Driver

```
<!-- * Make a creative, "original" plot--perhaps use grid? -->
<!-- * Make 3-d plot interactive! -->
<!-- * trellis plots to combine the variables -->
<!--     * Cleveland Caveliers vs. other teams, and other variables  -->
```

```
library(ggplot2)
library(scatterplot3d)
library(glmnet)
```

```
## Loading required package: Matrix
```

```
## Loading required package: foreach
```

```
## Loaded glmnet 2.0-13
```

```
sal_1718 <- read.csv("NBA_season1718_salary.csv")
sal_1819 <- read.csv("NBA_season1819_salary.csv")
player_data <- read.csv("player_data.csv")
Players <- read.csv("Players.csv")
season <- read.csv("Seasons_Stats.csv")
```

Merge the Data

Salary with season, for years

```
# getting rid of index in both datasets
# getting rid of two blank columns in season data set, as well as Year, since it'll all be 2017
sal_season <- merge(sal_1718[,c(-1)], season[season$Year == 2017,-c(1, 2, 22, 27)], by = c("Player", "Tm

write.csv(sal_season, file = "sal_season.csv")
```

Linear Model

Correlation Analysis

```
sal_season_mat <- model.matrix(~ ., sal_season[,c(-1, -3)])[,-1]
# inner negatives: remove name and salary (dependent variable). Outer negative: remove intercept

sal <- sal_season$season17_18[rowSums(is.na(sal_season)) == 0]

R <- cor(sal_season_mat, method = "spearman")
```

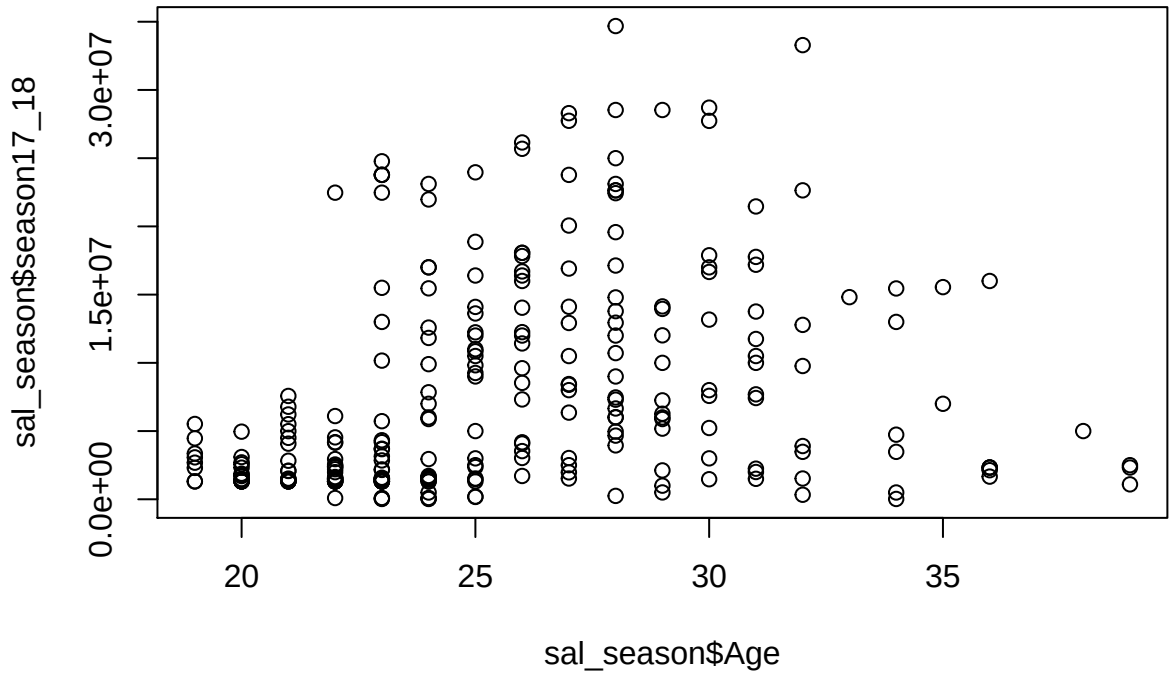
```
## Warning in cor(sal_season_mat, method = "spearman"): the standard deviation
## is zero
```

Too big of a correlation matrix to display.

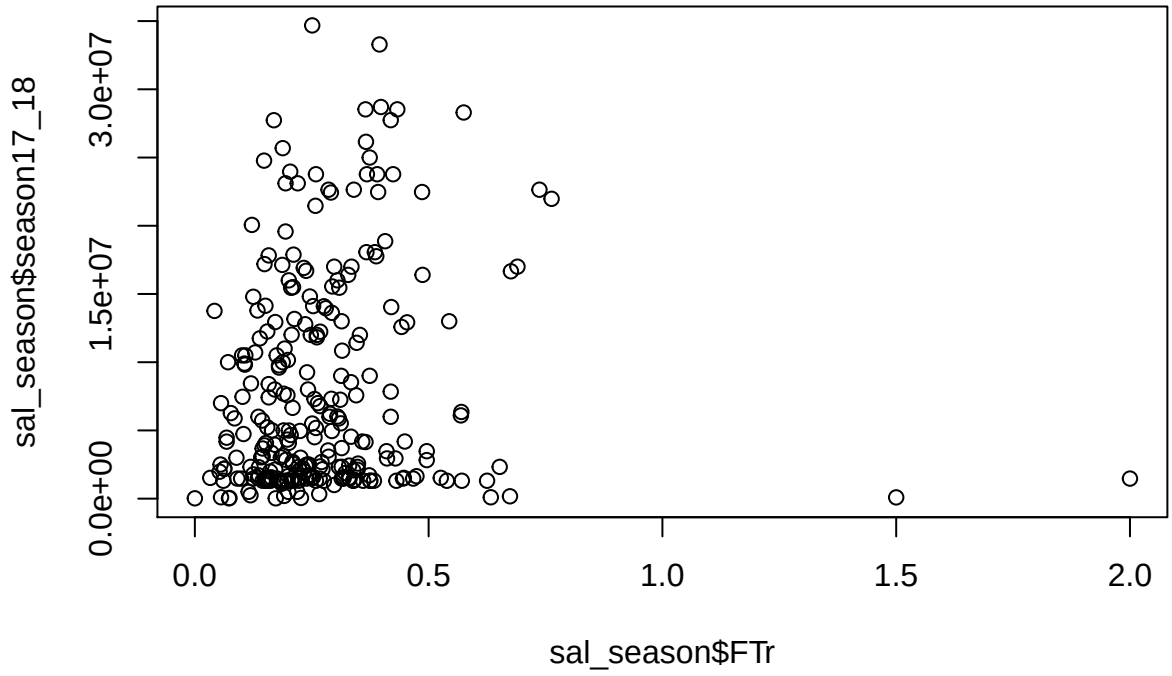
Graphics

Scatterplots between salary and arbitrarily chosen predictors

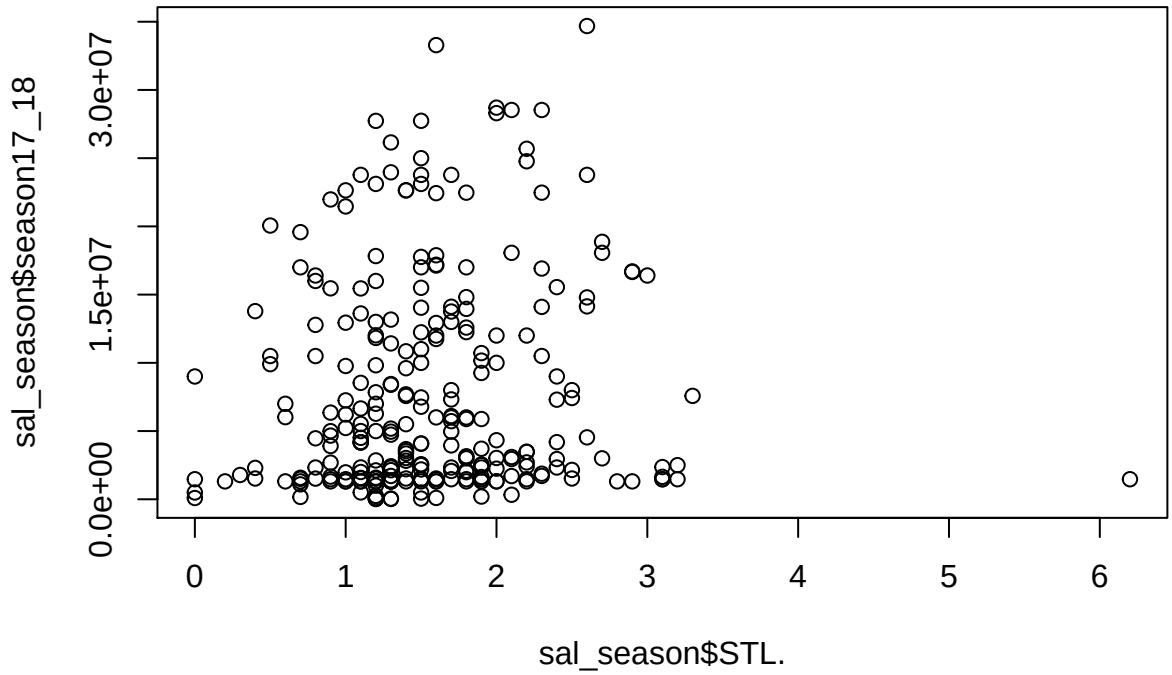
```
plot(sal_season$Age, sal_season$season17_18)
```



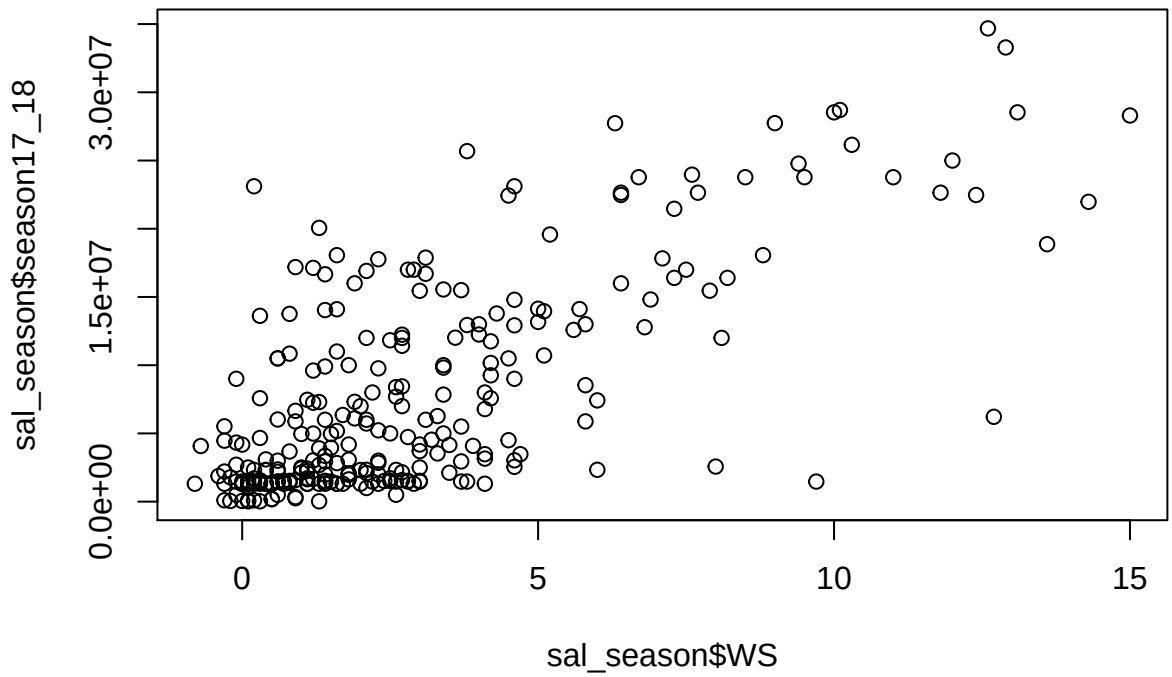

```
plot(sal_season$FTr, sal_season$season17_18)
```



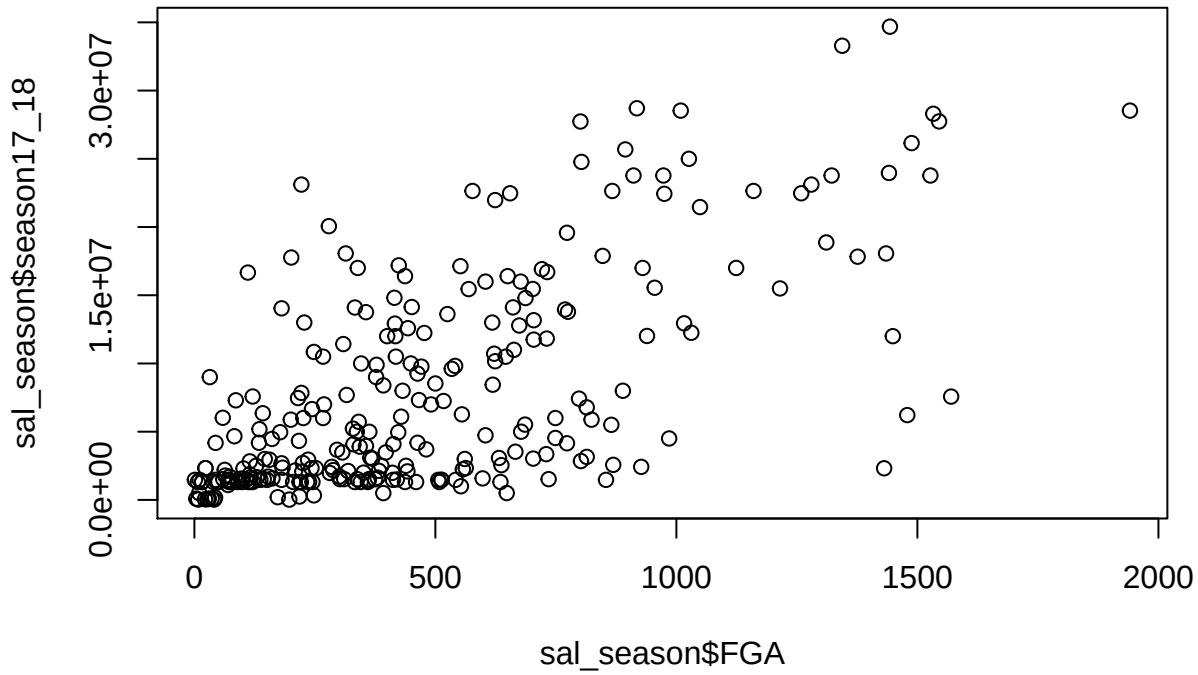
```
plot(sal_season$STL., sal_season$season17_18)
```



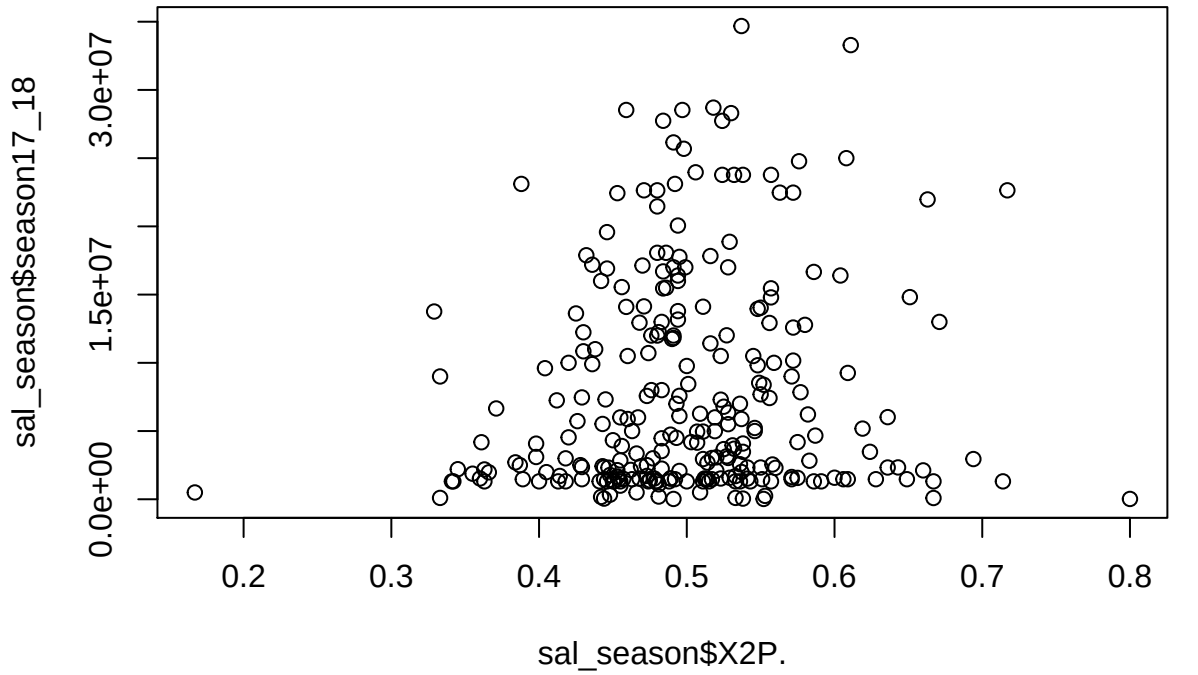

```
plot(sal_season$WS, sal_season$season17_18)
```



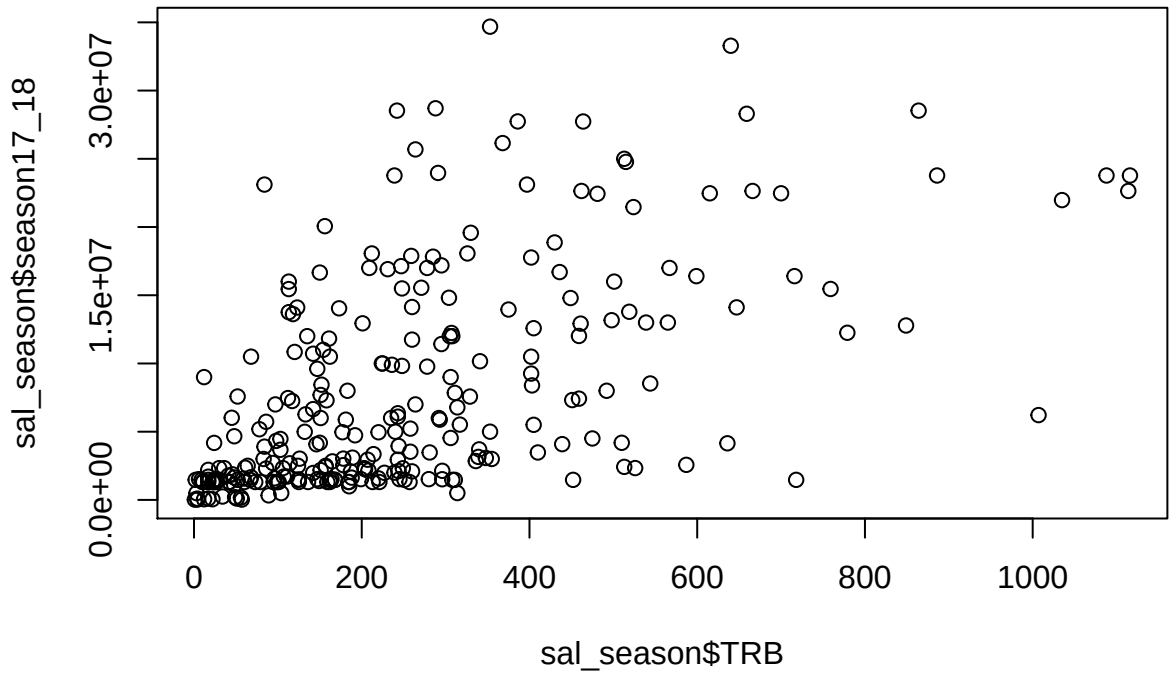

```
plot(sal_season$FGA, sal_season$season17_18)
```



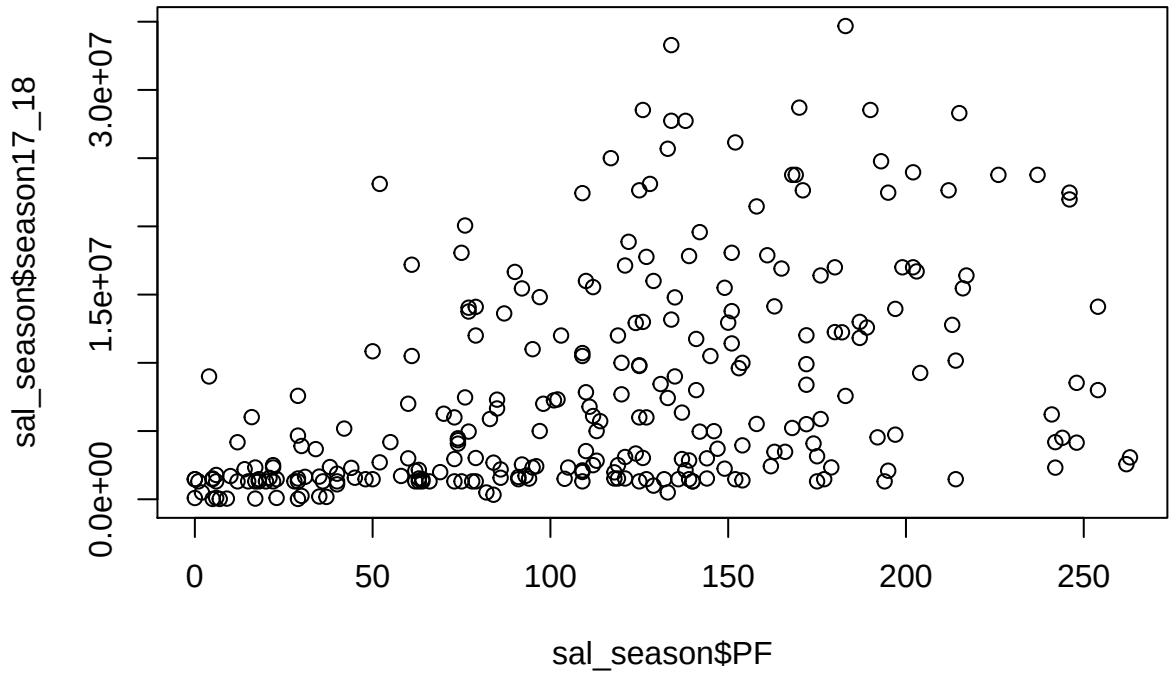

```
plot(sal_season$X2P., sal_season$season17_18)
```




```
plot(sal_season$TRB, sal_season$season17_18)
```

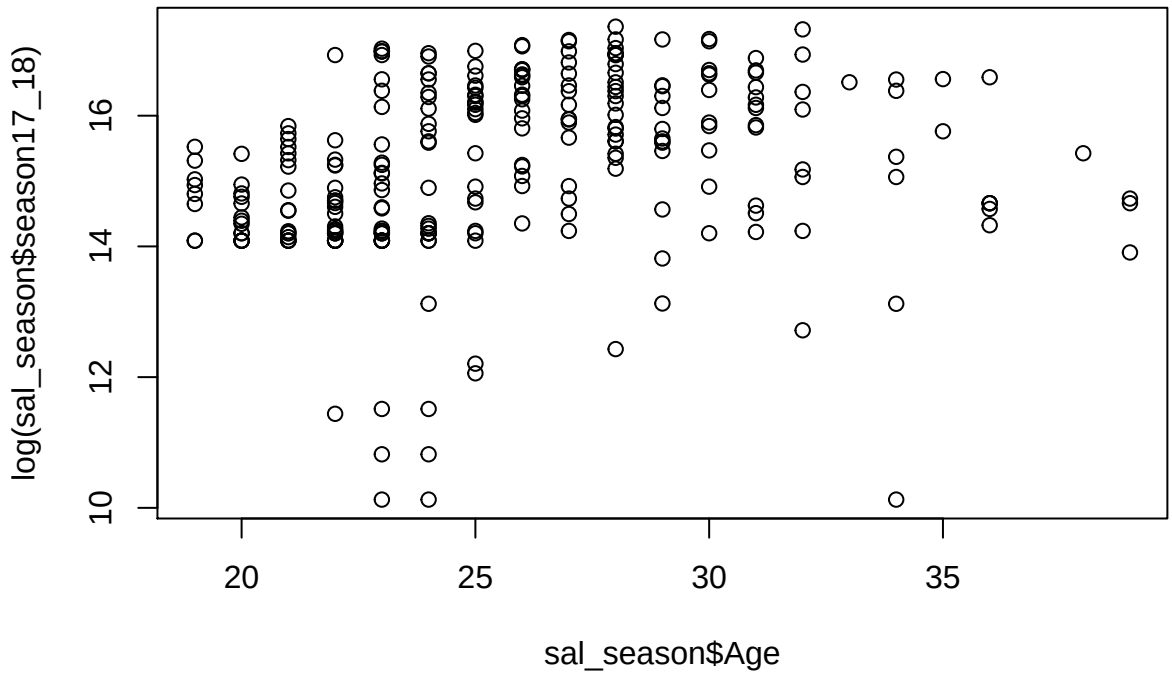



```
plot(sal_season$PF, sal_season$season17_18)
```

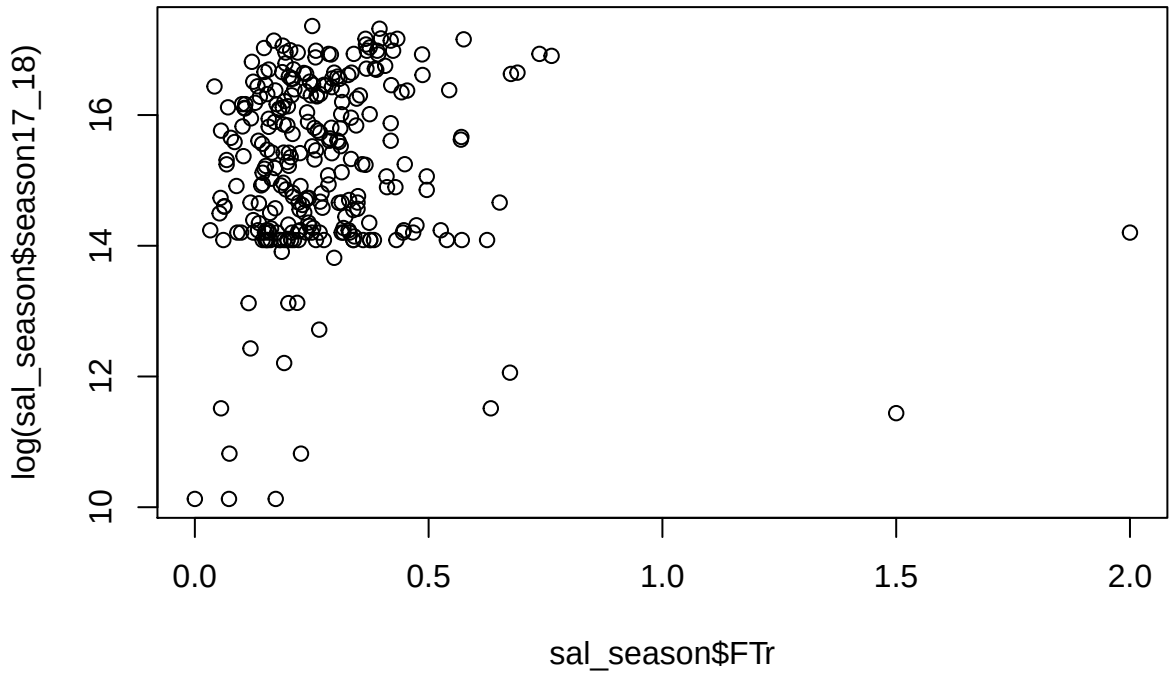


log-transforming salary

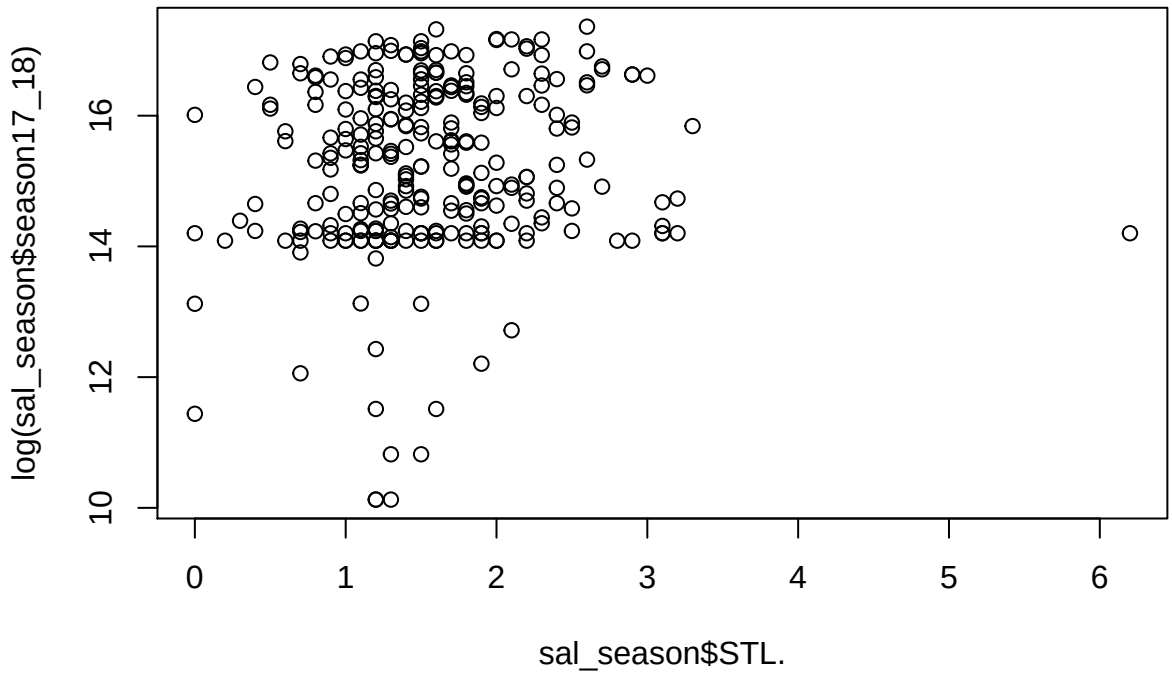
```
plot(sal_season$Age, log(sal_season$season17_18))
```



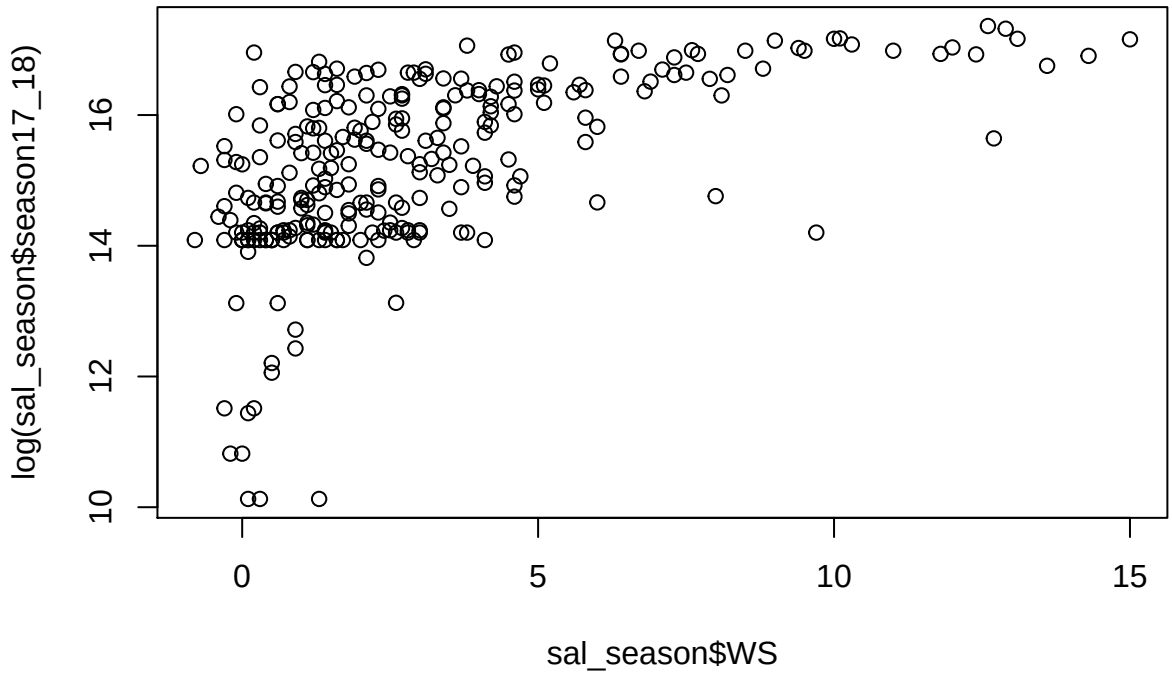

```
plot(sal_season$FTr, log(sal_season$season17_18))
```



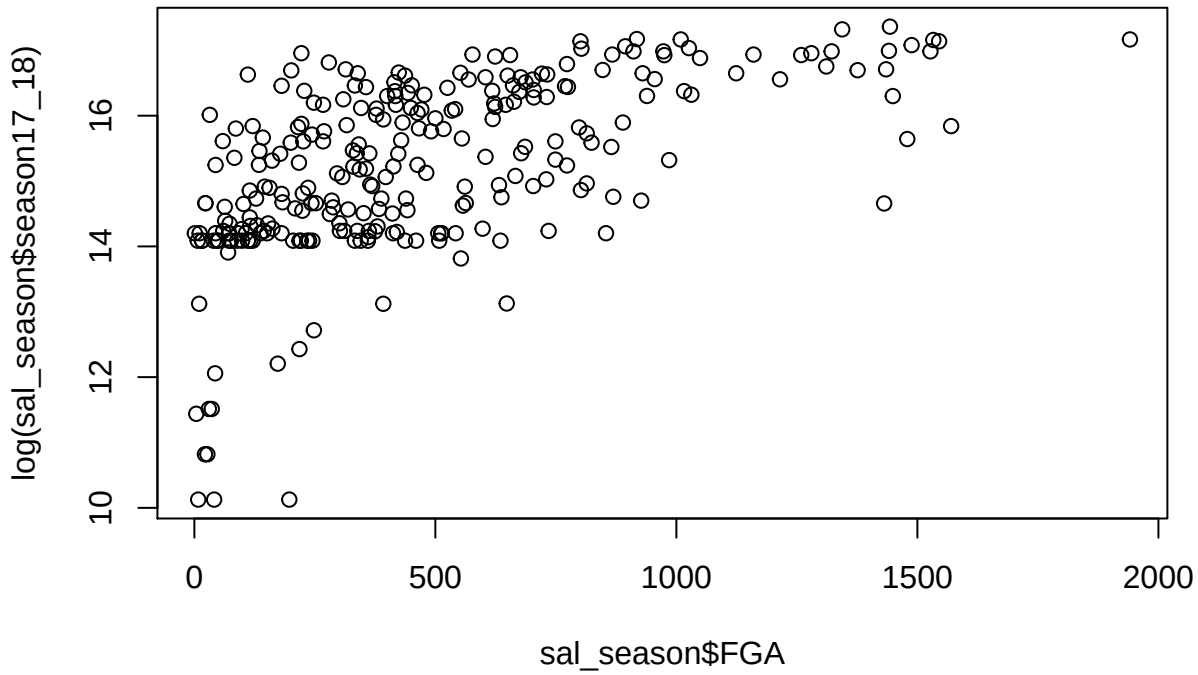

```
plot(sal_season$STL., log(sal_season$season17_18))
```



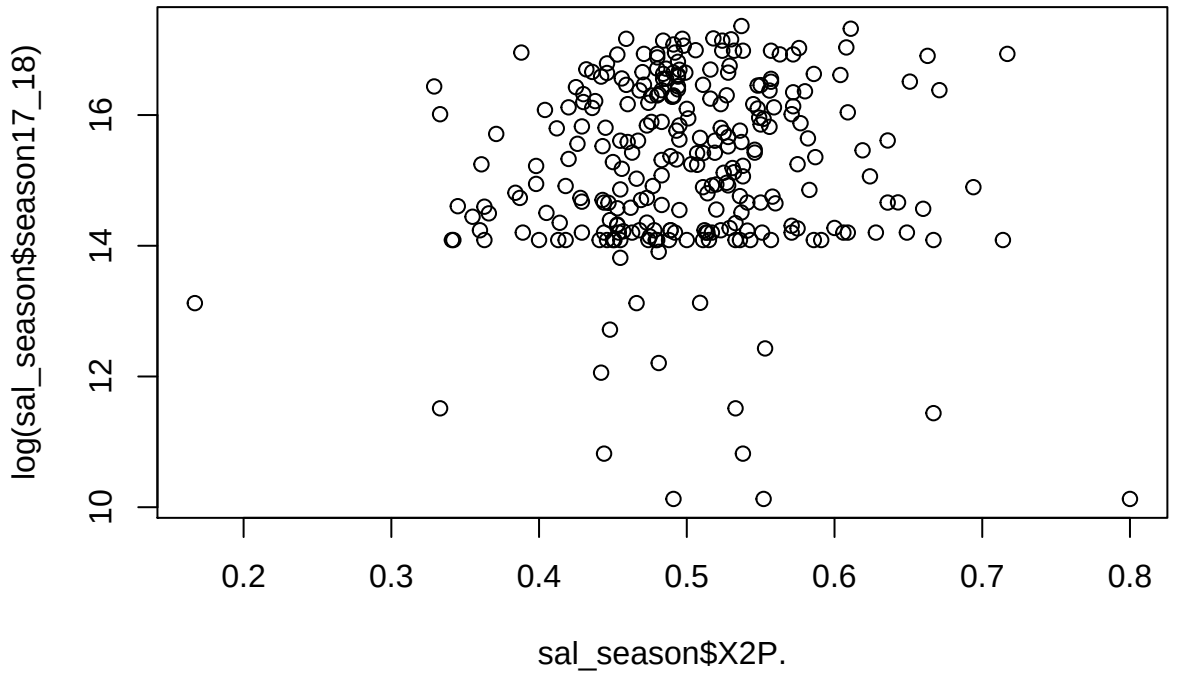

```
plot(sal_season$WS, log(sal_season$season17_18))
```



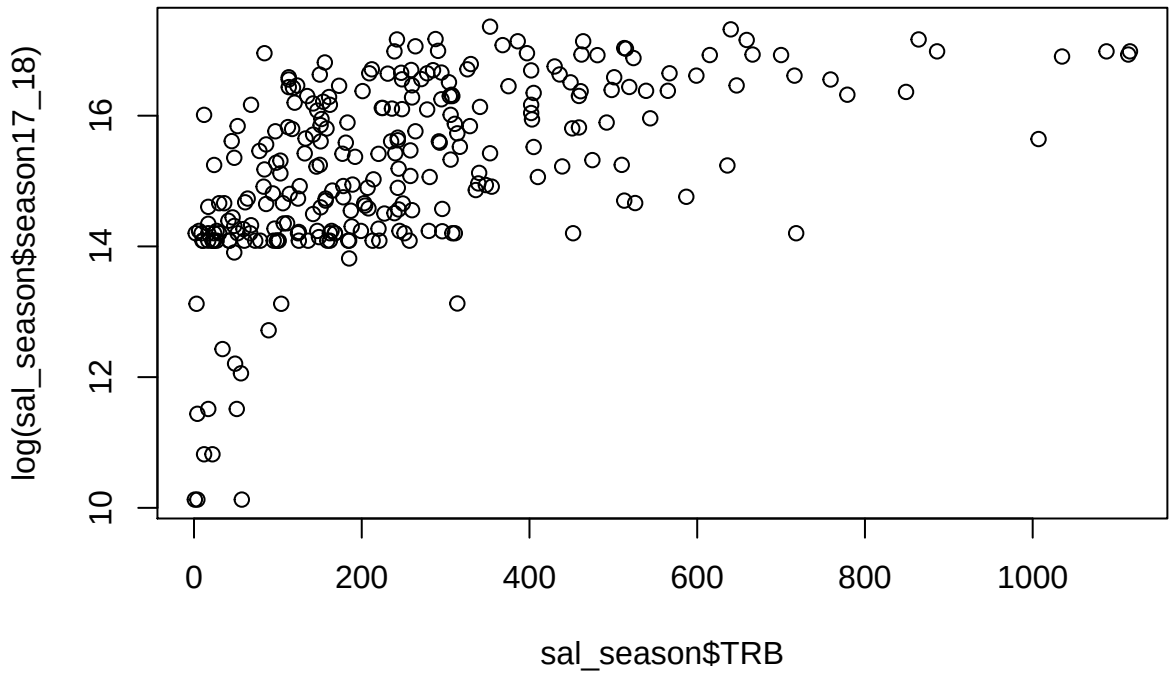

```
plot(sal_season$FGA, log(sal_season$season17_18))
```



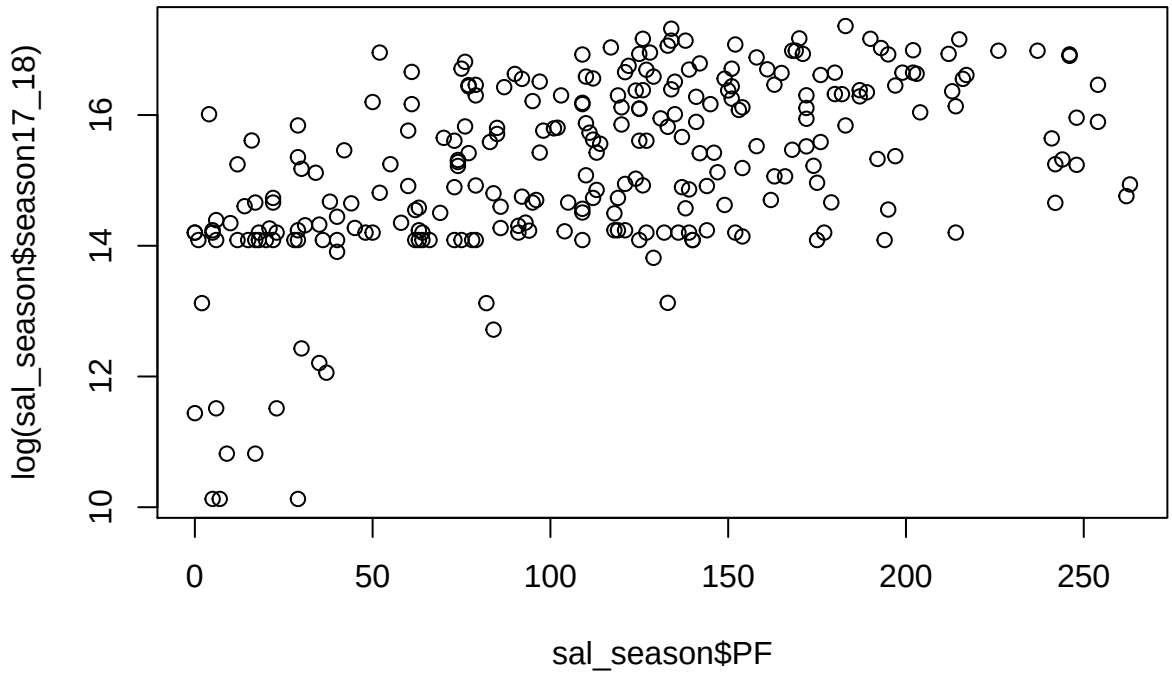

```
plot(sal_season$X2P., log(sal_season$season17_18))
```




```
plot(sal_season$TRB, log(sal_season$season17_18))
```

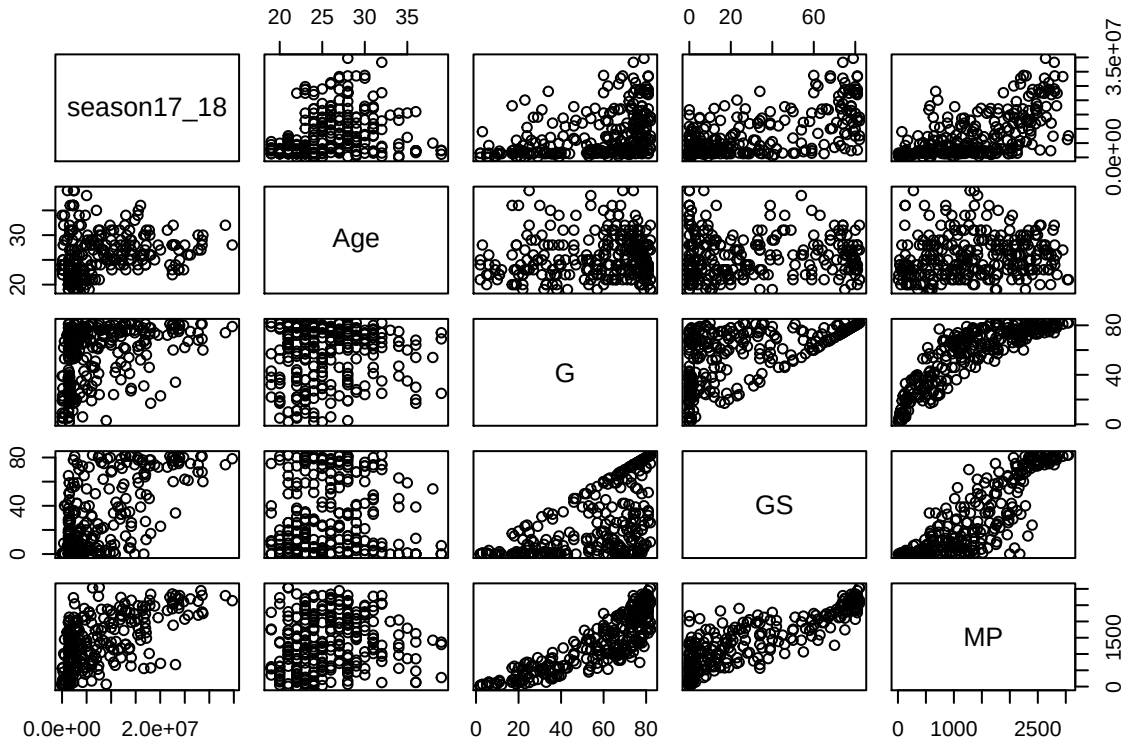



```
plot(sal_season$PF, log(sal_season$season17_18))
```

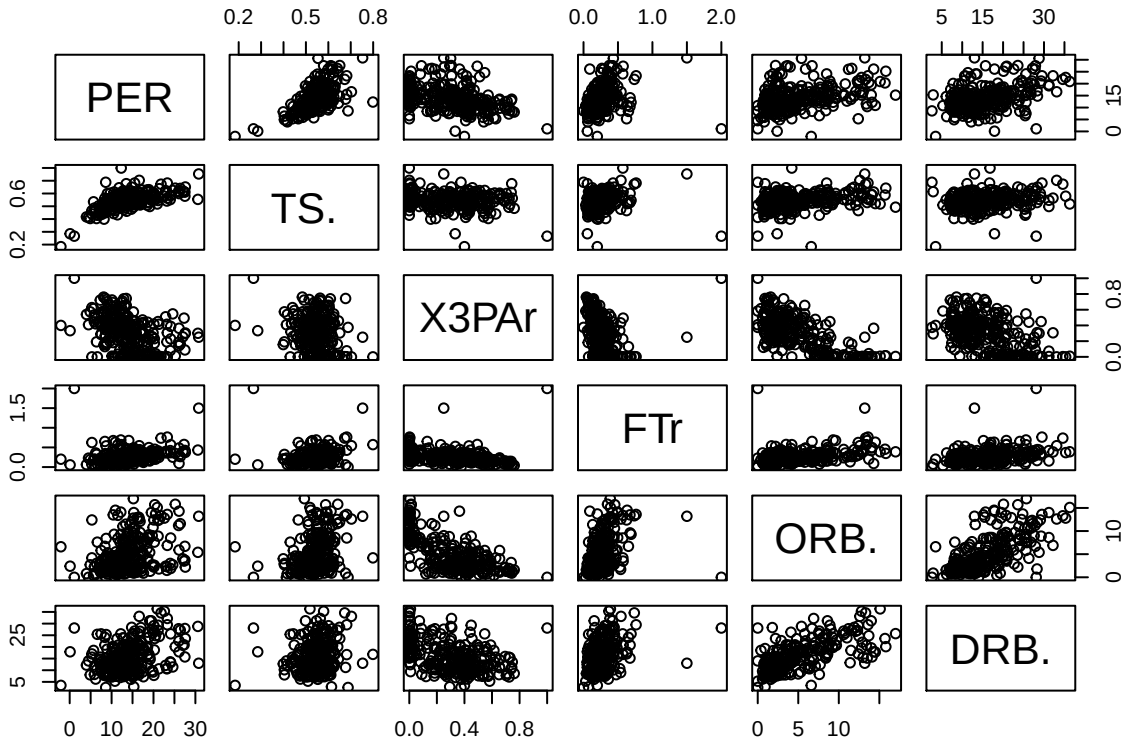


Pairwise Plots

```
pairs(~ ., data = sal_season[,-c(1, 2, 4, 9:50)])
```




```
pairs(~ ., data = sal_season[,-c(1:8, 15:50)])
```




```
# pairs(~ ., data = sal_season[,-c(1:8, 15:50)])
#
# pairs(~ ., data = sal_season[,-c(1, 2, 3, 4, 5:26)])
#
# pairs(~ ., data = sal_season[,-c(1, 2, 4, 27:50)])
```

```
#
# pairs(~ ., data = sal_season[,-c(1, 2, 3, 4, 5:26)])
#
# pairs(~ ., data = sal_season[,-c(1, 2, 3, 4, 5:26)])
#
# pairs(~ ., data = sal_season[,-c(1, 2, 3, 4, 5:26)])
```

Model

Ersan Ilyasova appears twice in the data

```
sal_lm <- lm(season17_18 ~ ., data = sal_season[,-1])

# taking out the Player name as a predictor

# sal_lm_log <- lm(log(season17_18) ~ ., data = sal_season[,-1])
```

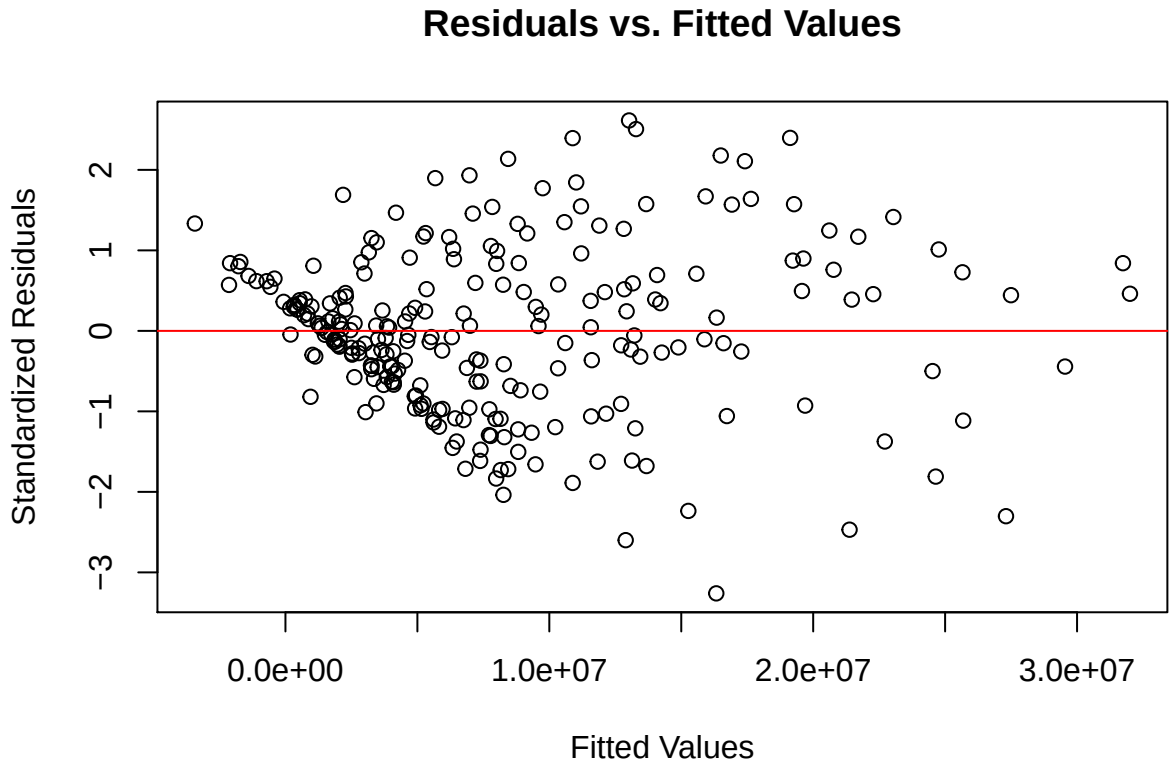
```
summary(sal_lm)
```

```
##
## Call:
## lm(formula = season17_18 ~ ., data = sal_season[, -1])
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -11327353  -2417586    -54162   2313882  10089039
##
## Coefficients: (4 not defined because of singularities)
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   6542174   18862010   0.347  0.72913
## TmBOS         4869014    3037047   1.603  0.11069
## TmBRK         6604862    3459351   1.909  0.05786 .
## TmCHI         -378713    2574477  -0.147  0.88322
## TmCHO         3403176    3106585   1.095  0.27482
## TmCLE        10744377    3629557   2.960  0.00350 **
## TmDAL         -392379    3299365  -0.119  0.90547
## TmDEN        10357000    4341774   2.385  0.01813 *
## TmDET         -223243    3023989  -0.074  0.94123
## TmGSW         -585550    2833224  -0.207  0.83651
## TmHOU         1391480    2944455   0.473  0.63710
## TmIND         3473134    3389907   1.025  0.30699
## TmLAC         5185363    3633072   1.427  0.15528
## TmLAL        10914462    4906780   2.224  0.02740 *
## TmMEM         3899026    2887935   1.350  0.17872
## TmMIA        -3260942    2520834  -1.294  0.19751
## TmMIL         7136036    3278238   2.177  0.03084 *
## TmMIN         8701474    4766103   1.826  0.06960 .
## TmNOP         5872974    2816140   2.085  0.03848 *
## TmNYK        10389313    4288685   2.422  0.01643 *
## TmOKC         3734971    2721479   1.372  0.17169
## TmORL         8413103    3849386   2.186  0.03018 *
## TmPHI         1852862    2884751   0.642  0.52152
## TmPHO        10841609    4477291   2.421  0.01648 *
```

```
## TmPOR        10984764    3581743   3.067  0.00251 **
## TmSAC         7843066    3949733   1.986  0.04863 *
## TmSAS        -4945344    2625115  -1.884  0.06124 .
## TmTOR         7338783    3345516   2.194  0.02958 *
## TmUTA         2334998    2729434   0.855  0.39345
## TmWAS         7340902    3404251   2.156  0.03242 *
## PosPF          231001    1412731   0.164  0.87030
## PosPG        -1632450    2189281  -0.746  0.45688
## PosSF          339244    1706913   0.199  0.84269
## PosSG        -1252795    1965683  -0.637  0.52474
## Age            266825      85339   3.127  0.00207 **
## G              -65981      41067  -1.607  0.10993
## GS              51588      24206   2.131  0.03447 *
## MP               3694       4067   0.908  0.36489
## PER          -2143579    1032157  -2.077  0.03928 *
## TS.         -52847734   39820914  -1.327  0.18619
## X3PAr        -5680313   18674735  -0.304  0.76136
## FTr          -2661102    4751624  -0.560  0.57617
## ORB.          3972224    2573212   1.544  0.12447
## DRB.          4227811    2505566   1.687  0.09331 .
## TRB.         -8384512    5104829  -1.642  0.10229
## AST.            82823     203135   0.408  0.68397
## STL.         -2471393    1609603  -1.535  0.12649
## BLK.          -401444    1104612  -0.363  0.71673
## TOV.          -102053     243704  -0.419  0.67591
## USG.          1391694     554589   2.509  0.01300 *
## OWS           3063430    7227332   0.424  0.67218
## DWS           9869363    7291266   1.354  0.17761
## WS           -2699408    7277393  -0.371  0.71114
## WS.48       137688835   76497363   1.800  0.07360 .
## OBPM         -6332477    7737798  -0.818  0.41425
## DBPM         -4689502    7630962  -0.615  0.53966
## BPM           6271605    7655516   0.819  0.41377
## VORP         -1804701    1261651  -1.430  0.15438
## FG              43788      75361   0.581  0.56196
## FGA            -30811      38987  -0.790  0.43042
## FG.          -2128139  101144379  -0.021  0.98324
## X3P            114531      90205   1.270  0.20589
## X3PA           -17639      32348  -0.545  0.58624
## X3P.         -2338510    4389086  -0.533  0.59485
## X2P                NA         NA      NA       NA
## X2PA               NA         NA      NA       NA
## X2P.          -523092   18933404  -0.028  0.97799
## eFG.         33025293   84542965   0.391  0.69654
## FT              -3652      60087  -0.061  0.95161
## FTA             15832      31203   0.507  0.61253
## FT.          -470924    4649432  -0.101  0.91944
## ORB             42219      34386   1.228  0.22117
## DRB            -28239      12582  -2.244  0.02606 *
## TRB                NA         NA      NA       NA
## AST             12424      21704   0.572  0.56776
## STL            -20677      44257  -0.467  0.64093
## BLK            -28875      39329  -0.734  0.46381
## TOV             -4205      57252  -0.073  0.94153
```

```
## PF             -50114      17739  -2.825  0.00528 **
## PTS                NA         NA      NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4567000 on 175 degrees of freedom
##   (19 observations deleted due to missingness)
## Multiple R-squared:  0.7649, Adjusted R-squared:  0.6641
## F-statistic:  7.59 on 75 and 175 DF,  p-value: < 2.2e-16
```

```
plot(sal_lm$fitted.values, rstandard(sal_lm), main = "Residuals vs. Fitted Values", xlab = "Fitted Valu
abline(h = 0, col = "red")
```




```
# plot(sal_lm_log$fitted.values, rstandard(sal_lm_log), main = "Residuals vs. Fitted Values", xlab = "F
# abline(h = 0, col = "red")
```

Looks screwy. Let's log-transform the data–looking on kaggle, other ppl didn't do it

Elastic-Net Regularization

Finding the optimal alpha

```
set.seed(1262019)
n <- nrow(sal_season_mat)
# Determining a random foldid
foldid <- sample(1:10, # default is 10 folds
                 size = n,
                 replace = TRUE
                 )
```

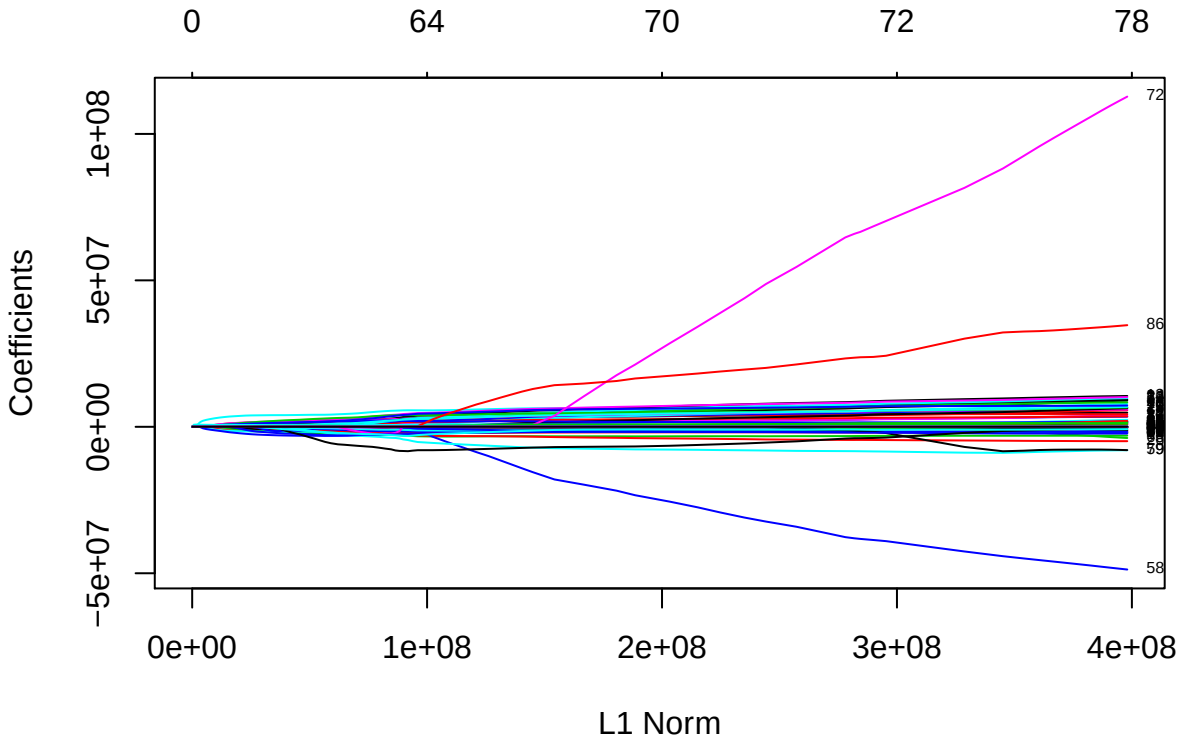
```
# Sequence of alphas to test
alphas <- seq(0, 1, by = .05)

devs <- rep(0, length(alphas))

for (i in 1:length(alphas)) {
cv <- cv.glmnet(sal_season_mat, sal, foldid = foldid, alpha = alphas[i], keep = TRUE)
  devs[i] <- min(cv$cvm)
}

devs
```

```
##  [1] 2.721649e+13 2.650403e+13 2.612741e+13 2.596997e+13 2.583762e+13
##  [6] 2.572320e+13 2.564012e+13 2.557848e+13 2.551959e+13 2.547592e+13
## [11] 2.544159e+13 2.541254e+13 2.538695e+13 2.536389e+13 2.534238e+13
## [16] 2.531901e+13 2.529810e+13 2.527939e+13 2.526370e+13 2.525401e+13
## [21] 2.524761e+13
```

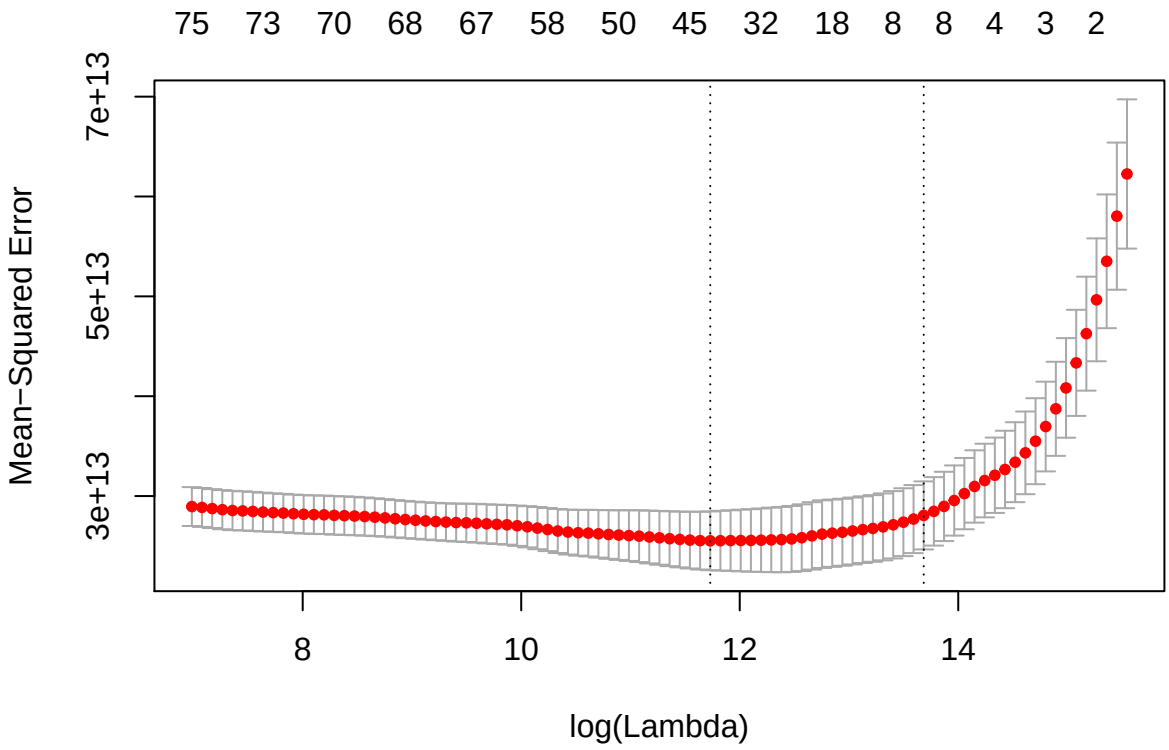
alpha of 1 is the best–just use alpha = .95 to strike a balance b/w ridge and lasso

```
set.seed(123)

cv_lm <- cv.glmnet(sal_season_mat, sal, alpha = 1)
```

```
plot(cv_lm$glmnet.fit, label = TRUE)
```




```
plot(cv_lm)
```




```
coef(cv_lm, s = cv_lm$lambda.min)
```

```
## 99 x 1 sparse Matrix of class "dgCMatrix"
##                        1
## (Intercept) -4520088.331
## TmBOS         929018.349
## TmBRK       -1731595.425
## TmCHI       -1384481.620
## TmCHO              .
## TmCLE        4011772.700
## TmDAL       -1046138.025
## TmDEN              .
## TmDET              .
## TmGSW         -54995.700
## TmHOU       -1132742.314
## TmIND       -1687568.406
## TmLAC         268172.603
## TmLAL         324618.659
## TmMEM        1771182.569
## TmMIA       -2020399.084
## TmMIL              .
## TmMIN       -1687614.143
## TmNOP        1588940.421
## TmNYK              .
## TmOKC          51880.331
## TmORL          43200.684
## TmPHI       -3011871.087
## TmPHO              .
## TmPOR        2051130.808
## TmSAC       -1098729.611
```

```
## TmSAS        -889652.987
## TmTOR        2306553.965
## TmUTA        1368837.697
## TmWAS         547257.563
## PosC                .
## PosC-F              .
## PosC-PF             .
## PosC-SF             .
## PosF                .
## PosF-C              .
## PosF-G              .
## PosG                .
## PosG-F              .
## PosPF               .
## PosPF-C             .
## PosPF-SF            .
## PosPG        -369101.381
## PosPG-SF            .
## PosPG-SG            .
## PosSF         751846.847
## PosSF-PF            .
## PosSF-PG            .
## PosSF-SG            .
## PosSG        -353222.339
## PosSG-PF            .
## PosSG-PG            .
## PosSG-SF            .
## Age           280231.438
## G             -33714.975
## GS             72512.820
## MP                  .
## PER                 .
## TS.          -1311692.435
## X3PAr        -453676.559
## FTr                 .
## ORB.                .
## DRB.                .
## TRB.                .
## AST.                .
## STL.         -320868.188
## BLK.                .
## TOV.                .
## USG.          230913.696
## OWS                 .
## DWS          1450814.970
## WS            572725.016
## WS.48               .
## OBPM                .
## DBPM                .
## BPM                 .
## VORP                .
## FG                  .
## FGA                 .
## FG.                 .
```

```
## X3P            22509.283
## X3PA               .
## X3P.        -2357619.326
## X2P                .
## X2PA               .
## X2P.        -1984215.987
## eFG.               .
## FT                 .
## FTA                .
## FT.          -986707.797
## ORB                .
## DRB                .
## TRB                .
## AST             1796.633
## STL                .
## BLK                .
## TOV                .
## PF              -8247.319
## PTS                .
```

Constructing LM from the elastic-net regularization

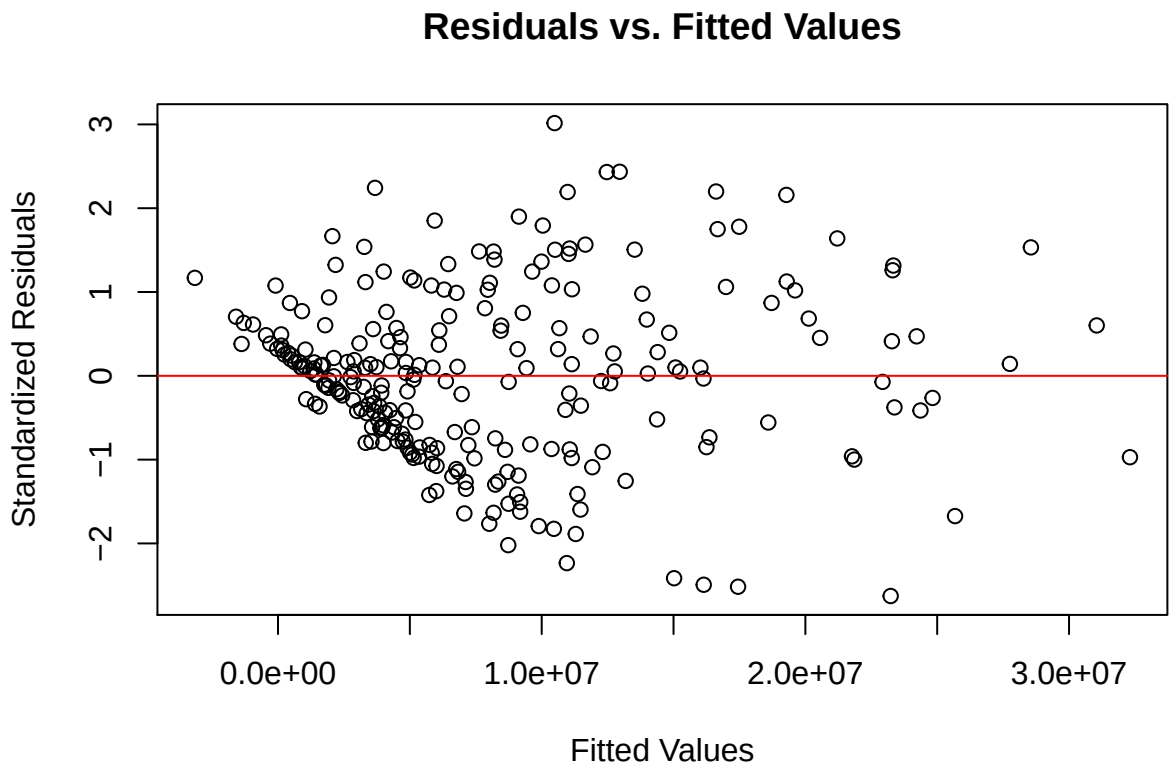
```
TmBOS <- ifelse(sal_season$Tm == "BOS", 1, 0)
TmBRK <- ifelse(sal_season$Tm == "BRK", 1, 0)
TmCHI <- ifelse(sal_season$Tm == "CHI", 1, 0)
TmCLE <- ifelse(sal_season$Tm == "CLE", 1, 0)
TmDAL <- ifelse(sal_season$Tm == "DAL", 1, 0)
TmGSW <- ifelse(sal_season$Tm == "GSW", 1, 0)
TmHOU <- ifelse(sal_season$Tm == "HOU", 1, 0)
TmIND <- ifelse(sal_season$Tm == "IND", 1, 0)
TmLAC <- ifelse(sal_season$Tm == "LAC", 1, 0)
TmLAL <- ifelse(sal_season$Tm == "LAL", 1, 0)
TmMEM <- ifelse(sal_season$Tm == "MEM", 1, 0)
TmMIA <- ifelse(sal_season$Tm == "MIA", 1, 0)
TmMIN <- ifelse(sal_season$Tm == "MIN", 1, 0)
TmNOP <- ifelse(sal_season$Tm == "NOP", 1, 0)
TmOKC <- ifelse(sal_season$Tm == "OKC", 1, 0)
TmORL <- ifelse(sal_season$Tm == "ORL", 1, 0)
TmPHI <- ifelse(sal_season$Tm == "PHI", 1, 0)
TmPOR <- ifelse(sal_season$Tm == "POR", 1, 0)
TmSAC <- ifelse(sal_season$Tm == "SAC", 1, 0)
TmSAS <- ifelse(sal_season$Tm == "SAS", 1, 0)
TmTOR <- ifelse(sal_season$Tm == "TOR", 1, 0)
TmUTA <- ifelse(sal_season$Tm == "UTA", 1, 0)
TmWAS <- ifelse(sal_season$Tm == "WAS", 1, 0)

PosPG <- ifelse(sal_season$Pos == "PG", 1, 0)
PosSF <- ifelse(sal_season$Pos == "SF", 1, 0)
PosSG <- ifelse(sal_season$Pos == "SG", 1, 0)

sal_lm_en <- lm(season17_18 ~ TmBOS + TmBRK + TmCHI + TmCLE + TmDAL + TmGSW + TmHOU + TmIND + TmLAC + Tm
```

```r
summary(sal_lm_en)
```

```
##
## Call:
## lm(formula = season17_18 ~ TmBOS + TmBRK + TmCHI + TmCLE + TmDAL +
##     TmGSW + TmHOU + TmIND + TmLAC + TmLAL + TmMEM + TmMIA + TmMIN +
##     TmNOP + TmOKC + TmORL + TmPHI + TmPOR + TmSAC + TmSAS + TmTOR +
##     TmTOR + TmUTA + TmWAS + PosPG + PosSF + PosSG + Age + G +
##     GS + TS. + X3PAr + STL. + USG. + DWS + WS + X3P + X3P. +
##     X2P. + FT. + AST + PF, data = sal_season)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -11235126  -2944417     61674   2402683  12623714
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3375462    5698496   0.592 0.554261
## TmBOS        1676023    2215299   0.757 0.450161
## TmBRK       -1825455    1754003  -1.041 0.299200
## TmCHI       -2153804    1685102  -1.278 0.202616
## TmCLE        4209533    2250980   1.870 0.062870 .
## TmDAL       -1827678    1699143  -1.076 0.283325
## TmGSW       -1461321    1660606  -0.880 0.379874
## TmHOU       -2424264    1919075  -1.263 0.207908
## TmIND       -2755507    1774764  -1.553 0.122031
## TmLAC        1533070    2500246   0.613 0.540432
## TmLAL        1079247    2005521   0.538 0.591054
## TmMEM        2112121    1655625   1.276 0.203469
## TmMIA       -3100015    1611065  -1.924 0.055687 .
## TmMIN       -1718518    2069112  -0.831 0.407171
## TmNOP        2114580    1784516   1.185 0.237379
## TmOKC         303143    1812058   0.167 0.867302
## TmORL        1017660    1850484   0.550 0.582946
## TmPHI       -3538304    1613211  -2.193 0.029387 *
## TmPOR        2693271    1572733   1.712 0.088292 .
## TmSAC       -1746274    1650691  -1.058 0.291320
## TmSAS       -2168731    1674643  -1.295 0.196735
## TmTOR        3249152    1699118   1.912 0.057210 .
## TmUTA        2212978    1916239   1.155 0.249469
## TmWAS        1445212    1751810   0.825 0.410322
## PosPG       -1463261    1148268  -1.274 0.203965
## PosSF         657915     947719   0.694 0.488322
## PosSG       -1131154    1015143  -1.114 0.266439
## Age           304538      76018   4.006 8.58e-05 ***
## G             -41210      28260  -1.458 0.146283
## GS             69450      18080   3.841 0.000162 ***
## TS.         -1425030   10249681  -0.139 0.889559
## X3PAr       -4436288    2936470  -1.511 0.132361
## STL.         -768712     578400  -1.329 0.185287
## USG.          171119      83490   2.050 0.041654 *
## DWS          2076072     729058   2.848 0.004845 **
## WS            511643     276229   1.852 0.065401 .
## X3P            41384      11716   3.532 0.000507 ***
```

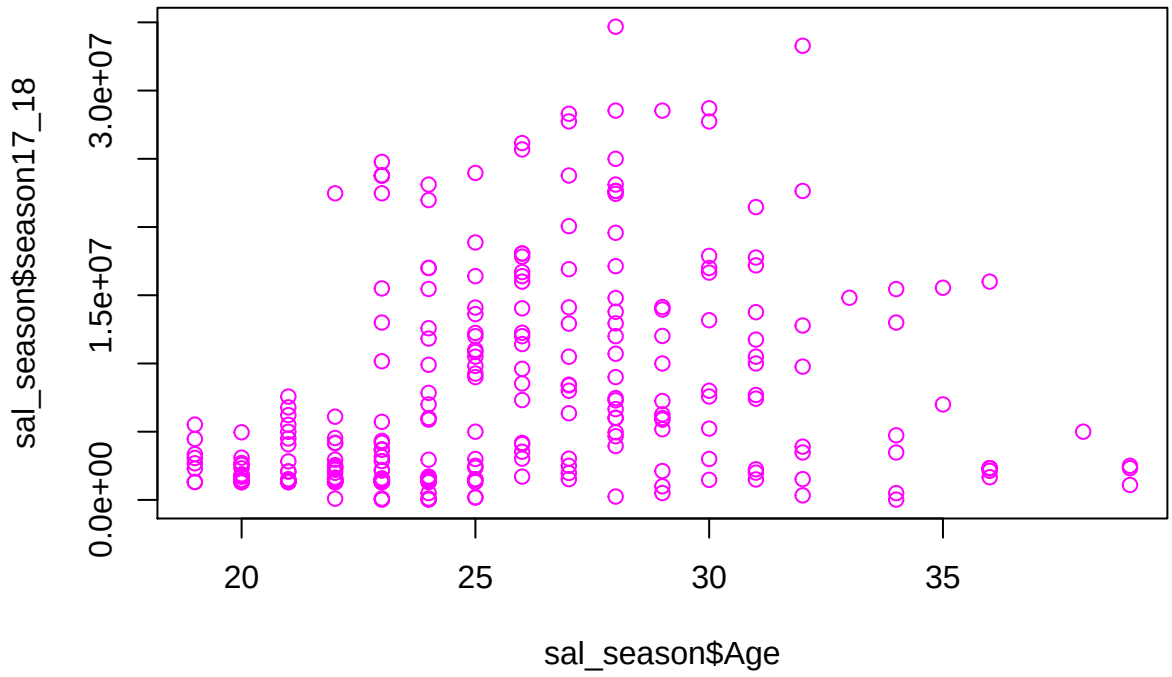
```
## X3P.        -2761067    3301119  -0.836 0.403883
## X2P.        -8895502    8128466  -1.094 0.275055
## FT.         -2398835    3103408  -0.773 0.440415
## AST             3470       3888   0.892 0.373208
## PF            -22911      10802  -2.121 0.035108 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4552000 on 209 degrees of freedom
##   (19 observations deleted due to missingness)
## Multiple R-squared:  0.721,  Adjusted R-squared:  0.6662
## F-statistic: 13.17 on 41 and 209 DF,  p-value: < 2.2e-16
```

```
plot(sal_lm_en$fitted.values, rstandard(sal_lm_en), main = "Residuals vs. Fitted Values", xlab = "Fitte
abline(h = 0, col = "red")
```

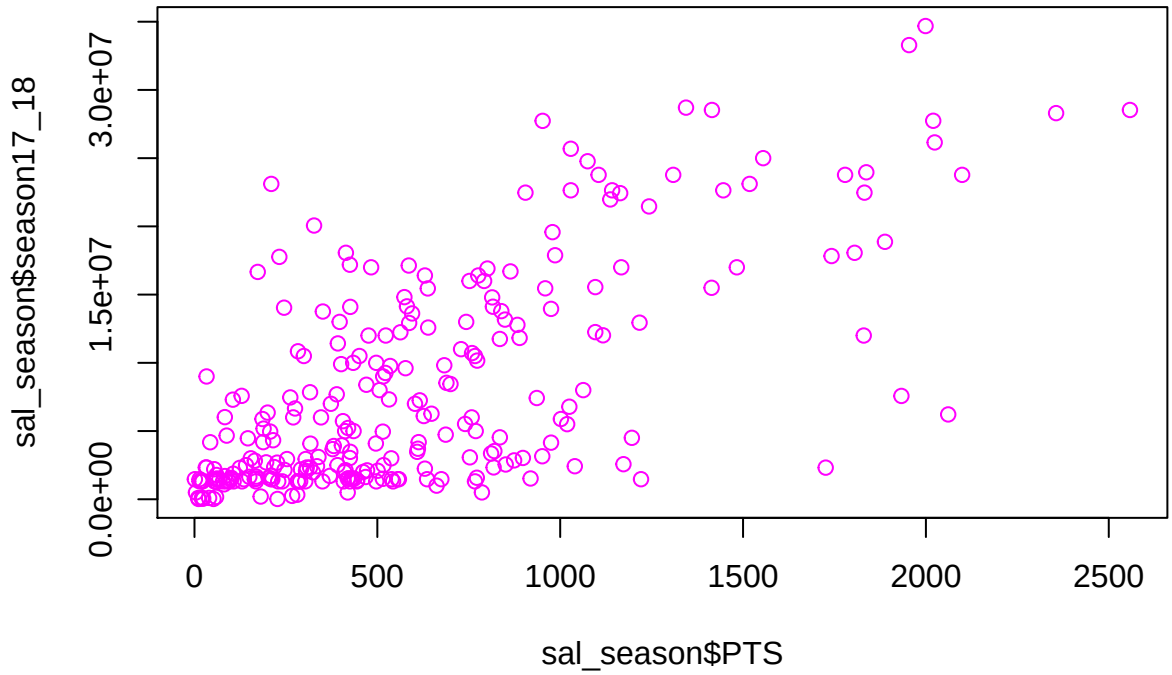


Plots based on model

```
plot(sal_season$Age, sal_season$season17_18, col = 6)
```



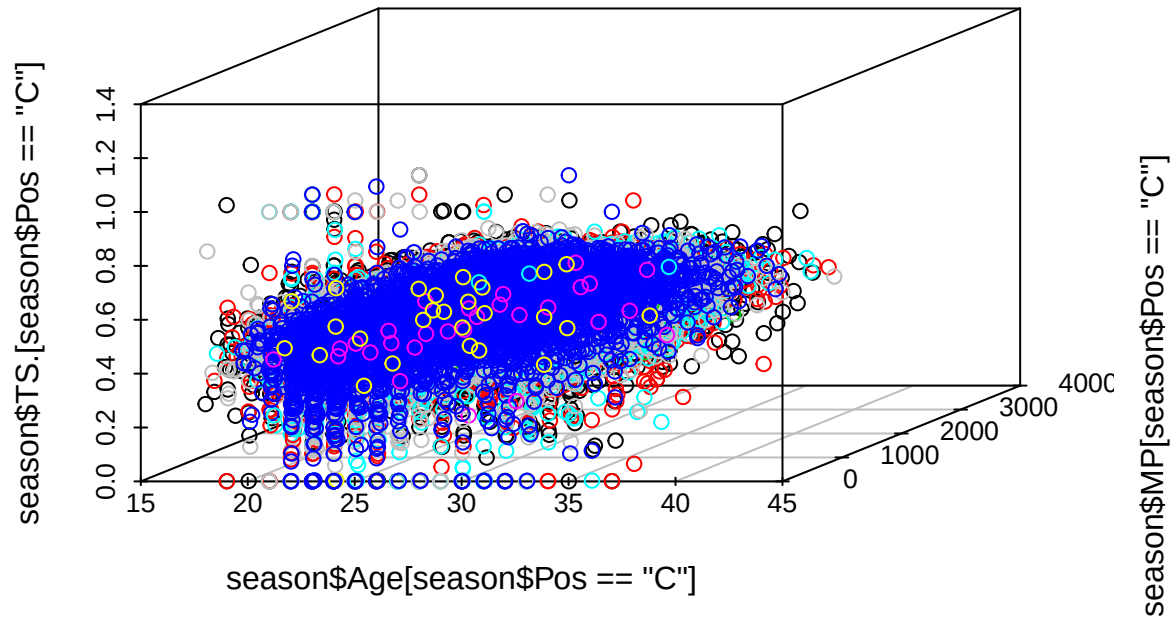

```
plot(sal_season$PTS, sal_season$season17_18, col = 6)
```



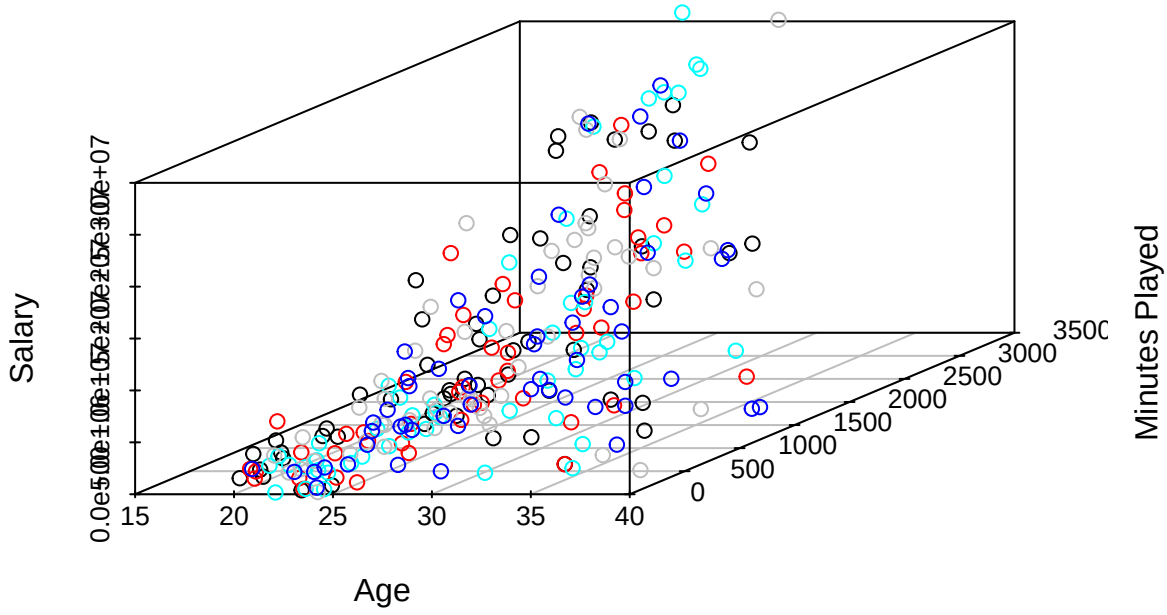
3d scatterplots

```
s3d_1 <- scatterplot3d(x = season$Age[season$Pos == "C"], y = season$MP[season$Pos == "C"], z = season$

positions <- levels(season$Pos)[3:24]
index <- 2
```

```
for (pos in positions) {
  s3d_1$points3d(x = season$Age[season$Pos == pos], y = season$MP[season$Pos == pos], z = season$TS.[sea
  index = index + 1
}
```

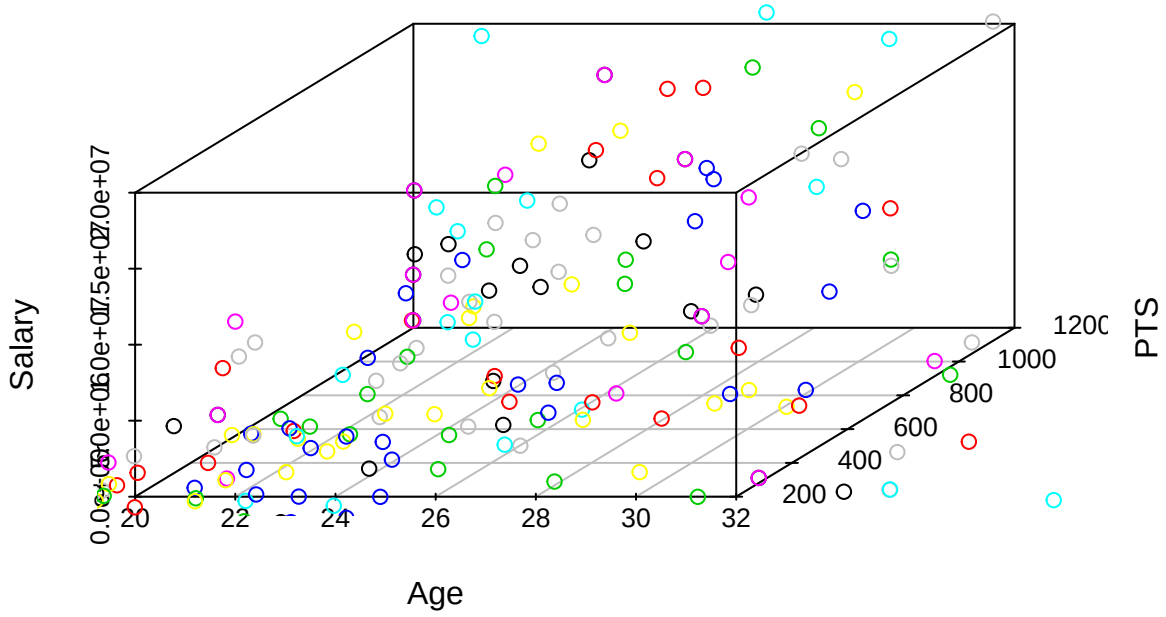



```
s3d_1 <- scatterplot3d(x = sal_season$Age[sal_season$Pos == "C"], y = sal_season$MP[sal_season$Pos == "
positions <- levels(sal_season$Pos)[3:24]
index <- 2

for (pos in positions) {
  s3d_1$points3d(x = sal_season$Age[sal_season$Pos == pos], y = sal_season$MP[sal_season$Pos == pos], z
  index = index + 1
}
```



Factor variable: Team

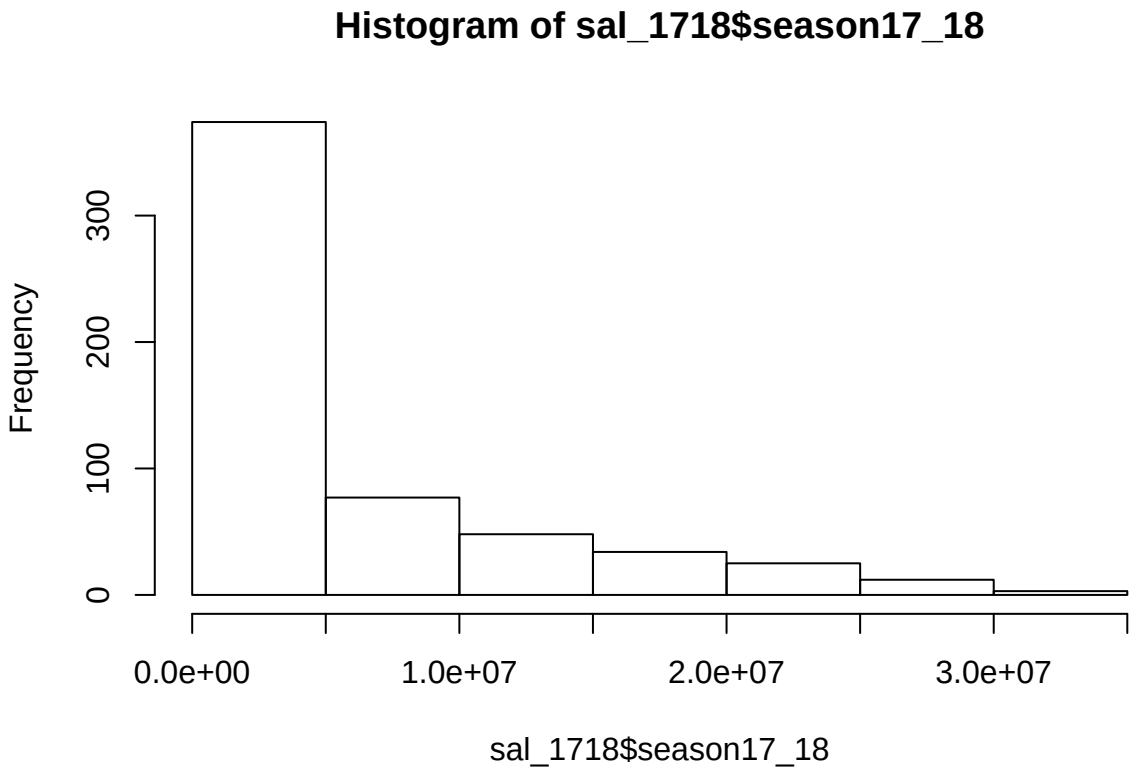
```
s3d_1 <- scatterplot3d(x = sal_season$Age[sal_season$Tm == "ORL"], y = sal_season$PTS[sal_season$Tm ==

teams <- levels(sal_season$Tm)[2:24]
index <- 2

for (tm in teams) {
  s3d_1$points3d(x = sal_season$Age[sal_season$Tm == tm], y = sal_season$PTS[sal_season$Tm == tm], z =
  index = index + 1
}
```



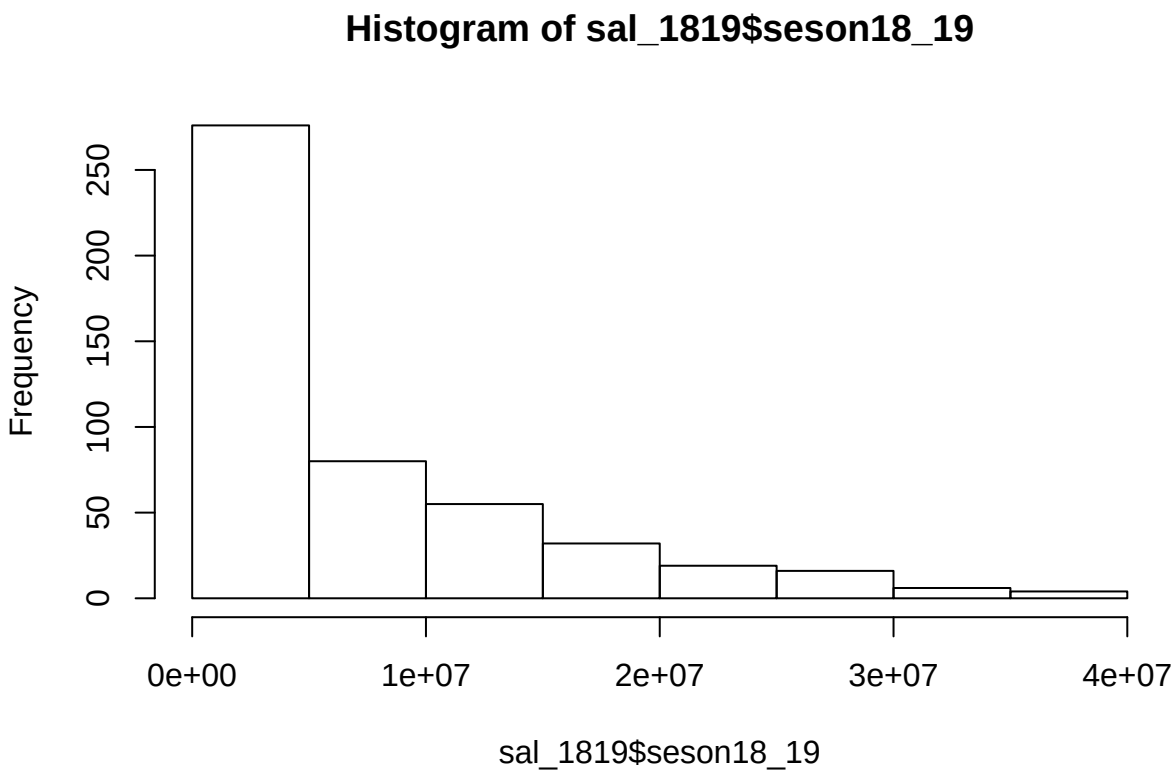
Exploration of the salary files

Exploration of sal_1718

```
hist(sal_1718$season17_18)
```




```
hist(sal_1819$seson18_19)
```



Exploratory Data Analysis of Seasons_Stats

```
sub_idx <- sample(nrow(season), size = 1000)

sub_season <- season[sub_idx,]
```

```
summary(season)
```

```
##        X              Year                     Player           Pos
##  Min.   :    0   Min.   :1950                    :   67   PF     :4966
##  1st Qu.: 6172   1st Qu.:1981   Eddie Johnson    :   33   SG     :4811
##  Median :12345   Median :1996   Mike Dunleavy    :   32   C      :4759
##  Mean   :12345   Mean   :1993   Gerald Henderson:   29   SF     :4699
##  3rd Qu.:18518   3rd Qu.:2007   Nazr Mohammed    :   28   PG     :4648
##  Max.   :24690   Max.   :2017   Kevin Willis     :   27   G      : 139
##                  NA's   :67     (Other)          :24475   (Other): 669
##       Age             Tm              G              GS
##  Min.   :18.00   TOT    : 2123   Min.   : 1.00   Min.   : 0.00
##  1st Qu.:24.00   NYK    : 1043   1st Qu.:27.00   1st Qu.: 0.00
##  Median :26.00   BOS    :  998   Median :58.00   Median : 8.00
##  Mean   :26.66   DET    :  917   Mean   :50.84   Mean   :23.59
##  3rd Qu.:29.00   PHI    :  871   3rd Qu.:75.00   3rd Qu.:45.00
##  Max.   :44.00   LAL    :  834   Max.   :88.00   Max.   :83.00
##  NA's   :75      (Other):17905   NA's   :67      NA's   :6458
##        MP             PER              TS.             X3PAr
##  Min.   :   0   Min.   :-90.60   Min.   :0.000   Min.   :0.000
##  1st Qu.: 340   1st Qu.:  9.80   1st Qu.:0.458   1st Qu.:0.005
##  Median :1053   Median : 12.70   Median :0.506   Median :0.064
##  Mean   :1210   Mean   : 12.48   Mean   :0.493   Mean   :0.159
##  3rd Qu.:1971   3rd Qu.: 15.60   3rd Qu.:0.544   3rd Qu.:0.288
##  Max.   :3882   Max.   :129.10   Max.   :1.136   Max.   :1.000
##  NA's   :553    NA's   :590      NA's   :153     NA's   :5852
##       FTr              ORB.             DRB.             TRB.
##  Min.   :0.0000   Min.   :  0.000   Min.   :  0.00   Min.   :  0.000
##  1st Qu.:0.2080   1st Qu.:  2.600   1st Qu.:  8.80   1st Qu.:  5.900
##  Median :0.2960   Median :  5.400   Median : 12.70   Median :  9.200
##  Mean   :0.3255   Mean   :  6.182   Mean   : 13.71   Mean   :  9.949
##  3rd Qu.:0.4000   3rd Qu.:  9.000   3rd Qu.: 18.10   3rd Qu.: 13.500
##  Max.   :6.0000   Max.   :100.000   Max.   :100.00   Max.   :100.000
##  NA's   :166      NA's   :3899      NA's   :3899     NA's   :3120
##       AST.             STL.             BLK.             TOV.
##  Min.   :  0.00   Min.   : 0.000   Min.   : 0.000   Min.   :  0.00
##  1st Qu.:  6.50   1st Qu.: 1.100   1st Qu.: 0.300   1st Qu.: 11.40
##  Median : 10.50   Median : 1.500   Median : 0.900   Median : 14.20
##  Mean   : 13.01   Mean   : 1.648   Mean   : 1.411   Mean   : 15.09
##  3rd Qu.: 17.60   3rd Qu.: 2.100   3rd Qu.: 1.900   3rd Qu.: 17.70
##  Max.   :100.00   Max.   :24.200   Max.   :77.800   Max.   :100.00
##  NA's   :2136     NA's   :3899     NA's   :3899     NA's   :5109
##       USG.          blanl              OWS              DWS
##  Min.   :  0.00   Mode:logical   Min.   :-5.100   Min.   :-1.000
##  1st Qu.: 15.40   NA's:24691     1st Qu.:-0.100   1st Qu.: 0.200
##  Median : 18.60                  Median : 0.400   Median : 0.800
##  Mean   : 18.91                  Mean   : 1.257   Mean   : 1.227
##  3rd Qu.: 22.20                  3rd Qu.: 1.900   3rd Qu.: 1.800
```

```
##  Max.   :100.00                 Max.   :18.300   Max.   :16.000
##  NA's   :5051                    NA's   :106      NA's   :106
##        WS              WS.48          blank2             OBPM
##  Min.   :-2.800   Min.   :-2.519   Mode:logical   Min.   :-73.800
##  1st Qu.: 0.200   1st Qu.: 0.031   NA's:24691     1st Qu.: -3.400
##  Median : 1.400   Median : 0.075                  Median : -1.500
##  Mean   : 2.486   Mean   : 0.065                  Mean   : -1.778
##  3rd Qu.: 3.800   3rd Qu.: 0.115                  3rd Qu.:  0.300
##  Max.   :25.400   Max.   : 2.123                  Max.   : 47.800
##  NA's   :106      NA's   :590                     NA's   :3894
##       DBPM              BPM               VORP              FG
##  Min.   :-30.400   Min.   :-86.700   Min.   :-2.60   Min.   :   0.0
##  1st Qu.: -1.700   1st Qu.: -4.200   1st Qu.:-0.20   1st Qu.:  41.0
##  Median : -0.500   Median : -1.800   Median : 0.00   Median : 141.0
##  Mean   : -0.549   Mean   : -2.327   Mean   : 0.56   Mean   : 195.3
##  3rd Qu.:  0.700   3rd Qu.:  0.300   3rd Qu.: 0.90   3rd Qu.: 299.0
##  Max.   : 46.800   Max.   : 36.200   Max.   :12.40   Max.   :1597.0
##  NA's   :3894      NA's   :3894      NA's   :3894    NA's   :67
##       FGA              FG.              X3P              X3PA
##  Min.   :   0.0   Min.   :0.0000   Min.   :  0.00   Min.   :  0.0
##  1st Qu.:  99.0   1st Qu.:0.3930   1st Qu.:  0.00   1st Qu.:  1.0
##  Median : 321.0   Median :0.4390   Median :  2.00   Median : 11.0
##  Mean   : 430.6   Mean   :0.4308   Mean   : 22.21   Mean   : 63.6
##  3rd Qu.: 661.0   3rd Qu.:0.4800   3rd Qu.: 27.00   3rd Qu.: 84.0
##  Max.   :3159.0   Max.   :1.0000   Max.   :402.00   Max.   :886.0
##  NA's   :67       NA's   :166      NA's   :5764     NA's   :5764
##       X3P.             X2P              X2PA             X2P.
##  Min.   :0.000   Min.   :   0.0   Min.   :   0.0   Min.   :0.0000
##  1st Qu.:0.100   1st Qu.:  35.0   1st Qu.:  82.0   1st Qu.:0.4070
##  Median :0.292   Median : 122.0   Median : 270.0   Median :0.4560
##  Mean   :0.249   Mean   : 178.3   Mean   : 381.8   Mean   :0.4453
##  3rd Qu.:0.363   3rd Qu.: 268.0   3rd Qu.: 579.2   3rd Qu.:0.4960
##  Max.   :1.000   Max.   :1597.0   Max.   :3159.0   Max.   :1.0000
##  NA's   :9275    NA's   :67       NA's   :67       NA's   :195
##       eFG.              FT               FTA              FT.
##  Min.   :0.0000   Min.   :  0.0   Min.   :   0.0   Min.   :0.0000
##  1st Qu.:0.4140   1st Qu.: 18.0   1st Qu.:  27.0   1st Qu.:0.6570
##  Median :0.4630   Median : 63.0   Median :  88.0   Median :0.7430
##  Mean   :0.4507   Mean   :102.4   Mean   : 136.8   Mean   :0.7193
##  3rd Qu.:0.5010   3rd Qu.:149.0   3rd Qu.: 201.0   3rd Qu.:0.8080
##  Max.   :1.5000   Max.   :840.0   Max.   :1363.0   Max.   :1.0000
##  NA's   :166      NA's   :67      NA's   :67       NA's   :925
##       ORB              DRB              TRB              AST
##  Min.   :  0.00   Min.   :   0.0   Min.   :   0.0   Min.   :   0.0
##  1st Qu.: 12.00   1st Qu.:  33.0   1st Qu.:  51.0   1st Qu.:  19.0
##  Median : 38.00   Median : 106.0   Median : 159.0   Median :  68.0
##  Mean   : 62.19   Mean   : 147.2   Mean   : 224.6   Mean   : 114.9
##  3rd Qu.: 91.00   3rd Qu.: 212.0   3rd Qu.: 322.0   3rd Qu.: 160.0
##  Max.   :587.00   Max.   :1111.0   Max.   :2149.0   Max.   :1164.0
##  NA's   :3894     NA's   :3894     NA's   :379      NA's   :67
##       STL              BLK              TOV              PF
##  Min.   :  0.0   Min.   :  0.00   Min.   :  0.00   Min.   :  0.0
##  1st Qu.:  9.0   1st Qu.:  3.00   1st Qu.: 18.00   1st Qu.: 39.0
##  Median : 29.0   Median : 11.00   Median : 55.00   Median :109.0
```

```
##  Mean   : 39.9   Mean   : 24.47   Mean   : 73.94   Mean   :116.3
##  3rd Qu.: 60.0   3rd Qu.: 29.00   3rd Qu.:112.00   3rd Qu.:182.0
##  Max.   :301.0   Max.   :456.00   Max.   :464.00   Max.   :386.0
##  NA's   :3894    NA's   :3894     NA's   :5046     NA's   :67
##       PTS
##  Min.   :   0.0
##  1st Qu.: 106.0
##  Median : 364.0
##  Mean   : 510.1
##  3rd Qu.: 778.0
##  Max.   :4029.0
##  NA's   :67
```

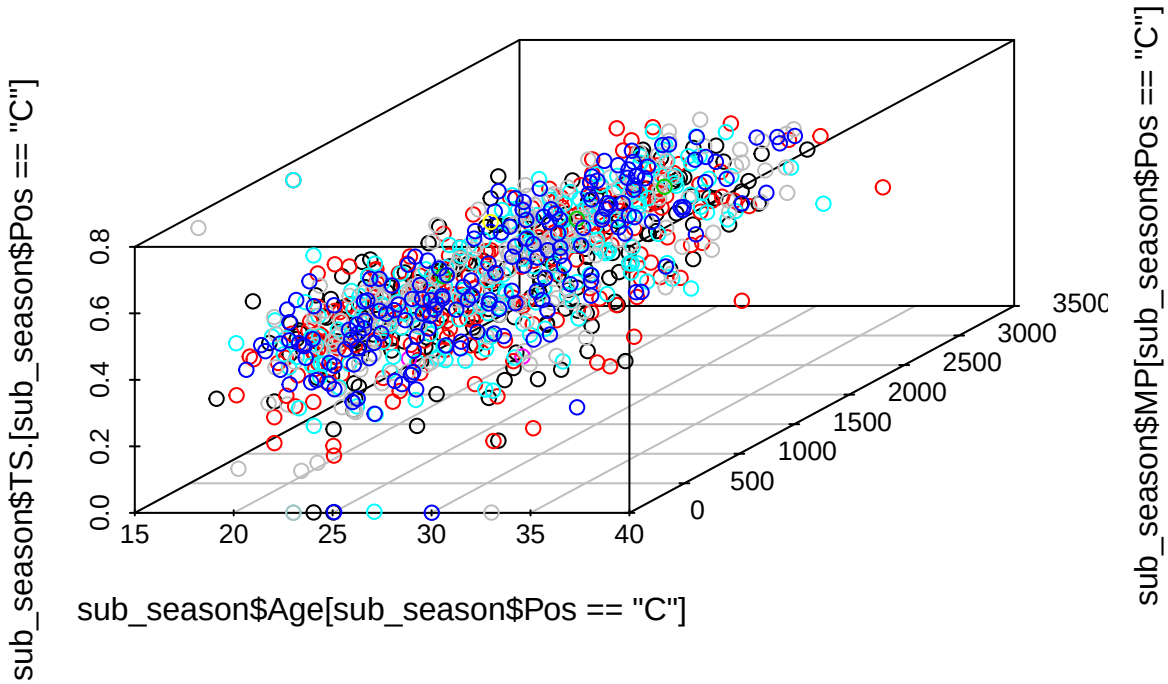
```
str(season)
```

```
## 'data.frame':    24691 obs. of  53 variables:
##  $ X     : int  0 1 2 3 4 5 6 7 8 9 ...
##  $ Year  : int  1950 1950 1950 1950 1950 1950 1950 1950 1950 1950 ...
##  $ Player: Factor w/ 3922 levels "","A.C. Green",..: 792 722 2419 1196 1196 1196 3044 1412 611 611 .
##  $ Pos   : Factor w/ 24 levels "","C","C-F","C-PF",..: 10 21 17 6 6 6 9 10 7 7 ...
##  $ Age   : int  31 29 25 24 24 24 22 23 28 28 ...
##  $ Tm    : Factor w/ 70 levels "","AND","ATL",..: 23 27 15 63 22 43 27 64 63 23 ...
##  $ G     : int  63 49 67 15 13 2 60 3 65 36 ...
##  $ GS    : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ MP    : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ PER   : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ TS.   : num  0.368 0.435 0.394 0.312 0.308 0.376 0.422 0.275 0.346 0.362 ...
##  $ X3PAr : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ FTr   : num  0.467 0.387 0.259 0.395 0.378 0.75 0.301 0.313 0.395 0.48 ...
##  $ ORB.  : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ DRB.  : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ TRB.  : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ AST.  : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ STL.  : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ BLK.  : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ TOV.  : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ USG.  : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ blanl : logi  NA NA NA NA NA NA ...
##  $ OWS   : num  -0.1 1.6 0.9 -0.5 -0.5 0 3.6 -0.1 -2.2 -0.7 ...
##  $ DWS   : num  3.6 0.6 2.8 -0.1 -0.1 0 1.2 0 5 2.2 ...
##  $ WS    : num  3.5 2.2 3.6 -0.6 -0.6 0 4.8 -0.1 2.8 1.5 ...
##  $ WS.48 : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ blank2: logi  NA NA NA NA NA NA ...
##  $ OBPM  : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ DBPM  : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ BPM   : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ VORP  : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ FG    : int  144 102 174 22 21 1 340 5 226 125 ...
##  $ FGA   : int  516 274 499 86 82 4 936 16 813 435 ...
##  $ FG.   : num  0.279 0.372 0.349 0.256 0.256 0.25 0.363 0.313 0.278 0.287 ...
##  $ X3P   : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ X3PA  : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ X3P.  : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ X2P   : int  144 102 174 22 21 1 340 5 226 125 ...
##  $ X2PA  : int  516 274 499 86 82 4 936 16 813 435 ...
```

```
##  $ X2P.   : num  0.279 0.372 0.349 0.256 0.256 0.25 0.363 0.313 0.278 0.287 ...
##  $ eFG.   : num  0.279 0.372 0.349 0.256 0.256 0.25 0.363 0.313 0.278 0.287 ...
##  $ FT     : int  170 75 90 19 17 2 215 0 209 132 ...
##  $ FTA    : int  241 106 129 34 31 3 282 5 321 209 ...
##  $ FT.    : num  0.705 0.708 0.698 0.559 0.548 0.667 0.762 0 0.651 0.632 ...
##  $ ORB    : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ DRB    : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ TRB    : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ AST    : int  176 109 140 20 20 0 233 2 163 75 ...
##  $ STL    : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ BLK    : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ TOV    : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ PF     : int  217 99 192 29 27 2 132 6 273 140 ...
##  $ PTS    : int  458 279 438 63 59 4 895 10 661 382 ...
```

3d Scatterplot of subseason

```
s3d_1 <- scatterplot3d(x = sub_season$Age[sub_season$Pos == "C"], y = sub_season$MP[sub_season$Pos == "

positions <- levels(sub_season$Pos)[3:24]
index <- 2

for (pos in positions) {
  s3d_1$points3d(x = sub_season$Age[sub_season$Pos == pos], y = sub_season$MP[sub_season$Pos == pos], z
  index = index + 1
}
```

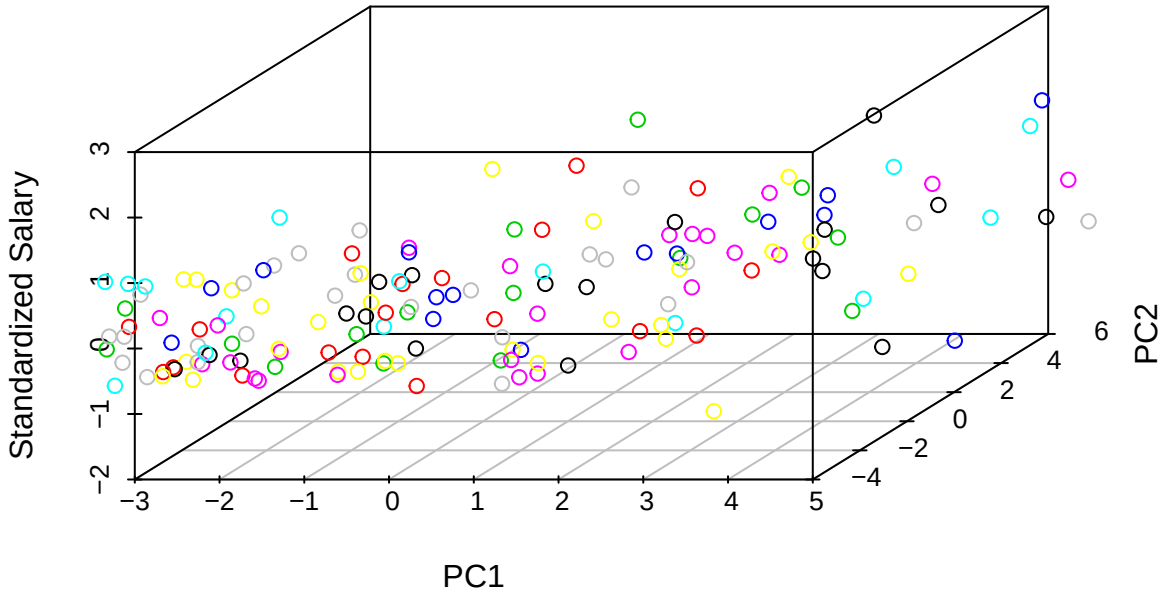


Principal Components Analysis

```
pr.out <- prcomp(sal_season_mat[,-which(colSums(sal_season_mat) == 0)], scale = TRUE)
# remember, sal_season_mat doesn't have the dependent variable.
# therefore, the principal components can be used to predict the dependent variable
# -which... takes out rows with no zeros
```

```
pc_dat <- pr.out$x
```

```
sal_st <- (sal - mean(sal)) / sd(sal)

s3d_pc <- scatterplot3d(x = pc_dat[sal_season_mat,"TmBOS"] == 1,1], y = pc_dat[sal_season_mat,"TmBOS"]

teams <- colnames(sal_season_mat)[2:24]
index <- 2

for (tm in teams) {
  s3d_pc$points3d(x = pc_dat[sal_season_mat,tm] == 1,1], y = pc_dat[sal_season_mat,tm] == 1,2], z = sa
  index = index + 1
}
```



Model

```
sal_lm_pc <- lm(sal ~ ., data = as.data.frame(pc_dat))

# taking out the Player name as a predictor

# sal_lm_log <- lm(log(season17_18) ~ ., data = sal_season[,-1])
```

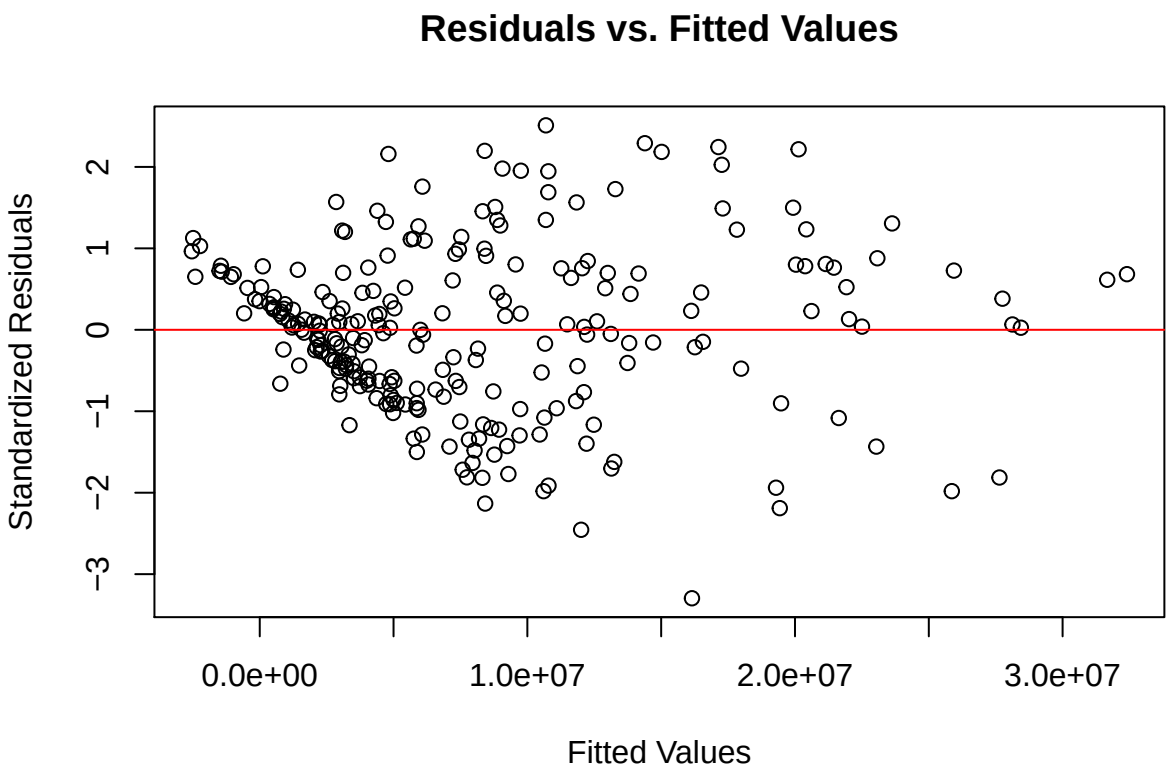
```
summary(sal_lm_pc)
```

```
##
## Call:
## lm(formula = sal ~ ., data = as.data.frame(pc_dat))
```

```
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -11150792  -2455081    123991   2356822   9368058
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  8.858e+06  6.961e+05  12.725  < 2e-16 ***
## PC1          7.189e+05  2.994e+05   2.401 0.017415 *
## PC2         -9.695e+05  9.888e+05  -0.981 0.328217
## PC3          2.158e+06  8.910e+05   2.422 0.016473 *
## PC4         -8.569e+05  1.135e+06  -0.755 0.451489
## PC5         -1.011e+06  6.888e+05  -1.467 0.144188
## PC6         -2.328e+06  6.881e+05  -3.383 0.000889 ***
## PC7         -4.164e+05  3.795e+06  -0.110 0.912753
## PC8         -1.122e+06  7.145e+05  -1.570 0.118291
## PC9          1.603e+06  1.234e+06   1.299 0.195722
## PC10        -5.528e+05  8.289e+05  -0.667 0.505729
## PC11         2.547e+06  3.489e+06   0.730 0.466363
## PC12        -1.999e+06  1.203e+06  -1.662 0.098394 .
## PC13         3.662e+06  1.358e+06   2.697 0.007704 **
## PC14         2.192e+06  1.457e+06   1.505 0.134124
## PC15        -3.216e+05  5.258e+05  -0.612 0.541518
## PC16         2.622e+06  1.289e+06   2.035 0.043450 *
## PC17        -1.288e+06  2.469e+06  -0.522 0.602607
## PC18         1.036e+06  1.235e+06   0.839 0.402916
## PC19         2.096e+06  2.121e+06   0.988 0.324582
## PC20         1.576e+06  1.818e+06   0.867 0.387267
## PC21        -6.164e+05  9.829e+05  -0.627 0.531416
## PC22         3.774e+06  3.430e+06   1.101 0.272665
## PC23        -5.086e+06  3.732e+06  -1.363 0.174735
## PC24        -5.295e+05  2.500e+06  -0.212 0.832503
## PC25         4.674e+06  2.702e+06   1.730 0.085504 .
## PC26        -1.519e+06  1.622e+06  -0.937 0.350154
## PC27         3.919e+06  1.945e+06   2.015 0.045526 *
## PC28         6.937e+05  8.156e+05   0.851 0.396192
## PC29         5.672e+05  2.079e+06   0.273 0.785323
## PC30         1.101e+06  5.350e+05   2.058 0.041132 *
## PC31        -2.079e+04  1.285e+06  -0.016 0.987112
## PC32        -3.229e+06  2.049e+06  -1.576 0.116894
## PC33         2.522e+06  1.650e+06   1.529 0.128246
## PC34        -2.425e+06  1.631e+06  -1.487 0.138896
## PC35         6.619e+05  1.871e+06   0.354 0.723890
## PC36         7.494e+05  8.131e+05   0.922 0.358008
## PC37        -1.786e+06  2.463e+06  -0.725 0.469326
## PC38        -4.487e+05  1.573e+06  -0.285 0.775753
## PC39         8.112e+05  1.424e+06   0.570 0.569646
## PC40        -5.376e+06  1.812e+06  -2.966 0.003449 **
## PC41        -1.949e+06  1.620e+06  -1.203 0.230706
## PC42        -1.649e+06  1.694e+06  -0.974 0.331646
## PC43         1.522e+06  1.205e+06   1.263 0.208354
## PC44         1.123e+06  1.473e+06   0.763 0.446804
## PC45         2.928e+06  2.063e+06   1.419 0.157749
## PC46         2.736e+06  1.878e+06   1.457 0.146986
```

```
## PC47         1.622e+06  1.495e+06   1.085 0.279651
## PC48         1.504e+06  1.915e+06   0.785 0.433265
## PC49         4.284e+05  2.066e+06   0.207 0.835998
## PC50        -4.777e+06  2.898e+06  -1.648 0.101183
## PC51        -3.700e+06  2.931e+06  -1.262 0.208656
## PC52         4.376e+06  3.087e+06   1.418 0.158075
## PC53        -2.230e+06  2.513e+06  -0.887 0.376298
## PC54        -1.072e+07  4.428e+06  -2.421 0.016543 *
## PC55        -4.356e+06  4.497e+06  -0.969 0.334099
## PC56        -4.919e+06  3.200e+06  -1.537 0.126137
## PC57         7.042e+06  2.422e+06   2.908 0.004122 **
## PC58         3.313e+06  4.169e+06   0.795 0.427900
## PC59        -1.754e+06  6.038e+06  -0.291 0.771736
## PC60         8.951e+06  5.118e+06   1.749 0.082080 .
## PC61         3.577e+06  5.190e+06   0.689 0.491555
## PC62         5.536e+06  5.977e+06   0.926 0.355591
## PC63         7.078e+05  8.868e+06   0.080 0.936475
## PC64        -9.219e+06  7.711e+06  -1.196 0.233512
## PC65         2.111e+07  1.908e+07   1.106 0.270192
## PC66         1.070e+06  1.410e+07   0.076 0.939578
## PC67         2.789e+07  1.516e+07   1.840 0.067546 .
## PC68        -1.841e+07  1.046e+07  -1.761 0.080106 .
## PC69         2.674e+07  1.629e+07   1.641 0.102616
## PC70         1.135e+07  1.458e+07   0.779 0.437205
## PC71         2.621e+06  2.516e+07   0.104 0.917163
## PC72        -1.803e+07  4.608e+07  -0.391 0.696120
## PC73         6.556e+07  5.232e+07   1.253 0.211885
## PC74        -1.189e+08  8.345e+07  -1.425 0.155873
## PC75        -2.245e+07  1.634e+08  -0.137 0.890886
## PC76         1.649e+21  5.844e+21   0.282 0.778159
## PC77         5.972e+21  3.260e+21   1.832 0.068722 .
## PC78         5.574e+21  3.377e+21   1.651 0.100677
## PC79        -9.972e+21  4.371e+21  -2.282 0.023759 *
## PC80         3.303e+21  2.725e+21   1.212 0.227202
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4486000 on 170 degrees of freedom
## Multiple R-squared:  0.7797, Adjusted R-squared:  0.676
## F-statistic: 7.519 on 80 and 170 DF,  p-value: < 2.2e-16
```

```r
plot(sal_lm_pc$fitted.values, rstandard(sal_lm_pc), main = "Residuals vs. Fitted Values", xlab = "Fitte
abline(h = 0, col = "red")
```




```
# plot(sal_lm_log$fitted.values, rstandard(sal_lm_log), main = "Residuals vs. Fitted Values", xlab = "F
# abline(h = 0, col = "red")
```

Elastic-Net Regularization

Finding the optimal alpha

```
set.seed(1262019)
n <- nrow(sal_season_mat)
# Determining a random foldid
foldid <- sample(1:10, # default is 10 folds
                 size = n,
                 replace = TRUE
                 )

# Sequence of alphas to test
alphas <- seq(0, 1, by = .05)

devs <- rep(0, length(alphas))

for (i in 1:length(alphas)) {
cv <- cv.glmnet(pc_dat, sal, foldid = foldid, alpha = alphas[i], keep = TRUE)
  devs[i] <- min(cv$cvm)
}

devs
```

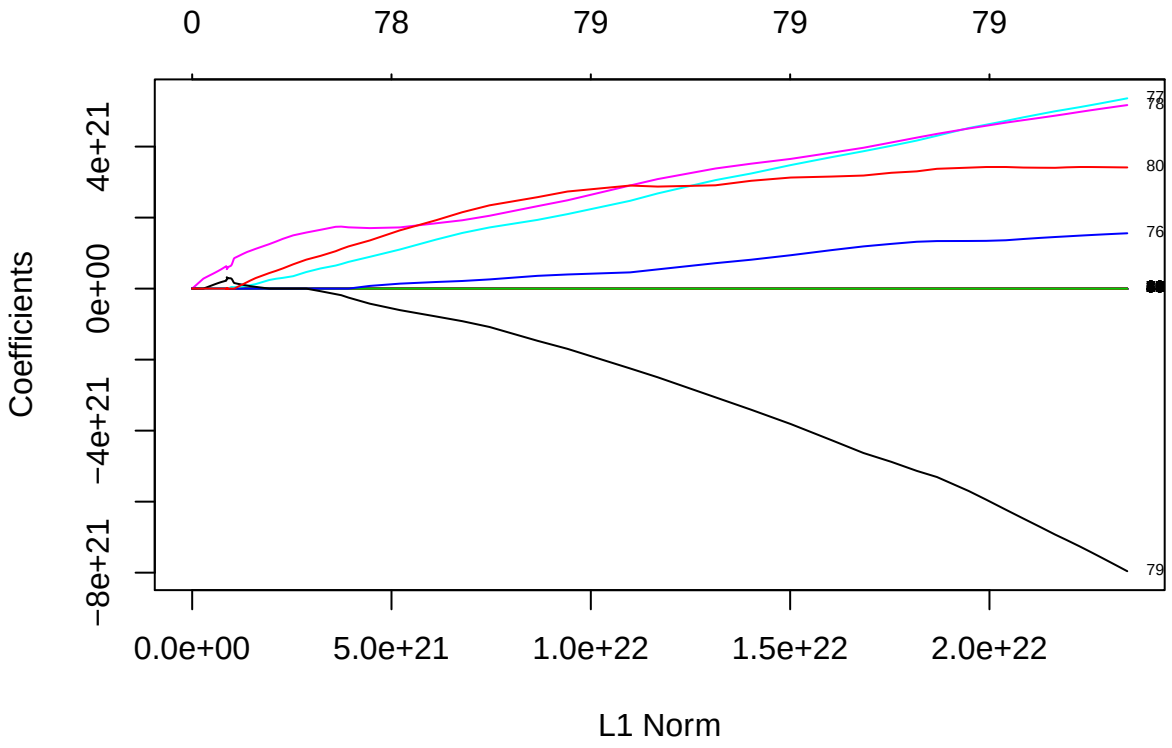
```
##  [1] 3.273366e+13 2.935139e+13 2.785179e+13 2.698035e+13 2.644814e+13
##  [6] 2.607851e+13 2.582010e+13 2.564716e+13 2.551392e+13 2.541935e+13
```

```
## [11] 2.535031e+13 2.528404e+13 2.523257e+13 2.519229e+13 2.516001e+13
## [16] 2.513499e+13 2.511456e+13 2.509738e+13 2.508346e+13 2.507156e+13
## [21] 2.506149e+13
```

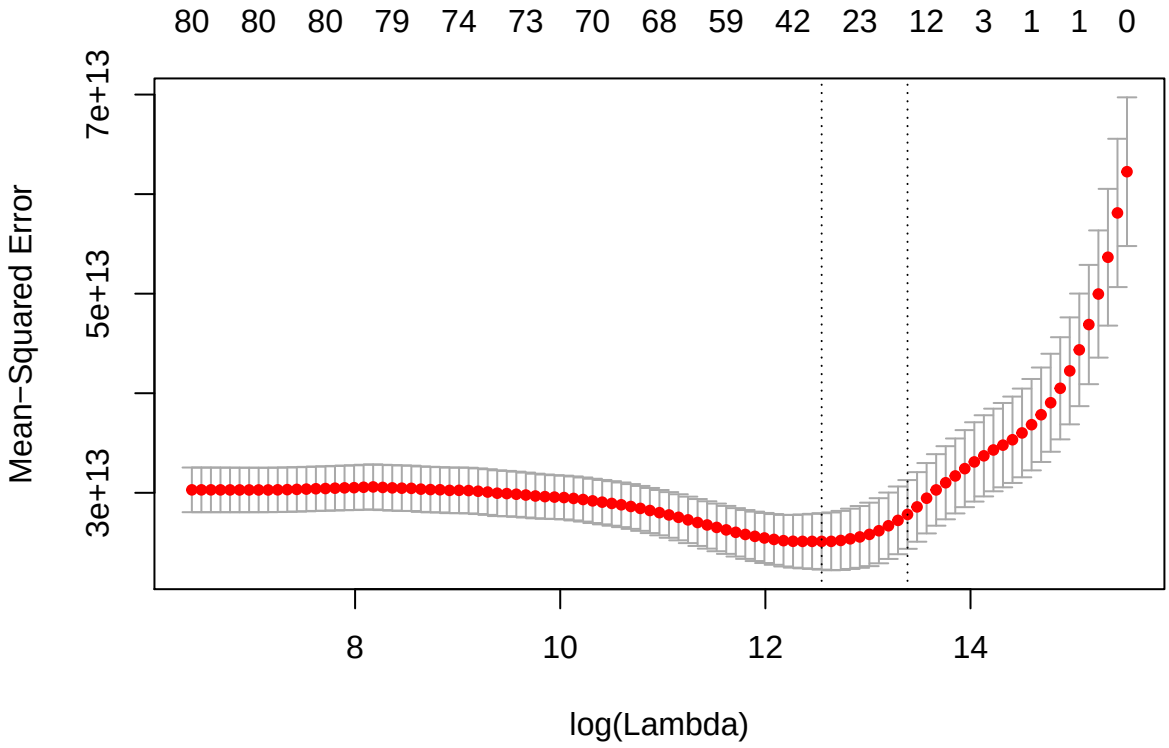
alpha of 1 is the best

```
set.seed(123)

cv_lm <- cv.glmnet(pc_dat, sal, alpha = 1)
```

```
plot(cv_lm$glmnet.fit, label = TRUE)
```




```
plot(cv_lm)
```




```
coef(cv_lm, s = cv_lm$lambda.min)
```

```
## 81 x 1 sparse Matrix of class "dgCMatrix"
##                          1
## (Intercept)  7.986379e+06
## PC1          1.135934e+06
## PC2         -2.040806e+05
## PC3          .
## PC4          .
## PC5          .
## PC6         -9.257954e+05
## PC7         -2.880867e+04
## PC8         -7.378798e+05
## PC9          .
## PC10         3.828081e+04
## PC11         .
## PC12         .
## PC13         2.569783e+05
## PC14         .
## PC15         .
## PC16         .
## PC17         .
## PC18         .
## PC19        -1.357362e+05
## PC20         .
## PC21         .
## PC22        -3.517527e+04
## PC23         .
## PC24         .
## PC25         .
```

```
## PC26         .
## PC27         5.921623e+05
## PC28         3.658381e+05
## PC29        -7.289587e+04
## PC30         .
## PC31        -8.143343e+04
## PC32        -2.579021e+05
## PC33         .
## PC34        -2.009806e+05
## PC35         .
## PC36         .
## PC37        -7.159848e+05
## PC38         .
## PC39         .
## PC40         .
## PC41        -5.722643e+05
## PC42         .
## PC43         1.938220e+05
## PC44         5.417600e+05
## PC45        -1.176844e+06
## PC46         1.055835e+06
## PC47         .
## PC48         .
## PC49         .
## PC50         .
## PC51        -1.782018e+06
## PC52        -6.089953e+04
## PC53         1.082469e+05
## PC54         .
## PC55         .
## PC56         .
## PC57         7.526151e+05
## PC58         .
## PC59         .
## PC60         1.197263e+06
## PC61         7.794888e+05
## PC62         .
## PC63         6.272157e+06
## PC64        -8.375874e+05
## PC65         .
## PC66        -5.200773e+04
## PC67         .
## PC68         .
## PC69         .
## PC70         9.171156e+06
## PC71         .
## PC72         1.790158e+06
## PC73         1.209739e+07
## PC74        -2.274982e+06
## PC75         .
## PC76         .
## PC77         .
## PC78         6.124860e+20
## PC79         2.417716e+20
```

```
## PC80          .
```

Constructing LM from the elastic-net regularization

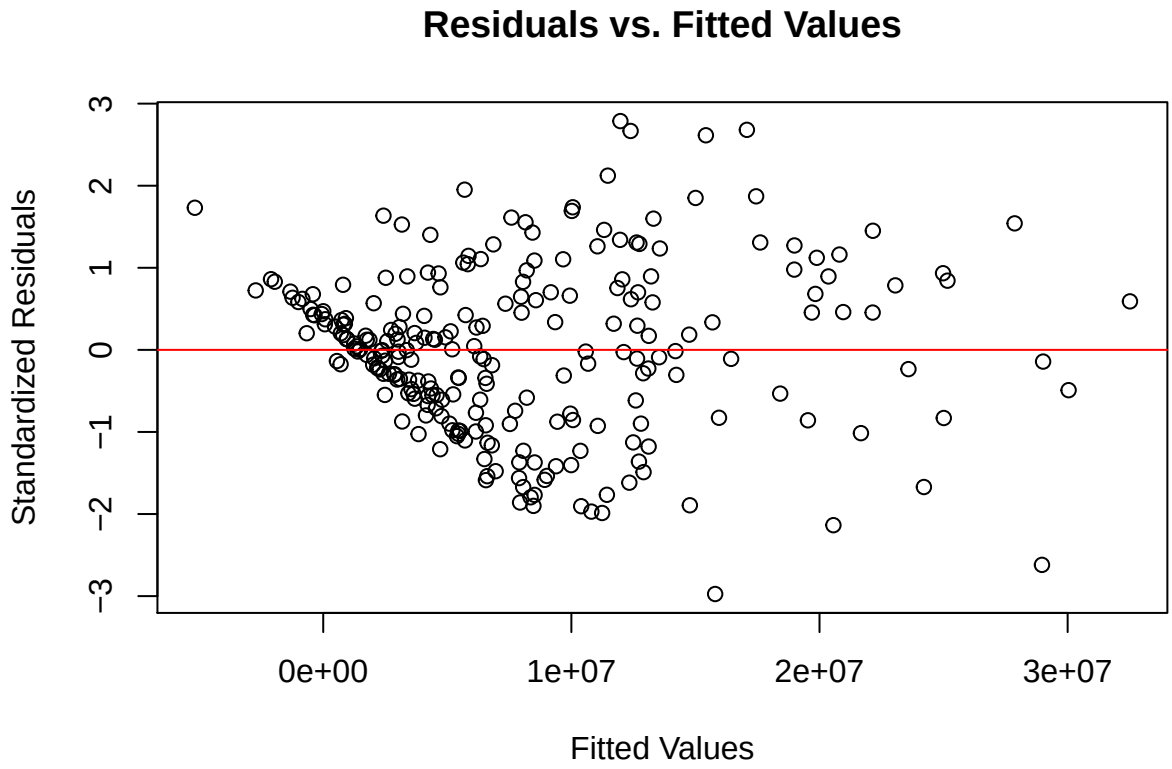
```
sal_lm_en_pc <- lm(sal ~ PC1 + PC2 + PC6 + PC7 + PC8 + PC10 + PC13 + PC19 + PC22 + PC27 + PC28 + PC29 +
```

```
summary(sal_lm_en_pc)
```

```
##
## Call:
## lm(formula = sal ~ PC1 + PC2 + PC6 + PC7 + PC8 + PC10 + PC13 +
##     PC19 + PC22 + PC27 + PC28 + PC29 + PC31 + PC32 + PC34 + PC37 +
##     PC41 + PC43 + PC44 + PC45 + PC46 + PC51 + PC52 + PC53 + PC57 +
##     PC60 + PC61 + PC63 + PC64 + PC66 + PC70 + PC72 + PC73 + PC74 +
##     PC78 + PC79, data = as.data.frame(pc_dat))
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -10794095  -2426784     28818   2531388  11137019
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  7.978e+06  2.687e+05  29.690  < 2e-16 ***
## PC1          1.210e+06  6.707e+04  18.047  < 2e-16 ***
## PC2         -2.900e+05  1.125e+05  -2.578 0.010606 *
## PC6         -1.119e+06  2.045e+05  -5.470 1.25e-07 ***
## PC7         -2.580e+05  2.133e+05  -1.210 0.227794
## PC8         -9.665e+05  2.239e+05  -4.317 2.41e-05 ***
## PC10         3.037e+05  2.359e+05   1.288 0.199204
## PC13         4.861e+05  2.612e+05   1.861 0.064111 .
## PC19        -4.456e+05  2.785e+05  -1.600 0.111053
## PC22        -3.753e+05  2.749e+05  -1.365 0.173593
## PC27         8.147e+05  2.821e+05   2.888 0.004273 **
## PC28         6.472e+05  2.724e+05   2.376 0.018384 *
## PC29        -3.905e+05  2.726e+05  -1.433 0.153382
## PC31        -3.533e+05  2.772e+05  -1.275 0.203808
## PC32        -5.397e+05  2.788e+05  -1.936 0.054197 .
## PC34        -4.758e+05  2.860e+05  -1.664 0.097589 .
## PC37        -1.023e+06  3.133e+05  -3.265 0.001275 **
## PC41        -9.855e+05  3.724e+05  -2.647 0.008734 **
## PC43         5.865e+05  4.100e+05   1.430 0.154069
## PC44         9.711e+05  4.204e+05   2.310 0.021833 *
## PC45        -1.731e+06  4.796e+05  -3.609 0.000382 ***
## PC46         1.533e+06  4.860e+05   3.154 0.001841 **
## PC51        -2.524e+06  7.524e+05  -3.354 0.000943 ***
## PC52        -9.504e+05  7.980e+05  -1.191 0.234971
## PC53         1.005e+06  8.544e+05   1.176 0.240768
## PC57         1.930e+06  1.161e+06   1.662 0.097921 .
## PC60         2.830e+06  1.625e+06   1.742 0.083008 .
## PC61         2.595e+06  1.788e+06   1.451 0.148146
## PC63         8.766e+06  2.312e+06   3.792 0.000194 ***
## PC64        -3.413e+06  2.603e+06  -1.311 0.191182
## PC66        -3.847e+06  3.509e+06  -1.096 0.274218
```

```
## PC70          1.604e+07  6.573e+06   2.440 0.015496 *
## PC72          1.778e+07  1.493e+07   1.191 0.234906
## PC73          3.853e+07  2.623e+07   1.469 0.143272
## PC74         -3.103e+07  2.895e+07  -1.072 0.285033
## PC78          5.269e+20  3.954e+20   1.333 0.184070
## PC79          4.095e+20  2.405e+20   1.703 0.090082 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4252000 on 214 degrees of freedom
## Multiple R-squared:  0.7508, Adjusted R-squared:  0.7089
## F-statistic: 17.91 on 36 and 214 DF,  p-value: < 2.2e-16
```

```
plot(sal_lm_en_pc$fitted.values, rstandard(sal_lm_en_pc), main = "Residuals vs. Fitted Values", xlab =
abline(h = 0, col = "red")
```



Cluster Analysis