An introduction to

What is R?

- Software environment for statistical computing and graphics
- Programming language based on language *S*
- “Open source” → free, anyone can use it (and also contribute!)
- Latest R version: R 4.3.2 (October 2024)
- Download from www.cran.r-project.org (version according to your operating system, e.g. Unix, Microsoft, Mac, etc.)

Structure of R

- R is organized in packages
- Each package provides particular functions
- Pre installed packages include most common functions (simple descriptive statistics, graphs, models, etc.)
- Public server („CRAN mirrors“) provide additonal packages

Load additonal packages in R

- Example: you would like to install the *psych* package

- First, download the *psych* package to make it available in your personal library using the command

```
install.package(psych)
```

- Second, make the functions of *psych* available in your actual working directory using the command

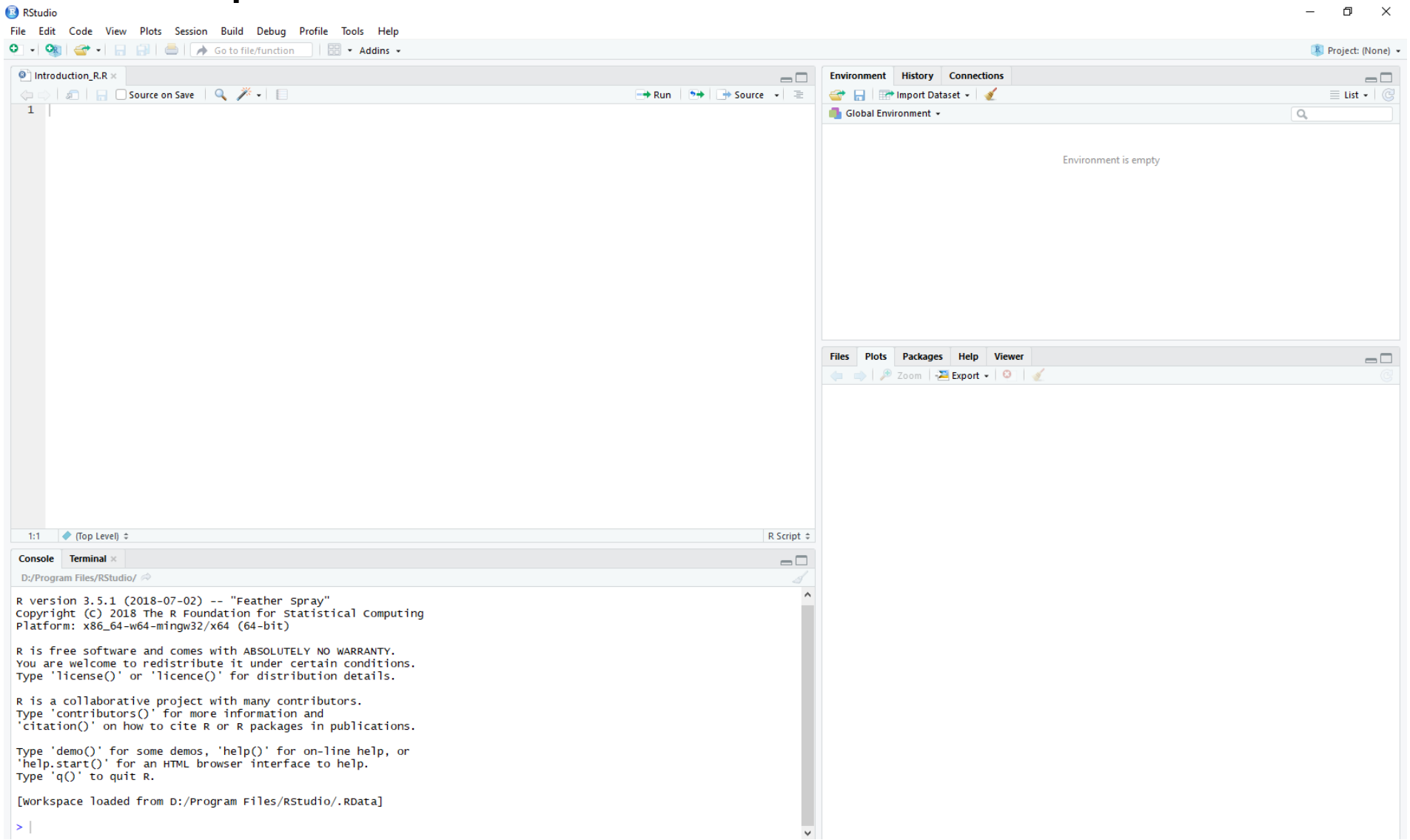
```
library(psych)
```

R Studio

- R is not very intuitive if you are not experienced with programming
- R Studio is a user interface for R which is more user friendly, but this will not protect you from a little programming…

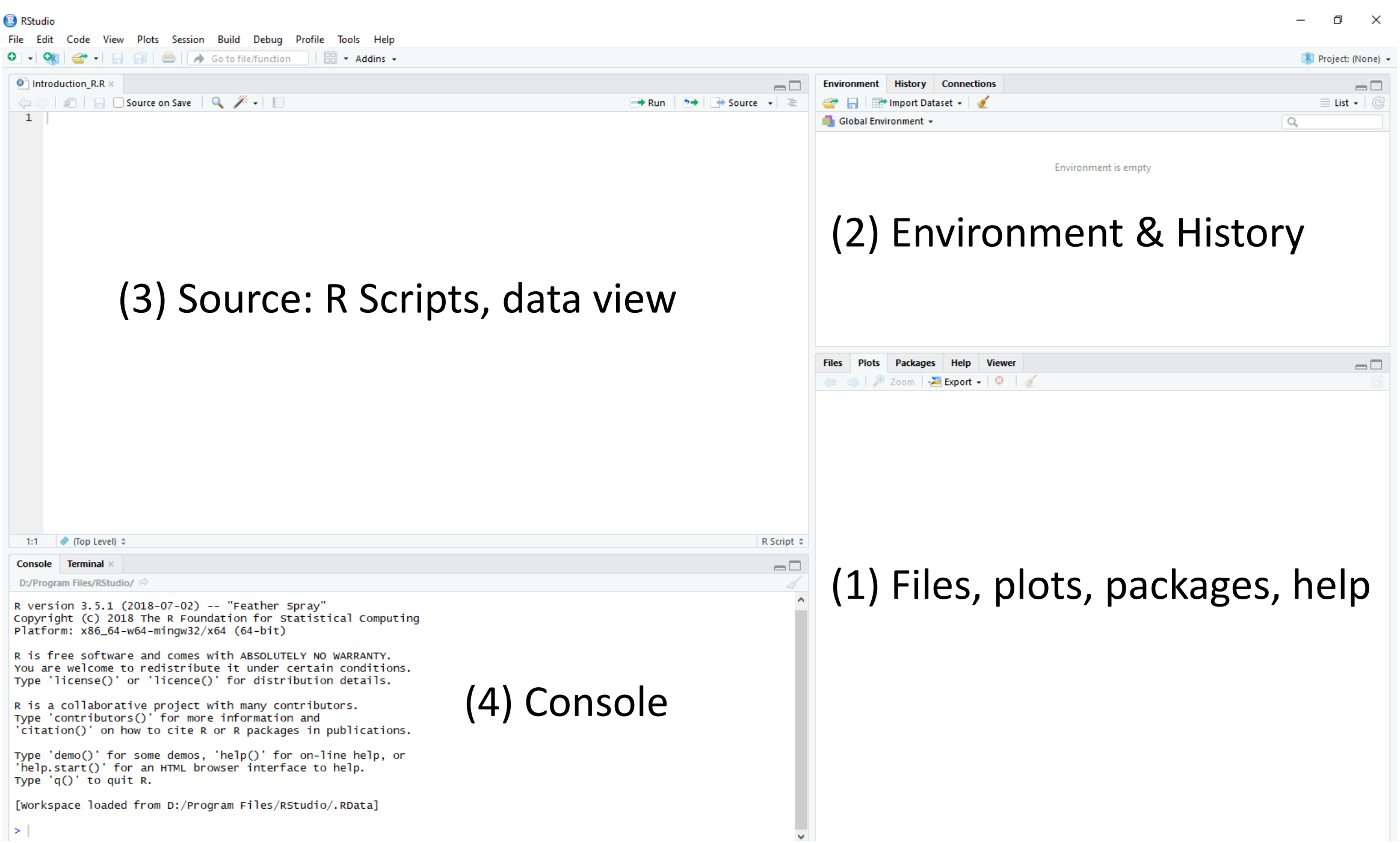
- Download free open source license from www.rstudio.com
- We will work with R Studio, so you need to download R and R Studio

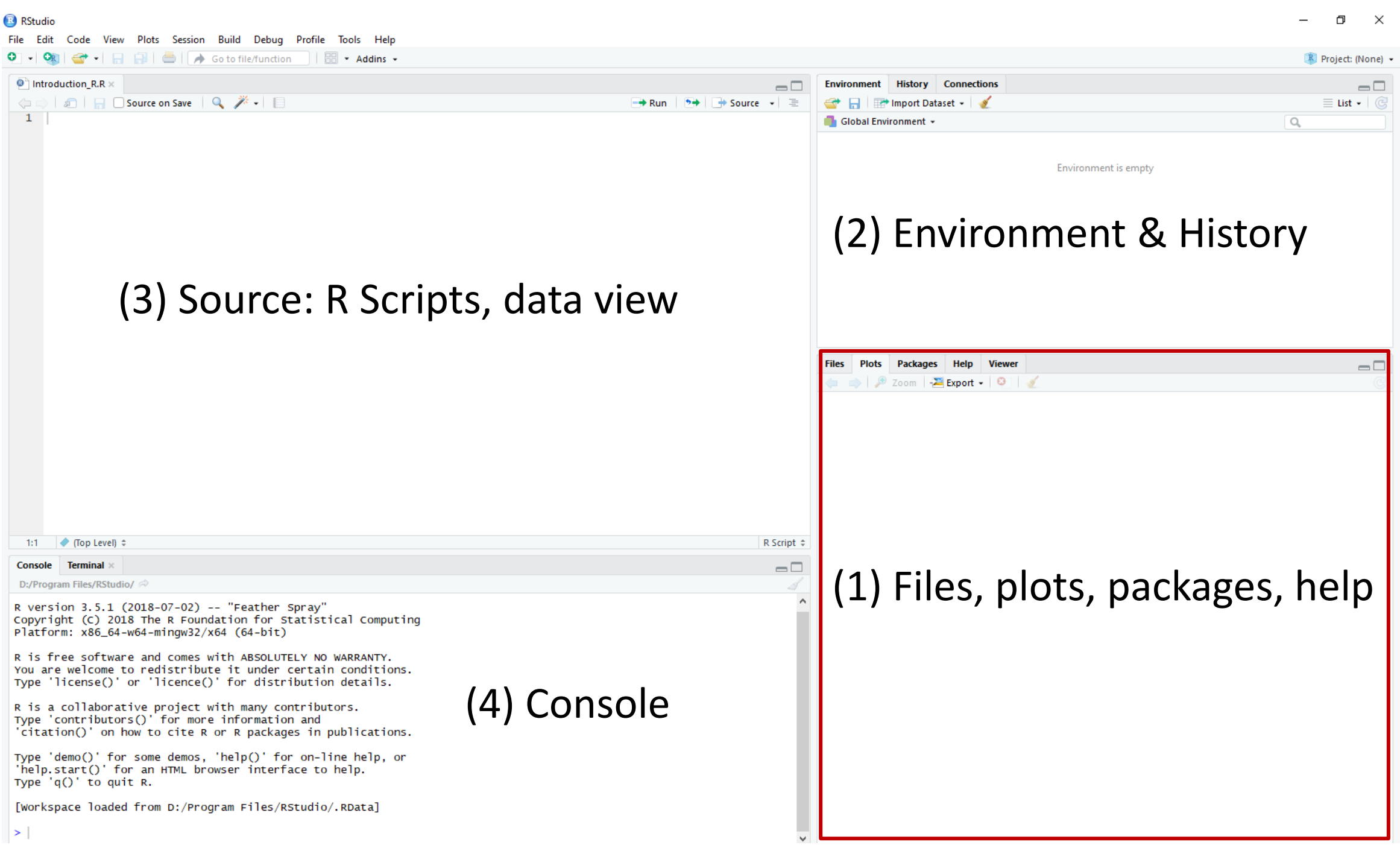
First steps with R Studio

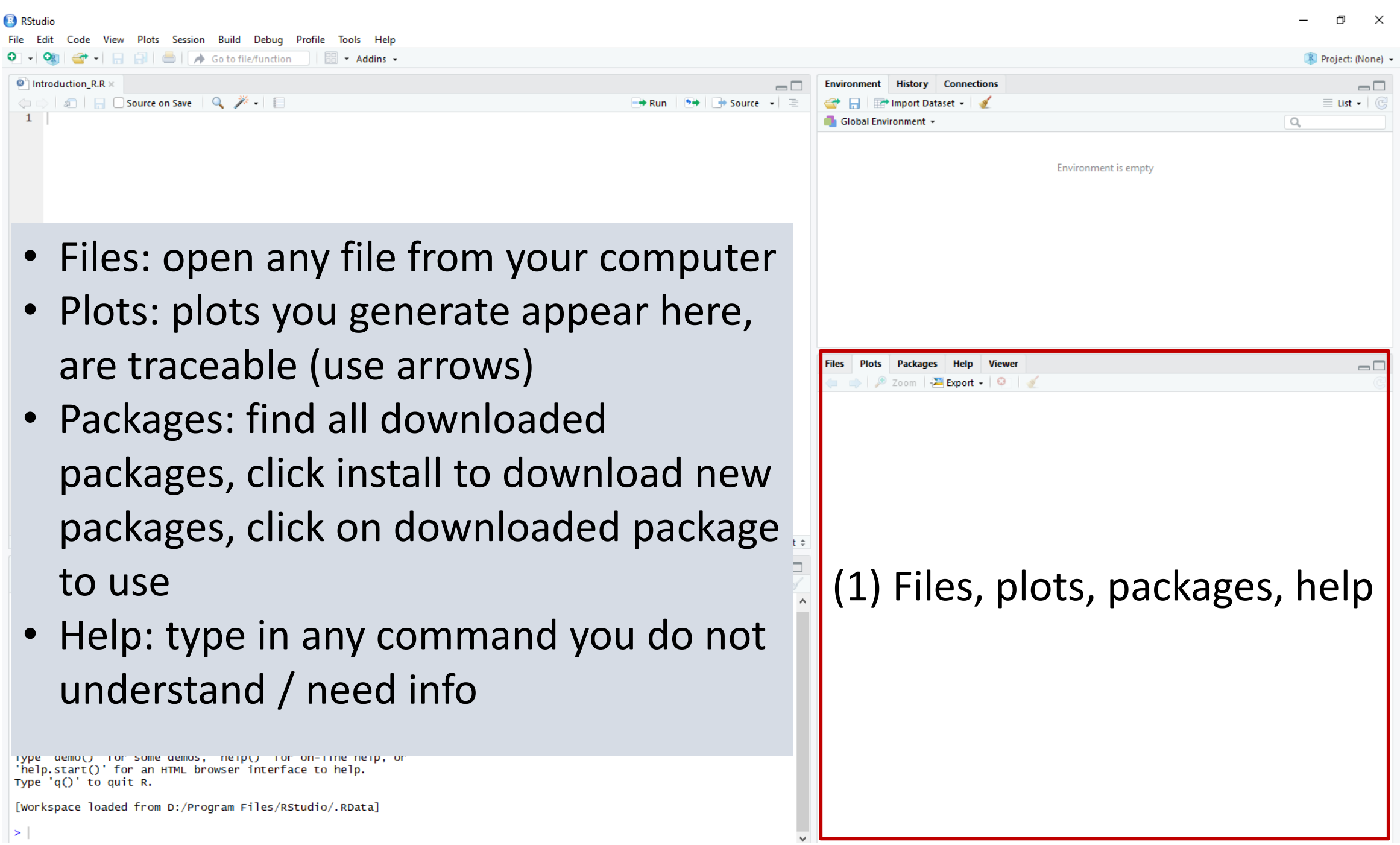


First steps with R Studio

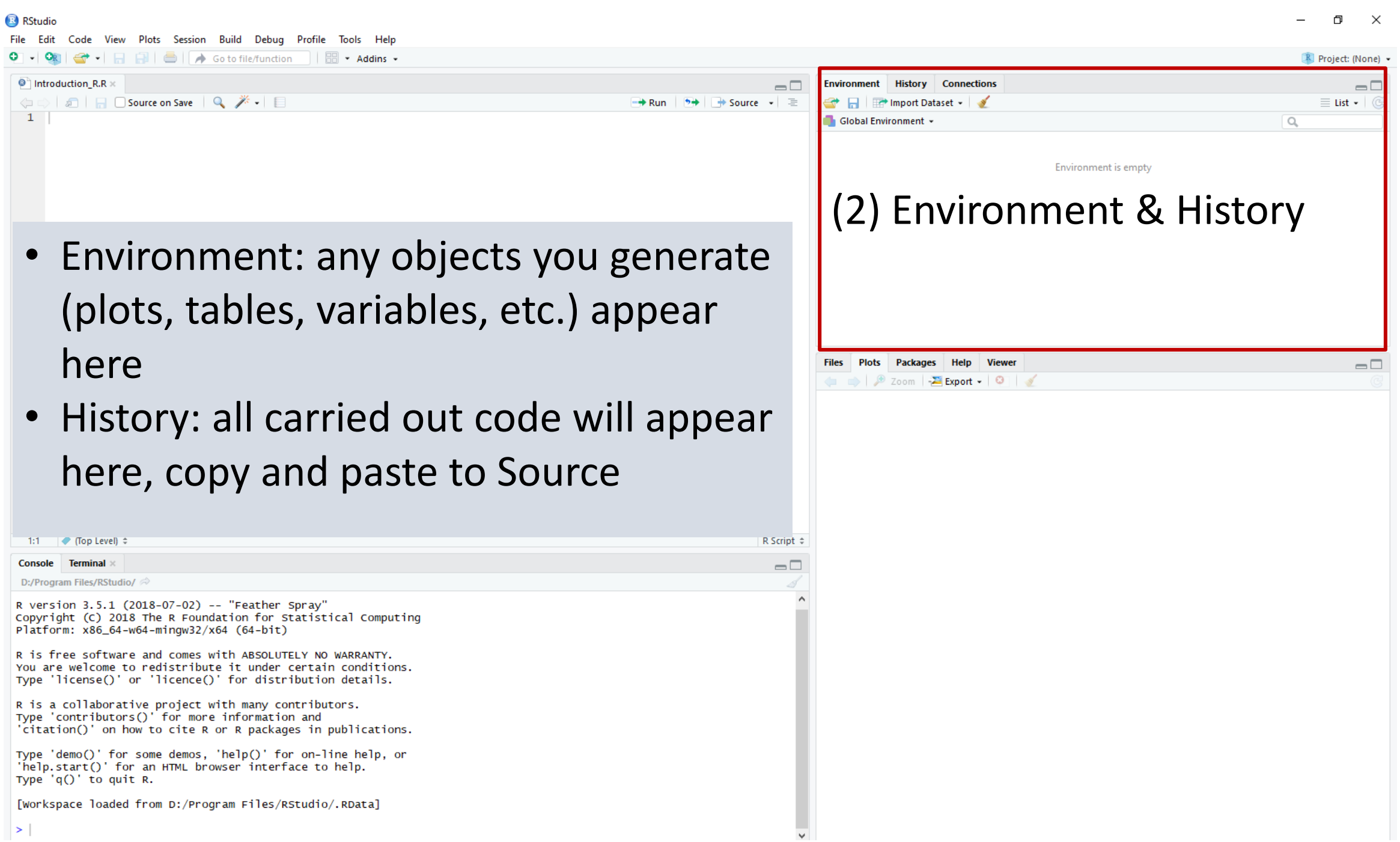
What do you see? 4 windows:

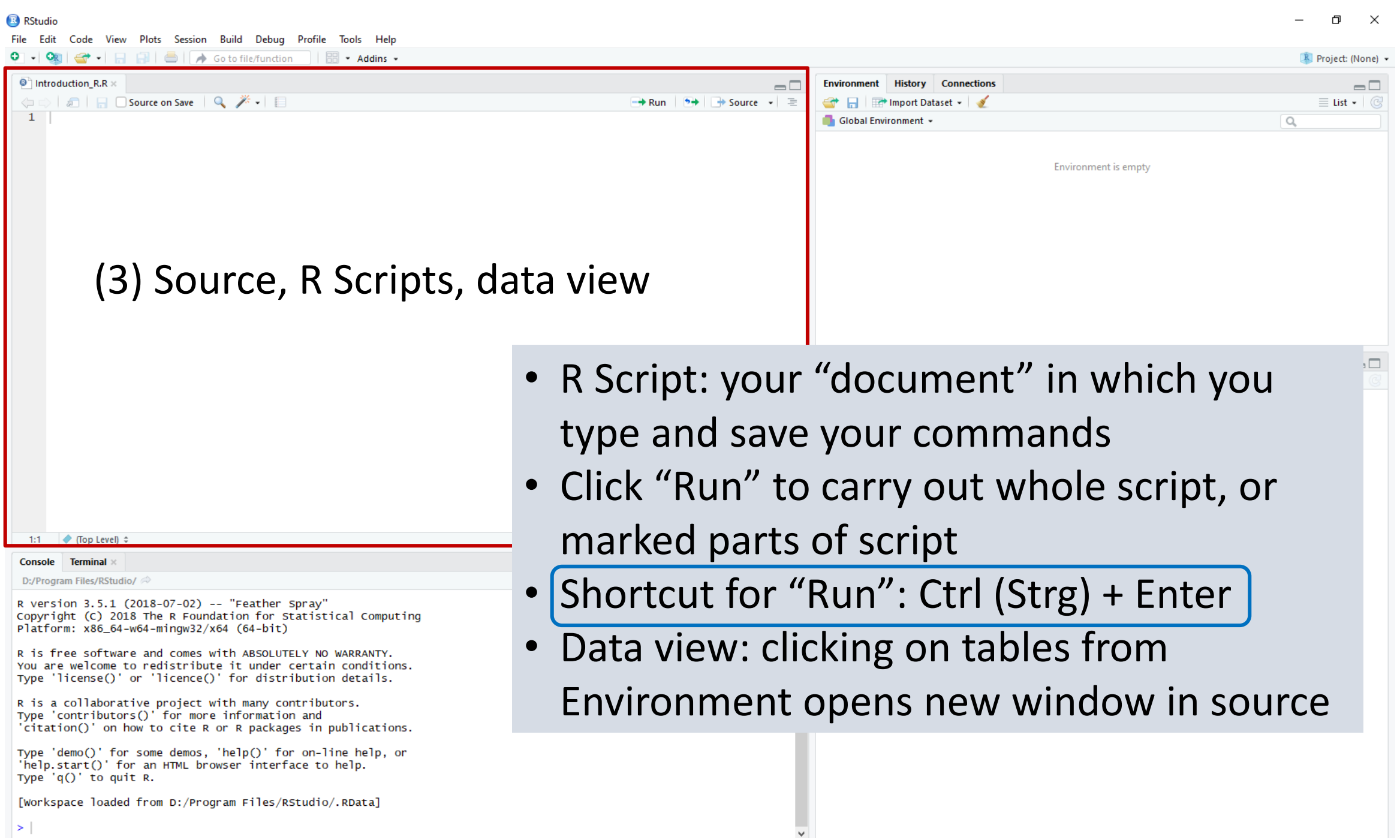

RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Project: (None)
Introduction_R.R
Source on Save
Run
Source
1
1:1
(Top Level)
R Script
Environment
History
Connections
Import Dataset
List
Global Environment
Environment is empty
(2) Environment & History
Files
Plots
Packages
Help
Viewer
Zoom
Export
(3) Source: R Scripts, data view
(1) Files, plots, packages, help
Console
Terminal
D:/Program Files/RStudio/
R version 3.5.1 (2018-07-02) -- "Feather Spray"
Copyright (C) 2018 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)
R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.
R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.
Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.
[Workspace loaded from D:/Program Files/RStudio/.RData]
>
(4) Console

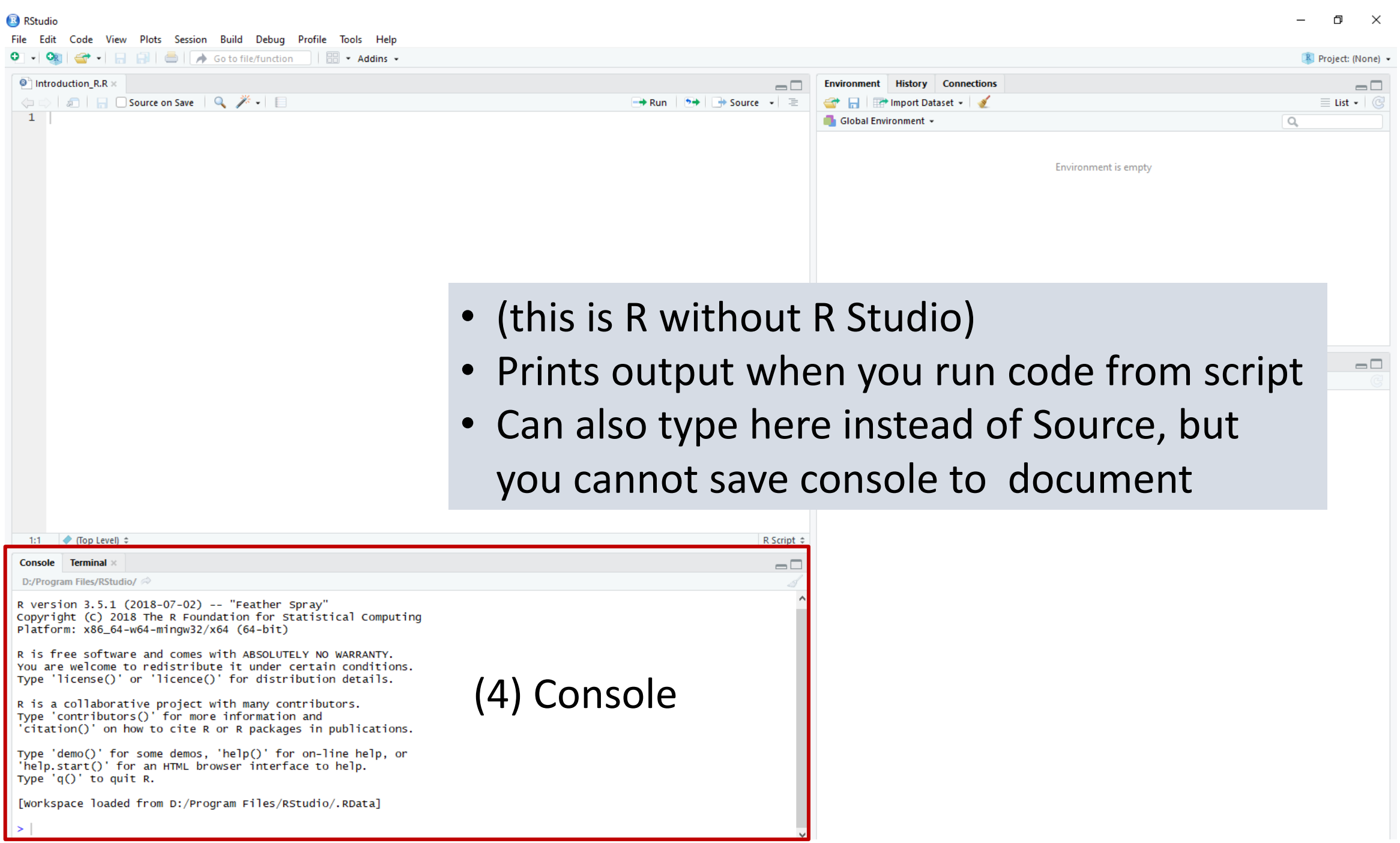

RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Project: (None)
Introduction_R.R
Source on Save
Run
Source
1
1:1
(Top Level)
R Script
Environment
History
Connections
Import Dataset
List
Global Environment
Environment is empty
(2) Environment & History
Files
Plots
Packages
Help
Viewer
Zoom
Export
(1) Files, plots, packages, help
(3) Source: R Scripts, data view
Console
Terminal
D:/Program Files/RStudio/
R version 3.5.1 (2018-07-02) -- "Feather Spray"
Copyright (C) 2018 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)
R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.
R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.
Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.
[Workspace loaded from D:/Program Files/RStudio/.RData]
>
(4) Console



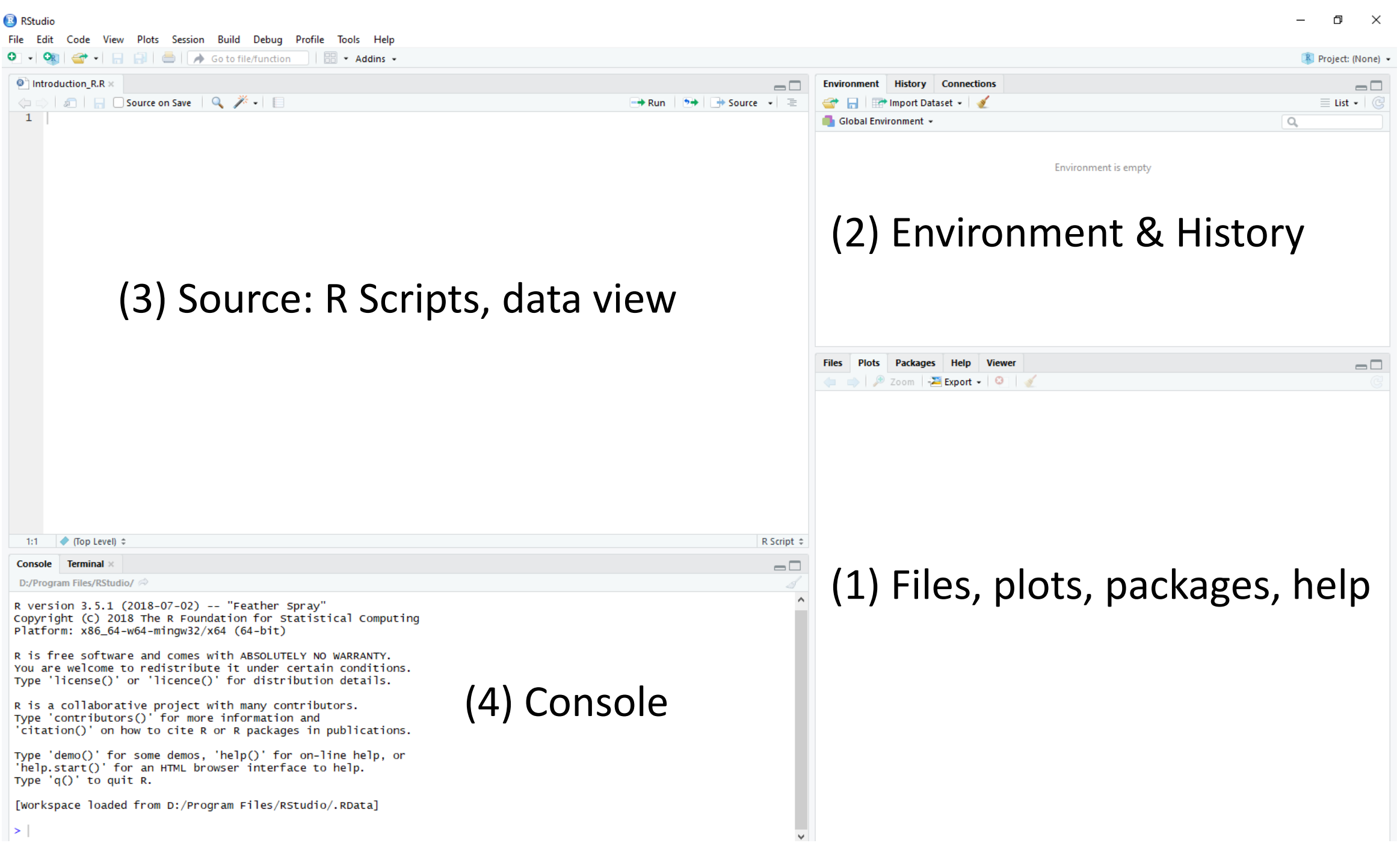
- Files: open any file from your computer
- Plots: plots you generate appear here, are traceable (use arrows)
- Packages: find all downloaded packages, click install to download new packages, click on downloaded package to use
- Help: type in any command you do not understand / need info


RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Project: (None)
Introduction_R.R
Source on Save
Run
Source
1
Environment History Connections
Import Dataset
List
Global Environment
Environment is empty
(2) Environment & History
Files Plots Packages Help Viewer
Zoom
Export
1:1
(Top Level)
R Script
Console Terminal
D:/Program Files/RStudio/
R version 3.5.1 (2018-07-02) -- "Feather Spray"
Copyright (C) 2018 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)
R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.
R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.
Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.
[Workspace loaded from D:/Program Files/RStudio/.RData]
>
- Environment: any objects you generate (plots, tables, variables, etc.) appear here
- History: all carried out code will appear here, copy and paste to Source


RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Project: (None)
Introduction_R.R
Source on Save
Run
Source
1
(3) Source, R Scripts, data view
1:1
(Top Level)
Environment
History
Connections
Import Dataset
List
Global Environment
Environment is empty
Console
Terminal
D:/Program Files/RStudio/
R version 3.5.1 (2018-07-02) -- "Feather Spray"
Copyright (C) 2018 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)
R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.
R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.
Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.
[Workspace loaded from D:/Program Files/RStudio/.RData]
>
• R Script: your “document” in which you type and save your commands
• Click “Run” to carry out whole script, or marked parts of script
• Shortcut for “Run”: Ctrl (Strg) + Enter
• Data view: clicking on tables from Environment opens new window in source



- (this is R without R Studio)
- Prints output when you run code from script
- Can also type here instead of Source, but you cannot save console to document


RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Project: (None)
Introduction_R.R
Source on Save
Run
Source
1
1:1
(Top Level)
R Script
Environment
History
Connections
Import Dataset
List
Global Environment
Environment is empty
(2) Environment & History
Files
Plots
Packages
Help
Viewer
Zoom
Export
(3) Source: R Scripts, data view
(1) Files, plots, packages, help
Console
Terminal
D:/Program Files/RStudio/
R version 3.5.1 (2018-07-02) -- "Feather Spray"
Copyright (C) 2018 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)
R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.
R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.
Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.
[Workspace loaded from D:/Program Files/RStudio/.RData]
>
(4) Console

Your R Script

- Use commands to communicate what you want
- Commands in R have two parts: objects and functions

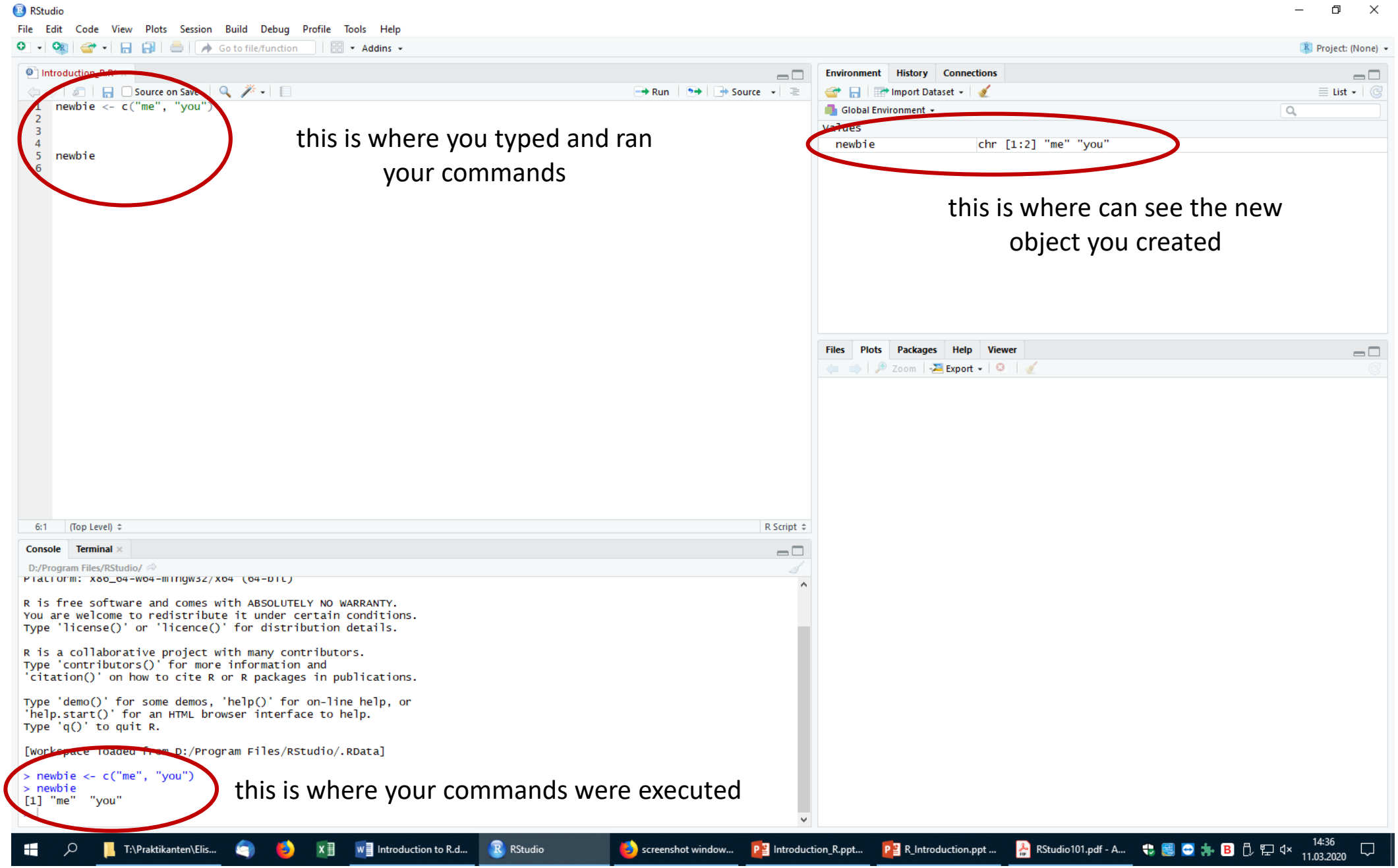
```
object <- function()
```

- Example in R:

```
newbie <- c("me", "you")
```

  - Function `c` creates new object `newbie` with elements `me` and `you`
  - In particular, the function `c` creates vector objects
  - The object now appears in your Environment

- Use # for comments (R will ignore what comes after #)


RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Addins
Project: (None)
Source on Save
Run
Source
1 newbie <- c("me", "you")
5 newbie
this is where you typed and ran your commands
Environment History Connections
Import Dataset
List
Global Environment
Values
newbie chr [1:2] "me" "you"
this is where can see the new object you created
Files Plots Packages Help Viewer
Zoom
Export
6:1 (Top Level)
R Script
Console Terminal
D:/Program Files/RStudio/
R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.
R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.
Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.
[Workspace loaded from D:/Program Files/RStudio/.RData]
> newbie <- c("me", "you")
> newbie
[1] "me" "you"
this is where your commands were executed
T:\Praktikanten\Elis...
Introduction to R.d...
RStudio
screenshot window...
Introduction_R.ppt...
R_Introduction.ppt ...
RStudio101.pdf - A...
14:36
11.03.2020

Functions and Objects

- Functions:
  - They “do” something
  - Always have a default setting
  - Change default by specifying

    Example: **`factor`**`(variable)` turns numeric variable into factor.

    Specify to create a more detailed factor variable

```
factor(variable,
            levels = c(0, 1),
            labels = c(“male”, “female”)
```

Functions and Objects

- Objects:
  - Character variables: always in quotes, stored as `chr` (character)
  - Numeric variables: stored as `int` (integer)
  - Factor variables (categorical): numeric variable, numbers represent names.

    Example: gender coded as 0 = “male” and 1 = “female”.

    In R: `Factor (w/ 2 levels: “male”, “female”)`

  - Check the type of your object with a function:

```
> mode(newbie)
[1] "character"
```

Vectors, Matrices and Data Frames

- Vectors:
  - Hold only one kind of data (numeric or character)

    Example: `newbie <- c("me", "you")` (character vector)

    `ages <- (25, 36)` (numeric vector)

- Matrices:
  - Hold only one kind of data, but contain multiple "rows" of vectors

- Data frames

Vectors, Matrices and Data Frames

- Data frames:
  - Can contain different kind of vectors, but have same column length
  - In our context used like tables and for importing excel files

    Example: create data frame `persons` with variables `newbie` and `ages`

```
newbie <- c("me", "you")
ages <- c(25, 36)

persons <- data.frame(newbie, ages)
```

Vectors, Matrices and Data Frames

- Data frames:
  - Can contain different kind of vectors, but have same column length
  - In our context used like tables and for importing excel files

    Example: create data frame `persons` with variables `newbie` and `ages`

```
persons <- data.frame(newbie, ages)
```

    this is your object (data frame) `persons`

```
newbie    ages
me        25
you       36
```

Vectors, Matrices and Data Frames

- Matrices vs. data frames:
  - Mostly use data frames (can contain different kinds of variables)
  - Some functions only work with matrices, but conversion is easy:
  - Matrix → data frame:

```
data.frame(yourmatrix)
dataframe <- data.frame(yourmatrix)
```

  - Data frame → matrix (only if numeric or characters!)

```
as.matrix(yourdata)
matrix <- as.matrix(yourdata)
```

Accessing Data Frames

- Always with `table[rows, columns]`
- Selecting areas:

`tab[ , 1:3]`	select columns 1 – 3
`tab[1:20, 1:3]`	select rows 1 – 20, columns 1 – 3
`tab[c(1, 5), ]`	select only rows 1 and 5

- Selecting by logic:

`tab[sex == "male", ]`	select rows where variable sex is male
`tab[age > 15, ]`	select rows where age is over 15

Accessing Data Frames

- Always with `table[rows, columns]`
- Selecting variables:

`tab$names`	select column “names”
`tab[,c(“names”, “sex”)]`	select columns “names” and “sex”

Importing Data into R

- Usually don't type in data manually
- Importing excel files:
  - Use package `library(readxl)`
  - Use `read_excel()` to import excel file

How to work with imported excel files: next lecture!

- Working Directory
  - Files in this folder will appear in "Files" (bottom right) → easy access
  - Set your working directory: `setwd("file path")`
  - See file path of your working directory: `getwd()`

Importing Data into R

- Exporting excel files:
  - Use package `library(xlsx)` (enough to open once!)
  - Use write.table(table, “table1.xlsx”) to save table in working directory

Important commands

COMMAND	EFFECT
library(packagename)	opens package from library (must be already installed)
install.package("packagename")	installs package into your library
setwd("C:/Path_to_your_WD")	sets your working directory (you can access files directly now)
getwd()	will display the path to your working directory
help("function") or **?**function	will display the manual page of given function
#	used for commenting as R will ignore anything after hash
c(…)	Combining values or strings to a vector (if using strings, put values in parentheses)
factor(variable, **levels** = c(1, 2), **labels** = c("male", "female")	Turns numeric variable into a factor. Level 1 = male, level 2 = female.
as.numeric()	Turns character variable into numeric variable
data.frame(var1, var2)	Combines two variables into one data frame
as.matrix()	Turns data frame into matrix
**$** (Example: table$column_1)	Used to select a particular column in a table
dataframe[rows, columns]	To specify which rows and columns will be used