Epidemiologic data analysis using R

Practicals 1

Faculty of Social Sciences, University of Tampere

–

Janne Pitkäniemi

28.02.2018

Topics of practical 1

Learning objectives of this practical

- get familiar with R scripting and running R commands
- utilize R-studio in your daily work
- learn to use R packages
- data input from an external SPSS file,
- basic data manipulation tasks,
- tabulation using function stat.table() in package Epi.

1. Basics with R and R-Studio

Defining working directory and launching R with R-studio

1. Create a special subdirectory within your own user account as a **working directory** to contain the necessary R scripts and data files to be used in this course.
2. For example, my working directory for this course on my laptop is

```
setwd("C:/Users/janne.pitkaniemi/Projects/TRE2018")
```

, but yours is probably something else.

3. Open R-studio by clicking on the appropriate icon.
4. Change the default working directory of R by choosing *Session - Set Working Directory - Choose Directory* Use your own directory name here! instead of the default directory offered by R.

5. Check whether there are any objects in the memory from the upper risgt hand corner - Environment

Comment: When working with R it is useful to allocate for each project its own directory in which the files pertaining to that project are located, and to which especially the files created during an R session will be saved. When this directory is declared as the working directory in the beginning of a session, it will specify the default directory path for the files to be loaded and saved. However, files can still be loaded from and saved to other directories, but then the whole directory path must be specified in the file names.

Attention! Within an R script a slash '/' must be then used instead of backslash '\' in the directory paths.

2. Working with script files in R-Studio

Writing and saving commands in a R-script file and running them from it in R-studio

1. Open new file in R-studio : *File - New file - R script.*
2. Install an R-package called foreign that will enable you to read several datafiles from other statistical programs like SPSS: *Tools - Install Packages . . .* . Types foreign and choose Install

Installation of a new package needs to be done only once when start to use a new package (or it's latest version in your use). 3. Type the following two lines:

```
x<-2
print(x)
```

```
[1] 2
```

4. Save the script from Save As and give a name for the script file *.R
5. Close R-studio

3. Reading external data

Data file *breastca.sav* in SPSS-format is found from the Moodle site designated for this course. It contains data on 11 variables from 1207 women with breast cancer. These describe characteristics related to the survival time of the patient. We will read the data set into an R data frame and analyze it in subsequent tasks. First we'll view the data in SPSS.

1. Write on the editor window the following R command lines, which will load some packages, read in the SPSS data set into a data frame and view its properties. Comments after # in each line can be omitted.

```
library(foreign)
library(Epi)
```

```
Attaching package: 'Epi'

The following object is masked from 'package:base':

    merge.data.frame
```

```
bca <- read.spss("C:/Users/janne.pitkaniemi/Projects/TRE2018/breastca.sav",
                 to.data.frame=TRUE)
bca[1:10, ]      # listing the first 20 rows of the data frame
```

```
   ID AGE PATHSIZE LNPOS  HISTGRAD       ER       PR   STATUS PATHSCAT
1   1  60       NA     0 Grade III Negative Negative Censored     <NA>
2   2  79       NA     0      <NA>     <NA>     <NA> Censored     <NA>
3   3  82       NA     0  Grade II     <NA>     <NA> Censored     <NA>
4   4  66       NA     0  Grade II Positive Positive Censored     <NA>
5   5  52       NA     0 Grade III     <NA>     <NA> Censored     <NA>
6   6  58       NA     0      <NA>     <NA>     <NA> Censored     <NA>
7   7  50       NA     0  Grade II Positive Negative Censored     <NA>
8   8  83       NA     0 Grade III Negative Negative Censored     <NA>
9   9  46       NA    17      <NA>     <NA>     <NA> Censored     <NA>
10 10  54       NA     6  Grade II Positive Positive Censored     <NA>
   LN_YESNO      TIME
1        No  9.466667
```

```
2        No  8.600000
3        No 19.333333
4        No 16.333333
5        No  8.500000
6        No  9.400000
7        No 17.666667
8        No  9.300000
9       Yes 27.633333
10      Yes 11.133333
```

```
str(bca)         # viewing the structure
```

```
'data.frame':   1207 obs. of  11 variables:
 $ ID      : num  1 2 3 4 5 6 7 8 9 10 ...
 $ AGE     : num  60 79 82 66 52 58 50 83 46 54 ...
 $ PATHSIZE: num  NA NA NA NA NA NA NA NA NA NA ...
 $ LNPOS   : num  0 0 0 0 0 0 0 0 17 6 ...
 $ HISTGRAD: Factor w/ 3 levels "Grade I","Grade II",..: 3 NA 2 2 3 NA 2 3 NA 2 ...
 $ ER      : Factor w/ 2 levels "Negative","Positive": 1 NA NA 2 NA NA 2 1 NA 2 ...
 $ PR      : Factor w/ 2 levels "Negative","Positive": 1 NA NA 2 NA NA 1 1 NA 2 ...
 $ STATUS  : Factor w/ 2 levels "Censored","Died": 1 1 1 1 1 1 1 1 1 1 ...
 $ PATHSCAT: Factor w/ 4 levels "0 cm","<= 2 cm",..: NA NA NA NA NA NA NA NA NA NA ...
 $ LN_YESNO: Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 2 2 ...
 $ TIME    : num  9.47 8.6 19.33 16.33 8.5 ...
 - attr(*, "variable.labels")= Named chr  "" "Age (years)" "Pathologic Tumor Size (cm)" "Positive Axilla
  ..- attr(*, "names")= chr  "ID" "AGE" "PATHSIZE" "LNPOS" ...
```

```
attributes(bca)$variable.labels # description of variables
```

```
                                  ID
                                  ""
                                 AGE
                       "Age (years)"
                            PATHSIZE
         "Pathologic Tumor Size (cm)"
                               LNPOS
      "Positive Axillary Lymph Nodes"
                            HISTGRAD
                  "Histologic Grade"
                                  ER
           "Estrogen Receptor Status"
                                  PR
        "Progesterone Receptor Status"
                              STATUS
                            "Status"
                            PATHSCAT
"Pathological Tumor Size (Categories)"
                            LN_YESNO
                       "Lymph Nodes?"
                                TIME
                     "Time (months)"
```

```
summary(bca)     # summary statistics of the variables
```

```
       ID              AGE           PATHSIZE         LNPOS
 Min.   :   1.0   Min.   :22.00   Min.   :0.100   Min.   : 0.0000
```

```
 1st Qu.: 310.5   1st Qu.:46.00   1st Qu.:1.000   1st Qu.: 0.0000
 Median : 619.0   Median :56.00   Median :1.500   Median : 0.0000
 Mean   : 621.1   Mean   :56.39   Mean   :1.733   Mean   : 0.8807
 3rd Qu.: 931.5   3rd Qu.:66.50   3rd Qu.:2.200   3rd Qu.: 0.0000
 Max.   :1266.0   Max.   :88.00   Max.   :7.000   Max.   :35.0000
                                  NA's   :86
     HISTGRAD          ER             PR           STATUS
 Grade I  : 79   Negative:338   Negative:389   Censored:1135
 Grade II :514   Positive:531   Positive:462   Died     :  72
 Grade III:327   NA's    :338   NA's    :356
 NA's     :287


   PATHSCAT   LN_YESNO       TIME
 0 cm   :  0   No :929   Min.   :  2.633
 <= 2 cm:826   Yes:278   1st Qu.: 22.550
 2-5 cm :283             Median : 42.967
 > 5 cm : 12             Mean   : 46.956
 NA's   : 86             3rd Qu.: 65.583
                         Max.   :133.800
```

Data table package is an alternative for handling (large) datasets and making summary tables

```
library(haven)
```

```
Warning: package 'haven' was built under R version 3.4.3
```

```
library(data.table)
bca <- read_sav("C:/Users/janne.pitkaniemi/Projects/TRE2018/breastca.sav")
bcadt<-data.table(bca) # This converts data.frame to data.table object
bcadt      # listing the first 20 rows of the data frame
```

```
         ID AGE PATHSIZE LNPOS HISTGRAD ER PR STATUS PATHSCAT LN_YESNO
   1:     1  60       NA     0        3  0  0      0       NA        0
   2:     2  79       NA     0       NA NA NA      0       NA        0
   3:     3  82       NA     0        2 NA NA      0       NA        0
   4:     4  66       NA     0        2  1  1      0       NA        0
   5:     5  52       NA     0        3 NA NA      0       NA        0
  ---
1203:  1259  72      3.0     0        2 NA NA      0        2        0
1204:  1261  41      1.2     0        2  1  1      0        1        0
1205:  1262  71      1.6     0        3  0  0      0        1        0
1206:  1263  48      2.5     4        3  0  0      0        2        1
1207:  1266  73      2.4     0        3  1  1      0        2        0
           TIME
   1:  9.466667
   2:  8.600000
   3: 19.333333
   4: 16.333333
   5:  8.500000
  ---
1203: 88.933333
1204: 90.166667
1205: 22.566667
```

```
1206: 45.200000
1207:  6.100000
```

```
str(bcadt)          # viewing the structure
```

```
Classes 'data.table' and 'data.frame':  1207 obs. of  11 variables:
 $ ID      : atomic  1 2 3 4 5 6 7 8 9 10 ...
  ..- attr(*, "format.spss")= chr "F8.0"
 $ AGE     : atomic  60 79 82 66 52 58 50 83 46 54 ...
  ..- attr(*, "label")= chr "Age (years)"
  ..- attr(*, "format.spss")= chr "F8.0"
 $ PATHSIZE: atomic  NA NA NA NA NA NA NA NA NA NA ...
  ..- attr(*, "label")= chr "Pathologic Tumor Size (cm)"
  ..- attr(*, "format.spss")= chr "F8.2"
 $ LNPOS   : atomic  0 0 0 0 0 0 0 0 17 6 ...
  ..- attr(*, "label")= chr "Positive Axillary Lymph Nodes"
  ..- attr(*, "format.spss")= chr "F8.0"
 $ HISTGRAD:Class 'labelled'  atomic [1:1207] 3 NA 2 2 3 NA 2 3 NA 2 ...
  .. ..- attr(*, "label")= chr "Histologic Grade"
  .. ..- attr(*, "format.spss")= chr "F8.0"
  .. ..- attr(*, "labels")= Named num [1:4] 1 2 3 4
  .. .. ..- attr(*, "names")= chr [1:4] "Grade I" "Grade II" "Grade III" "Unknown"
 $ ER      :Class 'labelled'  atomic [1:1207] 0 NA NA 1 NA NA 1 0 NA 1 ...
  .. ..- attr(*, "label")= chr "Estrogen Receptor Status"
  .. ..- attr(*, "format.spss")= chr "F6.0"
  .. ..- attr(*, "labels")= Named num [1:3] 0 1 2
  .. .. ..- attr(*, "names")= chr [1:3] "Negative" "Positive" "Unknown"
 $ PR      :Class 'labelled'  atomic [1:1207] 0 NA NA 1 NA NA 0 0 NA 1 ...
  .. ..- attr(*, "label")= chr "Progesterone Receptor Status"
  .. ..- attr(*, "format.spss")= chr "F6.0"
  .. ..- attr(*, "labels")= Named num [1:3] 0 1 2
  .. .. ..- attr(*, "names")= chr [1:3] "Negative" "Positive" "Unknown"
 $ STATUS  :Class 'labelled'  atomic [1:1207] 0 0 0 0 0 0 0 0 0 0 ...
  .. ..- attr(*, "label")= chr "Status"
  .. ..- attr(*, "format.spss")= chr "F8.0"
  .. ..- attr(*, "labels")= Named num [1:2] 0 1
  .. .. ..- attr(*, "names")= chr [1:2] "Censored" "Died"
 $ PATHSCAT:Class 'labelled'  atomic [1:1207] NA NA NA NA NA NA NA NA NA NA ...
  .. ..- attr(*, "label")= chr "Pathological Tumor Size (Categories)"
  .. ..- attr(*, "format.spss")= chr "F8.0"
  .. ..- attr(*, "labels")= Named num [1:4] 0 1 2 3
  .. .. ..- attr(*, "names")= chr [1:4] "0 cm" "<= 2 cm" "2-5 cm" "> 5 cm"
 $ LN_YESNO:Class 'labelled'  atomic [1:1207] 0 0 0 0 0 0 0 0 1 1 ...
  .. ..- attr(*, "label")= chr "Lymph Nodes?"
  .. ..- attr(*, "format.spss")= chr "F8.0"
  .. ..- attr(*, "labels")= Named num [1:2] 0 1
  .. .. ..- attr(*, "names")= chr [1:2] "No" "Yes"
 $ TIME    : atomic  9.47 8.6 19.33 16.33 8.5 ...
  ..- attr(*, "label")= chr "Time (months)"
  ..- attr(*, "format.spss")= chr "F8.2"
 - attr(*, ".internal.selfref")=<externalptr>
```

```
summary(bcadt)      # summary statistics of the variables
```

```
       ID              AGE           PATHSIZE         LNPOS
```

```
Min.   :   1.0   Min.   :22.00   Min.   :0.100   Min.   : 0.0000
1st Qu.: 310.5   1st Qu.:46.00   1st Qu.:1.000   1st Qu.: 0.0000
Median : 619.0   Median :56.00   Median :1.500   Median : 0.0000
Mean   : 621.1   Mean   :56.39   Mean   :1.733   Mean   : 0.8807
3rd Qu.: 931.5   3rd Qu.:66.50   3rd Qu.:2.200   3rd Qu.: 0.0000
Max.   :1266.0   Max.   :88.00   Max.   :7.000   Max.   :35.0000
                                 NA's   :86
   HISTGRAD          ER              PR             STATUS
Min.   :1.00   Min.   :0.000   Min.   :0.0000   Min.   :0.00000
1st Qu.:2.00   1st Qu.:0.000   1st Qu.:0.0000   1st Qu.:0.00000
Median :2.00   Median :1.000   Median :1.0000   Median :0.00000
Mean   :2.27   Mean   :0.611   Mean   :0.5429   Mean   :0.05965
3rd Qu.:3.00   3rd Qu.:1.000   3rd Qu.:1.0000   3rd Qu.:0.00000
Max.   :3.00   Max.   :1.000   Max.   :1.0000   Max.   :1.00000
NA's   :287    NA's   :338     NA's   :356
   PATHSCAT        LN_YESNO          TIME
Min.   :1.000   Min.   :0.0000   Min.   :  2.633
1st Qu.:1.000   1st Qu.:0.0000   1st Qu.: 22.550
Median :1.000   Median :0.0000   Median : 42.967
Mean   :1.274   Mean   :0.2303   Mean   : 46.956
3rd Qu.:2.000   3rd Qu.:0.0000   3rd Qu.: 65.583
Max.   :3.000   Max.   :1.0000   Max.   :133.800
NA's   :86
```

2. Save the script file (*File - Save - etc.*) into your own directory with name *bca.R*.
3. Run the commands written to the script file as follows: click the Run command in the scritping window
4. View the resulting output. Which of the variables are numeric and which are factors? Which values are referring to missing data for each of the variables? % Take your time!
5. Continue with writing each command in the following tasks to the script file, save, and run selected command lines as above.

4. Working with variables in R

Categorization of a numeric variable and forming 1-way frequency and percentage tables. 1. Create a factor named *age.gr* to this data frame with levels or age groups (years) 20-49, 50-64, 65-89 from variable *AGE* using function *cut( )*.

Get familiar with the syntax of this function by visiting its *help* page. Specify *right=F* so that each breakpoint is also an exact lower limit for an age group:

```
bca$age.gr <- cut(bca$AGE, br=c(20, 50, 65, 90), right=F)
head(bca[1:5,c(2,12)])
```

```
# A tibble: 5 x 2
    AGE age.gr
  <dbl> <fctr>
1  60.0 [50,65)
2  79.0 [65,90)
3  82.0 [65,90)
4  66.0 [65,90)
5  52.0 [50,65)
```

Data.table version of the same

```
bcadt[,age.gr:=cut(bca$AGE, br=c(20, 50, 65, 90), right=F)]
head(bcadt[1:5,c(2,12)])
```

```
   AGE  age.gr
1:  60 [50,65)
2:  79 [65,90)
3:  82 [65,90)
4:  66 [65,90)
5:  52 [50,65)
```

2. It is possible to label the age groups to be more of publication style:

```
levels(bca$age.gr) <-  c('20-49', '50-64', '65-89')
head(bca[1:5,c(2,12)])
```

```
# A tibble: 5 x 2
    AGE age.gr
  <dbl> <fctr>
1  60.0 50-64
2  79.0 65-89
3  82.0 65-89
4  66.0 65-89
5  52.0 50-64
```

You may attach the data frame after this: *attach(bca)*

3. Form a marginal frequency table for *age.gr* using function *table()* and print:

```
table(bca$age.gr)
```

```
20-49 50-64 65-89
  416   423   368
```

Data.table version

```
bcadt[order(age.gr),.N,by=age.gr]
```

```
    age.gr   N
1: [20,50) 416
2: [50,65) 423
3: [65,90) 368
```

4. Print a frequency table of variable *HISTGRAD* in the same way as for *age.gr*.

```
table(bca$HISTGRAD)
```

```
  1   2   3
 79 514 327
```

Data.table version

```
bcadt[order(HISTGRAD),.N,by=HISTGRAD]
```

```
   HISTGRAD   N
1:        1  79
2:        2 514
3:        3 327
4:       NA 287
```

5. Tabulations using stat.table (or data.table)

Do tabulations using the *stat.table()* function in the *Epi* package. First install package *Epi* from R-studio *Tools – install packages* and run following scrpit

```
library(Epi)
```

1. Frequencies and percentages of histological grade simultaneously, and marginal totals:

```
stat.table( HISTGRAD, list(count(), percent(HISTGRAD)),
                      margins=T,
                      data=bca);
```

```
 ------------------------------------
 HISTGRAD   count() percent(HISTGRAD)
 ------------------------------------
 1               79               8.6
 2              514              55.9
 3              327              35.5

 Total         1207             100.0
 ------------------------------------
```

Data.table version without missing values

```
res<-bcadt[order(HISTGRAD) & !is.na(HISTGRAD),.N,by=HISTGRAD] [, prop := 100*(N/sum(N)), ]
print(res)
```

```
   HISTGRAD   N      prop
1:        3 327 35.543478
2:        2 514 55.869565
3:        1  79  8.586957
```

Data.table version with missing values

```
res<-bcadt[order(HISTGRAD),.N,by=HISTGRAD] [, prop := 100*(N/sum(N)), ]
print(res)
```

```
   HISTGRAD   N      prop
1:        1  79  6.545153
2:        2 514 42.584921
3:        3 327 27.091964
4:       NA 287 23.777962
```

2. The columns can be neatly labelled:

```
stat.table(  HISTGRAD,
             list(Number = count(), 'Per cent' = percent(HISTGRAD)),
             margins=T,
             data=bca);
```

```
 --------------------------
 HISTGRAD    Number     Per
                       cent
 --------------------------
 1               79     8.6
 2              514    55.9
 3              327    35.5
```

```
 Total          1207   100.0
 --------------------------
```

Almost like a publication quality table!

3. Tabulate the mean age of patients as well as the minimum and maximum ages in each grade group:\

```
stat.table( HISTGRAD,
 list(Number = count(),
   'Mean age' = mean(AGE), min(AGE), max(AGE)),
  margins=T,data=bca);
```

```
 ---------------------------------------------
 HISTGRAD    Number    Mean min(AGE) max(AGE)
                        age
 ---------------------------------------------
 1               79   57.76    33.00    88.00
 2              514   57.53    24.00    87.00
 3              327   53.23    22.00    84.00

 Total         1207   56.39    22.00    88.00
 ---------------------------------------------
```

5. The numerical precision of numbers representing other quantities than counts or percentages is two decimal points by default. For us, two decimals in mean ages is exaggerating, so we wish to cut it into one decimal. There is a special **print method** for **stat.table()**, by which one can tune the number of decimal points. Before that, save the table into an own object.

```
mage.gr <- stat.table( HISTGRAD,
list(Number = count(),
'Mean age' = mean(AGE), min(AGE), max(AGE)),
 margins=T,data=bca);

print(mage.gr,  digits=c(mean=1, min=0, max=0));
```

```
 ---------------------------------------------
 HISTGRAD    Number    Mean min(AGE) max(AGE)
                        age
 ---------------------------------------------
 1               79    57.8       33       88
 2              514    57.5       24       87
 3              327    53.2       22       84

 Total         1207    56.4       22       88
 ---------------------------------------------
```

6. Two-way contingency tables, row & column percentages,and chi-square testing.

1. Form a 2-way contingency table with *age.gr* as the row variable and *HISTGRAD* as the column variable using function *table()*. Assign it to object *grbyage* and print. Take a look at the table. Can you judge anything about the association between age and histological grade from the table?

```
grbyage<-table(bca$age.gr,bca$HISTGRAD)
grbyage
```

```
       1   2   3
20-49  25 159 140
50-64  28 183 110
65-89  26 172  77
```

2. Using function *stat.table()* compute and print the frequency (counts) and percentage distribution of *HISTGRAD* in the three age groups:

```
stat.table( index = list(age.gr, HISTGRAD),
 contents = list(count(), percent(HISTGRAD) ), data=bca );
```

```
 ----------------------------------
         --------HISTGRAD---------
 age.gr          1       2       3
 ----------------------------------
 20-49          25     159     140
               7.7    49.1    43.2

 50-64          28     183     110
               8.7    57.0    34.3

 65-89          26     172      77
               9.5    62.5    28.0
 ----------------------------------
```

Can you now say more about, how the grade distribution depends on age?

3. Maybe you wish to add marginal distributions to the table. For later purposes we also add the data frame as a *data* argument indicating that *stat.table()* can also operate on variables that are hidden in unattached data frames:

```
stat.table( index = list(age.gr, HISTGRAD),
            contents = list(count(), percent(HISTGRAD) ),
            margins = T, data = bca)
```

```
 ----------------------------------------
         ------------HISTGRAD-------------
 age.gr          1       2       3   Total
 ----------------------------------------
 20-49          25     159     140     416
               7.7    49.1    43.2   100.0

 50-64          28     183     110     423
               8.7    57.0    34.3   100.0

 65-89          26     172      77     368
```

```
              9.5    62.5    28.0   100.0


 Total         79     514     327    1207
              8.6    55.9    35.5   100.0
 ----------------------------------------
```

4. Print a similar table as in (c) that contains the row percentages only. This can be obtained by dropping the *count()* argument (and the comma after it!) from the list of *contents*.

```
stat.table( index = list(age.gr, HISTGRAD),
            contents = list(percent(HISTGRAD) ),
            margins = T, data = bca)
```

```
 ----------------------------------------
         ------------HISTGRAD-------------
 age.gr         1       2       3   Total
 ----------------------------------------
 20-49        7.7    49.1    43.2   100.0
 50-64        8.7    57.0    34.3   100.0
 65-89        9.5    62.5    28.0   100.0

 Total        8.6    55.9    35.5   100.0
 ----------------------------------------
```

5. Perform a chi-square test for independence between *age.gr* and *HISTGRAD* using function *chisq.test()* with these variables as its main (and only) arguments.

```
chisq.test(bca$age.gr,bca$HISTGRAD)
```

```
    Pearson's Chi-squared test

data:  bca$age.gr and bca$HISTGRAD
X-squared = 15.388, df = 4, p-value = 0.003961
```

6. Assign the value of the chi-square test function in the previous item to an object with name *res*, say, and view its structure usint *str()* function.

```
res<-chisq.test(bca$age.gr,bca$HISTGRAD)
str(res)
```

```
List of 9
 $ statistic: Named num 15.4
  ..- attr(*, "names")= chr "X-squared"
 $ parameter: Named int 4
  ..- attr(*, "names")= chr "df"
 $ p.value  : num 0.00396
 $ method   : chr "Pearson's Chi-squared test"
 $ data.name: chr "bca$age.gr and bca$HISTGRAD"
 $ observed : 'table' int [1:3, 1:3] 25 28 26 159 183 172 140 110 77
  ..- attr(*, "dimnames")=List of 2
  .. ..$ bca$age.gr  : chr [1:3] "20-49" "50-64" "65-89"
  .. ..$ bca$HISTGRAD: chr [1:3] "1" "2" "3"
 $ expected : num [1:3, 1:3] 27.8 27.6 23.6 181 179.3 ...
  ..- attr(*, "dimnames")=List of 2
  .. ..$ bca$age.gr  : chr [1:3] "20-49" "50-64" "65-89"
  .. ..$ bca$HISTGRAD: chr [1:3] "1" "2" "3"
```

```
 $ residuals: table [1:3, 1:3] -0.535 0.083 0.491 -1.636 0.273 ...
  ..- attr(*, "dimnames")=List of 2
  .. ..$ bca$age.gr  : chr [1:3] "20-49" "50-64" "65-89"
  .. ..$ bca$HISTGRAD: chr [1:3] "1" "2" "3"
 $ stdres   : table [1:3, 1:3] -0.695 0.108 0.613 -3.061 0.51 ...
  ..- attr(*, "dimnames")=List of 2
  .. ..$ bca$age.gr  : chr [1:3] "20-49" "50-64" "65-89"
  .. ..$ bca$HISTGRAD: chr [1:3] "1" "2" "3"
 - attr(*, "class")= chr "htest"
```

7. Can you extract the *expected frequencies* from it? If yes, print them.

```
res$expected
```

```
          bca$HISTGRAD
bca$age.gr        1        2         3
     20-49 27.82174 181.0174 115.16087
     50-64 27.56413 179.3413 114.09457
     65-89 23.61413 153.6413  97.74457
```

7. Two- and three-dimensional tables

1. We are interested to know how does the presenec of lymphatic nodes (*LN_YESNO*) in breast cancer patients seem to depend on age?

Create and print by *stat.table()* a 2-way frequency table with *age.gr* as the row variable and variable *LN_YESNO* (presence vs. absence of ≥ 1 lymph nodes) as the column variable. Present the row percentages, too.

```
stat.table( index = list(age.gr, LN_YESNO),
            contents = list(percent(LN_YESNO) ),
            margins = T, data = bca)
```

```
 ---------------------------------
         --------LN_YESNO---------
 age.gr          0       1   Total
 ---------------------------------
 20-49        67.8    32.2   100.0
 50-64        80.6    19.4   100.0
 65-89        83.2    16.8   100.0

 Total        77.0    23.0   100.0
 ---------------------------------
```

2. How does *LN_YESNO* seem to depend on *HISTGRAD*?

Create another 2-way frequency and percentage table as in 1. but now with *HISTGRAD* as the row variable.

```
stat.table( index = list(age.gr, HISTGRAD),
            contents = list(percent(HISTGRAD) ),
            margins = T, data = bca)
```

```
 ----------------------------------------
         -----------HISTGRAD-------------
 age.gr          1       2       3   Total
 ----------------------------------------
 20-49         7.7    49.1    43.2   100.0
 50-64         8.7    57.0    34.3   100.0
 65-89         9.5    62.5    28.0   100.0

 Total         8.6    55.9    35.5   100.0
 ----------------------------------------
```

3. Multidimensional tables are challenching, especially when they need to be interpreted to people. However, suppose we are interested knowing if the association between histological grading (*HISTGRAD*) and lymphatic nodes (*LN_YESNO*) is the same by age groups (*age.gr*) of patients.

Ignoring the age:

```
stat.table( index = list(HISTGRAD,LN_YESNO),
            contents = list(count(),percent(LN_YESNO) ),
            margins = T, data = bca)
```

```
 ---------------------------------
           --------LN_YESNO---------
 HISTGRAD          0       1   Total
 ---------------------------------
 1                71       8      79
```

```
               89.9    10.1   100.0

 2              394     120     514
               76.7    23.3   100.0

 3              227     100     327
               69.4    30.6   100.0


 Total          929     278    1207
               77.0    23.0   100.0
 -----------------------------------
```

Let's do this with data.table to illustrate the usefullness of it.

```
bcadt<-bcadt[!is.na(bca$HISTGRAD),] #remove missing observations for histological grading
freq_tab<-bcadt[order(age.gr,HISTGRAD,LN_YESNO), .(.N), by = list(age.gr,HISTGRAD,LN_YESNO)] # make tab

res<-bcadt[order(age.gr,HISTGRAD),
           .(percentage = round(100*tabulate(LN_YESNO)/.N)),
           by = list(age.gr,HISTGRAD)]

print(res)
```

```
    age.gr HISTGRAD percentage
1: [20,50)        1         16
2: [20,50)        2         32
3: [20,50)        3         38
4: [50,65)        1          7
5: [50,65)        2         21
6: [50,65)        3         28
7: [65,90)        1          8
8: [65,90)        2         17
9: [65,90)        3         21
```

8. Examining the properties of a table object.

1. Print again the 2-way table *grbyage* created above. Apply the functions *length()* and *sum()* on the table object *grbyage*. It seems like these functions treat the table as if it were a numeric vector ...

```
length(grbyage)
```

```
[1] 9
```

```
sum(grbyage)
```

```
[1] 920
```

2. Continue examining the inner structure of the table object by functions *class(), mode(), dim() and str()* What information do these provide?

```
class(grbyage)
```

```
[1] "table"
```

```
mode(grbyage)
```

```
[1] "numeric"
```

```
dim(grbyage)
```

```
[1] 3 3
```

```
str(grbyage)
```

```
 'table' int [1:3, 1:3] 25 28 26 159 183 172 140 110 77
 - attr(*, "dimnames")=List of 2
  ..$ : chr [1:3] "20-49" "50-64" "65-89"
  ..$ : chr [1:3] "1" "2" "3"
```

3. The cells of any 2-dimensional table are accessed using double indexing A given row (column) is identified by leaving the column number (row number) empty. – Leaning on this instruction, print only the following selected items from the table object *grbyage*:

- the cell frequency in the crossing of the 2nd row ja 2nd column,
- the frequencies of the whole 2nd row; compute and print also their sum,
- the frequencies of the whole 2nd column; compute and print also their sum.

```
grbyage[2,2]
```

```
[1] 183
```

```
print(grbyage[2,])
```

```
  1   2   3
 28 183 110
```

```
print(sum(grbyage[2,]))
```

```
[1] 321
```

```
print(grbyage[,2])
```

```
20-49 50-64 65-89
  159   183   172
```

```
print(sum(grbyage[,2]))
```

```
[1] 514
```

9. Additional task

If you managed to do the previous tasks within the time allocated, get familial with data.table functions https://cran.r-project.org/web/packages/data.table/vignettes/datatable-intro.html