02441 Applied Statistics and Statistical Software

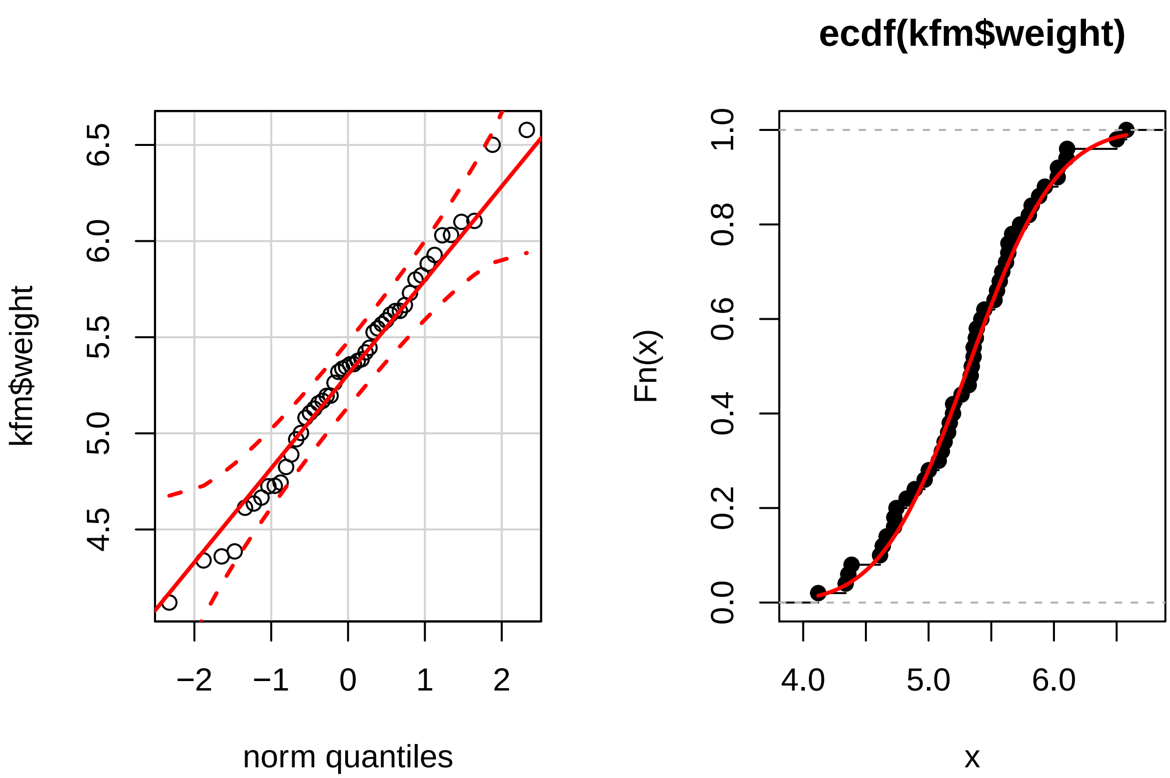
Exercise 4D - KFM

The dataset kfm contains measurements of the newborn babies, their mother and milk consumption

Variable name	Description
dl.milk	amount of breast milk (dl)
sex	gender of body
weight	baby weight (kg)
ml.suppl	amount of milk supplement (ml)
mat.weight	mothers weight (kg)
mat.height	mothers height (cm)

1. Make apropriate plots of the birthweight in order to check wether the weight is normally distributed

```
kfm <- read.table("kfm.txt", header = TRUE)
library(car)
par(mfrow=c(1,2))
qqPlot(kfm$weight)
plot(ecdf(kfm$weight))
lines(seq(min(kfm$weight), max(kfm$weight), length.out = 100),
      pnorm(seq(min(kfm$weight), max(kfm$weight), length.out = 100),
            mean(kfm$weight), sd(kfm$weight)), col = 'red', lw = 2)
```



2. Make a χ^2-test to test if the birthweights for the babies can be assumed normally distributed

To address this we want to get counts for well defined intervals. To do so use the histogram function `hist`. We can extract the breaks from the `hist` result and apply these to find the probabilities for the corresponding normal distribution. The breaks should include `-Inf` and `Inf` to include all probability mass.

```
# Get counts in weight bins using histogram
hist_weight <- hist(kfm$weight, plot = FALSE)
breaks <- hist_weight$breaks
counts <- hist_weight$counts
# Get expected counts according to normal distribtuion
prob <- pnorm(c(-Inf, breaks[-c(1, length(breaks))], Inf), mean(kfm$weight), sd(kfm$weight))
prob <- diff(prob)
(expected <- prob*length(kfm$weight))
```

```
## [1]  3.384451 10.641150 17.450426 13.172565  4.571175  0.780233
```

The breaks have to be adjusted because the expected counts in some bins are below 5.

```
# Get counts in weight bins using histogram
breaks <- c(4,4.75,5,5.25,5.5,5.75,6.0,7)
hist_weight <- hist(kfm$weight, breaks = breaks, plot = FALSE)
counts <- hist_weight$counts
# Get expected counts according to normal distribtuion
prob <- pnorm(c(-Inf, breaks[-c(1, length(breaks))], Inf), mean(kfm$weight), sd(kfm$weight))
prob <- diff(prob)
(expected <- prob*length(kfm$weight))
```

```
## [1] 7.490209 6.535392 8.480191 8.970235 7.735137 5.437427 5.351408
```

All expected counts are above 5. We can now calculate the χ^2 test statistic and the corresponding p-value.

```
chisq.test(counts, p = prob)
```

```
##
##  Chi-squared test for given probabilities
##
## data:  counts
## X-squared = 3.5643, df = 6, p-value = 0.7354
```

These results are not correct. We have estimated two additional parameters (mean and sd) and must therefore correct the degrees of freedom accordingly.

```
chi_statistic <- sum((counts-expected)^2/expected)
(p.value <- pchisq(chi_statistic, length(expected)-2-1, lower.tail = FALSE))
```

```
## [1] 0.4681656
```

Alternatively you can use a the `nortest` package specifically designed for this purpose. The p-value differs slightly from the above because of our customized breaks.

```
install.packages("nortest")
```

```
## Installing package into '/home/andba/R/x86_64-pc-linux-gnu-library/3.4'
## (as 'lib' is unspecified)
```

```r
library("nortest")
pearson.test(kfm$weight, n.classes = 6, adjust = TRUE)
```

```
##
##  Pearson chi-square normality test
##
## data:  kfm$weight
## P = 2.8, p-value = 0.4235
```