0.9395715 1.6324525 1.9450877 1.1316036 1.5687619
##       305       306       307       308       309       310       311       312
## 1.1376532 1.2934071 1.4864010 1.8023448 1.3746788 1.2418569 1.1052357 1.0664510
##       313       314       315       316       317       318       319       320
## 1.3119601 0.8597407 0.8710766 1.9218161 1.8900991 1.9709496 1.4564569 1.2504777
##       321       322       323       324       325       326       327       328
## 0.9785477 1.4503223 1.3564960 1.2179022 1.5290516 1.4105632 1.5468447 1.6567612
##       329       330       331       332       333       334       335       336
## 1.3914675 1.2272738 1.4247622 1.5499907 1.2265364 1.7625301 1.4674323 2.0277715
##       337       338       339       340       341       342       343       344
## 1.6943218 1.2510748 1.4837353 0.6331720 0.6945507 1.3125447 1.2261524 1.3783069
##       345       346       347       348       349       350       351       352
## 1.2782025 1.1651525 0.9906941 1.4101081 1.3599656 1.4005272 1.5926637 1.6853193
##       353       354       355       356       357       358       359       360
## 1.1323749 1.5541882 0.7932698 0.8156051 0.9207423 1.2211528 0.9399587 1.7265214
##       361       362       363       364       365       366       367       368
## 1.4390094 1.3230584 1.3124743 2.4357307 1.4529387 0.8911791 0.7956525 2.4934654
##       369       370       371       372       373       374       375       376
## 1.9099705 1.4983765 3.1920993 1.3698768 1.4155392 1.5776039 1.2215102 1.2285634
##       377       378       379       380       381       382       383       384
## 1.1446226 2.0768002 1.3281595 1.1933085 1.1398135 1.0543388 1.5541932 2.0343405
##       385       386       387       388       389       390       391       392
## 1.1055079 1.3241155 1.3889804 2.7627768 1.5542545 1.1978710 1.3110896 4.4215441
##       393       394       395       396       397       398       399       400
## 1.2028134 1.3036396 1.9291976 1.3893871 1.0448355 0.9676749 0.8617604 0.9246808
```

```
##       401       402       403       404       405       406       407       408
## 1.0920171 2.0095089 1.3502695 0.8854298 0.8891516 1.2015709 1.2295988 1.0159006
##       409       410       411       412       413       414       415       416
## 1.0771011 1.2941295 1.1584424 1.0496292 1.6410670 1.8064696 1.5094747 1.2284924
##       417       418
## 1.2809321 1.2557781
```

Final Model

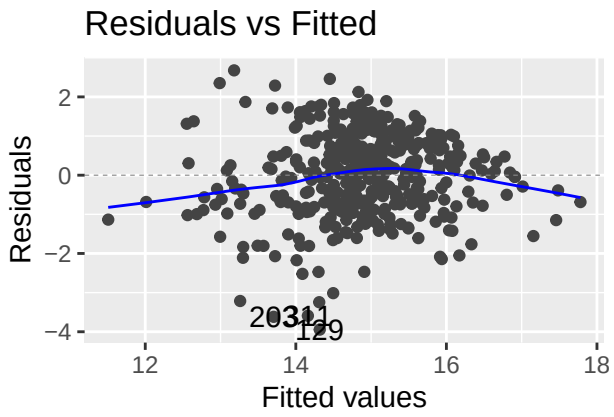
```
lm_wls <- lm(lnsalary ~ ., data = newdummy, weights = wts)

tidy(lm_wls)
```

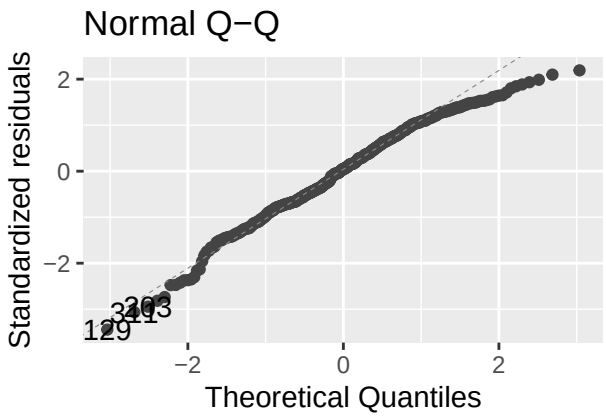
```
## # A tibble: 14 x 5
##    term            estimate std.error statistic   p.value
##    <chr>              <dbl>     <dbl>     <dbl>     <dbl>
##  1 (Intercept)     12.2      0.465       26.2    1.78e-89
##  2 `fg%`            6.67     1.60         4.16   3.90e- 5
##  3 `3p`             0.00796  0.00106      7.48   4.63e-13
##  4 `3p%`            0.176    0.474        0.372  7.10e- 1
##  5 `2p%`           -5.02     1.31        -3.84   1.45e- 4
##  6 trb              0.00249  0.000402     6.18   1.57e- 9
##  7 blk             -0.00374  0.00270     -1.39   1.66e- 1
##  8 pos_C            0.178    0.227        0.785  4.33e- 1
##  9 pos_PF           0.333    0.169        1.97   4.95e- 2
## 10 pos_PG           0.268    0.155        1.73   8.38e- 2
## 11 pos_SF           0.412    0.164        2.51   1.25e- 2
## 12 pos_SG          NA       NA           NA     NA
## 13 trans_team_none  1.08     0.269        4.02   6.86e- 5
## 14 trans           NA       NA           NA     NA
```

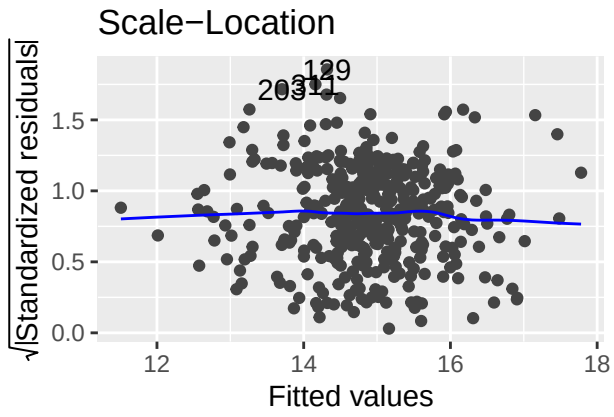
```
glance(lm_wls)
```

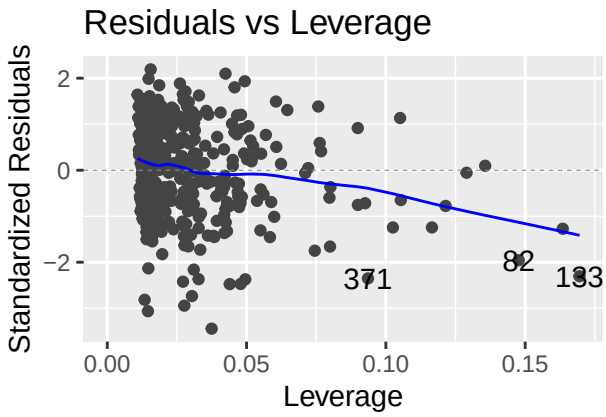
```
## # A tibble: 1 x 12
##   r.squared adj.r.squared sigma statistic  p.value    df logLik   AIC   BIC
##       <dbl>         <dbl> <dbl>     <dbl>    <dbl> <dbl>  <dbl> <dbl> <dbl>
## 1     0.444         0.429  1.24      29.4 2.63e-45    11  -608. 1242. 1294.
## # ... with 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>
```

```
# Assumtion
autoplot(lm_wls)
```










```
# Residual normality test
shapiro.test(lm_wls$residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  lm_wls$residuals
## W = 0.98586, p-value = 0.0004277
```

```
# Residual independence test
durbinWatsonTest(lm_wls)
```

```
##  lag Autocorrelation D-W Statistic p-value
##    1      0.09582103      1.805773    0.05
##  Alternative hypothesis: rho != 0
```

```
# Residual variance homogeneity test
ncvTest(lm_wls)
```

```
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 0.9445787, Df = 1, p = 0.3311
```