Introduction to R

R Basics

Arun Frey

Basic commands

Command	What it does
`class(x)`	See the class of an object
`cor.test(x,y)`	Conduct a correlation test between variables between x and y
`head(data)`	Shows first six rows of the data
`hist(x)`	Create a histogram of a variable
`length(x)`	Compute the length of a vector
`ls()`	Lists all the objects that are stored in your environment
`mean(x)`	Compute the mean of a variable
`median(x)`	Compute the median of a variable
`quantile(x)`	Compute the quantiles of the variable
`read.csv(data)`	Load data into R (remember to assign this to an object, by using `<-`!)
`plot(y, x)`	Create a scatterplot that show the relationship between variables x and y
`sum(x)`	Calculate the sum of the vector's values
`summary(data)`	Compute summary statistics of the data/variable (mean, standard deviation, etc.)
`sd(x)`	Compute the standard deviation of a variable
`tail(data)`	Shows last six rows of the data

- To look up how to use commands/options, use the help function in R: `help(command)` or `?` command.

- Use `$` to call a variable. For example, if I have a dataset called data that contains a variable called `economic_growth`, use `dataset$economic_growth`. (Hint: you can use auto-complete by using the tab button on your keyboard once you have entered the first letter of the variable name after `$`.)

- Use `<-` to assign a value or name to an object. For example, to assign the value 5 to an object called a, use `a <- 5`.

- When you want to calculate the mean/median/standard deviation of a variable that has missing values, use the `na.rm=TRUE` option. E.g. `mean(dataset$economic_growth,na.rm=TRUE)`.

- To combine multiple plots, use the `par(mfrow=c(number_of_rows,number_of_columns))`.

Exercise 1

Revisit the code we saw at the beginning of the lecture yesterday:
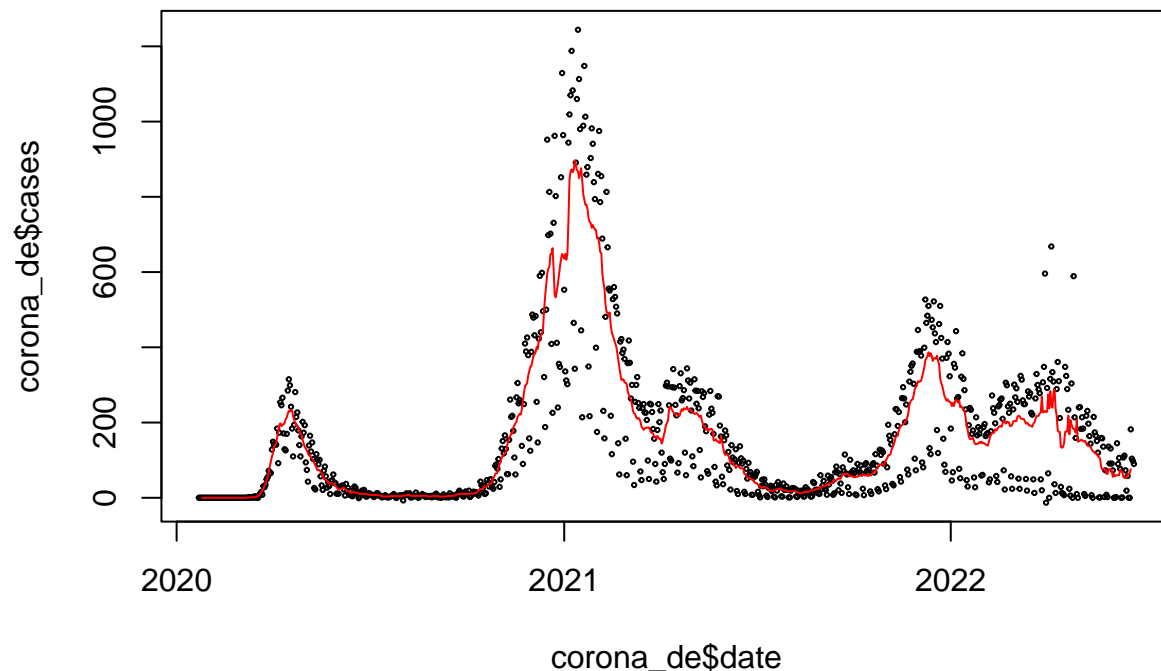
```r
library(coronavirus)

corona_de <- coronavirus[coronavirus$country=="Germany" &
                          coronavirus$type=="death", ]

corona_de$death_7 <-  zoo::rollmean(corona_de$cases, k = 7, fill = NA)

plot(x = corona_de$date,
     y = corona_de$cases,
     cex = 0.3)

lines(x = corona_de$date,
      y = corona_de$death_7,
      type = "l",
      cex = 1.5,
      col = "red")
```


1. Try to understand what each step is doing. Once you understand what's going on, recreate the plot in RStudio. Note that you might have to install some packages for this to work.

2. This plot isn't great: no legible axes labels, no title. Try to improve the formatting of the plot by adding axes labels and a title.

Exercise 2:

Next we will start working on our own little project.

1. Create a new folder on your Desktop called "intro-to-R_project".
2. In RStudio, click on File > New Project > Existing Directory, and then select the folder "intro-to-R_project" on your Desktop as your project directory.

3. Inside this folder, you will now have an RProject object. It's good practice to create a new R Project whenever you start working on a new project.

Head over to the `02-into-to-R_worksheet.R` file and start exploring the data.