R Language

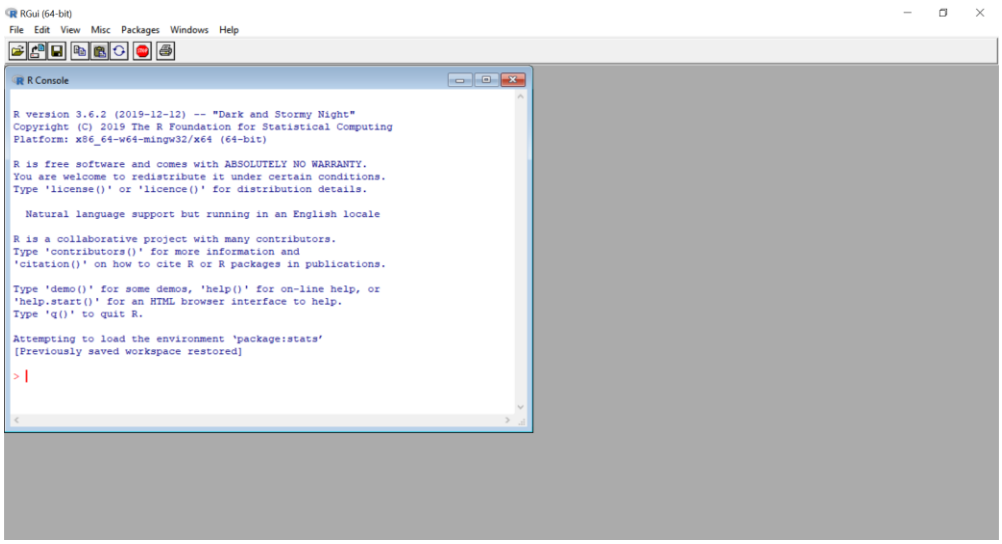
R is an open source language and it is available to download and install to run on Windows, OS X and a wide variety of Unix platforms.

The primary part or R language is available from its Comprehensive R Archive Network (CRAN) - https://cran.r-project.org/

Many add-on packages used to extend the functionality of R language are also hosted in the CRAN.

On overall it consists of 2 conceptual parts:

1. The "base" R part that can be download from CRAN: Linux, Windows or Mac. It consists of the most fundamental functions or R
2. The add on part with all other packages to carry out data preparation and data analytics such as class, cluster, nlme, rpart, spatial and nnet



R Integrated Development Environment

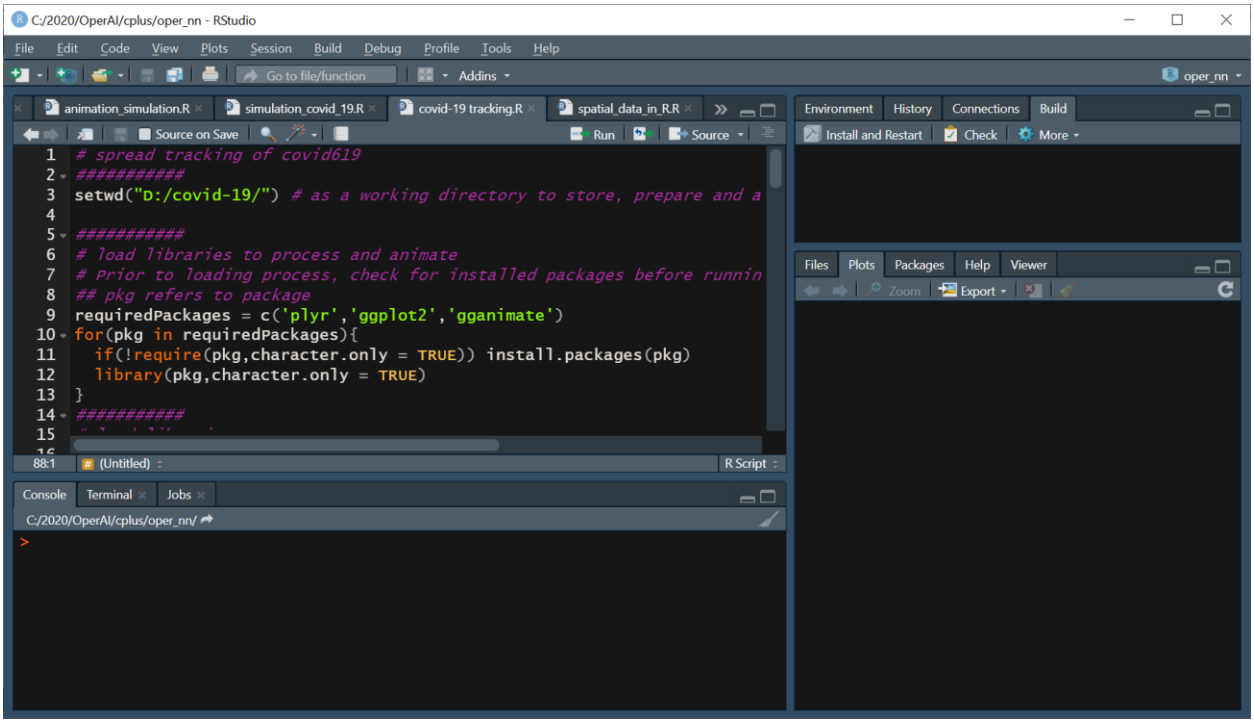
R has also a drop-menu as well as an integrated development environment (IDE) to keep track of the work environment - RStudio

R has also its integrated development environment (IDE) RStudio with features similar to other coding platforms such as Matlab.

RStudio has as syntax highlighting with four-pane workspace to type and view commands and to view results and visualizations.

File | Edit | Code	
command-line scripting and editing with Data (input)	Tracking space of current and Previous commands/files
Interactive console One line command at a time	Displays of graphs and plots (output)

1. The top left window of RStudio is in the console where command-line scripting and editing space to create a file with multiple lines of R code.
2. Bottom left space is the interactive console for only one line at a time. It displays the output of the calls as well as any lines of code that are run from the editor window.
3. The top right window is for tracking the work and also displaying the objects that are currently in use or connected to (via the cloud) along with a history tab with a list of prior code commands.
4. The space at the bottom right displays the outputs as graphs and plots. It can helped also to trace back the display based also on a history tab with options to export the outputs as images in different formats.



Installing R packages

In addition to the development of scripts and commands R allow to use also plenty packages.

Most of these packages can be installed from the Comprehensive R Archive Network. The command for install¬ing a package is:

```
> install.packages("package-name")
```

In RStudio there is a packages tab to "Install Packages." To know which packages are already available with the installation on the current working platform, type:

```
> installed.packages()
```

Prior to use any package in your working platform once it's installed, it needs to be loaded with the following command:

```
> library("package-name")
```

These packages are changing all the time and to stay up to date with any package, type:

```
> update.packages()
```

There also the option to drop any package from your system using the function:

```
> remove.packages("thepackagename")
```