STA302 Final Project ~ Predicting the systolic blood pressure using multiple linear model based on NHANES study from 2011-2012

Ruike Xu 1006562550

10/06/2021

Introduction

The prevalence of high blood pressure has become a major threat to people's lives all over the world. It's essential for the government to keep track of the health status of people and give proper suggestions to adjust for a better lifestyle.

NHANES is a study that was designed to assess the health and nutritional status of adults and children in the United States since 1960. The survey includes interviews and physical examination components and examines a nationally representative sample of approximately 5000 people each year. (National Health and Nutrition Examination Survey, 2020) We have specifically selected 17 variables from the original data set. Furthermore, only observations that are aged greater than 17 years old are selected to get a full representation of the data. (some of the measurements have a minimum age requirement)

Select the chosen columns/variables from the NHANES data set 2009-2012 with adjusted weighting and the observations are all aged greater than 17.

```
## If the package is not already installed then use ##
# install.packages('NHANES')
# install.packages('tidyverse')
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.3     v purrr   0.3.4
## v tibble  3.1.2     v dplyr   1.0.6
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(NHANES)
small.nhanes <- na.omit(NHANES[NHANES$SurveyYr=="2011_12"
& NHANES$Age > 17,c(1,3,4,8:11,13,17,20,21,25,46,50,51,52,61)])
small.nhanes <- as.data.frame(small.nhanes %>%
group_by(ID) %>% filter(row_number()==1) )
nrow(small.nhanes)
```

```
## [1] 743
```

```
## Checking whether there are any ID that was repeated. If not ##
## then length(unique(small.nhanes$ID)) and nrow(small.nhanes) are same ##
length(unique(small.nhanes$ID))
```

```
## [1] 743
```

Traning data and testing data from the data set

```
## Create training and test set ##
set.seed(1006562550)
train <- small.nhanes[sample(seq_len(nrow(small.nhanes)), size = 500),]
nrow(train)
```

```
## [1] 500
```

```
length(which(small.nhanes$ID %in% train$ID))
```

```
## [1] 500
```

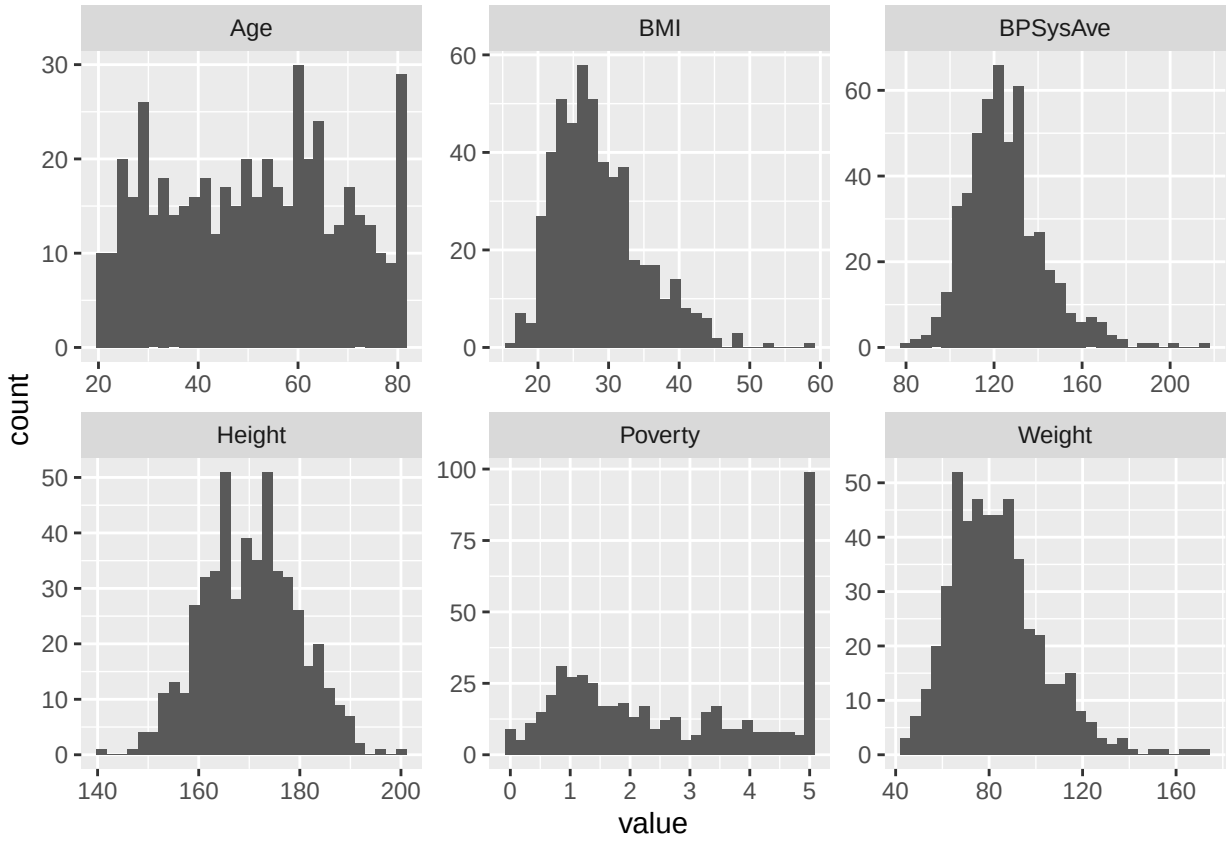
```
test <- small.nhanes[!small.nhanes$ID %in% train$ID,]
nrow(test)
```

```
## [1] 243
```

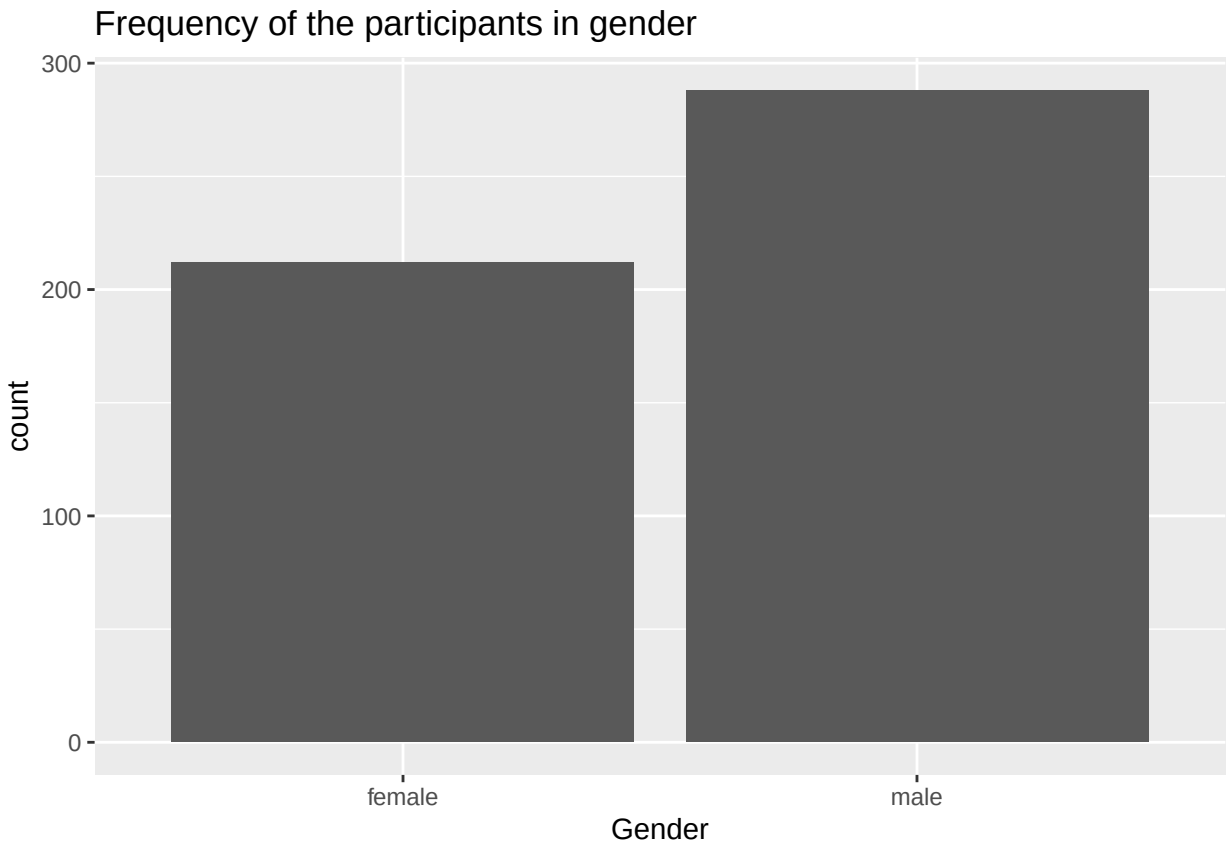
Plot histograms of all possible predictors and response variable

```
# install.packages('purrr')
# install.packages('tidyr')
# install.packages('ggplot2')
library(purrr)
library(tidyr)
library(ggplot2)
library(tidyverse)

train %>% select(-c(ID, SleepHrsNight)) %>% keep(is.numeric) %>%
  gather() %>% ggplot(aes(value)) +
  facet_wrap(~ key, scales = "free") +
  geom_histogram()
```

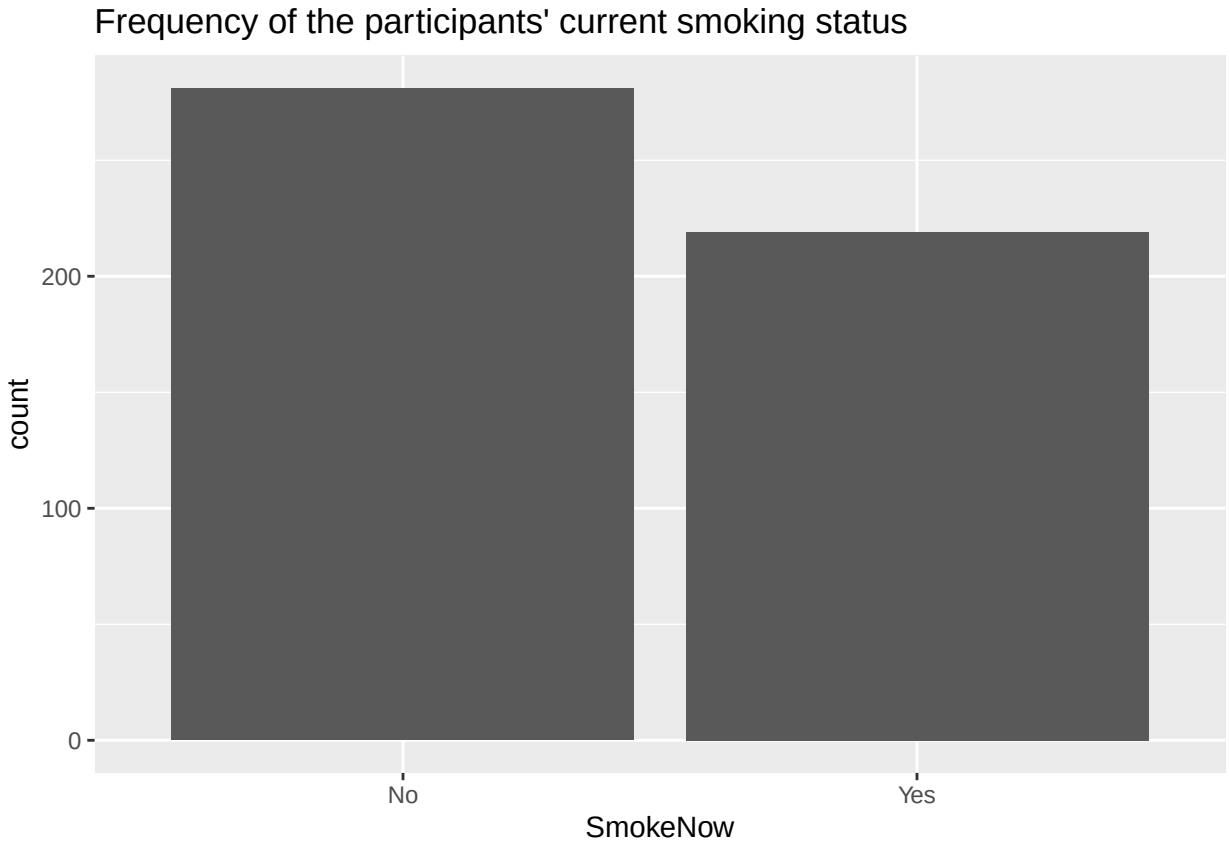
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



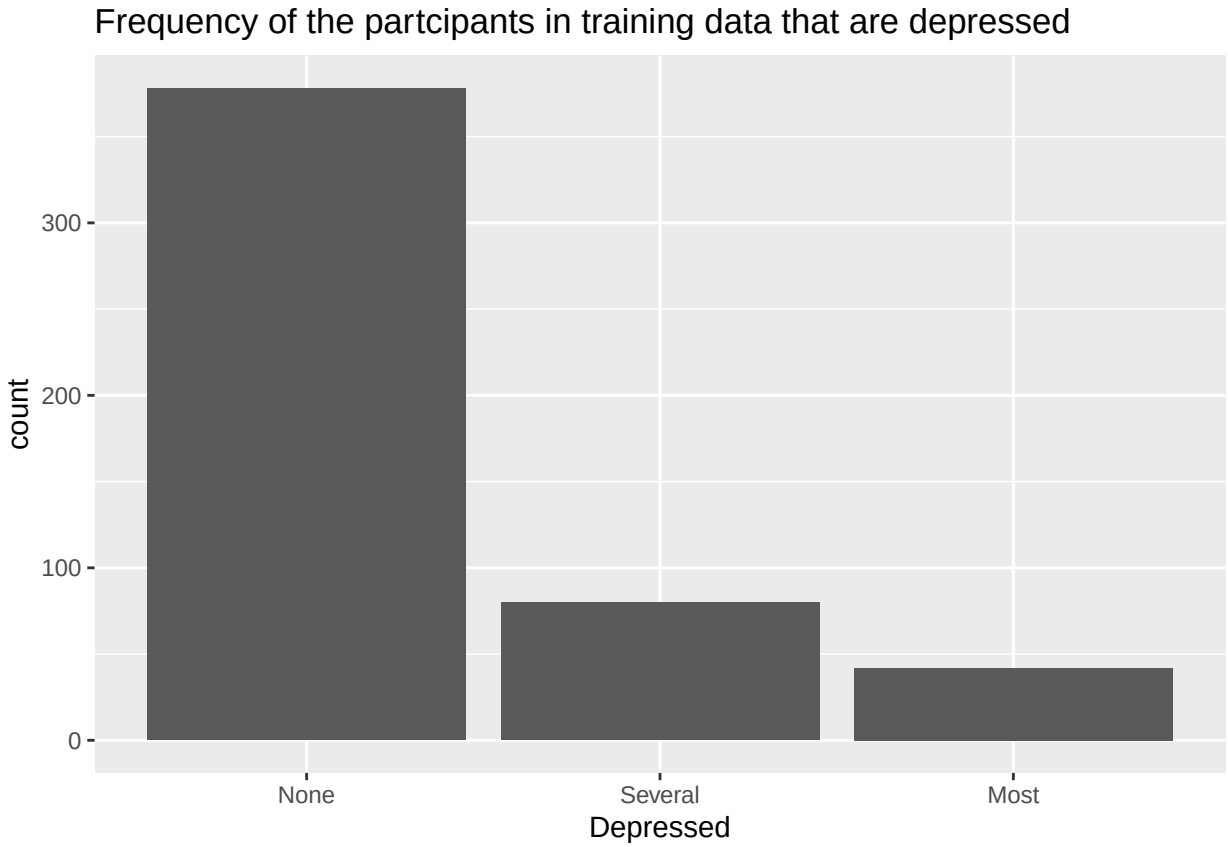

```
ggplot(train, aes(x = as.factor(Gender))) +
  geom_bar() + labs(x="Gender") + ggtitle("Frequency of the participants in gender")
```



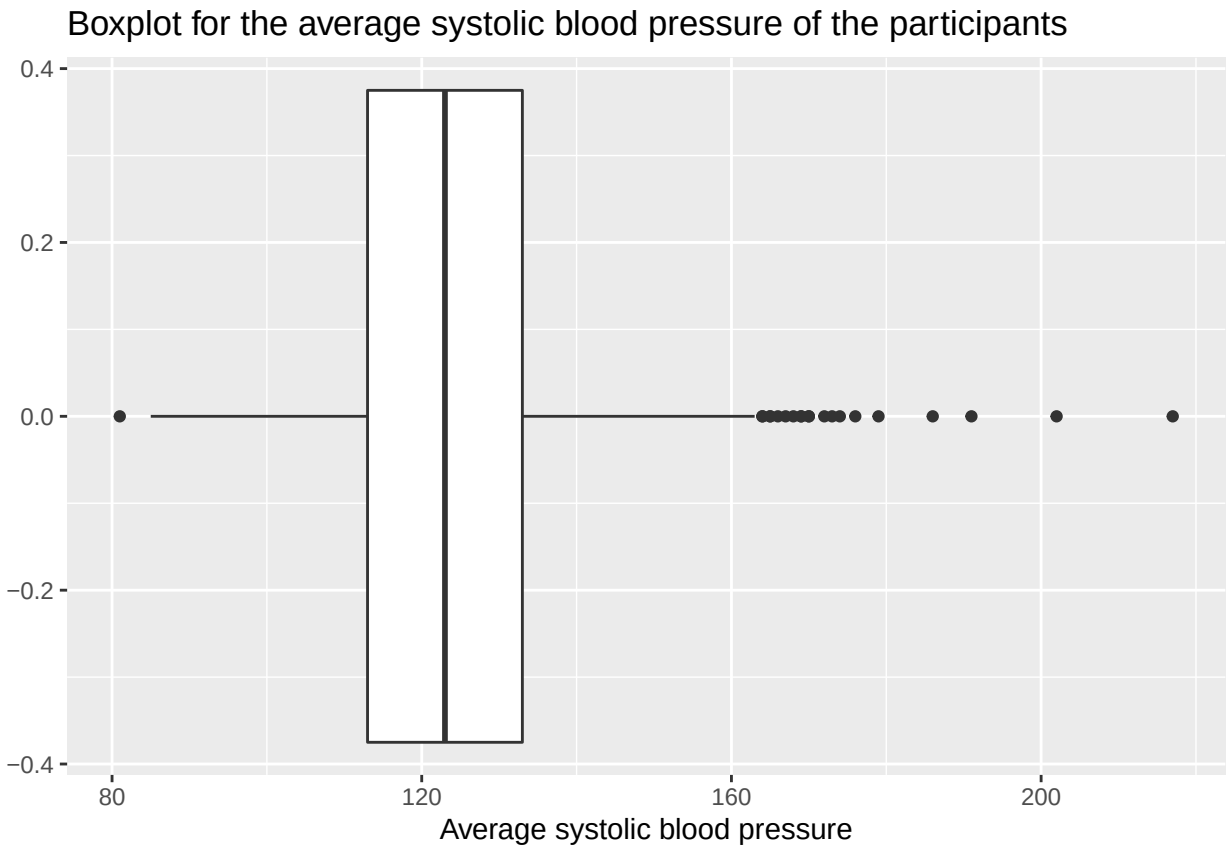

```
ggplot(train, aes(x = as.factor(SmokeNow))) +
  geom_bar() + labs(x="SmokeNow") + ggtitle("Frequency of the participants' current smoking status")
```




```
ggplot(train, aes(x = as.factor(Depressed))) +
  geom_bar() + labs(x="Depressed") + ggtitle("Frequency of the partcipants in training data that are dep
```




```
ggplot(train, aes(x = BPSysAve)) +
  geom_boxplot() + labs(x="Average systolic blood pressure") + ggtitle("Boxplot for the average systoli
```



Methodology

Full model with all potential predictors

```
# install.packages("car")
library(UsingR)
```

```
## Loading required package: MASS
```

```
##
## Attaching package: 'MASS'
```

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

```
## Loading required package: HistData
```

```
## Loading required package: Hmisc
```

```
## Loading required package: lattice
```

```
## Loading required package: survival

## Loading required package: Formula

##
## Attaching package: 'Hmisc'

## The following objects are masked from 'package:dplyr':
##
##     src, summarize

## The following objects are masked from 'package:base':
##
##     format.pval, units

##
## Attaching package: 'UsingR'

## The following object is masked from 'package:survival':
##
##     cancer
```

```
library(scatterplot3d)
library(xtable)
```

```
##
## Attaching package: 'xtable'

## The following objects are masked from 'package:Hmisc':
##
##     label, label<-
```

```
library(car)
```

```
## Loading required package: carData

##
## Attaching package: 'car'

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

## The following object is masked from 'package:purrr':
##
##     some
```

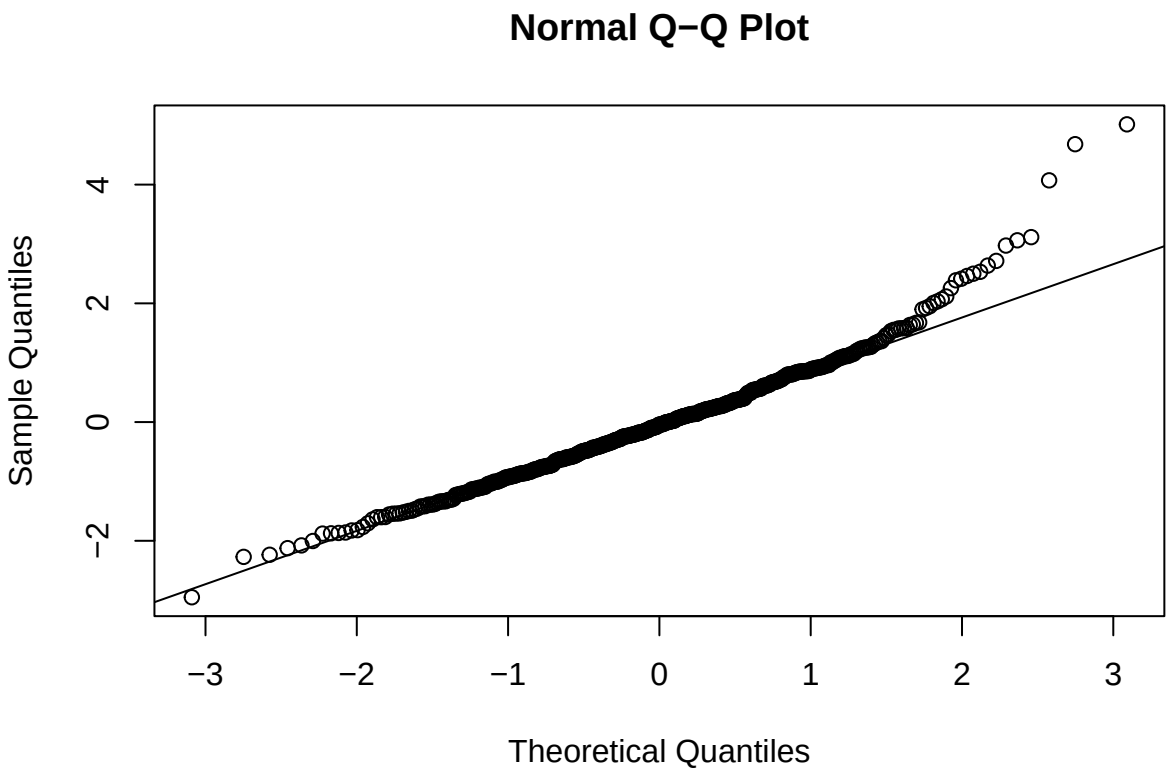
```r
# Fitting a full model
model_full <- lm(BPSysAve ~ ., data = train[, -c(1)])
vif(model_full)
```

```
##                     GVIF Df GVIF^(1/(2*Df))
## Gender          2.157190  1        1.468737
## Age             1.935847  1        1.391347
## Race3           1.947170  5        1.068908
## Education       2.012027  4        1.091325
## MaritalStatus   2.328322  5        1.088189
## HHIncome        8.717879 11        1.103433
## Poverty         4.876774  1        2.208342
## Weight        102.691700  1       10.133691
## Height         23.931611  1        4.891995
## BMI            87.254025  1        9.340986
## Depressed       1.369429  2        1.081770
## SleepHrsNight   1.181730  1        1.087074
## SleepTrouble    1.288569  1        1.135151
## PhysActive      1.289435  1        1.135533
## SmokeNow        1.324688  1        1.150951
```

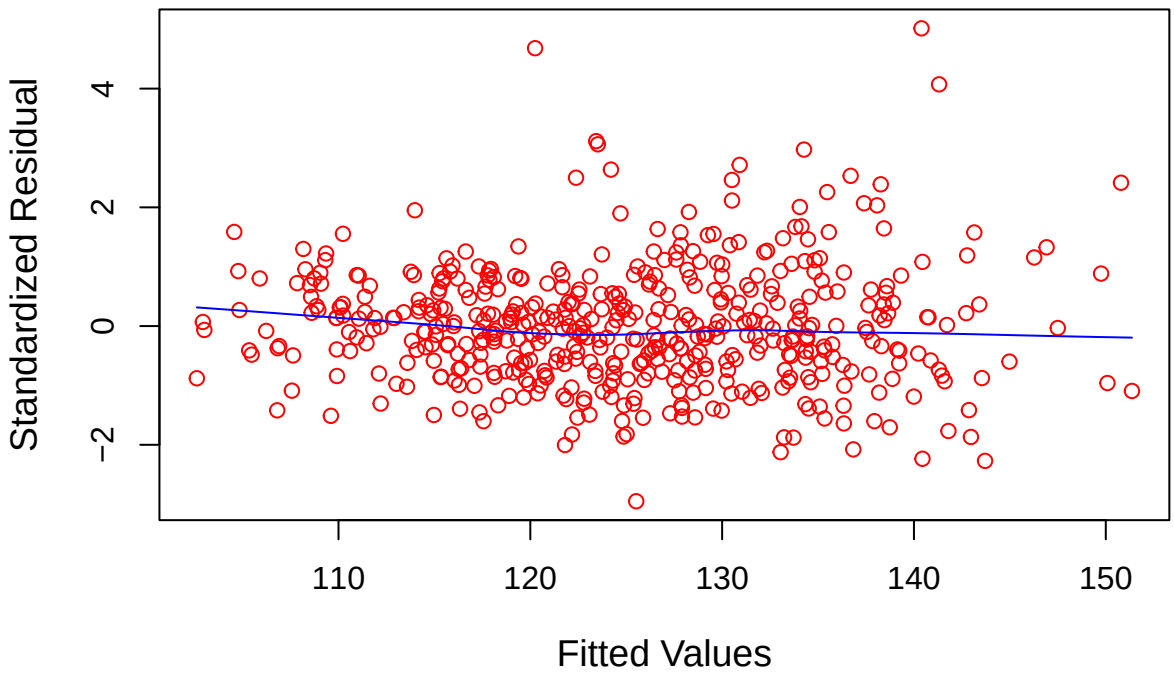
```r
anova(model_full)
```

```
## Analysis of Variance Table
##
## Response: BPSysAve
##                Df Sum Sq Mean Sq  F value    Pr(>F)
## Gender          1   1823  1823.2   7.1419  0.007797 **
## Age             1  29892 29891.6 117.0936 < 2.2e-16 ***
## Race3           5   1806   361.2   1.4150  0.217371
## Education       4   2127   531.8   2.0833  0.081949 .
## MaritalStatus   5   2679   535.9   2.0992  0.064359 .
## HHIncome       11   3994   363.1   1.4223  0.159259
## Poverty         1   1897  1896.9   7.4306  0.006656 **
## Weight          1     11    10.8   0.0424  0.836871
## Height          1    889   889.1   3.4828  0.062642 .
## BMI             1    835   835.2   3.2718  0.071131 .
## Depressed       2     68    33.8   0.1323  0.876080
## SleepHrsNight   1     91    91.3   0.3578  0.550013
## SleepTrouble    1    101   101.4   0.3974  0.528755
## PhysActive      1     12    11.9   0.0467  0.828961
## SmokeNow        1     47    47.4   0.1855  0.666855
## Residuals     462 117939   255.3
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

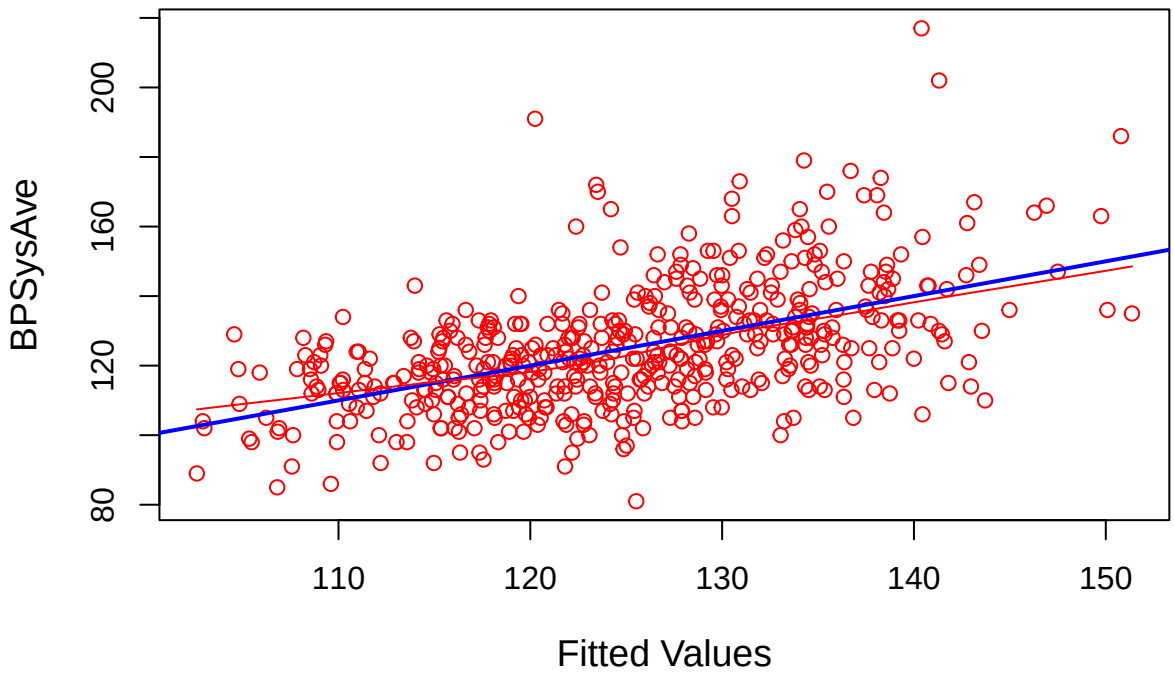
```r
capture.output(anova(model_full),file="an1.pdf")

# Residual plots
resid_full <- rstudent(model_full)
fitted_full <- predict(model_full)
qqnorm(resid_full)
qqline(resid_full)
```



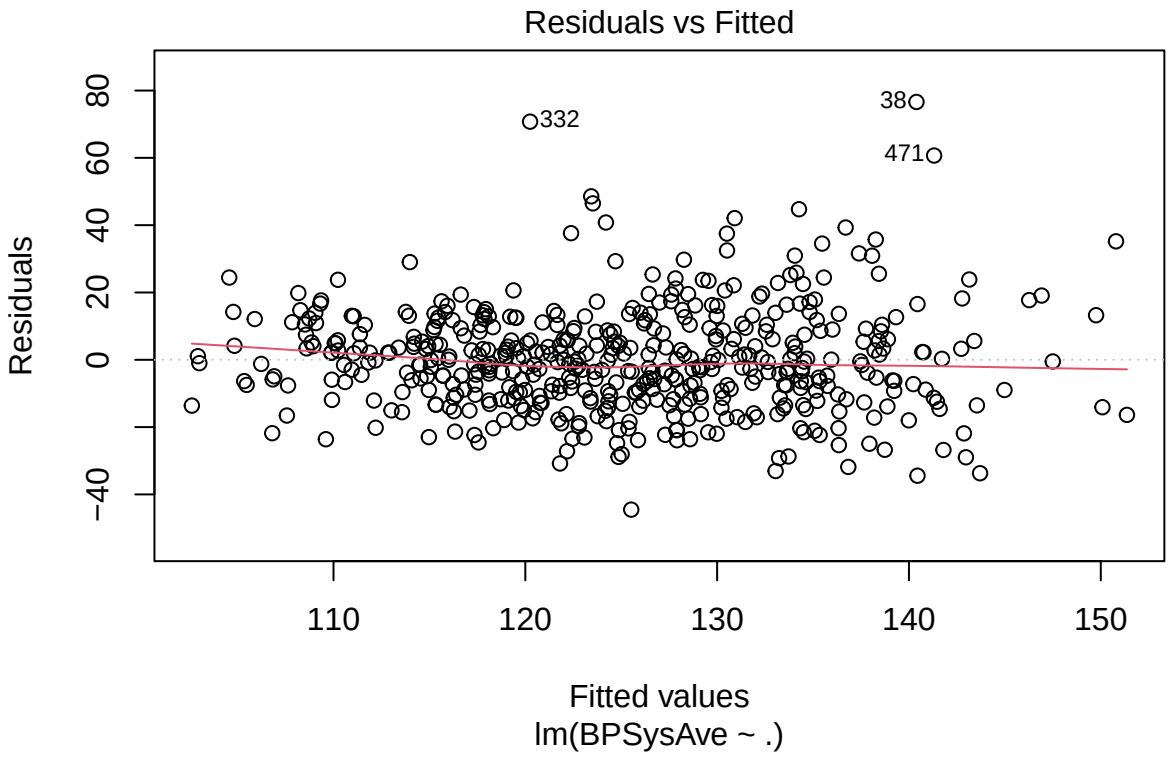

```
plot(resid_full ~ fitted_full, type = "p", xlab = "Fitted Values",
     ylab = "Standardized Residual", cex.lab = 1.2,
     col = "red")
lines(lowess(fitted_full, resid_full), col = "blue")
```

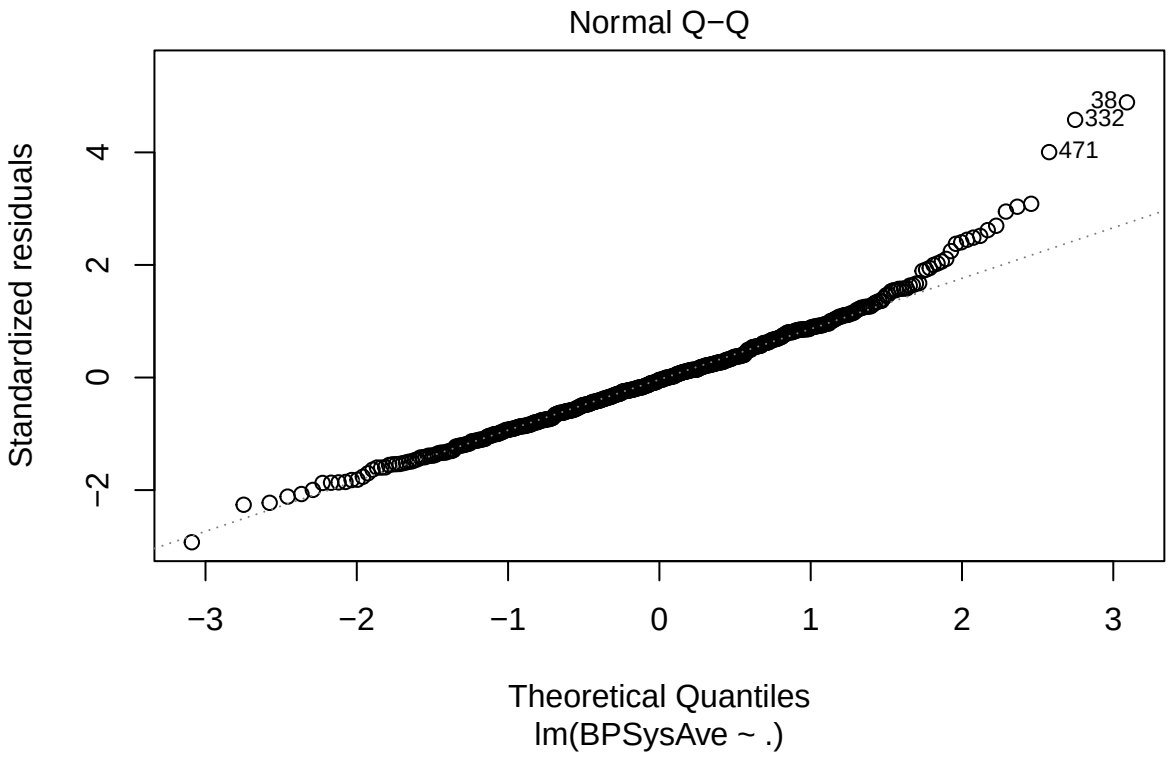


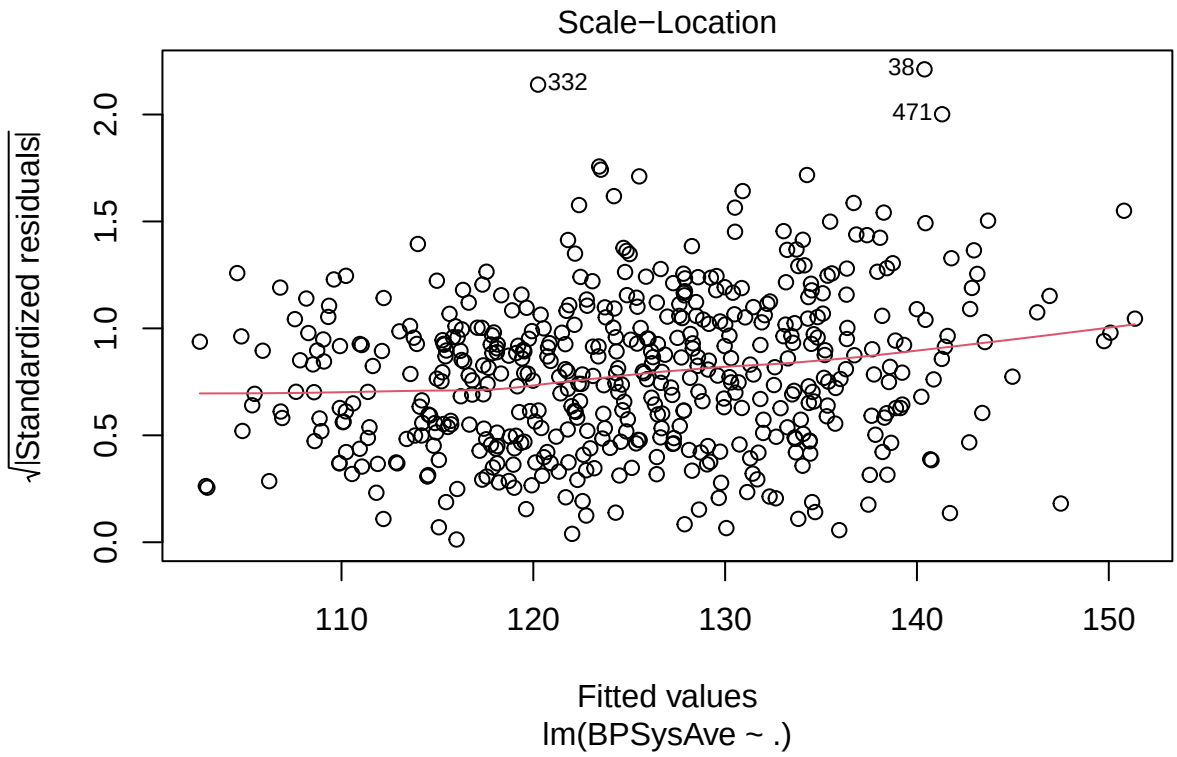

```
# Response vs Fitted values ##
plot(train$BPSysAve ~ fitted_full, type = "p", xlab = "Fitted Values",
     ylab = "BPSysAve", cex.lab = 1.2,
     col = "red")
abline(lm(train$BPSysAve ~ fitted_full), lwd = 2, col = "blue")
lines(lowess(fitted_full, train$BPSysAve), col = "red")
```

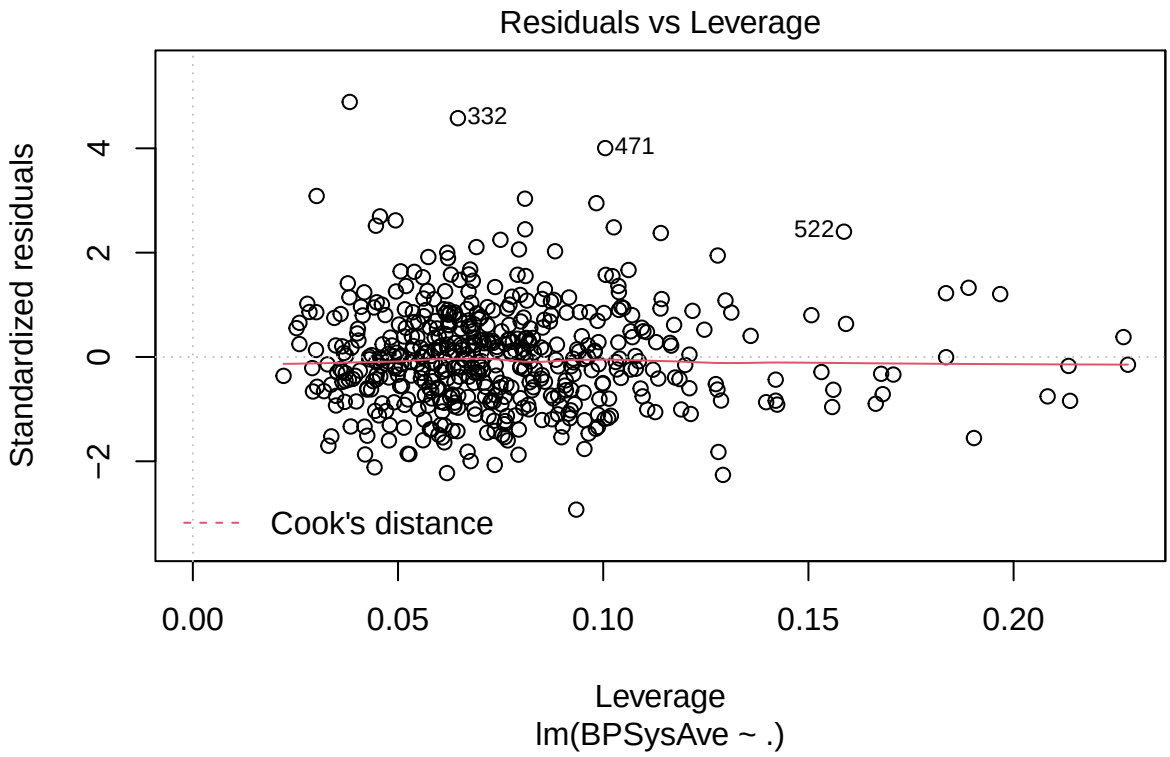



```r
# Prediction
pred_full <- predict(model_full, newdata = test[, -c(1)], type = "response")
# prediction error
pred_error_full <- mean((test$BPSysAve - pred_full)^2)

# Model selection criteria
criteria <- function(model){
    n <- length(model$residuals)
    p <- length(model$coefficients) - 1
    RSS <- sum(model$residuals^2)
    R2 <- summary(model)$r.squared
    R2.adj <- summary(model)$adj.r.squared
    AIC <- n*log(RSS/n) + 2*p
    AICc <- AIC + (2*(p+2)*(p+3))/(n-p-1)
    BIC <- n*log(RSS/n) + (p+2)*log(n)
    res <- c(R2, R2.adj, AIC, AICc, BIC)
    names(res) <- c("R Squared", "Adjusted R Squared", "AIC", "AICc", "BIC")
    return(res)
}

plot(model_full)
```


Residuals vs Fitted
Residuals
Fitted values
lm(BPSysAve ~ .)
332
38
471
−40
0
20
40
60
80
110
120
130
140
150


Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(BPSysAve ~ .)
38
332
471
−3
−2
−1
0
1
2
3
−2
0
2
4


Scale−Location
332
38
471
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
110
120
130
140
150
Fitted values
lm(BPSysAve ~ .)




```
# Criteria for full model
crit1 <- criteria(model = model_full)

# Diagnostics check in Cook's distance, DFFITS, DFBETAS
n_train = 500
p_full = 37

D_full <- cooks.distance(model_full)
which(D_full > qf(0.5, p_full+1, n_train-p_full-1))
```

```
## named integer(0)
```

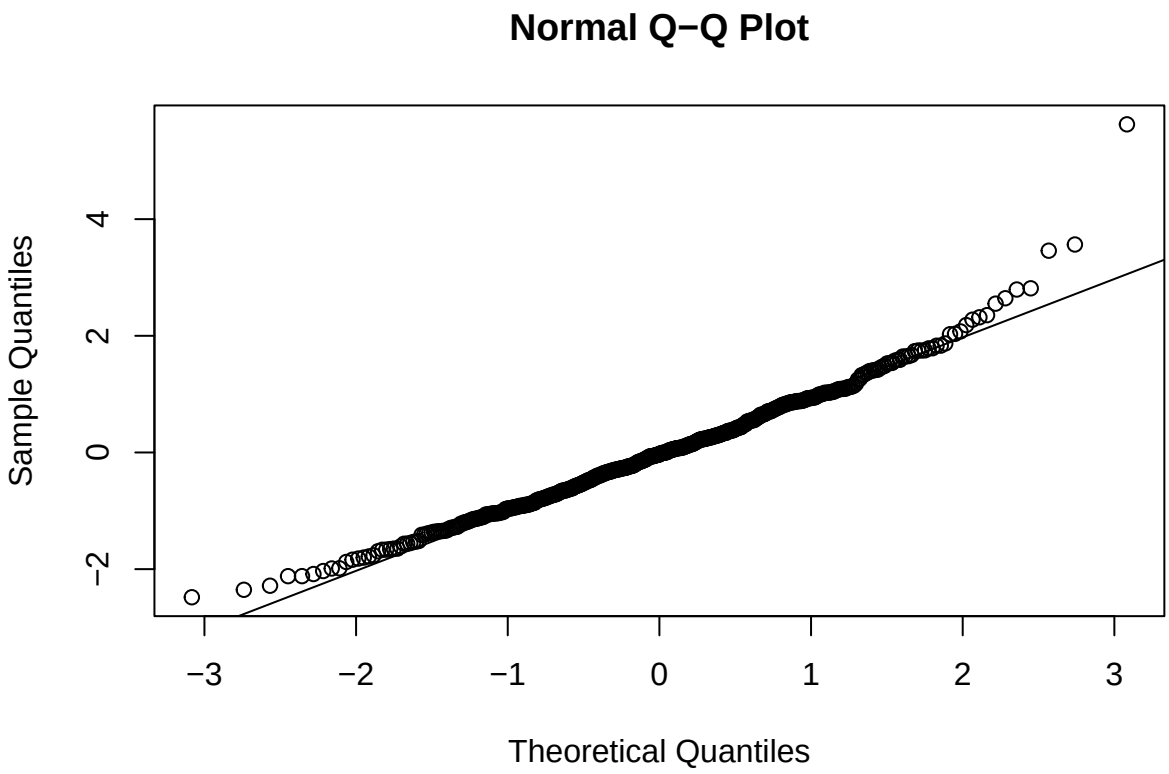
```
dfits_full <- dffits(model_full)
dfits_full_ben <- which(abs(dfits_full) > 2*sqrt((p_full+1)/n_train))

dfb_full <- dfbetas(model_full)
dfb_full_ben <- which(abs(dfb_full[,1]) > 2/sqrt(n_train))

# Remove potential outliers
full_outliers <- intersect(dfits_full_ben, dfb_full_ben)
train_modified <- train[-c(full_outliers),]

# Fit a new multiple linear model with modified training data
model_full_ad <- lm(BPSysAve ~ ., data = train_modified[, -c(1)])
vif(model_full_ad)
```

```
##                     GVIF Df GVIF^(1/(2*Df))
## Gender          2.151830  1        1.466912
## Age             1.947050  1        1.395367
## Race3           1.944817  5        1.068779
## Education       1.983393  4        1.089372
## MaritalStatus   2.314268  5        1.087530
## HHIncome        9.098763 11        1.105580
## Poverty         5.016542  1        2.239764
## Weight        103.248339  1       10.161119
## Height         24.255040  1        4.924941
## BMI            88.632196  1        9.414467
## Depressed       1.354570  2        1.078823
## SleepHrsNight   1.192866  1        1.092184
## SleepTrouble    1.309241  1        1.144221
## PhysActive      1.273236  1        1.128378
## SmokeNow        1.318577  1        1.148293
```

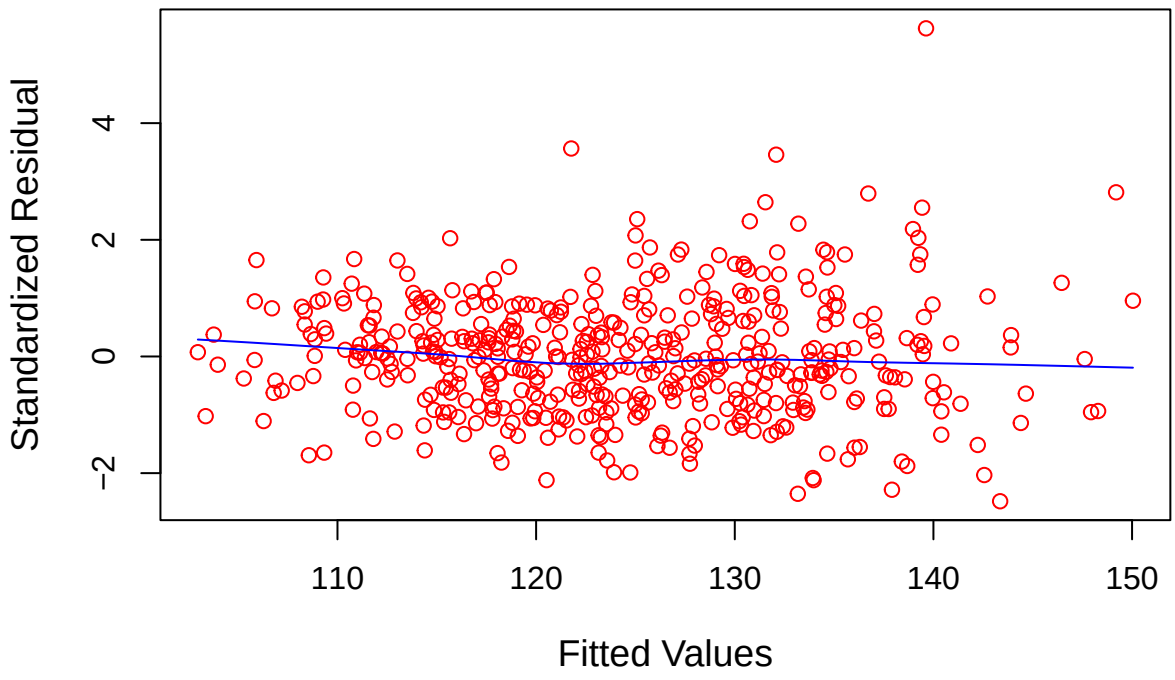
```
anova(model_full_ad)
```

```
## Analysis of Variance Table
##
## Response: BPSysAve
##                Df Sum Sq Mean Sq  F value  Pr(>F)
## Gender          1   1688  1688.2   8.0331 0.00480 **
## Age             1  29044 29043.7 138.1995 < 2e-16 ***
## Race3           5   2200   440.0   2.0936 0.06510 .
## Education       4   2782   695.4   3.3088 0.01092 *
## MaritalStatus   5   2533   506.6   2.4106 0.03570 *
## HHIncome       11   2688   244.4   1.1627 0.31087
## Poverty         1    934   934.3   4.4459 0.03554 *
## Weight          1     11    10.6   0.0506 0.82216
## Height          1    504   504.4   2.4000 0.12203
## BMI             1    153   153.1   0.7284 0.39385
## Depressed       2    123    61.5   0.2927 0.74641
## SleepHrsNight   1    278   278.0   1.3230 0.25067
## SleepTrouble    1    138   138.5   0.6590 0.41735
## PhysActive      1      0     0.2   0.0008 0.97735
## SmokeNow        1     51    51.0   0.2427 0.62248
## Residuals     450  94571   210.2
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

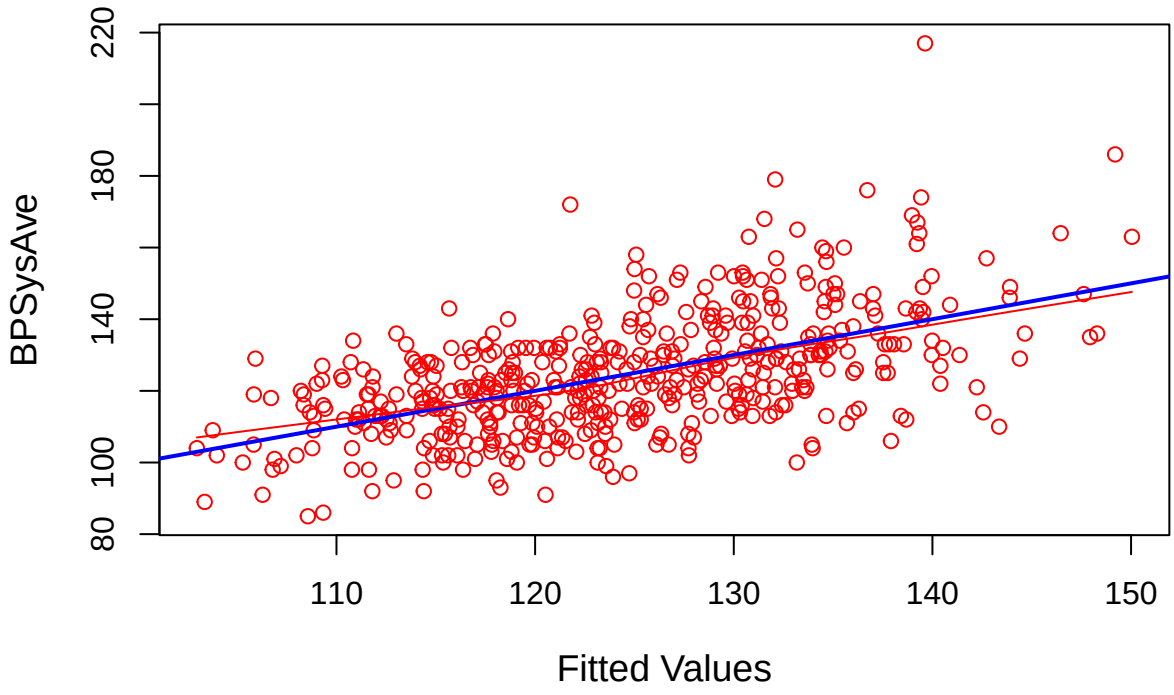
```
capture.output(anova(model_full_ad),file="an2.png")

# Residual plots
resid_full_ad <- rstudent(model_full_ad)
fitted_full_ad <- predict(model_full_ad)
qqnorm(resid_full_ad)
qqline(resid_full_ad)
```




```
plot(resid_full_ad ~ fitted_full_ad, type = "p", xlab = "Fitted Values",
     ylab = "Standardized Residual", cex.lab = 1.2,
     col = "red")
lines(lowess(fitted_full_ad, resid_full_ad), col = "blue")
```




```
# Response vs Fitted values ##
plot(train_modified$BPSysAve ~ fitted_full_ad, type = "p", xlab = "Fitted Values",
     ylab = "BPSysAve", cex.lab = 1.2,
     col = "red")
abline(lm(train_modified$BPSysAve ~ fitted_full_ad), lwd = 2, col = "blue")
lines(lowess(fitted_full_ad, train_modified$BPSysAve), col = "red")
```




```
# Prediction
pred_full_ad <- predict(model_full_ad, newdata = test[, -c(1)], type = "response")
# prediction error
pred_error_full_ad <- mean((test$BPSysAve - pred_full_ad)^2)

# Criteria for adjusted full model
crit2 <- criteria(model = model_full_ad)

c(pred_error_full, pred_error_full_ad)
```

```
## [1] 246.2328 244.0803
```

```
crit1
```

```
##          R Squared Adjusted R Squared                AIC               AICc
##          0.2817897          0.2242707       2805.6579850       2812.4112317
##                BIC
##       2974.0277008
```

```
crit2
```

```
##          R Squared Adjusted R Squared                AIC               AICc
##          0.3132020          0.2567319       2644.1942823       2651.1276156
##                BIC
##       2811.6165831
```

Ridge regression (not able to use for variable selection)

The ridge penalty shrinks the regression coefficient estimate toward zero, but not exactly zero, so I would prefer to employ Stepwise variable selection method and LASSO variable selection

Variable selection

Stepwise Variable selection (backward direction)

In the backward Stepwise variable selection method, all the predictor variables we have chose in the data set are added into the model sequentially, then the predictors that don't have statistical significance in predicting anything on the response variable are removed from the model one by one. The backward method is generally preferred because it avoids suppressor effect that often occurs in forward method. (predictors are only significant when another predictor is held constant)

```
n <- nrow(train)
sel_var_aic_back <- step(model_full, trace = 0, k = 2, direction = "backward")
sel_var_aic_back_mol <- sel_var_aic_back
sel_var_aic_back <- attr(terms(sel_var_aic_back), "term.labels")
sel_var_aic_back
```

Based on AIC

```
## [1] "Gender"  "Age"     "Poverty" "Weight"  "Height"  "BMI"
```

```
n <- nrow(train)
sel_var_bic_back <- step(model_full, trace = 0, k = log(n), direction = "backward")
sel_var_bic_back_mol <- sel_var_bic_back
sel_var_bic_back <- attr(terms(sel_var_bic_back), "term.labels")
sel_var_bic_back
```

Based on BIC

```
## [1] "Gender"  "Age"     "Poverty"
```

LASSO Variable selection

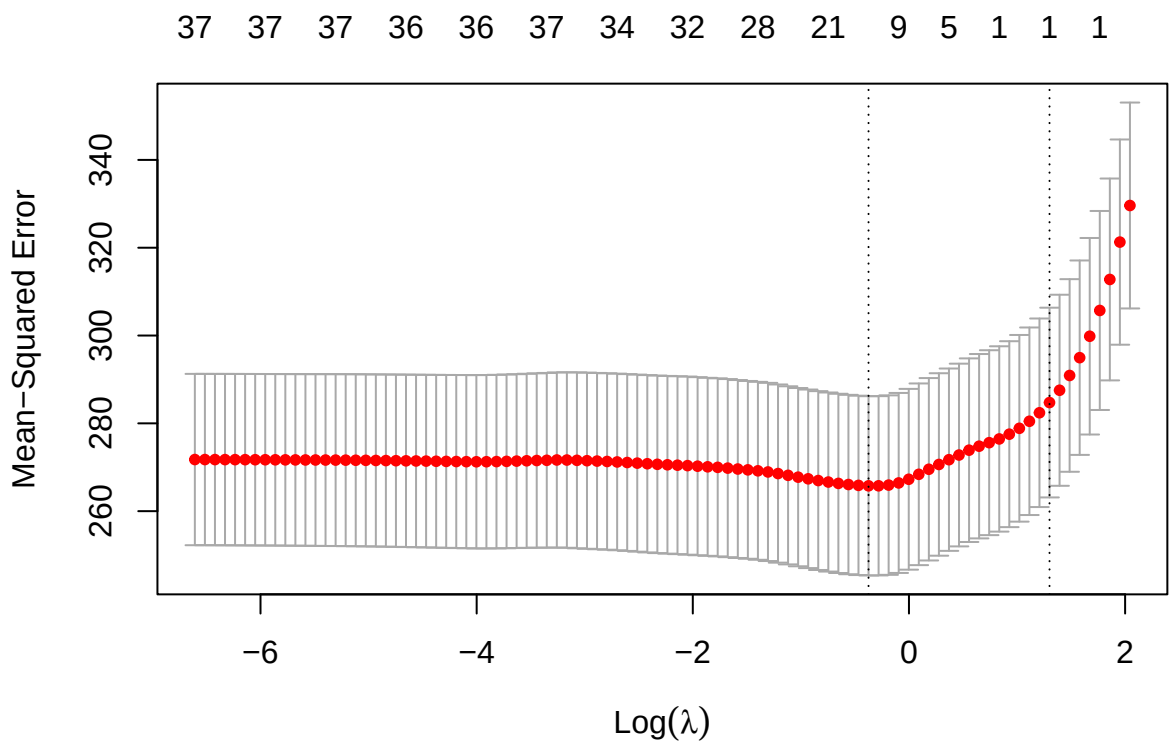
The LASSO variable selection method is a way to automatically select potential predictor variables of the response variable from a large set of candidate predictors in the training data. LASSO penalizes the absolute sum of the regression coefficient, based on tuning parameter λ, so that LASSO can reduce the coefficients of irrelevant variables to zero. We would apply cross validation of the training data to determine the severity of LASSO penalty λ

```
library(glmnet)
```

cross validation to choose lambda

```
## Loading required package: Matrix

##
## Attaching package: 'Matrix'

## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack

## Loaded glmnet 4.1-1
```

```
set.seed(1006562550)
cv.out <- cv.glmnet(x = model.matrix(~., train[-c(1, 12)]), y = train$BPSysAve, standardize = T, alpha =
plot(cv.out)
```




```
best.lambda <- cv.out$lambda.1se
co<-coef(cv.out, s = "lambda.1se")
```

```r
#Selection of the significant features(predictors)

## threshold for variable selection ##

thresh <- 0.00
# select variables #
inds<-which(abs(co) > thresh )
variables<-row.names(co)[inds]
sel.var.lasso<-variables[!(variables %in% '(Intercept)')]
sel.var.lasso
```

```
## [1] "Age"
```

```r
best.lambda
```

```
## [1] 3.67078
```

Variable choosing after backward Stepwise and and LASSO selection procedure

There are three possible sets of variable selections for predicting the person's systolic blood pressure from our data set.After we examine all of them, there is no sets of predictors containing our goal of interests - whether the participant is currently smoking (SmokeNow). Therefore, I would using diagnostics checking techniques and variance inflation factor to see which variables we should select.

```r
# Fitting a model based on backward Stepwise AIC selection and add SmokeNow
vif(sel_var_aic_back_mol)
```

```
##    Gender       Age   Poverty    Weight    Height       BMI
##  1.718817  1.072217  1.053032 97.160513 22.057871 82.294452
```

```r
model_1 <- lm(BPSysAve ~ ., data = train[c(2, 3, 8, 9, 10, 12, 17)])
summary(model_1)
```

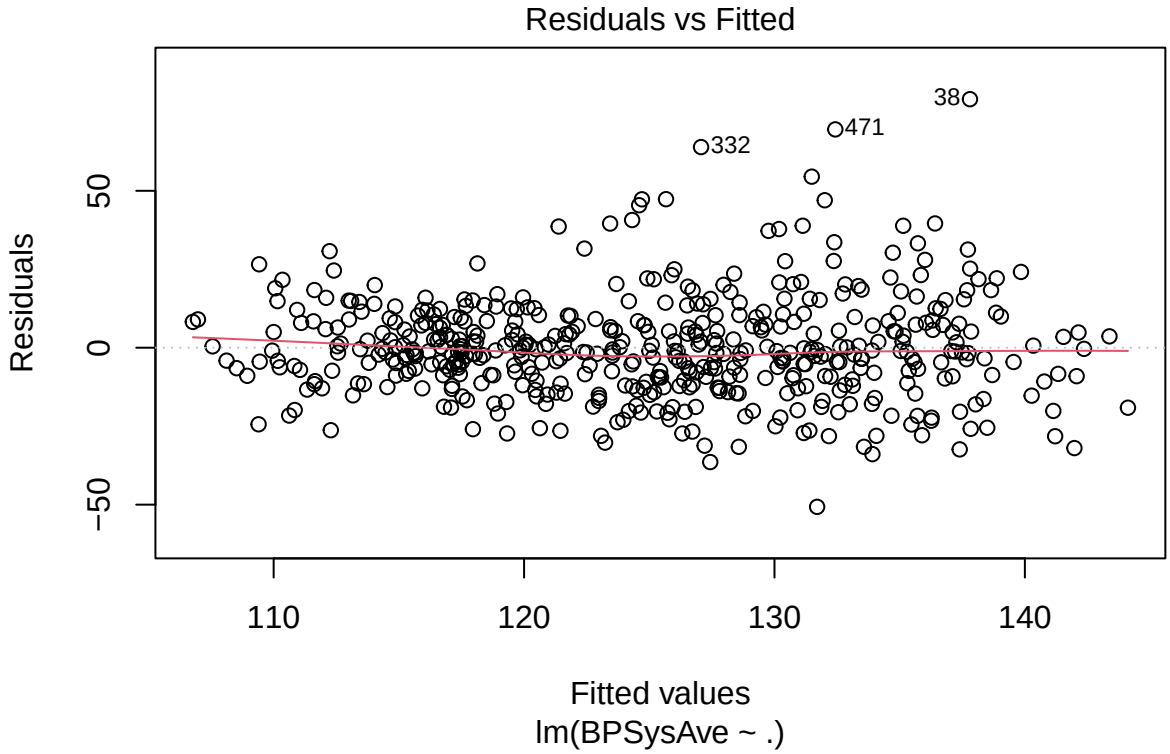
```
##
## Call:
## lm(formula = BPSysAve ~ ., data = train[c(2, 3, 8, 9, 10, 12,
##     17)])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -50.703  -9.422  -1.201   8.259  79.193
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 130.57738   17.63250   7.405  5.7e-13 ***
## Gendermale    5.88180    1.91841   3.066  0.00229 **
## Age           0.44169    0.04509   9.797  < 2e-16 ***
## Poverty      -1.32024    0.45297  -2.915  0.00372 **
## Weight        0.04664    0.03989   1.169  0.24288
## Height       -0.18565    0.11028  -1.683  0.09292 .
## SmokeNowYes  -0.56250    1.56386  -0.360  0.71923
```

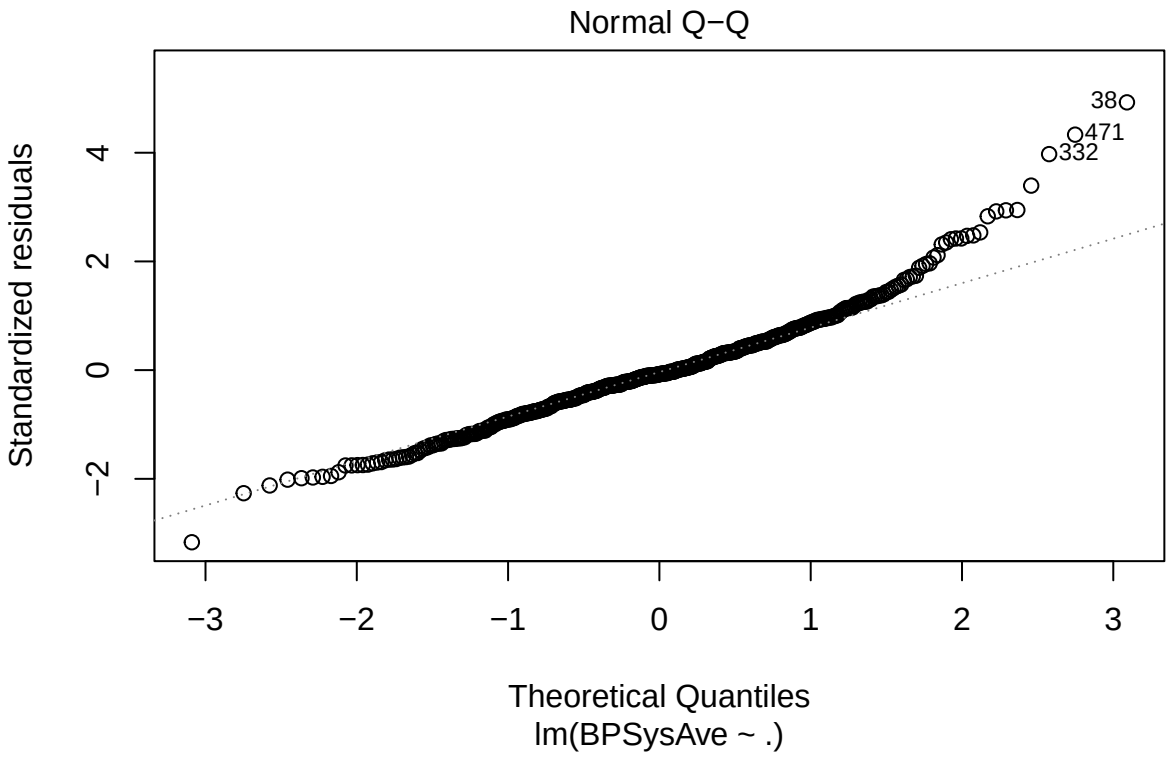
```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 16.17 on 493 degrees of freedom
## Multiple R-squared:  0.2151, Adjusted R-squared:  0.2055
## F-statistic: 22.51 on 6 and 493 DF,  p-value: < 2.2e-16
```

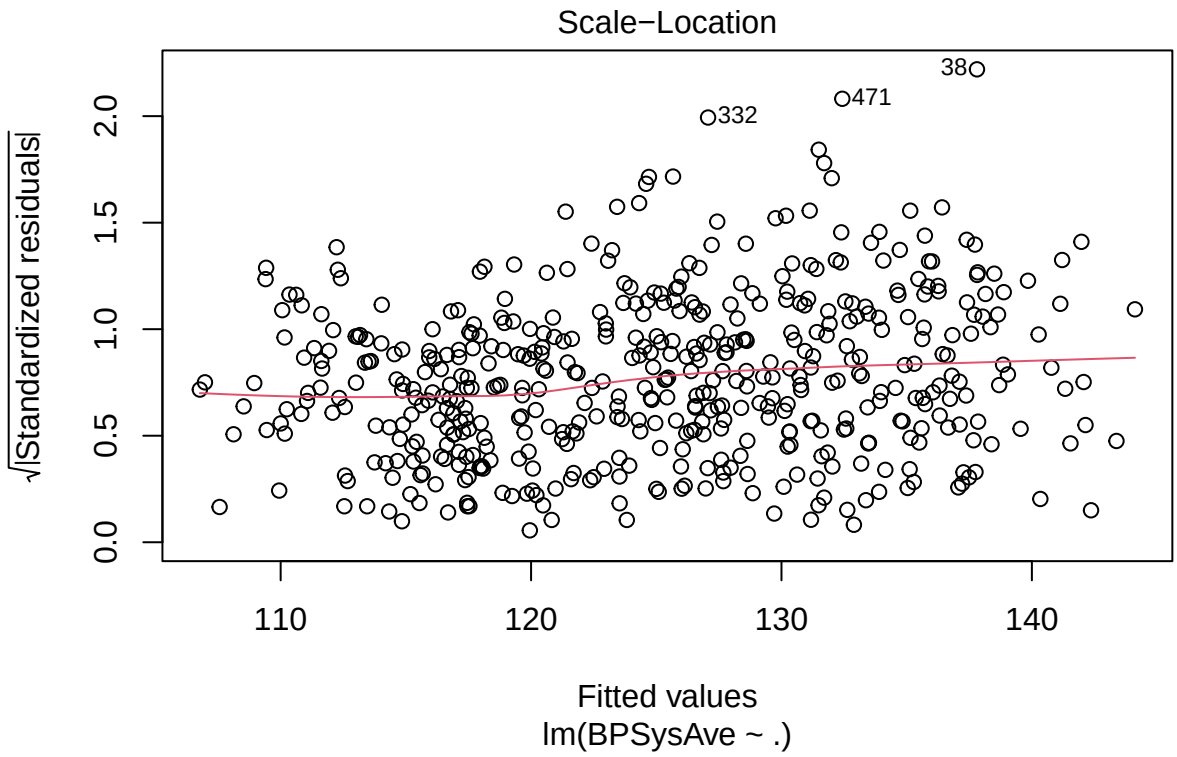
```
vif(model_1)
```

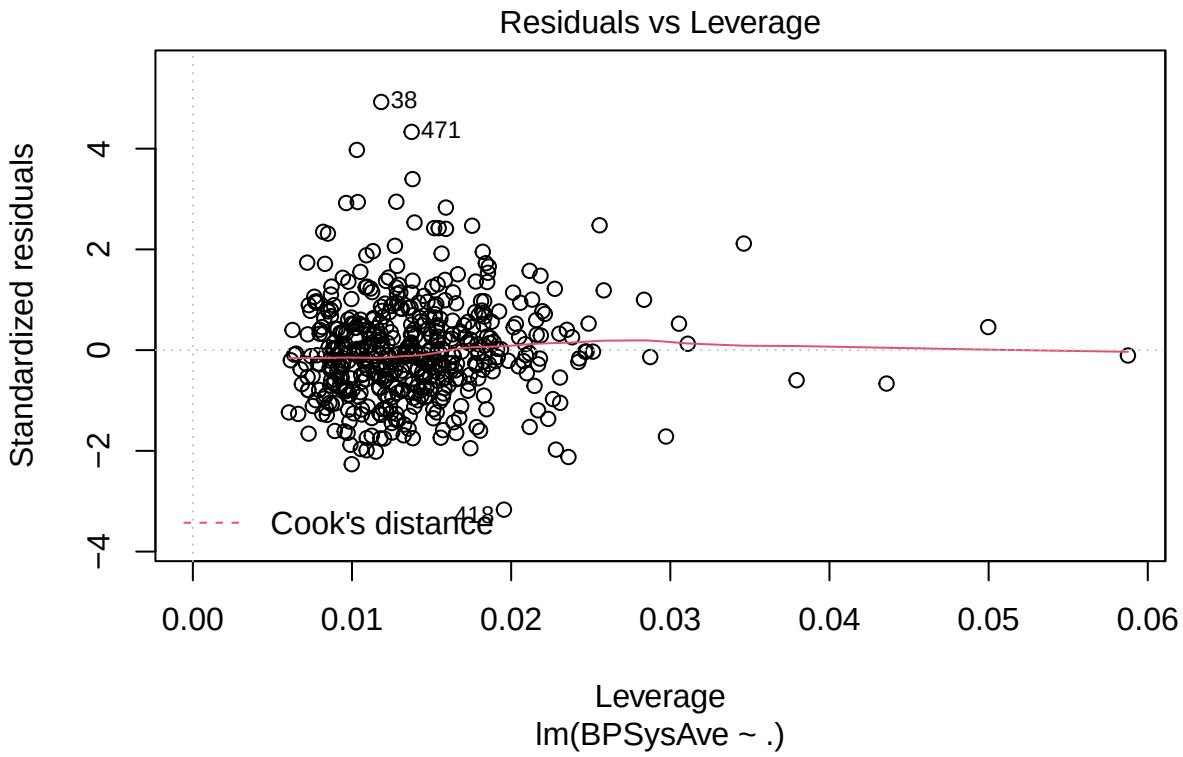
```
##   Gender      Age  Poverty   Weight   Height SmokeNow
## 1.718893 1.172668 1.064002 1.203755 2.041411 1.151284
```

```
plot(model_1)
```




Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(BPSysAve ~ .)
38
471
332
−3
−2
−1
0
1
2
3
−2
0
2
4


Scale−Location
38
471
332
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
110
120
130
140
Fitted values
lm(BPSysAve ~ .)




```
crit_1 <- criteria(model = model_1)

# Diagnostics check in Cook's distance, DFFITS, DFBETAS
n_1 = 500
p_1 = 6

D_1 <- cooks.distance(model_1)
which(D_1 > qf(0.5, p_1+1, n_1-p_1-1))
```

```
## named integer(0)
```

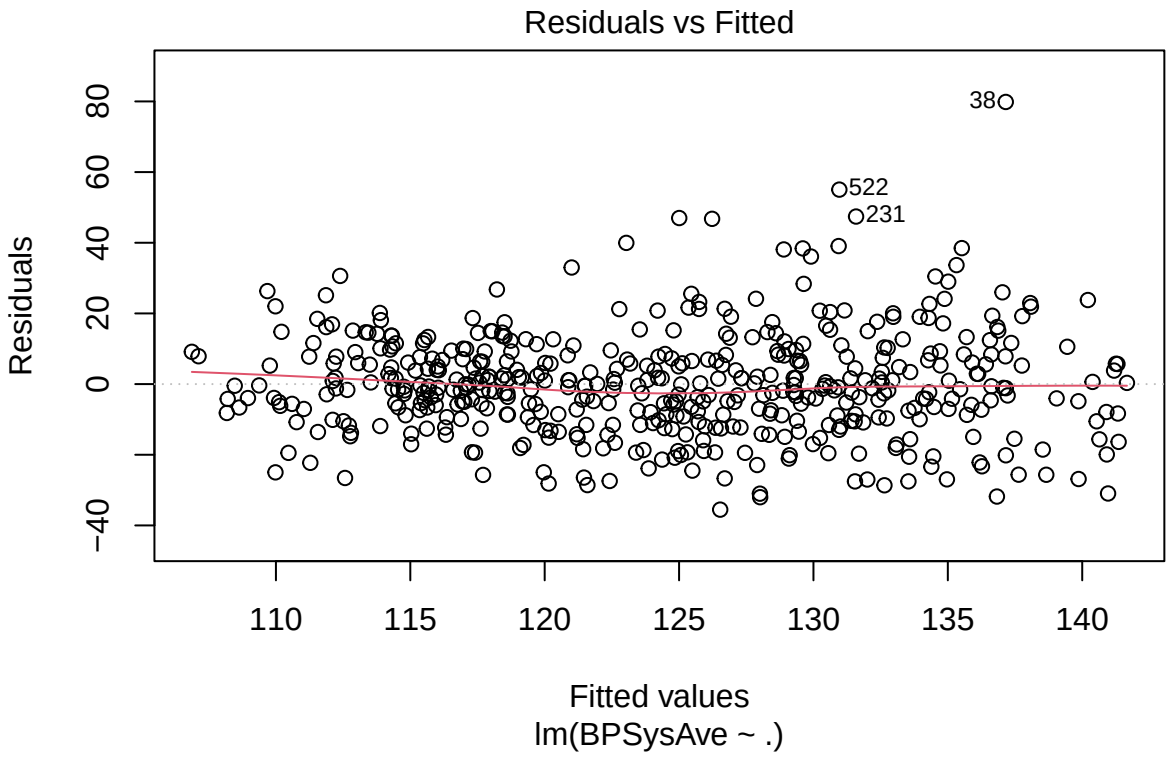
```
dfits_1 <- dffits(model_1)
dfits_ben_1 <- which(abs(dfits_1) > 2*sqrt((p_1+1)/n_1))

dfb_1 <- dfbetas(model_1)
dfb_ben_1 <- which(abs(dfb_1[,1]) > 2/sqrt(n_1))

# Remove potential outliers
outliers_1 <- intersect(dfits_ben_1, dfb_ben_1)
train_1 <- train[-c(outliers_1),]

# Fit new model
model_1_ad <- lm(BPSysAve ~ ., data = train_1[c(2, 3, 8, 9, 10, 12, 17)])
summary(model_1_ad)
```

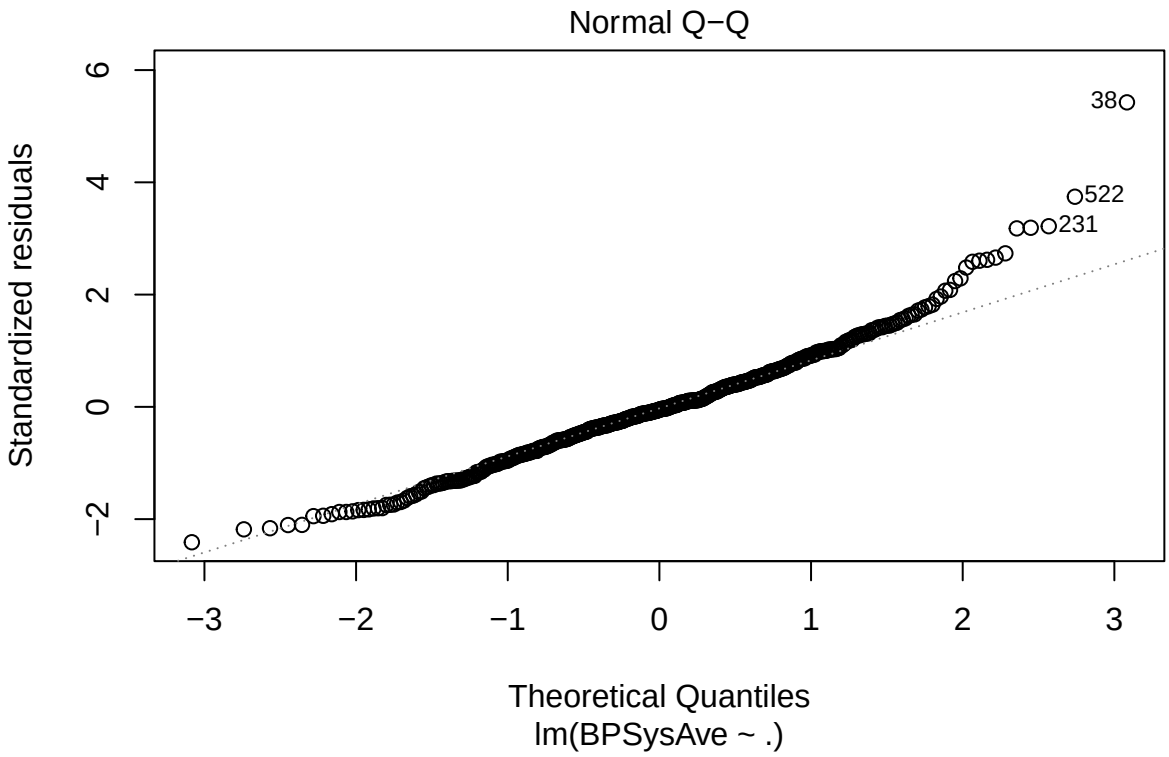
```
##
## Call:
## lm(formula = BPSysAve ~ ., data = train_1[c(2, 3, 8, 9, 10, 12,
##     17)])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -35.530  -8.894  -0.726   8.103  79.845
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 110.51351   16.89603   6.541 1.57e-10 ***
## Gendermale    4.56779    1.79005   2.552   0.0110 *
## Age           0.46027    0.04225  10.893  < 2e-16 ***
## Poverty      -1.18036    0.41956  -2.813   0.0051 **
## Weight        0.01659    0.03697   0.449   0.6538
## Height       -0.05982    0.10510  -0.569   0.5695
## SmokeNowYes  -0.19203    1.45245  -0.132   0.8949
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 14.81 on 481 degrees of freedom
## Multiple R-squared:  0.2437, Adjusted R-squared:  0.2342
## F-statistic: 25.83 on 6 and 481 DF,  p-value: < 2.2e-16
```

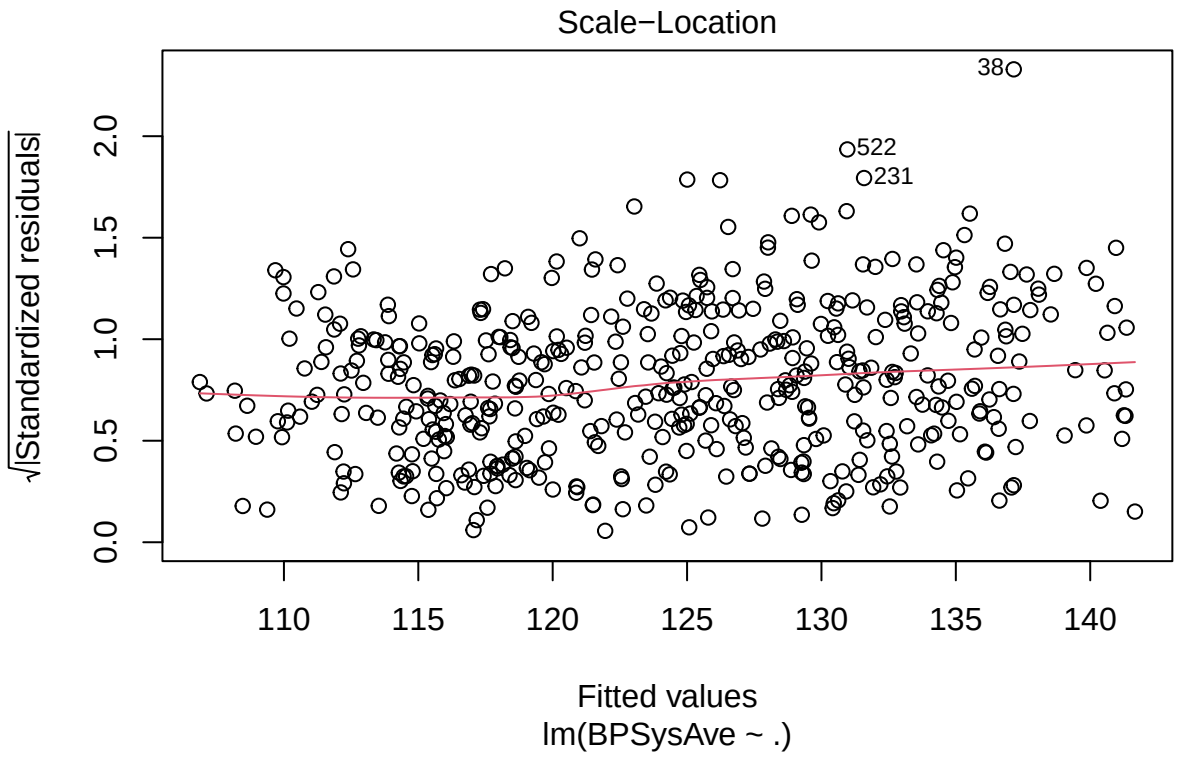
```
vif(model_1_ad)
```

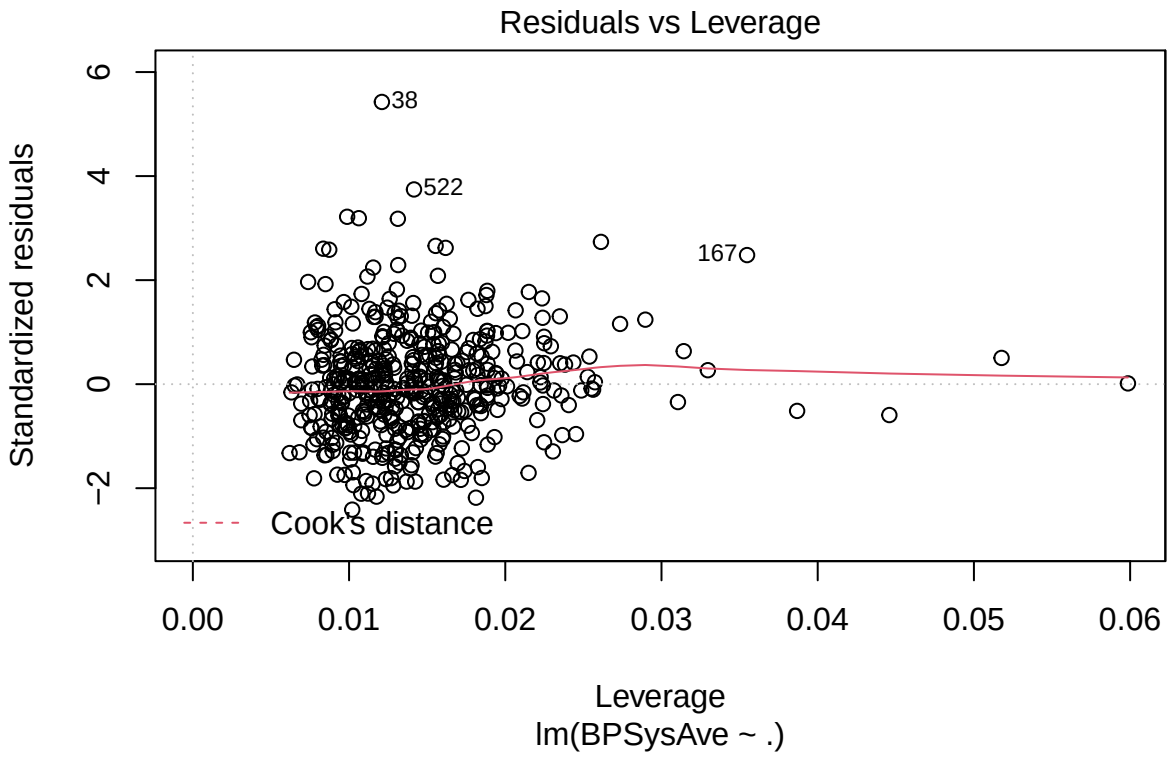
```
##   Gender      Age  Poverty   Weight   Height SmokeNow
## 1.744185 1.198577 1.061752 1.206422 2.089301 1.154925
```

```
plot(model_1_ad)
```


Residuals vs Fitted
Residuals
Fitted values
lm(BPSysAve ~ .)
38
522
231
−40
0
20
40
60
80
110
115
120
125
130
135
140


Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(BPSysAve ~ .)
38
522
231
−3
−2
−1
0
1
2
3
−2
0
2
4
6


Scale−Location
38
522
231
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
110
115
120
125
130
135
140
Fitted values
lm(BPSysAve ~ .)




```
crit_1_ad <- criteria(model = model_1_ad)
crit_1
```

```
##          R Squared Adjusted R Squared                AIC               AICc
##          0.2150677          0.2055147       2788.0755232       2788.3676124
##                BIC
##       2825.7923880
```

```
crit_1_ad
```

```
##          R Squared Adjusted R Squared                AIC               AICc
##          0.2436593          0.2342247       2635.3746373       2635.6740136
##                BIC
##       2672.8971605
```

From the variance inflation factor of the previous backward Stepwise AIC model, we can see that the predictors 'Weight', 'Height', 'BMI' have very high VIF, which are larger than the common cutoff 5. By the definition of the BMI, which is Weight/Height^2, I would drop the predictor BMI. After checking the influential observations from the training data and remove them from the training data, the prediction accuracy of the model has significantly improved in AIC, BIC, adjusted R^2. However, in the summary table, 'SmokeNow' has become less significant in the predicting model.

```
# Fitting a model based on backward Stepwise BIC selection and add SmokeNow
vif(sel_var_bic_back_mol)
```

```
##   Gender      Age  Poverty
## 1.000772 1.016094 1.016830
```

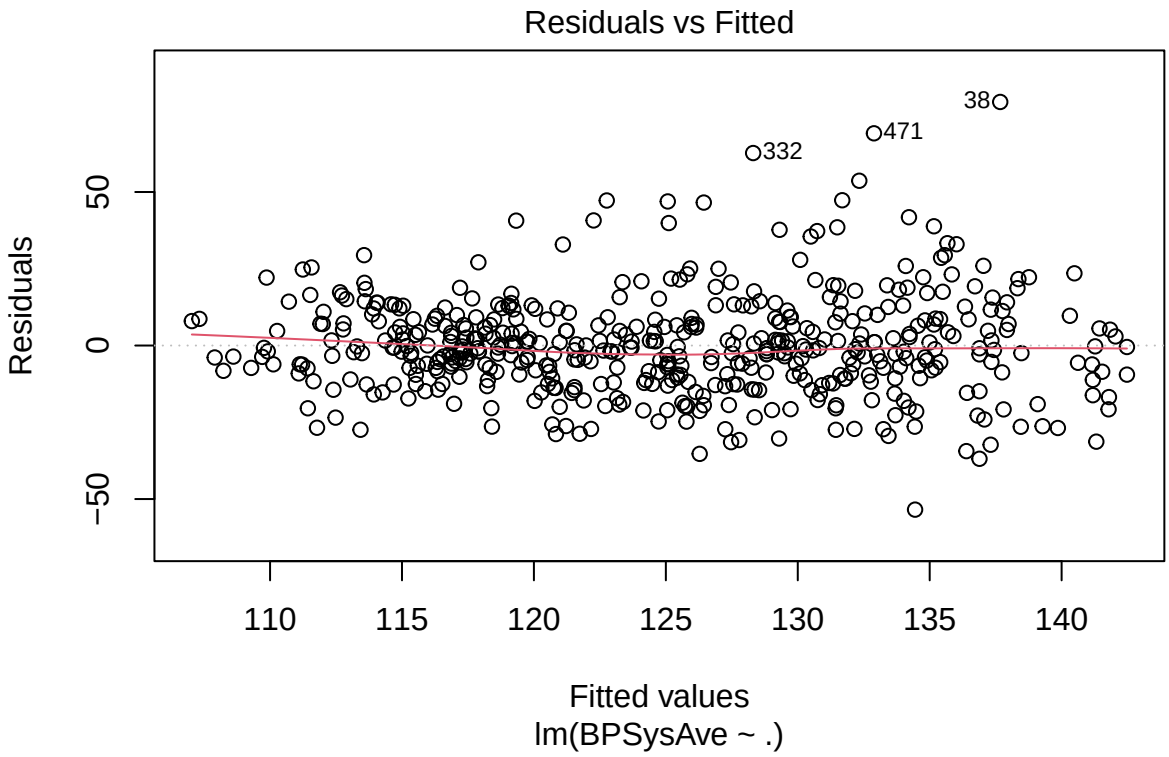
```
model_2 <- lm(BPSysAve ~ ., data = train[c(2, 3, 8, 12, 17)])
summary(model_2)
```

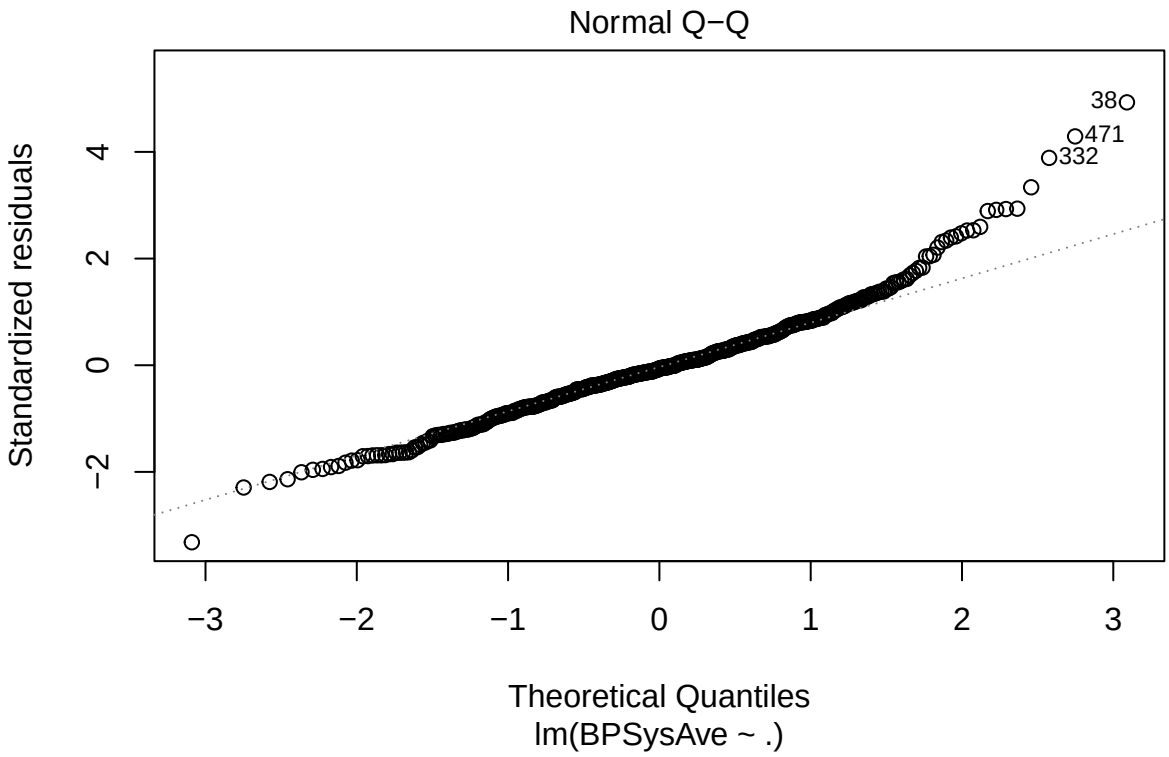
```
##
## Call:
## lm(formula = BPSysAve ~ ., data = train[c(2, 3, 8, 12, 17)])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -53.446  -9.565  -1.034   8.500  79.327
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 103.65873    2.96927  34.910  < 2e-16 ***
## Gendermale    4.07690    1.46771   2.778  0.00568 **
## Age           0.45512    0.04426  10.282  < 2e-16 ***
## Poverty      -1.45173    0.44727  -3.246  0.00125 **
## SmokeNowYes  -0.78610    1.56034  -0.504  0.61463
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 16.19 on 495 degrees of freedom
## Multiple R-squared:   0.21,  Adjusted R-squared:  0.2036
## F-statistic: 32.89 on 4 and 495 DF,  p-value: < 2.2e-16
```

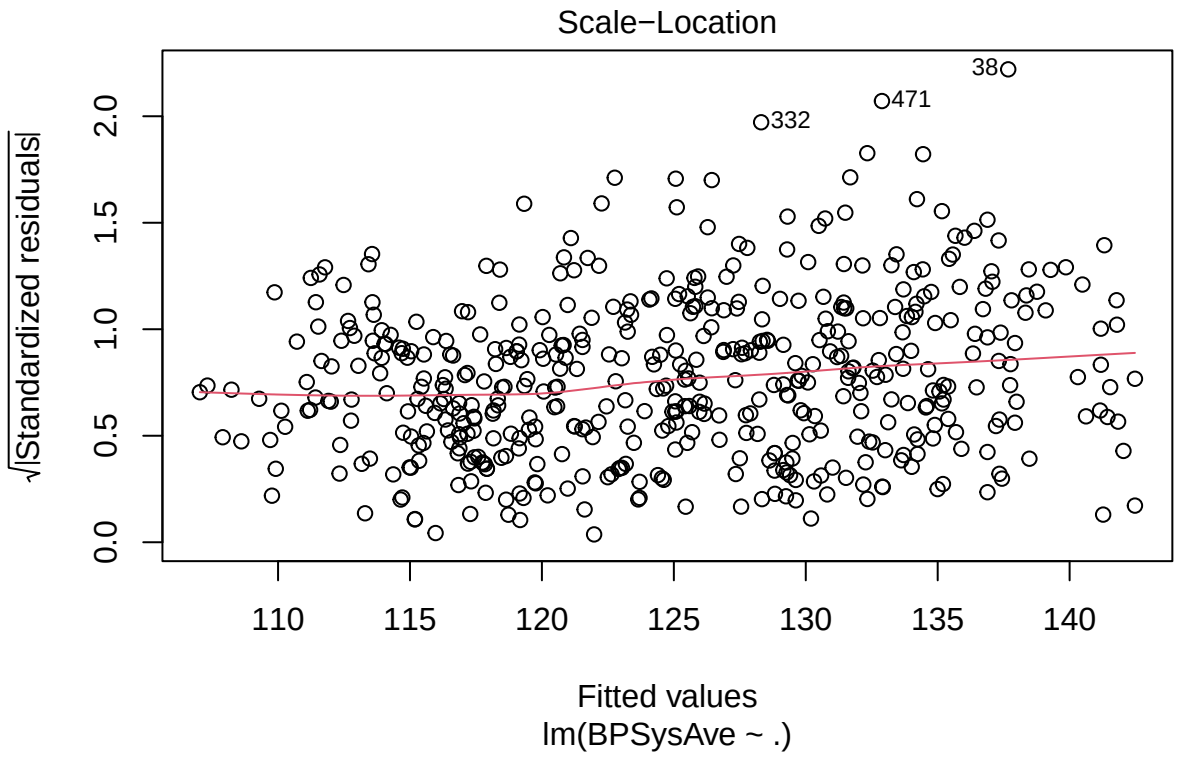
```
vif(model_2)
```

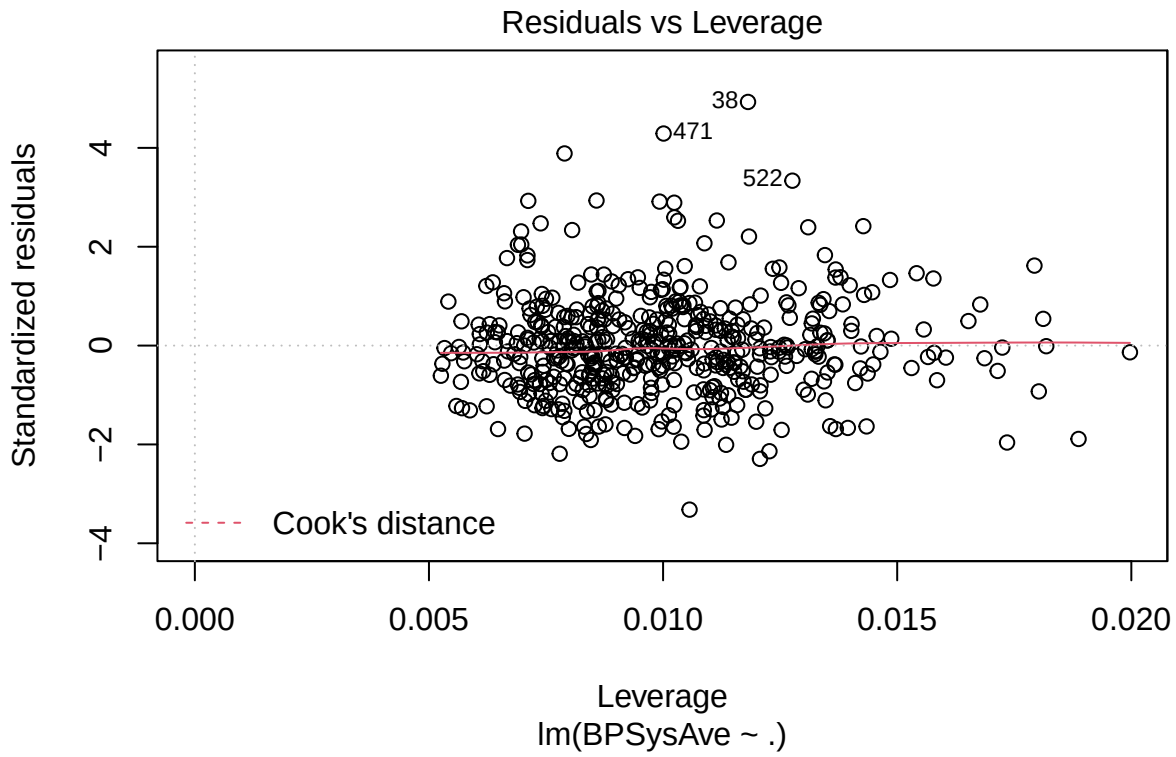
```
##   Gender      Age  Poverty SmokeNow
## 1.003664 1.127515 1.034859 1.143319
```

```
plot(model_2)
```


Residuals vs Fitted
Residuals
50
0
−50
38
471
332
110
115
120
125
130
135
140
Fitted values
lm(BPSysAve ~ .)


Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(BPSysAve ~ .)
38
471
332
−3
−2
−1
0
1
2
3
−2
0
2
4


Scale−Location
38
471
332
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
110
115
120
125
130
135
140
Fitted values
lm(BPSysAve ~ .)




```
crit_2 <- criteria(model = model_2)

# Diagnostics check in Cook's distance, DFFITS, DFBETAS
n_2 = 500
p_2 = 4

D_2 <- cooks.distance(model_2)
which(D_2 > qf(0.5, p_2+1, n_2-p_2-1))
```

```
## named integer(0)
```

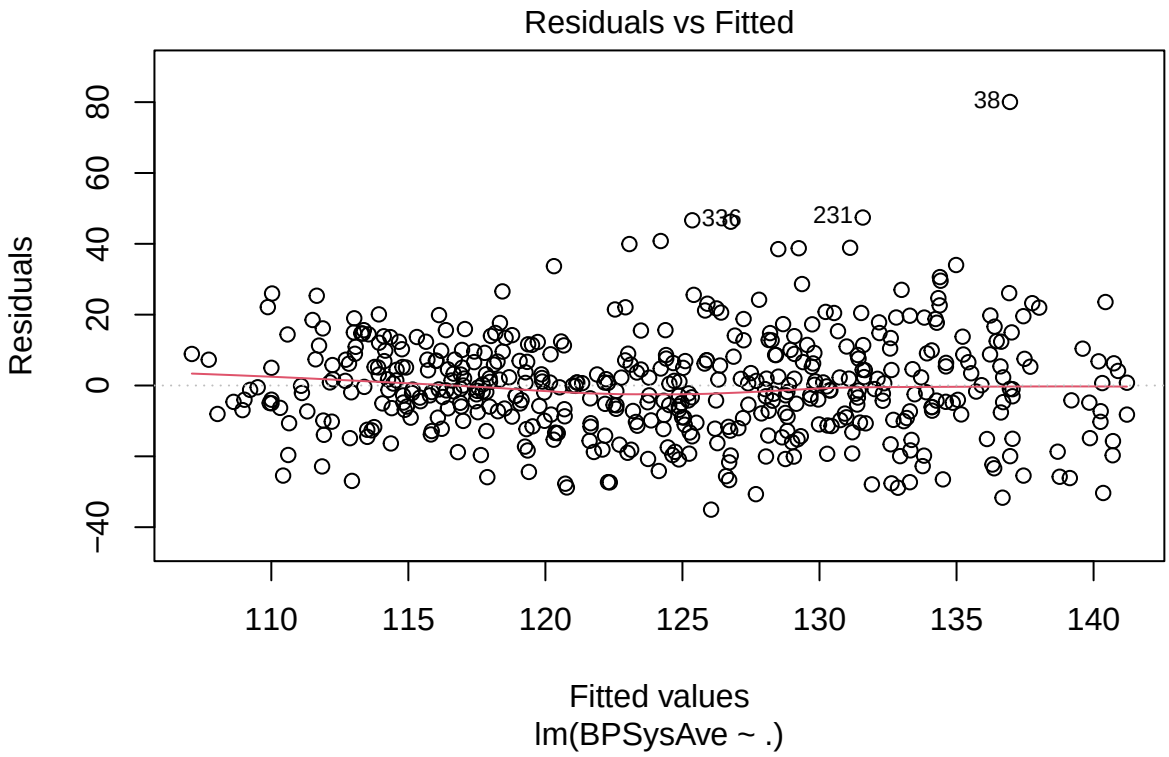
```
dfits_2 <- dffits(model_2)
dfits_ben_2 <- which(abs(dfits_2) > 2*sqrt((p_2+1)/n_2))

dfb_2 <- dfbetas(model_2)
dfb_ben_2 <- which(abs(dfb_2[,1]) > 2/sqrt(n_2))

# Remove potential outliers
outliers_2 <- intersect(dfits_ben_2, dfb_ben_2)
train_2 <- train[-c(outliers_2),]

# Fit new model
model_2_ad <- lm(BPSysAve ~ ., data = train_2[c(2, 3, 8, 12, 17)])
summary(model_2_ad)
```

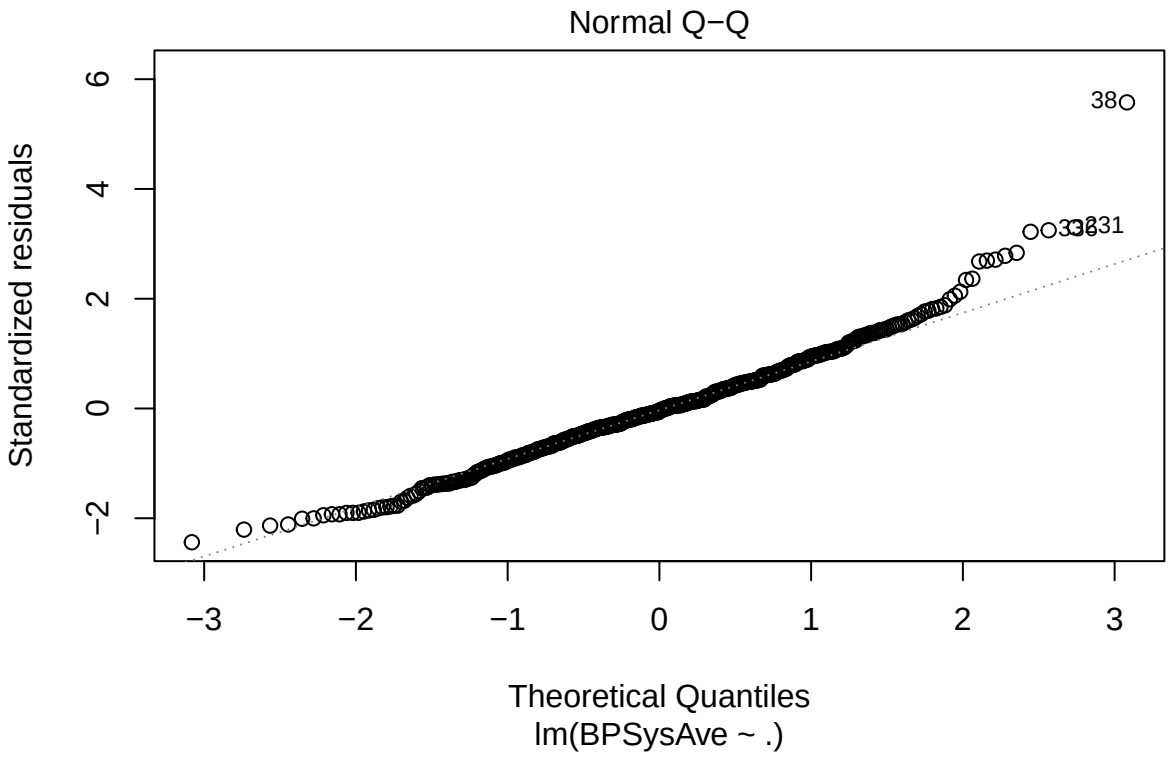
```
##
## Call:
## lm(formula = BPSysAve ~ ., data = train_2[c(2, 3, 8, 12, 17)])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -35.046  -9.059  -0.552   8.136  80.051
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 101.28750    2.73329  37.057  < 2e-16 ***
## Gendermale    3.72578    1.32762   2.806  0.00521 **
## Age           0.46814    0.04051  11.556  < 2e-16 ***
## Poverty      -1.08441    0.40710  -2.664  0.00799 **
## SmokeNowYes   0.11615    1.43074   0.081  0.93533
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 14.44 on 480 degrees of freedom
## Multiple R-squared:  0.2488, Adjusted R-squared:  0.2425
## F-statistic: 39.74 on 4 and 480 DF,  p-value: < 2.2e-16
```

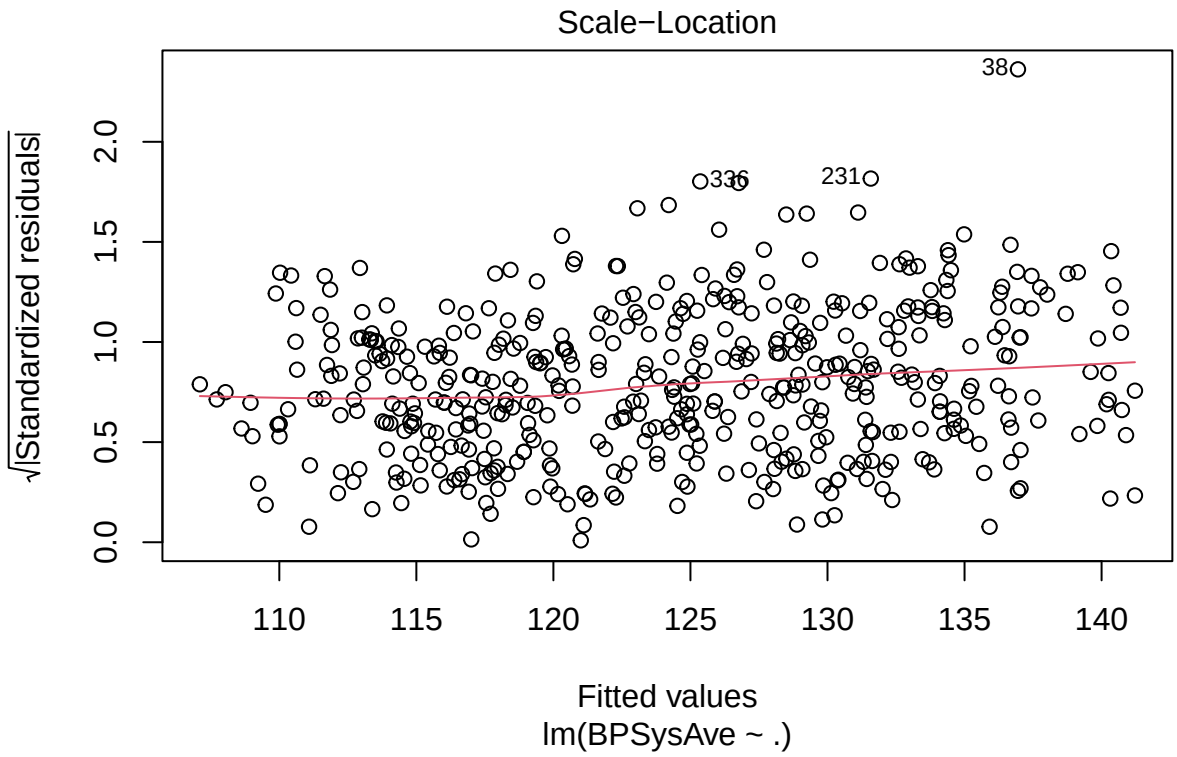
```
vif(model_2_ad)
```

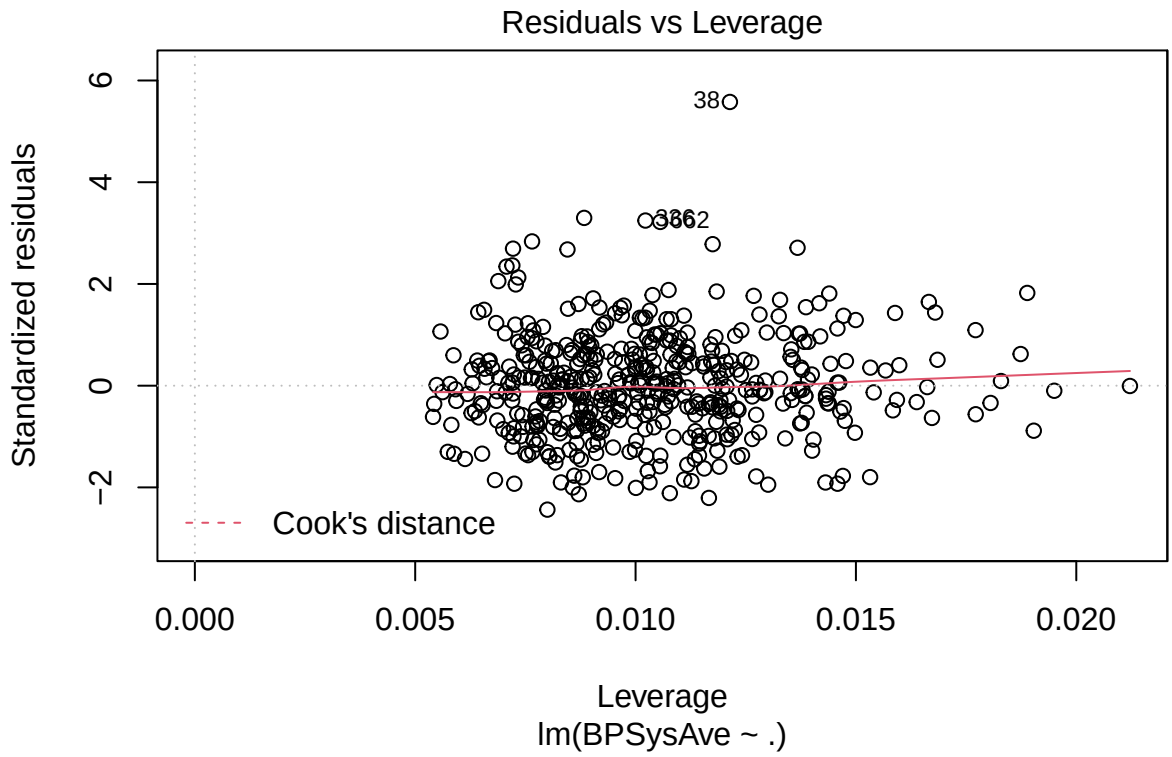
```
##   Gender      Age  Poverty SmokeNow
## 1.003091 1.148209 1.036456 1.170385
```

```
plot(model_2_ad)
```


Residuals vs Fitted
Residuals
Fitted values
lm(BPSysAve ~ .)
38
336
231
80
60
40
20
0
−40
110
115
120
125
130
135
140


Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(BPSysAve ~ .)
38
−3
−2
−1
0
1
2
3
6
4
2
0
−2


Scale−Location
38
336
231
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
110
115
120
125
130
135
140
Fitted values
lm(BPSysAve ~ .)




```
crit_2_ad <- criteria(model = model_2_ad)
crit_2
```

```
##          R Squared Adjusted R Squared               AIC              AICc
##          0.2099558          0.2035716       2787.3211923       2787.4908892
##                BIC
##       2816.6088409
```

```
crit_2_ad
```

```
##          R Squared Adjusted R Squared               AIC              AICc
##          0.2487891          0.2425290       2592.8164034       2592.9914034
##                BIC
##       2621.9212968
```

There is no significantly large VIF in this model, so we would examine the influential observations in the training data that potentially affect the model prediction. The prediction accuracy of the model has significantly improved in AIC, BIC, adjusted R^2 after we removed the potential outliers.

```
# Fitting a model based on LASSO selection and add SmokeNow
model_3 <- lm(BPSysAve ~ ., data = train[c(3, 12, 17)])
summary(model_3)
```

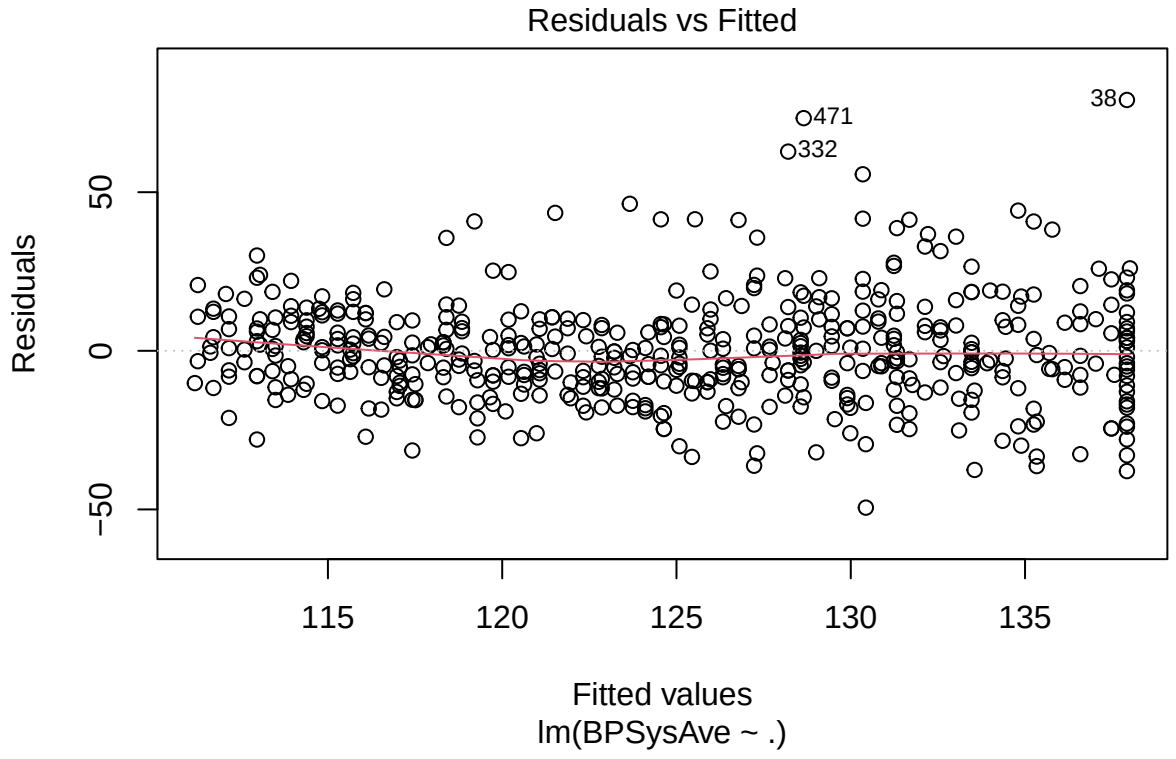
```
##
```

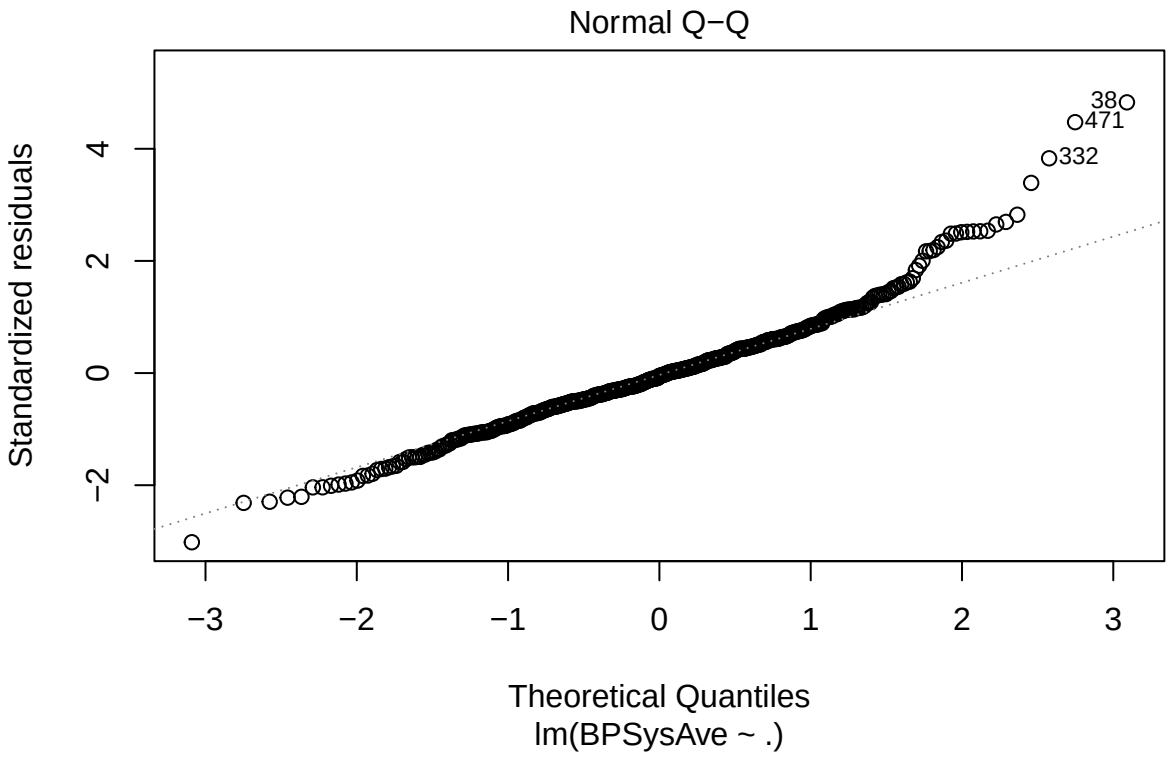
```
## Call:
## lm(formula = BPSysAve ~ ., data = train[c(3, 12, 17)])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -49.433  -9.664  -0.862   8.551  79.077
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 102.26805    2.69910  37.890   <2e-16 ***
## Age           0.44569    0.04482   9.944   <2e-16 ***
## SmokeNowYes   0.08702    1.56892   0.055    0.956
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 16.44 on 497 degrees of freedom
## Multiple R-squared:  0.1818, Adjusted R-squared:  0.1785
## F-statistic: 55.21 on 2 and 497 DF,  p-value: < 2.2e-16
```

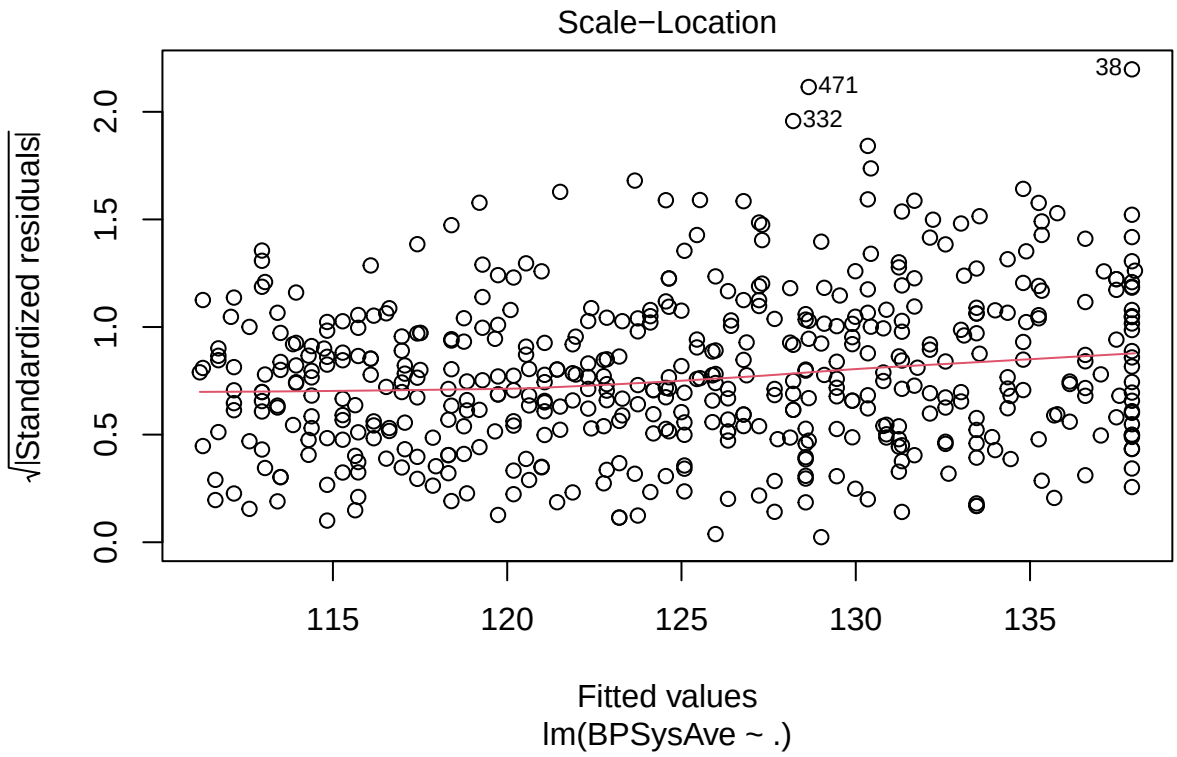
```
vif(model_3)
```

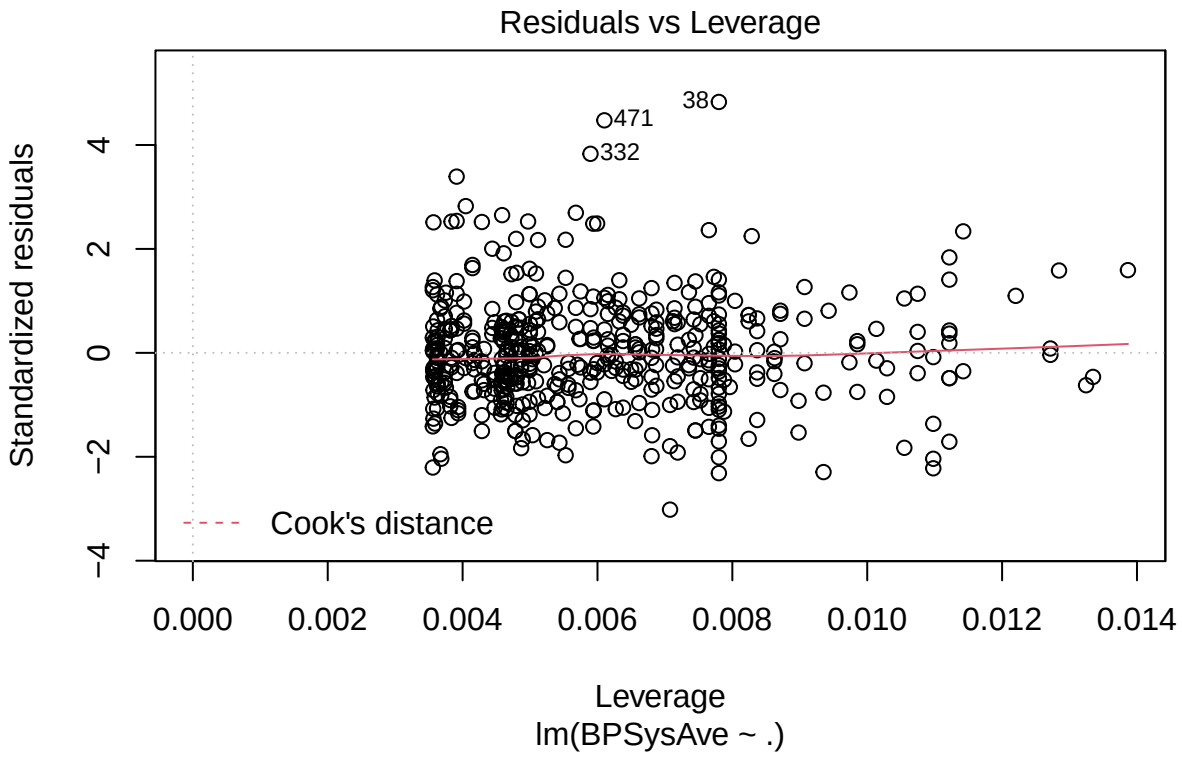
```
##      Age SmokeNow
## 1.120637 1.120637
```

```
plot(model_3)
```




Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(BPSysAve ~ .)
38
471
332
−3
−2
−1
0
1
2
3
−2
0
2
4


Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
471
332
38
115
120
125
130
135
Fitted values
lm(BPSysAve ~ .)




```
crit_3 <- criteria(model = model_3)

# Diagnostics check in Cook's distance, DFFITS, DFBETAS
n_3 = 500
p_3 = 2

D_3 <- cooks.distance(model_3)
which(D_3 > qf(0.5, p_3+1, n_3-p_3-1))
```

```
## named integer(0)
```

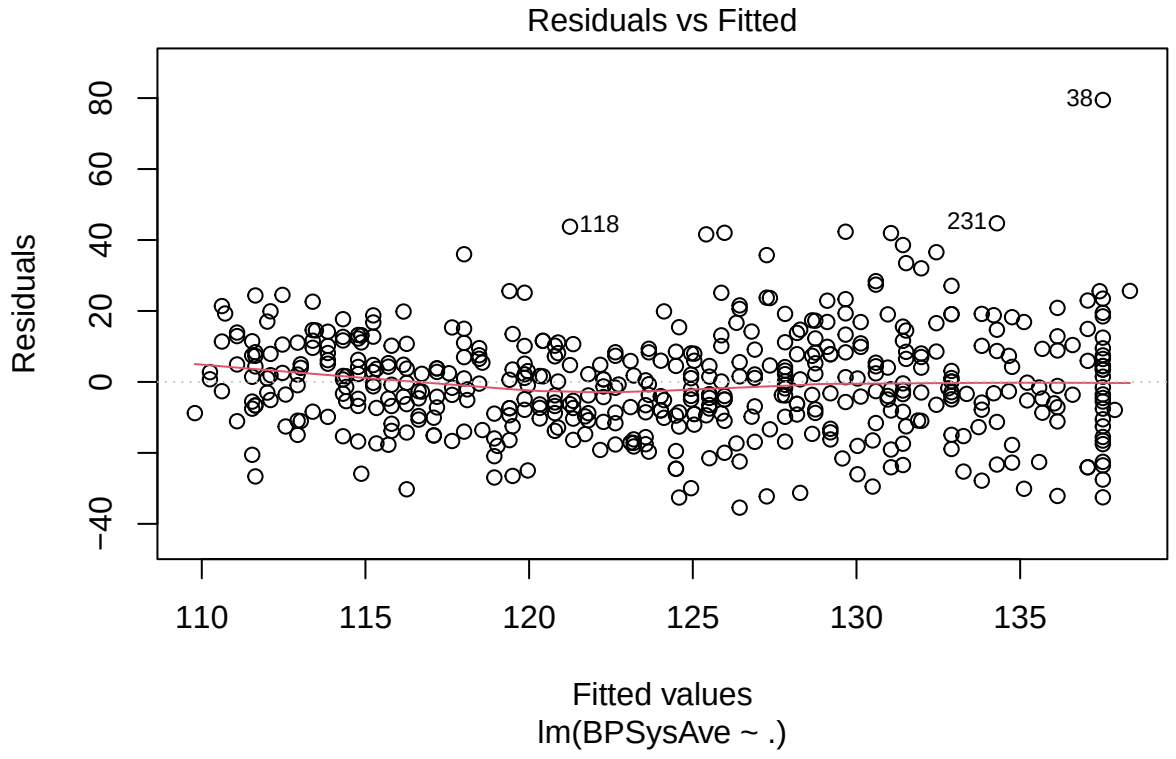
```
dfits_3 <- dffits(model_3)
dfits_ben_3 <- which(abs(dfits_3) > 2*sqrt((p_3+1)/n_3))

dfb_3 <- dfbetas(model_3)
dfb_ben_3 <- which(abs(dfb_3[,1]) > 2/sqrt(n_3))

# Remove potential outliers
outliers_3 <- intersect(dfits_ben_3, dfb_ben_3)
train_3 <- train[-c(outliers_3),]

# Fit new model
model_3_ad <- lm(BPSysAve ~ ., data = train_2[c(3, 12, 17)])
summary(model_3_ad)
```

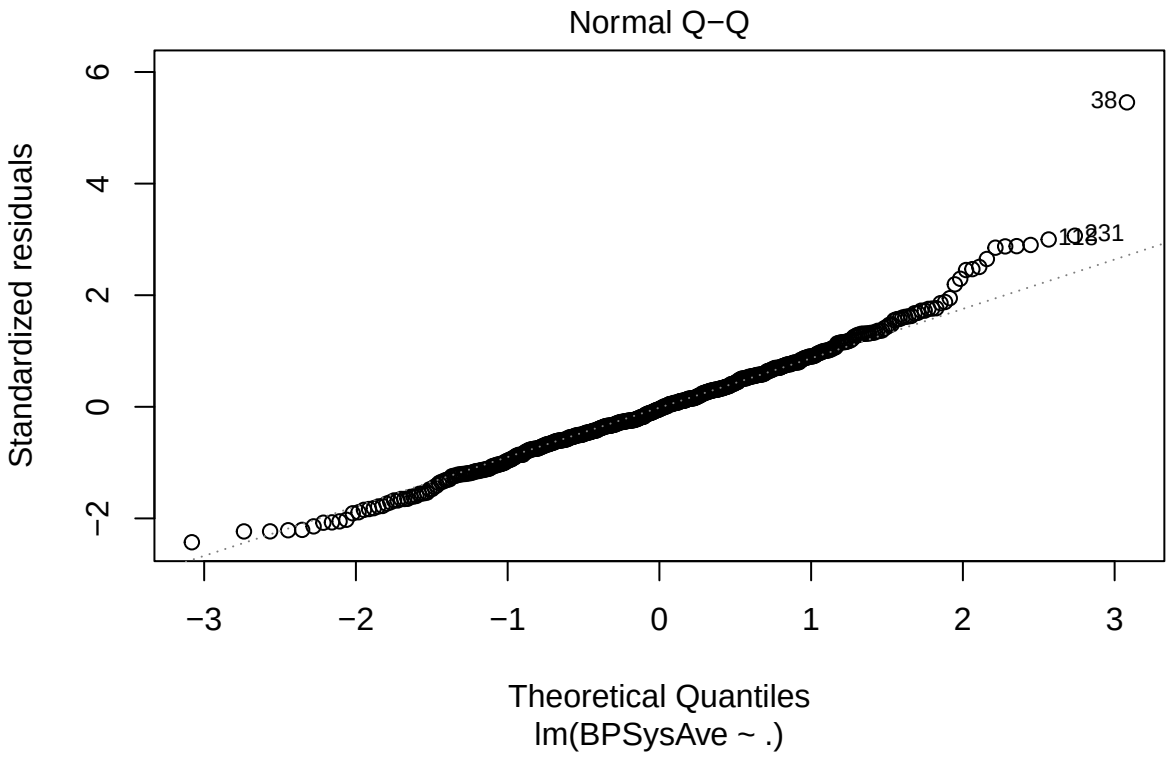
```
##
## Call:
## lm(formula = BPSysAve ~ ., data = train_2[c(3, 12, 17)])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -35.431  -8.937  -0.658   8.483  79.473
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 100.54207    2.46403  40.804   <2e-16 ***
## Age           0.46231    0.04095  11.288   <2e-16 ***
## SmokeNowYes   0.82742    1.43268   0.578    0.564
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 14.62 on 482 degrees of freedom
## Multiple R-squared:  0.2262, Adjusted R-squared:  0.2229
## F-statistic: 70.43 on 2 and 482 DF,  p-value: < 2.2e-16
```

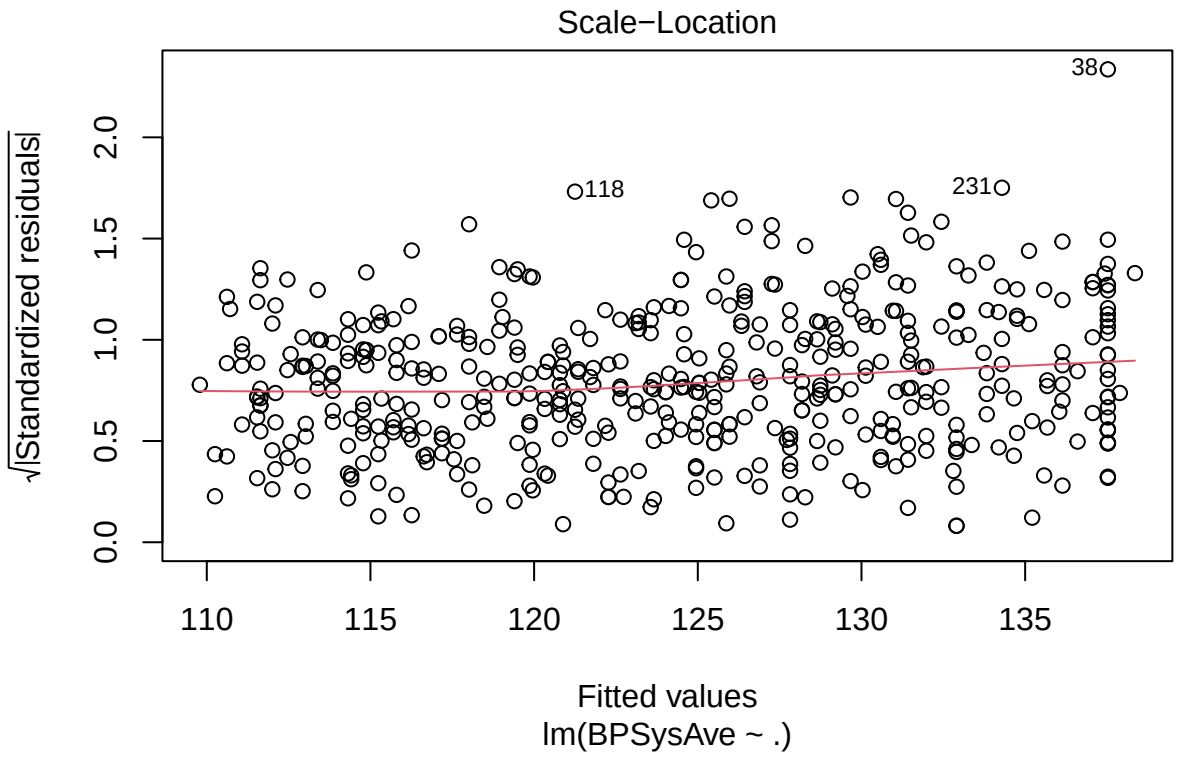
```
vif(model_3_ad)
```

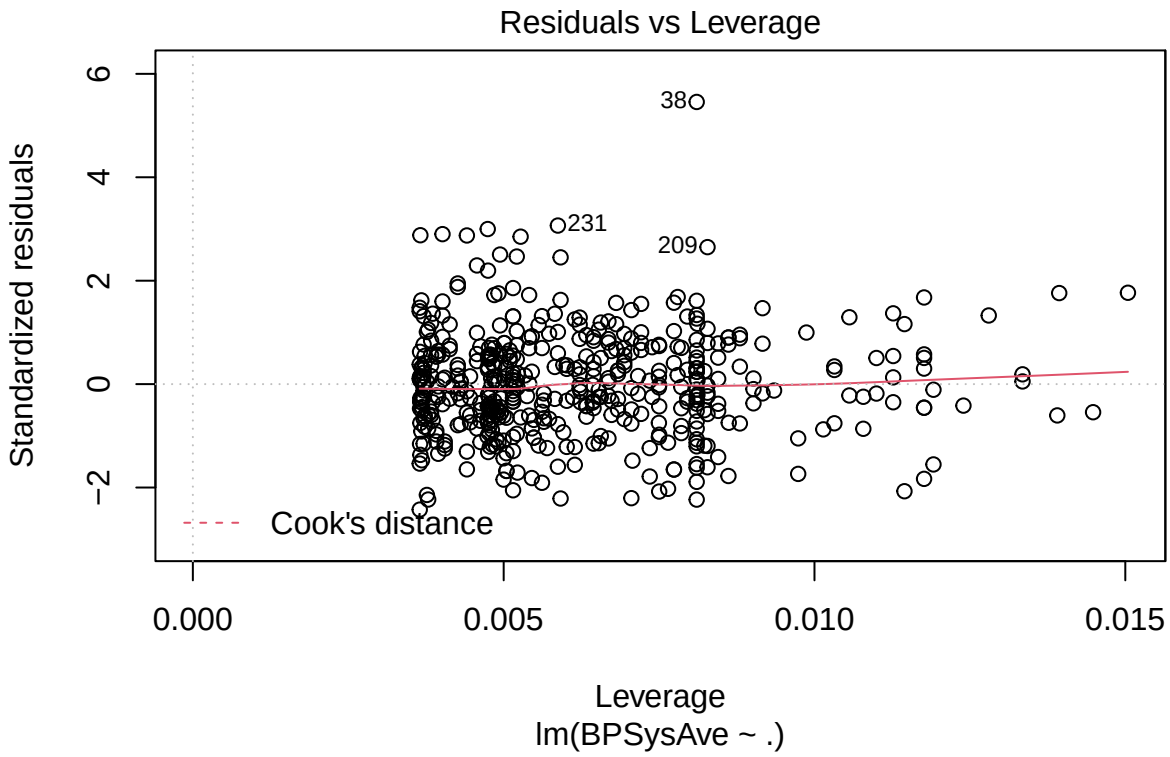
```
##      Age SmokeNow
## 1.143987 1.143987
```

```
plot(model_3_ad)
```




Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(BPSysAve ~ .)
38
231
−3
−2
−1
0
1
2
3
−2
0
2
4
6


Scale−Location
38
118
231
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
110
115
120
125
130
135
Fitted values
lm(BPSysAve ~ .)




```
crit_3_ad <- criteria(model = model_3_ad)
crit_3
```

```
##          R Squared Adjusted R Squared               AIC              AICc
##          0.1817846          0.1784920       2800.8395797       2800.9200626
##                BIC
##       2821.6980120
```

```
crit_3_ad
```

```
##          R Squared Adjusted R Squared               AIC              AICc
##          0.2261542          0.2229432       2603.2142294       2603.2972169
##                BIC
##       2623.9508249
```

The VIF for the model consisting predictor 'Age' and 'SmokeNow' indicates no strong multicollinearity of the model. After we check the influential observations for the model in training data and remove potential outliers, the prediction accuracy measure has significantly improved. Also, in this model, the level of statistical significance for predictor 'SmokeNow' greatly improved.

Model Validation

K-fold Cross validation

Cross validation is a resampling technique of the training data to evaluate the model we constructed. K refers to the number of groups that the training data is to be split into. We would first shuffle the data set randomly and split the data set into k groups. Each group of data is used to be testing data once and used to train the model K-1 times. K is chosen such that divided training data and testing data is large enough to be representative of the broader data set. We would fix K to be 10, which is value that generally found to generate relatively low variance and bias through experimentation.

shrinkage method with k-fold cross validation

```
## Ridge regression for Stepwise backward aic, bic models, and for LASSO selection model with 10-fold c
set.seed(1006562550)
library(glmnet)
library(rms)
```

```
## Loading required package: SparseM
```

```
##
## Attaching package: 'SparseM'
```

```
## The following object is masked from 'package:base':
##
##     backsolve
```

```
##
## Attaching package: 'rms'
```

```
## The following objects are masked from 'package:car':
##
##     Predict, vif
```

```
library(MASS)
## model_1
cv_ridge_1 <- cv.glmnet(x = model.matrix(~., train_1[c(2, 3, 8, 9, 10, 17)]), y = train_1$BPSysAve, sta
# fit best model
lambda_ridge_1 <- cv_ridge_1$lambda.min
model_ridge_1 <- glmnet(x = model.matrix(~., train_1[c(2, 3, 8, 9, 10, 17)]), y = train_1$BPSysAve, sta
# Prediction
pred_ridge_1 <- predict(model_ridge_1, newx = model.matrix(~., test[c(2, 3, 8, 9, 10, 17)]), type = "re
# Prediction error
pred_err1 <- mean((test$BPSysAve - pred_ridge_1)^2)
coef(model_ridge_1)
```

```
## 8 x 1 sparse Matrix of class "dgCMatrix"
##                       s0
## (Intercept) 111.22147431
## (Intercept)   .
## Gendermale    4.32639136
## Age           0.43735768
```

```
## Poverty      -1.11066428
## Weight        0.01584661
## Height       -0.05650993
## SmokeNowYes  -0.40206125
```

```
## model_2
cv_ridge_2 <- cv.glmnet(x = model.matrix(~., train_2[c(2, 3, 8, 17)]), y = train_2$BPSysAve, standardiz
# fit best model
lambda_ridge_2 <- cv_ridge_2$lambda.min
model_ridge_2 <- glmnet(x = model.matrix(~., train_2[c(2, 3, 8, 17)]), y = train_2$BPSysAve, standardiz
# Prediction
pred_ridge_2 <- predict(model_ridge_2, newx = model.matrix(~., test[c(2, 3, 8, 17)]), type = "response"
# Prediction error
pred_err2 <- mean((test$BPSysAve - pred_ridge_2)^2)
coef(model_ridge_2)
```

```
## 6 x 1 sparse Matrix of class "dgCMatrix"
##                       s0
## (Intercept) 102.5574535
## (Intercept)   .
## Gendermale    3.5536187
## Age           0.4437296
## Poverty      -1.0175193
## SmokeNowYes  -0.1344950
```

```
## model_3
cv_ridge_3 <- cv.glmnet(x = model.matrix(~., train_3[c(3, 17)]), y = train_3$BPSysAve, standardize = T,
# fit best model
lambda_ridge_3 <- cv_ridge_3$lambda.min
model_ridge_3 <- glmnet(x = model.matrix(~., train_3[c(3, 17)]), y = train_3$BPSysAve, standardize = T,
# Prediction
pred_ridge_3 <- predict(model_ridge_3, newx = model.matrix(~., test[c(3, 17)]), type = "response")
# Prediction error
pred_err3 <- mean((test$BPSysAve - pred_ridge_3)^2)
coef(model_ridge_3)
```

```
## 4 x 1 sparse Matrix of class "dgCMatrix"
##                       s0
## (Intercept) 103.9302826
## (Intercept)   .
## Age           0.4069142
## SmokeNowYes  -0.4459712
```

```
# Comparing the three model prediction error
c(pred_err1, pred_err2, pred_err3)
```

```
## [1] 229.2896 230.0168 233.9152
```

```
## LASSO regression for Stepwise backward aic, bic models, and for LASSO selection model with 10-fold c

set.seed(1006562550)
library(glmnet)
```

```r
library(rms)
library(MASS)
## model_1
cv_lasso_1 <- cv.glmnet(x = model.matrix(~., train_1[c(2, 3, 8, 9, 10, 17)]), y = train_1$BPSysAve, sta
# fit best model
lambda_lasso_1 <- cv_lasso_1$lambda.min
model_lasso_1 <- glmnet(x = model.matrix(~., train_1[c(2, 3, 8, 9, 10, 17)]), y = train_1$BPSysAve, sta
# Prediction
pred_lasso_1 <- predict(model_lasso_1, newx = model.matrix(~., test[c(2, 3, 8, 9, 10, 17)]), type = "re
# Prediction error
pred_err4 <- mean((test$BPSysAve - pred_lasso_1)^2)
coef(model_lasso_1)
```

```
## 8 x 1 sparse Matrix of class "dgCMatrix"
##                       s0
## (Intercept) 102.4954387
## (Intercept)   .
## Gendermale    3.3578628
## Age           0.4474672
## Poverty      -0.9981372
## Weight        .
## Height        .
## SmokeNowYes   .
```

```r
## model_2
cv_lasso_2 <- cv.glmnet(x = model.matrix(~., train_2[c(2, 3, 8, 17)]), y = train_2$BPSysAve, standardiz
# fit best model
lambda_lasso_2 <- cv_lasso_2$lambda.min
model_lasso_2 <- glmnet(x = model.matrix(~., train_2[c(2, 3, 8, 17)]), y = train_2$BPSysAve, standardiz
# Prediction
pred_lasso_2 <- predict(model_lasso_2, newx = model.matrix(~., test[c(2, 3, 8, 17)]), type = "response"
# Prediction error
pred_err5 <- mean((test$BPSysAve - pred_lasso_2)^2)
coef(model_lasso_2)
```

```
## 6 x 1 sparse Matrix of class "dgCMatrix"
##                       s0
## (Intercept) 101.9668271
## (Intercept)   .
## Gendermale    3.2900178
## Age           0.4531920
## Poverty      -0.9408069
## SmokeNowYes   .
```

```r
## model_3
cv_lasso_3 <- cv.glmnet(x = model.matrix(~., train_3[c(3, 17)]), y = train_3$BPSysAve, standardize = T,
# fit best model
lambda_lasso_3 <- cv_lasso_3$lambda.min
model_lasso_3 <- glmnet(x = model.matrix(~., train_3[c(3, 17)]), y = train_3$BPSysAve, standardize = T,
# Prediction
pred_lasso_3 <- predict(model_lasso_3, newx = model.matrix(~., test[c(3, 17)]), type = "response")
# Prediction error
```

```r
pred_err6 <- mean((test$BPSysAve - pred_lasso_3)^2)
coef(model_lasso_3)
```

```
## 4 x 1 sparse Matrix of class "dgCMatrix"
##                       s0
## (Intercept) 103.3751662
## (Intercept)   .
## Age           0.4140734
## SmokeNowYes   .
```

```r
# Comparing the three model prediction error
c(pred_err4, pred_err5, pred_err6)
```

```
## [1] 230.5053 230.1152 233.8187
```

```r
## Elastic-Net with (alpha = 0.5) regression for Stepwise backward aic, bic models, and for LASSO selec
set.seed(1006562550)
library(glmnet)
library(rms)
library(MASS)
## model_1
cv_en_1 <- cv.glmnet(x = model.matrix(~., train_1[c(2, 3, 8, 9, 10, 17)]), y = train_1$BPSysAve, standa
# fit best model
lambda_en_1 <- cv_en_1$lambda.min
model_en_1 <- glmnet(x = model.matrix(~., train_1[c(2, 3, 8, 9, 10, 17)]), y = train_1$BPSysAve, standa
# Prediction
pred_en_1 <- predict(model_en_1, newx = model.matrix(~., test[c(2, 3, 8, 9, 10, 17)]), type = "response
# Prediction error
pred_err7 <- mean((test$BPSysAve - pred_en_1)^2)
coef(model_en_1)
```

```
## 8 x 1 sparse Matrix of class "dgCMatrix"
##                       s0
## (Intercept) 102.7703565
## (Intercept)   .
## Gendermale    3.3547513
## Age           0.4417230
## Poverty      -0.9910965
## Weight        .
## Height        .
## SmokeNowYes   .
```

```r
## model_2
cv_en_2 <- cv.glmnet(x = model.matrix(~., train_2[c(2, 3, 8, 17)]), y = train_2$BPSysAve, standardize =
# fit best model
lambda_en_2 <- cv_en_2$lambda.min
model_en_2 <- glmnet(x = model.matrix(~., train_2[c(2, 3, 8, 17)]), y = train_2$BPSysAve, standardize =
# Prediction
pred_en_2 <- predict(model_en_2, newx = model.matrix(~., test[c(2, 3, 8, 17)]), type = "response")
# Prediction error
pred_err8 <- mean((test$BPSysAve - pred_en_2)^2)
coef(model_en_2)
```

```
## 6 x 1 sparse Matrix of class "dgCMatrix"
##                       s0
## (Intercept) 101.9824010
## (Intercept)   .
## Gendermale    3.3948637
## Age           0.4533824
## Poverty      -0.9730788
## SmokeNowYes   .
```

```
## model_3
cv_en_3 <- cv.glmnet(x = model.matrix(~., train_3[c(3, 17)]), y = train_3$BPSysAve, standardize = T, al
# fit best model
lambda_en_3 <- cv_en_3$lambda.min
model_en_3 <- glmnet(x = model.matrix(~., train_3[c(3, 17)]), y = train_3$BPSysAve, standardize = T, al
# Prediction
pred_en_3 <- predict(model_en_3, newx = model.matrix(~., test[c(3, 17)]), type = "response")
# Prediction error
pred_err9 <- mean((test$BPSysAve - pred_en_3)^2)
coef(model_en_3)
```

```
## 4 x 1 sparse Matrix of class "dgCMatrix"
##                        s0
## (Intercept) 103.19025257
## (Intercept)   .
## Age           0.41789747
## SmokeNowYes  -0.02057598
```

```
# Comparing the three model prediction error
c(pred_err7, pred_err8, pred_err9)
```

```
## [1] 230.3959 230.1459 233.8328
```

```
# The lowest prediction error of all shrinkage regression model
which.min(c(pred_err1, pred_err2, pred_err3, pred_err4, pred_err5, pred_err6, pred_err7, pred_err8, pre
```

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

```
c(pred_err1, pred_err2, pred_err3, pred_err4, pred_err5, pred_err6, pred_err7, pred_err8, pred_err9)
```

```
## [1] 229.2896 230.0168 233.9152 230.5053 230.1152 233.8187 230.3959 230.1459
## [9] 233.8328
```

```
min(c(pred_err1, pred_err2, pred_err3, pred_err4, pred_err5, pred_err6, pred_err7, pred_err8, pred_err9
```

```
## [1] 229.2896
```

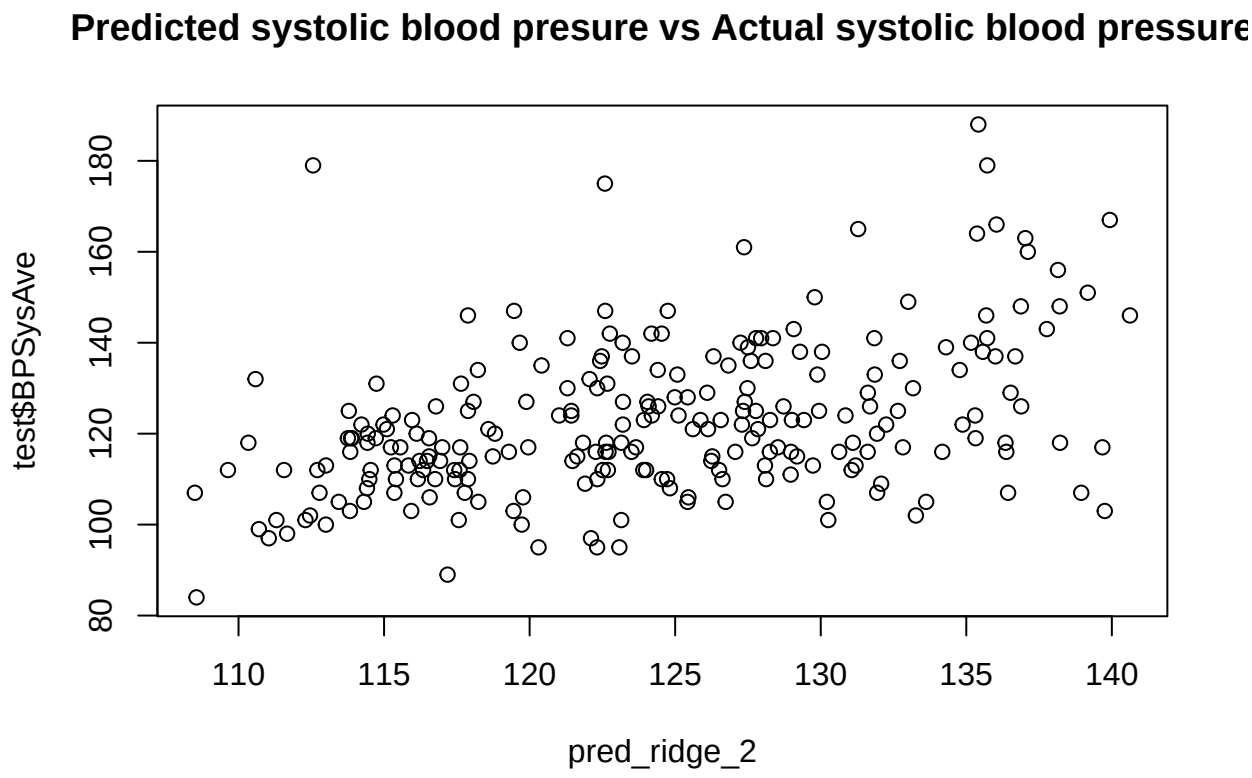
```
# The Stepwise variable BIC selection with additional 'SmokeNow' variable under ridge penalty has the l
pred_err2
```

```
## [1] 230.0168
```

```
coef(model_ridge_2)
```

```
## 6 x 1 sparse Matrix of class "dgCMatrix"
##                       s0
## (Intercept) 102.5574535
## (Intercept)   .
## Gendermale    3.5536187
## Age           0.4437296
## Poverty      -1.0175193
## SmokeNowYes  -0.1344950
```

```
bp_plot <- plot(pred_ridge_2, test$BPSysAve, main = "Predicted systolic blood presure vs Actual systolic
```




```
capture.output(bp_plot,file="BPSysAve.predicted.vs.Actual")
```

Reference

National Health and Nutrition Examination Survey (NHANES). (2020, August 27). Retrieved June 12, 2021, from https://www.cdc.gov/aging/publications/nhanes/index.html

Nhanes 2011-2012 overview. (2021, April 23). Retrieved June 12, 2021, from https://wwwn.cdc.gov/nchs/nhanes/continuousnhanes/overview.aspx?BeginYear=2011