UiT The Arctic University of Norway

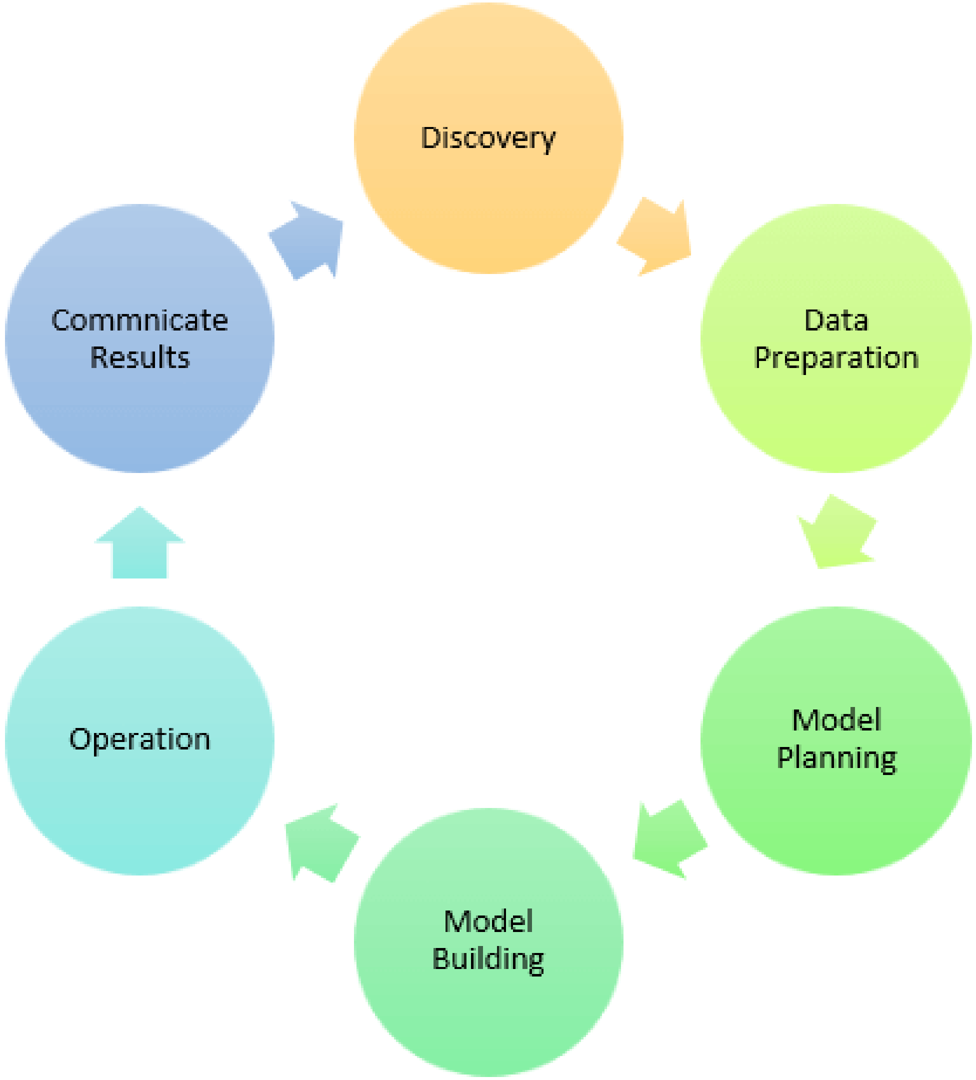
FSK-2053 Data science & bioinformatics for fisheries and aquaculture

Lecture 2 – Data Wrangling

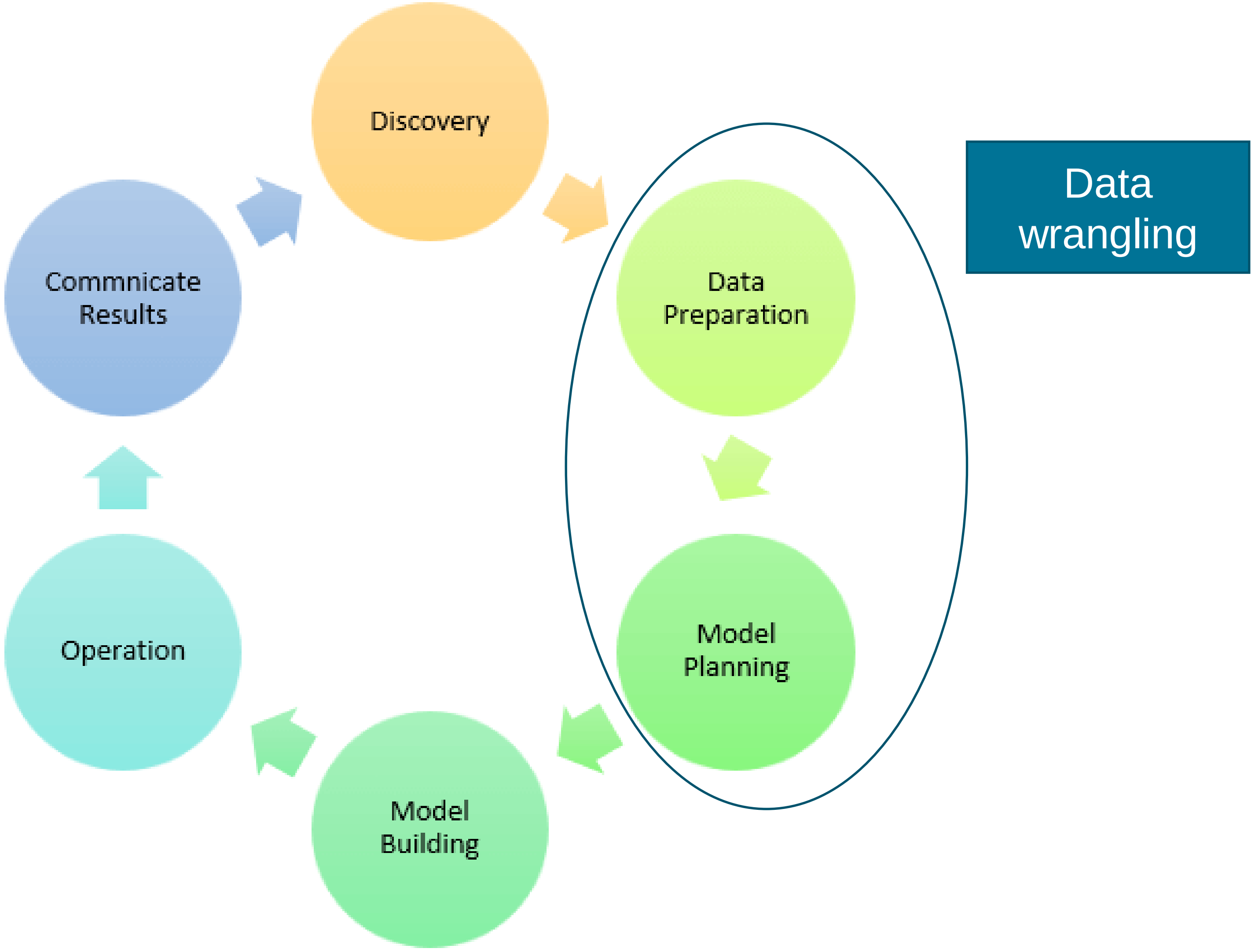
Daniel Kumazawa Morais
daniel.morais@uit.no

https://en.wikipedia.org/wiki/Wrangler_(profession)#/media/File:07350u_The_horse_wrangler.tif

The Data Science Workflow



The Data Science Workflow



1. Discovery:

Discovery step involves acquiring data from all the identified internal & external sources which helps you to shape and describe the problem.

The data can be:

- Retrieved from online public databases (using direct queries or automated APIs)

- Census and governmental databases

- Other… e.g. gathered from social media

Data Import : : CHEAT SHEET

R's **tidyverse** is built around **tidy data** stored in **tibbles**, which are enhanced data frames.

The front side of this sheet shows how to read text files into R with **readr**.

The reverse side shows how to create tibbles with **tibble** and to layout tidy data with **tidyr**.

OTHER TYPES OF DATA

Try one of the following packages to import other types of files

- **haven** - SPSS, Stata, and SAS files
- **readxl** - excel files (.xls and .xlsx)
- **DBI** - databases
- **jsonlite** - json
- **xml2** - XML
- **httr** - Web APIs
- **rvest** - HTML (Web Scraping)

Save Data

Save **x**, an R object, to **path**, a file path, as:

Comma delimited file

write_csv(x, path, na = "NA", append = FALSE, col_names = !append)

File with arbitrary delimiter

write_delim(x, path, delim = " ", na = "NA"

Read Tabular Data - These functions share the common arguments:

read_*(file, col_names = TRUE, col_types = NULL, locale = default_locale(), na = c("", "NA"), quoted_na = TRUE, comment = "", trim_ws = TRUE, skip = 0, n_max = Inf, guess_max = min(1000, n_max), progress = interactive())

a,b,c
1,2,3
4,5,NA
→

A	B	C
1	2	3
4	5	NA

Comma Delimited Files

read_csv("file.csv")

To make file.csv run:
write_file(x = "a,b,c\n1,2,3\n4,5,NA", path = "file.csv")

a;b;c
1;2;3
4;5;NA
→

A	B	C
1	2	3
4	5	NA

Semi-colon Delimited Files

read_csv2("file2.csv")

write_file(x = "a;b;c\n1;2;3\n4;5;NA", path = "file2.csv")

a|b|c
1|2|3
4|5|NA
→

A	B	C
1	2	3
4	5	NA

Files with Any Delimiter

read_delim("file.txt", delim = "|")

write_file(x = "a|b|c\n1|2|3\n4|5|NA", path = "file.txt")

a b c
1 2 3
4 5 NA
→

A	B	C
1	2	3
4	5	NA

Fixed Width Files

read_fwf("file.fwf", col_positions = c(1, 3, 5))

write_file(x = "a b c\n1 2 3\n4 5 NA", path = "file.fwf")

Tab Delimited Files

read_tsv("file.tsv") Also **read_table()**.

write_file(x = "a\tb\tc\n1\t2\t3\n4\t5\tNA", path = "file.tsv")

USEFUL ARGUMENTS

a,b,c
1,2,3
4,5,NA

Example file

write_file("a,b,c\n1,2,3\n4,5,NA","file.csv")
f <- "file.csv"

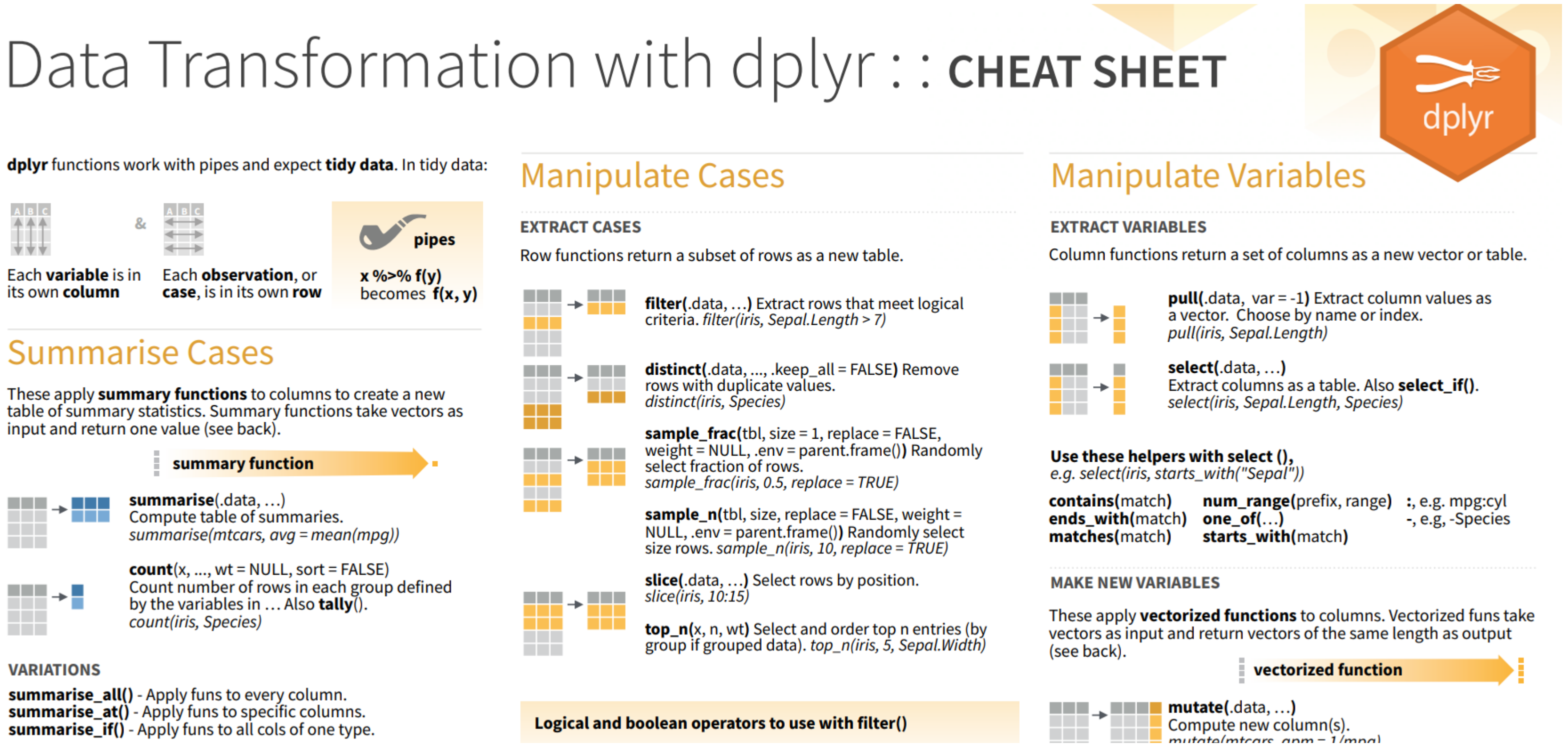
1	2	3
4	5	NA

Skip lines

read_csv(f, **skip = 1**)

2. Data preparation (data wrangling):

Data can have lots of inconsistencies like missing values, blank columns, incorrect data format which needs to be cleaned. You need to process, explore, filter, and condition data before modelling. The cleaner your data, the better are your predictions.



3. Model planning:

In this stage, you need to determine the method and technique to draw the relation between input variables. Planning for a model is performed by using different summarizing statistical tools and visualization tools.

4. Model building:

In this step, the actual model building process starts. Here, data scientist typically distributes datasets for training and testing. Techniques like association, classification, and clustering are applied to the training data set. The model once prepared is tested against the "testing" dataset.

3. Model planning:

In this stage, you need to determine the method and technique to draw the relation between input variables. Planning for a model is performed by using different summarizing statistical tools and visualization tools.

4. Model building:

In this step, the actual model building process starts. Here, data scientist typically distributes datasets for training and testing. Techniques like association, classification, and clustering are applied to the training data set. The model once prepared is tested against the "testing" dataset.

$$\bar{X} = \frac{\sum_{i=1}^{n} x_i}{n} \qquad \mu = \frac{\sum_{i=1}^{N} x_i}{N}$$

The simplest of the models

5. Operationalize:

In this stage, you deliver the final baselined model with reports, code, and technical documents. Model can be deployed into a real-time production environment after thorough testing.

6. Communicate results

In this stage, the key findings are communicated to all stakeholders. This helps you to decide if the results of the project are a success or a failure based on information from the model.

```
X Mean: 54.26
Y Mean: 47.83
X SD  : 16.76
Y SD  : 26.93
Corr. : -0.06
```

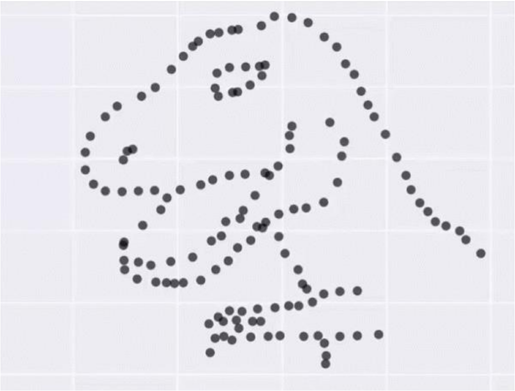
5. Operationalize:

In this stage, you deliver the final baselined model with reports, code, and technical documents. Model can be deployed into a real-time production environment after thorough testing.

6. Communicate results

In this stage, the key findings are communicated to all stakeholders. This helps you to decide if the results of the project are a success or a failure based on information from the model.

X Mean:	54.26
Y Mean:	47.83
X SD :	16.76
Y SD :	26.93
Corr. :	-0.06



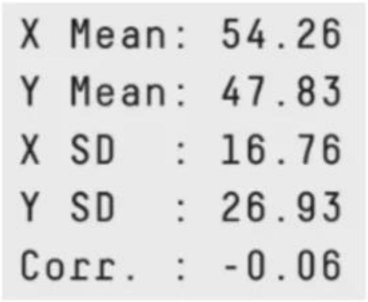
https://towardsdatascience.com/a-comprehensive-guide-to-the-grammar-of-graphics-for-effective-visualization-of-multi-dimensional-1f92b4ed4149

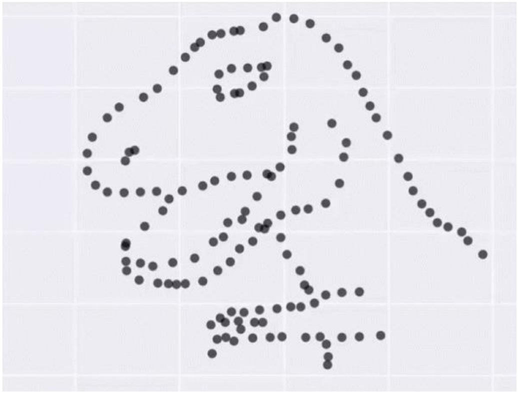
5. Operationalize:

In this stage, you deliver the final baselined model with reports, code, and technical documents. Model can be deployed into a real-time production environment after thorough testing.

6. Communicate results

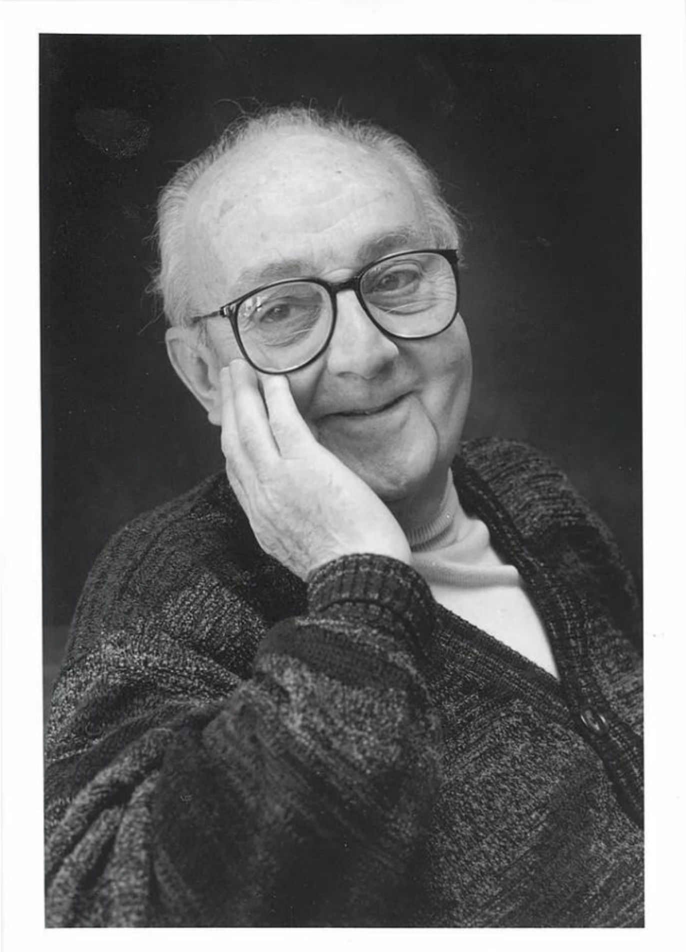
In this stage, the key findings are communicated to all stakeholders. This helps you to decide if the results of the project are a success or a failure based on information from the model.





https://towardsdatascience.com/a-comprehensive-guide-to-the-grammar-of-graphics-for-effective-visualization-of-multi-dimensional-1f92b4ed4149

"All models are wrong, but some are useful"

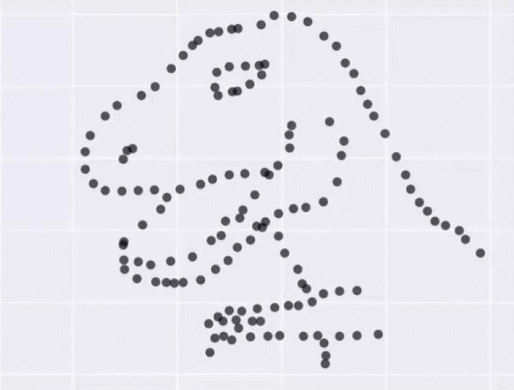


George E. P. Box.
British mathematician and professor of statistics

Simple models to describe the data

```
X Mean: 54.26
Y Mean: 47.83
X SD  : 16.76
Y SD  : 26.93
Corr. : -0.06
```

The actual data



How can we translate it to our fields of expertise?

https://medium.com/@chasecottle/all-models-are-wrong-but-some-are-useful-c97d8f169a8e

Communicate results
Critical step for evidence-base management

Sources of Data:
Online databases and resources for biology, fisheries & aquaculture

- **BarentsWatch**: https://www.barentswatch.no/
- **Global Fishing Watch**: https://globalfishingwatch.org/
- **FAO Global Fishery Databases**: http://www.fao.org/fishery/topic/16054
- **FishBase**: https://www.fishbase.se/
- **Global Biodiversity Information Facility**: https://www.gbif.org/
- **WORMS (World Register Of Marine Species)**: http://marinespecies.org/
- **Ocean Data Platform**: https://www.oceandata.earth/

- **NCBI/Genbank**: https://www.ncbi.nlm.nih.gov/
- **BOLD database**: https://boldsystems.org/

- **DRYAD**: https://datadryad.org/stash
- **Mendeley Data**: https://data.mendeley.com/
- **UiT Open Research Data**: https://dataverse.no/dataverse/uit

Sources of Data:
Online databases and resources for biology, fisheries & aquaculture

- **BarentsWatch**: https://www.barentswatch.no/

- **Global Fishing Watch**: https://globalfishingwatch.org/

- **FAO Global Fishery Databases**: https://www.fao.org/fishery/statistics-query/en/capture

- **FishBase**: https://www.fishbase.se/

- **Global Biodiversity Information Facility**: https://www.gbif.org/

- **WORMS (World Register Of Marine Species)**: http://marinespecies.org/

- **Ocean Data Platform**: https://www.oceandata.earth/

- **NCBI/Genbank**: https://www.ncbi.nlm.nih.gov/

- **BOLD database**: https://boldsystems.org/

- **DRYAD**: https://datadryad.org/stash

- **Mendeley Data**: https://data.mendeley.com/

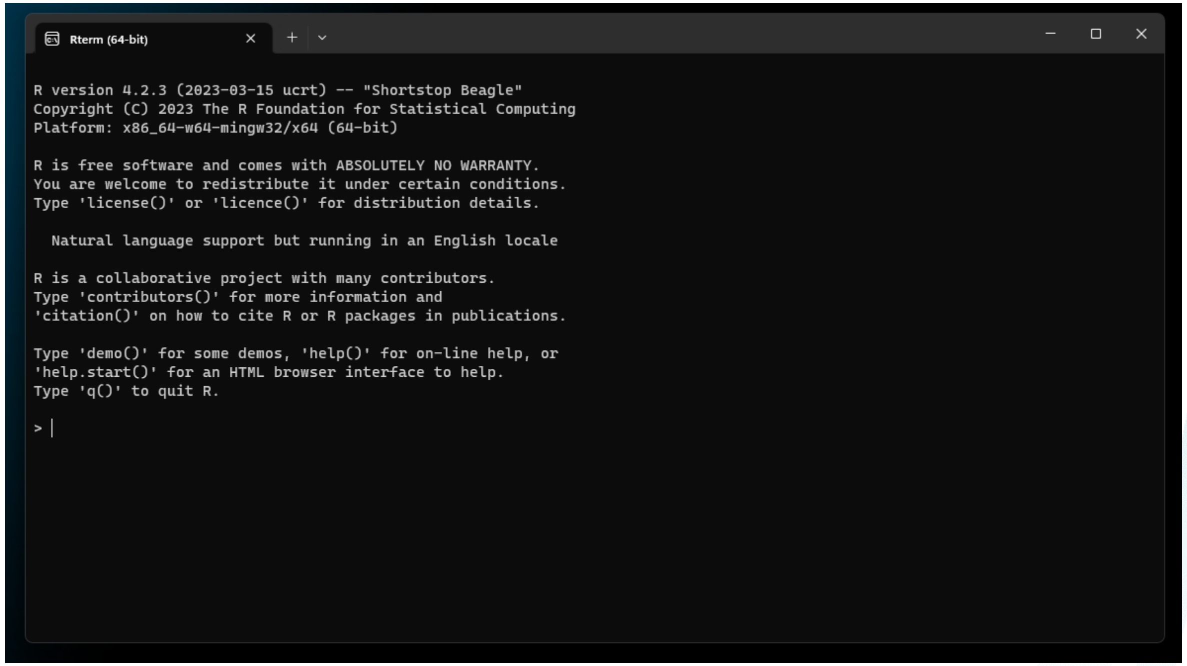
- **UiT Open Research Data**: https://dataverse.no/dataverse/uit

The R prompt

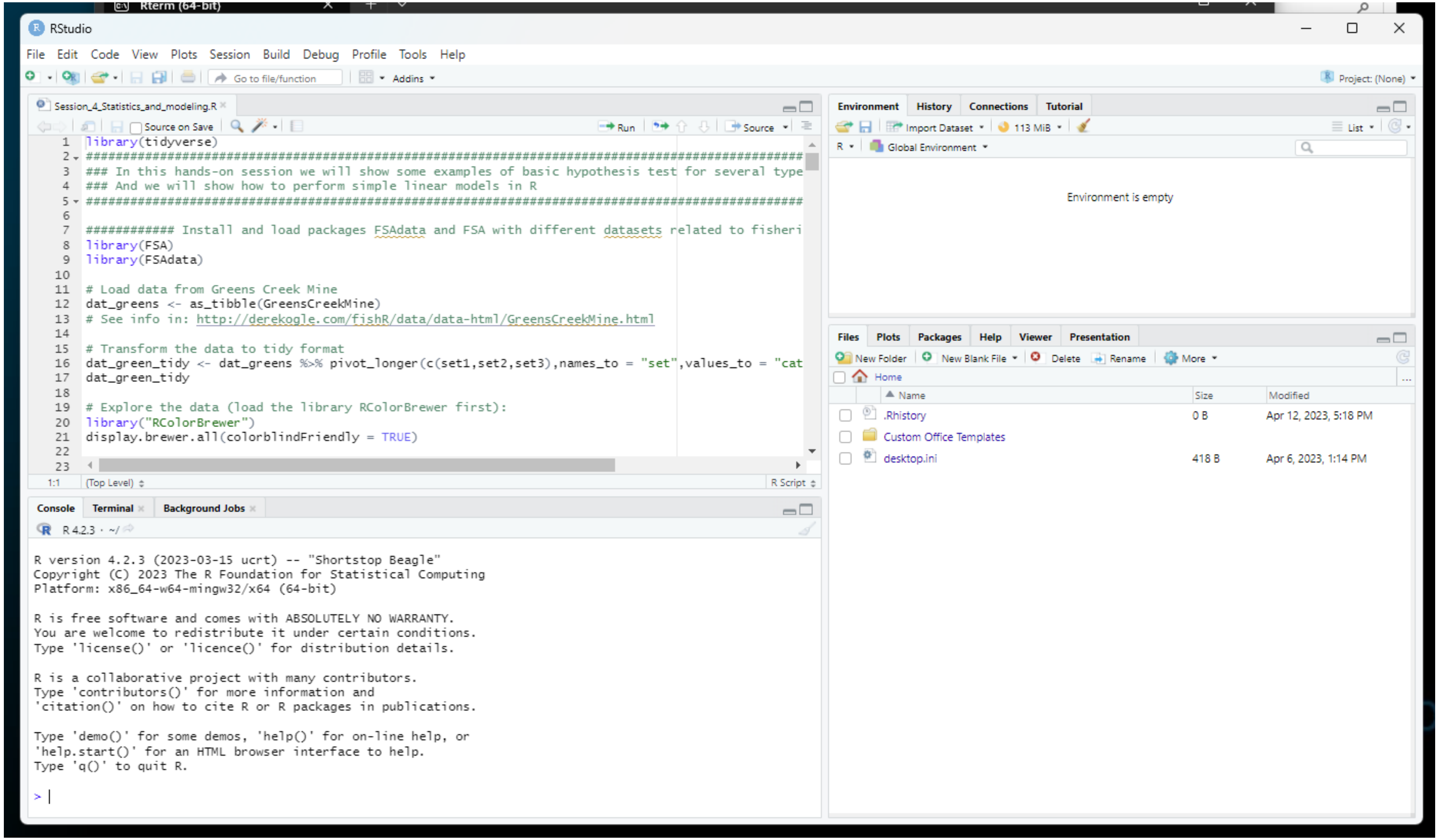
The R programming language was officially released in the early 2000.

It was created by Ross Ihaka and Robert Gentleman as a tool for teaching statistics classes.

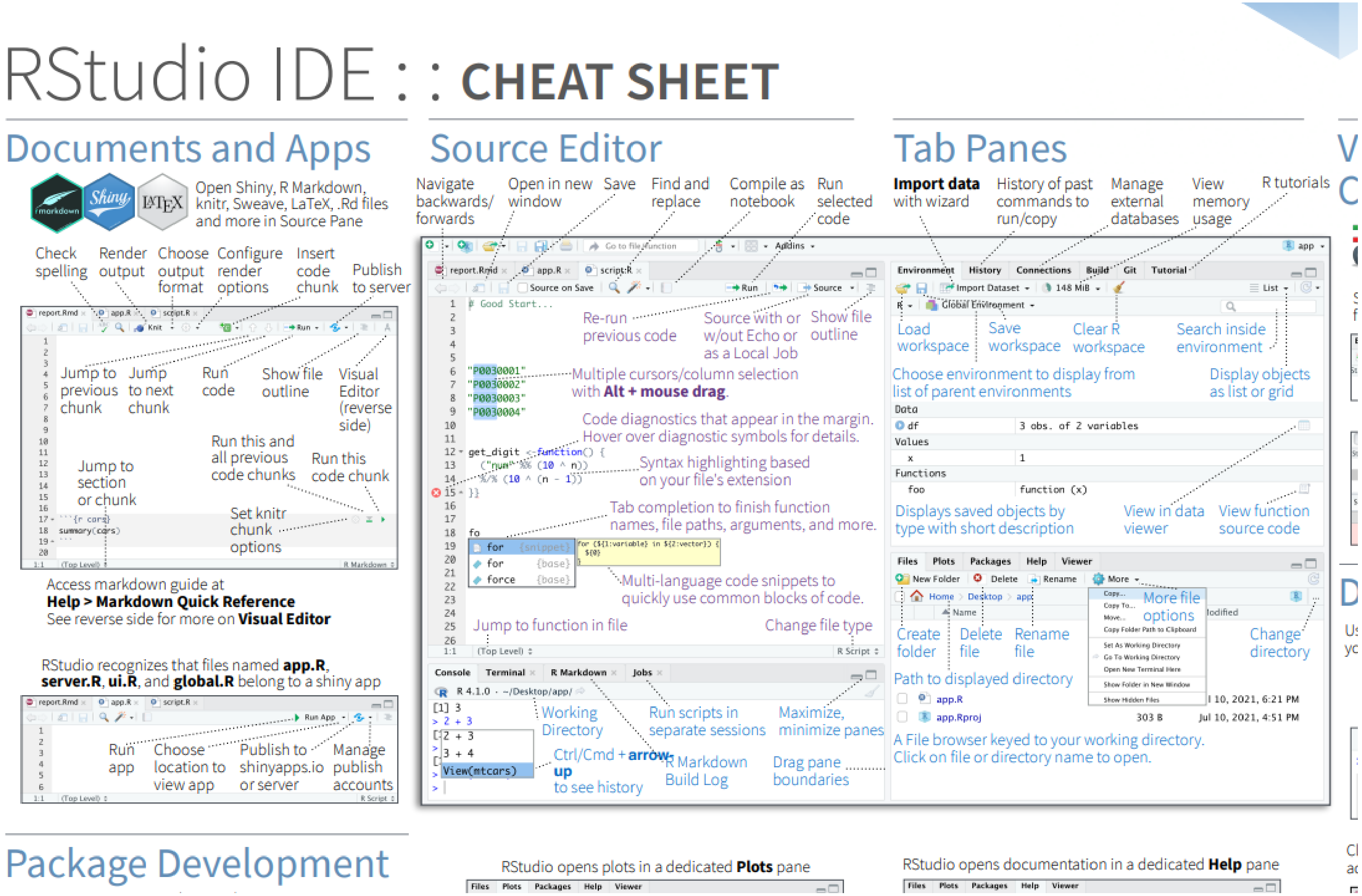
Therefore, its is considered that R doesn’t behave as most general-purpose programming languages. Being quite unique in many aspects.



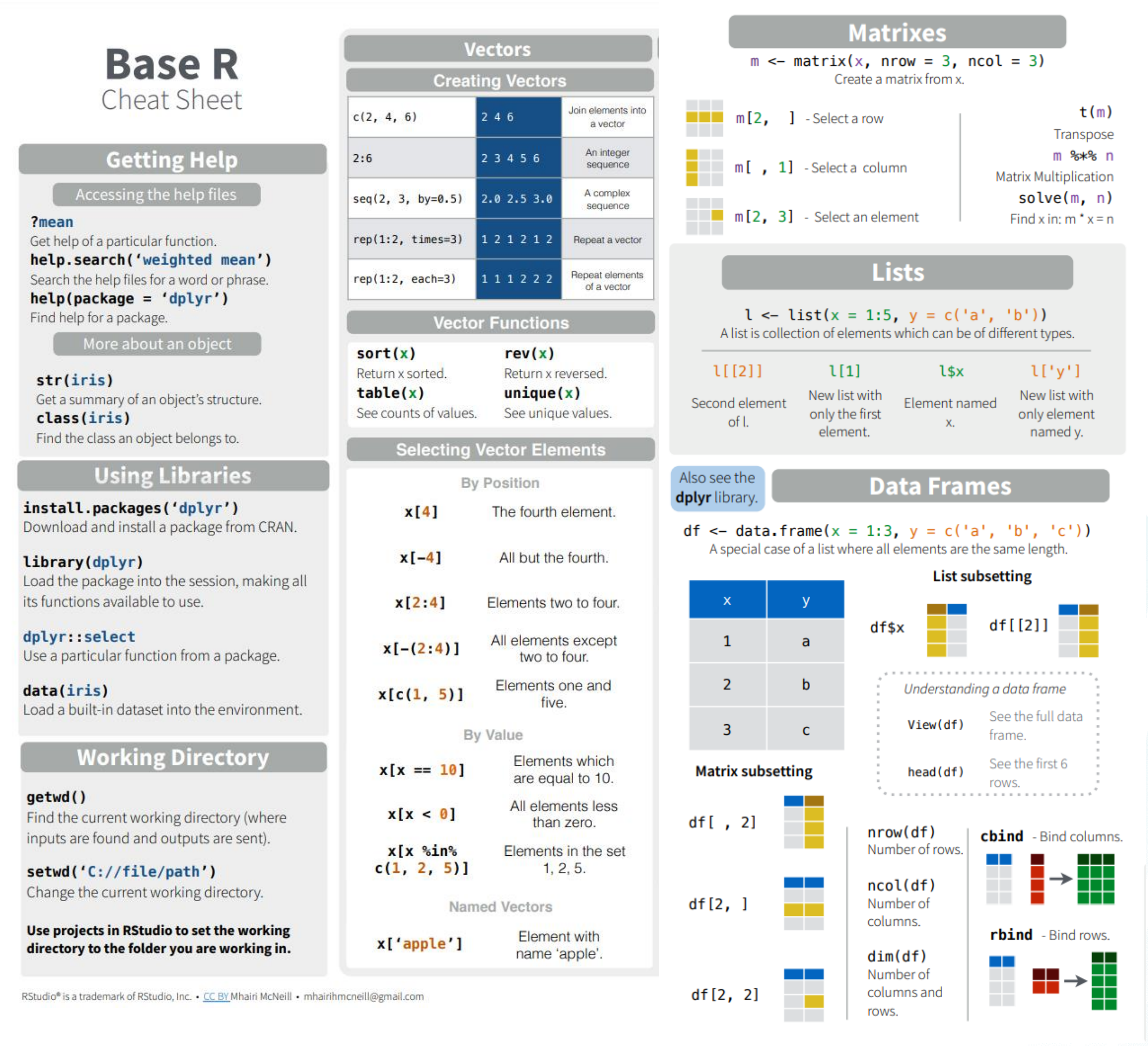
IDE - Integrated Development Environment



IDE - Integrated Development Environment



Data types in R



Base R
Cheat Sheet

Getting Help

Accessing the help files

`?mean`
Get help of a particular function.

`help.search('weighted mean')`
Search the help files for a word or phrase.

`help(package = 'dplyr')`
Find help for a package.

More about an object

`str(iris)`
Get a summary of an object's structure.

`class(iris)`
Find the class an object belongs to.

Using Libraries

`install.packages('dplyr')`
Download and install a package from CRAN.

`library(dplyr)`
Load the package into the session, making all its functions available to use.

`dplyr::select`
Use a particular function from a package.

`data(iris)`
Load a built-in dataset into the environment.

Working Directory

`getwd()`
Find the current working directory (where inputs are found and outputs are sent).

`setwd('C://file/path')`
Change the current working directory.

Use projects in RStudio to set the working directory to the folder you are working in.

Vectors

Creating Vectors

`c(2, 4, 6)`	2 4 6	Join elements into a vector
`2:6`	2 3 4 5 6	An integer sequence
`seq(2, 3, by=0.5)`	2.0 2.5 3.0	A complex sequence
`rep(1:2, times=3)`	1 2 1 2 1 2	Repeat a vector
`rep(1:2, each=3)`	1 1 1 2 2 2	Repeat elements of a vector

Vector Functions

`sort(x)`
Return x sorted.

`rev(x)`
Return x reversed.

`table(x)`
See counts of values.

`unique(x)`
See unique values.

Selecting Vector Elements

By Position

`x[4]`	The fourth element.
`x[-4]`	All but the fourth.
`x[2:4]`	Elements two to four.
`x[-(2:4)]`	All elements except two to four.
`x[c(1, 5)]`	Elements one and five.

By Value

`x[x == 10]`	Elements which are equal to 10.
`x[x < 0]`	All elements less than zero.
`x[x %in% c(1, 2, 5)]`	Elements in the set 1, 2, 5.

Named Vectors

`x['apple']`	Element with name 'apple'.

Matrixes

`m <- matrix(x, nrow = 3, ncol = 3)`
Create a matrix from x.

`m[2, ]` - Select a row

`m[ , 1]` - Select a column

`m[2, 3]` - Select an element

`t(m)`
Transpose

`m %*% n`
Matrix Multiplication

`solve(m, n)`
Find x in: m * x = n

Lists

`l <- list(x = 1:5, y = c('a', 'b'))`
A list is collection of elements which can be of different types.

`l[[2]]`	`l[1]`	`l$x`	`l['y']`
Second element of l.	New list with only the first element.	Element named x.	New list with only element named y.

Also see the **dplyr** library.

Data Frames

`df <- data.frame(x = 1:3, y = c('a', 'b', 'c'))`
A special case of a list where all elements are the same length.

x	y
1	a
2	b
3	c

List subsetting

`df$x` `df[[2]]`

Understanding a data frame

`View(df)` See the full data frame.

`head(df)` See the first 6 rows.

Matrix subsetting

`df[ , 2]`

`df[2, ]`

`df[2, 2]`

`nrow(df)`
Number of rows.

`ncol(df)`
Number of columns.

`dim(df)`
Number of columns and rows.

cbind - Bind columns.

rbind - Bind rows.

RStudio® is a trademark of RStudio, Inc. • CC BY Mhairi McNeill • mhairihmcneill@gmail.com

Introduction to the Tidyverse

readr

tidyr

dplyr

tibble

ggplot2

stringr

forcats

purrr

https://www.tidyverse.org/packages/

Tidy Data

“Happy families are all alike; every unhappy family is unhappy in its own way.”
— Leo Tolstoy

“Tidy datasets are all alike, but every messy dataset is messy in its own way.”
— Hadley Wickham

There are three interrelated rules which make a dataset tidy:

- Each variable must have its own column.

- Each observation must have its own row.

- Each value must have its own cell.

variables observations values

Tidyverse is based on using Tibbles

```
> as_tibble(iris)
# A tibble: 150 × 5
   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
          <dbl>       <dbl>        <dbl>       <dbl> <fct>  
 1          5.1         3.5          1.4         0.2 setosa 
 2          4.9         3            1.4         0.2 setosa 
 3          4.7         3.2          1.3         0.2 setosa 
 4          4.6         3.1          1.5         0.2 setosa 
 5          5           3.6          1.4         0.2 setosa 
 6          5.4         3.9          1.7         0.4 setosa 
 7          4.6         3.4          1.4         0.3 setosa 
 8          5           3.4          1.5         0.2 setosa 
 9          4.4         2.9          1.4         0.2 setosa 
10          4.9         3.1          1.5         0.1 setosa 
# ℹ 140 more rows
# ℹ Use `print(n = ...)` to see more rows
> 
```

```
> iris
    Sepal.Length Sepal.Width Petal.Length Petal.Width    Species
1            5.1         3.5          1.4         0.2     setosa
2            4.9         3.0          1.4         0.2     setosa
3            4.7         3.2          1.3         0.2     setosa
4            4.6         3.1          1.5         0.2     setosa
5            5.0         3.6          1.4         0.2     setosa
6            5.4         3.9          1.7         0.4     setosa
7            4.6         3.4          1.4         0.3     setosa
8            5.0         3.4          1.5         0.2     setosa
9            4.4         2.9          1.4         0.2     setosa
10           4.9         3.1          1.5         0.1     setosa
11           5.4         3.7          1.5         0.2     setosa
12           4.8         3.4          1.6         0.2     setosa
13           4.8         3.0          1.4         0.1     setosa
14           4.3         3.0          1.1         0.1     setosa
15           5.8         4.0          1.2         0.2     setosa
16           5.7         4.4          1.5         0.4     setosa
17           5.4         3.9          1.3         0.4     setosa
18           5.1         3.5          1.4         0.3     setosa
19           5.7         3.8          1.7         0.3     setosa
20           5.1         3.8          1.5         0.3     setosa
21           5.4         3.4          1.7         0.2     setosa
22           5.1         3.7          1.5         0.4     setosa
23           4.6         3.6          1.0         0.2     setosa
24           5.1         3.3          1.7         0.5     setosa
25           4.8         3.4          1.9         0.2     setosa
26           5.0         3.0          1.6         0.2     setosa
27           5.0         3.4          1.6         0.4     setosa
```

Tidyverse is based on using Tibbles

Tibbles are data frames, but with a series of advantages:

- Printing is much tidier for large data

- Creating a tibble never changes the type of the inputs

  (e.g. it never converts strings to factors!)

- Creating a tibble never changes the names of variables

- Creating a tibble never creates row names

- Non-syntactic names are allowed for columns (delimited by backticks ` `)

- Subsetting of a tibble always gives another tibble

- You can still use [ ] for subsetting a tibble. However, functions dplyr::filter() and dplyr::select() allow you to solve the same problems with clearer code

https://r4ds.had.co.nz/tibbles.html

DOWNLOAD

RStudio IDE

The most popular coding environment for R, built with love by Posit.

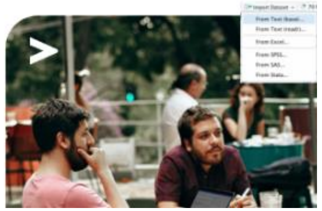
Used by millions of people weekly, the RStudio integrated development environment (IDE) is a set of tools built to help you be more productive with R and Python. It includes a console, syntax-highlighting editor that supports direct code execution. It also features tools for plotting, viewing history, debugging and managing your workspace.

RStudio Desktop RStudio Server

RStudio Desktop

Find out more about RStudio Desktop and RStudio Desktop Pro below.

DOWNLOAD RSTUDIO



https://posit.co/downloads/