
DESE71006

Tutorial 1: Intro to R and R Shiny

The aim of this tutorial session is to get you started with R and with R Shiny.

R is a programming language that is quite easy to work with. It has similarities with Matlab – for instance, a Matlab/R reference guide is available online ([Matlab/R reference guide](#)) and is very useful if you are comfortable with Matlab. However, in contrast to Matlab, R was originally developed with a 100% focus on data science. It had great data science and visualisation packages much before Python!!

There are also a number of very good cheat sheet for basics of R, see e.g., [R cheat sheet 1](#) and [R cheat sheet 2](#).

More generally in terms of resources, excellent introduction books to R are the following:

Alex Douglas, Deon Roos, Francesca Mancini, Ana Couto, David Lusseau (2023) *An introduction to R*. available online at intro2r.com/

Hadley Wickham, Mine Çetinkaya-Rundel, Garrett Golemund (2023) *R for Data Science* (2nd edition). O'Reilley, available online at r4ds.hadley.nz/

For R Shiny, you may rely on the following resources (which we will be using as a basis anyway):

Hadley Wickham (2020) *Mastering Shiny*. O'Reilley, available online at [mastering-shiny.org](https://mastering-shiny.org/shiny.posit.co/r/getstarted/shiny-basics/lesson1/index.html)
shiny.posit.co/r/getstarted/shiny-basics/lesson1/index.html

And, you may want to have this cheat sheet available: rstudio/cheatsheets/main/shiny.pdf

Finally, for inspiration and code to produce nice visualisations, you can always have a look at:

data-to-viz.com

r-graph-gallery.com

Activity 1: Making some simple plots in R... and monthly weather data at Heathrow

Weather data are some of the easiest data to find and use. We will then use data I got from Heathrow airport as input to this problem, to get you used to data handling and plotting with R. The data was downloaded from metoffice.gov.uk/pub/data/weather/uk/climate/stationdata/heathrowdata.txt

The historic station data at Heathrow includes (on a monthly basis, for many many years):

- Mean daily maximum temperature (**tmax**)
- Mean daily minimum temperature (**tmin**)
- Days of air frost (**af**)
- Total rainfall (**rain**)
- Total sunshine duration (**sun**)

Step 1. We always need to get the data first. In R, there are various options to scrape data or to load data directly from a url. We will use the latter for now. To efficiently read csv files, we can use:

```
WeatherData <- read.csv("url", header=TRUE)
```

The first entry is the actual url, while the second one tells that the file has a header (i.e., giving the variable names). To know what a R function does, you can always use "?". For example,

```
?read.csv
```

will tell you what you need to know about that function. In parallel, in the above, `WeatherData` is the variable that will then contain the data, and "`<=`" is the assignment operator.

Get the data from the following variable and assign it to the variable of your choice (you can also use "`WeatherData`"):

<http://pierrepinson.com/wp-content/uploads/2023/10/Heathrow-weather-data-1948-2022.csv>

Step 2. When the data is loaded, we can explore some basic things with the data. For instance, you can visualise the raw data by simply typing

```
WeatherData
```

It is also possible to check the dimensions of that matrix with the `dim` function,

```
dim(WeatherData)
```

You can also look at a summary of the contents and characteristics/statistics of the variables composing `WeatherData`:

```
summary(WeatherData)
```

Since we have variables for each column, we can then specifically look at a specific variable:

```
WeatherData$tmin
```

Step 3. Let us try our first plot then. Have a look at the help for the `plot` function first. Then, we use the simplest way to plot a time-series (e.g., minimum temperature), by simply plotting the vector itself:

```
plot(WeatherData$tmin)
```

Does that look good to you? What should we change there?

Step 4. Looking at the documentation for the `plot` function, find a way to make it a line plot, with the colour of your choice, and to control the labels for the x and y axes. You may call them "months [starting from 1.1948]" and "degrees [Celsius]" for instance.

Step 5. We can have even more control on the axes, by first removing them from the plot, and then adding them back in the way we want. For instance, try

```
plot(WeatherData$tmin, type="l", col="blue", ylim=c(-5,20),
     xlab="years", ylab="degrees [Celsius]", axes=FALSE)
axis(1, c(seq(1,length(WeatherData$tmin),20*12),length(WeatherData$tmin)),
     c(seq(1948,2008,20), 2022))
axis(2, seq(-5,20,5))
```

Note that we also control here the range of the y-axis. In parallel, look at the R cheat sheet mentioned above to see what `c()` and `seq()` are for.

Step 6. What if instead I want to look at the climatology (i.e., average conditions, as a function of time of year)? We would first need to calculate averages, per month, over the 75 years of data available. For that, we would first need to extract the vector interest, reshape it, and then take an average per row. This may be done for instance with

```
tmin.vect <- WeatherData$tmin # extract the variable of interest
dim(tmin.vect) <- c(12, length(tmin.vect)/12) # transform the vector into a matrix
tmin.mean <- rowMeans(tmin.vect) # calculate the mean per row
```

Now, produce a nice plot for the climatology of min temperature at Heathrow, as a function of the month of the year.

Step 7. It may be good to have both min and max temperatures on the same plot. To add an extra plot in the same figure, we can use `par(new=TRUE)` and then add the second plot command. Generate that plot.

Optional: Add a legend...

Step 8. Finally, let us export that figure. Look at the `png` function to see how this works. Typically you should start with

```
png("Heathrow-T-climatology.png", width=600, height=450, pointsize=12)
```

then, have the set of plot commands (i.e., plot and axis), to finally use

```
dev.off()
```

to get the export completed.

Activity 2: My first steps with R Shiny

Go through the activities and exercises at the following links. Note that we will start these activities during the Tutorial session and you will have to finish them as homework for the following week (i.e., by next tutorial).

You will need to focus on:

- Your first R Shiny app – mastering-shiny.org/basic-app.html
- Basic UI – mastering-shiny.org/basic-ui.html

Activity 3 (optional - only if you have time and want to go the extra mile): A first dashboard

Produce a dashboard requiring 2 inputs from the user:

- the url to load the Heathrow weather data (free text input)
- the variable they would like to plot (using, e.g., radio buttons)

Based on these inputs, the dashboard produces a plot showing the climatology for that meteorological variable at Heathrow. Keep it as simple as possible...