Contents

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
```

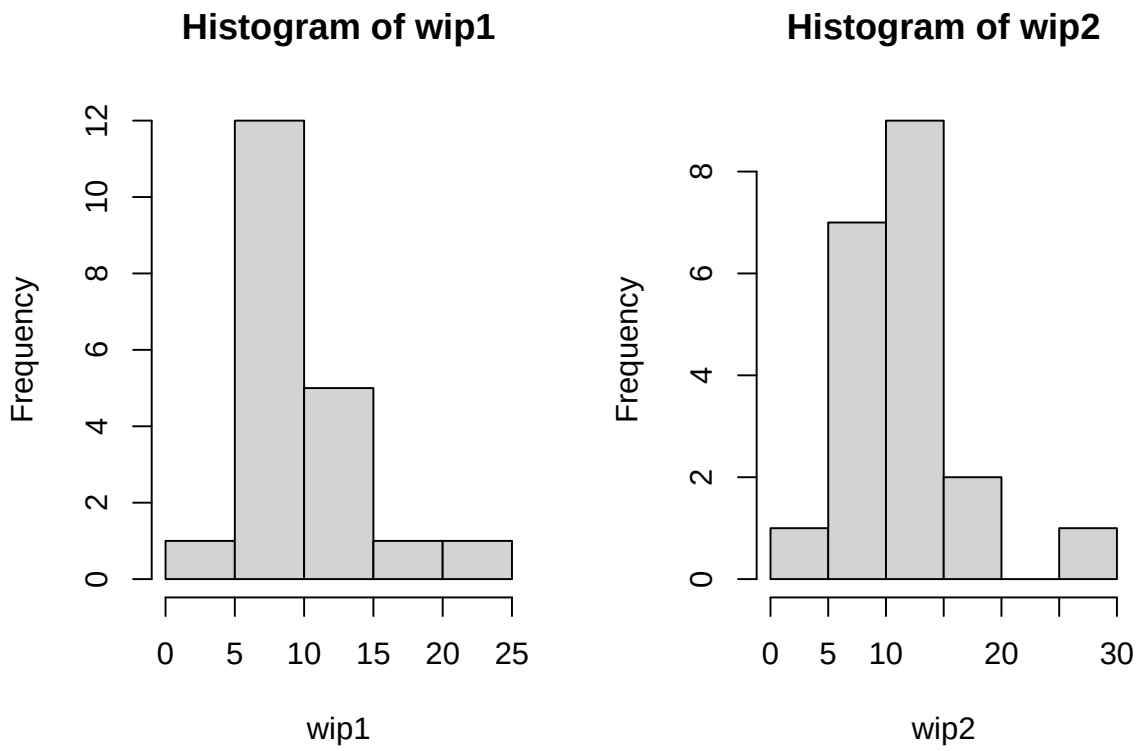
```
## [1] 26.27748
```

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

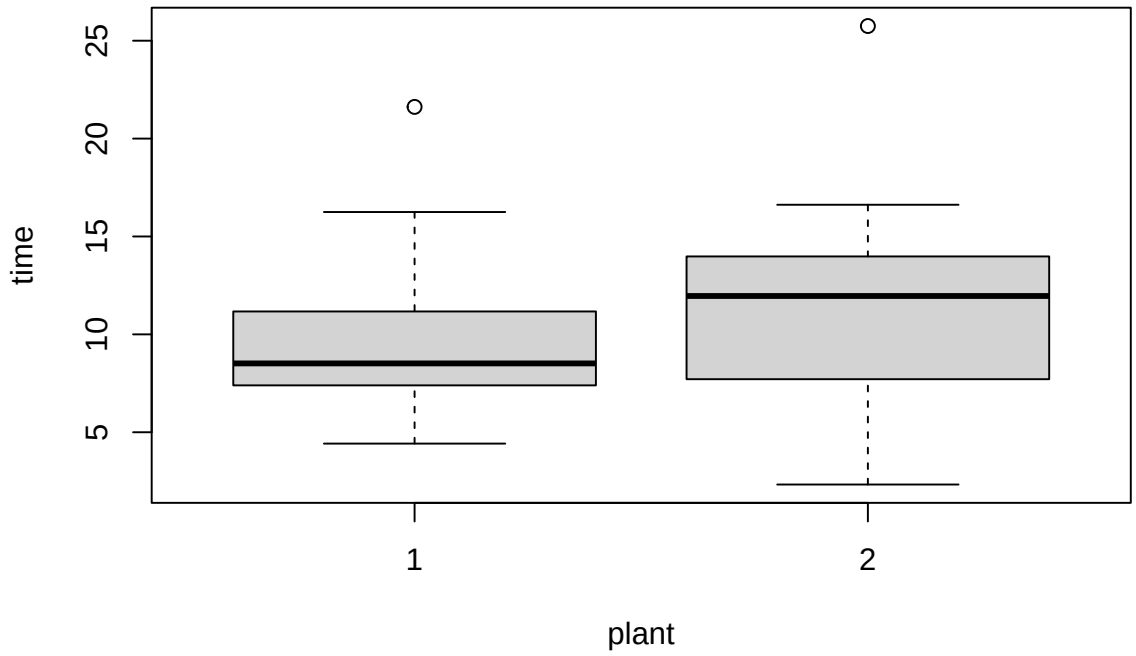
```
## [1] 3.997653
```

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

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




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



Plant 2 has a higher median and a wider IQR.

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
```

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

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

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

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

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

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

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

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

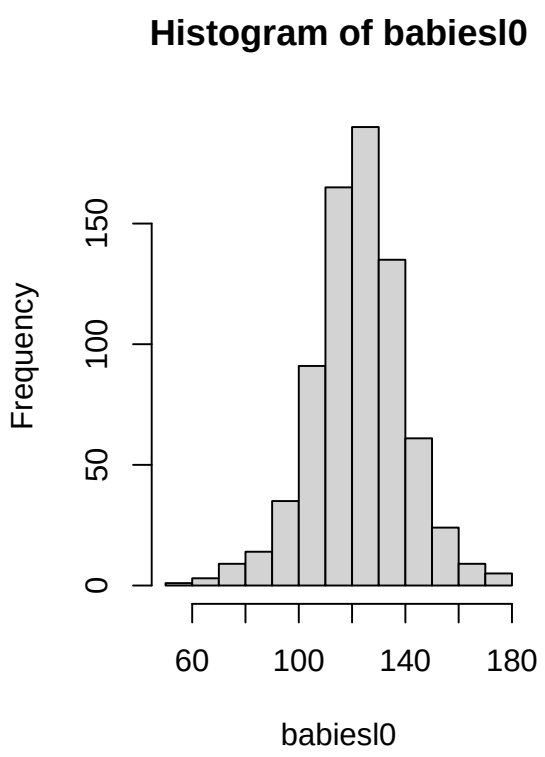
```r
babiesl0 <- babiesl[babiesl$smoke==0, "bwt"]
babiesl1 <- babiesl[babiesl$smoke==1, "bwt"]
summary(babiesl0)
```

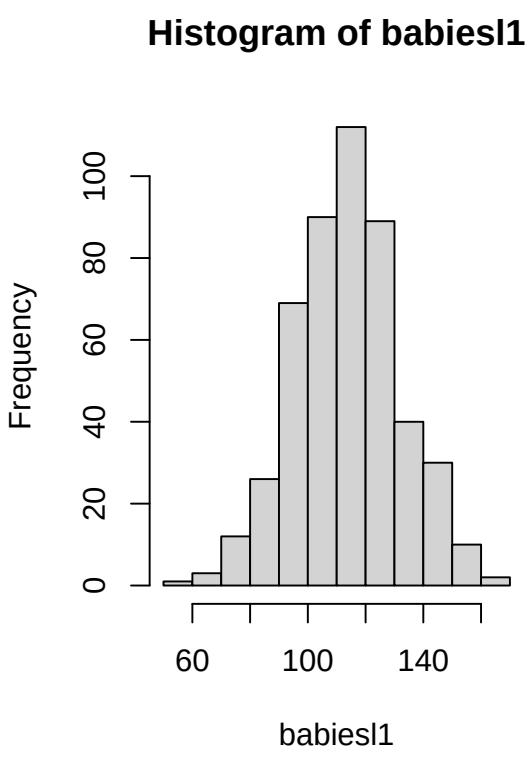
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      55     113     123     123     134     176
```

```r
summary(babiesl1)
```

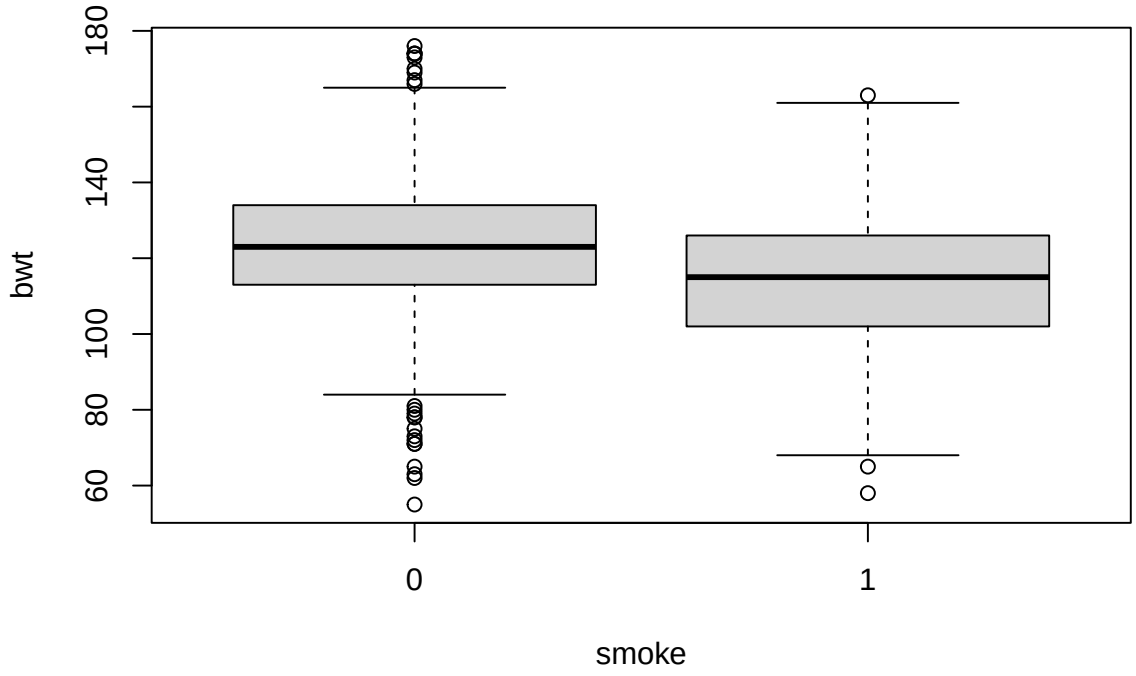
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    58.0   102.0   115.0   114.1   126.0   163.0
```

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

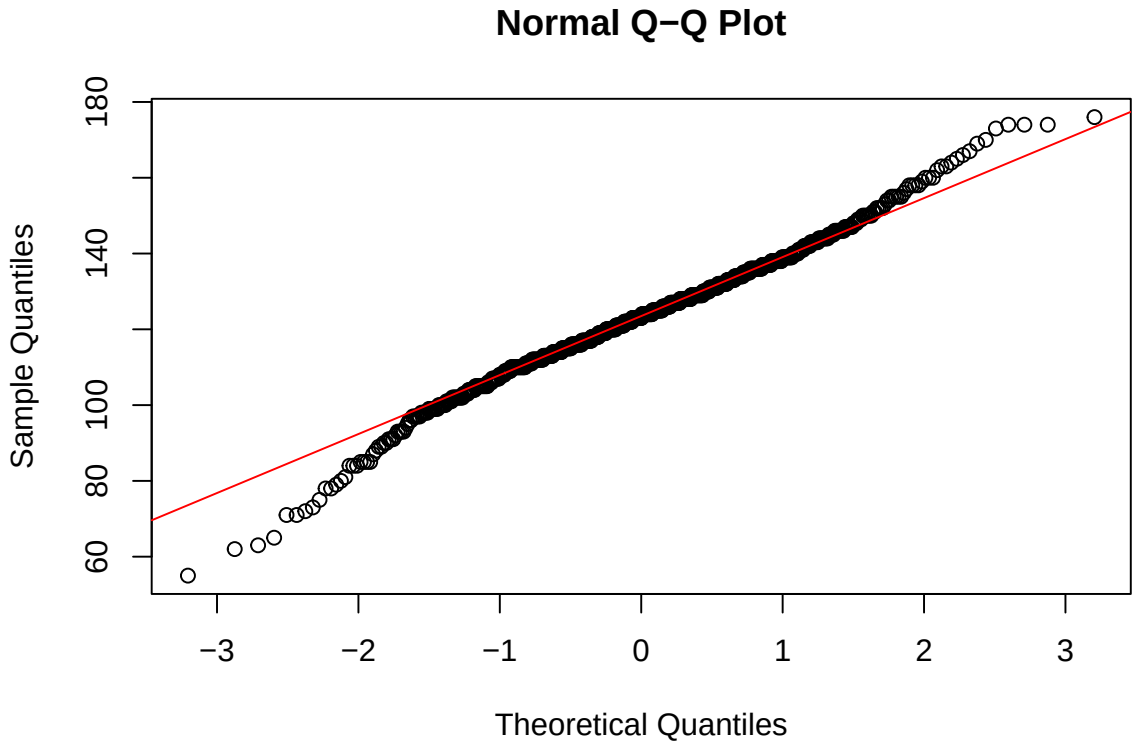




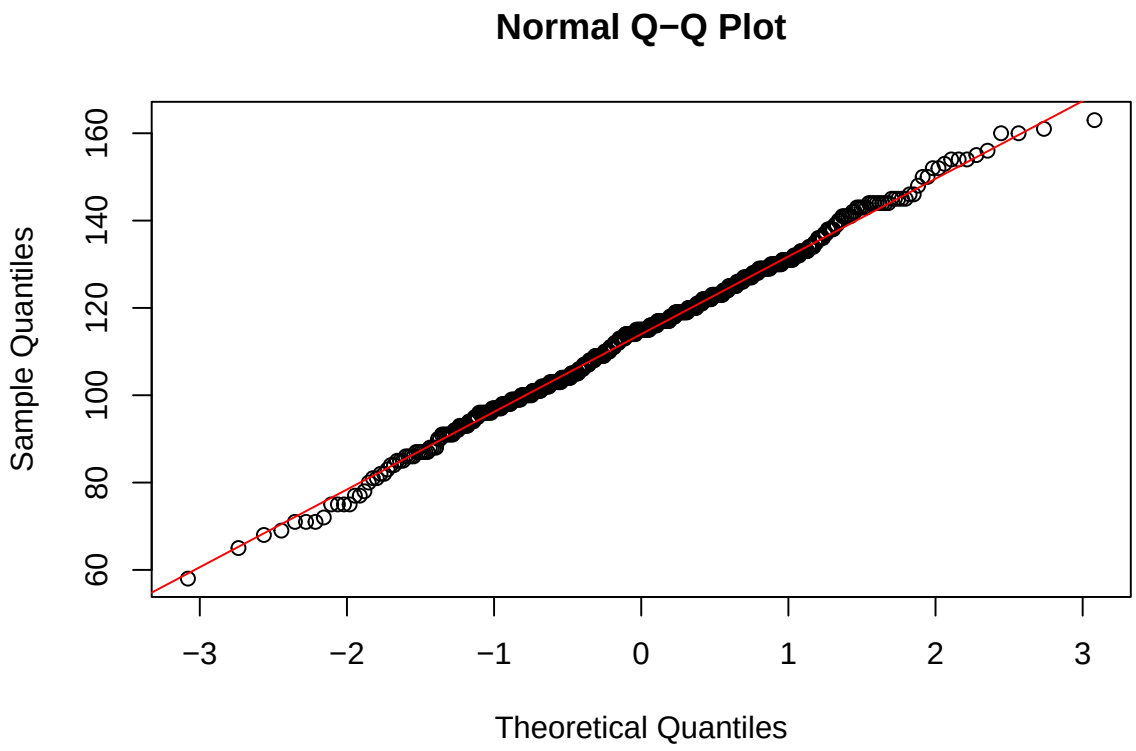

```r
par(mfrow=c(1,1))

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




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



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



As the q-q plot follows a nice straight line, the sample quantiles (data) follows the theoretical quantiles (normal distribution).

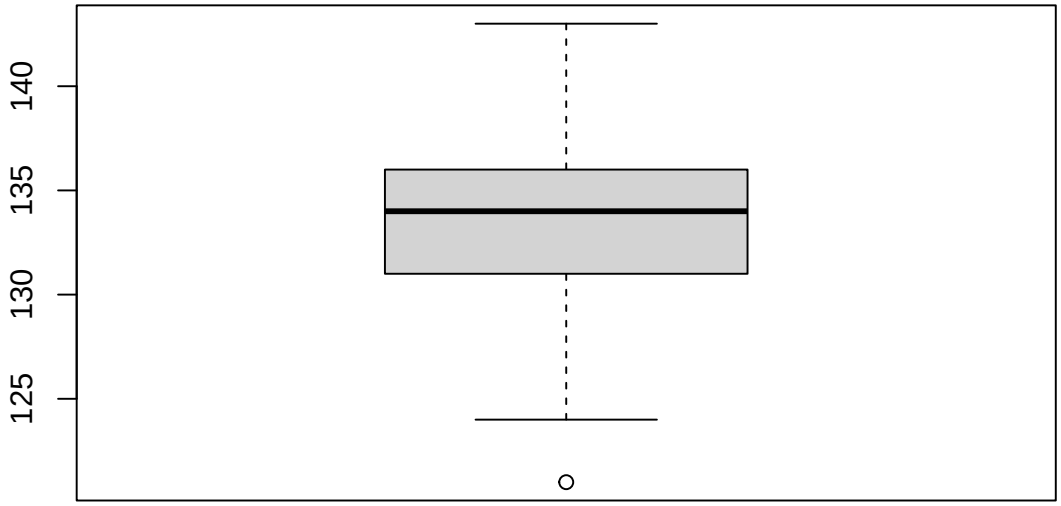
3 Exercise 3.3

```r
skull <- read.table("Lab03skull_height.txt", col.names="height")
skull <- skull$height
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
```

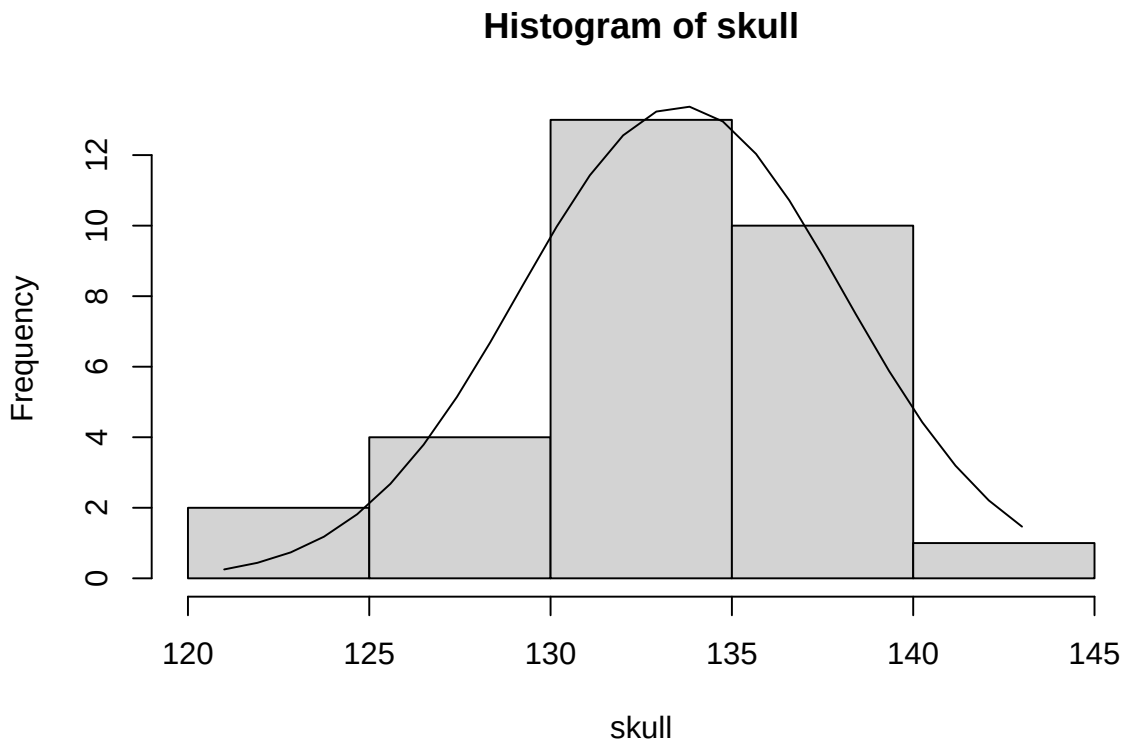
```
quantile(skull, 0.25) - 1.5*IQR(skull)
```

```
##      25%
## 124.125
```

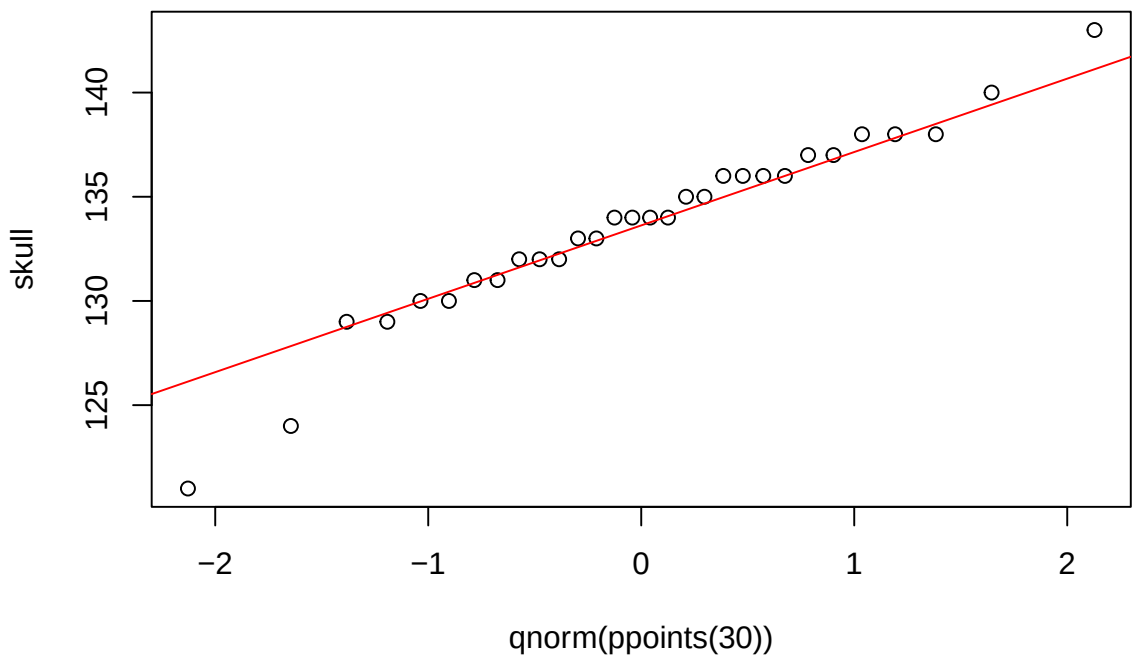
```
quantile(skull, 0.25, names=F)
```

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

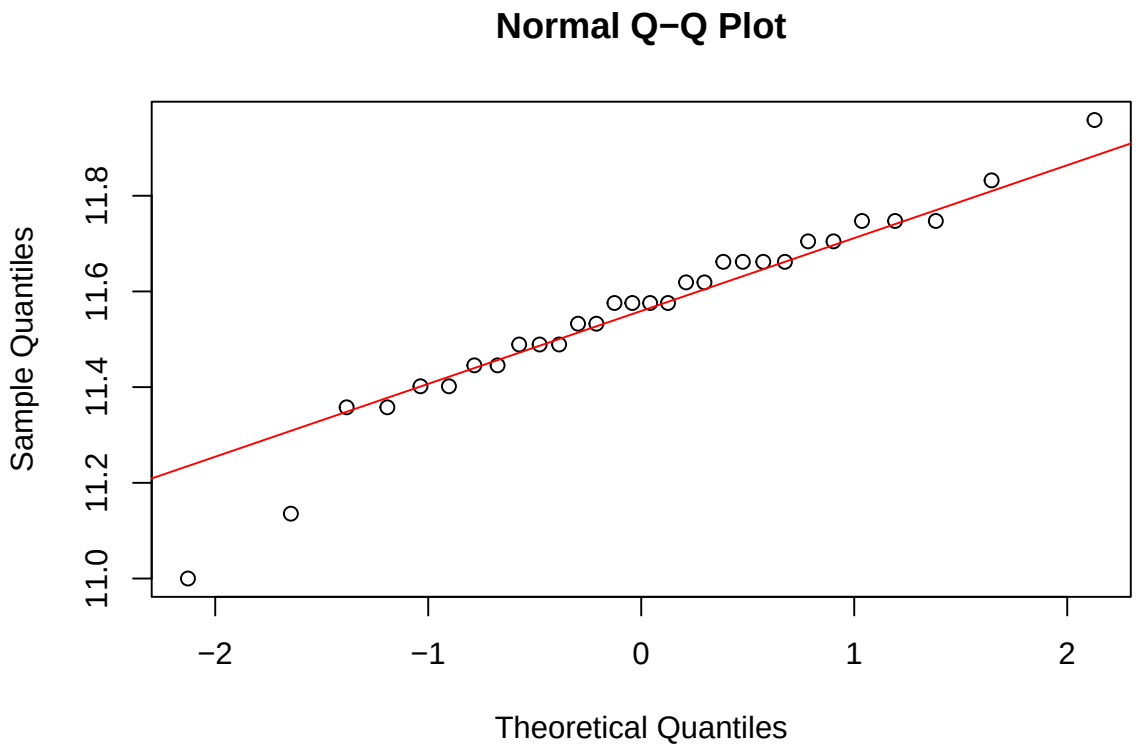
```r
hist(skull)
x <- seq(min(skull), max(skull), length.out=25)
p <- dnorm(x, mean=mean(skull), sd=sd(skull))
p <- p*length(skull)*5 # bin width
lines(x, p)
```




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




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



4 Exercise 3.4

a) IQR: 41.5 - 14 = 27.5

b) There is a clear right-skew given the maximum value of 2510 and third quartile of 41.5

c) 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) `exec.pay[exec.pay > 100] / length(exec.pay)`

e)

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

5 Exercise 3.5

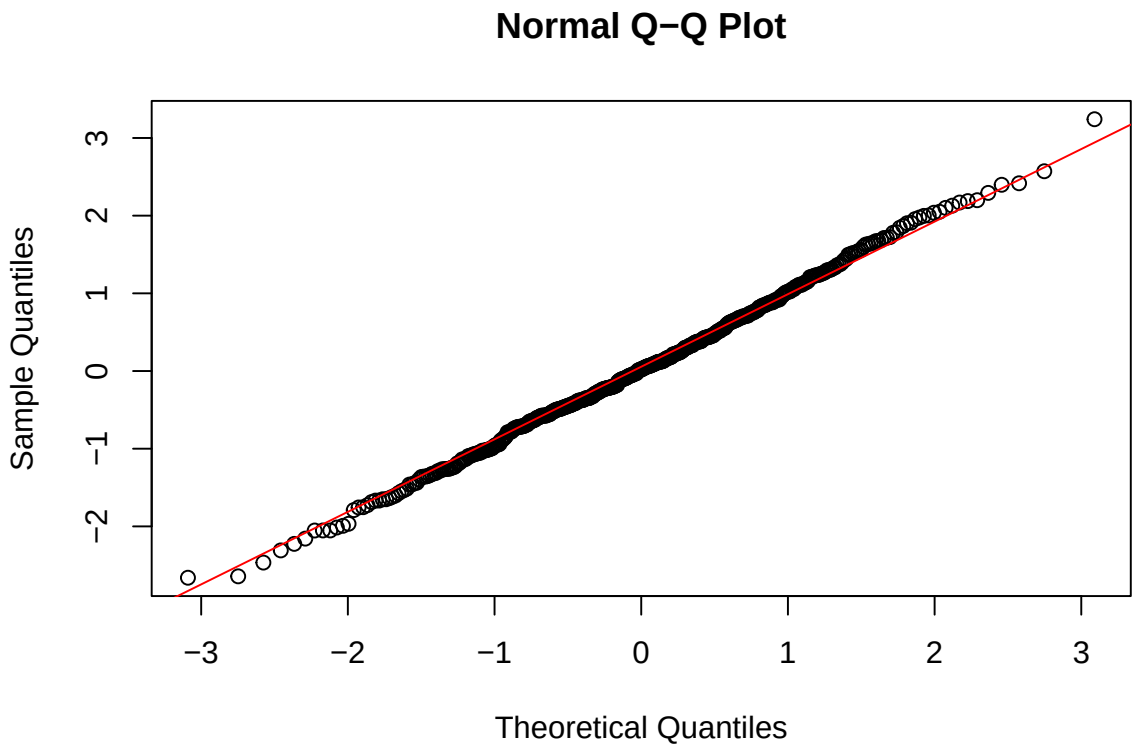
5.1 Quantile-quantile plots

It might be helpful to understand how quantile-quantile plots (qqplots) are created, and their purpose.

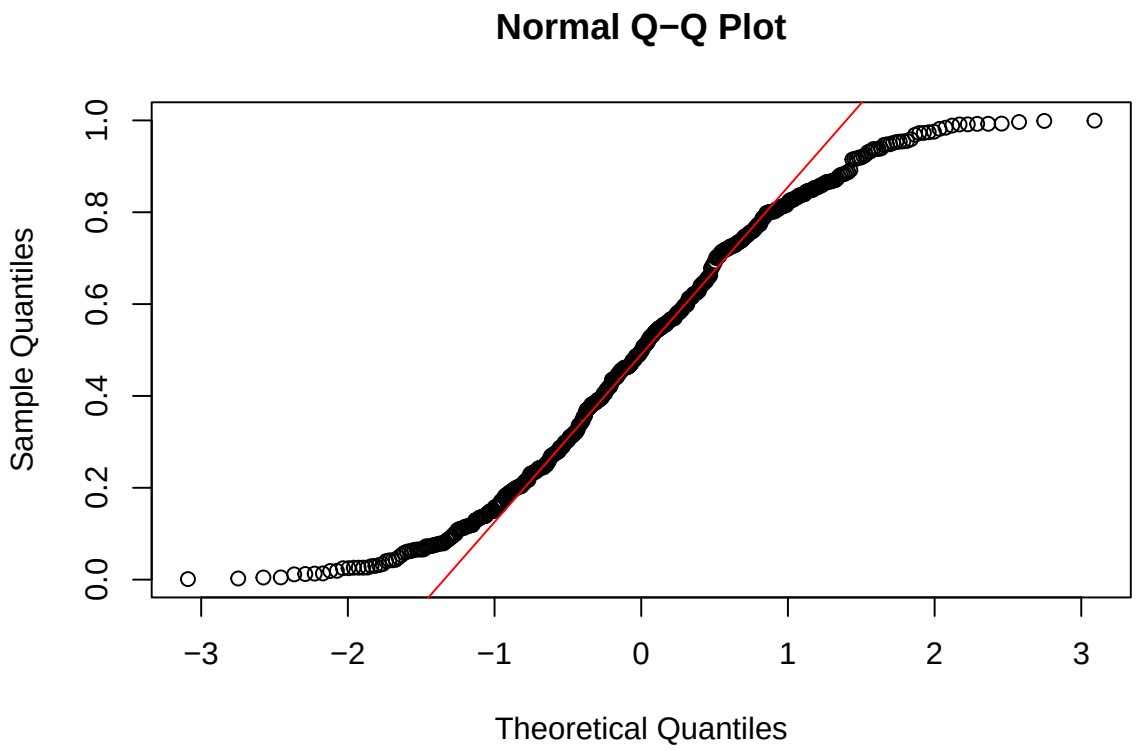
Generally, the purpose of a qqplot is to compare the distribution of two variables, *usually* with reference to a theoretical distribution. When a theoretical distribution is not available, one may rely on an empirical distribution (based on data).

An example: we have a set of data, which we want to check follows the normal distribution. The `qqnorm()` function helps us do that. It automatically arranges the data for plotting.

```
set.seed(123)
x.sample <- rnorm(500)
qqnorm(x.sample)
qqline(x.sample, col="red")
```



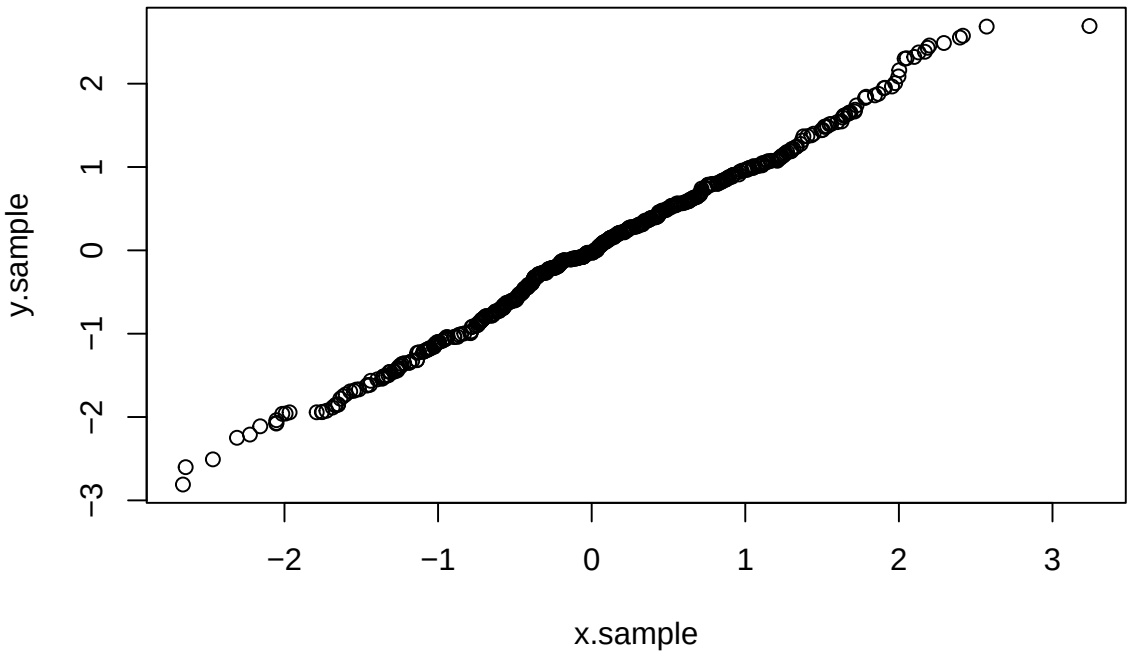
```r
x.sample <- runif(500)
qqnorm(x.sample)
qqline(x.sample, col="red")
```



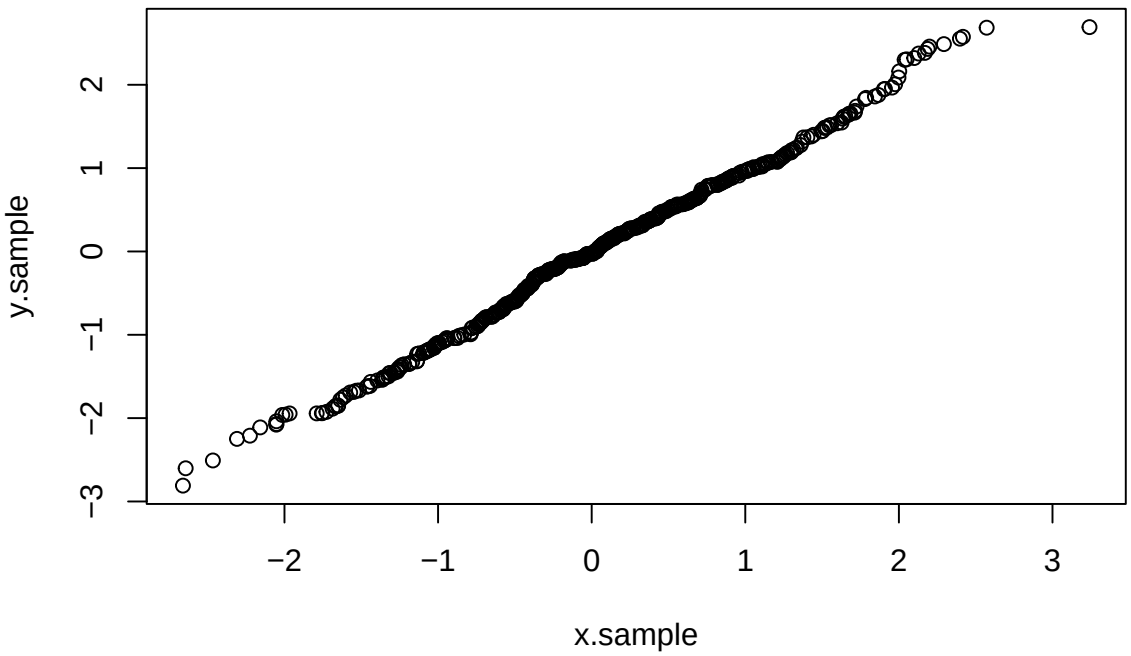
Second example: we have two sets of data, which we want to compare have the same distribution. Assume we don't have a theoretical distribution to compare it with. In this case, we use the `qqplot()` function. The base `plot()` function also works. The data must be arranged in ascending order, here it is done with `sort()`.

`qqline()` is harder to use in this case, as it requires a distribution function. The qq-line just draws a line between the 25th and 75th quantiles, though, so it can be implemented.

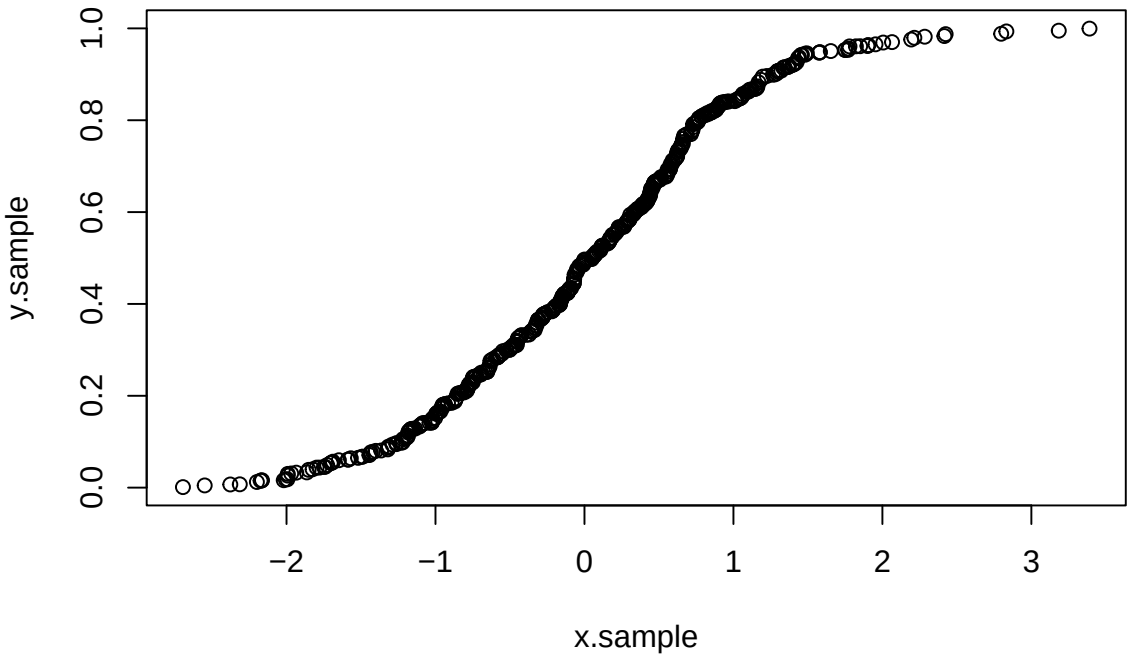
```r
set.seed(123)
x.sample <- sort(rnorm(500))
y.sample <- sort(rnorm(500))
plot(x.sample, y.sample)
```



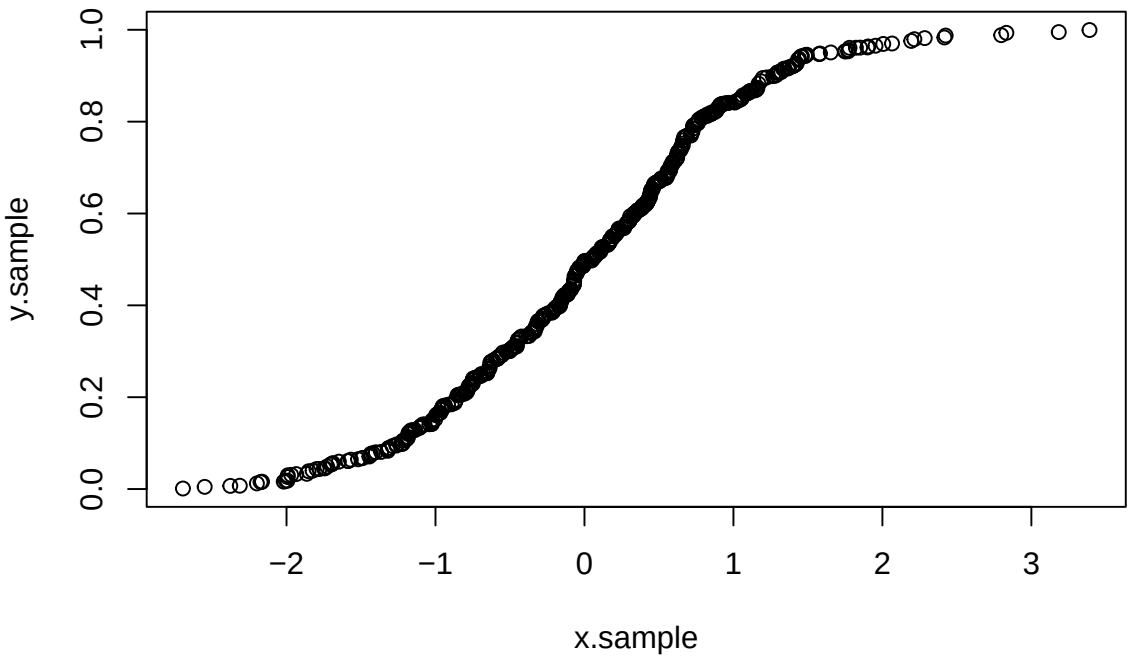

```
qqplot(x.sample, y.sample)
```



```r
x.sample <- sort(rnorm(500))
y.sample <- sort(runif(500))
plot(x.sample, y.sample)
```



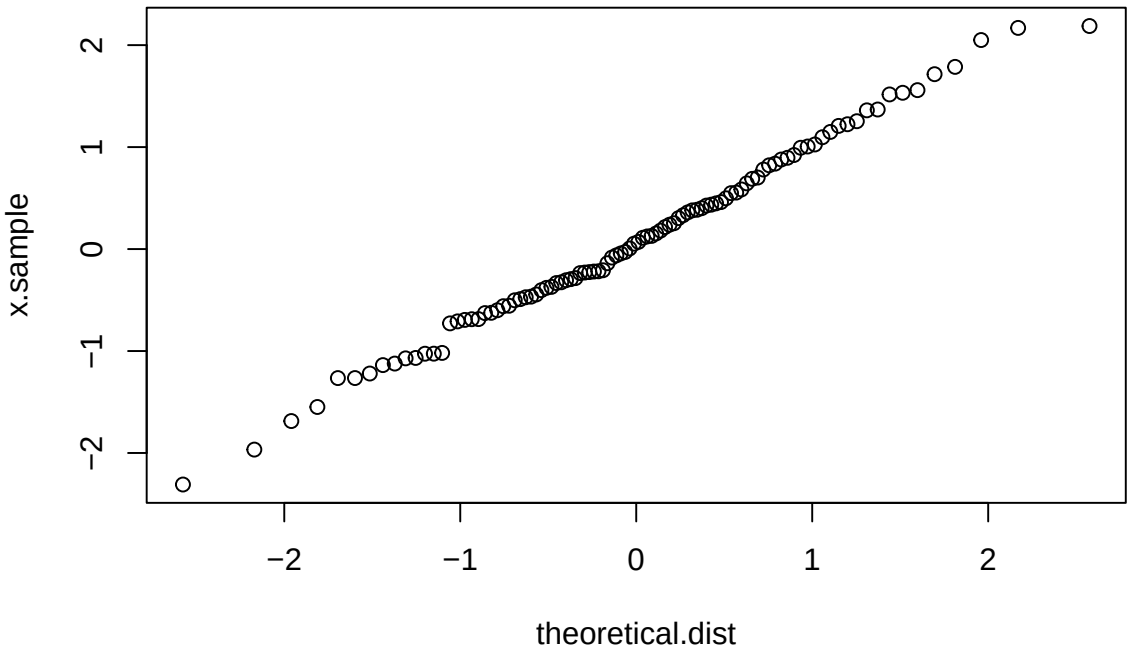

```r
qqplot(x.sample, y.sample)
```



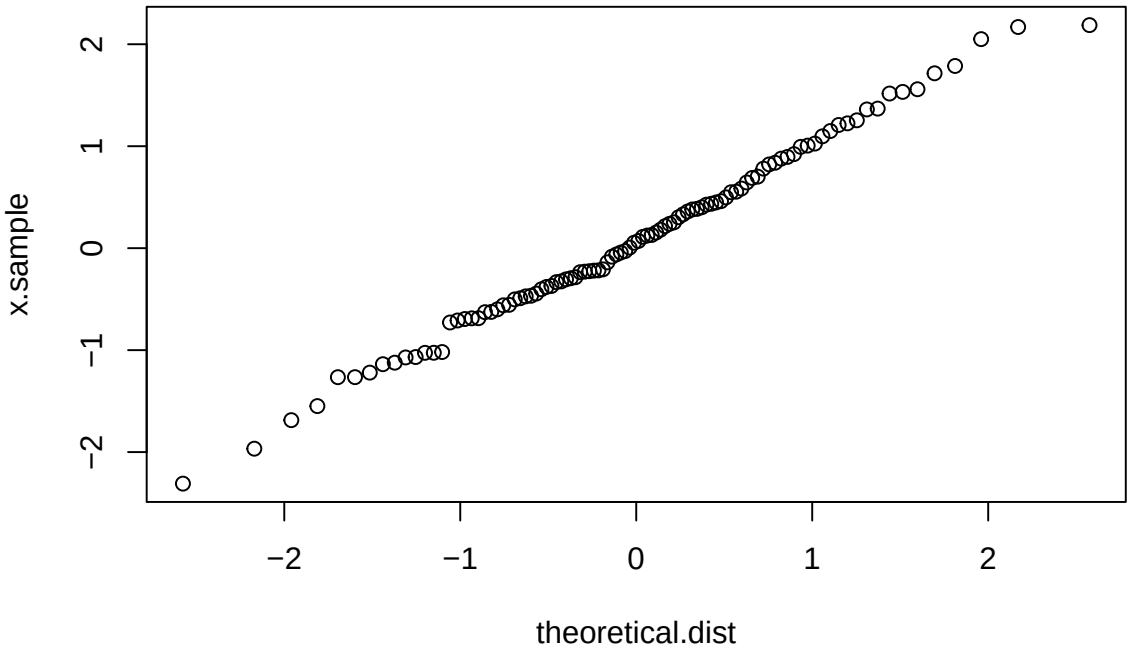
5.2 Optional: theoretical distribution without `qqplot()`/`qqnorm()`

We need to get the quantiles of the theoretical distribution. Thus, we space out numbers evenly in the interval $(0, 1)$, acting as the probabilities. Importantly, we avoid $p = 0$ and $p = 1$ to avoid extremities, e.g. `qnorm(0) = -Inf` and `qnorm(1) = +Inf`.

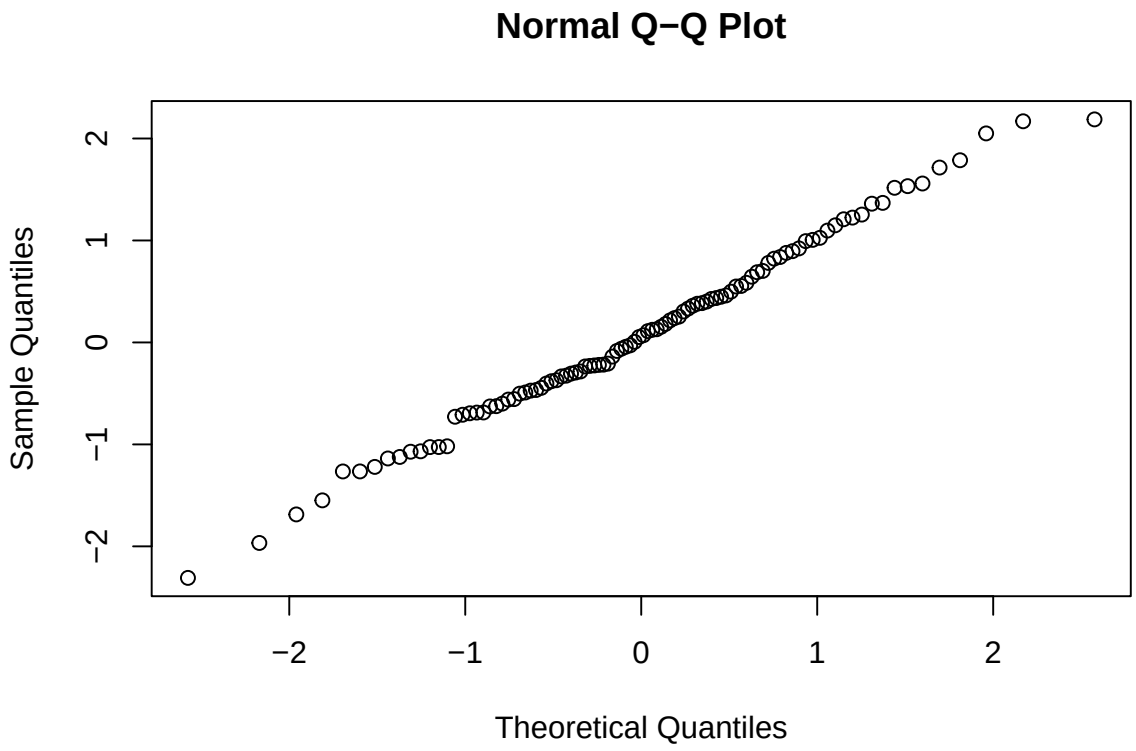
```r
set.seed(123)
x.sample <- sort(rnorm(100))
p <- ppoints(100)
theoretical.dist <- qnorm(p)

# All three create the same plot.
plot(theoretical.dist, x.sample)
```




```r
qqplot(theoretical.dist, x.sample)
```



```r
qqnorm(x.sample)
```



5.3 Back to the exercise

a) 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 on 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.

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.

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 $y = 0.5$.