R for Data Science (IV): Essential Statistics with R

Alex Sanchez, Miriam Mota, Ricardo Gonzalo & Mireia Ferrer

Statistics and Bioinformatics Unit. Vall d'Hebron Institut de Recerca

Outline: Essential Statistics with R*

- Descriptive Statistics
  - Numerical summaries
  - Graphical exploration
- Statistical test
  - Continuous variables
  - Discrete variables
- BONUS: ANOVA and Linear Models

Based on this Course: [*BIMS 8382, University of Virginia School of Medicine (USA)*] (https://bioconnector.github.io/workshops/index.html).

What this class is *not*

- This is not a statistics course. Not covering:
  - Underlying mathematical motivation
  - How to choose the correct statistical procedure
  - Model assumptions
  - Interpreting every aspect of model output

What packages we will use today?

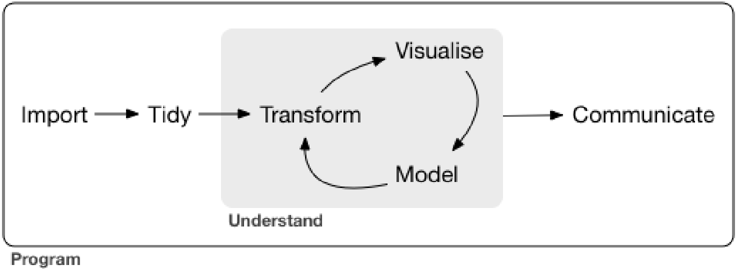
Please be sure you have the following packages installed:

- **dplyr** - subletting, sorting, transforming variables, grouping
- **ggplot2** - system for creating graphics
- **readxl** - reading .xls files

```
# install.packages("dplyr", dependencies = TRUE)
# install.packages("ggplot2", dependencies = TRUE)
# install.packages("readxl", dependencies = TRUE)

library(dplyr)
library(ggplot2)
library(readxl)
```

The Data Science Approach in R



Getting started

Getting started (I)

1. Load dataset: today we will continue working with *diabetes* dataset:

```
diab <- read_excel("datasets/diabetes_mod.xls")
```

2. Check if we have loaded it correctly:

```
diab[1:4, 1:8]
```

```
## # A tibble: 4 x 8
##   numpacie mort  tempsviu  edat   bmi edatdiag tabac        sbp
##      <dbl> <chr>    <dbl> <dbl> <dbl>    <dbl> <chr>      <dbl>
## 1        1 Vivo      12.4    44  34.2       41 No fumador   132
## 2        2 Vivo      12.4    49  32.6       48 Fumador      130
## 3        3 Vivo       9.6    49  22         35 Fumador      108
## 4        4 Vivo       7.2    47  37.9       45 No fumador   128
```

Getting started (II)

Some useful functions in `R` to check a dataframe:

- Content
```
- head(name of dataframe): shows the first few rows
- tail(): shows the last few rows
```
- Size
```
- dim(): returns the number of rows and the number of c
- nrow(): returns the number of rows
- ncol(): returns the number of columns
```
- Summary
```
- colnames() or names(): returns the column names
- glimpse(): returns a glimpse of your data, telling
             you structure, class, length and content
```

Getting started (III)

```
head(diab)
```

```
## # A tibble: 6 x 11
##   numpacie mort  tempsviu  edat   bmi edatdiag tabac   sbp   dbp ecg  
##      <dbl> <chr>    <dbl> <dbl> <dbl>    <dbl> <chr> <dbl> <dbl> <chr>
## 1        1 Vivo      12.4    44  34.2       41 No f~   132    96 Norm~
## 2        2 Vivo      12.4    49  32.6       48 Fuma~   130    72 Norm~
## 3        3 Vivo       9.6    49  22         35 Fuma~   108    58 Norm~
## 4        4 Vivo       7.2    47  37.9       45 No f~   128    76 Fron~
## 5        5 Vivo      14.1    43  42.2       42 Fuma~   142    80 Norm~
## 6        6 Vivo      14.1    47  33.1       44 No f~   156    94 Norm~
## # ... with 1 more variable: chd <chr>
```

Getting started (IV)

```
dim(diab)
```

```
## [1] 149  11
```

```
nrow(diab)
```

```
## [1] 149
```

```
colnames(diab)
```

```
##  [1] "numpacie" "mort"     "tempsviu" "edat"     "bmi"      "edatdia
##  [7] "tabac"    "sbp"      "dbp"      "ecg"      "chd"
```

Getting started (IV)

```r
glimpse(diab)
```

```
## Observations: 149
## Variables: 11
## $ numpacie <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16...
## $ mort     <chr> "Vivo", "Vivo", "Vivo", "Vivo", "Vivo", "Vivo", "Vivo...
## $ tempsviu <dbl> 12.4, 12.4, 9.6, 7.2, 14.1, 14.1, 12.4, 14.2, 12.4, 1...
## $ edat     <dbl> 44, 49, 49, 47, 43, 47, 50, 36, 50, 49, 50, 54, 42, 4...
## $ bmi      <dbl> 34.2, 32.6, 22.0, 37.9, 42.2, 33.1, 36.5, 38.5, 41.5,...
## $ edatdiag <dbl> 41, 48, 35, 45, 42, 44, 48, NA, 47, 45, 48, 43, 36, 4...
## $ tabac    <chr> "No fumador", "Fumador", "Fumador", "No fumador", "Fu...
## $ sbp      <dbl> 132, 130, 108, 128, 142, 156, 140, 144, 134, 102, 142...
## $ dbp      <dbl> 96, 72, 58, 76, 80, 94, 86, 88, 78, 68, 84, 74, 86, 5...
## $ ecg      <chr> "Normal", "Normal", "Normal", "Frontera", "Normal", "...
## $ chd      <chr> "No", "No", "Si", "Si", "No", "No", "Si", "No", "Si",...
```

Changing *characters (chr)* to *factors (Factor)*

Use dplyr function `mutate_if` can do it easily:

```
diab <- diab %>% mutate_if(is.character, as.factor)

glimpse(diab)
```

```
## Observations: 149
## Variables: 11
## $ numpacie <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16...
## $ mort     <fct> Vivo, Vivo, Vivo, Vivo, Vivo, Vivo, Vivo, Vivo, Vivo,...
## $ tempsviu <dbl> 12.4, 12.4, 9.6, 7.2, 14.1, 14.1, 12.4, 14.2, 12.4, 1...
## $ edat     <dbl> 44, 49, 49, 47, 43, 47, 50, 36, 50, 49, 50, 54, 42, 4...
## $ bmi      <dbl> 34.2, 32.6, 22.0, 37.9, 42.2, 33.1, 36.5, 38.5, 41.5,...
## $ edatdiag <dbl> 41, 48, 35, 45, 42, 44, 48, NA, 47, 45, 48, 43, 36, 4...
## $ tabac    <fct> No fumador, Fumador, Fumador, No fumador, Fumador, No...
## $ sbp      <dbl> 132, 130, 108, 128, 142, 156, 140, 144, 134, 102, 142...
## $ dbp      <dbl> 96, 72, 58, 76, 80, 94, 86, 88, 78, 68, 84, 74, 86, 5...
## $ ecg      <fct> Normal, Normal, Normal, Frontera, Normal, Normal, Fro...
## $ chd      <fct> No, No, Si, Si, No, No, Si, No, Si, No, No, No, No, N...
```

Check the levels of a factor

Usually when humans fill the database... a plenty of errors could be found :(

```
- An answer like "SI", could be entered like:
      "SI", "Si", "si", "SI ", "SÍ", .....
```

All this possible answers **will be differents levels for the same variable**

How to correct it?

We can use: `recode_factor`:

```
diab$mort <- recode_factor(diab$mort, "Muerto" = "muerto")
levels(diab$mort)
```

```
## [1] "muerto" "Vivo"
```

Return to the original version:

```
diab$mort <- recode_factor(diab$mort, "muerto" = "Muerto")
levels(diab$mort)
```

```
## [1] "Muerto" "Vivo"
```

Descriptive Statistics: Numerical summaries

Numerical Summaries (I)

We can access individual variables within a data frame using the $ operator. Let's print out all the *edat* values in the data. Let's then see what are `unique` values of each. Then let's calculate the `mean` , `median` and `range` for the *edad* variable:

```
#display all the values
diab$edat
```

```
##   [1] 44 49 49 47 43 47 50 36 50 49 50 54 42 44 40 48 50 47 38 35 51 40 54
##  [24] 53 45 41 34 38 43 45 40 44 48 51 36 52 41 49 44 37 51 47 45 38 35 50
##  [47] 53 48 40 43 54 52 69 38 50 64 44 38 62 47 78 49 63 71 51 59 50 66 42
##  [70] 40 67 86 52 42 60 75 81 60 60 63 62 57 71 58 42 45 66 61 48 82 35 57
##  [93] 56 49 50 53 71 55 69 59 47 75 80 57 52 48 57 58 51 33 52 52 64 31 69
## [116] 59 38 49 49 68 40 36 60 74 61 54 35 46 40 53 66 61 41 41 41 46 80 63
## [139] 72 41 52 53 61 53 75 40 61 62 49
```

Numerical Summaries (II)

```r
# Get the unique values of edat
diab$edat %>% unique()
```

```
##  [1] 44 49 47 43 50 36 54 42 40 48 38 35 51 53 45 41 34 52 37 69 64 62 78
## [24] 63 71 59 66 67 86 60 75 81 57 58 61 82 56 55 80 33 31 68 74 46 72
```

```r
diab$edat %>% unique() %>% length()
```

```
## [1] 45
```

Numerical Summaries (III)

```
#Mean, median and rang
mean(diab$edat)
```

```
## [1] 52.16779
```

```
median(diab$edat)
```

```
## [1] 50
```

```
sd(diab$edat)
```

```
## [1] 11.77285
```

```
var(diab$edat)
```

```
## [1] 138.6
```

```
range(diab$edat)
```

```
## [1] 31 86
```

Numerical Summaries (IV)

If we want to group the descriptive summaries by other variables we can use `group_by` function:

```
diab %>%
  group_by(tabac, ecg) %>%
  summarize(mean(edat))
```

```
## # A tibble: 9 x 3
## # Groups:   tabac [?]
##   tabac      ecg      `mean(edat)`
##   <fct>      <fct>           <dbl>
## 1 Ex fumador Anormal          68.5
## 2 Ex fumador Frontera         59.8
## 3 Ex fumador Normal           51.1
## 4 Fumador    Anormal          58
## 5 Fumador    Frontera         44.8
## 6 Fumador    Normal           44.7
## 7 No fumador Anormal          66.5
## 8 No fumador Frontera         53.8
## 9 No fumador Normal           56.0
```

Numerical Summaries (V)

A general summary of all variables:

```
summary(diab[, 2:11])
```

```
##      mort        tempsviu           edat            bmi       
##  Muerto: 25   Min.   : 0.00   Min.   :31.00   Min.   :18.20  
##  Vivo   :124   1st Qu.: 7.30   1st Qu.:43.00   1st Qu.:26.60  
##               Median :11.60   Median :50.00   Median :31.20  
##               Mean   :10.52   Mean   :52.17   Mean   :31.78  
##               3rd Qu.:13.90   3rd Qu.:60.00   3rd Qu.:35.20  
##               Max.   :16.90   Max.   :86.00   Max.   :59.70  
##                                                               
##     edatdiag            tabac         sbp             dbp        
##  Min.   :26.00   Ex fumador:41   Min.   : 98.0   Min.   : 58.00  
##  1st Qu.:38.00   Fumador   :51   1st Qu.:124.5   1st Qu.: 74.00  
##  Median :45.00   No fumador:57   Median :138.0   Median : 80.00  
##  Mean   :46.01                   Mean   :139.3   Mean   : 90.04  
##  3rd Qu.:53.25                   3rd Qu.:152.0   3rd Qu.: 88.00  
##  Max.   :81.00                   Max.   :222.0   Max.   :862.00  
##  NA's   :5                       NA's   :3                        
##        ecg       chd    
##  Anormal : 11   No:99  
##  Frontera: 27   Si:50  
##  Normal  :111          
##                        
```

Numerical Summaries (VI)

What happens if we have missing data in our dataset?

```
mean(diab$sbp)
```

```
## [1] NA
```

NA indicates *missing data* in the variable

Let's look the `sbp` variable:

```
diab$sbp
```

```
##   [1] 132 130 108 128 142 156 140 144 134 102 142 128 156 102 146 120 142
##  [18] 144  NA 134 130 122 132 150 134 142 124 102 134 118 192 122 122 112
##  [35] 142 152 112 118 152 136 134 130 108 126 132 144 126 128  NA 128 142
##  [52] 132 148 170 140 138 112 140 138 130 178 158 168 146 128 132 154 154
##  [69] 122 144 178 162 142 120 124 174 142 160 122 162 132 116 152 144  98
##  [86] 138 138 184 158 176 118 172 182 144 142 154 122 222 150 142 128 122
## [103] 162 172 132 112 138 128 132 120 140 140 172 136 152 126 104 142 128
## [120] 122 122 122 122 168 162  NA 126 180 132 150 106 154 122 120 120 144
## [137] 134 148 170 160 154 124 130 156 162 132 120 160 146
```

Numerical Summaries (VII)

How to work with *missing data*:

```
?mean
mean(diab$sbp, na.rm = TRUE)
```

```
## [1] 139.2603
```

```
is.na(diab$sbp)
```

```
##   [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [12] FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE
##  [23] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [34] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [45] FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE
##  [56] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [67] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [78] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [89] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [100] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [111] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [122] FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE
## [133] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [144] FALSE FALSE FALSE FALSE FALSE FALSE
```

Numerical Summaries (VIII)

How to work with *missing data*:

```
sum(is.na(diab$sbp))
```

```
## [1] 3
```

```
sum(is.na(diab$dbp))
```

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

EXERCISE

1. With the `diab` dataset

- Show only the rows from 35 to 98 and columns 5, 7, and from 9 to 11
- Change the level of the variable *tabac*, from **No Fumador** to **No_Fumador**
- Display the unique values for the variable *bmi*. Count how many exist.
- Display the mean of *edatdiag*, grouped by *ecg*

Descriptive Statistics: Graphical summaries

Exploratory Data Analysis (EDA)

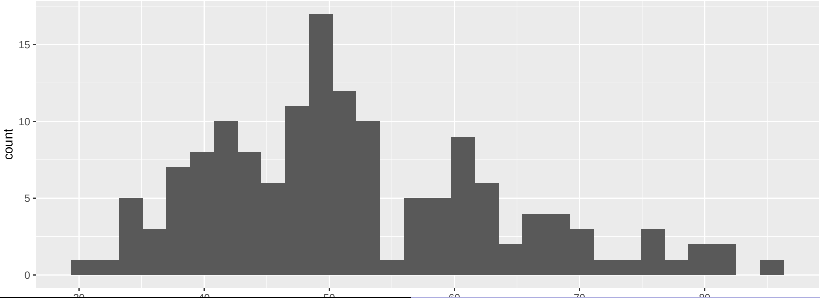
We could dedicate half of the course only to EDA. Here we will only see the most common approaches to visualize data:

```
- Histograms
- Scatterplots
- Boxplots
```

Histograms

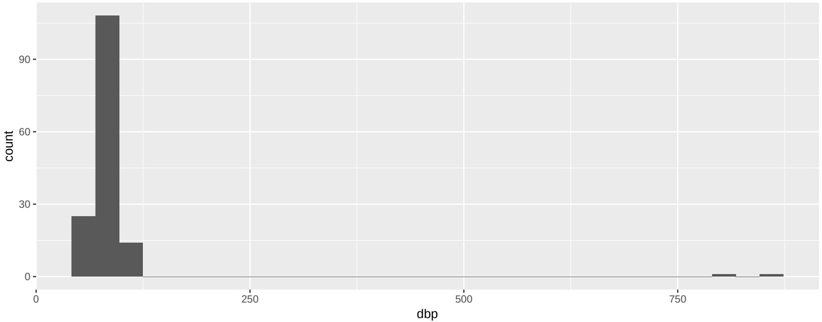
We will use histograms to plot the frequencies of each level of variables. This is the way to see the data distribution of particulars variables.

```
ggplot(diab, aes(edat)) +
  geom_histogram(bins = 30)
```



Histograms (II)

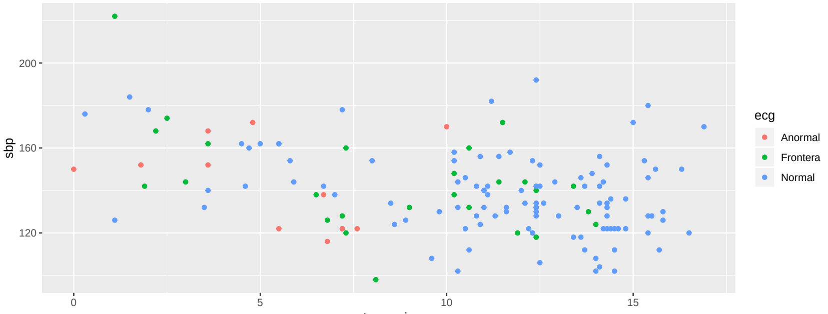
```
ggplot(diab, aes(dbp)) +
  geom_histogram(bins = 30)
```



Scatterplots. Two Continuous variables

This is the graphical way to check the relation between two variables:

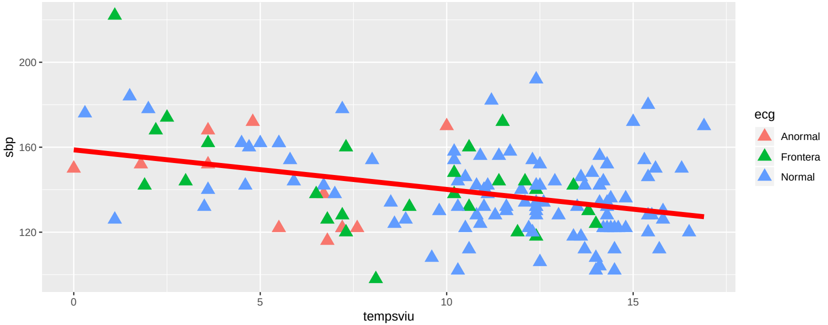
```
ggplot(diab, aes(tempsviu, sbp, col = ecg)) +
  geom_point()
```



Scatterplots (II)

```
ggplot(diab, aes(tempsviu, sbp, col = ecg)) +
  geom_point(size = 4, pch = 17) +
  geom_smooth(lwd=2, se=FALSE, method="lm", col="red")
```

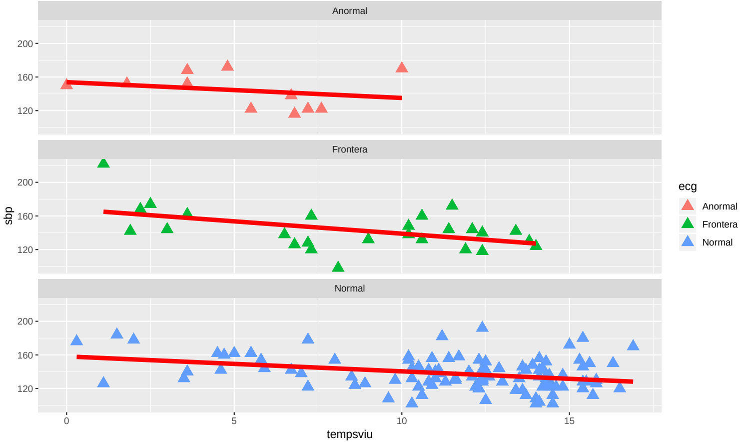
Scatterplots (II)



Faceting

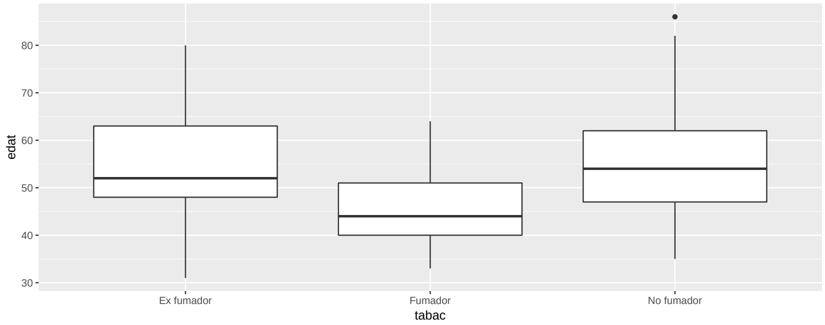
```
ggplot(diab, aes(tempsviu, sbp, col = ecg)) +
  geom_point(size = 4, pch = 17) +
  geom_smooth(lwd = 2, se=FALSE, method="lm", col="red") +
  facet_wrap(~ ecg, ncol = 1)
```

Faceting


Anormal
Frontera
Normal
200
160
120
sbp
0
5
10
15
tempsviu
ecg
Anormal
Frontera
Normal

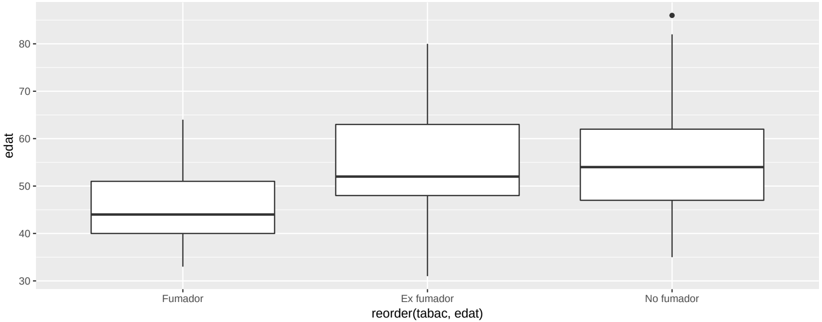
Boxplot. Continuous versus categorical

```
ggplot(diab, aes(tabac, edat)) +
  geom_boxplot()
```



Boxplot (II)

```
ggplot(diab, aes(x= reorder(tabac, edat), y = edat)) +
  geom_boxplot()
```



EXERCISE

2. With the `diab` dataset

- Use the best graphic type to plot the relation between *sbp* and *dbp*
- Show graphically the relation between *edat* and *ecg*
- Plot the *sbp* frequencies
- Improve the first graphic (add linear regression, avoid strange data in dbp, . . .)

Statistics test. Continuous Variables

Two-sample t-test

- Two-sample t-test will assess the differences in means between two groups
- The function for a t-test is t.test()
- The usage is t.test(response~group, data=myDataFrame)

Recode levels of *tabac* variable

Let's recode the levels of the variable to gain more opportunities of "playing" with the dataset

```
levels(diab$tabac)
```

```
## [1] "Ex fumador" "Fumador"    "No fumador"
```

```
diab$tabac <- recode_factor(diab$tabac, "Ex fumador" = "Fumador")

levels(diab$tabac)
```

```
## [1] "Fumador"    "No fumador"
```

Two-sample t-test (II)

Are there differences in coronary heart disease (*chd*) depending of body mass index (*bmi*) in this dataset?

Two-sample t-test (II)

```
t.test(bmi ~ chd, data = diab)
```

```
##
##  Welch Two Sample t-test
##
## data:  bmi by chd
## t = 2.3387, df = 102.14, p-value = 0.0213
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.4072447 4.9543310
## sample estimates:
## mean in group No mean in group Si
##         32.67879         29.99800
```

Two-sample t-test (III)

Are there differences in *edat* depending of *tabac* in this dataset?

Two-sample t-test (III)

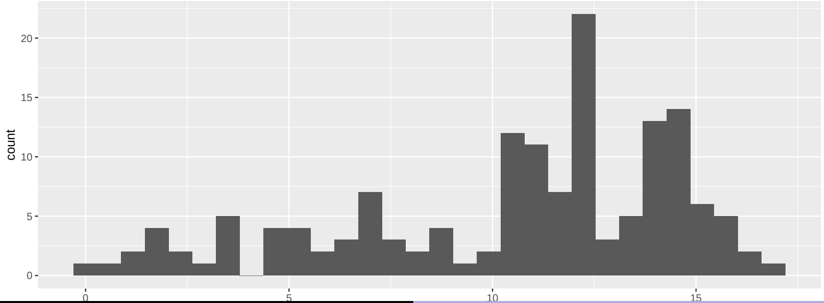
```
t.test(edat ~ tabac, data = diab)
```

```
##
##  Welch Two Sample t-test
##
## data:  edat by tabac
## t = -3.3969, df = 111.3, p-value = 0.0009461
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -10.501812  -2.763635
## sample estimates:
##    mean in group Fumador mean in group No fumador
##                 49.63043                 56.26316
```

Wilcoxon rank-sum test (a.k.a. Mann-Whitney *U* test)

When the Continuous variable has not a normal distribution

```
ggplot(diab, aes(tempsviu)) +
  geom_histogram(bins = 30)
```



Wilcoxon rank-sum test (a.k.a. Mann-Whitney *U* test) II

```
wilcox.test(tempsviu ~ chd, data = diab)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  tempsviu by chd
## W = 3711, p-value = 6.738e-07
## alternative hypothesis: true location shift is not equal to 0
```

Statistics test. Discrete Variables

Contingency tables

```
xtabs(~ chd + tabac, data = diab)
```

```
##      tabac
## chd   Fumador No fumador
##   No       61         38
##   Si       31         19
```

```
addmargins(xtabs(~ chd + tabac, data = diab))
```

```
##       tabac
## chd    Fumador No fumador Sum
##   No        61         38  99
##   Si        31         19  50
##   Sum       92         57 149
```

Chi-square test

Chi-square test is used top assess the independence of these two factors. That is, if the null hypothesis that smoking habits and coronary heart disease history are independent is true, then we would expect a proportionally equal number of smokers across each coronary heart disease history level. Smokers seem to be slightly higher risk than non-smokers, but the difference is just short of statistically significant.

Chi-square test

```
chisq.test(xtabs(~ chd + tabac, data = diab))
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  xtabs(~chd + tabac, data = diab)
## X-squared = 5.3975e-31, df = 1, p-value = 1
```

Fisher's exact test

Useful when $n < 5$ in some of the groups.

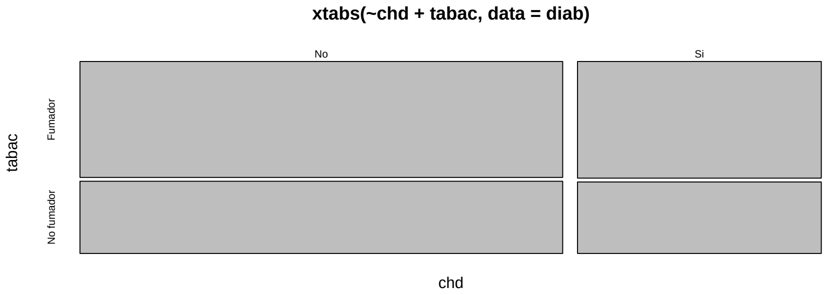
```
fisher.test(xtabs(~ chd + tabac, data = diab))
```

```
##
##  Fisher's Exact Test for Count Data
##
## data:  xtabs(~chd + tabac, data = diab)
## p-value = 1
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
##  0.4569698 2.0900449
## sample estimates:
## odds ratio
##  0.9839764
```

Plot the results

Mosaic plot is a useful way to visualize contingency table data

```
mosaicplot(xtabs(~ chd + tabac, data = diab))
```



EXERCISE

3. With the `diab` dataset

- Are there any differences in *sbp* values between *chd* groups
- Show graphically the relation between *sbp* and *chd*
- Create a contingency table between *mort* and *tabac*. Plot the table
- Test if smoking habits is related with *mort* variable.

Bonus: ANOVA and Linear Regression

ANOVA

Remember that t-tests are for assessing the differences in means between two groups. A t-test is a specific case of ANOVA, which is a specific case of a linear model.

```
t.test(edat ~ ecg, data = diab)

Error in t.test.formula(edat ~ ecg, data = diab) :
  grouping factor must have exactly 2 levels
```

ANOVA (II)

Let's look the relationship between *edat* and *ecg* with ANOVA:

```
fit <- lm(edat ~ ecg, data = diab)
anova(fit)
```

```
## Analysis of Variance Table
##
## Response: edat
##            Df  Sum Sq Mean Sq F value    Pr(>F)
## ecg         2  2166.1 1083.04  8.6186 0.0002897 ***
## Residuals 146 18346.7  125.66
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

ANOVA (II)

ANOVA only says if there exists differences among the levels (in general), but does not say anything about differences within the levels. We have to draw on **Tukey's test**

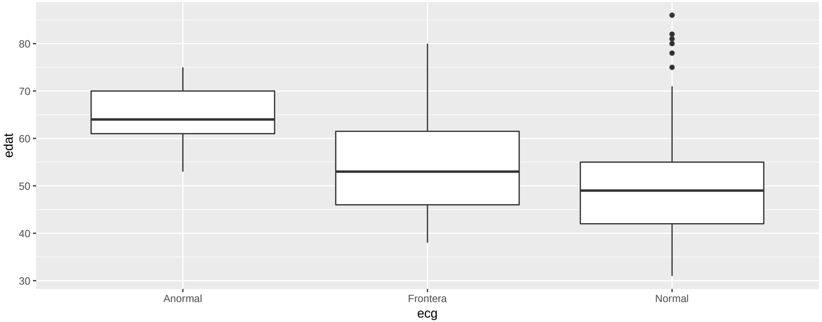
```
TukeyHSD(aov(fit))
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = fit)
##
## $ecg
##                       diff        lwr       upr     p adj
## Frontera-Anormal -11.09428 -20.588782 -1.599771 0.0174729
## Normal-Anormal   -14.40459 -22.794951 -6.014221 0.0002297
## Normal-Frontera   -3.31031  -9.006113  2.385493 0.3560430
```

ANOVA (III)

It is very useful to plot the two variables

```
ggplot(diab, aes(ecg, edat)) +
  geom_boxplot()
```



Linear Models

Linear model seeks to explain the relationship between a variable of interest, our Y, outcome, response, or dependent variable, and one or more X, predictor, or independent variables.

```
Y = beta0 + beta1·X + error
```

where

```
Y is the response
X is the predictor variable
beta0 is the intercept
beta1 is the coefficient
error is the random error
```

Linear Models (II)

Let's look the relationship between *sbp* and *dbp*

```
fit <- lm(edat ~ sbp, data = diab)
summary(fit)
```

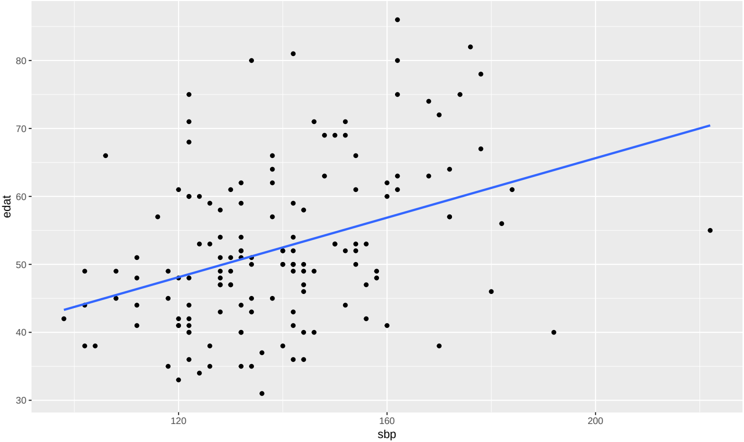
```
##
## Call:
## lm(formula = edat ~ sbp, data = diab)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -23.882  -7.012  -1.498   5.624  28.816
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 21.84633    6.30134   3.467 0.000694 ***
## sbp          0.21894    0.04478   4.889 2.67e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.96 on 144 degrees of freedom
##   (3 observations deleted due to missingness)
## Multiple R-squared:  0.1424, Adjusted R-squared:  0.1364
## F-statistic: 23.91 on 1 and 144 DF,  p-value: 2.668e-06
```

Linear Models (III)

Let's plot the relationship

```
ggplot(diab, aes(sbp, edat)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE)
```

Linear Models (III)



Your turn

EXERCISE

4. Using the *osteoporosis.csv* dataset
   - Load the dataset and check if it is correctly loaded
   - Calculate the mean and standard deviation of imc grouped by clasific
   - Plot the distribution of edat
   - Plot the relationship between talla and peso
   - Compute the model between talla and peso. Add the linear regression to previous plot
   - Is bua values different between levels of menop?
   - Is imc values different among levels of grupedad? Between which levels?
   - Build a contingency table between clasific and grupedad. Check if there is independence between the levels of clasific and grupedad.