Department of Psychology, The University of Edinburgh

Data Analysis for Psychology in R 1

Contents

List of Tables

List of Figures

Course overview

Course description

Data analysis for psychology in R 1 or, in short, **DAPR1**, is your first step on the exciting road of analysing data, and using data to answer interesting questions in the ways that statisticians, researchers, data scientists do.

This year-long course is designed to work gently through conceptual content that form the basis of understanding and working with data to perform statistical testing. It will provide you with the foundations in working with data and probability theory, and build to learning about how we make inferences about our hypotheses in psychology. Concepts are demonstrated in R throughout, taking you from basic calculations to the foundations of data management, plotting and use of simple statistical tests.

The overall aim of the course is to provide you with all the necessary skills to feel confident in the basics of statistics, R and data, before we move on to discuss a broader array of statistical methods in year 2.

Team

- Dr Tom Booth, *Senior Lecturer*: Tom.Booth@ed.ac.uk

- Dr Josiah King, *Senior Teaching Coordinator*: ug.ppls.stats@ed.ac.uk
- Dr Umberto Noe, *Senior Teaching Coordinator*: ug.ppls.stats@ed.ac.uk
- And not forgetting your friendly tutoring team! Ask them anything!

Schedule

SEMESTER 1

Week	Topic
1	Collecting data
2	Types of data
3	Visualising distributions
4	Describing distributions
5	Visualising and describing relationships
Break	
6	Basics of probability theory
7	Probability rules
8	Random variables
9	Sampling variability and sampling distributions
10	Bias-variance trade-off

SEMESTER 2

Week	Topic
11	Bootstrap & Confidence Intervals
12	Hypothesis testing with the p-value approach
13	Hypothesis testing with the critical values approach
14	Hypothesis testing & Confidence Intervals
15	Making decisions - Effect sizes, Power, Errors
Break	
16	Test for one mean
17	Test for two means (paired samples)
18	Test for two means (independent samples)

Week	Topic
19	Chi-square test
20	Covariance, correlation, and looking ahead

Textbook

TBC

0.1 Collecting Data

Learning Objectives

- LO1: Install R & Rstudio, and get comfortable with the layout
- LO2: Learn about how data is stored in R
- LO3: Produce your first Rmarkdown document

R & Rstudio

Installing R & Rstudio

install tidyverse, rmarkdown, knitr etc.

Intro to Rstudio

useful settings - environment clearing soft-wrapping text

R as a calculator

Rscripts, Rmarkdown

Using R packages

Data in R

assignment `<-` vector matrix data.frame tibble

accessors [], $

Tasks for today

Task Open an Rmarkdown document.

File > New File > R Markdown..

When the box pops-up, give a title of your choice ("Intro lab", maybe?) and your name as the author.

The file which opens will have some template stuff in it. Delete everything below the first code chunk to start with a fresh document:

TODO IMAGE

Task Insert a new code chunk by either using the Insert button in the top right of the document, and clicking R, or typing *Ctrl + Alt + i*

Inside the chunk, type:
`print("Hello world! My name is ?")`.
Run the chunk and see that the output will be printed below.

Task We're going to use some functions which are in the **tidverse** package, which we have already installed earlier on.
However, to *use* the package, we need to load it using `library(tidyverse)`.

Important: When writing analysis, we want it to be *reproducible*. We want to be able to give somebody else our code and the data,

and ensure that they can get the same results. To do this, we need to show what packages we use.
It is good practice to load any packages you use at the top of your code.

In your first code chunk, type:

```
#I'm going to use these packages in this document:
library(tidyverse)
```

and run the chunk.

Note that the `#` in R code makes that line a comment, which basically means that R will ignore the line. Comments are useful for you to remind yourself of what your code is doing.

Task Below the code chunk, add a new line with the following:

```
# R code examples
```

When the # is used in a Rmarkdown file, **outside** of a code-chunk, it makes that line a heading when we finally get to *compiling* the document. Below, what you see on the left will be compiled to look like those on the right:

Task In your Rmarkdown document, give an explanation of what each of the following does in R. Write the explanation, and give an example in a code chunk.

- `+`
- `-`
- `*`
- `/`
- `()`
- `^`
- `<-`
- `=`
- `<`

- `>`
- `<=`
- `>=`
- `==`
- `!=`
- `data.frame()`
- `c()`
- `[]`
- `$`

You can see an example of the first few below: TODO IMAGE

Solution Hopefully you managed to include examples for all of these, but we have provided a table below describing what each symbol does.

Symbol	Description	Example
`+`	Adds two numbers together	`2+2` - two plus two
`-`	Subtract one number from another	`3-1` - three minus one
`*`	Multiply two numbers together	`3*3` - three times three
`/`	Divide one number by another	`9/3` - nine divided by three
`()`	group operations together	`(2+2)/4` is different from `2+2/4`
`^`	to the power of..	`4^2` - four to the power of two, or four squared
`<-`	stores an object in R with the left hand side (LHS) as the name, and the RHS as the value	`x<-10`

Symbol	Description	Example
`=`	stores an object in R with the left hand side (LHS) as the name, and the RHS as the value	`x = 10`
`<`	is less than?	`2<3`
`>`	is greater than?	`2>3`
`<=`	is less than or equal to?	`2<=3`
`>=`	is greater than or equal to?	`2>=2`
`==`	is equal to?	`(5+5) == 10`
`!=`	is not equal to?	`(2+3) != 4`
`c()`	combines values into a vector (a sequence of values)	`c(1,2,3,4)`
`data.frame()`	converts whatever is inside the brackets to a dataframe	`data.frame(values = c(1,2,3,4))`
`[]`	used to extract the 1st, 2nd, … i^{th} elements in a set of numbers	`myvector[3]`
`$`	used to extract a named column from a dataframe	`mydata$age_variable`

Task Outside of a code chunk, add a new heading with the following:

```
# Vectors and dataframes
```

Task In a new code chunk, do the following:

1. store the following numbers as an object in R:

4,7,3,1,8,9,5,2,2,6,9,9,5,20

2. Try using the function `sum()`, with the name of your object inside the brackets.

Solution Because this is just a set of numbers, we store it as a **vector**, using `c()`.
We have named it `myvec`, but you can call yours whatever you like.

The `sum()` function will add all of the numbers together!

```
myvec <- c(4,7,3,1,8,9,5,2,2,6,9,9,5,20)
sum(myvec)
```

```
[1] 90
```

Task Using the square brackets - `[]` - pull out the 2nd, 4th and 6th values in the object you just created.

Hint: You will need to put inside the square brackets a *sequence of* numbers. How do we combine numbers in to a sequence in R? using `c()`!

Solution

```
myvec[c(2,4,6)]
```

```
[1] 7 1 9
```

Task Store the names and birth-years of the Beatle in the appropriate format in R. Name it `beatles`. John was born in 1940 Paul was born in 1942 George was born in 1943 Ringo was born in 1940

Hint: We're going to have two sequences here, the names, and the birth-years. The easiest way to think of this would be to have a row for each Beatle, and a column for each of name and birth-year.

Check dimensions of the object using `dim()`. How many rows and how many columns are there?

Solution TODO - decide dataframe/tibble We want a dataframe/tibble

```
beatles <-
  data.frame(
    name = c("John","Paul","George","Ringo"),
    birthyear = c(1940,1942,1943,1940)
)
beatles <-
  tibble(
    name = c("John","Paul","George","Ringo"),
    birthyear = c(1940,1942,1943,1940)
)
```

```
dim(beatles)
```

```
[1] 4 2
```

Four rows, and two columns!

So far, we've been manually inputting our data. However, R can read in data which has been created elsewhere (like in excel, or by some software which is used to present participants with experiments).

TODO - link to data we're going to use next couple of weeks

First, click on the following link: link

It should open a webpage and show you a dataset. There are two things to note.

- the values are separated by commas
- the url (in the top bar of your browser) ends with **.csv**.

This stands for 'comma separated value'.
The **tidyverse** package which we loaded at the top has a function to read this sort of data: `read_csv()`.

Task Read the data into R. You can do this by giving `read_csv()` the url.

Hint: If you just type:

```
read_csv("https://uoe-psychology.github.io/uoe_psystats/multivar/data/women_
```

```
# A tibble: 192 x 3
    date field            pct_women_majors
   <dbl> <chr>                       <dbl>
 1  1966 Computer.science            0.146
 2  1967 Computer.science            0.108
 3  1968 Computer.science            0.12
 4  1969 Computer.science            0.13
 5  1970 Computer.science            0.129
 6  1971 Computer.science            0.136
 7  1972 Computer.science            0.136
 8  1973 Computer.science            0.149
 9  1974 Computer.science            0.164
10  1975 Computer.science            0.19
# ... with 182 more rows
```

It will print out the dataset, but it won't store it in R for you to do things with. To do that, we want to assign a name to the data:

```
mydata <- read_csv("https://uoe-psychology.github.io/uoe_psystats/multivar/d
```

Note that it now turns up in the *Environment* pane of Rstudio.

Task Check how many rows and columns you have in the dataset. You can do this with the `dim()` function.

Solution

```
dim(mydata)
```

```
[1] 192   3
```

Task Using the square brackets, show the 167th row, with all columns.

Remember: When you are using `[]` with a dataframe, you specify `data[rows, columns]`. If you leave either rows or columns blank it will give all of them - for instance, `data[ , columns]` will give you all rows for some specified columns.

Solution

```
mydata[167,]
```

```
# A tibble: 1 x 3
   date field             pct_women_majors
  <dbl> <chr>                        <dbl>
1  1988 Physical.Sciences             0.32
```

Task By now, you should have an Rmardkown document (**.Rmd**) with your answers to the tasks we've been through today.

Compile the document by clicking on the **Knit** button at the top. The little arrow to the right allows you to compile to either **.pdf** or **.html**.

Glossary

0.2 Types of data

Learning Objectives

- LO1:
- LO2:
- LO3:

0.3 Visualising distributions

Learning Objectives

- LO1:
- LO2:
- LO3:

0.4 Describing distributions

Learning Objectives

- LO1:
- LO2:
- LO3:

0.5 Visualising and describing relationships

Learning Objectives

- LO1:
- LO2:
- LO3:

0.6 Basics of probability theory

Learning Objectives

- LO1:
- LO2:
- LO3:

0.7 Probability rules

Learning Objectives

- LO1:
- LO2:
- LO3:

0.8 Random variables

Learning Objectives

- LO1:
- LO2:
- LO3:

0.9 Sampling variability and sampling distributions

Learning Objectives

- LO1:
- LO2:
- LO3:

0.10 Bias-Variance Trade-off

Learning Objectives

- LO1:
- LO2:
- LO3: