HW 3

Philip Lin

10/14/2017

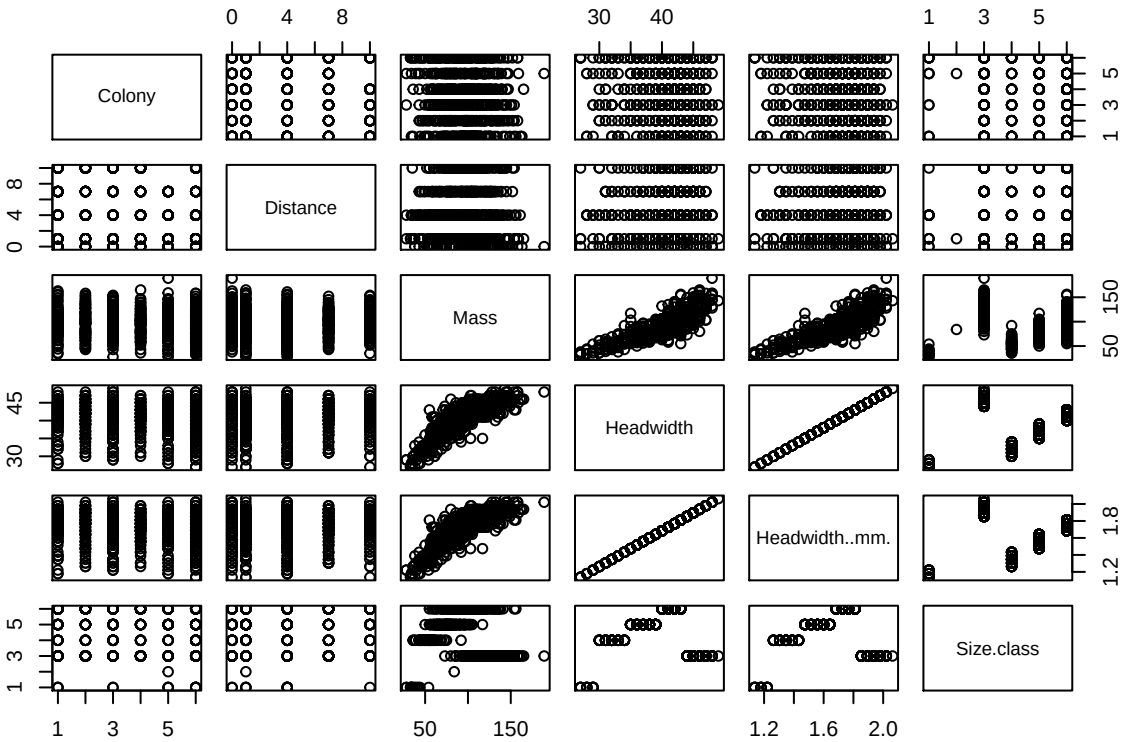
Question 1

```
full = read.csv('thatch-ant.dat.txt')
# full = read.csv('http://www.stat.ucla.edu/projects/datasets/thatch-ant.dat')
```

```
dat = full
for(i in 7:11) {
  dat = dat[dat$Colony != i, ]
}
```

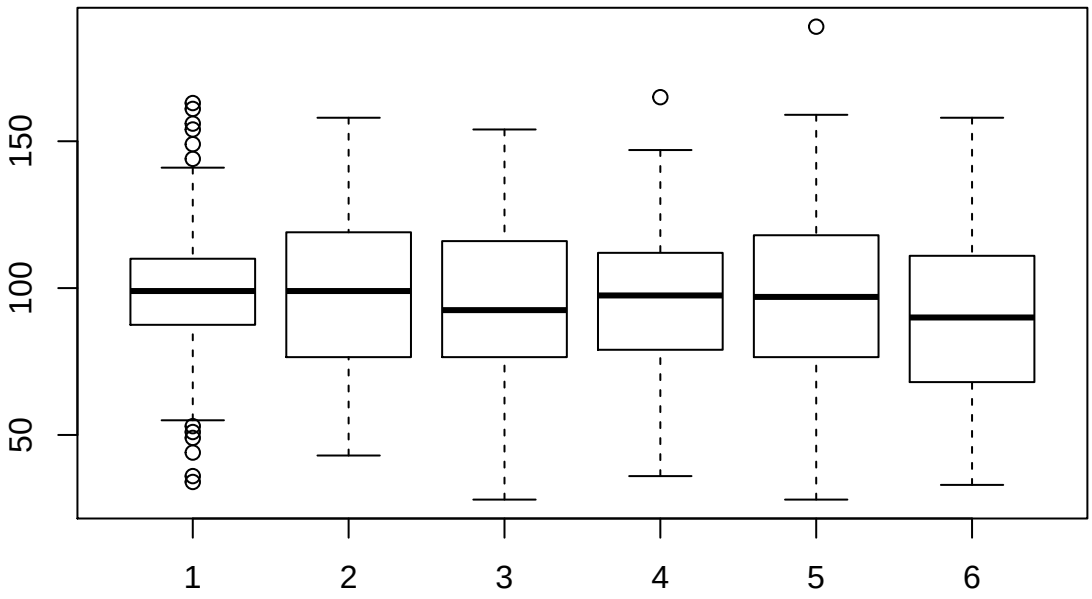
a)

```
pairs(dat)
```

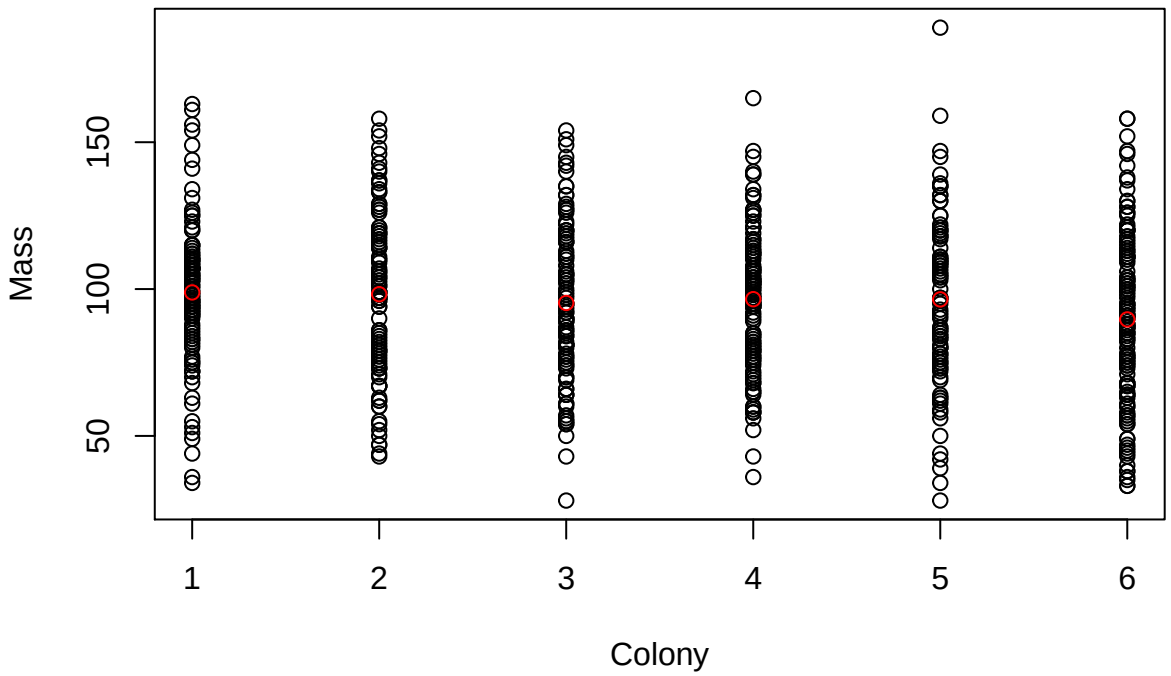


By looking at the pair plot above, we can easily see that mass and headwidth are highly correlated. Moreover, headwidth and headwidth..mm.. are perfectly correlated because they measures the same quantity, but using a different scale.

```
boxplot(Mass ~ Colony, data = dat)
```

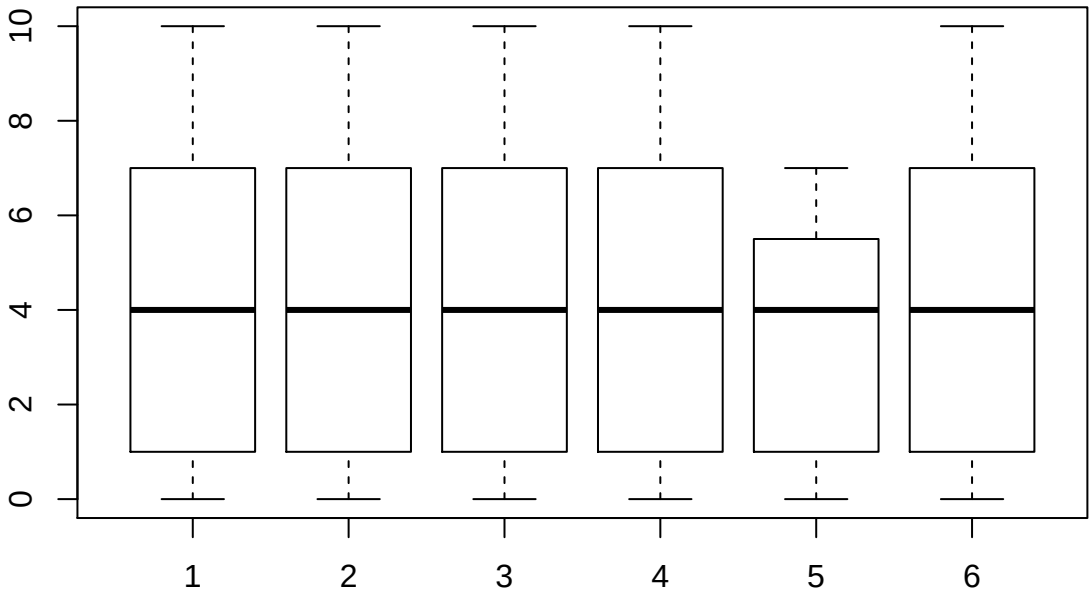



```
means = c()
for(i in 1:6) {
  m = mean(dat[dat$Colony == i, ]$Mass)
  means = c(means, m)
}
plot(dat$Colony, dat$Mass, xlab = 'Colony', ylab = 'Mass')
points(means, col = 'red')
```

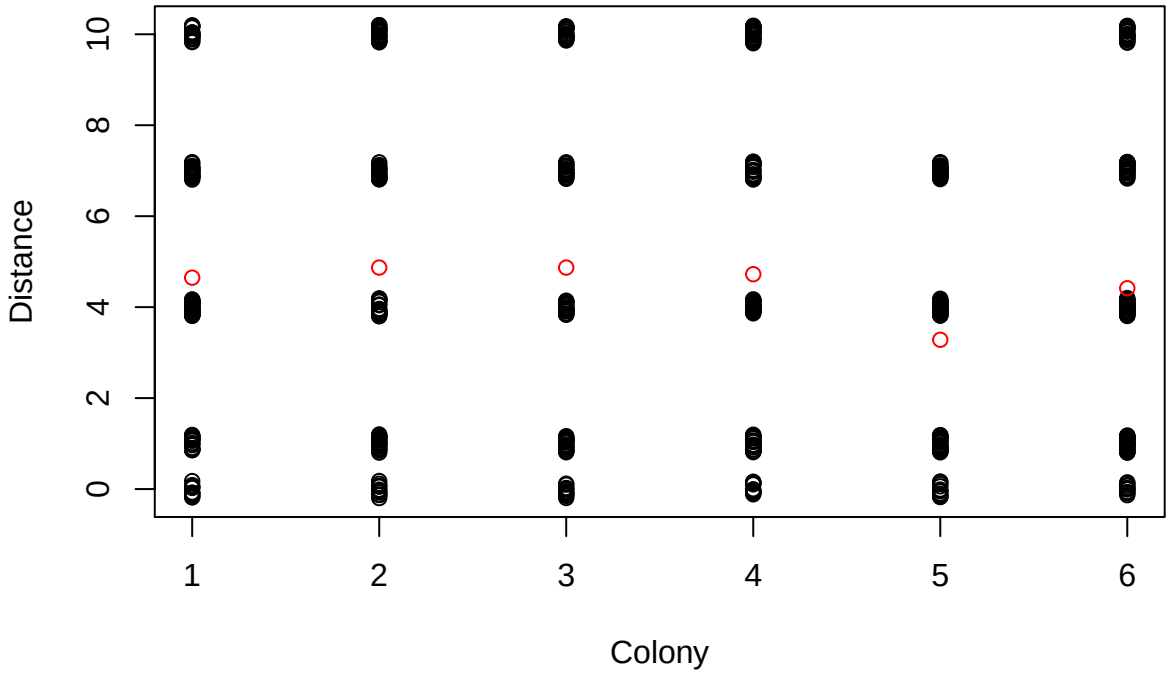


By looking at both boxplot and scatter plot above, we can see colony 6 may have a more energy conservative strategy since the mean of the mass in colony 6 is lower than others, but we don't know yet.

```r
boxplot(Distance ~ Colony, data = dat)
```



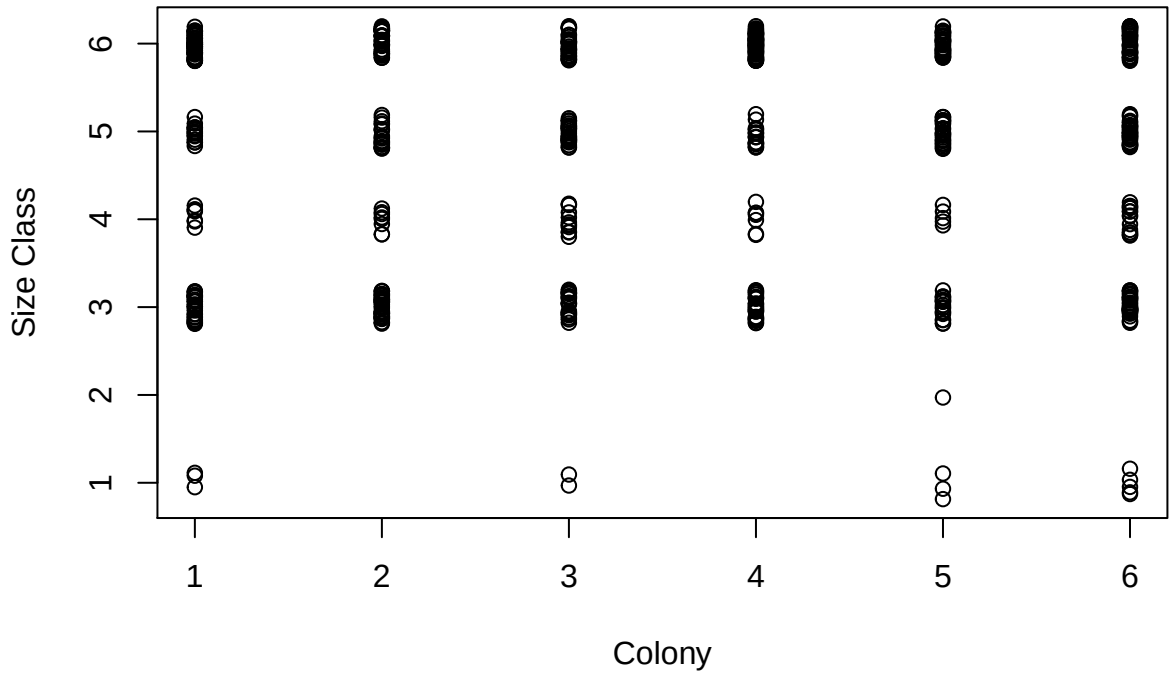

```r
means = c()
for(i in 1:6) {
  m = mean(dat[dat$Colony == i, ]$Distance)
  means = c(means, m)
}
plot(dat$Colony, jitter(dat$Distance), xlab = 'Colony', ylab = 'Distance')
points(means, col = 'red')
```



By looking at the plots above, we can discover that colony 5 do no go too far to forage food.

```r
original_levels = levels(dat$Size.class)
```

```
plot(dat$Colony, jitter(as.numeric(dat$Size.class)), xlab = 'Colony', ylab = 'Size Class')
```



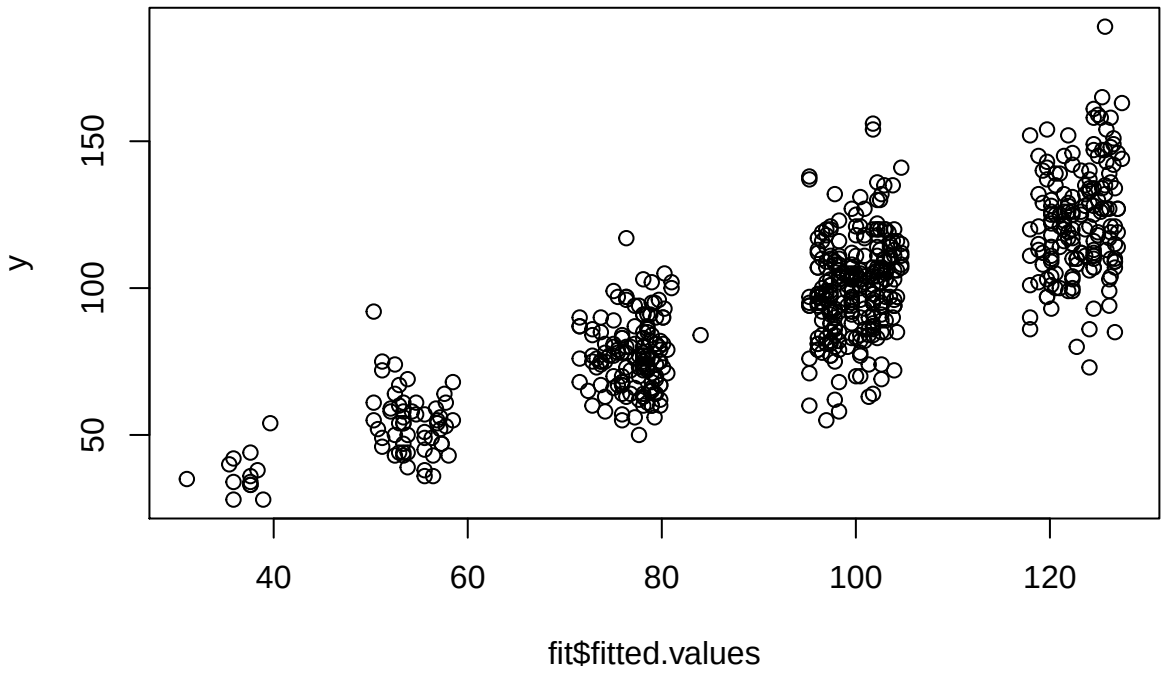

```
print(original_levels)
```

```
## [1] "<30"   "\x80"  ">43"   "30-34" "35-39" "40-43"
```

In the y axis, 1 represents < 30, 2 represents N/A, 3 represents > 43, 4 represents 30-40, 5 represents 35-39, and 6 represents 40-43.

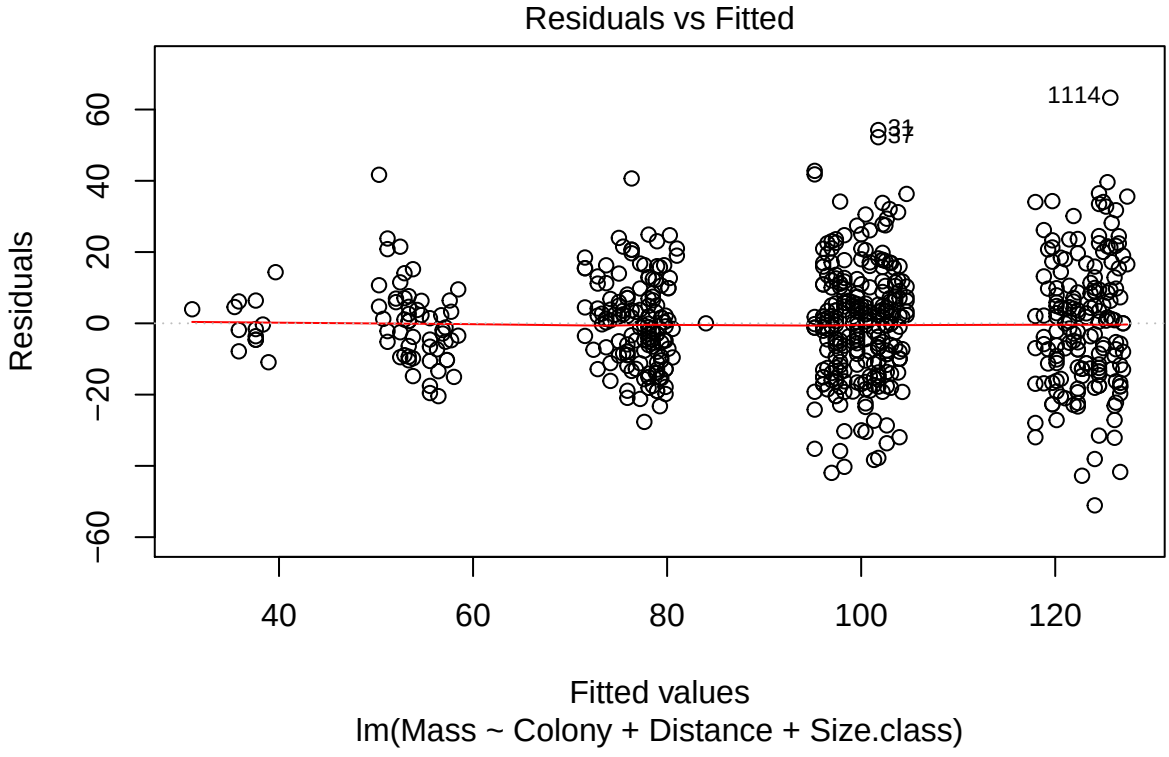
b)

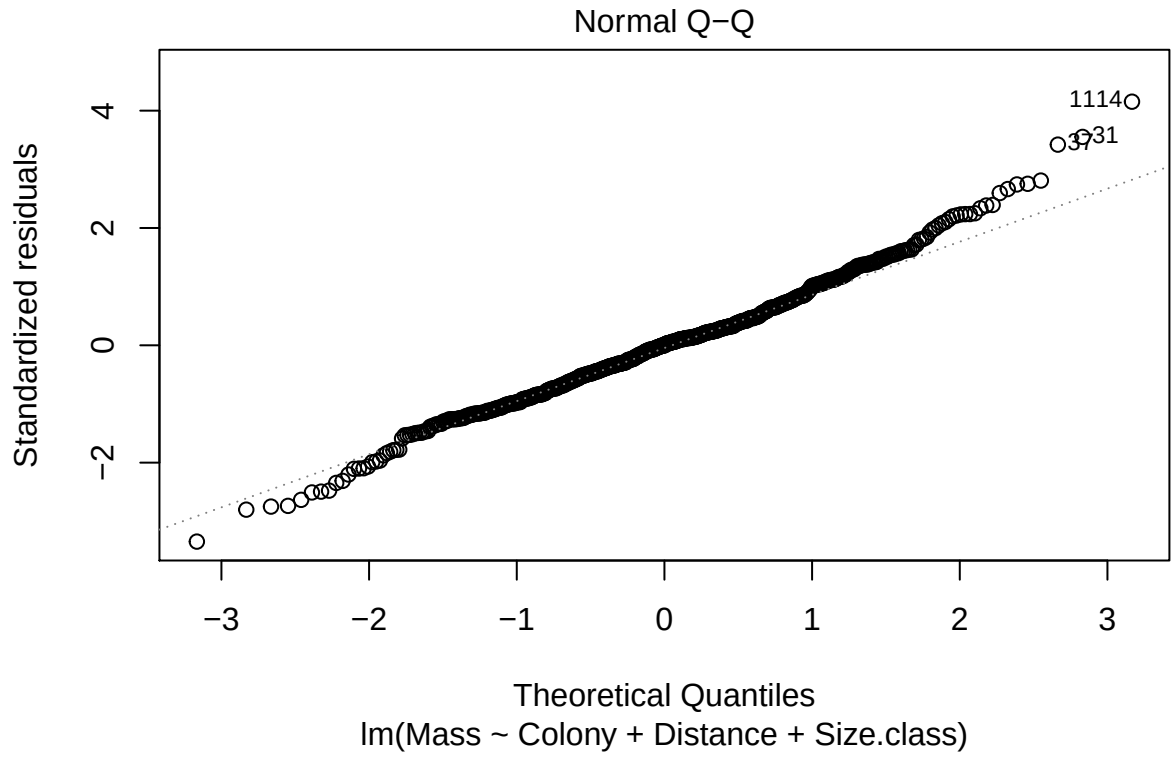
```
y = dat$Mass
fit = lm(Mass ~ Colony + Distance + Size.class, data = dat)
plot(y ~ fit$fitted.values)
```



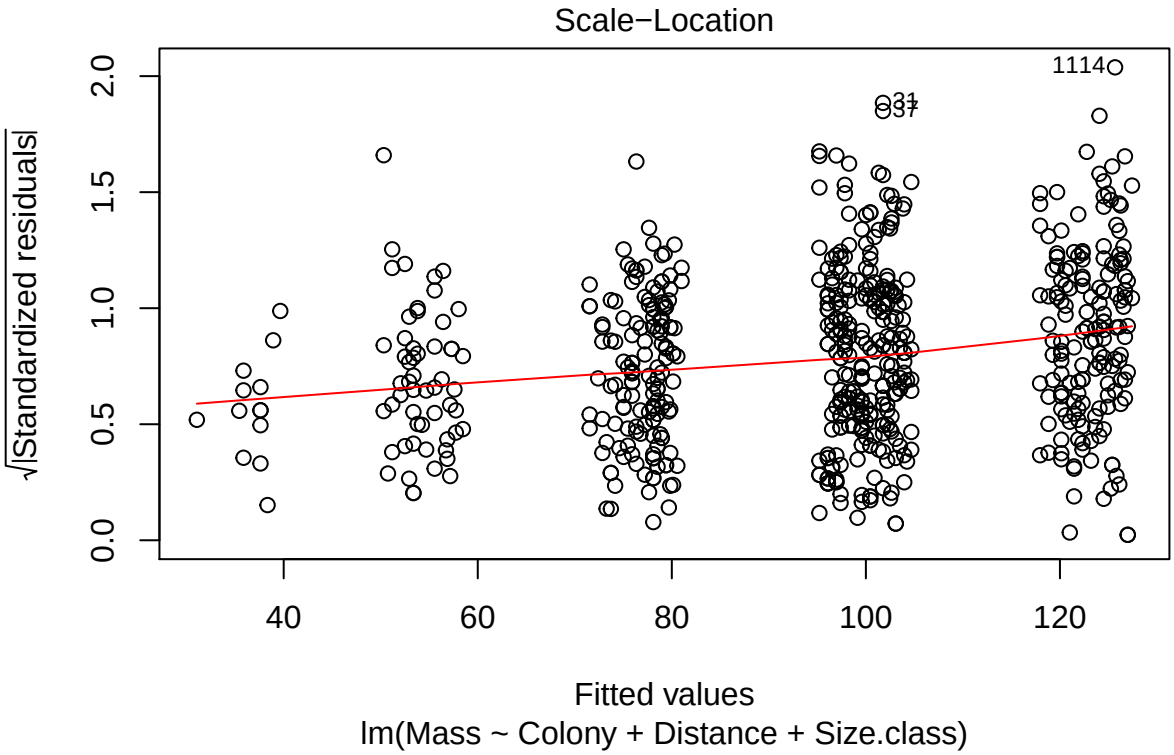

```
plot(fit)
```

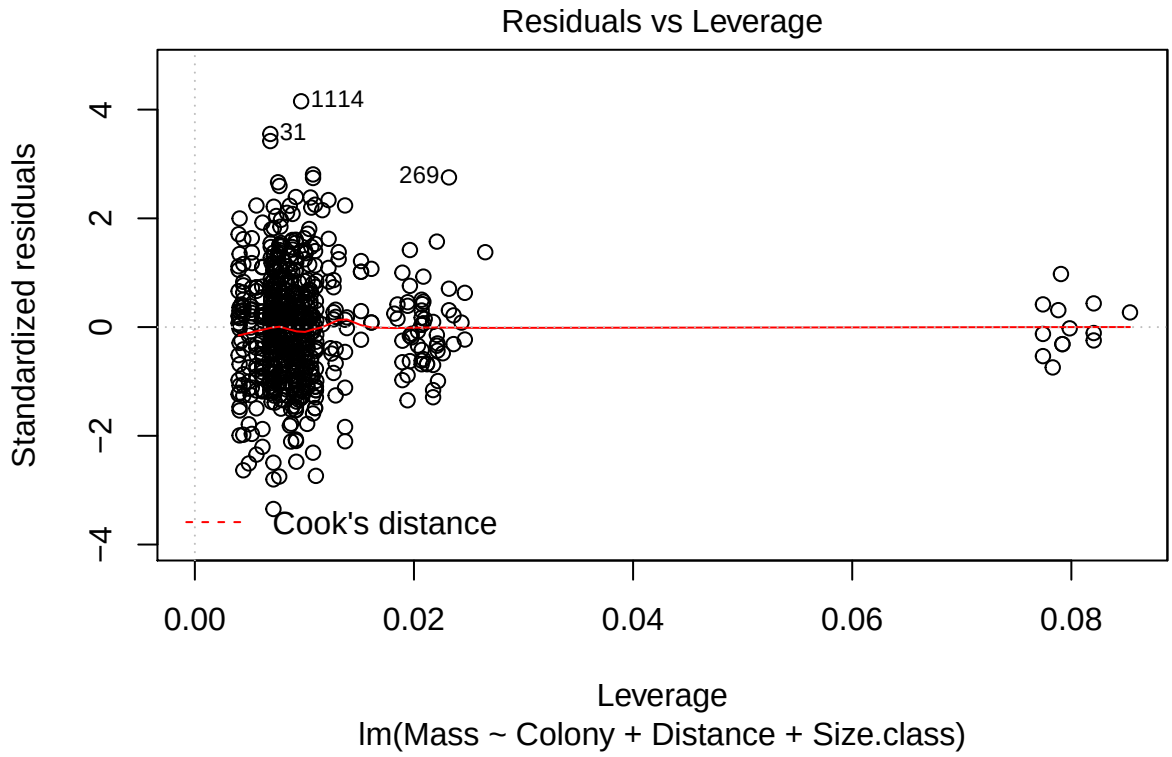
```
## Warning: not plotting observations with leverage one:
##   590
```



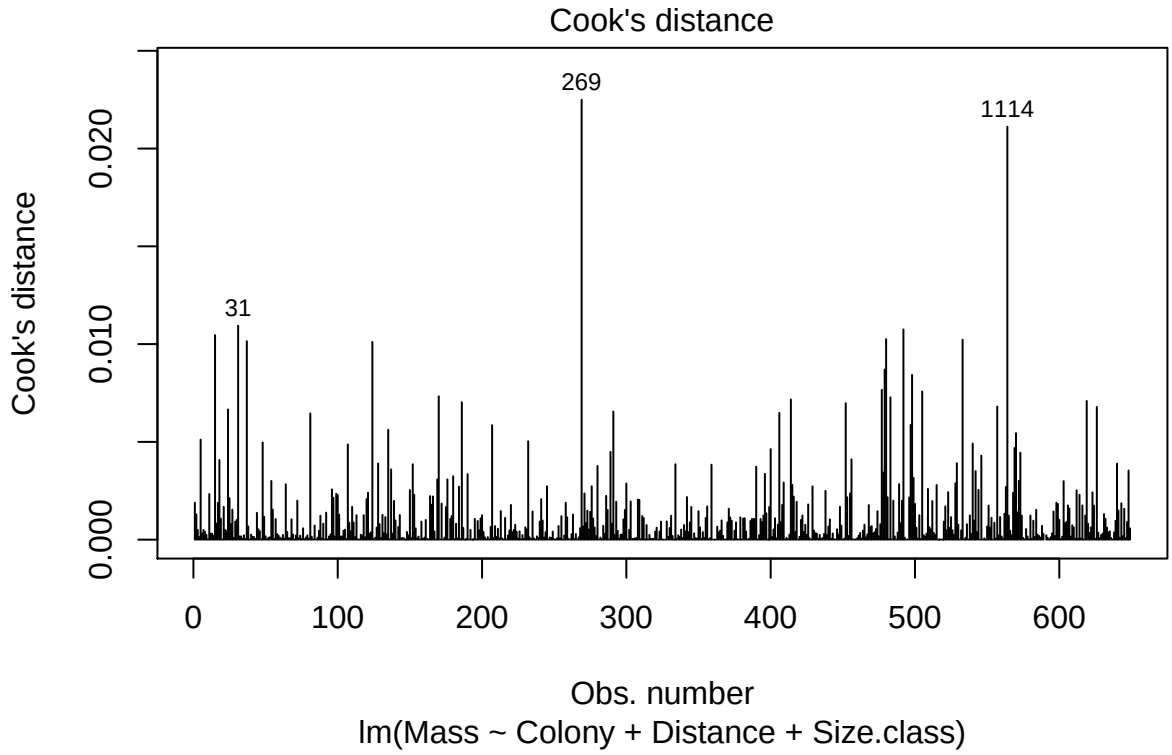



```
## Warning: not plotting observations with leverage one:
##   590
```






```
plot(fit, which = 4)
```



By looking at the residual plot, we can see as the fitted value goes up, the variance of residuals also goes up. This suggest that there might be a better fit other than linear fit to capture this pattern.

By looking at the cook's distance plot, although we can see there are two observations seem to have "high"

cook's distances, they are actually less than 0.03. Therefore, I will say there those two points also have low influential factor.

c)

```
sums = list()
for(i in 1:6) {
  dat_con = dat[dat$Colony == i, ]
  fit_con = lm(Mass ~ Distance + Size.class, data = dat_con)
  sums[[i]] = summary(fit_con)
}

sums[[1]]
```

```
##
## Call:
## lm(formula = Mass ~ Distance + Size.class, data = dat_con)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -37.997  -9.731  -2.150   6.740  53.435
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)      42.4217     8.9736   4.727 7.10e-06 ***
## Distance         -1.1054     0.4351  -2.541   0.0125 *
## Size.class>43    81.6804     9.2818   8.800 2.98e-14 ***
## Size.class30-34  19.1652    10.5336   1.819   0.0717 .
## Size.class35-39  45.7825     9.6456   4.746 6.57e-06 ***
## Size.class40-43  64.5648     9.0351   7.146 1.23e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15.25 on 105 degrees of freedom
## Multiple R-squared:  0.6163, Adjusted R-squared:  0.598
## F-statistic: 33.73 on 5 and 105 DF,  p-value: < 2.2e-16
```

```
sums[[2]]
```

```
##
## Call:
## lm(formula = Mass ~ Distance + Size.class, data = dat_con)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -50.982  -7.501   1.645   9.181  33.968
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     124.04819    3.34865  37.044   <2e-16 ***
## Distance         -0.01648    0.44484  -0.037    0.971
## Size.class30-34 -72.21582    6.20988 -11.629   <2e-16 ***
## Size.class35-39 -50.74812    4.37708 -11.594   <2e-16 ***
## Size.class40-43 -26.55311    3.80506  -6.978    4e-10 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15.88 on 95 degrees of freedom
## Multiple R-squared:  0.6938, Adjusted R-squared:  0.6809
## F-statistic:  53.8 on 4 and 95 DF,  p-value: < 2.2e-16
```

```
sums[[3]]
```

```
##
## Call:
## lm(formula = Mass ~ Distance + Size.class, data = dat_con)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -22.482  -9.803  -0.259   9.055  33.947
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)      41.3209     8.8476   4.670 9.99e-06 ***
## Distance         -0.6417     0.3499  -1.834   0.0698 .
## Size.class>43    91.3101     9.3508   9.765 5.81e-16 ***
## Size.class30-34  23.1488     9.7241   2.381   0.0193 *
## Size.class35-39  39.7282     9.2514   4.294 4.26e-05 ***
## Size.class40-43  64.0409     9.2464   6.926 5.31e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.51 on 94 degrees of freedom
## Multiple R-squared:  0.7966, Adjusted R-squared:  0.7858
## F-statistic: 73.62 on 5 and 94 DF,  p-value: < 2.2e-16
```

```
sums[[4]]
```

```
##
## Call:
## lm(formula = Mass ~ Distance + Size.class, data = dat_con)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -38.393 -10.683   1.153   8.452  41.374
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)     123.8079     3.4355  36.038  < 2e-16 ***
## Distance         -0.1818     0.4190  -0.434    0.665
## Size.class30-34 -71.5743     6.3972 -11.188  < 2e-16 ***
## Size.class35-39 -48.0568     4.5977 -10.452  < 2e-16 ***
## Size.class40-43 -26.1429     3.4999  -7.470 1.96e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15 on 111 degrees of freedom
## Multiple R-squared:  0.6244, Adjusted R-squared:  0.6109
## F-statistic: 46.14 on 4 and 111 DF,  p-value: < 2.2e-16
```

```
sums[[5]]
```

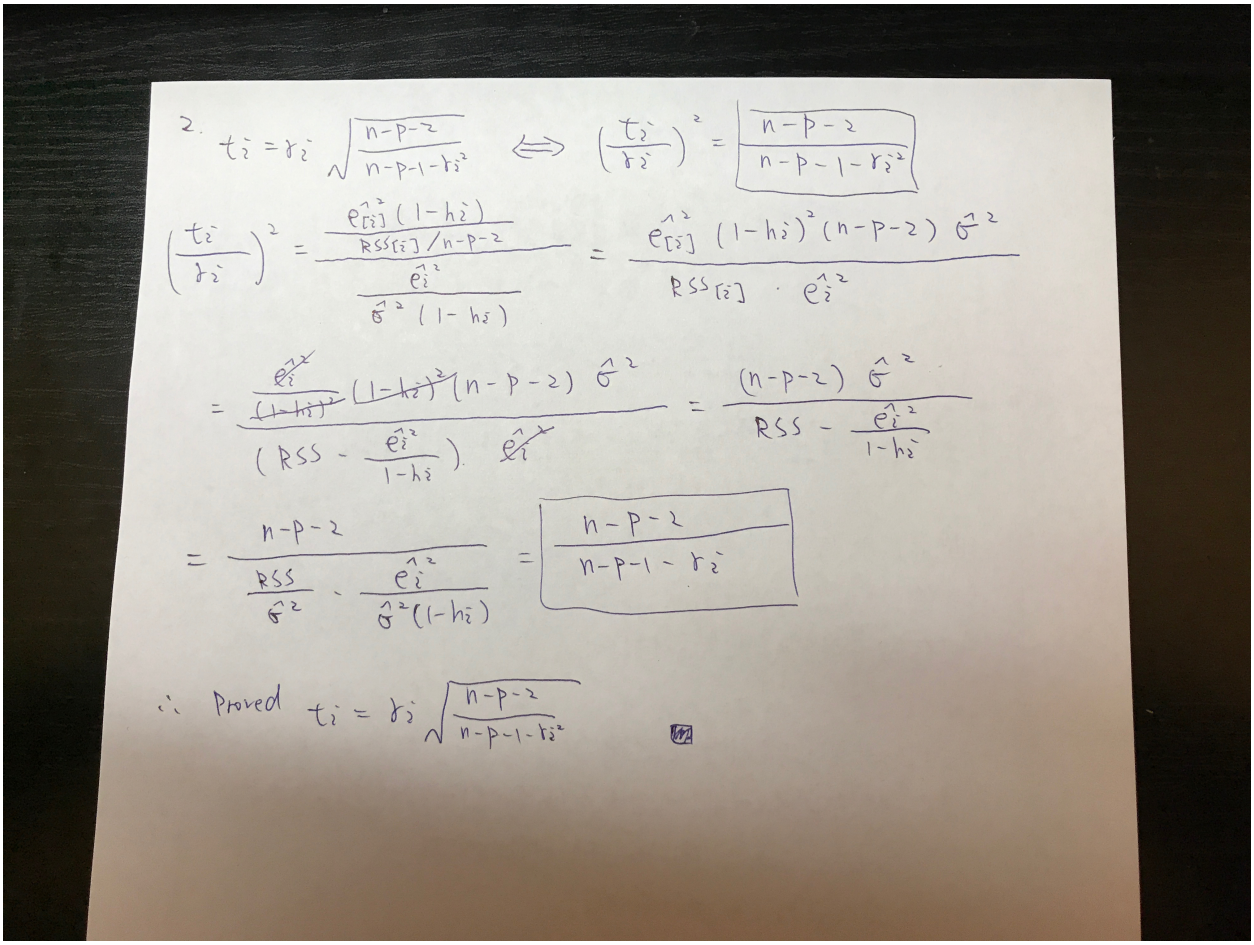
```
##
## Call:
## lm(formula = Mass ~ Distance + Size.class, data = dat_con)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -44.598 -10.456  -0.702  10.003  53.592
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)      45.4765    10.2002   4.458 2.51e-05 ***
## Distance         -2.7024     0.6946  -3.891 0.000198 ***
## Size.class\x80   41.2260    19.7394   2.089 0.039743 *
## Size.class>43    89.9312    10.6189   8.469 6.41e-13 ***
## Size.class30-34  15.3714    12.4270   1.237 0.219518
## Size.class35-39  41.4885    10.3904   3.993 0.000138 ***
## Size.class40-43  69.0708    10.1890   6.779 1.50e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 17 on 85 degrees of freedom
## Multiple R-squared:  0.6844, Adjusted R-squared:  0.6621
## F-statistic: 30.72 on 6 and 85 DF,  p-value: < 2.2e-16
```

```
sums[[6]]
```

```
##
## Call:
## lm(formula = Mass ~ Distance + Size.class, data = dat_con)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -36.867  -8.680  -0.733   7.265  42.811
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)      37.8189     6.8841   5.494 2.14e-07 ***
## Distance         -0.6309     0.4022  -1.568   0.1193
## Size.class>43    84.3313     7.2479  11.635  < 2e-16 ***
## Size.class30-34  15.1787     7.7490   1.959   0.0524 .
## Size.class35-39  40.0884     7.2319   5.543 1.70e-07 ***
## Size.class40-43  63.6790     7.2391   8.796 1.00e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15.12 on 124 degrees of freedom
## Multiple R-squared:  0.7387, Adjusted R-squared:  0.7282
## F-statistic: 70.11 on 5 and 124 DF,  p-value: < 2.2e-16
```

Also, noticed that all the distance coefficients are negative, which means that if we fix other variables, as the mass goes up, the distance goes down. Therefore, I would say ants may have an energy conservative strategy.

Question 2



Question 3

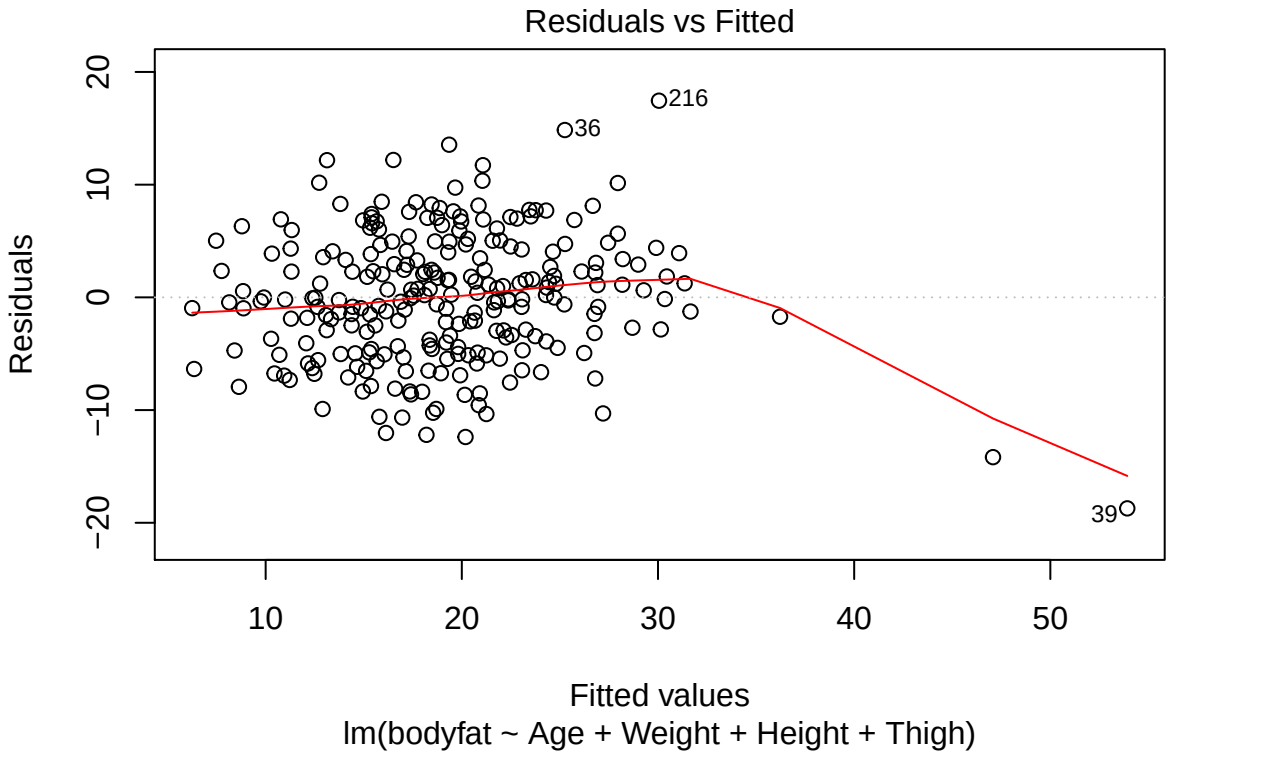
```
dat3 = read.csv('bodyfat.csv')
fit3 = lm(bodyfat ~ Age + Weight + Height + Thigh, data = dat3)
fit3_sum = summary(fit3)
fit3_sum
```

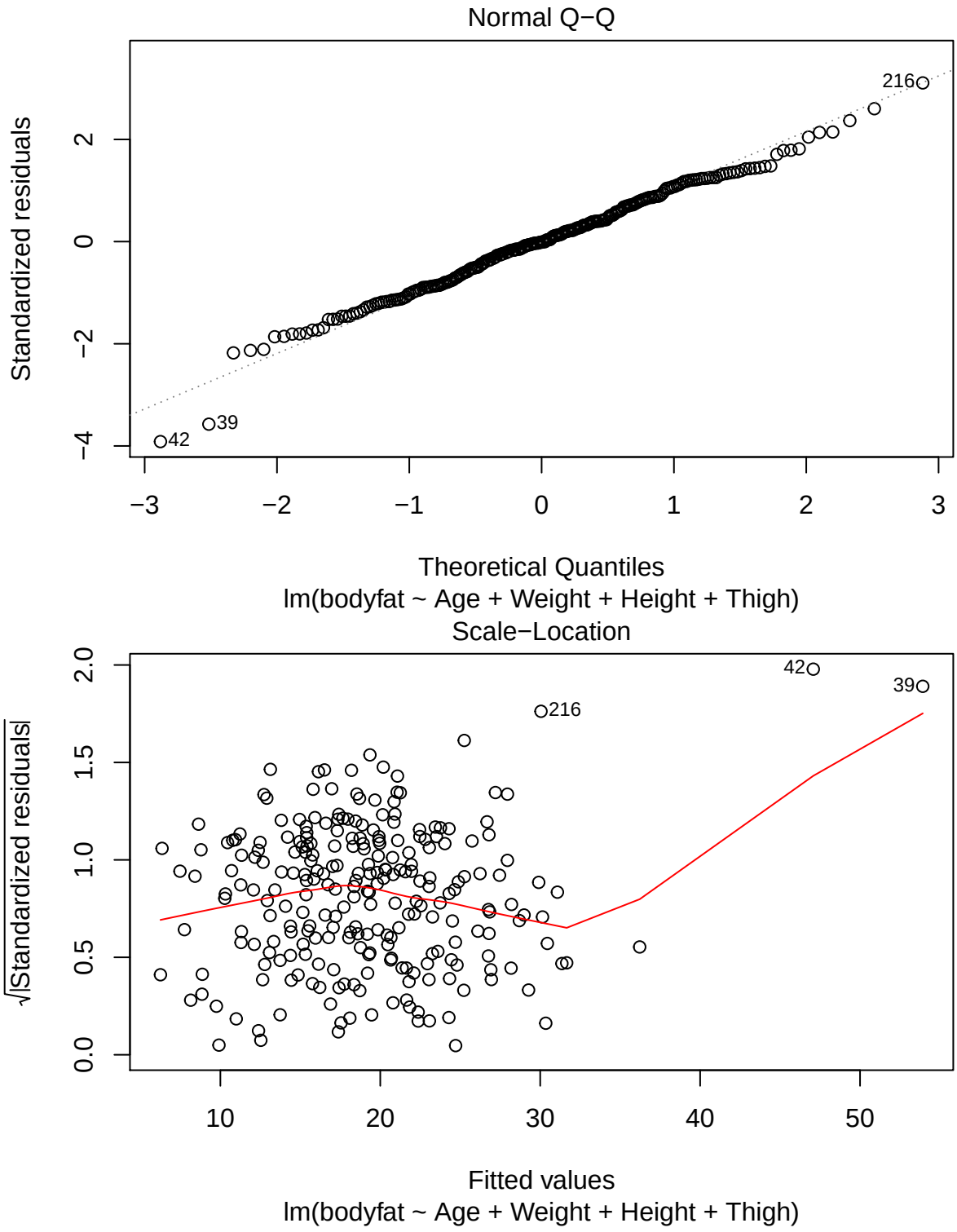
```
##
## Call:
## lm(formula = bodyfat ~ Age + Weight + Height + Thigh, data = dat3)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -18.722  -4.283  -0.055   4.061  17.449
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.27488   11.12642  -0.204   0.8382
## Age          0.20517    0.03274   6.267 1.63e-09 ***
## Weight       0.13417    0.02952   4.545 8.59e-06 ***
```

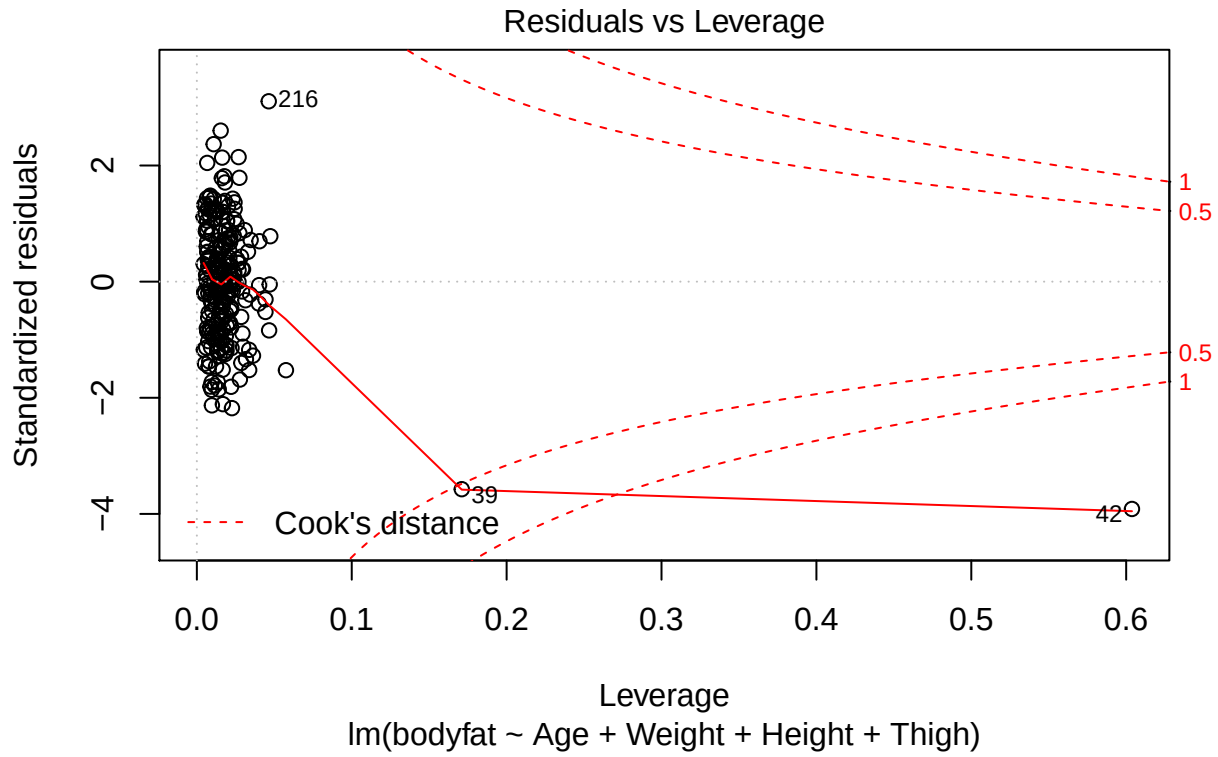
```
## Height      -0.49810    0.11313  -4.403 1.59e-05 ***
## Thigh        0.38970    0.16142   2.414   0.0165 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.753 on 247 degrees of freedom
## Multiple R-squared:  0.5349, Adjusted R-squared:  0.5274
## F-statistic: 71.03 on 4 and 247 DF,  p-value: < 2.2e-16
```

a)

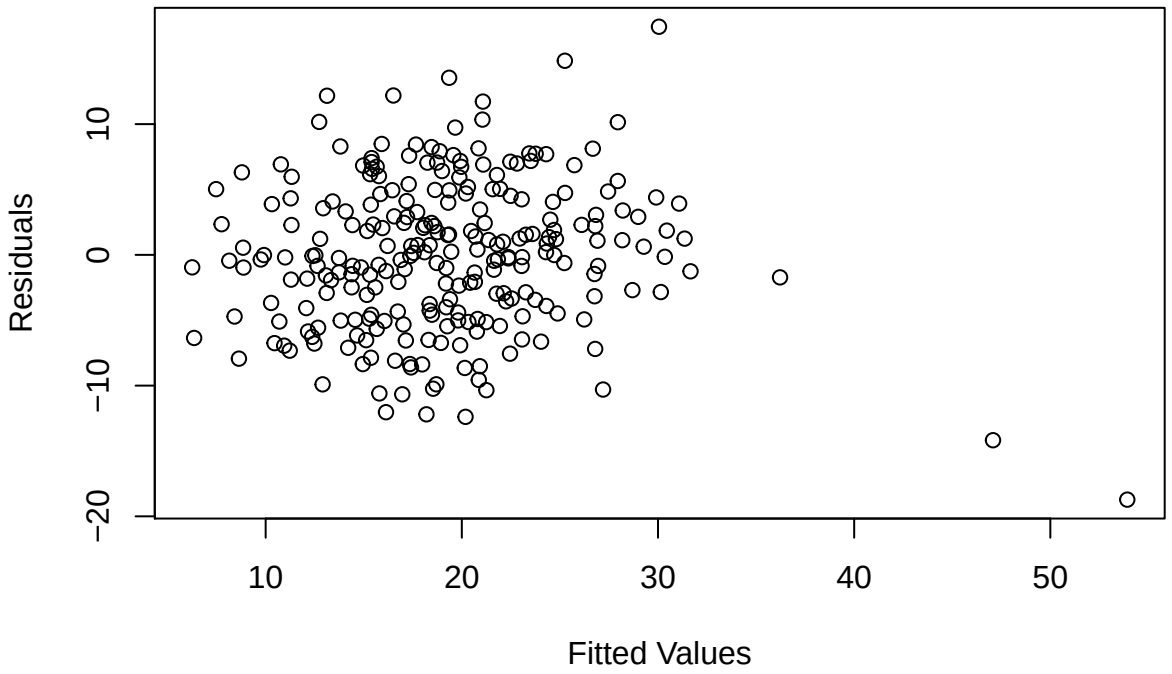
```
plot(fit3)
```




Normal Q−Q
Standardized residuals
216
39
42
−3
−2
−1
0
1
2
3
−4
−2
0
2
Theoretical Quantiles
lm(bodyfat ~ Age + Weight + Height + Thigh)
Scale−Location
√|Standardized residuals|
42
39
216
0.0
0.5
1.0
1.5
2.0
10
20
30
40
50
Fitted values
lm(bodyfat ~ Age + Weight + Height + Thigh)



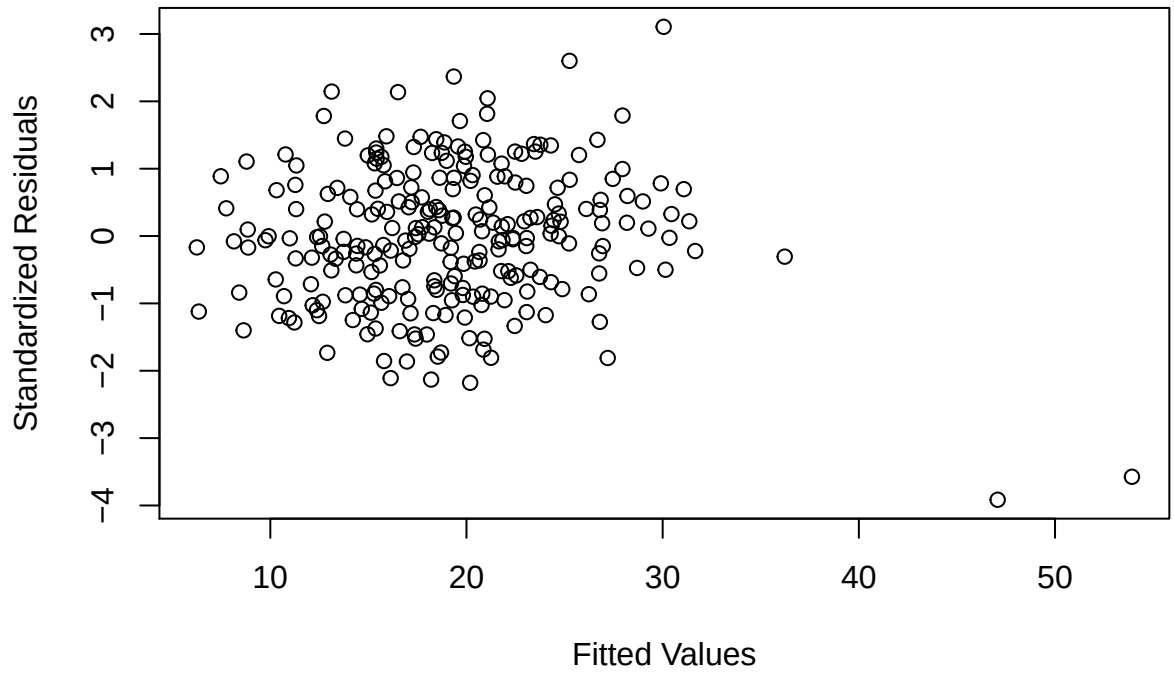

```
plot(fit3$residuals ~ fit3$fitted.values,
     xlab = 'Fitted Values', ylab = 'Residuals')
```



b)

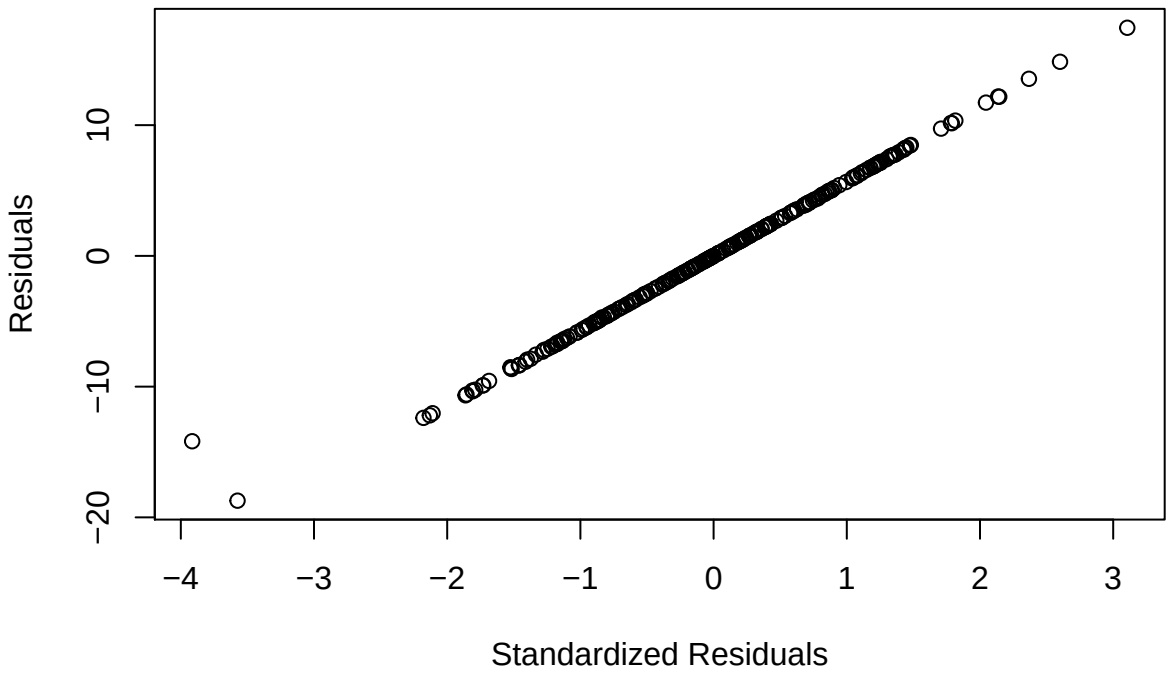
```
X = model.matrix(fit3)
p = ncol(X) - 1
n = nrow(X)
```

```
H = X %*% solve(crossprod(X)) %*% t(X)
std_residuals = fit3$residuals / (fit3_sum$sigma * sqrt(1 - diag(H)))
plot(std_residuals ~ fit3$fitted.values,
     xlab = 'Fitted Values', ylab = 'Standardized Residuals')
```



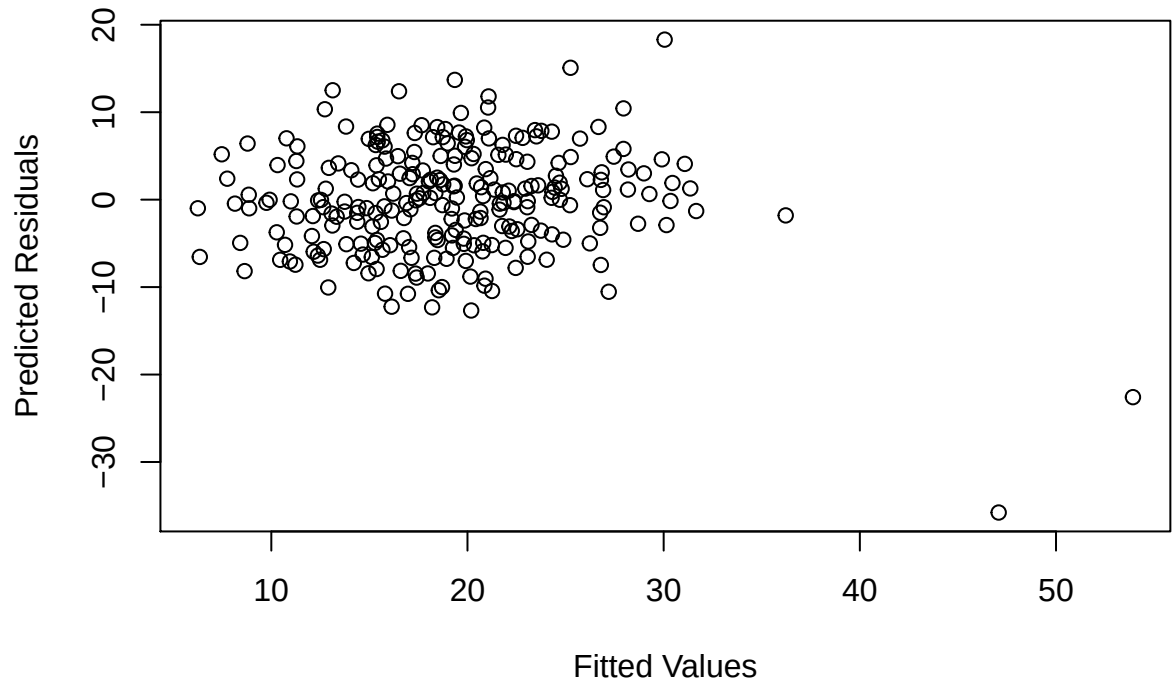
c)

```
plot(fit3$residuals ~ std_residuals,
     xlab = 'Standardized Residuals', ylab = 'Residuals')
```



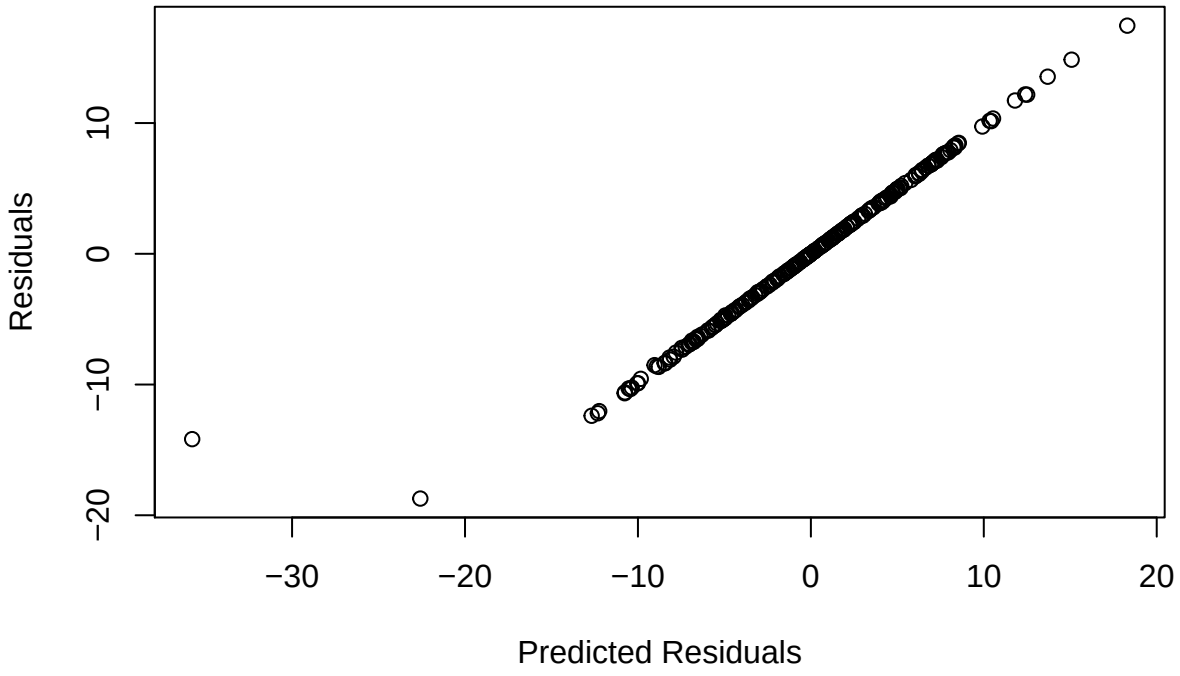
d)

```r
pred_residuals = fit3$residuals / (1 - diag(H))
plot(pred_residuals ~ fit3$fitted.values,
     xlab = 'Fitted Values', ylab = 'Predicted Residuals')
```



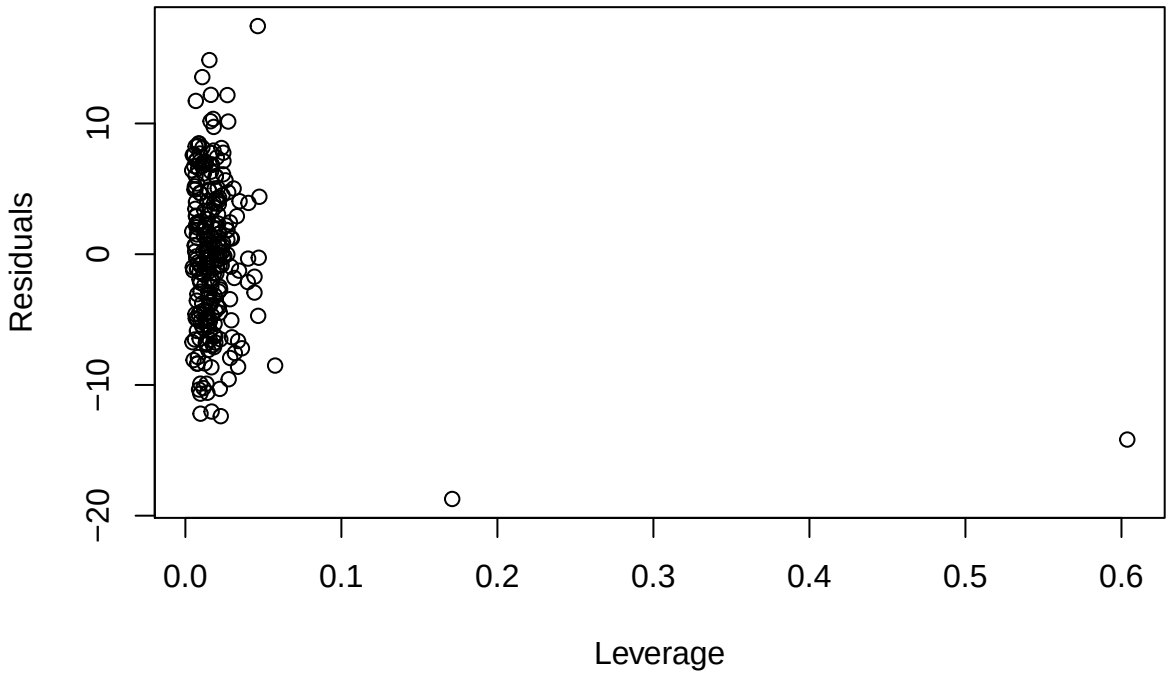
e)

```r
plot(fit3$residuals ~ pred_residuals,
     xlab = 'Predicted Residuals', ylab = 'Residuals')
```



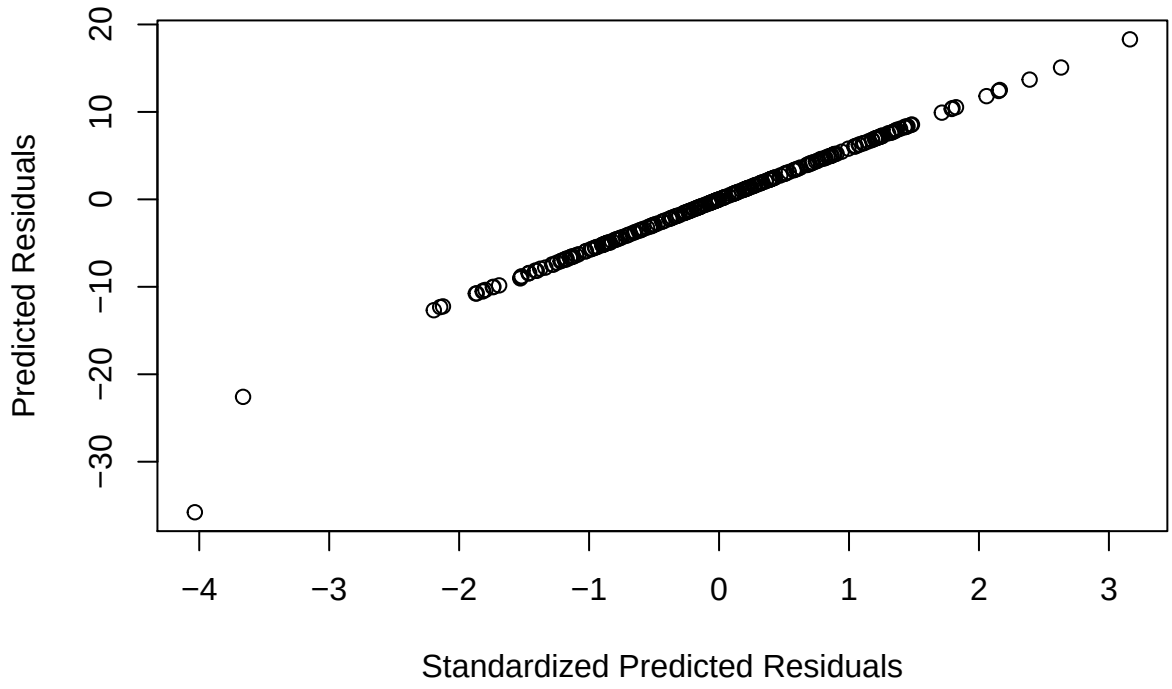
f)

```
plot(fit3$residuals ~ diag(H), xlab = 'Leverage', ylab = 'Residuals')
```



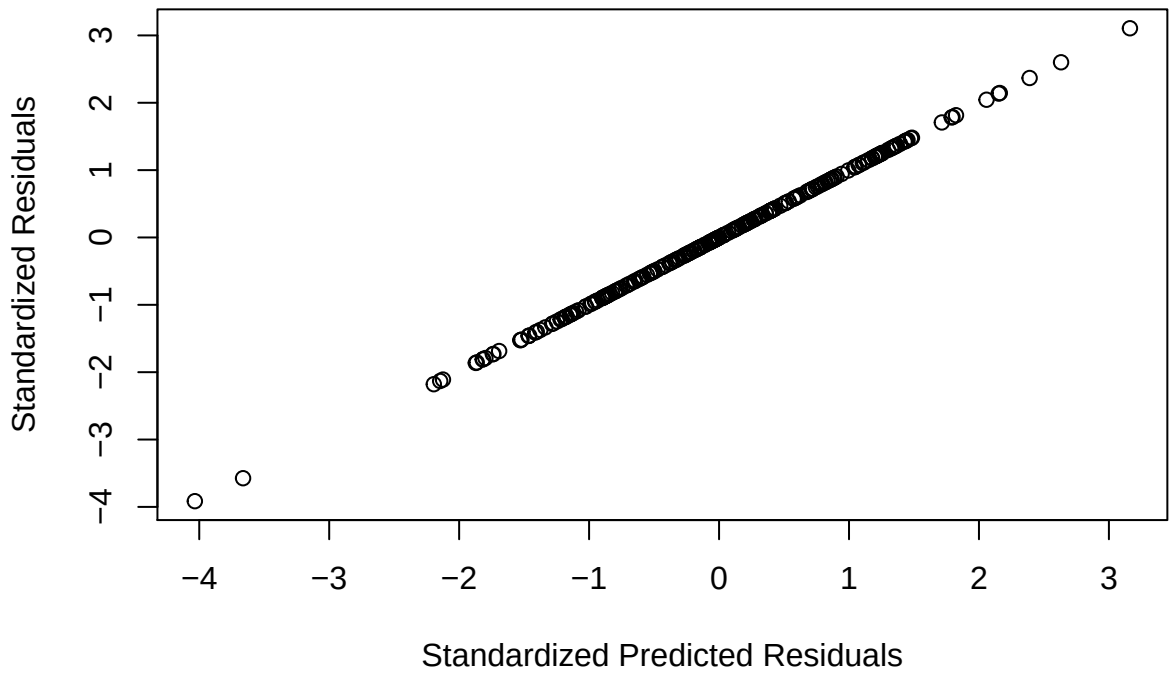
g)

```
std_pred_residuals =
  std_residuals *
  sqrt((n - p - 2) / (n - p - 1 - std_residuals^2))
plot(pred_residuals ~ std_pred_residuals,
     xlab = 'Standardized Predicted Residuals', ylab = 'Predicted Residuals')
```



h)

```
plot(std_residuals ~ std_pred_residuals,
     xlab = 'Standardized Predicted Residuals',
     ylab = 'Standardized Residuals')
```

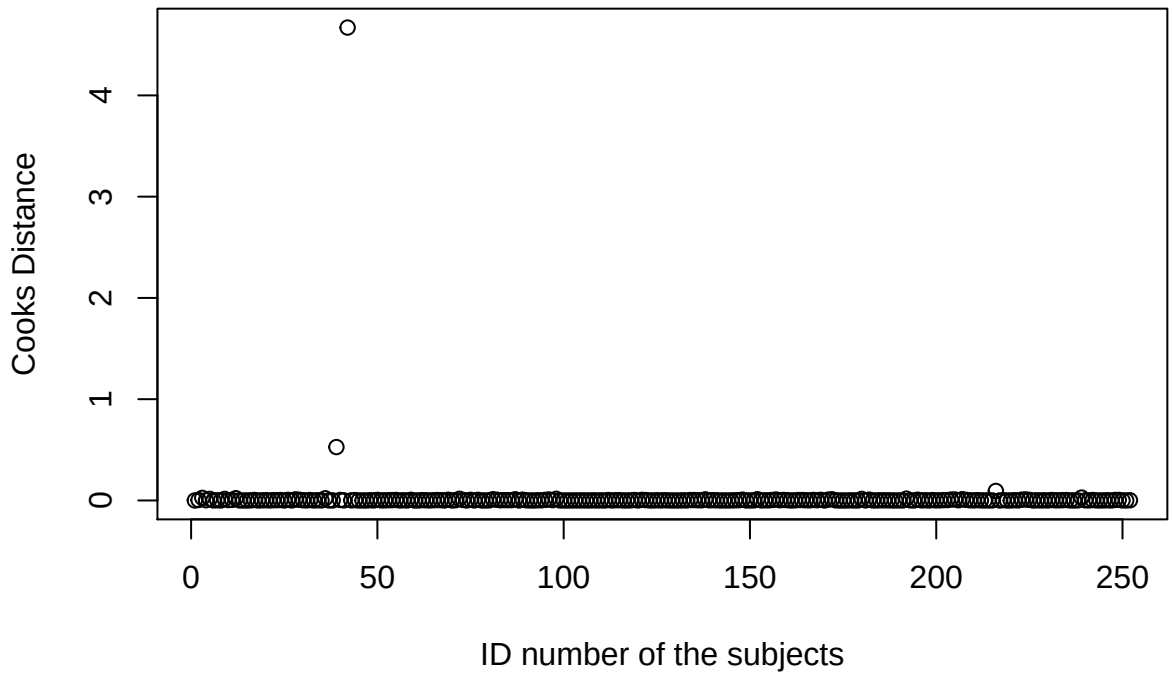


i)

```r
cook = std_residuals^2 * diag(H) / ((1-diag(H)) * (ncol(X)))
# Check the quantities with R function
all(round(cook, 6) == round(cooks.distance(fit3), 6))
```

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

```r
ID = 1:length(cook)
plot(cook ~ ID,
     xlab = 'ID number of the subjects', ylab = 'Cooks Distance')
```



j) Comment on these plots. Based on these plots, assess whether there are any outliers in the dataset; are there any influential observations.

Based on the plot in part i), we can see there is a point with high cooks distance, and another one has higher cooks distance than others, but since we don't know how high should we consider "high," we may need to take this point in our consideration as well.

Based on the plot in part f), we can see there are two points with high leverages. We know there are two "outliers in the direction of x."

Based on the plot in part a), we can see there are two points being far away from the cloud.

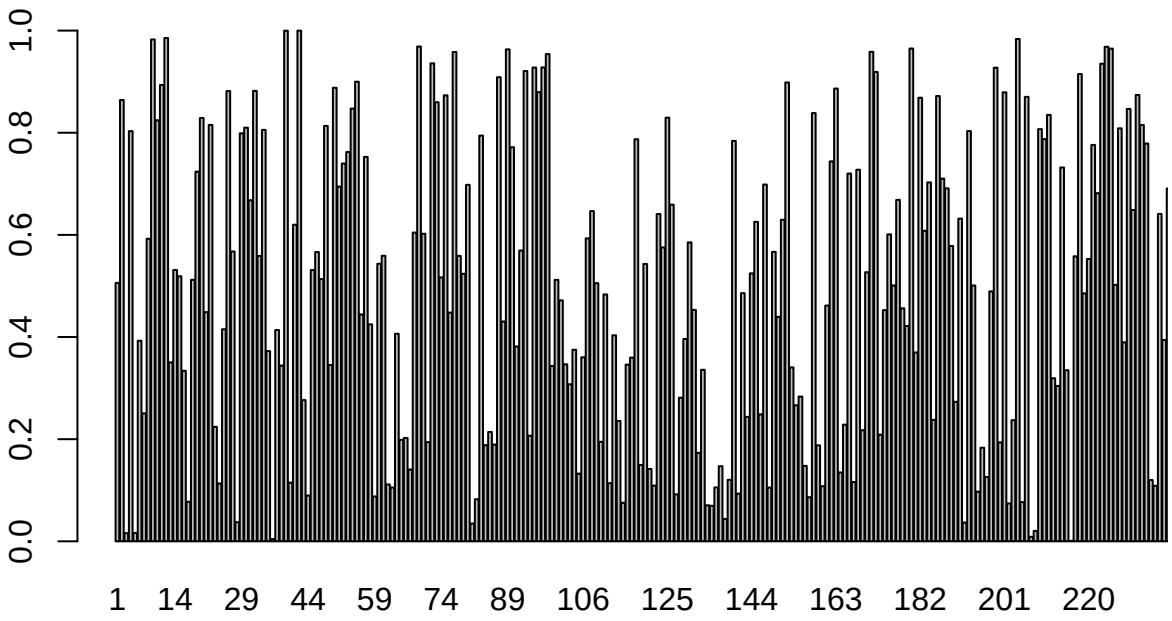
To conclude, I will say there may be two points in our dataset we can consider them as outliers.

(k) For each subject, calculate the p-value for testing whether the ith subject is an outlier based on the standardized predicted residual. Plot these p-values against the ID number of the subjects. How may of these p-values are less than 0.05? Does it make sense to rule all such subjects as outliers?

```r
p_values = c()
df = n - p - 2
for(i in 1:n) {
  p_val = pt(std_pred_residuals[i], 246, lower.tail = FALSE)
```

```
  p_values = c(p_values, p_val)
}

barplot(p_values, ylim = c(0, 1), xlim = c(0, n + 1))
```




```
num_less_than_.05 = sum(p_values < .05)
paste('There are', num_less_than_.05, 'subjects have p-values less than 0.05. ')
```

```
## [1] "There are 10 subjects have p-values less than 0.05. "
```

It doesn;t really make sense to rule out all 10 subjects as outliers. If we assume all observations are independent, then we will expect around 252 * 0.05 equals 12.6 observations to be within the rejection region. Therefore, if we want to test on $\alpha = .05$, we should consider the Bonferroni correction and use 0.05 / 252 as our new "α".

l) Based on the analysis, does it make sense to fit the linear model with any of the subjects removed? If not, why not? If so, which ones; and in this case, report the summary for the linear model with the subjects removed.

It does make sense to remove the two observations with the highest cook distances.

```
largest_2_indices = as.numeric(names(sort(cook, decreasing = TRUE)[1:2]))
paste('The observations should be removed are the ones with indices', largest_2_indices[1], 'and', large
```

```
## [1] "The observations should be removed are the ones with indices 42 and 39"
```

```
dat3_new = dat3[-largest_2_indices, ]
fit3_new = lm(bodyfat ~ Age + Weight + Height + Thigh,
              data = dat3_new)
summary(fit3_new)
```

```
##
## Call:
## lm(formula = bodyfat ~ Age + Weight + Height + Thigh, data = dat3_new)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
```

```
## -11.4982  -3.7381  -0.0034   3.7581  12.0943
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 42.82844   13.74245   3.117  0.00205 **
## Age          0.16101    0.03164   5.089 7.18e-07 ***
## Weight       0.21150    0.03020   7.003 2.39e-11 ***
## Height      -1.18281    0.16753  -7.060 1.70e-11 ***
## Thigh        0.24418    0.15252   1.601  0.11068
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.365 on 245 degrees of freedom
## Multiple R-squared:  0.5883, Adjusted R-squared:  0.5816
## F-statistic: 87.54 on 4 and 245 DF,  p-value: < 2.2e-16
```