EDA A3 Q1:Population of blocks

Consider a study population $\mathcal{P}_{Study}$ consisting of $N = 100$ blocks labelled $u = 1, 2, 3, \ldots, 100$.

The blocks are of uniform thickness and density (all blocks were cut from the same opaque plastic sheet of about 5 mm thickness), but have different shapes.

Data on this population of 100 blocks are available as an R data set 'blocks'. This data set has four variates: block id number, weight in grams, perimeter in centimetres, and the group of the block (being either A or B). It can be loaded from the assignment data directory as follows:

```
load("/Users/rudranibhadra/Downloads/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
```

In this question, you will examine different possible attributes of interest for this population.

a. Simple numerical attributes.

b. *(1 mark)* Summarize this population by the following attributes on the variates 'weight' and 'perimeter': the population median, mean, and standard deviation (here computed using 'sd()' with denominator $N - 1$).

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

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

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

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

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

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

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

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

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

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

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

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

ii. *(1 mark)* Repeat the above summaries but now conditional on the group to which each block belongs. Now include the number in each group.

```
#For group A
mean(blocks$weight[blocks$group=="A"])
```

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

```
median(blocks$weight[blocks$group=="A"])
```

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

```r
sd(blocks$weight[blocks$group=="A"])
```

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

```r
mean(blocks$perimeter[blocks$group=="A"])
```

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

```r
median(blocks$perimeter[blocks$group=="A"])
```

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

```r
sd(blocks$perimeter[blocks$group=="A"])
```

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

```r
#For group B
mean(blocks$weight[blocks$group=="B"])
```

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

```r
median(blocks$weight[blocks$group=="B"])
```

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

```r
sd(blocks$weight[blocks$group=="B"])
```

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

```r
mean(blocks$perimeter[blocks$group=="B"])
```

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

```r
median(blocks$perimeter[blocks$group=="B"])
```

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

```r
sd(blocks$perimeter[blocks$group=="B"])
```

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

iii. *(3 marks)* On the basis of the above computed attributes, describe how each group differs from the whole population and from each other.

Based on weights: Mean and median of group A is lesser than the mean and median of all weights and of group B. Mean and median of group B is greater than mean and median of all weights. It means that group A contains lighter weights out of the whole collection compared to group B.

Standard deviation of group A is lower than the standard deviation of all weights and of group B. It means that group A contains weight measurements that are not that spread out as compared to combined group and group B. The entire group has the highest sd since it contains all weights of varying ranges.

Based on perimeter: Mean and median of group A is lesser than the mean and median of all perimeters and of group B. Mean and median of group B is greater than mean and median of all perimeters. It means that group A contains smaller blocks compared to group B.

Standard deviation of group A is lower than the standard deviation of all weights and of group B. It means that group A contains perimeter measurements are not that spread out as compared to combined group and group B. The entire group has the highest sd since it contains all perimeters of varying ranges.

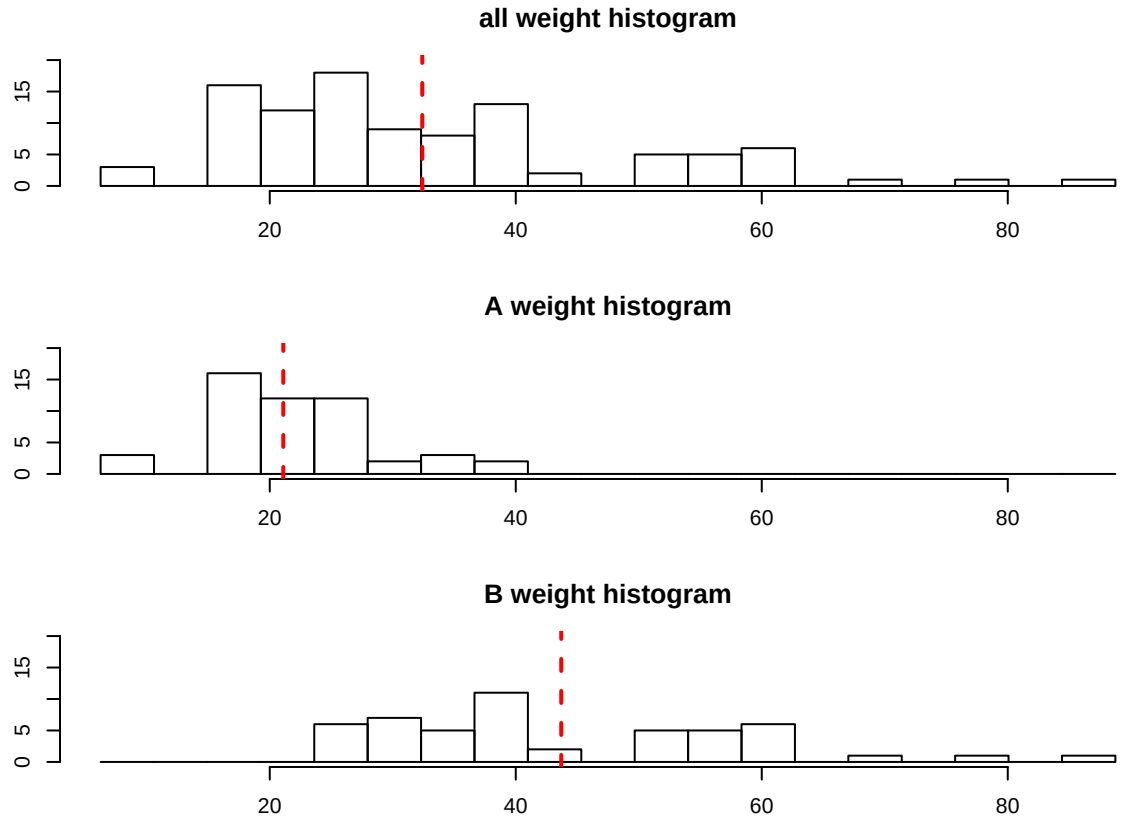
b. Simple graphical attributes.

c. *(8 marks)* Draw (suitably labelled) histograms of the weight for the whole population, only the blocks in group 'A', and only the blocks in group 'B'. Make sure you use the same 'xlim = extendrange(blocks$weight)', the same 'ylim = c(0,20)', and the same 'breaks <- seq(min(xlim), max(xlim), length.out = 20)' in all histograms. Add a vertical dashed red line at the average of the blocks in each case. Arrange the three plots so that they appear above one another in your display.

Comment on the differences between histograms.

Show your code.

```
par(mfrow=c(3,1),mar=c(3,3,3,3))

xlim=extendrange(blocks$weight)
hist(blocks$weight, main="all weight histogram",xlab="weights",xlim=extendrange(blocks$weight),
ylim = c(0,20),breaks=seq(min(xlim), max(xlim), length.out = 20) )
abline(v=mean(blocks$weight),col="red",lwd=2,lty=2)

hist(blocks$weight[blocks$group=="A"], main="A weight histogram",xlab="weights of A group",
xlim=extendrange(blocks$weight),ylim = c(0,20),breaks=seq(min(xlim), max(xlim), length.out = 20) )
abline(v=mean(blocks$weight[blocks$group=="A"]),col="red",lwd=2,lty=2)

hist(blocks$weight[blocks$group=="B"], main="B weight histogram",xlab="weights of B group",
xlim=extendrange(blocks$weight),ylim = c(0,20),breaks=seq(min(xlim), max(xlim), length.out = 20) )
abline(v=mean(blocks$weight[blocks$group=="B"]),col="red",lwd=2,lty=2)
```



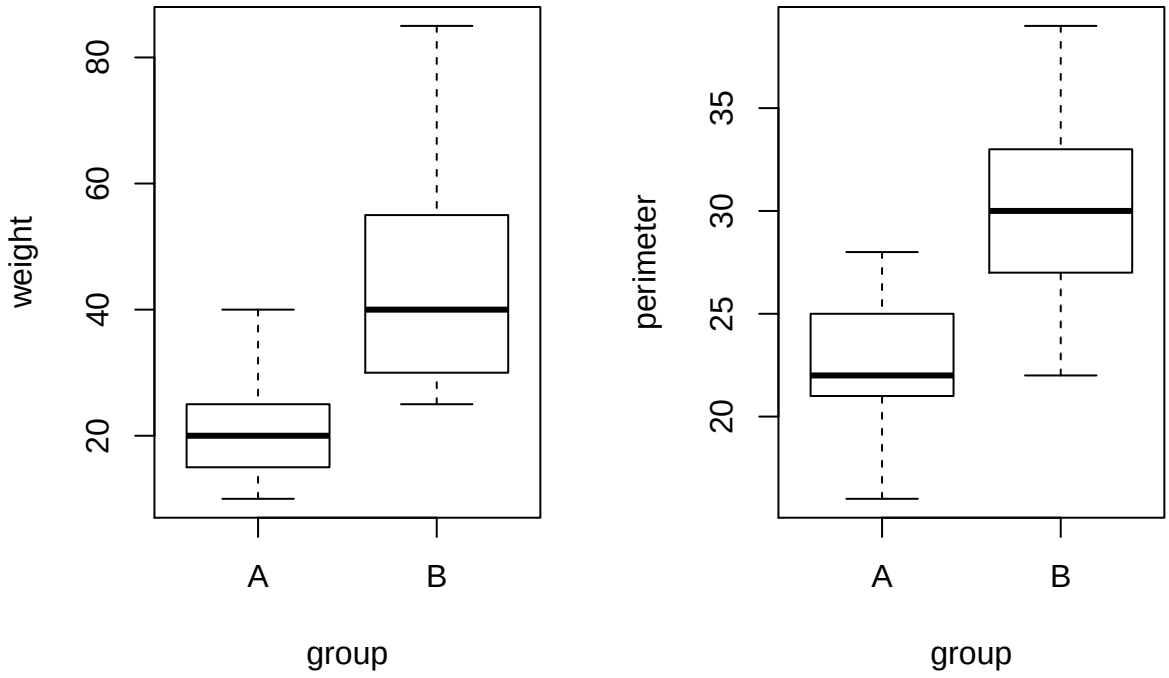
From the above histograms we can see that the maximum range of A weight histogram is around 40. The maximum range for B weight histogram exceeds 80. The all weight histogram and A weight histogram seem to be right skewed compared to B weight histogram. The mean value of A weight histogram is lower than

the mean of other two histograms.

ii. *(6 marks)* Using formula notation, draw pairs of (suitably labelled) boxplots comparing the two groups, first with respect to a difference in block weights and then with respect to the perimeters. Comment on how the two groups compare.

Show your code.

```
par(mfrow=c(1,2))
boxplot(weight~group, data=blocks)
boxplot(perimeter~group, data=blocks)
```



From the above boxplots, we can see that the mean weight and mean perimeter of group B is higher than group A. The interquartile range of group B is also higher tha group A's.

iii. *(9 marks)* Quantile plots. A sample quantile plot is a scatterplot of the point pairs $(\frac{i-0.5}{n}, y_{(i)})$ for $i = 1, 2, \ldots, n$ and $y_{(1)} \leq y_{(2)} \leq \cdots \leq y_{(n)}$ is the sample *order statistic*. The values $\frac{i-0.5}{n}$ are returned by the R function 'ppoints()'.

On a single plot, overlay the sample quantiles for the perimeters of all the blocks, of the perimeters of those in group `A`, and of those in group 'B'. Use different colours and point symbols for each set. Add a legend to distinguish the groups.

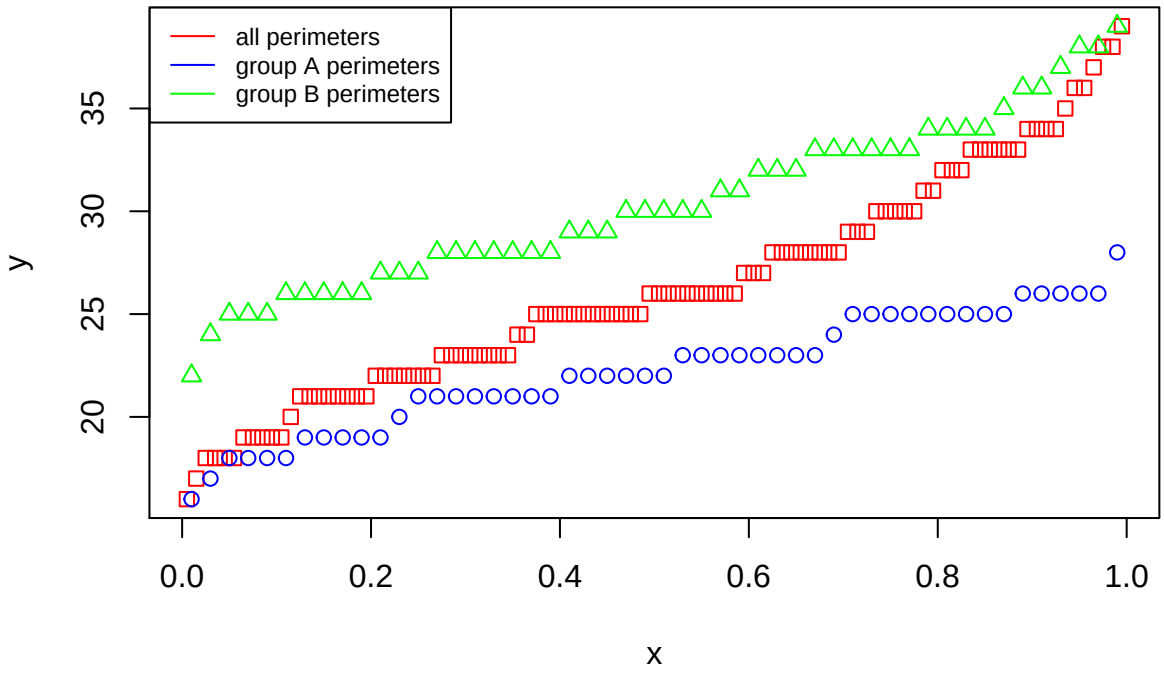
The three groups of points trace out different curves.

- what does a difference in heights of the curves near the middle show?
- what does a difference in the slopes of the curves near the middle tell you?

Comment on the differences between the three groups.

Show your code.

```r
n<-nrow(blocks)
x<-ppoints(n)
#x<-sort(blocks$perimeter)
#y<-(c(1:n)-0.5)/n
y<-sort(blocks$perimeter)
#y<-ppoints(n)
plot(x,y,col="red",pch=0)
#par(new=TRUE)
n1<-sum(blocks$group=="A")
x1<-ppoints(n1)
y1<-sort(blocks$perimeter[blocks$group=="A"])
#plot(x1,y1,col="blue")
#par(new=TRUE)
n2<-sum(blocks$group=="B")
x2<-ppoints(n2)
y2<-sort(blocks$perimeter[blocks$group=="B"])
#plot(x2,y2,col="green")
points(x1,y1,col="blue",pch=1)
points(x2,y2,col="green",pch=2)


legend("topleft", lty=c(1,1,1), col=c("red", "blue","green"),
       legend = c("all perimeters", "group A perimeters", "group B perimeters"),cex=0.75)
```



The difference in the heights in the middle tells us which group has the highest median. From the plot we can see that group B points have the highest median compared to group A and the entire population curve.

The difference in slopes in the middle tells us which group has a small or large interquartile range. Group A curve has a flatter slope compared to the otehr two curves so it has the smallest interquartile range. The combined groups curve and the group B has a steeper slope as compared to group A curve and thus have a higher interquartile range than A curve.

c. *(5 marks)* Scatterplots. In loon, use 'l_plot()' to plot 'weight' versus 'perimeter' (i.e. the pairs '(perimeter, weight)'); make sure you save the plot on an R variable (e.g. 'p <- l_plot(...)').

Then,

- select all of the points in the plot and jitter them
- this can be done programmatically using 'p["selected"] <- TRUE'
- and 'l_move_jitter(p)'
  - deselect the points programmatically
  - print the plot as part of your assignment
  - swap the axes programmatically and again print the plot
  - note: 'grid.arrange()' from the 'pkg{gridExtra}' package can be used to place the two plots side by side in your output

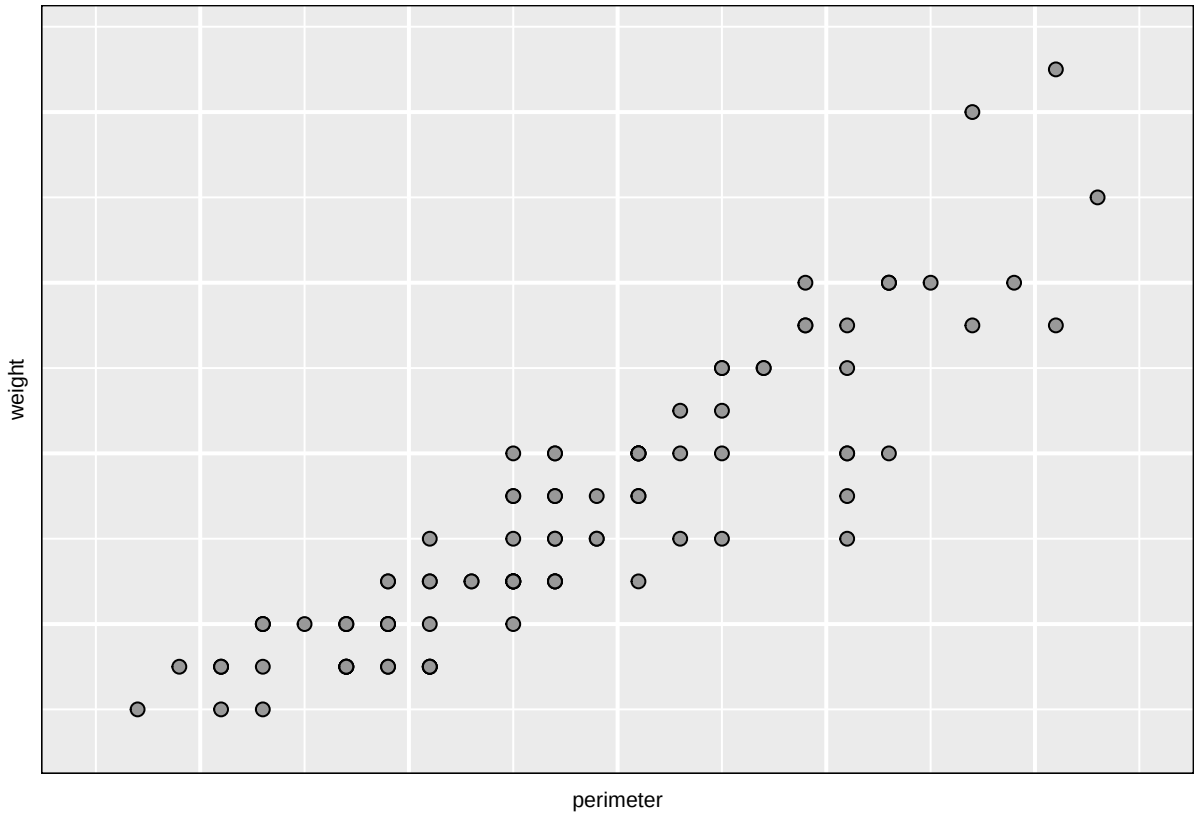
    Show your code.

Describe how these two variables appear to be related (if at all).

```
library(loon)
```

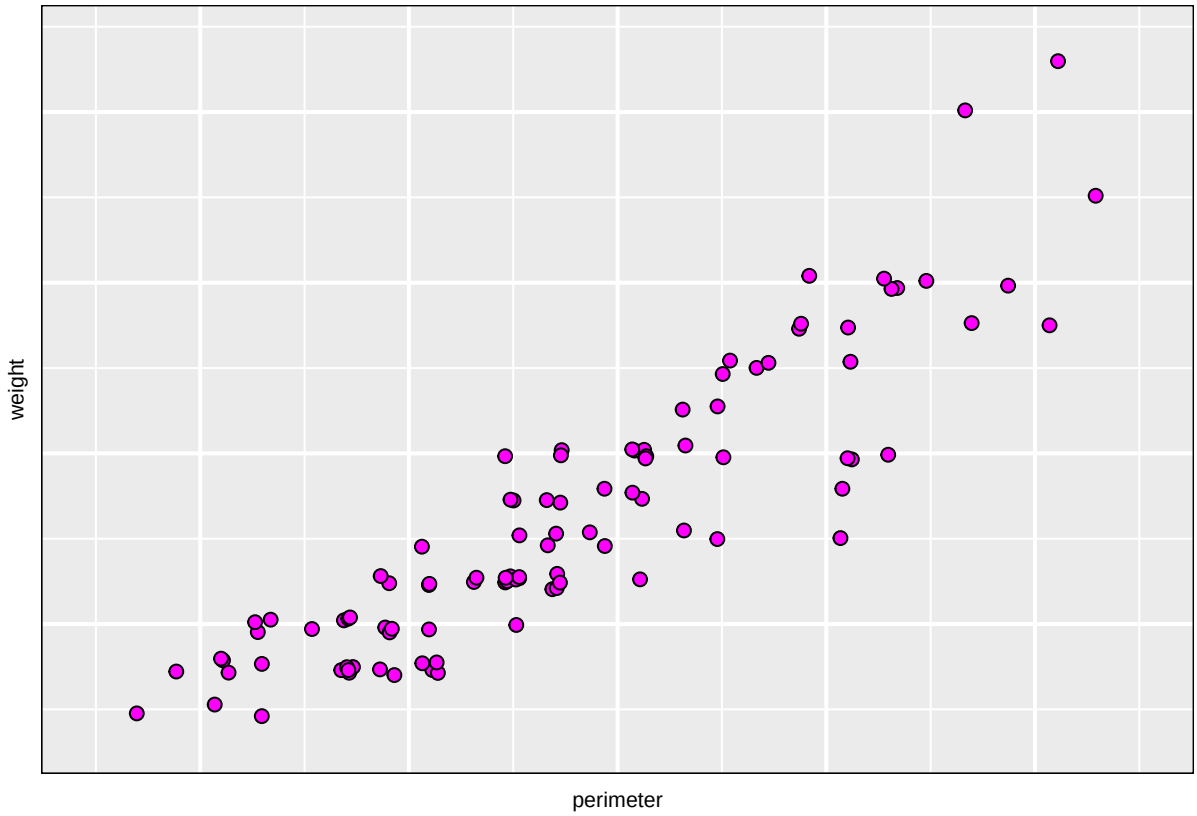
```
## Loading required package: tcltk
```

```
p<-l_plot(blocks$perimeter,blocks$weight,xlabel="perimeter",ylabel="weight")
plot(p)
```



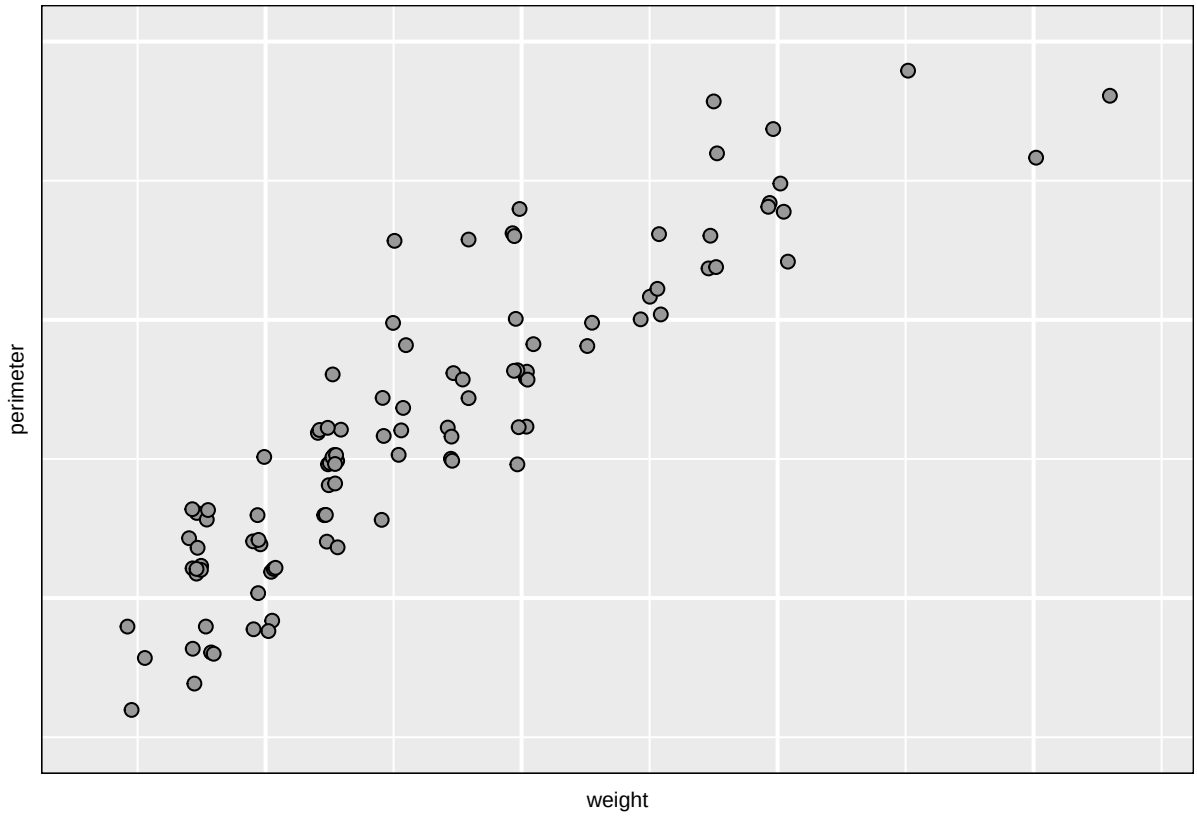

```
#p["selected"]<-TRUE
#l_move_jitter(p)
```

```
p["selected"]<-TRUE
l_move_jitter(p)
q<-plot(p)
```

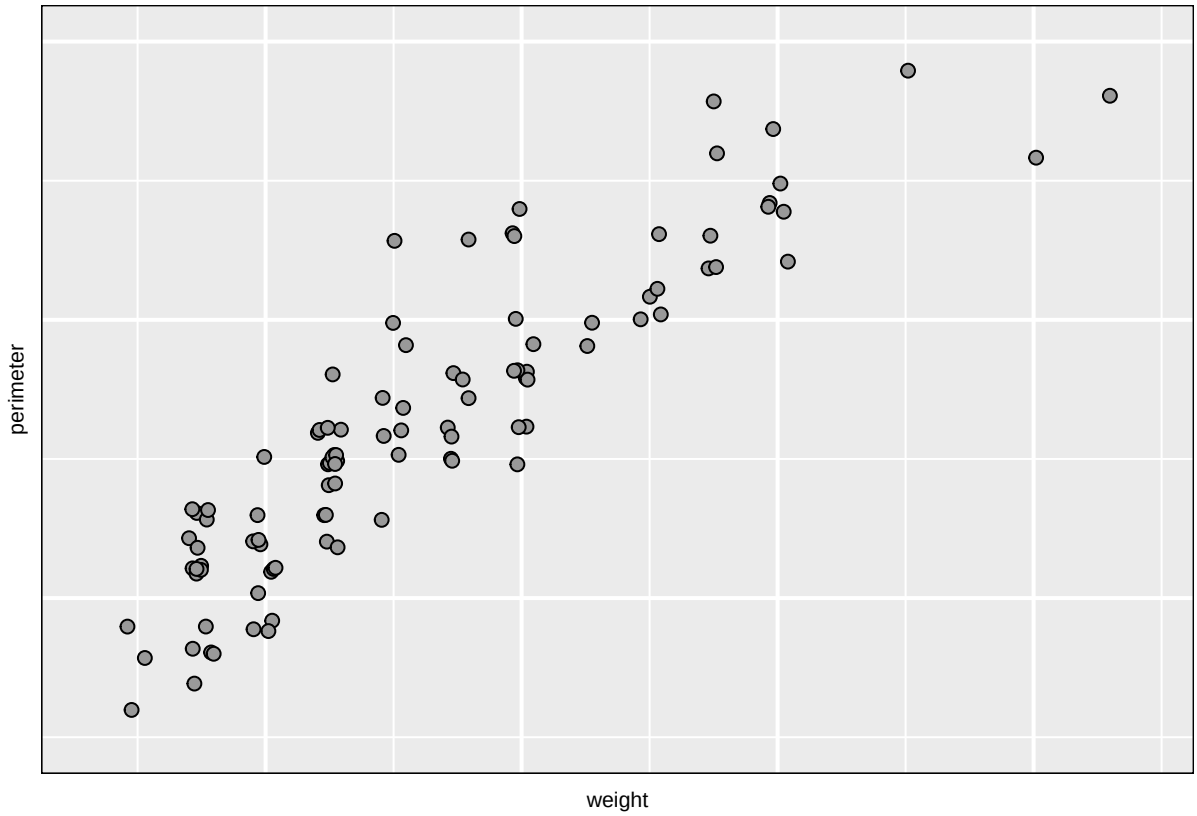



```
p["selected"]<-FALSE
```

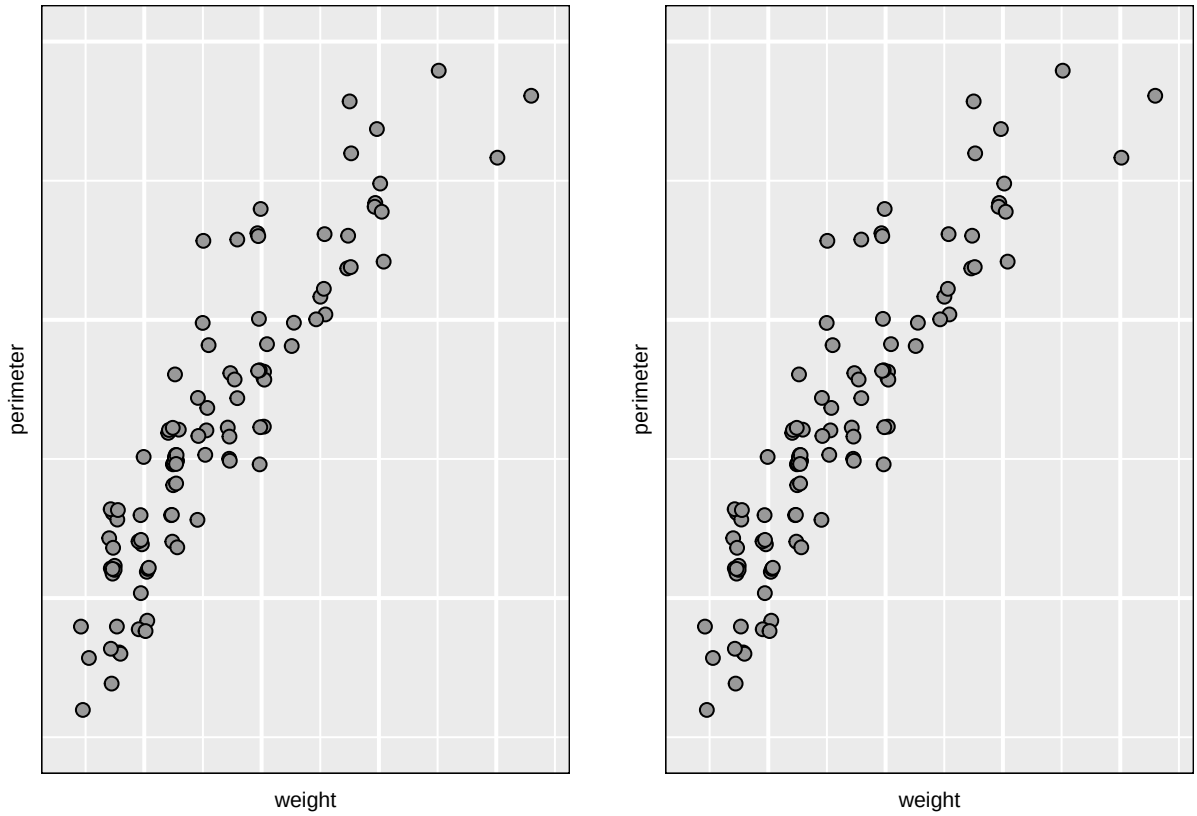
```
p["swapAxes"]<-TRUE
plot(p)
```




```
library(gridExtra)
#p<-l_plot(blocks$perimeter,blocks$weight,xlabel="perimeter",ylabel="weight")
p1<-loonGrob(p)
p["swapAxes"]<-TRUE
p2<-plot(p)
```




```
grid.arrange(grobs=list(p1,p2), nrow=1)
```



From the above loon plots, there seems to be a positive correlation between weight and perimeter. As one of them increase, the other variable also tends to increase.

d. Attributes given by fitted models.

e. *(5 marks)* Find the simplest polynomial that fits the 'weight' as a function of 'perimeter' for these blocks.

Show your code for fitting (only) the final model you choose and print a summary of the fitted model.

Using the 'fit' of your model examine (and submit) the two plots from 'plot(fit, which = c(1,2))'.

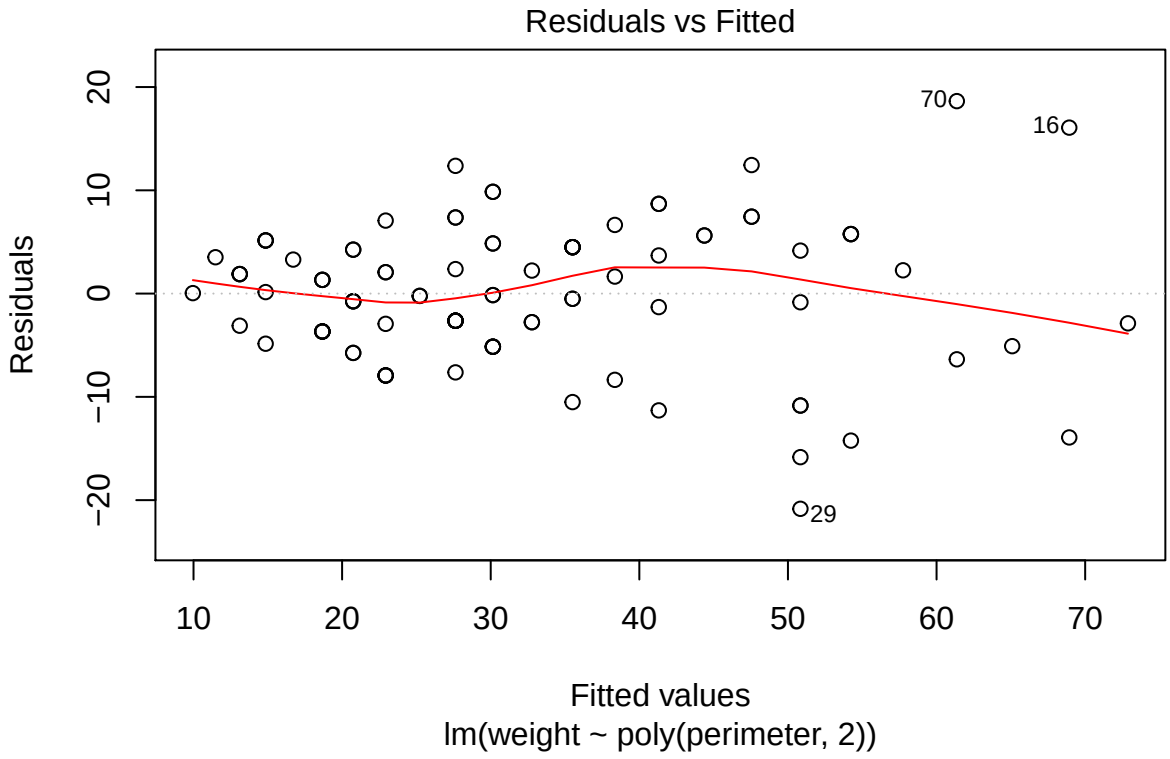
Describe the model you have selected and common on the quality of its fit.

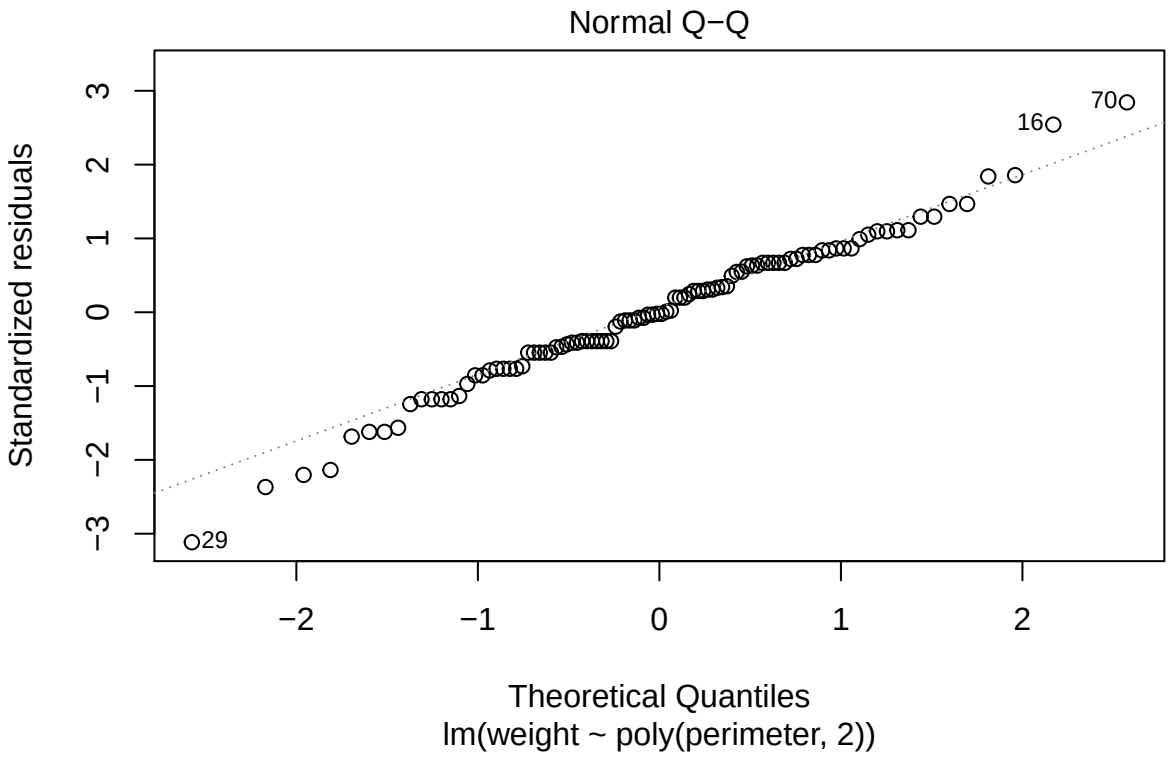
```
#fit<-lm(weight~perimeter, data=blocks)
#fit<-lm(weight~poly(perimeter,3), data=blocks)
#fit<-lm(weight~poly(perimeter,4), data=blocks)
#fit<-lm(weight~poly(perimeter,5), data=blocks)
fit<-lm(weight~poly(perimeter,2), data=blocks)
summary(fit)
```

```
##
## Call:
## lm(formula = weight ~ poly(perimeter, 2), data = blocks)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -20.8397  -3.6715  -0.1448   4.4938  18.6367
##
## Coefficients:
```

```
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)            32.4000     0.6777  47.806  < 2e-16 ***
## poly(perimeter, 2)1 143.8728     6.7774  21.228  < 2e-16 ***
## poly(perimeter, 2)2  17.8644     6.7774   2.636  0.00977 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.777 on 97 degrees of freedom
## Multiple R-squared:  0.8251, Adjusted R-squared:  0.8215
## F-statistic: 228.8 on 2 and 97 DF,  p-value: < 2.2e-16
```

```
plot(fit,which = c(1,2))
```





The model chosen is based on the relationship between (weight)^1 and (perimeter)^2. It has a good fit since its R squared value is close to 1 (0.8215)

ii. *(5 marks)* Use the 'power_xy()' function (from the course slides) on the perimeter (as x) and weight (as y). Find values of α_x and α_y on Tukey's ladder that make the least-squares line a plausible summary.

Use these values with 'lm()' to fit the model appearing as the straight line in your transformed plot. Note, if you have an integer power for 'perimeter', fit the model using 'poly()' with degree equal to the integer power.

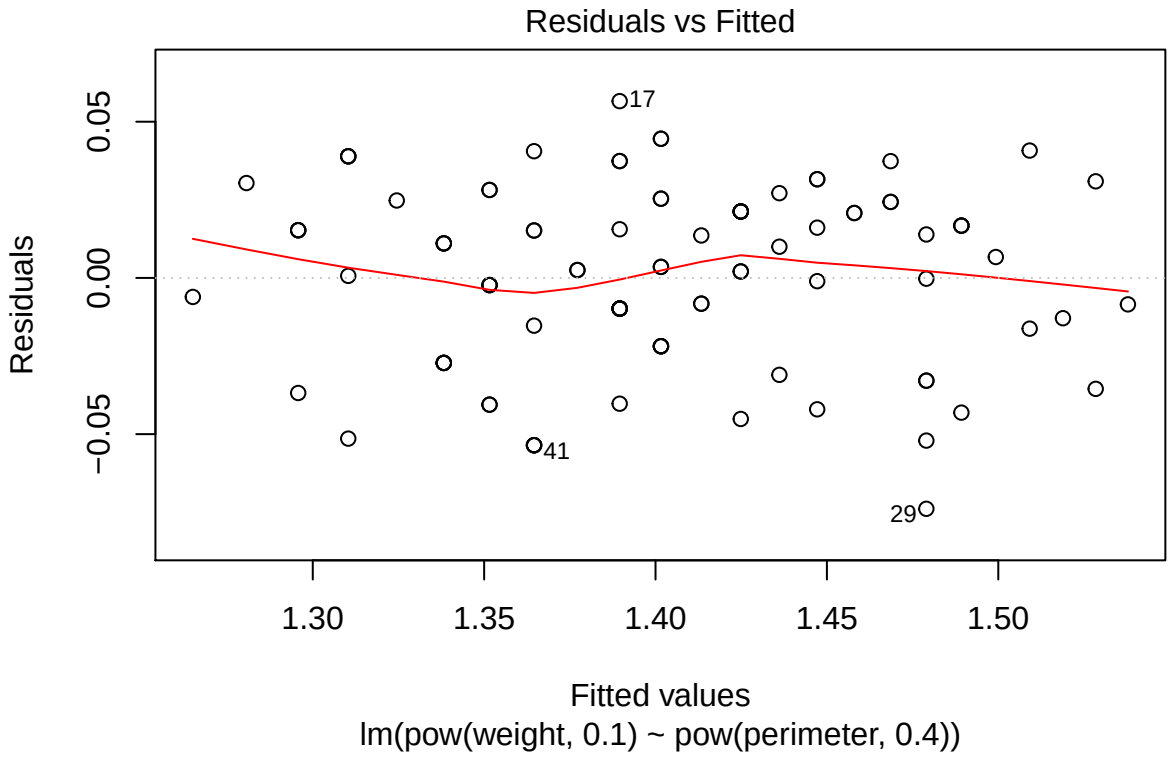
Show a summary of your fitted model and comment on the quality of its fit.

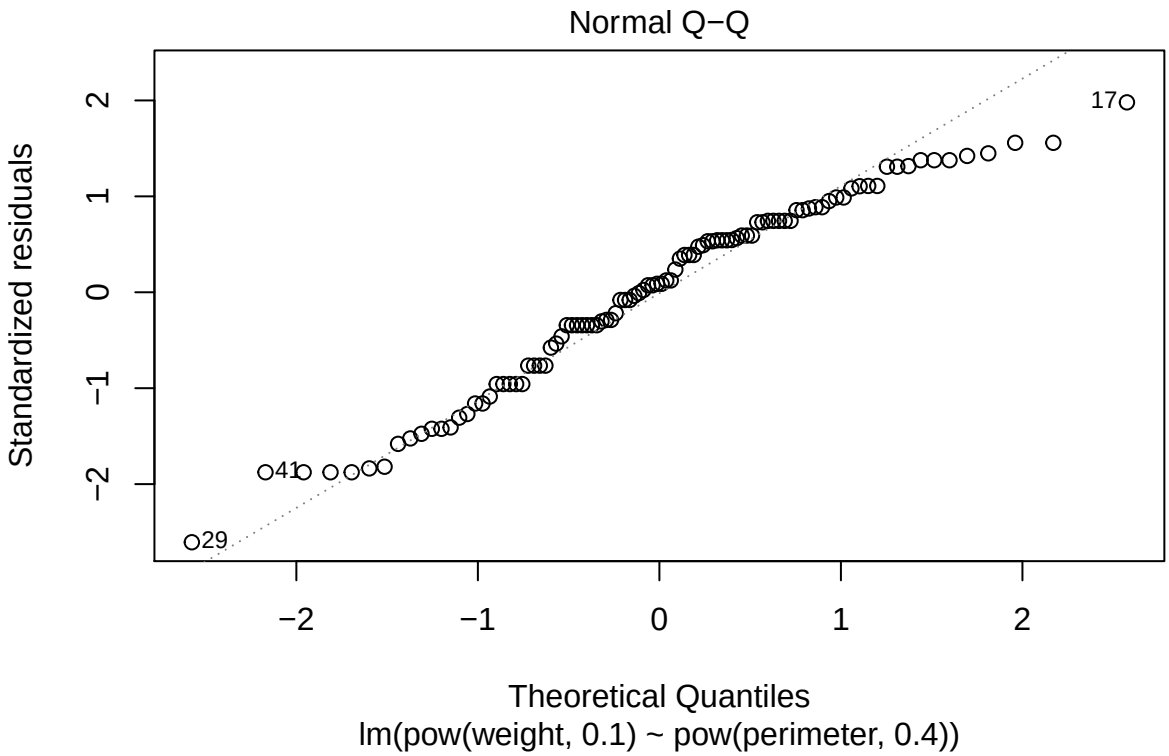
```
source("/Users/rudranibhadra/Downloads/power_xy.R")


p<-with(blocks,power_xy(x=perimeter,y=weight,xlab="weight",ylab="perimeter",linkingGroup = "b"))

pow<-function(x,y){x^y}
fit1<-lm(pow(weight,0.1) ~ pow(perimeter,0.4), data=blocks)
summary(fit1)
```

```
##
## Call:
## lm(formula = pow(weight, 0.1) ~ pow(perimeter, 0.4), data = blocks)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.073892 -0.021857  0.002523  0.021259  0.056583
##
```

```
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)          0.627839   0.035609   17.63   <2e-16 ***
## pow(perimeter, 0.4) 0.210189   0.009649   21.78   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.02873 on 98 degrees of freedom
## Multiple R-squared:  0.8288, Adjusted R-squared:  0.8271
## F-statistic: 474.5 on 1 and 98 DF,  p-value: < 2.2e-16
```

```
plot(fit1,which = c(1,2))
```





iii. *(3 marks)* Which of the two previous fitted models do you prefer? Explain why you choose that model.

The second model is better since the adjusted R-squared of the second model is greater than the first model. Since, closer the R-squared value to 1, better the fit of model. Since the adjusted R-squared value of second model is 0.8271 which is higher than the adjusted R-squared value of first model is 0.8215. The residual vs fitted plots for both models are similar so they cannot be compared.

iv. *(4 marks)* Explain why a model relating weight as a function only of perimeter makes no physical sense. How might the concept of study error be used to describe the problem? If, for example, our target population were to include all shaped blocks then any model summary based on the relationship between weight and perimeter would be quite different for our study population of convex blocks. This constitutes study error.

Study error is difference between the attribute values for the study population and the target population. The model which is based on the relationship between weight and perimeter according to the dataset given may not be used on the target dataset since it is not given. So the target population might consist of completely different blocks from the given study population. If a model is fitted on the target population, it might be completely different from the model used to fit the study population.