Mulitple Linear Regression Modelling of TotChol from NHANES Data

Edward J. Lee

2025-04-05

##Data Cleaning and Preliminary model

```
library(NHANES)  # NHANES dataset
library(dplyr)   # Data wrangling
library(ggplot2) # Visualization
library(car)     # Multicollinearity check (VIF)
library(ggResidpanel) # Advanced diagnostic plots
library(knitr) #for KABLE
library(gridExtra) #for scatterplot matrix

#Turning of scientific notation for interpretability
options(scipen = 999)

#Reading NHANES data package
data("NHANES")
nrow(NHANES) #10,000 observations
```

```
## [1] 10000
```

```
#Applying filters on Age for sample of adults
#Removing 0 entries/misinputs prevalent in BP variables based on problematic
# observations screening
nhanes_filtered <- NHANES %>% filter(Age>20,Height>0,Weight>0,
                                     BPDia1>10,BPDia2>10,BPDia3>10,BPDiaAve>10,
                                     BPSys1>10,BPSys2>10,BPSys3>10,BPSysAve>10,
                                     TotChol>0)

nrow(nhanes_filtered) #5989 observations
```

```
## [1] 5989
```

```
# Remove NA entries and only select columns of interest
nhanes_data <- nhanes_filtered %>%
  dplyr::select(Height, Age, Weight, BPSysAve, BPDiaAve,
                TotChol, SmokeNow, PhysActiveDays) %>%
  na.omit()

n <- nrow(nhanes_data) #1289 observations

#Categorical predictors
nhanes_data$SmokeNow <- as.factor(nhanes_data$SmokeNow)
nhanes_data <- data.frame(nhanes_data)
```

```r
#Preliminary model
model <- lm(TotChol ~ Age + Weight + Height + BPSysAve + BPDiaAve + SmokeNow +
              PhysActiveDays,
            data = nhanes_data)

summary(model)
```

```
##
## Call:
## lm(formula = TotChol ~ Age + Weight + Height + BPSysAve + BPDiaAve +
##     SmokeNow + PhysActiveDays, data = nhanes_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.9808 -0.7045 -0.0874  0.6185  5.2610
##
## Coefficients:
##                  Estimate Std. Error t value          Pr(>|t|)
## (Intercept)     4.2510002  0.6049335   7.027 0.000000000003414 ***
## Age             0.0079190  0.0021614   3.664          0.000259 ***
## Weight          0.0004503  0.0017764   0.253          0.799939
## Height         -0.0084607  0.0035455  -2.386          0.017163 *
## BPSysAve        0.0020588  0.0020282   1.015          0.310258
## BPDiaAve        0.0218750  0.0028088   7.788 0.000000000000014 ***
## SmokeNowYes     0.0327591  0.0631304   0.519          0.603912
## PhysActiveDays -0.0001130  0.0163047  -0.007          0.994471
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.042 on 1281 degrees of freedom
## Multiple R-squared:  0.08085,    Adjusted R-squared:  0.07583
## F-statistic:  16.1 on 7 and 1281 DF,  p-value: < 0.00000000000000022
```

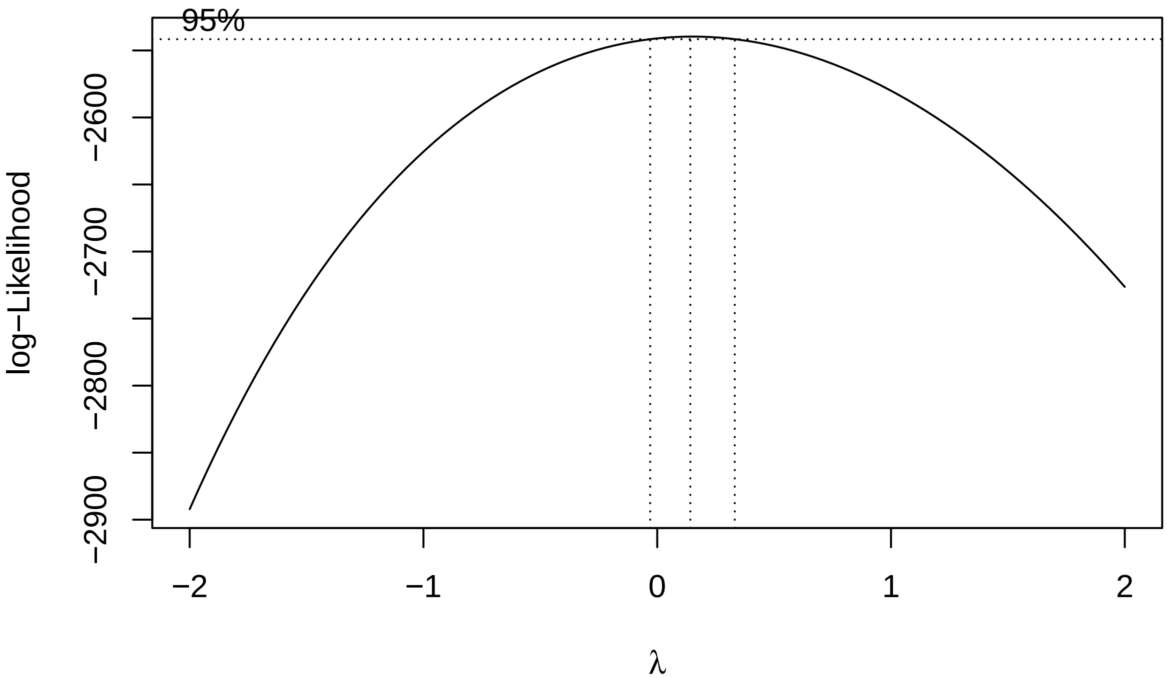
```r
#Multicollinearity Check
vif(model) #No serious multicollinearity, all <5
```

```
##            Age         Weight         Height       BPSysAve       BPDiaAve
##       1.446074       1.322625       1.300929       1.474744       1.242024
##       SmokeNow PhysActiveDays
##       1.159322       1.012534
```

Box-Cox Transformation and Polynomial Term

```r
#POLYNOMIAL "AGE" TERM
poly_data <- nhanes_data %>%
  dplyr::select(Height, Age, Weight, BPSysAve, BPDiaAve,
                TotChol, SmokeNow, PhysActiveDays) %>%
  mutate(Age2 = Age^2)

poly_model <- lm(TotChol~Age+Age2+Height+Weight+BPSysAve+BPDiaAve+
                 SmokeNow+PhysActiveDays, data=poly_data)

#BOX COX TRANSFORMATION
library(MASS)
```

```r
b <- boxcox(poly_model)
```



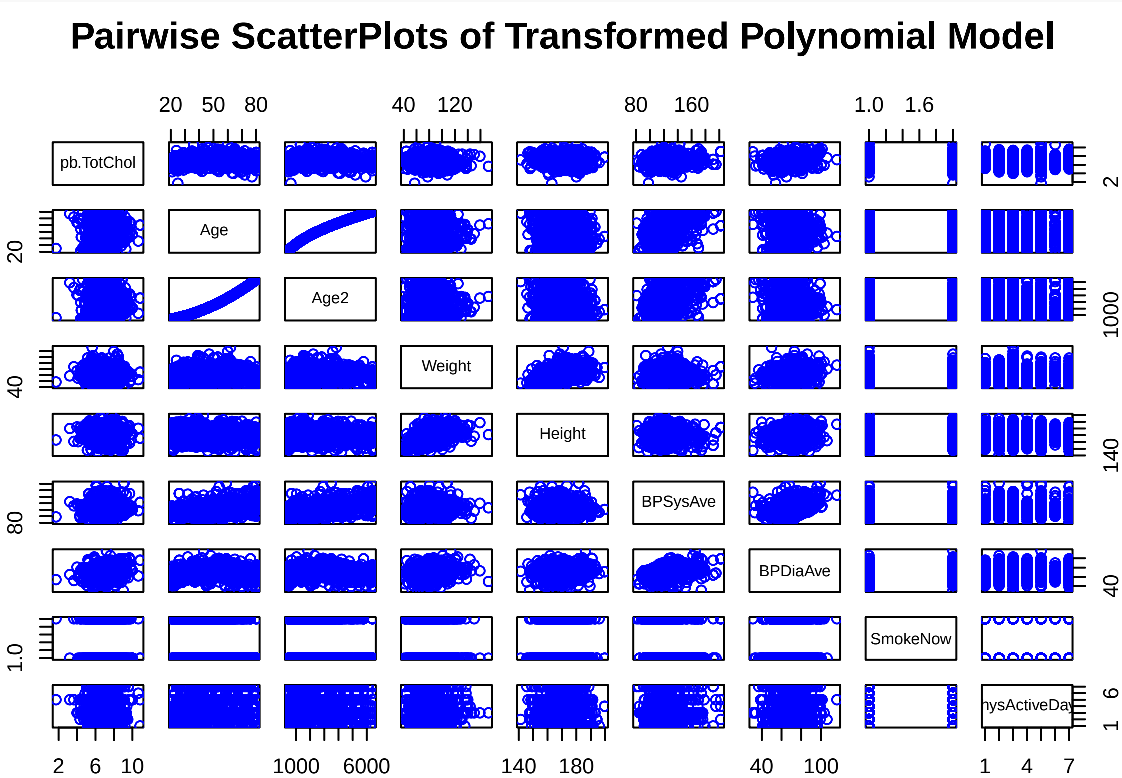

```r
lambda <- b$x[which.max(b$y)]

log_product <- sum(log(poly_data$TotChol))
geo_mean <- exp(log_product/n)

pb.TotChol <- geo_mean^(1-lambda)*(poly_data$TotChol^lambda - 1)/lambda


p.BXCX.frame <- poly_data %>%
  dplyr::select(-TotChol) %>%
  mutate(pb.TotChol = pb.TotChol)

p.BXCX.model <- lm(pb.TotChol ~ Age + Age2 + Weight + Height + BPSysAve +
                          BPDiaAve + SmokeNow + PhysActiveDays,
                        data = p.BXCX.frame)

summary(p.BXCX.model)
```

```
##
## Call:
## lm(formula = pb.TotChol ~ Age + Age2 + Weight + Height + BPSysAve +
##     BPDiaAve + SmokeNow + PhysActiveDays, data = p.BXCX.frame)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.5764 -0.6158 -0.0084  0.6574  3.8416
##
## Coefficients:
##                    Estimate Std. Error t value             Pr(>|t|)
## (Intercept)       4.6682540  0.6104391   7.647   0.0000000000000401 ***
```

```
## Age            0.0993535  0.0112143   8.860 < 0.0000000000000002 ***
## Age2          -0.0009453  0.0001135  -8.331 < 0.0000000000000002 ***
## Weight        -0.0006614  0.0016858  -0.392             0.69487
## Height        -0.0087700  0.0033509  -2.617             0.00897 **
## BPSysAve       0.0057045  0.0019803   2.881             0.00404 **
## BPDiaAve       0.0128515  0.0028416   4.523   0.0000066733181871 ***
## SmokeNowYes    0.0127777  0.0596913   0.214             0.83053
## PhysActiveDays -0.0128377  0.0154387  -0.832             0.40583
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9849 on 1280 degrees of freedom
## Multiple R-squared:  0.1264, Adjusted R-squared:  0.121
## F-statistic: 23.15 on 8 and 1280 DF,  p-value: < 0.000000000000000022
```

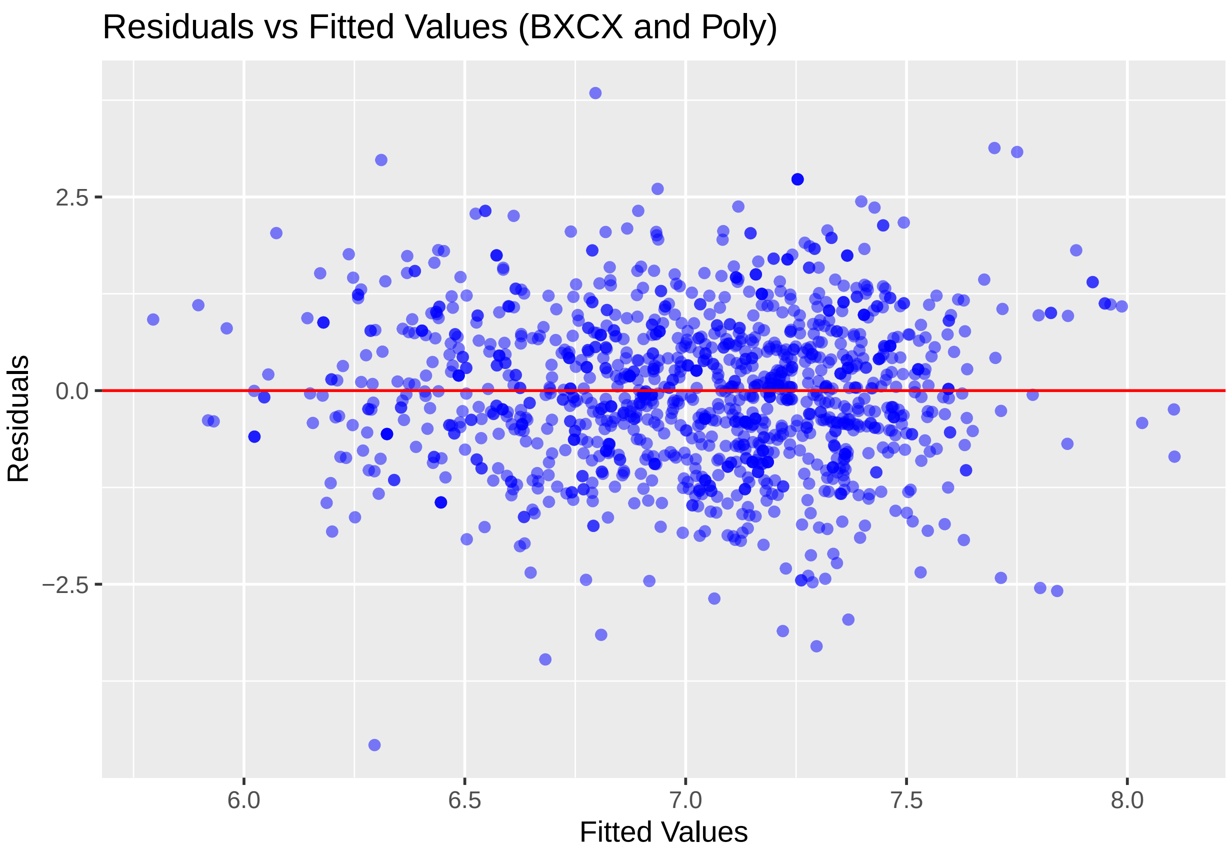
##Transformed Poly Model Diagnostics

```r
#FITTED AND RESIDUAL VALUES FROM TRANSFORMED
fitted <- fitted(p.BXCX.model)
residuals <- resid(p.BXCX.model)

#DATA FRAME FOR PLOTTING
plot_data <- data.frame(fitted = fitted, residuals = residuals)

#PAIRWISE PLOTS OF ORIGINAL MODEL
pairs(~pb.TotChol+Age+Age2+Weight+Height+
        BPSysAve+BPDiaAve+SmokeNow+PhysActiveDays,
      data = p.BXCX.frame,
      main = "Pairwise ScatterPlots of Transformed Polynomial Model",
      col = "blue")
```

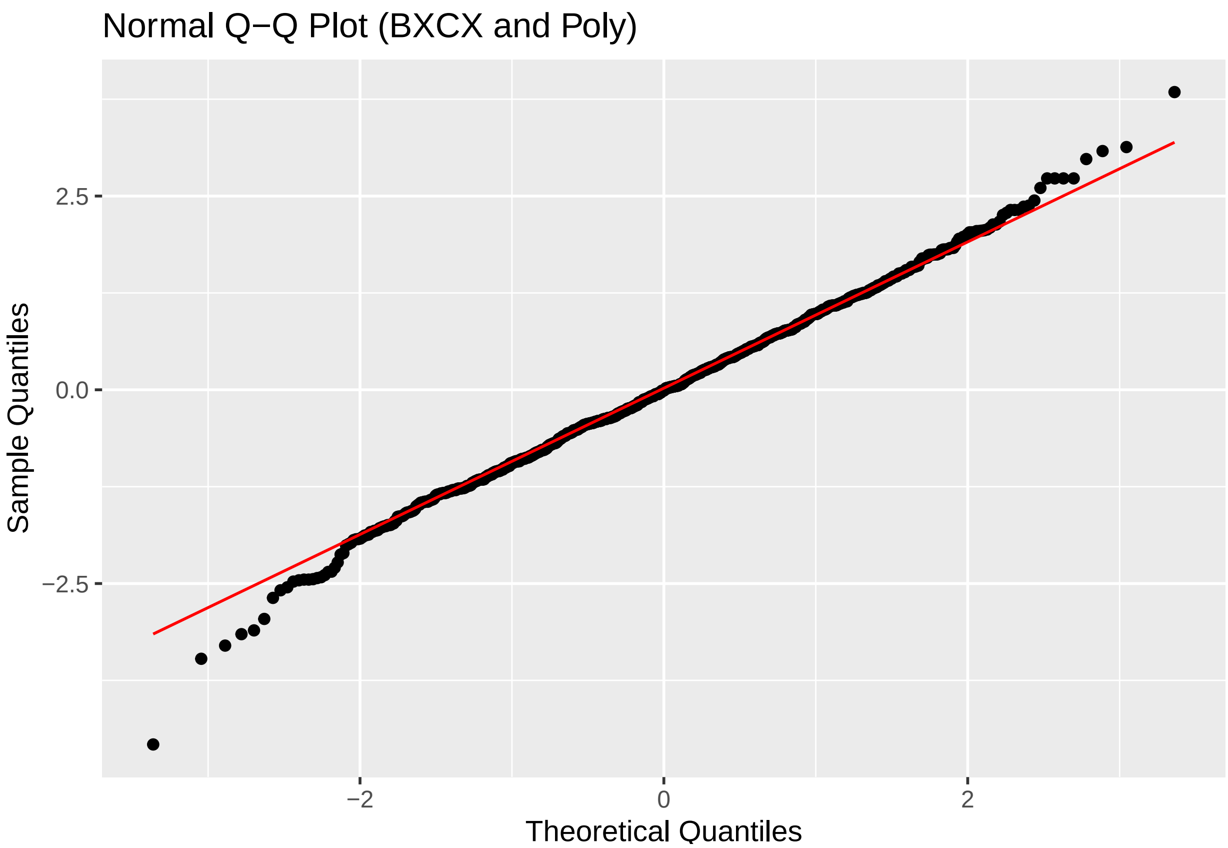


Residual Plots

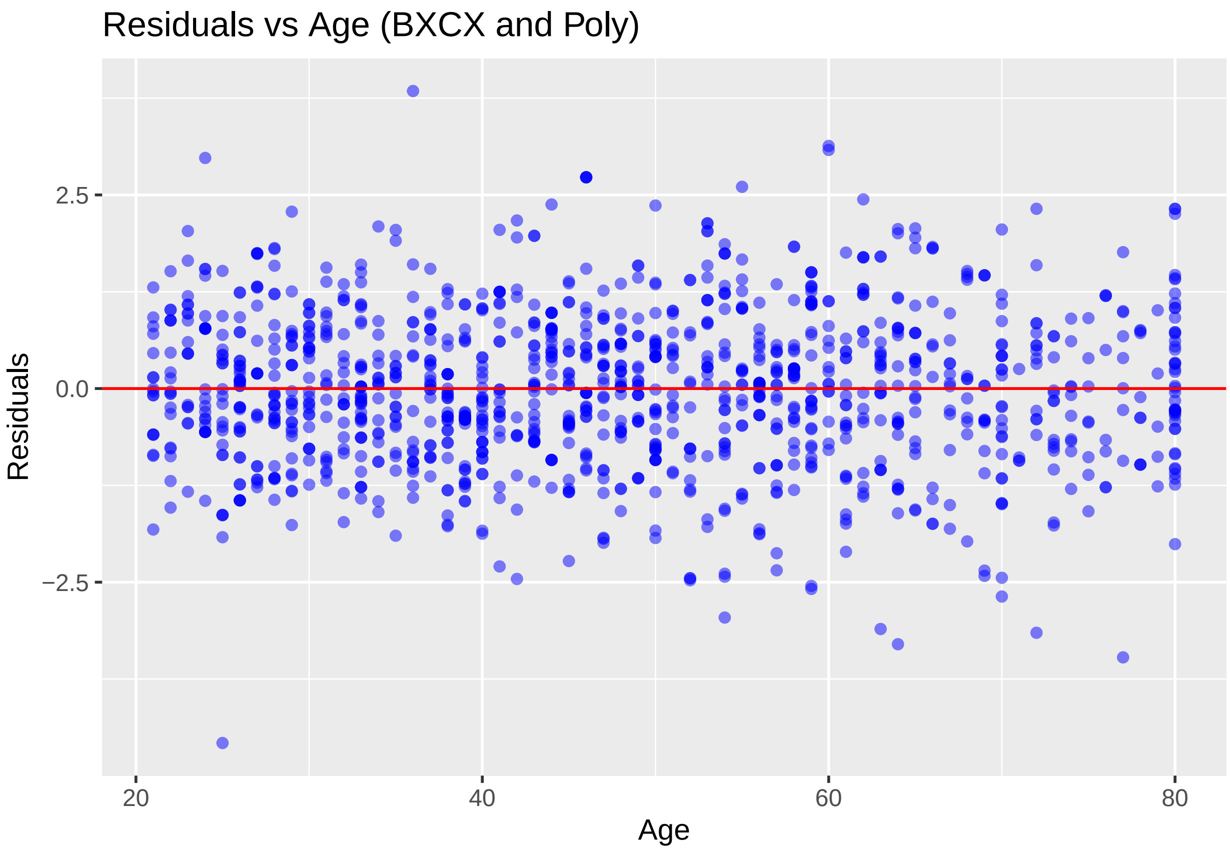
```
#RESIDUALS VS FITTED
res_fitted_plot <- ggplot(data = plot_data,
                          aes(x = fitted, y = residuals)) +
  geom_point(color = "blue", alpha = 0.5) +
  geom_hline(yintercept = 0, color = "red") +
  labs(title = "Residuals vs Fitted Values (BXCX and Poly)",
       x = "Fitted Values", y = "Residuals")

print(res_fitted_plot)
```



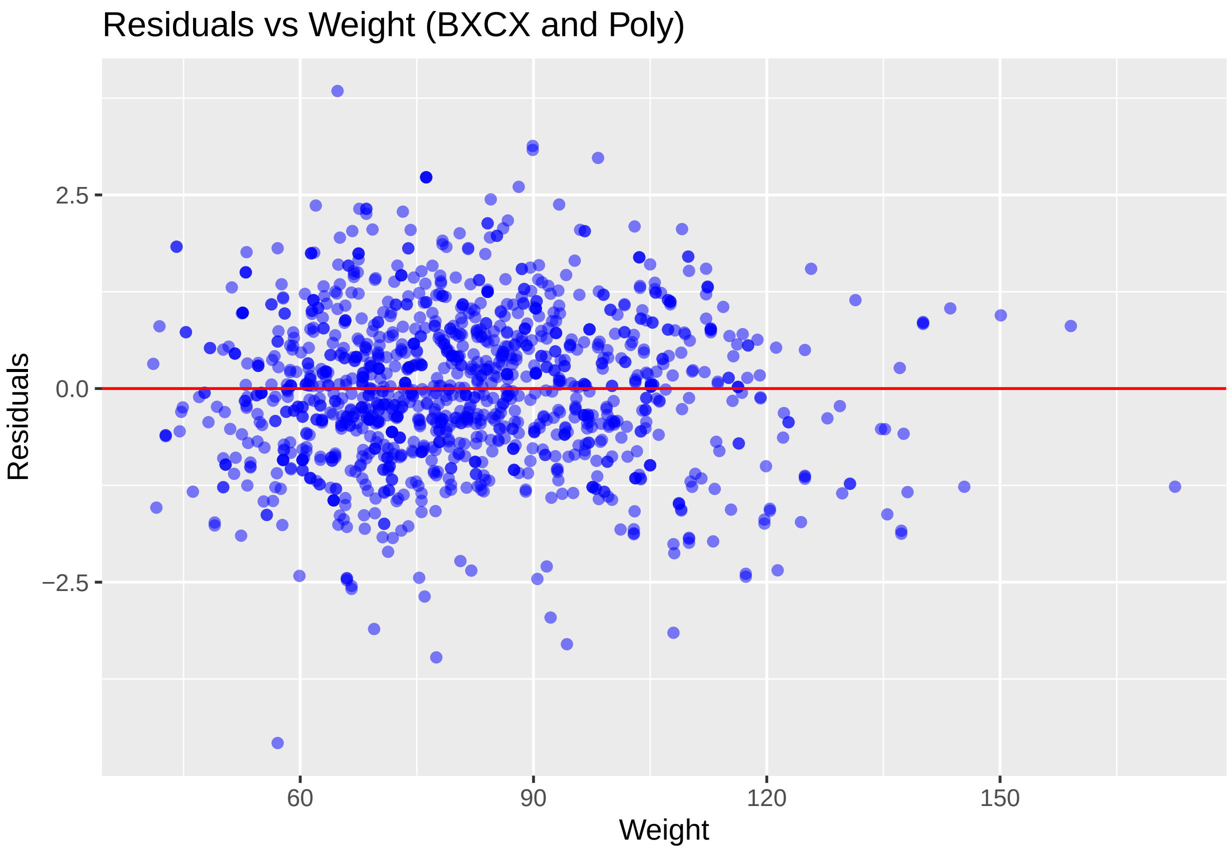
```r
#NORMAL QQ PLOT
qq_plot <- ggplot(data = data.frame(residuals = residuals),
                  aes(sample = residuals)) +
  stat_qq() +
  stat_qq_line(color = "red") +
  labs(title = "Normal Q-Q Plot (BXCX and Poly)",
       x = "Theoretical Quantiles", y = "Sample Quantiles")

print(qq_plot)
```



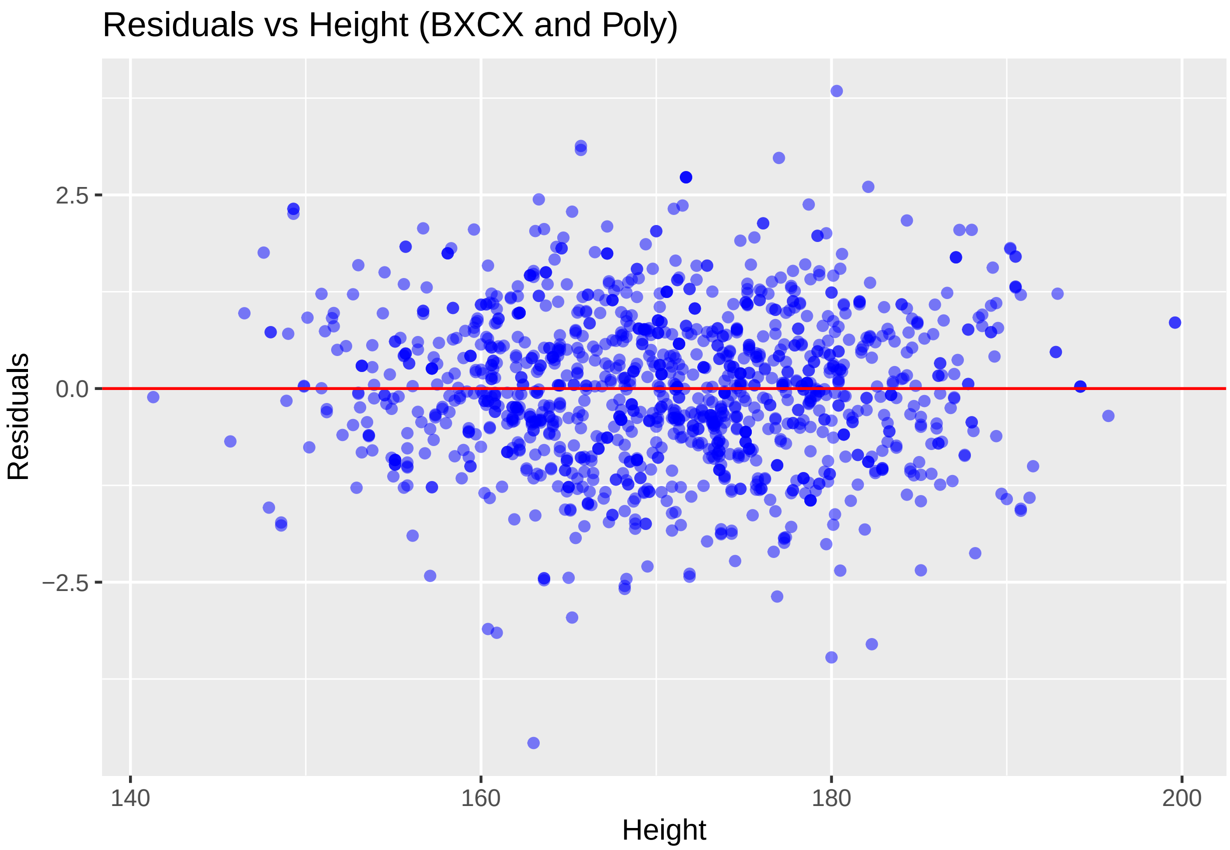

```r
#RESIDUALS VS AGE
res_age_plot <- ggplot(p.BXCX.frame,
                       aes(x = Age, y = residuals)) +
  geom_point(color = "blue", alpha = 0.5) +
  geom_hline(yintercept = 0, color = "red") +
  labs(title = "Residuals vs Age (BXCX and Poly)",
       x = "Age", y = "Residuals")

print(res_age_plot)
```



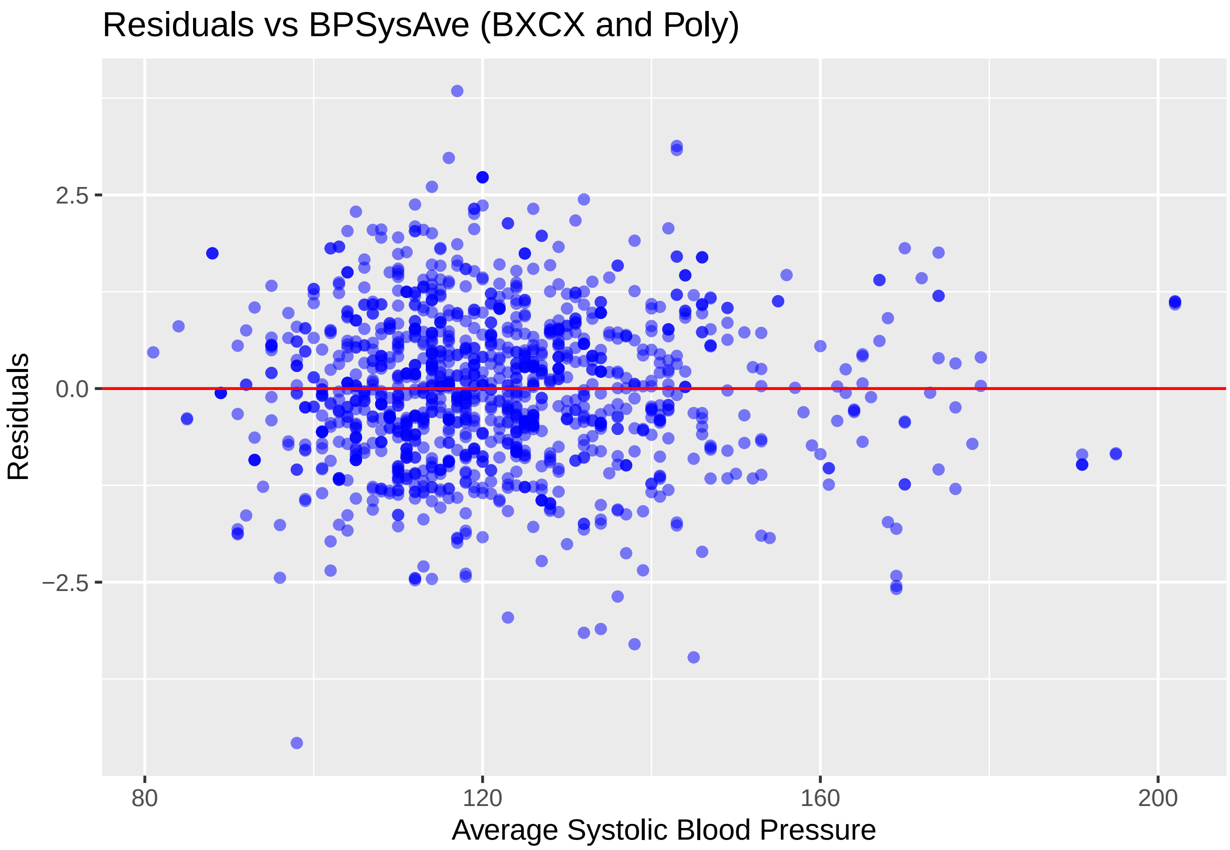

```r
#RESIDUALS VS WEIGHT
res_weight_plot <- ggplot(p.BXCX.frame,
                          aes(x = Weight, y = residuals)) +
  geom_point(color = "blue", alpha = 0.5) +
  geom_hline(yintercept = 0, color = "red") +
  labs(title = "Residuals vs Weight (BXCX and Poly)",
       x = "Weight", y = "Residuals")

print(res_weight_plot)
```



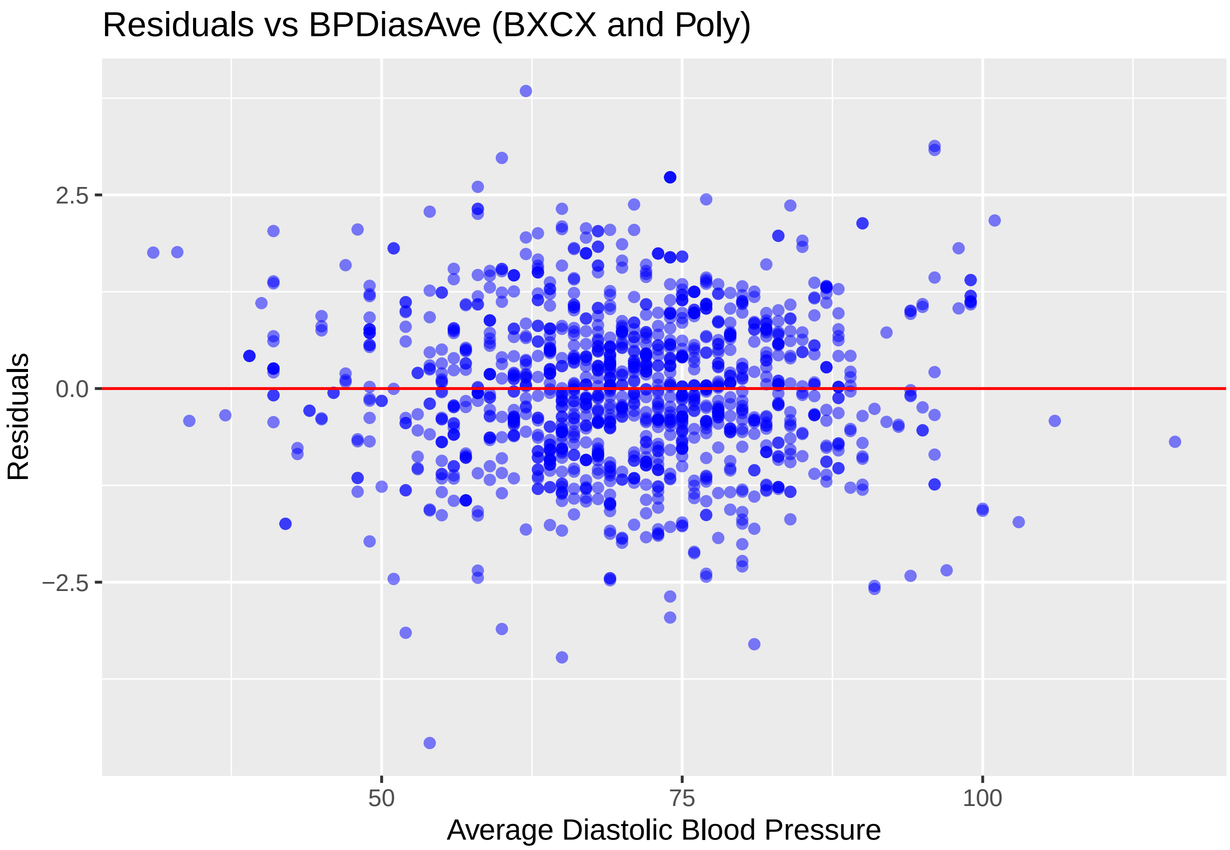

```r
#RESIDUALS VS HEIGHT
res_height_plot <- ggplot(p.BXCX.frame,
                          aes(x = Height, y = residuals)) +
  geom_point(color = "blue", alpha = 0.5) +
  geom_hline(yintercept = 0, color = "red") +
  labs(title = "Residuals vs Height (BXCX and Poly)",
       x = "Height", y = "Residuals")

print(res_height_plot)
```



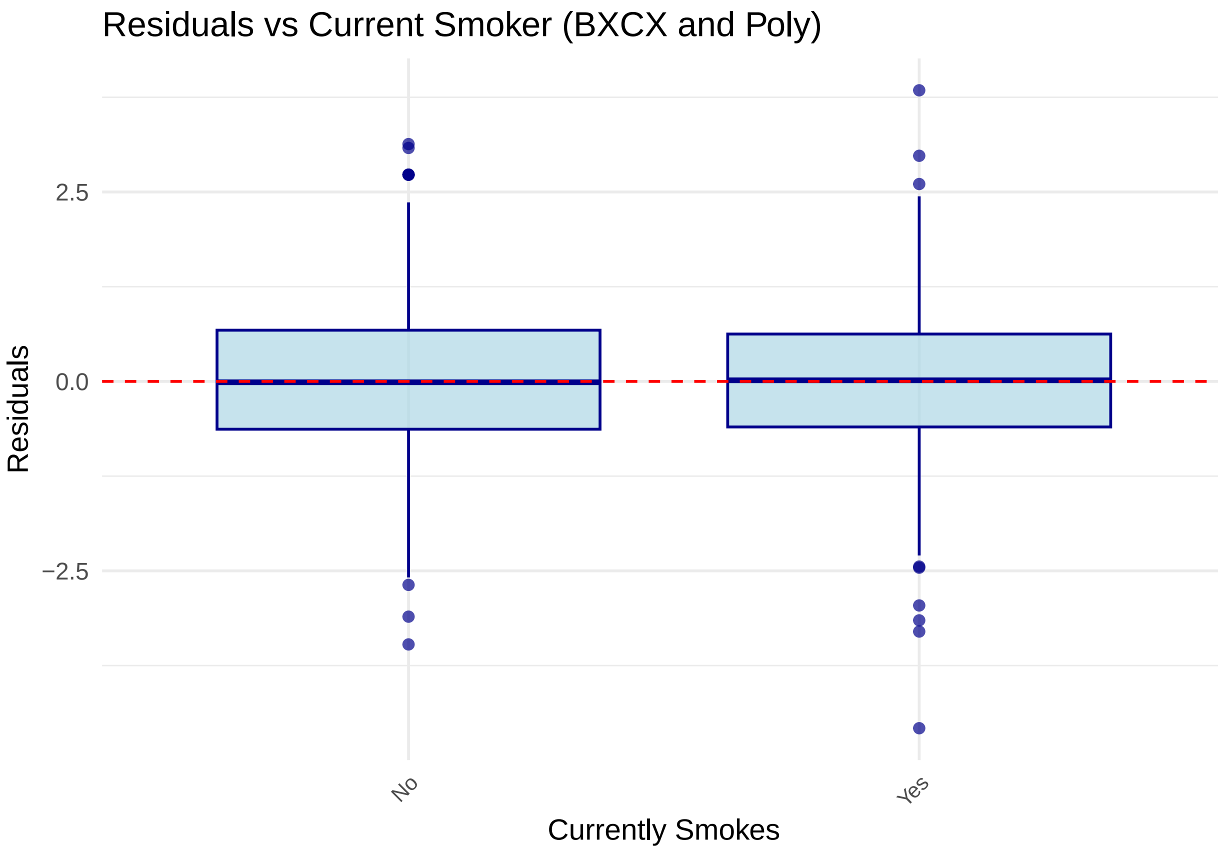

```r
#RESIDUALS VS BPSysAve
res_BPSysAve_plot <- ggplot(p.BXCX.frame,
                            aes(x = BPSysAve, y = residuals)) +
  geom_point(color = "blue", alpha = 0.5) +
  geom_hline(yintercept = 0, color = "red") +
  labs(title = "Residuals vs BPSysAve (BXCX and Poly)",
       x = "Average Systolic Blood Pressure", y = "Residuals")

print(res_BPSysAve_plot)
```



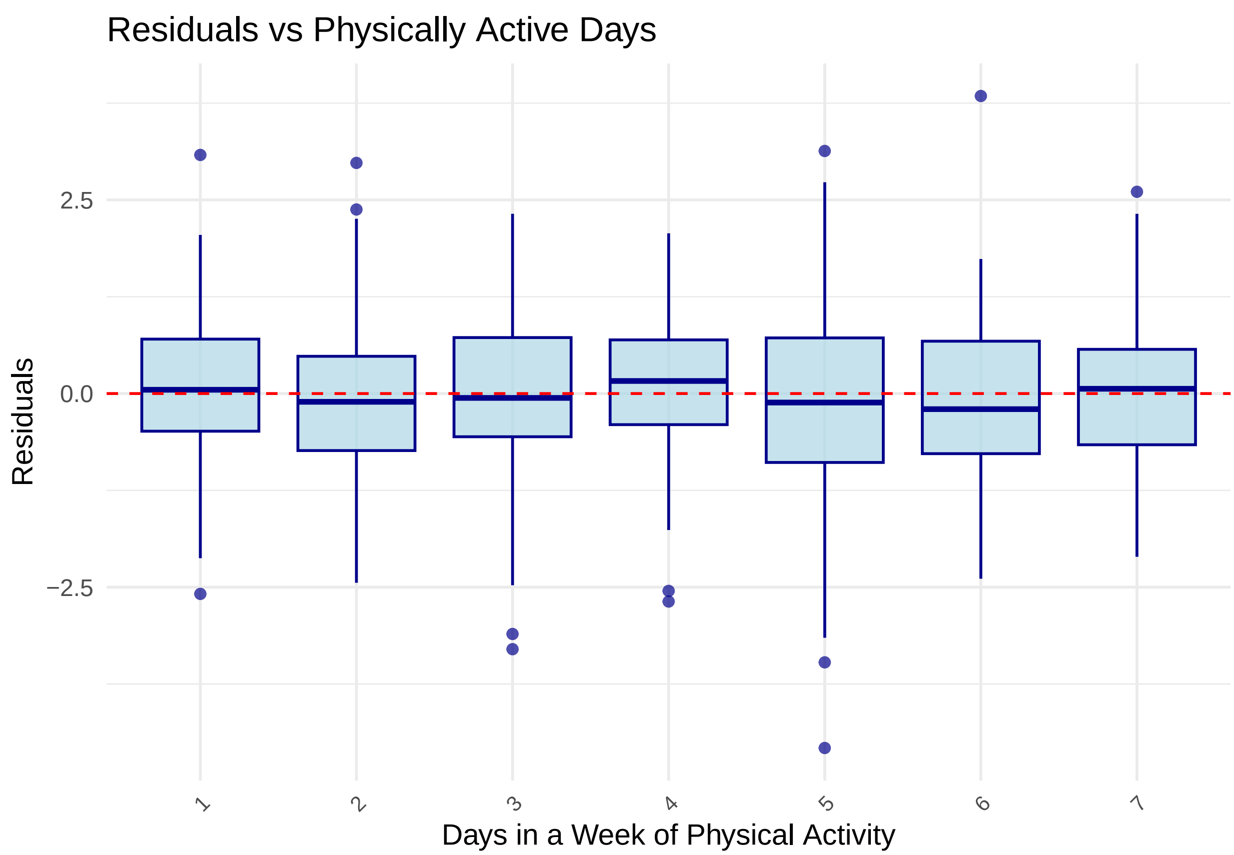

```r
#RESIDUALS VS BPDiaAve
res_BPDiaAve_plot <- ggplot(p.BXCX.frame,
                            aes(x = BPDiaAve, y = residuals)) +
  geom_point(color = "blue", alpha = 0.5) +
  geom_hline(yintercept = 0, color = "red") +
  labs(title = "Residuals vs BPDiasAve (BXCX and Poly)",
       x = "Average Diastolic Blood Pressure", y = "Residuals")

print(res_BPDiaAve_plot)
```



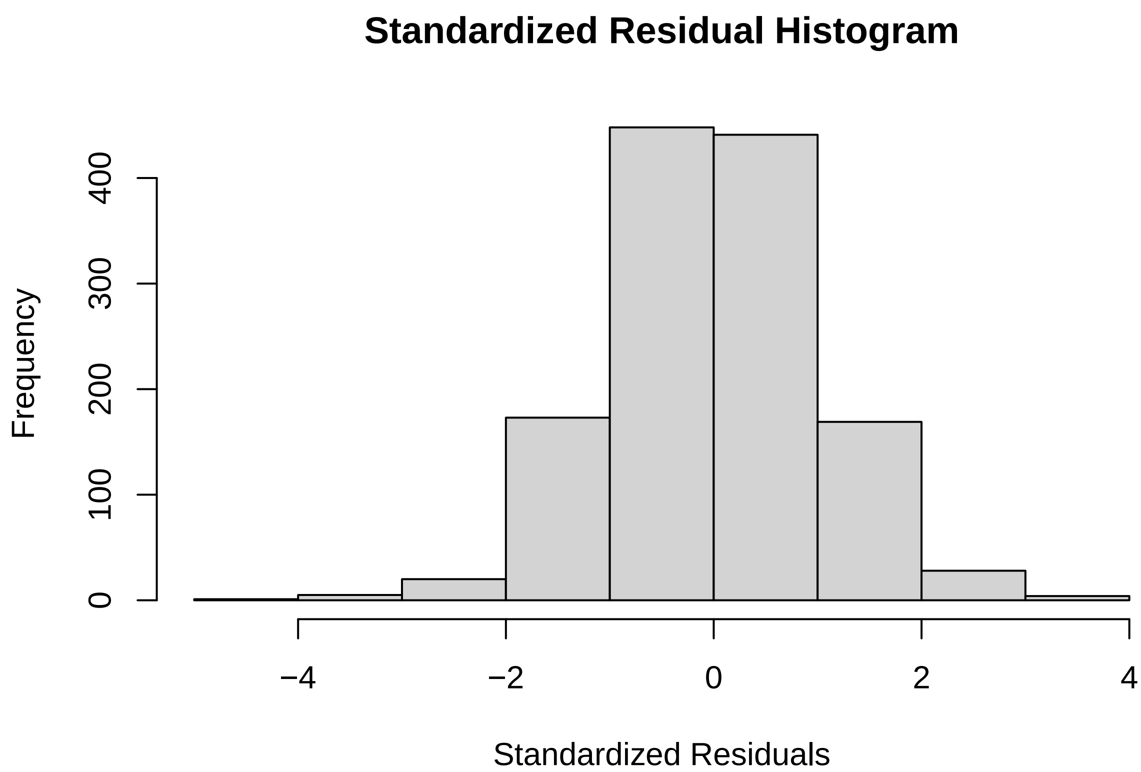

```r
#RESIDUALS VS SmokeNow (BOXPLOT)
res_smoke_plot <- ggplot(
  p.BXCX.frame, aes(x = as.factor(SmokeNow), y = residuals)) +
  geom_boxplot(fill = "lightblue", color = "darkblue", alpha = 0.7) +
  geom_hline(yintercept = 0, color = "red", linetype = "dashed") +
  theme_minimal() +
  ggtitle("Residuals vs Current Smoker (BXCX and Poly)") +
  xlab("Currently Smokes") +
  ylab("Residuals") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1, size = 8))

print(res_smoke_plot)
```




```r
#RESIDUALS VS PhysActiveDays (BOXPLOT)
res_active_plot <- ggplot(
  p.BXCX.frame,
  aes(x = as.factor(PhysActiveDays), y = residuals)) +
  geom_boxplot(fill = "lightblue", color = "darkblue", alpha = 0.7) +
  geom_hline(yintercept = 0, color = "red", linetype = "dashed") +
  theme_minimal() +
  ggtitle("Residuals vs Physically Active Days") +
  xlab("Days in a Week of Physical Activity") +
  ylab("Residuals") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1, size = 8))

print(res_active_plot)
```




```
tr_stres_values <- rstandard(p.BXCX.model)

tr_stres_plot <- hist(tr_stres_values,
                      xlab = "Standardized Residuals",
                      main = "Standardized Residual Histogram")
```



Problematic Observations

```
##LEVERAGE POINTS
leverage <- hatvalues(p.BXCX.model)

p <- 8
high_lev <- 2*(p+1)/n

leverage_points <- p.BXCX.frame[leverage > high_lev,]
leverage_points <- leverage_points %>%
  mutate(row = row.names(leverage_points))

#FINDING OUTLIERS
st.residuals <- rstandard(p.BXCX.model)

outlier_points <- p.BXCX.frame[abs(st.residuals) > 4,]

#COOKS DISTANCE
cooks_value <- cooks.distance(p.BXCX.model)

f_value <- qf(0.50, 8, 1280)

cooks_points <- p.BXCX.frame[cooks_value > f_value,]

#DFFITS
dffits_cutoff <- 2*(sqrt((p+1)/n))

dffits_value = dffits(p.BXCX.model)

dffits_points <- p.BXCX.frame[(abs(dffits_value) > dffits_cutoff),]
dffits_points <- dffits_points %>%
```

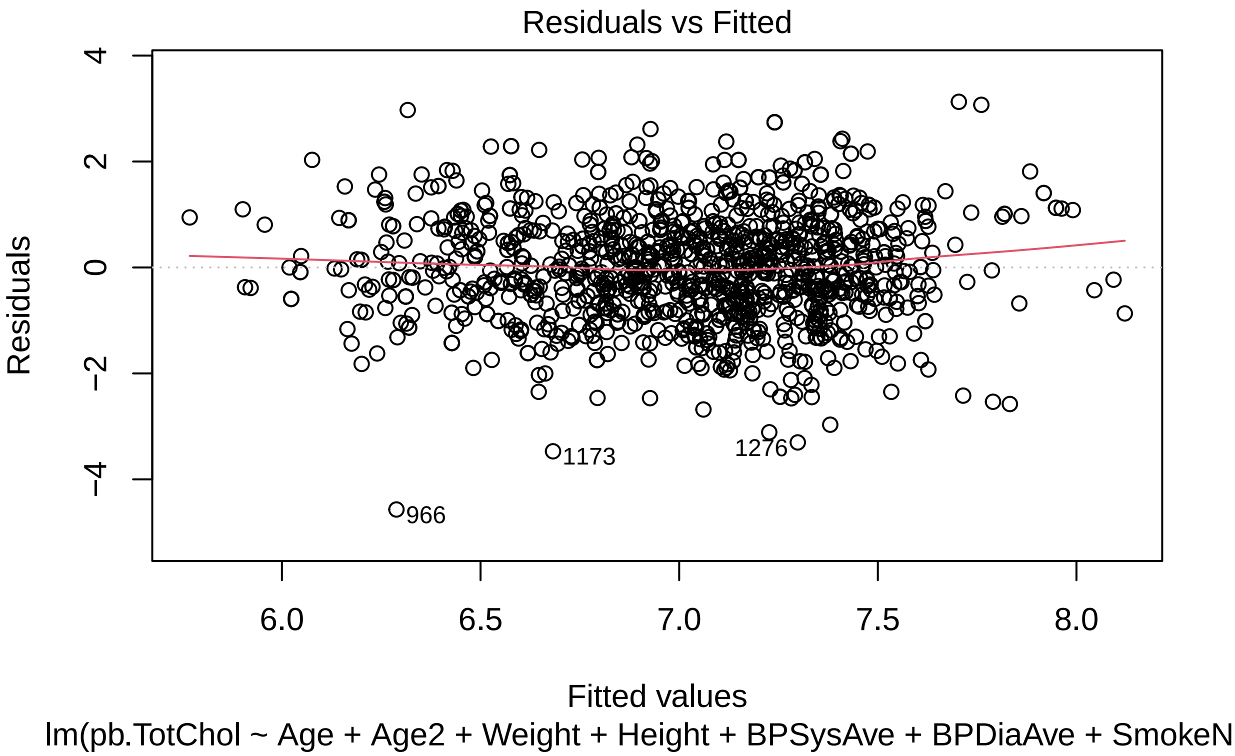
```r
  mutate(row = row.names(dffits_points))

#DFBETAS
dfbetas_cutoff <- 2/sqrt(n)

dfbeta_frame <- as.data.frame(dfbetas(p.BXCX.model))

dfbeta_points <- round(dfbeta_frame[apply(
  abs(dfbeta_frame)>dfbetas_cutoff,1,any),],4)
dfbeta_points <- dfbeta_points %>%
  mutate(row = row.names(dfbeta_points))

#Problematic observations
influential_points <- c(728,823)
p.BXCX.frame[influential_points, ]
```

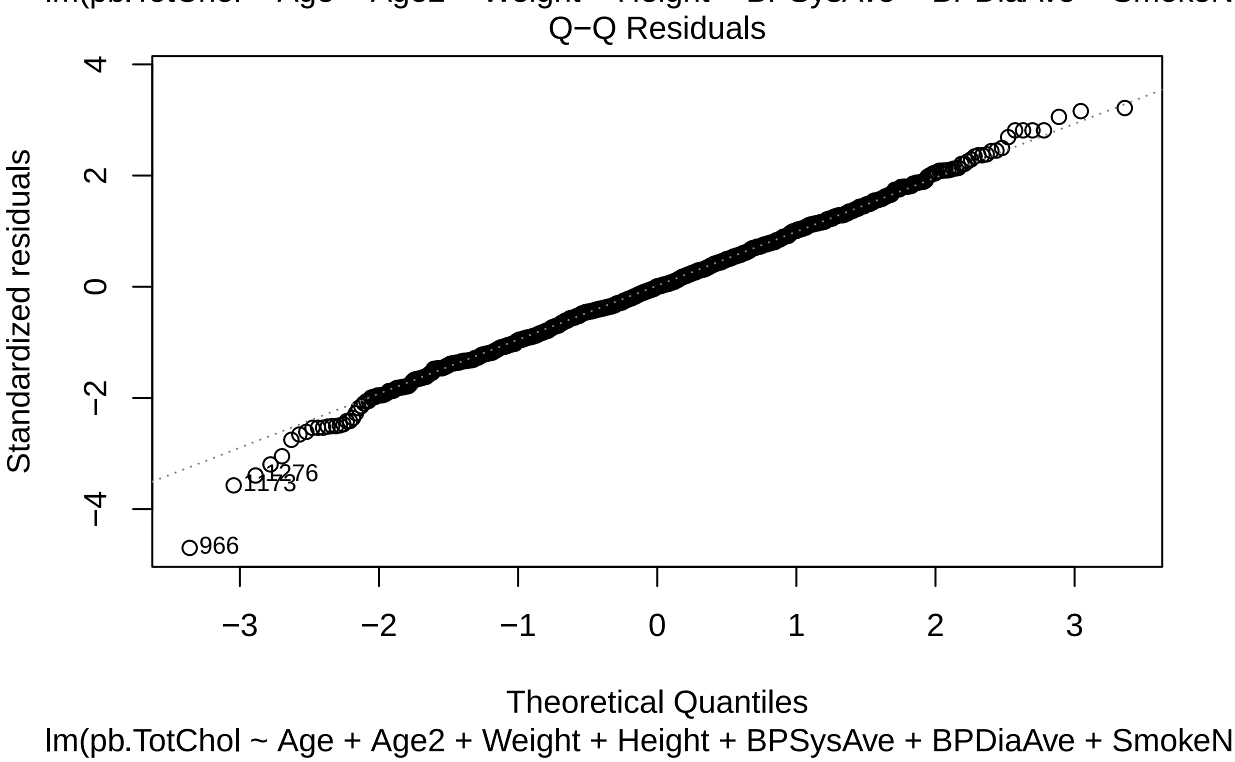
```
##     Height Age Weight BPSysAve BPDiaAve SmokeNow PhysActiveDays Age2 pb.TotChol
## 728  160.9  72  108.0      132       52      Yes              5 5184    3.65555
## 823  180.3  36   64.8      117       62      Yes              6 1296   10.63743
```

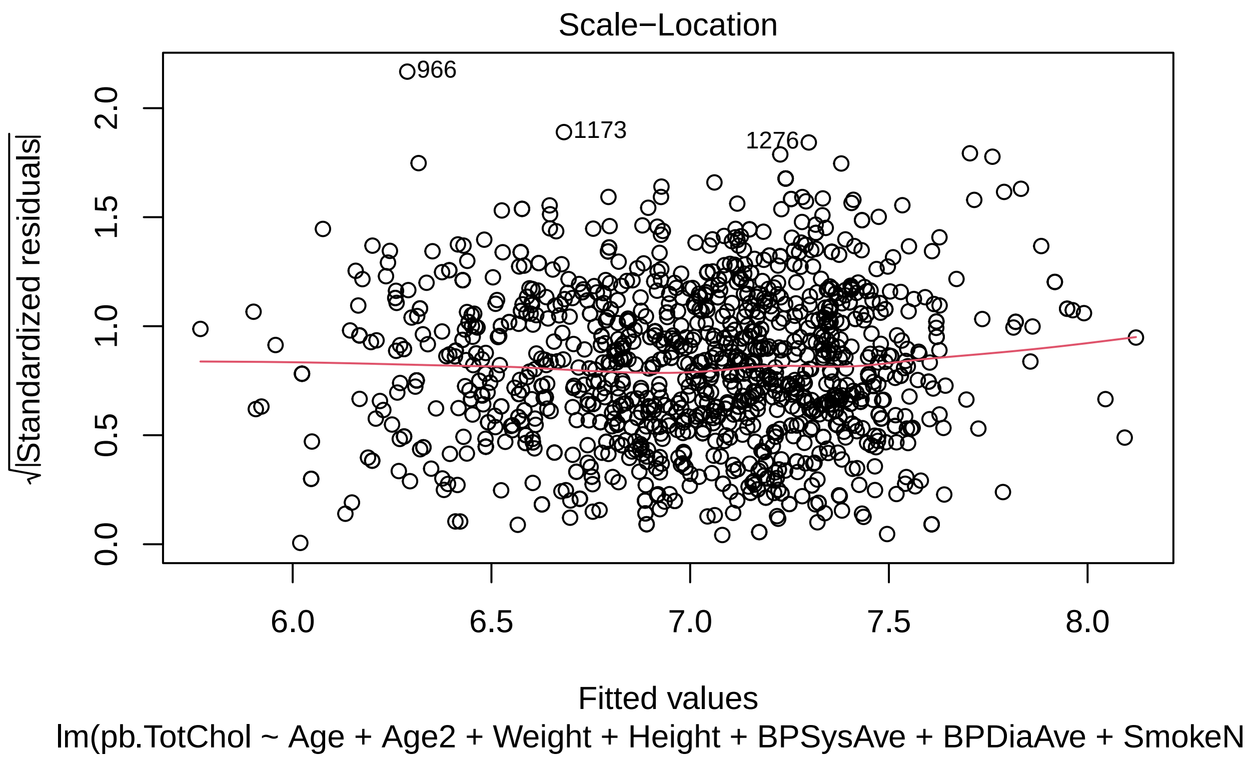
```r
clean.frame <- p.BXCX.frame %>%
dplyr::filter(!row_number() %in% influential_points)

clean_model <- lm(pb.TotChol ~ Age + Age2 + Weight + Height + BPSysAve +
    BPDiaAve + SmokeNow + PhysActiveDays, data = clean.frame)

summary(clean_model)
```

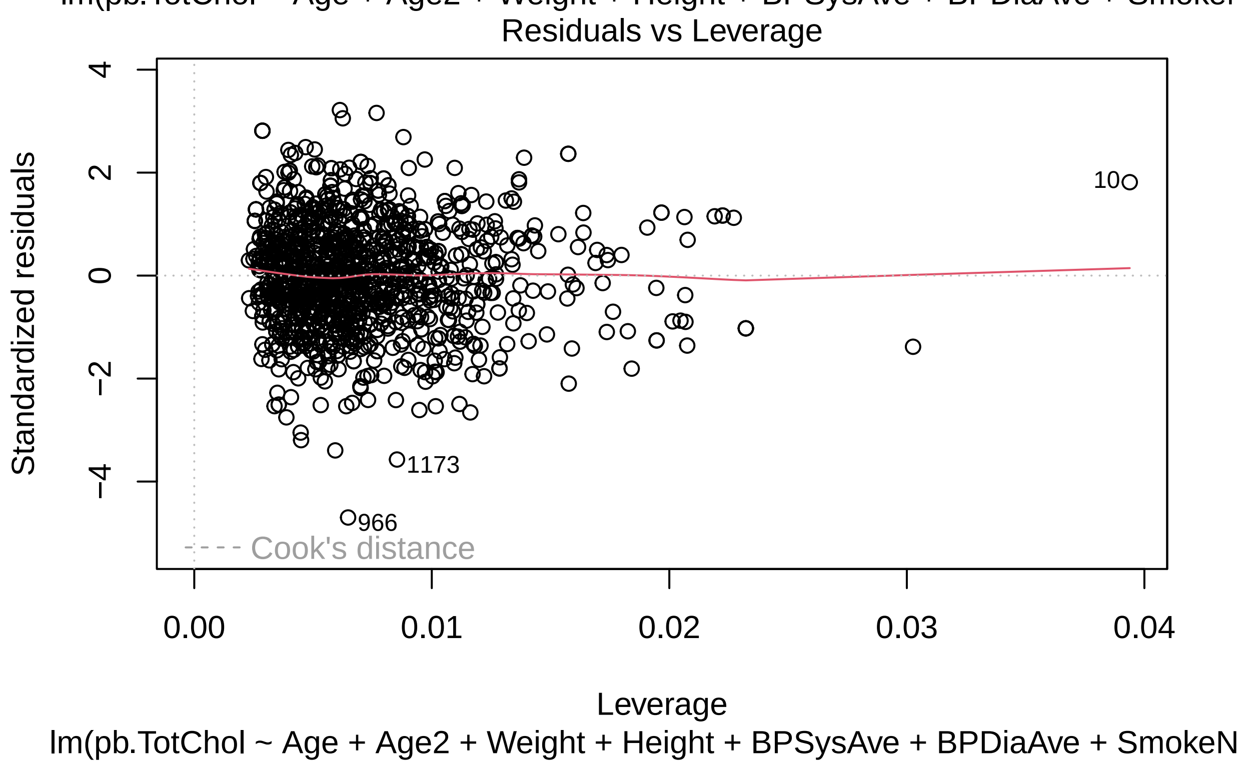
```
##
## Call:
## lm(formula = pb.TotChol ~ Age + Age2 + Weight + Height + BPSysAve +
##     BPDiaAve + SmokeNow + PhysActiveDays, data = clean.frame)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.5691 -0.6185  0.0030  0.6555  3.1272
##
## Coefficients:
##                  Estimate  Std. Error t value             Pr(>|t|)
## (Intercept)     4.82829934  0.60556204   7.973 0.000000000000003403 ***
## Age             0.09839179  0.01111150   8.855 < 0.0000000000000002 ***
## Age2           -0.00093174  0.00011244  -8.287 0.000000000000000291 ***
## Weight         -0.00003742  0.00167529  -0.022              0.98218
## Height         -0.00984142  0.00332638  -2.959              0.00315 **
## BPSysAve        0.00564643  0.00196213   2.878              0.00407 **
## BPDiaAve        0.01274747  0.00281842   4.523 0.000006665615466859 ***
## SmokeNowYes     0.01780620  0.05923101   0.301              0.76375
## PhysActiveDays -0.01413432  0.01530911  -0.923              0.35604
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9757 on 1278 degrees of freedom
## Multiple R-squared:  0.129,  Adjusted R-squared:  0.1236
## F-statistic: 23.66 on 8 and 1278 DF,  p-value: < 0.00000000000000022
```

```r
plots <- plot(clean_model)
```







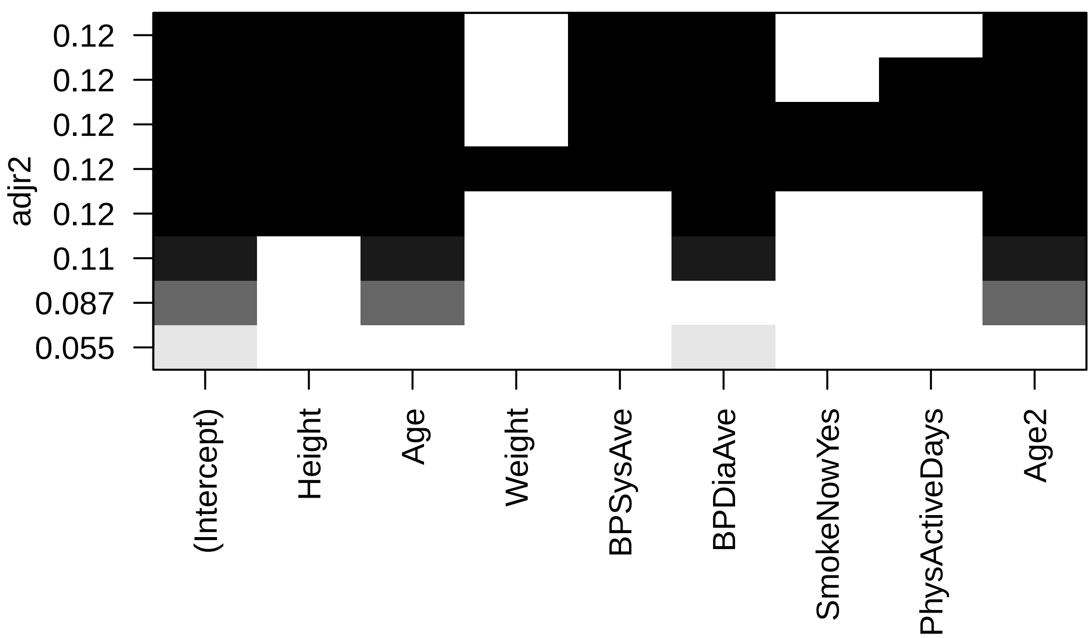


Variable Selection

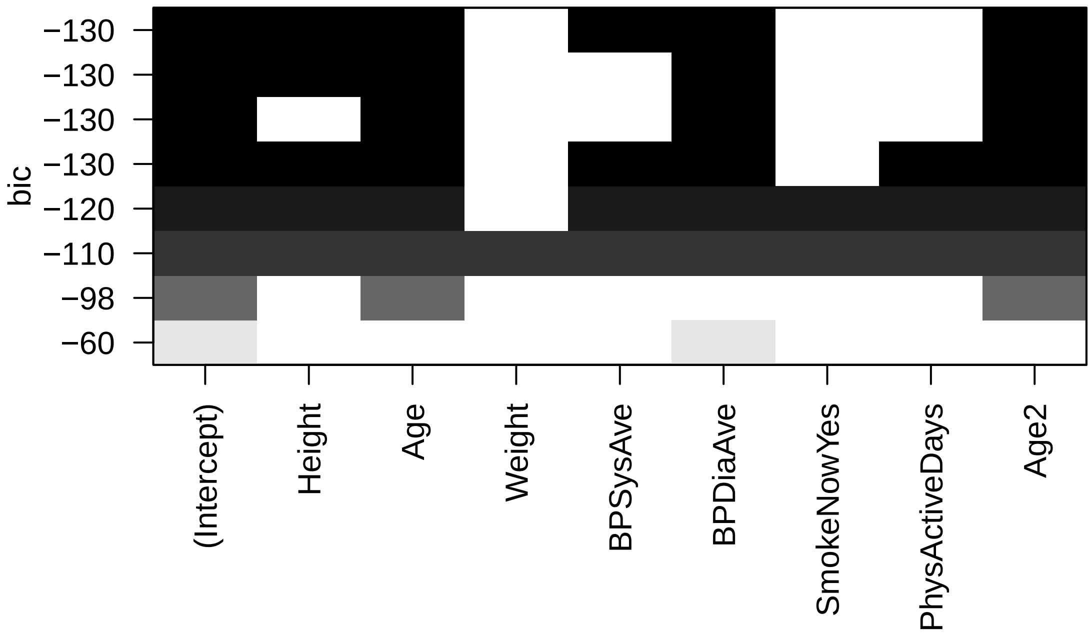
```
library(leaps)

best_subset <- regsubsets(pb.TotChol~., data=clean.frame,nvmax=8,                    nbest=1,really.big=T

summary(best_subset)
```

```
## Subset selection object
## Call: regsubsets.formula(pb.TotChol ~ ., data = clean.frame, nvmax = 8,
##     nbest = 1, really.big = TRUE, method = "exhaustive")
## 8 Variables  (and intercept)
##                 Forced in Forced out
## Height              FALSE      FALSE
## Age                 FALSE      FALSE
## Weight              FALSE      FALSE
## BPSysAve            FALSE      FALSE
## BPDiaAve            FALSE      FALSE
## SmokeNowYes         FALSE      FALSE
## PhysActiveDays      FALSE      FALSE
## Age2                FALSE      FALSE
## 1 subsets of each size up to 8
## Selection Algorithm: exhaustive
##          Height Age Weight BPSysAve BPDiaAve SmokeNowYes PhysActiveDays Age2
## 1  ( 1 ) " "    " " " "    " "      "*"      " "         " "            " "
## 2  ( 1 ) " "    "*" " "    " "      " "      " "         " "            "*"
## 3  ( 1 ) " "    "*" " "    " "      "*"      " "         " "            "*"
## 4  ( 1 ) "*"    "*" " "    " "      "*"      " "         " "            "*"
## 5  ( 1 ) "*"    "*" " "    "*"      "*"      " "         " "            "*"
## 6  ( 1 ) "*"    "*" " "    "*"      "*"      " "         "*"            "*"
## 7  ( 1 ) "*"    "*" " "    "*"      "*"      "*"         "*"            "*"
## 8  ( 1 ) "*"    "*" "*"    "*"      "*"      "*"         "*"            "*"
```

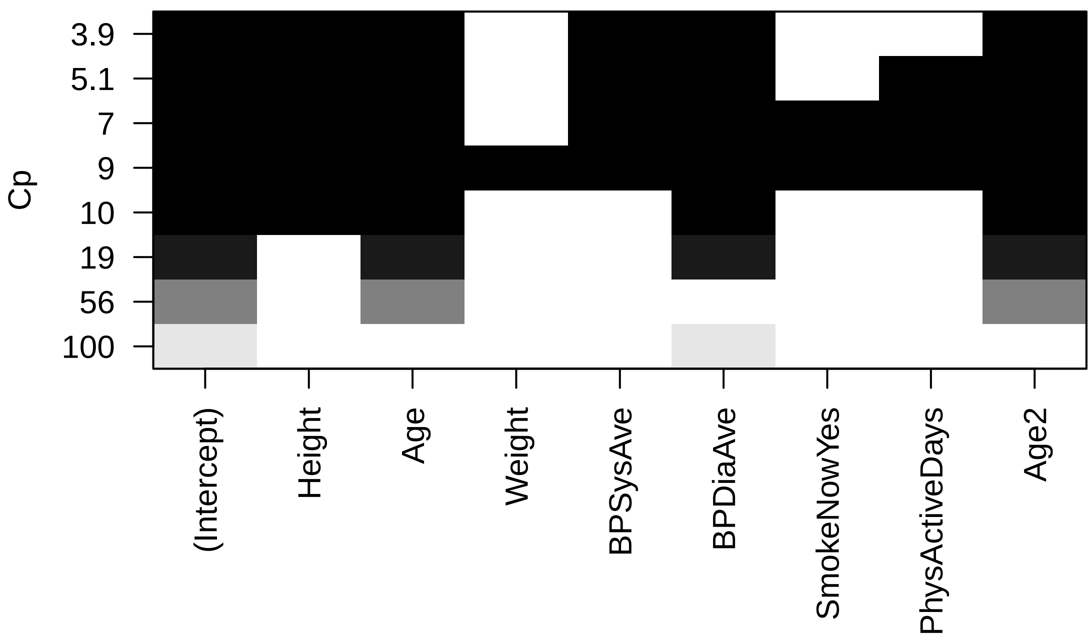
```
plot(best_subset,scale='adjr2')
```




```
plot(best_subset,scale='bic');
```




```r
plot(best_subset,scale='Cp')
```




```r
AIC <- step(clean_model, direction="both")
```

```
## Start:  AIC=-54.33
## pb.TotChol ~ Age + Age2 + Weight + Height + BPSysAve + BPDiaAve +
##     SmokeNow + PhysActiveDays
##
##                  Df Sum of Sq    RSS     AIC
## - Weight          1     0.000 1216.7 -56.334
## - SmokeNow        1     0.086 1216.8 -56.244
## - PhysActiveDays  1     0.811 1217.5 -55.476
## <none>                        1216.7 -54.335
## - BPSysAve        1     7.884 1224.5 -48.022
```

```
## - Height          1     8.333 1225.0 -47.550
## - BPDiaAve        1    19.475 1236.1 -35.897
## - Age2            1    65.377 1282.0  11.028
## - Age             1    74.647 1291.3  20.300
##
## Step:  AIC=-56.33
## pb.TotChol ~ Age + Age2 + Height + BPSysAve + BPDiaAve + SmokeNow +
##     PhysActiveDays
##
##                  Df Sum of Sq    RSS     AIC
## - SmokeNow        1     0.088 1216.8 -58.241
## - PhysActiveDays  1     0.811 1217.5 -57.476
## <none>                        1216.7 -56.334
## + Weight          1     0.000 1216.7 -54.335
## - BPSysAve        1     7.936 1224.6 -49.967
## - Height          1    10.536 1227.2 -47.237
## - BPDiaAve        1    19.546 1236.2 -37.823
## - Age2            1    65.904 1282.6   9.557
## - Age             1    75.216 1291.9  18.868
##
## Step:  AIC=-58.24
## pb.TotChol ~ Age + Age2 + Height + BPSysAve + BPDiaAve + PhysActiveDays
##
##                  Df Sum of Sq    RSS     AIC
## - PhysActiveDays  1     0.811 1217.6 -59.384
## <none>                        1216.8 -58.241
## + SmokeNow        1     0.088 1216.7 -56.334
## + Weight          1     0.003 1216.8 -56.244
## - BPSysAve        1     8.071 1224.8 -51.731
## - Height          1    10.615 1227.4 -49.062
## - BPDiaAve        1    19.459 1236.2 -39.821
## - Age2            1    66.037 1282.8   7.779
## - Age             1    75.131 1291.9  16.872
##
## Step:  AIC=-59.38
## pb.TotChol ~ Age + Age2 + Height + BPSysAve + BPDiaAve
##
##                  Df Sum of Sq    RSS     AIC
## <none>                        1217.6 -59.384
## + PhysActiveDays  1     0.811 1216.8 -58.241
## + SmokeNow        1     0.088 1217.5 -57.476
## + Weight          1     0.000 1217.6 -57.384
## - BPSysAve        1     7.982 1225.5 -52.974
## - Height          1    10.444 1228.0 -50.391
## - BPDiaAve        1    19.562 1237.1 -40.870
## - Age2            1    65.411 1283.0   5.965
## - Age             1    74.398 1292.0  14.949
```

```r
summary(AIC)
```

```
##
## Call:
## lm(formula = pb.TotChol ~ Age + Age2 + Height + BPSysAve + BPDiaAve,
##     data = clean.frame)
##
```

```
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.5880 -0.6170 -0.0140  0.6438  3.1057
##
## Coefficients:
##               Estimate Std. Error t value            Pr(>|t|)
## (Intercept)  4.8098341  0.5741846   8.377 < 0.0000000000000002 ***
## Age          0.0974977  0.0110200   8.847 < 0.0000000000000002 ***
## Age2        -0.0009263  0.0001117  -8.296  0.00000000000000027 ***
## Height      -0.0098211  0.0029628  -3.315            0.000943 ***
## BPSysAve     0.0056469  0.0019487   2.898            0.003821 **
## BPDiaAve     0.0127101  0.0028016   4.537  0.00000624991387098 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9749 on 1281 degrees of freedom
## Multiple R-squared:  0.1284, Adjusted R-squared:  0.125
## F-statistic: 37.73 on 5 and 1281 DF,  p-value: < 0.00000000000000022
```

FINAL MODEL

```r
final_model <- lm(pb.TotChol ~ Age+Age2+Height+BPSysAve+BPDiaAve,
                    data=clean.frame)

summary(final_model)
```

```
##
## Call:
## lm(formula = pb.TotChol ~ Age + Age2 + Height + BPSysAve + BPDiaAve,
##     data = clean.frame)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.5880 -0.6170 -0.0140  0.6438  3.1057
##
## Coefficients:
##               Estimate Std. Error t value            Pr(>|t|)
## (Intercept)  4.8098341  0.5741846   8.377 < 0.0000000000000002 ***
## Age          0.0974977  0.0110200   8.847 < 0.0000000000000002 ***
## Age2        -0.0009263  0.0001117  -8.296  0.00000000000000027 ***
## Height      -0.0098211  0.0029628  -3.315            0.000943 ***
## BPSysAve     0.0056469  0.0019487   2.898            0.003821 **
## BPDiaAve     0.0127101  0.0028016   4.537  0.00000624991387098 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9749 on 1281 degrees of freedom
## Multiple R-squared:  0.1284, Adjusted R-squared:  0.125
## F-statistic: 37.73 on 5 and 1281 DF,  p-value: < 0.00000000000000022
```

```r
confint(final_model)
```

```
##                    2.5 %       97.5 %
## (Intercept)  3.683388750  5.9362795086
## Age          0.075878426  0.1191170717
```

```
## Age2        -0.001145301 -0.0007072109
## Height      -0.015633630 -0.0040086511
## BPSysAve     0.001823995  0.0094698317
## BPDiaAve     0.007213819  0.0182063184
```

Prediction Accuracy and Model Validation

```r
#PREDICTION ACCURACY
set.seed(123)
train_index <- sample(1:nrow(clean.frame), 0.7 * nrow(clean.frame))
train_data <- clean.frame[train_index, ]
test_data <- clean.frame[-train_index, ]

validation_model <- lm(pb.TotChol ~ Age + Age2 + Height + BPSysAve + BPDiaAve,
                        data = train_data)
predictions <- predict(validation_model, newdata = test_data)

# Compare predictions to actual
mean((predictions - test_data$pb.TotChol)^2)  # MSE
```

```
## [1] 0.9542581
```

```r
sqrt(mean((predictions - test_data$pb.TotChol)^2))  # RMSE
```

```
## [1] 0.9768613
```

```r
#K-Fold (10-Fold) MODEL VALIDATION
library(caret)

#FINAL_MODEL VALIDATION
train_control <- trainControl(method = "cv", number = 10)
cv_model <- train(
  pb.TotChol ~ Age+Age2+Height+BPSysAve+BPDiaAve,
  data = clean.frame,
  method = "lm",
  trControl = train_control
)

print(cv_model)
```

```
## Linear Regression
##
## 1287 samples
##    5 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 1158, 1159, 1158, 1158, 1158, 1159, ...
## Resampling results:
##
##   RMSE       Rsquared   MAE
##   0.9751559  0.1373103  0.7694394
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

```r
#FULL_MODEL VALIDATION
train_control_full <- trainControl(method = "cv", number = 10)
cv_full_model <- train(
  pb.TotChol ~ .,
  data = clean.frame,
  method = "lm",
  trControl = train_control_full
)

print(cv_full_model)
```

```
## Linear Regression
##
## 1287 samples
##    8 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 1159, 1160, 1160, 1157, 1159, 1158, ...
## Resampling results:
##
##   RMSE       Rsquared   MAE
##   0.9751863  0.1249023  0.7700059
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

```r
#NULL_MODEL VALIDATION

train_control_null <- trainControl(method = "cv", number = 10)


cv_null_model <- train(
  pb.TotChol ~ SmokeNow,
  data = clean.frame,
  method = "lm",
  trControl = train_control_null
)

print(cv_null_model)
```

```
## Linear Regression
##
## 1287 samples
##    1 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 1159, 1159, 1159, 1159, 1158, 1158, ...
## Resampling results:
##
##   RMSE      Rsquared    MAE
##   1.040307  0.01430996  0.822368
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

```r
#ORIGINAL MODEL VALIDATION
train_original <- trainControl(method = "cv", number = 10)
cv_original_model <- train(
  TotChol ~ Age+Height+BPSysAve+BPDiaAve,
  data = nhanes_data,
  method = "lm",
  trControl = train_original
)

print(cv_original_model)
```

```
## Linear Regression
##
## 1289 samples
##    4 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 1161, 1159, 1162, 1160, 1160, 1159, ...
## Resampling results:
##
##   RMSE      Rsquared    MAE
##   1.040968  0.08495365  0.811146
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

```r
#ORIGINAL FULL MODEL VALIDATION
train_full.og <- trainControl(method = "cv", number = 10)
cv_full.og_model <- train(
  TotChol ~ .,
  data = nhanes_data,
  method = "lm",
  trControl = train_full.og
)

print(cv_full.og_model)
```

```
## Linear Regression
##
## 1289 samples
##    7 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 1159, 1160, 1161, 1160, 1160, 1160, ...
## Resampling results:
##
##   RMSE      Rsquared    MAE
##   1.042695  0.07761385  0.8123572
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

```r
library(glmnet)

lasso_model <- train(
```

```r
  pb.TotChol ~ .,
  data = clean.frame,
  method = "glmnet",
  trControl = train_control,
  tuneGrid = expand.grid(
    alpha = 1,          # Lasso
    lambda = 10^seq(-4, 1, length = 100)  # Lambda grid
  )
)
```

```
## Warning in nominalTrainWorkflow(x = x, y = y, wts = weights, info = trainInfo,
## : There were missing values in resampled performance measures.
```

```r
print(lasso_model)
```

```
## glmnet
##
## 1287 samples
##    8 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 1157, 1159, 1159, 1159, 1157, 1157, ...
## Resampling results across tuning parameters:
##
##   lambda        RMSE       Rsquared    MAE
##    0.0001000000  0.9758988  0.12666823  0.7709795
##    0.0001123324  0.9758988  0.12666823  0.7709795
##    0.0001261857  0.9758999  0.12666666  0.7709812
##    0.0001417474  0.9759009  0.12666285  0.7709869
##    0.0001592283  0.9758998  0.12666193  0.7709929
##    0.0001788650  0.9758984  0.12666299  0.7709989
##    0.0002009233  0.9758963  0.12666452  0.7710054
##    0.0002257020  0.9758946  0.12666494  0.7710128
##    0.0002535364  0.9758924  0.12666557  0.7710213
##    0.0002848036  0.9758909  0.12666603  0.7710309
##    0.0003199267  0.9758887  0.12666674  0.7710411
##    0.0003593814  0.9758860  0.12666812  0.7710534
##    0.0004037017  0.9758836  0.12666920  0.7710672
##    0.0004534879  0.9758815  0.12666968  0.7710833
##    0.0005094138  0.9758796  0.12666983  0.7711006
##    0.0005722368  0.9758785  0.12666887  0.7711209
##    0.0006428073  0.9758784  0.12666621  0.7711437
##    0.0007220809  0.9758798  0.12666199  0.7711696
##    0.0008111308  0.9758819  0.12665717  0.7712009
##    0.0009111628  0.9758863  0.12665024  0.7712405
##    0.0010235310  0.9758936  0.12663926  0.7712890
##    0.0011497570  0.9759057  0.12662259  0.7713466
##    0.0012915497  0.9759227  0.12660071  0.7714115
##    0.0014508288  0.9759474  0.12656973  0.7714889
##    0.0016297508  0.9759794  0.12653072  0.7715771
##    0.0018307383  0.9760233  0.12647811  0.7716849
##    0.0020565123  0.9760816  0.12640925  0.7718162
##    0.0023101297  0.9761577  0.12631877  0.7719638
##    0.0025950242  0.9762578  0.12619962  0.7721312
```

```
## 0.0029150531 0.9763887 0.12604303 0.7723414
## 0.0032745492 0.9765584 0.12583663 0.7725954
## 0.0036783798 0.9767756 0.12557097 0.7728794
## 0.0041320124 0.9770538 0.12522643 0.7732245
## 0.0046415888 0.9774168 0.12476170 0.7736467
## 0.0052140083 0.9778836 0.12414338 0.7741313
## 0.0058570208 0.9784840 0.12331670 0.7747493
## 0.0065793322 0.9792591 0.12220102 0.7755570
## 0.0073907220 0.9802429 0.12072598 0.7765613
## 0.0083021757 0.9814818 0.11878954 0.7778023
## 0.0093260335 0.9829998 0.11633587 0.7792333
## 0.0104761575 0.9849164 0.11308017 0.7809314
## 0.0117681195 0.9873332 0.10877773 0.7829471
## 0.0132194115 0.9904062 0.10305807 0.7853466
## 0.0148496826 0.9943045 0.09554966 0.7883378
## 0.0166810054 0.9986372 0.08715637 0.7917896
## 0.0187381742 1.0000214 0.08473350 0.7929129
## 0.0210490414 1.0000562 0.08494733 0.7929042
## 0.0236448941 1.0001824 0.08502703 0.7929373
## 0.0265608778 1.0004079 0.08495207 0.7930512
## 0.0298364724 1.0007144 0.08481217 0.7932449
## 0.0335160265 1.0010958 0.08463082 0.7935107
## 0.0376493581 1.0015599 0.08441186 0.7938229
## 0.0422924287 1.0021305 0.08413908 0.7942505
## 0.0475081016 1.0028265 0.08379651 0.7947587
## 0.0533669923 1.0036968 0.08331091 0.7953587
## 0.0599484250 1.0047976 0.08259270 0.7960690
## 0.0673415066 1.0061616 0.08156668 0.7969913
## 0.0756463328 1.0076840 0.08041895 0.7980038
## 0.0849753436 1.0092618 0.07970835 0.7991879
## 0.0954548457 1.0110954 0.07890115 0.8006928
## 0.1072267222 1.0134043 0.07743096 0.8026675
## 0.1204503540 1.0163388 0.07457984 0.8051721
## 0.1353047775 1.0197924 0.07010347 0.8080000
## 0.1519911083 1.0230368 0.06637875 0.8105885
## 0.1707352647 1.0260602 0.06605730 0.8126892
## 0.1917910262 1.0297810 0.06605634 0.8152489
## 0.2154434690 1.0344526 0.06605634 0.8184422
## 0.2420128265 1.0395529 0.03679097 0.8215551
## 0.2718588243 1.0403394        NaN 0.8219620
## 0.3053855509 1.0403394        NaN 0.8219620
## 0.3430469286 1.0403394        NaN 0.8219620
## 0.3853528594 1.0403394        NaN 0.8219620
## 0.4328761281 1.0403394        NaN 0.8219620
## 0.4862601580 1.0403394        NaN 0.8219620
## 0.5462277218 1.0403394        NaN 0.8219620
## 0.6135907273 1.0403394        NaN 0.8219620
## 0.6892612104 1.0403394        NaN 0.8219620
## 0.7742636827 1.0403394        NaN 0.8219620
## 0.8697490026 1.0403394        NaN 0.8219620
## 0.9770099573 1.0403394        NaN 0.8219620
## 1.0974987655 1.0403394        NaN 0.8219620
## 1.2328467394 1.0403394        NaN 0.8219620
## 1.3848863714 1.0403394        NaN 0.8219620
```

```
##    1.5556761439  1.0403394         NaN  0.8219620
##    1.7475284000  1.0403394         NaN  0.8219620
##    1.9630406500  1.0403394         NaN  0.8219620
##    2.2051307399  1.0403394         NaN  0.8219620
##    2.4770763560  1.0403394         NaN  0.8219620
##    2.7825594022  1.0403394         NaN  0.8219620
##    3.1257158497  1.0403394         NaN  0.8219620
##    3.5111917342  1.0403394         NaN  0.8219620
##    3.9442060594  1.0403394         NaN  0.8219620
##    4.4306214576  1.0403394         NaN  0.8219620
##    4.9770235643  1.0403394         NaN  0.8219620
##    5.5908101825  1.0403394         NaN  0.8219620
##    6.2802914418  1.0403394         NaN  0.8219620
##    7.0548023107  1.0403394         NaN  0.8219620
##    7.9248289835  1.0403394         NaN  0.8219620
##    8.9021508545  1.0403394         NaN  0.8219620
##   10.0000000000  1.0403394         NaN  0.8219620
##
## Tuning parameter 'alpha' was held constant at a value of 1
## RMSE was used to select the optimal model using the smallest value.
## The final values used for the model were alpha = 1 and lambda = 0.0006428073.
```

```r
# Get the best lambda chosen by caret
best_lambda <- lasso_model$bestTune$lambda

# Extract the coefficients at the best lambda
lasso_coefs <- coef(lasso_model$finalModel, s = best_lambda)

# To convert to a tidy data frame (optional)
lasso_coefs_df <- as.data.frame(as.matrix(lasso_coefs))
lasso_coefs_df$Variable <- rownames(lasso_coefs_df)
colnames(lasso_coefs_df)[1] <- "Coefficient"

# View non-zero coefficients only (optional)
subset(lasso_coefs_df, Coefficient != 0)
```

```
##                  Coefficient       Variable
## (Intercept)     4.9053321709    (Intercept)
## Height         -0.0097653068         Height
## Age             0.0941196741            Age
## BPSysAve        0.0054486565       BPSysAve
## BPDiaAve        0.0130758073       BPDiaAve
## SmokeNowYes     0.0163610102    SmokeNowYes
## PhysActiveDays -0.0133636084 PhysActiveDays
## Age2           -0.0008885573           Age2
```

```r
#LASSO MODEL VALIDATION
train_lasso <- trainControl(method = "cv", number = 10)
cv_lasso <- train(
  pb.TotChol ~ Height+Age+BPSysAve+BPDiaAve+SmokeNow+PhysActiveDays+Age2,
  data = clean.frame,
  method = "lm",
  trControl = train_lasso
)
```

```r
print(cv_lasso)
```

```
## Linear Regression
##
## 1287 samples
##    7 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 1158, 1158, 1158, 1159, 1159, 1159, ...
## Resampling results:
##
##   RMSE       Rsquared   MAE
##   0.9772544  0.1289544  0.7713758
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```