Applied Statistics Using R

Introduction to R

- R is a comprehensive statistical and graphical programming language and is origin from the S language:
  - 1988 - S2: RA Becker, JM Chambers, A Wilks
  - 1992 - S3: JM Chambers, TJ Hastie
  - 1998 - S4: JM Chambers
- R: initially written by Ross Ihaka and Robert Gentleman at Dep. of Statistics of U of Auckland, New Zealand during 1993s.
- Since 1997: international “R-core” team of 15 people with access to common CVS archive.

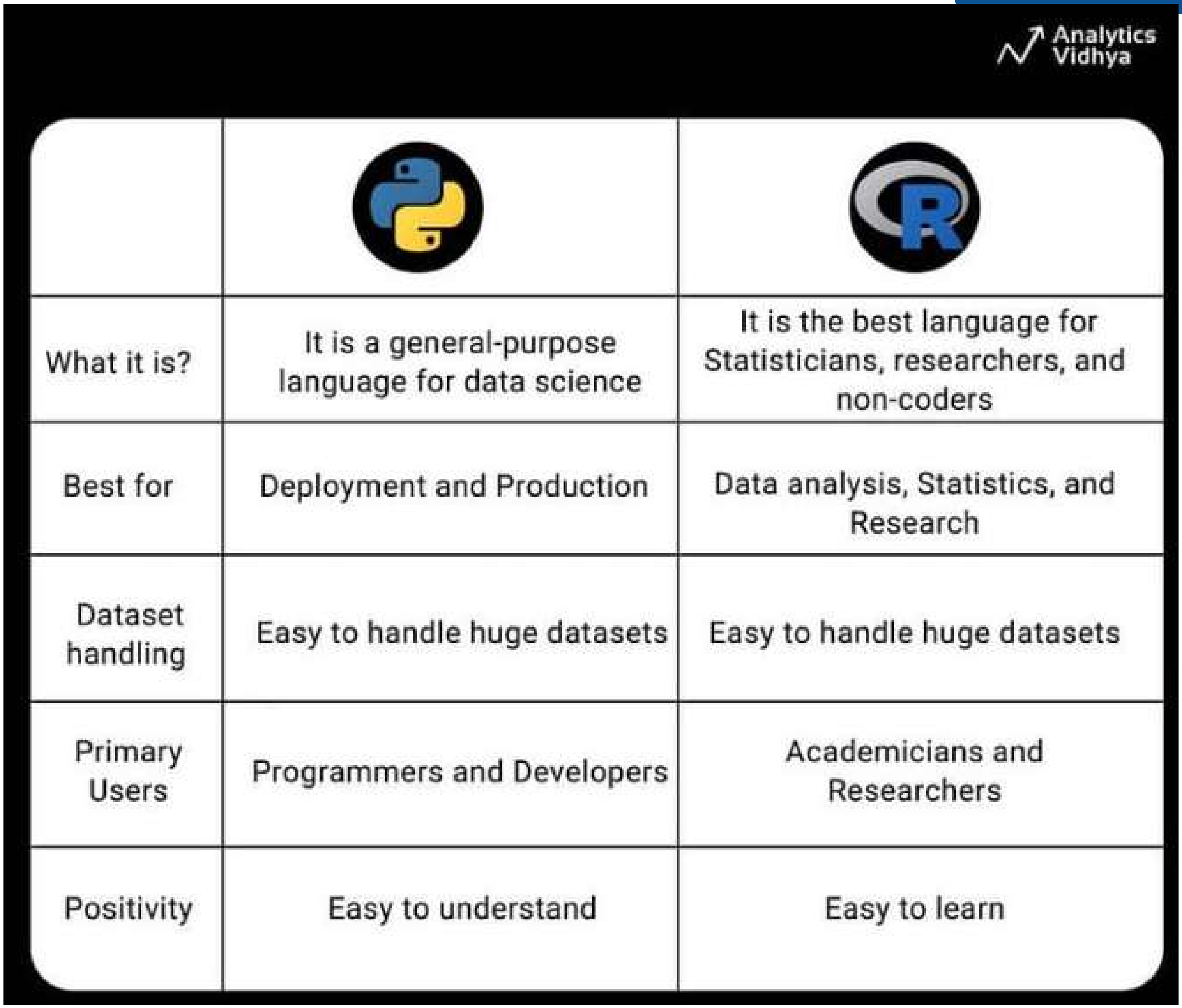
What is R

- Software for Statistical Data Analysis
- Based on S programming language
- Powerful Programming Environment
- Interpreted Language
- Many statistical built in functions
- Data Storage, Analysis, Graphing
- Free and Open Source Software
- R – Studio : open-source integrated development environment (IDE) for R

Strengths and Weaknesses

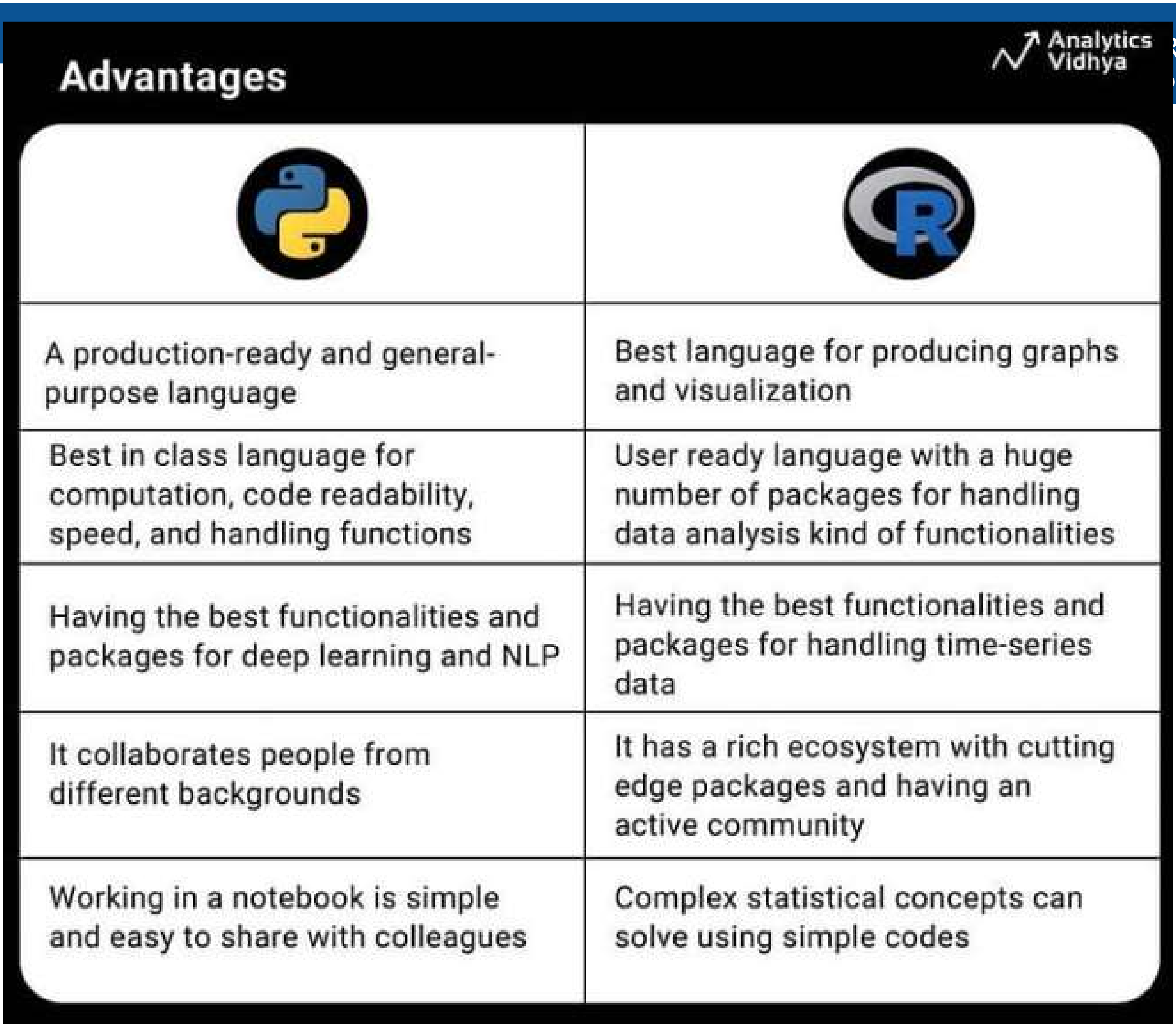
- **Strengths**
  - Free and Open Source
  - Strong User Community
  - Highly extensible, flexible
  - Implementation of high end statistical methods
  - Flexible graphics and intelligent defaults
- **Weakness**
  - Steep learning curve
  - Slow for large datasets

Why R?



Analytics Vidhya

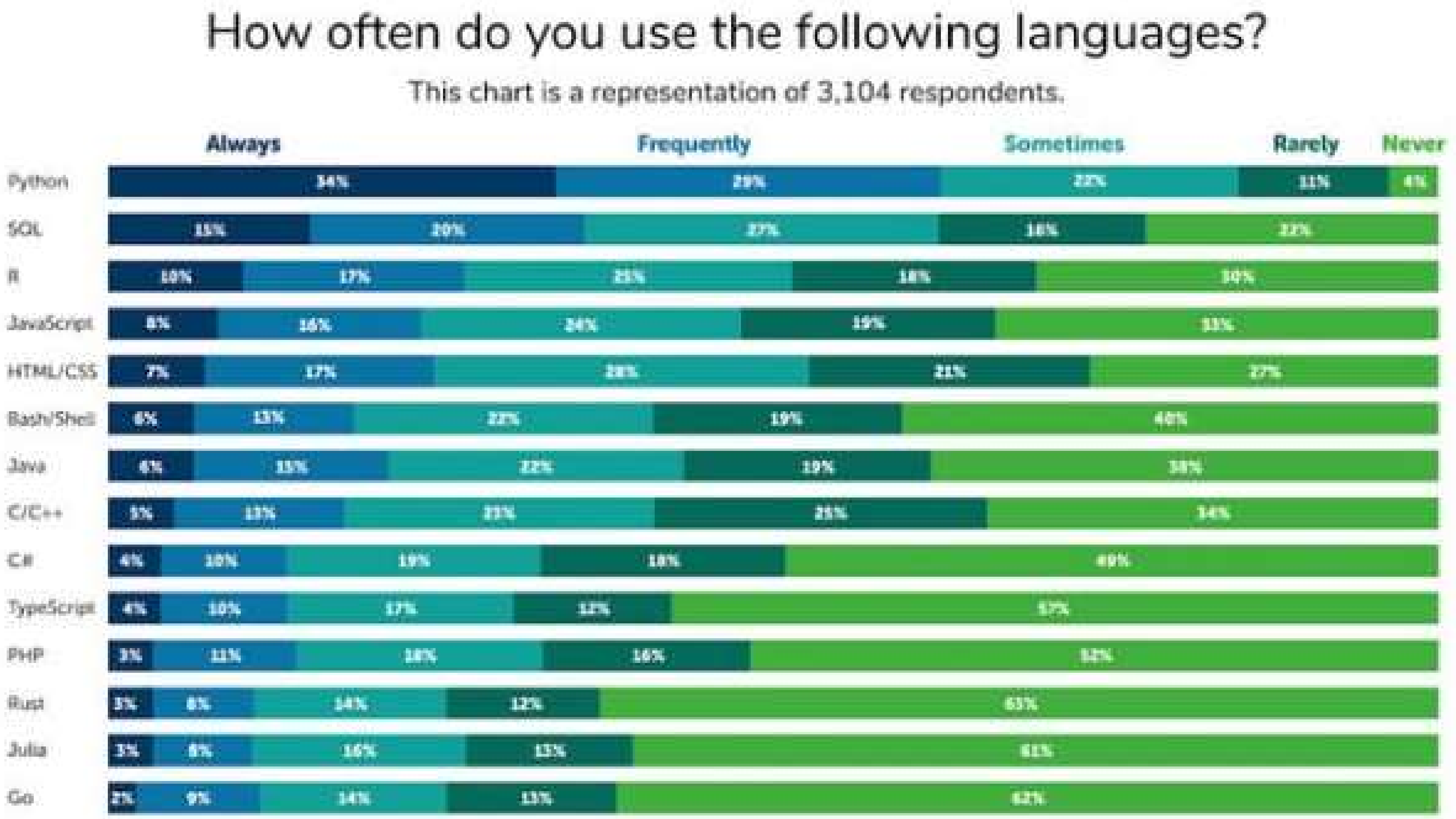
		R
What it is?	It is a general-purpose language for data science	It is the best language for Statisticians, researchers, and non-coders
Best for	Deployment and Production	Data analysis, Statistics, and Research
Dataset handling	Easy to handle huge datasets	Easy to handle huge datasets
Primary Users	Programmers and Developers	Academicians and Researchers
Positivity	Easy to understand	Easy to learn



Advantages

Analytics Vidhya

A production-ready and general-purpose language	Best language for producing graphs and visualization
Best in class language for computation, code readability, speed, and handling functions	User ready language with a huge number of packages for handling data analysis kind of functionalities
Having the best functionalities and packages for deep learning and NLP	Having the best functionalities and packages for handling time-series data
It collaborates people from different backgrounds	It has a rich ecosystem with cutting edge packages and having an active community
Working in a notebook is simple and easy to share with colleagues	Complex statistical concepts can solve using simple codes



How often do you use the following languages?

This chart is a representation of 3,104 respondents.

Installing R & R-Studio

- RStudio requires R 2.11.1 (or higher) to run
- Link to Download R
  - https://www.r-project.org/

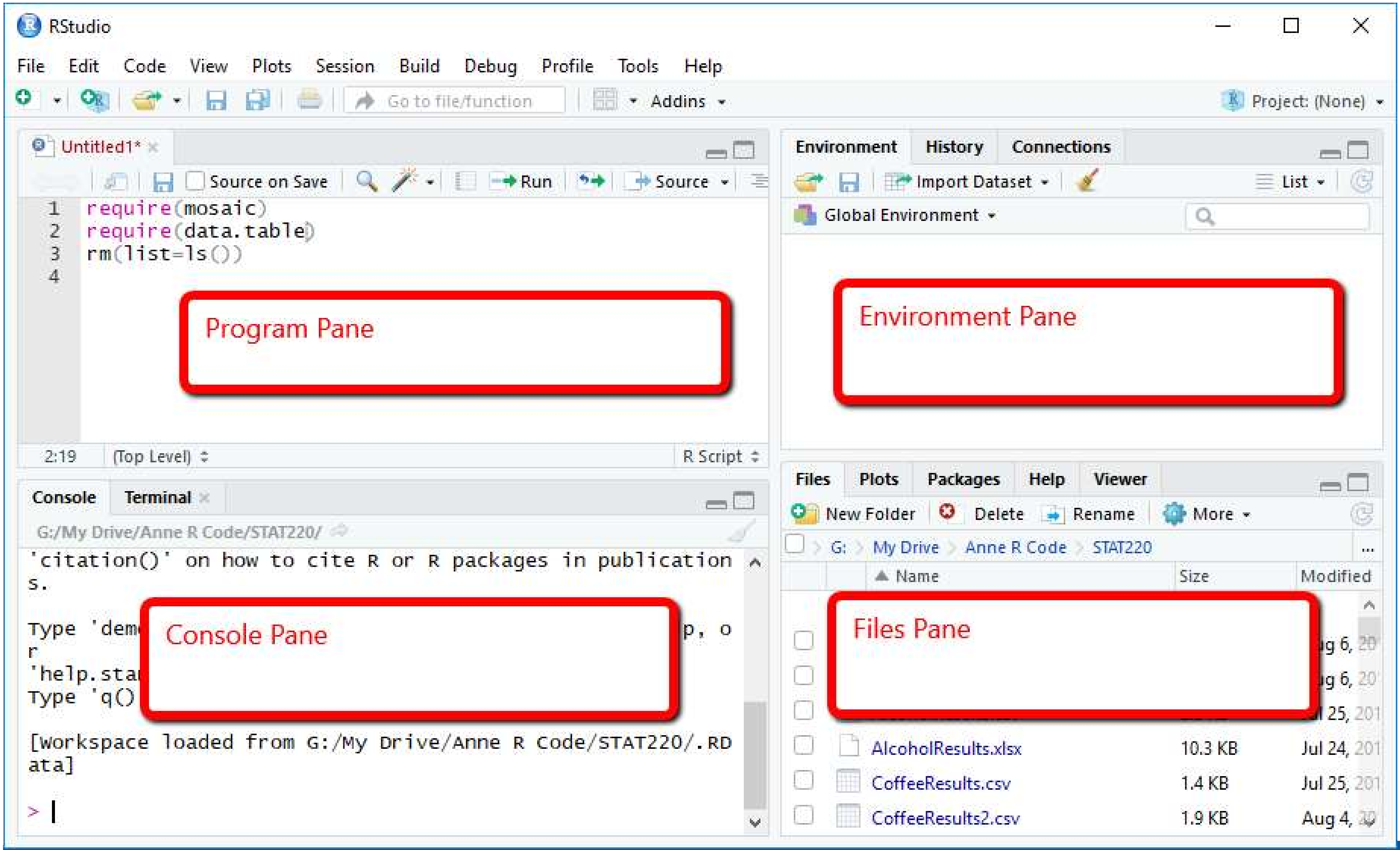
- Link to Download R Studio
  - https://www.rstudio.com/products/rstudio/download/

- Installation Help
  - https://www.rstudio.com/products/rstudio/download/

What is R Studio?

- Customizable workbench with all of the tools required to work with R in one place (console, source, plots, workspace, help, history, etc.).
- Syntax highlighting editor with code completion.
- Execute code directly from the source editor (line, selection, or file).
- Full support for authoring Sweave and TeX documents.
- Runs on all major platforms (Windows, Mac, and Linux) and can also be run as a server, enabling multiple users to access the RStudio IDE using a web browser.

RStudio User Interface


RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Project: (None)
Untitled1*
Source on Save
Run
Source
Program Pane
2:19
(Top Level)
R Script
Environment History Connections
Import Dataset
List
Global Environment
Environment Pane
Console Terminal
G:/My Drive/Anne R Code/STAT220/
Console Pane
Files Plots Packages Help Viewer
New Folder Delete Rename More
G: > My Drive > Anne R Code > STAT220
Files Pane

```
require(mosaic)
require(data.table)
rm(list=ls())
```

```
'citation()' on how to cite R or R packages in publication
s.

Type 'dem                                              p, o
r
'help.sta
Type 'q()

[Workspace loaded from G:/My Drive/Anne R Code/STAT220/.RD
ata]

> |
```

Name	Size	Modified
		ug 6, 20
		ug 6, 20
		l 25, 201
AlcoholResults.xlsx	10.3 KB	Jul 24, 201
CoffeeResults.csv	1.4 KB	Jul 25, 201
CoffeeResults2.csv	1.9 KB	Aug 4, 20

Tips and Reminders

- R is case-sensitive
- comments are preceded with #
- R scripts are simply text files with a .R extension
- Use Ctrl + R to submit code
- Use the Tab key to let R/R Studio finish typing commands for you
- Use Shift + down arrow to highlight lines or blocks of code
- In R Studio: Ctrl + 1 and Ctrl + 2 switches between script and console
- Use up and down arrows to cycle through previous commands in console

R Workspace

- collection of objects that you currently have
- When you close the RGui or the R console window, R session save the workspace image then all the objects in your current are saved in a file .RData. (a binary file located in the working directory of R)
- explicitly save the workspace image

```
## save to the current working directory
save.image()
## just checking what the current working directory is
getwd()
## save to a specific file and location
save.image("C :\\Program Files\\R\\R-2.5.0\\bin\\.RData ")
```

- To restore the workspace
  - Click File menu → Load workspace

Getting help with functions and features

- R has inbuilt help facility
- To get more information on any specific named function
  - help(mean)
  - ?mean
- Allows searching for help in various ways
  - ??mean

Brackets

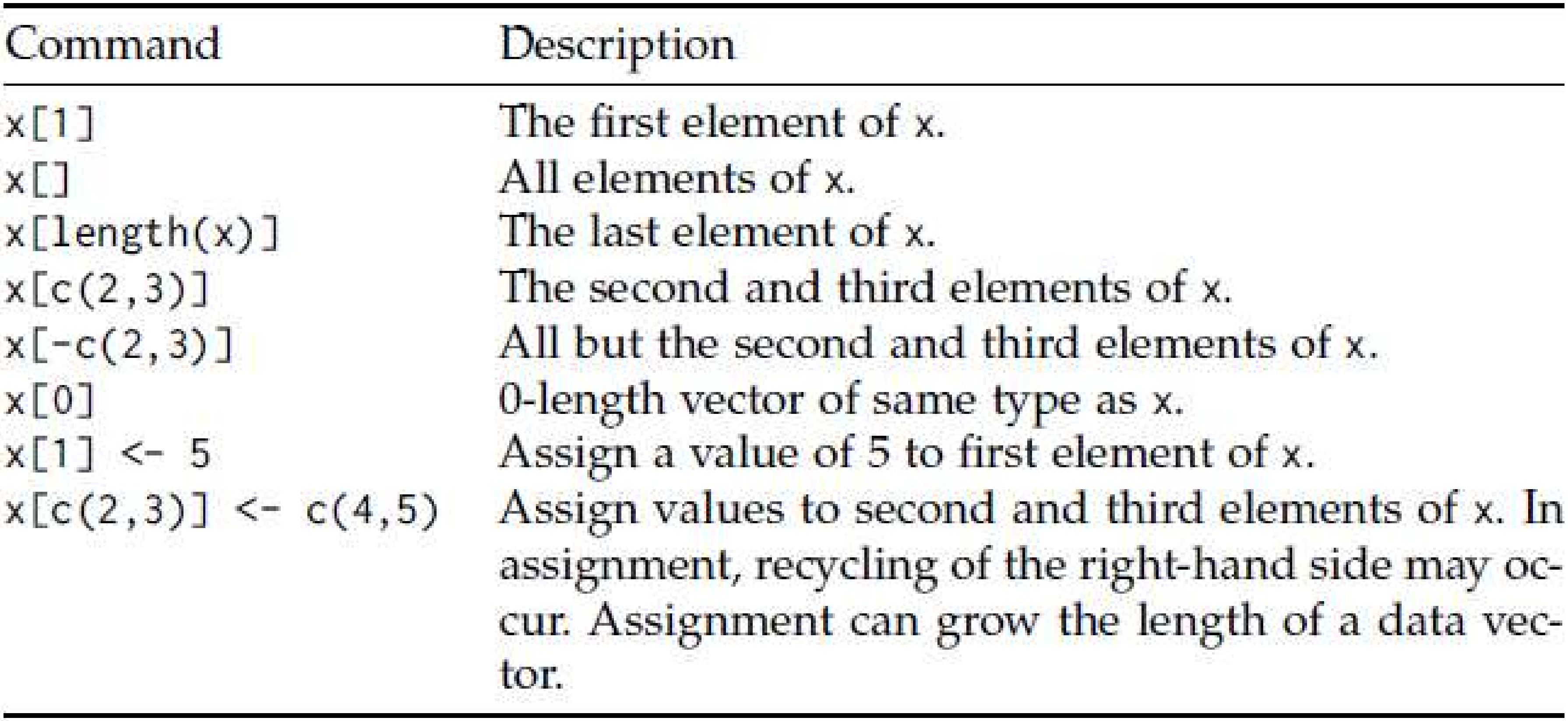
- () round brackets as the standard brackets
- [] box brackets if we are dealing with index positons of vectors
  - x=c(3.1, 4, 3, 4.1, 5)
  - y1=x[1:4]
  - Y2 = x[-(1:5)]
- {} curled brackets for conditional statements, loops, and functions

Vector Examples

- assign("x", c(10.4, 5.6, 3.1, 6.4, 21.7))
- c(10.4, 5.6, 3.1, 6.4, 21.7) = x
- x = c(10.4, 5.6, 3.1, 6.4, 21.7)
- X = c(10.4, 8.6, 3, 6, 21)

  1/X
- y = c(x, 0, x)
- sum((x-mean(x))^2)/(length(x)-1)

Various uses of indexing by numeric indices

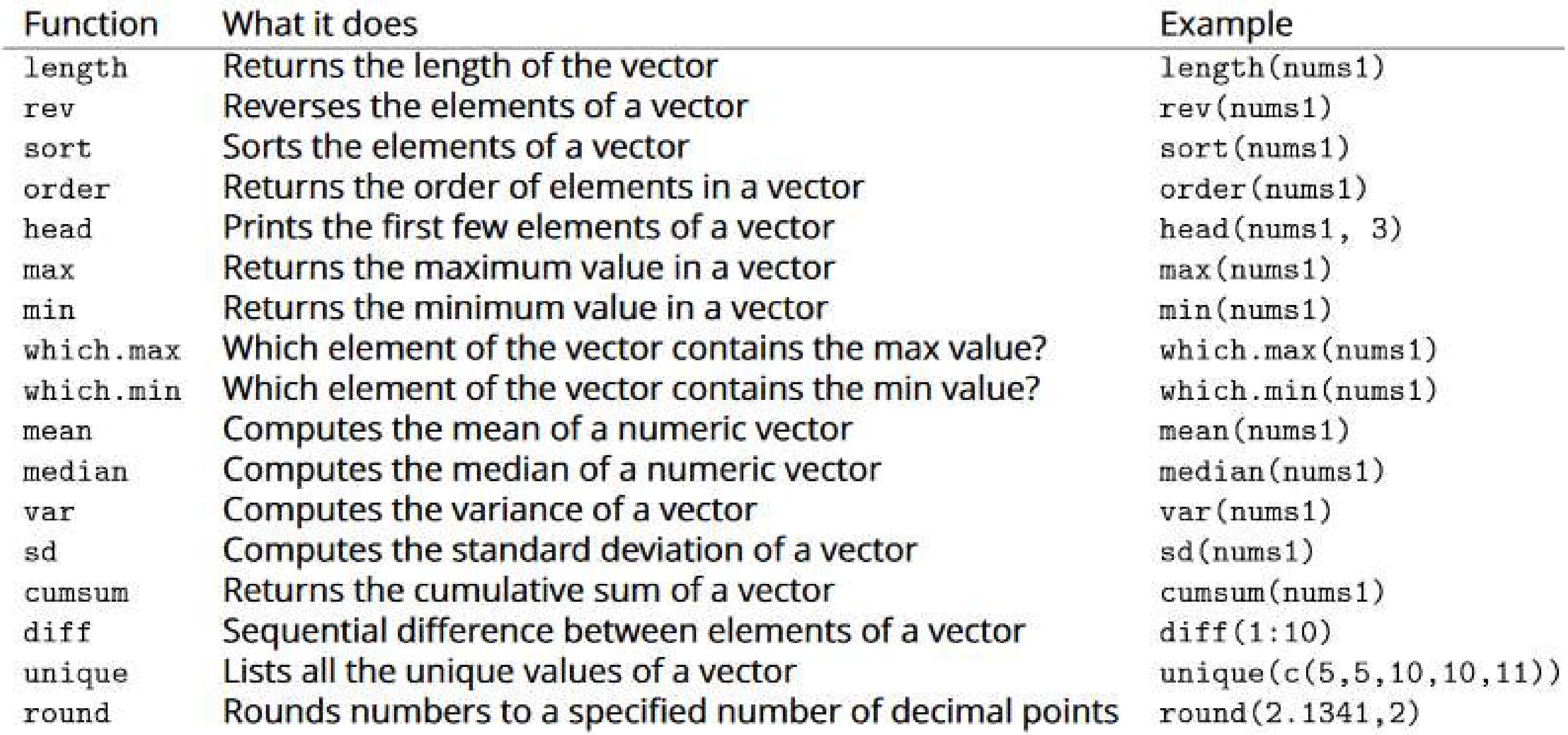


Command	Description
`x[1]`	The first element of x.
`x[]`	All elements of x.
`x[length(x)]`	The last element of x.
`x[c(2,3)]`	The second and third elements of x.
`x[-c(2,3)]`	All but the second and third elements of x.
`x[0]`	0-length vector of same type as x.
`x[1] <- 5`	Assign a value of 5 to first element of x.
`x[c(2,3)] <- c(4,5)`	Assign values to second and third elements of x. In assignment, recycling of the right-hand side may occur. Assignment can grow the length of a data vector.

Missing Values

- NA – Not Availabe
  - z = c(1:3,NA)
  - ind = is.na(z)
- NaN -Not a Number
  - 0/0
  - Inf – Inf
- <NA>
  - character vectors are printed without quotes

Useful functions for vectors



Function	What it does	Example
`length`	Returns the length of the vector	`length(nums1)`
`rev`	Reverses the elements of a vector	`rev(nums1)`
`sort`	Sorts the elements of a vector	`sort(nums1)`
`order`	Returns the order of elements in a vector	`order(nums1)`
`head`	Prints the first few elements of a vector	`head(nums1, 3)`
`max`	Returns the maximum value in a vector	`max(nums1)`
`min`	Returns the minimum value in a vector	`min(nums1)`
`which.max`	Which element of the vector contains the max value?	`which.max(nums1)`
`which.min`	Which element of the vector contains the min value?	`which.min(nums1)`
`mean`	Computes the mean of a numeric vector	`mean(nums1)`
`median`	Computes the median of a numeric vector	`median(nums1)`
`var`	Computes the variance of a vector	`var(nums1)`
`sd`	Computes the standard deviation of a vector	`sd(nums1)`
`cumsum`	Returns the cumulative sum of a vector	`cumsum(nums1)`
`diff`	Sequential difference between elements of a vector	`diff(1:10)`
`unique`	Lists all the unique values of a vector	`unique(c(5,5,10,10,11))`
`round`	Rounds numbers to a specified number of decimal points	`round(2.1341,2)`

Exercises-1

- Suppose you keep track of your mileage each time you fill up. At your last fllups the mileage was

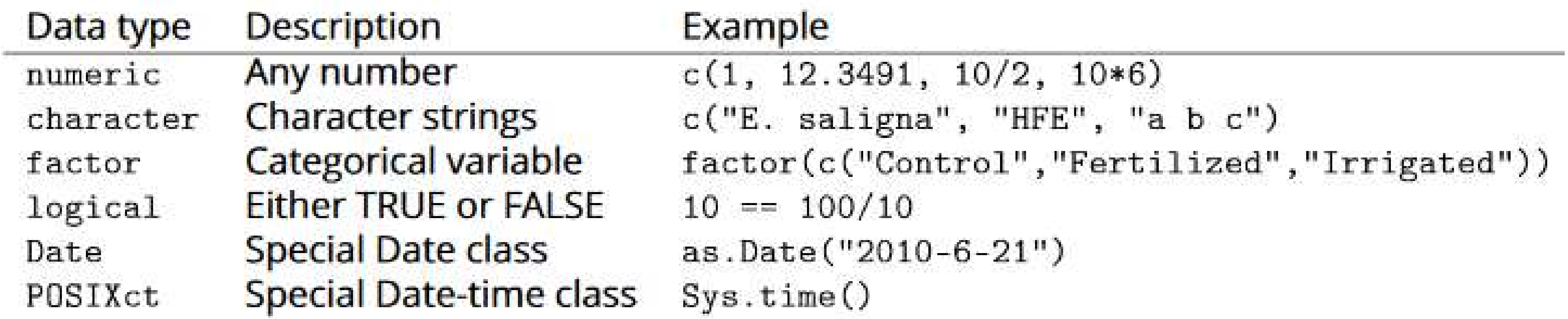
65311 65624 65908 66219 66499 66821 67145 67447

Enter these numbers into R. Use the function diff on the data. What does it give?

- Your cell phone bill varies from month to month. Suppose the year has following monthly amounts
  460 330 390 370 460 300 480 320 490 350 300 480

Enter this data into a object called bill. Use the sum command to add the amount you spent this year on the cell phone. What is the smallest amount you spent in a month? What is the largest? How many months was the amount greater than 400? What percentage was this?

R – Data Types



Data type	Description	Example
`numeric`	Any number	`c(1, 12.3491, 10/2, 10*6)`
`character`	Character strings	`c("E. saligna", "HFE", "a b c")`
`factor`	Categorical variable	`factor(c("Control","Fertilized","Irrigated"))`
`logical`	Either TRUE or FALSE	`10 == 100/10`
`Date`	Special Date class	`as.Date("2010-6-21")`
`POSIXct`	Special Date-time class	`Sys.time()`

Operators

- **Arithmetic Operator**
  - **+ - * / ^ %% %/%**
  - x = 5
  - y = 16
  - x+y
  - y%/%x
- **Relational Operator**
  - **<, >, <=, >=, ==, !=**
  - x<y
  - y>=20
  - x != 5

- **Operation on Vector**
  - x = c(2,8,3)
  - y = c(6,4,1)
  - x+y
  - x>y

- **Logical operator**
  - **!, &, &&, |, ||**
- **Assignment Operator**
  - **<-, <<-, =**
  - **->, ->>**
  - <<-, ->> is used for assigning to variables in the parent environments

Data Sequences

- seq(3, 5)
- seq(from = 3, to = 5)
- seq(from = 3, length =5)
- seq(from = 3, length =3, by = 0.5)
- seq(from = 3, by = 0.5, length =3)
  - Note: Argument order does not matter

Paste Function - Characters

- paste ("xyz", 1:10)
- paste ("xyz", c(2,5,7,"test",4.5))
- paste("xyz", 1:10, sep = "")
- paste("xyz", 1:10, sep = "/")

Repeat Sequences

- rep (c(3,4,5), 3)
- rep(1:10, times = 3)
- X = c(1,2,3)
- rep (x, each = 3)
- rep (x, each = 3, times = 3)

Position and Value

- x = c(4:20)
- which (x==10) #position of 10 in x
- x[3]   # value in the third position of x

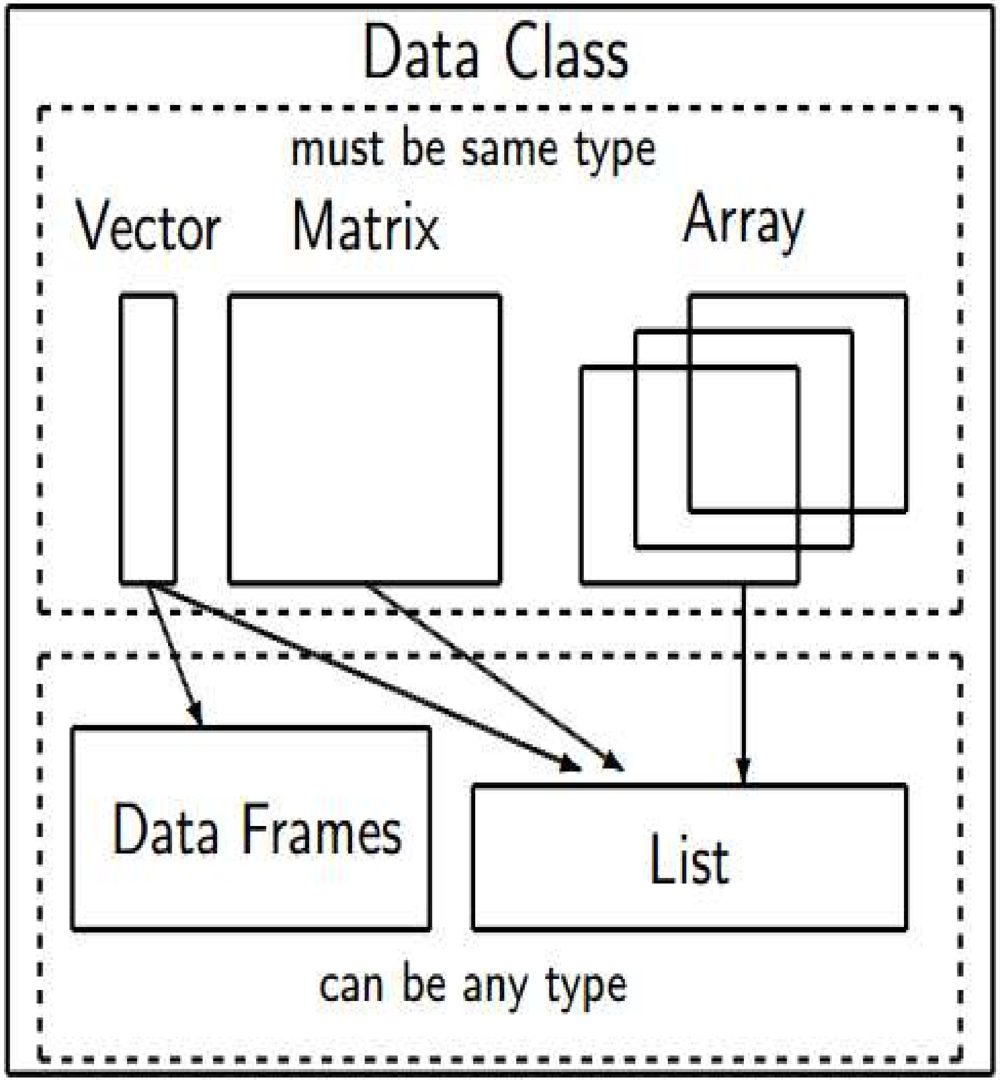
Exercises - 2

1. Define the object "myobject" and assign the vector 1:10 in at least 3 different ways
2. Get the sum of your object
3. Create the following vector by using the paste function
   - [1] "R is great 4 and I will love it"
   - [2] "R is great 7 and I will love it"
   - [3] "R is great 45 and I will love it"
4. Vector of 1,2,3, repeat the vector to get eleven 1s, ten 2s and ten 3s
5. What is the value of this vector on position 7?

Answers

1. Define the object "myobject" and assign the vector 1:10 in at least 3 different ways
   - ○ myobject <- (1:10)
   - ○ myobject = (1:10)
   - ○ (1:10) -> myobject
   - ○ assign("myobject", 1:10)
2. Get the sum of your object
   - ○ sum (myobject)
3. Create the following vector by using the paste function
   - ○ [1] "R is great 4 and I will love it"
   - ○ [2] "R is great 7 and I will love it"
   - ○ [3] "R is great 45 and I will love it"
   - ○ paste ("R is great", c(4,7,45), "and I will love it")
4. Vector of 1,2,3, repeat the vector to get eleven 1s, ten 2s and ten 3s
   - ○ rep (1:3, length = 31)
5. What is the value of this vector on position 7?
   - ○ x <- rep (1:3, length = 31)
   - ○ x[7] ➔ ?

Data Structure in R



Data Structure in R Cont…

- str() – to know the structure of data
- Vector – Basic data structure in R
    - numeric_vector = c(1, 2, 3)
    - character_vector = c("a", "b", "c")
    - boolean_vector =c(T,F,T,F,F)
  - typeof() # what it is.
  - length() # how many elements it contains.

Data Structure in R Cont…

- **Matrix -** collection of elements of the same data type (numeric, character, or logical) arranged into a fixed number of rows and columns
- **matrix() – construct matrix**
- **factor() - used to store Categorical data**
  - gender_vector <- c("Male","Female","Female","Male","Male")
  - factor_gender_vector <- factor(gender_vector)

Data Structure in R Cont…

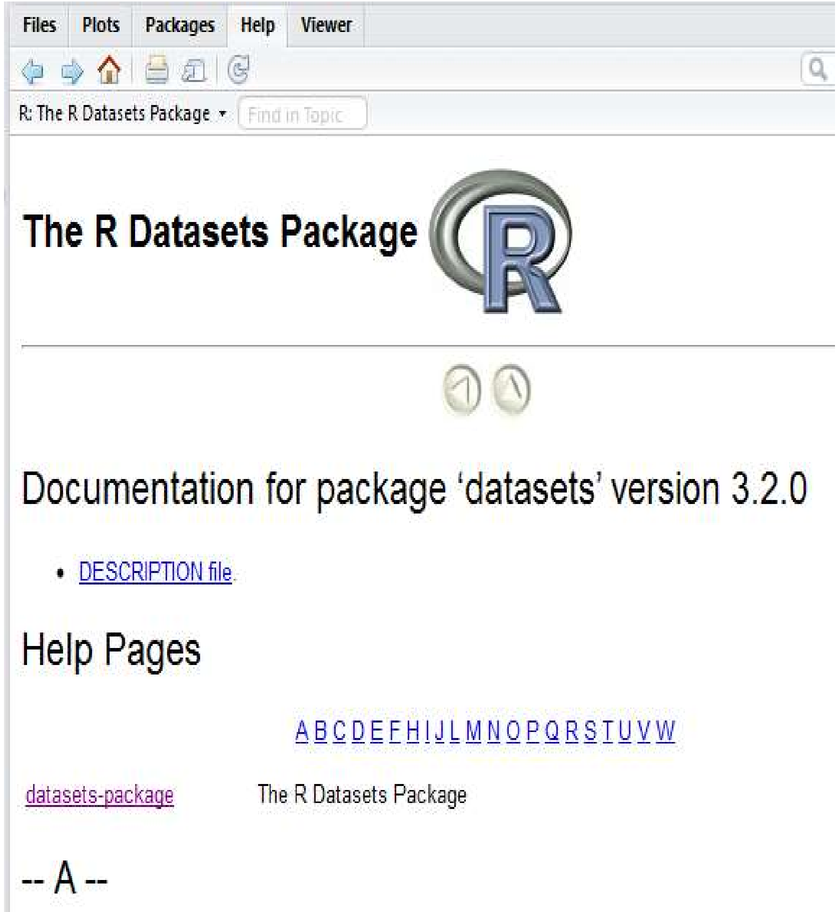
- **Lists -** hold components of different types

  -
    ```
    n = c(2, 3, 5)
     s = c("aa", "bb", "cc", "dd", "ee")
     b = c(TRUE, FALSE, TRUE, FALSE, FALSE)
    x = list(n, s, b, 3)
    ```
  - `x[c(2, 4)] ; x[2]`
- **DataFrames – used to store data table with different type**

  ```
  df<- data.frame(x =1:3, y = c("a", "b", "c"), stringsAsFactors =FALSE)
  ```

  - **combine data frames using cbind() and rbind()**
    - `cbind(df, data.frame(z = 3:1))`
    - `rbind(df, data.frame(x = 10, y = "z"))`

Internal Datasets



The R Datasets Package

Documentation for package ‘datasets’ version 3.2.0

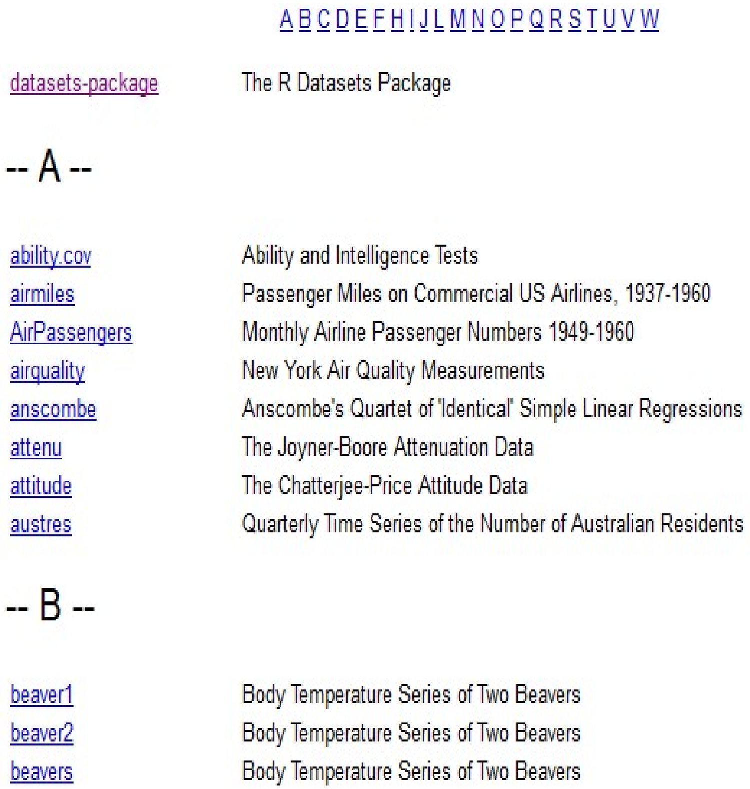
- DESCRIPTION file.

Help Pages

A B C D E F H I J L M N O P Q R S T U V W

datasets-package	The R Datasets Package

-- A --



A B C D E F H I J L M N O P Q R S T U V W

datasets-package	The R Datasets Package

-- A --

ability.cov	Ability and Intelligence Tests
airmiles	Passenger Miles on Commercial US Airlines, 1937-1960
AirPassengers	Monthly Airline Passenger Numbers 1949-1960
airquality	New York Air Quality Measurements
anscombe	Anscombe's Quartet of 'Identical' Simple Linear Regressions
attenu	The Joyner-Boore Attenuation Data
attitude	The Chatterjee-Price Attitude Data
austres	Quarterly Time Series of the Number of Australian Residents

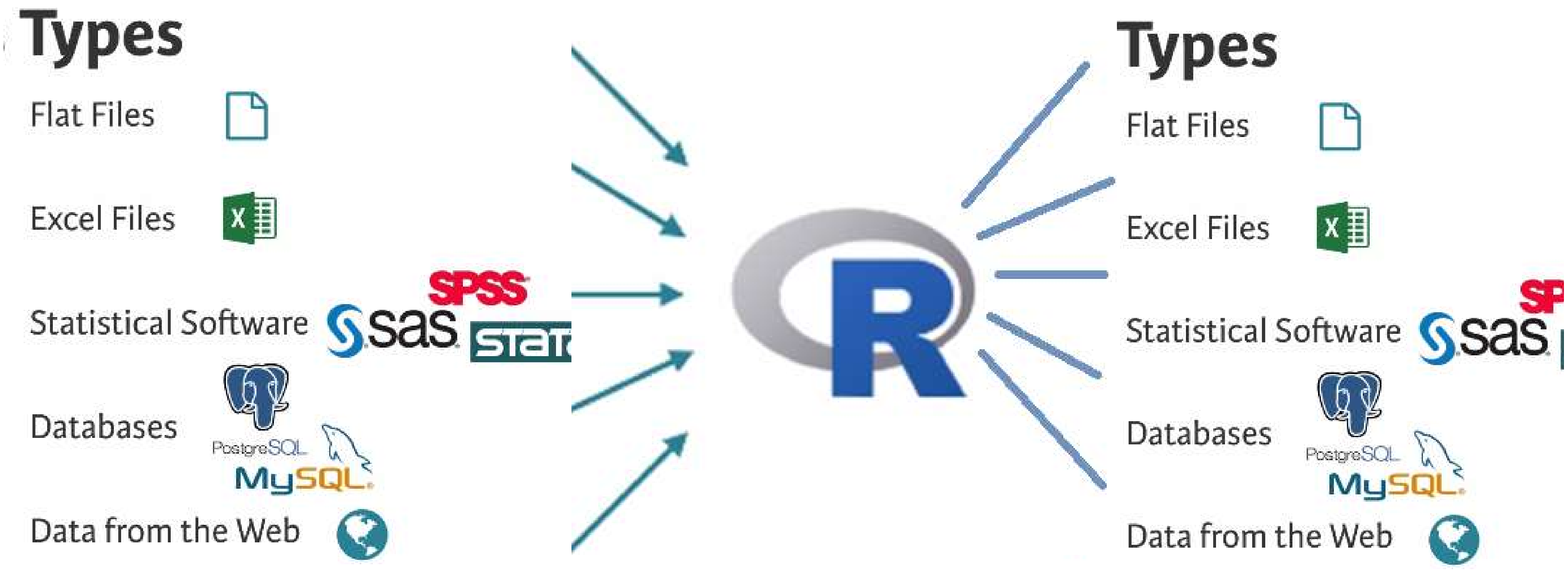
-- B --

beaver1	Body Temperature Series of Two Beavers
beaver2	Body Temperature Series of Two Beavers
beavers	Body Temperature Series of Two Beavers

Exercises - 3

- help(mtcars)
- head(mtcars)
- ncol(mtcars)
- mtcars[1, 2]
- mtcars["Mazda RX4", "cyl"]
- mtcars[[9]]
- mtcars[["am"]]
- mtcars$am
- mtcars[,"am"]

External Data Sources



Types

Flat Files

Excel Files

Statistical Software

Databases

Data from the Web

Types

Flat Files

Excel Files

Statistical Software

Databases

Data from the Web

Utils package

- Loaded by default when you start R
- read.table()
- Read any tabular files as a data frame
- Number of arguments is huge

Import Dataset

- **From Internet**
  - mydata 1<- read.table(url("http://assets.datacamp.com/blog_assets/chol.txt"), header = TRUE)
- **From Text File**
  - mydata1 <- read.table(“iris.txt")
- **From csv File**
  - mydata2 <- read.csv(“churn.csv")
- help(read.table)

Export Dataset

write.table(mtcars, file = "mtcars.txt", sep = “\t",   row.names = T, col.names = T)

write.csv(lynx, file = "lynx.csv")

Data Frame's Row And Column Names

```
names(mtcars)      # see the column names
mtcars1=mtcars   #Copy to other object
mtcars1
names(mtcars1) <- c("MilesPerGallon","NumOfCylinders",
                    "Displacement", "HorsePower",
                    "RearAxleRatio","Weight","MileTime",
                    "EngineType", "Transmission",
                    "NumOfGears", "NumOfCarburetors")
View(mtcars1)
dim(mtcars1)         # check number of rows and columns
length(mtcars1)      #check number of columns
mtcars1[1:2,3]
attach(mtcars1)     #attach data frame
```

- **DataFrames – used to store data table with different type**

```
nrow(mtcars)
ncol(mtcars)
tail(mtcars)
head(mtcars)
```

- Column vector

```
mtcars[[9]]
mtcars$am
mtcars[,"am"]
mtcars[,9]
```

- mtcars[mtcars$am == 0,]

- Row Slice

```
mtcars[24,]
mtcars[c(3, 24),]
mtcars[c(3:14),]
```

- Column Slice

```
mtcars[9]
mtcars$am
mtcars["am"]
mtcars[c("mpg", "am")]
```