

Contents

1	Exercise 3.1	2
2	Exercise 3.2	5
3	Exercise 3.3	9
4	Exercise 3.4	13
5	Exercise 3.5	14

1 Exercise 3.1

```
wip <- read.table("Lab03wip.txt", header=T)
wip
```

```
## # A tibble: 40 x 2
##   time plant
##   <dbl> <int>
## 1  5.62     1
## 2  5.29     1
## 3 16.2     1
## 4 10.9     1
## 5 11.5     1
## 6 21.6     1
## 7  8.45     1
## 8  8.58     1
## 9  5.41     1
## 10 11.4     1
## # i 30 more rows
```

Base R style:

```
wip1 <- wip[wip$plant==1, "time"]
wip2 <- wip[wip$plant==2, "time"]

mean(wip1); mean(wip2)
```

```
## [1] 9.382
```

```
## [1] 11.3535
```

```
median(wip1); median(wip2)
```

```
## [1] 8.515
```

```
## [1] 11.96
```

```
quantile(wip1, c(0, 0.25, 0.5, 0.75, 1)); quantile(wip2, c(0, 0.25, 0.5,
↪ 0.75, 1))
```

```
##      0%      25%      50%      75%     100%  
##  4.4200  7.4475  8.5150 11.0450 21.6200
```

```
##      0%      25%      50%      75%     100%  
##  2.330  8.440 11.960 13.845 25.750
```

```
min(wip1); min(wip2)
```

```
## [1] 4.42
```

```
## [1] 2.33
```

```
max(wip1); max(wip2)
```

```
## [1] 21.62
```

```
## [1] 25.75
```

```
range(wip1); range(wip2)
```

```
## [1]  4.42 21.62
```

```
## [1]  2.33 25.75
```

```
IQR(wip1); IQR(wip2)
```

```
## [1] 3.5975
```

```
## [1] 5.405
```

```
var(wip1); var(wip2)
```

```
## [1] 15.98123
```

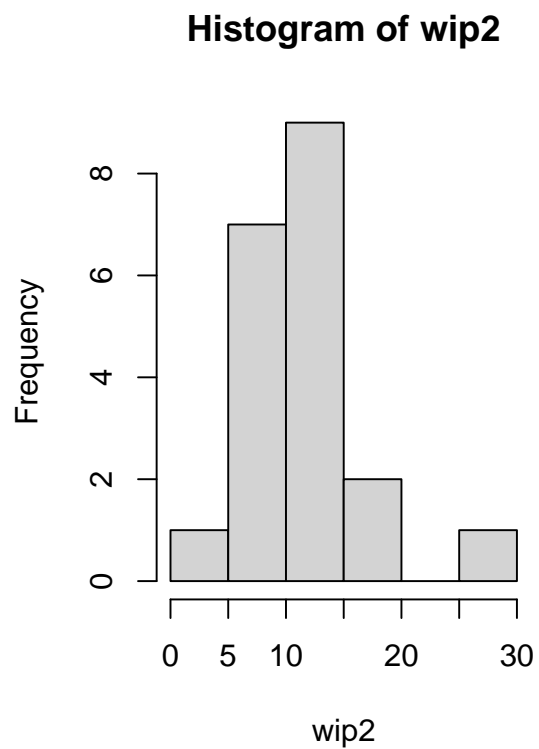
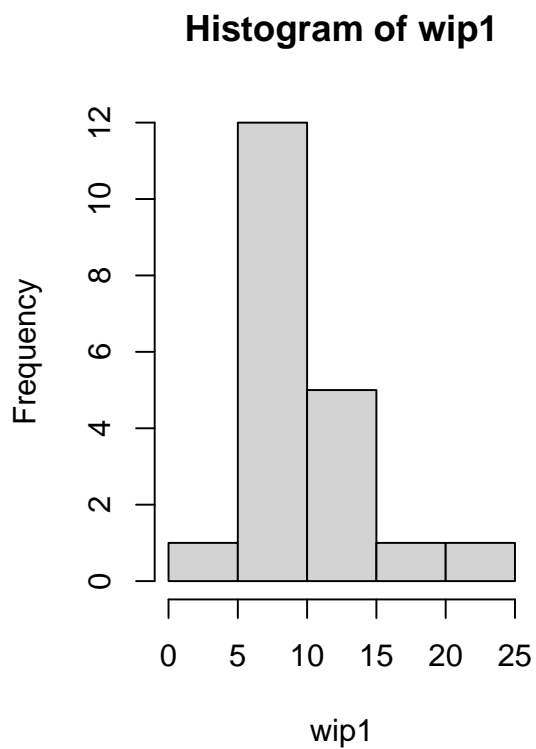
```
## [1] 26.27748
```

```
sd(wip1); sd(wip2)
```

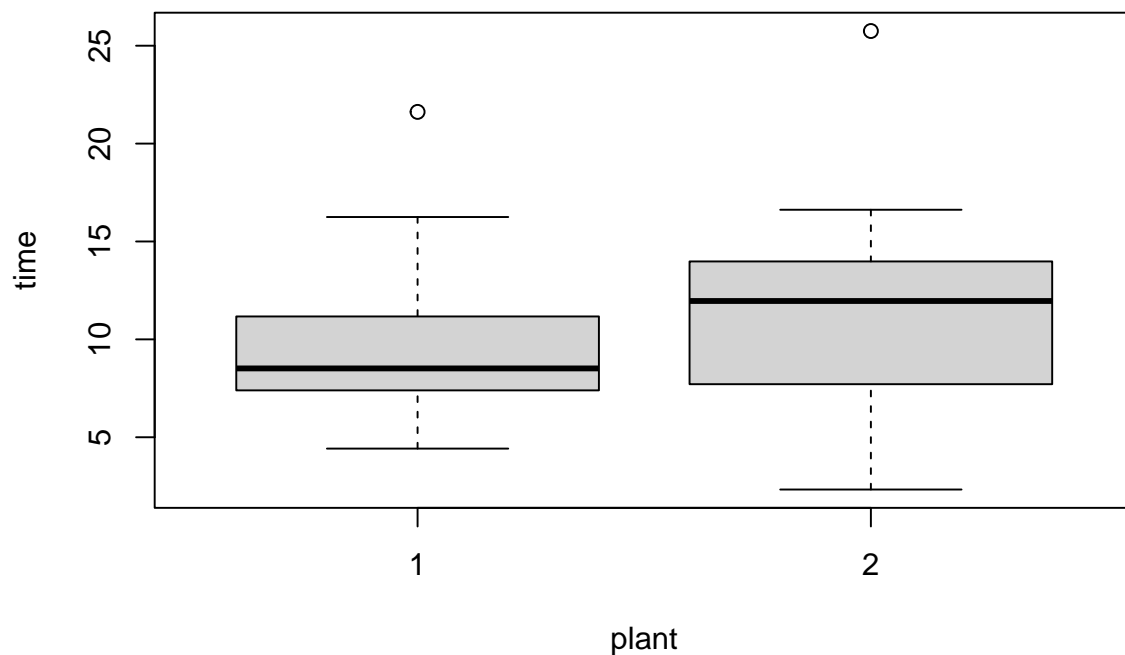
```
## [1] 3.997653
```

```
## [1] 5.126156
```

```
par(mfrow=c(1,2))  
hist(wip1)  
hist(wip2)
```



```
par(mfrow=c(1,1))  
boxplot(time~plant, wip)
```



The boxplot above is made in a single function using the tilde `~` operator. Here, in the `boxplot()` function, the left-hand side of the tilde is the variable we want to plot, the right-hand side of the tilde is what to group by. In this case, we plot `time`, grouping by `plant`.

Note that `wip` has *both* variables `time` and `plant` inside it.

Plant 2 has a higher median and a wider spread/distribution.

2 Exercise 3.2

```
babiesl <- read.table("Lab03babiesl.data", header=T)
babiesl
```

```
## # A tibble: 1,236 x 2
##       bwt smoke
##   <int> <int>
## 1   120     0
## 2   113     0
## 3   128     1
## 4   123     0
## 5   108     1
## 6   136     0
```

```
## 7 138 0
## 8 132 0
## 9 120 0
## 10 143 1
## # i 1,226 more rows
```

```
unique(babiesl$smoke)
```

```
## [1] 0 1 9
```

```
sum(babiesl$smoke == 9)
```

```
## [1] 10
```

```
babiesl.f <- babiesl[babiesl$smoke != 9,]
sum(babiesl.f$smoke == 0)
```

```
## [1] 742
```

```
sum(babiesl.f$smoke == 1)
```

```
## [1] 484
```

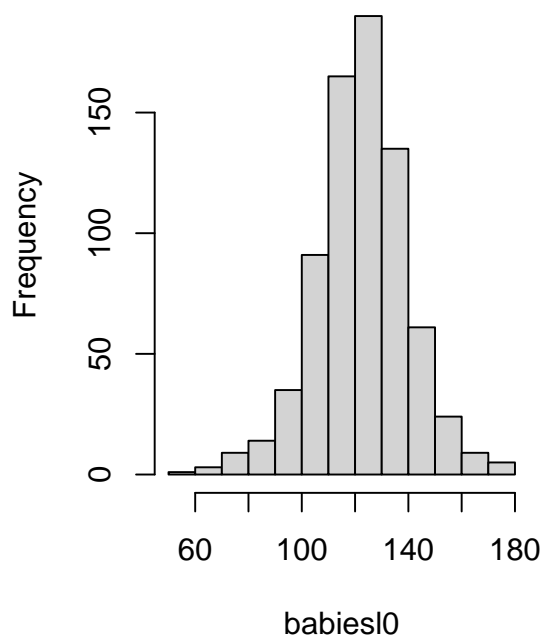
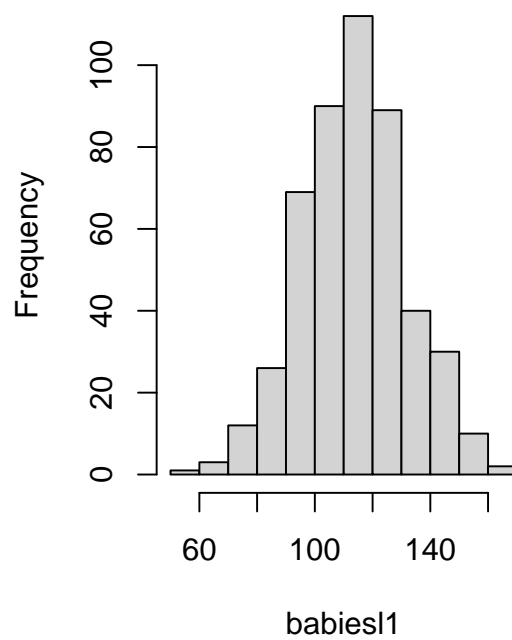
```
babiesl0 <- babiesl[babiesl$smoke==0, "bwt"]
babiesl1 <- babiesl[babiesl$smoke==1, "bwt"]
summary(babiesl0)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       55     113     123     123     134     176
```

```
summary(babiesl1)
```

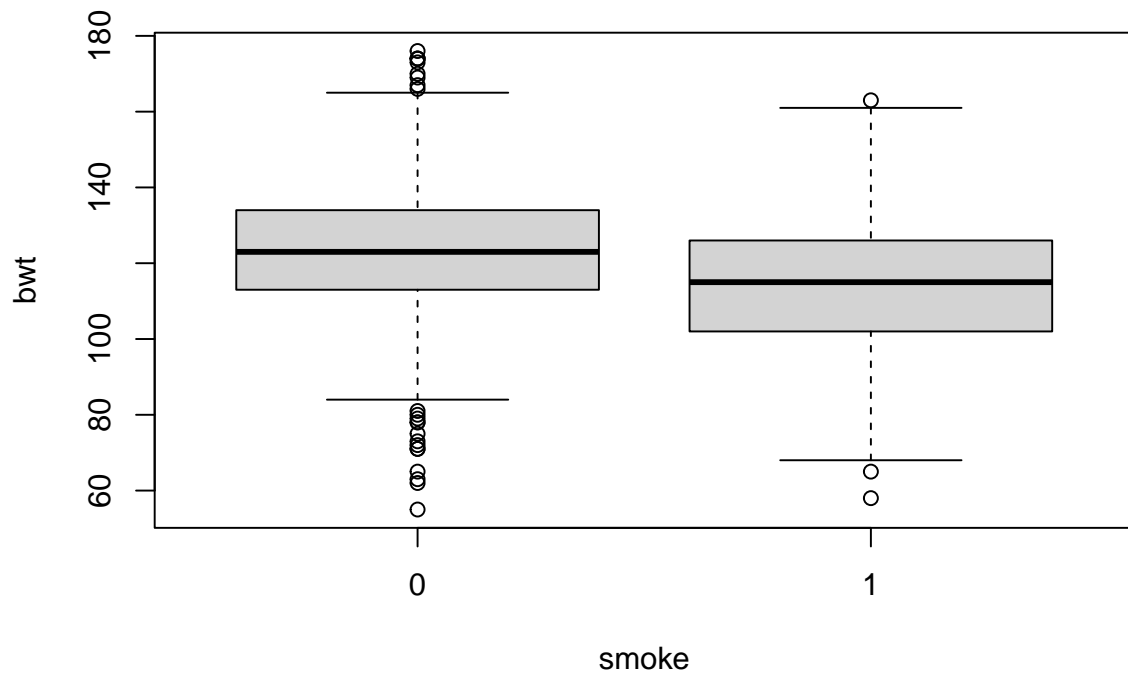
```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    58.0   102.0   115.0   114.1   126.0   163.0
```

```
par(mfrow=c(1,2))
hist(babiesl0)
hist(babiesl1)
```

Histogram of babiesl0**Histogram of babiesl1**

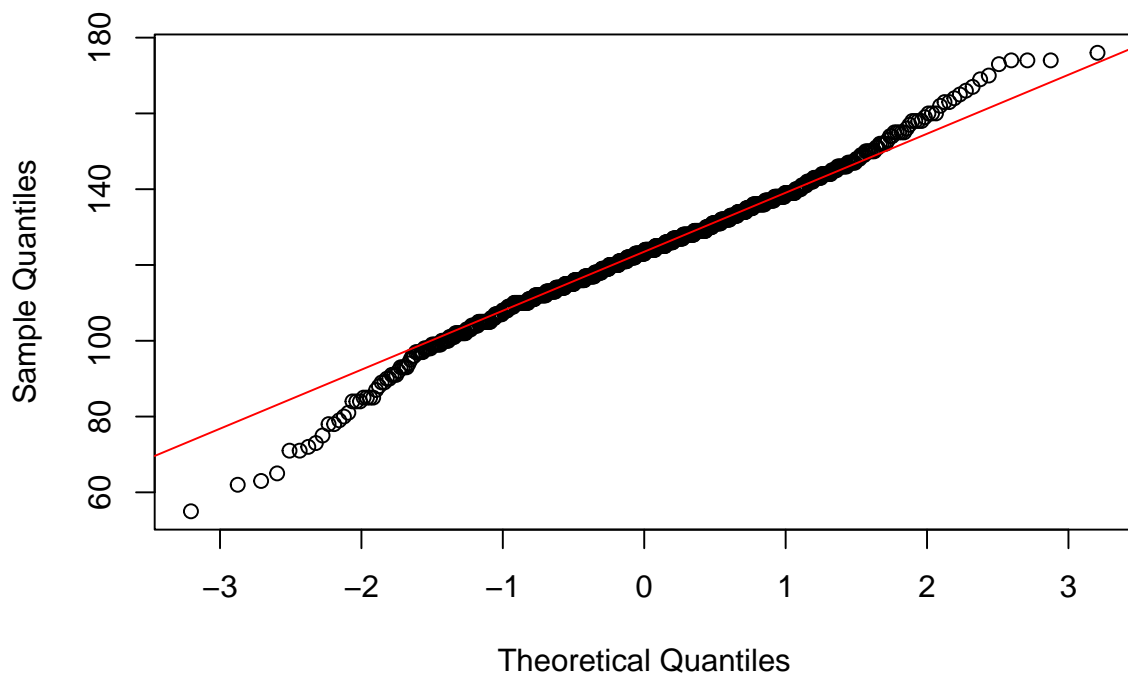
```
par(mfrow=c(1,1))
```

```
boxplot(bwt~smoke, babiesl.f)
```

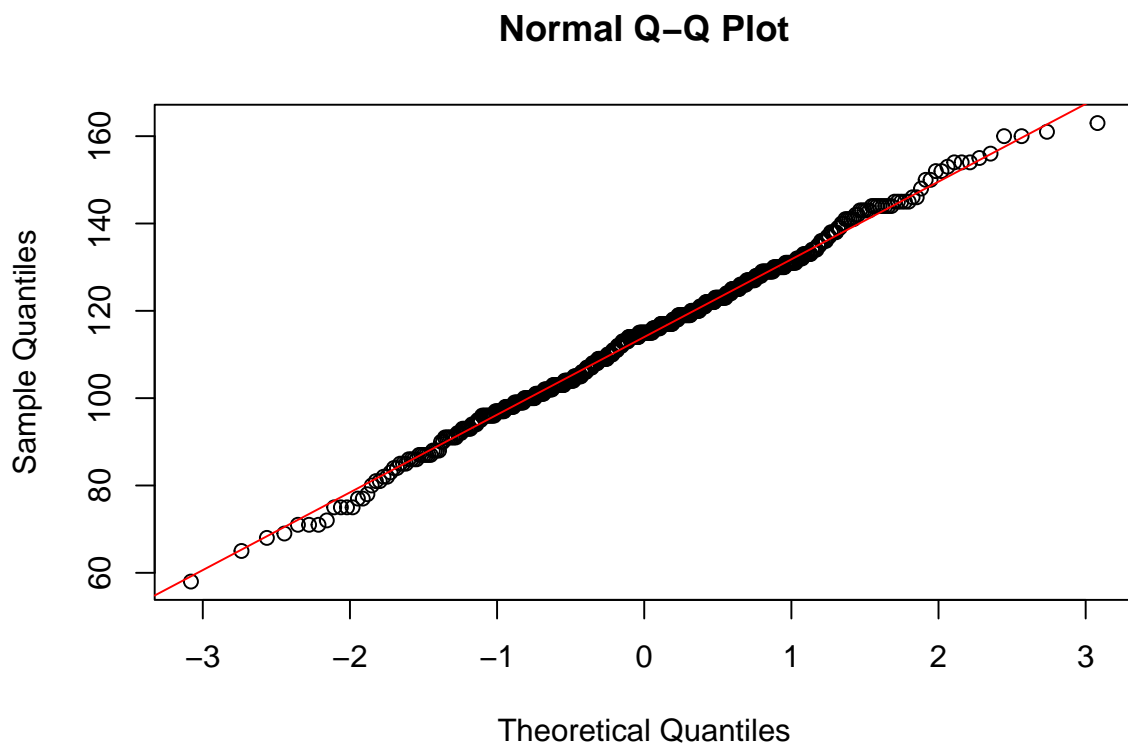


```
qqnorm(babiesl0); qqline(babiesl0, col="red")
```

Normal Q-Q Plot



```
qqnorm(babiesl1); qqline(babiesl1, col="red")
```



For the non-smoking group, while *most* of the data follows the line nicely, the tails of the data deviate greatly from the tails of the normal distribution. Thus, the data should not be taken to be normally distributed.

The smoking group, however, follows the normal distribution throughout its range, as seen by the close adherence to the line.

3 Exercise 3.3

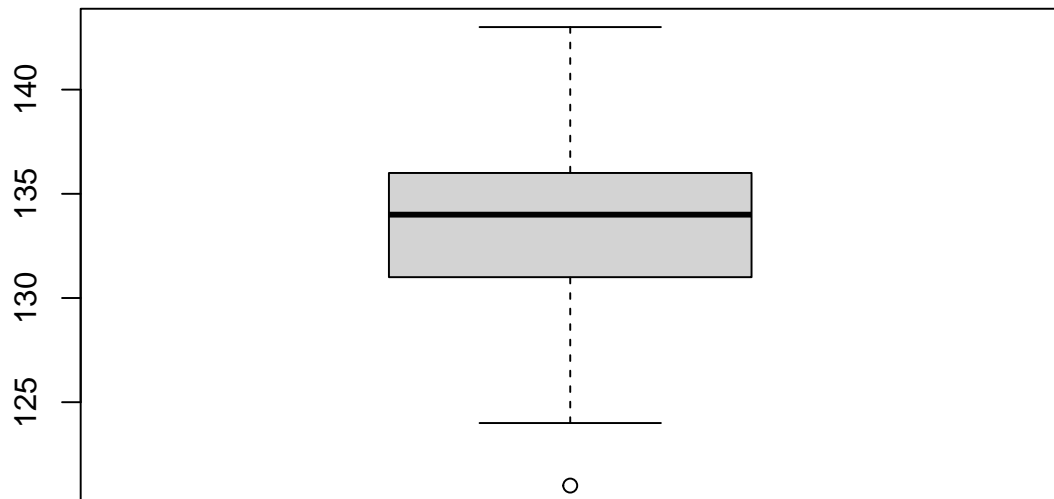
```
skull <- read.table("Lab03skull_height.txt", col.names="height")
skull <- skull$height
skull <- skull# + rnorm(length(skull))
sort(skull)
```

```
## [1] 121 124 129 129 130 130 131 131 132 132 132 133 133 134 134 134 134 135
## [20] 136 136 136 136 137 137 138 138 138 140 143
```

```
stem(skull)
```

```
##
##  The decimal point is at the |
##
##  120 | 0
##  122 |
##  124 | 0
##  126 |
##  128 | 00
##  130 | 0000
##  132 | 00000
##  134 | 000000
##  136 | 000000
##  138 | 000
##  140 | 0
##  142 | 0
```

```
boxplot(skull)
```



```
IQR(skull)
```

```
## [1] 4.75
```

```
quantile(skull, c(0.25, 0.75))
```

```
##      25%      75%  
## 131.25 136.00
```

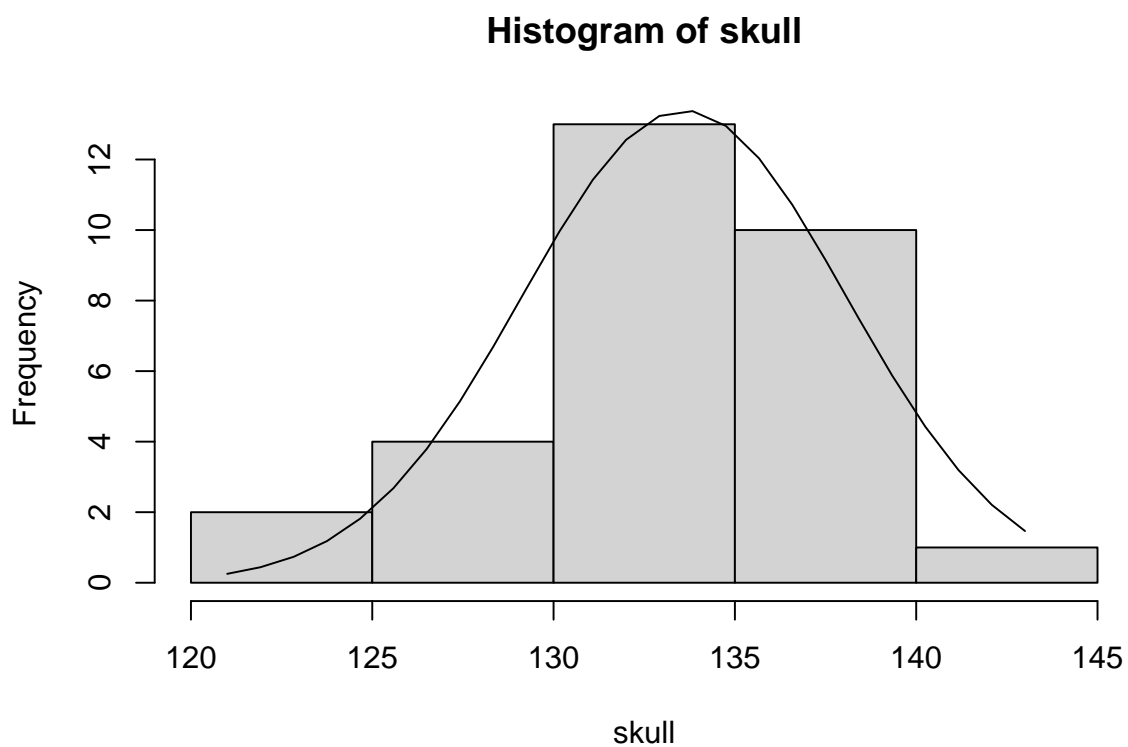
```
quantile(skull, 0.25, names=F) - 1.5*IQR(skull)
```

```
## [1] 124.125
```

```
quantile(skull, 0.25, names=F)
```

```
## [1] 131.25
```

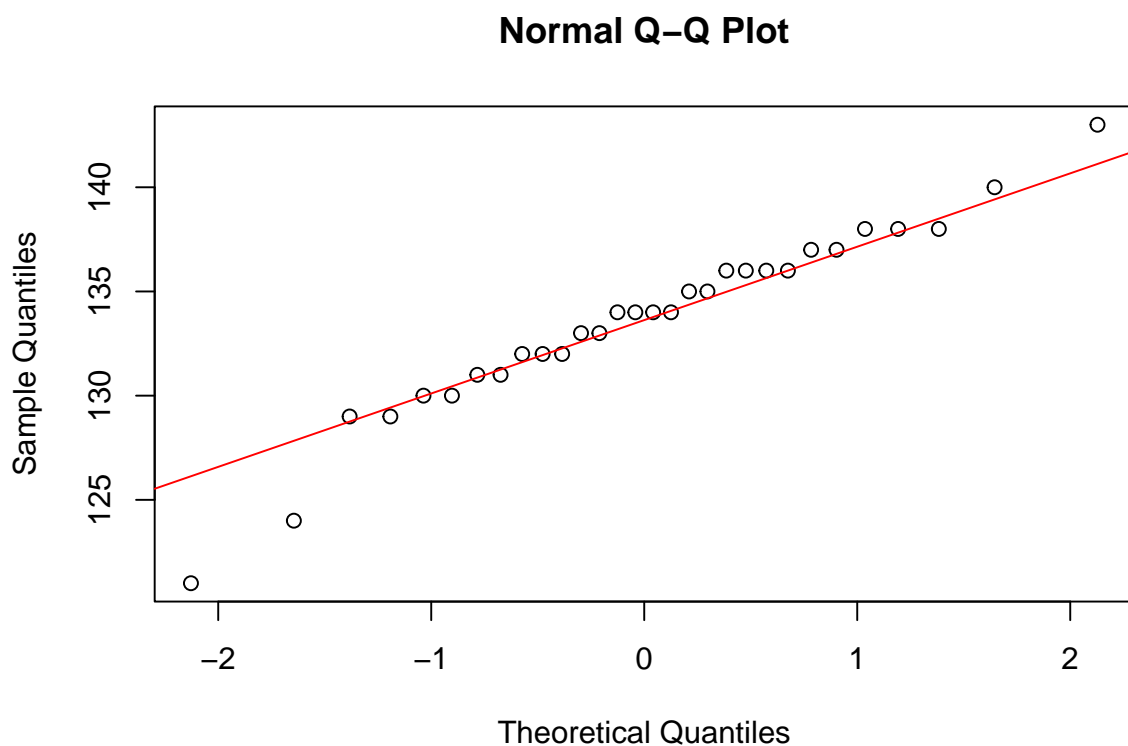
```
hist(skull)  
x <- seq(min(skull), max(skull), length.out=25)  
p <- dnorm(x, mean=mean(skull), sd=sd(skull))  
p <- p*(length(skull)*5) # bin width  
lines(x, p)
```



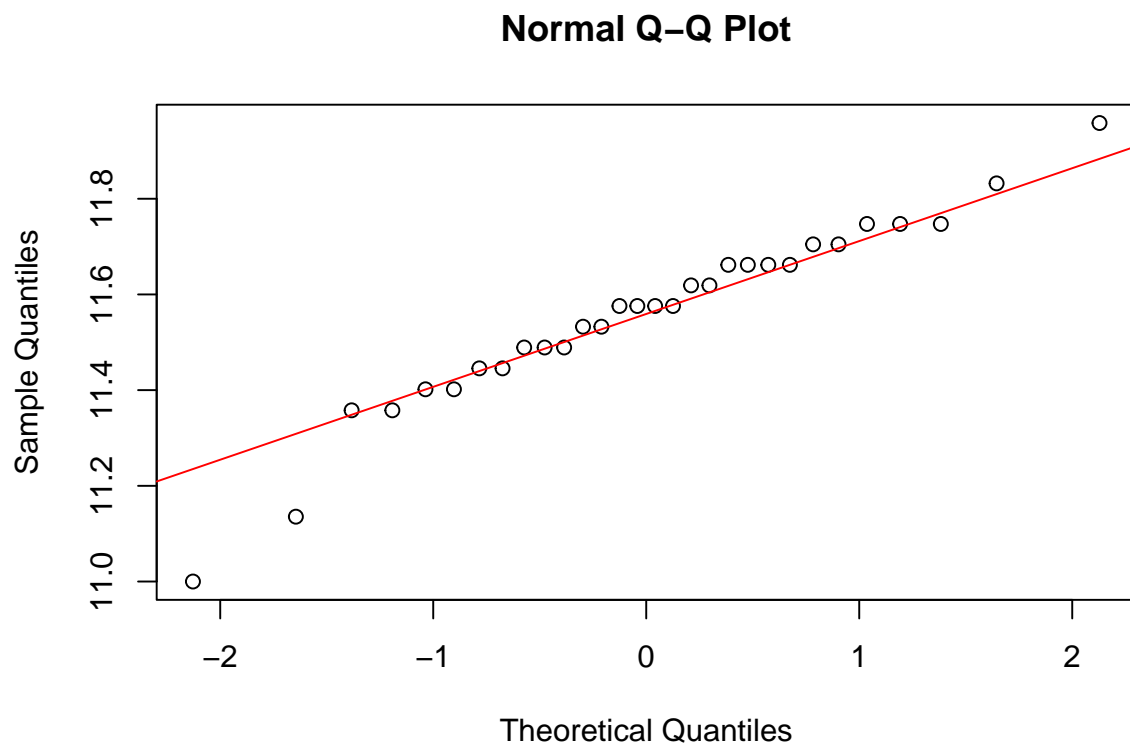
To impose the normal distribution on the histogram, we must create another vector containing the points to plot. So, we get the range of the points we want to plot (saved to x), and get the probability density function evaluated at those points (saved to p).

The values in p must be normalised to match the scale of the histogram. Since the area under the pdf integrates to 1, we multiply it by the area of the histogram, which, since each bin has width 5, each observation takes an area of 5.

```
# qqplot(qnorm(ppoints(30)), skull)  
qqnorm(skull)  
qqline(skull, col="red")
```



```
sqrtskull <- sqrt(skull)  
qqnorm(sqrtskull)  
qqline(sqrtskull, col="red")
```



4 Exercise 3.4

- IQR: $41.5 - 14 = 27.5$
- There is a clear right-skew given the maximum value of 2510 and third quartile of 41.5
- Based on the output of the summary function, we are unable to determine any other values. We cannot interpolate the values to estimate the 40th percentile. For example, we may have a dataset that looks like:

```
x <- c(0, rep(14, 3), 27, rep(41.5, 3), 2510)
```

```
x
```

```
## [1] 0.0 14.0 14.0 14.0 27.0 41.5 41.5 41.5 2510.0
```

```
summary(x)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      0.0   14.0   27.0   300.4   41.5   2510.0
```

For the 40th percentile, we can only bound it between the 25th percentile (1st quartile) and 50th percentile (median), and so: lower bound: 14, upper bound: 27

d) `sum(exec.pay > 100) / length(exec.pay)` or `mean(exec.pay > 100)`

```
x > 100
```

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

```
sum(x > 100) / length(x)
```

```
## [1] 0.1111111
```

```
mean(x > 100)
```

```
## [1] 0.1111111
```

e)

```
quantile.10 <- quantile(exec.pay, 0.1)
mean(exec.pay[exec.pay <= quantile.10])
```

```
quantile.10 <- quantile(x, 0.1)
quantile.10
```

```
## 10%
```

```
## 11.2
```

```
x <= quantile.10
```

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

```
mean(x[x <= quantile.10])
```

```
## [1] 0
```

5 Exercise 3.5

a) Understanding how the qqplots are constructed should help greatly in this question.

The tail distributions can be interpreted from the qq-plot. The qq-plot shows one distribution plotted against another. If the two distributions were exactly the same, they would line up as a straight line.

The tail distributions can be inferred from the spacing between each point at the ends of each axis.

Focusing on the top right corner of the qq-plot, the points are much more spread along the x-axis, compared to the y-axis. This suggests the right tail (the more positive values) are more spread out for the x-axis values, as compared to the y-axis values, i.e. X has a longer right tail than Y.

Likewise, in the bottom left corner of the qq-plot, the points are closer together along the x-axis compared to the y-axis. This suggests the left tail (the more negative values) are closer together for the x-axis values, as compared to the y-axis values, i.e. Y has a longer left tail than X.

- b) Given the 0.6-sample quantile for X is 2, we can draw a vertical line across at $x = 2$, finding it intersects the qq-plot at around $y = 0.5$. Thus the 0.6 sample quantile for Y is about 0.5