HW04

caojilin

10/15/2018

Problem 1

a) Suppose X1,...,Xn are i.i.d observations from a distribution with known variance σ^2. Describe a bootstrap-based algorithm to compute a 95% confidence interval for σ. (0.5 points)

First we calculate the estimate of σ, $\widehat{\sigma}$ from sample observations. Then we can do nonparametric bootstrap.

```
Let N=5000
vec = rep(0,N)
for i = 1 to N:
    resample from sample with replacement
    calculate estiamted sigma from resample and store it in vec[i]
```

or parametric bootstrap

```
Let N=5000
vec = rep(0,N)
for i = 1 to N:
    generate a sample from a known distribution with estimated sigma
    calculate estiamted sigma from sample and store it in vec[i]
```

Now vec - σ forms a distribution. We pick 2.5% and 97.5% percentile of this distribution $b_{0.025}$ and $b_{0.975}$, the confidence interval is $[\widehat{\beta}_i - b_{0.975}, \widehat{\beta}_i - b_{0.025}]$

b) Take M = 1000. For each i = 1,...,M, simulate n = 100 observations from a normal distribution with σ = 1. Construct your confidence interval in the previous part and check if the interval contains the true value σ = 1. For how many i = 1,...,M, does your interval contain the true value? (0.5 points)

For this problem, we can use nonparametric bootstrap.

```
calc_CI =  function(samp, N=1000){
  samp_sd = sd(samp)
  vec = rep(0,N)
  for (i in 1:N) {
    # resample size does matter here!
    resamp = sample(samp,length(samp), replace = TRUE)
    vec[i] = sd(resamp)
  }
  b1 = quantile(vec-samp_sd,0.025)
  b2 = quantile(vec-samp_sd,0.975)
  # ci = quantile(vec,c(0.025,0.975))
  ci = c(samp_sd - b2, samp_sd - b1)
}
```

```
M= 1000
count = 0
for (i in 1:M) {
  samp = rnorm(100,sd=1)
  ci = calc_CI(samp)
  if (ci[1]<=1 & ci[2]>=1)
    count = count + 1
  }
count
```

```
## [1] 952
```

Problem 2

```
lmod1 = lm(lwage ~ jc + univ + exper, data=twoyear)
summary(lmod1)
```

```
##
## Call:
## lm(formula = lwage ~ jc + univ + exper, data = twoyear)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.10362 -0.28132  0.00551  0.28518  1.78167
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.4723256  0.0210602  69.910   <2e-16 ***
## jc          0.0666967  0.0068288   9.767   <2e-16 ***
## univ        0.0768762  0.0023087  33.298   <2e-16 ***
## exper       0.0049442  0.0001575  31.397   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4301 on 6759 degrees of freedom
## Multiple R-squared:  0.2224, Adjusted R-squared:  0.2221
## F-statistic: 644.5 on 3 and 6759 DF,  p-value: < 2.2e-16
```

```
X = model.matrix(lmod1)
sigma_square = 0.4301^2
```

null hypothesis H0 :$\widehat{\beta}_1 = \widehat{\beta}_2$ is equivalent to $\widehat{\beta}_1 - \widehat{\beta}_2 = 0$

a) Find the value of the t-statistic for this test. Does the t-test reject the null hypothesis at the 95 % level? (0.5 points).

$$\text{For } \beta_1 - \beta_2 = 0, \text{we can rewrite model as}$$
$$y_i = \beta_0 + (\beta_1 - \beta_2)jc + \beta_2(jc + univ) + \beta_3 expr + e_i,$$
$$\text{which is the same as original model}$$
$$y_i = \beta_0 + \beta_1 jc + \beta_2 univ + \beta_3 expr + e_i$$

```
lmod1 = lm(lwage ~ jc + I(jc+univ) + exper, data=twoyear)
summary(lmod1)
```

```
##
## Call:
## lm(formula = lwage ~ jc + I(jc + univ) + exper, data = twoyear)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.10362 -0.28132  0.00551  0.28518  1.78167
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.4723256  0.0210602  69.910   <2e-16 ***
## jc           -0.0101795  0.0069359  -1.468    0.142
## I(jc + univ)  0.0768762  0.0023087  33.298   <2e-16 ***
## exper         0.0049442  0.0001575  31.397   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4301 on 6759 degrees of freedom
## Multiple R-squared:  0.2224, Adjusted R-squared:  0.2221
## F-statistic: 644.5 on 3 and 6759 DF,  p-value: < 2.2e-16
```

Now β_1 in this model here is $\beta_1 - \beta_2$ in original model we want to test for. We can directly read t-value, which is -1.468, it's p-value is 0.142, thus we cannot reject the null at a=5% level.

b) Find the value of the F-statistic for this test. Does the F-test reject the null hypothesis at the 95 %

level? (0.5 points).

$$\text{For } \beta_1 = \beta_2, \text{the model m becomes}$$
$$y_i = \beta_0 + \beta_1(x_{i1} + x_{i2}) + \beta_3 x_{i3} + \cdots + \beta_p x_{ip} + e_i$$

we can use R to fit reduced model m

```
lmod2 = lm(lwage ~ I(jc + univ) + exper, data=twoyear)
summary(lmod2)
```

```
##
## Call:
## lm(formula = lwage ~ I(jc + univ) + exper, data = twoyear)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.09708 -0.28069  0.00532  0.28324  1.78332
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.4719702  0.0210606   69.89   <2e-16 ***
## I(jc + univ) 0.0761563  0.0022562   33.75   <2e-16 ***
## exper        0.0049323  0.0001573   31.36   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4302 on 6760 degrees of freedom
## Multiple R-squared:  0.2222, Adjusted R-squared:  0.222
## F-statistic: 965.6 on 2 and 6760 DF,  p-value: < 2.2e-16
```

then f statistic is

$$f_{1,6759} = \frac{RSS(m) - RSS(M)}{RSS(M)/(n-3-1)}$$
$$\text{where } n = 6763$$
$$RSS(m) = 1250.942$$
$$RSS(M) = 1250.544$$

```
## [1] 2.154016
```

and we get f = 2.154016, and p-value is `1 - pf(2.154016, df1=1,df2=6759)=0.142244` We do not reject the null at a=5% level

c) Design a permutation test for testing this hypothesis. Does your test reject the null hypothesis at the 95% level? (0.8 points).

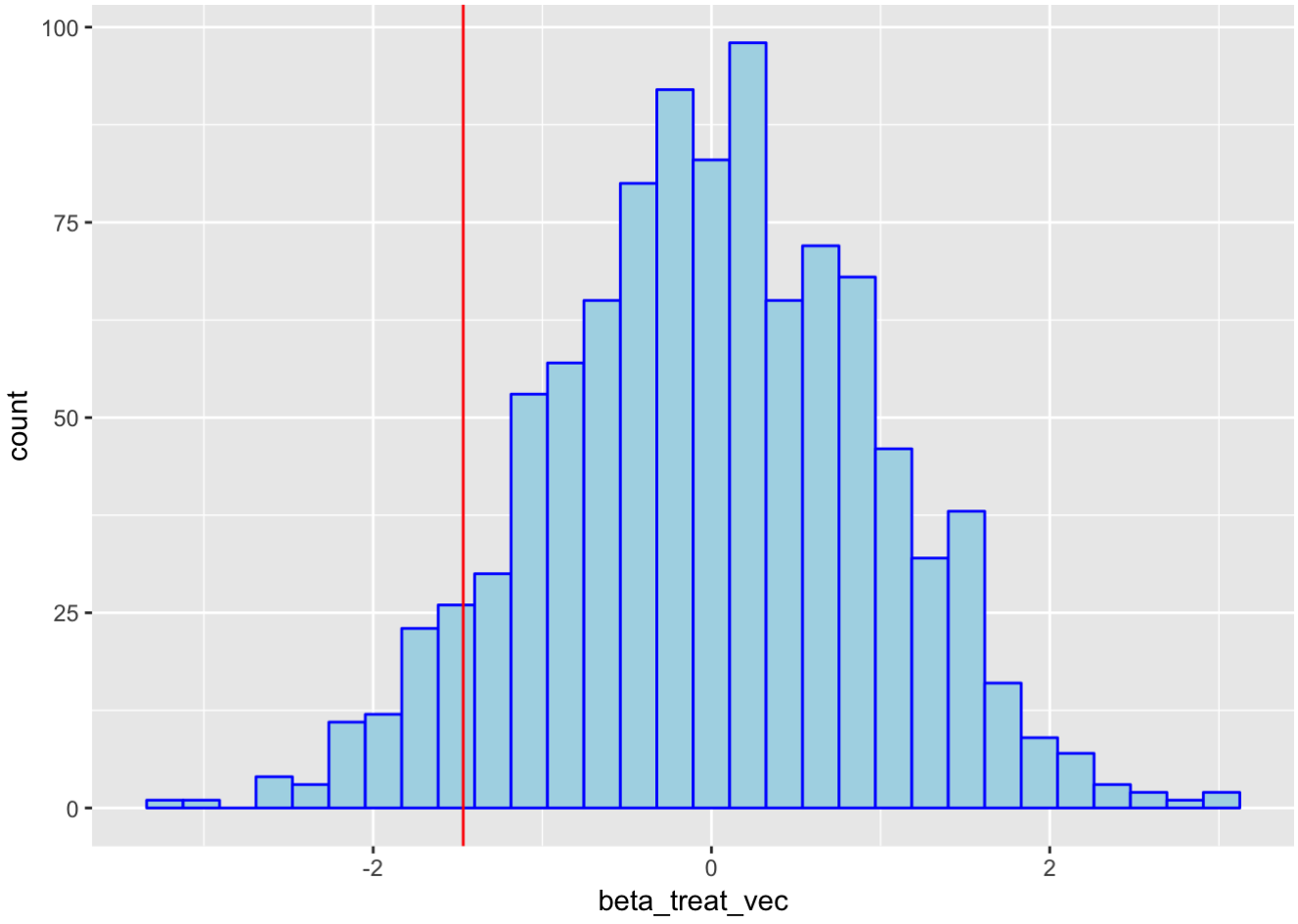
$$\text{For } \beta_1 - \beta_2 = 0, \text{we can rewrite model as}$$
$$y_i = \beta_0 + (\beta_1 - \beta_2)jc + \beta_2(jc + univ) + \beta_3 expr + e_i,$$
$$\text{which is the same as original model}$$
$$y_i = \beta_0 + \beta_1 jc + \beta_2 univ + \beta_3 expr + e_i$$

We can still use modified model and permute *jc* on new model, in this way we can test the null $\beta_1 - \beta_2 = 0$ and only permute one column

```
# run the permutation test
n_perm <- 1000
beta_treat_vec <- rep(0, n_perm)
observed_t = summary(lmod1)$coefficients[2,3]
for(i in 1:n_perm){
  ss= sample(twoyear$jc)
  lmod_perm = lm(lwage ~ ss + I(jc + univ) + exper, data=twoyear)
  beta_treat_vec[i] <- summary(lmod_perm)$coefficients[2,3]
}

# plot the histogram of the treatment coefficient under
# the permutation distribution
# red line was our originally computed statistic
ggplot() + geom_histogram(aes(x = beta_treat_vec),
                          color = 'blue', fill = 'light blue') +
  geom_vline(xintercept = observed_t, color = 'red')
```

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



p-value is

```
mean(abs(beta_treat_vec) >= abs(observed_t))
```

```
## [1] 0.136
```

we do not reject the null

d) Construct a 95 % confidence interval for $\beta_1 - \beta_2$ via bootstrap. Does this interval contain the value zero? (0.8 points).

```
full.model = lm(lwage ~ jc + univ + exper, data=twoyear)
beta1 = full.model$coefficients["jc"]
beta2 = full.model$coefficients["univ"]
vec1 = rep(0,2000)
#residual bootstrap, can we get any linear combinations?
for(i in 1:2000){
  e_temp = sample(full.model$residuals, nrow(twoyear),replace = TRUE)
  new_y = X %*% full.model$coefficients + e_temp
  tempmodel = lm(new_y ~ jc + univ + exper, data=twoyear)
  vec1[i] = tempmodel$coefficients[2] - tempmodel$coefficients[3]
}
```

95% CI for $\beta_1 - \beta_2$ contains 0

```
b1 = quantile(vec1-(beta1-beta2),0.025)
b2 = quantile(vec1-(beta1-beta2),0.975)
ci = c(beta1-beta2-b2, beta1-beta2-b1)
names(ci) = c("","")
ci
```

```
##
## -0.023210654  0.003103643
```

Problem 3

a) Use R to report the usual normality based confidence intervals for each of $\beta_1, \cdots, \beta_4$ (0.4 points)

```
##
## Call:
## lm(formula = sr ~ pop15 + pop75 + dpi + ddpi, data = savings)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.2422 -2.6857 -0.2488  2.4280  9.7509
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 28.5660865  7.3545161   3.884 0.000334 ***
## pop15       -0.4611931  0.1446422  -3.189 0.002603 **
## pop75       -1.6914977  1.0835989  -1.561 0.125530
## dpi         -0.0003369  0.0009311  -0.362 0.719173
## ddpi         0.4096949  0.1961971   2.088 0.042471 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.803 on 45 degrees of freedom
## Multiple R-squared:  0.3385, Adjusted R-squared:  0.2797
## F-statistic: 5.756 on 4 and 45 DF,  p-value: 0.0007904
```

we use

$$\widehat{\beta}_j \pm t_{n-p-1}^{a/2} s.e(\widehat{\beta}_j)$$

$$t_{n-p-1}^{a/2} = qt(0.975, 45) = 2.014103$$

from

$$\widehat{\beta}_1 \; to \; \widehat{\beta}_4$$

the 95% CI:

$$-0.4611931 \pm 0.2913243$$

$$-1.6914977 \pm 2.18248$$

$$-0.0003369 \pm 0.001875$$

$$0.4096949 \pm 0.395161$$

b) Compute confidence intervals for$\beta_1, \cdots, \beta_4$ using residual bootstrap. How do these intervals compare with those in part (a) above? (0.8 points).

```
vec_residual = lmod3$residuals
beta_matrix = matrix(0,2000,5,byrow = TRUE)
for (i in 1:2000) {
  e_hat_i = sample(vec_residual,length(vec_residual),replace = TRUE)
  y_i = model.matrix(lmod3) %*% lmod3$coefficients + e_hat_i
  beta_matrix[i,] = lm(y_i ~ pop15 + pop75 + dpi + ddpi, data = savings)$coefficients -
 lmod3$coefficients
}
ma = matrix(0,5,2,byrow = TRUE)
for (i in 1:5) {
  a = quantile(beta_matrix[,i],0.025)
  b = quantile(beta_matrix[,i],0.975)
  ma[i,] = c(a,b)
}
ma = as.data.frame(ma)
colnames(ma) = c(0.025,0.975)
```

percentile matrix for $\beta_0, \cdots, \beta_4$

```
ma
```

```
##           0.025        0.975
## 1 -13.959022559 13.200299024
## 2  -0.265754384  0.268882572
## 3  -1.906662326  2.126133797
## 4  -0.001695754  0.001770856
## 5  -0.365086821  0.381485541
```

we use

$$[\widehat{\beta}_i - b_{0.975}, \widehat{\beta}_i - b_{0.025}]$$

from

$$\widehat{\beta}_1 \; to \; \widehat{\beta}_4$$

the 95% CI:

```
ci = matrix(0,4,2,byrow = TRUE)
for (i in 1:4) {
  ci[i,] = c(lmod3$coefficients[i+1]-ma[i+1,2],lmod3$coefficients[i+1]-ma[i+1,1])
}
ci
```

```
##              [,1]         [,2]
## [1,] -0.730075719 -0.195438764
## [2,] -3.817631474  0.215164649
## [3,] -0.002107758  0.001358852
## [4,]  0.028209386  0.774781749
```

we found that these confidence intervals are **narrower** than part a)

Problem 4 how do we actually identify outliers? any standards?

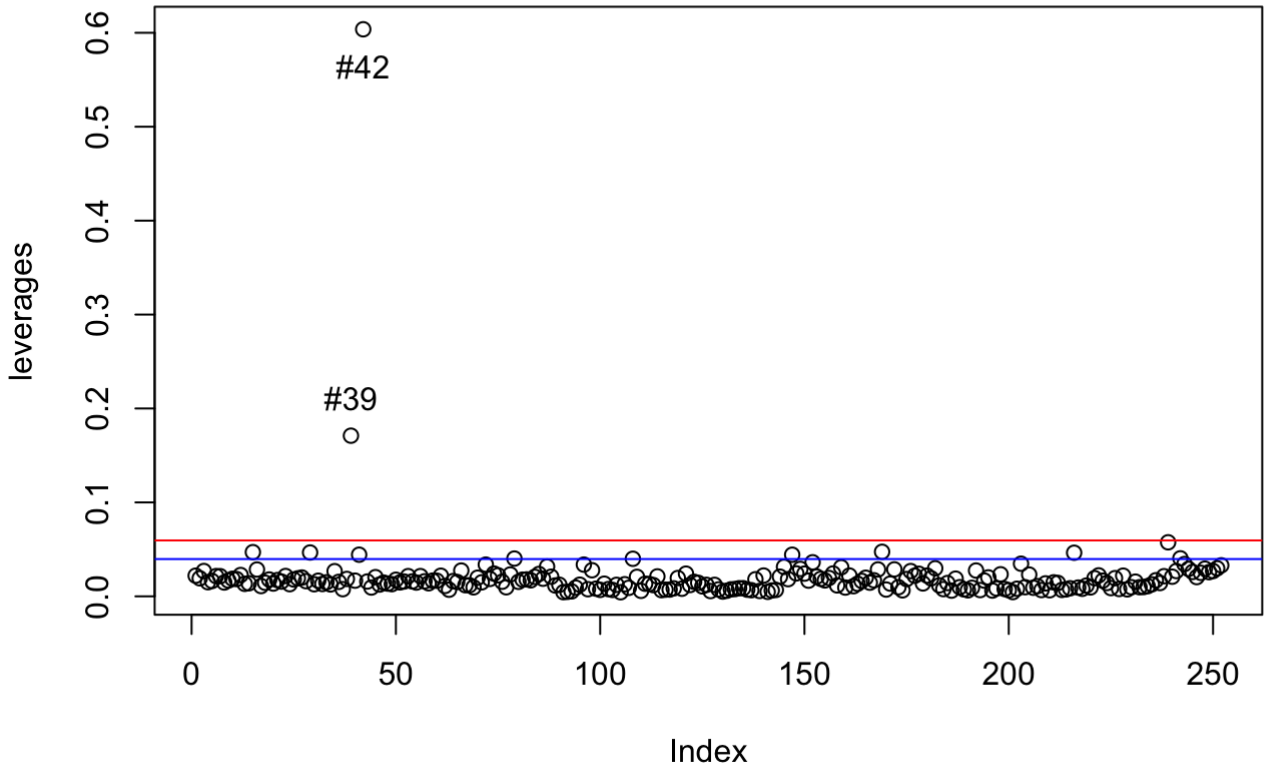
Comment on these plots. Based on these plots, assess whether there are any outliers in the dataset; are there any influential observations. (0.5 points)

```
lmod4 = lm(bodyfat ~ Age +Weight + Height+Thigh,data=body)
n = nrow(body)
p = 4
y = body$bodyfat
X = model.matrix(lmod4)
H = X %*% solve(t(X) %*% X) %*% t(X)
e_hat = lmod4$residuals
y_hat = H %*% y
sigma_hat <- sqrt(sum(e_hat**2) / (n - p - 1))
leverages = diag(H)
h_bar = (1+p)/n
head(sort(leverages,decreasing = TRUE))
```

```
##         42         39        239        169         15         29
## 0.60373733 0.17103211 0.05752533 0.04742070 0.04703008 0.04665892
```

```
plot(leverages)
abline(b=0,a=2*h_bar,col="blue")
abline(b=0,a=3*h_bar,col="red")

text(x = 42,y=0.60373733-0.04,"#42")
text(x = 39,y=0.17103211+0.04,"#39")
```



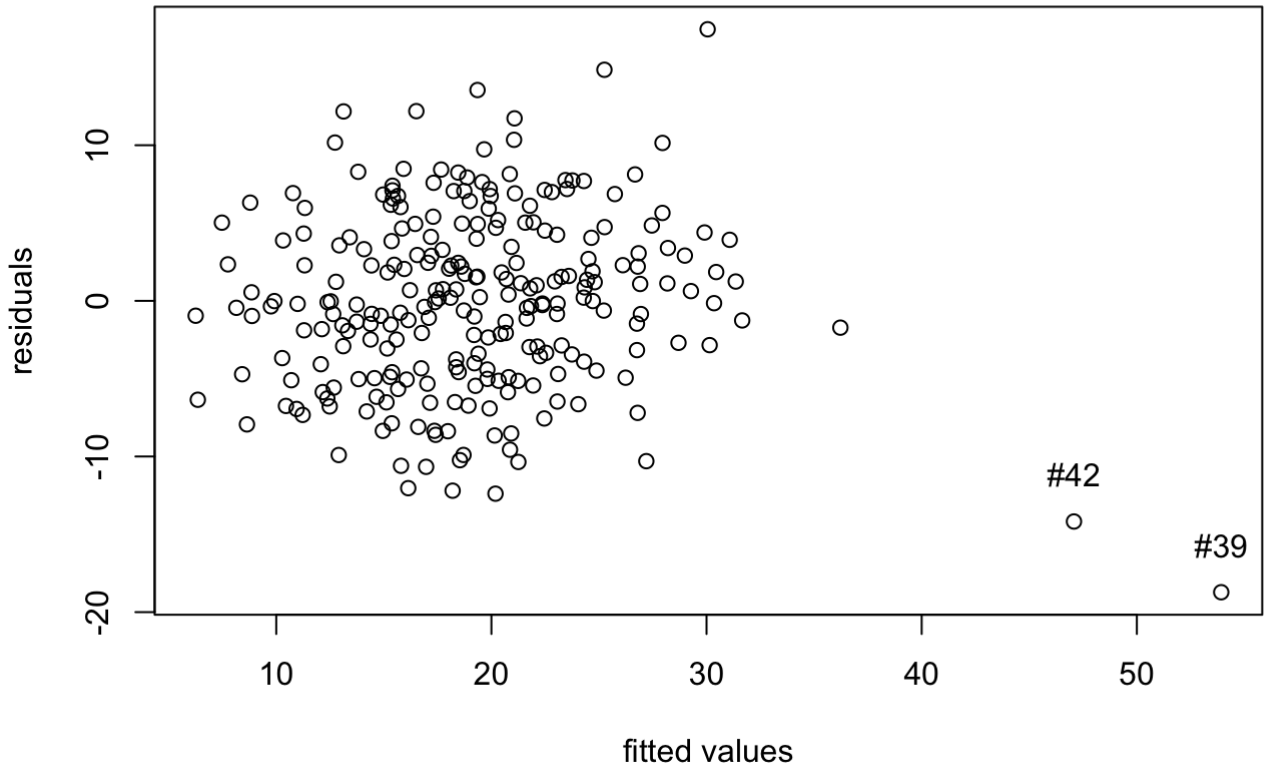
Index 39 and 42 subject have high leverage

a) Residuals against fitted values.

```
fit_value = lmod4$fitted.values
head(sort(fit_value,decreasing = TRUE))
```

```
##       39       42       41      243      244      242
## 53.92162 47.07736 36.22114 31.65766 31.35755 31.07688
```

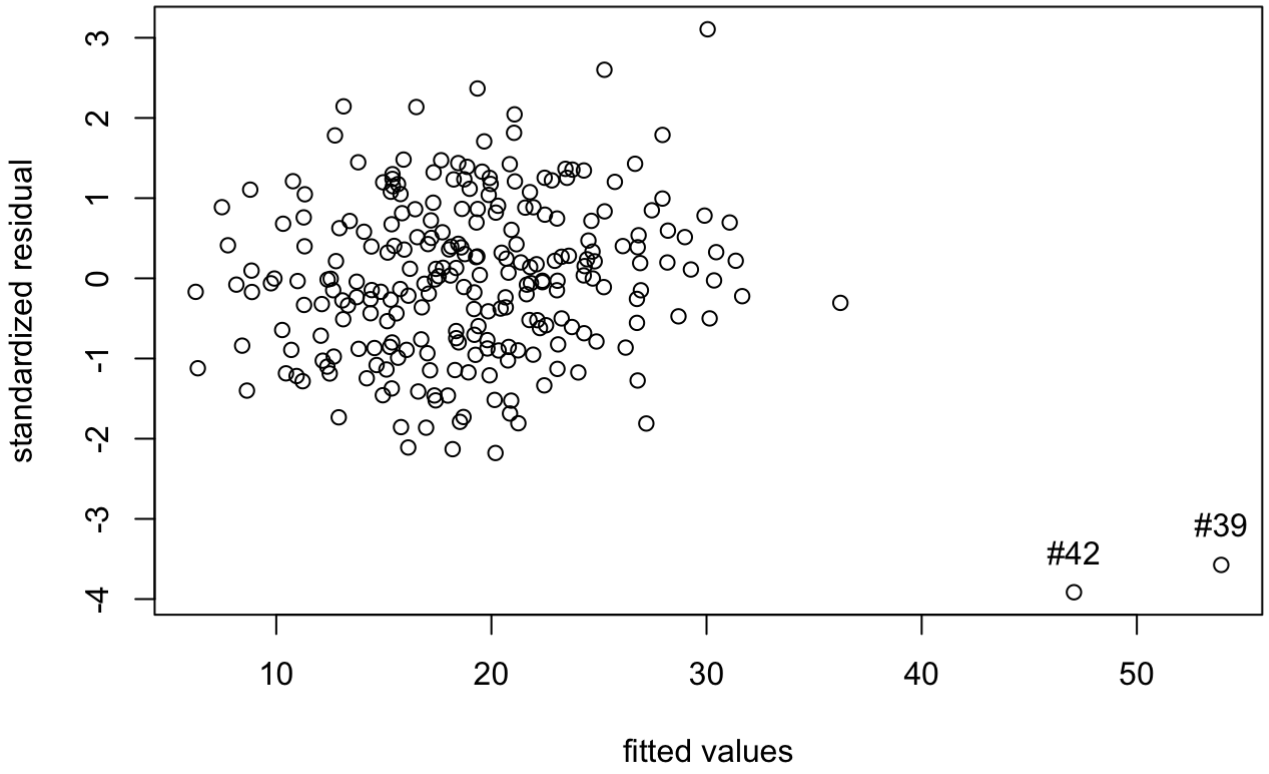
```
res = lmod4$residuals
plot(res ~ fit_value,xlab="fitted values", ylab="residuals")
text(fit_value[39],res[39]+3,"#39")
text(fit_value[42],res[42]+3,"#42")
```



b) Standardized Residuals against fitted values.

```
e_hat_std <- e_hat / (sigma_hat * sqrt(1 - diag(H)))

plot(e_hat_std ~ fit_value,xlab="fitted values", ylab = 'standardized residual')
text(fit_value[39],e_hat_std[39]+0.5,"#39")
text(fit_value[42],e_hat_std[42]+0.5,"#42")
```

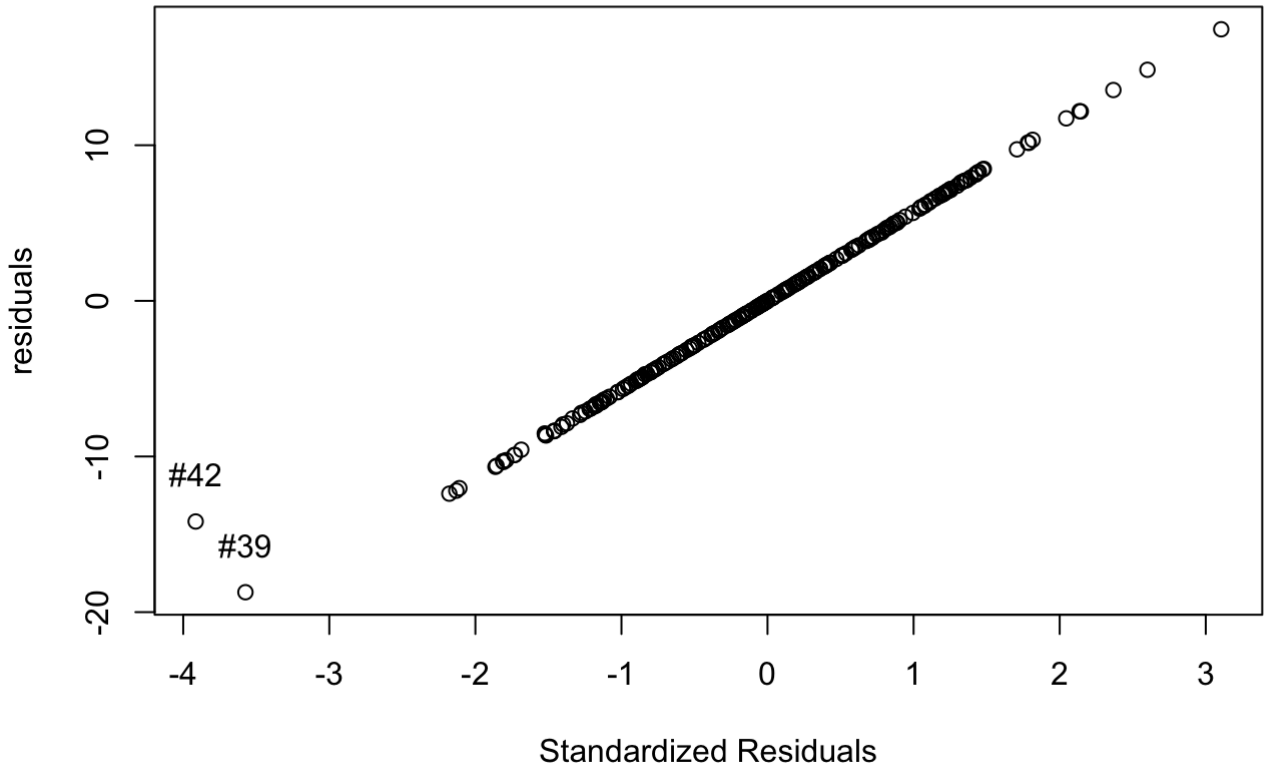


c) Residuals against Standardized Residuals.

```
head(sort(res))
```

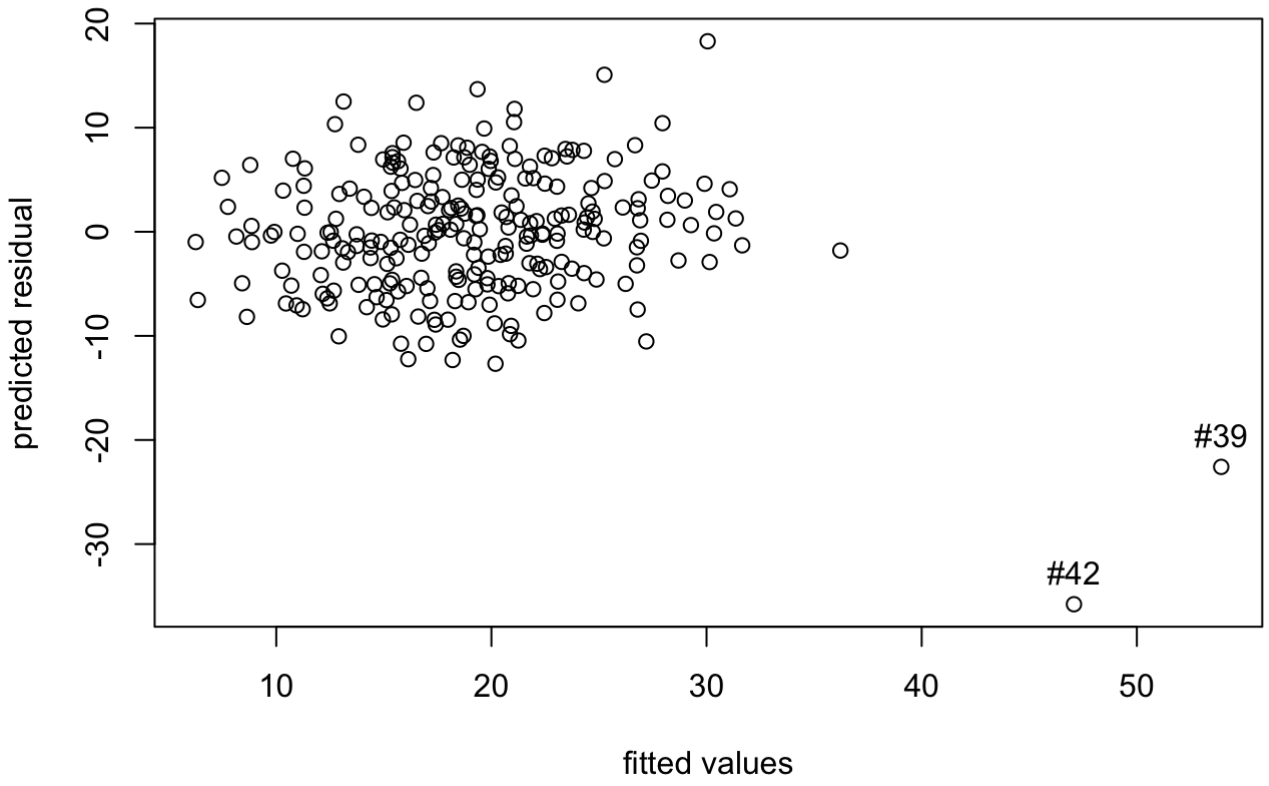
```
##        39        42        12       204         9        69
## -18.72162 -14.17736 -12.38902 -12.19729 -12.03450 -10.66488
```

```
plot(res ~ e_hat_std,xlab= "Standardized Residuals",ylab = "residuals")
text(e_hat_std[42],res[42]+3,"#42")
text(e_hat_std[39],res[39]+3,"#39")
```



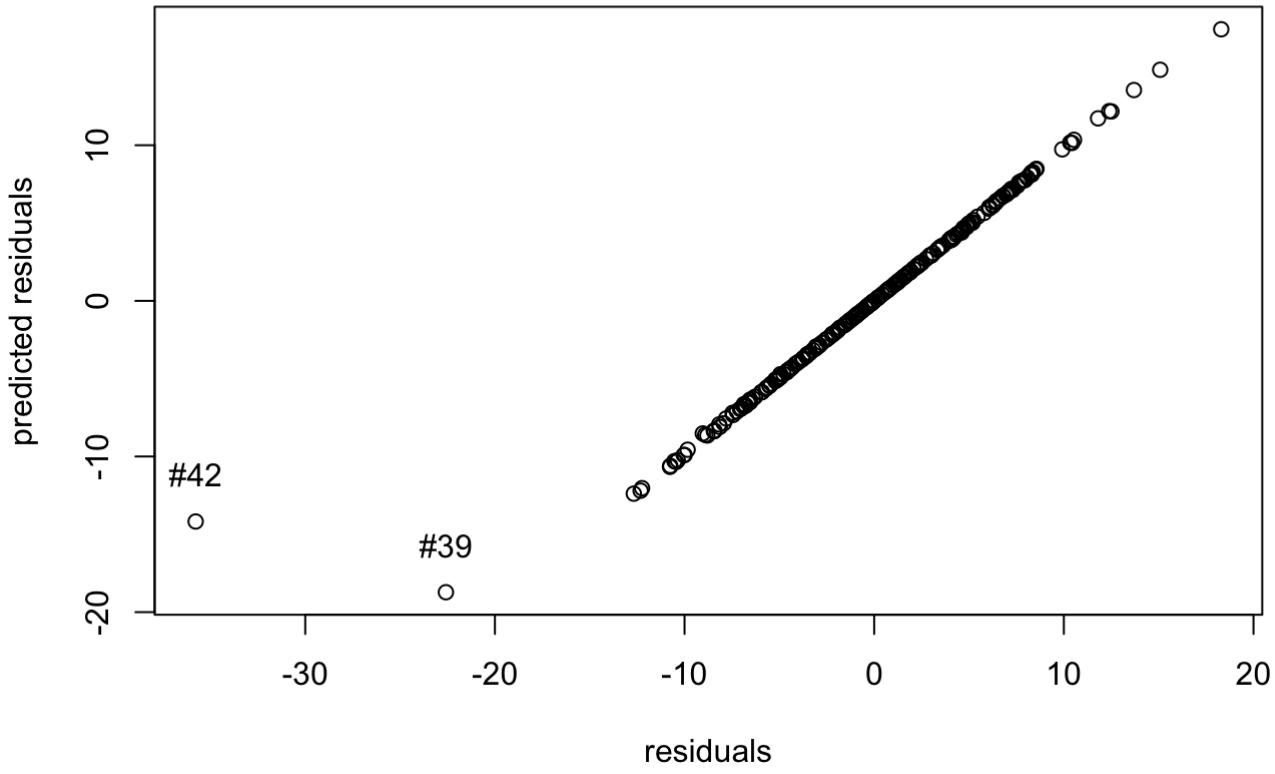
d) Predicted residuals against fitted values.

```
e_hat_pred <- e_hat / (1 - diag(H))
plot(e_hat_pred ~ fit_value, xlab = "fitted values",ylab = 'predicted residual')
text(fit_value[39],e_hat_pred[39]+3,"#39")
text(fit_value[42],e_hat_pred[42]+3,"#42")
```



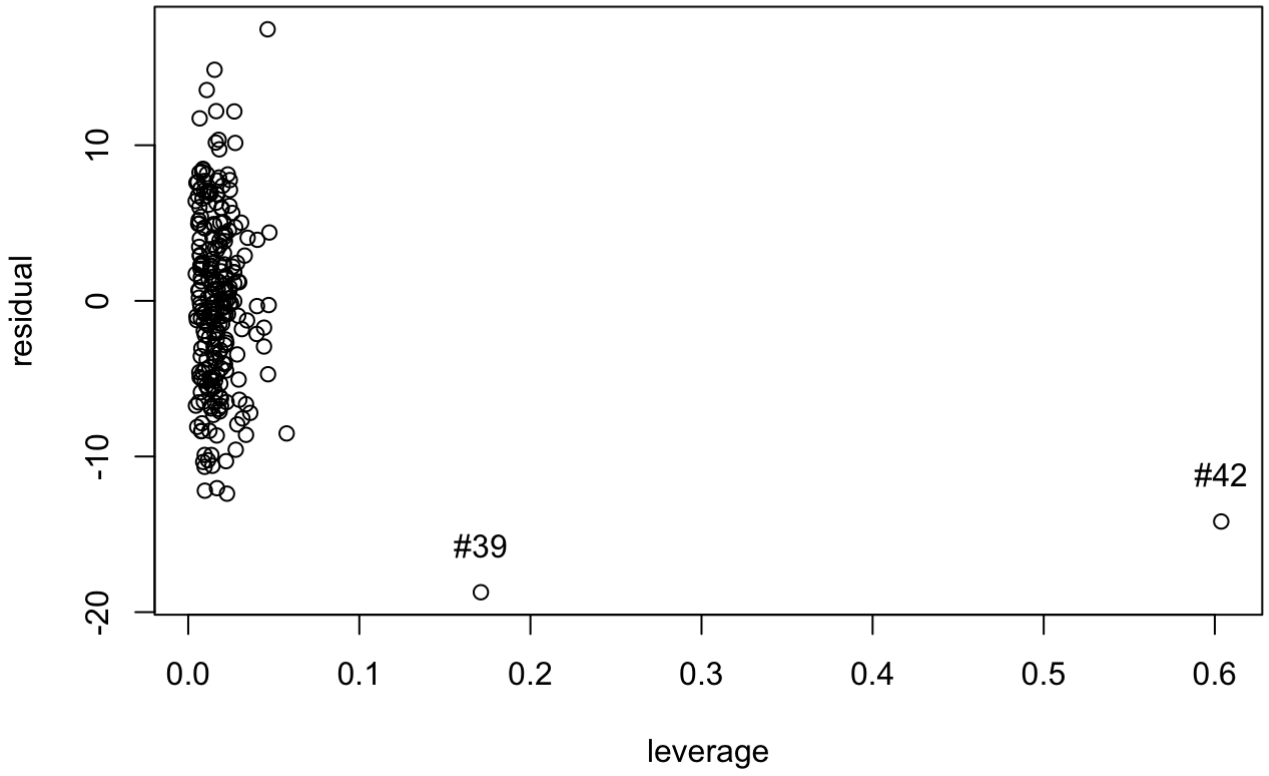
e) Residuals against predicted residuals.

```
plot(e_hat ~ e_hat_pred,xlab="residuals",ylab="predicted residuals")
text(e_hat_pred[39],e_hat[39]+3,"#39")
text(e_hat_pred[42],e_hat[42]+3,"#42")
```



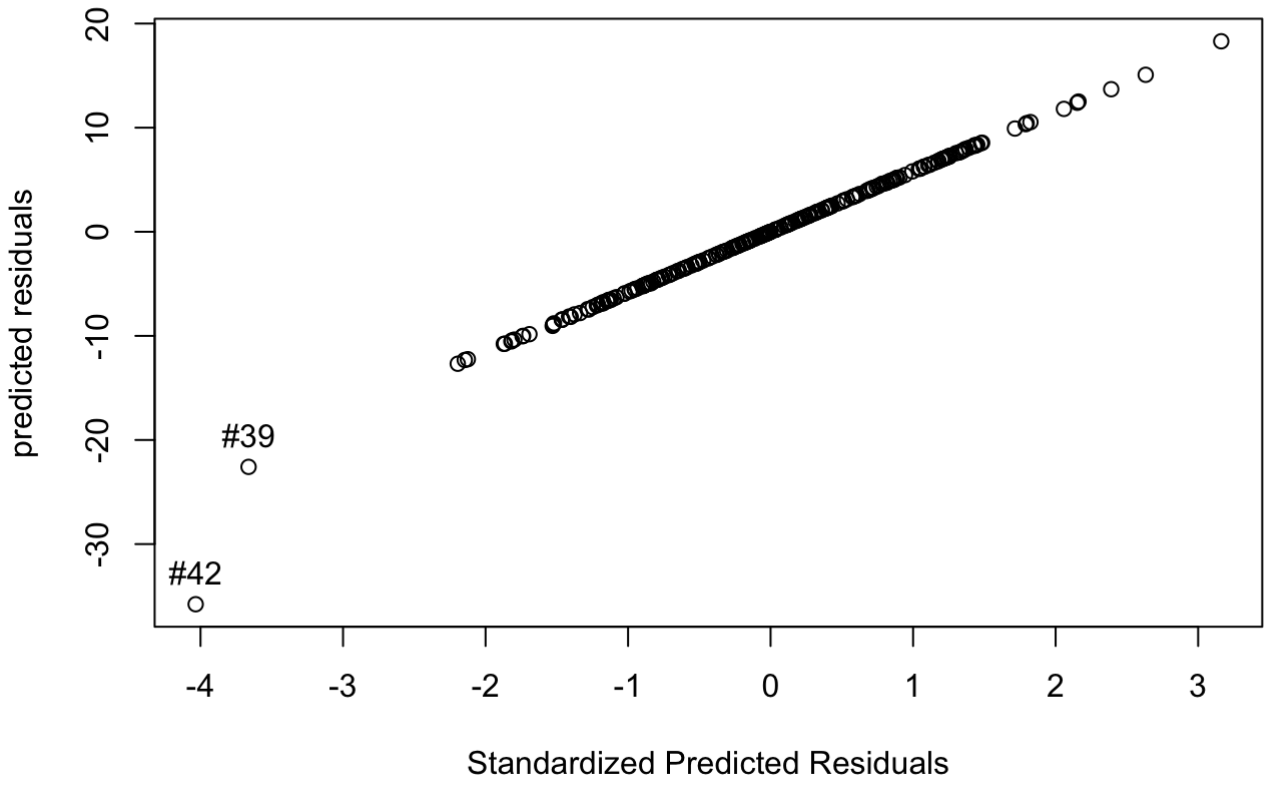
f) Residuals against leverage.

```
plot(e_hat~diag(H),xlab="leverage",ylab="residual")
text(leverages[39],e_hat[39]+3,"#39")
text(leverages[42],e_hat[42]+3,"#42")
```



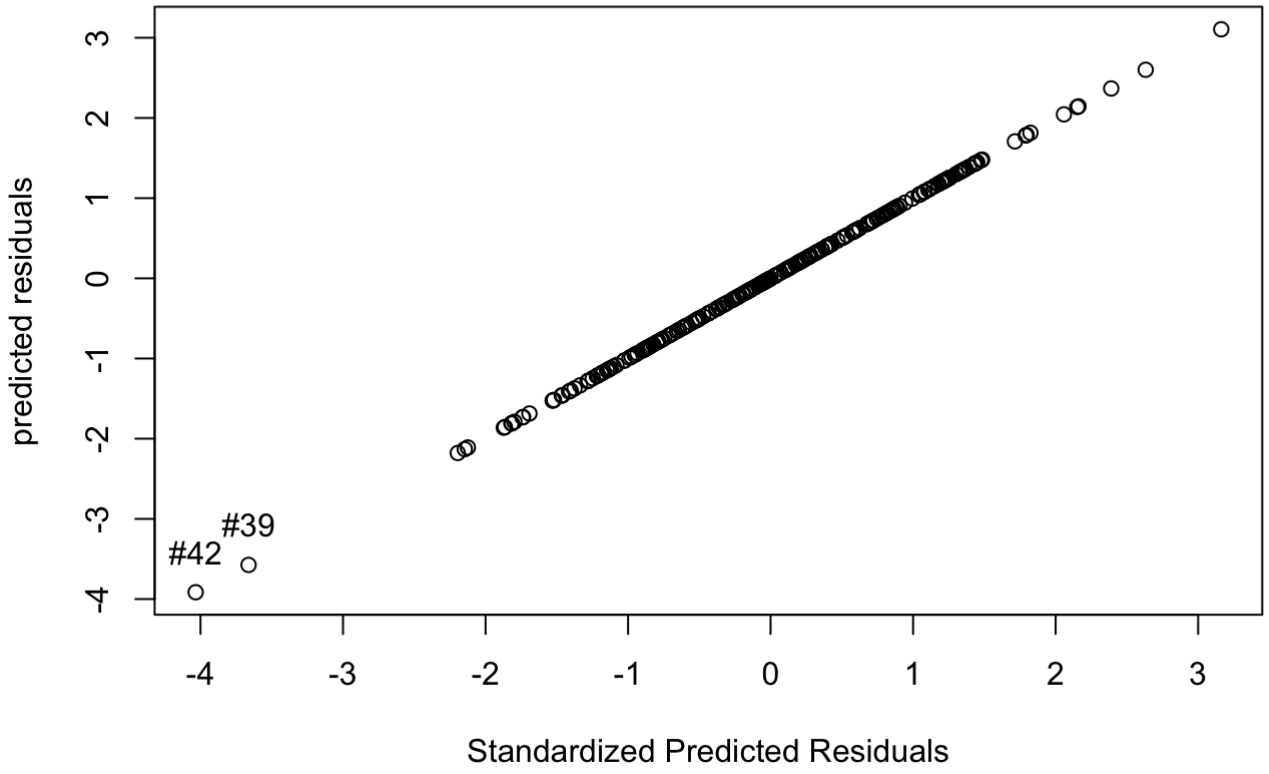
g) Predicted residuals against Standardized Predicted Residuals.

```
e_hat_pred_std = e_hat_std * sqrt((n-p-2)/(n-p-1-e_hat_std^2))
plot(e_hat_pred ~ e_hat_pred_std,xlab="Standardized Predicted Residuals",ylab="predicted
 residuals")
text(e_hat_pred_std[39],e_hat_pred[39]+3,"#39")
text(e_hat_pred_std[42],e_hat_pred[42]+3,"#42")
```



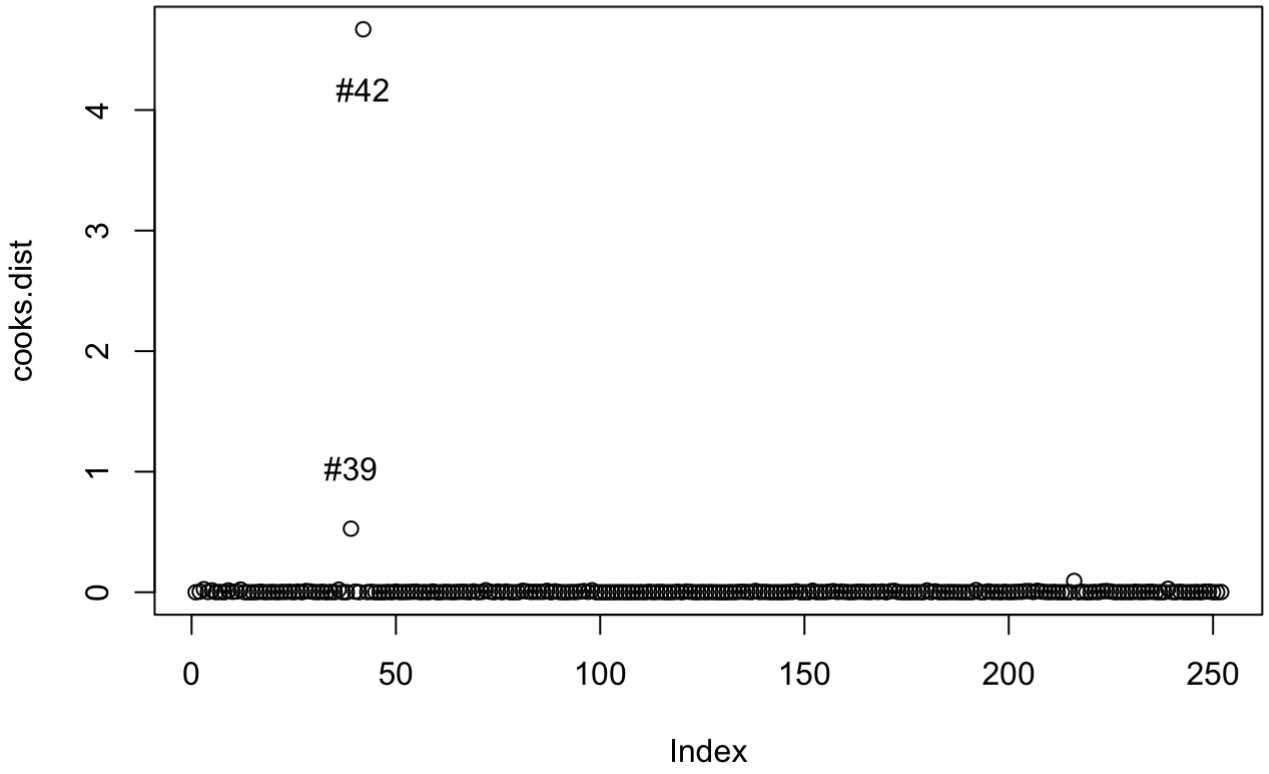
h) Standardized residuals against Standardized Predicted residuals.

```
# any convenient way to calculate RSS[i]?
# RSS[i] = RSS-(ei^2/(1-hi))
plot(e_hat_std ~ e_hat_pred_std,xlab="Standardized Predicted Residuals",ylab="predicted
 residuals")
text(e_hat_pred_std[39],e_hat_std[39]+0.5,"#39")
text(e_hat_pred_std[42],e_hat_std[42]+0.5,"#42")
```



i) Cooks Distance against the ID number of the subjects.

```
cooks.dist = cooks.distance(lmod4)
plot(cooks.dist)
text(39,cooks.dist[39]+0.5,"#39")
text(42,cooks.dist[42]-0.5,"#42")
```

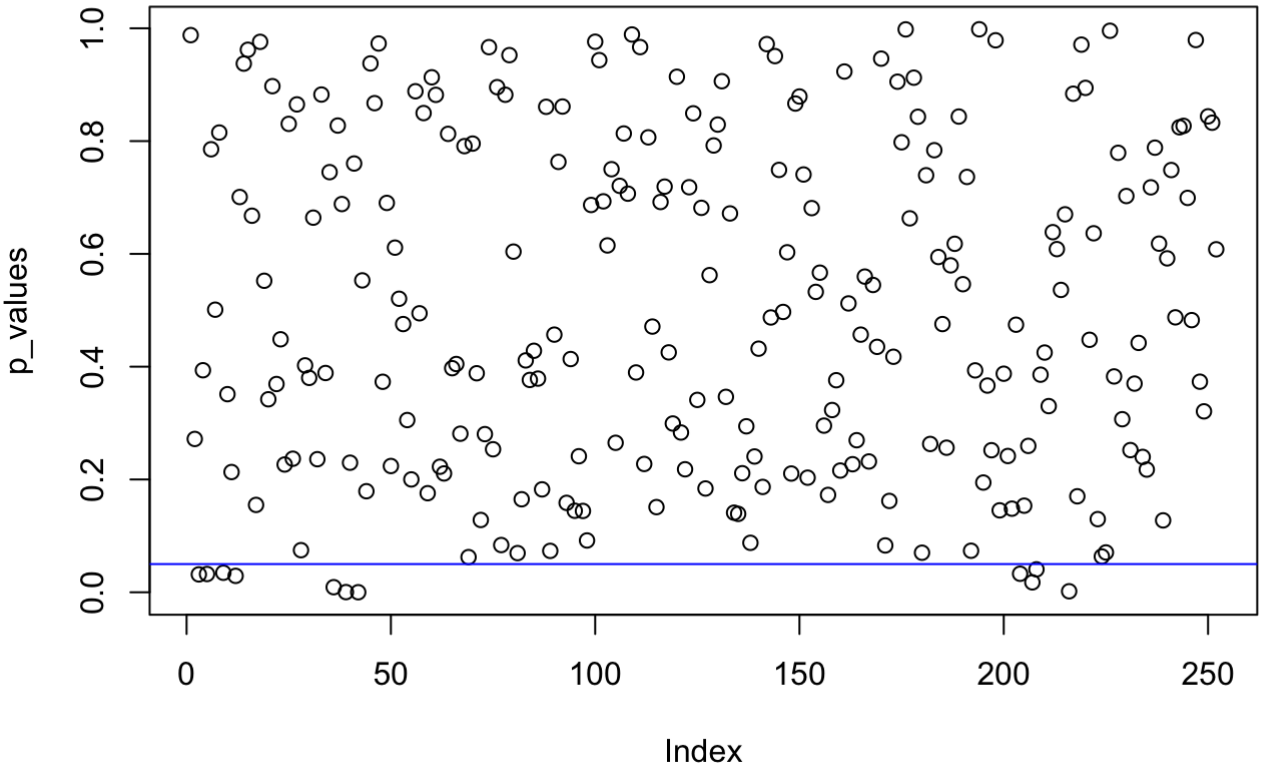


Comment on these plots. Based on these plots, assess whether there are any outliers in the dataset; are there any inuential observations. (0.5 points)

The index 39 and 42 points have high leverages. It's obviously to see in the residual against fitted values graph. Standardized residual graph also support this evidence. They also stand out in the other graphs.

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 many of these p-values are less than 0.05? Does it make sense to rule all such subjects as outliers? (1 points)

```
p_values = 2*(1-pt(abs(e_hat_pred_std),n-p-2))
plot(p_values)
abline(a=0.05,b=0,col="blue")
```




```
sort(p_values[which(p_values<0.05)])
```

```
##           42           39          216           36          207
## 7.325343e-05 3.054138e-04 1.764147e-03 9.029060e-03 1.760332e-02
##           12            3            5          204            9
## 2.909586e-02 3.169823e-02 3.237542e-02 3.285382e-02 3.461962e-02
##          208
## 4.062107e-02
```

There are 11 points whose p-value less than 0.05. But it doesn't make sense to rule out all these points as outliers

do a **Bonferroni correction**

```
p_values[which(p_values < 0.05/n)]
```

```
##           42
## 7.325343e-05
```

We found that only #42 stood out.

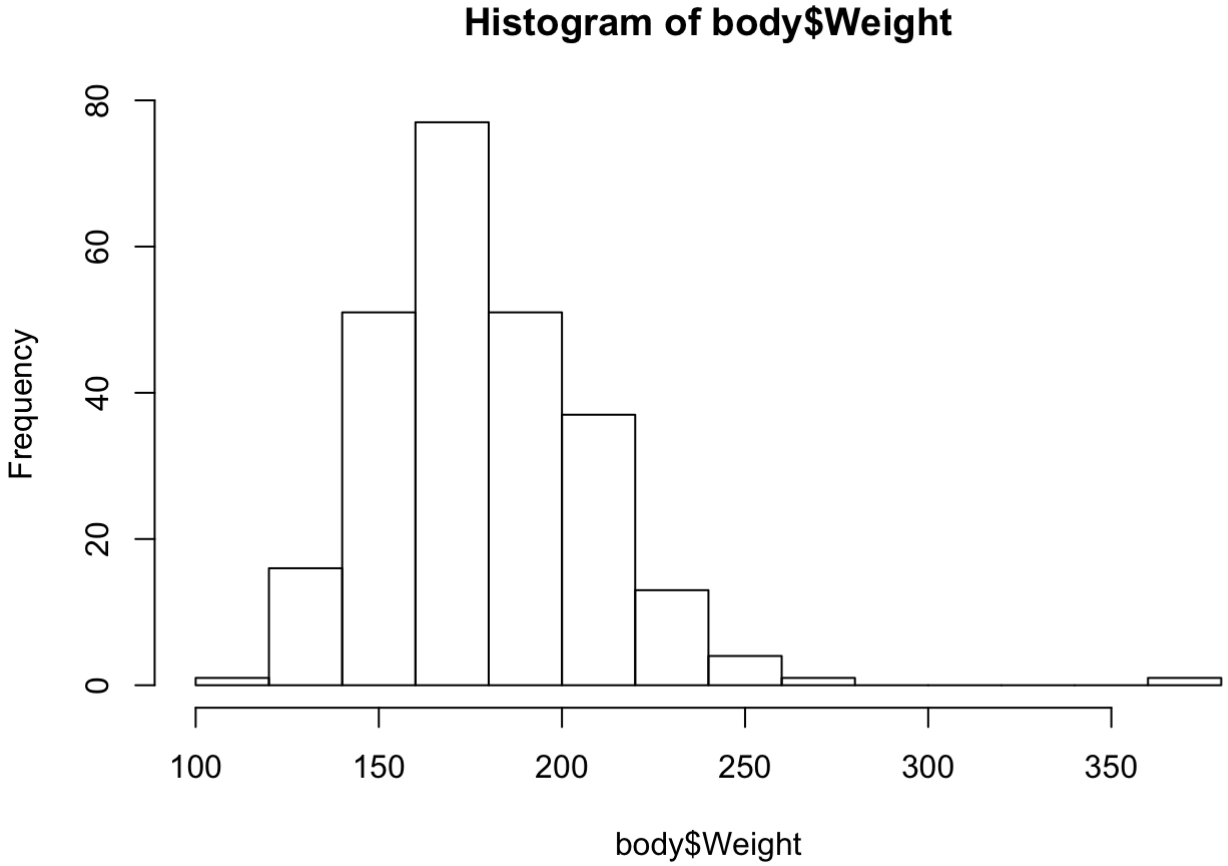
Based on the analysis, does it make sense to fit the linear model with any of the subjects removed? If not,why not? If so, which ones; and in this case, report the summary for the linear model with the subjects removed. (1 points)

we investigate these two points 39 and 42

```
body[c(39,42),]
```

```
##    bodyfat Age Weight Height Thigh
## 39    35.2  46 363.15  72.25  87.3
## 42    32.9  44 205.00  29.50  70.6
```

```
hist(body$Weight)
```




```
sd(body$Weight);mean(body$Weight)
```

```
## [1] 29.38916
```

```
## [1] 178.9244
```

And we see that on the weight histogram, subject 39's weight is 363.15 lbs, which is almost 6 standard deviation from mean of weight 178.9244. This is an unusualy observation, but other data looks legit. So this may be an error or a guy who is really this heavy

```
hist(body$Height)
```

We notice that subject 42's height is weird, because how can one man's weight be 205 pounds and only 29.5 inches, which is 74.93 cm. This probably is an error in data.

original model

```
summary(lm(bodyfat ~ Age +Weight + Height+Thigh,data=body))
```

```
##
## Call:
## lm(formula = bodyfat ~ Age + Weight + Height + Thigh, data = body)
##
## 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
```

remove 42

```
summary(lm(bodyfat ~ Age +Weight + Height+Thigh,data=body[-42,]))
```

```
##
## Call:
## lm(formula = bodyfat ~ Age + Weight + Height + Thigh, data = body[-42,
##     ])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -22.2729  -3.7828  -0.0947   3.9254  13.0096
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 34.86048   14.18928   2.457   0.0147 *
## Age          0.17168    0.03284   5.228 3.66e-07 ***
## Weight       0.17257    0.03019   5.717 3.13e-08 ***
## Height      -1.02550    0.17072  -6.007 6.77e-09 ***
## Thigh        0.29942    0.15824   1.892   0.0596 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.583 on 246 degrees of freedom
## Multiple R-squared:  0.559,  Adjusted R-squared:  0.5519
## F-statistic: 77.96 on 4 and 246 DF,  p-value: < 2.2e-16
```

remove both 42 and 39

```
summary(lm(bodyfat ~ Age +Weight + Height+Thigh,data=body[c(-42,-39),]))
```

```
##
## Call:
## lm(formula = bodyfat ~ Age + Weight + Height + Thigh, data = body[c(-42,
##     -39), ])
##
## 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
```

We see an increase R^2 in three models