Using R for Statistics in Medical Research

Elrozy Andrinopoulou
e.andrinopoulou@erasmusmc.nl
Sten Willemsen
s.willemsen@erasmusmc.nl

25 February - 1 March 2019

Department of Biostatistics, Erasmus University Medical Center

Practical 1

Practical 1.1

- Create 2 vectors of size 50
  - Sex: takes 2 values 0 and 1
  - Age: takes values from 20 till 80 (hint: use `sample()`)
  - convert “Sex” to a factor with levels 0: “female” and 1: “male”
  - define the new variable “AgeCat” as dichotomous with “Age”$<=$50 to be 0 and 1 otherwise
  - convert “AgeCat” to a factor with levels 0: “young” and 1: “old”
  - overwrite “Age” variable by $\frac{Age - mean(Age)}{sd(Age)}$

Save your code

Practical 1.2

- Create a data.frame with the name “DF”
  - including the following vectors: “Sex,” “Age”, “AgeCat”
  - with names: “Gender”, “StandardizedAge”, “DichotomousAge”
  - What are the dimensions of the data.frame (hint: use `dim()`)?

Practical 1.2.1 -extra

- Create 2 vectors of size 150
  - Treatment: takes 2 values 1 and 2
  - Weight: takes values from 50 till 100
  - convert “Treatment” to a factor with levels 1: “no” and 2: “yes”
  - overwrite “Weight” variable by “Weight”∗1000
  - create a data.frame including “Treatment” and “Weight”

Practical 1.2.2 -extra

- Create a list called “my_list” with the following:
  - let: “*a*” to “*i*” (hint: use `letters`)
  - mat: matrix $\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$
  - sex: factor taking the values “males” and “females” and length 50

Practical 2

Practical 2.1

- Using the data.frame “DF” from Practical 1
  - calculate the mean of the variable “StandardizedAge”
  - calculate the standard deviation of the variable “StandardizedAge”
  - calculate the frequencies of the variable “Gender” (hint: use function `length()` or `table()`)
  - calculate the frequencies of the variable “DichotomousAge”

Practical 2.2

- Using the data.frame “DF” from Practical 1
  - calculate the mean of the variable “StandardizedAge” for males and females
  - show the crosstab including “Gender” and “DichotomousAge” (hint: use function `table()`)
  - select all males that are young
  - select all females or old patients

Practical 2.2.1 -extra

- Using the data.frame “DF” from Practical 1
  - calculate the median of the variable “StandardizedAge”
  - calculate the median of the variable “StandardizedAge” for males and females
  - show the 2nd column of DF

Practical 3

Practical 3.1

- Create a function with the name “Descriptives” that
  - takes as values 2 vectors
  - calculates the min, median, max, mean, sd and length of both vectors
  - returns a matrix where rows indicate the statistic and columns the vectors (hind: to name the rows of the matrix use `rownames()`)
- Use the function for the following vectors
  - vec1: sample “age” from 20 till 80 and with n = 100
  - vec2: sample “weight” from 60 till 100 and with n = 200

Practical 3.1.1 -extra

- Create a function with the name “Summaries” that
  - takes as value a data.frame that includes 2 columns: 1st categorical, 2nd continuous
  - calculates the mean and median per group (variable in 1st column) for the variable in the 2nd column
  - returns a matrix where rows indicate the mean/median and columns the groups
- Use the function for the data.frame that includes the following
  - vec1: sample “age” from 20 till 80 and with n = 100
  - vec2: sample “sex” takes 0 and 1, with n = 100 (convert also to a factor)

Practical 4

Practical 4.1

- Import the spss dataset: Cancer.sav
  - we are interested in testing if there is a difference in the average age between the 2 treatment groups (t-test)
  - we obtained the result, but it was nowhere saved! Save the result of the t-test in an object named ’res’ for post processing
  - using indexing extract the p-value
  - using indexing extract the 95% CI

Practical 4.2

- We want to save those results for reporting in our paper. Create a data.frame with the name “Restest”, with columns
  - mean1: mean for TRT = 0
  - mean2: mean for TRT = 1
  - diffres: difference of the means
  - lowCI: lower limit of confidence interval
  - upCI: upper limit of confidence interval
  - pVal: p-value of the test

Practical 4.3

- Using the dataset: Cancer.sav
  - create a dichotomous variable for “WEIGHIN”: 0 if “WEIGHIN” < mean(“WEIGHIN”) and 1 otherwise. Use the name “WEIGHINcat”
  - perform a t-test comparing “AGE” between the “TRT” groups
  - perform a t-test comparing “AGE” between the “WEIGHINcat” groups
  - create a vector with the name “results”, including the p-values of the above tests
  - calculate the significance level after the bonferroni correction. Use the name “bonf.corr_alpha”
  - using an if statement, print whether the null hypothesis is rejected or not

Practical 4.4

- Using the Cancer.sav dataset
  - perform a regression analysis with dependent variable: WEIGHIN and independent: TRT - use the name fm1
  - perform a regression analysis with dependent variable: WEIGHIN and independent: TRT + STAGE - use the name fm2
  - compare the 2 models
  - perform a regression analysis with dependent variable: WEIGHIN and independent: TRT + STAGE + AGE - use the name fm3
  - include an interaction term of AGE and STAGE in the previous model - use the name fm4

Practical 4.5

- Using the Cancer.sav dataset
  - create a function with the name “Pred” that takes as values: the treatment, stage and age of a new patient and returns the predicted weight using model fm4
    - Predict the weight of a new patient: treatment = 1, stage = 2 and age = 77
  - plot the weight vs the age per treatment group (xyplot with 2 figures). The x-axis will have the name “age”, the y-axis will have the name “weight” and the main title will be “weight vs age”

Practical 4.5.1 -extra

- Create the following vectors of size 100
  - sex: takes 2 values 1 and 2
  - tr: takes 4 values 1 to 4
  - score: normal distribution with mean 20 and sd 5
- Create boxplots of score per sex and tr groups
- Replace the above vectors using n = 20 instead of 100
- Create new boxplots of score per sex and tr groups
- What can you say about the distrubution?

Practical 4.5.2 -extra

- Using the Cancer.sav dataset
  - what is the Pearson correlation between “WEIGHIN” and “AGE”?
  - perform the test, what is hypothesis and the decision?
  - create a scatterplot to support your findings
  - create a boxplot and investigate the col argument

Practical 5

Practical 5.1

- Using the Cancer.sav dataset, create a function “uniRegr” that
  - takes a data.frame and a vector with the names of the independent covariates as c(“ ”, “ ”, “ ”, …)
  - performs univariable regression analysis of the outcome:“WEIGHIN” and each covariate (hint: use `as.formula()` and `paste()`)
  - returns the estimate, std error and the p-values as a data.frame
- Using the Cancer.sav dataset, perform univariable regression analysis using the covariates “AGE”, “STAGE” and “TRT”
- Save the results as “results_regr”

Practical 5.1.1 -extra

- Using the Cancer.sav dataset, create a function “uniLogRegr” that
  - takes a data.frame and a vector with the names of the independent covariates as c(“ ”, “ ”, “ ”, …)
  - performs univariable **logistic** regression analysis of the outcome:“TRT” and each covariate (hint: use `as.formula()` and `paste()`)
  - returns the estimate, std error and the p-values as a data.frame
- Using the Cancer.sav dataset, perform univariable logistic regression analysis using the covariates “AGE”, “STAGE” and “WEIGHIN”
- Save the results as “results_log_regr”