Ram Yoogesh Gopu A3 - 20867060

Setting up

```
knitr::opts_chunk$set(echo = TRUE,
                      warning = FALSE,
                      message = FALSE,
                      fig.align = "center",
                      fig.width = 7,
                      fig.height = 6,
                      out.height = "60%")
set.seed(12314159)
library(loon.data)
library(loon)
```

```
## Loading required package: tcltk
```

```
library(gridExtra)

imageDirectory <- "./img"
dataDirectory <- "./data"
codeDirectory <- "/home/yoogesh/Desktop/stat847assignments/a3/"
path_concat <- function(path1, path2, sep="/") paste(path1, path2, sep = sep)


imageDirectory <- "/home/yoogesh/Desktop/stat847assignments/a3/img"  # e.g. in current "./img"
dataDirectory <- "/home/yoogesh/Desktop/stat847assignments/a3/data"  # e.g. in current "./data"
path_concat <- function(path1, path2, sep="/") paste(path1, path2, sep = sep)
```

(B)

```
load("blocks.rda")
head(blocks, n=3)
```

```
##   id weight perimeter group
## 1  1     55        32     B
## 2  2     35        27     B
## 3  3     35        25     A
```

(A)

(i)

```
print("printing All Mean")
```

```
## [1] "printing All Mean"
```

```
print("Weight")
```

```
## [1] "Weight"
```

```
mean(blocks$weight)
```

```
## [1] 32.4
```

```
print("Perimeter")
```

```
## [1] "Perimeter"
```

```
mean(blocks$perimeter)
```

```
## [1] 26.27
```

```
print("Printing Median")
```

```
## [1] "Printing Median"
```

```
print("Weights")
```

```
## [1] "Weights"
```

```
median(blocks$weight)
```

```
## [1] 30
```

```
print("Perimeter")
```

```
## [1] "Perimeter"
```

```
median(blocks$perimeter)
```

```
## [1] 26
```

```
print("printing Standard Deviation")
```

```
## [1] "printing Standard Deviation"
```

```
print("Weights")
```

```
## [1] "Weights"
```

```r
sd(blocks$weight)
```

```
## [1] 16.04098
```

```r
print("Perimeter")
```

```
## [1] "Perimeter"
```

```r
sd(blocks$perimeter)
```

```
## [1] 5.340629
```

(ii)

```r
library(dplyr)
bone = slice(blocks, which(blocks$group == "A"))
btwo = slice(blocks, which(blocks$group == "B"))

print("Block One")
```

```
## [1] "Block One"
```

```r
print("Mean weight")
```

```
## [1] "Mean weight"
```

```r
mean(bone$weight)
```

```
## [1] 21.1
```

```r
print("Mean perimeter")
```

```
## [1] "Mean perimeter"
```

```r
mean(bone$perimeter)
```

```
## [1] 22.22
```

```r
print("Median weight")
```

```
## [1] "Median weight"
```

```r
median(bone$weight)
```

```
## [1] 20
```

```r
print("Median perimeter")
```

```
## [1] "Median perimeter"
```

```r
median(bone$perimeter)
```

```
## [1] 22
```

```r
print("Standard Deviation weight")
```

```
## [1] "Standard Deviation weight"
```

```r
sd(bone$weight)
```

```
## [1] 7.304597
```

```r
print("Standard Deviation perimete")
```

```
## [1] "Standard Deviation perimete"
```

```r
sd(bone$perimeter)
```

```
## [1] 2.816062
```

```r
print("Block Two")
```

```
## [1] "Block Two"
```

```r
print("Mean weightw")
```

```
## [1] "Mean weightw"
```

```r
mean(btwo$weight)
```

```
## [1] 43.7
```

```r
print("Mean perimeter")
```

```
## [1] "Mean perimeter"
```

```r
mean(btwo$perimeter)
```

```
## [1] 30.32
```

```r
print("Median weight")
```

```
## [1] "Median weight"
```

```r
median(btwo$weight)
```

```
## [1] 40
```

```r
print("Median of Perimeter")
```

```
## [1] "Median of Perimeter"
```

```r
median(btwo$perimeter)
```

```
## [1] 30
```

```r
print("Standard Deviation Weight")
```

```
## [1] "Standard Deviation Weight"
```

```r
sd(btwo$weight)
```

```
## [1] 14.35021
```

```r
print("Standard Deviation perimeter")
```

```
## [1] "Standard Deviation perimeter"
```

```r
sd(btwo$perimeter)
```

```
## [1] 4.027659
```

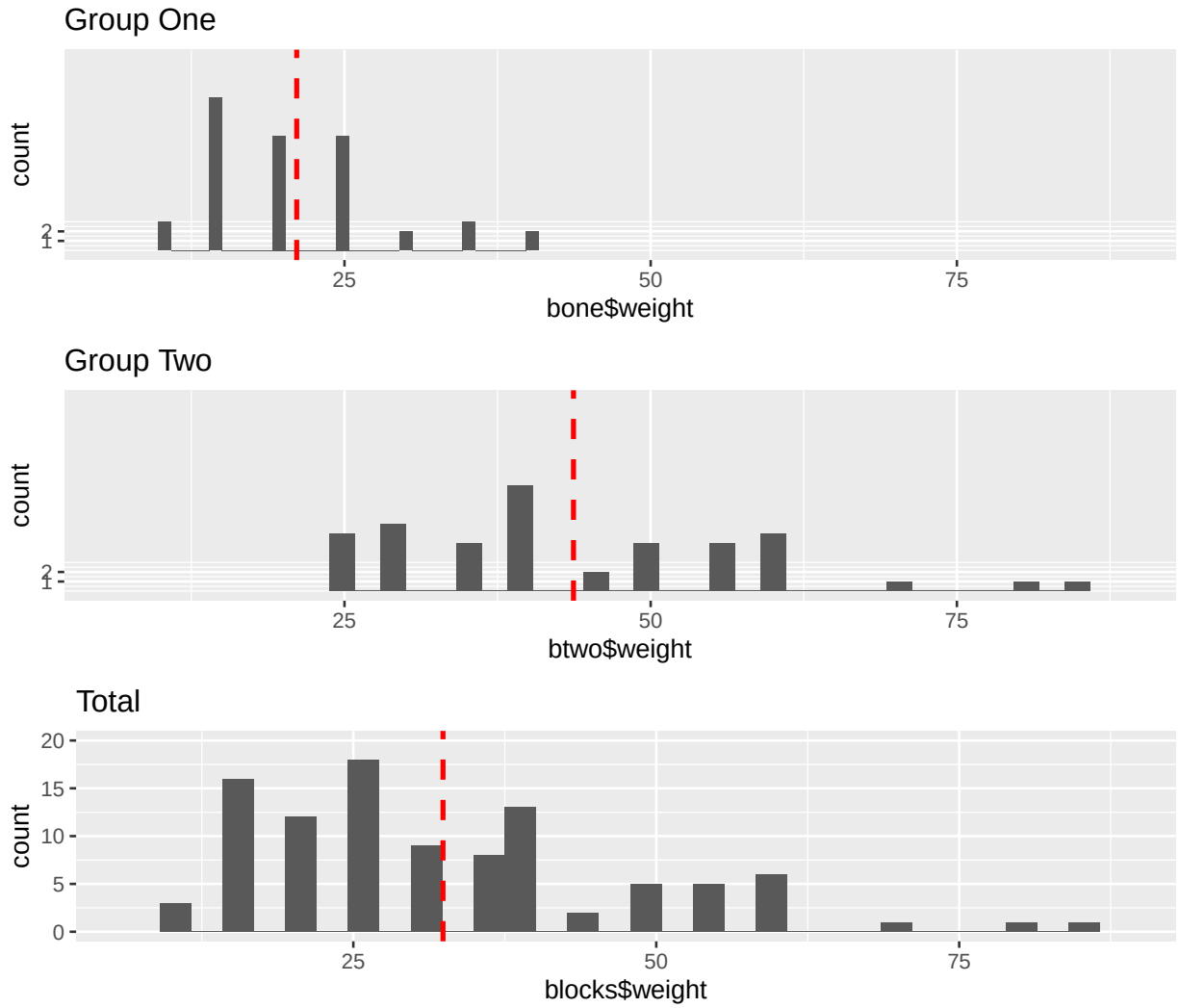
(iii)

From the above results we can infer that all the mean, median and weights are almost double for the block Two. The standard deviation for Block 1 is very low. Standard deviation for block two is also, but the overall value of the standard deviation is very high. Also, the mean of the perimeter of block Two is almost close to the perimeter of the whole group. Hence we can conclude that Block Two has the largest blocks.

(B)

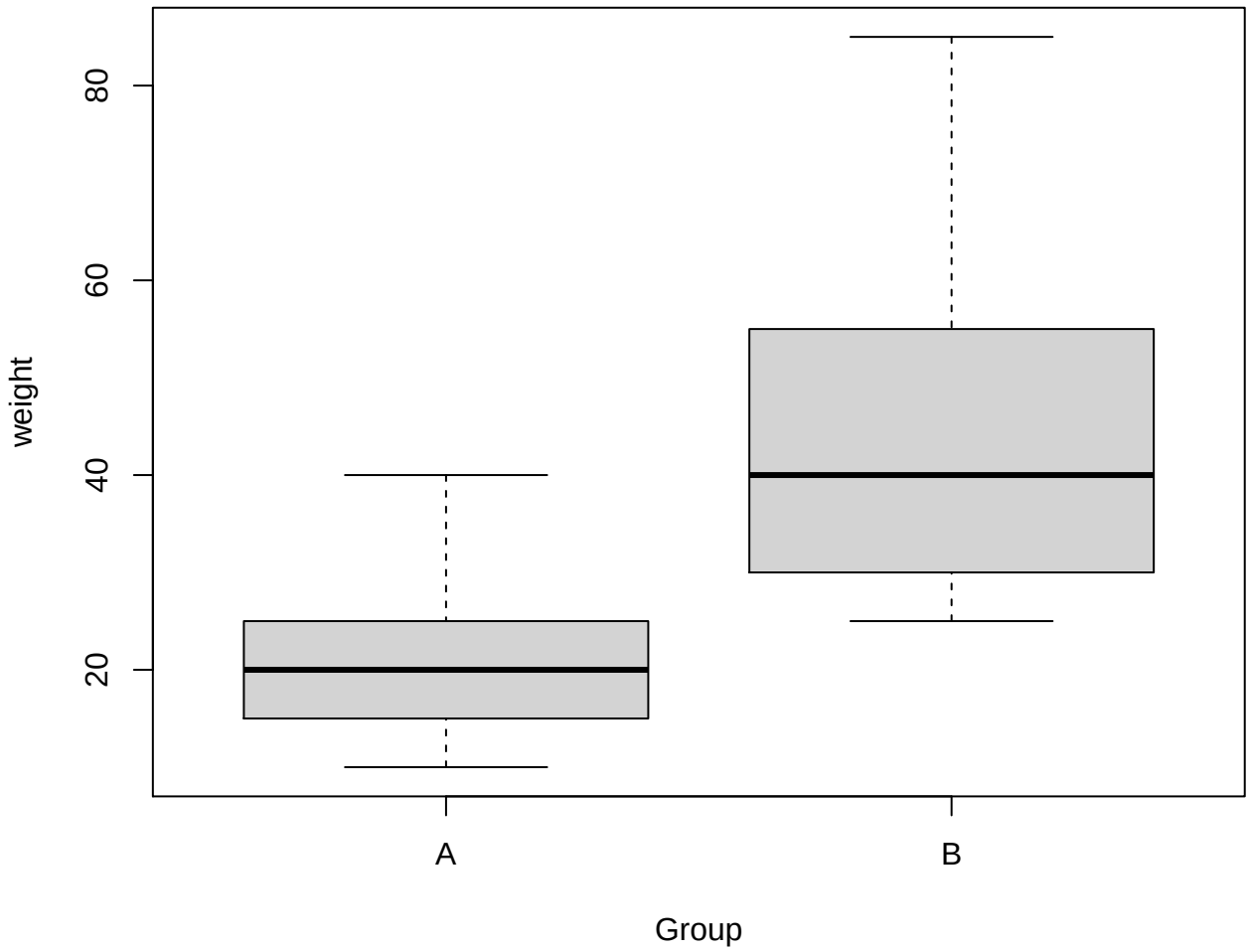
(i)

```r
library(ggplot2)
library(gridExtra)

hone <- ggplot(data=bone, aes(bone$weight)) +
                coord_cartesian(xlim = extendrange(blocks$weight), ylim = c(0,20)) +
                scale_y_continuous(breaks=seq) +
                geom_histogram() +
                geom_vline(xintercept=mean(bone$weight), linetype="dashed",
                color = "red", size=1) +
                ggtitle("Group One")


htwo <-  ggplot(data=btwo, aes(btwo$weight)) +
                coord_cartesian(xlim = extendrange(blocks$weight), ylim = c(0,20)) +
                scale_y_continuous(breaks=seq) +
                geom_histogram() +
                geom_vline(xintercept=mean(btwo$weight), linetype="dashed",
                color = "red", size=1) +
                ggtitle("Group Two")


histtotal <- ggplot(data=blocks, aes(blocks$weight)) +
                coord_cartesian(xlim = extendrange(blocks$weight), ylim = c(0,20)) +
                scale_y_continuous() +
                geom_histogram() +
                geom_vline(xintercept=mean(blocks$weight), linetype="dashed",
                color = "red", size=1) +
                ggtitle("Total")
grid.arrange(hone, htwo, histtotal,  ncol = 1)
```

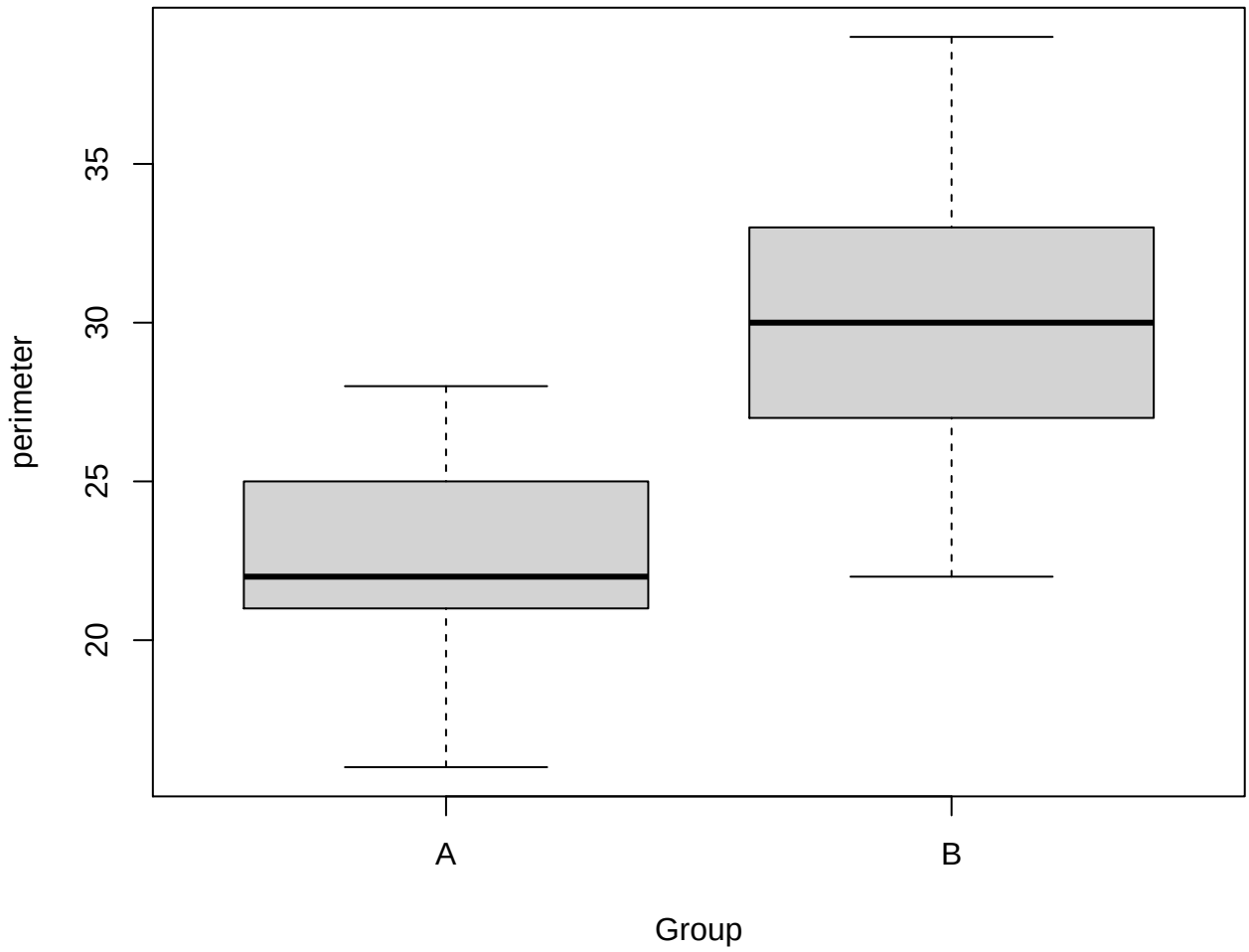


(ii)

```
boxplot(weight ~ group, xlab = "Group", data = blocks, col = "lightgrey")
```

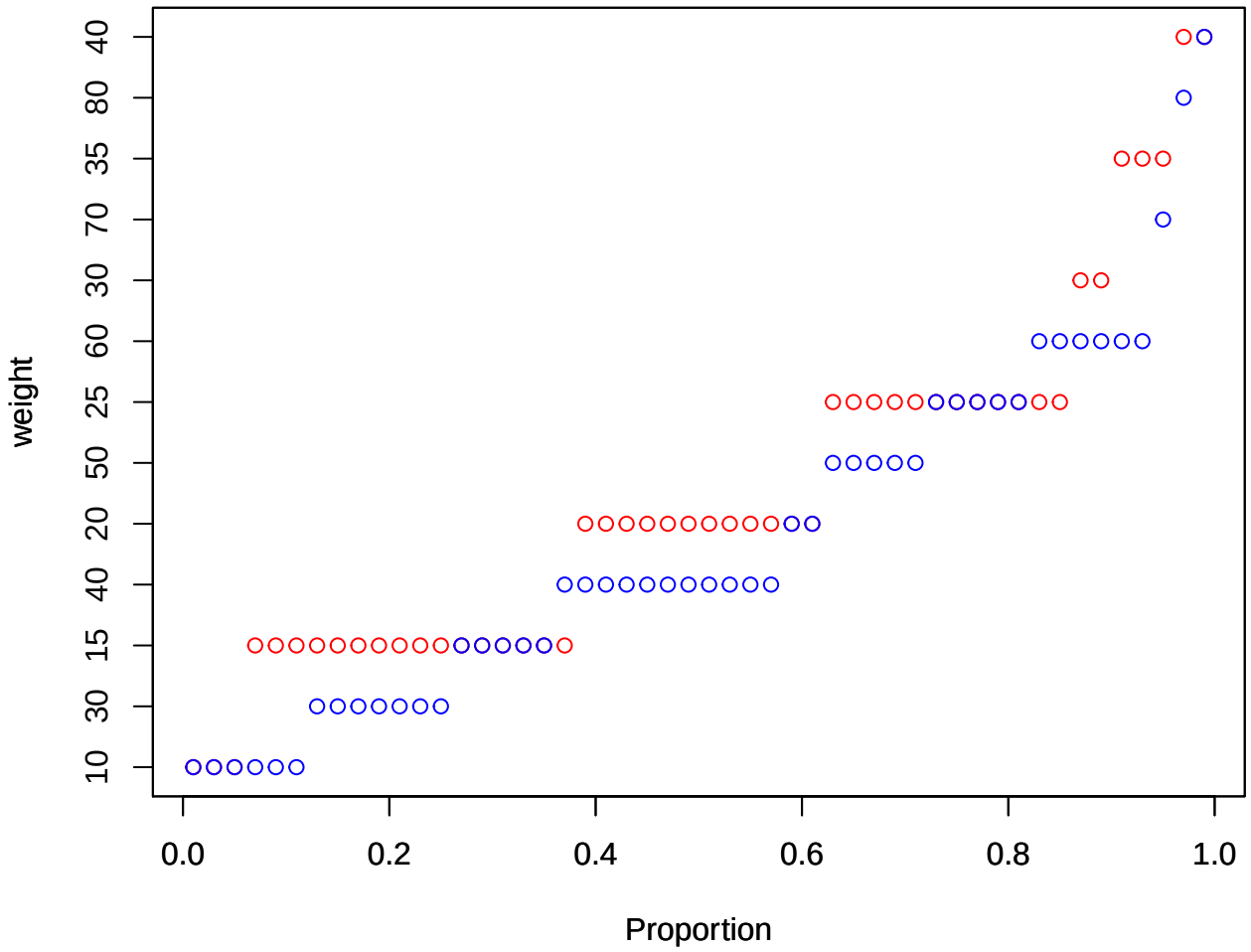



```
boxplot(perimeter ~ group, xlab = "Group", data = blocks, col = "lightgrey")
```



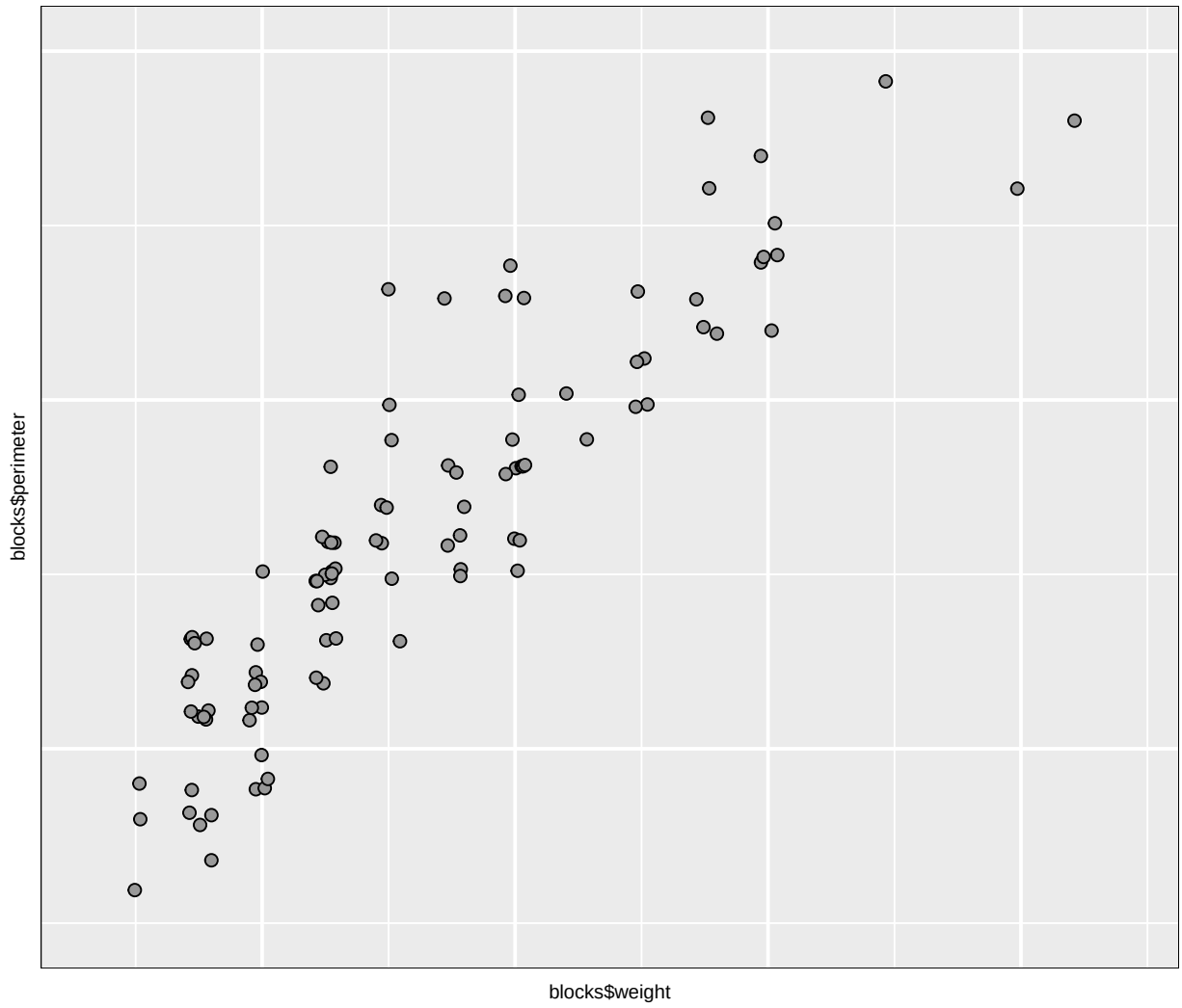
(iii)

```
a <- ppoints(length(bone$perimeter))
b <- ppoints(length(btwo$perimeter))
plot(x = a, y = sort(bone$weight), col = "red", xlab = "Proportion", ylab = "weight")
par(new = TRUE)
plot(x = b, y = sort(btwo$weight), col = "blue", xlab = "Proportion", ylab = "weight")
```

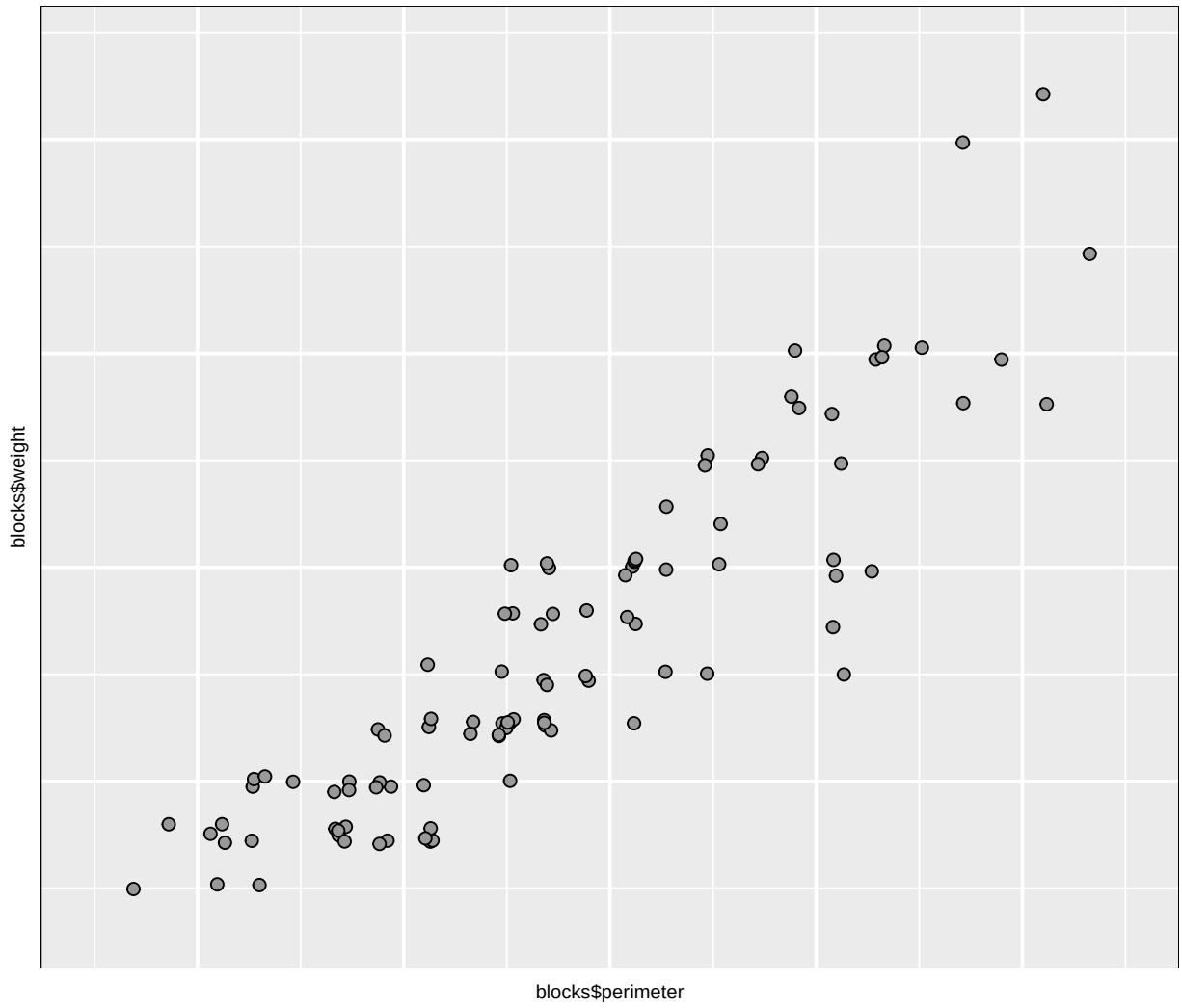


(C)

```
p <- l_plot(blocks$weight, blocks$perimeter)
p["selected"] <- TRUE
l_move_jitter(p)
p["selected"] <- FALSE
plot(p)
```



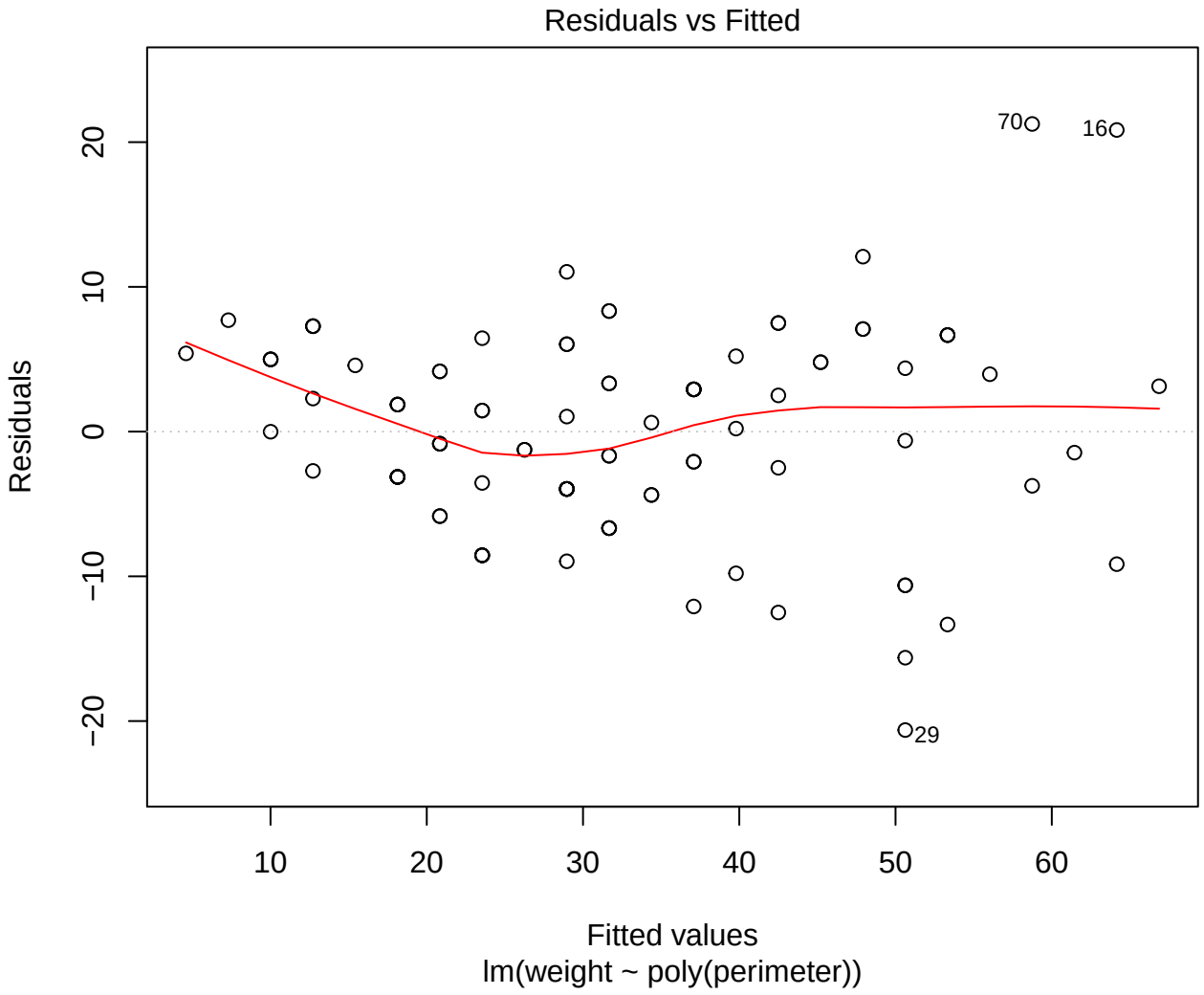

```
p["swapAxes"] <- TRUE
plot(p)
```

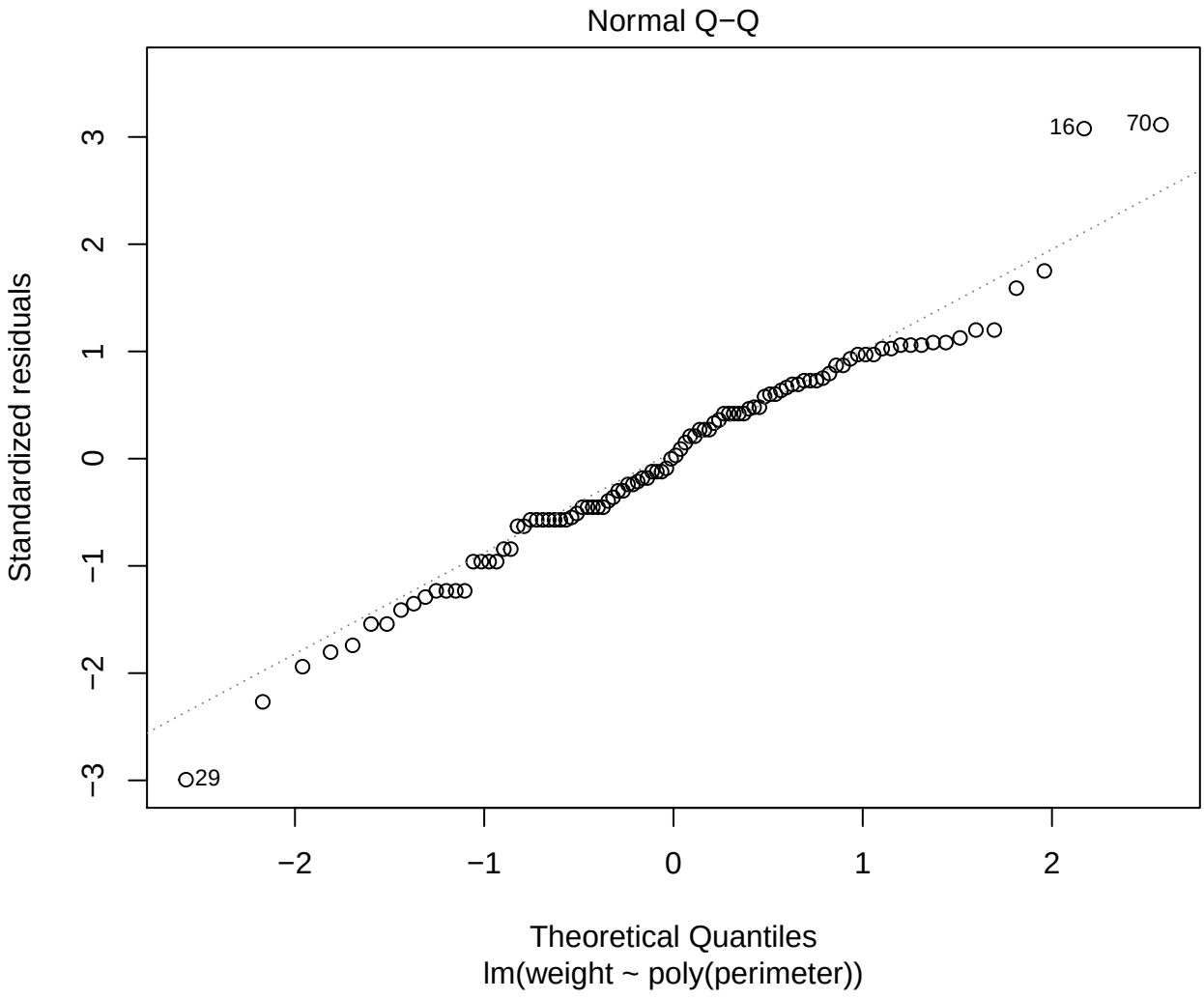


(D)

(i)

```
fit <- lm(weight ~ poly(perimeter), data = blocks)
plot(fit, which = c(1,2))
```


Residuals vs Fitted
Residuals
Fitted values
lm(weight ~ poly(perimeter))
70
16
29
20
10
0
−10
−20
10
20
30
40
50
60



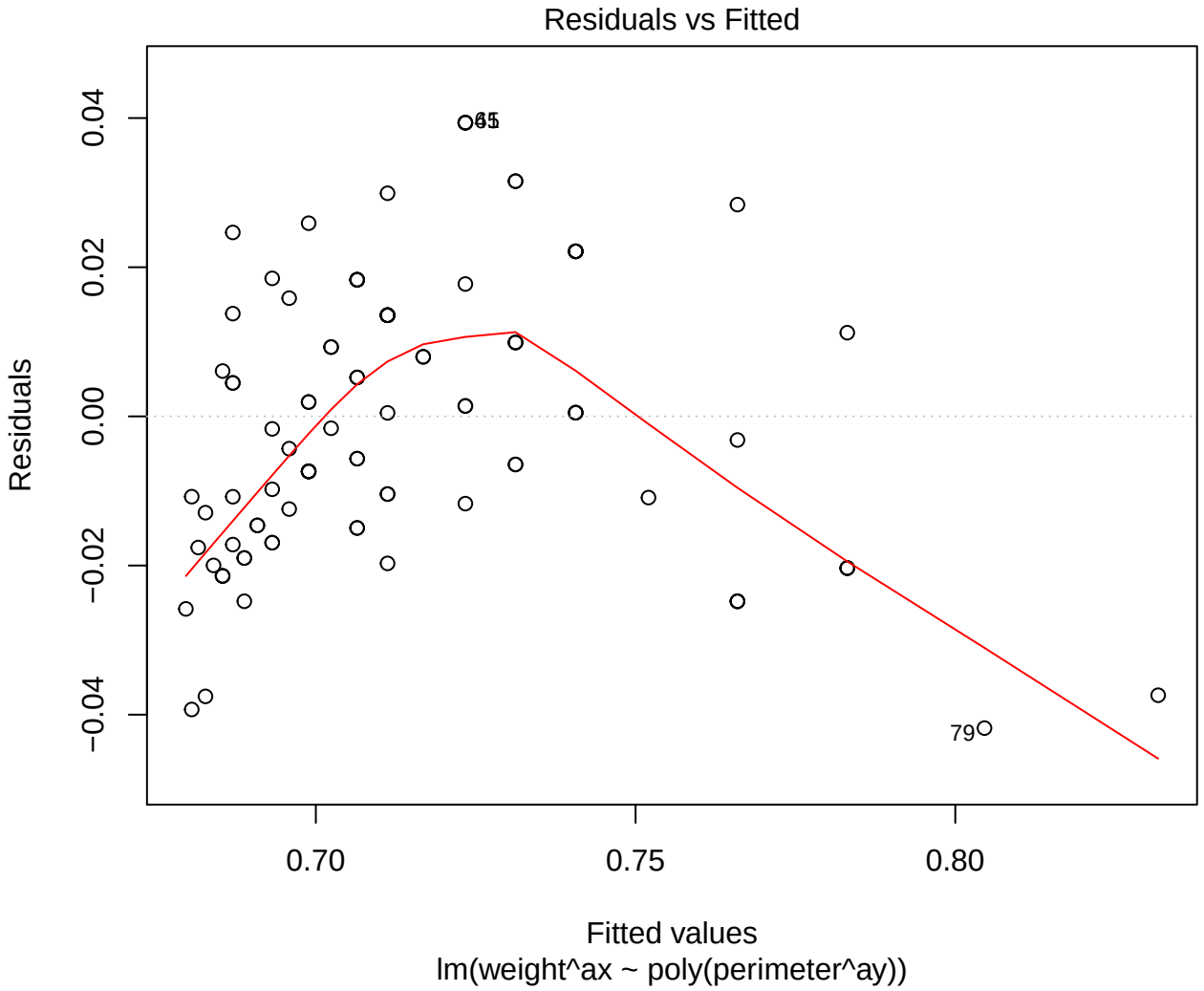

```
summary(fit)
```

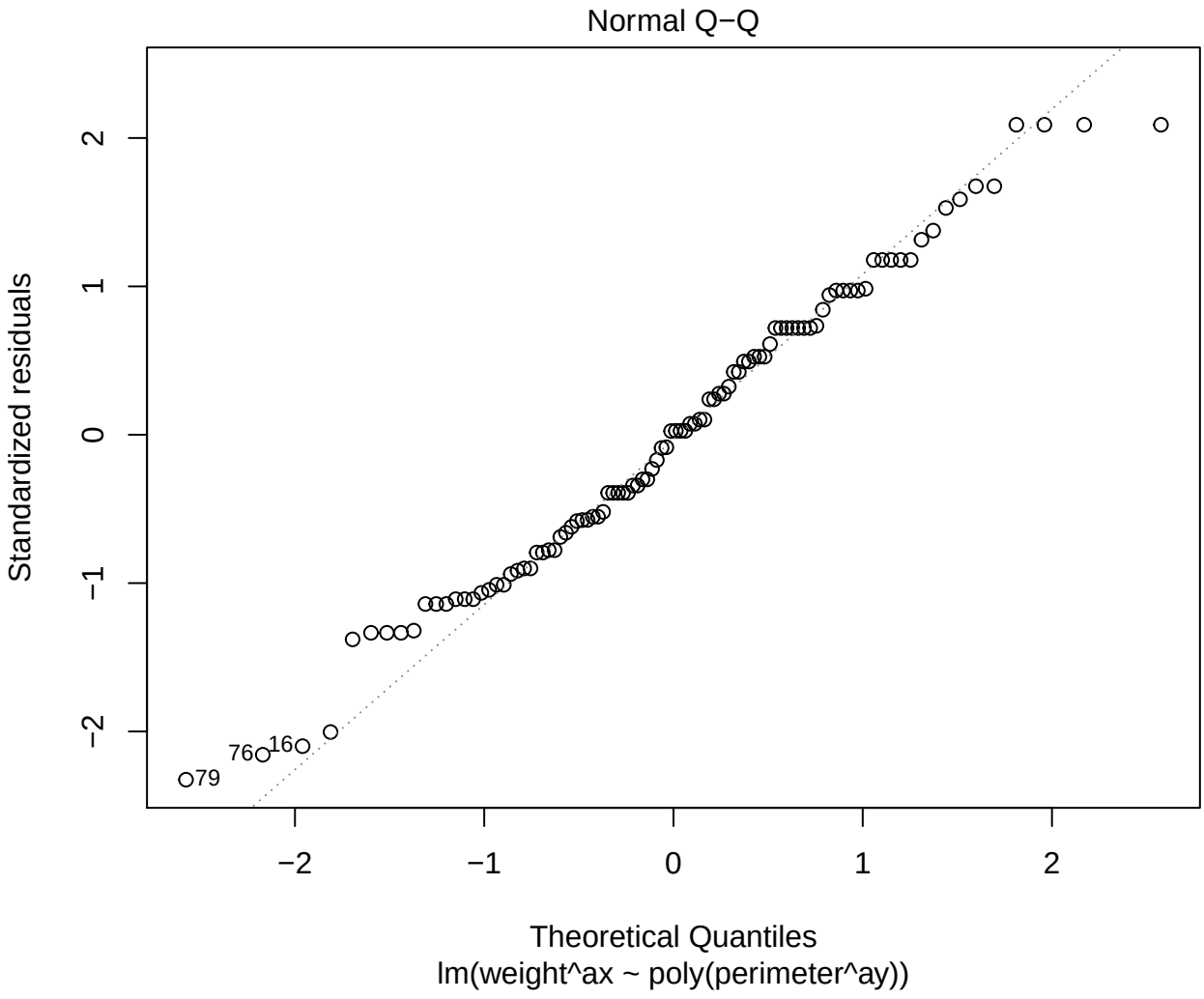
```
##
## Call:
## lm(formula = weight ~ poly(perimeter), data = blocks)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -20.6215  -3.9615   0.0998   4.8429  21.2560
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)       32.400      0.698   46.42   <2e-16 ***
## poly(perimeter)  143.873      6.980   20.61   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.98 on 98 degrees of freedom
## Multiple R-squared:  0.8126, Adjusted R-squared:  0.8107
## F-statistic: 424.9 on 1 and 98 DF,  p-value: < 2.2e-16
```

Here we fit the model , though the line doesnt fit all the points perfectly , it does cover up most of the lien. So i can say that the variability is really low.

(ii)

```
source("power_xy.r")

p.weight.perimeter <- with(blocks, power_xy(x=perimeter, y=weight, xlab="Perimeter", ylab="Weights", ti
ax = -0.1
ay = -3.0
fit <- lm(weight ^ ax ~ poly(perimeter ^ ay), data = blocks)
plot(fit, which = c(1,2))
```






```
summary(fit)
```

```
##
## Call:
## lm(formula = weight^ax ~ poly(perimeter^ay), data = blocks)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.041798 -0.014703  0.000482  0.013562  0.039380
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)         0.715497   0.001895  377.57   <2e-16 ***
## poly(perimeter^ay) 0.297180   0.018950   15.68   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.01895 on 98 degrees of freedom
## Multiple R-squared:  0.7151, Adjusted R-squared:  0.7122
## F-statistic: 245.9 on 1 and 98 DF,  p-value: < 2.2e-16
```

(iii)

We can infer from the first line that there is no proper fit , and it seems to have a patters. On the contrary we can see that the second line fits very well , i.e Most of the points are near the line. Hence we can say that the adjusted R square will be high for the first line. So i would choose the second model.

(iv)

There is no meaning in it. Because for the same value of the perimeter, we tend to have larger surface area for different 2 different blocks.