STAT 210
Applied Statistics and Data Analysis:
Homework 3

Due on Sep. 25/2022

Question 1

Access the data from url http://www.stat.berkeley.edu/users/statlabs/data/babies.data and store the information in an object named `BABIES` using the function `read.table()`. Use the option that reads the first line as header.

A description of the variables can be found at http://www.stat.berkeley.edu/users/statlabs/labs.html. Look for the data set Birth Weight II. These data are a subset from a much larger study dealing with child health and development.

(i) Create a "clean" data set that removes subjects if any observations on the subject are "unknown." Note that `bwt, gestation, parity, height, weight`, and `smoke` use values of 999, 999, 9, 99, 999, and 9, respectively, to denote "unknown." Store the modified data set in an object named `CLEAN`. The function `subset` may be useful here.

(ii) Use the information in `CLEAN` to create a histogram of the birth weights of babies whose mothers have never smoked (`smoke=0`) and another histogram placed directly below the first in the same graphics device for the birth weights of babies whose mothers currently smoke (`smoke=1`). Use a common range of the x-axis for both histograms. Superimpose a density curve over each histogram. Use informative titles and labels for your graphs. Comment on what you observe.

(iii) The body weight index or body mass index (bmi) is defined as the weight of a person divided by the height squared and is measured in units of kg/m^2. Compute the bmi for each mother in `CLEAN`. Observe that you have to convert the measurements in the data frame to metric (0.0254 m= 1 in., and 0.45359 kg= 1 lb.). Modify the variables `weight` and `height` so that they now appear in metric units (kg and m), and add `bmi` to `CLEAN` and store the result in `CLEANP`. Count how many subjects have `bmi` above 30.

Solution We start by loading the data and looking at the structure:

```
#BABIES <- read.table("http://www.stat.berkeley.edu/users/statlabs/data/babies.data",
BABIES <- read.table('babies.data', header = T)
str(BABIES)
```

```
## 'data.frame':    1236 obs. of  7 variables:
##  $ bwt      : int  120 113 128 123 108 136 138 132 120 143 ...
##  $ gestation: int  284 282 279 999 282 286 244 245 289 299 ...
##  $ parity   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ age      : int  27 33 28 36 23 25 33 23 25 30 ...
##  $ height   : int  62 64 64 69 67 62 62 65 62 66 ...
##  $ weight   : int  100 135 115 190 125 93 178 140 125 136 ...
##  $ smoke    : int  0 0 1 0 1 0 0 0 0 1 ...
```

Part (i)

We check how many records have missing information:

```
with(BABIES, sum(bwt == 999 | gestation == 999 | parity == 9 |
                   height == 99 | weight == 999 | smoke == 9))
```

```
## [1] 61
```

Next, we clean the data and check again

```
CLEAN <- subset(BABIES,bwt!= 999
                & gestation!= 999
                & parity != 9
                & height != 99
                & weight != 999
                & smoke != 9)
str(CLEAN)
```

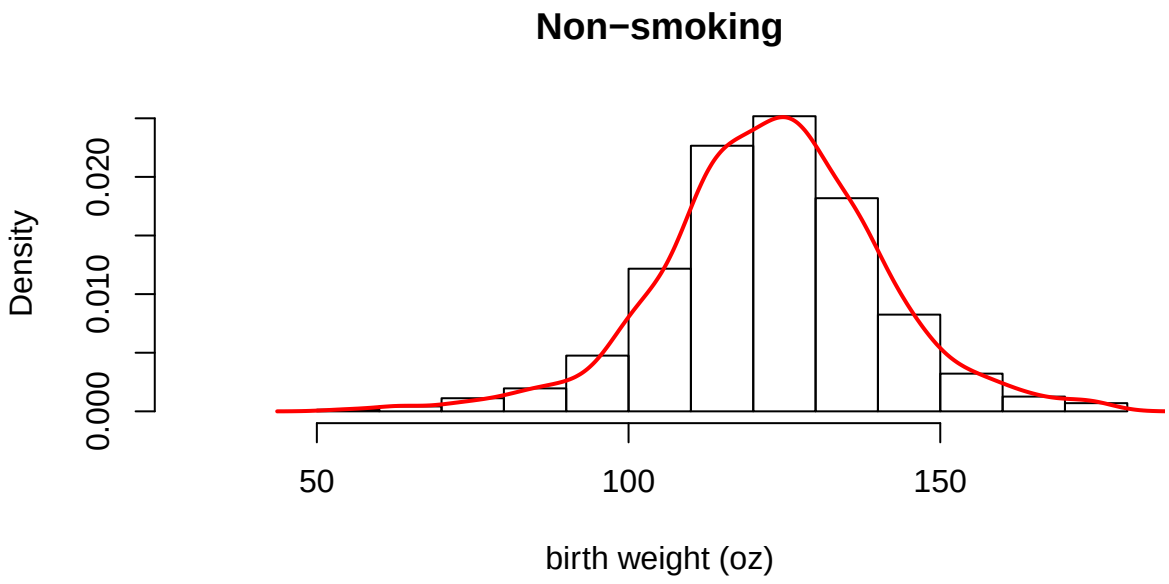
```
## 'data.frame':    1175 obs. of  7 variables:
##  $ bwt      : int  120 113 128 108 136 138 132 120 143 140 ...
##  $ gestation: int  284 282 279 282 286 244 245 289 299 351 ...
##  $ parity   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ age      : int  27 33 28 23 25 33 23 25 30 27 ...
##  $ height   : int  62 64 64 67 62 62 65 62 66 68 ...
##  $ weight   : int  100 135 115 125 93 178 140 125 136 120 ...
##  $ smoke    : int  0 0 1 1 0 0 0 0 1 0 ...
```

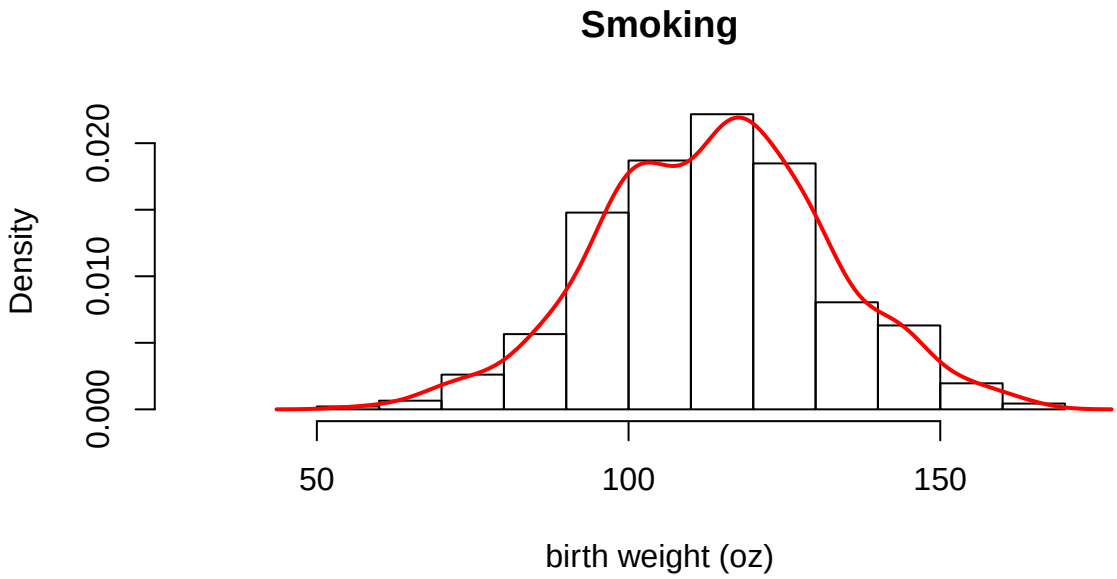
```
with(CLEAN, sum(bwt == 999 | gestation == 999 | parity == 9 |
                  height == 99 | weight == 999 | smoke == 9))
```

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

Part (ii)

```
attach(CLEAN)
par(mfrow=c(2,1))
hist(bwt[smoke==0], xlim=c(30, 180), xlab='birth weight (oz)',
     main='Non-smoking', freq = FALSE)
lines(density(bwt[smoke==0]), col='red',lwd=2)
hist(bwt[smoke==1], xlim=c(30, 180), xlab='birth weight (oz)',
     main='Smoking', freq = FALSE)
lines(density(bwt[smoke==1]), col='red',lwd=2)
```






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

We observe that the distribution for the smoking mothers is shifted to smaller values with respect to the non-smoking mothers.

Part (iii)

```
CLEANP <- CLEAN
CLEANP$height <- CLEANP$height*.0254
CLEANP$weight <- CLEANP$weight*0.45359
CLEANP$bmi <- CLEANP$weight/(CLEANP$height^2)
str(CLEANP)
```

```
## 'data.frame':    1175 obs. of  8 variables:
##  $ bwt      : int  120 113 128 108 136 138 132 120 143 140 ...
##  $ gestation: int  284 282 279 282 286 244 245 289 299 351 ...
```

```
##  $ parity   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ age      : int  27 33 28 23 25 33 23 25 30 27 ...
##  $ height   : num  1.57 1.63 1.63 1.7 1.57 ...
##  $ weight   : num  45.4 61.2 52.2 56.7 42.2 ...
##  $ smoke    : int  0 0 1 1 0 0 0 0 1 0 ...
##  $ bmi      : num  18.3 23.2 19.7 19.6 17 ...
```

```
sum(CLEANP$bmi > 30)
```

```
## [1] 34
```

Question 2

The file `dataQ2` has four simulated samples of size 20 coming from the following distributions

- Standard Cauchy, (`rcauchy(20)`)
- Chi-square with 2 degrees of freedom, χ^2_2, (`rchisq(20,2)`)
- Lognormal with standard parameters, (`rlnorm(20)`)
- Weibull with shape parameter 1 (`rweibull(20,2)`)

You have to identify which is which using quantile plots. Since you will need to draw quantile plots with respect to distributions other than the normal, it will be convenient to use a new function named `qqPlot` in the package `car`. You will need to install this package. If you are using RStudio, select the `Packages` tab on the panel on the right and then select the `Install` tab. Type `car` on the pop-up window and click install. After installing, you need to load the package using `library(car)`.

The function `qqPlot` has syntax

```
qqPlot(x, dist = 'weibull', shape = 2)
```

for plotting a quantile graph of vector `x` with respect to the Weibull distribution with shape parameter 2. The default distribution for `qqPlot` is the normal distribution. You can find more details in the help for `qqPlot`. By default, this function draws confidence bands which I find in many cases of little use, and in some cases misleading. If you don't want them in your graph, add `envelope = FALSE` in your call.

Explain clearly the reasons for your choices.

Solution

Start by reading the data and looking at the structure of the data set.

```
dataQ2 <- read.table('dataQ2', header = T)
str(dataQ2)
```

```
## 'data.frame':    20 obs. of  4 variables:
##  $ smpl1: num  1.796 2.033 0.896 0.635 1.833 ...
##  $ smpl2: num  13.9784 -0.0295 -2.8789 -4.0742 -0.6839 ...
##  $ smpl3: num  0.496 0.92 0.275 0.507 1.161 ...
##  $ smpl4: num  2.144 1.537 0.299 1.048 1.13 ...
```

We do quantile plots for all combinations of distributions and simulated samples. Since in this problem you know that each sample comes from a different distribution, you can either choose a distribution for each sample or a sample for each distribution. I will do this choosing a distribution for each sample.

Sample 1.

```
attach(dataQ2)
library(car)
par(mfrow = c(2,2))
qqPlot(smpl1, dist = 'cauchy', envelope = FALSE)
```

```
## [1] 12 19
```

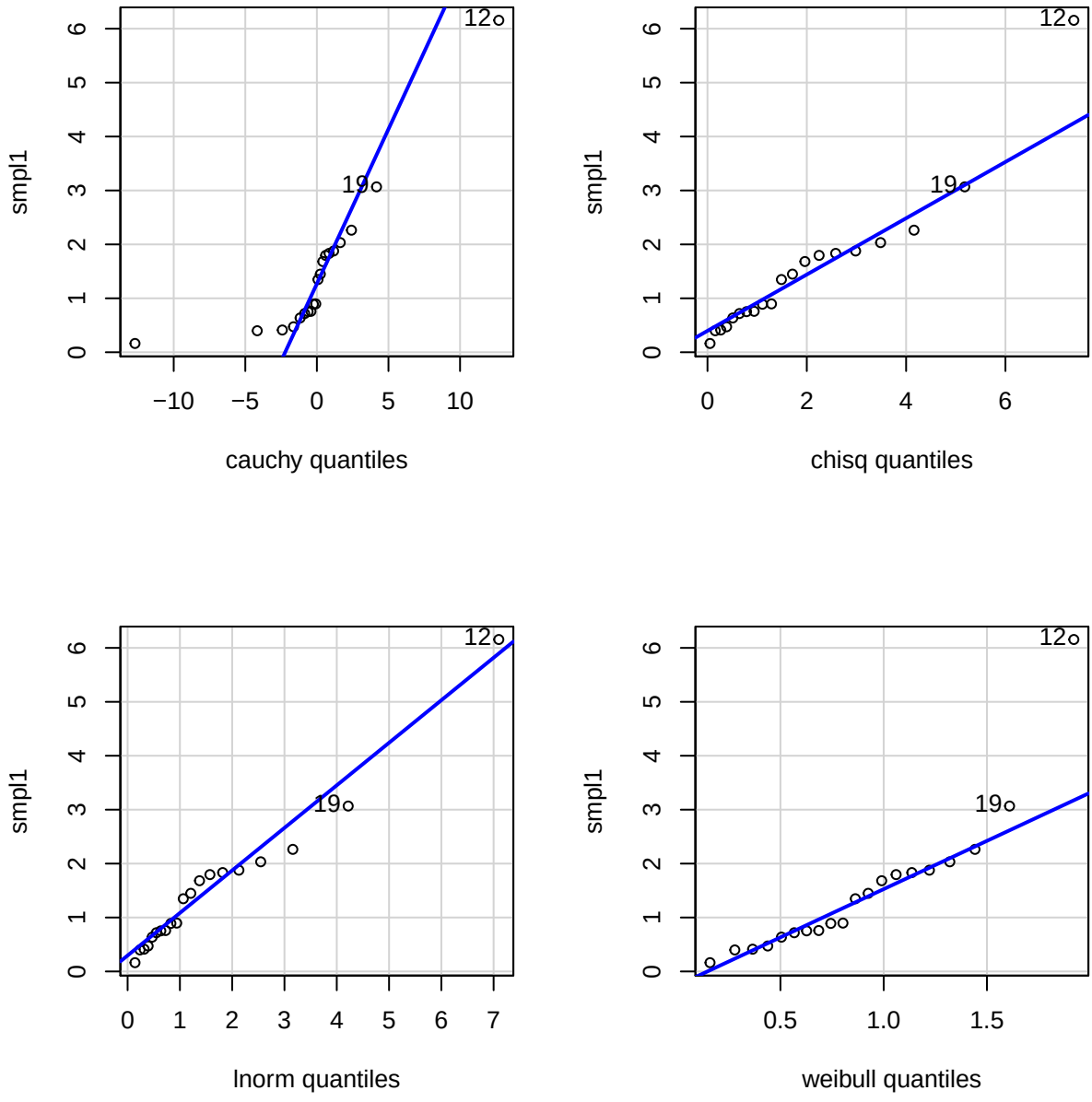
```
qqPlot(smpl1, dist = 'chisq', df = 2, envelope = FALSE)
```

```
## [1] 12 19
```

```
qqPlot(smpl1, dist = 'lnorm', envelope = FALSE)
```

```
## [1] 12 19
```

```
qqPlot(smpl1, dist = 'weibull', shape = 2, envelope = FALSE)
```



```
## [1] 12 19
```

In this case the best fit corresponds to the lognormal distribution. Observe also that in this sample all the values are positive (look at the y-axis in all the plots) and there are three distributions in the list that have only positive values: lognormal, Weibull, and Chi square.

Sample 2.

```
par(mfrow = c(2,2))
qqPlot(smpl2, dist = 'cauchy', envelope = FALSE)
```

```
## [1] 1 4
```

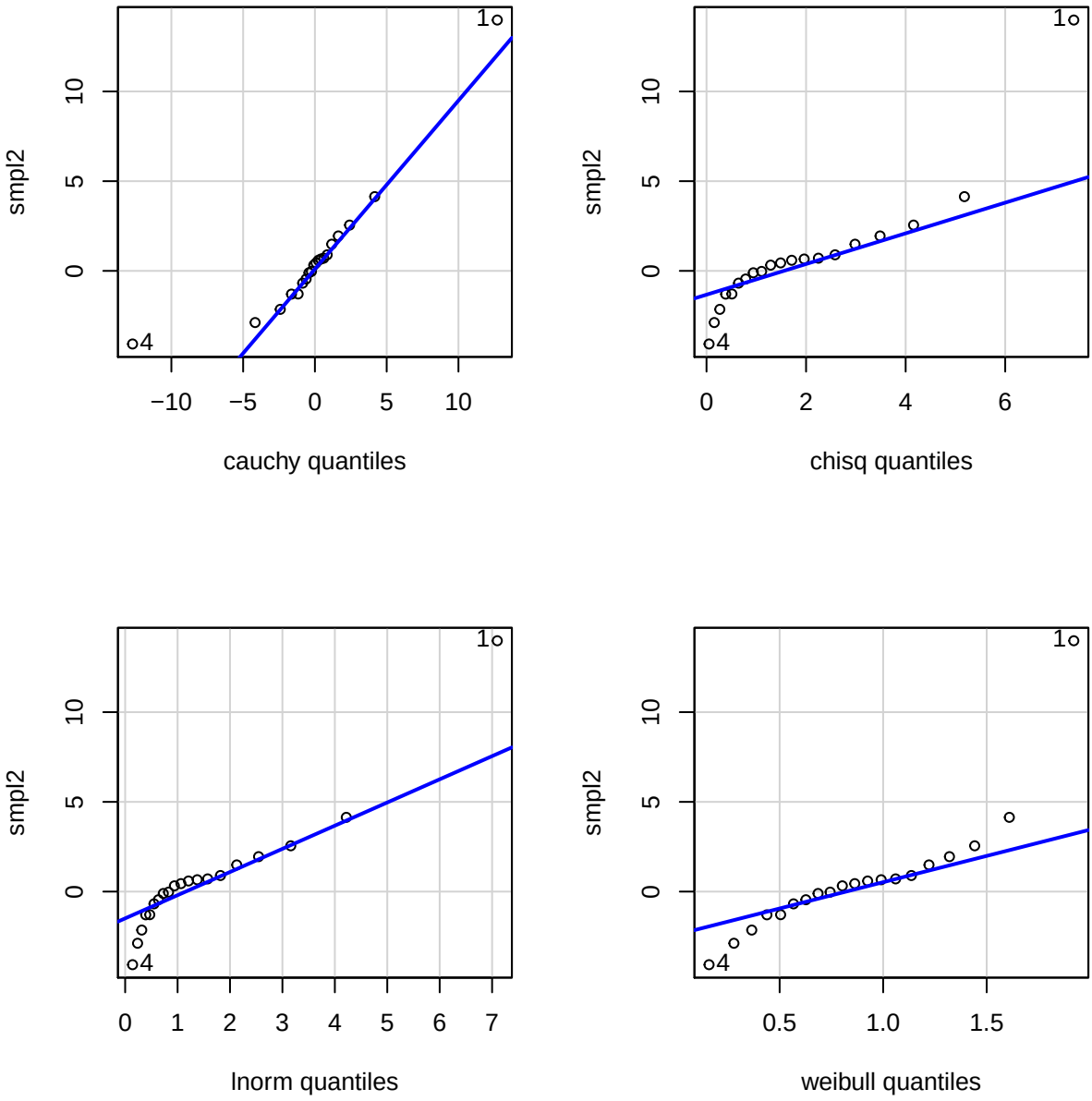
```
qqPlot(smpl2, dist = 'chisq', df = 2, envelope = FALSE)
```

```
## [1] 1 4
```

```
qqPlot(smpl2, dist = 'lnorm', envelope = FALSE)
```

```
## [1] 1 4
```

```
qqPlot(smpl2, dist = 'weibull', shape = 2, envelope = FALSE)
```




```
## [1] 1 4
```

In this case the fit to the Cauchy distribution is very good, except for a point, but all the other plots look much worse, so we identify this as the Cauchy sample. Observe that the sample has both positive and negative values, and the Cauchy distribution is the only option that has values in the real line.

Sample 3.

```
par(mfrow = c(2,2))
qqPlot(smpl3, dist = 'cauchy', envelope = FALSE)
```

```
## [1] 8 9
```

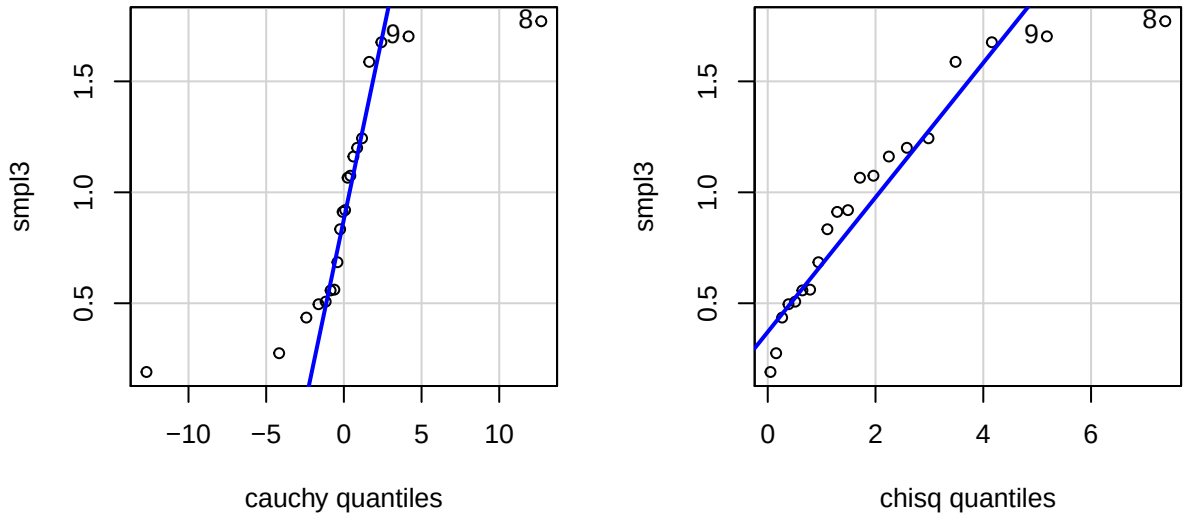
```
qqPlot(smpl3, dist = 'chisq', df = 2, envelope = FALSE)
```

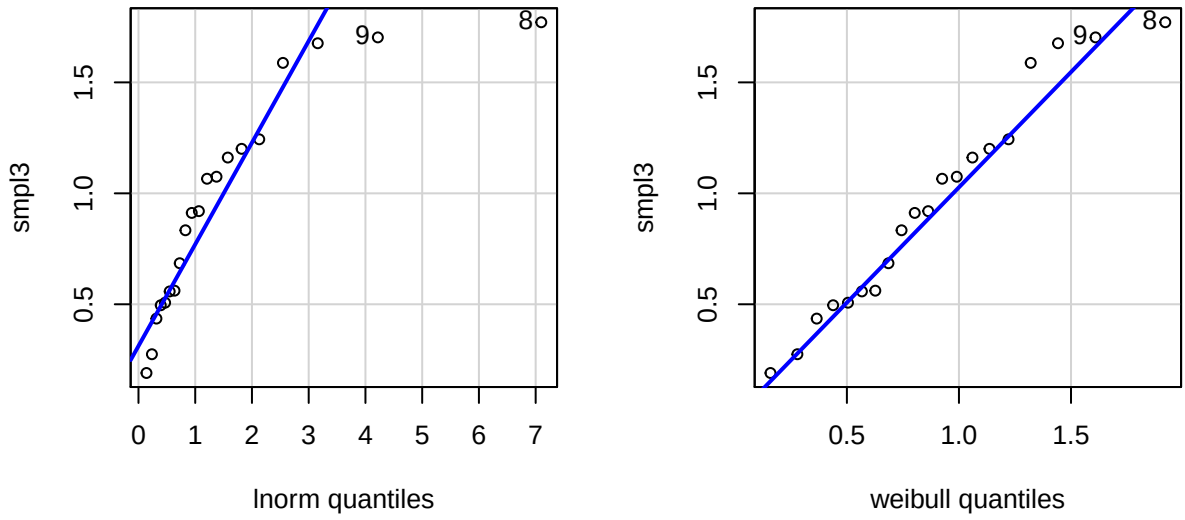
```
## [1] 8 9
```

```
qqPlot(smpl3, dist = 'lnorm', envelope = FALSE)
```

```
## [1] 8 9
```

```
qqPlot(smpl3, dist = 'weibull', shape = 2, envelope = FALSE)
```






```
## [1] 8 9
```

In this case the best fit corresponds to the Weibull distribution

Sample 2.

```
par(mfrow = c(2,2))
qqPlot(smpl4, dist = 'cauchy', envelope = FALSE)
```

```
## [1]  8 11
```

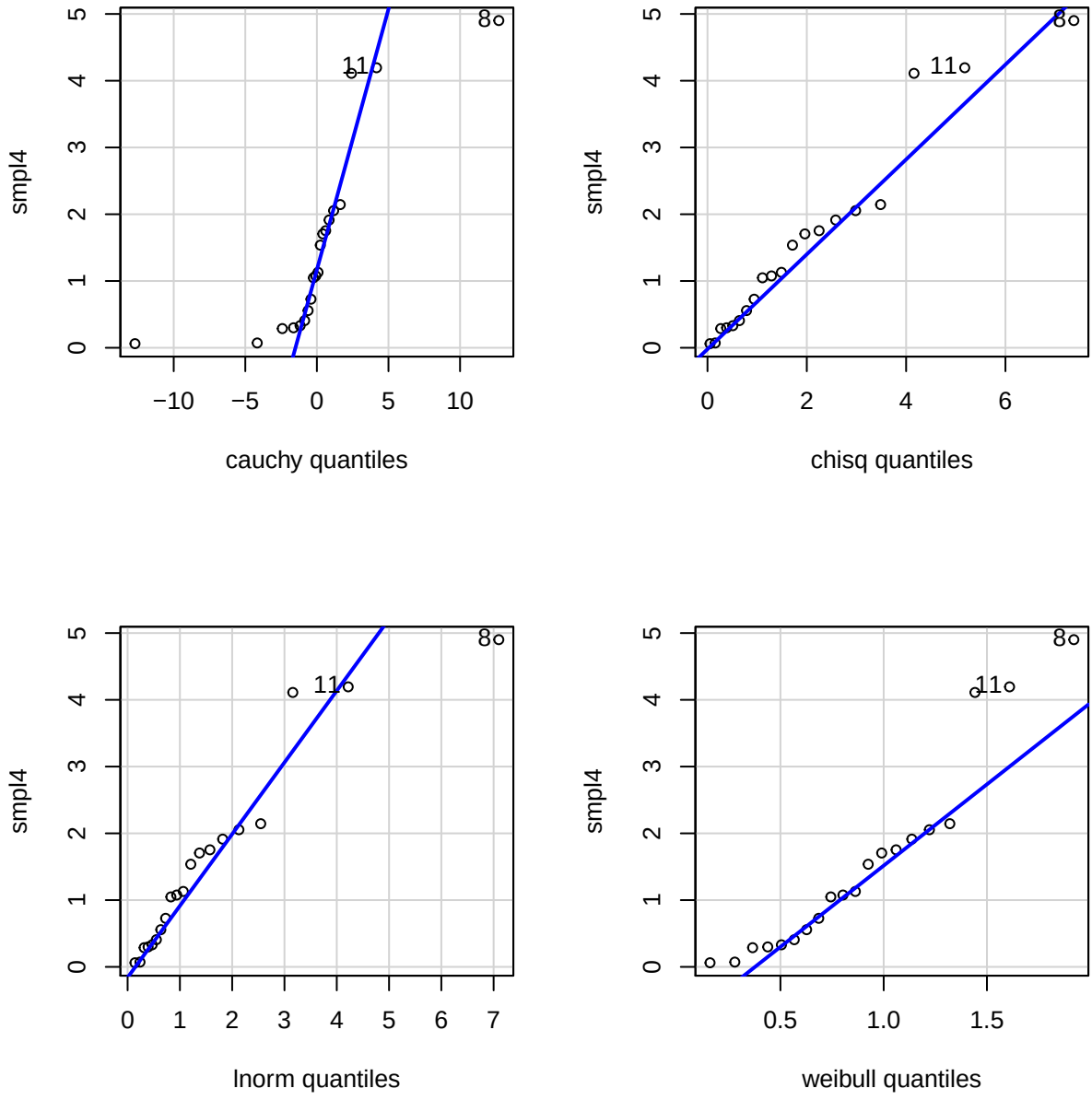
```
qqPlot(smpl4, dist = 'chisq', df = 2, envelope = FALSE)
```

```
## [1]  8 11
```

```
qqPlot(smpl4, dist = 'lnorm', envelope = FALSE)
```

```
## [1]  8 11
```

```
qqPlot(smpl4, dist = 'weibull', shape = 2, envelope = FALSE)
```




```
## [1]  8 11
```

For this remaining sample the best fit corresponds to the chi-square distribution which, fortunately, is the only remaining distribution.

Our classification is

Sample	Distribution
smpl1	Lognormal
smpl2	Cauchy
smpl3	Weibull
smpl4	Chi-square