project-final

Reading dataset from directory

```
setwd("/Users/matthewkilleen/Desktop/School/UAlbany Folder/Fall-2021-Semester/AMAT-465-Applied-Statisti
project_dataset <- read.table("./Datasets/project-dataset.csv", sep=",")
```

Editting variable names

```
colnames(project_dataset) = c('Fat_Percent', 'Density', 'Age', 'Weight',
                              'Height', 'Neck Circ.', 'Chest Circ.',
                              'Abdomen 2 Circ.', 'Hip Circ.', 'Thigh Circ.',
                              'Knee Circ.', 'Ankle Circ.', 'Bicep Circ.',
                              'Forearm Circ.', 'Wrist Circ.')
```

We know that Fat_Percent is determined via the rearranged formula for density, i.e. 100*B = 495/D - 450 where B is Fat_Percent and D is Density. That being said, Density has nearly 100% correlation with Fat_Percent. Furthermore, our goal is to predict Fat_Percent as calculated using the formula containing Density using the other body measurements. Therefore, we must delete Density from our data set.

In addition, when looking at the data set we can see that the data points in rows 39 and 42 are outliers. The data point in row 39 has an extremely high weight value of 363.15 lbs and other inflated measurements (to a lesser extent), while the data point at row 42 has an extremely low height value of 29.5 inches. Therefore, I feel that it would be best to omit these data points from the data set.

```
# Getting rid of column 'Density'
project_dataset = project_dataset[, -grep('Density', colnames(project_dataset))]

# Getting rid of data point with extremely high weight value, i.e. row 39
project_dataset = project_dataset[-c(which(project_dataset$Weight == max(project_dataset$Weight)),which
```

Finally, for all data points, I decided to remove all data points that had values for explanatory variables more than 5 standard deviations away from the mean. This is to ensure that the data for the explanatory variables is within the normal distribution.

```
for (val in 2:length(project_dataset)){
  mean = mean(project_dataset[, val])
  sdv = sd(project_dataset[, val])
  list = c(which(project_dataset[, val] > mean + 5 * sdv))
  if (length(list) > 0)
    project_dataset = project_dataset[-list, ]
}
```

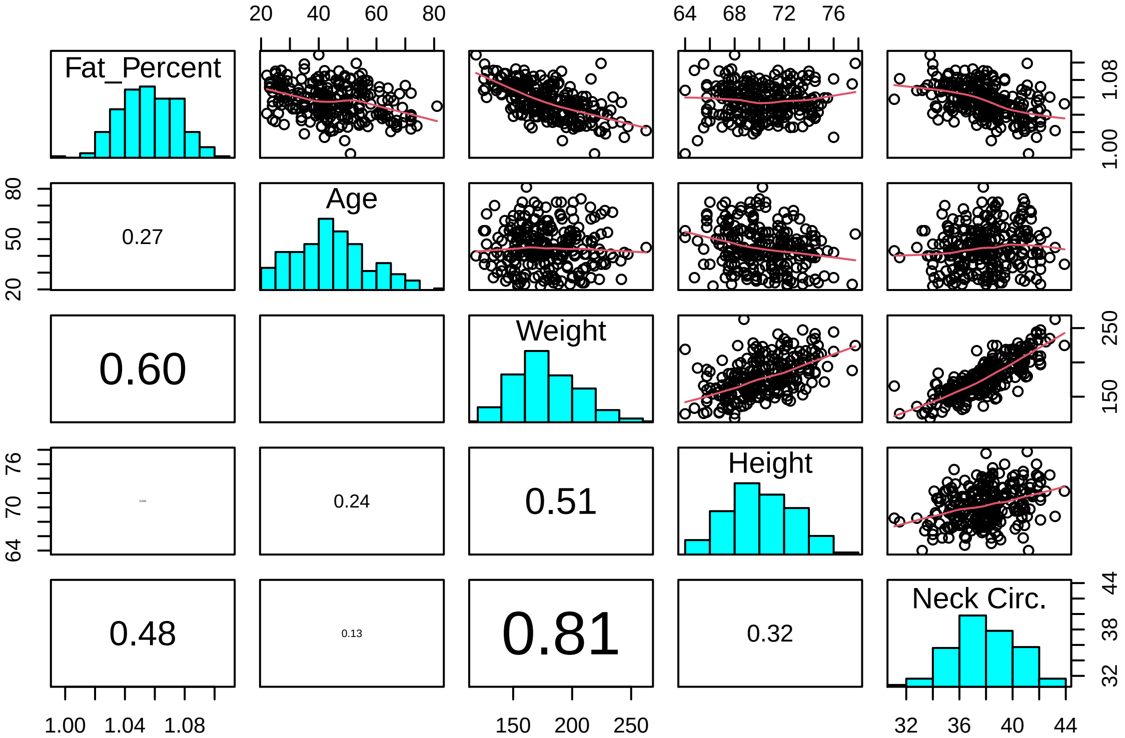
In order to derive optimal model(s), I will later carry out cross validation. Therefore, I will split the data set into training and testing data sets; the training data set will have 80% of the data and the testing data set will have the remaining 20%.

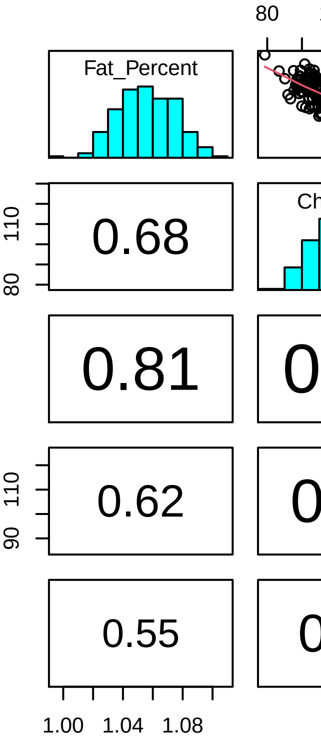
```
set.seed(100)
n<-length(project_dataset$`Fat_Percent`)
cvindex<-sample(1:n,.8*n,replace=FALSE)

train<-project_dataset[cvindex,]
test<-project_dataset[-cvindex,]
```

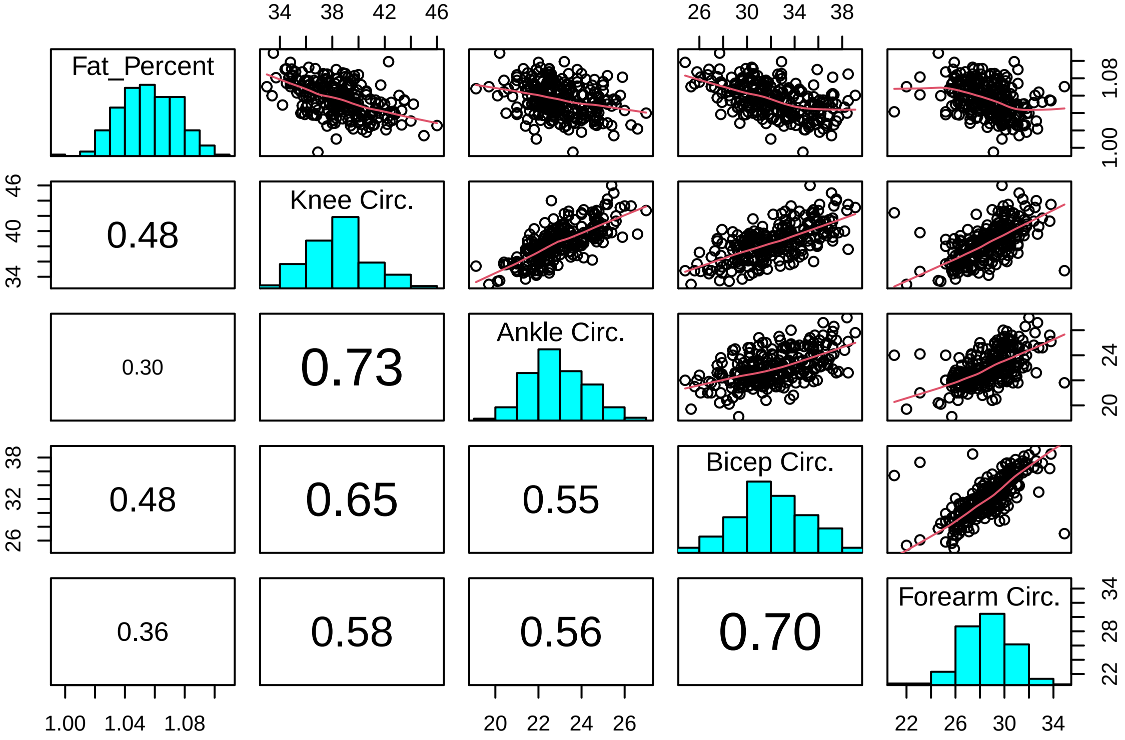
Before we create an initial model for the dataset, let's take a look at the output of pairs.R to get a sense of the relationships between variables:

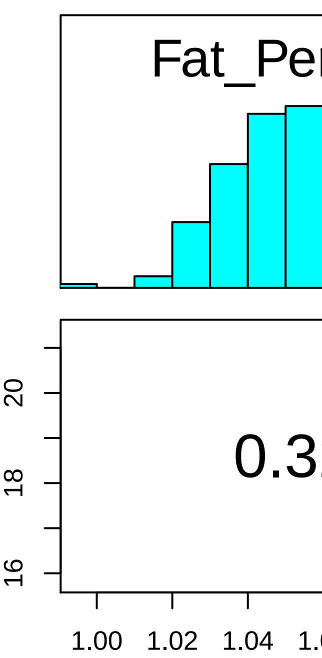
Partioning data to call pairs.r in a way to make plots more visible

```
#TODO: Make this look better later, it works for now
start = 2
end = if (length(project_dataset) < 5) length(project_dataset) else 5
source('pairs.R')
while (TRUE){
  pairs(project_dataset[, c(1, start:end)],panel=panel.smooth,diag.panel=panel.hist,lower.panel=panel.c
  start = end + 1
  if (start == length(project_dataset))
    end = start
  else if (length(project_dataset) - start < 3)
    break
  else
    end = start + 3
}
```









RELATIONSHIPS: We can see that a decent amount of the explanatory variables have a linear relationship with the response variable. However, there are some variables which have a curved relationship with the response variable (there are also a few explanatory variables that have curved relationships with other explanatory variables, though to a lesser extent). That being said, I believe it will be worthwhile to investigate the use of polynomial terms in the model as well as the transformation of the response and/or explanatory variables.

CORRELATIONS: It seems that the variables in the data set are relatively highly correlated with one another, which may lead to negative coefficients for the explanatory variables and / or intercept. This may also lead to a model that yields a significant F statistic for the global significance test for the model but has many insignificant explanatory variables.

Now that we've taken a look at the interactions between our variables and made some remarks about them, let's create a naive model where we use all variables:

generating and plotting model wherein Fat_Percent is the response variable

and all other variables are predictor variables

```
naive_model = lm(`Fat_Percent` ~ ., data = train)
summary(naive_model)
```

```
##
## Call:
## lm(formula = Fat_Percent ~ ., data = train)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.020529 -0.006811  0.000115  0.006552  0.034360
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)        1.114e+00  5.979e-02  18.624  < 2e-16 ***
## Age               -1.285e-04  8.642e-05  -1.487  0.13875
## Weight             1.228e-04  1.697e-04   0.724  0.47006
## Height             5.506e-04  5.003e-04   1.100  0.27259
## ‘Neck Circ.‘       4.459e-04  6.101e-04   0.731  0.46579
## ‘Chest Circ.‘      3.468e-04  2.832e-04   1.224  0.22236
## ‘Abdomen 2 Circ.‘ -2.237e-03  2.482e-04  -9.013 2.58e-16 ***
## ‘Hip Circ.‘        4.437e-04  3.743e-04   1.185  0.23744
## ‘Thigh Circ.‘     -3.090e-04  3.954e-04  -0.781  0.43555
## ‘Knee Circ.‘      -1.323e-04  6.559e-04  -0.202  0.84042
## ‘Ankle Circ.‘     -7.412e-04  9.478e-04  -0.782  0.43522
## ‘Bicep Circ.‘     -6.607e-04  4.491e-04  -1.471  0.14296
## ‘Forearm Circ.‘   -6.291e-04  5.117e-04  -1.230  0.22044
## ‘Wrist Circ.‘      4.276e-03  1.448e-03   2.953  0.00356 **
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 0.01002 on 184 degrees of freedom
## Multiple R-squared:  0.736,  Adjusted R-squared:  0.7173
## F-statistic: 39.45 on 13 and 184 DF,  p-value: < 2.2e-16
```

Looking at the above model, we can see a few interesting things:

1. A few of the coefficients for the explanatory variables are negative in our naive model (namely Age, Abdomen 2 Circ., Thigh Circ., Knee Circ., Ankle Circ., Bicep Circ. and Forearm Circ.)

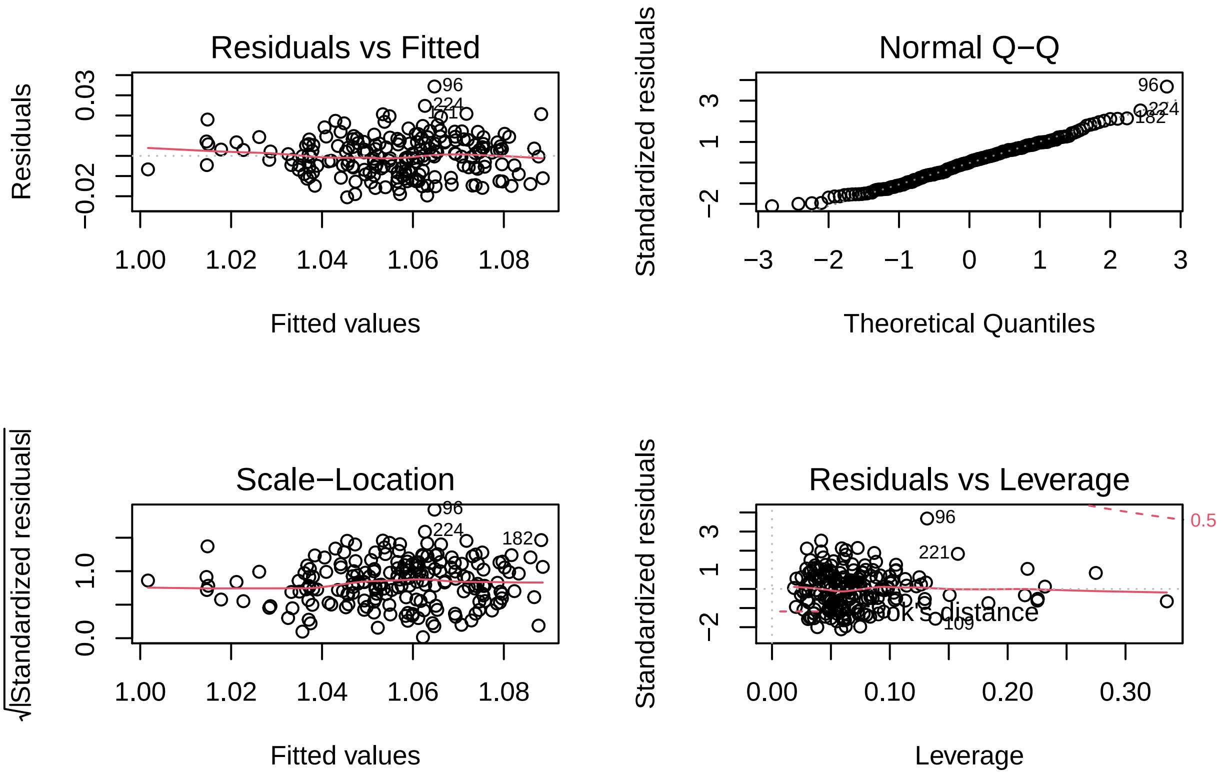
2. Many of the explanatory variables fail to be significant in the model
3. The glabal F test for the model indicates that the model is significant with all of these variables

This indicates that multicolinearity (correlation) is present within the model.

Looking at the other metrics produced for the model, we can see that the sum of square errors (AKA residual standard error) is low, indicating that the model fits the data well. In addition, the multiple R-squared and the adjusted R-squared have relatively high magnitudes; this indicates that there is a relatively high proportion of the variance for Fat_Percent that is explained by the model's explanatory variables.

Let's take a look at some of the diagnostic plots we can produce for the model:

```
par(mfrow = c(2,2))
plot(naive_model)
```



Residuals vs. Fitted plot: Used to determine if the residuals exhibit a non-linear pattern; the red line across the center of the plot is pretty horizontal, therefore it would be reasonable to assume that the residuals follow a linear pattern

Normal Q-Q plot: Used to determine if the residuals of the regression model are normally distributed; if the points fall roughly on the diagonal line, then we can assume they are normally distributed. Some of the points at both ends of the line start to deviate a bit from the diagonal, though most lie nearly on it so it's safe to assume they are normally distributed.

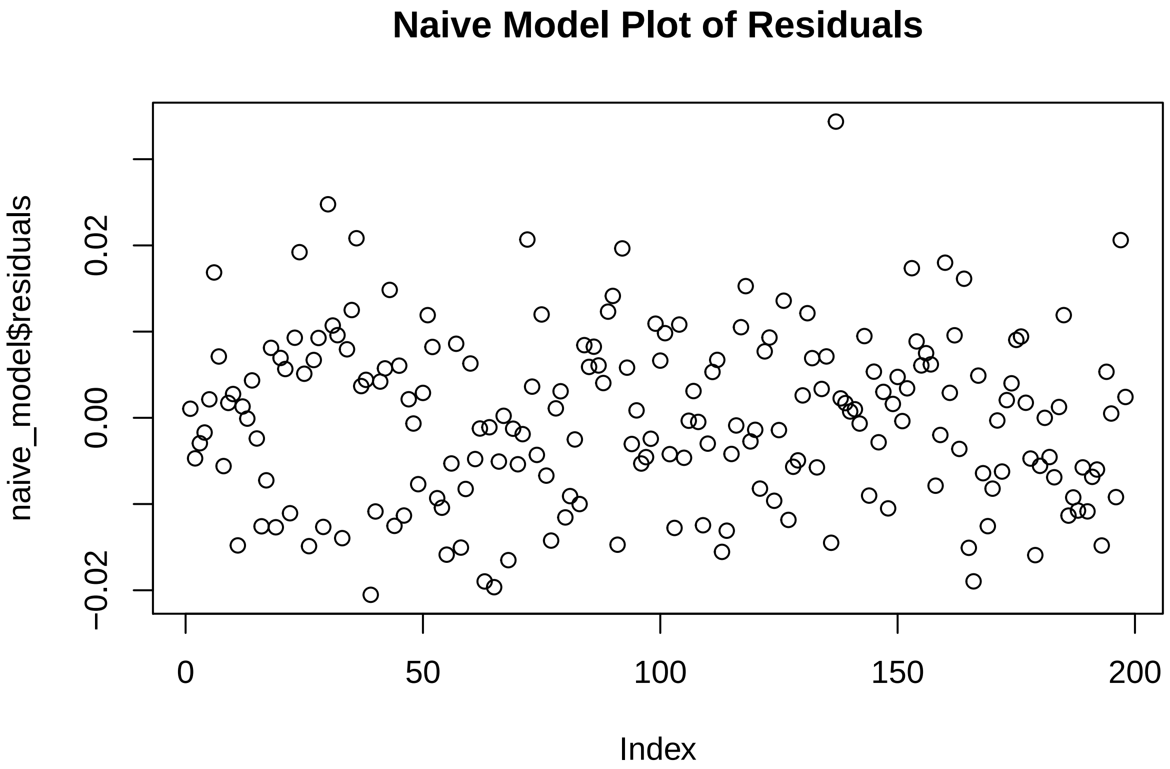
Scale-Location plot: Used to check the assumption of homoscedasticity / equal variance among the residuals of the regression model. If the red line is roughly horizontal across the plot, then the assumption is likely met. The red line in our plot is nearly horizontal, therefore it is safe to assume constant variance.

Residuals vs. Leverage plot: Used to identify influential observations. If any points in this plot fall outside of Cook's distance (the dashed line(s)), then the point(s) are influential. In our plot, none of the points cross and therefore there aren't any overly influential points in the data set.

To test our hypotheses that the residuals are normally distributed and have constant variance, we can use the Shapiro-Wilks test and Breusch-Pagan tests respectively. In the Shapiro-Wilks test, the null hypothesis is that the residuals are normally distributed, the alternative hypothesis being they are not normally distributed. In the Breusch-Pagan test, the null hypothesis is that the residuals have constant variance, the alternative being they have non constant variance. If the p-values < 0.5 for either of these tests, we reject the null hypothesis and accept the alternative. Otherwise, we fail to reject the null hypothesis.

Let's take a look at the naive model's residuals plot:

```
plot(naive_model$residuals, main="Naive Model Plot of Residuals")
```



We can see that there is no discernable pattern within the plot of residuals, indicating no correlation

Conducting Shapiro-Wilks and Breusch-Pagan tests on residuals

```
library(lmtest)
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
shapiro.test(naive_model$residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  naive_model$residuals
## W = 0.98885, p-value = 0.1249
```

```
bptest(naive_model)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  naive_model
## BP = 13.025, df = 13, p-value = 0.4459
```

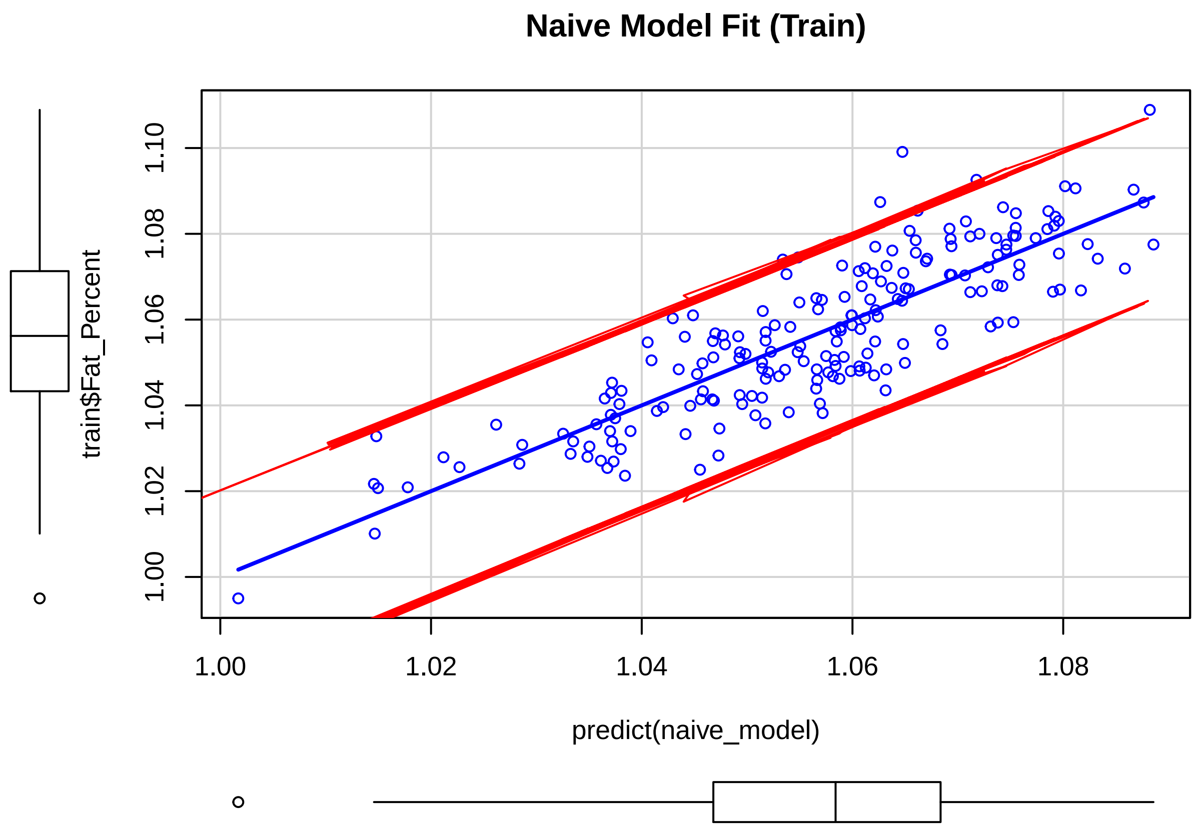
We can see that we fail to reject the null hypothesis for each test, therefore we conclude that the residuals of our model are normally distributed and have constant variance.

TAKING A LOOK AT HOW THE NAIVE MODEL FITS THE TRAINING DATA:

```
library(car)
```

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

```
scatterplot(predict(naive_model), train$Fat_Percent, smooth = FALSE, main = "Naive Model Fit (Train)")
pred_interval <- predict(naive_model, newdata=train, interval="prediction", level = 0.95)
lines(pred_interval[,1], pred_interval[,2], col="red")
lines(pred_interval[,1], pred_interval[,3], col="red")
```



REMEDYING CURVED RELATIONSHIPS:

Before we start doing cross validation, let's see if we can add any interactions, polynomial terms or transform our variables in order to produce a more easily interpretable model.

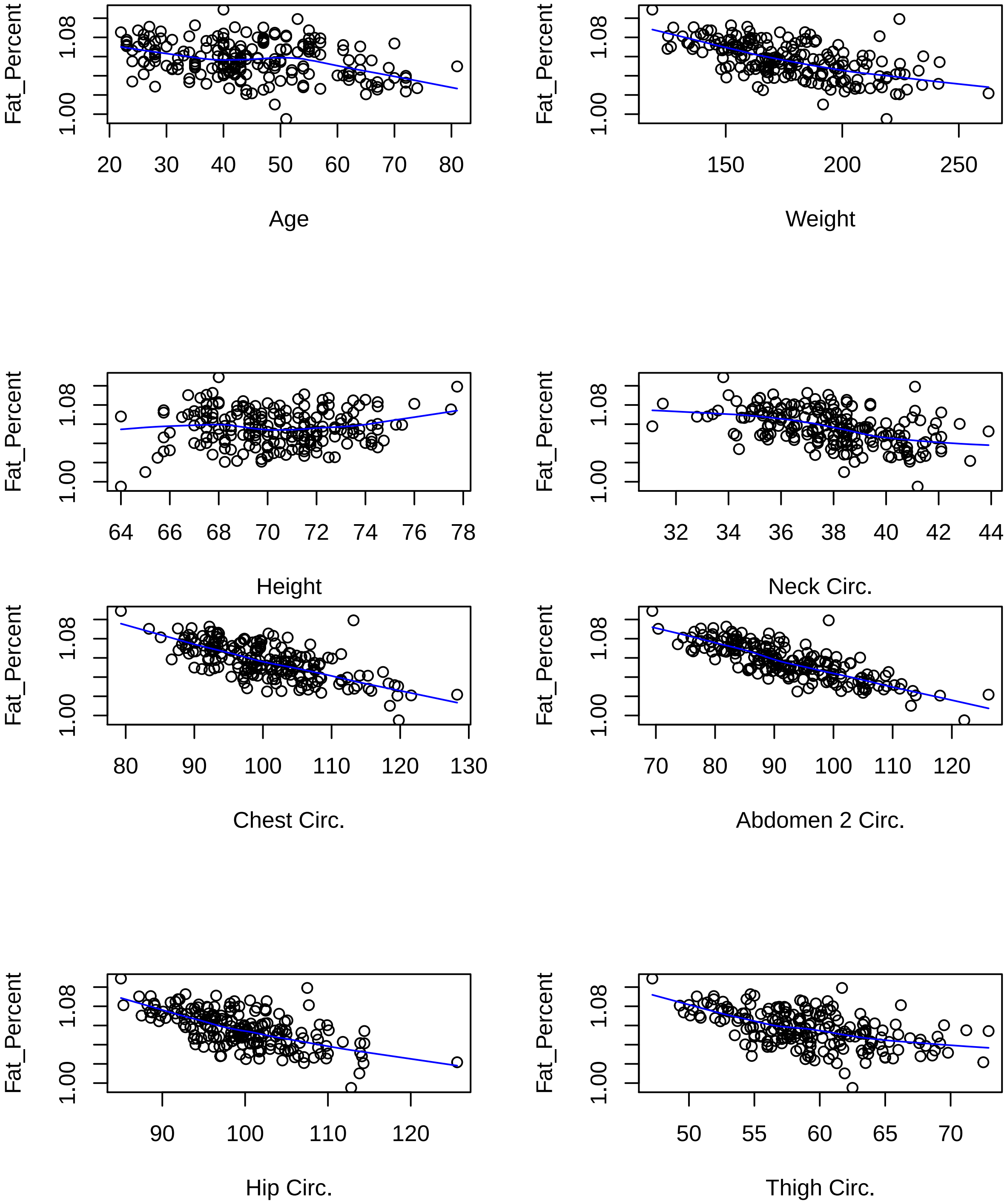
Interaction Effects:

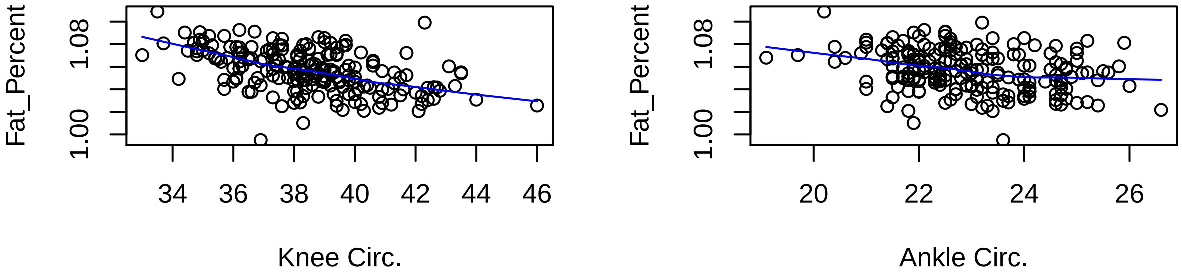
Since there are no categorical variables in the data set we are using (all variables are numeric), there is no need to include interaction effects in our model.

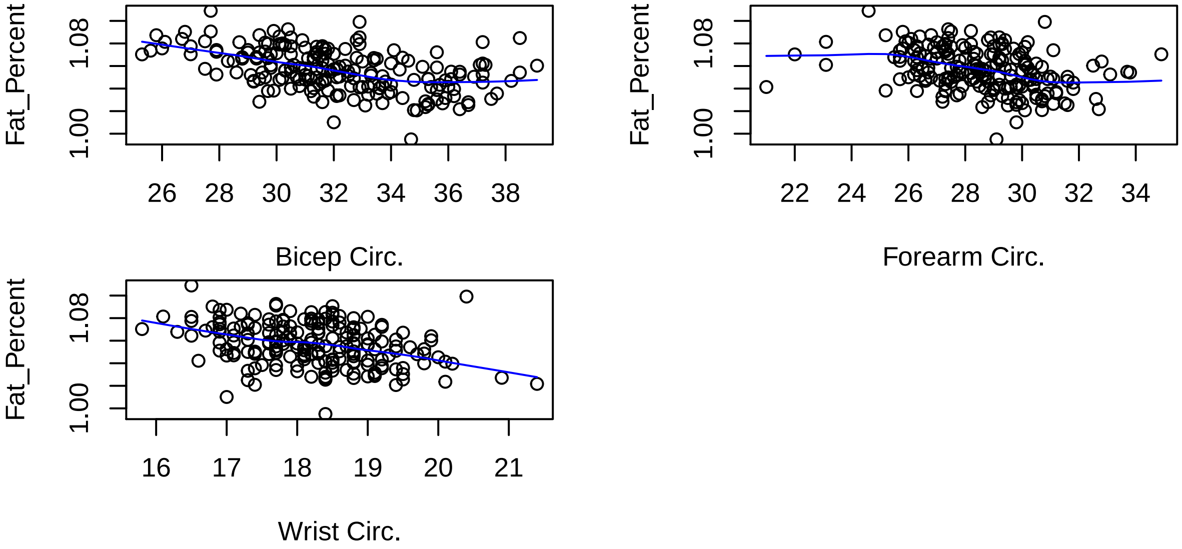
When we used pairs.R before, we plotted the relationships between our variables. Let's take a closer look at the relationships between the explanatory variables and the response variable:

Plotting explanatory variables vs. response variable

```
attach(train)
par(mfrow=c(2,2))
for (val in 2:length(train)){
  plot(train[, val], `Fat_Percent`, xlab = colnames(train[val]))
  lines(lowess(train[, val], `Fat_Percent`), col="blue")
}
```


Fat_Percent
1.00
1.08
20
30
40
50
60
70
80
Age
Fat_Percent
1.00
1.08
150
200
250
Weight
Fat_Percent
1.00
1.08
64
66
68
70
72
74
76
78
Height
Fat_Percent
1.00
1.08
32
34
36
38
40
42
44
Neck Circ.
Fat_Percent
1.00
1.08
80
90
100
110
120
130
Chest Circ.
Fat_Percent
1.00
1.08
70
80
90
100
110
120
Abdomen 2 Circ.
Fat_Percent
1.00
1.08
90
100
110
120
Hip Circ.
Fat_Percent
1.00
1.08
50
55
60
65
70
Thigh Circ.



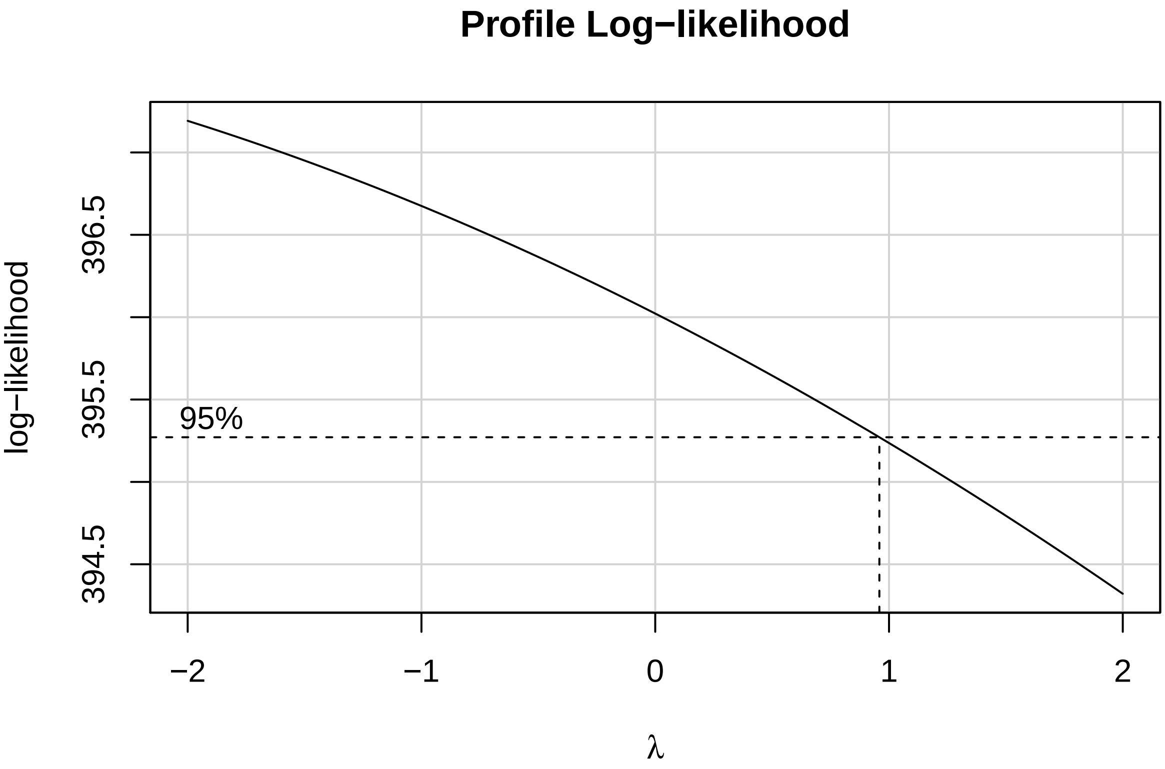


We can see a somewhat curved relationship in some of the plots, therefore let's look into whether we should try and transform our variables.

Transformation of Response:

Calling boxCox() on response variable

```
library(car)
boxCox(naive_model)
```



The boxCox plot indicates that an optimal choice for lambda is extremtly close to 1, indicating that raising our response variable to the power of 1 would be optimal. The reason that we'd transform our response variable would be in an effort to achieve constant variance and normality of the residuals for our model. We already have these, however, and we don't want to risk losing this. Therefore, it would be best to leave the response variable as it is.

Transformations on Explanatory Variables and Polynomial Terms:

As we saw in the plots of our explanatory variables vs. the response variable, some of the explanatory variables have a somewhat curvilinear relationship with the response variable, while the rest are more or less linear. This would indicate that we should attempt to transform some of our explanatory variables and / or add some polynomial terms into our model. The two that I have identified as most ploblematic are Neck Circ. and Bicep Circ. Both of these curves seem to have the shape of a cubic function stretched out across the x axis. Therefore, I am going to add a cubic polynomial term for both.

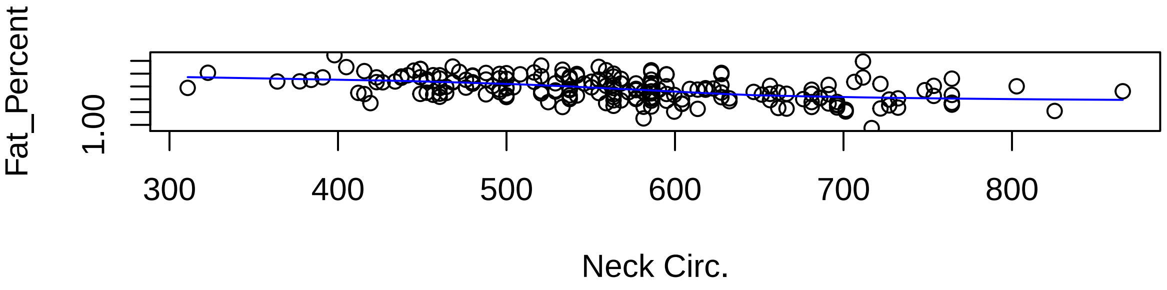
Plot of the problematic explanatory variables

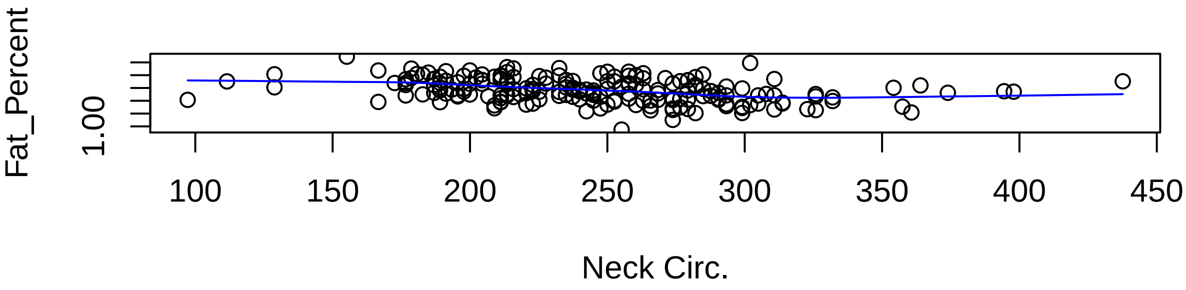
```
attach(train)
```

```
## The following objects are masked from train (pos = 3):
##
##     Abdomen 2 Circ., Age, Ankle Circ., Bicep Circ., Chest Circ.,
##     Fat_Percent, Forearm Circ., Height, Hip Circ., Knee Circ., Neck
##     Circ., Thigh Circ., Weight, Wrist Circ.
```

```
par(mfrow=c(2,1))
plot(.01 * (train[, 5] + train[, 5]^2 + train[, 5]^3), `Fat_Percent`, xlab = colnames(train[5]))
```

```
lines(lowess(.01 * (train[, 5] + train[, 5]^2 + train[, 5]^3), `Fat_Percent`), col="blue")

plot(.01 * (train[, 13] + train[, 13]^2 + train[, 13]^3), `Fat_Percent`, xlab = colnames(train[5]))
lines(lowess(.01 * (train[, 13] + train[, 13]^2 + train[, 13]^3), `Fat_Percent`), col="blue")
```





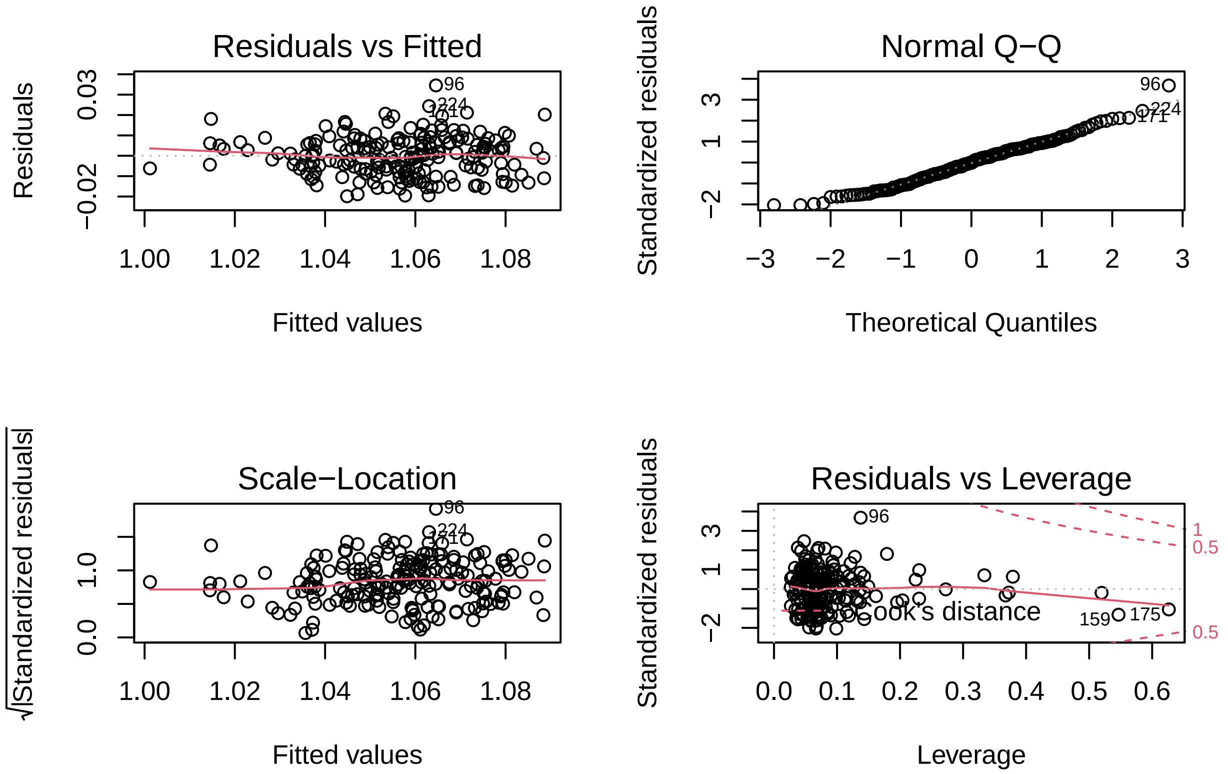
Creating model with polynomial terms

```
poly_model = lm(Fat_Percent ~ . -`Neck Circ.` + poly(.01 * `Neck Circ.`, 3)
                -`Forearm Circ.` + poly(.01 * `Forearm Circ.`, 3), data = train)
summary(poly_model)
```

```
##
## Call:
## lm(formula = Fat_Percent ~ . - ‘Neck Circ.‘ + poly(0.01 * ‘Neck Circ.‘,
##     3) - ‘Forearm Circ.‘ + poly(0.01 * ‘Forearm Circ.‘, 3), data = train)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.019871 -0.006882 -0.000080  0.006348  0.034536
##
## Coefficients:
##                                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)                       1.101e+00  6.051e-02  18.200  < 2e-16 ***
## Age                              -1.247e-04  8.780e-05  -1.421  0.15717
## Weight                            9.204e-05  1.851e-04   0.497  0.61969
## Height                            6.288e-04  5.387e-04   1.167  0.24460
## ‘Chest Circ.‘                     3.776e-04  2.966e-04   1.273  0.20452
## ‘Abdomen 2 Circ.‘                   -2.209e-03  2.566e-04  -8.608 3.63e-15 ***
## ‘Hip Circ.‘                          4.425e-04  3.815e-04   1.160  0.24760
## ‘Thigh Circ.‘                       -2.793e-04  4.087e-04  -0.683  0.49519
## ‘Knee Circ.‘                        -1.049e-04  6.648e-04  -0.158  0.87482
## ‘Ankle Circ.‘                       -6.680e-04  9.647e-04  -0.692  0.48955
## ‘Bicep Circ.‘                       -6.314e-04  4.674e-04  -1.351  0.17844
## ‘Wrist Circ.‘                        4.279e-03  1.490e-03   2.873  0.00456 **
## poly(0.01 * ‘Neck Circ.‘, 3)1        1.384e-02  2.106e-02   0.657  0.51192
## poly(0.01 * ‘Neck Circ.‘, 3)2        1.295e-03  1.268e-02   0.102  0.91875
## poly(0.01 * ‘Neck Circ.‘, 3)3        3.801e-03  1.147e-02   0.332  0.74064
## poly(0.01 * ‘Forearm Circ.‘, 3)1 -1.805e-02  1.772e-02  -1.019  0.30975
## poly(0.01 * ‘Forearm Circ.‘, 3)2  5.622e-03  1.215e-02   0.463  0.64404
## poly(0.01 * ‘Forearm Circ.‘, 3)3  3.233e-03  1.362e-02   0.237  0.81258
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 0.01011 on 180 degrees of freedom
## Multiple R-squared:  0.7369, Adjusted R-squared:  0.7121
## F-statistic: 29.66 on 17 and 180 DF,  p-value: < 2.2e-16
```

Plotting diagnostics for poly model

```
par(mfrow=c(2,2))
plot(poly_model)
```



Conducting Breusch-Pagan and Shapiro-Wilks tests to check for constant variance

and normally distributed residuals respectively

```
library(lmtest)
shapiro.test(poly_model$residuals)
```

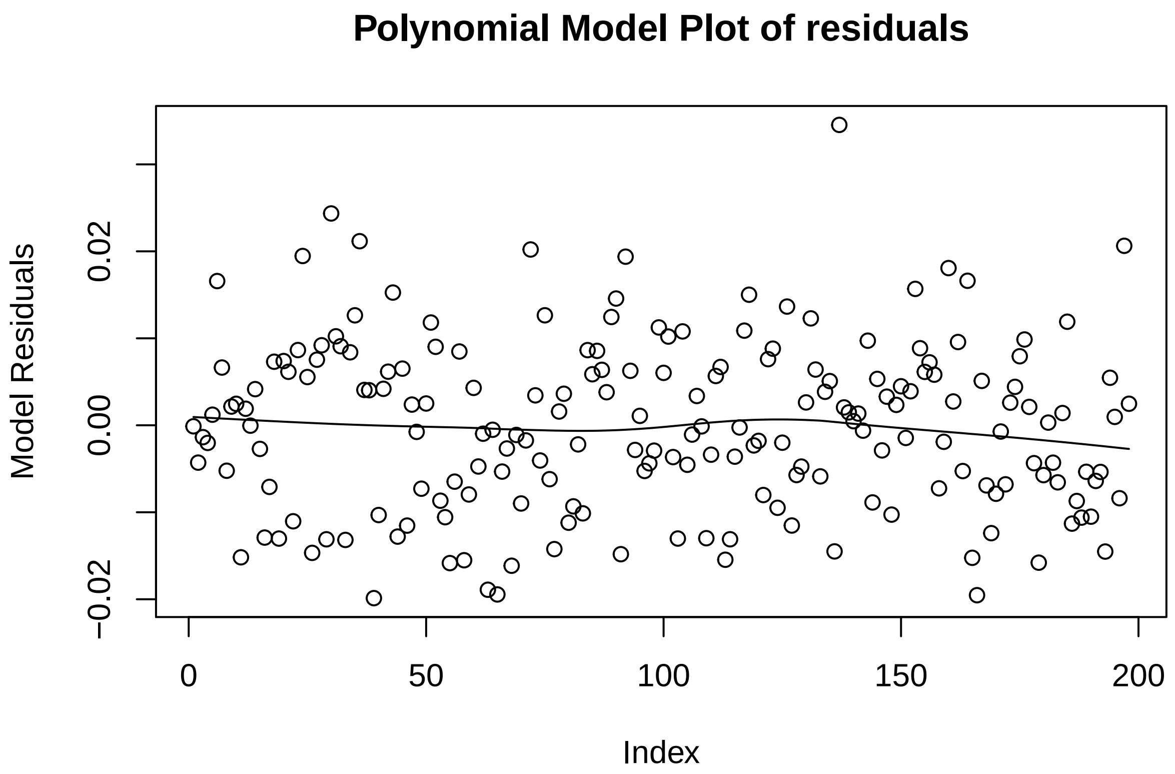
```
##
##  Shapiro-Wilk normality test
##
## data:  poly_model$residuals
## W = 0.98853, p-value = 0.1121
```

```
bptest(poly_model)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  poly_model
## BP = 14.881, df = 17, p-value = 0.6041
```

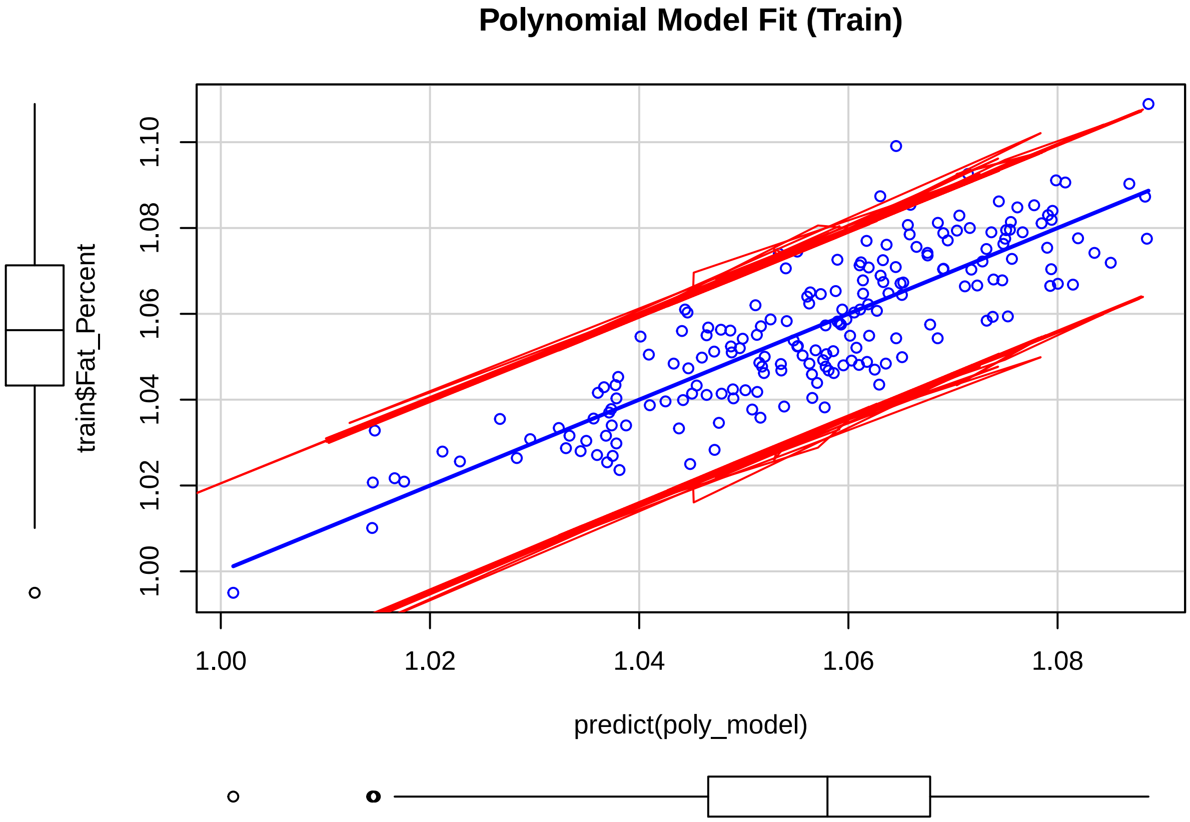
TAKING A LOOK AT MODEL RESIDUALS:

```
plot(poly_model$residuals, ylab = "Model Residuals", main = "Polynomial Model Plot of residuals")
lines(lowess(poly_model$residuals))
```



TAKING A LOOK AT HOW THE MODEL FITS THE TRAINING DATA:

```
library(car)
scatterplot(predict(poly_model), train$Fat_Percent, smooth = FALSE, main = "Polynomial Model Fit (Train
pred_interval <- predict(poly_model, newdata=train, interval="prediction", level = 0.95)
lines(pred_interval[,1], pred_interval[,2], col="red")
lines(pred_interval[,1], pred_interval[,3], col="red")
```



There doesn't seem to be any difference or improvement, so let's just stick with the naive model.

Cross Validation:

```
fullMSE<-summary(naive_model)$sig^2      # needed to compute Cp


backsel = step(naive_model, direction = "backward")
```

```
## Start:  AIC=-1809.55
## Fat_Percent ~ Age + Weight + Height + `Neck Circ.` + `Chest Circ.` +
##     `Abdomen 2 Circ.` + `Hip Circ.` + `Thigh Circ.` + `Knee Circ.` +
##     `Ankle Circ.` + `Bicep Circ.` + `Forearm Circ.` + `Wrist Circ.`
##
##                       Df Sum of Sq      RSS     AIC
## - `Knee Circ.`         1 0.0000041 0.018462 -1811.5
## - Weight               1 0.0000526 0.018510 -1811.0
## - `Neck Circ.`         1 0.0000536 0.018511 -1811.0
## - `Thigh Circ.`        1 0.0000613 0.018519 -1810.9
## - `Ankle Circ.`        1 0.0000613 0.018519 -1810.9
## - Height             1 0.0001215 0.018579 -1810.2
## - ‘Hip Circ.‘        1 0.0001409 0.018598 -1810.0
## - ‘Chest Circ.‘      1 0.0001504 0.018608 -1809.9
## - ‘Forearm Circ.‘    1 0.0001516 0.018609 -1809.9
## <none>                           0.018458 -1809.5
## - ‘Bicep Circ.‘      1 0.0002171 0.018675 -1809.2
## - Age                1 0.0002218 0.018679 -1809.2
## - ‘Wrist Circ.‘      1 0.0008746 0.019332 -1802.4
## - ‘Abdomen 2 Circ.‘  1 0.0081482 0.026606 -1739.2
##
## Step:  AIC=-1811.51
## Fat_Percent ~ Age + Weight + Height + ‘Neck Circ.‘ + ‘Chest Circ.‘ +
##     ‘Abdomen 2 Circ.‘ + ‘Hip Circ.‘ + ‘Thigh Circ.‘ + ‘Ankle Circ.‘ +
##     ‘Bicep Circ.‘ + ‘Forearm Circ.‘ + ‘Wrist Circ.‘
##
##                     Df Sum of Sq      RSS     AIC
## - Weight             1 0.0000497 0.018511 -1813.0
## - ‘Neck Circ.‘       1 0.0000569 0.018519 -1812.9
## - ‘Ankle Circ.‘      1 0.0000689 0.018530 -1812.8
## - ‘Thigh Circ.‘      1 0.0000774 0.018539 -1812.7
## - Height             1 0.0001174 0.018579 -1812.2
## - ‘Hip Circ.‘        1 0.0001384 0.018600 -1812.0
## - ‘Forearm Circ.‘    1 0.0001541 0.018616 -1811.9
## - ‘Chest Circ.‘      1 0.0001575 0.018619 -1811.8
## <none>                           0.018462 -1811.5
## - ‘Bicep Circ.‘      1 0.0002150 0.018677 -1811.2
## - Age                1 0.0002602 0.018722 -1810.7
## - ‘Wrist Circ.‘      1 0.0008718 0.019333 -1804.4
## - ‘Abdomen 2 Circ.‘  1 0.0081788 0.026640 -1740.9
##
## Step:  AIC=-1812.97
## Fat_Percent ~ Age + Height + ‘Neck Circ.‘ + ‘Chest Circ.‘ + ‘Abdomen 2 Circ.‘ +
##     ‘Hip Circ.‘ + ‘Thigh Circ.‘ + ‘Ankle Circ.‘ + ‘Bicep Circ.‘ +
##     ‘Forearm Circ.‘ + ‘Wrist Circ.‘
##
##                     Df Sum of Sq      RSS     AIC
## - ‘Ankle Circ.‘      1 0.0000440 0.018555 -1814.5
## - ‘Thigh Circ.‘      1 0.0000579 0.018569 -1814.3
## - ‘Neck Circ.‘       1 0.0001012 0.018613 -1813.9
## - ‘Forearm Circ.‘    1 0.0001425 0.018654 -1813.5
## - ‘Bicep Circ.‘      1 0.0001812 0.018693 -1813.0
## <none>                           0.018511 -1813.0
## - ‘Hip Circ.‘        1 0.0002544 0.018766 -1812.3
## - Age                1 0.0003086 0.018820 -1811.7
## - ‘Chest Circ.‘      1 0.0003737 0.018885 -1811.0
## - Height             1 0.0004880 0.018999 -1809.8
## - ‘Wrist Circ.‘      1 0.0009275 0.019439 -1805.3
## - ‘Abdomen 2 Circ.‘  1 0.0087551 0.027266 -1738.3
##
## Step:  AIC=-1814.5
## Fat_Percent ~ Age + Height + ‘Neck Circ.‘ + ‘Chest Circ.‘ + ‘Abdomen 2 Circ.‘ +
##     ‘Hip Circ.‘ + ‘Thigh Circ.‘ + ‘Bicep Circ.‘ + ‘Forearm Circ.‘ +
##     ‘Wrist Circ.‘
##
```

```
##                     Df Sum of Sq      RSS     AIC
## - ‘Thigh Circ.‘      1 0.0000846 0.018640 -1815.6
## - ‘Neck Circ.‘       1 0.0001260 0.018681 -1815.2
## - ‘Bicep Circ.‘      1 0.0001631 0.018718 -1814.8
## - ‘Forearm Circ.‘    1 0.0001649 0.018720 -1814.8
## <none>                           0.018555 -1814.5
## - ‘Hip Circ.‘        1 0.0002374 0.018793 -1814.0
## - Age                1 0.0002850 0.018840 -1813.5
## - ‘Chest Circ.‘      1 0.0003635 0.018919 -1812.7
## - Height             1 0.0004559 0.019011 -1811.7
## - ‘Wrist Circ.‘      1 0.0009581 0.019513 -1806.5
## - ‘Abdomen 2 Circ.‘  1 0.0087189 0.027274 -1740.2
##
## Step:  AIC=-1815.6
## Fat_Percent ~ Age + Height + ‘Neck Circ.‘ + ‘Chest Circ.‘ + ‘Abdomen 2 Circ.‘ +
##     ‘Hip Circ.‘ + ‘Bicep Circ.‘ + ‘Forearm Circ.‘ + ‘Wrist Circ.‘
##
##                     Df Sum of Sq      RSS     AIC
## - ‘Neck Circ.‘       1 0.0000992 0.018739 -1816.5
## - ‘Hip Circ.‘        1 0.0001556 0.018795 -1816.0
## - ‘Forearm Circ.‘    1 0.0001580 0.018798 -1815.9
## <none>                           0.018640 -1815.6
## - Age                1 0.0002127 0.018853 -1815.4
## - ‘Bicep Circ.‘      1 0.0002697 0.018910 -1814.8
## - ‘Chest Circ.‘      1 0.0004060 0.019046 -1813.3
## - Height             1 0.0004910 0.019131 -1812.5
## - ‘Wrist Circ.‘      1 0.0009590 0.019599 -1807.7
## - ‘Abdomen 2 Circ.‘  1 0.0089833 0.027623 -1739.7
##
## Step:  AIC=-1816.55
## Fat_Percent ~ Age + Height + ‘Chest Circ.‘ + ‘Abdomen 2 Circ.‘ +
##     ‘Hip Circ.‘ + ‘Bicep Circ.‘ + ‘Forearm Circ.‘ + ‘Wrist Circ.‘
##
##                     Df Sum of Sq      RSS     AIC
## - ‘Forearm Circ.‘    1 0.0001173 0.018856 -1817.3
## - ‘Hip Circ.‘        1 0.0001456 0.018885 -1817.0
## <none>                           0.018739 -1816.5
## - Age                1 0.0002165 0.018956 -1816.3
## - ‘Bicep Circ.‘      1 0.0002325 0.018972 -1816.1
## - ‘Chest Circ.‘      1 0.0004886 0.019228 -1813.5
## - Height             1 0.0005285 0.019268 -1813.0
## - ‘Wrist Circ.‘      1 0.0013324 0.020071 -1805.0
## - ‘Abdomen 2 Circ.‘  1 0.0088849 0.027624 -1741.7
##
## Step:  AIC=-1817.32
## Fat_Percent ~ Age + Height + ‘Chest Circ.‘ + ‘Abdomen 2 Circ.‘ +
##     ‘Hip Circ.‘ + ‘Bicep Circ.‘ + ‘Wrist Circ.‘
##
##                     Df Sum of Sq      RSS     AIC
## - ‘Hip Circ.‘        1 0.0001432 0.018999 -1817.8
## - Age                1 0.0001729 0.019029 -1817.5
## <none>                           0.018856 -1817.3
## - ‘Bicep Circ.‘      1 0.0004024 0.019259 -1815.1
## - ‘Chest Circ.‘      1 0.0004401 0.019297 -1814.8
## - Height               1 0.0005262 0.019383 -1813.9
## - ‘Wrist Circ.‘        1 0.0012180 0.020074 -1806.9
## - ‘Abdomen 2 Circ.‘    1 0.0088211 0.027678 -1743.3
##
## Step:  AIC=-1817.82
## Fat_Percent ~ Age + Height + ‘Chest Circ.‘ + ‘Abdomen 2 Circ.‘ +
##     ‘Bicep Circ.‘ + ‘Wrist Circ.‘
##
##                       Df Sum of Sq      RSS     AIC
## <none>                            0.018999 -1817.8
## - ‘Bicep Circ.‘        1 0.0003259 0.019325 -1816.5
## - Age                  1 0.0004544 0.019454 -1815.1
## - ‘Chest Circ.‘        1 0.0004648 0.019464 -1815.0
## - Height               1 0.0007690 0.019769 -1812.0
## - ‘Wrist Circ.‘        1 0.0013497 0.020349 -1806.2
## - ‘Abdomen 2 Circ.‘    1 0.0121189 0.031119 -1722.1
```

```
bothsel<-step(naive_model, direction = "both")
```

```
## Start:  AIC=-1809.55
## Fat_Percent ~ Age + Weight + Height + ‘Neck Circ.‘ + ‘Chest Circ.‘ +
##     ‘Abdomen 2 Circ.‘ + ‘Hip Circ.‘ + ‘Thigh Circ.‘ + ‘Knee Circ.‘ +
##     ‘Ankle Circ.‘ + ‘Bicep Circ.‘ + ‘Forearm Circ.‘ + ‘Wrist Circ.‘
##
##                       Df Sum of Sq      RSS     AIC
## - ‘Knee Circ.‘         1 0.0000041 0.018462 -1811.5
## - Weight               1 0.0000526 0.018510 -1811.0
## - ‘Neck Circ.‘         1 0.0000536 0.018511 -1811.0
## - ‘Thigh Circ.‘        1 0.0000613 0.018519 -1810.9
## - ‘Ankle Circ.‘        1 0.0000613 0.018519 -1810.9
## - Height               1 0.0001215 0.018579 -1810.2
## - ‘Hip Circ.‘          1 0.0001409 0.018598 -1810.0
## - ‘Chest Circ.‘        1 0.0001504 0.018608 -1809.9
## - ‘Forearm Circ.‘      1 0.0001516 0.018609 -1809.9
## <none>                            0.018458 -1809.5
## - ‘Bicep Circ.‘        1 0.0002171 0.018675 -1809.2
## - Age                  1 0.0002218 0.018679 -1809.2
## - ‘Wrist Circ.‘        1 0.0008746 0.019332 -1802.4
## - ‘Abdomen 2 Circ.‘    1 0.0081482 0.026606 -1739.2
##
## Step:  AIC=-1811.51
## Fat_Percent ~ Age + Weight + Height + ‘Neck Circ.‘ + ‘Chest Circ.‘ +
##     ‘Abdomen 2 Circ.‘ + ‘Hip Circ.‘ + ‘Thigh Circ.‘ + ‘Ankle Circ.‘ +
##     ‘Bicep Circ.‘ + ‘Forearm Circ.‘ + ‘Wrist Circ.‘
##
##                       Df Sum of Sq      RSS     AIC
## - Weight               1 0.0000497 0.018511 -1813.0
## - ‘Neck Circ.‘         1 0.0000569 0.018519 -1812.9
## - ‘Ankle Circ.‘        1 0.0000689 0.018530 -1812.8
## - ‘Thigh Circ.‘        1 0.0000774 0.018539 -1812.7
## - Height               1 0.0001174 0.018579 -1812.2
## - ‘Hip Circ.‘          1 0.0001384 0.018600 -1812.0
## - ‘Forearm Circ.‘      1 0.0001541 0.018616 -1811.9
## - ‘Chest Circ.‘        1 0.0001575 0.018619 -1811.8
## <none>                           0.018462 -1811.5
## - ‘Bicep Circ.‘      1 0.0002150 0.018677 -1811.2
## - Age                1 0.0002602 0.018722 -1810.7
## + ‘Knee Circ.‘       1 0.0000041 0.018458 -1809.5
## - ‘Wrist Circ.‘      1 0.0008718 0.019333 -1804.4
## - ‘Abdomen 2 Circ.‘  1 0.0081788 0.026640 -1740.9
##
## Step:  AIC=-1812.97
## Fat_Percent ~ Age + Height + ‘Neck Circ.‘ + ‘Chest Circ.‘ + ‘Abdomen 2 Circ.‘ +
##     ‘Hip Circ.‘ + ‘Thigh Circ.‘ + ‘Ankle Circ.‘ + ‘Bicep Circ.‘ +
##     ‘Forearm Circ.‘ + ‘Wrist Circ.‘
##
##                      Df Sum of Sq      RSS     AIC
## - ‘Ankle Circ.‘      1 0.0000440 0.018555 -1814.5
## - ‘Thigh Circ.‘      1 0.0000579 0.018569 -1814.3
## - ‘Neck Circ.‘       1 0.0001012 0.018613 -1813.9
## - ‘Forearm Circ.‘    1 0.0001425 0.018654 -1813.5
## - ‘Bicep Circ.‘      1 0.0001812 0.018693 -1813.0
## <none>                           0.018511 -1813.0
## - ‘Hip Circ.‘        1 0.0002544 0.018766 -1812.3
## - Age                1 0.0003086 0.018820 -1811.7
## + Weight             1 0.0000497 0.018462 -1811.5
## - ‘Chest Circ.‘      1 0.0003737 0.018885 -1811.0
## + ‘Knee Circ.‘       1 0.0000012 0.018510 -1811.0
## - Height             1 0.0004880 0.018999 -1809.8
## - ‘Wrist Circ.‘      1 0.0009275 0.019439 -1805.3
## - ‘Abdomen 2 Circ.‘  1 0.0087551 0.027266 -1738.3
##
## Step:  AIC=-1814.5
## Fat_Percent ~ Age + Height + ‘Neck Circ.‘ + ‘Chest Circ.‘ + ‘Abdomen 2 Circ.‘ +
##     ‘Hip Circ.‘ + ‘Thigh Circ.‘ + ‘Bicep Circ.‘ + ‘Forearm Circ.‘ +
##     ‘Wrist Circ.‘
##
##                      Df Sum of Sq      RSS     AIC
## - ‘Thigh Circ.‘      1 0.0000846 0.018640 -1815.6
## - ‘Neck Circ.‘       1 0.0001260 0.018681 -1815.2
## - ‘Bicep Circ.‘      1 0.0001631 0.018718 -1814.8
## - ‘Forearm Circ.‘    1 0.0001649 0.018720 -1814.8
## <none>                           0.018555 -1814.5
## - ‘Hip Circ.‘        1 0.0002374 0.018793 -1814.0
## - Age                1 0.0002850 0.018840 -1813.5
## + ‘Ankle Circ.‘      1 0.0000440 0.018511 -1813.0
## + Weight             1 0.0000248 0.018530 -1812.8
## - ‘Chest Circ.‘      1 0.0003635 0.018919 -1812.7
## + ‘Knee Circ.‘       1 0.0000061 0.018549 -1812.6
## - Height             1 0.0004559 0.019011 -1811.7
## - ‘Wrist Circ.‘      1 0.0009581 0.019513 -1806.5
## - ‘Abdomen 2 Circ.‘  1 0.0087189 0.027274 -1740.2
##
## Step:  AIC=-1815.6
## Fat_Percent ~ Age + Height + ‘Neck Circ.‘ + ‘Chest Circ.‘ + ‘Abdomen 2 Circ.‘ +
##     ‘Hip Circ.‘ + ‘Bicep Circ.‘ + ‘Forearm Circ.‘ + ‘Wrist Circ.‘
##
##                      Df Sum of Sq      RSS     AIC
## - ‘Neck Circ.‘       1 0.0000992 0.018739 -1816.5
## - ‘Hip Circ.‘        1 0.0001556 0.018795 -1816.0
## - ‘Forearm Circ.‘    1 0.0001580 0.018798 -1815.9
## <none>                           0.018640 -1815.6
## - Age                1 0.0002127 0.018853 -1815.4
## - ‘Bicep Circ.‘      1 0.0002697 0.018910 -1814.8
## + ‘Thigh Circ.‘      1 0.0000846 0.018555 -1814.5
## + ‘Ankle Circ.‘      1 0.0000707 0.018569 -1814.3
## + ‘Knee Circ.‘       1 0.0000309 0.018609 -1813.9
## + Weight             1 0.0000070 0.018633 -1813.7
## - ‘Chest Circ.‘      1 0.0004060 0.019046 -1813.3
## - Height             1 0.0004910 0.019131 -1812.5
## - ‘Wrist Circ.‘      1 0.0009590 0.019599 -1807.7
## - ‘Abdomen 2 Circ.‘  1 0.0089833 0.027623 -1739.7
##
## Step:  AIC=-1816.55
## Fat_Percent ~ Age + Height + ‘Chest Circ.‘ + ‘Abdomen 2 Circ.‘ +
##     ‘Hip Circ.‘ + ‘Bicep Circ.‘ + ‘Forearm Circ.‘ + ‘Wrist Circ.‘
##
##                     Df Sum of Sq      RSS     AIC
## - ‘Forearm Circ.‘    1 0.0001173 0.018856 -1817.3
## - ‘Hip Circ.‘        1 0.0001456 0.018885 -1817.0
## <none>                           0.018739 -1816.5
## - Age                1 0.0002165 0.018956 -1816.3
## - ‘Bicep Circ.‘      1 0.0002325 0.018972 -1816.1
## + ‘Neck Circ.‘       1 0.0000992 0.018640 -1815.6
## + ‘Ankle Circ.‘      1 0.0000915 0.018648 -1815.5
## + ‘Thigh Circ.‘      1 0.0000578 0.018681 -1815.2
## + ‘Knee Circ.‘       1 0.0000348 0.018704 -1814.9
## + Weight             1 0.0000280 0.018711 -1814.8
## - ‘Chest Circ.‘      1 0.0004886 0.019228 -1813.5
## - Height             1 0.0005285 0.019268 -1813.0
## - ‘Wrist Circ.‘      1 0.0013324 0.020071 -1805.0
## - ‘Abdomen 2 Circ.‘  1 0.0088849 0.027624 -1741.7
##
## Step:  AIC=-1817.32
## Fat_Percent ~ Age + Height + ‘Chest Circ.‘ + ‘Abdomen 2 Circ.‘ +
##     ‘Hip Circ.‘ + ‘Bicep Circ.‘ + ‘Wrist Circ.‘
##
##                     Df Sum of Sq      RSS     AIC
## - ‘Hip Circ.‘        1 0.0001432 0.018999 -1817.8
## - Age                1 0.0001729 0.019029 -1817.5
## <none>                           0.018856 -1817.3
## + ‘Forearm Circ.‘    1 0.0001173 0.018739 -1816.5
## + ‘Ankle Circ.‘      1 0.0001104 0.018746 -1816.5
## + ‘Neck Circ.‘       1 0.0000585 0.018798 -1815.9
## + ‘Thigh Circ.‘      1 0.0000577 0.018799 -1815.9
## + ‘Knee Circ.‘       1 0.0000421 0.018814 -1815.8
## + Weight             1 0.0000135 0.018843 -1815.5
## - ‘Bicep Circ.‘      1 0.0004024 0.019259 -1815.1
## - ‘Chest Circ.‘      1 0.0004401 0.019297 -1814.8
## - Height             1 0.0005262 0.019383 -1813.9
## - ‘Wrist Circ.‘      1 0.0012180 0.020074 -1806.9
## - ‘Abdomen 2 Circ.‘  1 0.0088211 0.027678 -1743.3
```

```
##
## Step:  AIC=-1817.82
## Fat_Percent ~ Age + Height + ‘Chest Circ.‘ + ‘Abdomen 2 Circ.‘ +
##     ‘Bicep Circ.‘ + ‘Wrist Circ.‘
##
##                       Df Sum of Sq      RSS     AIC
## <none>                             0.018999 -1817.8
## + ‘Hip Circ.‘        1 0.0001432 0.018856 -1817.3
## + ‘Forearm Circ.‘    1 0.0001149 0.018885 -1817.0
## + Weight             1 0.0000868 0.018913 -1816.7
## - ‘Bicep Circ.‘      1 0.0003259 0.019325 -1816.5
## + ‘Ankle Circ.‘      1 0.0000573 0.018942 -1816.4
## + ‘Neck Circ.‘       1 0.0000514 0.018948 -1816.3
## + ‘Knee Circ.‘       1 0.0000042 0.018995 -1815.9
## + ‘Thigh Circ.‘      1 0.0000005 0.018999 -1815.8
## - Age                1 0.0004544 0.019454 -1815.1
## - ‘Chest Circ.‘      1 0.0004648 0.019464 -1815.0
## - Height             1 0.0007690 0.019769 -1812.0
## - ‘Wrist Circ.‘      1 0.0013497 0.020349 -1806.2
## - ‘Abdomen 2 Circ.‘  1 0.0121189 0.031119 -1722.1
```

```
source("modelselectionfunctions.R")
library(leaps)
lp2<-regsubsets(Fat_Percent ~ ., nbest=3, data=train, really.big=T)


lp2matrix<-matrix.selection(lp2,Xnames=lp2$xnames[-1],Yname='Fat_Percent',fullMSE,train)


size<-apply((summary(lp2)$which*1),1,sum)                              #size=p+1

ibestbic<-which(summary(lp2)$bic==min(summary(lp2)$bic))
ibestadjr2<-which(summary(lp2)$adjr2==max(summary(lp2)$adjr2))
ibestcp<- which(abs(lp2matrix$Cp - size) == min(abs(lp2matrix$Cp - size)))
ibestaic<-which(lp2matrix$AIC==min(lp2matrix$AIC))

foo <- summary(lp2)$which[ibestbic, ]
form <- lp2$xnames[foo][-1]          #remove the intercept
form <- paste(form, collapse = " + ")
form <- paste("Fat_Percent~", form)
bicmod<- lm(as.formula(form), data=train)


foo <- summary(lp2)$which[ibestaic, ]
form <- lp2$xnames[foo][-1]
form <- paste(form, collapse = " + ")
form <- form <- paste("Fat_Percent~", form)
aicmod<- lm(as.formula(form),data=train)


foo <- summary(lp2)$which[ibestadjr2, ]
form <- lp2$xnames[foo][-1]
form <- paste(form, collapse = " + ")
form <- form <- paste("Fat_Percent~", form)
```

```
adjr2mod<- lm(as.formula(form),data=train)


foo <- summary(lp2)$which[ibestcp, ]
form <- lp2$xnames[foo][-1]
form <- paste(form, collapse = " + ")
form <- form <- paste("Fat_Percent~", form)
cpmod<- lm(as.formula(form),data=train)



rbind(bsel=Criteria(backsel,fullMSE,label=T), bicm=Criteria(bicmod,fullMSE,label=T),
      aicm=Criteria(aicmod,fullMSE),
      adjr2m=Criteria(adjr2mod,fullMSE),
      cpm=Criteria(cpmod,fullMSE))
```

```
##        p+1  R2adj   Cp      AIC      PRESS
## bsel     7 0.7197 5.40 -1817.82 0.02053340
## bicm     5 0.7182 4.38 -1818.73 0.02042259
## aicm     5 0.7182 4.38 -1818.73 0.02042259
## adjr2m   9 0.7206 6.81 -1816.55 0.02050852
## cpm      5 0.7182 4.38 -1818.73 0.02042259
```

```
#MSEpred = mean ((bicmod$fitted.values - test)^2)

bic_values = bicmod$fitted.values[test$`Fat_Percent`]
aic_values = aicmod$fitted.values[test$`Fat_Percent`]
adj_values = adjr2mod$fitted.values[test$`Fat_Percent`]
cp_values = cpmod$fitted.values[test$`Fat_Percent`]

MSEpredbic = mean((bic_values - test$`Fat_Percent`)^2)
MSEpredaic = mean((aic_values - test$`Fat_Percent`)^2)
MSEpredadj = mean((adj_values - test$`Fat_Percent`)^2)
MSEpredcp = mean((cp_values - test$`Fat_Percent`)^2)
```

```
summary(backsel)
```

```
##
## Call:
## lm(formula = Fat_Percent ~ Age + Height + ‘Chest Circ.‘ + ‘Abdomen 2 Circ.‘ +
##     ‘Bicep Circ.‘ + ‘Wrist Circ.‘, data = train)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.019746 -0.007174  0.000517  0.006511  0.036110
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)        1.0773047  0.0219167  49.154  < 2e-16 ***
## Age               -0.0001407  0.0000658  -2.137 0.033836 *
## Height             0.0009092  0.0003270   2.780 0.005974 **
## ‘Chest Circ.‘      0.0004976  0.0002302   2.162 0.031901 *
## ‘Abdomen 2 Circ.‘ -0.0019895  0.0001802 -11.038  < 2e-16 ***
## ‘Bicep Circ.‘     -0.0006826  0.0003771  -1.810 0.071854 .
## ‘Wrist Circ.‘      0.0042127  0.0011437   3.684 0.000299 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 0.009974 on 191 degrees of freedom
## Multiple R-squared:  0.7282, Adjusted R-squared:  0.7197
## F-statistic: 85.29 on 6 and 191 DF,  p-value: < 2.2e-16
```

```
# summary(bothsel) # backsel and bothsel produce same models, therefore just use backsel
summary(bicmod)
```

```
##
## Call:
## lm(formula = as.formula(form), data = train)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.020331 -0.007433  0.000518  0.006422  0.037560
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)         1.181e+00  1.784e-02  66.226  < 2e-16 ***
## Weight              3.201e-04  7.054e-05   4.538 9.99e-06 ***
## ‘Abdomen 2 Circ.‘ -2.272e-03  1.434e-04 -15.845  < 2e-16 ***
## ‘Bicep Circ.‘     -9.009e-04  3.938e-04  -2.288  0.02324 *
## ‘Wrist Circ.‘      3.102e-03  1.104e-03   2.809  0.00548 **
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 0.01 on 193 degrees of freedom
## Multiple R-squared:  0.7239, Adjusted R-squared:  0.7182
## F-statistic: 126.5 on 4 and 193 DF,  p-value: < 2.2e-16
```

```
summary(aicmod)
```

```
##
## Call:
## lm(formula = as.formula(form), data = train)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.020331 -0.007433  0.000518  0.006422  0.037560
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)         1.181e+00  1.784e-02  66.226  < 2e-16 ***
## Weight              3.201e-04  7.054e-05   4.538 9.99e-06 ***
## ‘Abdomen 2 Circ.‘ -2.272e-03  1.434e-04 -15.845  < 2e-16 ***
## ‘Bicep Circ.‘     -9.009e-04  3.938e-04  -2.288  0.02324 *
## ‘Wrist Circ.‘      3.102e-03  1.104e-03   2.809  0.00548 **
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 0.01 on 193 degrees of freedom
## Multiple R-squared:  0.7239, Adjusted R-squared:  0.7182
## F-statistic: 126.5 on 4 and 193 DF,  p-value: < 2.2e-16
```

```
summary(adjr2mod)
```

```
##
## Call:
## lm(formula = as.formula(form), data = train)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.019626 -0.006664  0.000608  0.006087  0.035813
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)         1.075e+00  2.244e-02  47.904  < 2e-16 ***
## Age                -1.116e-04  7.549e-05  -1.478  0.14115
## Height              7.893e-04  3.418e-04   2.309  0.02203 *
## ‘Chest Circ.‘       5.142e-04  2.317e-04   2.220  0.02762 *
## ‘Abdomen 2 Circ.‘ -2.169e-03  2.291e-04  -9.466  < 2e-16 ***
## ‘Hip Circ.‘         3.544e-04  2.925e-04   1.212  0.22715
## ‘Bicep Circ.‘      -6.245e-04  4.078e-04  -1.531  0.12735
## ‘Forearm Circ.‘    -5.364e-04  4.932e-04  -1.088  0.27816
## ‘Wrist Circ.‘       4.360e-03  1.189e-03   3.666  0.00032 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 0.009957 on 189 degrees of freedom
## Multiple R-squared:  0.7319, Adjusted R-squared:  0.7206
## F-statistic:  64.5 on 8 and 189 DF,  p-value: < 2.2e-16
```

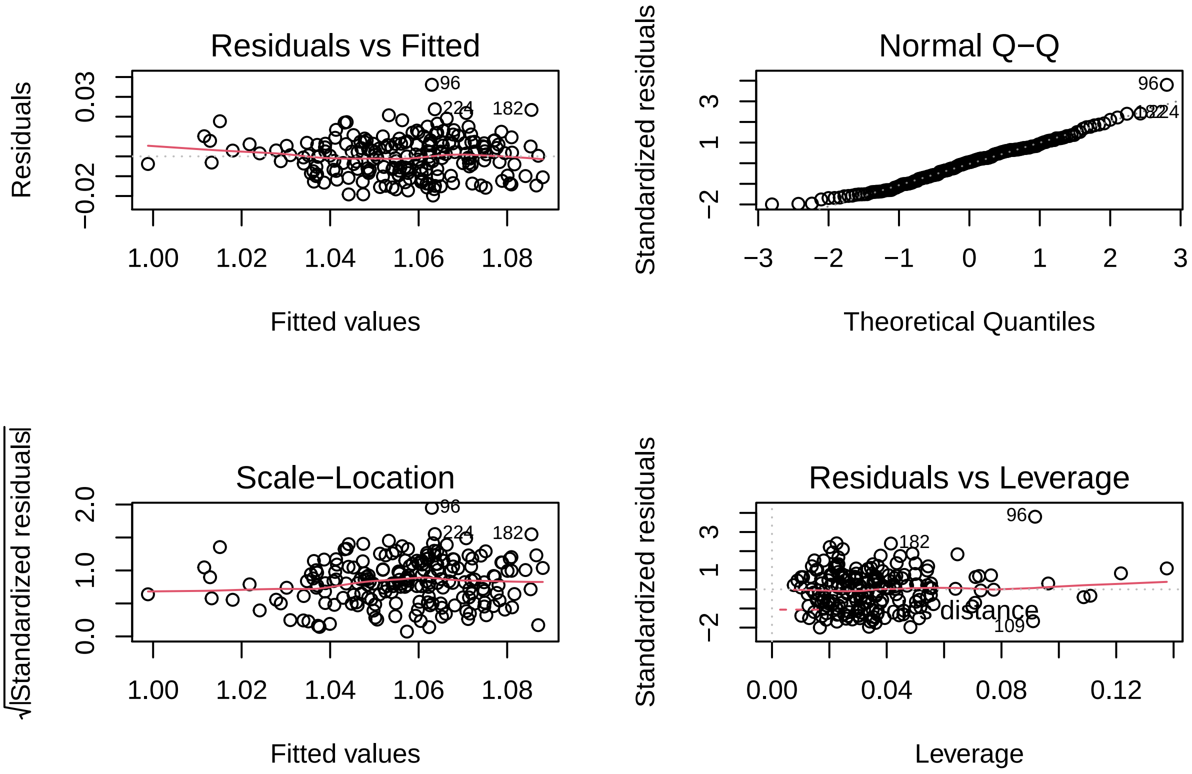
```
summary(cpmod)
```

```
##
## Call:
## lm(formula = as.formula(form), data = train)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.020331 -0.007433  0.000518  0.006422  0.037560
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)         1.181e+00  1.784e-02  66.226  < 2e-16 ***
## Weight              3.201e-04  7.054e-05   4.538 9.99e-06 ***
## ‘Abdomen 2 Circ.‘ -2.272e-03  1.434e-04 -15.845  < 2e-16 ***
## ‘Bicep Circ.‘      -9.009e-04  3.938e-04  -2.288  0.02324 *
## ‘Wrist Circ.‘       3.102e-03  1.104e-03   2.809  0.00548 **
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
```

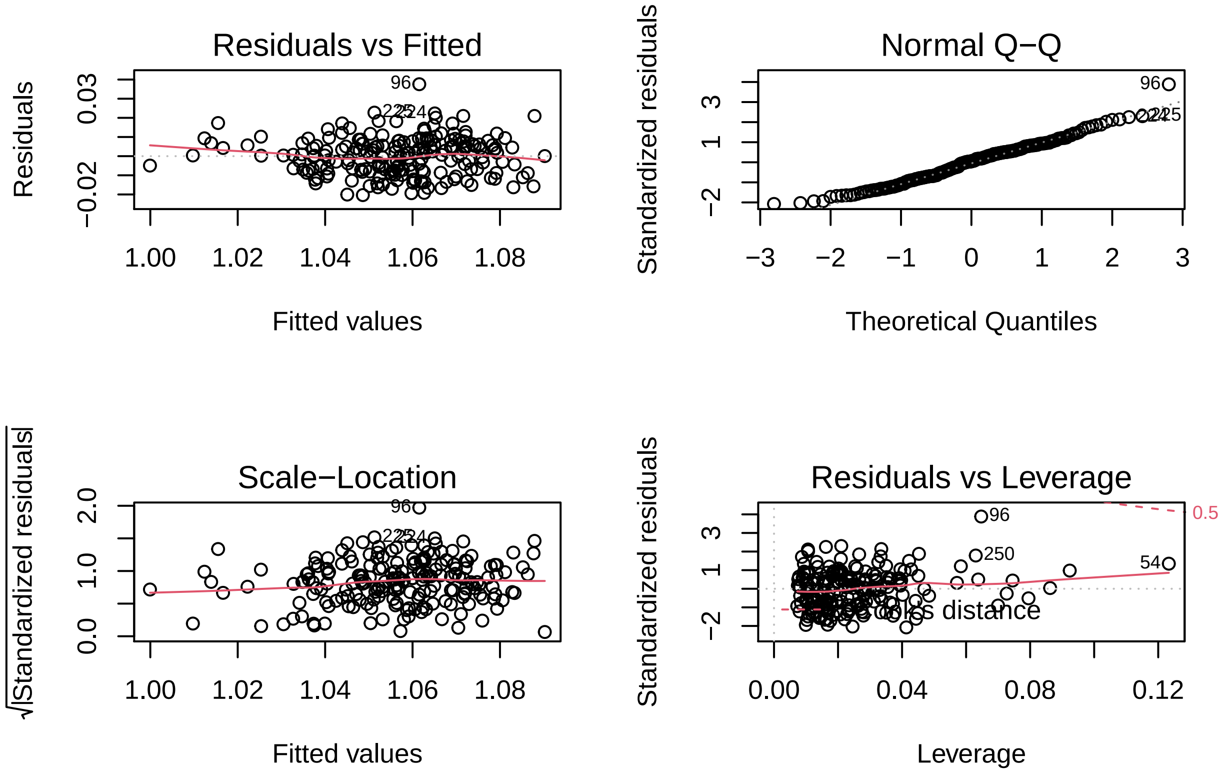
```
##
## Residual standard error: 0.01 on 193 degrees of freedom
## Multiple R-squared:  0.7239, Adjusted R-squared:  0.7182
## F-statistic: 126.5 on 4 and 193 DF,  p-value: < 2.2e-16
```

These models were selected based upon the best adjusted R-squared, Mallow's CP, Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC) respectively. As we did with the naive model, let's take a look at these models diagnostic plots:

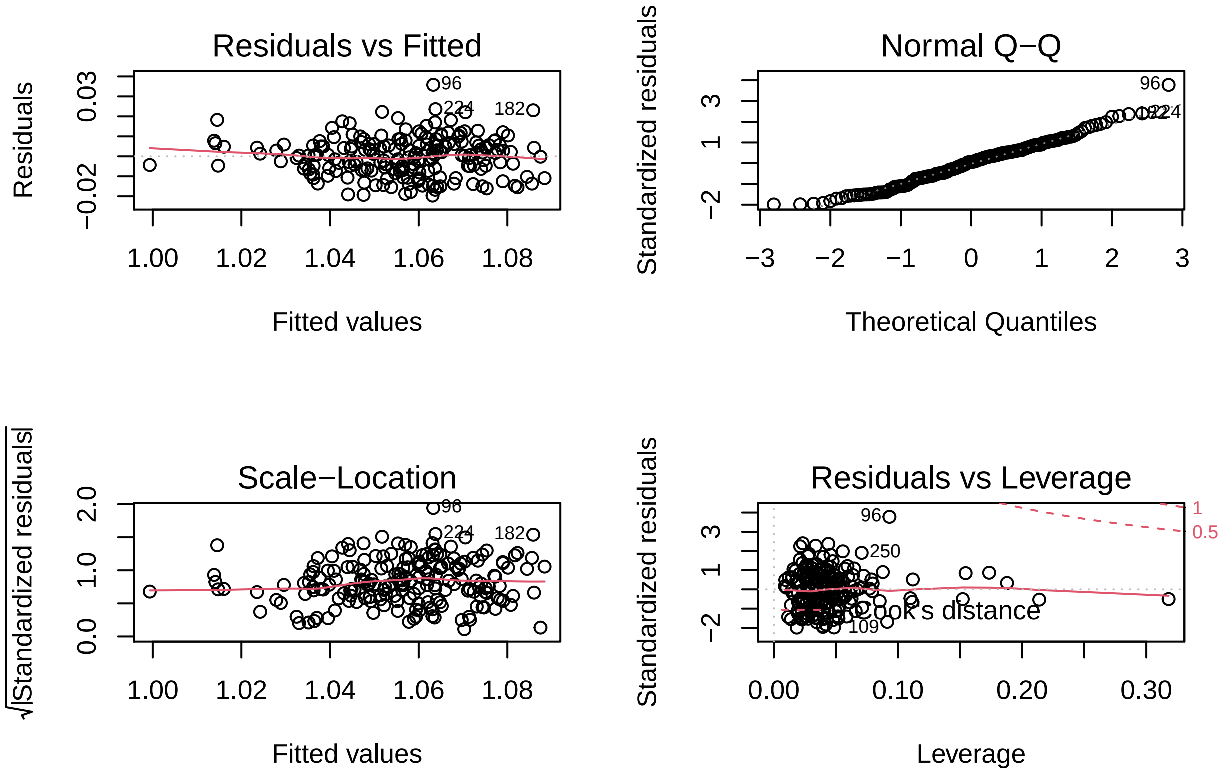
```
par(mfrow=c(2,2))
plot(backsel)
```



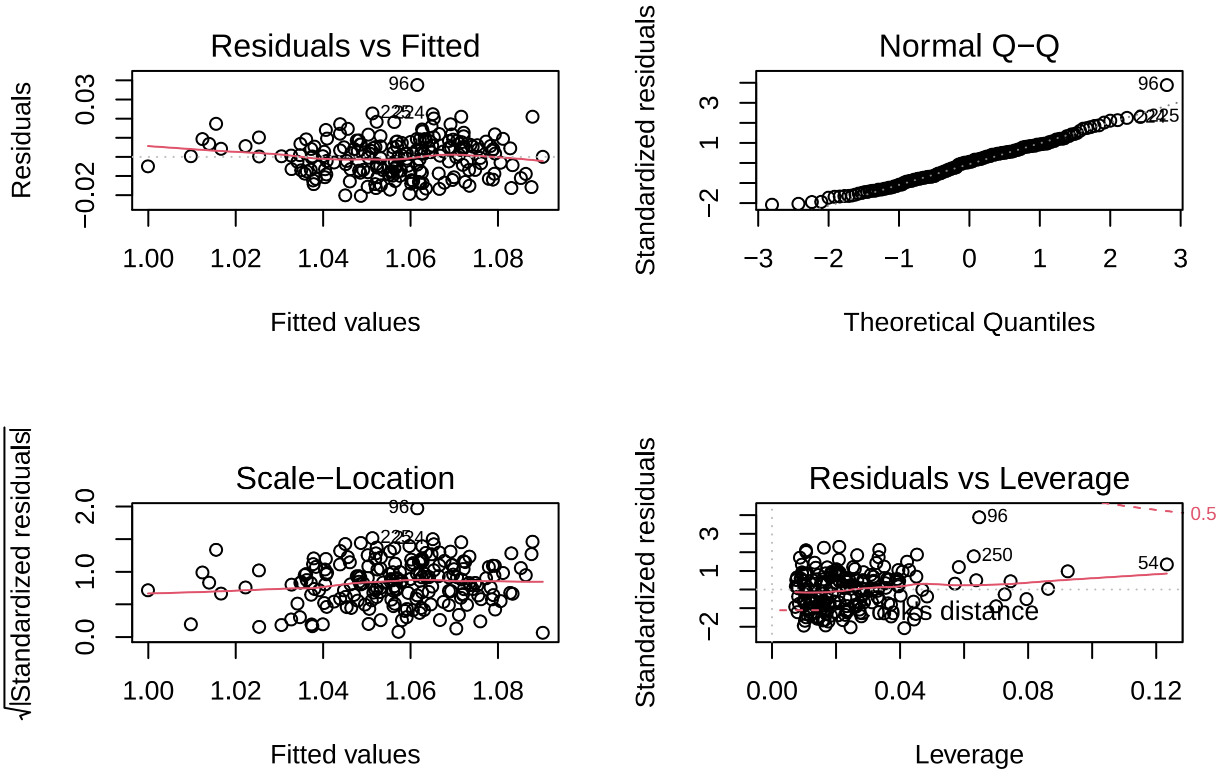
```
# plot(bothsel)
plot(bicmod)
plot(aicmod)
```



```
plot(adjr2mod)
```



```
plot(cpmod)
```



We can see that all the plots indicate constant variance and no overly significant outliers, though indicate there may be non-normally distributed residuals.

```
library(lmtest)
shapiro.test(backsel$residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  backsel$residuals
## W = 0.98469, p-value = 0.03011
```

```
bptest(backsel)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  backsel
## BP = 9.17, df = 6, p-value = 0.1642
```

```
# shapiro.test(bothsel$residuals)
# bptest(bothsel)
shapiro.test(bicmod$residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  bicmod$residuals
## W = 0.98543, p-value = 0.03872
```

```
bptest(bicmod)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  bicmod
## BP = 4.8075, df = 4, p-value = 0.3076
```

```
shapiro.test(aicmod$residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  aicmod$residuals
## W = 0.98543, p-value = 0.03872
```

```
bptest(aicmod)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  aicmod
## BP = 4.8075, df = 4, p-value = 0.3076
```

```
shapiro.test(adjr2mod$residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  adjr2mod$residuals
## W = 0.98571, p-value = 0.04261
```

```
bptest(adjr2mod)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  adjr2mod
## BP = 9.9807, df = 8, p-value = 0.2664
```

```
shapiro.test(cpmod$residuals)
```

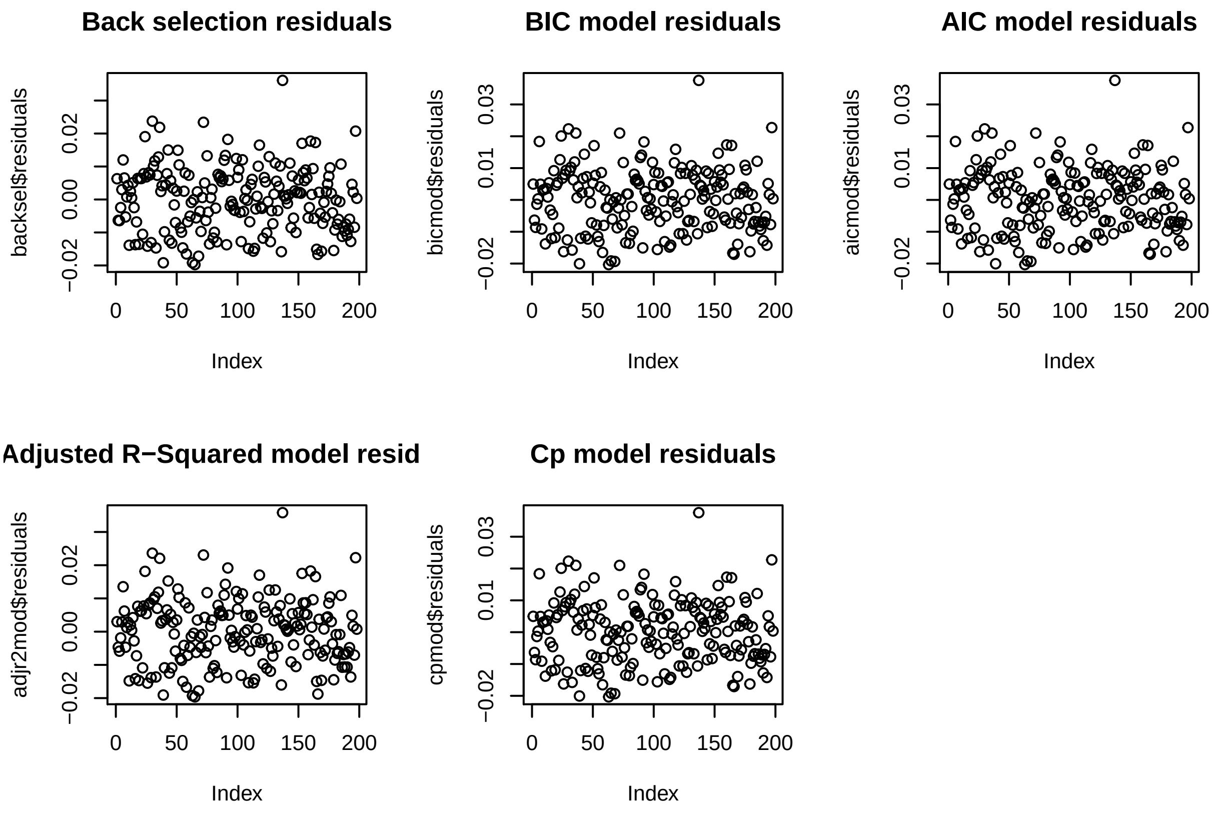
```
##
##  Shapiro-Wilk normality test
##
## data:  cpmod$residuals
## W = 0.98543, p-value = 0.03872
```

```
bptest(cpmod)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  cpmod
## BP = 4.8075, df = 4, p-value = 0.3076
```

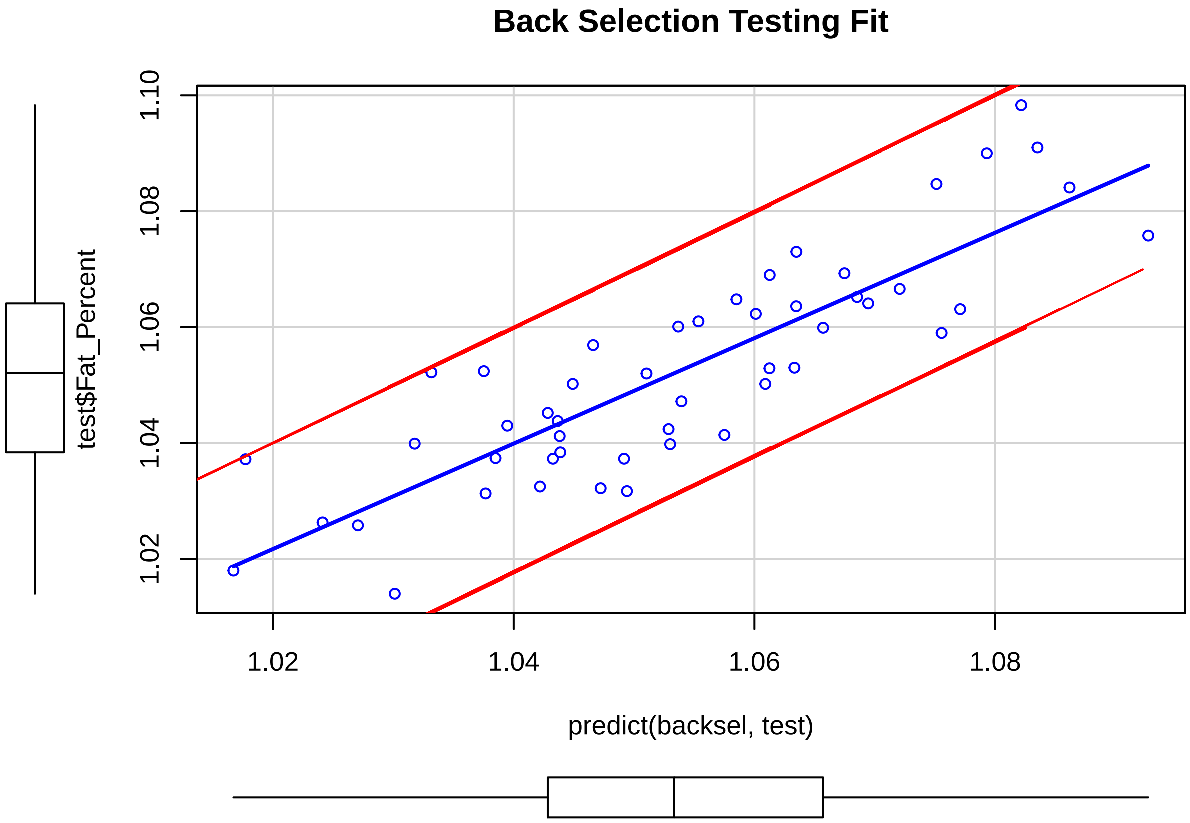
In addition, we can see that the Breusch-Pagan and Shapiro-Wilks tests (tests for constant variance and normality of residuals respectively) fail to reject the null hypothesis and reject the null hypothesis respectively at the 5% significance level. This indicates that the residuals of the model have constant variance, though are not normally distributed (face some level of correlation)

```
par(mfrow=c(2,3))
plot(backsel$residuals, main="Back selection residuals")
plot(bicmod$residuals, main="BIC model residuals")
plot(aicmod$residuals, main="AIC model residuals")
plot(adjr2mod$residuals, main="Adjusted R-Squared model residuals")
plot(cpmod$residuals, main="Cp model residuals")
```

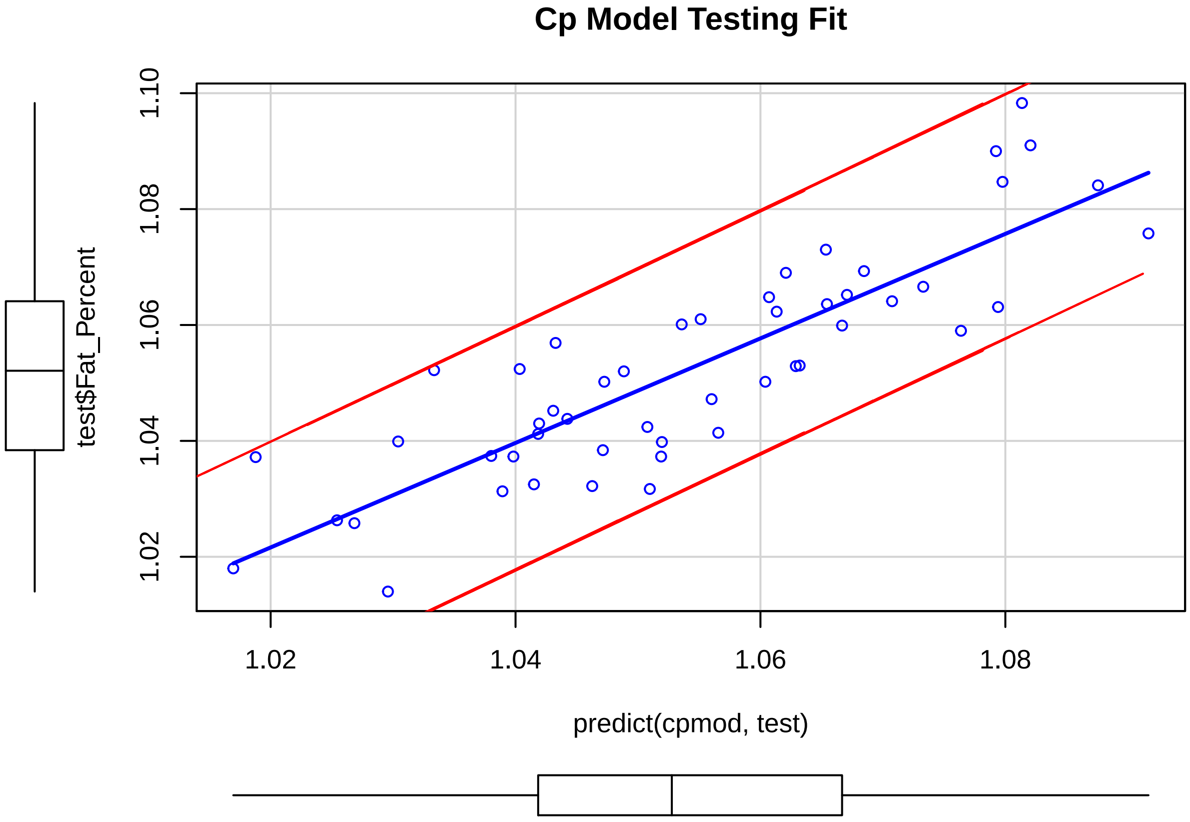


Let's see how these models perfrom on the test data:

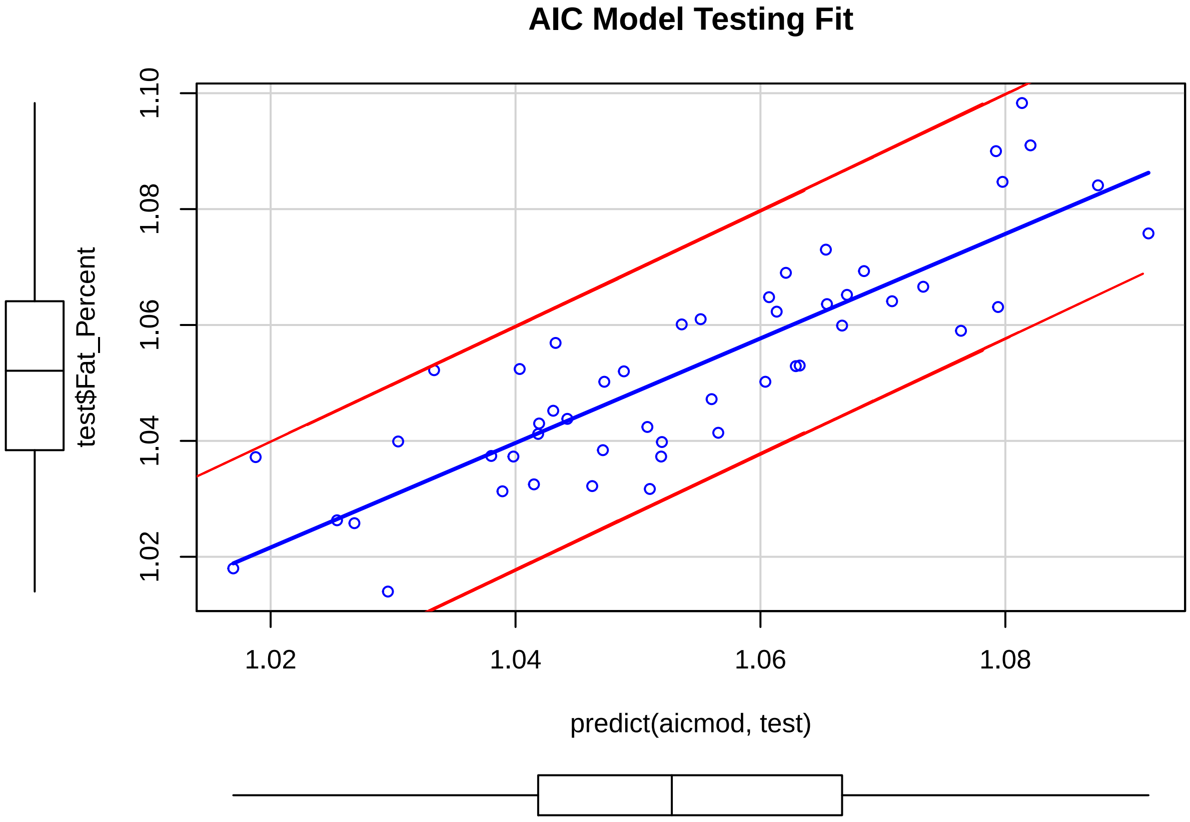
```
scatterplot(predict(backsel, test), test$Fat_Percent, smooth = FALSE, main="Back Selection Testing Fit")
pred_interval <- predict(backsel, newdata=test, interval="prediction", level = 0.95)
lines(pred_interval[,1], pred_interval[,2], col="red")
lines(pred_interval[,1], pred_interval[,3], col="red")
```



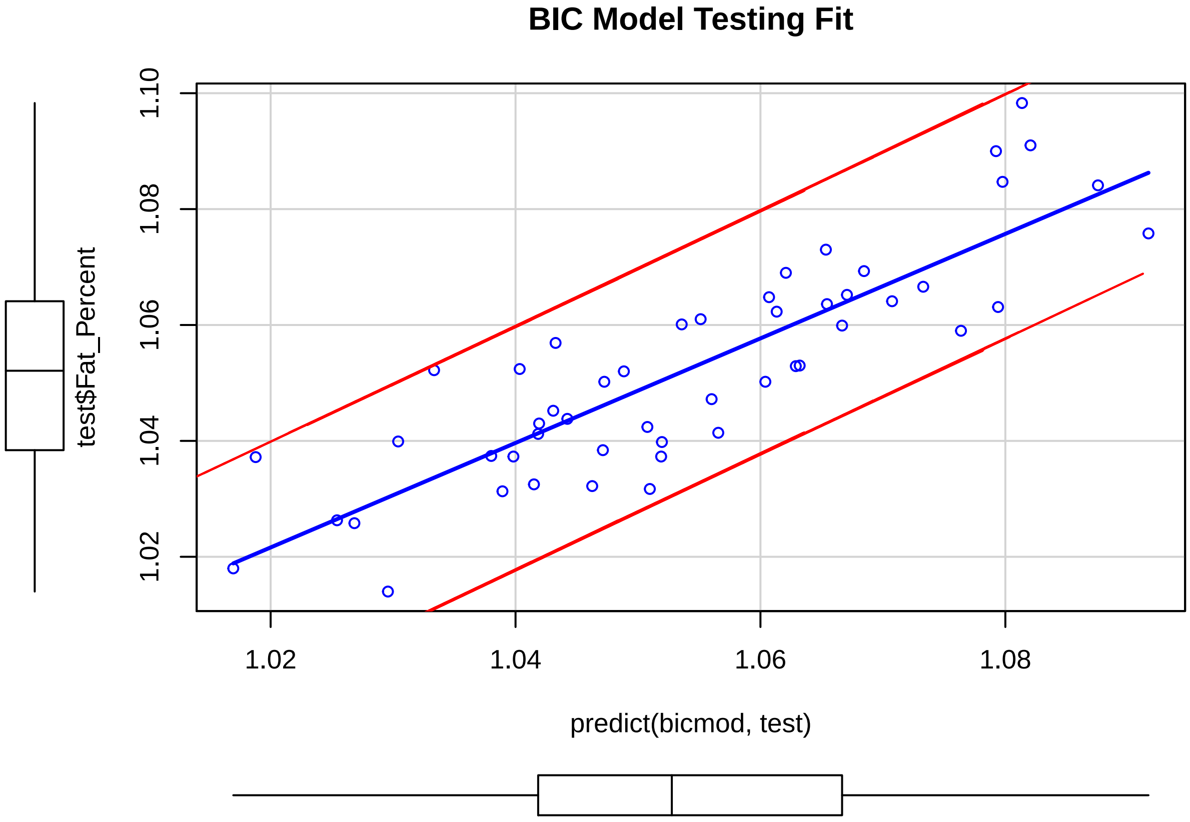
```
scatterplot(predict(cpmod, test), test$Fat_Percent, smooth = FALSE, main="Cp Model Testing Fit")
pred_interval <- predict(cpmod, newdata=test, interval="prediction", level = 0.95)
lines(pred_interval[,1], pred_interval[,2], col="red")
lines(pred_interval[,1], pred_interval[,3], col="red")
```



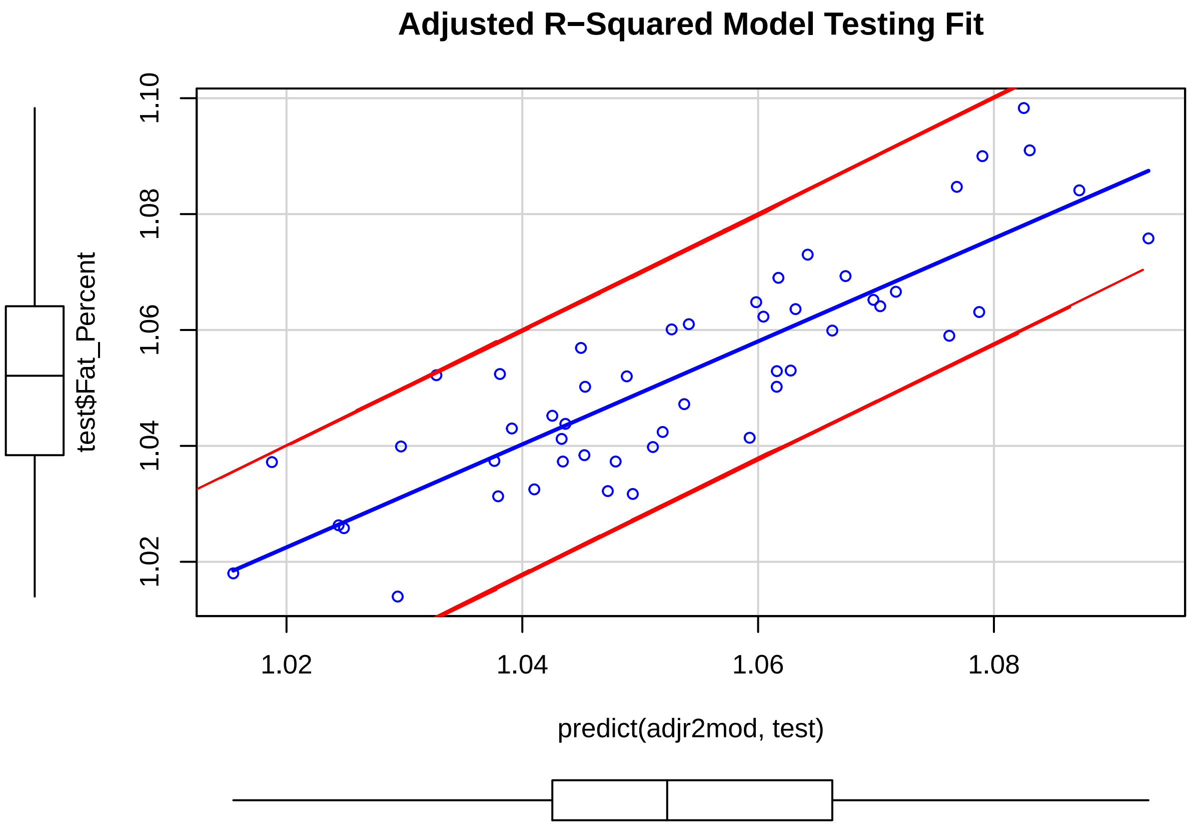
```
scatterplot(predict(aicmod, test), test$Fat_Percent, smooth = FALSE, main="AIC Model Testing Fit")
pred_interval <- predict(aicmod, newdata=test, interval="prediction", level = 0.95)
lines(pred_interval[,1], pred_interval[,2], col="red")
lines(pred_interval[,1], pred_interval[,3], col="red")
```



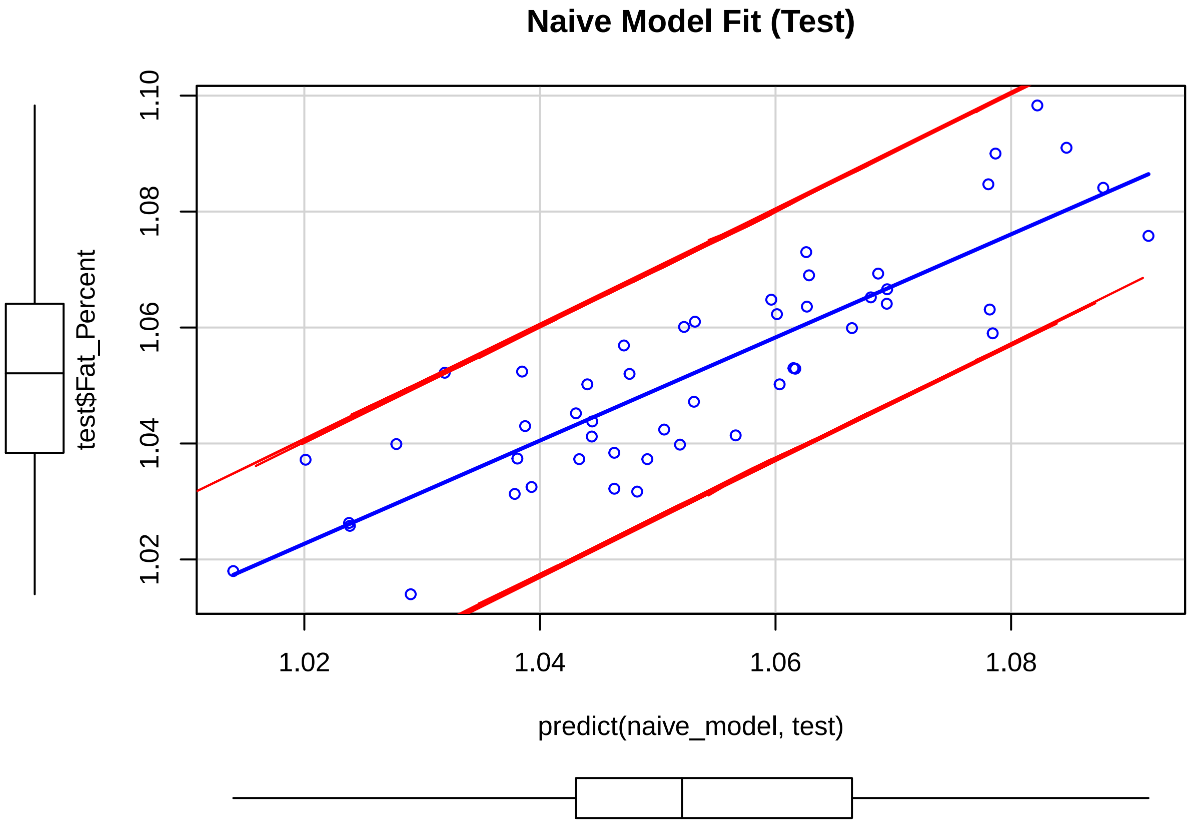
```
scatterplot(predict(bicmod, test), test$Fat_Percent, smooth = FALSE, main="BIC Model Testing Fit")
pred_interval <- predict(bicmod, newdata=test, interval="prediction", level = 0.95)
lines(pred_interval[,1], pred_interval[,2], col="red")
lines(pred_interval[,1], pred_interval[,3], col="red")
```



```
scatterplot(predict(adjr2mod, test), test$Fat_Percent, smooth = FALSE, main="Adjusted R-Squared Model Te
pred_interval <- predict(adjr2mod, newdata=test, interval="prediction", level = 0.95)
lines(pred_interval[,1], pred_interval[,2], col="red")
lines(pred_interval[,1], pred_interval[,3], col="red")
```



```
scatterplot(predict(naive_model, test), test$Fat_Percent, smooth = FALSE, main = "Naive Model Fit (Test
pred_interval <- predict(naive_model, newdata=test, interval="prediction", level = 0.95)
lines(pred_interval[,1], pred_interval[,2], col="red")
lines(pred_interval[,1], pred_interval[,3], col="red")
```



We can see that all models predict the data quite well, with all points being within the generated 95% prediction intervals (indicated by red bounds)