151AHW3

Jiyoon Clover Jeong

10/13/2017

1 Ant Colonies

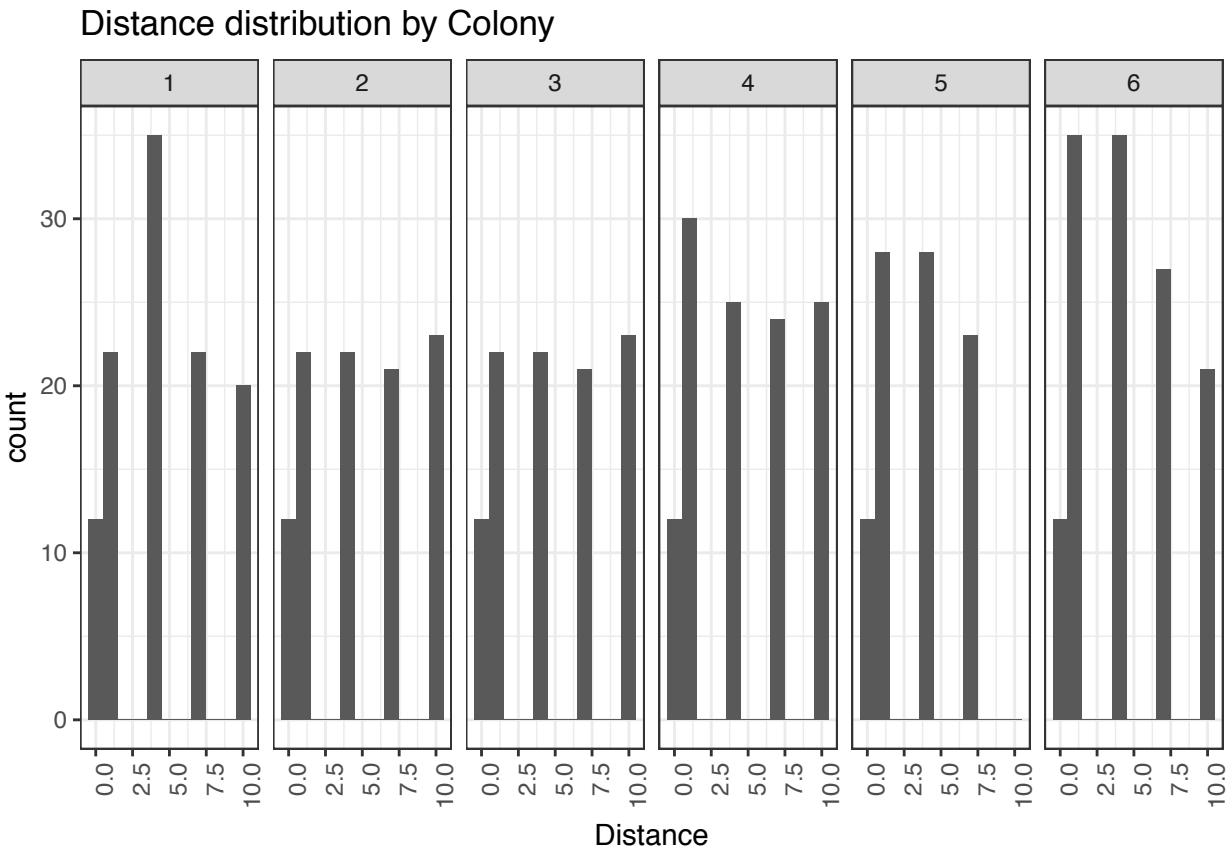
```
ant <- read.table(file="/Users/cloverjiyoon/2017Fall/Stat 151A/HW/HW3/thatch-ant.dat.txt", header=T, sep
head(data)
```

```
##
## 1 function (..., list = character(), package = NULL, lib.loc = NULL,
## 2     verbose = getOption("verbose"), envir = .GlobalEnv)
## 3 {
## 4     fileExt <- function(x) {
## 5         db <- grepl("\\\\.[^.]+\\\\.(gz|bz2|xz)$", x)
## 6         ans <- sub(".*\\\\.", "", x)
```

```
ant <- na.omit(ant)

ant$Headwidth..mm. <- NULL


ant <- ant[ant$Colony %in% c("1","2","3","4","5","6"),]
ant$Colony <- as.factor(ant$Colony)
```

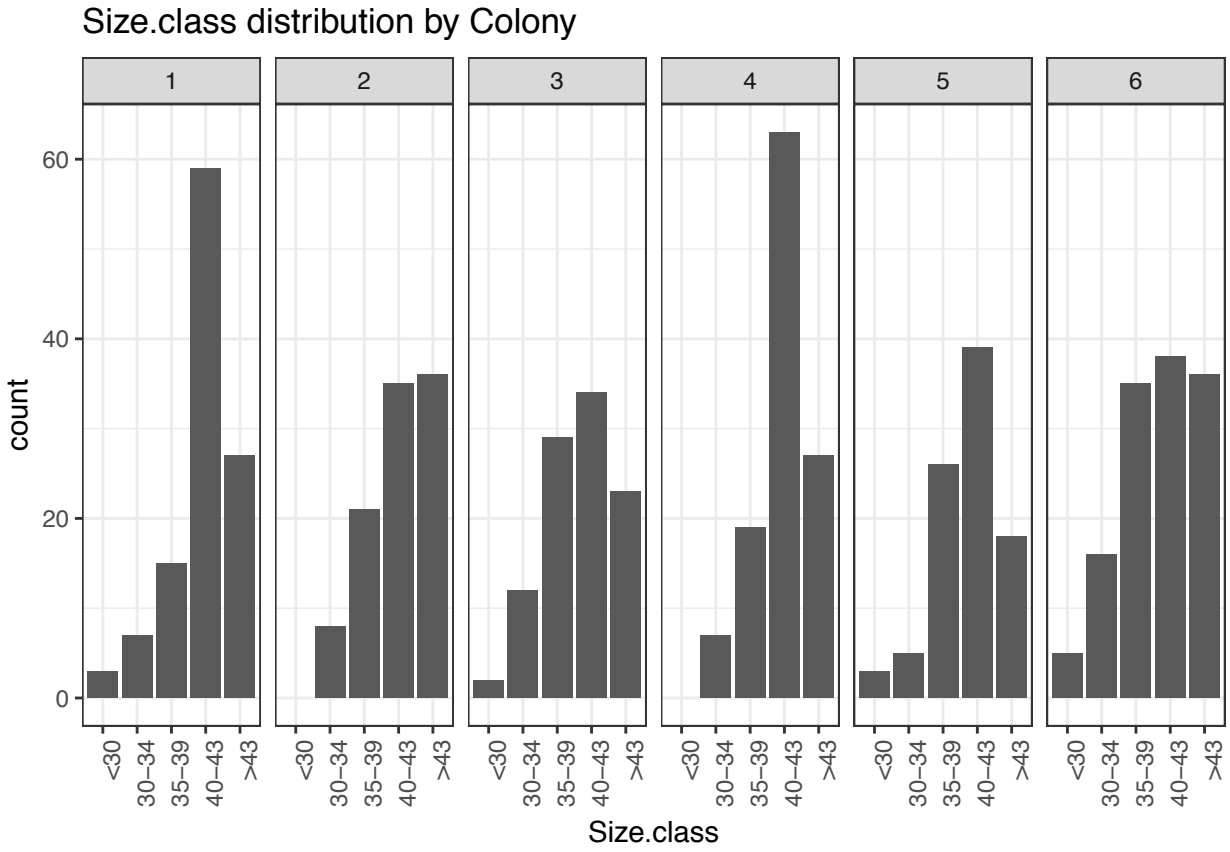
(a)

```
ggplot(ant,aes(x=Distance))+geom_histogram(binwidth =1)+facet_grid(~Colony)+theme_bw() + labs(title = "
```



Colony 1 sends large amount of ants to Distance '4' while Colony 2 and 3 sends equal amounts of ants to each distance. The graph suggests that colony 2 and 3 ahs similar Distance distributions. Colony 4 and 6 like to keep workers near and Colony 5 never send their works far away.

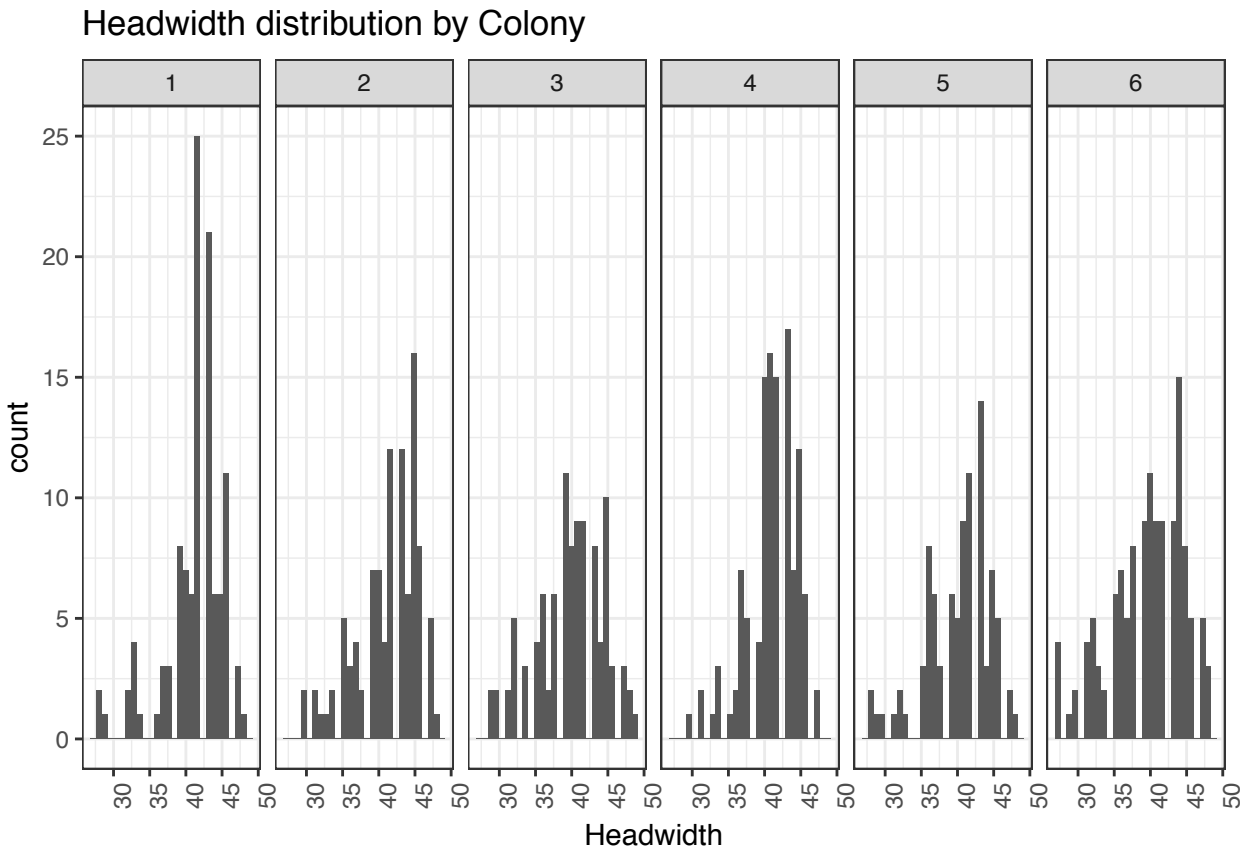
```
ant$Size.class <- factor(ant$Size.class, levels = c("<30", "30-34","35-39","40-43",">43"))

ggplot(ant,aes(x=Size.class))+
  geom_bar(aes(y = ..count..))+facet_grid(~Colony)+theme_bw() + labs(title = "Size.class distribution by
```



I will define size 1 : <30 size 2 : 30-34 size 3 : 35 - 39 size 4 : 40 - 43 size 5 : >43

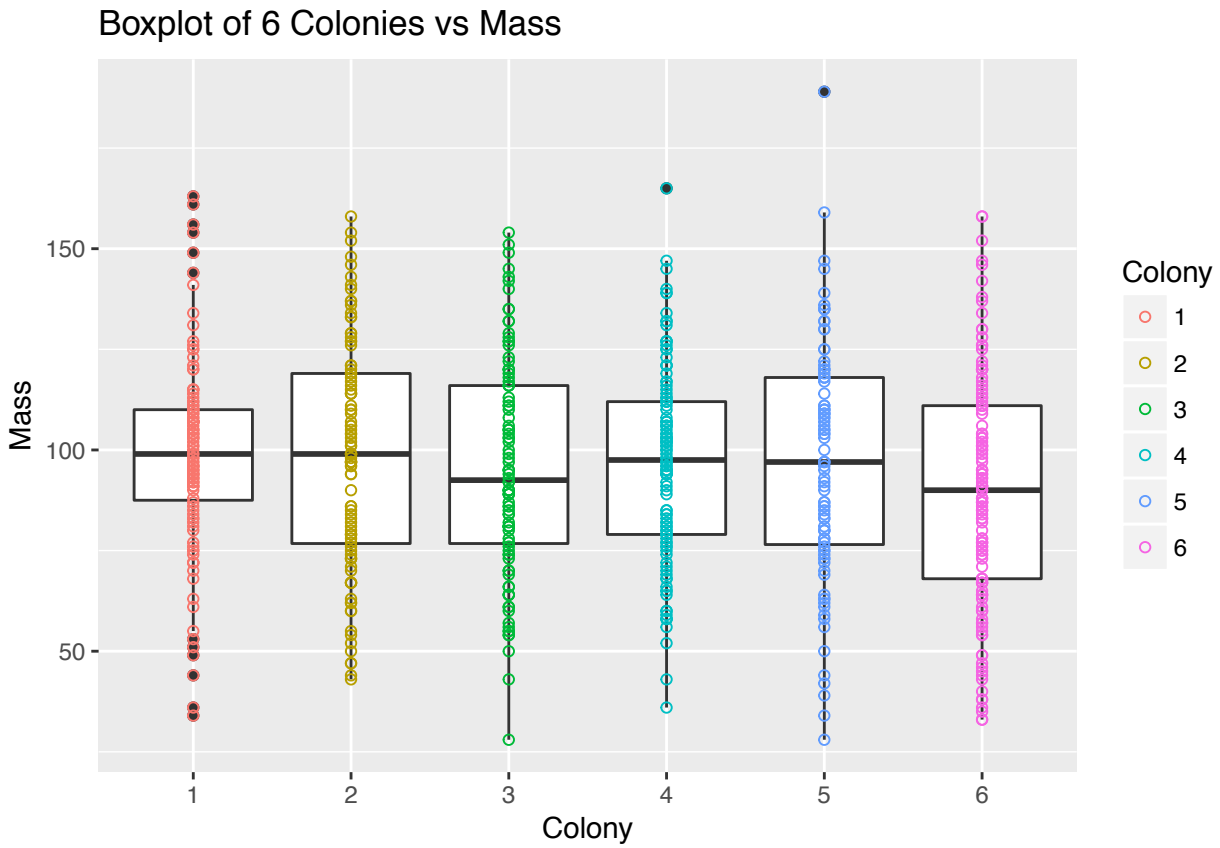
Colony 1 and 4 have the similar distribution of ant's size since they have huge amounts of size 4 workers. Colony 2 and 6 also have the similar distribution since they have more big size workers than the small size workers. Colony 3 and 5 also have a smiliar distribution but colony 5 has more size 4 workers.

```
ggplot(ant,aes(x=Headwidth))+
  geom_histogram(binwidth = 0.8, aes(y = ..count..))+ facet_grid(~Colony)+theme_bw() + labs(title = "Hea
```

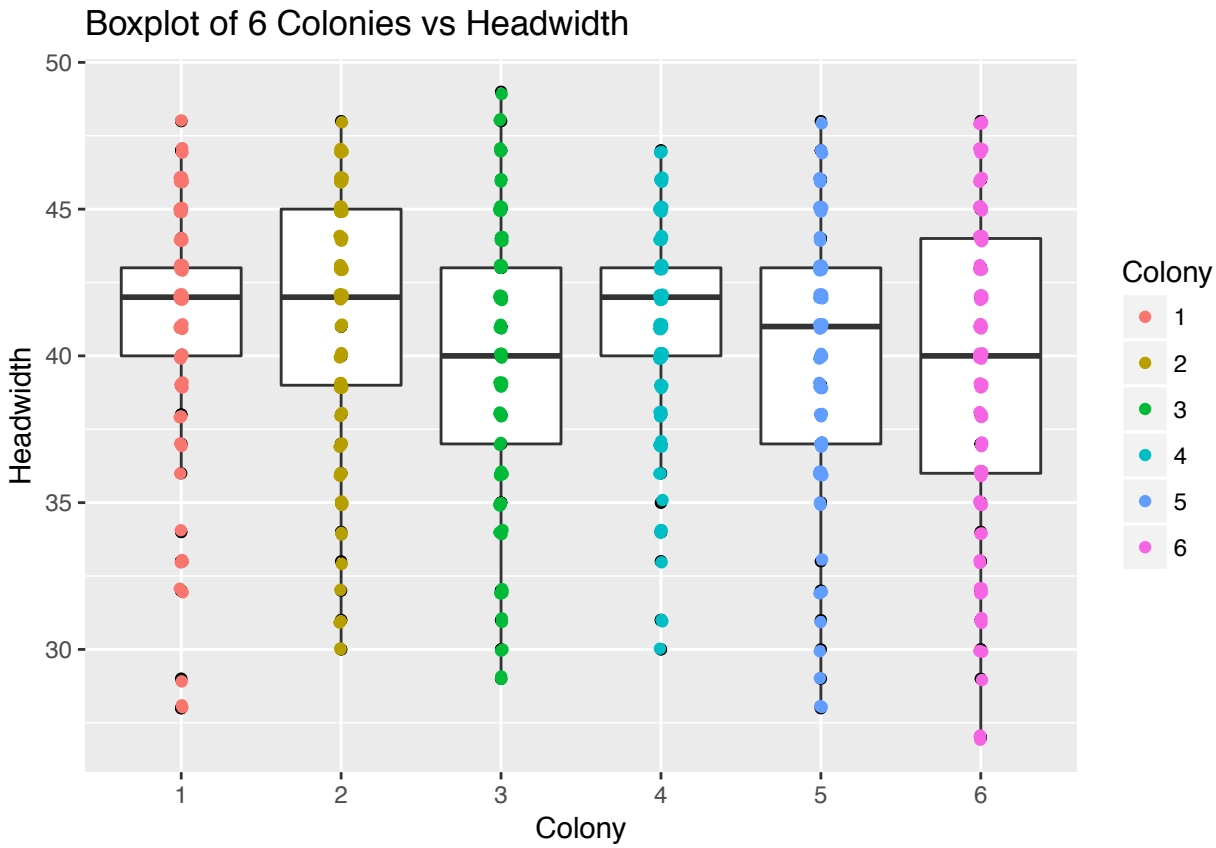



```
# ggplot(ant[ant$Colony == "1",],aes(x=Distance))+geom_histogram(binwidth =1)+theme_bw() + labs(title =
```

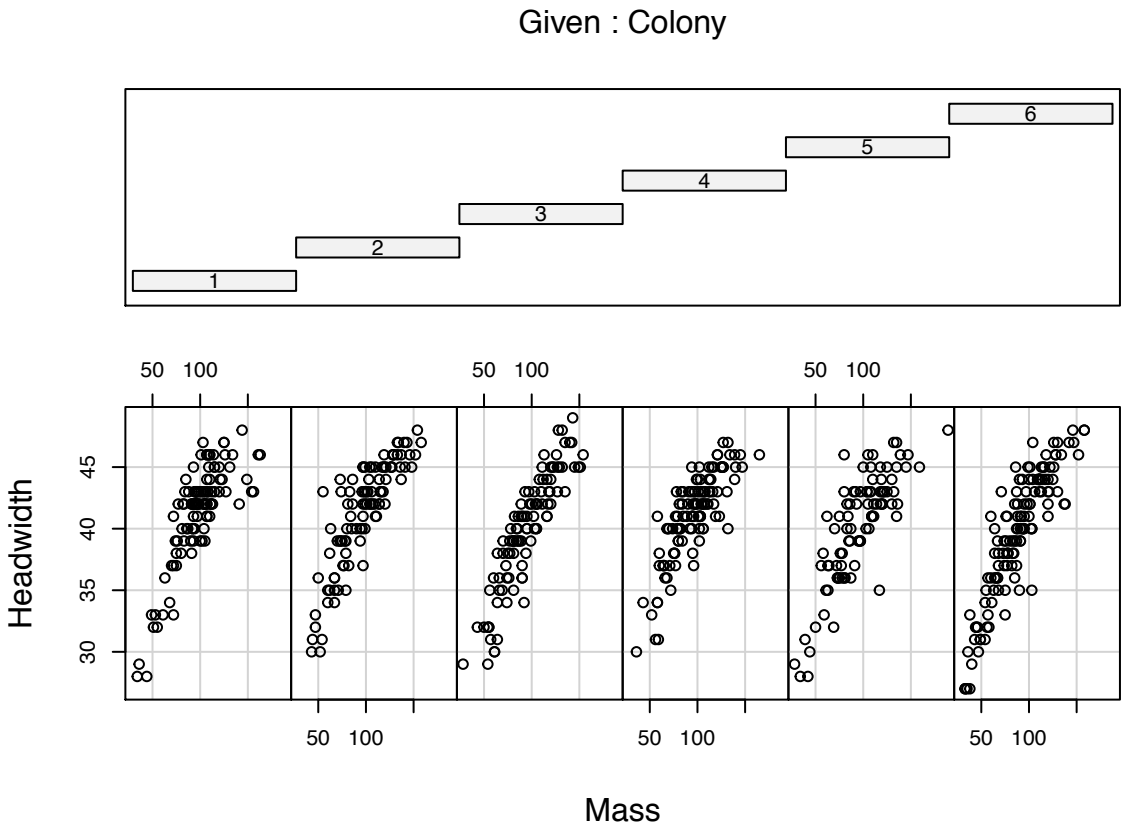
```
ggplot(data = ant, aes(x=Colony, y = Mass)) + geom_boxplot() + geom_point(aes(y=Mass, x=Colony, color=
```



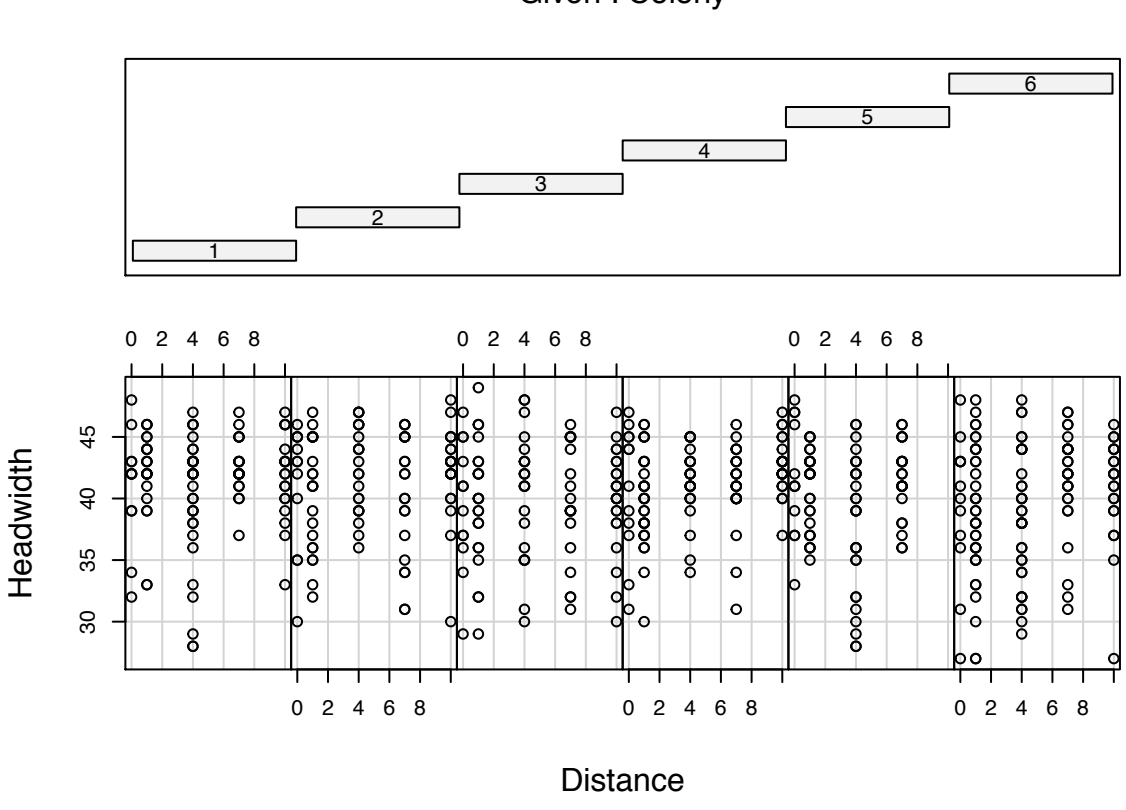

```
ggplot(data = ant, aes(x=Colony, y = Headwidth))+ geom_boxplot() + geom_point(aes(x=Colony, y=Headwidth
```



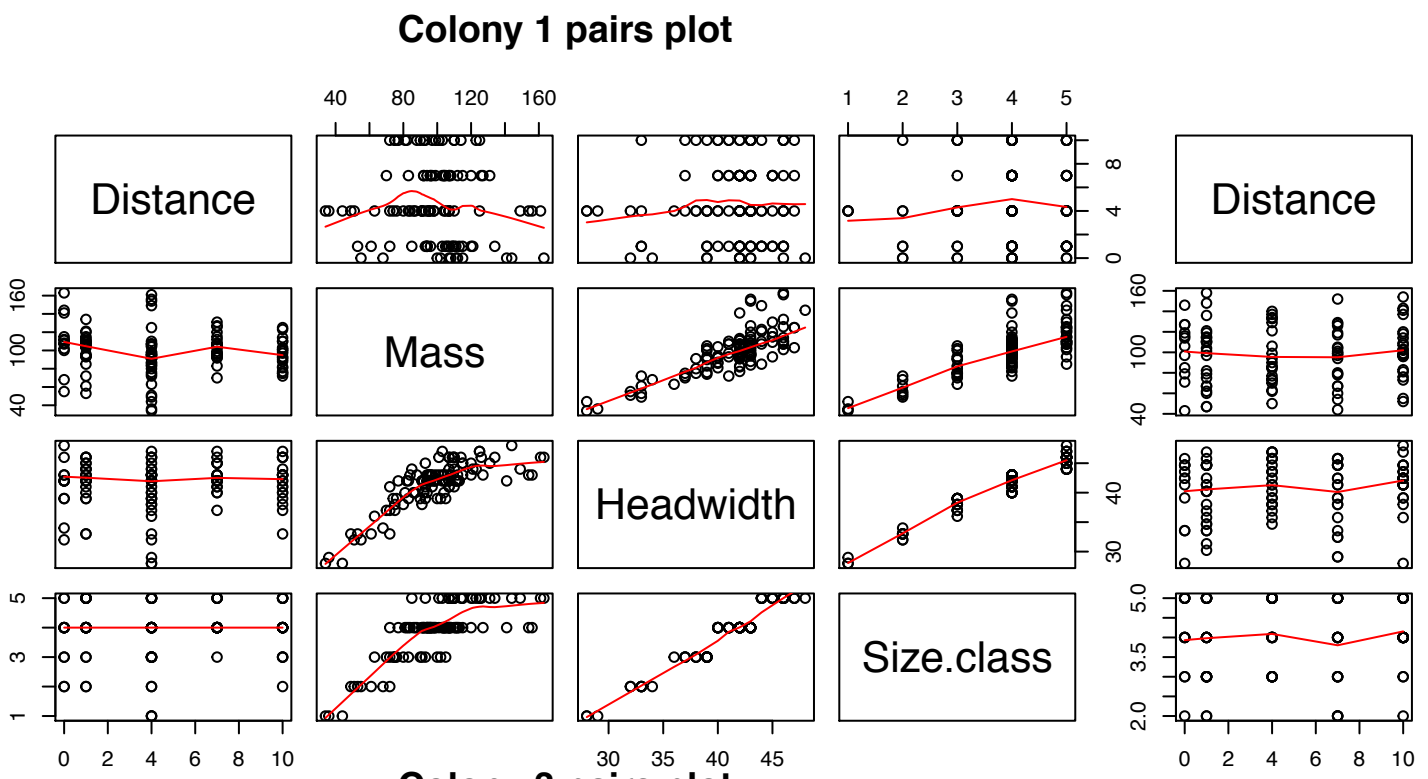

```
coplot( Headwidth ~ Mass | Colony, data=ant, rows=1)
```

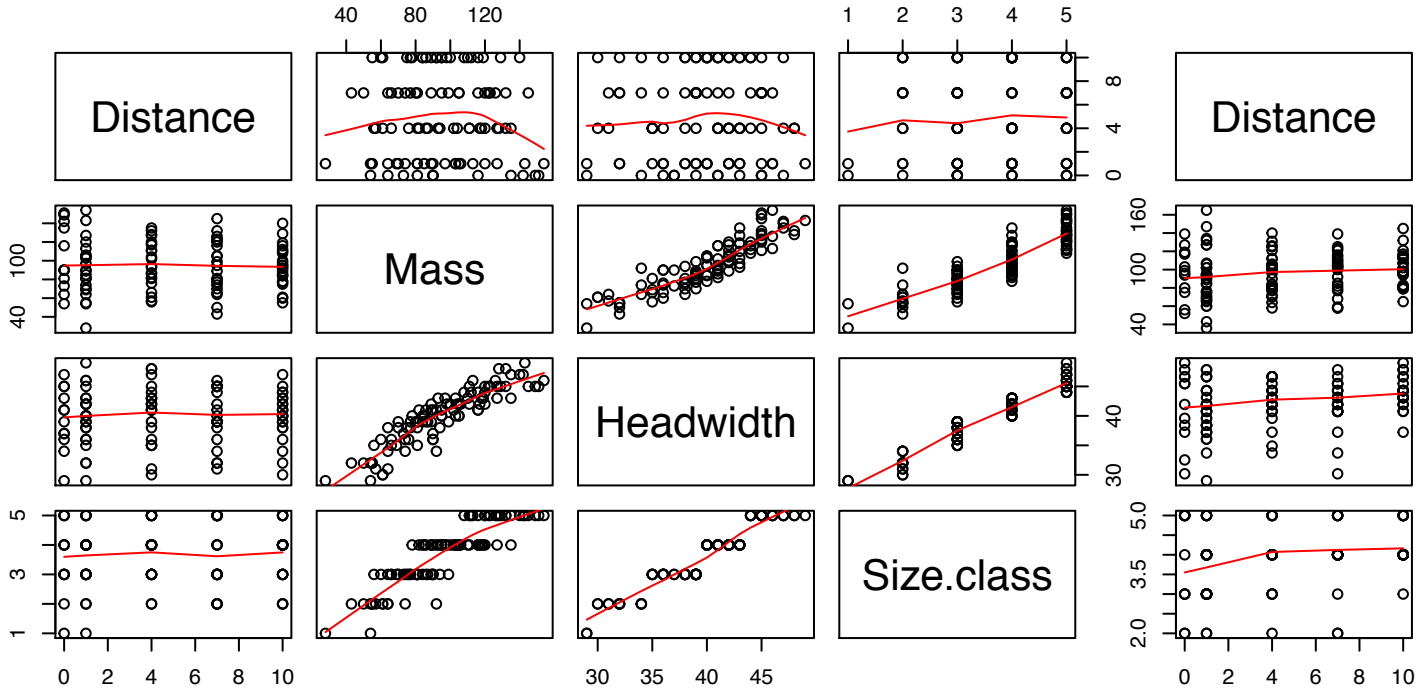


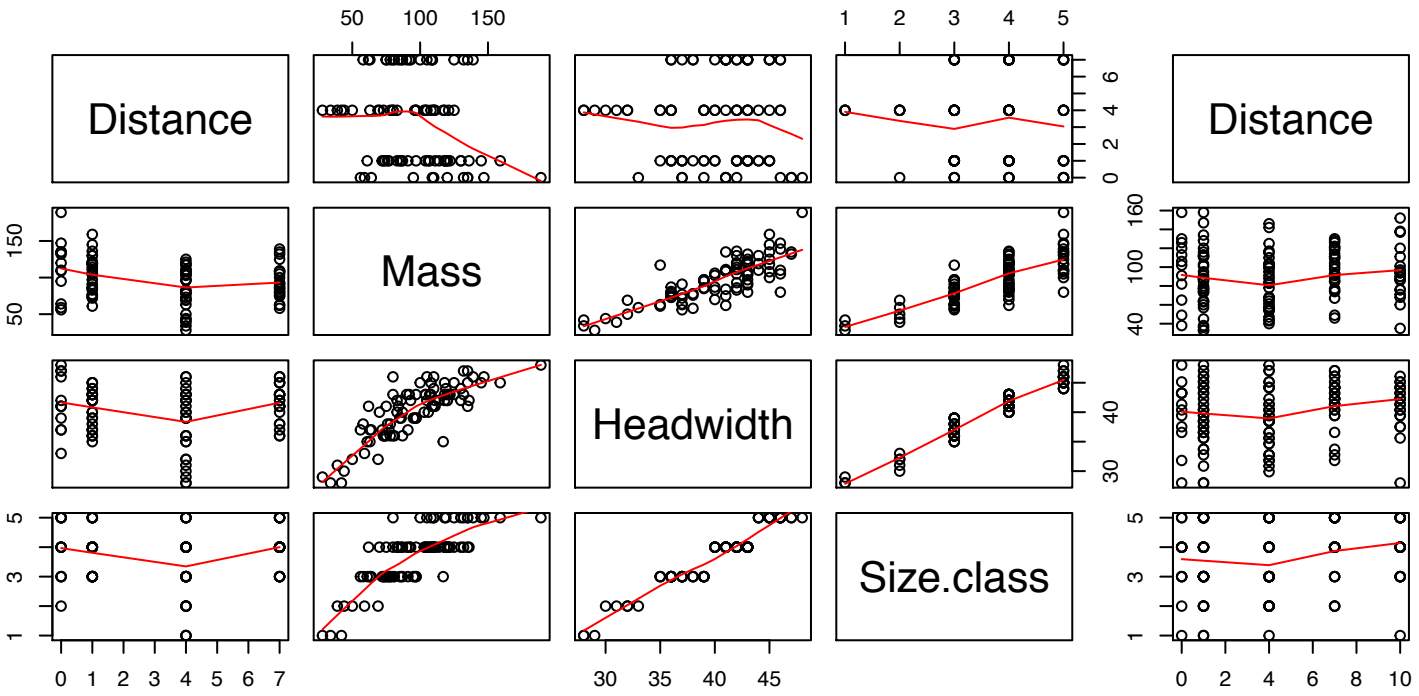

```
coplot(Headwidth ~ Distance | Colony, data = ant, rows= 1)
```



```
pairsgraph <- sapply(levels(ant$Colony), function(x) {
  pairs(subset(ant, Colony == x, select=-Colony),
        panel=panel.smooth, main= paste("Colony", x, "pairs plot"))
})
```







(b)

```
ant$Distance <- as.factor(ant$Distance)


# Q: add intercept??
fullfit <- lm(Mass ~. -1, data=ant)
summary(fullfit)
```

```
##
## Call:
## lm(formula = Mass ~ . - 1, data = ant)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -47.804  -8.022  -0.555   8.003  51.511
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## Colony1         -98.8873    13.7297  -7.202 1.69e-12 ***
## Colony2         -99.5553    13.5927  -7.324 7.34e-13 ***
## Colony3         -94.7184    13.5871  -6.971 7.92e-12 ***
## Colony4         -99.7989    13.5455  -7.368 5.44e-13 ***
## Colony5         -95.2742    13.5730  -7.019 5.76e-12 ***
## Colony6         -99.2774    13.4595  -7.376 5.14e-13 ***
## Distance1        -6.6332     1.9680  -3.371 0.000796 ***
## Distance4        -9.5713     1.9502  -4.908 1.17e-06 ***
## Distance7        -9.4464     2.0165  -4.685 3.44e-06 ***
```

```
## Distance10      -10.5882     2.1188  -4.997 7.54e-07 ***
## Headwidth         5.0542     0.4555  11.097  < 2e-16 ***
## Size.class30-34  -2.2907     4.6612  -0.491 0.623278
## Size.class35-39  -6.4351     5.8475  -1.100 0.271539
## Size.class40-43  -4.3846     7.3629  -0.596 0.551722
## Size.class>43    -0.2594     8.8522  -0.029 0.976627
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.77 on 633 degrees of freedom
## Multiple R-squared:  0.9812, Adjusted R-squared:  0.9808
## F-statistic:  2207 on 15 and 633 DF,  p-value: < 2.2e-16
```

```
fullfitR2 <- round(x=summary(fullfit)$adj.r.squared, digits=5)


cat("Adjusted R^2 for model including all variables in ant data is ", fullfitR2, "\n")
```

```
## Adjusted R^2 for model including all variables in ant data is  0.98079
```

```
#smallfit <- lm(Mass ~ 0 + Colony + Size.class + Distance, data=ant)

smallfit <- lm(Mass ~ Colony + Size.class + Distance -1, data=ant)
summary(smallfit)
```

```
##
## Call:
## lm(formula = Mass ~ Colony + Size.class + Distance - 1, data = ant)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -46.728  -8.899  -0.431   8.231  56.492
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## Colony1           45.865      4.677   9.807  < 2e-16 ***
## Colony2           43.221      4.787   9.029  < 2e-16 ***
## Colony3           48.259      4.709  10.248  < 2e-16 ***
## Colony4           42.512      4.761   8.929  < 2e-16 ***
## Colony5           47.602      4.690  10.149  < 2e-16 ***
## Colony6           42.550      4.608   9.234  < 2e-16 ***
## Size.class30-34   18.171      4.675   3.887 0.000112 ***
## Size.class35-39   40.694      4.389   9.271  < 2e-16 ***
## Size.class40-43   64.416      4.337  14.853  < 2e-16 ***
## Size.class>43     87.280      4.386  19.901  < 2e-16 ***
## Distance1         -8.017      2.145  -3.738 0.000202 ***
## Distance4        -10.773      2.126  -5.066 5.33e-07 ***
## Distance7        -10.644      2.199  -4.840 1.63e-06 ***
## Distance10       -11.208      2.313  -4.845 1.59e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15.04 on 634 degrees of freedom
## Multiple R-squared:  0.9776, Adjusted R-squared:  0.9771
## F-statistic:  1975 on 14 and 634 DF,  p-value: < 2.2e-16
```

```r
smallfitR2 <- round(x=summary(smallfit)$adj.r.squared, digits=5)
cat("Adjusted R^2 for model only including variable Colony, Distance, and Size.class in ant data is ", s
```

```
## Adjusted R^2 for model only including variable Colony, Distance, and Size.class in ant data is  0.97
```

```r
# check if adding headwidth is appropriate
smallfit2 <- lm(Headwidth ~  Colony + Size.class + Distance -1, data=ant)
summary(smallfit2)
```

```
##
## Call:
## lm(formula = Headwidth ~ Colony + Size.class + Distance - 1,
##     data = ant)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.5736 -0.9709 -0.0154  0.9846  3.6651
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## Colony1          28.6399     0.3734  76.699   <2e-16 ***
## Colony2          28.2489     0.3822  73.913   <2e-16 ***
## Colony3          28.2887     0.3760  75.239   <2e-16 ***
## Colony4          28.1567     0.3802  74.066   <2e-16 ***
## Colony5          28.2686     0.3745  75.488   <2e-16 ***
## Colony6          28.0611     0.3679  76.274   <2e-16 ***
## Size.class30-34   4.0485     0.3733  10.846   <2e-16 ***
## Size.class35-39   9.3247     0.3505  26.607   <2e-16 ***
## Size.class40-43  13.6124     0.3463  39.311   <2e-16 ***
## Size.class>43    17.3200     0.3502  49.461   <2e-16 ***
## Distance1        -0.2738     0.1713  -1.599    0.110
## Distance4        -0.2378     0.1698  -1.400    0.162
## Distance7        -0.2369     0.1756  -1.349    0.178
## Distance10       -0.1226     0.1847  -0.664    0.507
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.201 on 634 degrees of freedom
## Multiple R-squared:  0.9992, Adjusted R-squared:  0.9991
## F-statistic: 5.346e+04 on 14 and 634 DF,  p-value: < 2.2e-16
```

```r
# Q
smallfit3 <- lm(Headwidth + I(Headwidth^2) ~ . -1, data=ant[, -3])
summary(smallfit3)
```

```
##
## Call:
## lm(formula = Headwidth + I(Headwidth^2) ~ . - 1, data = ant[,
##     -3])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -194.28  -72.43   -1.18   76.28  347.15
##
## Coefficients:
```

```
##                 Estimate Std. Error t value Pr(>|t|)
## Colony1           862.16      30.49  28.273  < 2e-16 ***
## Colony2           832.30      31.21  26.666  < 2e-16 ***
## Colony3           834.94      30.70  27.192  < 2e-16 ***
## Colony4           822.36      31.05  26.489  < 2e-16 ***
## Colony5           834.17      30.58  27.277  < 2e-16 ***
## Colony6           816.99      30.04  27.193  < 2e-16 ***
## Distance1         -23.40      13.99  -1.673   0.0947 .
## Distance4         -20.07      13.87  -1.447   0.1483
## Distance7         -22.31      14.34  -1.556   0.1202
## Distance10        -11.11      15.08  -0.737   0.4615
## Size.class30-34   250.14      30.48   8.206 1.28e-15 ***
## Size.class35-39   621.98      28.62  21.732  < 2e-16 ***
## Size.class40-43   965.09      28.28  34.128  < 2e-16 ***
## Size.class>43    1291.32      28.60  45.156  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 98.08 on 634 degrees of freedom
## Multiple R-squared:  0.9969, Adjusted R-squared:  0.9968
## F-statistic: 1.457e+04 on 14 and 634 DF,  p-value: < 2.2e-16
```

```
# smallfit3 <- lm( I(Headwidth^2) ~ . -1, data=ant[, -c(3,5)])
# summary(smallfit3)



fullfit2 <- lm(Mass ~ . -1 + I(Headwidth^2), data=ant)
summary(fullfit2)
```

```
##
## Call:
## lm(formula = Mass ~ . - 1 + I(Headwidth^2), data = ant)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -48.397  -7.714  -0.633   7.835  50.652
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## Colony1         162.47770   76.03678   2.137  0.03300 *
## Colony2         161.47260   75.91767   2.127  0.03381 *
## Colony3         166.41567   75.94658   2.191  0.02880 *
## Colony4         161.66773   76.03296   2.126  0.03387 *
## Colony5         165.70549   75.90061   2.183  0.02939 *
## Colony6         161.76229   75.89766   2.131  0.03345 *
## Distance1        -6.42364    1.95175  -3.291  0.00105 **
## Distance4        -9.43454    1.93351  -4.879 1.35e-06 ***
## Distance7        -8.89666    2.00505  -4.437 1.08e-05 ***
## Distance10      -10.38091    2.10114  -4.941 9.98e-07 ***
## Headwidth        -9.27095    4.12499  -2.248  0.02495 *
## Size.class30-34  11.74121    6.12200   1.918  0.05558 .
## Size.class35-39  17.69214    9.01601   1.962  0.05017 .
## Size.class40-43  20.63467   10.22500   2.018  0.04401 *
## Size.class>43    20.25507   10.55822   1.918  0.05551 .
```

```
## I(Headwidth^2)    0.17865    0.05113   3.494  0.00051 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.65 on 632 degrees of freedom
## Multiple R-squared:  0.9816, Adjusted R-squared:  0.9811
## F-statistic:  2106 on 16 and 632 DF,  p-value: < 2.2e-16
```

```
fullfit2R2 <- round(summary(fullfit2)$adj.r.squared, 5)


# check removing Headwidth..mm. is appropriate
fullfit3 <- lm(Mass~ Colony + Distance + Headwidth + Size.class, data = ant)
summary(fullfit3)$adj.r.squared
```

```
## [1] 0.7407833
```

```
summary(fullfit)$adj.r.squared
```

```
## [1] 0.980793
```

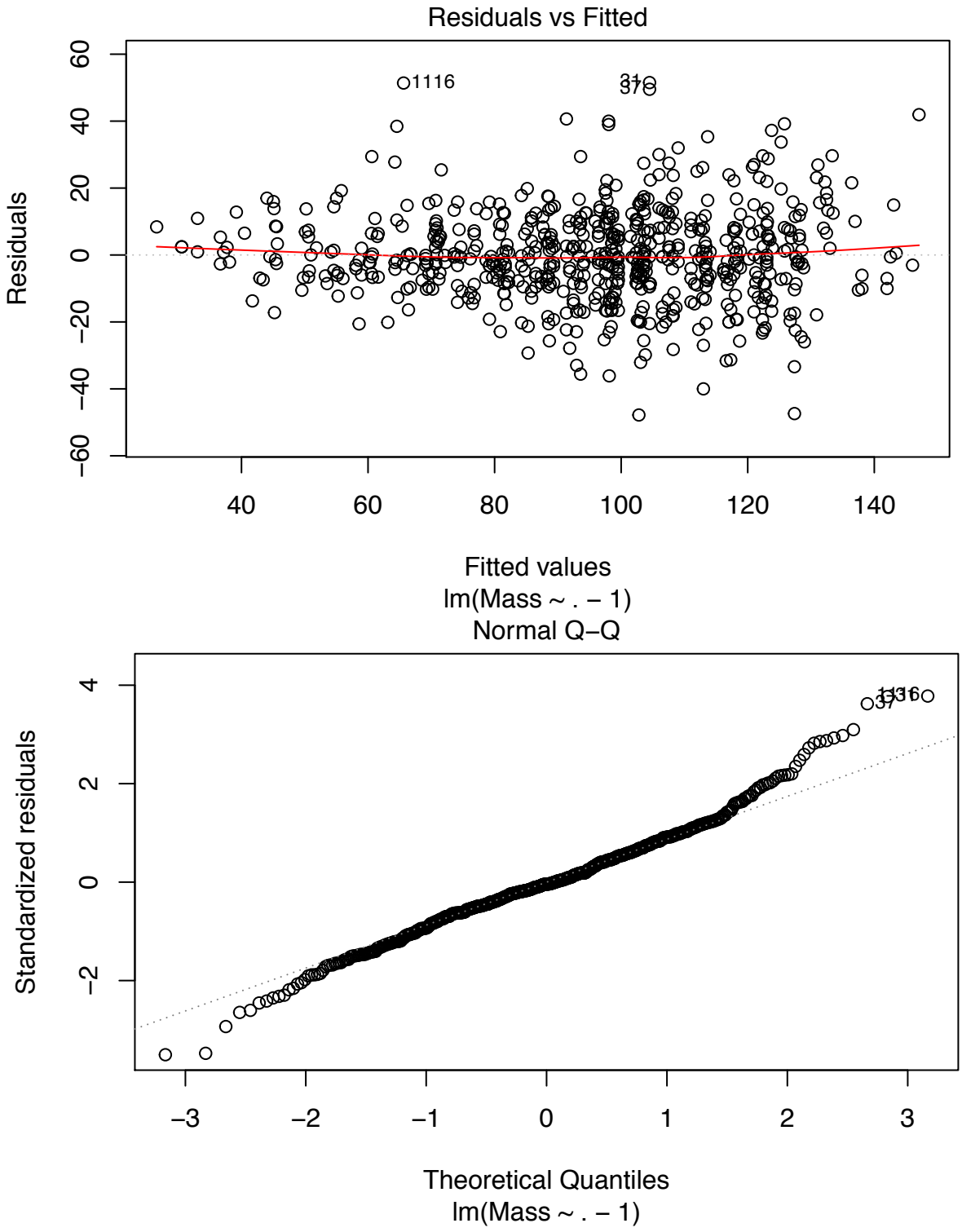
```
cat("Adjusted R^2 for model including variable Colony, Distance, and Size as well as variable 'Headwidth
```

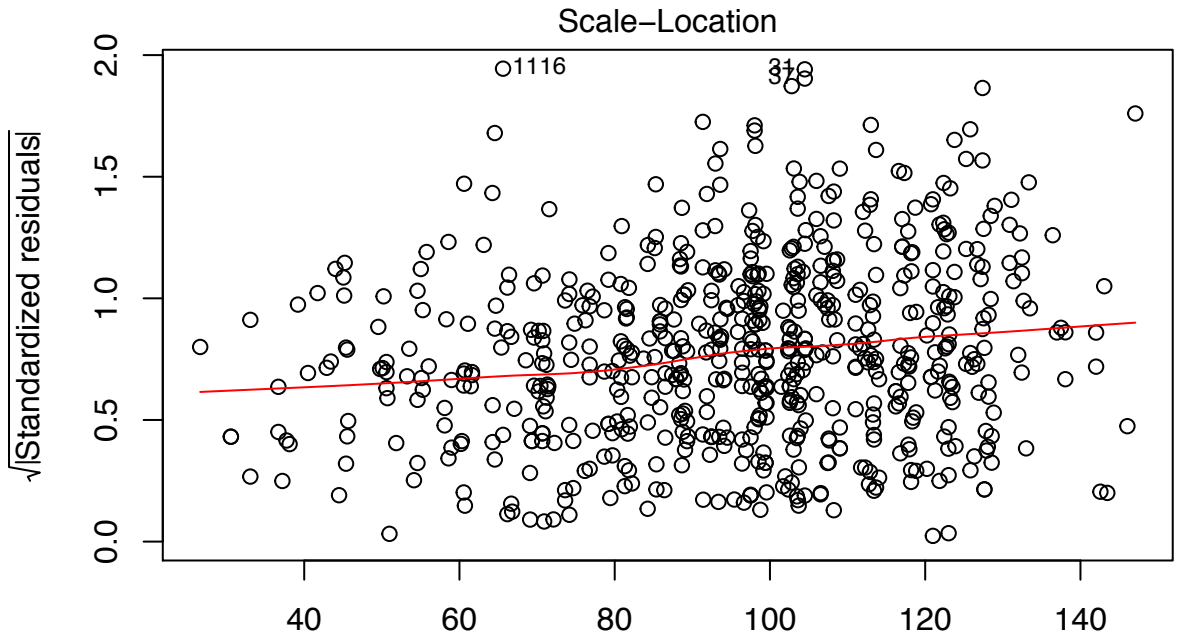
```
## Adjusted R^2 for model including variable Colony, Distance, and Size as well as variable 'Headwidth^
##  This is bigger than Adjusted R^2 for model including variable Colony, Distance, and Size.class whic
```

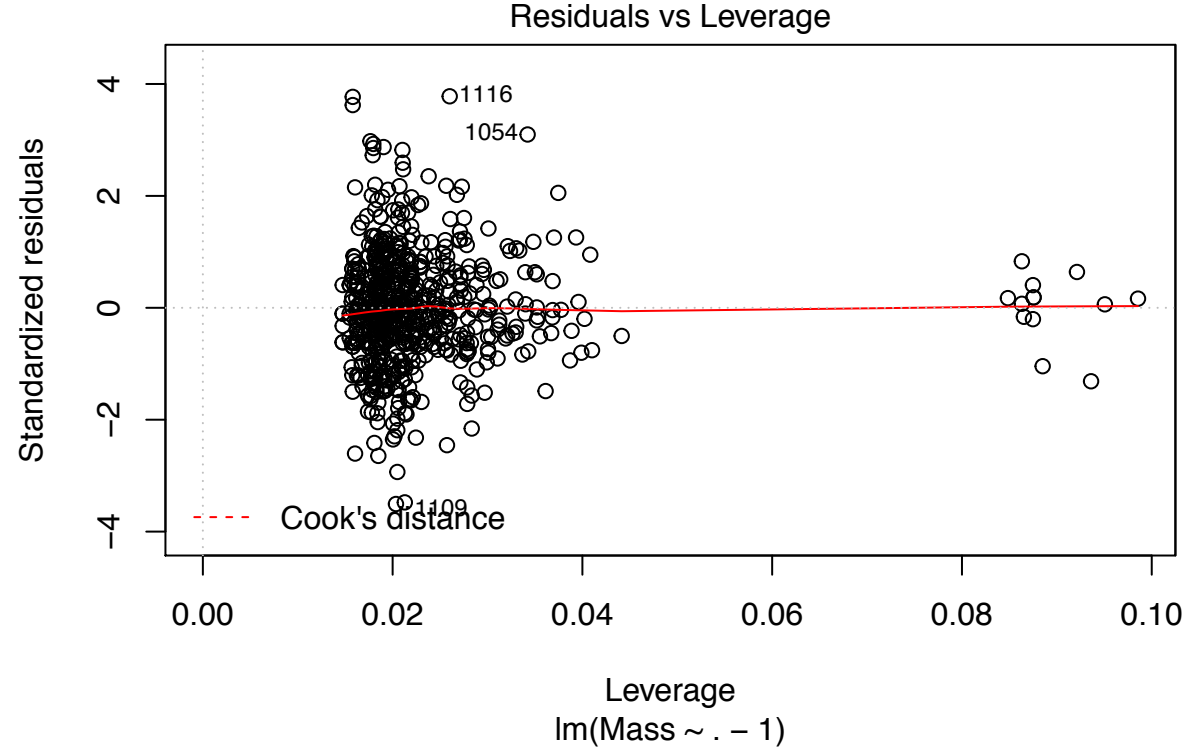
Since the modified model including Colony, Distance, size, and Headwidth^2 has the highest adjusted R^2 value, I conclude that this transformation gives us more accurate fit. Also, from the code above, we can see that removing Headwidth..mm. variable is not a good idea since the adjusted R^2 becomes around 0.7 (Originall around 0.9).

(b) Visualization - graphical techniques

```
plot(fullfit)
```

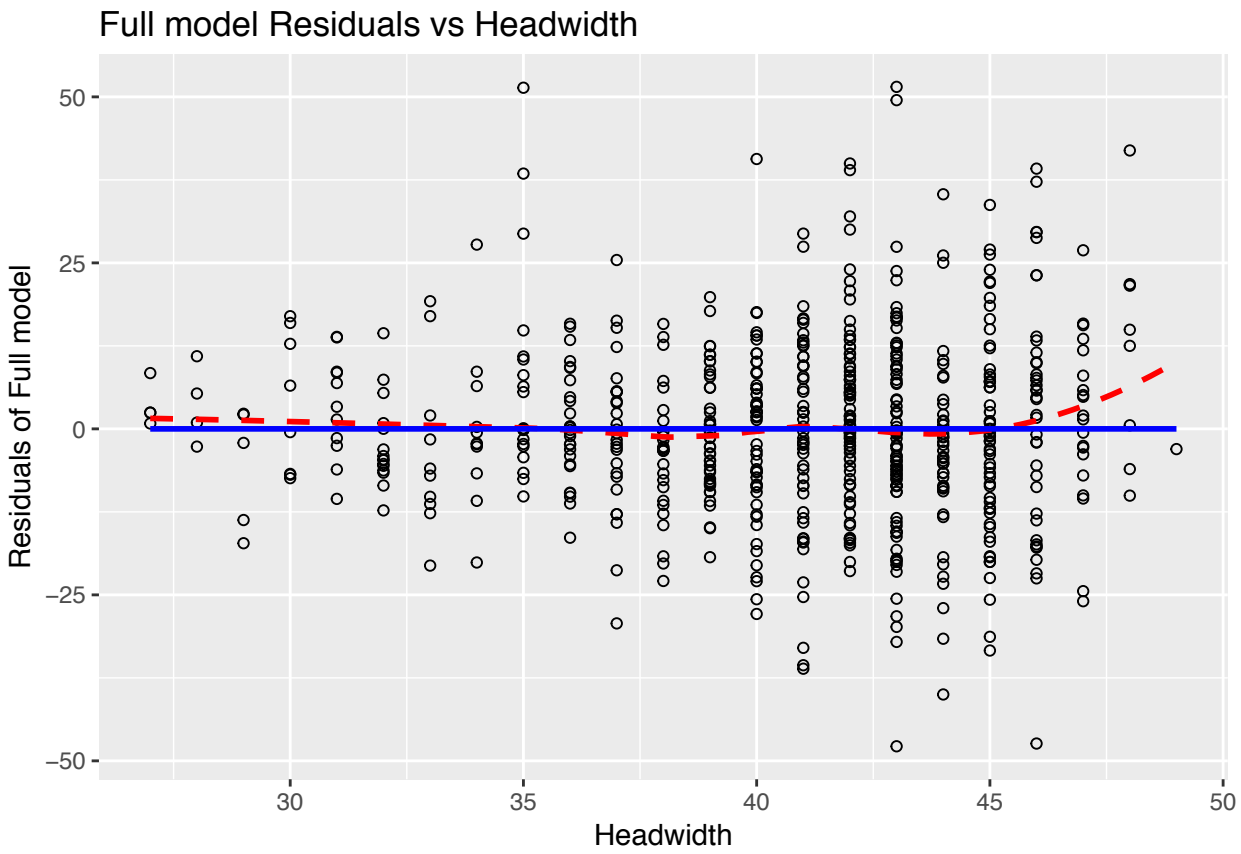

Residuals vs Fitted
Residuals
Fitted values
lm(Mass ~ . – 1)
Normal Q–Q
Standardized residuals
Theoretical Quantiles
lm(Mass ~ . – 1)



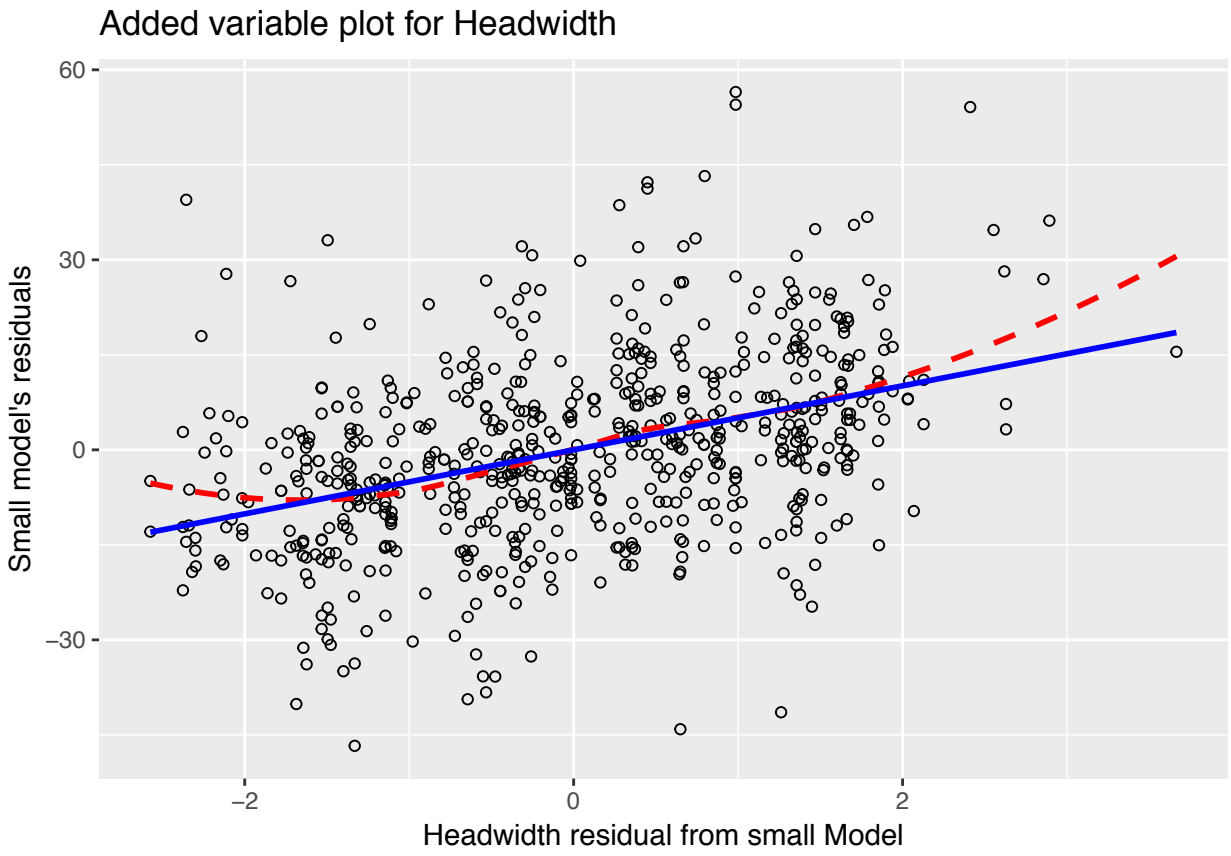



```
data <- data.frame(X=ant$Headwidth, Y=fullfit$residuals)


ggplot(data,aes(x=X,y=Y)) + geom_point(shape=1) +
  stat_smooth(method="loess", se=FALSE, color='red', lty=2) +
  stat_smooth(method="lm", se=FALSE, color='blue', alpha=0.65) +
```

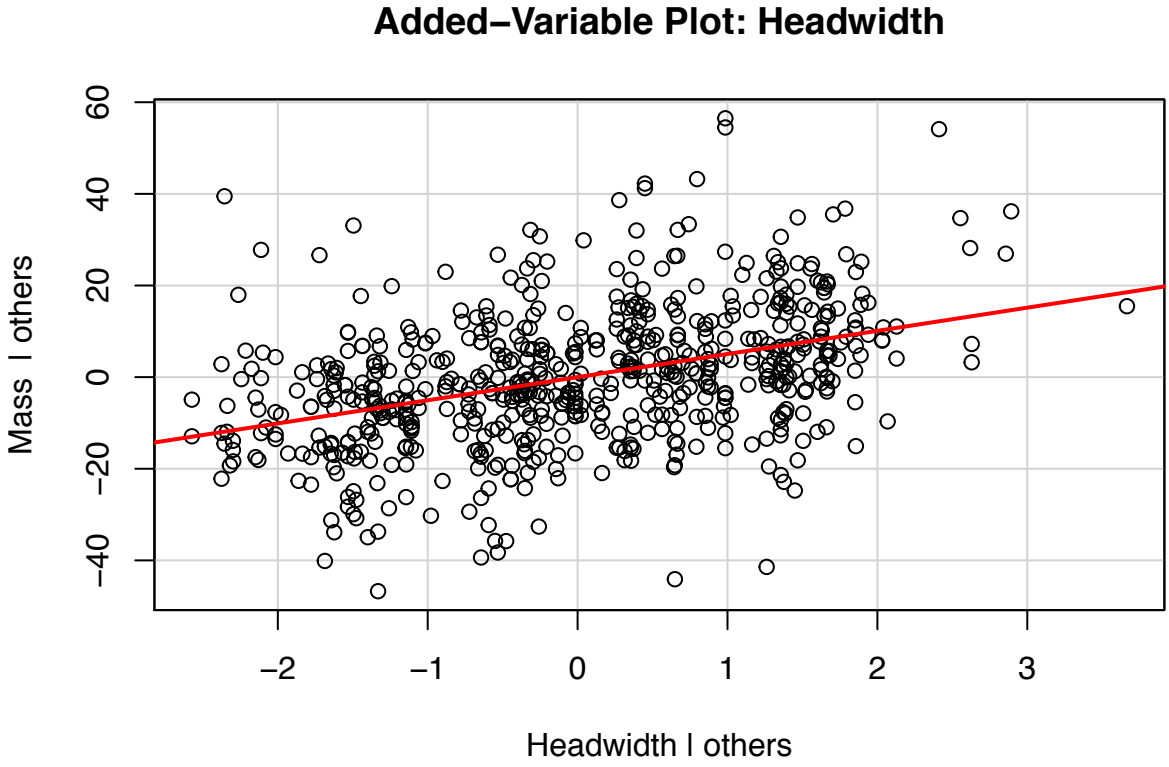
```
  labs(x="Headwidth",  y="Residuals of Full model",
       title="Full model Residuals vs Headwidth")
```




```
# Added variable plot / partial regression plot - Headwidth

# smallfit2 <- lm(Headwidth ~  Colony + Size.class + Distance -1, data=ant)
# smallfit <- lm(Mass ~ Colony + Size.class + Distance -1, data=ant)

data <- data.frame(X=smallfit2$residuals, Y=smallfit$residuals)

ggplot(data, aes(x=X,y=Y)) + geom_point(shape=1) +
  stat_smooth(method="loess", se=FALSE, color='red', lty=2) +
  stat_smooth(method="lm", se=FALSE, color='blue') +
  labs(x="Headwidth residual from small Model",
       y="Small model's residuals",
       title="Added variable plot for Headwidth")
```




```
avPlot(fullfit, variable = "Headwidth")
```



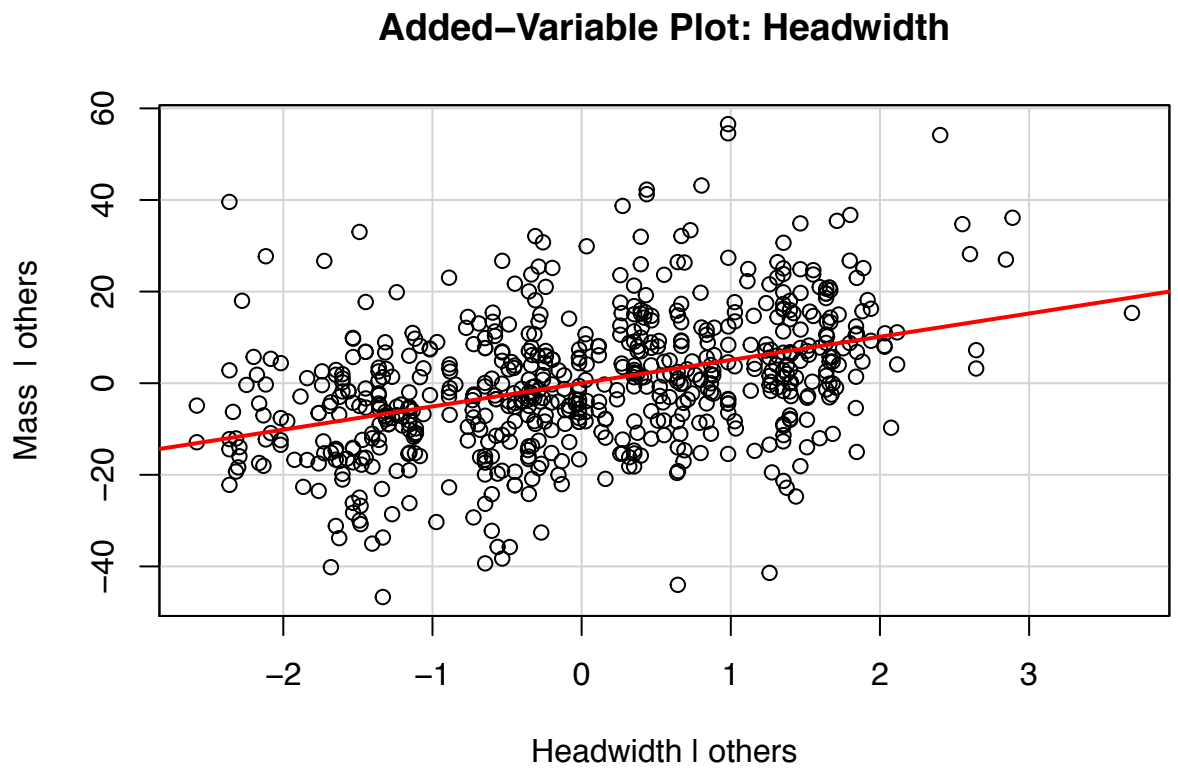
```r
#high leverage points
X <- model.matrix(Mass~., data= ant)
H <- X %*% solve(t(X) %*% X) %*% t(X)

lev.sorted <- sort(diag(H), decreasing=T, index.return=T)
rownames(ant)[lev.sorted$ix[1:3]]
```

```
## [1] "1023" "1039" "207"
```

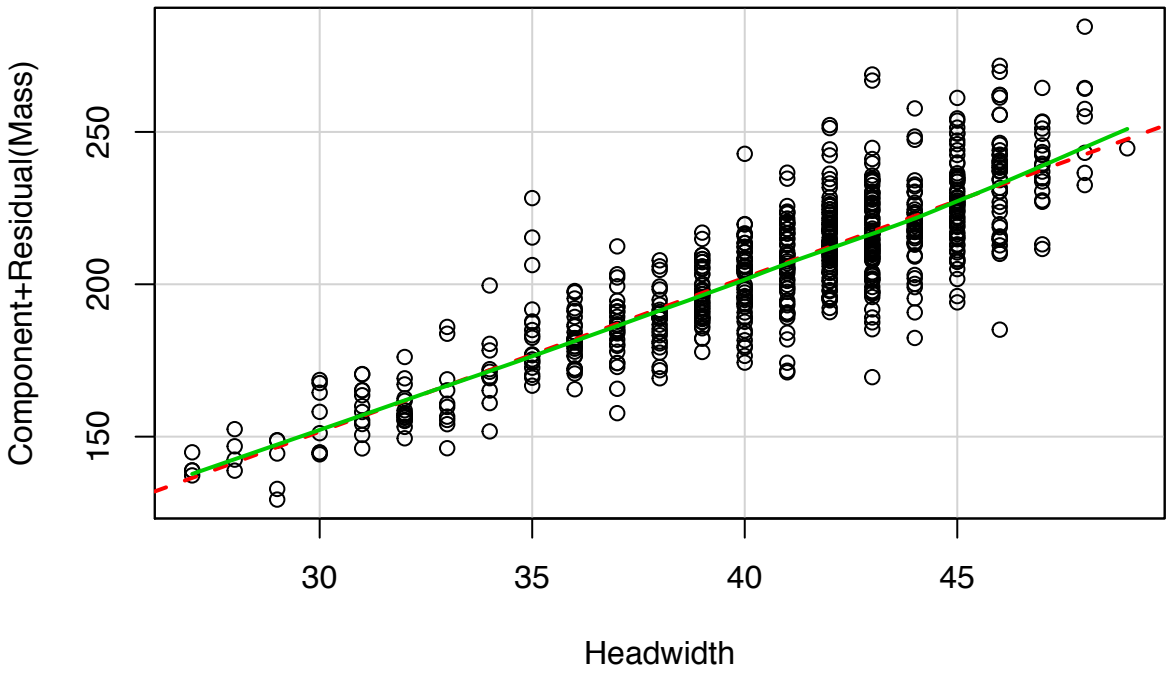
```r
ant_minus <- ant[-lev.sorted$ix[1:3],]
mod_del <- lm(Mass ~. , data=ant_minus)
coef(mod_del)
```

```
##     (Intercept)         Colony2         Colony3         Colony4
##     -98.0214095      -0.6231565       4.3697194      -0.8718741
##         Colony5         Colony6       Distance1       Distance4
##       3.5985392      -0.3985782      -6.4877290      -9.5791552
##       Distance7      Distance10       Headwidth Size.class30-34
##      -9.3995521     -10.5753053       5.0736344      -3.8723974
## Size.class35-39 Size.class40-43   Size.class>43
##      -8.1291571      -6.1399873      -2.0906395
```

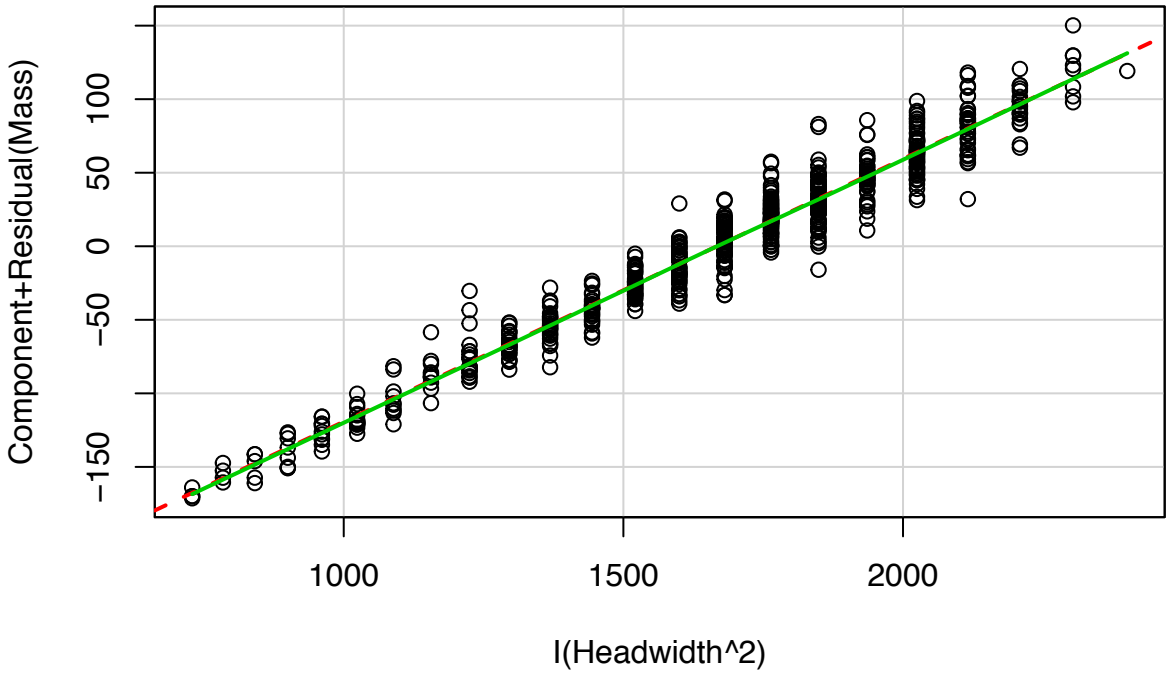
```r
avPlot(mod_del, variable = "Headwidth")
```




```r
crPlot(fullfit, "Headwidth")
```




```
crfit <- lm(Mass ~. + I(Headwidth^2), data = ant)

crPlot(crfit, "I(Headwidth^2)")
```



As we can see the smallfit and fullfit of the R^2 values from summary, I conclude that adding Headwidth^2 leads to a good fit. After that, I also checked it with added variable plot and component plus residual plot and the plots shows almost perfect linearity in both cases. Therefore, I conclude that we have to add Headwidth^2 variable as well as Colony, Distance, Headwidth, and Size.class in order to get a good lm fit.

(c) Interpret the coefficients relative to the scientific contributions and discuss what conclusions you can draw.

```
summary(fullfit2)
```

```
##
## Call:
## lm(formula = Mass ~ . - 1 + I(Headwidth^2), data = ant)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -48.397  -7.714  -0.633   7.835  50.652
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## Colony1         162.47770   76.03678   2.137  0.03300 *
## Colony2         161.47260   75.91767   2.127  0.03381 *
## Colony3         166.41567   75.94658   2.191  0.02880 *
## Colony4         161.66773   76.03296   2.126  0.03387 *
## Colony5         165.70549   75.90061   2.183  0.02939 *
## Colony6         161.76229   75.89766   2.131  0.03345 *
## Distance1        -6.42364    1.95175  -3.291  0.00105 **
## Distance4        -9.43454    1.93351  -4.879 1.35e-06 ***
## Distance7        -8.89666    2.00505  -4.437 1.08e-05 ***
## Distance10      -10.38091    2.10114  -4.941 9.98e-07 ***
## Headwidth        -9.27095    4.12499  -2.248  0.02495 *
## Size.class30-34  11.74121    6.12200   1.918  0.05558 .
## Size.class35-39  17.69214    9.01601   1.962  0.05017 .
## Size.class40-43  20.63467   10.22500   2.018  0.04401 *
## Size.class>43    20.25507   10.55822   1.918  0.05551 .
## I(Headwidth^2)    0.17865    0.05113   3.494  0.00051 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.65 on 632 degrees of freedom
## Multiple R-squared:  0.9816, Adjusted R-squared:  0.9811
## F-statistic:  2106 on 16 and 632 DF,  p-value: < 2.2e-16
```

```
for(i in 1:6){
  col1 <- ant[ant$Colony == i,]
  col1fit <- lm(Mass ~ Distance + Headwidth + Size.class, data = col1)
  cat("Summary statistics for Colony", i)
  print(summary(col1fit))
}
```

```
## Summary statistics for Colony 1
## Call:
## lm(formula = Mass ~ Distance + Headwidth + Size.class, data = col1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -26.414  -8.001  -1.837   4.658  50.651
##
## Coefficients:
```

```
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)     -73.6886    41.9216  -1.758 0.081816 .
## Distance1       -15.7381     5.1856  -3.035 0.003059 **
## Distance4       -14.6927     4.8682  -3.018 0.003220 **
## Distance7       -14.3449     5.2636  -2.725 0.007573 **
## Distance10      -19.7046     5.2399  -3.760 0.000284 ***
## Headwidth         4.4605     1.4116   3.160 0.002082 **
## Size.class30-34  -2.9332    11.7266  -0.250 0.802994
## Size.class35-39   0.4158    16.5231   0.025 0.979974
## Size.class40-43   1.9278    21.1207   0.091 0.927455
## Size.class>43     4.5785    25.8171   0.177 0.859595
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 14.04 on 101 degrees of freedom
## Multiple R-squared:  0.6869, Adjusted R-squared:  0.659
## F-statistic: 24.62 on 9 and 101 DF,  p-value: < 2.2e-16
##
## Summary statistics for Colony 2
## Call:
## lm(formula = Mass ~ Distance + Headwidth + Size.class, data = col1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -48.895  -7.318  -0.725  10.381  22.685
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)     -130.894     36.880  -3.549 0.000614 ***
## Distance1         -1.740      5.059  -0.344 0.731664
## Distance4        -13.639      5.120  -2.664 0.009140 **
## Distance7         -3.851      5.157  -0.747 0.457111
## Distance10        -5.771      5.102  -1.131 0.261001
## Headwidth          5.827      1.162   5.015 2.61e-06 ***
## Size.class35-39   -6.528      8.525  -0.766 0.445843
## Size.class40-43   -9.988     12.812  -0.780 0.437672
## Size.class>43     -3.765     16.679  -0.226 0.821932
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.94 on 91 degrees of freedom
## Multiple R-squared:  0.7741, Adjusted R-squared:  0.7542
## F-statistic: 38.98 on 8 and 91 DF,  p-value: < 2.2e-16
##
## Summary statistics for Colony 3
## Call:
## lm(formula = Mass ~ Distance + Headwidth + Size.class, data = col1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -20.6043  -7.2897  -0.6298   6.8340  25.0227
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)      -75.552     25.442  -2.970  0.00382 **
## Distance1        -12.045      3.916  -3.076  0.00278 **
## Distance4        -10.615      3.968  -2.675  0.00888 **
## Distance7        -11.582      4.007  -2.891  0.00482 **
## Distance10       -12.343      3.967  -3.111  0.00250 **
## Headwidth          4.227      0.833   5.074 2.08e-06 ***
## Size.class30-34   11.854      8.889   1.333  0.18574
## Size.class35-39    5.323     10.796   0.493  0.62318
## Size.class40-43   12.862     13.288   0.968  0.33568
## Size.class>43     22.146     16.182   1.369  0.17454
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.85 on 90 degrees of freedom
## Multiple R-squared:  0.8535, Adjusted R-squared:  0.8388
## F-statistic: 58.24 on 9 and 90 DF,  p-value: < 2.2e-16
##
## Summary statistics for Colony 4
## Call:
## lm(formula = Mass ~ Distance + Headwidth + Size.class, data = col1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -37.880  -7.397  -0.868   7.456  42.145
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     -133.3029    39.8570  -3.345  0.00114 **
## Distance1          2.3652     4.9231   0.480  0.63191
## Distance4          0.7435     5.1446   0.145  0.88536
## Distance7          2.7324     5.2073   0.525  0.60086
## Distance10        -3.2568     5.2593  -0.619  0.53707
## Headwidth          5.6572     1.2088   4.680 8.44e-06 ***
## Size.class35-39   -4.7835     8.6736  -0.551  0.58244
## Size.class40-43   -5.4945    12.2712  -0.448  0.65524
## Size.class>43      0.7429    16.2851   0.046  0.96370
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.91 on 107 degrees of freedom
## Multiple R-squared:  0.6888, Adjusted R-squared:  0.6655
## F-statistic:  29.6 on 8 and 107 DF,  p-value: < 2.2e-16
##
## Summary statistics for Colony 5
## Call:
## lm(formula = Mass ~ Distance + Headwidth + Size.class, data = col1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -48.242  -9.947  -0.554   9.443  48.749
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)     -85.1098    46.7467  -1.821  0.07231 .
```

```
## Distance1        -3.8703     5.8381  -0.663  0.50923
## Distance4        -9.3870     5.9634  -1.574  0.11931
## Distance7       -18.6780     6.0145  -3.105  0.00261 **
## Headwidth         4.5587     1.5603   2.922  0.00450 **
## Size.class30-34   0.7641    12.9949   0.059  0.95325
## Size.class35-39   3.1925    16.9004   0.189  0.85064
## Size.class40-43   8.7568    23.2710   0.376  0.70767
## Size.class>43    13.0378    28.4905   0.458  0.64844
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 16.43 on 82 degrees of freedom
## Multiple R-squared:  0.7152, Adjusted R-squared:  0.6874
## F-statistic: 25.74 on 8 and 82 DF,  p-value: < 2.2e-16
##
## Summary statistics for Colony 6
## Call:
## lm(formula = Mass ~ Distance + Headwidth + Size.class, data = col1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -33.940  -6.150  -1.083   6.006  38.753
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     -103.6695    26.1411  -3.966 0.000125 ***
## Distance1         -8.9946     4.4863  -2.005 0.047224 *
## Distance4        -11.4267     4.5155  -2.531 0.012683 *
## Distance7        -11.0054     4.6521  -2.366 0.019599 *
## Distance10       -11.3335     4.8185  -2.352 0.020298 *
## Headwidth          5.3876     0.9092   5.926 3.05e-08 ***
## Size.class30-34   -7.6205     8.0224  -0.950 0.344072
## Size.class35-39  -11.6536    10.9595  -1.063 0.289768
## Size.class40-43  -11.9476    14.2211  -0.840 0.402505
## Size.class>43    -10.9833    17.4441  -0.630 0.530133
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.21 on 120 degrees of freedom
## Multiple R-squared:  0.807,  Adjusted R-squared:  0.7925
## F-statistic: 55.76 on 9 and 120 DF,  p-value: < 2.2e-16
```

The coefficient of colony-level contribution is almost indifferent as the mean(coefficients of Colony 1 ~ 6) are similar(all around 160). The sign of Distance1 ~ 10 and Headwidth are negative and it means variable Mass and Distance and Headwidth are inversely proportional to each other. Since the definition of variable Mass is 'How much the ant weighed in milligrams' and it related to how much food (energy) the ant was carrying, it suggests that ants prefer **energy conservative** strategy **generally**.(Distant goes up —> Mass goes down)

However, if we see the 6 summary statistics above by Colony, it suggests different information. The Distance variable has positive coefficient in Colony 4 except 'Distance 10'. It potentially indicates that **Colony 4 tends to choose worker conservative strategy** except when woker ants are not seriously far away (Distance 10)

Therefore, I conclude that Colony 4 prefers worker conservative strategy and other colonies prefer energy conservative strategy.

#2. $t_i = \dfrac{\hat{e}_{[i]} \cdot \sqrt{1-h_i}}{\hat{\sigma}} = \dfrac{\frac{\hat{e}_i}{1-h_i} \cdot \sqrt{1-h_i}}{\sqrt{\frac{RSS_{[i]}}{n-p-2}}} = \dfrac{\frac{\hat{e}_i}{\sqrt{1-h_i}}}{\sqrt{\frac{RSS - \frac{\hat{e}_i^2}{1-h_i}}{n-p-2}}}$

※ $r_i = \dfrac{\hat{e}_i}{\hat{\sigma}\sqrt{1-h_i}} = \dfrac{\hat{e}_i}{\sqrt{\frac{RSS}{n-p-1}} \cdot \sqrt{1-h_i}} = \dfrac{r_i \cdot \hat{\sigma}}{\sqrt{\frac{RSS - \frac{\hat{e}_i^2}{1-h_i}}{n-p-2}}} = r_i \cdot \sqrt{\dfrac{\frac{RSS}{n-p-1}}{\frac{RSS - (r_i\hat{\sigma})^2}{n-p-2}}}$

$= r_i \cdot \sqrt{\dfrac{(n-p-2)(RSS)}{(n-p-1)(RSS-(r_i\hat{\sigma})^2)}} = r_i \sqrt{\dfrac{(n-p-2)RSS}{(n-p-1)RSS - (n-p-1)r_i^2\hat{\sigma}^2}}$

$= r_i \sqrt{\dfrac{(n-p-2)RSS}{(n-p-1)RSS - (n-p-1)r_i^2 \cdot \frac{RSS}{(n-p-1)}}} = r_i \sqrt{\dfrac{(n-p-2)RSS}{(n-p-1)RSS - RSS \cdot r_i^2}}$

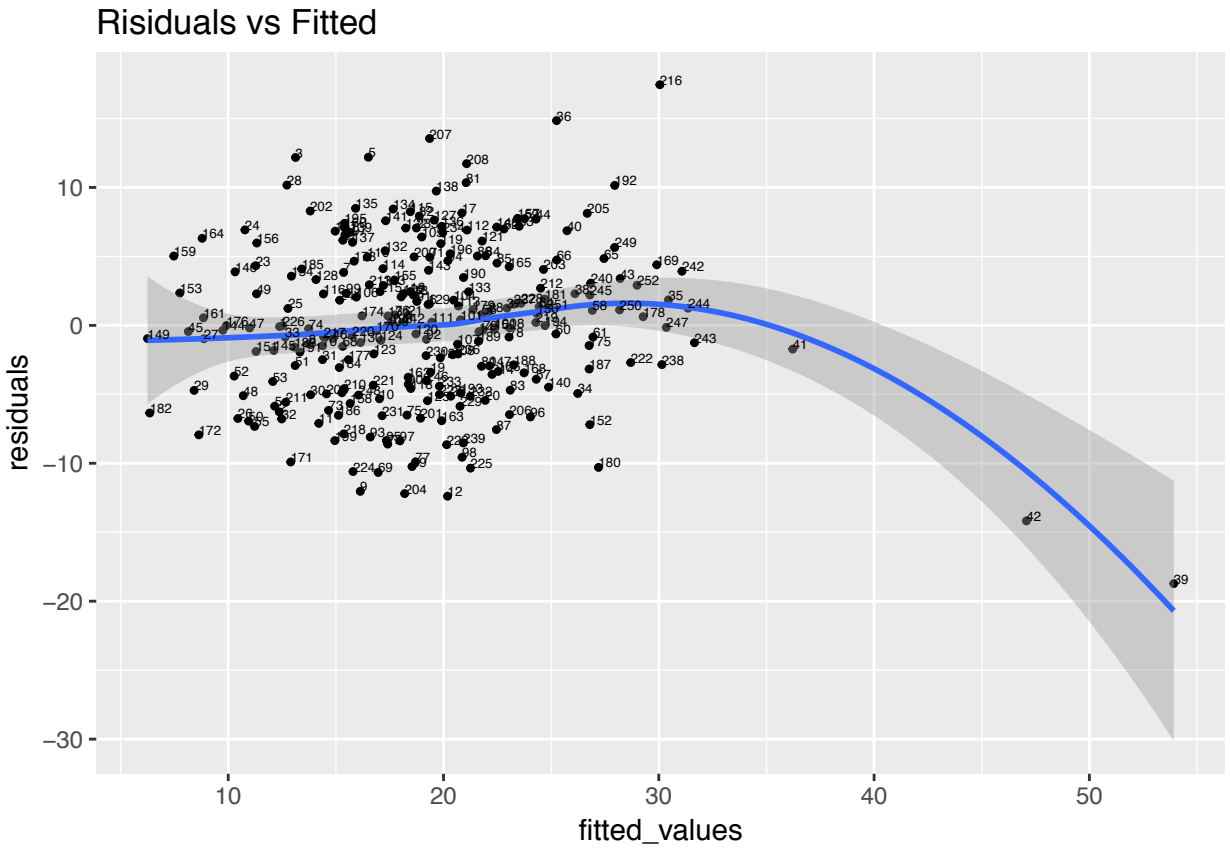
$= r_i \sqrt{\dfrac{(n-p-2)RSS}{(n-p-1-r_i^2)RSS}} = r_i \sqrt{\dfrac{(n-p-2)}{(n-p-1-r_i^2)}}$

3. Bodyfat

(a) Residuals against fitted values.

```
bodyfat <- read.csv("/Users/cloverjiyoon/2017Fall/Stat 151A/Lab/Lab3/bodyfat.csv")


# fitting linear model and getting diagnostics
fit = lm(bodyfat~ Age + Weight + Height + Thigh, data = bodyfat)



ggplot(data.frame(fitted_values = fit$fitted.values, residuals =fit$residuals), aes(x = fitted_values, y
```

```
## `geom_smooth()` using method = 'loess'
```



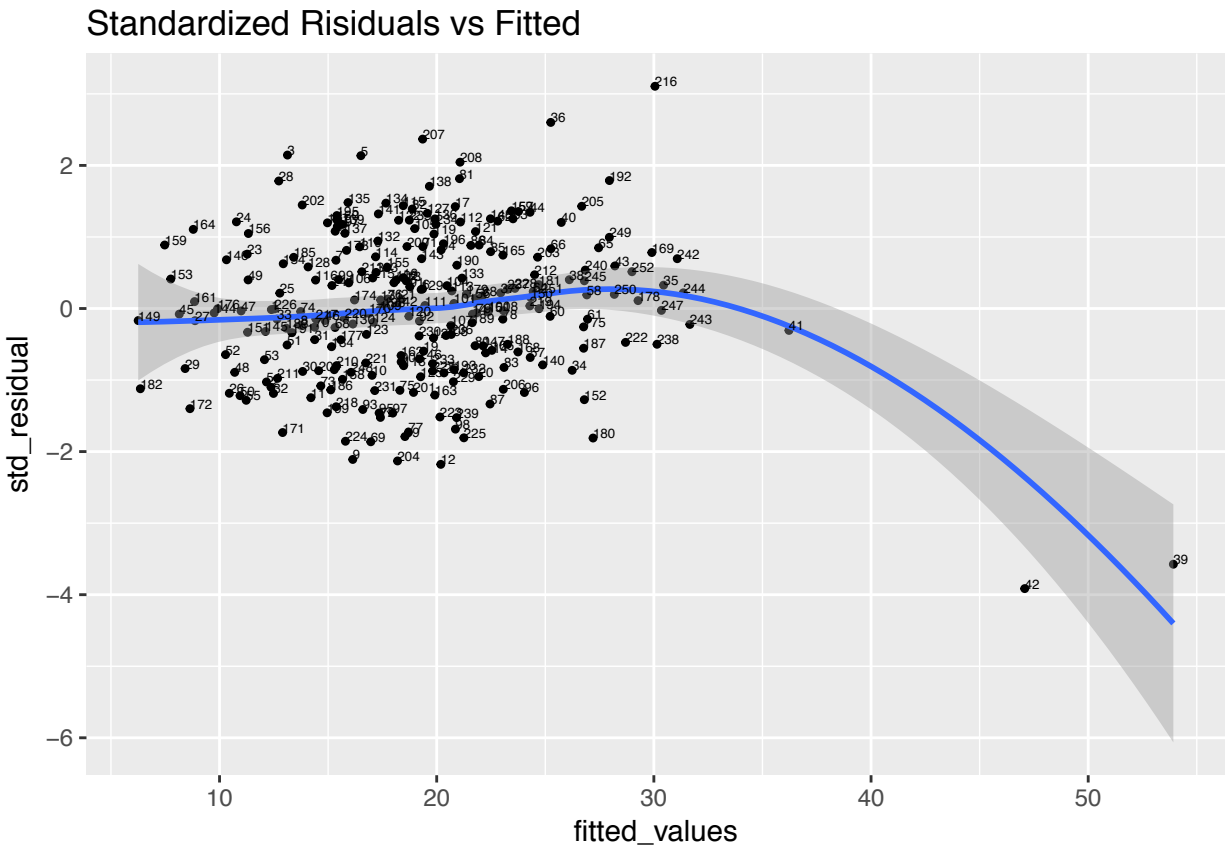
The residuals vs fitted values plot shows that the possibilites of being outliers for bottom left points on the graph. Since the loess line in the graph is almost a straight line until it reaches to the 42th elements in the graph, probably 42th and 39th elements can be an outliers.

(b) Standardized Residuals against fitted values.

```
n = nrow(bodyfat)
p = 4
```

```
X <- as.matrix(cbind(1,bodyfat[,c(3,4,5,10)]))
H <- X %*% solve(t(X) %*% X) %*% t(X)

RSS <- sum(fit$residuals^2)

std_residual <- fit$residuals / ( sqrt(RSS/ (n-p-1)) * sqrt(1-diag(H)))



ggplot(data.frame(fitted_values = fit$fitted.values, Std_residual =std_residual), aes(x = fitted_values
```

```
## `geom_smooth()` using method = 'loess'
```

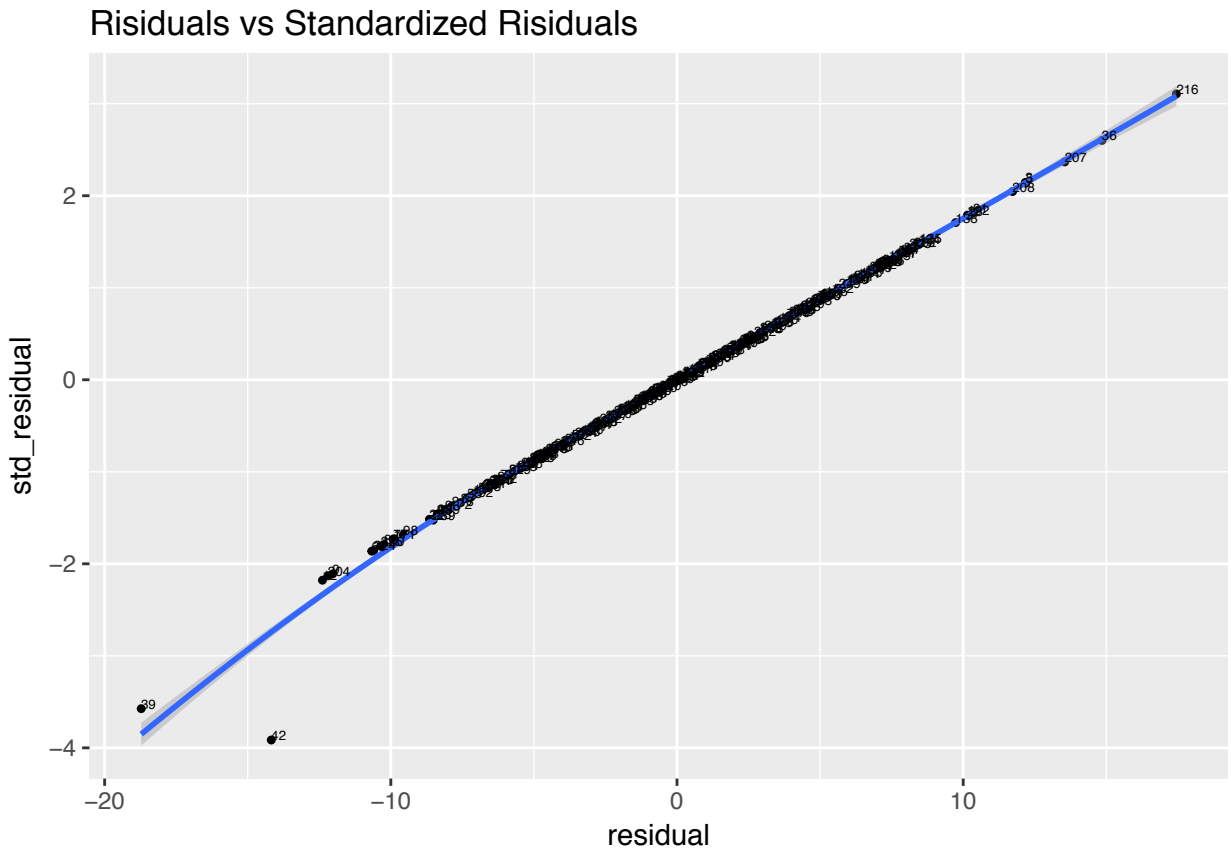


This graph looks similar as the residual vs fitted values plot as shown previously. Still 42th and 39th elements look like outliers

(c) Residuals against Standardized Residuals.

```
ggplot(data.frame(residual= fit$residuals, Std_residual =std_residual), aes(x = residual, y = std_residu
```

```
## `geom_smooth()` using method = 'loess'
```



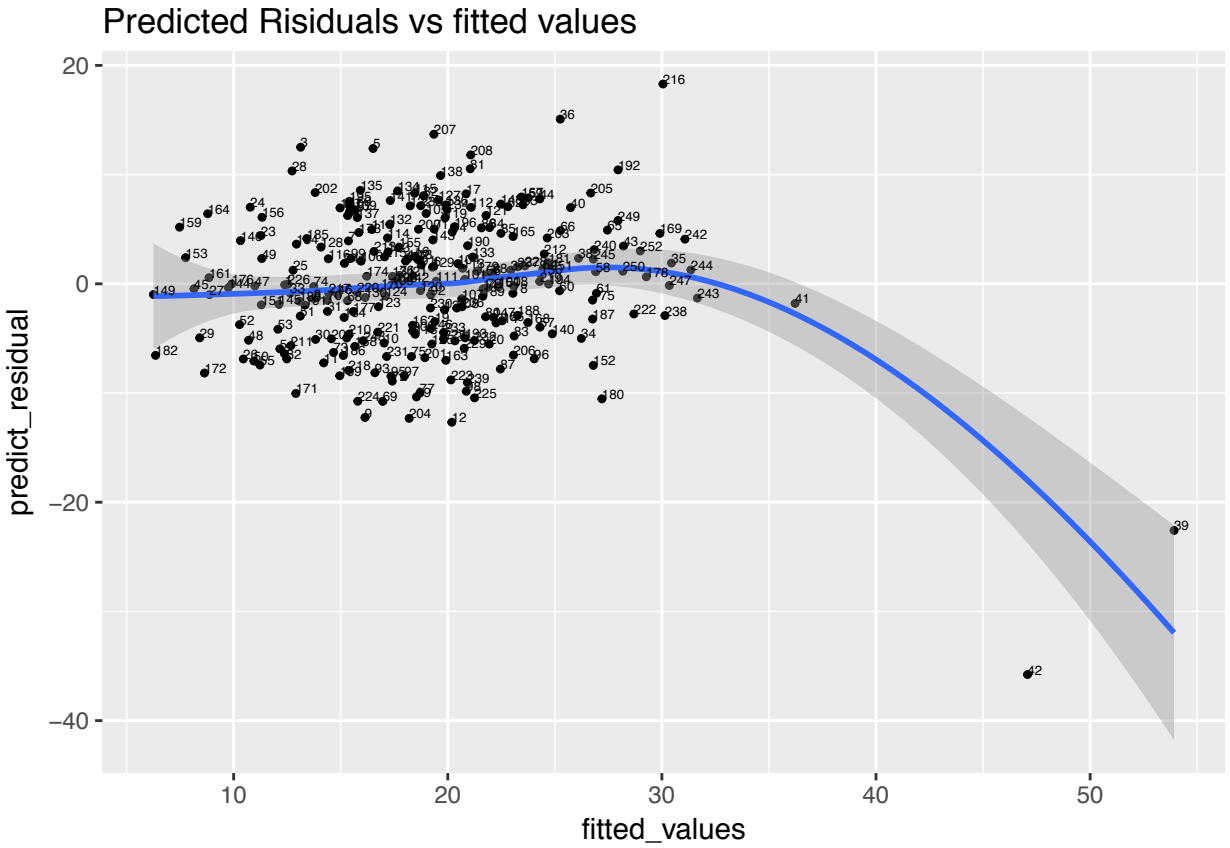
From Residuals against Standardized Residuals plot, we can also see that 39th and 42 elements are not on the loess line unlike the other points.

(d) Predicted residuals against fitted values.

$$e_{\hat{[i]}} = \frac{\hat{e_i}}{1 - h_i}$$

```
predict_residuals <- fit$residuals / (1- diag(H))



ggplot(data.frame(predict_residual= predict_residuals, fitted_values =fit$fitted.values), aes(x = fitte
```

```
## `geom_smooth()` using method = 'loess'
```

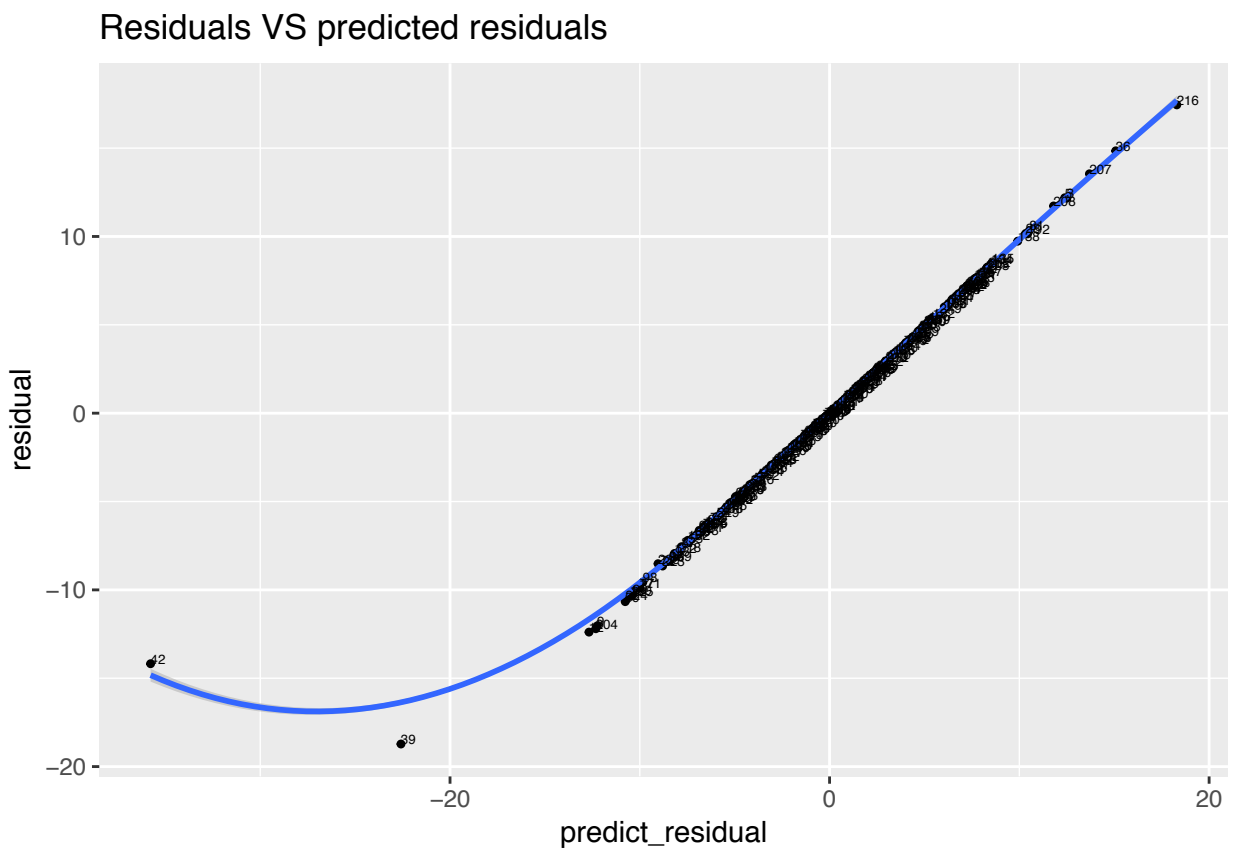


This graph looks similar as the residual vs fitted values plot as shown previously. Still 42th and 39th elements look like outliers.

(e) Residuals against predicted residuals.

```
ggplot(data.frame(predict_residual= predict_residuals, residual =fit$residuals), aes(x = predict_residua
```

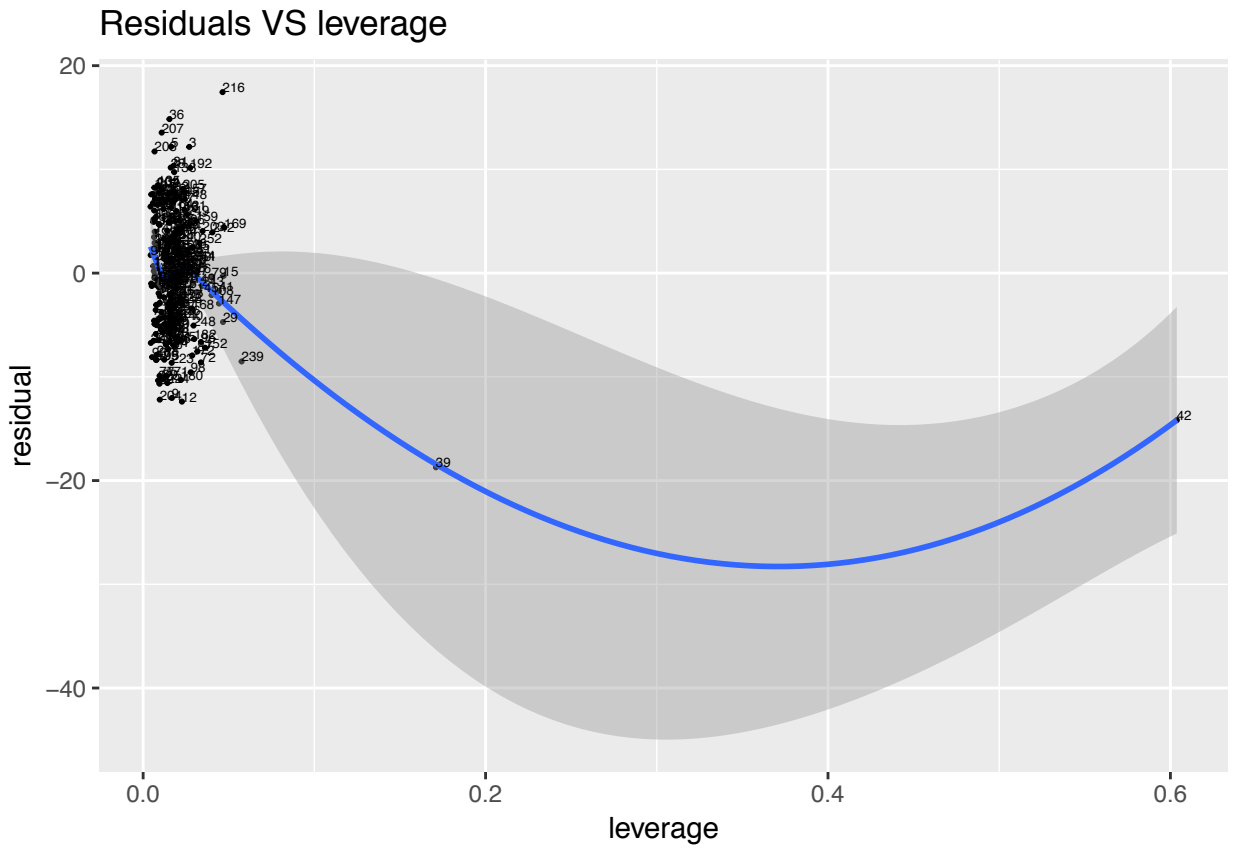
```
## `geom_smooth()` using method = 'loess'
```



(f) Residuals against leverage.

```
ggplot(data.frame(residual= fit$residuals, leverage = diag(H)), aes(y = residual, x = leverage)) + geom_
```

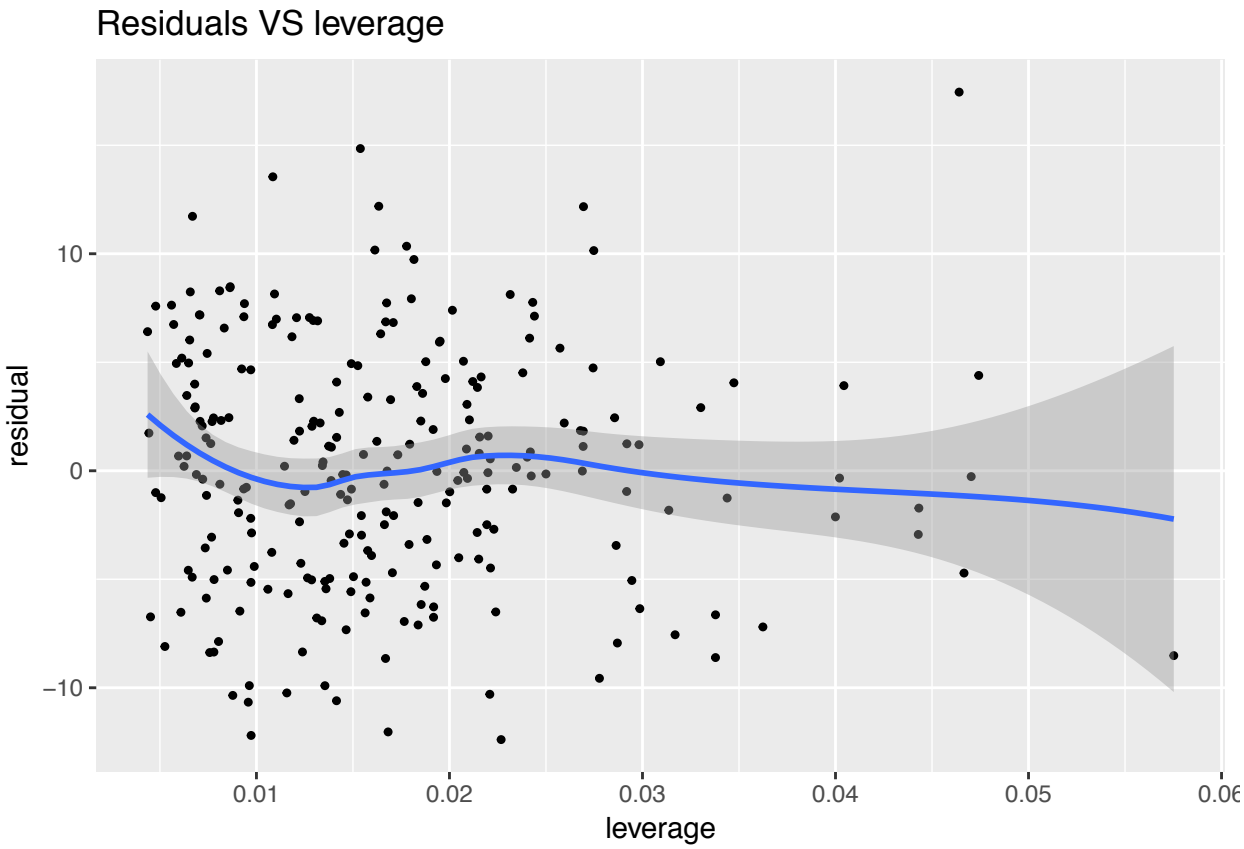
```
## `geom_smooth()` using method = 'loess'
```



Let's remove 42th and 39th elements.

```
ggplot(data.frame(residual= fit$residuals[-c(42,39)], leverage = diag(H)[-c(42,39)]), aes(y = residual,
```

```
## `geom_smooth()` using method = 'loess'
```



(g) Predicted residuals against Standardized Predicted Residuals.

```
# Q : why different?

RSS_i <- sum(fit$residuals^2) - (fit$residuals)^2/(1-diag(H))

std_predict_residuals <- (predict_residuals * sqrt(1 - diag(H))) /
  sqrt( RSS_i / (n - p - 2))



# Check

rstudent(fit)
```

```
##           1           2           3           4           5
## -0.015179424 -1.101321284  2.160521227 -0.854488155  2.151942093
##           6           7           8           9          10
##  0.272441517  0.673775061 -0.234008108 -2.124574955 -0.933932061
##          11          12          13          14          15
## -1.248106234 -2.195036056  0.384618125 -0.078642352 -0.047762816
##          16          17          18          19          20
##  0.429846505  1.426857268 -0.030141830 -0.595034940 -0.951741781
##          21          22          23          24          25
##  0.129084011 -0.899540353  0.759037906  1.212410739  0.214259764
##          26          27          28          29          30
```

```
## -1.185907826 -0.170105724  1.790214878 -0.838958860 -0.879244947
##           31           32           33           34           35
## -0.434560658 -1.188056644 -0.147917522 -0.863366401  0.325691053
##           36           37           38           39           40
##  2.631849424  0.218068131  0.401532975 -3.662837008  1.204064577
##           41           42           43           44           45
## -0.305459471 -4.033893314  0.594038102  1.347179270 -0.078249241
##           46           47           48           49           50
## -0.167309762 -0.033836321 -0.892054630  0.398797919 -1.219370005
##           51           52           53           54           55
## -0.509165820 -0.643557011 -0.714529997 -1.027162587 -1.284744513
##           56           57           58           59           60
##  0.140525406 -0.683929450  0.189714274  1.358099002 -0.109213551
##           61           62           63           64           65
## -0.148619657  1.222714786  1.254947683  0.237031521  0.847916885
##           66           67           68           69           70
##  0.834658388  1.080119553 -0.265226532 -1.872100579 -0.259354446
##           71           72           73           74           75
##  0.864163829 -1.526375723 -1.081793307 -0.041940423 -1.144446880
##           76           77           78           79           80
##  0.131301063 -1.736427808 -0.148371880 -0.059720795 -0.519257571
##           81           82           83           84           85
##  1.823364210  1.392852163 -0.823369023  0.885963653  0.793584608
##           86           87           88           89           90
##  0.881568267 -1.337036059  0.175534122 -1.797985031 -0.745217924
##           91           92           93           94           95
##  0.301772269 -0.174936481 -1.414229212  0.819083533 -1.463867959
##           96           97           98           99          100
## -1.174827423 -1.465030770 -1.692290255  0.403758131 -0.029877803
##          101          102          103          104          105
##  0.070867334  0.395017609  0.503592464  0.318792456  1.117131691
##          106          107          108          109          110
##  0.357971599 -0.236196803 -0.376580884 -0.013785546  0.861637681
##          111          112          113          114          115
##  0.041833723  1.209858498  0.245008280  0.721758936  1.440443727
##          116          117          118          119          120
##  0.396609253  0.359859743 -0.798548156  1.040287674 -0.108025367
##          121          122          123          124          125
##  1.075392661  1.234835049 -0.361273043 -0.190309301 -0.953883665
##          126          127          128          129          130
## -0.410708074  1.332249953  0.580141607  0.263678588 -0.215847140
##          131          132          133          134          135
##  0.117989816  0.943259586  0.424286063  1.476341371  1.484446753
##          136          137          138          139          140
##  1.253881981  1.051041787  1.714492134  1.175996480 -0.787171854
##          141          142          143          144          145
##  1.323827281  0.035198127  0.695715764 -0.061871599 -0.320412641
##          146          147          148          149          150
##  0.680105193 -0.520982403  1.255394610 -0.168179249  0.152303944
##          151          152          153          154          155
## -0.331175233 -1.275878340  0.411355251  0.624841371  0.573663406
##          156          157          158          159          160
##  1.048406394  1.367775686 -0.990039837  0.886921517  1.240925548
##          161          162          163          164          165
```

```
##  0.096318706 -0.656407823 -1.211023344  1.106470110  0.745235315
##          166          167          168          169          170
## -0.583988195  1.198460062 -0.606288699  0.781697156 -0.067523896
##          171          172          173          174          175
## -1.740474409 -1.402753436  0.811959051  0.119084367 -0.256350261
##          176          177          178          179          180
## -0.002441090 -0.436245322  0.110090325  0.198083049 -1.818873201
##          181          182          183          184          185
##  0.332987080 -1.122032126 -0.274607341 -0.533069324  0.714540572
##          186          187          188          189          190
## -1.137772554 -0.554626517 -0.499446140 -0.197621156  0.604175838
##          191          192          193          194          195
## -0.336957340  1.796445402 -0.854956570 -0.002196885  1.300790176
##          196          197          198          199          200
##  0.904704501  1.148559989  0.026701571 -1.461120553  0.865623044
##          201          202          203          204          205
## -1.173789103  1.450269408  0.716378523 -2.145974191  1.431554936
##          206          207          208          209          210
## -1.130126392  2.389968655  2.058235353 -0.868739481 -0.799007566
##          211          212          213          214          215
## -0.975890571  0.470324713  0.512893363 -0.619553040  0.426453101
##          216          217          218          219          220
##  3.161915202 -0.146039530 -1.375696824  0.036363458 -0.132949116
##          221          222          223          224          225
## -0.760479342 -0.473118754 -1.520345218 -1.864647449 -1.815677537
##          226          227          228          229          230
## -0.005491539 -0.874516141  0.280660697 -1.024001547 -0.382235514
##          231          232          233          234          235
## -1.147954155 -0.898116866 -0.769812512  1.177854849  1.236271927
##          236          237          238          239          240
## -0.361541886  0.268771807 -0.499038004 -1.529313292  0.536544377
##          241          242          243          244          245
##  0.320746070  0.695397851 -0.222032205  0.218758989  0.386741513
##          246          247          248          249          250
## -0.703250782 -0.026119304 -0.891799261  0.994864284  0.197135156
##          251          252
##  0.211601572  0.513355597
```

```
std_predict_residuals
```

```
##           1           2           3           4           5
## -0.015179424 -1.101321284  2.160521227 -0.854488155  2.151942093
##           6           7           8           9          10
##  0.272441517  0.673775061 -0.234008108 -2.124574955 -0.933932061
##          11          12          13          14          15
## -1.248106234 -2.195036056  0.384618125 -0.078642352 -0.047762816
##          16          17          18          19          20
##  0.429846505  1.426857268 -0.030141830 -0.595034940 -0.951741781
##          21          22          23          24          25
##  0.129084011 -0.899540353  0.759037906  1.212410739  0.214259764
##          26          27          28          29          30
## -1.185907826 -0.170105724  1.790214878 -0.838958860 -0.879244947
##          31          32          33          34          35
## -0.434560658 -1.188056644 -0.147917522 -0.863366401  0.325691053
##          36          37          38          39          40
```

```
##  2.631849424  0.218068131  0.401532975 -3.662837008  1.204064577
##           41           42           43           44           45
## -0.305459471 -4.033893314  0.594038102  1.347179270 -0.078249241
##           46           47           48           49           50
## -0.167309762 -0.033836321 -0.892054630  0.398797919 -1.219370005
##           51           52           53           54           55
## -0.509165820 -0.643557011 -0.714529997 -1.027162587 -1.284744513
##           56           57           58           59           60
##  0.140525406 -0.683929450  0.189714274  1.358099002 -0.109213551
##           61           62           63           64           65
## -0.148619657  1.222714786  1.254947683  0.237031521  0.847916885
##           66           67           68           69           70
##  0.834658388  1.080119553 -0.265226532 -1.872100579 -0.259354446
##           71           72           73           74           75
##  0.864163829 -1.526375723 -1.081793307 -0.041940423 -1.144446880
##           76           77           78           79           80
##  0.131301063 -1.736427808 -0.148371880 -0.059720795 -0.519257571
##           81           82           83           84           85
##  1.823364210  1.392852163 -0.823369023  0.885963653  0.793584608
##           86           87           88           89           90
##  0.881568267 -1.337036059  0.175534122 -1.797985031 -0.745217924
##           91           92           93           94           95
##  0.301772269 -0.174936481 -1.414229212  0.819083533 -1.463867959
##           96           97           98           99          100
## -1.174827423 -1.465030770 -1.692290255  0.403758131 -0.029877803
##          101          102          103          104          105
##  0.070867334  0.395017609  0.503592464  0.318792456  1.117131691
##          106          107          108          109          110
##  0.357971599 -0.236196803 -0.376580884 -0.013785546  0.861637681
##          111          112          113          114          115
##  0.041833723  1.209858498  0.245008280  0.721758936  1.440443727
##          116          117          118          119          120
##  0.396609253  0.359859743 -0.798548156  1.040287674 -0.108025367
##          121          122          123          124          125
##  1.075392661  1.234835049 -0.361273043 -0.190309301 -0.953883665
##          126          127          128          129          130
## -0.410708074  1.332249953  0.580141607  0.263678588 -0.215847140
##          131          132          133          134          135
##  0.117989816  0.943259586  0.424286063  1.476341371  1.484446753
##          136          137          138          139          140
##  1.253881981  1.051041787  1.714492134  1.175996480 -0.787171854
##          141          142          143          144          145
##  1.323827281  0.035198127  0.695715764 -0.061871599 -0.320412641
##          146          147          148          149          150
##  0.680105193 -0.520982403  1.255394610 -0.168179249  0.152303944
##          151          152          153          154          155
## -0.331175233 -1.275878340  0.411355251  0.624841371  0.573663406
##          156          157          158          159          160
##  1.048406394  1.367775686 -0.990039837  0.886921517  1.240925548
##          161          162          163          164          165
##  0.096318706 -0.656407823 -1.211023344  1.106470110  0.745235315
##          166          167          168          169          170
## -0.583988195  1.198460062 -0.606288699  0.781697156 -0.067523896
##          171          172          173          174          175
```

```
## -1.740474409 -1.402753436  0.811959051  0.119084367 -0.256350261
##          176          177          178          179          180
## -0.002441090 -0.436245322  0.110090325  0.198083049 -1.818873201
##          181          182          183          184          185
##  0.332987080 -1.122032126 -0.274607341 -0.533069324  0.714540572
##          186          187          188          189          190
## -1.137772554 -0.554626517 -0.499446140 -0.197621156  0.604175838
##          191          192          193          194          195
## -0.336957340  1.796445402 -0.854956570 -0.002196885  1.300790176
##          196          197          198          199          200
##  0.904704501  1.148559989  0.026701571 -1.461120553  0.865623044
##          201          202          203          204          205
## -1.173789103  1.450269408  0.716378523 -2.145974191  1.431554936
##          206          207          208          209          210
## -1.130126392  2.389968655  2.058235353 -0.868739481 -0.799007566
##          211          212          213          214          215
## -0.975890571  0.470324713  0.512893363 -0.619553040  0.426453101
##          216          217          218          219          220
##  3.161915202 -0.146039530 -1.375696824  0.036363458 -0.132949116
##          221          222          223          224          225
## -0.760479342 -0.473118754 -1.520345218 -1.864647449 -1.815677537
##          226          227          228          229          230
## -0.005491539 -0.874516141  0.280660697 -1.024001547 -0.382235514
##          231          232          233          234          235
## -1.147954155 -0.898116866 -0.769812512  1.177854849  1.236271927
##          236          237          238          239          240
## -0.361541886  0.268771807 -0.499038004 -1.529313292  0.536544377
##          241          242          243          244          245
##  0.320746070  0.695397851 -0.222032205  0.218758989  0.386741513
##          246          247          248          249          250
## -0.703250782 -0.026119304 -0.891799261  0.994864284  0.197135156
##          251          252
##  0.211601572  0.513355597
```

```
rstandard(fit)
```

```
##           1           2           3           4           5
## -0.015210238 -1.100846934  2.144656260 -0.854955306  2.136297782
##           6           7           8           9          10
##  0.272953523  0.674521034 -0.234457158 -2.109622217 -0.934173712
##          11          12          13          14          15
## -1.246699395 -2.178264563  0.385283250 -0.078801042 -0.047859574
##          16          17          18          19          20
##  0.430557627  1.423874516 -0.030202976 -0.595814512 -0.951923295
##          21          22          23          24          25
##  0.129341731 -0.899888037  0.759690014  1.211259022  0.214674780
##          26          27          28          29          30
## -1.184933470 -0.170441093  1.782277714 -0.839462260 -0.879649123
##          31          32          33          34          35
## -0.435275977 -1.187068284 -0.148211271 -0.863811709  0.326282016
##          36          37          38          39          40
##  2.600831611  0.218489793  0.402216486 -3.574105576  1.202969808
##          41          42          43          44          45
## -0.306021664 -3.914683688  0.594817799  1.344962476 -0.078407147
##          46          47          48          49          50
```

```
## -0.167639940 -0.033904945 -0.892423668  0.399478550 -1.218170025
##           51           52           53           54           55
## -0.509931033 -0.644321566 -0.715238995 -1.027048115 -1.283055916
##           56           57           58           59           60
##  0.140805085 -0.684667515  0.190085577  1.355783433 -0.109432652
##           61           62           63           64           65
## -0.148914737  1.221491357  1.253489779  0.237485686  0.848399677
##           66           67           68           69           70
##  0.835171391  1.079755343 -0.265727073 -1.862679934 -0.259845532
##           71           72           73           74           75
##  0.864607134 -1.522283315 -1.081420616 -0.042025431 -1.143729939
##           76           77           78           79           80
##  0.131563055 -1.729387423 -0.148666491 -0.059841621 -0.520026993
##           81           82           83           84           85
##  1.814843948  1.390209233 -0.823906344  0.886349619  0.794180021
##           86           87           88           89           90
##  0.881966201 -1.334909288  0.175879522 -1.789913293 -0.745889602
##           91           92           93           94           95
##  0.302329050 -0.175280780 -1.411374937  0.819629751 -1.460492889
##           96           97           98           99          100
## -1.173924230 -1.461642988 -1.685941207  0.404443958 -0.029938415
##          101          102          103          104          105
##  0.071010503  0.395694201  0.504355081  0.319373787  1.116571324
##          106          107          108          109          110
##  0.358605057 -0.236649558 -0.377236797 -0.013813532  0.862087306
##          111          112          113          114          115
##  0.041918515  1.208724303  0.245475810  0.722459892  1.437319706
##          116          117          118          119          120
##  0.397287554  0.360495550 -0.799134491  1.040114620 -0.108242141
##          121          122          123          124          125
##  1.075052203  1.233525267 -0.361910598 -0.190681680 -0.954057702
##          126          127          128          129          130
## -0.411400976  1.330165085  0.580922302  0.264176647 -0.216264930
##          131          132          133          134          135
##  0.118226044  0.943470193  0.424992085  1.472828707  1.480843210
##          136          137          138          139          140
##  1.252432078  1.050819120  1.707800285  1.175085862 -0.787778646
##          141          142          143          144          145
##  1.321815269  0.035269506  0.696443575 -0.061996745 -0.320996252
##          146          147          148          149          150
##  0.680846337 -0.521752479  1.253933345 -0.168511043  0.152605996
##          151          152          153          154          155
## -0.331773720 -1.274259805  0.412048800  0.625613827  0.574444100
##          156          157          158          159          160
##  1.048196020  1.365370964 -0.990079564  0.887304847  1.239571551
##          161          162          163          164          165
##  0.096512458 -0.657165368 -1.209881164  1.105968114  0.745906970
##          166          167          168          169          170
## -0.584768752  1.197402968 -0.607066357  0.782313351 -0.067660373
##          171          172          173          174          175
## -1.733368649 -1.400013573  0.812519657  0.119322723 -0.256836467
##          176          177          178          179          180
## -0.002446046 -0.436962112  0.110311142  0.198469421 -1.810433314
##          181          182          183          184          185
```

```
##  0.333588026 -1.121444416 -0.275122755 -0.533843458  0.715249559
##          186          187          188          189          190
## -1.137094811 -0.555405519 -0.500206702 -0.198006701  0.604953927
##          191          192          193          194          195
## -0.337563627  1.788400432 -0.855422589 -0.002201346  1.298971691
##          196          197          198          199          200
##  0.905037099  1.147818584  0.026755748 -1.457775422  0.866062668
##          201          202          203          204          205
## -1.172892490  1.447041240  0.717085508 -2.130482286  1.428523697
##          206          207          208          209          210
## -1.129492806  2.367493332  2.044882352 -0.869171168 -0.799593048
##          211          212          213          214          215
## -0.975984692  0.471067941  0.513660205 -0.620327242  0.427161130
##          216          217          218          219          220
##  3.105851285 -0.146329714 -1.373217979  0.036437194 -0.133214278
##          221          222          223          224          225
## -0.761129308 -0.473863862 -1.516325073 -1.855367863 -1.807294577
##          226          227          228          229          230
## -0.005502689 -0.874932846  0.281185552 -1.023900863 -0.382897939
##          231          232          233          234          235
## -1.147216368 -0.898468659 -0.770448142  1.176932386  1.234951763
##          236          237          238          239          240
## -0.362179772  0.269278003 -0.499798358 -1.525185443  0.537319503
##          241          242          243          244          245
##  0.321330148  0.696125954 -0.222460743  0.219181852  0.387409021
##          246          247          248          249          250
## -0.703971422 -0.026172302 -0.892169018  0.994884917  0.197519830
##          251          252
##  0.212011927  0.514122634
```

```
std_residual
```

```
##           1           2           3           4           5
## -0.015210238 -1.100846934  2.144656260 -0.854955306  2.136297782
##           6           7           8           9          10
##  0.272953523  0.674521034 -0.234457158 -2.109622217 -0.934173712
##          11          12          13          14          15
## -1.246699395 -2.178264563  0.385283250 -0.078801042 -0.047859574
##          16          17          18          19          20
##  0.430557627  1.423874516 -0.030202976 -0.595814512 -0.951923295
##          21          22          23          24          25
##  0.129341731 -0.899888037  0.759690014  1.211259022  0.214674780
##          26          27          28          29          30
## -1.184933470 -0.170441093  1.782277714 -0.839462260 -0.879649123
##          31          32          33          34          35
## -0.435275977 -1.187068284 -0.148211271 -0.863811709  0.326282016
##          36          37          38          39          40
##  2.600831611  0.218489793  0.402216486 -3.574105576  1.202969808
##          41          42          43          44          45
## -0.306021664 -3.914683688  0.594817799  1.344962476 -0.078407147
##          46          47          48          49          50
## -0.167639940 -0.033904945 -0.892423668  0.399478550 -1.218170025
##          51          52          53          54          55
## -0.509931033 -0.644321566 -0.715238995 -1.027048115 -1.283055916
##          56          57          58          59          60
```

```
##  0.140805085 -0.684667515  0.190085577  1.355783433 -0.109432652
##           61           62           63           64           65
## -0.148914737  1.221491357  1.253489779  0.237485686  0.848399677
##           66           67           68           69           70
##  0.835171391  1.079755343 -0.265727073 -1.862679934 -0.259845532
##           71           72           73           74           75
##  0.864607134 -1.522283315 -1.081420616 -0.042025431 -1.143729939
##           76           77           78           79           80
##  0.131563055 -1.729387423 -0.148666491 -0.059841621 -0.520026993
##           81           82           83           84           85
##  1.814843948  1.390209233 -0.823906344  0.886349619  0.794180021
##           86           87           88           89           90
##  0.881966201 -1.334909288  0.175879522 -1.789913293 -0.745889602
##           91           92           93           94           95
##  0.302329050 -0.175280780 -1.411374937  0.819629751 -1.460492889
##           96           97           98           99          100
## -1.173924230 -1.461642988 -1.685941207  0.404443958 -0.029938415
##          101          102          103          104          105
##  0.071010503  0.395694201  0.504355081  0.319373787  1.116571324
##          106          107          108          109          110
##  0.358605057 -0.236649558 -0.377236797 -0.013813532  0.862087306
##          111          112          113          114          115
##  0.041918515  1.208724303  0.245475810  0.722459892  1.437319706
##          116          117          118          119          120
##  0.397287554  0.360495550 -0.799134491  1.040114620 -0.108242141
##          121          122          123          124          125
##  1.075052203  1.233525267 -0.361910598 -0.190681680 -0.954057702
##          126          127          128          129          130
## -0.411400976  1.330165085  0.580922302  0.264176647 -0.216264930
##          131          132          133          134          135
##  0.118226044  0.943470193  0.424992085  1.472828707  1.480843210
##          136          137          138          139          140
##  1.252432078  1.050819120  1.707800285  1.175085862 -0.787778646
##          141          142          143          144          145
##  1.321815269  0.035269506  0.696443575 -0.061996745 -0.320996252
##          146          147          148          149          150
##  0.680846337 -0.521752479  1.253933345 -0.168511043  0.152605996
##          151          152          153          154          155
## -0.331773720 -1.274259805  0.412048800  0.625613827  0.574444100
##          156          157          158          159          160
##  1.048196020  1.365370964 -0.990079564  0.887304847  1.239571551
##          161          162          163          164          165
##  0.096512458 -0.657165368 -1.209881164  1.105968114  0.745906970
##          166          167          168          169          170
## -0.584768752  1.197402968 -0.607066357  0.782313351 -0.067660373
##          171          172          173          174          175
## -1.733368649 -1.400013573  0.812519657  0.119322723 -0.256836467
##          176          177          178          179          180
## -0.002446046 -0.436962112  0.110311142  0.198469421 -1.810433314
##          181          182          183          184          185
##  0.333588026 -1.121444416 -0.275122755 -0.533843458  0.715249559
##          186          187          188          189          190
## -1.137094811 -0.555405519 -0.500206702 -0.198006701  0.604953927
##          191          192          193          194          195
```

```
## -0.337563627  1.788400432 -0.855422589 -0.002201346  1.298971691
##          196          197          198          199          200
##  0.905037099  1.147818584  0.026755748 -1.457775422  0.866062668
##          201          202          203          204          205
## -1.172892490  1.447041240  0.717085508 -2.130482286  1.428523697
##          206          207          208          209          210
## -1.129492806  2.367493332  2.044882352 -0.869171168 -0.799593048
##          211          212          213          214          215
## -0.975984692  0.471067941  0.513660205 -0.620327242  0.427161130
##          216          217          218          219          220
##  3.105851285 -0.146329714 -1.373217979  0.036437194 -0.133214278
##          221          222          223          224          225
## -0.761129308 -0.473863862 -1.516325073 -1.855367863 -1.807294577
##          226          227          228          229          230
## -0.005502689 -0.874932846  0.281185552 -1.023900863 -0.382897939
##          231          232          233          234          235
## -1.147216368 -0.898468659 -0.770448142  1.176932386  1.234951763
##          236          237          238          239          240
## -0.362179772  0.269278003 -0.499798358 -1.525185443  0.537319503
##          241          242          243          244          245
##  0.321330148  0.696125954 -0.222460743  0.219181852  0.387409021
##          246          247          248          249          250
## -0.703971422 -0.026172302 -0.892169018  0.994884917  0.197519830
##          251          252
##  0.212011927  0.514122634
```

```
head(fit$residuals / (1- diag(H)))
```

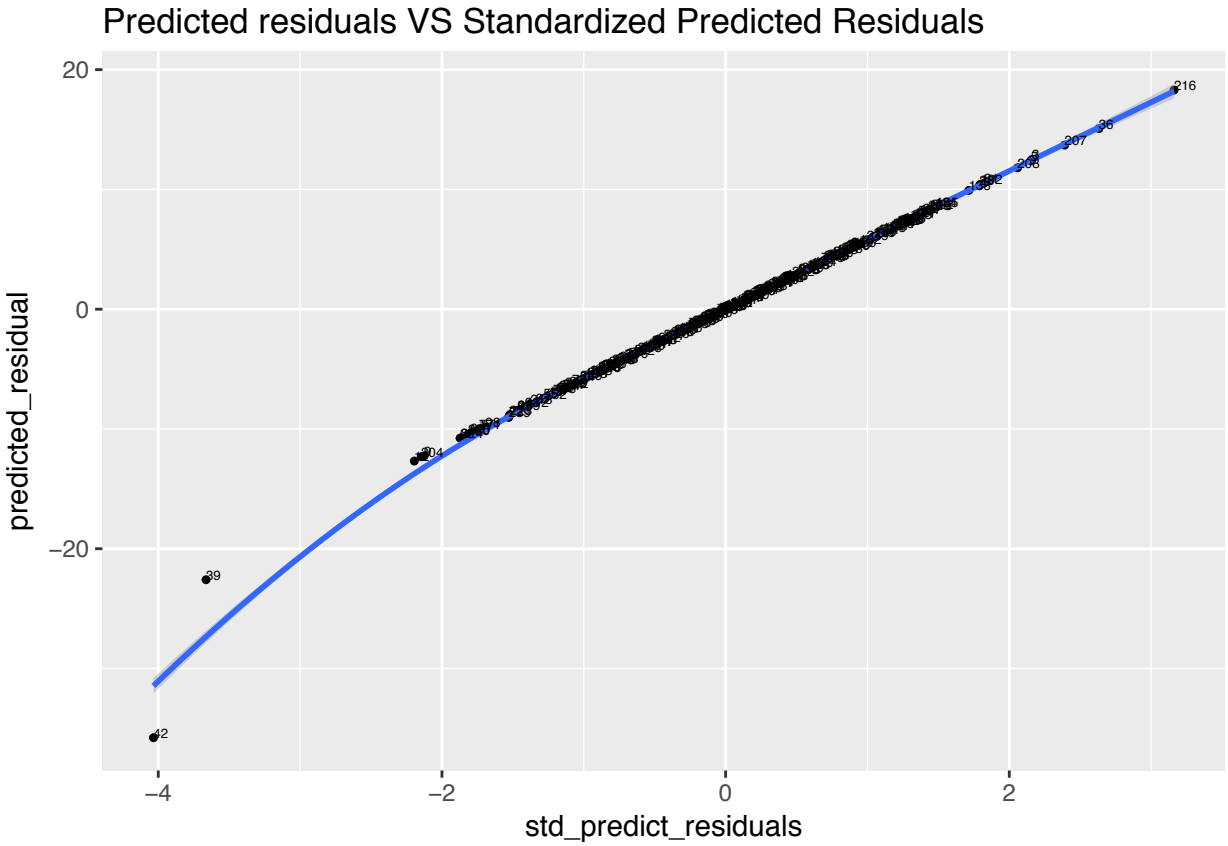
```
##           1           2           3           4           5           6
## -0.08848557 -6.39499076 12.50823447 -4.95608404 12.39213584  1.58756070
```

```
head(predict_residuals)
```

```
##           1           2           3           4           5           6
## -0.08848557 -6.39499076 12.50823447 -4.95608404 12.39213584  1.58756070
```

```
ggplot(data.frame(predicted_residual= predict_residuals, std_predict_residuals = std_predict_residuals)
```

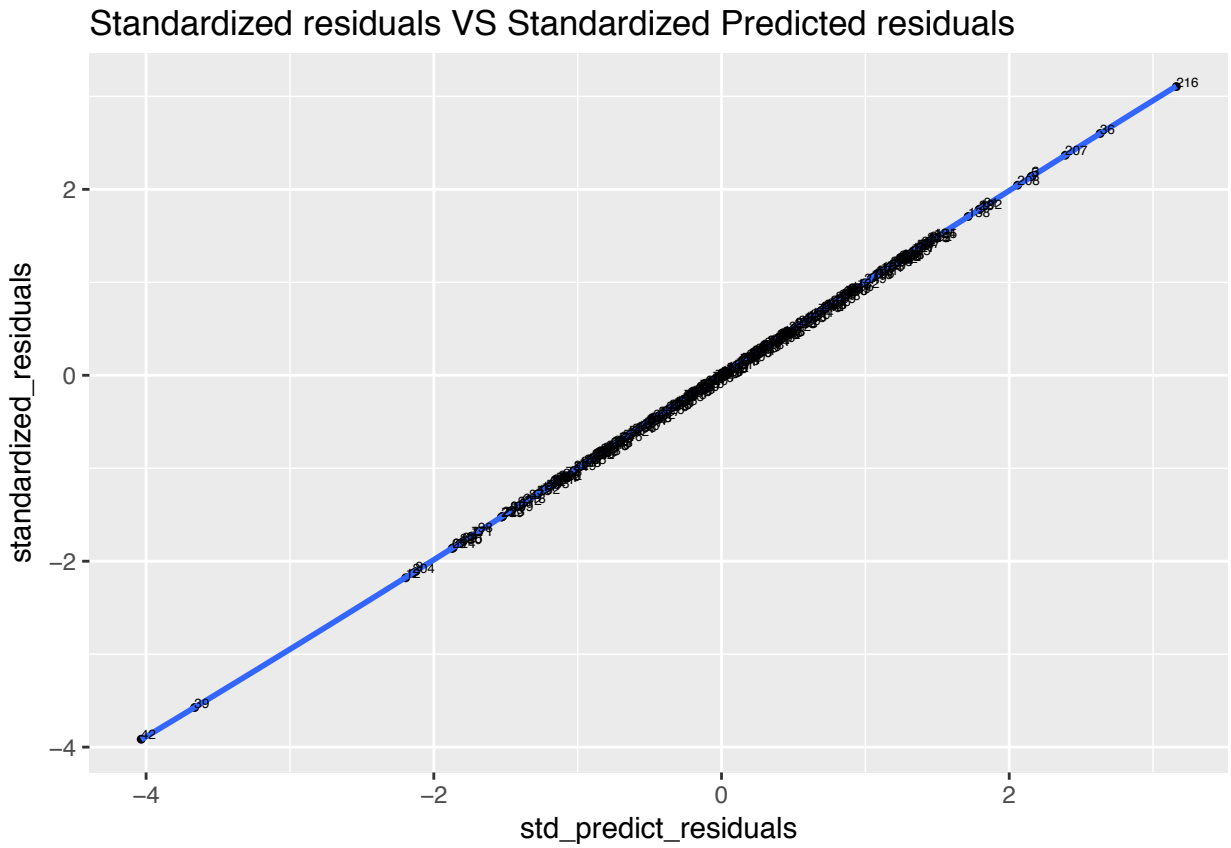
```
## `geom_smooth()` using method = 'loess'
```



(h) Standardized residuals against Standardized Predicted residuals.

```
ggplot(data.frame(standardized_residuals= std_residual, std_predict_residuals = std_predict_residuals),
```

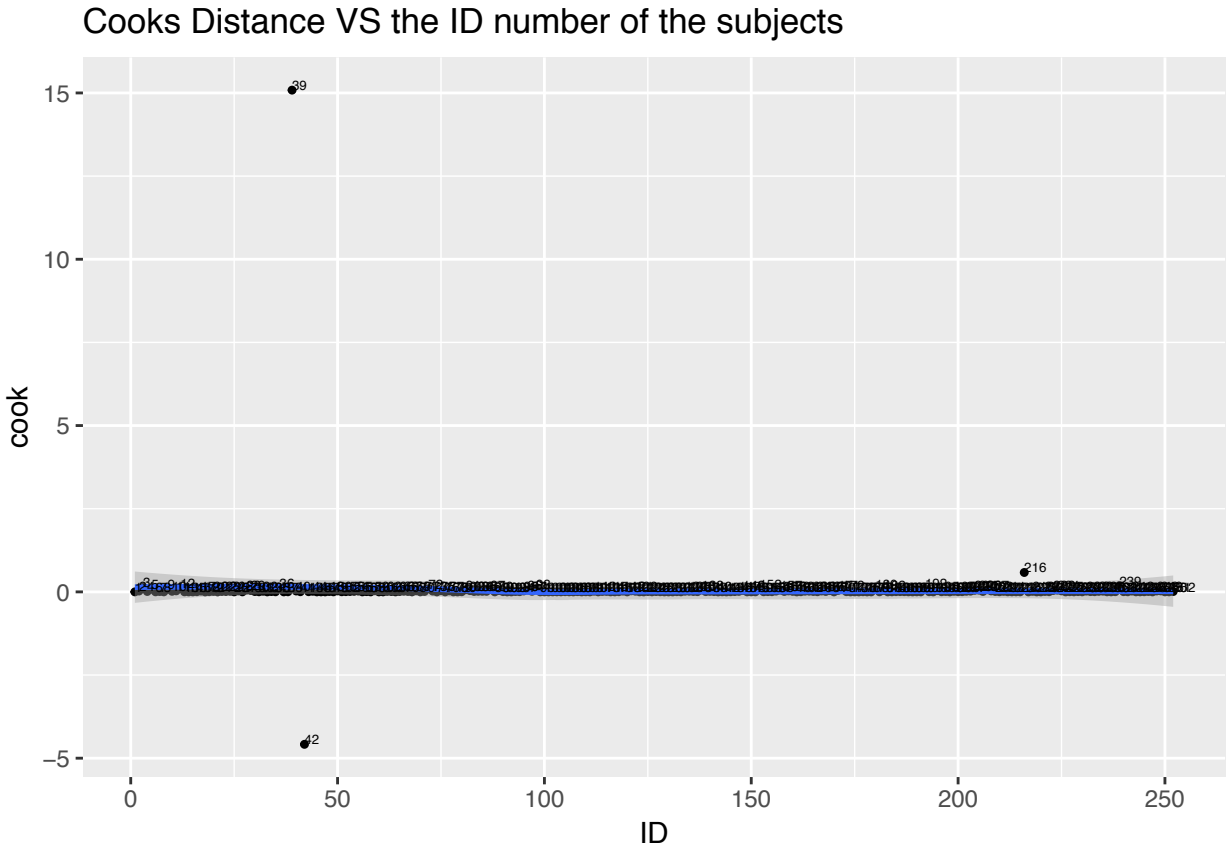
```
## `geom_smooth()` using method = 'loess'
```



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

```
cook <- std_residual^2 * diag(H) / ((1-diag(H) * (p+1)))

ggplot(data.frame(cook= cook, ID = 1:length(cook)), aes(y = cook, x = ID)) + geom_point(cex = 0.9) + lab
```

```
## `geom_smooth()` using method = 'loess'
```



Obviously we can see that 39th and 42 elements are far away from the clouds and 216th elements are a bit off from clouds also.

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

As shown above, these plots suggests some potential outliers and infuential points(42th, 39th elements).

First plot : The plot shows that 39th, 41th, 42th, and 216th elements are away from clouds and this observations potentially lead to poor fit.

Second plot: The plot also gives us the similar intuition as the first plot and indicates potential outlier which is 36th element.

Third plot: 42th and 39th elements are not on a loess line and it suggests that the difference between their residuals and standardized residuals is huge. —> potential unusal leverage

Fourth plot & Fifth plot : The plot also indicates that 39th and 42th elements have unusual leverage.

Sixth plot: The plot shows that 216th, 239th, 39th, and 42th elements have high leverage. Since $\sum_i h_i = p$ and the average leverage is p/n = 0.01587302, the high leverage points can be the one which have 2p/n leverage. The list below is the elements with high leverage.

```
match(diag(H)[diag(H) > 2*p/n], diag(H))
```

```
##  [1]  15  29  39  41  42  72  79  96 108 147 152 169 203 216 239 242 243
## [18] 252
```

As we assumed previously, 39th, 41th, 42th, and 216th elements have high leverage. The other elements listed above are not necessarily exact outliers since leverage doesn't take reponses into account.

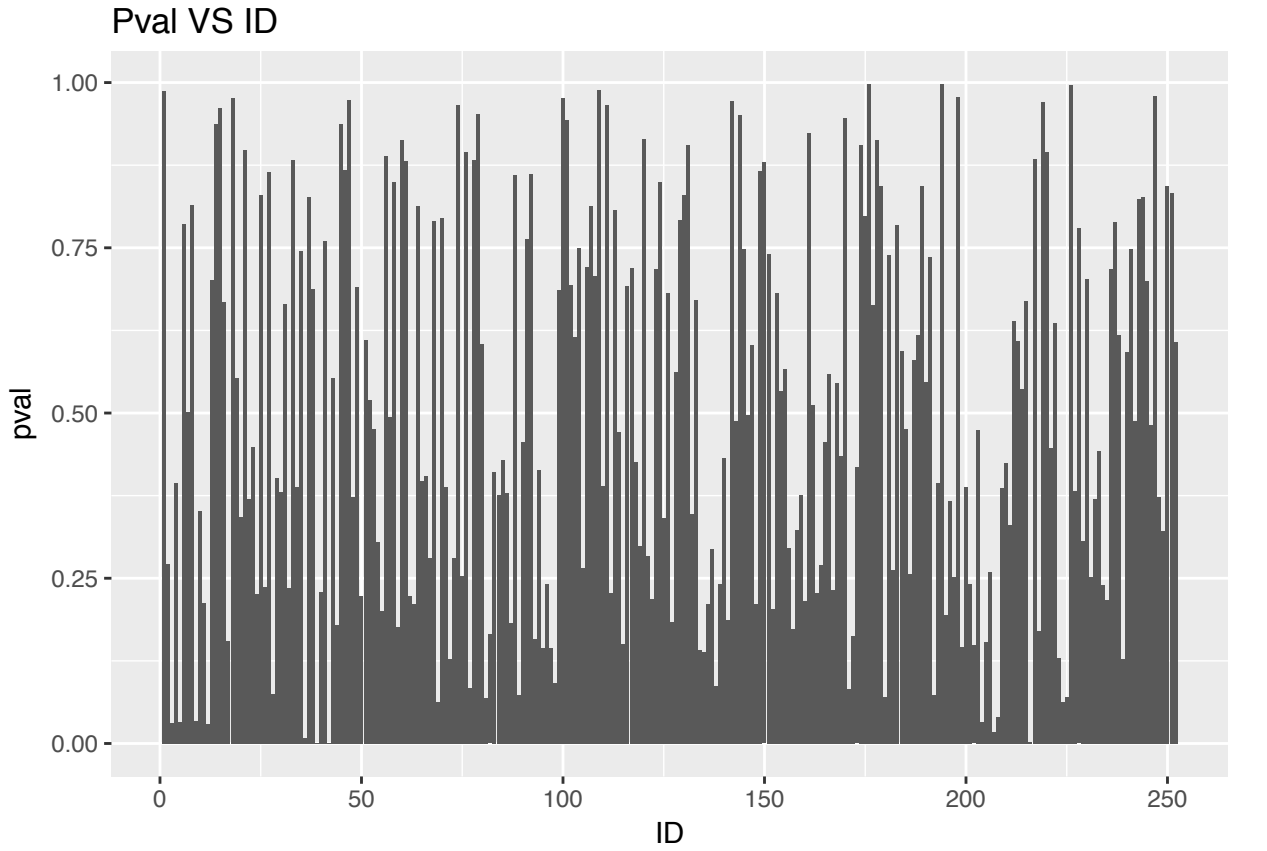
Seventh and Eighth plots: 39th and 42th elements are far away from the loess line and from the cloud(other points).

Ninth plot: The plot shows that 216th, 39th, and 42th elements have unusual Cook's distance.

We can confidently determine that 39th and 42th element as outliers and influential points. Also, 15th, 29th, 39th, 41th, 42th, 72th, 79th, 96th, 108th, 147th, 152th, 169th, 203th, 216th, 239th, 242th, 243th, and 252th elements can be potential outliers or influential points.

(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?

```
pval = sapply(std_predict_residuals, function(t) (pt(abs(t), n-p-2, lower.tail = F))*2)

ggplot(data = data.frame(ID = 1:length(pval), y = pval), aes(x = ID, y = pval)) + geom_col() + labs(tit
```




```
pval[pval <0.05]
```

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

This suggests 3th, 5th, 28th, 36th, 81th, 138th, 192th, 207th, 208th, and 216th as outliers and it's incorrect since each tests for each elements assume that there is 5% chance of being an outlier even when it is not. As we are doing 252 tests for each elements, it is expected that we can see as many as 13(n* 0.05) outliers even when there is no true outliers.

We can use Bonferroni correction to solve this issue by setting alpha = 0.05/n. However, since this correction doesn't give any outlier in this case since it is overly conservative.

```
pval[pval <0.05/n]
```

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

From the information that we shown, we can conclude that 39th and 42th elements are outliers and can be removed.

(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.

```
summary(fit)
```

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

```
bodyfat2 = bodyfat[-c(39,42), ]
summary(lm(bodyfat ~ Age + Weight + Height + Thigh, data = bodyfat2))
```

```
##
## Call:
## lm(formula = bodyfat ~ Age + Weight + Height + Thigh, data = bodyfat2)
##
## 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
```

Since F statistics increased after we removed two points that we determined as outliers, I believe it is better to fit the model without these two observations(39th and 42th elements).