group-project

Zhaoyang Wang

5/4/2022

R Markdown

```r
# ......preprocessing the data
`NBA.Stats.2020.2021(1)` <- read.csv("~/Desktop/Rworkplace/shiyan/NBA.Stats.2020.2021(1).csv")
data1 <- `NBA.Stats.2020.2021(1)`
str(data1)
```

```
## 'data.frame':    311 obs. of  30 variables:
##  $ FULL.NAME: chr  "Dennis Smith Jr." "Brandon Ingram" "Ivica Zubac" "Jamal Murray" ...
##  $ TEAM     : chr  "Nyk" "Nor" "Lac" "Den" ...
##  $ POS      : chr  "G" "F" "C" "G" ...
##  $ MAIN.POS : chr  "G" "F" "C" "G" ...
##  $ AGE      : num  23.5 23.7 24.2 24.2 24.2 ...
##  $ GP       : int  23 61 72 48 58 62 7 37 67 58 ...
##  $ MPG      : num  18.2 34.3 22.3 35.5 22.7 ...
##  $ MIN.     : num  38.1 71.5 46.5 73.9 47.3 ...
##  $ USG.     : num  19.2 28 15.2 24.7 12.3 ...
##  $ TO.      : num  13.8 11 14.7 11.1 8.5 ...
##  $ FTA      : int  26 320 171 153 91 218 0 80 392 250 ...
##  $ FT.      : num  0.717 0.878 0.789 0.869 0.648 ...
##  $ X2PA     : int  86 726 390 476 168 451 8 279 916 701 ...
##  $ X2P.     : num  0.446 0.51 0.656 0.523 0.619 ...
##  $ X3PA     : int  59 375 4 316 136 257 4 321 371 410 ...
##  $ X3P.     : num  0.3 0.381 0.25 0.408 0.316 0.3 0.5 0.399 0.34 0.398 ...
##  $ eFG.     : num  0.448 0.531 0.654 0.559 0.554 ...
##  $ TS.      : num  0.485 0.584 0.693 0.592 0.576 ...
##  $ PPG      : num  6.74 23.8 9 21.2 6.8 ...
##  $ RPG      : num  2.44 4.9 7.2 4 3.5 ...
##  $ TRB.     : num  7.03 7.8 18.1 6.5 8.3 ...
##  $ APG      : num  3.35 4.9 1.3 4.8 0.8 ...
##  $ AST.     : num  26.9 22.4 7.9 20.3 4.8 ...
##  $ SPG      : num  1 0.69 0.33 1.33 0.64 0.85 0.14 0.81 0.79 1.24 ...
##  $ BPG.     : num  0.609 0.59 0.86 0.27 0.93 ...
##  $ TOPG     : num  1.13 2.51 1.13 2.23 0.55 2.29 0.14 1.62 3.09 2.72 ...
##  $ VI       : num  8.12 9.7 7.9 8.4 4.7 ...
##  $ ORTG     : num  103 115 134 116 120 ...
##  $ DRTG     : num  105 111 101 111 110 ...
##  $ Salary   : int  5686677 27285000 7000000 27285000 9258000 4767000 449115 13425895 29467800 250351
```

```r
data1$MAIN.POS <- as.factor(data1$MAIN.POS)
sum(is.na(data1))
```

```
## [1] 0
```

```r
str(data1$MAIN.POS)
```

```
##  Factor w/ 3 levels "C","F","G": 3 2 1 3 2 3 3 3 3 3 ...
```

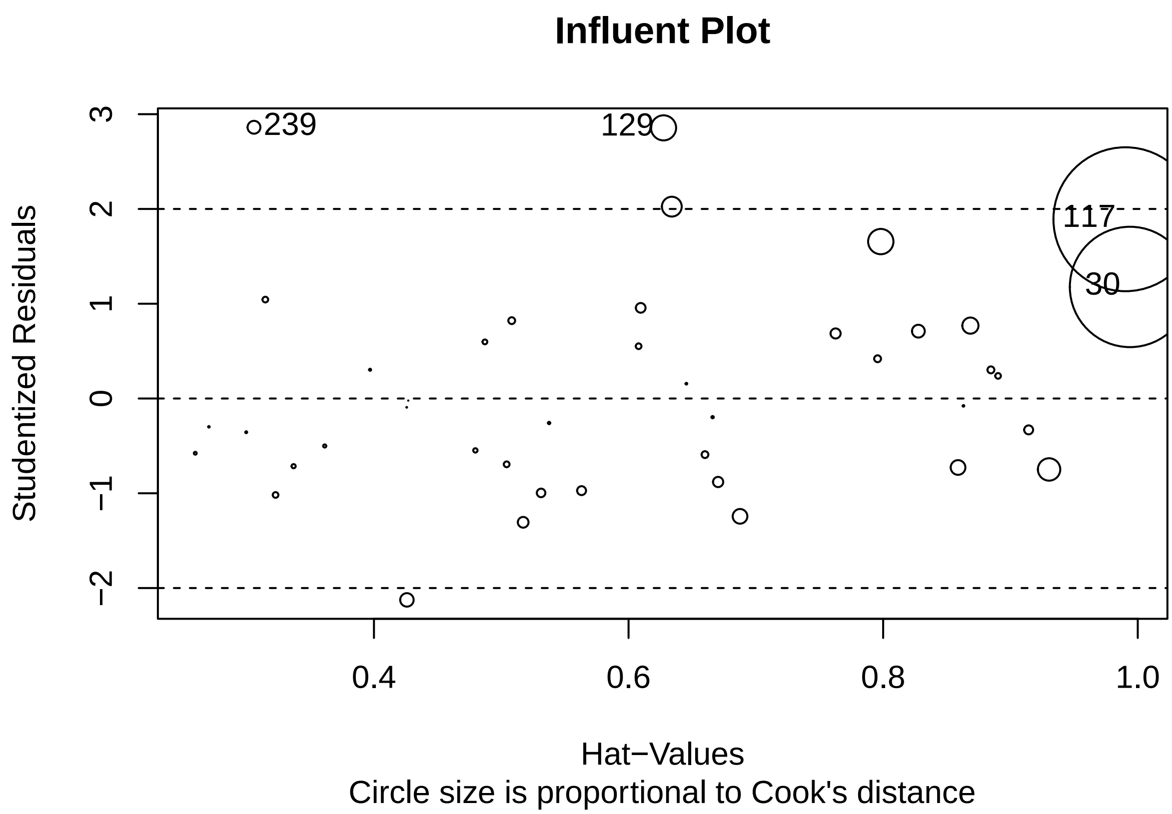
```r
# ...Classified by player's game role
main.pos.c <- subset(data1, MAIN.POS == "C")
main.pos.g <- subset(data1, MAIN.POS == "G")
main.pos.f <- subset(data1, MAIN.POS == "F")

# ...Among the players in the 'C POS', look for strong influence points
library(car)
```

```
## Loading required package: carData
```

```r
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following object is masked from 'package:car':
##
##     recode

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

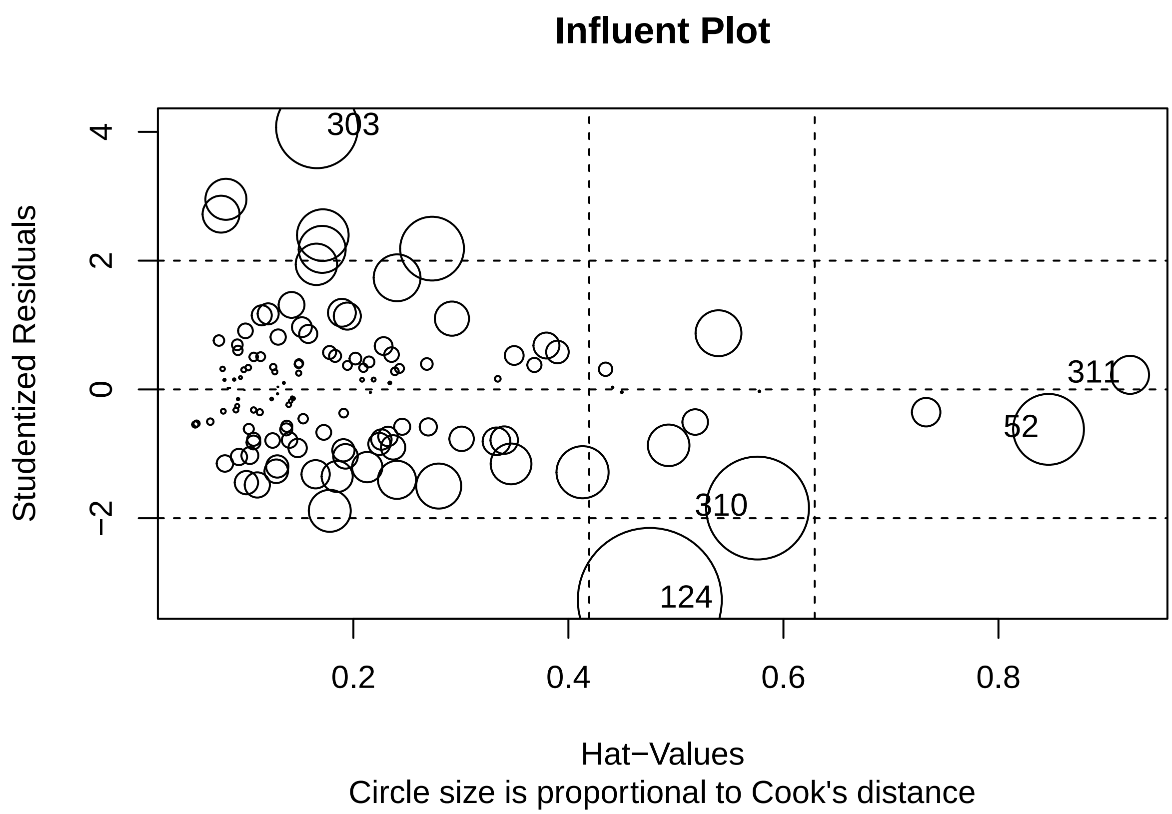
```r
options(warn = -1)
c <- subset(main.pos.c, select = AGE:Salary)
fit.c <- lm(Salary ~ ., data = c)

c.influenceplot = influencePlot(fit.c, id.method = "identify", main = "Influent Plot",
    sub = "Circle size is proportional to Cook's distance")
```




```
c.influenceplot
```

```
##       StudRes       Hat       CookD
## 30  1.177209 0.9939410  8.54960233
## 117 1.891659 0.9903180 12.22341228
## 129 2.855723 0.6274996  0.37186452
## 239 2.861920 0.3057653  0.09750479
```

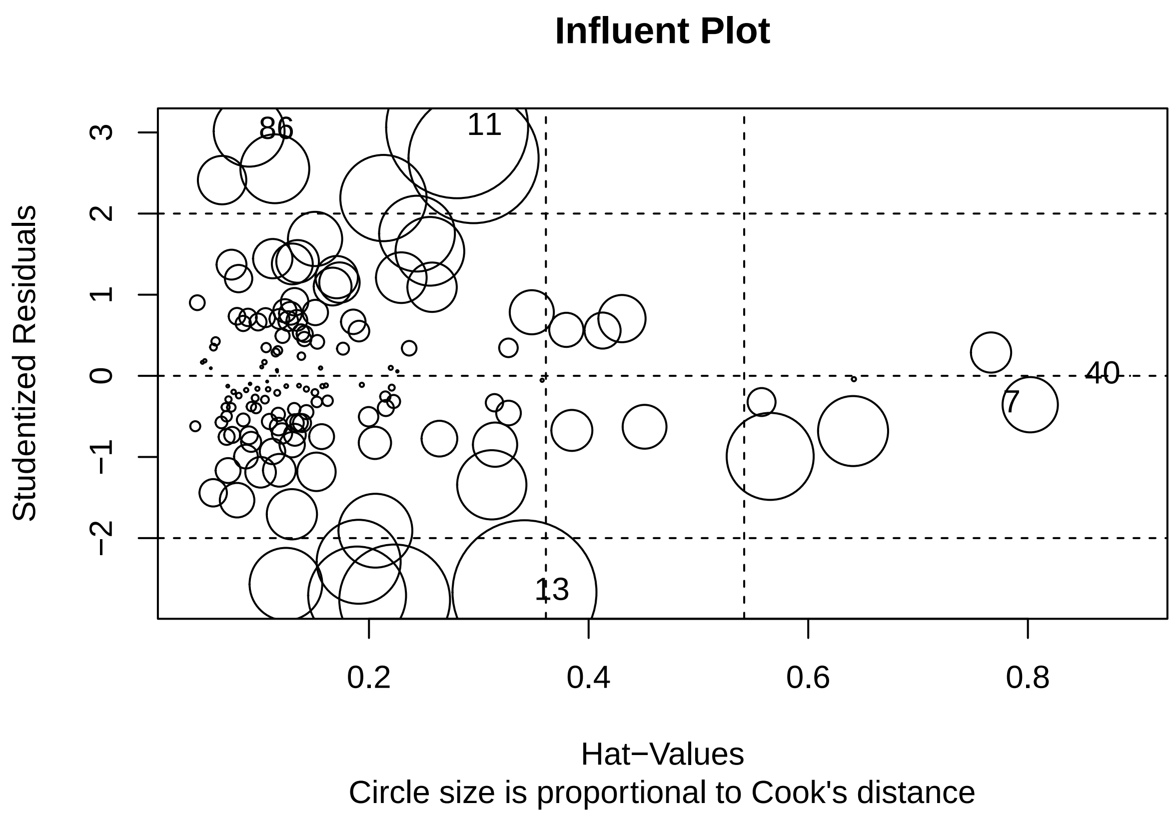
```
main.pos.c.filtered <- filter(main.pos.c, rownames(main.pos.c) != "30" & rownames(main.pos.c) !=
    "117" & rownames(main.pos.c) != "129" & rownames(main.pos.c) != "239")



# ...Among the players in the 'F POS', look for strong influence points
f <- subset(main.pos.f, select = AGE:Salary)
fit.f <- lm(Salary ~ ., data = f)

f.influenceplot = influencePlot(fit.f, id.method = "identify", main = "Influent Plot",
    sub = "Circle size is proportional to Cook's distance")
```




```
f.influenceplot
```

```
##         StudRes       Hat       CookD
## 52  -0.6199317 0.8465937 0.08208868
## 124 -3.2682747 0.4756924 0.33922532
## 303  4.0710638 0.1661399 0.10959031
## 310 -1.8424256 0.5759303 0.17308336
## 311  0.2264782 0.9224069 0.02368120
```

```
main.pos.f.filtered <- filter(main.pos.f, rownames(main.pos.f) != "52" & rownames(main.pos.f) !=
    "124" & rownames(main.pos.f) != "303" & rownames(main.pos.f) != "310" & rownames(main.pos.f) !=
    "311")



# ...Among the players in the 'G POS', look for strong influence points
g <- subset(main.pos.g, select = AGE:Salary)
fit.g <- lm(Salary ~ ., data = g)

g.influenceplot = influencePlot(fit.g, id.method = "identify", main = "Influent Plot",
    sub = "Circle size is proportional to Cook's distance")
```

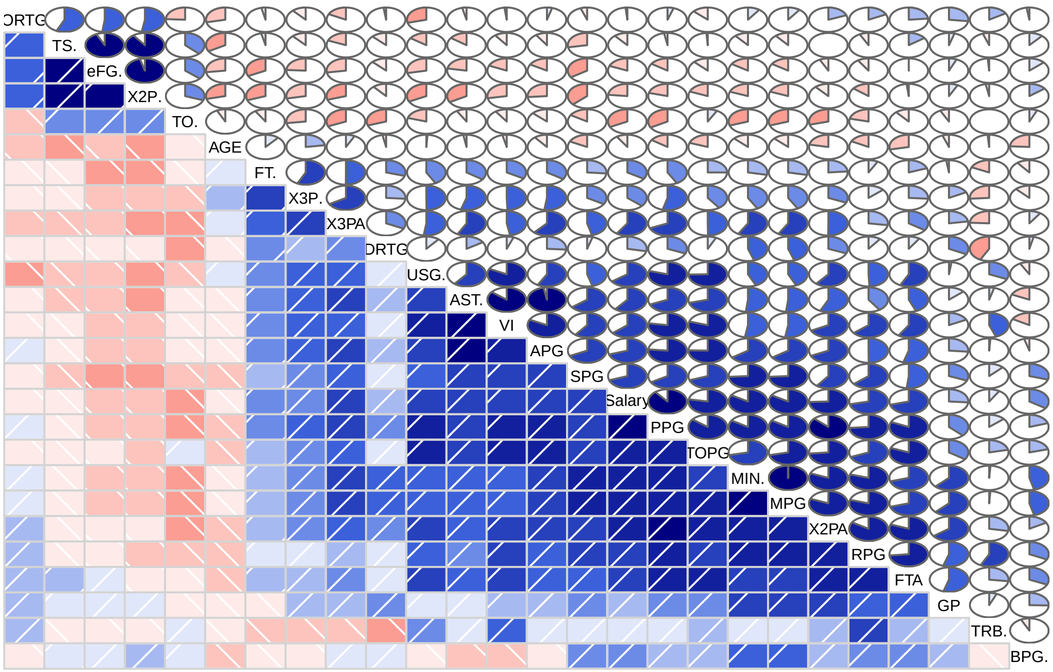



```
g.influenceplot
```

```
##           StudRes        Hat        CookD
## 7  -0.3559977154 0.80212158 1.990622e-02
## 11  3.0635698206 0.28028991 1.312553e-01
## 13 -2.6672889829 0.34159594 1.349729e-01
## 40  0.0004212965 0.89300507 5.746311e-08
## 86  3.0152290552 0.09085789 3.270348e-02
```

```
main.pos.g.filtered <- filter(main.pos.g, rownames(main.pos.g) != "7" & rownames(main.pos.g) !=
    "11" & rownames(main.pos.g) != "13" & rownames(main.pos.g) != "40" & rownames(main.pos.g) !=
    "86")



# ...Build training and test sets for each class of players, with the training
# set being two-thirds of the dataset

library(caTools)
split.c <- sample.split(main.pos.c.filtered$MAIN.POS, SplitRatio = 2/3)
main.pos.c.train <- subset(main.pos.c.filtered, split.c == TRUE)
main.pos.c.test <- subset(main.pos.c.filtered, split.c == FALSE)

split.f <- sample.split(main.pos.f.filtered$MAIN.POS, SplitRatio = 2/3)
main.pos.f.train <- subset(main.pos.f.filtered, split.f == TRUE)
main.pos.f.test <- subset(main.pos.f.filtered, split.f == FALSE)

split.g <- sample.split(main.pos.g.filtered$MAIN.POS, SplitRatio = 2/3)
main.pos.g.train <- subset(main.pos.g.filtered, split.g == TRUE)
```

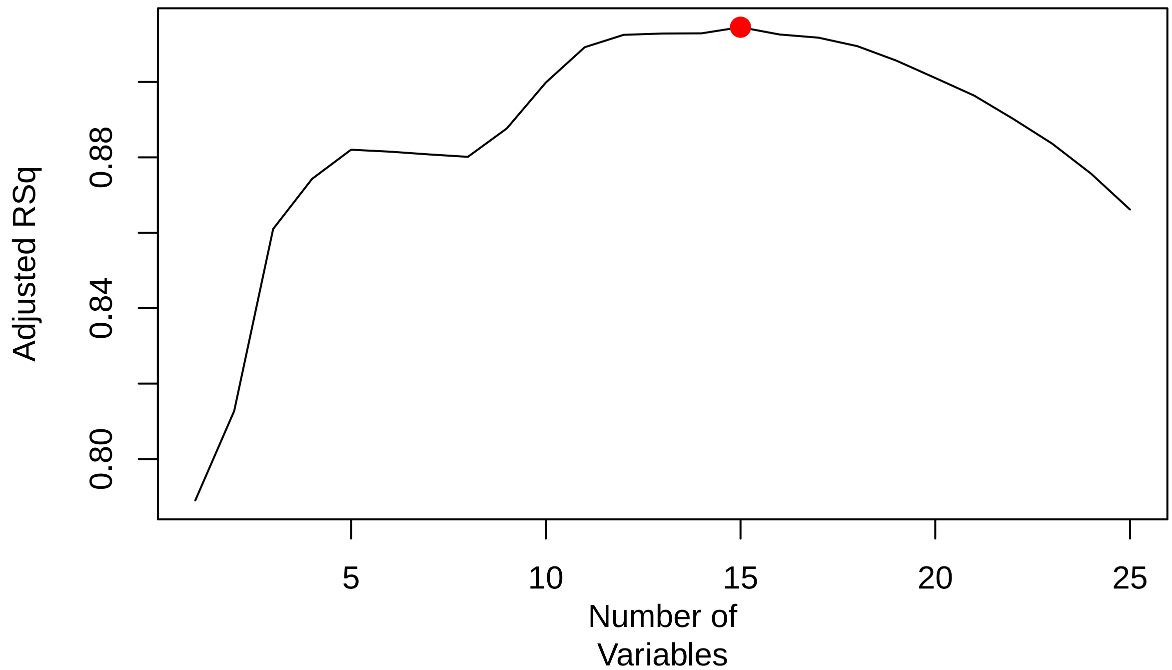
```
main.pos.g.test <- subset(main.pos.g.filtered, split.g == FALSE)

# ...visualize the data
library(leaps)
library(corrgram)
corrgram(main.pos.c.filtered, order = TRUE, lower.panel = panel.shade, upper.panel = panel.pie,
    text.panel = panel.txt)
```



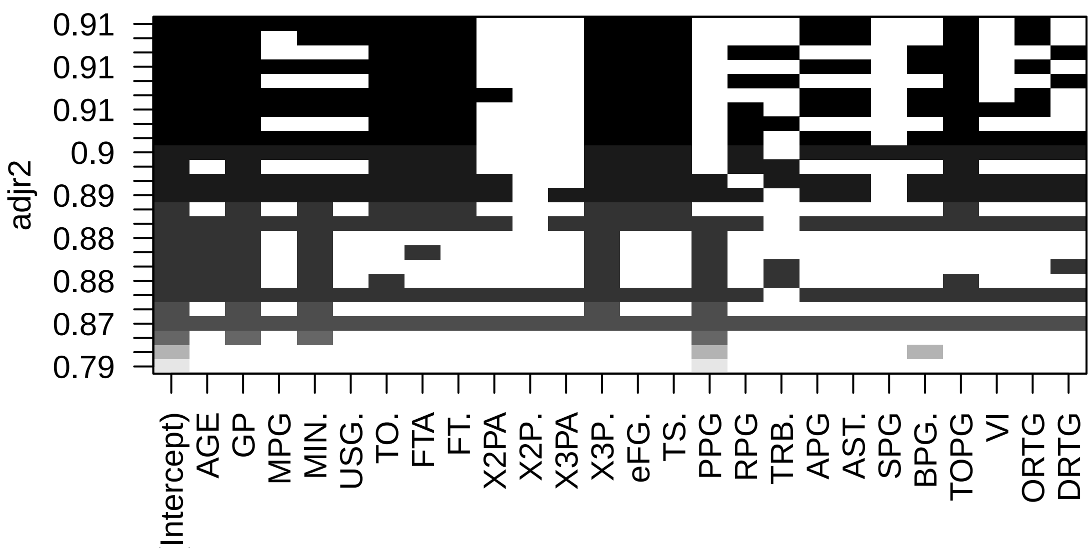

```
# ......Variable subsetting using regsubsets ...regsubsetting of POS 'C'
c.fit <- main.pos.c.filtered[, 5:30]
regfit.full.c = regsubsets(Salary ~ ., data = c.fit, nvmax = 25)

reg.summary.c = summary(regfit.full.c)

par(mfrow = c(1, 1))

plot(reg.summary.c$adjr2, xlab = "Number of
Variables", ylab = "Adjusted RSq", type = "l")
which.max(reg.summary.c$adjr2)
```

```
## [1] 15
```

```
points(15, reg.summary.c$adjr2[15], col = "red", cex = 2, pch = 20)
```



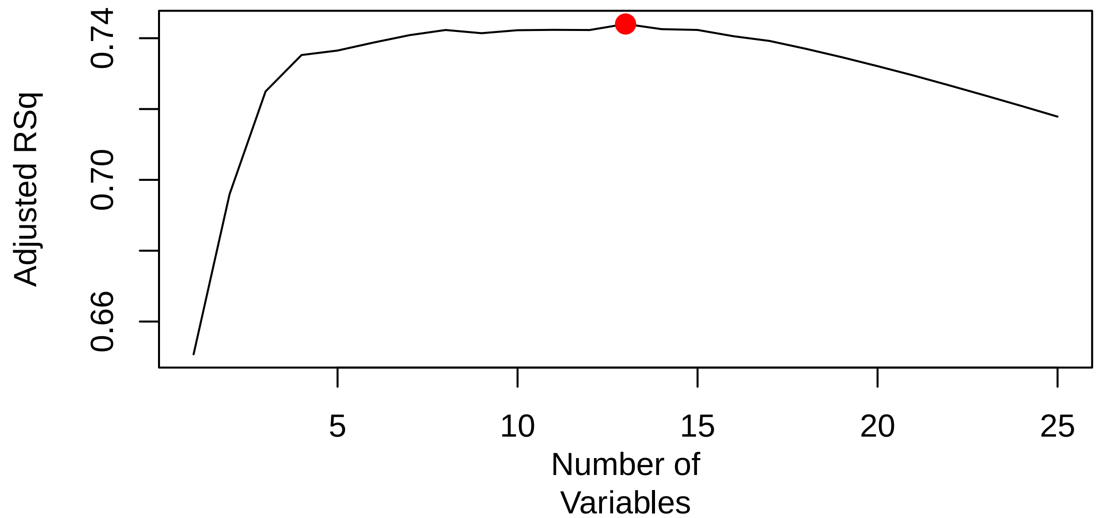

```
# best varible number is 15, and the name of the varibles are in the
# #reg.summary.c$which
plot(regfit.full.c, scale = "adjr2")
```



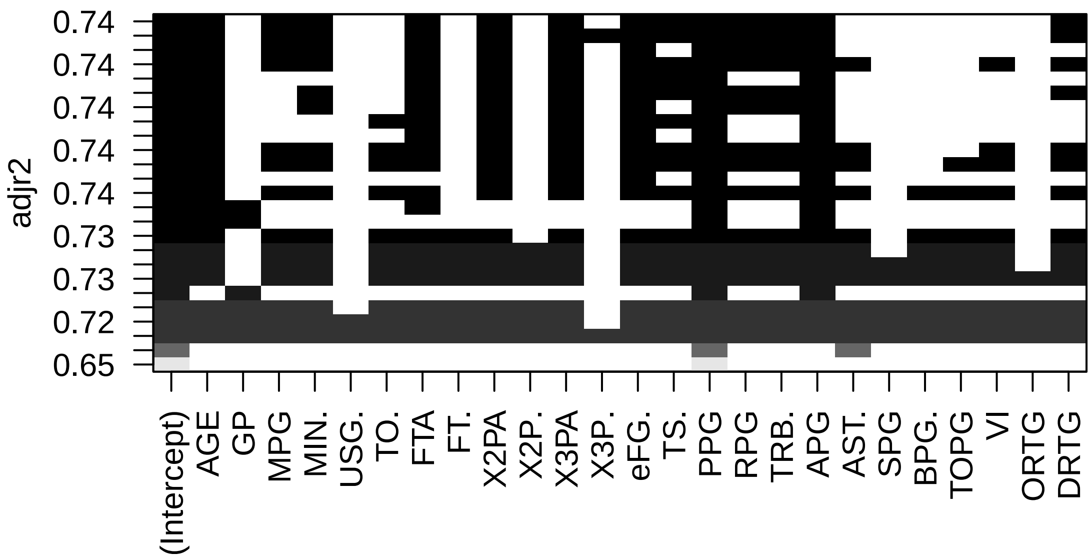

```
# ...regsubsetting of POS 'F'
f.fit <- main.pos.f.filtered[, 5:30]
regfit.full.f = regsubsets(Salary ~ ., data = f.fit, nvmax = 25)

reg.summary.f = summary(regfit.full.f)

par(mfrow = c(1, 1))

plot(reg.summary.f$adjr2, xlab = "Number of
Variables", ylab = "Adjusted RSq", type = "l")
which.max(reg.summary.f$adjr2)
```

```
## [1] 13
```

```
points(13, reg.summary.f$adjr2[13], col = "red", cex = 2, pch = 20)
```



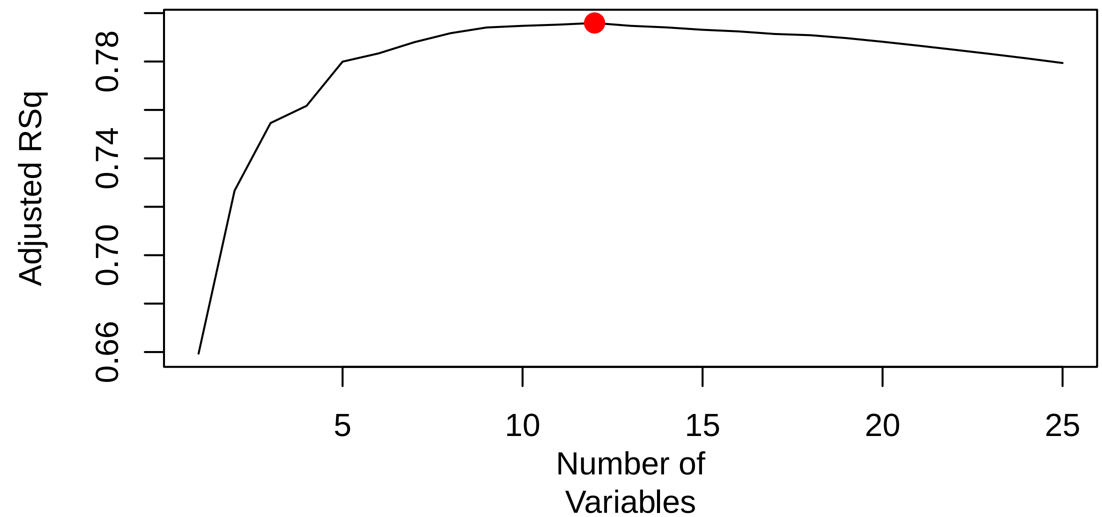

```
# best varible number is 13, and the name of the varibles are in the
# #reg.summary.f$which
plot(regfit.full.f, scale = "adjr2")
```



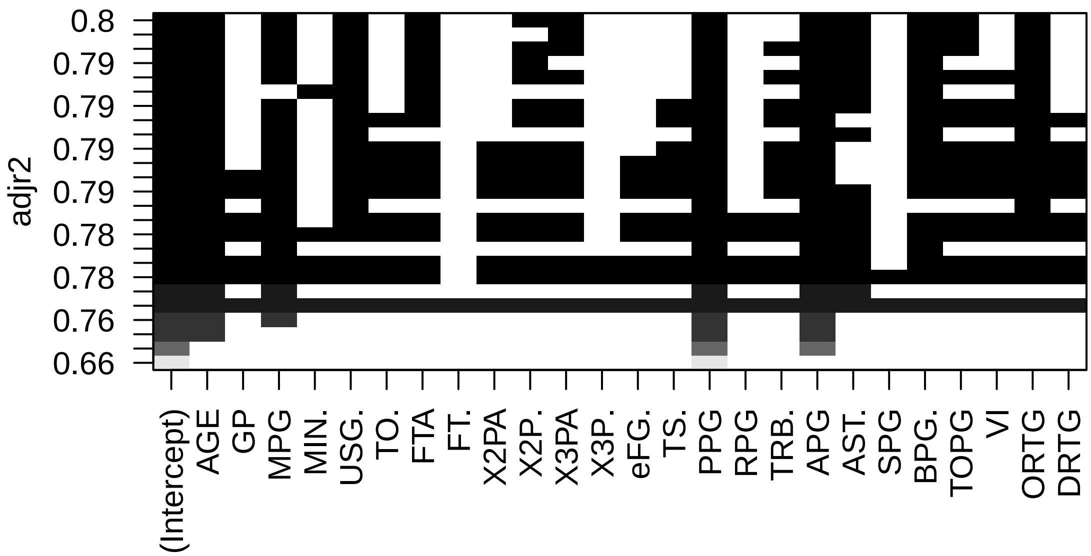

```
# ...regsubsetting of POS 'G'
g.fit <- main.pos.g.filtered[, 5:30]
regfit.full.g = regsubsets(Salary ~ ., data = g.fit, nvmax = 25)

reg.summary.g = summary(regfit.full.g)

par(mfrow = c(1, 1))

plot(reg.summary.g$adjr2, xlab = "Number of
Variables", ylab = "Adjusted RSq", type = "l")
which.max(reg.summary.g$adjr2)
```

```
## [1] 12
```

```r
points(12, reg.summary.g$adjr2[12], col = "red", cex = 2, pch = 20)
```




```r
# best varible number is 12, and the name of the varibles are in the
# #reg.summary.g$which
plot(regfit.full.g, scale = "adjr2")
```




```r
# ......linear regression ...linear regression for POS 'C'(the variables based
# on the regsubsetting #results of POS 'C')
library(dplyr)
c.fit.train = main.pos.c.train[, 5:30]
c.fit.test = main.pos.c.test[, 5:30]
reg.c.variables <- data.frame(reg.summary.c$which)
reg.c.variables <- reg.c.variables[, -1]
a <- which(reg.c.variables[15, ] == TRUE)
a. <- colnames(c.fit.train)[a]

final.c.reg <- select(c.fit.train, col = a)
```

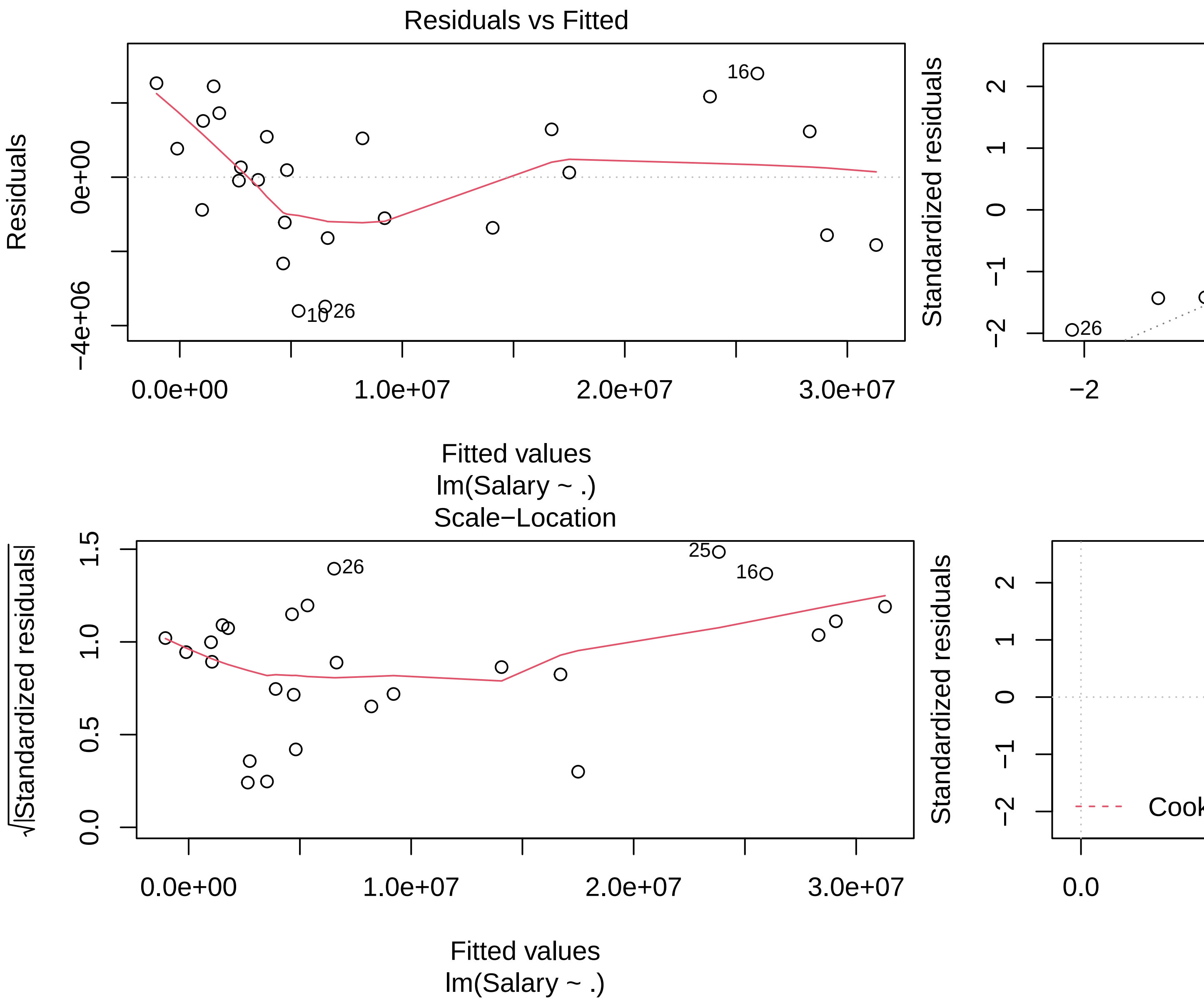
```
## Note: Using an external vector in selections is ambiguous.
## i Use `all_of(a)` instead of `a` to silence this message.
```

```
## i See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.
## This message is displayed once per session.
```

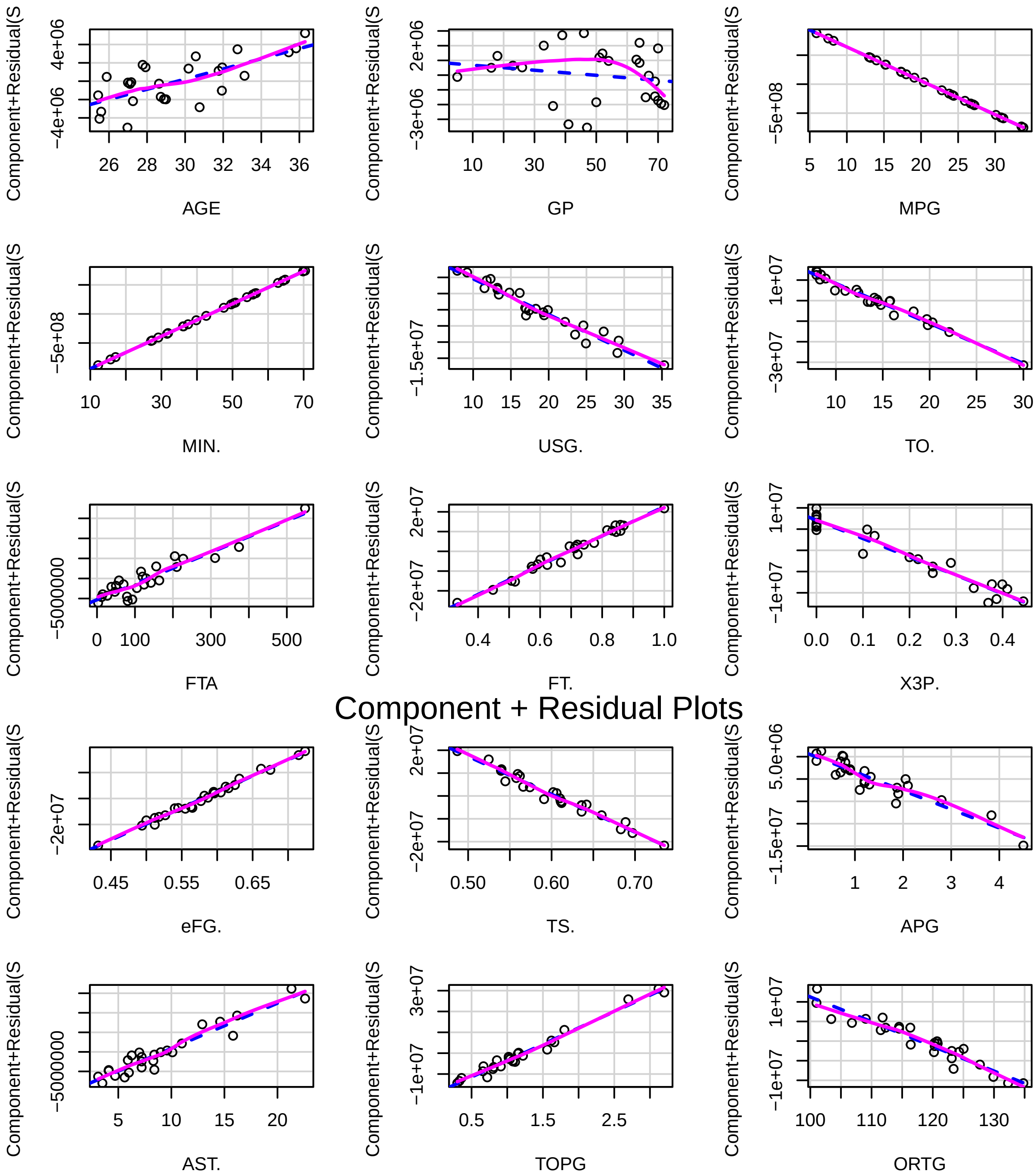
```
colnames(final.c.reg) <- a.
final.c.reg <- cbind(final.c.reg, c.fit.train[, 26])
colnames(final.c.reg)[16] <- "Salary"

final.c.reg.test <- select(c.fit.test, col = a)
colnames(final.c.reg.test) <- a.
final.c.reg.test <- cbind(final.c.reg.test, c.fit.test[, 26])
colnames(final.c.reg.test)[16] <- "Salary"

# linear fit
c.linearfit <- lm(Salary ~ ., data = final.c.reg)
summary(c.linearfit)
```

```
##
## Call:
## lm(formula = Salary ~ ., data = final.c.reg)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3604393 -1328958   156843  1274144  2793485
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)    2074618   23961404   0.087  0.93271
## AGE             554762     315619   1.758  0.10931
## GP              -17204      69048  -0.249  0.80828
## MPG          -58351701   34096177  -1.711  0.11779
## MIN.          27949561   16329528   1.712  0.11775
## USG.          -1104184     434165  -2.543  0.02920 *
## TO.           -1944864     622028  -3.127  0.01075 *
## FTA              39365      18600   2.116  0.06039 .
## FT.           74154018   16327683   4.542  0.00107 **
## X3P.         -43275945   11180901  -3.871  0.00311 **
## eFG.         247167123   75565179   3.271  0.00842 **
## TS.         -165586750   96596341  -1.714  0.11726
## APG           -4185598    4160753  -1.006  0.33815
## AST.           1157372     765045   1.513  0.16127
## TOPG          15698202    4956541   3.167  0.01004 *
## ORTG           -629184     245295  -2.565  0.02813 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2842000 on 10 degrees of freedom
## Multiple R-squared:  0.9694, Adjusted R-squared:  0.9235
## F-statistic: 21.11 on 15 and 10 DF,  p-value: 1.384e-05
```

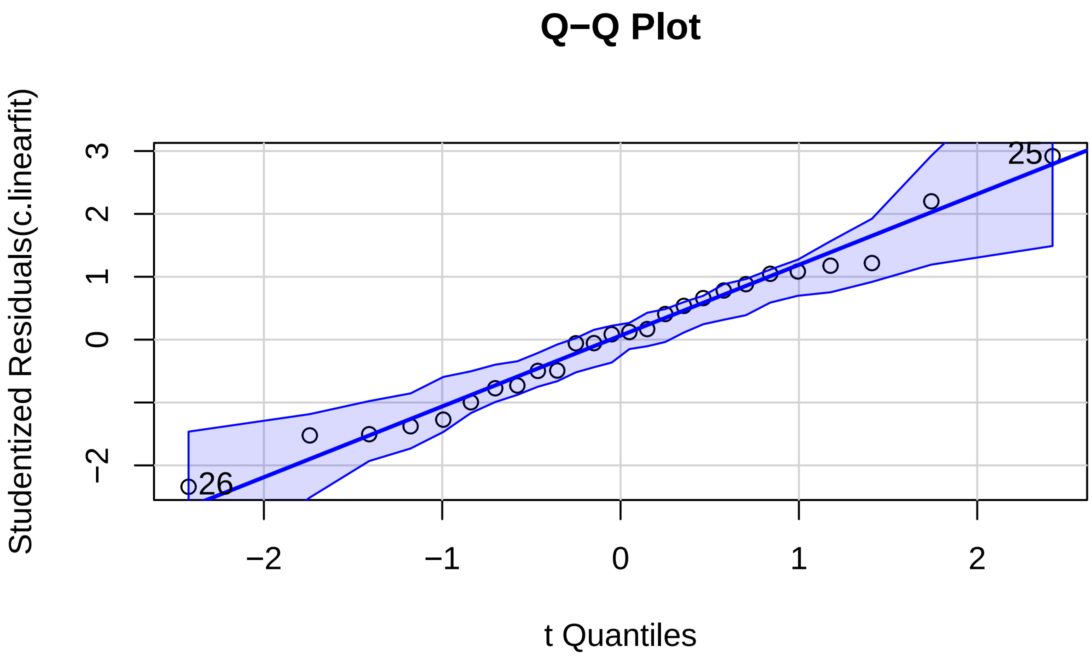
```
plot(c.linearfit)
```




```
library(car)
crPlots(c.linearfit)
```




```
qqPlot(c.linearfit, id.method = "identify", simulate = TRUE, main = "Q-Q Plot")
```

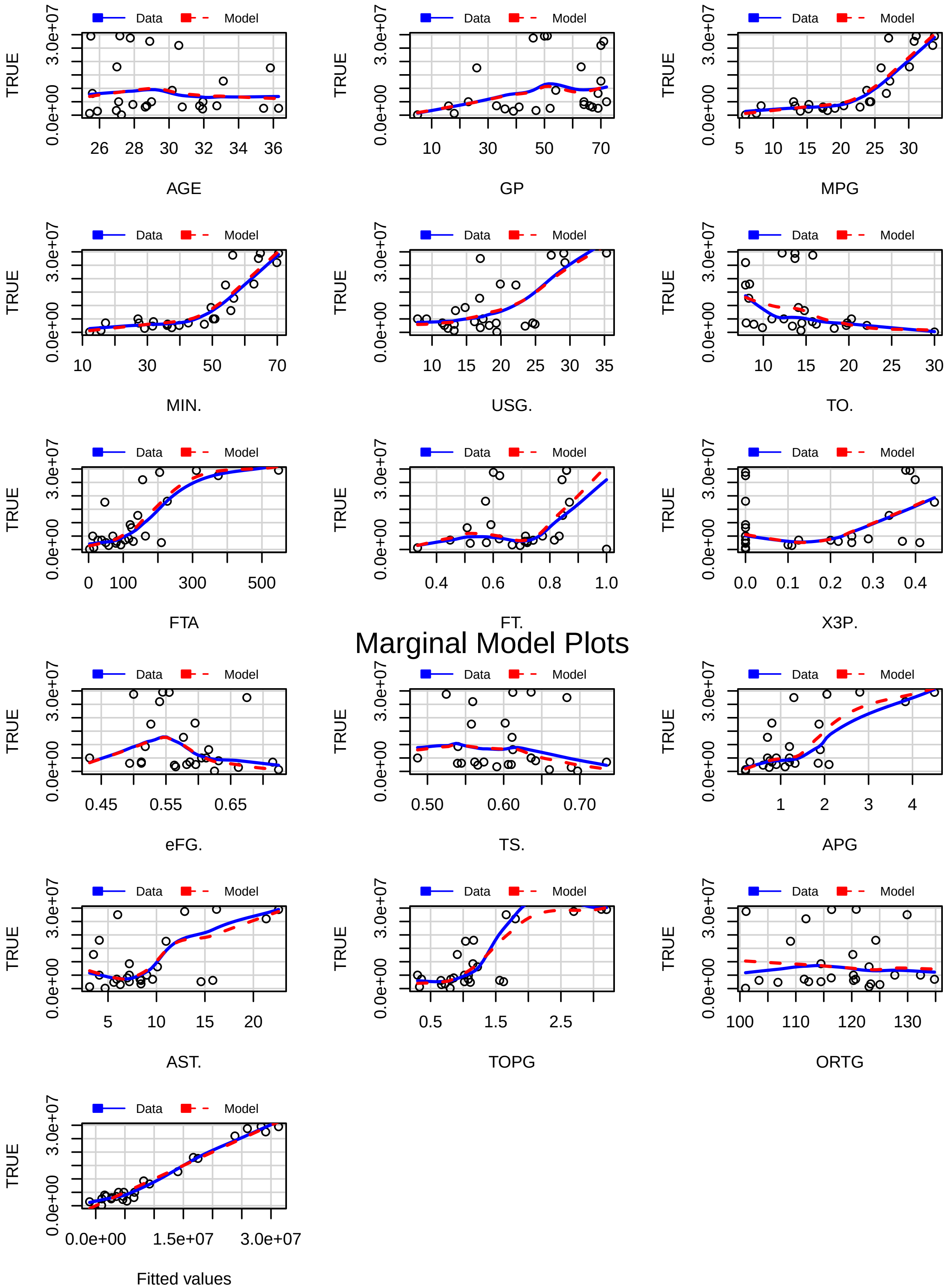



```
## 25 26
## 17 18
```

```r
ncvTest(c.linearfit)
```

```
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 0.02333897, Df = 1, p = 0.87858
```

```r
marginalModelPlots(c.linearfit)
```




```
# the accuracy of linear model prediction
pre.c <- predict(c.linearfit, newdata = final.c.reg.test)
```

```r
mean((pre.c - c.fit.test$Salary)^2)
```

```
## [1] 2.512578e+13
```

```r
# check the collinearity using vif function
vif(c.linearfit)
```

```
##          AGE           GP          MPG         MIN.         USG.          TO.
## 3.237944e+00 5.775209e+00 2.333313e+05 2.323715e+05 2.679513e+01 3.294018e+01
##          FTA          FT.         X3P.         eFG.          TS.          APG
## 1.634610e+01 1.925958e+01 1.085411e+01 8.374403e+01 1.008352e+02 5.808678e+01
##         AST.         TOPG         ORTG
## 4.855407e+01 4.414166e+01 1.533703e+01
```

```r
# it's bigger than 4, collinearity exsits, so delete the varible with the
# #biggest vif-value and build a new linear model


# new linear model and predict again
c.linearfit1 <- lm(Salary ~ AGE + GP + MIN. + USG. + TO. + FTA + FT. + X3P. + eFG. +
    TS. + APG + AST. + TOPG + ORTG, data = final.c.reg)
summary(c.linearfit1)
```

```
##
## Call:
## lm(formula = Salary ~ AGE + GP + MIN. + USG. + TO. + FTA + FT. +
##     X3P. + eFG. + TS. + APG + AST. + TOPG + ORTG, data = final.c.reg)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -4322297  -763457   -79829   941366  3586486
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   18550550   23788431   0.780  0.45195
## AGE             354980     317908   1.117  0.28796
## GP              -61487      69402  -0.886  0.39460
## MIN.              4727     169650   0.028  0.97827
## USG.           -960127     461763  -2.079  0.06178 .
## TO.           -1598705     637713  -2.507  0.02914 *
## FTA              57382      16625   3.452  0.00541 **
## FT.           67025661   17115741   3.916  0.00241 **
## X3P.         -34281651   10699024  -3.204  0.00839 **
## eFG.         267740856   80879503   3.310  0.00695 **
## TS.         -216202758   99693943  -2.169  0.05290 .
## APG           -4230880    4510728  -0.938  0.36840
## AST.           1197072     829031   1.444  0.17663
## TOPG          11243584    4572950   2.459  0.03175 *
## ORTG           -576078     263796  -2.184  0.05152 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3081000 on 11 degrees of freedom
## Multiple R-squared:  0.9604, Adjusted R-squared:  0.9101
## F-statistic: 19.07 on 14 and 11 DF,  p-value: 1.039e-05
```

```r
vif(c.linearfit1)
```

```
##       AGE        GP      MIN.      USG.       TO.       FTA       FT.      X3P.
##  2.794986  4.964154 21.339157 25.787917 29.457104 11.109903 18.006204  8.455946
##      eFG.       TS.       APG      AST.      TOPG      ORTG
## 81.624495 91.382162 58.084433 48.509434 31.968126 15.091591
```

```r
pre.c <- predict(c.linearfit1, newdata = final.c.reg.test)
mean((pre.c - c.fit.test$Salary)^2)
```

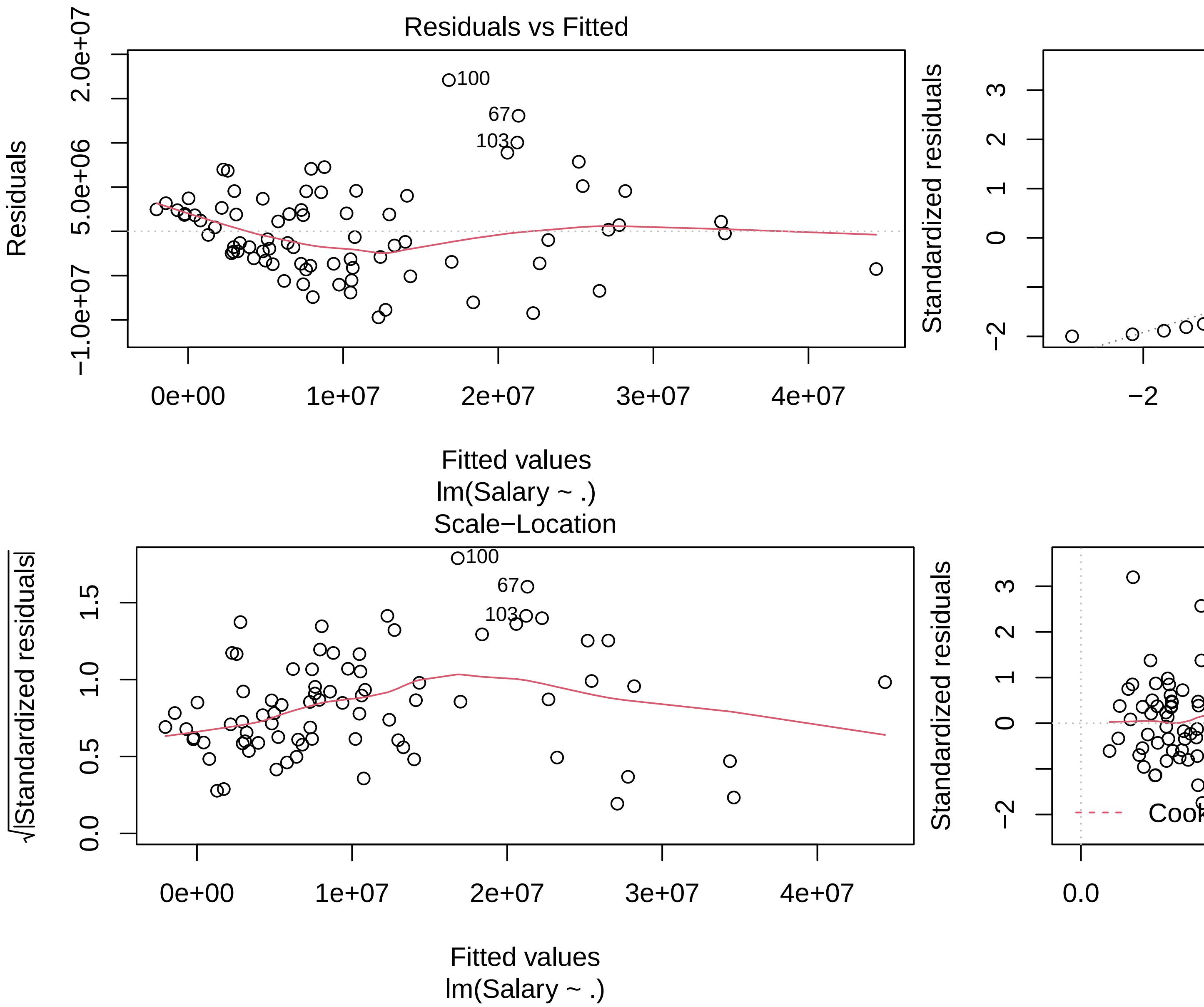
```
## [1] 1.8899e+13
```

```r
# ...linear regression for POS 'F'(the variables based on the regsubsetting
# #results of POS 'F')
library(dplyr)
f.fit.train = main.pos.f.train[, 5:30]
f.fit.test = main.pos.f.test[, 5:30]
reg.f.variables <- data.frame(reg.summary.f$which)
reg.f.variables <- reg.f.variables[, -1]
a <- which(reg.f.variables[13, ] == TRUE)
a. <- colnames(f.fit.train)[a]
final.f.reg <- select(f.fit.train, col = a)
colnames(final.f.reg) <- a.
final.f.reg <- cbind(final.f.reg, f.fit.train[, 26])
colnames(final.f.reg)[14] <- "Salary"

final.f.reg.test <- select(f.fit.test, col = a)
colnames(final.f.reg.test) <- a.
final.f.reg.test <- cbind(final.f.reg.test, f.fit.test[, 26])
colnames(final.f.reg.test)[14] <- "Salary"

# linear fit
f.linearfit <- lm(Salary ~ ., data = final.f.reg)
summary(f.linearfit)
```

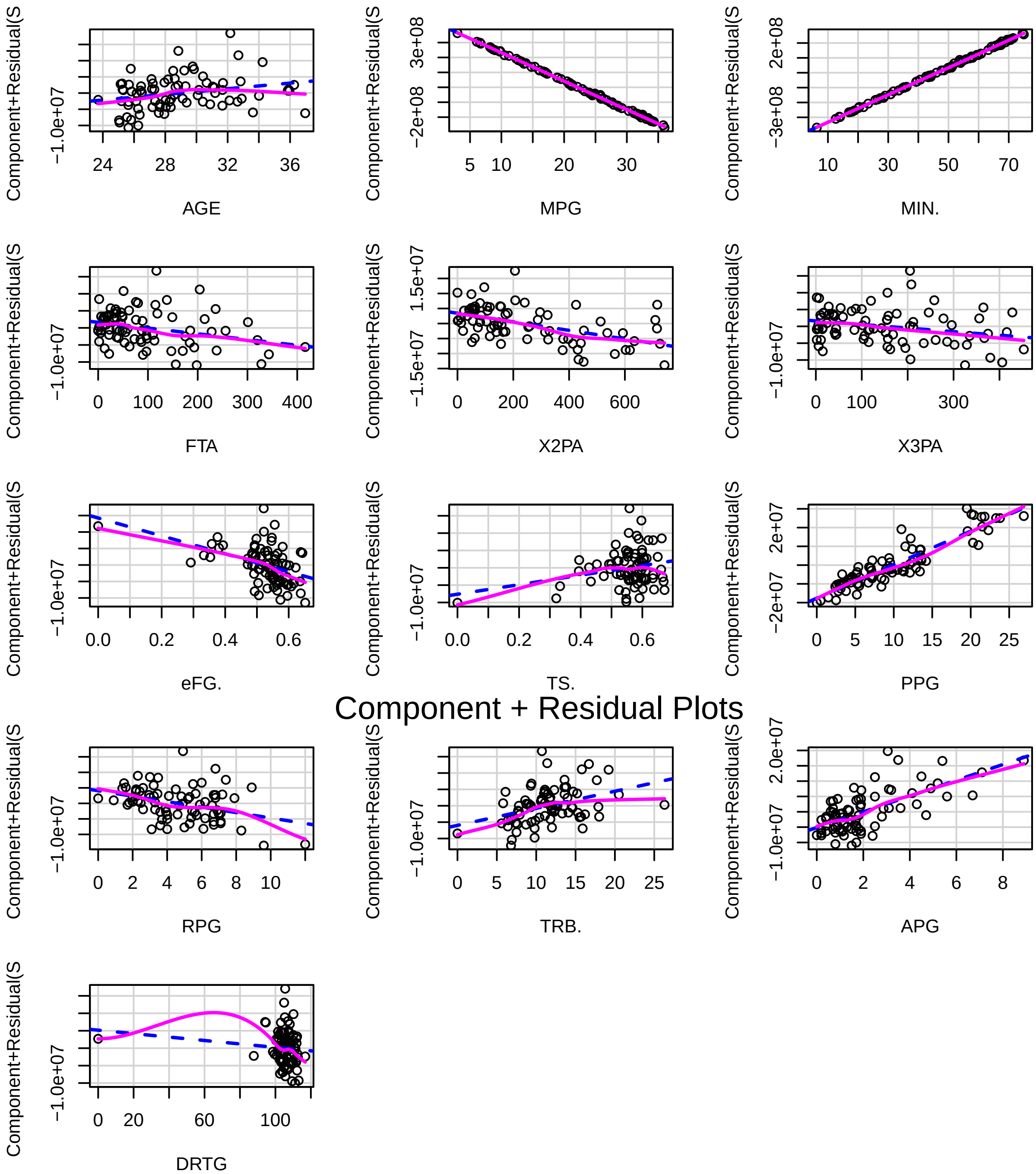
```
##
## Call:
## lm(formula = Salary ~ ., data = final.f.reg)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -9713169 -3531952  -650157  2566960 17085253
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -10184093    8489168  -1.200 0.234627
## AGE            434170     212441   2.044 0.045037 *
## MPG         -19047339   22619665  -0.842 0.402836
## MIN.          9189456   10865894   0.846 0.400814
## FTA            -16799      21001  -0.800 0.426682
## X2PA           -14256       8169  -1.745 0.085684 .
## X3PA           -10615       9199  -1.154 0.252745
## eFG.        -27155221   48674724  -0.558 0.578835
## TS.          13748222   51648632   0.266 0.790937
## PPG           1786041     297404   6.005 9.48e-08 ***
```

```
## RPG           -879662    1174468  -0.749 0.456566
## TRB.           520996     404895   1.287 0.202747
## APG           2570848     621726   4.135 0.000104 ***
## DRTG           -49288      76223  -0.647 0.520150
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5534000 on 65 degrees of freedom
## Multiple R-squared:  0.7789, Adjusted R-squared:  0.7347
## F-statistic: 17.61 on 13 and 65 DF,  p-value: < 2.2e-16
```

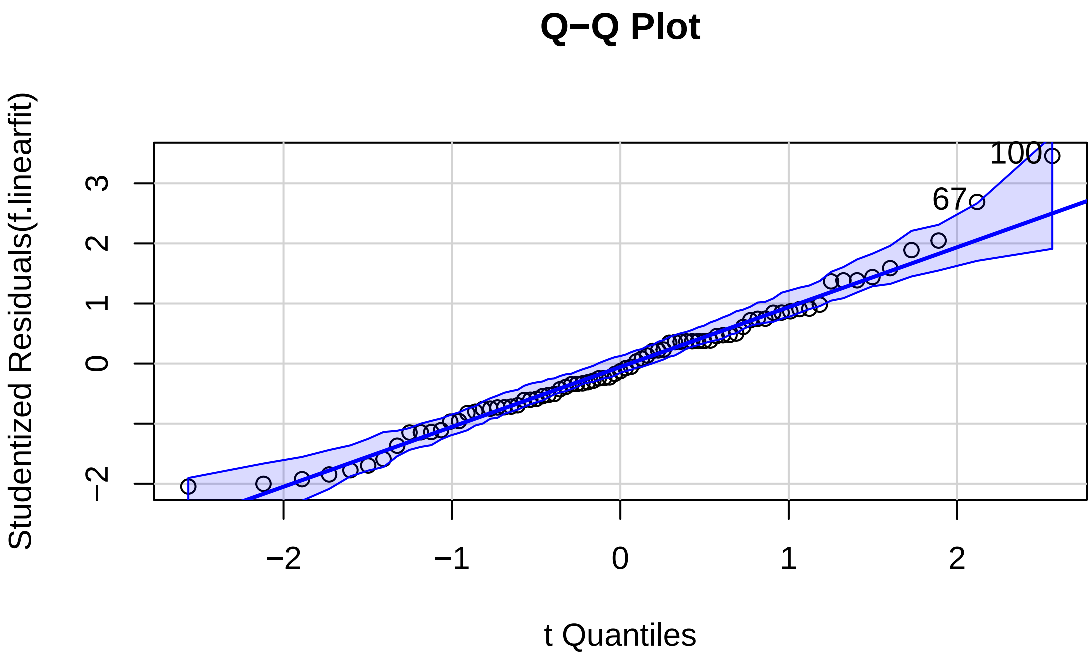
```
plot(f.linearfit)
```




```
library(car)
crPlots(f.linearfit)
```




```
qqPlot(f.linearfit, id.method = "identify", simulate = TRUE, main = "Q-Q Plot")
```

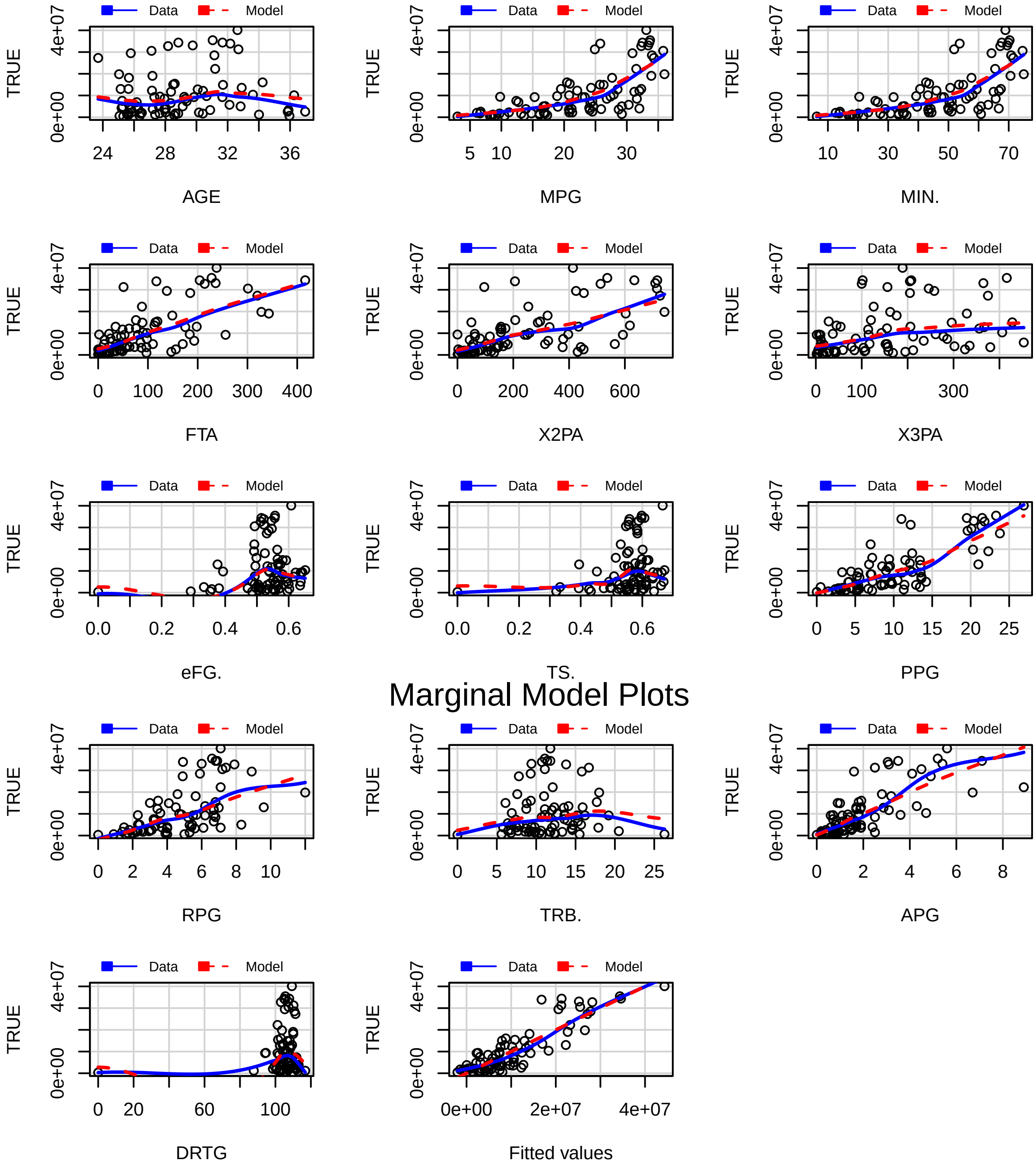



```
##  67 100
##  47  67
```

```
ncvTest(f.linearfit)
```

```
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 6.542806, Df = 1, p = 0.010531
```

```
marginalModelPlots(f.linearfit)
```




```r
# the accuracy of linear model prediction
pre.f <- predict(f.linearfit, newdata = final.f.reg.test)
mean((pre.f - final.f.reg.test$Salary)^2)
```

```
## [1] 2.896804e+13
```

```r
# check the collinearity using vif function
vif(f.linearfit)
```

```
##          AGE          MPG         MIN.          FTA         X2PA         X3PA
##     1.127903 96592.679392 96749.093963     9.717891     7.786187     3.475798
##         eFG.          TS.          PPG          RPG         TRB.          APG
```

```
##    51.618158    59.398852     9.238909    17.584824     6.671034     2.978623
##         DRTG
##     2.431683
```

```r
# it's bigger than 4, collinearity exsits, so delete the varible with the
# #biggest vif-value and build a new linear model


# new linear model and predict again
f.linearfit1 <- lm(Salary ~ AGE + MPG + FTA + X2PA + X3PA + eFG. + TS. + PPG + RPG +
    TRB. + APG + DRTG, data = final.f.reg)
summary(f.linearfit1)
```

```
##
## Call:
## lm(formula = Salary ~ AGE + MPG + FTA + X2PA + X3PA + eFG. +
##     TS. + PPG + RPG + TRB. + APG + DRTG, data = final.f.reg)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -10220301  -3383561   -806964   2631228  17836228
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -9812555    8459483  -1.160  0.25025
## AGE            436504     211964   2.059  0.04341 *
## MPG             80165     349995   0.229  0.81954
## FTA            -16140      20942  -0.771  0.44362
## X2PA           -14442       8148  -1.772  0.08096 .
## X3PA           -10192       9166  -1.112  0.27018
## eFG.        -19619437   47748835  -0.411  0.68249
## TS.           4111882   50267301   0.082  0.93505
## PPG           1808400     295587   6.118 5.78e-08 ***
## RPG           -745232    1161148  -0.642  0.52322
## TRB.           467861     399127   1.172  0.24533
## APG           2430452     597859   4.065  0.00013 ***
## DRTG           -37997      74882  -0.507  0.61355
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5522000 on 66 degrees of freedom
## Multiple R-squared:  0.7765, Adjusted R-squared:  0.7358
## F-statistic:  19.1 on 12 and 66 DF,  p-value: < 2.2e-16
```

```r
vif(f.linearfit1)
```

```
##       AGE       MPG       FTA      X2PA      X3PA      eFG.       TS.       PPG
##  1.127713 23.225914  9.704524  7.780580  3.465505 49.888326 56.507933  9.165897
##       RPG      TRB.       APG      DRTG
## 17.262715  6.510403  2.766258  2.357089
```

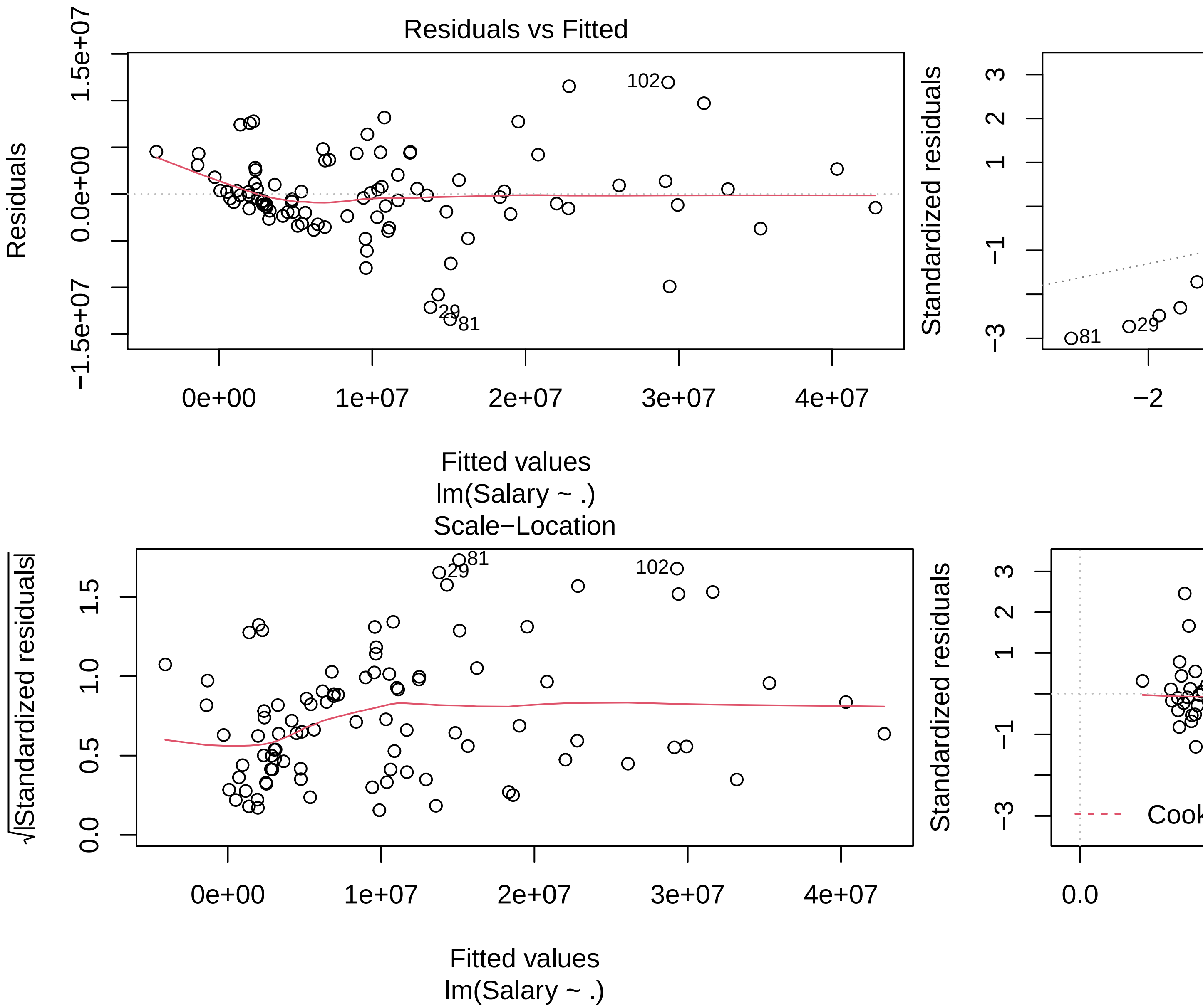
```r
pre.f <- predict(f.linearfit1, newdata = final.f.reg.test)
mean((pre.f - final.f.reg.test$Salary)^2)
```

```
## [1] 2.977197e+13
```

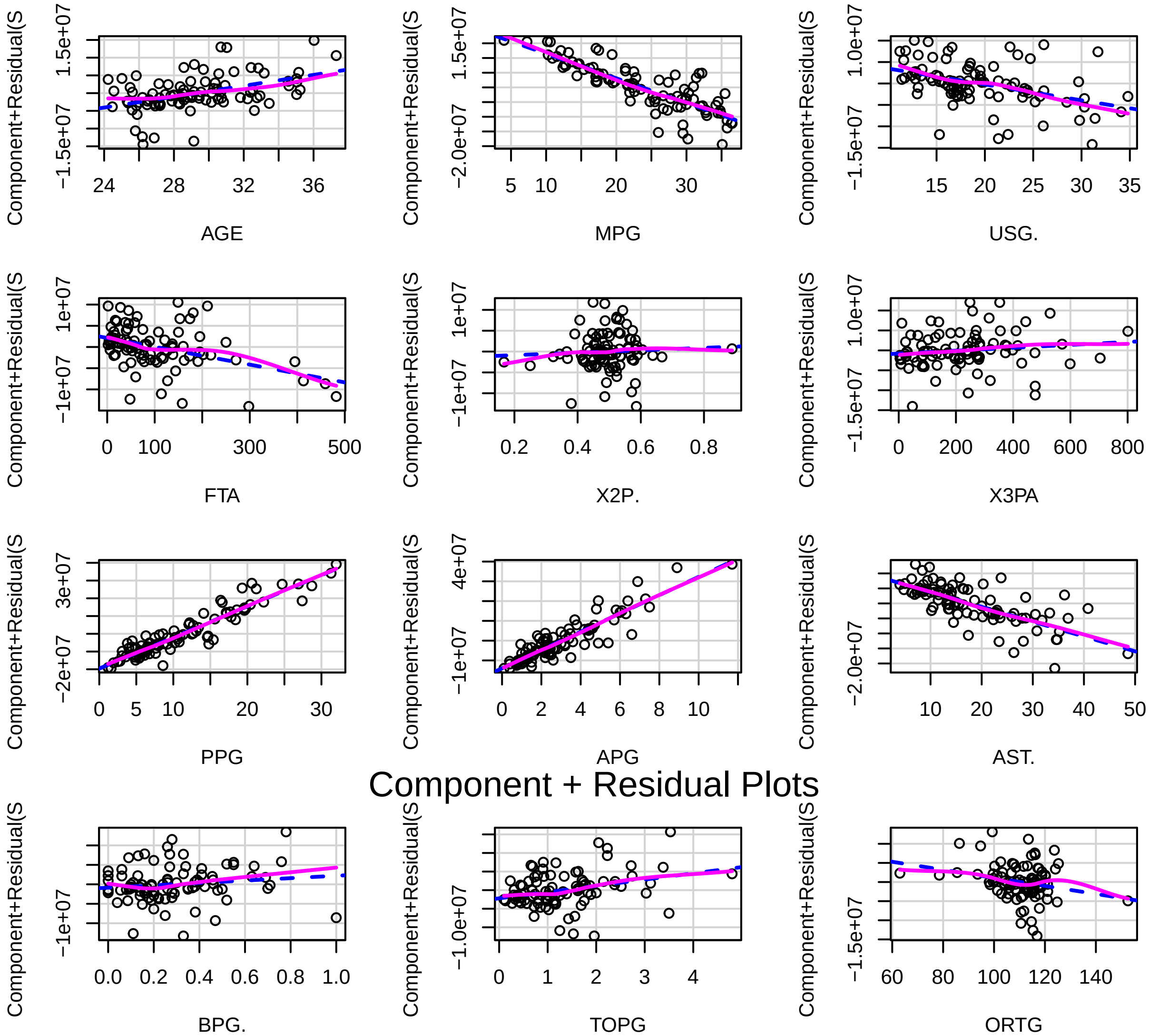
```r
# ...linear regression for POS 'G'(the variables based on the regsubsetting
# #results of POS 'G')
library(dplyr)
g.fit.train = main.pos.g.train[, 5:30]
g.fit.test = main.pos.g.test[, 5:30]
reg.g.variables <- data.frame(reg.summary.g$which)
reg.g.variables <- reg.g.variables[, -1]
a <- which(reg.g.variables[12, ] == TRUE)
a. <- colnames(g.fit.train)[a]
final.g.reg <- select(g.fit.train, col = a)
colnames(final.g.reg) <- a.
final.g.reg <- cbind(final.g.reg, g.fit.train[, 26])
colnames(final.g.reg)[13] <- "Salary"

final.g.reg.test <- select(g.fit.test, col = a)
colnames(final.g.reg.test) <- a.
final.g.reg.test <- cbind(final.g.reg.test, g.fit.test[, 26])
colnames(final.g.reg.test)[13] <- "Salary"

# linear fit
g.linearfit <- lm(Salary ~ ., data = final.g.reg)
summary(g.linearfit)
```

```
##
## Call:
## lm(formula = Salary ~ ., data = final.g.reg)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -13423066  -1954857   -376156   2176253  11953038
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1403698   11546165  -0.122 0.903546
## AGE           767326     186324   4.118 9.33e-05 ***
## MPG          -836431     255384  -3.275 0.001569 **
## USG.         -370194     332320  -1.114 0.268669
## FTA           -20938      14036  -1.492 0.139744
## X2P.         2922813   10356105   0.282 0.778505
## X3PA            3637       6840   0.532 0.596447
## PPG          1754193     481854   3.641 0.000484 ***
## APG          4624692    1184944   3.903 0.000199 ***
## AST.         -494337     193976  -2.548 0.012759 *
## BPG.         3060906    3222666   0.950 0.345108
## TOPG         1680147    2302238   0.730 0.467678
## ORTG         -105191      90983  -1.156 0.251101
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4849000 on 79 degrees of freedom
## Multiple R-squared:  0.8289, Adjusted R-squared:  0.8029
## F-statistic: 31.89 on 12 and 79 DF,  p-value: < 2.2e-16
```

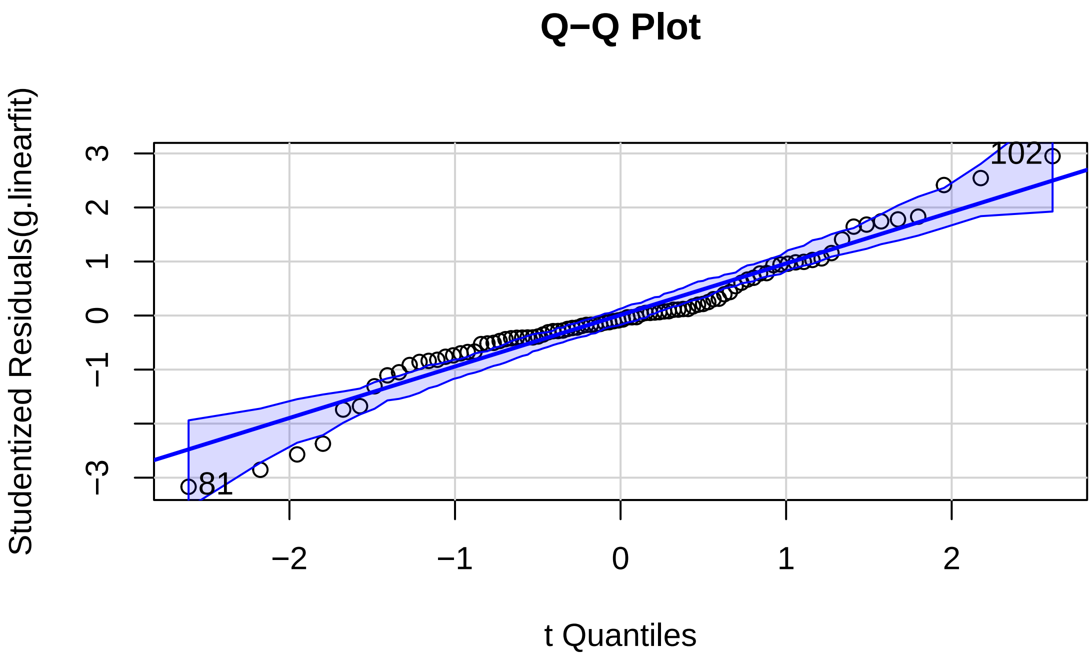
```
plot(g.linearfit)
```




```
library(car)
crPlots(g.linearfit)
```




```
qqPlot(g.linearfit, id.method = "identify", simulate = TRUE, main = "Q-Q Plot")
```

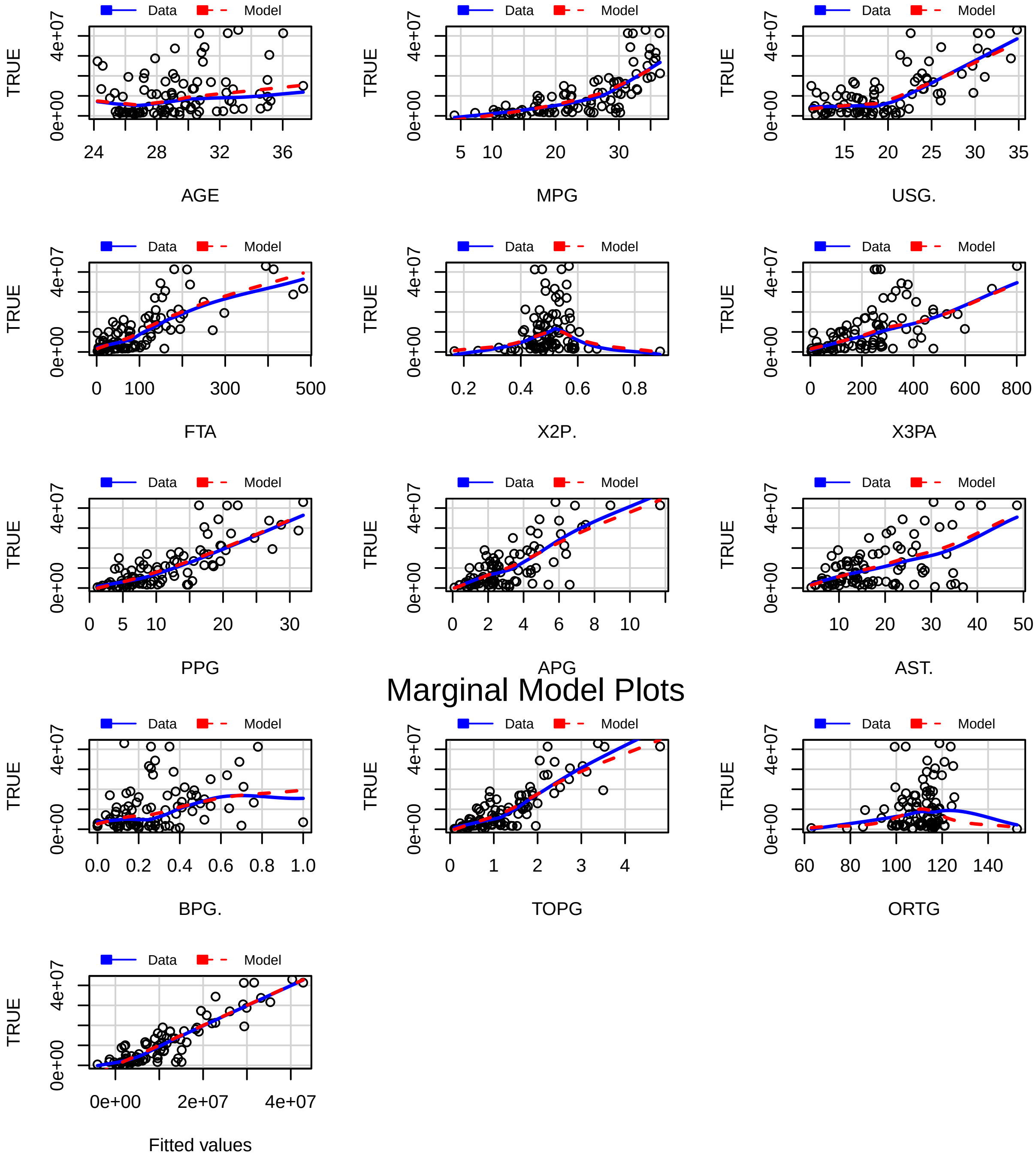



```
##  81 102
##  50  67
```

```
ncvTest(g.linearfit)
```

```
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 10.50656, Df = 1, p = 0.0011895
```

```
marginalModelPlots(g.linearfit)
```




```
# the accuracy of linear model prediction
pre.g <- predict(g.linearfit, newdata = final.g.reg.test)
mean((pre.g - final.g.reg.test$Salary)^2)
```

```
## [1] 3.524769e+13
```

```
# ......stepwise regression ...stepwise regression for POS 'C' too fewer
# observations that it can not be in stepwise process

# ...stepwise regression for POS 'F'
library(MASS)
```

```
##
## Attaching package: 'MASS'
## The following object is masked from 'package:dplyr':
##
##     select
```

```r
f.linearfit <- lm(Salary ~ ., data = f.fit.train)
stepAIC(f.linearfit, direction = "backward")
```

```
## Start:  AIC=2482.45
## Salary ~ AGE + GP + MPG + MIN. + USG. + TO. + FTA + FT. + X2PA +
##     X2P. + X3PA + X3P. + eFG. + TS. + PPG + RPG + TRB. + APG +
##     AST. + SPG + BPG. + TOPG + VI + ORTG + DRTG
##
##        Df  Sum of Sq        RSS    AIC
## - TRB.  1 6.8762e+10 1.8147e+15 2480.4
## - USG.  1 3.2779e+11 1.8149e+15 2480.5
## - ORTG  1 5.0876e+11 1.8151e+15 2480.5
## - eFG.  1 2.1382e+12 1.8167e+15 2480.5
## - BPG.  1 3.4831e+12 1.8181e+15 2480.6
## - SPG   1 4.1374e+12 1.8187e+15 2480.6
## - DRTG  1 5.8354e+12 1.8204e+15 2480.7
## - FTA   1 6.2088e+12 1.8208e+15 2480.7
## - TS.   1 7.0872e+12 1.8217e+15 2480.8
## - X3P.  1 1.2043e+13 1.8266e+15 2481.0
## - RPG   1 1.7188e+13 1.8318e+15 2481.2
## - FT.   1 2.0985e+13 1.8356e+15 2481.4
## - X2P.  1 2.3289e+13 1.8379e+15 2481.5
## - AST.  1 2.9525e+13 1.8441e+15 2481.7
## - TO.   1 3.5003e+13 1.8496e+15 2482.0
## - VI    1 4.0107e+13 1.8547e+15 2482.2
## - GP    1 4.5153e+13 1.8598e+15 2482.4
## <none>                1.8146e+15 2482.4
## - TOPG  1 4.7526e+13 1.8621e+15 2482.5
## - MPG   1 6.5782e+13 1.8804e+15 2483.3
## - MIN.  1 6.6307e+13 1.8809e+15 2483.3
## - AGE   1 8.6470e+13 1.9011e+15 2484.1
## - APG   1 9.5059e+13 1.9097e+15 2484.5
## - X3PA  1 9.6925e+13 1.9115e+15 2484.6
## - X2PA  1 1.8721e+14 2.0018e+15 2488.2
## - PPG   1 2.5325e+14 2.0678e+15 2490.8
##
## Step:  AIC=2480.45
## Salary ~ AGE + GP + MPG + MIN. + USG. + TO. + FTA + FT. + X2PA +
##     X2P. + X3PA + X3P. + eFG. + TS. + PPG + RPG + APG + AST. +
##     SPG + BPG. + TOPG + VI + ORTG + DRTG
##
##        Df  Sum of Sq        RSS    AIC
## - USG.  1 3.5190e+11 1.8150e+15 2478.5
## - ORTG  1 4.6275e+11 1.8151e+15 2478.5
## - eFG.  1 2.4985e+12 1.8172e+15 2478.6
## - BPG.  1 3.4797e+12 1.8181e+15 2478.6
## - SPG   1 4.0785e+12 1.8187e+15 2478.6
## - FTA   1 6.1459e+12 1.8208e+15 2478.7
```

```
## - DRTG  1 6.2487e+12 1.8209e+15 2478.7
## - TS.   1 8.8729e+12 1.8235e+15 2478.8
## - X3P.  1 1.5099e+13 1.8298e+15 2479.1
## - FT.   1 2.1676e+13 1.8363e+15 2479.4
## - RPG   1 2.2135e+13 1.8368e+15 2479.4
## - X2P.  1 2.3491e+13 1.8382e+15 2479.5
## - AST.  1 4.1649e+13 1.8563e+15 2480.2
## <none>              1.8147e+15 2480.4
## - GP    1 4.7547e+13 1.8622e+15 2480.5
## - TOPG  1 5.2453e+13 1.8671e+15 2480.7
## - TO.   1 5.4310e+13 1.8690e+15 2480.8
## - VI    1 6.6113e+13 1.8808e+15 2481.3
## - MPG   1 6.6157e+13 1.8808e+15 2481.3
## - MIN.  1 6.6612e+13 1.8813e+15 2481.3
## - AGE   1 8.6438e+13 1.9011e+15 2482.1
## - APG   1 1.0162e+14 1.9163e+15 2482.8
## - X3PA  1 1.0282e+14 1.9175e+15 2482.8
## - X2PA  1 1.8911e+14 2.0038e+15 2486.3
## - PPG   1 2.5318e+14 2.0678e+15 2488.8
##
## Step:  AIC=2478.47
## Salary ~ AGE + GP + MPG + MIN. + TO. + FTA + FT. + X2PA + X2P. +
##     X3PA + X3P. + eFG. + TS. + PPG + RPG + APG + AST. + SPG +
##     BPG. + TOPG + VI + ORTG + DRTG
##
##        Df  Sum of Sq        RSS    AIC
## - ORTG  1 9.9144e+11 1.8160e+15 2476.5
## - BPG.  1 3.2079e+12 1.8182e+15 2476.6
## - eFG.  1 3.2425e+12 1.8183e+15 2476.6
## - SPG   1 3.7593e+12 1.8188e+15 2476.6
## - DRTG  1 5.9272e+12 1.8209e+15 2476.7
## - FTA   1 5.9513e+12 1.8210e+15 2476.7
## - TS.   1 1.0014e+13 1.8250e+15 2476.9
## - X3P.  1 1.5690e+13 1.8307e+15 2477.2
## - X2P.  1 2.3373e+13 1.8384e+15 2477.5
## - FT.   1 2.6559e+13 1.8416e+15 2477.6
## - RPG   1 2.7096e+13 1.8421e+15 2477.6
## - AST.  1 4.5899e+13 1.8609e+15 2478.4
## <none>              1.8150e+15 2478.5
## - GP    1 4.7510e+13 1.8625e+15 2478.5
## - TOPG  1 5.3330e+13 1.8683e+15 2478.8
## - TO.   1 5.5445e+13 1.8705e+15 2478.8
## - MPG   1 6.6169e+13 1.8812e+15 2479.3
## - MIN.  1 6.6576e+13 1.8816e+15 2479.3
## - AGE   1 8.7244e+13 1.9023e+15 2480.2
## - VI    1 9.6083e+13 1.9111e+15 2480.5
## - X3PA  1 1.0247e+14 1.9175e+15 2480.8
## - APG   1 1.0276e+14 1.9178e+15 2480.8
## - X2PA  1 1.8888e+14 2.0039e+15 2484.3
## - PPG   1 3.6861e+14 2.1836e+15 2491.1
##
## Step:  AIC=2476.51
## Salary ~ AGE + GP + MPG + MIN. + TO. + FTA + FT. + X2PA + X2P. +
##     X3PA + X3P. + eFG. + TS. + PPG + RPG + APG + AST. + SPG +
```

```
##      BPG. + TOPG + VI + DRTG
##
##         Df  Sum of Sq        RSS    AIC
## - SPG   1 3.2049e+12 1.8192e+15 2474.7
## - BPG.  1 3.4398e+12 1.8194e+15 2474.7
## - eFG.  1 4.2409e+12 1.8203e+15 2474.7
## - FTA   1 5.5043e+12 1.8215e+15 2474.8
## - DRTG  1 1.1695e+13 1.8277e+15 2475.0
## - X3P.  1 1.5242e+13 1.8313e+15 2475.2
## - TS.   1 1.6327e+13 1.8323e+15 2475.2
## - X2P.  1 2.3257e+13 1.8393e+15 2475.5
## - FT.   1 3.0153e+13 1.8462e+15 2475.8
## - RPG   1 3.5239e+13 1.8512e+15 2476.0
## <none>               1.8160e+15 2476.5
## - GP    1 4.6685e+13 1.8627e+15 2476.5
## - AST.  1 4.8956e+13 1.8650e+15 2476.6
## - TOPG  1 5.4531e+13 1.8705e+15 2476.8
## - TO.   1 6.0745e+13 1.8768e+15 2477.1
## - MPG   1 6.6231e+13 1.8822e+15 2477.3
## - MIN.  1 6.6676e+13 1.8827e+15 2477.4
## - AGE   1 8.7361e+13 1.9034e+15 2478.2
## - VI    1 9.9905e+13 1.9159e+15 2478.7
## - APG   1 1.0186e+14 1.9179e+15 2478.8
## - X3PA  1 1.0222e+14 1.9182e+15 2478.8
## - X2PA  1 1.9043e+14 2.0064e+15 2482.4
## - PPG   1 3.6838e+14 2.1844e+15 2489.1
##
## Step:  AIC=2474.65
## Salary ~ AGE + GP + MPG + MIN. + TO. + FTA + FT. + X2PA + X2P. +
##     X3PA + X3P. + eFG. + TS. + PPG + RPG + APG + AST. + BPG. +
##     TOPG + VI + DRTG
##
##         Df  Sum of Sq        RSS    AIC
## - BPG.  1 3.2301e+12 1.8224e+15 2472.8
## - eFG.  1 4.7160e+12 1.8239e+15 2472.9
## - FTA   1 7.6111e+12 1.8268e+15 2473.0
## - DRTG  1 1.0891e+13 1.8301e+15 2473.1
## - TS.   1 1.6742e+13 1.8360e+15 2473.4
## - X3P.  1 1.7539e+13 1.8368e+15 2473.4
## - X2P.  1 2.5803e+13 1.8450e+15 2473.8
## - FT.   1 2.9981e+13 1.8492e+15 2473.9
## - RPG   1 3.4426e+13 1.8536e+15 2474.1
## <none>               1.8192e+15 2474.7
## - GP    1 4.6883e+13 1.8661e+15 2474.7
## - AST.  1 4.8166e+13 1.8674e+15 2474.7
## - TOPG  1 5.3630e+13 1.8728e+15 2474.9
## - TO.   1 6.0878e+13 1.8801e+15 2475.2
## - MPG   1 6.5968e+13 1.8852e+15 2475.5
## - MIN.  1 6.6249e+13 1.8855e+15 2475.5
## - AGE   1 8.4518e+13 1.9037e+15 2476.2
## - VI    1 9.7302e+13 1.9165e+15 2476.8
## - X3PA  1 9.9265e+13 1.9185e+15 2476.8
## - APG   1 9.9602e+13 1.9188e+15 2476.9
## - X2PA  1 1.9010e+14 2.0093e+15 2480.5
```

```
## - PPG   1 4.0451e+14 2.2237e+15 2488.5
##
## Step:  AIC=2472.79
## Salary ~ AGE + GP + MPG + MIN. + TO. + FTA + FT. + X2PA + X2P. +
##     X3PA + X3P. + eFG. + TS. + PPG + RPG + APG + AST. + TOPG +
##     VI + DRTG
##
##        Df  Sum of Sq        RSS    AIC
## - eFG.  1 4.7939e+12 1.8272e+15 2471.0
## - FTA   1 7.6297e+12 1.8301e+15 2471.1
## - DRTG  1 1.1413e+13 1.8339e+15 2471.3
## - X3P.  1 1.6620e+13 1.8391e+15 2471.5
## - TS.   1 1.6811e+13 1.8393e+15 2471.5
## - X2P.  1 2.9335e+13 1.8518e+15 2472.1
## - FT.   1 3.2633e+13 1.8551e+15 2472.2
## - RPG   1 4.0550e+13 1.8630e+15 2472.5
## - GP    1 4.6122e+13 1.8686e+15 2472.8
## <none>              1.8224e+15 2472.8
## - AST.  1 4.7106e+13 1.8696e+15 2472.8
## - TOPG  1 5.2563e+13 1.8750e+15 2473.0
## - TO.   1 5.9218e+13 1.8817e+15 2473.3
## - MPG   1 6.8528e+13 1.8910e+15 2473.7
## - MIN.  1 6.8812e+13 1.8913e+15 2473.7
## - AGE   1 8.1322e+13 1.9038e+15 2474.2
## - X3PA  1 9.6645e+13 1.9191e+15 2474.9
## - APG   1 9.9398e+13 1.9218e+15 2475.0
## - VI    1 1.0030e+14 1.9227e+15 2475.0
## - X2PA  1 1.8788e+14 2.0103e+15 2478.5
## - PPG   1 4.0395e+14 2.2264e+15 2486.6
##
## Step:  AIC=2471
## Salary ~ AGE + GP + MPG + MIN. + TO. + FTA + FT. + X2PA + X2P. +
##     X3PA + X3P. + TS. + PPG + RPG + APG + AST. + TOPG + VI +
##     DRTG
##
##        Df  Sum of Sq        RSS    AIC
## - X3P.  1 1.3283e+13 1.8405e+15 2469.6
## - DRTG  1 2.1336e+13 1.8486e+15 2469.9
## - FTA   1 2.2049e+13 1.8493e+15 2469.9
## - X2P.  1 2.8928e+13 1.8562e+15 2470.2
## - FT.   1 3.9941e+13 1.8672e+15 2470.7
## - RPG   1 4.4284e+13 1.8715e+15 2470.9
## <none>              1.8272e+15 2471.0
## - TOPG  1 4.8758e+13 1.8760e+15 2471.1
## - AST.  1 4.9828e+13 1.8771e+15 2471.1
## - GP    1 5.5850e+13 1.8831e+15 2471.4
## - TO.   1 5.6707e+13 1.8839e+15 2471.4
## - MPG   1 6.7713e+13 1.8950e+15 2471.9
## - MIN.  1 6.8053e+13 1.8953e+15 2471.9
## - AGE   1 7.6629e+13 1.9039e+15 2472.2
## - X3PA  1 9.3184e+13 1.9204e+15 2472.9
## - APG   1 9.7350e+13 1.9246e+15 2473.1
## - TS.   1 9.9011e+13 1.9262e+15 2473.2
## - VI    1 1.1030e+14 1.9375e+15 2473.6
```

```
## - X2PA  1 1.8308e+14 2.0103e+15 2476.5
## - PPG   1 3.9920e+14 2.2264e+15 2484.6
##
## Step:  AIC=2469.57
## Salary ~ AGE + GP + MPG + MIN. + TO. + FTA + FT. + X2PA + X2P. +
##     X3PA + TS. + PPG + RPG + APG + AST. + TOPG + VI + DRTG
##
##        Df  Sum of Sq        RSS    AIC
## - FTA   1 1.2757e+13 1.8533e+15 2468.1
## - X2P.  1 1.6501e+13 1.8570e+15 2468.3
## - DRTG  1 3.5476e+13 1.8760e+15 2469.1
## - RPG   1 4.2639e+13 1.8832e+15 2469.4
## - AST.  1 4.6628e+13 1.8871e+15 2469.6
## - FT.   1 4.6793e+13 1.8873e+15 2469.6
## <none>              1.8405e+15 2469.6
## - GP    1 4.8918e+13 1.8894e+15 2469.6
## - TOPG  1 5.9213e+13 1.8997e+15 2470.1
## - MPG   1 7.4508e+13 1.9150e+15 2470.7
## - MIN.  1 7.4987e+13 1.9155e+15 2470.7
## - AGE   1 7.5209e+13 1.9157e+15 2470.7
## - TO.   1 7.8173e+13 1.9187e+15 2470.9
## - APG   1 9.0003e+13 1.9305e+15 2471.3
## - X3PA  1 1.0866e+14 1.9492e+15 2472.1
## - VI    1 1.1171e+14 1.9522e+15 2472.2
## - TS.   1 1.5859e+14 1.9991e+15 2474.1
## - X2PA  1 1.7648e+14 2.0170e+15 2474.8
## - PPG   1 3.8736e+14 2.2279e+15 2482.7
##
## Step:  AIC=2468.12
## Salary ~ AGE + GP + MPG + MIN. + TO. + FT. + X2PA + X2P. + X3PA +
##     TS. + PPG + RPG + APG + AST. + TOPG + VI + DRTG
##
##        Df  Sum of Sq        RSS    AIC
## - X2P.  1 1.3698e+13 1.8670e+15 2466.7
## - DRTG  1 3.6920e+13 1.8902e+15 2467.7
## - RPG   1 3.9932e+13 1.8932e+15 2467.8
## - GP    1 4.0406e+13 1.8937e+15 2467.8
## - AST.  1 4.2305e+13 1.8956e+15 2467.9
## <none>              1.8533e+15 2468.1
## - FT.   1 5.0104e+13 1.9034e+15 2468.2
## - TOPG  1 5.1047e+13 1.9043e+15 2468.3
## - TO.   1 7.2740e+13 1.9260e+15 2469.2
## - MPG   1 7.3763e+13 1.9270e+15 2469.2
## - MIN.  1 7.4352e+13 1.9276e+15 2469.2
## - APG   1 7.9000e+13 1.9323e+15 2469.4
## - AGE   1 8.6625e+13 1.9399e+15 2469.7
## - X3PA  1 9.7903e+13 1.9512e+15 2470.2
## - VI    1 1.1327e+14 1.9665e+15 2470.8
## - TS.   1 1.5606e+14 2.0093e+15 2472.5
## - X2PA  1 2.0037e+14 2.0536e+15 2474.2
## - PPG   1 4.1215e+14 2.2654e+15 2482.0
##
## Step:  AIC=2466.7
## Salary ~ AGE + GP + MPG + MIN. + TO. + FT. + X2PA + X3PA + TS. +
```

```
##     PPG + RPG + APG + AST. + TOPG + VI + DRTG
##
##        Df  Sum of Sq        RSS    AIC
## - GP    1 2.8274e+13 1.8953e+15 2465.9
## - RPG   1 3.5118e+13 1.9021e+15 2466.2
## - AST.  1 3.8167e+13 1.9051e+15 2466.3
## - DRTG  1 4.2515e+13 1.9095e+15 2466.5
## <none>               1.8670e+15 2466.7
## - FT.   1 5.6966e+13 1.9239e+15 2467.1
## - TOPG  1 6.2252e+13 1.9292e+15 2467.3
## - APG   1 7.8187e+13 1.9452e+15 2467.9
## - MPG   1 8.4747e+13 1.9517e+15 2468.2
## - X3PA  1 8.5093e+13 1.9521e+15 2468.2
## - MIN.  1 8.5231e+13 1.9522e+15 2468.2
## - TO.   1 9.7883e+13 1.9649e+15 2468.7
## - VI    1 1.0134e+14 1.9683e+15 2468.9
## - AGE   1 1.2500e+14 1.9920e+15 2469.8
## - TS.   1 1.8084e+14 2.0478e+15 2472.0
## - X2PA  1 1.8669e+14 2.0537e+15 2472.2
## - PPG   1 4.2109e+14 2.2881e+15 2480.8
##
## Step:  AIC=2465.89
## Salary ~ AGE + MPG + MIN. + TO. + FT. + X2PA + X3PA + TS. + PPG +
##     RPG + APG + AST. + TOPG + VI + DRTG
##
##        Df  Sum of Sq        RSS    AIC
## - RPG   1 2.0072e+13 1.9153e+15 2464.7
## - AST.  1 2.3309e+13 1.9186e+15 2464.8
## - DRTG  1 3.7024e+13 1.9323e+15 2465.4
## - TOPG  1 4.6465e+13 1.9417e+15 2465.8
## - FT.   1 4.8055e+13 1.9433e+15 2465.9
## <none>               1.8953e+15 2465.9
## - X3PA  1 5.9875e+13 1.9551e+15 2466.3
## - APG   1 6.0366e+13 1.9556e+15 2466.4
## - MPG   1 6.2885e+13 1.9581e+15 2466.5
## - MIN.  1 6.3353e+13 1.9586e+15 2466.5
## - TO.   1 7.3857e+13 1.9691e+15 2466.9
## - VI    1 7.7356e+13 1.9726e+15 2467.1
## - AGE   1 1.2936e+14 2.0246e+15 2469.1
## - TS.   1 1.5973e+14 2.0550e+15 2470.3
## - X2PA  1 2.4905e+14 2.1443e+15 2473.6
## - PPG   1 5.1133e+14 2.4066e+15 2482.8
##
## Step:  AIC=2464.72
## Salary ~ AGE + MPG + MIN. + TO. + FT. + X2PA + X3PA + TS. + PPG +
##     APG + AST. + TOPG + VI + DRTG
##
##        Df  Sum of Sq        RSS    AIC
## - AST.  1 6.6750e+12 1.9220e+15 2463.0
## - DRTG  1 2.3077e+13 1.9384e+15 2463.7
## - TOPG  1 3.9703e+13 1.9550e+15 2464.3
## - X3PA  1 4.0025e+13 1.9553e+15 2464.3
## - APG   1 4.3423e+13 1.9587e+15 2464.5
## <none>               1.9153e+15 2464.7
```

```
## - FT.   1 5.3323e+13 1.9686e+15 2464.9
## - MIN.  1 5.4260e+13 1.9696e+15 2464.9
## - MPG   1 5.4394e+13 1.9697e+15 2464.9
## - TO.   1 6.2982e+13 1.9783e+15 2465.3
## - VI    1 7.9418e+13 1.9947e+15 2465.9
## - AGE   1 1.2134e+14 2.0367e+15 2467.6
## - TS.   1 1.4159e+14 2.0569e+15 2468.3
## - X2PA  1 2.4327e+14 2.1586e+15 2472.2
## - PPG   1 5.8007e+14 2.4954e+15 2483.6
##
## Step:  AIC=2462.99
## Salary ~ AGE + MPG + MIN. + TO. + FT. + X2PA + X3PA + TS. + PPG +
##     APG + TOPG + VI + DRTG
##
##        Df  Sum of Sq        RSS    AIC
## - DRTG  1 3.3785e+13 1.9558e+15 2462.4
## - TOPG  1 3.3984e+13 1.9560e+15 2462.4
## - X3PA  1 3.7326e+13 1.9593e+15 2462.5
## <none>              1.9220e+15 2463.0
## - MIN.  1 5.2147e+13 1.9741e+15 2463.1
## - MPG   1 5.2181e+13 1.9742e+15 2463.1
## - FT.   1 5.6185e+13 1.9782e+15 2463.3
## - TO.   1 5.8443e+13 1.9804e+15 2463.4
## - VI    1 8.4153e+13 2.0062e+15 2464.4
## - APG   1 9.9846e+13 2.0218e+15 2465.0
## - AGE   1 1.1527e+14 2.0373e+15 2465.6
## - TS.   1 1.3542e+14 2.0574e+15 2466.4
## - X2PA  1 2.3664e+14 2.1586e+15 2470.2
## - PPG   1 5.7842e+14 2.5004e+15 2481.8
##
## Step:  AIC=2462.37
## Salary ~ AGE + MPG + MIN. + TO. + FT. + X2PA + X3PA + TS. + PPG +
##     APG + TOPG + VI
##
##        Df  Sum of Sq        RSS    AIC
## - TOPG  1 1.9668e+13 1.9755e+15 2461.2
## - TO.   1 3.6716e+13 1.9925e+15 2461.8
## - MIN.  1 3.8044e+13 1.9938e+15 2461.9
## - MPG   1 3.8169e+13 1.9940e+15 2461.9
## - FT.   1 3.9763e+13 1.9955e+15 2462.0
## - X3PA  1 4.1941e+13 1.9977e+15 2462.1
## <none>              1.9558e+15 2462.4
## - VI    1 6.6411e+13 2.0222e+15 2463.0
## - APG   1 1.0104e+14 2.0568e+15 2464.3
## - AGE   1 1.4082e+14 2.0966e+15 2465.9
## - TS.   1 1.6759e+14 2.1234e+15 2466.9
## - X2PA  1 2.1485e+14 2.1706e+15 2468.6
## - PPG   1 5.4993e+14 2.5057e+15 2479.9
##
## Step:  AIC=2461.16
## Salary ~ AGE + MPG + MIN. + TO. + FT. + X2PA + X3PA + TS. + PPG +
##     APG + VI
##
##        Df  Sum of Sq        RSS    AIC
```

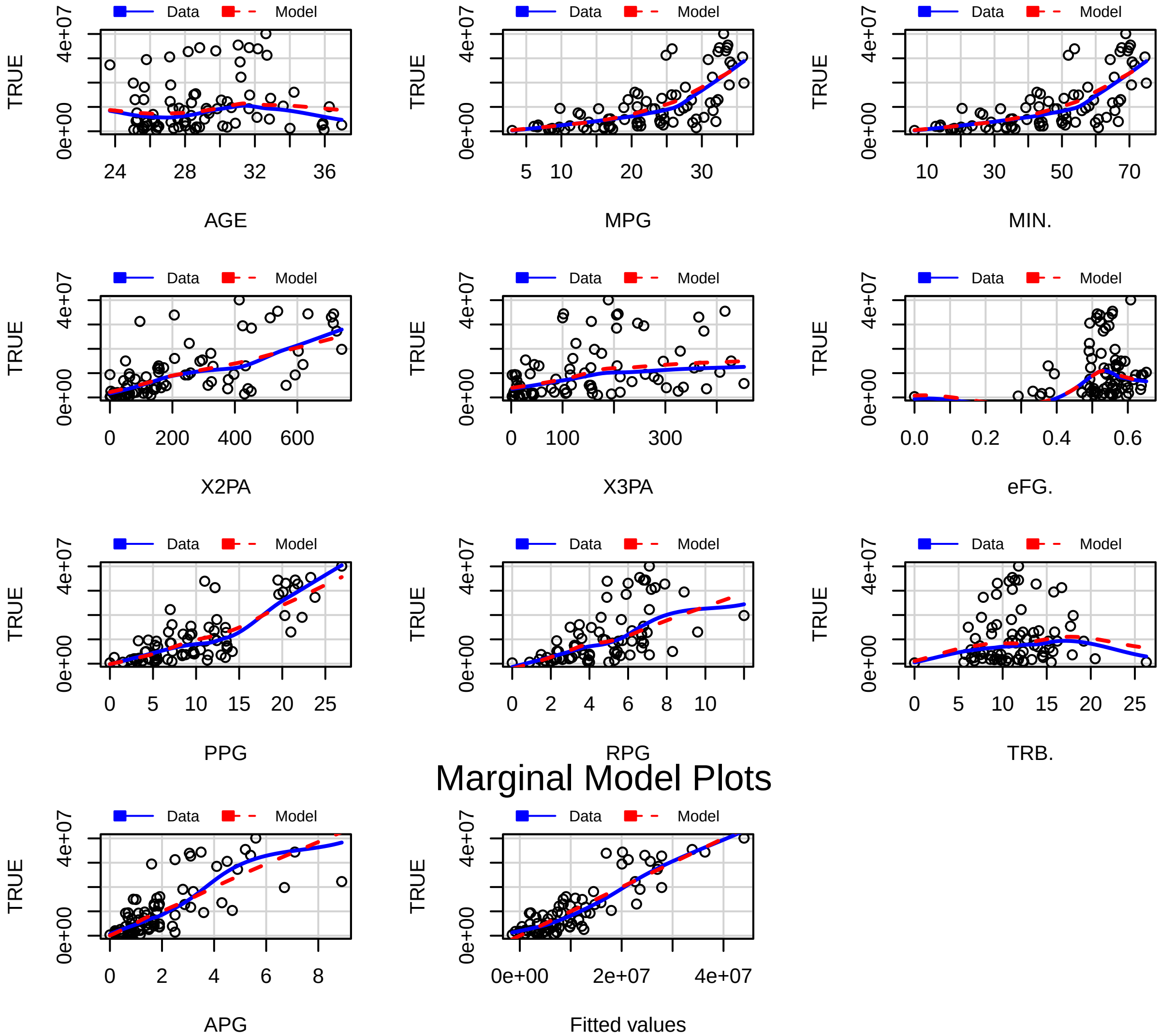
```
## - TO.   1 1.7049e+13 1.9925e+15 2459.8
## - MIN.  1 2.6099e+13 2.0015e+15 2460.2
## - MPG   1 2.6197e+13 2.0016e+15 2460.2
## - FT.   1 3.2909e+13 2.0084e+15 2460.5
## - X3PA  1 4.4836e+13 2.0203e+15 2460.9
## <none>              1.9755e+15 2461.2
## - VI    1 5.6862e+13 2.0323e+15 2461.4
## - APG   1 8.3500e+13 2.0590e+15 2462.4
## - TS.   1 1.4793e+14 2.1234e+15 2464.9
## - AGE   1 1.6079e+14 2.1362e+15 2465.3
## - X2PA  1 2.0974e+14 2.1852e+15 2467.1
## - PPG   1 7.9065e+14 2.7661e+15 2485.8
##
## Step:  AIC=2459.84
## Salary ~ AGE + MPG + MIN. + FT. + X2PA + X3PA + TS. + PPG + APG +
##     VI
##
##        Df  Sum of Sq        RSS    AIC
## - MIN.  1 2.5563e+13 2.0181e+15 2458.8
## - MPG   1 2.5625e+13 2.0181e+15 2458.8
## - FT.   1 2.6602e+13 2.0191e+15 2458.9
## - X3PA  1 5.0018e+13 2.0425e+15 2459.8
## <none>              1.9925e+15 2459.8
## - VI    1 7.3605e+13 2.0661e+15 2460.7
## - APG   1 1.3608e+14 2.1286e+15 2463.1
## - TS.   1 1.3812e+14 2.1306e+15 2463.1
## - AGE   1 1.7405e+14 2.1665e+15 2464.4
## - X2PA  1 2.2290e+14 2.2154e+15 2466.2
## - PPG   1 8.1148e+14 2.8040e+15 2484.8
##
## Step:  AIC=2458.85
## Salary ~ AGE + MPG + FT. + X2PA + X3PA + TS. + PPG + APG + VI
##
##        Df  Sum of Sq        RSS    AIC
## - MPG   1 5.6624e+11 2.0186e+15 2456.9
## - FT.   1 1.6244e+13 2.0343e+15 2457.5
## - X3PA  1 4.4162e+13 2.0622e+15 2458.6
## <none>              2.0181e+15 2458.8
## - VI    1 6.6908e+13 2.0850e+15 2459.4
## - APG   1 1.2027e+14 2.1383e+15 2461.4
## - TS.   1 1.4019e+14 2.1582e+15 2462.2
## - AGE   1 1.6934e+14 2.1874e+15 2463.2
## - X2PA  1 2.2628e+14 2.2443e+15 2465.2
## - PPG   1 8.4482e+14 2.8629e+15 2484.5
##
## Step:  AIC=2456.87
## Salary ~ AGE + FT. + X2PA + X3PA + TS. + PPG + APG + VI
##
##        Df  Sum of Sq        RSS    AIC
## - FT.   1 1.6276e+13 2.0349e+15 2455.5
## <none>              2.0186e+15 2456.9
## - X3PA  1 5.5302e+13 2.0739e+15 2457.0
## - VI    1 7.8543e+13 2.0972e+15 2457.9
## - APG   1 1.5785e+14 2.1765e+15 2460.8
```

```
## - TS.    1 1.5920e+14 2.1778e+15 2460.9
## - AGE    1 1.7478e+14 2.1934e+15 2461.4
## - X2PA   1 2.2987e+14 2.2485e+15 2463.4
## - PPG    1 9.8705e+14 3.0057e+15 2486.3
##
## Step:  AIC=2455.5
## Salary ~ AGE + X2PA + X3PA + TS. + PPG + APG + VI
##
##          Df  Sum of Sq        RSS    AIC
## - X3PA   1 4.9792e+13 2.0847e+15 2455.4
## <none>               2.0349e+15 2455.5
## - VI     1 6.9343e+13 2.1042e+15 2456.2
## - TS.    1 1.6014e+14 2.1950e+15 2459.5
## - AGE    1 1.7352e+14 2.2084e+15 2460.0
## - APG    1 1.7891e+14 2.2138e+15 2460.2
## - X2PA   1 2.4847e+14 2.2834e+15 2462.6
## - PPG    1 1.0540e+15 3.0889e+15 2486.5
##
## Step:  AIC=2455.41
## Salary ~ AGE + X2PA + TS. + PPG + APG + VI
##
##          Df  Sum of Sq        RSS    AIC
## <none>               2.0847e+15 2455.4
## - APG    1 1.3658e+14 2.2213e+15 2458.4
## - AGE    1 1.7899e+14 2.2637e+15 2459.9
## - VI     1 1.9830e+14 2.2830e+15 2460.6
## - X2PA   1 2.1369e+14 2.2984e+15 2461.1
## - TS.    1 2.3478e+14 2.3195e+15 2461.8
## - PPG    1 1.1977e+15 3.2824e+15 2489.3

##
## Call:
## lm(formula = Salary ~ AGE + X2PA + TS. + PPG + APG + VI, data = f.fit.train)
##
## Coefficients:
## (Intercept)          AGE         X2PA          TS.          PPG          APG
##   -13397286       503174       -16376    -23273452      1366557      1322836
##          VI
##     1403011
```

```r
# using the combination of varible with lowest AIC value and build a model
f.linearfit = lm(formula = Salary ~ AGE + MPG + MIN. + MPG + X2PA + X3PA + eFG. +
    PPG + RPG + TRB. + APG, data = f.fit.train)

# check the accuracy of prediction
pre.f <- predict(f.linearfit, newdata = f.fit.test)
mean((pre.f - f.fit.test$Salary)^2)
```

```
## [1] 3.108638e+13
```

```r
marginalModelPlots(f.linearfit)
```




```
# ...stepwise regression for POS 'G'
library(MASS)
g.linearfit <- lm(Salary ~ ., data = g.fit.train)
stepAIC(g.linearfit, direction = "backward")
```

```
## Start:  AIC=2861.11
## Salary ~ AGE + GP + MPG + MIN. + USG. + TO. + FTA + FT. + X2PA +
##     X2P. + X3PA + X3P. + eFG. + TS. + PPG + RPG + TRB. + APG +
##     AST. + SPG + BPG. + TOPG + VI + ORTG + DRTG
##
##        Df  Sum of Sq        RSS    AIC
## - TS.   1 1.3925e+09 1.6767e+15 2859.1
## - GP    1 7.9858e+09 1.6767e+15 2859.1
## - X2PA  1 6.3717e+10 1.6767e+15 2859.1
## - FT.   1 1.3656e+12 1.6780e+15 2859.2
## - MIN.  1 3.3926e+12 1.6801e+15 2859.3
## - ORTG  1 4.1592e+12 1.6808e+15 2859.3
## - MPG   1 4.1713e+12 1.6808e+15 2859.3
## - X3PA  1 9.0589e+12 1.6857e+15 2859.6
## - SPG   1 1.0595e+13 1.6873e+15 2859.7
```

```
## - BPG.  1 1.1258e+13 1.6879e+15 2859.7
## - eFG.  1 1.1468e+13 1.6881e+15 2859.7
## - VI    1 1.2962e+13 1.6896e+15 2859.8
## - TRB.  1 1.4873e+13 1.6915e+15 2859.9
## - TO.   1 1.5263e+13 1.6919e+15 2859.9
## - RPG   1 1.8730e+13 1.6954e+15 2860.1
## - FTA   1 2.2066e+13 1.6987e+15 2860.3
## - TOPG  1 2.9127e+13 1.7058e+15 2860.7
## - USG.  1 3.3071e+13 1.7097e+15 2860.9
## <none>               1.6767e+15 2861.1
## - X3P.  1 3.6910e+13 1.7136e+15 2861.1
## - DRTG  1 3.8633e+13 1.7153e+15 2861.2
## - AST.  1 4.0401e+13 1.7171e+15 2861.3
## - X2P.  1 6.1053e+13 1.7377e+15 2862.4
## - APG   1 1.3241e+14 1.8091e+15 2866.1
## - PPG   1 1.4980e+14 1.8265e+15 2867.0
## - AGE   1 4.3111e+14 2.1078e+15 2880.2
##
## Step:  AIC=2859.11
## Salary ~ AGE + GP + MPG + MIN. + USG. + TO. + FTA + FT. + X2PA +
##     X2P. + X3PA + X3P. + eFG. + PPG + RPG + TRB. + APG + AST. +
##     SPG + BPG. + TOPG + VI + ORTG + DRTG
##
##        Df  Sum of Sq        RSS    AIC
## - GP    1 1.2592e+10 1.6767e+15 2857.1
## - X2PA  1 8.4765e+10 1.6768e+15 2857.1
## - FT.   1 1.8388e+12 1.6785e+15 2857.2
## - MIN.  1 3.4139e+12 1.6801e+15 2857.3
## - MPG   1 4.1948e+12 1.6809e+15 2857.3
## - ORTG  1 6.1257e+12 1.6828e+15 2857.4
## - SPG   1 1.0601e+13 1.6873e+15 2857.7
## - BPG.  1 1.1290e+13 1.6880e+15 2857.7
## - X3PA  1 1.2385e+13 1.6891e+15 2857.8
## - VI    1 1.3188e+13 1.6899e+15 2857.8
## - TRB.  1 1.6947e+13 1.6936e+15 2858.0
## - RPG   1 1.9044e+13 1.6957e+15 2858.2
## - TO.   1 2.1505e+13 1.6982e+15 2858.3
## - TOPG  1 2.9778e+13 1.7065e+15 2858.7
## - FTA   1 3.4853e+13 1.7115e+15 2859.0
## <none>               1.6767e+15 2859.1
## - X3P.  1 3.7093e+13 1.7138e+15 2859.1
## - eFG.  1 3.7265e+13 1.7139e+15 2859.1
## - DRTG  1 4.1776e+13 1.7185e+15 2859.4
## - USG.  1 4.2932e+13 1.7196e+15 2859.4
## - AST.  1 4.8895e+13 1.7256e+15 2859.8
## - X2P.  1 6.1746e+13 1.7384e+15 2860.4
## - APG   1 1.3577e+14 1.8124e+15 2864.3
## - PPG   1 1.5013e+14 1.8268e+15 2865.0
## - AGE   1 4.3117e+14 2.1078e+15 2878.2
##
## Step:  AIC=2857.11
## Salary ~ AGE + MPG + MIN. + USG. + TO. + FTA + FT. + X2PA + X2P. +
##     X3PA + X3P. + eFG. + PPG + RPG + TRB. + APG + AST. + SPG +
##     BPG. + TOPG + VI + ORTG + DRTG
```

```
##
##        Df  Sum of Sq        RSS    AIC
## - X2PA  1 3.0045e+11 1.6770e+15 2855.1
## - FT.   1 1.9486e+12 1.6786e+15 2855.2
## - MIN.  1 3.4219e+12 1.6801e+15 2855.3
## - MPG   1 4.2107e+12 1.6809e+15 2855.3
## - ORTG  1 6.1131e+12 1.6828e+15 2855.4
## - SPG   1 1.0778e+13 1.6875e+15 2855.7
## - BPG.  1 1.1936e+13 1.6886e+15 2855.8
## - VI    1 1.3450e+13 1.6901e+15 2855.8
## - TRB.  1 1.8312e+13 1.6950e+15 2856.1
## - RPG   1 1.9515e+13 1.6962e+15 2856.2
## - TO.   1 2.1630e+13 1.6983e+15 2856.3
## - X3PA  1 2.9351e+13 1.7060e+15 2856.7
## - TOPG  1 3.0110e+13 1.7068e+15 2856.8
## - FTA   1 3.4853e+13 1.7115e+15 2857.0
## <none>                1.6767e+15 2857.1
## - X3P.  1 3.7106e+13 1.7138e+15 2857.1
## - eFG.  1 3.7326e+13 1.7140e+15 2857.1
## - USG.  1 4.3496e+13 1.7202e+15 2857.5
## - DRTG  1 4.6505e+13 1.7232e+15 2857.6
## - AST.  1 4.9419e+13 1.7261e+15 2857.8
## - X2P.  1 6.1819e+13 1.7385e+15 2858.4
## - APG   1 1.3649e+14 1.8132e+15 2862.3
## - PPG   1 2.1950e+14 1.8962e+15 2866.4
## - AGE   1 4.3305e+14 2.1097e+15 2876.2
##
## Step:  AIC=2855.13
## Salary ~ AGE + MPG + MIN. + USG. + TO. + FTA + FT. + X2P. + X3PA +
##     X3P. + eFG. + PPG + RPG + TRB. + APG + AST. + SPG + BPG. +
##     TOPG + VI + ORTG + DRTG
##
##        Df  Sum of Sq        RSS    AIC
## - FT.   1 1.7265e+12 1.6787e+15 2853.2
## - MIN.  1 3.8675e+12 1.6809e+15 2853.3
## - MPG   1 4.7090e+12 1.6817e+15 2853.4
## - ORTG  1 7.5321e+12 1.6845e+15 2853.5
## - BPG.  1 1.1639e+13 1.6886e+15 2853.8
## - VI    1 1.3329e+13 1.6903e+15 2853.9
## - SPG   1 1.4674e+13 1.6917e+15 2853.9
## - TRB.  1 1.8623e+13 1.6956e+15 2854.1
## - RPG   1 1.9310e+13 1.6963e+15 2854.2
## - TO.   1 2.2761e+13 1.6997e+15 2854.4
## - TOPG  1 2.9811e+13 1.7068e+15 2854.8
## - X3PA  1 2.9854e+13 1.7068e+15 2854.8
## <none>                1.6770e+15 2855.1
## - eFG.  1 3.7105e+13 1.7141e+15 2855.1
## - X3P.  1 3.9621e+13 1.7166e+15 2855.3
## - USG.  1 4.3196e+13 1.7202e+15 2855.5
## - FTA   1 4.3886e+13 1.7209e+15 2855.5
## - DRTG  1 4.6330e+13 1.7233e+15 2855.6
## - AST.  1 4.9483e+13 1.7265e+15 2855.8
## - X2P.  1 6.4629e+13 1.7416e+15 2856.6
## - APG   1 1.3626e+14 1.8132e+15 2860.3
```

```
## - PPG   1 2.2042e+14 1.8974e+15 2864.5
## - AGE   1 4.3395e+14 2.1109e+15 2874.3
##
## Step:  AIC=2853.22
## Salary ~ AGE + MPG + MIN. + USG. + TO. + FTA + X2P. + X3PA +
##     X3P. + eFG. + PPG + RPG + TRB. + APG + AST. + SPG + BPG. +
##     TOPG + VI + ORTG + DRTG
##
##        Df  Sum of Sq        RSS    AIC
## - MIN.  1 4.2829e+12 1.6830e+15 2851.5
## - MPG   1 5.1953e+12 1.6839e+15 2851.5
## - VI    1 1.2043e+13 1.6908e+15 2851.9
## - BPG.  1 1.4601e+13 1.6933e+15 2852.0
## - ORTG  1 1.4783e+13 1.6935e+15 2852.0
## - SPG   1 1.5065e+13 1.6938e+15 2852.0
## - RPG   1 1.8852e+13 1.6976e+15 2852.2
## - TRB.  1 2.0629e+13 1.6993e+15 2852.3
## - TO.   1 2.5403e+13 1.7041e+15 2852.6
## - X3PA  1 2.8491e+13 1.7072e+15 2852.8
## - TOPG  1 2.8936e+13 1.7076e+15 2852.8
## - eFG.  1 3.5447e+13 1.7142e+15 2853.1
## <none>              1.6787e+15 2853.2
## - FTA   1 4.2170e+13 1.7209e+15 2853.5
## - X3P.  1 4.2400e+13 1.7211e+15 2853.5
## - DRTG  1 4.5241e+13 1.7240e+15 2853.7
## - AST.  1 4.7956e+13 1.7267e+15 2853.8
## - USG.  1 4.8729e+13 1.7274e+15 2853.8
## - X2P.  1 6.5805e+13 1.7445e+15 2854.8
## - APG   1 1.3770e+14 1.8164e+15 2858.5
## - PPG   1 2.3556e+14 1.9143e+15 2863.3
## - AGE   1 4.4093e+14 2.1196e+15 2872.7
##
## Step:  AIC=2851.46
## Salary ~ AGE + MPG + USG. + TO. + FTA + X2P. + X3PA + X3P. +
##     eFG. + PPG + RPG + TRB. + APG + AST. + SPG + BPG. + TOPG +
##     VI + ORTG + DRTG
##
##        Df  Sum of Sq        RSS    AIC
## - VI    1 1.2267e+13 1.6953e+15 2850.1
## - SPG   1 1.2351e+13 1.6953e+15 2850.1
## - ORTG  1 1.3148e+13 1.6961e+15 2850.2
## - BPG.  1 1.5592e+13 1.6986e+15 2850.3
## - RPG   1 1.7167e+13 1.7002e+15 2850.4
## - TRB.  1 1.7777e+13 1.7008e+15 2850.4
## - TO.   1 2.3055e+13 1.7061e+15 2850.7
## - TOPG  1 2.6993e+13 1.7100e+15 2850.9
## - X3PA  1 2.7776e+13 1.7108e+15 2851.0
## <none>              1.6830e+15 2851.5
## - eFG.  1 3.9287e+13 1.7223e+15 2851.6
## - DRTG  1 4.1736e+13 1.7247e+15 2851.7
## - X3P.  1 4.4605e+13 1.7276e+15 2851.9
## - FTA   1 4.5426e+13 1.7284e+15 2851.9
## - USG.  1 4.6696e+13 1.7297e+15 2852.0
## - AST.  1 5.8516e+13 1.7415e+15 2852.6
```

```
## - X2P.  1 6.9871e+13 1.7529e+15 2853.2
## - MPG   1 1.0812e+14 1.7911e+15 2855.2
## - APG   1 1.6971e+14 1.8527e+15 2858.3
## - PPG   1 2.3538e+14 1.9184e+15 2861.5
## - AGE   1 4.4189e+14 2.1249e+15 2870.9
##
## Step:  AIC=2850.12
## Salary ~ AGE + MPG + USG. + TO. + FTA + X2P. + X3PA + X3P. +
##     eFG. + PPG + RPG + TRB. + APG + AST. + SPG + BPG. + TOPG +
##     ORTG + DRTG
##
##        Df  Sum of Sq        RSS    AIC
## - RPG   1 7.4657e+12 1.7027e+15 2848.5
## - ORTG  1 9.7804e+12 1.7050e+15 2848.7
## - SPG   1 1.4784e+13 1.7100e+15 2848.9
## - BPG.  1 1.6595e+13 1.7119e+15 2849.0
## - TO.   1 1.8130e+13 1.7134e+15 2849.1
## - TOPG  1 2.4404e+13 1.7197e+15 2849.4
## - TRB.  1 2.6378e+13 1.7216e+15 2849.5
## - X3PA  1 2.7231e+13 1.7225e+15 2849.6
## - USG.  1 3.4457e+13 1.7297e+15 2850.0
## - eFG.  1 3.5549e+13 1.7308e+15 2850.0
## <none>                1.6953e+15 2850.1
## - DRTG  1 4.2978e+13 1.7382e+15 2850.4
## - FTA   1 4.6085e+13 1.7413e+15 2850.6
## - AST.  1 4.8538e+13 1.7438e+15 2850.7
## - X3P.  1 5.1397e+13 1.7467e+15 2850.9
## - X2P.  1 6.3322e+13 1.7586e+15 2851.5
## - MPG   1 1.3887e+14 1.8341e+15 2855.4
## - APG   1 1.7926e+14 1.8745e+15 2857.4
## - PPG   1 2.4100e+14 1.9363e+15 2860.3
## - AGE   1 4.3009e+14 2.1254e+15 2868.9
##
## Step:  AIC=2848.53
## Salary ~ AGE + MPG + USG. + TO. + FTA + X2P. + X3PA + X3P. +
##     eFG. + PPG + TRB. + APG + AST. + SPG + BPG. + TOPG + ORTG +
##     DRTG
##
##        Df  Sum of Sq        RSS    AIC
## - ORTG  1 4.9788e+12 1.7077e+15 2846.8
## - TO.   1 1.2717e+13 1.7154e+15 2847.2
## - SPG   1 1.4510e+13 1.7172e+15 2847.3
## - BPG.  1 1.6924e+13 1.7197e+15 2847.4
## - TOPG  1 2.2468e+13 1.7252e+15 2847.7
## - USG.  1 3.0018e+13 1.7327e+15 2848.1
## - X3PA  1 3.0331e+13 1.7331e+15 2848.2
## - TRB.  1 3.6434e+13 1.7392e+15 2848.5
## <none>                1.7027e+15 2848.5
## - DRTG  1 3.7682e+13 1.7404e+15 2848.5
## - eFG.  1 4.1909e+13 1.7446e+15 2848.8
## - X3P.  1 4.8002e+13 1.7507e+15 2849.1
## - AST.  1 6.1319e+13 1.7640e+15 2849.8
## - X2P.  1 6.2787e+13 1.7655e+15 2849.9
## - FTA   1 6.3498e+13 1.7662e+15 2849.9
```

```
## - APG   1 1.8457e+14 1.8873e+15 2856.0
## - PPG   1 2.5292e+14 1.9557e+15 2859.3
## - MPG   1 2.5981e+14 1.9625e+15 2859.6
## - AGE   1 4.3318e+14 2.1359e+15 2867.4
##
## Step:  AIC=2846.8
## Salary ~ AGE + MPG + USG. + TO. + FTA + X2P. + X3PA + X3P. +
##     eFG. + PPG + TRB. + APG + AST. + SPG + BPG. + TOPG + DRTG
##
##        Df  Sum of Sq        RSS    AIC
## - TO.   1 8.9573e+12 1.7167e+15 2845.3
## - SPG   1 1.2142e+13 1.7198e+15 2845.4
## - BPG.  1 1.6046e+13 1.7238e+15 2845.7
## - USG.  1 2.5550e+13 1.7333e+15 2846.2
## - TRB.  1 3.1680e+13 1.7394e+15 2846.5
## - TOPG  1 3.5971e+13 1.7437e+15 2846.7
## <none>               1.7077e+15 2846.8
## - DRTG  1 3.8466e+13 1.7462e+15 2846.8
## - X3PA  1 3.8977e+13 1.7467e+15 2846.9
## - X3P.  1 4.6187e+13 1.7539e+15 2847.2
## - X2P.  1 6.6998e+13 1.7747e+15 2848.3
## - eFG.  1 7.8008e+13 1.7857e+15 2848.9
## - AST.  1 9.8794e+13 1.8065e+15 2850.0
## - FTA   1 1.0427e+14 1.8120e+15 2850.2
## - APG   1 1.8852e+14 1.8962e+15 2854.4
## - PPG   1 2.4823e+14 1.9559e+15 2857.3
## - MPG   1 2.6367e+14 1.9714e+15 2858.0
## - AGE   1 4.2879e+14 2.1365e+15 2865.4
##
## Step:  AIC=2845.28
## Salary ~ AGE + MPG + USG. + FTA + X2P. + X3PA + X3P. + eFG. +
##     PPG + TRB. + APG + AST. + SPG + BPG. + TOPG + DRTG
##
##        Df  Sum of Sq        RSS    AIC
## - SPG   1 6.4805e+12 1.7231e+15 2843.6
## - USG.  1 1.7397e+13 1.7341e+15 2844.2
## - BPG.  1 1.7427e+13 1.7341e+15 2844.2
## - X3PA  1 3.4080e+13 1.7507e+15 2845.1
## - DRTG  1 3.6444e+13 1.7531e+15 2845.2
## <none>               1.7167e+15 2845.3
## - TOPG  1 3.8445e+13 1.7551e+15 2845.3
## - TRB.  1 3.9448e+13 1.7561e+15 2845.4
## - X3P.  1 4.4180e+13 1.7608e+15 2845.6
## - X2P.  1 6.3334e+13 1.7800e+15 2846.6
## - eFG.  1 7.5214e+13 1.7919e+15 2847.2
## - FTA   1 9.9521e+13 1.8162e+15 2848.5
## - AST.  1 1.6212e+14 1.8788e+15 2851.6
## - MPG   1 2.5635e+14 1.9730e+15 2856.1
## - PPG   1 2.8728e+14 2.0039e+15 2857.5
## - APG   1 3.1761e+14 2.0343e+15 2858.9
## - AGE   1 4.2060e+14 2.1373e+15 2863.4
##
## Step:  AIC=2843.62
## Salary ~ AGE + MPG + USG. + FTA + X2P. + X3PA + X3P. + eFG. +
```

```
##     PPG + TRB. + APG + AST. + BPG. + TOPG + DRTG
##
##        Df  Sum of Sq        RSS    AIC
## - USG.  1 1.8520e+13 1.7417e+15 2842.6
## - BPG.  1 2.2285e+13 1.7454e+15 2842.8
## - X3PA  1 3.1174e+13 1.7543e+15 2843.3
## - DRTG  1 3.2074e+13 1.7552e+15 2843.3
## - TOPG  1 3.7420e+13 1.7606e+15 2843.6
## <none>               1.7231e+15 2843.6
## - TRB.  1 3.8406e+13 1.7616e+15 2843.7
## - X3P.  1 4.0213e+13 1.7634e+15 2843.8
## - X2P.  1 6.0084e+13 1.7832e+15 2844.8
## - eFG.  1 7.0505e+13 1.7937e+15 2845.3
## - FTA   1 9.6179e+13 1.8193e+15 2846.6
## - AST.  1 1.6591e+14 1.8891e+15 2850.1
## - MPG   1 2.5103e+14 1.9742e+15 2854.1
## - PPG   1 2.8605e+14 2.0092e+15 2855.8
## - APG   1 3.4779e+14 2.0709e+15 2858.5
## - AGE   1 4.1420e+14 2.1373e+15 2861.4
##
## Step:  AIC=2842.61
## Salary ~ AGE + MPG + FTA + X2P. + X3PA + X3P. + eFG. + PPG +
##     TRB. + APG + AST. + BPG. + TOPG + DRTG
##
##        Df  Sum of Sq        RSS    AIC
## - BPG.  1 1.8962e+13 1.7606e+15 2841.6
## - TOPG  1 2.4040e+13 1.7657e+15 2841.9
## - X3PA  1 2.6753e+13 1.7684e+15 2842.0
## - X3P.  1 3.1061e+13 1.7727e+15 2842.2
## <none>               1.7417e+15 2842.6
## - DRTG  1 4.5245e+13 1.7869e+15 2843.0
## - X2P.  1 5.3867e+13 1.7955e+15 2843.4
## - TRB.  1 5.6243e+13 1.7979e+15 2843.5
## - eFG.  1 5.7697e+13 1.7994e+15 2843.6
## - FTA   1 8.2150e+13 1.8238e+15 2844.8
## - AST.  1 1.9011e+14 1.9318e+15 2850.1
## - MPG   1 2.8399e+14 2.0257e+15 2854.5
## - APG   1 3.7216e+14 2.1138e+15 2858.4
## - AGE   1 4.3657e+14 2.1782e+15 2861.2
## - PPG   1 4.9314e+14 2.2348e+15 2863.5
##
## Step:  AIC=2841.6
## Salary ~ AGE + MPG + FTA + X2P. + X3PA + X3P. + eFG. + PPG +
##     TRB. + APG + AST. + TOPG + DRTG
##
##        Df  Sum of Sq        RSS    AIC
## - X3PA  1 1.3983e+13 1.7746e+15 2840.3
## - TOPG  1 1.7285e+13 1.7779e+15 2840.5
## - X3P.  1 2.6215e+13 1.7868e+15 2841.0
## <none>               1.7606e+15 2841.6
## - DRTG  1 3.8952e+13 1.7996e+15 2841.6
## - X2P.  1 5.3700e+13 1.8143e+15 2842.4
## - eFG.  1 5.5985e+13 1.8166e+15 2842.5
## - TRB.  1 5.9117e+13 1.8197e+15 2842.6
```

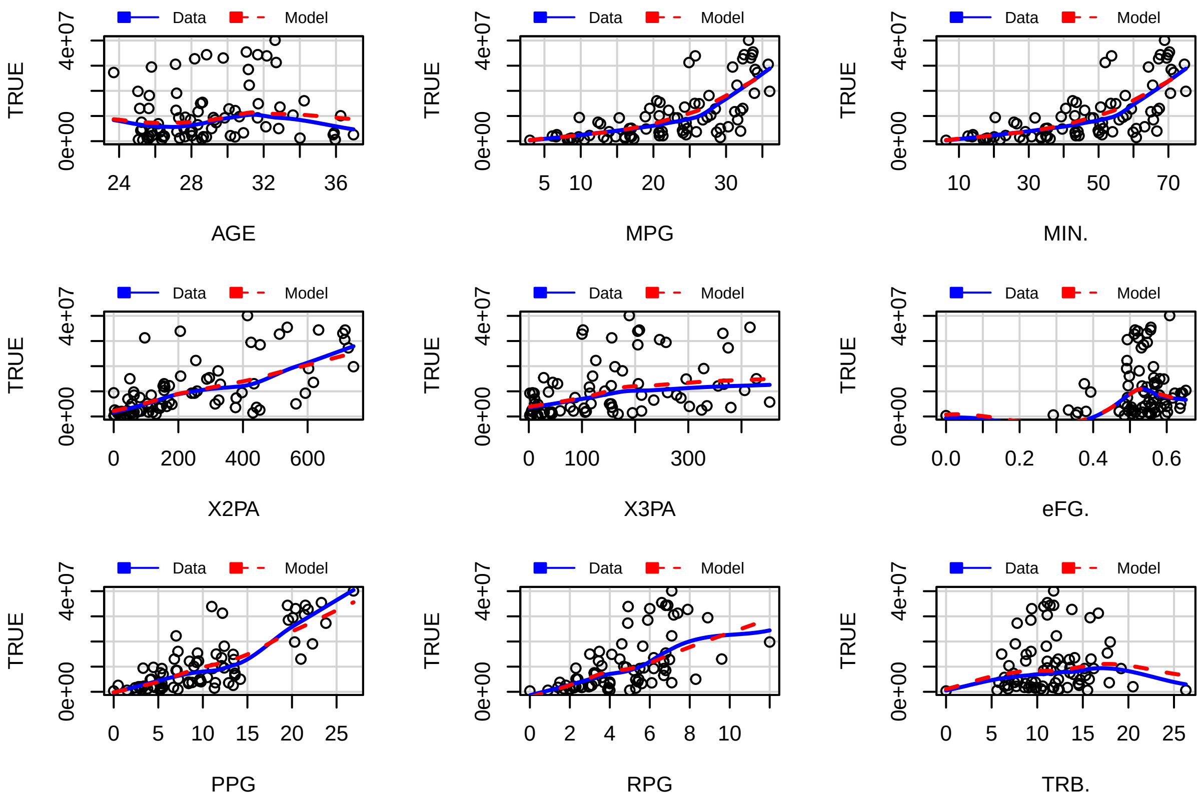
```
## - FTA   1 8.2740e+13 1.8434e+15 2843.8
## - AST.  1 1.9778e+14 1.9584e+15 2849.4
## - MPG   1 2.6604e+14 2.0267e+15 2852.6
## - APG   1 3.7739e+14 2.1380e+15 2857.5
## - AGE   1 4.5512e+14 2.2157e+15 2860.8
## - PPG   1 5.7591e+14 2.3365e+15 2865.6
##
## Step:  AIC=2840.33
## Salary ~ AGE + MPG + FTA + X2P. + X3P. + eFG. + PPG + TRB. +
##     APG + AST. + TOPG + DRTG
##
##        Df  Sum of Sq        RSS    AIC
## - TOPG  1 1.0335e+13 1.7849e+15 2838.9
## - X3P.  1 1.9457e+13 1.7941e+15 2839.3
## <none>              1.7746e+15 2840.3
## - DRTG  1 3.9913e+13 1.8145e+15 2840.4
## - X2P.  1 4.2562e+13 1.8172e+15 2840.5
## - eFG.  1 4.4936e+13 1.8195e+15 2840.6
## - TRB.  1 5.9444e+13 1.8341e+15 2841.4
## - FTA   1 6.9842e+13 1.8445e+15 2841.9
## - AST.  1 1.9614e+14 1.9708e+15 2848.0
## - MPG   1 2.5208e+14 2.0267e+15 2850.6
## - APG   1 3.7068e+14 2.1453e+15 2855.8
## - AGE   1 4.8336e+14 2.2580e+15 2860.5
## - PPG   1 6.9223e+14 2.4668e+15 2868.6
##
## Step:  AIC=2838.87
## Salary ~ AGE + MPG + FTA + X2P. + X3P. + eFG. + PPG + TRB. +
##     APG + AST. + DRTG
##
##        Df  Sum of Sq        RSS    AIC
## - X3P.  1 2.1551e+13 1.8065e+15 2838.0
## <none>              1.7849e+15 2838.9
## - DRTG  1 3.9261e+13 1.8242e+15 2838.9
## - X2P.  1 5.1939e+13 1.8369e+15 2839.5
## - eFG.  1 5.3269e+13 1.8382e+15 2839.6
## - FTA   1 6.2386e+13 1.8473e+15 2840.0
## - TRB.  1 6.9397e+13 1.8543e+15 2840.4
## - AST.  1 1.9534e+14 1.9803e+15 2846.4
## - MPG   1 2.5270e+14 2.0376e+15 2849.1
## - APG   1 4.2941e+14 2.2144e+15 2856.7
## - AGE   1 5.2290e+14 2.3078e+15 2860.5
## - PPG   1 8.7930e+14 2.6642e+15 2873.7
##
## Step:  AIC=2837.97
## Salary ~ AGE + MPG + FTA + X2P. + eFG. + PPG + TRB. + APG + AST. +
##     DRTG
##
##        Df  Sum of Sq        RSS    AIC
## - X2P.  1 3.3013e+13 1.8395e+15 2837.6
## <none>              1.8065e+15 2838.0
## - DRTG  1 5.2738e+13 1.8592e+15 2838.6
## - eFG.  1 5.6758e+13 1.8633e+15 2838.8
## - FTA   1 6.3485e+13 1.8700e+15 2839.2
```

```
## - TRB.  1 8.1530e+13 1.8880e+15 2840.0
## - AST.  1 1.8257e+14 1.9891e+15 2844.8
## - MPG   1 2.5519e+14 2.0617e+15 2848.1
## - APG   1 4.1289e+14 2.2194e+15 2854.9
## - AGE   1 5.5258e+14 2.3591e+15 2860.5
## - PPG   1 9.0904e+14 2.7155e+15 2873.5
##
## Step:  AIC=2837.64
## Salary ~ AGE + MPG + FTA + eFG. + PPG + TRB. + APG + AST. + DRTG
##
##        Df  Sum of Sq        RSS    AIC
## - eFG.  1 2.4803e+13 1.8643e+15 2836.9
## <none>              1.8395e+15 2837.6
## - DRTG  1 4.3742e+13 1.8833e+15 2837.8
## - FTA   1 5.7428e+13 1.8969e+15 2838.5
## - TRB.  1 9.3641e+13 1.9332e+15 2840.2
## - AST.  1 1.7483e+14 2.0143e+15 2844.0
## - MPG   1 2.5036e+14 2.0899e+15 2847.4
## - APG   1 4.0636e+14 2.2459e+15 2854.0
## - AGE   1 5.2960e+14 2.3691e+15 2858.9
## - PPG   1 8.8583e+14 2.7253e+15 2871.8
##
## Step:  AIC=2836.87
## Salary ~ AGE + MPG + FTA + PPG + TRB. + APG + AST. + DRTG
##
##        Df  Sum of Sq        RSS    AIC
## <none>              1.8643e+15 2836.9
## - FTA   1 4.4999e+13 1.9093e+15 2837.1
## - DRTG  1 6.9768e+13 1.9341e+15 2838.2
## - TRB.  1 1.0628e+14 1.9706e+15 2840.0
## - AST.  1 1.5003e+14 2.0143e+15 2842.0
## - MPG   1 2.2785e+14 2.0922e+15 2845.5
## - APG   1 3.8187e+14 2.2462e+15 2852.0
## - AGE   1 5.1784e+14 2.3822e+15 2857.4
## - PPG   1 8.9711e+14 2.7614e+15 2871.0

##
## Call:
## lm(formula = Salary ~ AGE + MPG + FTA + PPG + TRB. + APG + AST. +
##     DRTG, data = g.fit.train)
##
## Coefficients:
## (Intercept)          AGE          MPG          FTA          PPG         TRB.
##   -54681999       861296      -586434       -17539      1428627       563348
##         APG         AST.         DRTG
##     4393450      -466052       281213
```

```r
# using the combination of varible with lowest AIC value and build a model
g.linearfit = lm(formula = Salary ~ AGE + MPG + X2P. + X3PA + TS. + PPG + TRB. +
    APG + VI, data = g.fit.train)


# check the accuracy of prediction
pre.g <- predict(g.linearfit, newdata = g.fit.test)
mean((pre.g - g.fit.test$Salary)^2)
```

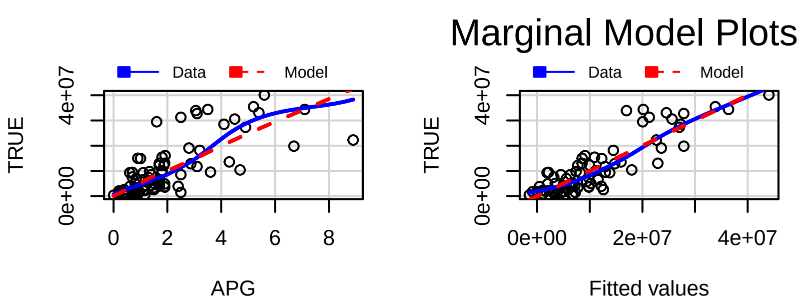
```
## [1] 3.727693e+13
```

```r
marginalModelPlots(f.linearfit)
```




```r
# ......lasso regression ...lasso regression for POS 'C'
library(glmnet)
```

```
## Loading required package: Matrix
## Loaded glmnet 4.1-3
```

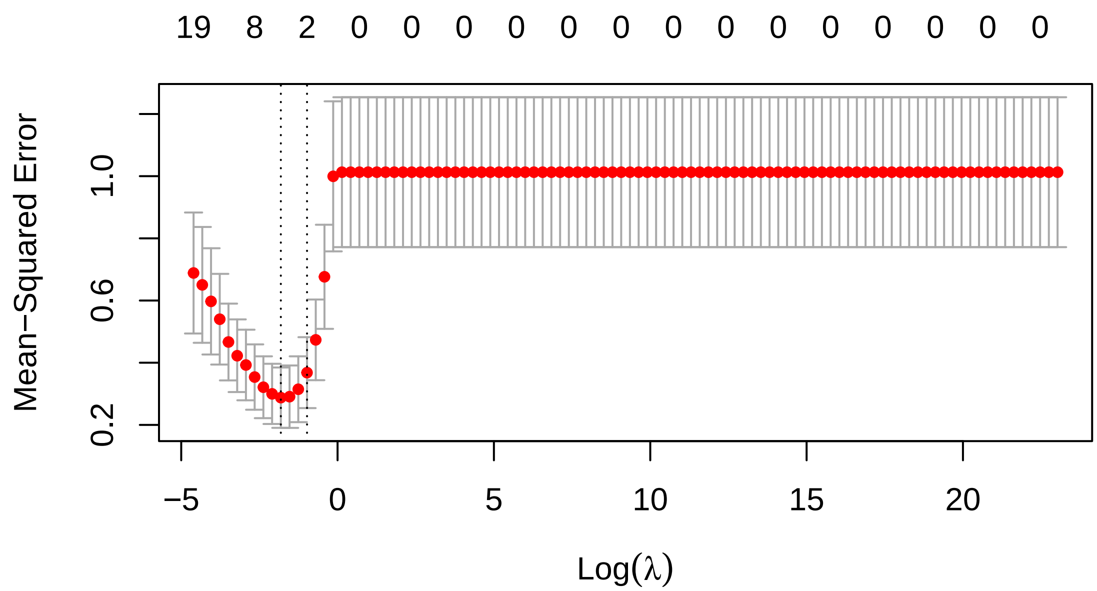



```r
c.fit.train <- main.pos.c.train[, 5:30]
c.fit.test <- main.pos.c.test[, 5:30]
library(caret)
```

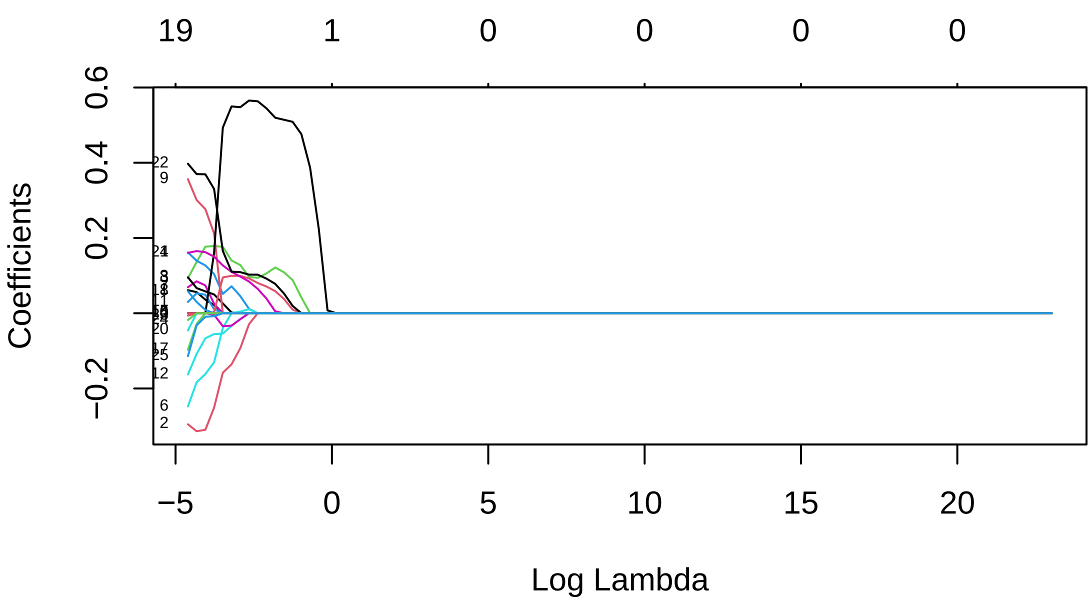
```
## Loading required package: ggplot2
## Loading required package: lattice
##
## Attaching package: 'lattice'
## The following object is masked from 'package:corrgram':
```

```
##
##     panel.fill
```

```r
# center and scale the data
scal <- preProcess(c.fit.train, method = c("center", "scale"))
c.fit.trains <- predict(scal, c.fit.train)
c.fit.tests <- predict(scal, c.fit.test)

# build the model with the best lambda parameter
grid = 10^seq(10, -2, length = 100)
X <- as.matrix(c.fit.trains[, 1:25])
Y <- c.fit.trains[, 26]
set.seed(1245)
c.lasso <- cv.glmnet(X, Y, alpha = 1, lambda = grid, nfolds = 10)
plot(c.lasso)
```




```r
plot(c.lasso$glmnet.fit, "lambda", label = T)
```



```r
c.lasso.min <- c.lasso$lambda.min
c.lasso.best <- glmnet(X, Y, alpha = 1, lambda = c.lasso.min)
coef(c.lasso.best)
```

```
## 26 x 1 sparse Matrix of class "dgCMatrix"
##                        s0
## (Intercept) -1.227050e-17
## AGE          .
## GP           .
## MPG          1.221473e-01
## MIN.         .
## USG.         .
## TO.          .
## FTA          .
## FT.          .
## X2PA         .
## X2P.         .
## X3PA         .
## X3P.         .
## eFG.         .
## TS.          .
## PPG          5.201153e-01
## RPG          5.851592e-02
## TRB.         .
## APG          .
## AST.         .
## SPG          .
## BPG.         4.913806e-03
## TOPG         7.760515e-02
## VI           .
## ORTG         .
## DRTG         .
```

```r
# check the accuracy of the prediction
c.fit.tests1 = as.matrix(c.fit.test[, 1:25])
c.test.pre <- predict(c.lasso.best, newx = c.fit.tests1)
library(mlr3measures)
```

```
## In order to avoid name clashes, do not attach 'mlr3measures'. Instead, only load the namespace with
```

```
##
## Attaching package: 'mlr3measures'
```

```
## The following objects are masked from 'package:caret':
##
##     precision, recall, sensitivity, specificity
```

```
## The following object is masked from 'package:MASS':
##
##     fbeta
```

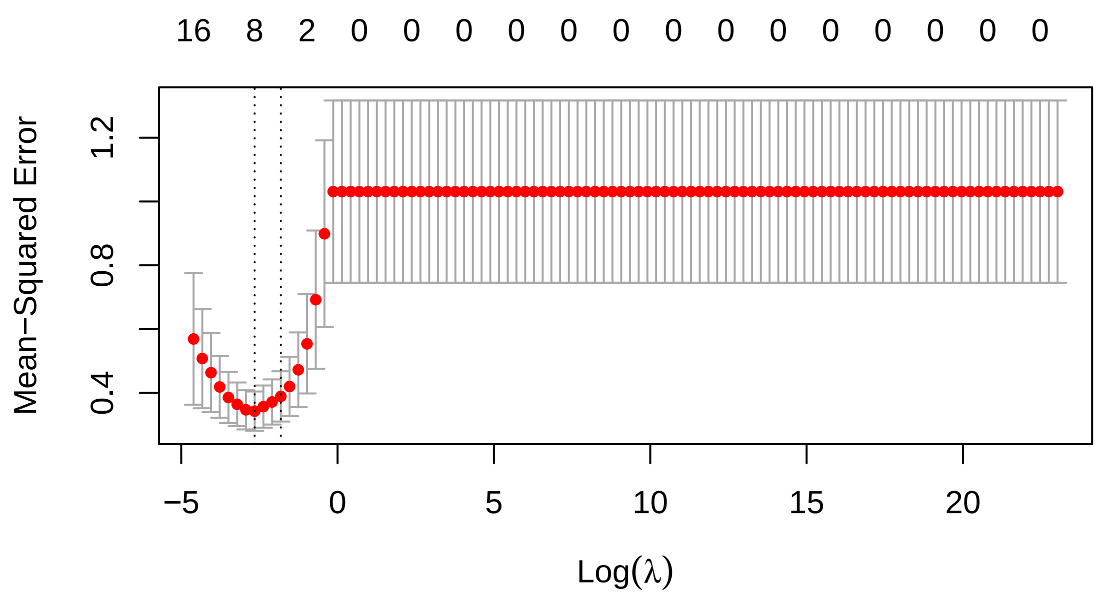
```r
sprintf("mean absolute error after standardization: %f", mae(c.fit.tests$Salary,
    c.test.pre))
```

```
## [1] "mean absolute error after standardization: 8.270367"
```

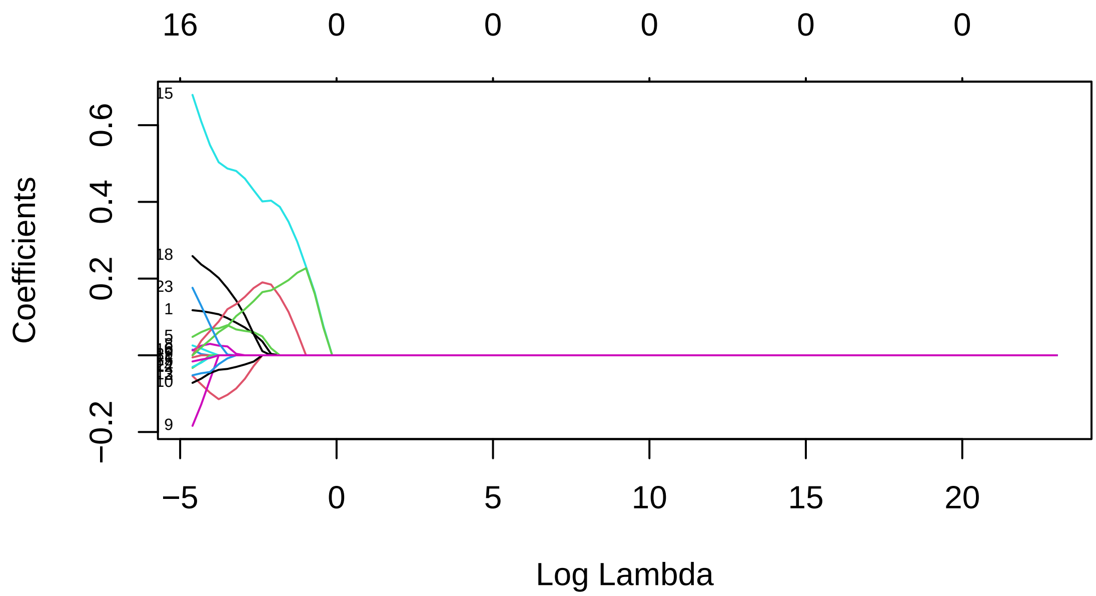
```
test_pre_o <- as.vector(c.test.pre[, 1] * scal$std[26] + scal$mean[26])
sprintf("mean absolute error before standardization: %f", mae(c.fit.test$Salary,
    test_pre_o))
```

```
## [1] "mean absolute error before standardization: 84969131.728028"
```

```
# ...lasso regression for POS 'F'
library(glmnet)
f.fit.train <- main.pos.f.train[, 5:30]
f.fit.test <- main.pos.f.test[, 5:30]
library(caret)

# center and scale the data
scal <- preProcess(f.fit.train, method = c("center", "scale"))
f.fit.trains <- predict(scal, f.fit.train)
f.fit.tests <- predict(scal, f.fit.test)

# build the model with the best lambda parameter
grid = 10^seq(10, -2, length = 100)
X <- as.matrix(f.fit.trains[, 1:25])
Y <- f.fit.trains[, 26]
set.seed(1245)
f.lasso <- cv.glmnet(X, Y, alpha = 1, lambda = grid, nfolds = 10)
plot(f.lasso)
```




```
plot(f.lasso$glmnet.fit, "lambda", label = T)
```




```
f.lasso.min <- f.lasso$lambda.min
f.lasso.best <- glmnet(X, Y, alpha = 1, lambda = f.lasso.min)
coef(f.lasso.best)
```

```
## 26 x 1 sparse Matrix of class "dgCMatrix"
##                        s0
## (Intercept)  6.095936e-17
## AGE          5.618338e-02
## GP          -2.892345e-02
## MPG          .
## MIN.         .
## USG.         6.315251e-02
## TO.          .
## FTA          .
## FT.          .
## X2PA         .
## X2P.        -1.606911e-02
## X3PA         .
## X3P.         .
## eFG.         .
## TS.          .
## PPG          4.293303e-01
## RPG          .
## TRB.         .
## APG          6.843523e-02
## AST.         1.672493e-01
## SPG          .
## BPG.         .
## TOPG         1.358074e-01
## VI           .
## ORTG         .
## DRTG         .
```

```
# check the accuracy of the prediction
f.fit.tests1 = as.matrix(f.fit.test[, 1:25])
f.test.pre <- predict(f.lasso.best, newx = f.fit.tests1)
```

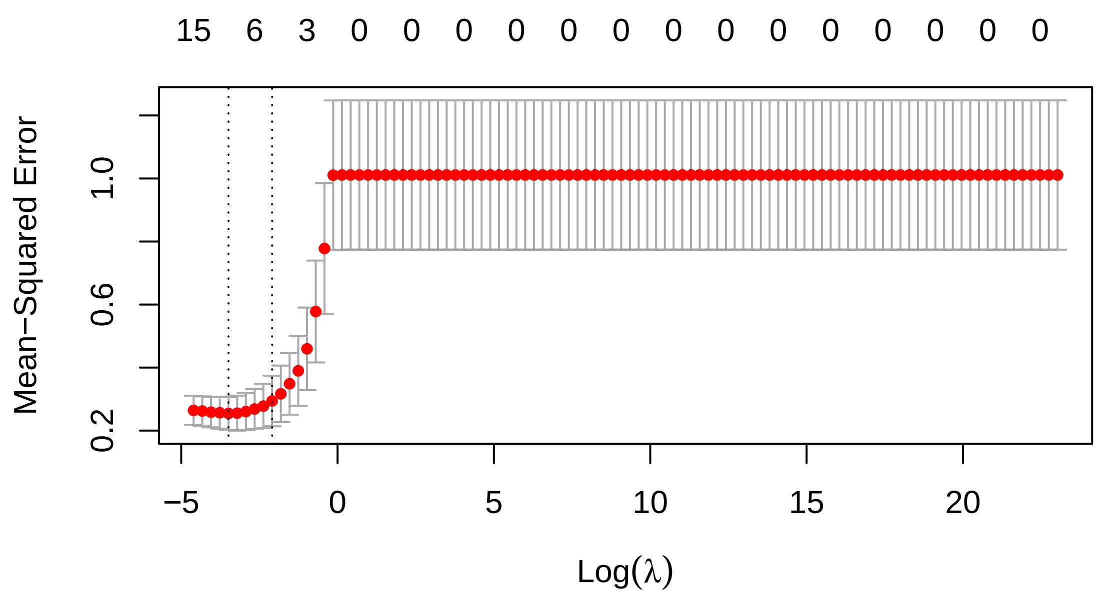
```r
library(mlr3measures)
sprintf("mean absolute error after standardization: %f", mae(f.fit.tests$Salary,
    f.test.pre))
```

```
## [1] "mean absolute error after standardization: 7.974427"
```

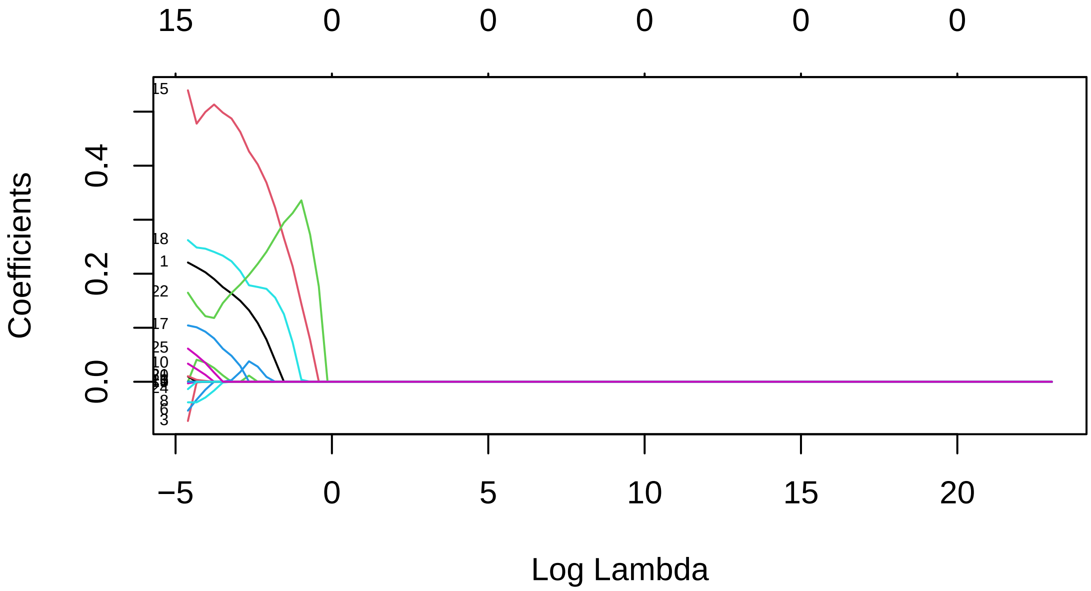
```r
test_pre_o <- as.vector(f.test.pre[, 1] * scal$std[26] + scal$mean[26])
sprintf("mean absolute error before standardization: %f", mae(f.fit.test$Salary,
    test_pre_o))
```

```
## [1] "mean absolute error before standardization: 85671715.156205"
```

```r
# ...lasso regression for POS 'G'
library(glmnet)
g.fit.train <- main.pos.g.train[, 5:30]
g.fit.test <- main.pos.g.test[, 5:30]
library(caret)

# center and scale the data
scal <- preProcess(g.fit.train, method = c("center", "scale"))
g.fit.trains <- predict(scal, g.fit.train)
g.fit.tests <- predict(scal, g.fit.test)

# build the model with the best lambda parameter
grid = 10^seq(10, -2, length = 100)
X <- as.matrix(g.fit.trains[, 1:25])
Y <- g.fit.trains[, 26]
set.seed(1245)
g.lasso <- cv.glmnet(X, Y, alpha = 1, lambda = grid, nfolds = 10)
plot(g.lasso)
```




```r
plot(g.lasso$glmnet.fit, "lambda", label = T)
```




```
g.lasso.min <- g.lasso$lambda.min
g.lasso.best <- glmnet(X, Y, alpha = 1, lambda = g.lasso.min)
coef(c.lasso.best)
```

```
## 26 x 1 sparse Matrix of class "dgCMatrix"
##                          s0
## (Intercept) -1.227050e-17
## AGE          .
## GP           .
## MPG          1.221473e-01
## MIN.         .
## USG.         .
## TO.          .
## FTA          .
## FT.          .
## X2PA         .
## X2P.         .
## X3PA         .
## X3P.         .
## eFG.         .
## TS.          .
## PPG          5.201153e-01
## RPG          5.851592e-02
## TRB.         .
## APG          .
## AST.         .
## SPG          .
## BPG.         4.913806e-03
## TOPG         7.760515e-02
## VI           .
## ORTG         .
## DRTG         .
```

```
# check the accuracy of the prediction
g.fit.tests1 = as.matrix(g.fit.test[, 1:25])
g.test.pre <- predict(g.lasso.best, newx = g.fit.tests1)
```

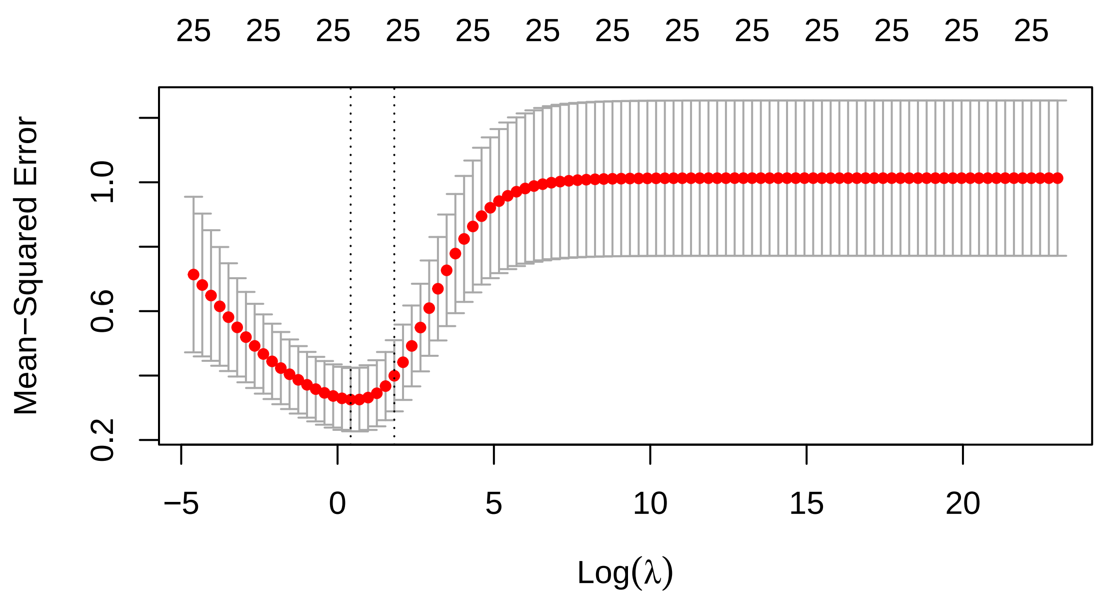
```r
library(mlr3measures)
sprintf("mean absolute error after standardization: %f", mae(g.fit.tests$Salary,
    g.test.pre))
```

```
## [1] "mean absolute error after standardization: 11.434472"
```

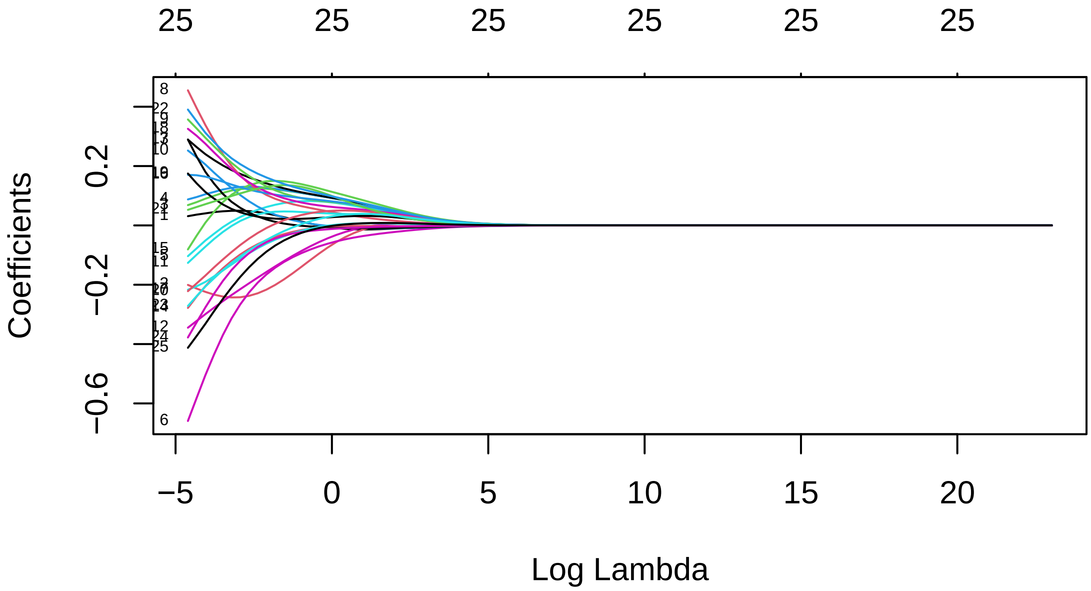
```r
test_pre_o <- as.vector(g.test.pre[, 1] * scal$std[26] + scal$mean[26])
sprintf("mean absolute error before standardization: %f", mae(g.fit.test$Salary,
    test_pre_o))
```

```
## [1] "mean absolute error before standardization: 124892906.026643"
```

```r
# ......ridge regression ...ridge regression for POS 'C'
library(glmnet)
c.fit.train <- main.pos.c.train[, 5:30]
c.fit.test <- main.pos.c.test[, 5:30]
library(caret)

# center and scale the data
scal <- preProcess(c.fit.train, method = c("center", "scale"))
c.fit.trains <- predict(scal, c.fit.train)
c.fit.tests <- predict(scal, c.fit.test)

# build the model with the best lambda parameter
grid = 10^seq(10, -2, length = 100)
X <- as.matrix(c.fit.trains[, 1:25])
Y <- c.fit.trains[, 26]
set.seed(1245)
c.ridge <- cv.glmnet(X, Y, alpha = 0, lambda = grid, nfolds = 10)
plot(c.ridge)
```




```r
plot(c.ridge$glmnet.fit, "lambda", label = T)
```




```r
c.ridge.min <- c.ridge$lambda.min
c.ridge.best <- glmnet(X, Y, alpha = 0, lambda = c.ridge.min)
coef(c.ridge.best)
```

```
## 26 x 1 sparse Matrix of class "dgCMatrix"
##                        s0
## (Intercept) -1.451766e-17
## AGE         -1.162385e-02
## GP          -3.994749e-02
## MPG          8.372485e-02
## MIN.         8.373680e-02
## USG.         7.111320e-02
## TO.         -4.689437e-02
## FTA          8.489800e-02
## FT.          3.679364e-02
## X2PA         7.444989e-02
## X2P.        -6.007538e-03
## X3PA         3.830703e-02
## X3P.        -2.218945e-02
## eFG.        -7.760405e-03
## TS.         -1.698393e-03
## PPG          1.016230e-01
## RPG          7.575903e-02
## TRB.         4.742928e-03
## APG          5.854676e-02
## AST.         3.032789e-02
## SPG          5.030390e-02
## BPG.         8.309024e-02
## TOPG         8.822283e-02
## VI           3.620137e-02
## ORTG        -9.235314e-03
## DRTG         3.396385e-03
```

```r
# check the accuracy of the prediction
c.test.pre <- predict(c.ridge.best, newx = as.matrix(c.fit.tests[, 1:25]))
library(mlr3measures)
```

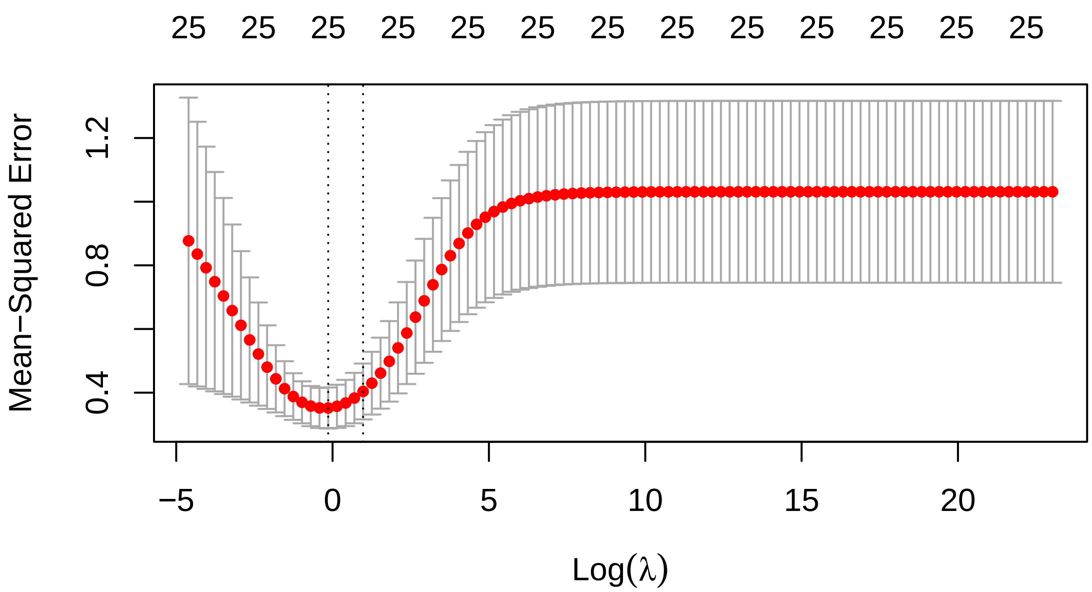
```r
sprintf("mean absolute error after standardization: %f", mae(c.fit.tests$Salary,
    c.test.pre))
```

```
## [1] "mean absolute error after standardization: 0.282627"
```

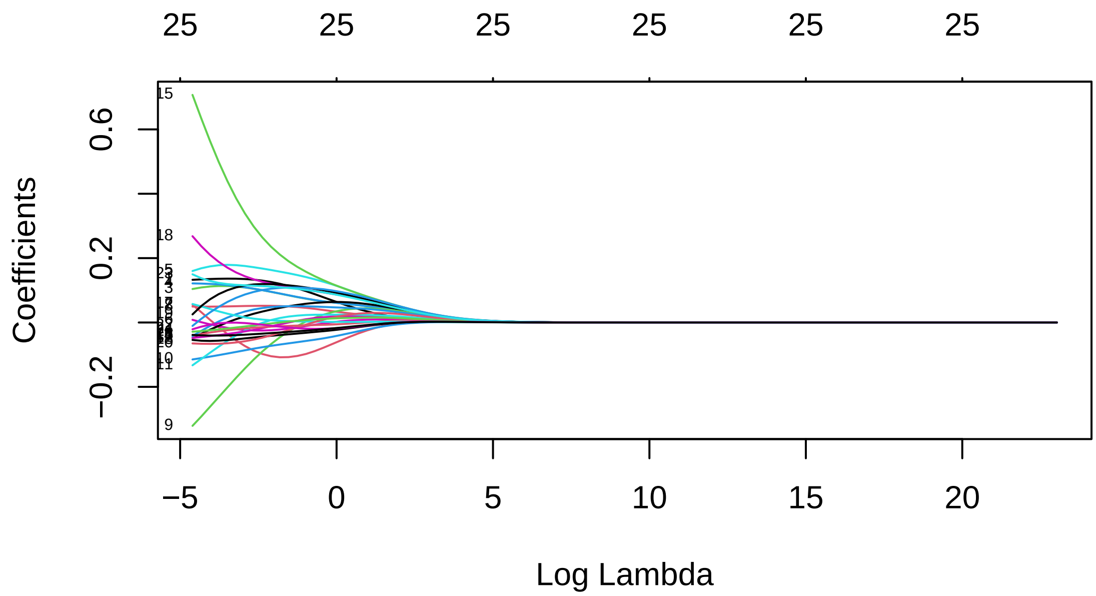
```r
test_pre_o <- as.vector(c.test.pre[, 1] * scal$std[11] + scal$mean[11])
sprintf("mean absolute error before standardization: %f", mae(c.fit.test$Salary,
    test_pre_o))
```

```
## [1] "mean absolute error before standardization: 11215653.802023"
```

```r
# ...ridge regression for POS 'F'
library(glmnet)
f.fit.train <- main.pos.f.train[, 5:30]
f.fit.test <- main.pos.f.test[, 5:30]
library(caret)

# center and scale the data
scal <- preProcess(f.fit.train, method = c("center", "scale"))
f.fit.trains <- predict(scal, f.fit.train)
f.fit.tests <- predict(scal, f.fit.test)

# build the model with the best lambda parameter
grid = 10^seq(10, -2, length = 100)
X <- as.matrix(f.fit.trains[, 1:25])
Y <- f.fit.trains[, 26]
set.seed(1245)
f.ridge <- cv.glmnet(X, Y, alpha = 0, lambda = grid, nfolds = 10)
plot(f.ridge)
```




```r
plot(f.ridge$glmnet.fit, "lambda", label = T)
```




```r
f.ridge.min <- f.ridge$lambda.min
f.ridge.best <- glmnet(X, Y, alpha = 0, lambda = f.ridge.min)
coef(f.ridge.best)
```

```
## 26 x 1 sparse Matrix of class "dgCMatrix"
##                        s0
## (Intercept)  3.618113e-17
## AGE          6.864519e-02
## GP          -6.700274e-02
## MPG          6.268046e-02
## MIN.         6.253836e-02
## USG.         1.183253e-01
## TO.          6.128762e-04
## FTA          6.357430e-02
## FT.          3.521234e-02
## X2PA         3.191836e-02
## X2P.        -4.445824e-02
## X3PA         2.427753e-02
## X3P.         1.807349e-02
## eFG.        -2.422528e-02
## TS.         -6.378917e-03
## PPG          1.196962e-01
## RPG          4.724108e-02
## TRB.         1.849145e-03
## APG          8.892079e-02
## AST.         9.483427e-02
## SPG          1.511128e-02
## BPG.         1.242033e-02
## TOPG         9.945924e-02
## VI           8.895540e-02
## ORTG        -1.837332e-02
## DRTG        -1.839699e-02
```

```r
# check the accuracy of the prediction
f.test.pre <- predict(f.ridge.best, newx = as.matrix(f.fit.tests[, 1:25]))
library(mlr3measures)
```

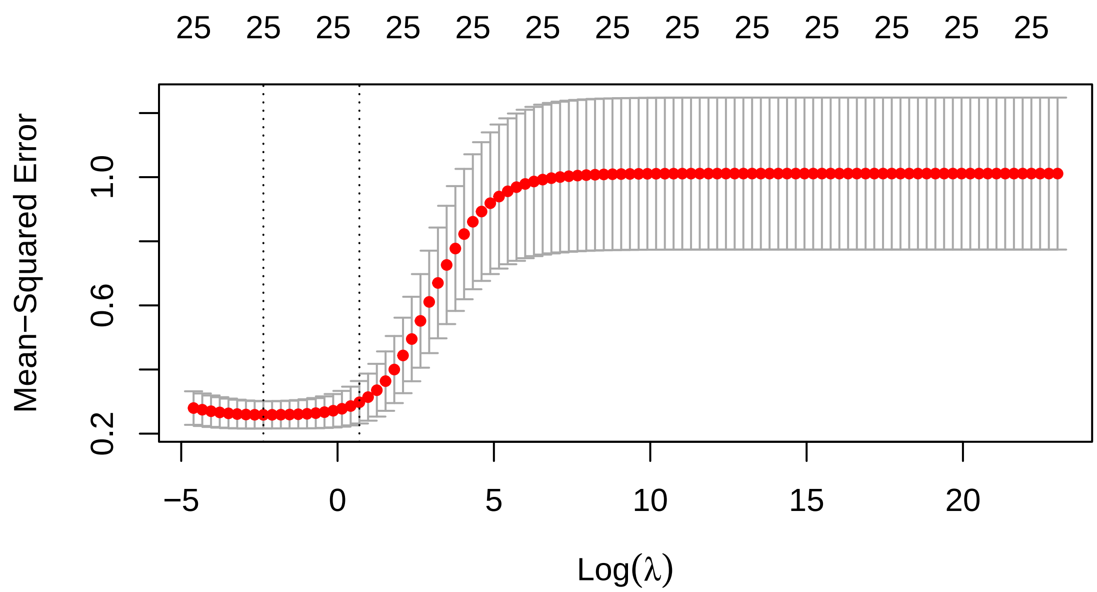
```r
sprintf("mean absolute error after standardization: %f", mae(f.fit.tests$Salary,
    f.test.pre))
```

```
## [1] "mean absolute error after standardization: 0.416047"
```

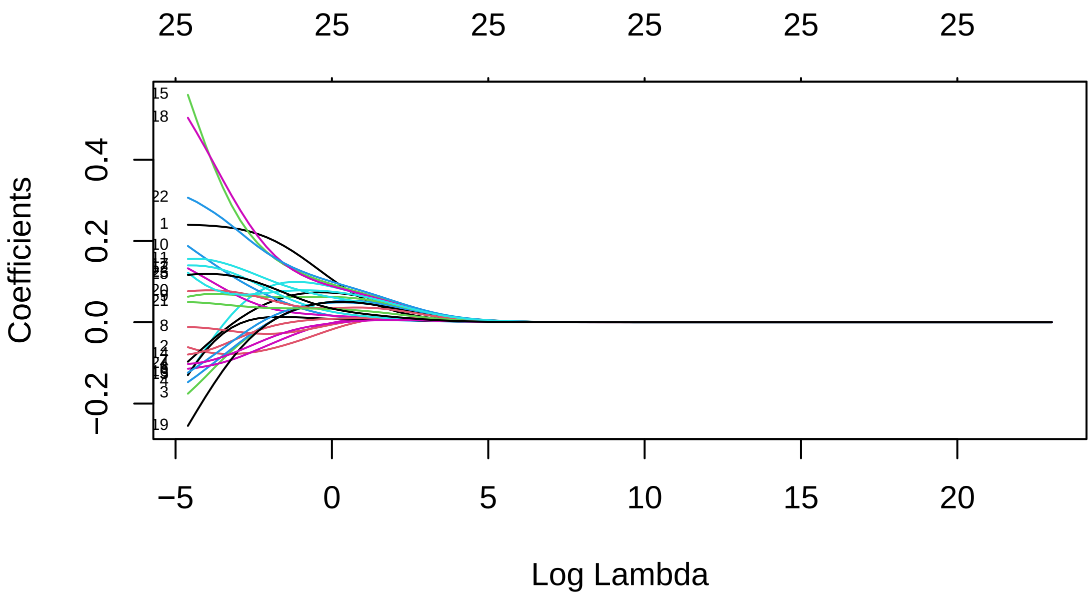
```r
test_pre_o <- as.vector(f.test.pre[, 1] * scal$std[26] + scal$mean[26])
sprintf("mean absolute error before standardization: %f", mae(f.fit.test$Salary,
    test_pre_o))
```

```
## [1] "mean absolute error before standardization: 4469724.039373"
```

```r
# ...ridge regression for POS 'G'
library(glmnet)
g.fit.train <- main.pos.g.train[, 5:30]
g.fit.test <- main.pos.g.test[, 5:30]
library(caret)

# center and scale the data
scal <- preProcess(g.fit.train, method = c("center", "scale"))
g.fit.trains <- predict(scal, g.fit.train)
g.fit.tests <- predict(scal, g.fit.test)

# build the model with the best lambda parameter
grid = 10^seq(10, -2, length = 100)
X <- as.matrix(g.fit.trains[, 1:25])
Y <- g.fit.trains[, 26]
set.seed(1245)
g.ridge <- cv.glmnet(X, Y, alpha = 0, lambda = grid, nfolds = 10)
plot(g.ridge)
```




```r
plot(g.ridge$glmnet.fit, "lambda", label = T)
```




```
g.ridge.min <- g.ridge$lambda.min
g.ridge.best <- glmnet(X, Y, alpha = 0, lambda = g.ridge.min)
coef(g.ridge.best)
```

```
## 26 x 1 sparse Matrix of class "dgCMatrix"
##                        s0
## (Intercept) -2.258617e-17
## AGE          2.174228e-01
## GP          -7.215619e-02
## MPG         -1.880459e-02
## MIN.        -1.688136e-02
## USG.         7.490516e-02
## TO.         -6.674509e-02
## FTA          3.639788e-02
## FT.         -2.777068e-02
## X2PA         6.440019e-02
## X2P.         7.831596e-02
## X3PA         1.156336e-01
## X3P.         4.382585e-02
## eFG.         9.689697e-03
## TS.         -2.020046e-02
## PPG          1.937493e-01
## RPG         -4.402235e-03
## TRB.         9.379257e-02
## APG          2.121888e-01
## AST.        -2.246214e-02
## SPG          6.250898e-02
## BPG.         3.649577e-02
## TOPG         1.852930e-01
## VI           6.973054e-02
## ORTG        -5.041559e-02
## DRTG         9.776647e-02
```

```
# check the accuracy of the prediction
g.test.pre <- predict(g.ridge.best, newx = as.matrix(g.fit.tests[, 1:25]))
library(mlr3measures)
```

```r
sprintf("mean absolute error after standardization: %f", mae(g.fit.tests$Salary,
    g.test.pre))
```

```
## [1] "mean absolute error after standardization: 0.640305"
```

```r
test_pre_o <- as.vector(g.test.pre[, 1] * scal$std[26] + scal$mean[26])
sprintf("mean absolute error before standardization: %f", mae(g.fit.test$Salary,
    test_pre_o))
```

```
## [1] "mean absolute error before standardization: 6993725.052379"
```

```r
# ......neural network ...neural network for POS 'c' center and scale the data
set.seed(12345)
data.c <- main.pos.c.filtered[, 5:30]

index <- sample(1:nrow(data.c), round(0.75 * nrow(data.c)))
train <- data.c[index, ]
test <- data.c[-index, ]

maxs <- apply(data.c, 2, max)
mins <- apply(data.c, 2, min)
scaled <- as.data.frame(scale(data.c, center = mins, scale = maxs - mins))
train_ <- scaled[index, ]
test_ <- scaled[-index, ]

# build the 2 layer 3 node neuralnetwork model
library(neuralnet)
```

```
##
## Attaching package: 'neuralnet'

## The following object is masked from 'package:dplyr':
##
##     compute
```

```r
n <- names(train_)
f <- as.formula(paste("Salary ~", paste(n[!n %in% "Salary"], collapse = " + ")))
nn <- neuralnet(f, data = train_, hidden = c(2, 3), linear.output = T)
plot(nn)

# use compute function to get the MSE on the test set
pr.nn <- neuralnet::compute(nn, test_[, 1:25])
pr.nn_ <- pr.nn$net.result * (max(data.c$Salary) - min(data.c$Salary)) + min(data.c$Salary)
test.r <- (test_$Salary) * (max(data.c$Salary) - min(data.c$Salary)) + min(data.c$Salary)
MAD <- abs(test.r - pr.nn_)/nrow(test_)
MAD
```

```
##            [,1]
## 5    90488.8761
## 15  121623.6969
## 18  144267.8672
## 20     647.8486
## 21 1446241.2916
## 30   48340.9421
## 31  794730.2559
## 34  101318.2737
## 36 1522409.8405
```

```
## 38  922673.0201
```

```
# build the 5 layer 5 node neuralnetwork model
nn.c <- neuralnet(f, data = train_, hidden = c(5, 5), linear.output = T)
plot(nn.c)

# use compute function to get the MSE on the test set
set.seed(12345)
pr.nn <- neuralnet::compute(nn.c, test_[, 1:25])
pr.nn_ <- pr.nn$net.result * (max(data.c$Salary) - min(data.c$Salary)) + min(data.c$Salary)
test.r <- (test_$Salary) * (max(data.c$Salary) - min(data.c$Salary)) + min(data.c$Salary)
MAD <- abs(test.r - pr.nn_)/nrow(test_)
MAD
```

```
##          [,1]
## 5   284788.37
## 15   91505.27
## 18  186611.77
## 20   30855.13
## 21  886767.67
## 30   32644.07
## 31  726573.13
## 34   20097.50
## 36 1647445.86
## 38  653554.64
```

```
# ...neural network for POS 'f' center and scale the data
set.seed(12345)
data.f <- main.pos.f.filtered[, 5:30]

index <- sample(1:nrow(data.f), round(0.75 * nrow(data.f)))
train <- data.f[index, ]
test <- data.f[-index, ]

maxs <- apply(data.f, 2, max)
mins <- apply(data.f, 2, min)
scaled <- as.data.frame(scale(data.f, center = mins, scale = maxs - mins))
train_ <- scaled[index, ]
test_ <- scaled[-index, ]

# build the 2 layer 3 node neuralnetwork model
library(neuralnet)
n <- names(train_)
f <- as.formula(paste("Salary ~", paste(n[!n %in% "Salary"], collapse = " + ")))
nn.f <- neuralnet(f, data = train_, hidden = c(2, 3), linear.output = T)
plot(nn.f)

# use compute function to get the MSE on the test set
pr.nn <- neuralnet::compute(nn.f, test_[, 1:25])
pr.nn_ <- pr.nn$net.result * (max(data.f$Salary) - min(data.f$Salary)) + min(data.f$Salary)
test.r <- (test_$Salary) * (max(data.f$Salary) - min(data.f$Salary)) + min(data.f$Salary)
MAD <- abs(test.r - pr.nn_)/nrow(test_)
MAD
```

```
##           [,1]
## 6   124890.016
```

```
## 8   132538.048
## 18   26178.372
## 19   34191.493
## 21   76687.646
## 28  195480.515
## 29  508319.300
## 33   28662.840
## 41   66336.695
## 47  152138.655
## 53   28894.619
## 54   51706.796
## 55   82693.031
## 59  290077.470
## 61  258928.716
## 68  168116.195
## 70   43920.091
## 77  158680.126
## 78  225951.307
## 79  123363.846
## 83  287557.080
## 85   84849.348
## 95  158420.356
## 99   55680.190
## 104   7781.237
## 108 242744.786
## 109 234187.655
## 110 413877.604
## 112 194821.911
## 115  18284.347
```

```r
# build the 5 layer 5 node neuralnetwork model
nn <- neuralnet(f, data = train_, hidden = c(5, 5), linear.output = T)
plot(nn)

# use compute function to get the MSE on the test set
set.seed(12345)
pr.nn <- neuralnet::compute(nn, test_[, 1:25])
pr.nn_ <- pr.nn$net.result * (max(data.f$Salary) - min(data.f$Salary)) + min(data.f$Salary)
test.r <- (test_$Salary) * (max(data.f$Salary) - min(data.f$Salary)) + min(data.f$Salary)
MAD <- abs(test.r - pr.nn_)/nrow(test_)
MAD
```

```
##             [,1]
## 6   173898.5397
## 8   213642.5933
## 18  102180.3365
## 19   28096.9631
## 21  160684.2948
## 28    2128.7067
## 29  802356.1707
## 33  193605.9141
## 41  131341.4147
## 47  135273.4181
## 53   60212.6400
## 54   24622.0187
```

```
## 55     273.7825
## 59  574565.7428
## 61  417626.1994
## 68  225782.2202
## 70    6731.9304
## 77    4990.3693
## 78  130580.5806
## 79  153013.3359
## 83  572419.6825
## 85  152286.1318
## 95    5580.4943
## 99   20948.5505
## 104 367069.4518
## 108 104700.6570
## 109  21091.0161
## 110 378119.4486
## 112  32074.0535
## 115 119012.0967
```

```
# ...neural network for POS 'g' center and scale the data
set.seed(12345)
data.g <- main.pos.g.filtered[, 5:30]

index <- sample(1:nrow(data.g), round(0.75 * nrow(data.g)))
train <- data.g[index, ]
test <- data.g[-index, ]

maxs <- apply(data.g, 2, max)
mins <- apply(data.g, 2, min)
scaled <- as.data.frame(scale(data.g, center = mins, scale = maxs - mins))
train_ <- scaled[index, ]
test_ <- scaled[-index, ]

# build the 2 layer 3 node neuralnetwork model
library(neuralnet)
n <- names(train_)
f <- as.formula(paste("Salary ~", paste(n[!n %in% "Salary"], collapse = " + ")))
nn.g <- neuralnet(f, data = train_, hidden = c(2, 3), linear.output = T)
plot(nn.g)

# use compute function to get the MSE on the test set
pr.nn <- neuralnet::compute(nn.g, test_[, 1:25])
pr.nn_ <- pr.nn$net.result * (max(data.g$Salary) - min(data.g$Salary)) + min(data.g$Salary)
test.r <- (test_$Salary) * (max(data.g$Salary) - min(data.g$Salary)) + min(data.g$Salary)
MAD <- abs(test.r - pr.nn_)/nrow(test_)
MAD
```

```
##           [,1]
## 6    86739.225
## 8    38138.525
## 19   16029.633
## 21   56552.090
## 24   66199.421
## 27  371875.022
## 29  522758.639
```

```
## 33   21713.339
## 41   33243.694
## 43   25878.591
## 47   39087.350
## 50   55085.249
## 52   89940.533
## 53  467646.400
## 54   49684.155
## 57  138574.871
## 63   94612.940
## 66    5563.645
## 69   14047.110
## 70   38841.772
## 73  131745.343
## 78   24838.724
## 88  132738.441
## 92   78298.455
## 99   61576.371
## 101  52511.576
## 102 267953.179
## 108 284208.253
## 109 183706.742
## 114  65550.139
## 119  81889.646
## 122 218769.229
## 124  36735.219
## 125 129843.162
## 134  57841.835
```

```r
# build the 5 layer 5 node neuralnetwork model
nn <- neuralnet(f, data = train_, hidden = c(5, 5), linear.output = T)
plot(nn)

# use compute function to get the MSE on the test set
set.seed(12345)
pr.nn <- neuralnet::compute(nn, test_[, 1:25])
pr.nn_ <- pr.nn$net.result * (max(data.g$Salary) - min(data.g$Salary)) + min(data.g$Salary)
test.r <- (test_$Salary) * (max(data.g$Salary) - min(data.g$Salary)) + min(data.g$Salary)
MAD <- abs(test.r - pr.nn_)/nrow(test_)
MAD
```

```
##           [,1]
## 6     7782.648
## 8   187820.396
## 19   22753.179
## 21   98914.366
## 24  166886.889
## 27  257257.079
## 29  524474.668
## 33   44428.429
## 41   80272.129
## 43   70588.100
## 47   94168.461
## 50  124533.484
## 52  178801.884
```

```
## 53  490158.525
## 54  116029.502
## 57  294421.580
## 63   94543.494
## 66  335490.177
## 69   44661.791
## 70  120106.595
## 73  204157.685
## 78    9218.574
## 88   13104.565
## 92  134749.745
## 99  133114.487
## 101   7760.159
## 102 164245.853
## 108 206793.653
## 109 188405.868
## 114 317359.512
## 119 134181.273
## 122  24899.665
## 124 181401.935
## 125 163212.392
## 134  33276.857
```

```
# ......Prediction for new data
rookie <- read.csv("~/Desktop/Rworkplace/group-project/rookie.csv")
data2 <- rookie
colnames(data2)[4] <- "main.pos"
data2$main.pos <- as.factor(data2$main.pos)

# ...Classified by player's game role
c <- subset(data2, main.pos == "C")
g <- subset(data2, main.pos == "G")
f <- subset(data2, main.pos == "F")

c <- c[, 5:30]
f <- f[, 5:30]
g <- g[, 5:30]


# center and scale data
set.seed(12345)
maxs <- apply(c, 2, max)
mins <- apply(c, 2, min)
scaled.c <- as.data.frame(scale(c, center = mins, scale = maxs - mins))

maxs <- apply(f, 2, max)
mins <- apply(f, 2, min)
scaled.f <- as.data.frame(scale(f, center = mins, scale = maxs - mins))

maxs <- apply(g, 2, max)
mins <- apply(g, 2, min)
scaled.g <- as.data.frame(scale(g, center = mins, scale = maxs - mins))

# prediction for pos.c
```

```r
set.seed(12345)
pr.nn.c <- neuralnet::compute(nn.c, scaled.c)
pr.nn_ <- pr.nn.c$net.result * (max(c$Salary) - min(c$Salary)) + min(c$Salary)

# compare prediction and observation of c
c <- data.frame(c$Salary, pr.nn_)
c
```

```
##   c.Salary   pr.nn_
## 4 28100000 29795827
## 6 20000000 20519309
## 9 28950000 23576265
```

```r
# prediction for pos.f
set.seed(12345)
pr.nn.f <- neuralnet::compute(nn.f, scaled.f)
pr.nn_ <- pr.nn.f$net.result * (max(f$Salary) - min(f$Salary)) + min(f$Salary)

# compare prediction and observation of f
f <- data.frame(f$Salary, pr.nn_)
f
```

```
##   f.Salary   pr.nn_
## 1 35700000 33715301
## 3 28100000 29021590
## 7 29470000 28880395
```

```r
# prediction for pos.g
set.seed(12345)
pr.nn.g <- neuralnet::compute(nn.g, scaled.g)
pr.nn_ <- pr.nn.g$net.result * (max(g$Salary) - min(g$Salary)) + min(g$Salary)

# compare prediction and observation of g
g <- data.frame(g$Salary, pr.nn_)
g
```

```
##   g.Salary   pr.nn_
## 2 29750000 29616881
## 5 28100000 28203180
## 8 29750000 28225130
```