INTRODUCTION TO R AND RSTUDIO

Part 1: Setup (Follow along in RStudio)

LEARNING OUTCOMES

What you will learn in this session:

- How to install R and RStudio
- What is the windows layout of RStudio
- How to setup RStudio
- How to create a project (folder) in RStudio
- How to use major functionalities of RStudio
- How to extend R’s functionality with R-packages
- Which packages you should install for this book

INSTALL AND SETUP R AND RSTUDIO

A typical setup to work with R consists of two components:

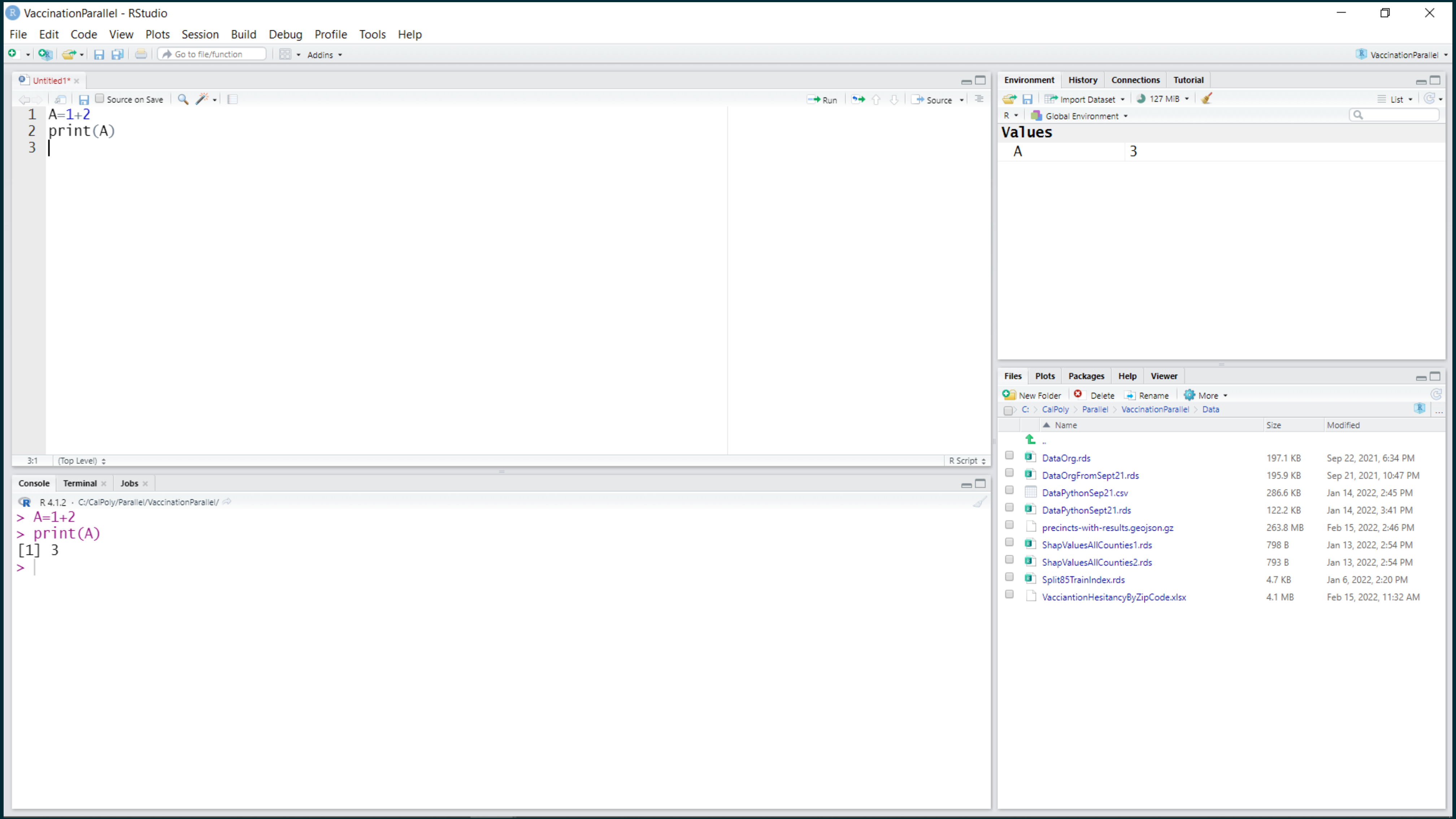
- the **R Console** which executes R code and
- an integrated development environment (IDE) such as **RStudio**.

You can download R here: Download R

You can download RStudio here: Download RStudio

Detailed installation guides are provided in the Book and the Online Resources sections of this chapter in book.

RSTUDIO – INTEGRATED DEVELOPMENT ENVIRONMENT (IDE) FOR R



```
A=1+2
print(A)
```

```
R 4.1.2 · C:/CalPoly/Parallel/VaccinationParallel/
> A=1+2
> print(A)
[1] 3
>
```

Values

A	3

C: › CalPoly › Parallel › VaccinationParallel › Data

Name	Size	Modified
..		
DataOrg.rds	197.1 KB	Sep 22, 2021, 6:34 PM
DataOrgFromSept21.rds	195.9 KB	Sep 21, 2021, 10:47 PM
DataPythonSep21.csv	286.6 KB	Jan 14, 2022, 2:45 PM
DataPythonSept21.rds	122.2 KB	Jan 14, 2022, 3:41 PM
precincts-with-results.geojson.gz	263.8 MB	Feb 15, 2022, 2:46 PM
ShapValuesAllCounties1.rds	798 B	Jan 13, 2022, 2:54 PM
ShapValuesAllCounties2.rds	793 B	Jan 13, 2022, 2:54 PM
Split85TrainIndex.rds	4.7 KB	Jan 6, 2022, 2:20 PM
VacciantionHesitancyByZipCode.xlsx	4.1 MB	Feb 15, 2022, 11:32 AM

RStudio Window

https://econ.lange-analytics.com/aibook/

RECOMMENDED RSTUDIO SETTINGS

1. **Do not Restore .RData into workspace at startup:** `Tools -> GlobalOptions`.

2. **Work with R Projects:** This assigns a directory on your hard drive to your R analysis: `File -> New Project`

R PACKAGES

R Packages extend R's functionality. They have to be **installed** only once:

`Tools -> Install Packages ...`

After installation they need to be **loaded** in every new R script with `library()`.

Packages frequently used in this course (**please install soon**):

- `tidyverse`: supports easy data processing .
- `rio`: allows loading various data resources with one `import()` command from the user's hard drive or the Internet.
- `janitor`: provides functionality to clean data and rename variable names to avoid spaces and special characters.
- `tidymodels`: streamlines data engineering and machine learning tasks.
- `kableExtra`: supports rendering tables in HTML.
- `shiny`: needed together with the `learnr` package for the interactive exercises in the book.
- `learnr` package: together with the `shiny` package for the interactive exercises in the book.

EXAMPLE: THE `rio` AND THE `tidyverse` PACKAGE

Assuming the `rio` packages is already installed.

```r
library(rio);library(tidyverse)
DataHousing =
  import("https://lange-analytics.com/AIBook/Data/HousingData.csv") %>%
  select(Price=price, Sqft=sqft_living, Bedrooms=bedrooms,Waterfront=waterfront)
print(DataHousing[1:3,])
```

```
   Price Sqft Bedrooms Waterfront
1 221900 1180        3         no
2 538000 2570        3         no
3 180000  770        2         no
```

`import()` would not work if the `rio` package were not loaded.

`select()` would not work if the `tidyverse` package were not loaded.

DATA TYPES & DATA OBJECTS

- **Data Types:** What can R store?
  - numerical `num`
  - character `chr`
  - `factor`
  - `logic`
- **Data Objects:** What are the **containers** R uses to store data?
  - single entry *single entry variable`
  - list of entries `vectors`
  - table `dataframe` and `tibble`
  - *advanced objects*. E.g., for plot, models, prediction results

DATA TYPES

Main | Numerical | Character | Factor | Logic | Truth Table

Numerical Data Type (`num`): Numerical values (e.g., 1, 523, 3.45) are used for calculation. In contrast, ZIP-Codes are not numerical data type.

Character Data Type (`chr`): Storing sequence of characters, numbers, and/or symbols to form a word or even a sentence is called a `character` data type (e.g. first or last names, street addresses, or Zip-codes)

Factor Data Type (`factor`): A `factor` is an R data type that stores *categorical* data in an effective way. `factor` data types are also required by many classification models in R.

Logic Data Type(`logic`): A data type that stores the logic states `TRUE` and `FALSE` is called a `logic` object (sometimes called Boolean)

DATA TYPES & DATA OBJECTS

Data Types: What can R store?

Data Objects: What are the containers R uses to store data?

DATA TYPES

- **Single Value Object**
- **Vector Object**
- **Data Frame (Tibble) Object**
- **List Object** (not covered in this course)
- **Advanced Object** such as plots, models, recipes

SINGLE VALUE OBJECT

Object just stores a single value:

```
A=123.768
B=3
C="Hello World"
IsLifeGood=TRUE
```

VECTOR-OBJECTS

A vector object stores a list of values (numerical, character, factor, or logic)

Example: Weather during the last three days in Stattown:

```r
VecTemp=c(70, 68, 55)
VecWindSpeed=c("low","low","high")
VecIsSunny=c(TRUE,TRUE,FALSE)
```

Vector objects can be used as arguments for an R command to calculate:

▶ Code

```
The average forecasted temperature is 64.33333
```

▶ Code

```
The forecast is for 3 days.
```

DATA FRAMES (TIBBLES)

A data frame is similar to an Excel table (note not all columns of the Titanic data frame are shown).

Survived	Pclass	Sex	Age	FareInPounds
0	3	male	22	7.2500
1	1	female	38	71.2833
1	3	female	26	7.9250
1	1	female	35	53.1000
0	3	male	35	8.0500
0	3	male	27	8.4583
0	1	male	54	51.8625
0	3	male	2	21.0750

Survived	Pclass	Sex	Age	FareInPounds
1	3	female	27	11.1333
1	2	female	14	30.0708
1	3	female	4	16.7000
1	1	female	58	26.5500

A data frame consist of vectors making up the columns. These are the variables for the data analysis (remember: observations are in the rows, variables are in the columns).

```r
DataTitanic=import("https://lange-analytics.com/AIBook/Data/TitanicDataCl.csv")
str(DataTitanic)
```

```
'data.frame':	887 obs. of  8 variables:
 $ Survived             : int  0 1 1 1 0 0 0 0 1 1 ...
 $ Pclass               : int  3 1 3 1 3 3 1 3 3 2 ...
 $ Name                 : chr  "Mr. Owen Harris Braund" "Mrs. John Bradley (Florence
Briggs Thayer) Cumings" "Miss. Laina Heikkinen" "Mrs. Jacques Heath (Lily May Peel)
Futrelle" ...
 $ Sex                  : chr  "male" "female" "female" "female" ...
 $ Age                  : num  22 38 26 35 35 27 54 2 27 14 ...
 $ SiblingsSpousesAboard: int  1 1 0 1 0 0 0 3 0 1 ...
```

EXTRACTING THE VECTORS AND PERFORM CALCULATIONS (NUMERICAL VECTORS)

```r
VecFareInPounds=DataTitanic$FareInPounds
AvgFare=mean(VecFareInPounds)
cat("The average fare of Titanic passengers was:", AvgFare, "British Pounds")
```

```
The average fare of Titanic passengers was: 32.30542 British Pounds
```

EXTRACTING THE VECTORS AND PERFORM CALCULATIONS (LOGICAL VECTORS)

```r
DataTitanic$Survived=as.logical(DataTitanic$Survived)
str(DataTitanic)
```

```
'data.frame':	887 obs. of  8 variables:
 $ Survived               : logi  FALSE TRUE TRUE TRUE FALSE FALSE ...
 $ Pclass                 : int  3 1 3 1 3 3 1 3 3 2 ...
 $ Name                   : chr  "Mr. Owen Harris Braund" "Mrs. John Bradley (Florence
Briggs Thayer) Cumings" "Miss. Laina Heikkinen" "Mrs. Jacques Heath (Lily May Peel)
Futrelle" ...
 $ Sex                    : chr  "male" "female" "female" "female" ...
 $ Age                    : num  22 38 26 35 35 27 54 2 27 14 ...
 $ SiblingsSpousesAboard: int  1 1 0 1 0 0 0 3 0 1 ...
 $ ParentsChildrenAboard: int  0 0 0 0 0 0 0 1 2 0 ...
 $ FareInPounds           : num  7.25 71.28 7.92 53.1 8.05 ...
```

```r
SurvRate=mean(DataTitanic$Survived)
cat("The average survival rate of Titanic passengers was:", SurvRate)
```

```
The average survival rate of Titanic passengers was: 0.3855693
```

DATA FRAMES VS. TIBBLES

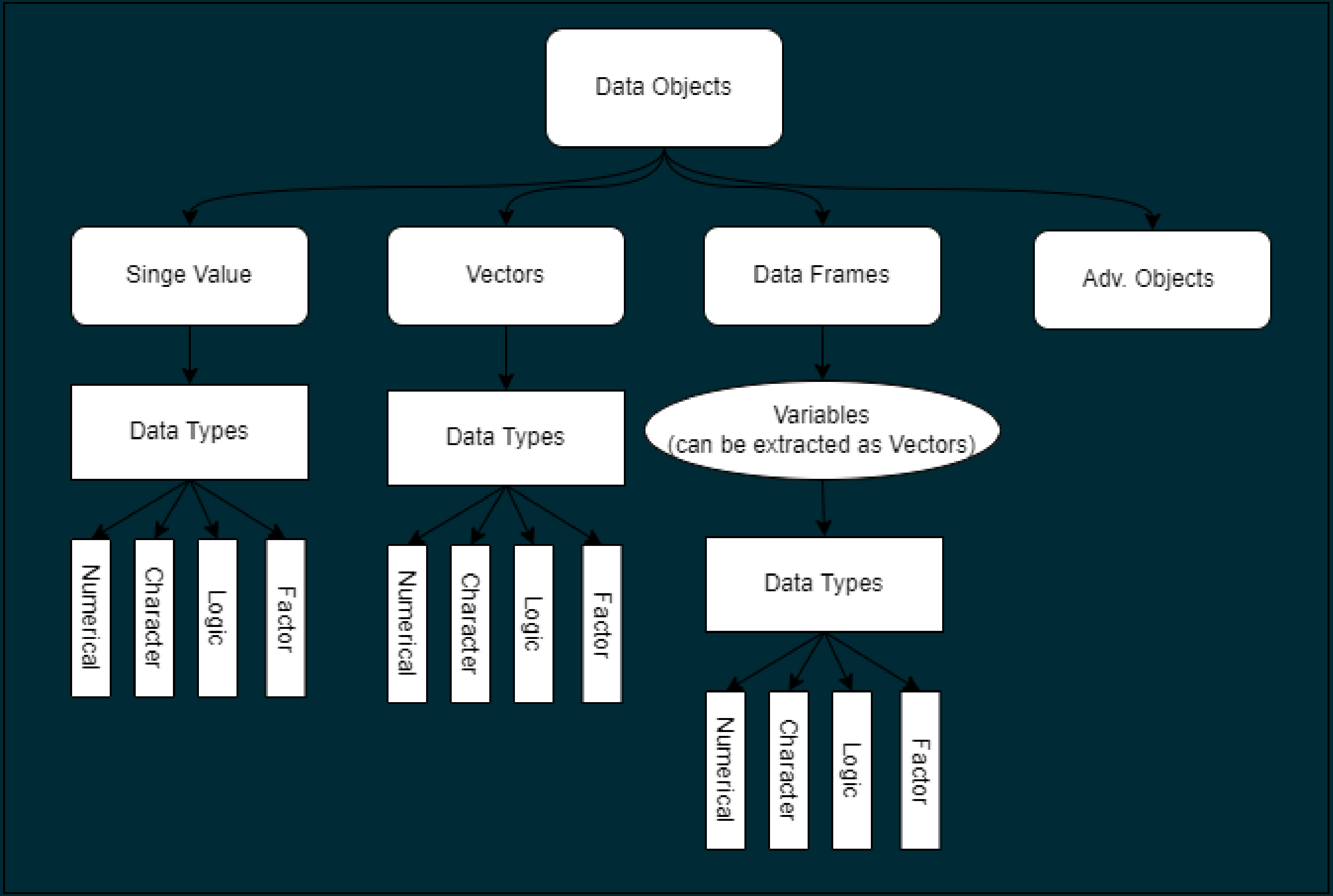
A **tibble** is a more advanced sub-type of a *data frame*. If needed, a regular *data frame* can be coerced into a *tibble* with the `as_tibble()` command.

A few of the differences between *data frames* and *tibbles*:

1. A *data frame* outputs all its rows and columns by default. A *tibble* outputs only the first 10 rows and the variables that fit on the screen but provides information about omitted variables and rows.
2. A *data frame* can have row names, while a *tibble* cannot.
3. In R version <4.1 a *data frame* converts all *character* values to *factor* type. This conversion was often confusing and annoying. In contrast, a *tibble* only coerces *character* values into *factor* on demand. Since R version 4.1 regular `data frames` behave the same as `tibbles`.

Among other reasons, points 1. and 3. make it more straightforward to work with *tibbles* rather than with basic *data frames*.

SUMMARY



https://econ.lange-analytics.com/aibook/