Session 5. Introduction to statistical inference

Example 1

- Read the Osteoporosis dataset and turn factors into variables automatically with Rbase function `read.delim`
- Take a sample of size 100 from the original file. Call it 'osteo100' and work with this file from now on.
- Compute the mean value of the variable containing bone density values `BUA`
- Split the computation between all subgroups from variable `classific` and variable `menop`.
- Compute the percentage of menopausic women from variable `menop`.

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
# Read data
osteoporosis <- read.delim2("datasets/osteoporosis.csv", stringsAsFactors=TRUE)
# Take subsample
osteo100 <- sample_n(osteoporosis, 100)
# mean bone density
buaMean <- mean(osteo100$bua)
print(buaMean)
```

```
## [1] 74.08
```

```
# Mean bone density ny groups
osteo100 %>%
  group_by(menop) %>%
  summarize(m = mean(bua))
```

```
## # A tibble: 2 x 2
##   menop     m
##   <fct> <dbl>
## 1 NO     83.4
## 2 SI     69.5
```

```
# Proportion of menop women (Proportion  is a mean of 0-1 values)
mean(ifelse(osteo100$menop=="SI",1,0))
```

```
## [1] 0.67
```

Exercise 1

- Read the diabetes dataset. Convert characters into factors before continuing.
- Provide an estimate of
  - The distribution of a numerical variable.
  - a proportion of at least one categorical variable and
  - the mean value of at least one numerical variable.
- Could you have used different estimators?
- How would you decide?

First we read data and recode character values into factors.

```
library(readxl)
library(dplyr)
library(magrittr)
diabetes <- read_excel("datasets/diabetes.xls")
sapply(diabetes, class)
```

```
##    numpacie        mort    tempsviu        edat         bmi    edatdiag
##   "numeric" "character"   "numeric"   "numeric"   "numeric"   "numeric"
##       tabac         sbp         dbp         ecg         chd
## "character"   "numeric"   "numeric" "character" "character"
```

```
diabetes_factor <- diabetes %>%
  mutate_if(sapply(diabetes, is.character), as.factor) %>%
  select (-numpacie)
sapply(diabetes_factor, class)
```

```
##      mort  tempsviu      edat       bmi  edatdiag     tabac       sbp       dbp
##  "factor" "numeric" "numeric" "numeric" "numeric"  "factor" "numeric" "numeric"
##       ecg       chd
##  "factor"  "factor"
```

Next provide a quick summary of each variable

```
summary(diabetes_factor)
```

```
##      mort        tempsviu          edat            bmi           edatdiag
##  Muerto: 25   Min.   : 0.00   Min.   :31.00   Min.   :18.20   Min.   :26.00
##  Vivo  :124   1st Qu.: 7.30   1st Qu.:43.00   1st Qu.:26.60   1st Qu.:38.00
##               Median :11.60   Median :50.00   Median :31.20   Median :45.00
##               Mean   :10.52   Mean   :52.17   Mean   :31.78   Mean   :45.99
##               3rd Qu.:13.90   3rd Qu.:60.00   3rd Qu.:35.20   3rd Qu.:53.00
##               Max.   :16.90   Max.   :86.00   Max.   :59.70   Max.   :81.00
##         tabac          sbp            dbp                ecg      chd
##  Ex fumador:41   Min.   : 98.0   Min.   : 58.00   Anormal : 11   No:99
##  Fumador   :51   1st Qu.:124.0   1st Qu.: 74.00   Frontera: 27   Si:50
##  No fumador:57   Median :138.0   Median : 80.00   Normal  :111
##                  Mean   :139.1   Mean   : 90.04
##                  3rd Qu.:152.0   3rd Qu.: 88.00
##                  Max.   :222.0   Max.   :862.00
```

Plotting all variables with an instruction is a bit tricky. May be easier to plot separately numerical and categorical variables.

```
library(ggplot2)
library(tidyr)
```

```
##
## Attaching package: 'tidyr'

## The following object is masked from 'package:magrittr':
##
##     extract
```

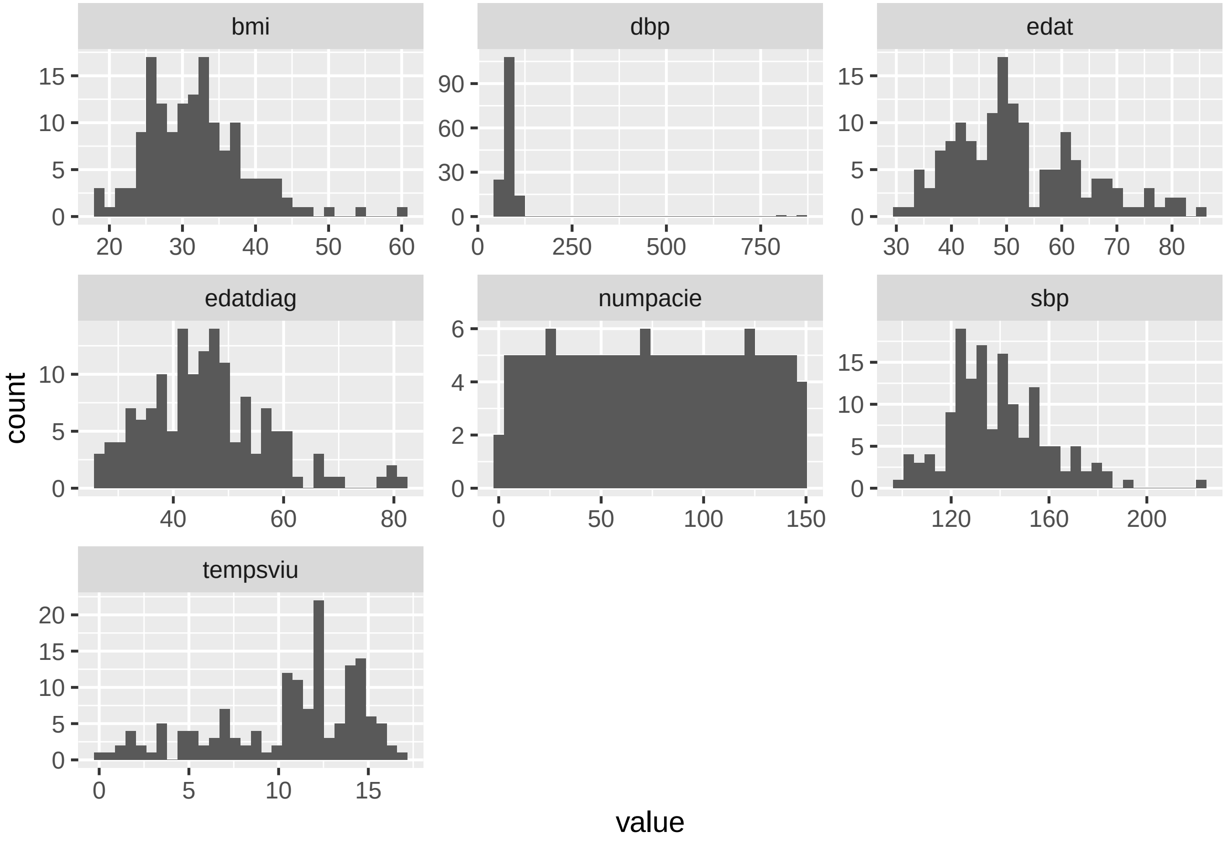
```
library(purrr)
```

```
##
## Attaching package: 'purrr'

## The following object is masked from 'package:magrittr':
##
##     set_names
```

```
diabetes %>%
  keep(is.numeric) %>%
  gather() %>%
  ggplot(aes(value)) +
    facet_wrap(~ key, scales = "free") +
    geom_histogram()
```

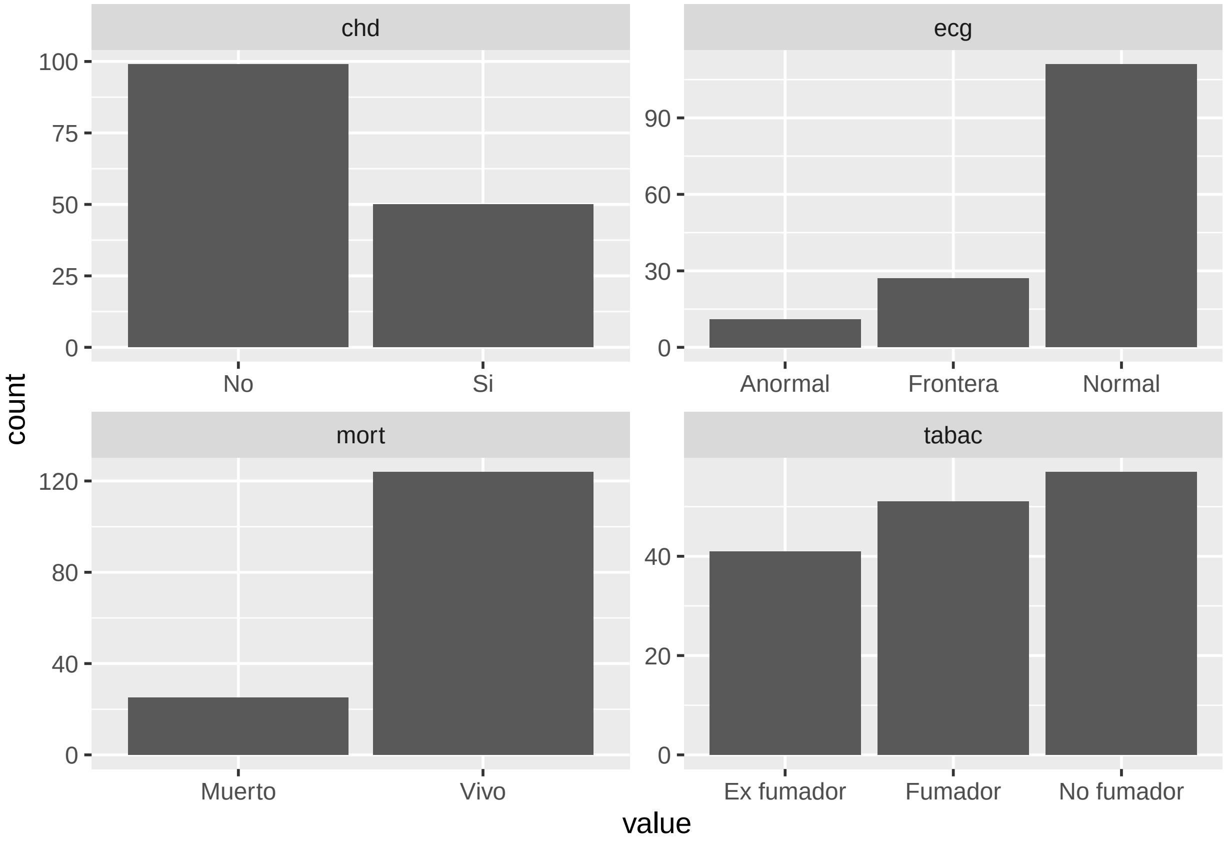
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Proceed similarly with categorical variables

```
diabetes %>%
  keep(is.character) %>%
```

```r
  gather() %>%
  ggplot(aes(value)) +
    facet_wrap(~ key, scales = "free") +
    geom_bar()
```



You may notice -or not- that the dataset has some outlier values.

Before removing them consider estimating the mean nvalue of SBP and DBP with distinct estimators

```r
with(diabetes_factor, {
    print("DBP")
    show(summary(dbp))
    print("SBP")
    show(summary(sbp))
  }
  )
```

```
## [1] "DBP"
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   58.00   74.00   80.00   90.04   88.00  862.00
## [1] "SBP"
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    98.0   124.0   138.0   139.1   152.0   222.0
```

What is prefereable to estimate the mean SBP or DBP?

Example 2. Computing Confidence Intervals with R (2)

```
t.test(osteo100[["bua"]])
```

```
##
##  One Sample t-test
##
## data:  osteo100[["bua"]]
## t = 41.62, df = 99, p-value < 2.2e-16
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  70.54831 77.61169
## sample estimates:
## mean of x
##     74.08
```

Example 2 . Computing Confidence Intervals with R (3)

```
cntMenop <- table(osteo100[["menop"]])["SI"]
ssize <- length(osteo100[["menop"]])
prop.test (x=cntMenop, n=ssize)
```

```
##
##  1-sample proportions test with continuity correction
##
## data:  cntMenop out of ssize, null probability 0.5
## X-squared = 10.89, df = 1, p-value = 0.0009668
## alternative hypothesis: true p is not equal to 0.5
## 95 percent confidence interval:
##  0.5679099 0.7588442
## sample estimates:
##    p
## 0.67
```

Exercise 2.1 Computing Confidence intervals

- Read the file "osteoporosis.csv" into a dataset and call it "osteoporosis"
- Compute confidence intervals for the BUA mean and for the percentage of menopausic women with **all the individuals in the dataset**.
- Compare these confidence intervals with those that you obtained in example 2. How do they differ?

La solución en el tema 6

Exercise 2.2 Computing Confidence intervals

- Read the diabetes dataset. Convert characters into factors before continuing.

```
library(readxl)
library(dplyr)
library(magrittr)
```

```r
diabetes <- read_excel("datasets/diabetes.xls")
sapply(diabetes, class)
```

```
##    numpacie        mort    tempsviu        edat         bmi    edatdiag 
##   "numeric" "character"   "numeric"   "numeric"   "numeric"   "numeric" 
##       tabac         sbp         dbp         ecg         chd 
## "character"   "numeric"   "numeric" "character" "character" 
```

```r
diabetes_factor <- diabetes %>%
  mutate_if(sapply(diabetes, is.character), as.factor) %>%
  select (-numpacie)
sapply(diabetes_factor, class)
```

```
##      mort  tempsviu      edat       bmi  edatdiag     tabac       sbp       dbp 
##  "factor" "numeric" "numeric" "numeric" "numeric"  "factor" "numeric" "numeric" 
##       ecg       chd 
##  "factor"  "factor" 
```

- Provide a confidence interval for:
    - a proportion of at least one categorical variable and
    - the mean value of at least one numerical variable.

```r
cnt <- table(diabetes[["mort"]])["Muerto"]
ssize <- length(diabetes[["mort"]])
prop.test (x=cnt, n=ssize)
```

```
## 
##  1-sample proportions test with continuity correction
## 
## data:  cnt out of ssize, null probability 0.5
## X-squared = 64.456, df = 1, p-value = 9.869e-16
## alternative hypothesis: true p is not equal to 0.5
## 95 percent confidence interval:
##  0.1134978 0.2396854
## sample estimates:
##         p 
## 0.1677852 
```

```r
t.test(diabetes[["edat"]])
```

```
## 
##  One Sample t-test
## 
## data:  diabetes[["edat"]]
## t = 54.09, df = 148, p-value < 2.2e-16
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  50.26188 54.07370
## sample estimates:
## mean of x 
##  52.16779 
```

- How would you find alternative approaches to compute these confidence intervals?
    - An option is to apply formulas directly, calculating from:

$$\bar{x} \pm t_{\alpha/2}\frac{s}{\sqrt{n}}$$

- Why would you want to do such a thing?

```
mostra<-diabetes[["edat"]]

m<-mean(mostra)  # Calculate mean

sd<-sd(mostra)  # Calculate standard deviation
se<-sd/sqrt(length(mostra)) # Calculate standard Error
li<- m-qt(.975,length(mostra)-1)*se # Calculate 95% CI lower bound

ls<- m+qt(.975,length(mostra)-1)*se # Calculate 95%CI upper bound


cat("Mean=",m,"\n")
```

```
## Mean= 52.16779
```

```
cat("Standard deviation=",sd,"\n")
```

```
## Standard deviation= 11.77285
```

```
cat("Standard error=",se,"\n")
```

```
## Standard error= 0.9644696
```

```
cat("95% Confidence interval=(",li,";",ls,")","\n")
```

```
## 95% Confidence interval=( 50.26188 ; 54.0737 )
```

- An *approximate* confidence interval for proportions can also be computed using a normal approximation such as:

$$\hat{p} \pm z_{1-\alpha/2}\sqrt{\frac{\hat{p}(1-\hat{p})}{n}}$$

```
cnt <- table(diabetes[["mort"]])["Muerto"]
ssize <- length(diabetes[["mort"]])
p<-cnt/ssize
n<-ssize
z<-qnorm(.975)
ee<-sqrt((p*(1-p))/n)
lowerli<- p-z*ee
upperli<- p+z*ee

cat("95% confidence interval for  ", p ,"=(",lowerli,";",upperli,")","\n")
```

```
## 95% confidence interval for   0.1677852 =( 0.1077855 ; 0.227785 )
```

Example 3. Sample size calculation

- Using the osteoporosis dataset, assume that the standard deviation is a good aproximation to σ.
- Find the sample size needed to achieve a margin of error equal to 5 with a 95% confidence interval.
- This can be computed with distinct packages.
- An option is the `sample.size.mean`function from the `samplingbook` package.
- TYpe `? sample.size.mean` to learn about it

```
library(samplingbook)
```

```
## Loading required package: pps

## Loading required package: sampling

## Loading required package: survey

## Loading required package: grid

## Loading required package: Matrix

##
## Attaching package: 'Matrix'

## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack

## Loading required package: survival

##
## Attaching package: 'survival'

## The following objects are masked from 'package:sampling':
##
##     cluster, strata

##
## Attaching package: 'survey'

## The following object is masked from 'package:graphics':
##
##     dotchart
```

```
osteoporosis <- read.delim2("datasets/osteoporosis.csv", stringsAsFactors=TRUE)
sdbua<-sd(osteoporosis$bua)
merror<-5
sample.size.mean(merror,sdbua)
```

```
##
## sample.size.mean object: Sample size for mean estimate
## Without finite population correction: N=Inf, precision e=5 and standard deviation S=16.8093
##
## Sample size needed: 44
```

Exercise 3. Sample size calculation

- The sample size formula for proportions is:

$$n = \frac{\hat{p}(1-\hat{p})z^2_{1-\alpha/2}}{\Delta^2},$$

where Δ is the margin error, that is the maximum expected difference between the true value and its estimation, that one expects to have with a probability of, at least $1-\alpha$.

- Write a function to compute the sample size for proportions in the worst case (p=q=0.5) or assuming p is known.

- Using a 50% planned proportion estimate, find the sample size needed to achieve 5 margin of error for a survey at 95 confidence level.

- How would this result change if we are told that a pilot study suggests that $p = 10\%$?

```
alpha<-1-.95
z<-qnorm(1-alpha/2)
merror<-0.05
p<-0.5 # Worst proportion
nsample<- (p*(1-p) * z^2)/0.05^2
cat("Sample size for 95% CI and 5% margin error s ",round(nsample))
```

```
## Sample size for 95% CI and 5% margin error s  384
```

```
p<-.1 # Worst proportion
nsample<- (p*(1-p) * z^2)/0.05^2
cat("Sample size for 95% CI and 10% margin error s ",round(nsample))
```

```
## Sample size for 95% CI and 10% margin error s  138
```

```
t.test(osteoporosis$bua,mu=72.5
       )
```

```
##
##  One Sample t-test
##
## data:  osteoporosis$bua
## t = 1.4994, df = 999, p-value = 0.1341
## alternative hypothesis: true mean is not equal to 72.5
## 95 percent confidence interval:
##  72.2539 74.3401
## sample estimates:
## mean of x
##    73.297
```