STAT 210
Applied Statistics and Data Analysis
Problem List 2 - Solution
(due on week 3)

Fall 2022

Exercise 1

This is an exercise on the use of `plot` and its arguments.

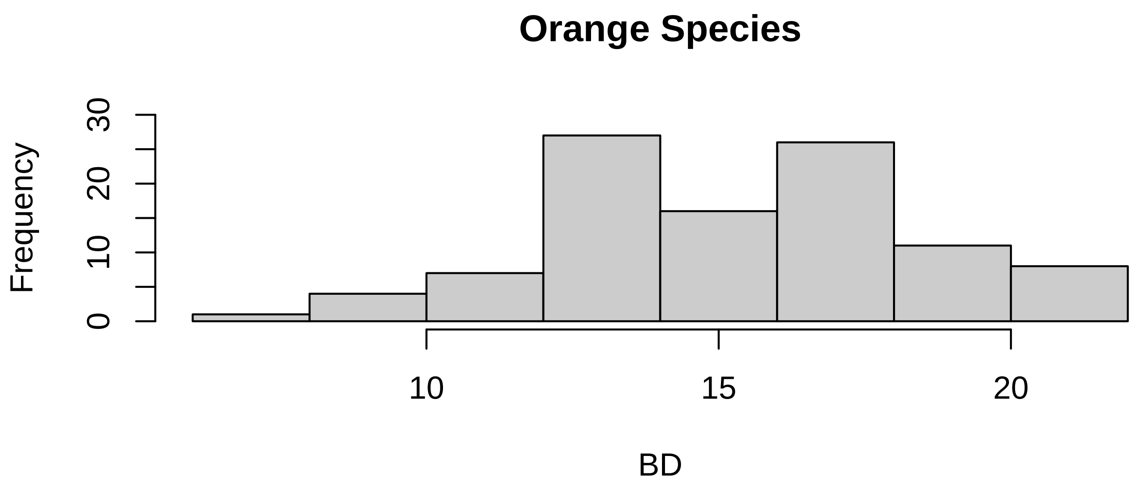
1. Load library MASS and use `str` to explore the structure of the set `crabs`.

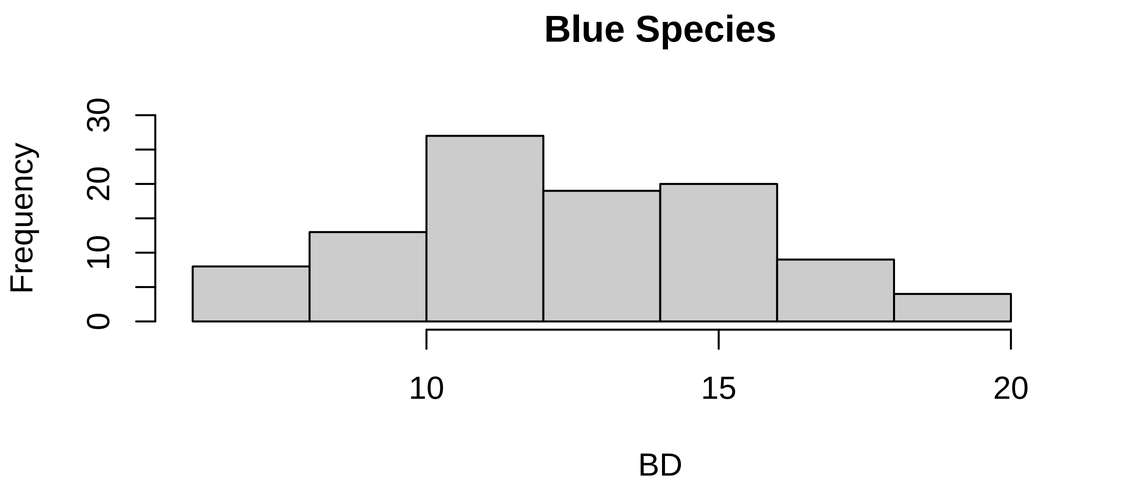
```
library(MASS)
data(crabs)
str(crabs)
```

```
## 'data.frame':    200 obs. of  8 variables:
##  $ sp   : Factor w/ 2 levels "B","O": 1 1 1 1 1 1 1 1 1 1 ...
##  $ sex  : Factor w/ 2 levels "F","M": 2 2 2 2 2 2 2 2 2 2 ...
##  $ index: int  1 2 3 4 5 6 7 8 9 10 ...
##  $ FL   : num  8.1 8.8 9.2 9.6 9.8 10.8 11.1 11.6 11.8 11.8 ...
##  $ RW   : num  6.7 7.7 7.8 7.9 8 9 9.9 9.1 9.6 10.5 ...
##  $ CL   : num  16.1 18.1 19 20.1 20.3 23 23.8 24.5 24.2 25.2 ...
##  $ CW   : num  19 20.8 22.4 23.1 23 26.5 27.1 28.4 27.8 29.3 ...
##  $ BD   : num  7 7.4 7.7 8.2 8.2 9.8 9.8 10.4 9.7 10.3 ...
```

2. Divide the plotting window in two rows using the function `par` with argument `mfrow`. In the first row draw a histogram of `BD` for the orange species and in the second row draw a histogram of the same variable for the blue species. Use appropriate titles for the plots and for the axes. Make sure to use the same scale in both axes for both plots. What do you see in these graphs?

```
par(mfrow = c(2,1))
hist(crabs$BD[crabs$sp == 'O'], col='gray80',
     xlim = c(6,22), ylim = c(0,30),
     xlab = 'BD', main = 'Orange Species')
hist(crabs$BD[crabs$sp == 'B'], col='gray80',
     xlim = c(6,22), ylim = c(0,30),
     xlab = 'BD', main = 'Blue Species')
```

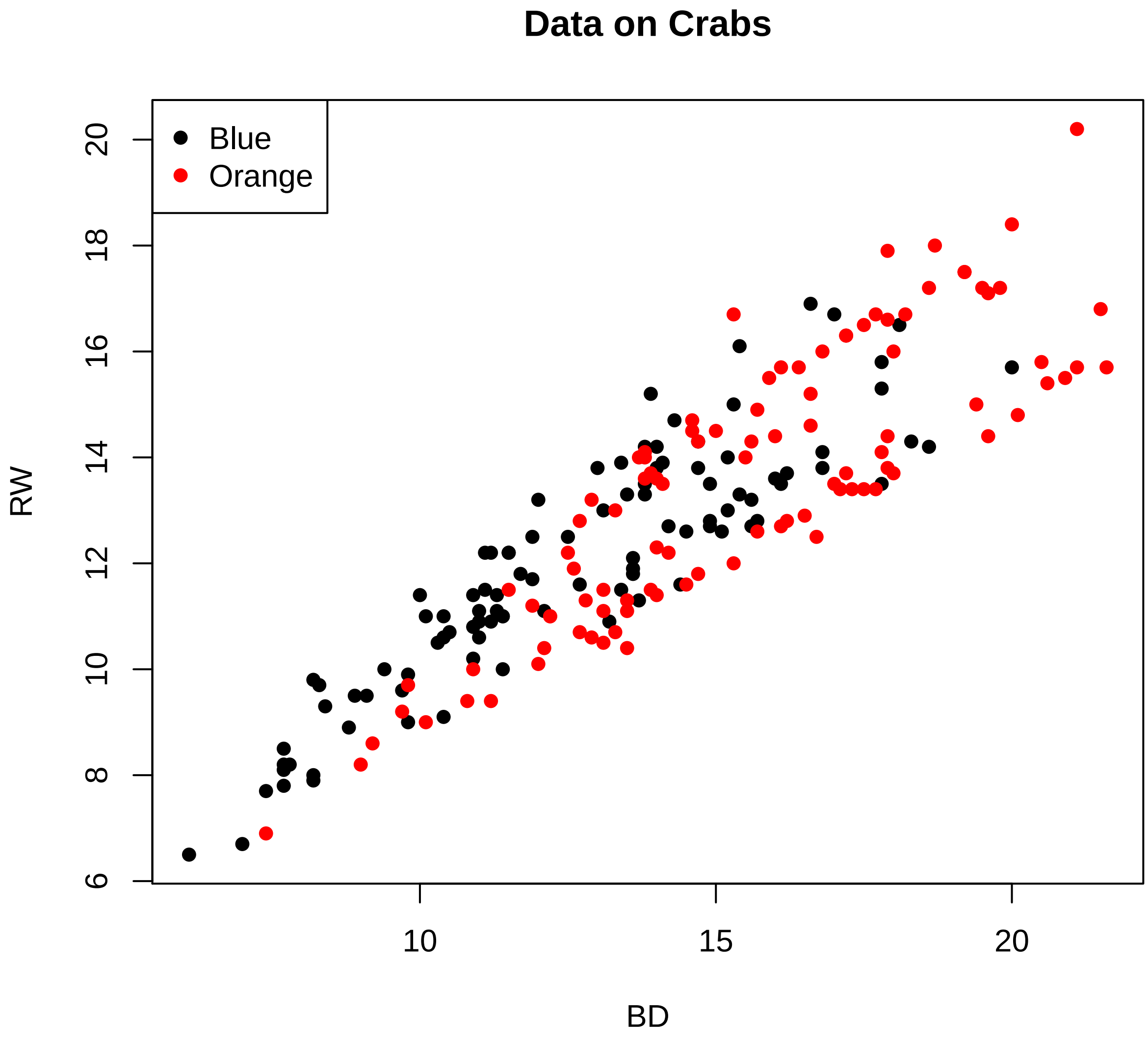




We see that the values for the blue species are lower than those for the orange. Both distributions look symmetric and have similar ranges.

3. Plot a graph of `RW` against `BD`. Include as title 'Data on Crabs' by using `main`. Use a solid dot as plotting symbol and use species (`sp`) to determine the color of the points. Add a legend on the upper left corner. Can you conclude anything from this graph?

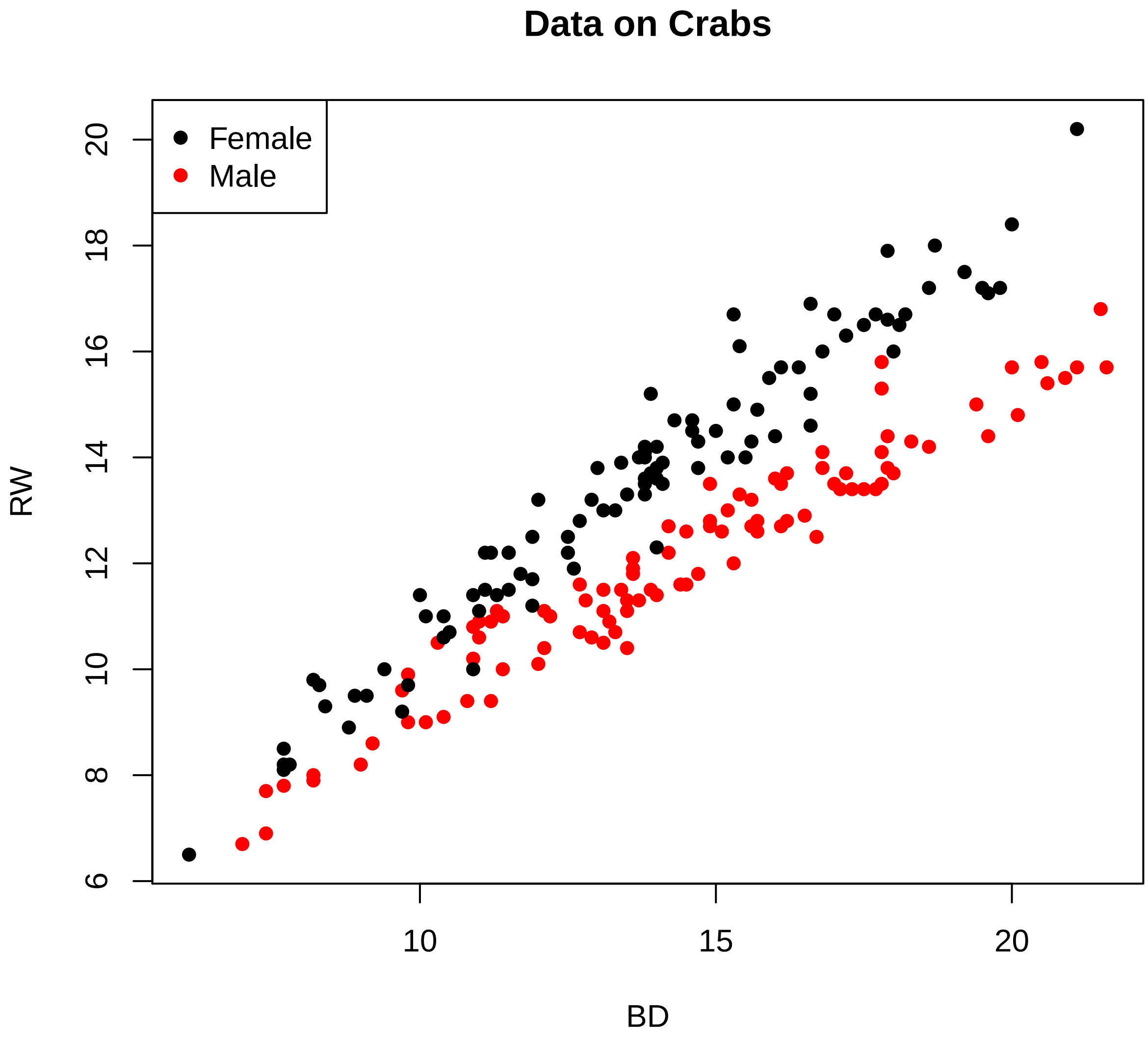
```
plot(RW ~ BD, data = crabs, main = 'Data on Crabs', pch = 16,
     col = sp)
legend('topleft',c('Blue','Orange'), pch = rep(16,2),
       col = 1:2)
```



It is difficult to conclude anything from this graph except that there are more black points in the left half of the plot and more red ones in the right half. This means that the orange species crabs have these two dimension, body depth and rear width larger than the blue crabs. Also, there is clearly a relation between these two variables: as BD increases, so does RW, but the graph looks like a funnel opening up for larger values of the variables.

4. Plot a graph of `RW` against `BD`. Include as title 'Data on Crabs' in the plot by using `main`. Use a solid dot as plotting symbol. Use `sex` to determine the color of the points. Add a legend on the upper left corner. Can you conclude anything from this graph?

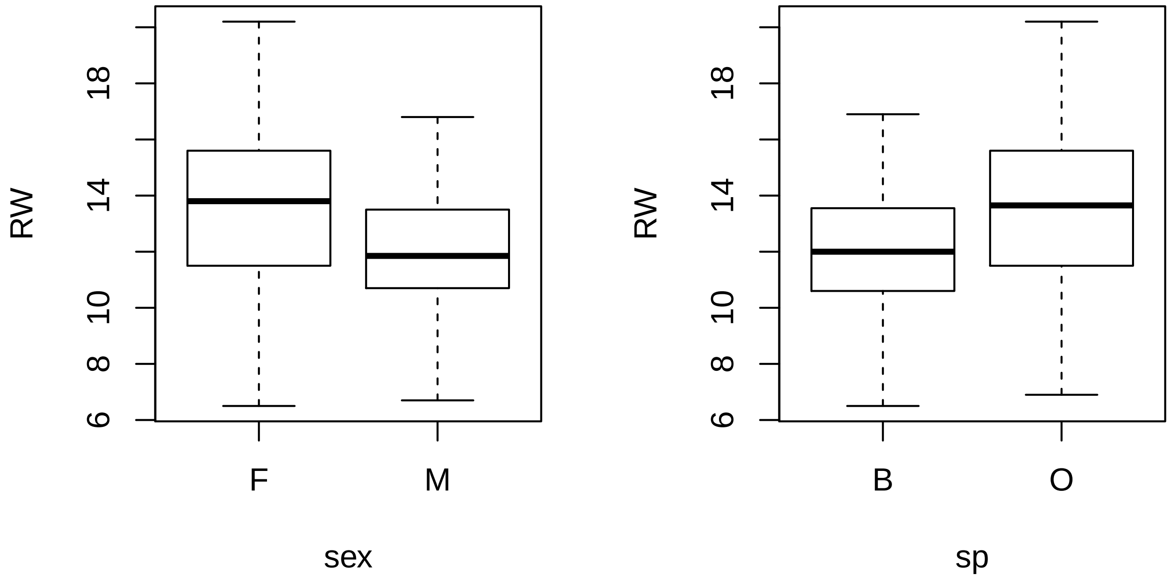
```
plot(RW ~ BD, data = crabs, main = 'Data on Crabs', pch = 16,
     col = sex)
legend('topleft',c('Female','Male'), pch = rep(16,2),
       col = 1:2)
```



In this plot the black dots, which correspond to females, are above the red dots, which correspond to males. This means that for a given value of body depth, females have larger RW than males. Also, there seems to be a linear relation between the two variables, but with different slopes.

5. Divide the plotting window in two columns. Plot on the left a boxplot of `RW` against `sex` and on the right `RW` against `sp`. Comment.

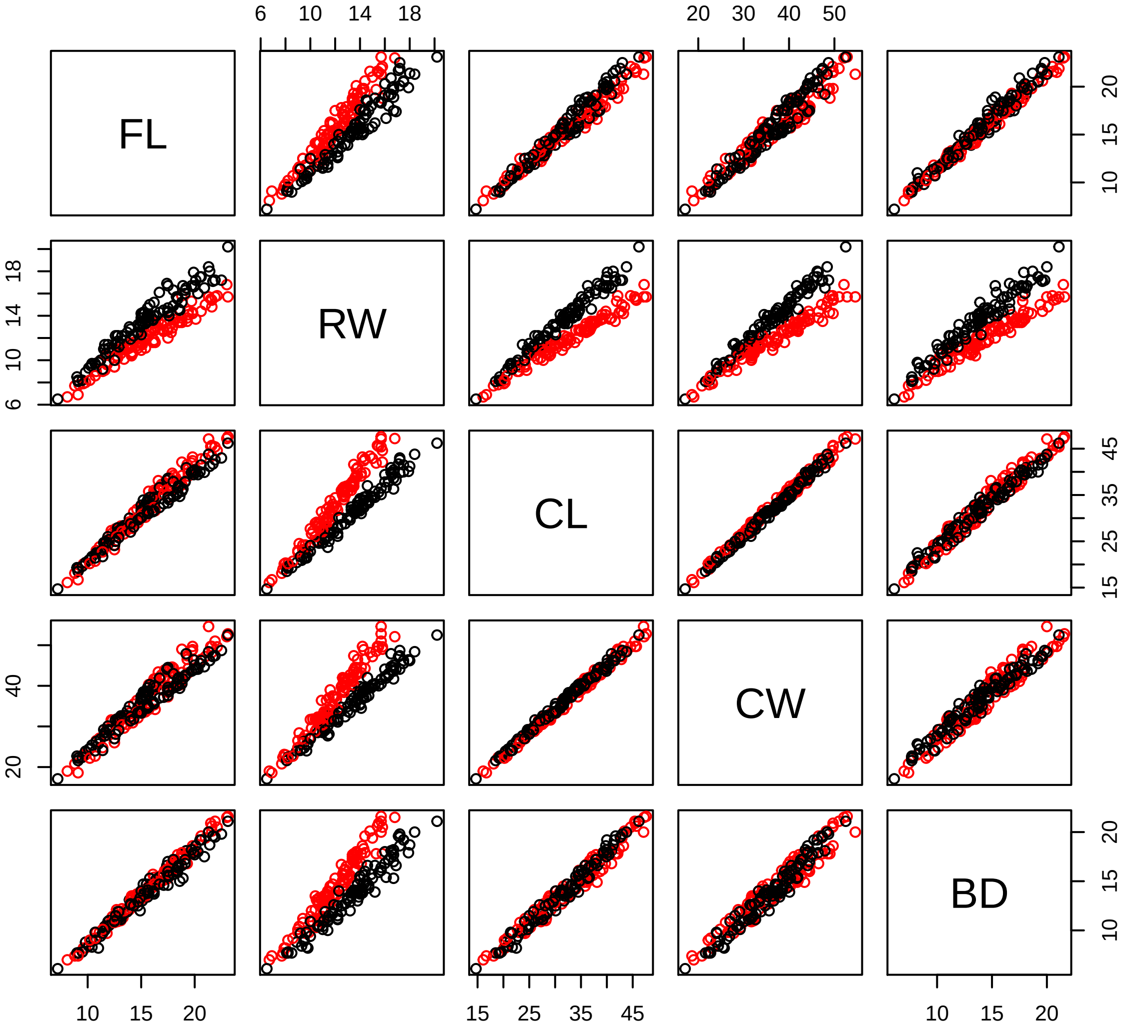
```
par(mfrow=c(1,2))
plot(RW ~ sex, data = crabs)
plot(RW ~ sp, data = crabs)
```



The left plot shows that females have larger values for the variable RW, and also that the values cover a wider range. For both sexes the distributions seem to be symmetrical. As for the plot on the right, the orange species has bigger values of RW and also the values cover a wider range.

6. Using `plot` draw a scatterplot matrix with the numerical variables in `crabs` (columns 4 to 8). Add color by sex. Comment.

```r
plot(crabs[,4:8], col = crabs$sex)
```



The plots that do not involve the variable RW do not show differences according to sex. The plots in the second row or second column, show differences for the two sexes. Hence, RW seems to be the variable that shows more difference between the two sexes.

Exercise 2

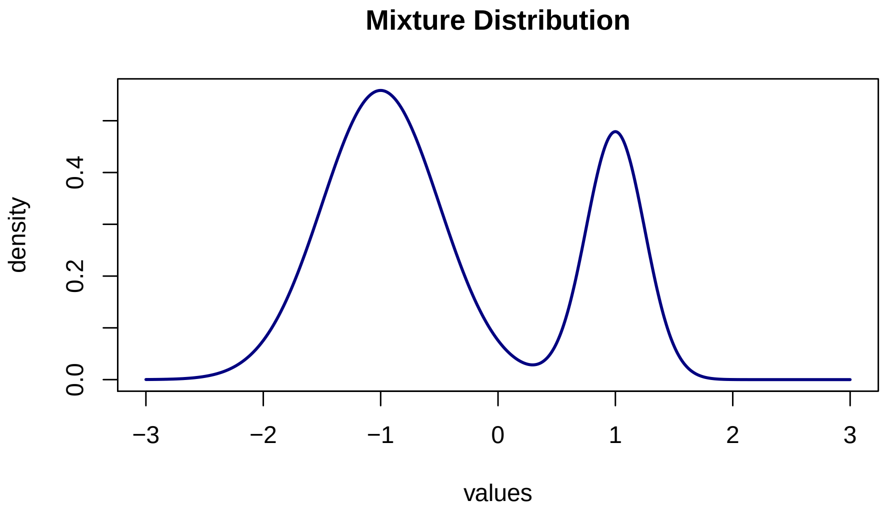
Histograms

For this exercise we are going to use simulated data from a mixture of normal distributions. In this population, 70% of the points come from a normal distribution with mean -1 and standard deviation 0.5, and 30% come from a normal distribution with mean 1 and standard deviation 0.25.

$$0.7 \times N(-1, 0.5^2) + 0.3N(1, 0.25^2)$$

The code below plots the density for this distribution.

```
points.x <- seq(-3,3,length=1000)
points.dens <- 0.7*(dnorm(points.x, mean=-1, sd = 0.5)) +
  0.3*(dnorm(points.x,mean=1, sd = 0.25))
plot(points.x,points.dens,type='l',xlab='values',ylab='density',lwd = 2,
     col = 'navyblue', main = 'Mixture Distribution')
```



The following commands draw a sample of size 500 from this mixture and print the range of values for the simulated data. The sample is stored in the vector `mix.sample`

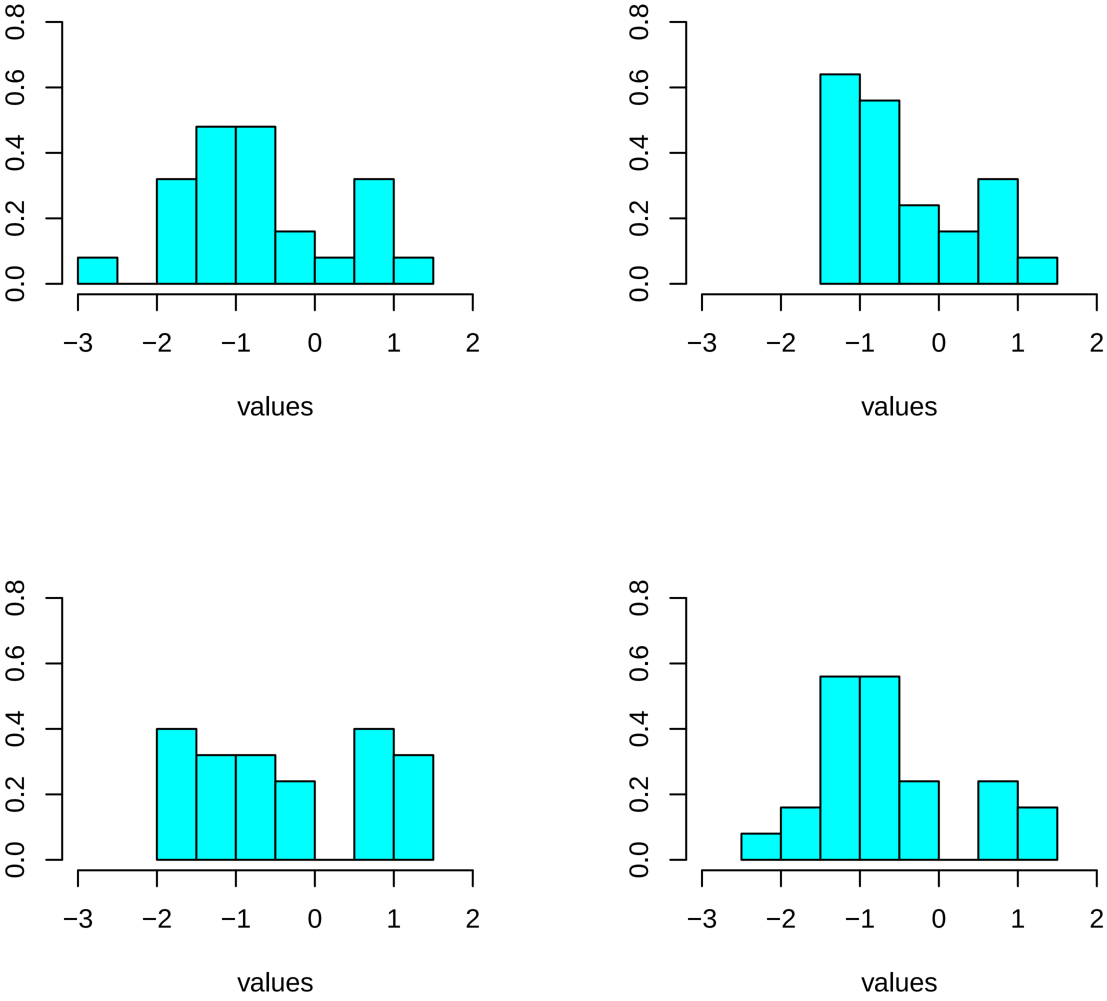
```
n <- 500; set.seed(5678)
unif.sample <- runif(n) <= 0.7
mix.sample <- unif.sample *rnorm(n,mean=-1, sd = 0.5) +
  (1-unif.sample)*rnorm(n,mean=1, sd = .25)
(rng <- range(mix.sample))
```

```
## [1] -2.540183  1.808056
```

We will use this sample to draw histograms with the function `truehist` in the `MASS` package. Look up the help for `truehist`. It is also a good idea to explore the use of the function `hist` on the base package by repeating this exercise using `hist`.

1. Divide the plotting window into 4 using the function `par` with argument `mfrow`. Select four disjoint subsets of data of length 25 and draw histograms for them. Set the bin width to 0.5 in all plots. Make sure that the scales are the same for all plots. Are these plots similar to the density in the previous slide?

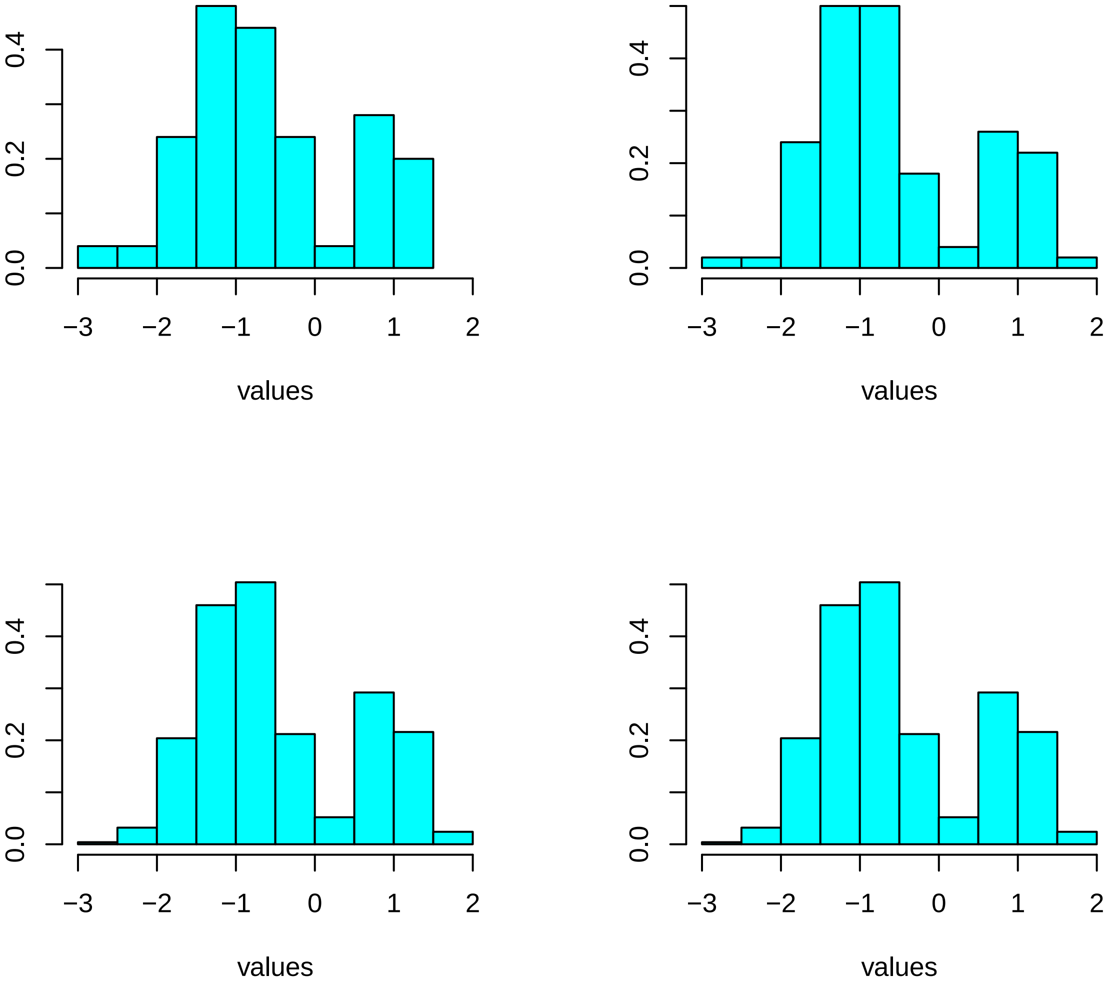
```
par(mfrow=c(2,2))
truehist(mix.sample[1:25], xlab = 'values', h = 0.5,
         xlim = c(floor(rng[1]),ceiling(rng[2])),
         ylim = c(0,0.8))
truehist(mix.sample[101:125], xlab = 'values', h = 0.5,
         xlim = c(floor(rng[1]),ceiling(rng[2])),
         ylim = c(0,0.8))
truehist(mix.sample[201:225], xlab = 'values', h = 0.5,
         xlim = c(floor(rng[1]),ceiling(rng[2])),
         ylim = c(0,0.8))
truehist(mix.sample[301:325], xlab = 'values', h = 0.5,
         xlim = c(floor(rng[1]),ceiling(rng[2])),
         ylim = c(0,0.8))
```



Not really. In the second and fourth plots the bimodality is not clear. The second plot looks like a right-skewed distribution. The third plot looks more like a uniform distribution with data missing in one of the intervals, and the fourth looks like a slightly skewed, unimodal distribution.

2. Divide the plotting window into 4 using the function `par` with argument `mfrow`. Draw successive histograms of relative frequency for the first 25, 50, 100, and 500 points in `mix.sample`. Set the bin width to 0.5 in all plots. Make sure that the scales are the same for all plots. Are these plots similar to the density in the previous slide?

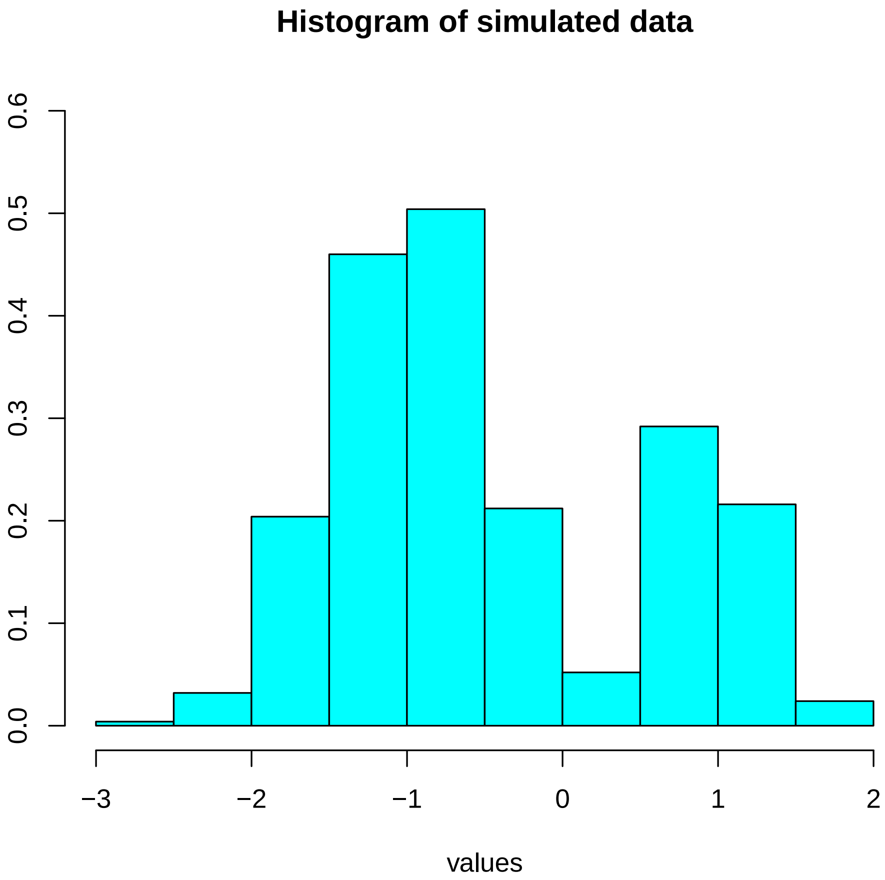
```r
par(mfrow=c(2,2))
truehist(mix.sample[1:50], xlab = 'values', h = 0.5,
         xlim = c(floor(rng[1]),ceiling(rng[2])))
truehist(mix.sample[1:100], xlab = 'values', h = 0.5,
         xlim = c(floor(rng[1]),ceiling(rng[2])))
truehist(mix.sample[1:500], xlab = 'values', h = 0.5,
         xlim = c(floor(rng[1]),ceiling(rng[2])))
truehist(mix.sample[1:1000], xlab = 'values', h = 0.5,
         xlim = c(floor(rng[1]),ceiling(rng[2])))
```



In this case all plots look similar to the populations density. They show clearly the bimodal nature of the data and the proportion between the two modes is approximately correct.

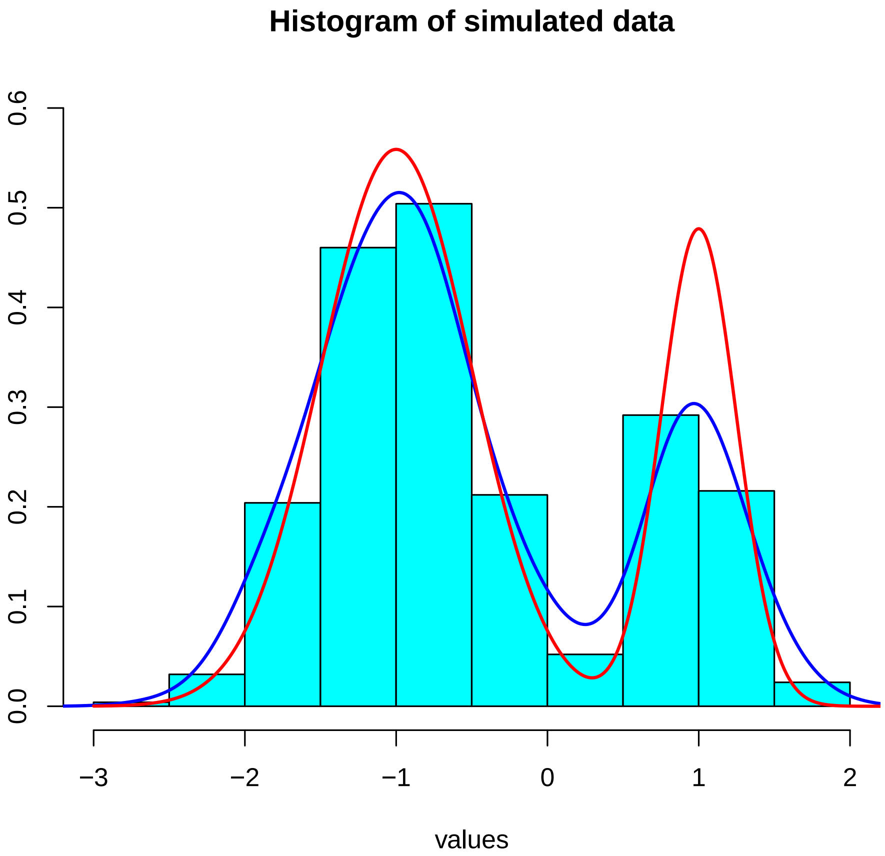
3. Using again the function `par` with argument `mfrow`, set the graphical window to a single graph. Draw a histogram of relative frequency using all the points in `mix.sample`. Choose the number of bins (`nbins`) using the `Scott` rule.

```
par(mfrow=c(1,1))
truehist(mix.sample[1:1000], xlab = 'values',
         xlim = c(floor(rng[1]),ceiling(rng[2])),
         main = 'Histogram of simulated data', nbins = 'Scott',
         ylim = c(0,0.6))
```



4. Using the function `lines` with argument `density(sample.mix)`, add an estimate of the density for this sample. Color the line in blue. Add also a graph of the theoretical density in red (look back to the previous page to see how this density was plotted before and make the necessary changes). Comment on what you observe.

```
par(mfrow=c(1,1))
truehist(mix.sample[1:1000], xlab = 'values',
         xlim = c(floor(rng[1]),ceiling(rng[2])),
         main = 'Histogram of simulated data', nbins = 'FD',
         ylim = c(0,0.6))
lines(density(mix.sample),col = 'blue', lwd=2)
lines(points.x,points.dens,type='l',col = 2, lwd=2)
```



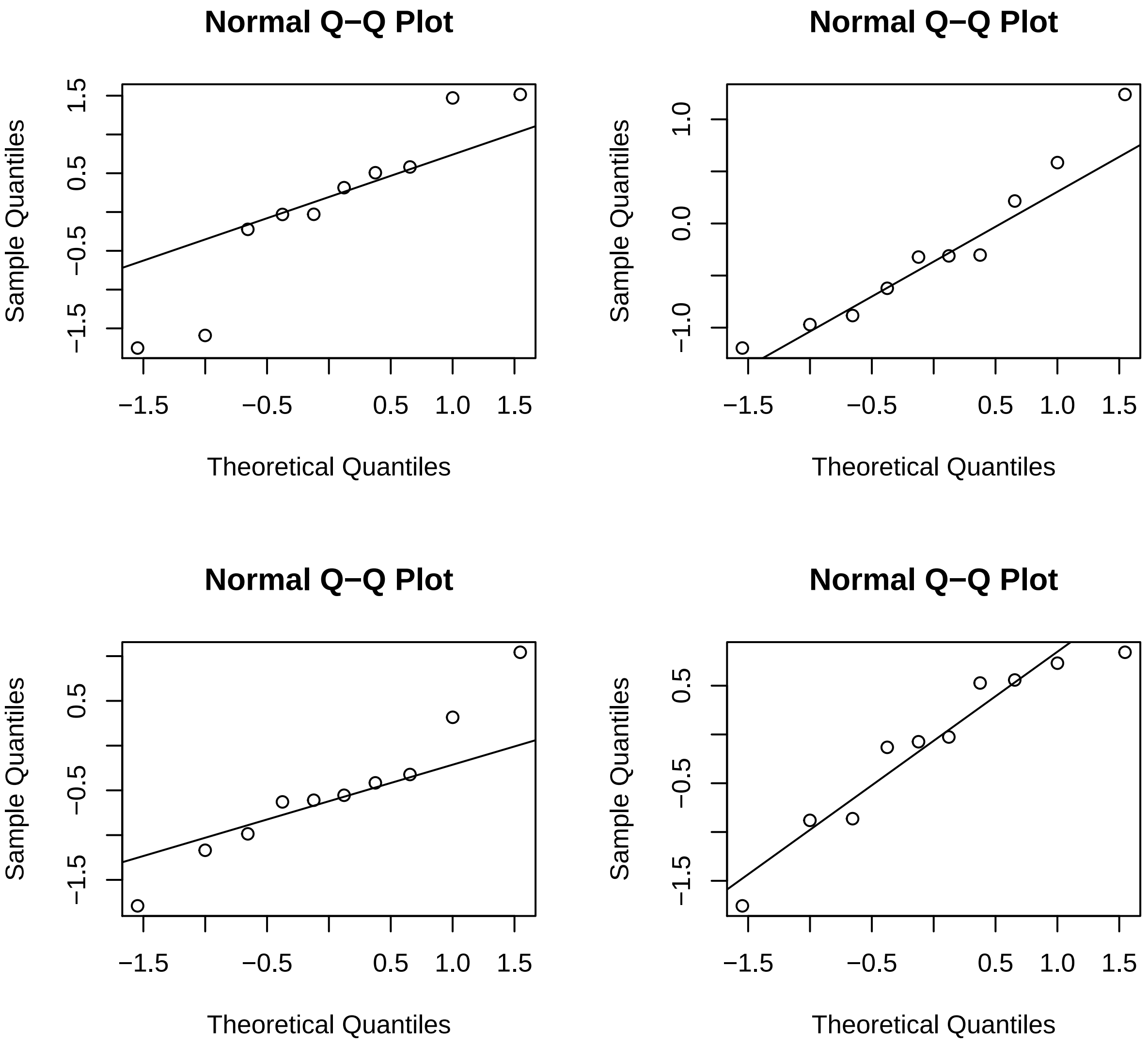
We see that the estimated density and the histogram are reasonably close to the population density. However, the positive part of the density is not sufficiently represented in this graph.

Exercise 3

In this exercise we look at quantile plots. In all cases we will consider samples simulated from the normal distribution. We explore the effect of size, mean, and variance, and also use `qqplot` to compare samples.

1. Divide the graphical window into four regions using `par` and `mfrow`. Generate four samples from the standard normal distribution of size 10 and draw normal quantile plots. Add lines with `qqline`. Comment on what you observe.

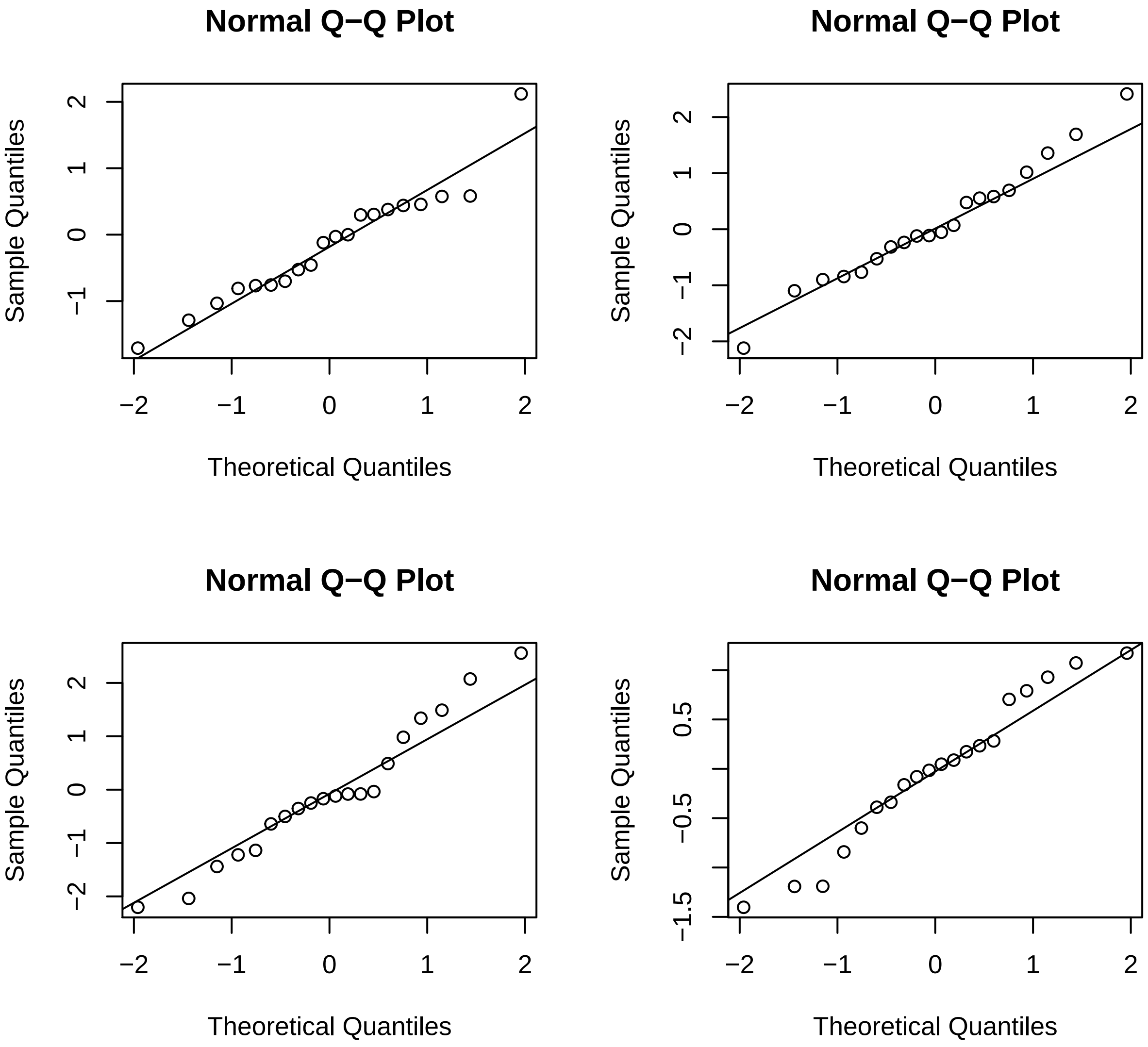
```r
par(mfrow=c(2,2))
for(i in 1:4) {samp1 <- rnorm(10); qqnorm(samp1); qqline(samp1)}
```



In plot 1 the six points in the middle are aligned but the other four points do not show a good fit. Something similar occurs with plot 3. Plots 2 and 4 show a good alignment of the points.

2. Repeat for sample sizes 20, 50, and 100. Comment on what you observe.

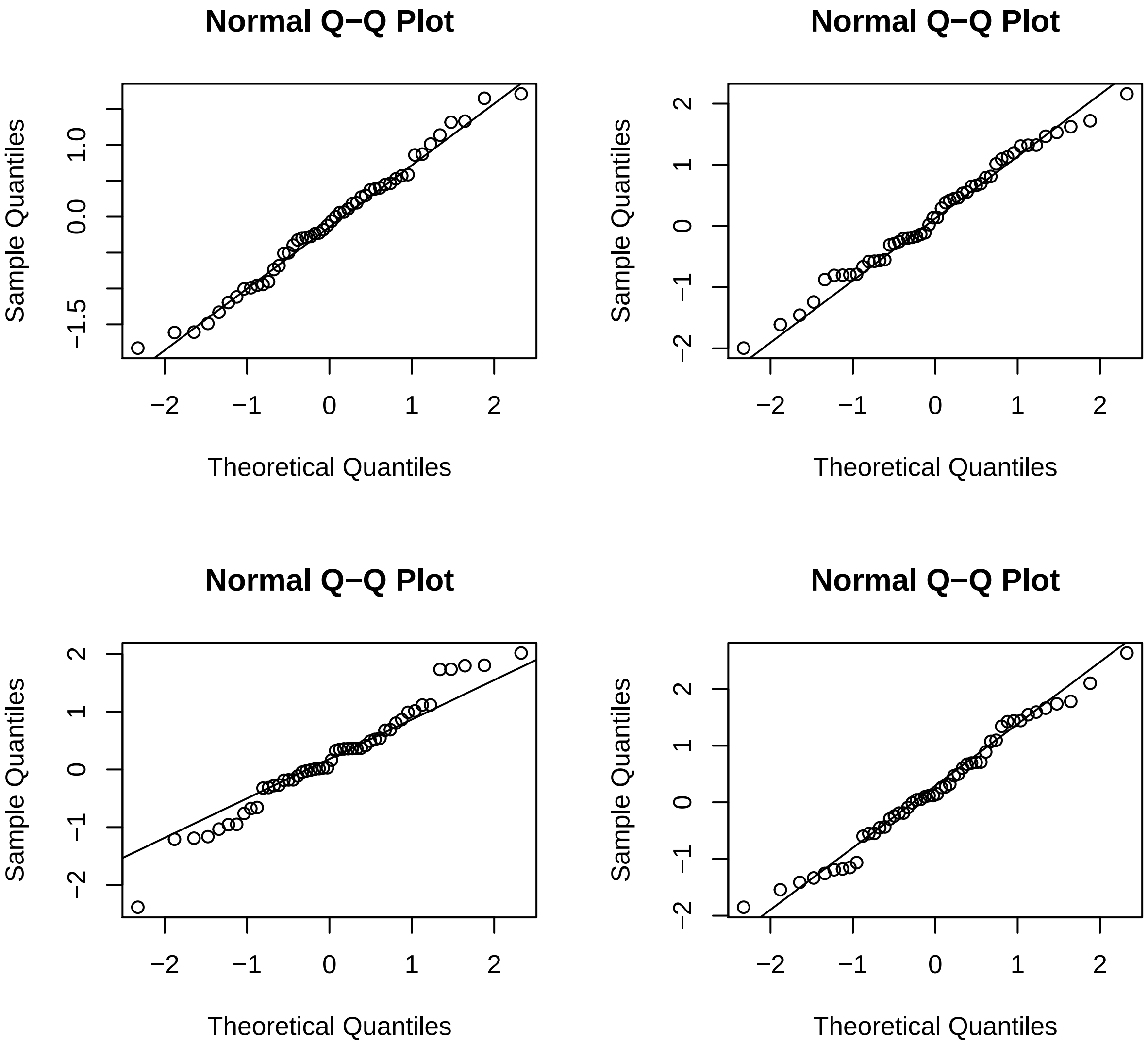
```r
par(mfrow=c(2,2))
for(i in 1:4) {samp1 <- rnorm(20); qqnorm(samp1); qqline(samp1)}
```



For sample size 20 the fit is reasonable, but there are still some points that deviated markedly from the line.

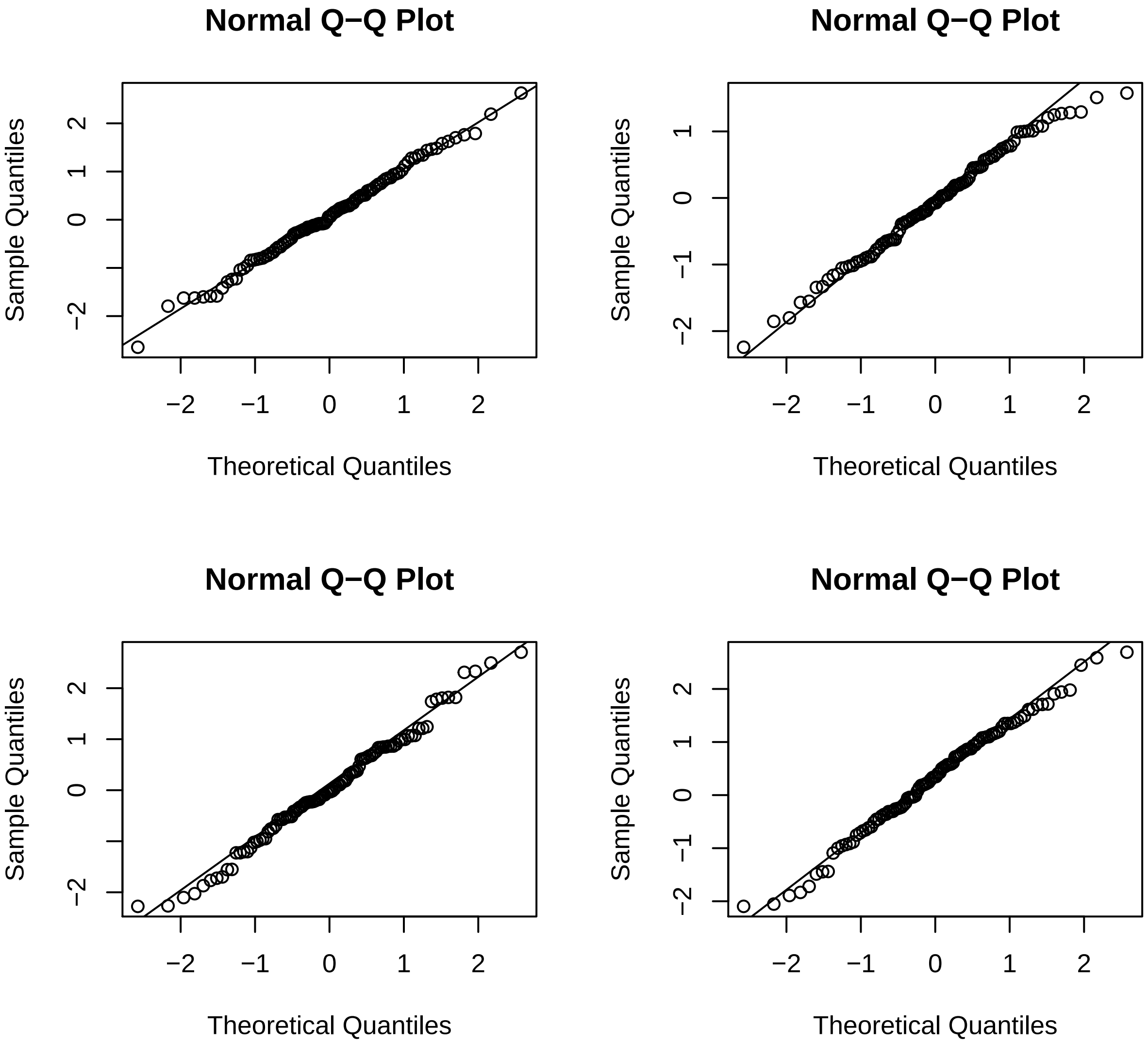
2. Repeat for sample sizes 20, 50, and 100.

```
par(mfrow=c(2,2))
for(i in 1:4) {samp1 <- rnorm(50); qqnorm(samp1); qqline(samp1)}
```



For sample size 50 the fit is better. In three out of four plots the fit is very good. Plot 3 has a large minimum value that deviates from the rest.

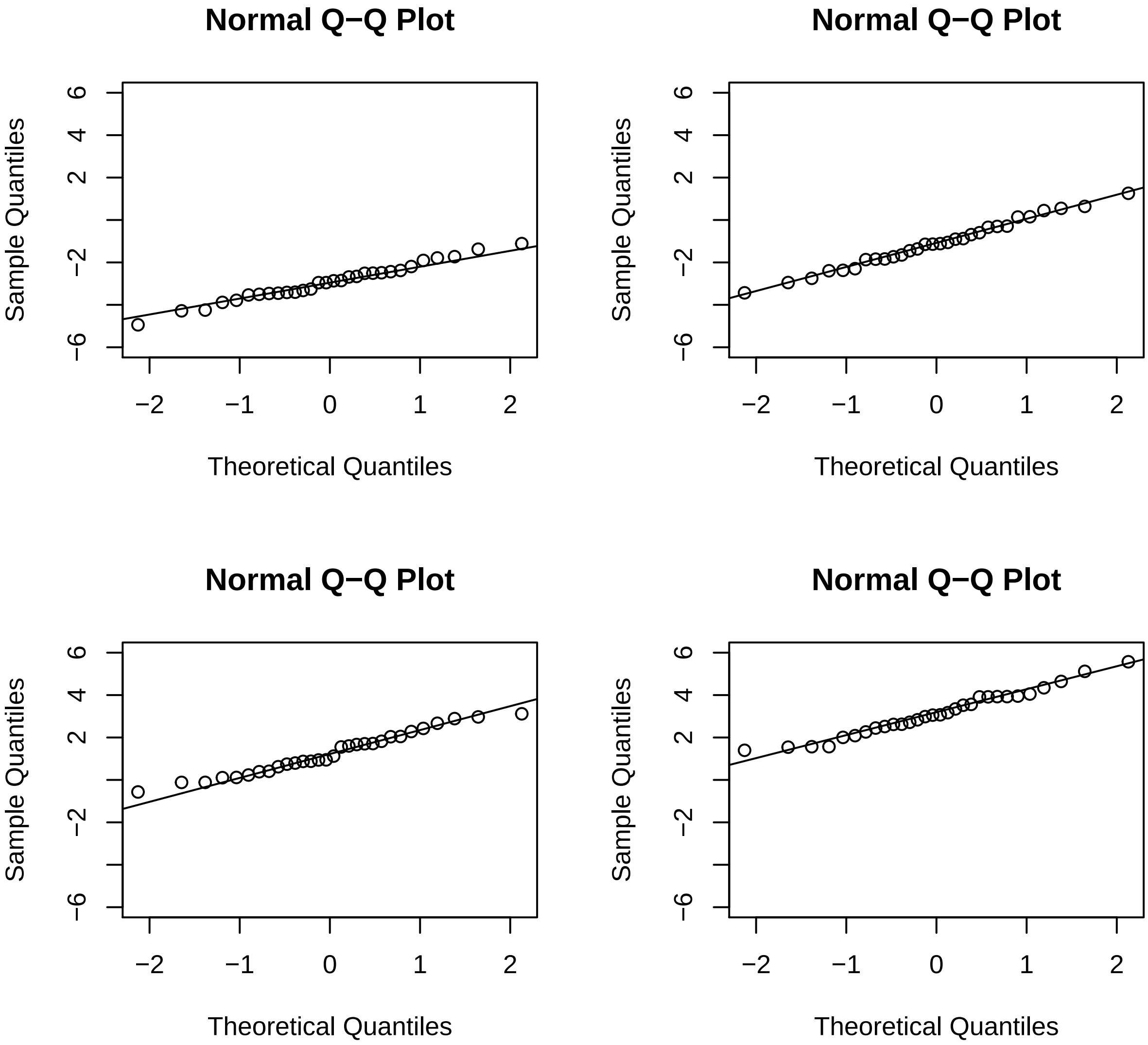
```r
par(mfrow=c(2,2))
for(i in 1:4) {samp1 <- rnorm(100); qqnorm(samp1); qqline(samp1)}
```



Now the fit is very good iin all cases. We see that, as the sample size grows, the fit improves.

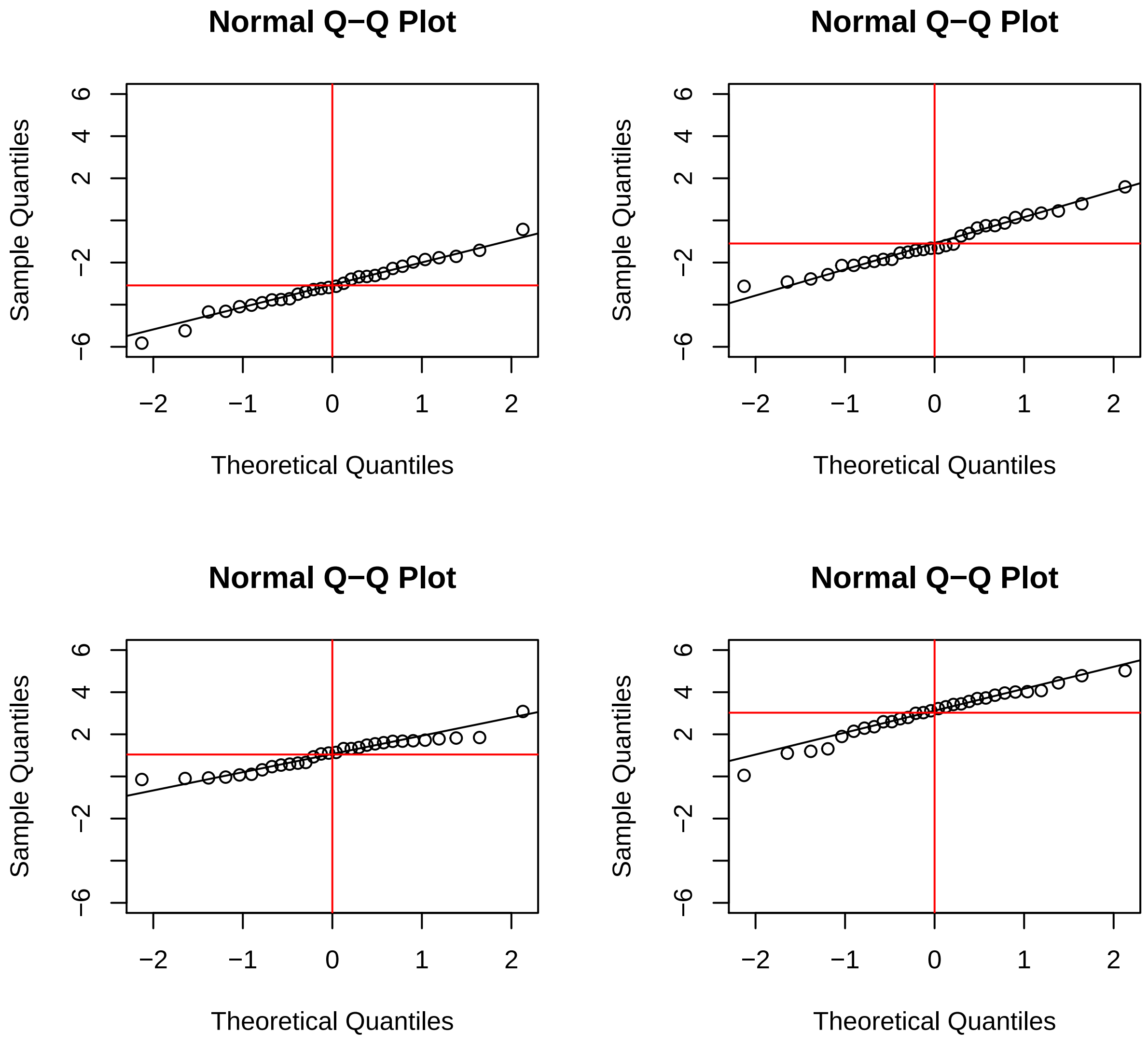
3. Draw samples of size 50 from normal distributions with means -6, -2, 2, and 6, all with variance 1 and draw the corresponding quantile plots. To be able to compare the four graphs, find a suitable common scale for the axes for all plots. Comment on the similarities and differences between the plots.

```
par(mfrow=c(2,2))
for (i in c(-3,-1,1,3)) {
  dat <- rnorm(30,i);qqnorm(dat,ylim=c(-6,6));qqline(dat)}
```



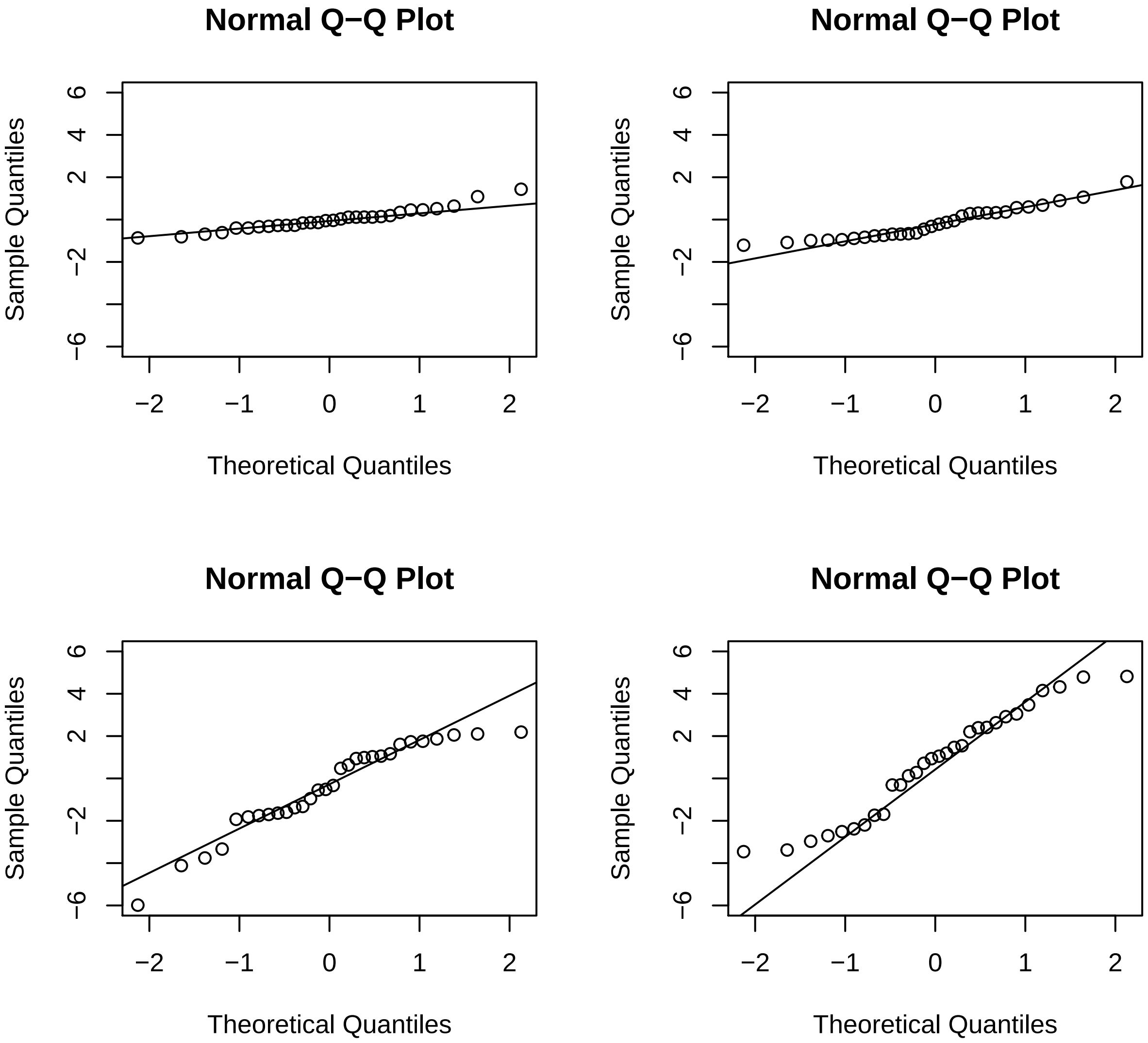
In the plots we see that the slope of the lines remain constant, but the lines shift upwards as the mean increases.

```
par(mfrow=c(2,2))
for (i in c(-3,-1,1,3)) {
  dat <- rnorm(30,i);qqnorm(dat,ylim=c(-6,6));qqline(dat)
  abline(v=0,col='red'); abline(h=mean(dat),col = 'red')}
```



4. Draw samples of size 50 from normal distributions with mean 1 and standard deviations 0.5, 2, 4, and 6, and draw the corresponding quantile plots. To be able to compare the four graphs, find a suitable common scale for the axes for all plots. Comment on the similarities and differences between the plots.

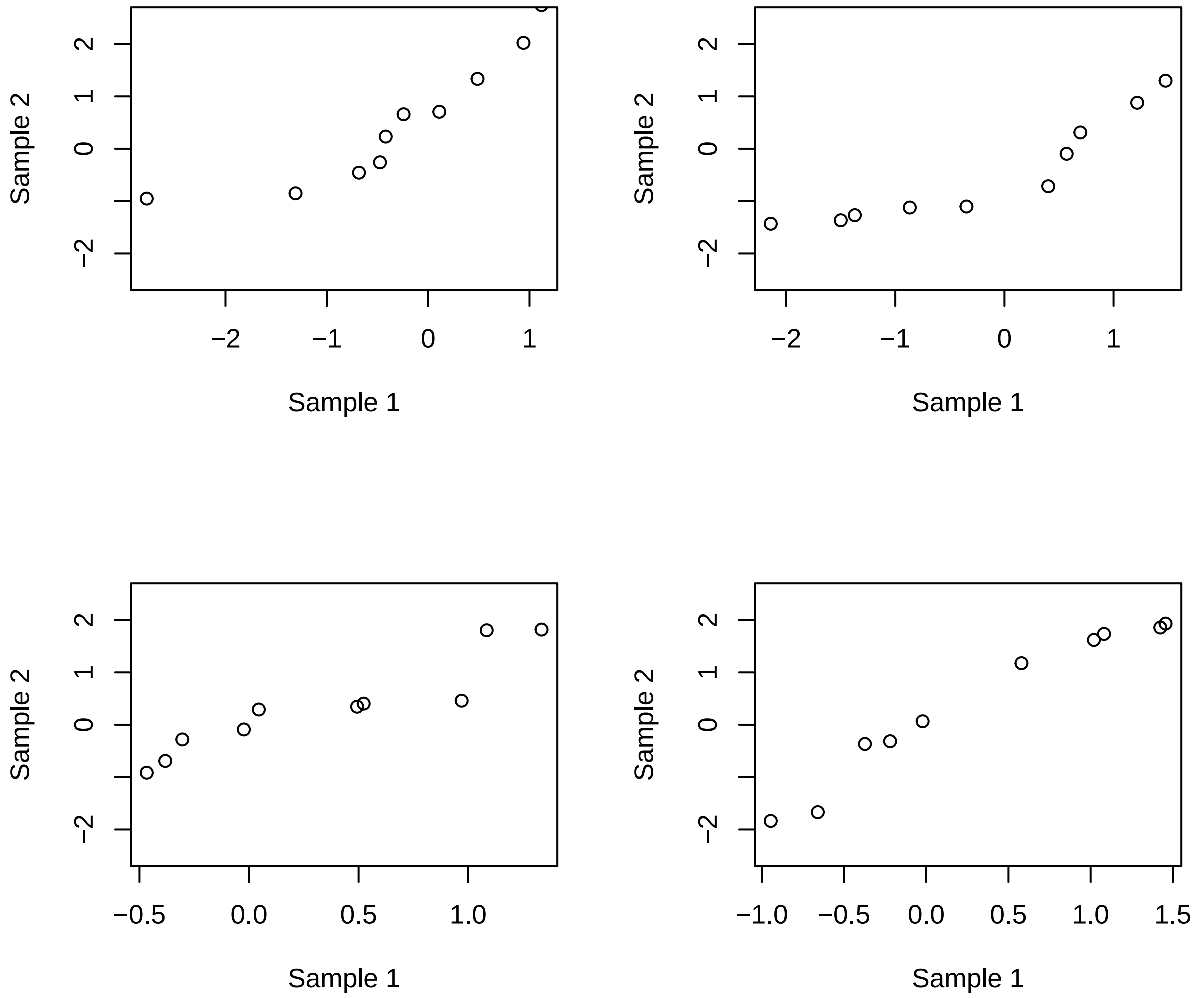
```
par(mfrow=c(2,2))
for (i in c(0.5,1,2,3)) {
  dat <- rnorm(30,0,i);qqnorm(dat,ylim=c(-6,6));qqline(dat)}
```



In these plots we see that the height of the central points remains constant, but the slope of the lines increases as the variance increases.

5. Draw two samples of size 10 from the standard normal distribution and compare them using `qqplot`. Repeat a total of four times. Plot the four graphs on the same window. Comment on what you see.

```
par(mfrow=c(2,2))
for (i in 1:4) {
  dat <- rnorm(20);qqplot(dat[1:10],dat[11:20], ylim=c(-2.5,2.5),
                          xlab = 'Sample 1', ylab = 'Sample 2')}
```



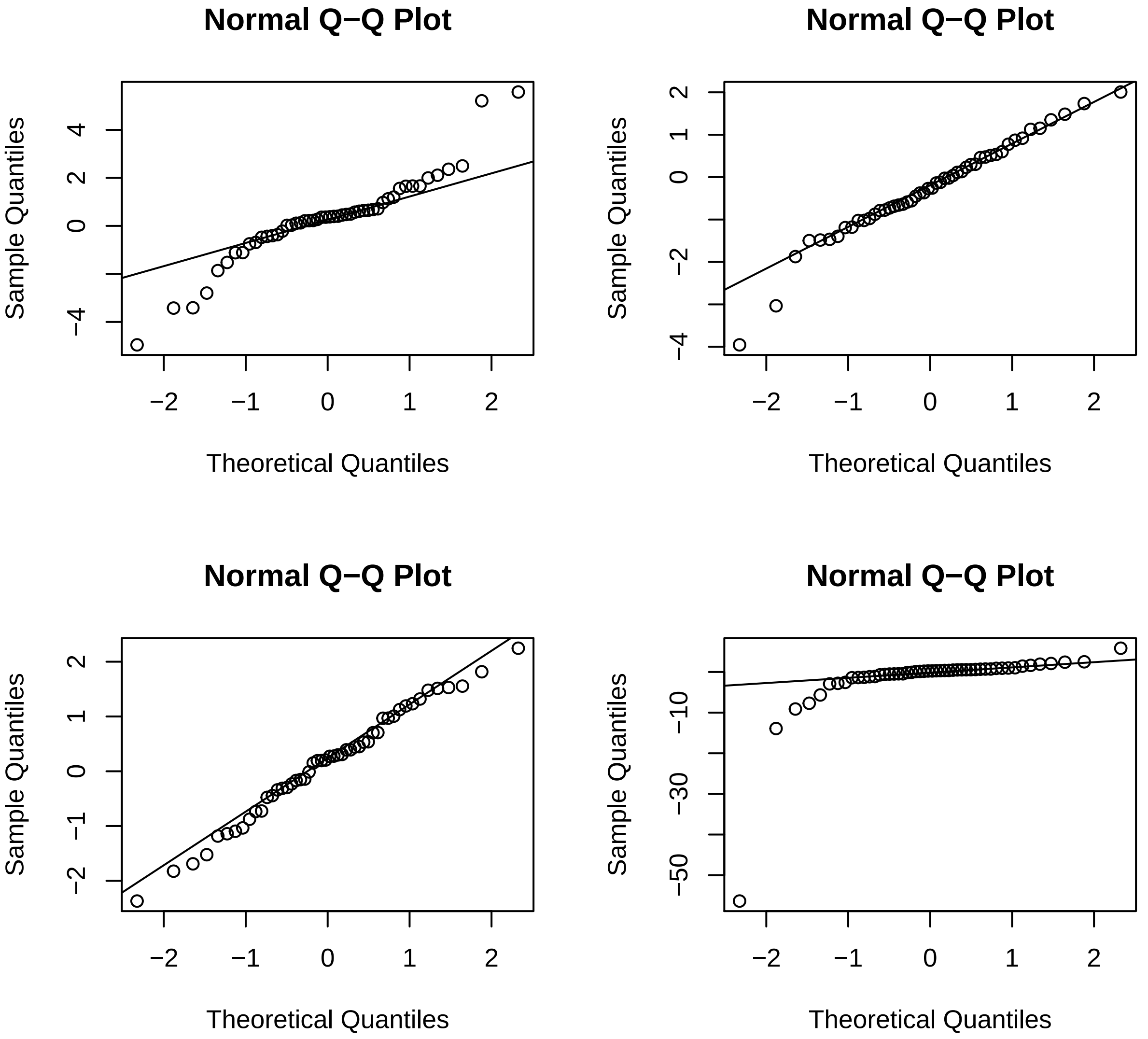
We see that the first two plots do not show adequate alignment, and we would probably conclude that the two samples come from different distributions. The last two plots show points that are reasonably aligned, and we would conclude that in this case they come from a common distribution function.

Exercise 4

In this exercise we look again at quantile plots, but now we explore the effect of comparing samples from other distributions with the normal.

1. **Distributions with heavy tails**. Draw samples of size 50 from the following distributions: t with 2 degrees of freedom, t with 5 degrees of freedom, t with 10 degrees of freedom, and Cauchy with standard parameters. For each of these samples draw normal quantile plots along with the reference line. Use a single graphic window divided in four. Comment on what you observe.

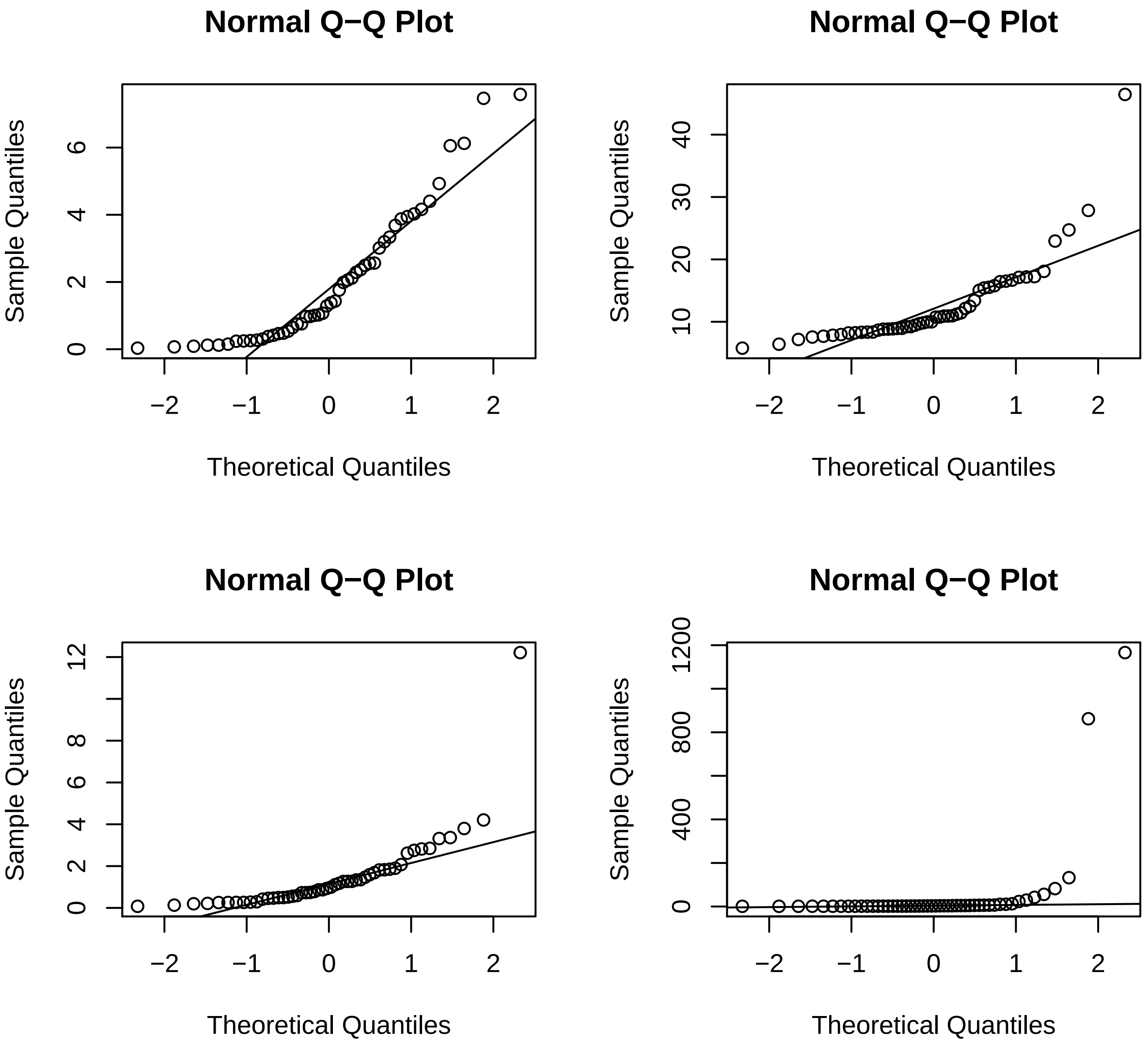
```r
sampt2 <- rt(50,2); sampt5 <- rt(50,5); sampt10 <- rt(50,10)
sampcauchy <- rcauchy(50)
par(mfrow = c(2,2))
qqnorm(sampt2); qqline(sampt2)
qqnorm(sampt5); qqline(sampt5)
qqnorm(sampt10); qqline(sampt10)
qqnorm(sampcauchy); qqline(sampcauchy)
```



As the degrees of freedom for the t distribution increase, the fit to the normal distribution gets better, because the t distribution converges to the normal as the degrees of freedom go to infinity. We see this in the first three graphs. On the other hand, the Cauchy distribution is symmetric as the normal, but it has much heavier tails. This shows in the fact that some values are completely out of the scale of a normal sample.

2. **Distributions bounded below**. Draw samples of size 50 from the following distributions: χ^2 with 2 degrees of freedom, F with 5 and 10 degrees of freedom, lognormal with standard parameters, and pareto with location parameter equal to 1. For the pareto distribution you need to install and load the package `EnvStat` and use the function `rpareto`. For each of these samples draw normal quantile plots along with the reference line. Use a single graphic window divided in four. Comment on what you observe.

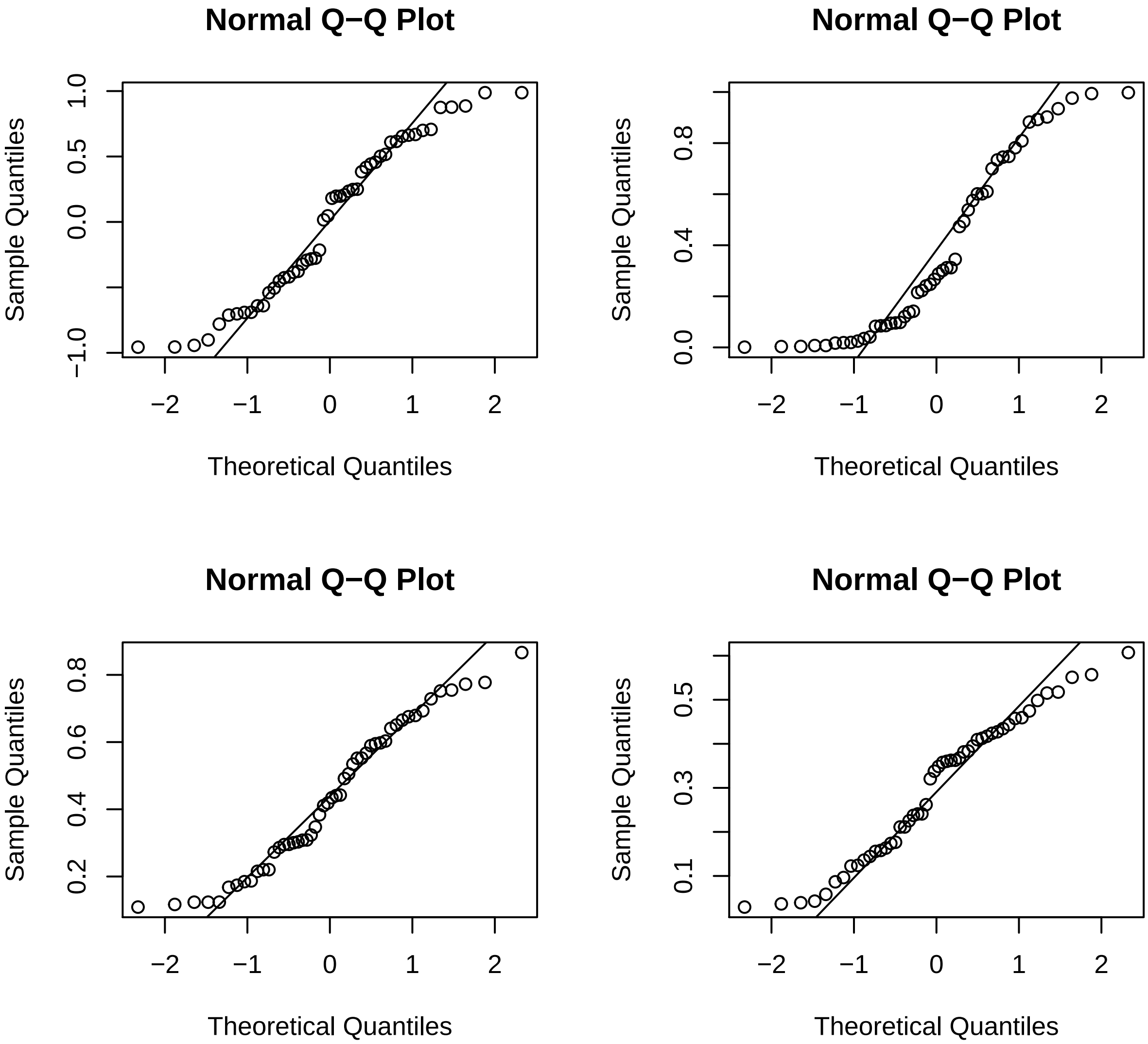
```r
library(EnvStats)
sampchi2 <- rchisq(50,2); sampf5 <- rt(50,5,10); sampln <- rlnorm(50)
samppareto <- rpareto(50, 1)
par(mfrow = c(2,2))
qqnorm(sampchi2); qqline(sampchi2)
qqnorm(sampf5); qqline(sampf5)
qqnorm(sampln); qqline(sampln)
qqnorm(samppareto); qqline(samppareto)
```



Observe that the four distributions in these graphs have only positive values, and therefore the lower tail of the plot is bounded by zero. In all cases the fit is bad but for different reasons. In the first plot (chi-square) the lower tail deviates for the reference line. In the second plot (F distribution) the points deviate at the two extremes, the lognormal distribution shows a similar pattern. Finally, the Pareto distribution has some extremely large values on the right tail.

3. **Bounded distributions**. Draw samples of size 50 from the following distributions: Uniform in $[-1, 1]$, Beta with both parameters equal to 0.5, Beta with both parameters equal to 2, and Beta with `shape1 =1`, `shape2 = 3`. For each of these samples draw normal quantile plots along with the reference line. Use a single graphic window divided in four. Comment on what you observe.

```r
sampunif <- runif(50,-1,1); sampbeta1 <- rbeta(50,.5,.5); sampbeta2 <- rbeta(50,2,2)
sampbeta3 <- rbeta(50,1,3)
par(mfrow = c(2,2))
qqnorm(sampunif); qqline(sampunif)
qqnorm(sampbeta1); qqline(sampbeta1)
qqnorm(sampbeta2); qqline(sampbeta2)
qqnorm(sampbeta3); qqline(sampbeta3)
```



In these graphs all distributions have bounded support, and the plots are similar. They all show lack of fit in the tails.

4. **Mixtures of normal distributions**. Modifying the commands in question 2 of this list, draw samples of size 50 from the following distributions:

   (a) A mixture of normal distributions with 50% of the population coming from a normal distribution with mean -1 and standard deviation 0.5, and 50% coming from a normal with mean 1 and standard deviation 0.5.

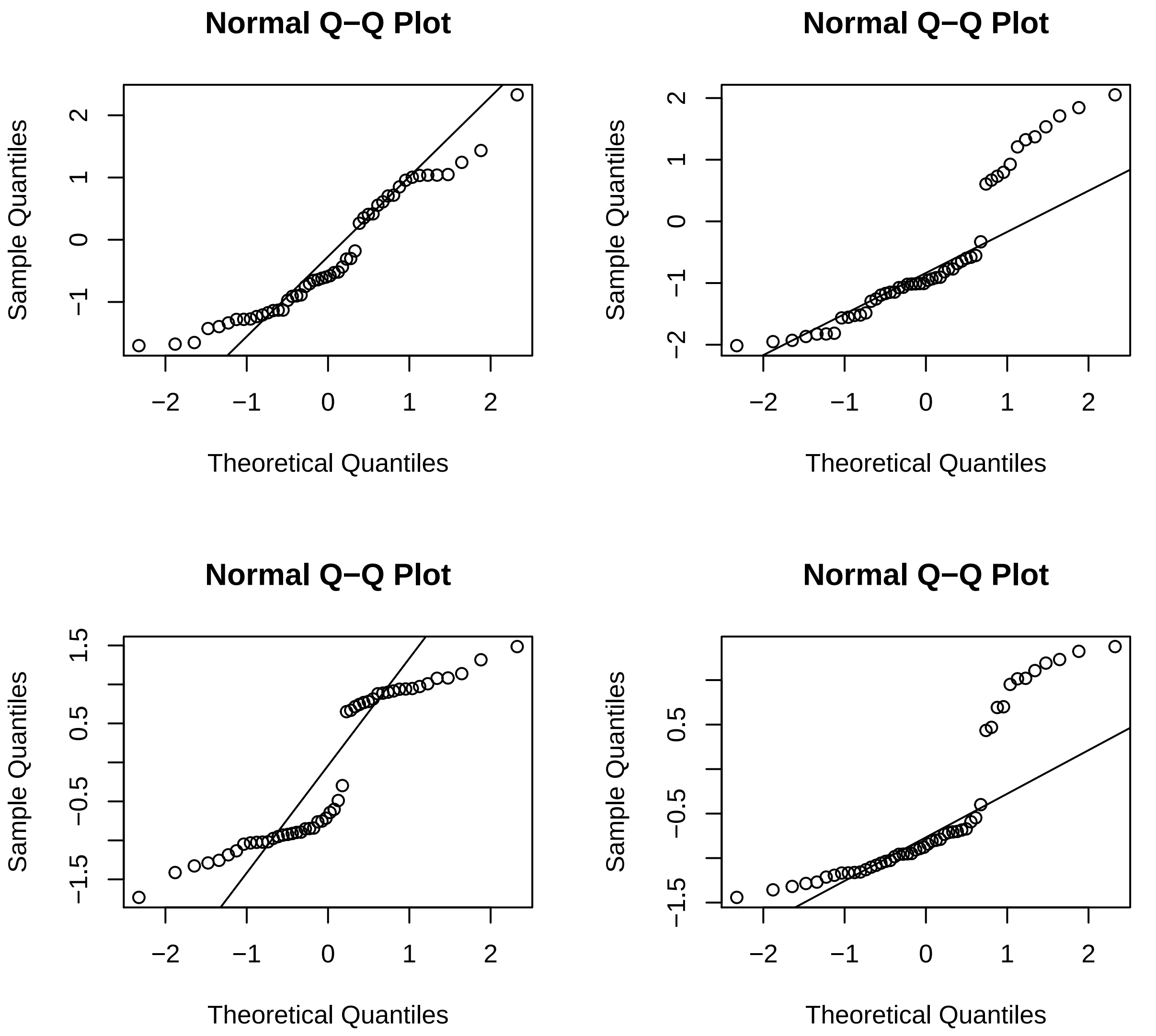
   (b) Same as above but change the proportions to 80% and 20%.

   (c) Same as (a) but change the standard deviations to 0.25.

   (d) Same as (b) but change the standard deviations to 0.25.

(a)

```
n <- 50
unif.sample <- runif(n) <= 0.5
mix.sample1 <- unif.sample*rnorm(n,mean=-1, sd = 0.5)+
  (1-unif.sample)*rnorm(n,mean=1, sd = .5)
unif.sample <- runif(n) <= 0.8
mix.sample2 <- unif.sample*rnorm(n,mean=-1, sd = 0.5)+
  (1-unif.sample)*rnorm(n,mean=1, sd = .5)
unif.sample <- runif(n) <= 0.5
mix.sample3 <- unif.sample*rnorm(n,mean=-1, sd = 0.25)+
  (1-unif.sample)*rnorm(n,mean=1, sd = .25)
unif.sample <- runif(n) <= 0.8
mix.sample4 <- unif.sample*rnorm(n,mean=-1, sd = 0.25)+
  (1-unif.sample)*rnorm(n,mean=1, sd = .25)
par(mfrow = c(2,2))
qqnorm(mix.sample1); qqline(mix.sample1)
qqnorm(mix.sample2); qqline(mix.sample2)
qqnorm(mix.sample3); qqline(mix.sample3)
qqnorm(mix.sample4); qqline(mix.sample4)
```



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

Mixture distribution show a total lack of fit. They frequently show an S shape, but in this case, since the means are a distance apart, we see two groups of points in the plot. the number of points in each depends on the weights of each population. The plots with smaller standard deviation show two clearly separated groups.