Exam 1

Part I: Estimating probabilities

Remember to load the `mosaic` package first:

```
library(mosaic)
```

```
## Warning: package 'mosaic' was built under R version 3.4.2
```

```
options(digits = 4)
library(pander)
```

chile referendum data

In this part we will use the dataset `chile`. Remember to read the description of the dataset as well as the Wikipedia entry about the background.

```
Chile <- read.table("http://asta.math.aau.dk/dan/static/datasets?file=Chile.dat", header=TRUE, quote="\
```

NB: This dataset has several missing values (`NA`). To remove these when you use `tally` you can add the argument `useNA = "no"`.

- Do a cross tabulation of the variables `vote` and `sex`.

1) Chile dataset has 2700 obs. After removing the missing values 2500 remain.
2) Cross tabulation in percentage

```
chile<-tally(~vote+sex,data=Chile, useNA="no")
n<-sum(chile)
n
```

```
## [1] 2532
```

```
chile<-tally(~vote+sex,data=Chile, useNA="no", format="percent")
chile
```

```
##     sex
## vote      F      M
##    A  4.107  3.278
##    N 14.336 20.774
##    U 14.297  8.926
##    Y 18.957 15.324
```

- Estimate the probability of `vote=N`.

```
sum(chile[2,1:2])
```

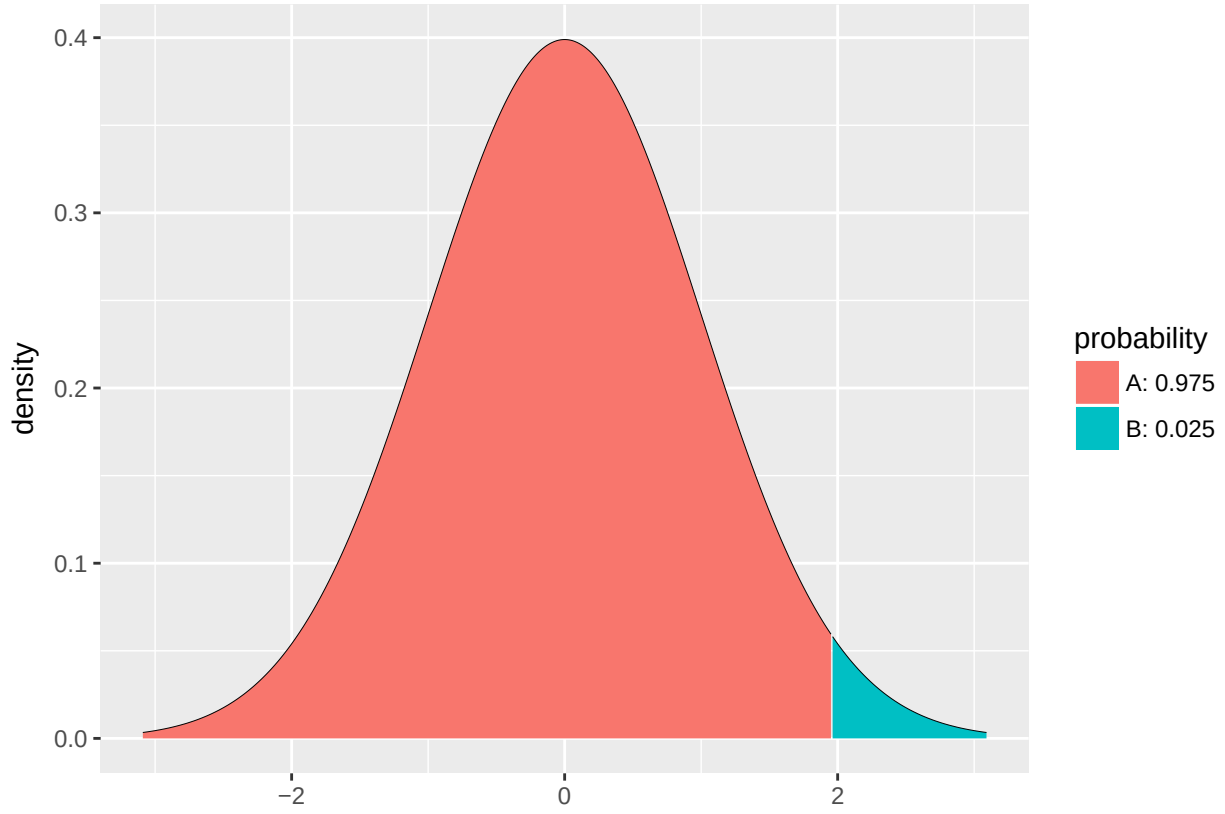
```
## [1] 35.11
```

```
pi<-14.336+20.774
pi
```

```
## [1] 35.11
```

- Make a 95% confidence interval for the probability of `vote=N`.

```
se=sqrt((pi/100*(1-pi/100))/n)
z = qdist("norm", 1-0.05/2) #Since qdist calculates the probability not taking into account the absolut
```




```
z
```

```
## [1] 1.96
```

```
piInt= pi+z*100*c(-se,se) #100 times z since we are giving the conf int in percentage.
piInt
```

```
## [1] 33.25 36.97
```

```
prop.test(~ vote, data = Chile, correct = FALSE, success= "N")
```

```
##
##  1-sample proportions test without continuity correction
##
## data:  Chile$vote  [with success = N]
## X-squared = 220, df = 1, p-value <2e-16
## alternative hypothesis: true p is not equal to 0.5
## 95 percent confidence interval:
##  0.3328 0.3699
## sample estimates:
##      p
## 0.3511
```

- Estimate the probability of `vote=N` given that `sex=F`.

```
tally(~vote+sex,data=Chile, useNA="no",format="percent",margin=1)
```

```
##        sex
## vote          F       M   Total
```

```
##   A       4.107   3.278   7.385
##   N      14.336  20.774  35.111
##   U      14.297   8.926  23.223
##   Y      18.957  15.324  34.281
##   Total  51.698  48.302 100.000
```

```
piCond<-14.336/(51.698)
piCond
```

```
## [1] 0.2773
```

- What would these probabilities satisfy if `vote` and `sex` were statistically independent?

  This fact would mean that P(`vote`) does not depend on P(`sex`) applying this: P(`vote`|P`sex`)=P(`vote`)*P(`sex`).

```
#P(`vote=N`)
pi
```

```
## [1] 35.11
```

```
#P(`sex=F`)
pif<-51.698
#P(P`vote`)*P(`sex`)
pi*0.01*pif*0.01
```

```
## [1] 0.1815
```

```
#Whereas P(P`vote`|P`sex`)
piCond
```

```
## [1] 0.2773
```

```
#So the vote is conditioned by the sex
```

Part II: Sampling distributions and the central limit theorem

This is a purely theoretical exercise where we investigate the random distribution of samples from a known population.

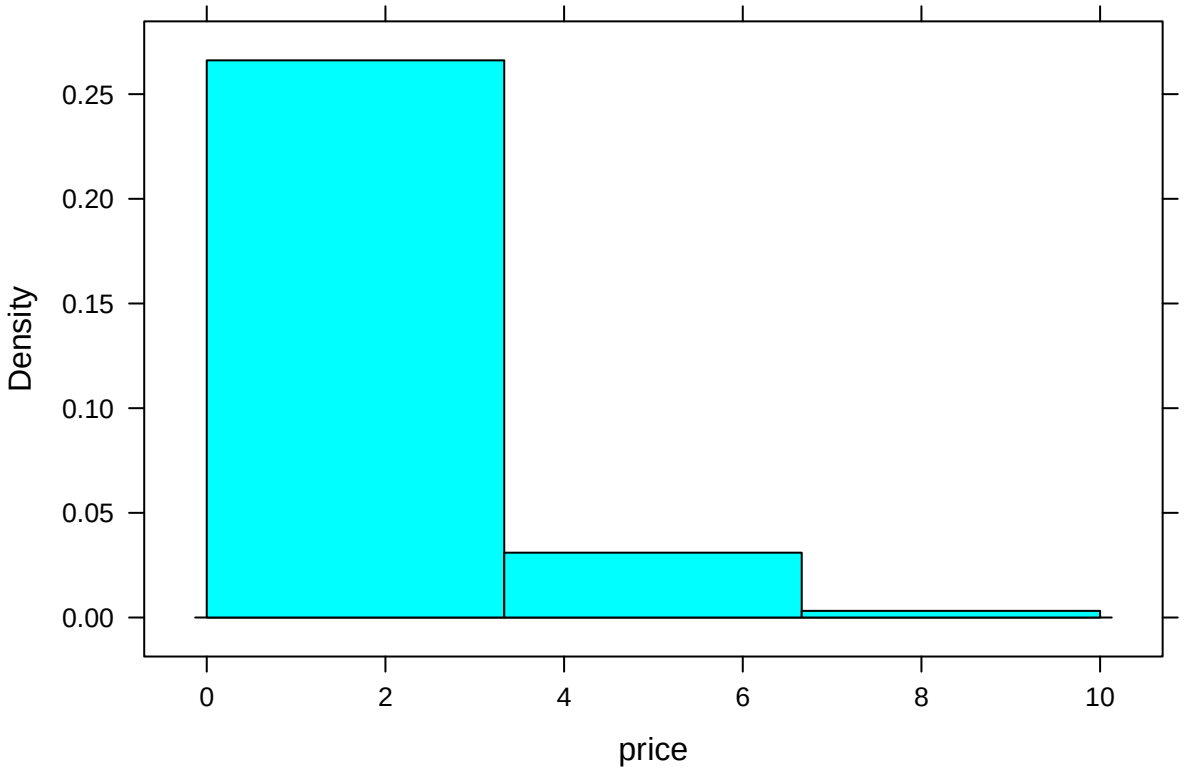
House prices in Denmark

The Danish real estate agency HOME has a database containing approximately 80.000 house prices for one-family houses under DKK 10 million for the period 2004-2016. The house prices (without all the additional information such as house size, address etc.) are available as a R data file `Home.RData` on the course webpage. If you download it you can load it using `load("Home.RData")` assuming you have saved it in the same directory as this Rmarkdown document. This will add the vector `price` to your work space. Alternatively, you can add it directly from the course website (this will download it every time you run the Rmarkdown document, so make sure you have a decent internet connection):

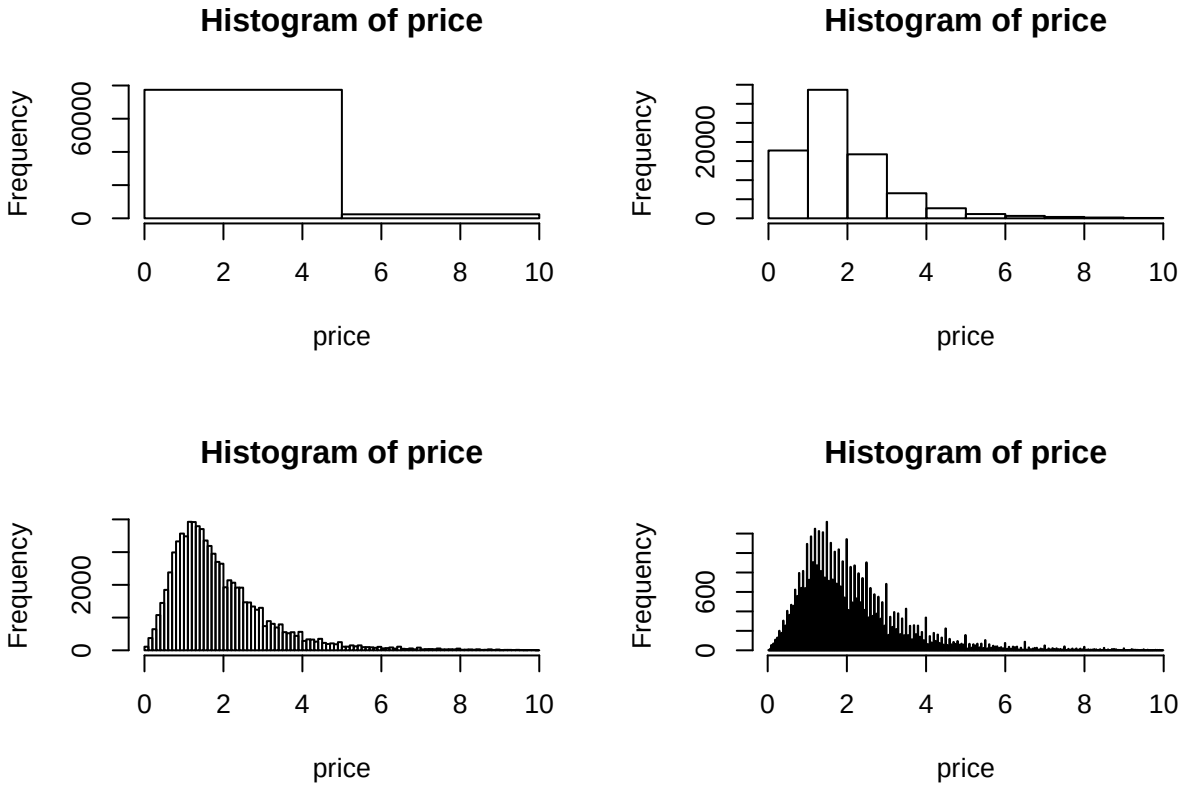
```
load(url("http://asta.math.aau.dk/dan/static/datasets?file=Home.RData"))
```

Make a histogram of all the house prices using a command like `histogram(price, breaks = 30)` inserted in a new code chunk (try to do experiments with the number of breaks):

```
#With my own deffinition for the breaks
histogram(price, breaks = c(0,3.33,6.66,10))
```




```
#Using the R default algorithm for making breaks
par(mfrow=c(2,2))
hist(price, breaks = 2)
hist(price, breaks = 10)
hist(price, breaks = 100)
hist(price, breaks = 1000)
```




```
#As we increase the number of breaks, the graph aproximates to a probability density function. Related
#The sum of the area of the bars is equal to 1 independently of the number of bars.
```

- Explain how a histogram is constructed.

To construct a histogram from a continuous variable the first is to split the data into intervals, called bins or breaks. In the example above, each bin contains the number of occurrences of dataset scores within that break. The y axis represents the frequency for each break. The sum of the area of all the bars has to be equal to 1.

- Does this histogram look like a normal distribution?

No. Although it has kind of a bell shape as the normal distribution, there is a notizable right skew. The normal distibution is symetrical around 0.

Since the median is lower than the mean, there are more samples to the left of the mean than to the right of the mean.

```
median(price)
```

```
## [1] 1.6
```

```
mean(price)
```

```
## [1] 1.929
```

In this database (our population) the mean price is 1.929 and the standard deviation is 1.2744. In many cases access to such databases is restrictive and in the following we imagine that we are only allowed access to a random sample of 40 prices and the mean of this sample will be denoted `y_bar`.

Before obtaining this sample we will use the Central Limit Theorem (CLT) to predict the distribution of `y_bar`:

Central Limit Theorem:

For random sampling with a large sample size n, the sampling distribution of the sample mean y is approximately a normal distribution.

- What is the expected value of `y_bar`?

```
mean(price)
```

```
## [1] 1.929
```

- What is the standard deviation of `y_bar` (also called the standard error)?

```
sd(price)/sqrt(40)
```

```
## [1] 0.2015
```

- What is the approximate distribution of `y_bar`? N(1.929,0.2015)

Now make a random sample of 40 house prices and calculate the sample mean:

```
y <- sample(price, 40)
#sample give us random values from a vector. With n we specify the number of values we want to take
mean(y)
```

```
## [1] 1.866
```

```
#Repeat this command a few times. Is each mean price close to what you expected?
y <- sample(price, 40)
mean(y)
```

```
## [1] 1.956
```

```
y <- sample(price, 40)
mean(y)
```

```
## [1] 1.761
```

```
#Yes they are close to the estimated mean.
```

Use `replicate` to repeat the sampling 200 times and save each mean value in the vector `y_bar`:

```
y_bar <- replicate(200, mean(sample(price, 40)))
```

Calculate the mean and standard deviation of the values in `y_bar`.

```
mean(y_bar)
```

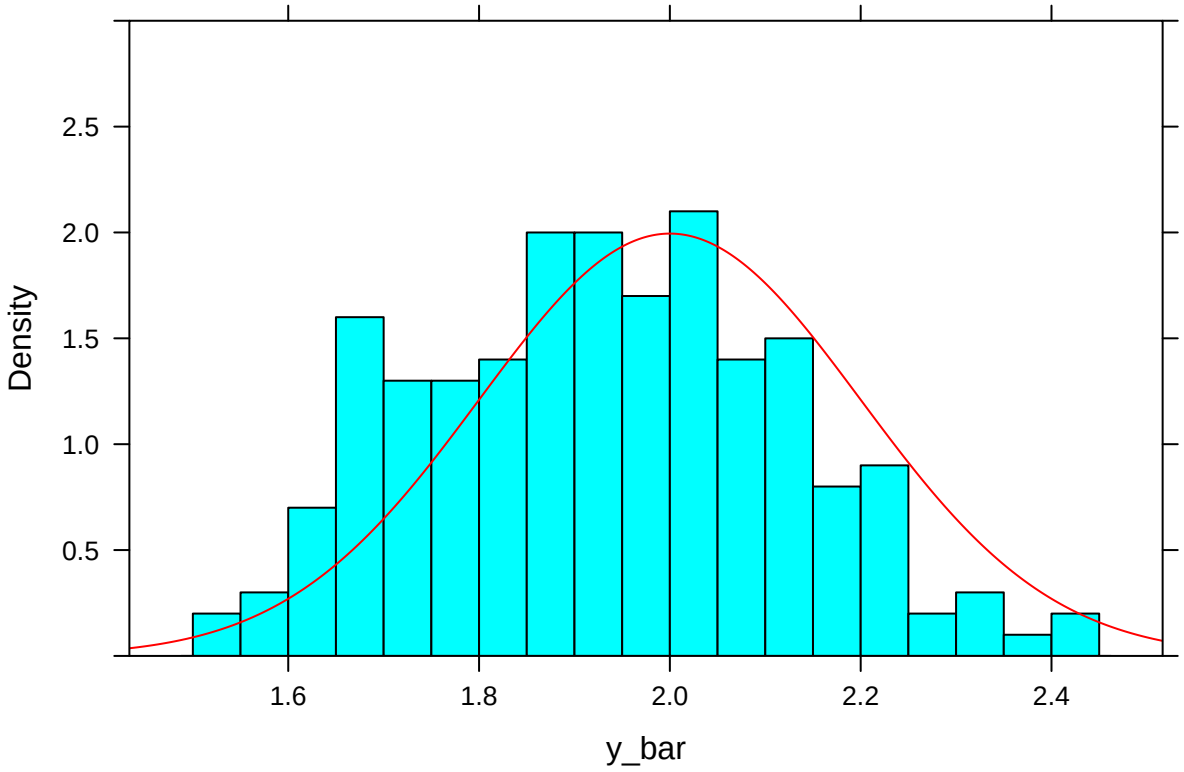
```
## [1] 1.933
```

```
sd(y_bar)
```

```
## [1] 0.1899
```

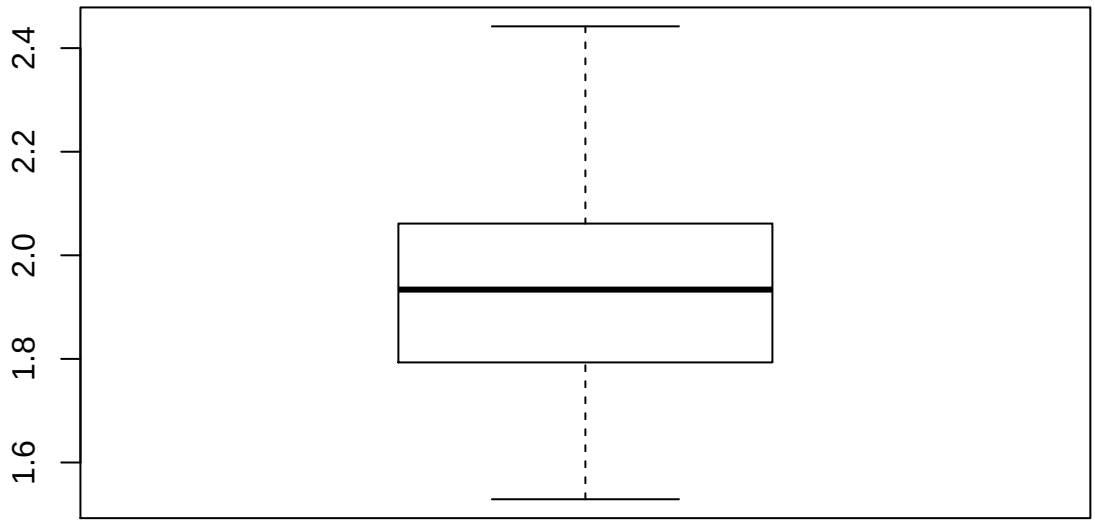
- How do they match with what you expected? They match perfectly with the CLT prediction as the number of samples increases enough (200 times).
- Make a histogram of the values in `y_bar` and add the density curve for the approximate distribution you predicted previously using `plotDist` with the argument `add = TRUE`. For example if you predicted a normal distribution with mean 2 and standard deviation 0.2:

```
histogram(y_bar, breaks = 15, type="density",ylim=c(0,3))
plotDist("norm", mean = 2, sd = 0.2, add = TRUE, col = "red", ylim=c(0,3))
```



- Make a boxplot of `y_bar` and explain how a boxplot is constructed.

```
boxplot(y_bar)
```



1) Calculate the median, lower and upper quartiles.
2) Draws a line at the median and a box between the upper and lower quartiles.
3) Calculate interquartile range (upper quartile - lower quartile).
4) Calculate the following values:

- L1 = lower quartile - 1.5·IQR
- L2 = lower quartile - 3.0·IQR
- U1 = upper quartile + 1.5·IQR
- U2 = upper quartile + 3.0·IQR

5. Draw a line for lower quartile limit (L1). Similarly, draw a line for upper quartile limit (U1).
6. Measurements between L2 and L1 / between U1 and U2 are drawn as circles. Measurements smaller than L2 / larger that U2 are marked by a x. # Part III: Theoretical boxplot for a normal distribution

Finally, consider the theoretical boxplot of a general normal distribution with mean μ and standard deviation σ, and find the probability of being an outlier according to the 1.5·IQR criterion:

- First find the z-score of the lower/upper quartile. I.e. the value of z such that $\mu \pm z\sigma$ is the lower/upper quartile.

q1=$\mu + z * sigma$ q3=$\mu - z * sigma$

```
q1 <- qdist("norm", 0.25, plot = FALSE)
q3 <-qdist("norm", 0.75, plot = FALSE)
q1
```

```
## [1] -0.6745
```

```
q3
```

```
## [1] 0.6745
```

- Use this to find the IQR (expressed in terms of σ). The IQR then is $(\mu + 0.6745\sigma) - (\mu - 0.6745\sigma) = 2 \cdot 0.6745\sigma = 1.349\sigma$
- Now find the z-score of the maximal extent of the whisker. I.e. the value of z such that $\mu \pm z\sigma$ is the endpoint of lower/upper whisker.

```
Ll = -0.6745-1.5*1.349
Ul = 0.6745+1.5*1.349
Ll
```

```
## [1] -2.698
```

```
Ul
```

```
## [1] 2.698
```

- Find the probability of being an outlier.

```
PLl = pdist("norm", q = Ll, mean = 0, sd = 1, plot = FALSE)
PUl = pdist("norm", q = Ul, mean = 0, sd = 1, plot = FALSE)
PIN = PUl-PLl
POUT = 1-PIN
POUT
```

```
## [1] 0.006976
```