Unit Admin

CITS4009 Computational Data Analysis

Dr. G. Mubashar Hassan

Department of Computer Science and Software Engineering
The University of Western Australia

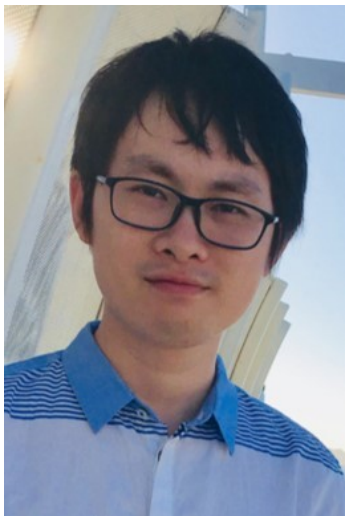
Semester 2, 2024

Teaching Team

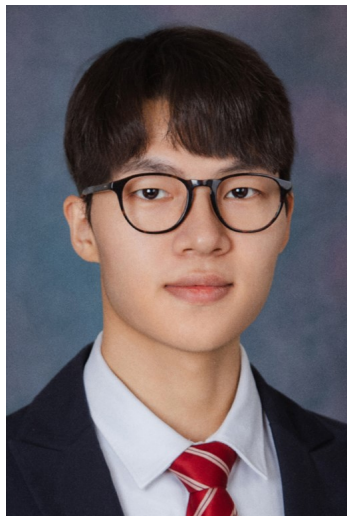
- **Unit Coordinator**: Dr. Ghulam Mubashar Hassan
- **Email Address**: ghulam.hassan@uwa.edu.au
- **Preferred Communication**: Email
- **Office**: CSSE Building 2.12
- **Consultation**: Tuesdays 1-2pm
- **Unit Website**: LMS
- **Lab Starts**: Week 2
- **Labs Venue**: CSSE 2.01
- **Lecture**: Wednesdays 12-2pm

Teaching Team - Lab Facilitators

Mila Zhang



Dr. Yangjinbo Zhang



Arthur Lian

Nicodemus Ong

Teaching Schedule

Week	Date	Lab	Lecture Topic
1	22 Jul	No Lab	Admin and DS Primer
2	29 Jul	Lab 01	Basic R and Visualization
3	5 Aug	Lab 02	EDA - Single Variable
4	12 Aug	Lab 03	EDA - Multiple Variables
5	19 Aug	Lab 04	Data Cleaning
6	26 Aug	Lab 05	Data Transformation
	4 Sep	**Study Break**	

Teaching Schedule (Contd.)

Week	Date	Lab	Lecture Topic
7	9 Sep		Classification Model Evaluation
7	13 Sep 3 pm	**Mid Sem Test**	ARTS: [G57] Alexander LT; ARTS: [G58] Murdoch LT WTLTS: [G106] Tattersal LT
8	16 Sep	Lab 06	Single Variable Models
9	23 Sep	Lab 07	Multi-Variable Models
10	30 Sep	Lab 08	Unsupervised Methods
11	7 Oct	Project	Linear and Logistic Regression
12	14 Oct	Revision	Exam Discussion

Assessment Components

Assessment	Description	Deadline	Percentage
Project	Exploratory Data Analytics and Predictive Data Models	Week 11	25%
Mid-Sem Test	w1-w6 inclusive	Week 7	15%
Final Exam	w1-w12 inclusive	Exam Period	60%

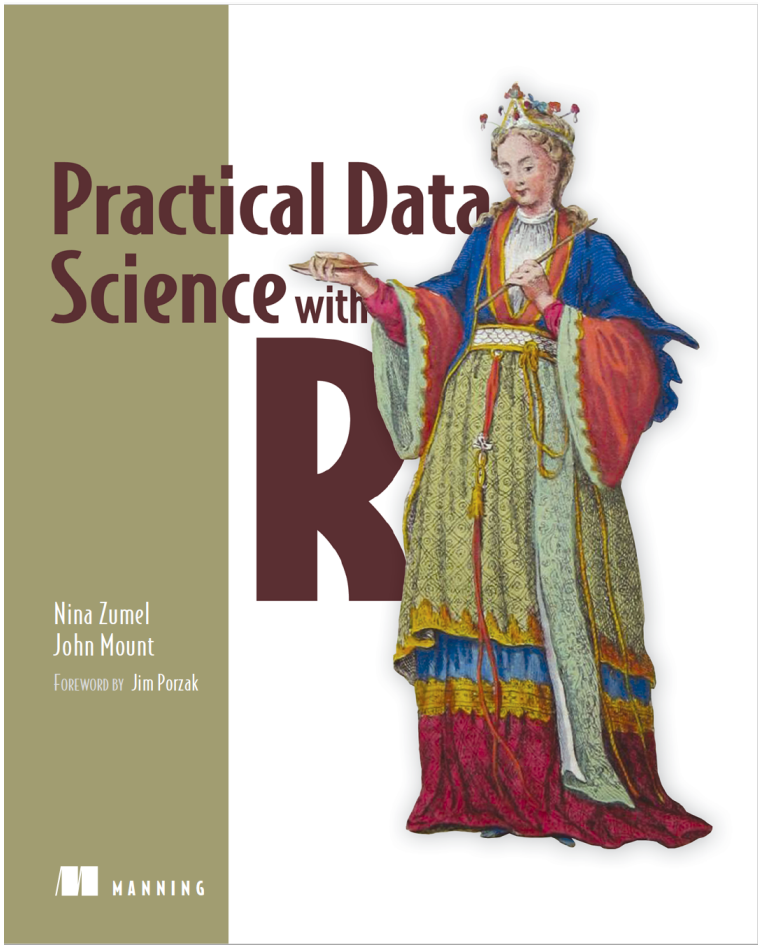
All extensions of deadlines or exemptions are handled by UWA student office located in EZONE North Building.

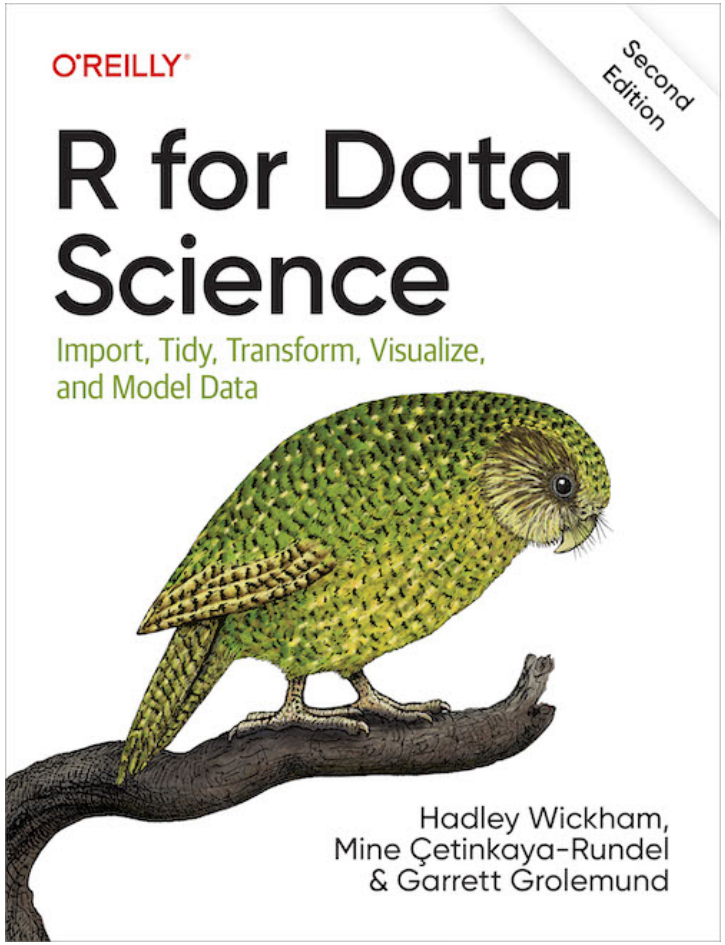
Project cannot be submitted after 7 days of deadline.

In case of exemption of any assessment, marks will be allocated to Final Exam.

Unit Readings

Free online: [R for Data Science (2e)] (https://r4ds.hadley.nz/)





Introduction to Data Science

CITS4009 Computational Data Analysis

Dr. Mubashar Hassan

Department of Computer Science and Software Engineering
The University of Western Australia

Semester 2, 2024

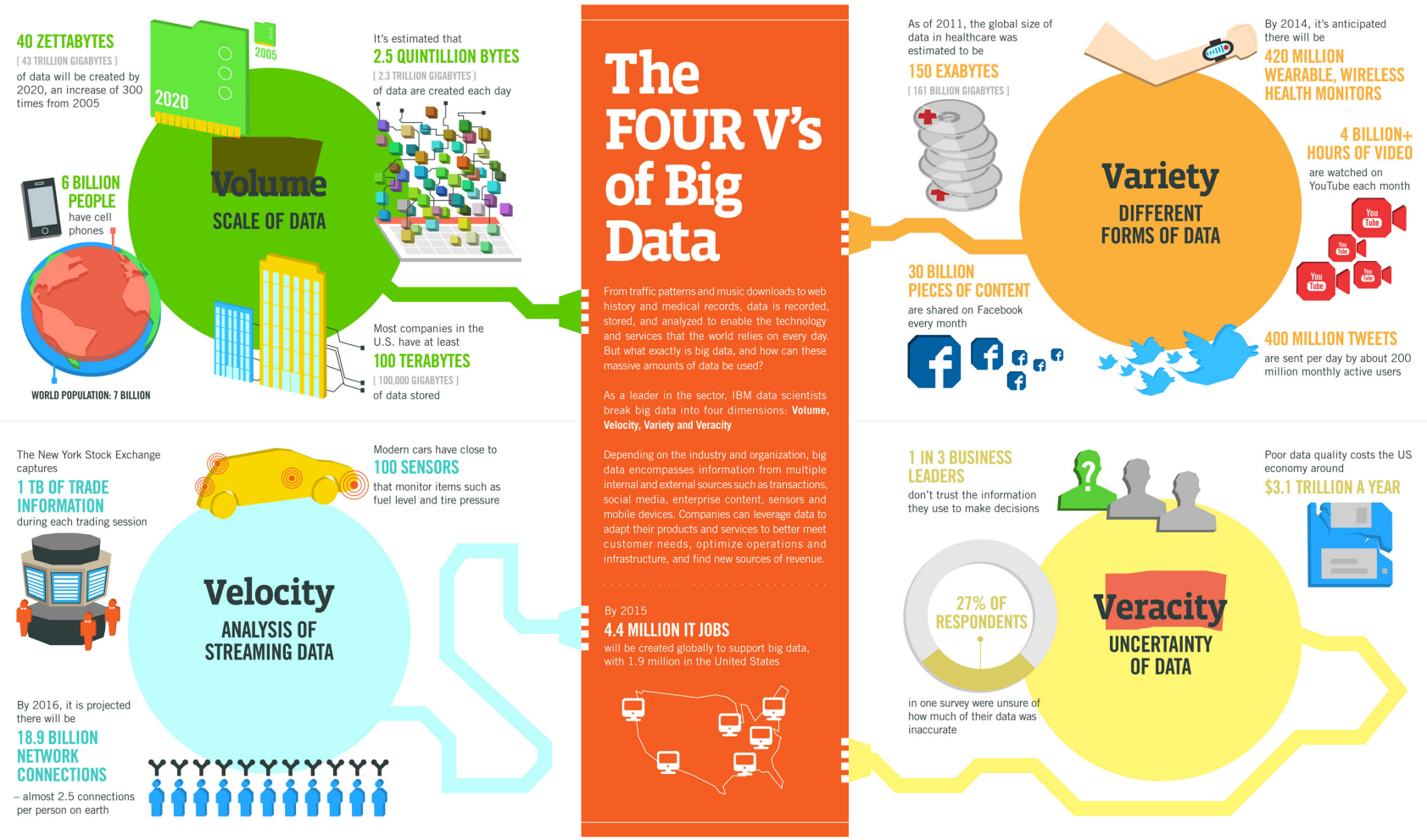
Section 1

Data Science Premier

What is Data Science?

- Data science is an multidisciplinary domain that allows you to turn raw data into understanding, insight, and knowledge.
- It typically involves in solving analytically complex problems using machine learning algorithms.
- Unlike traditional algorithms, machine learning algorithms are **data driven**.
  Data determines the algorithm's response.
- For example, for an application to detect faces, a non-machine learning algorithm would try to first define what a face is. In contrast, a machine learning algorithm "learns-by-examples" through lots of images grouped into *faces* and *non-faces* categories.

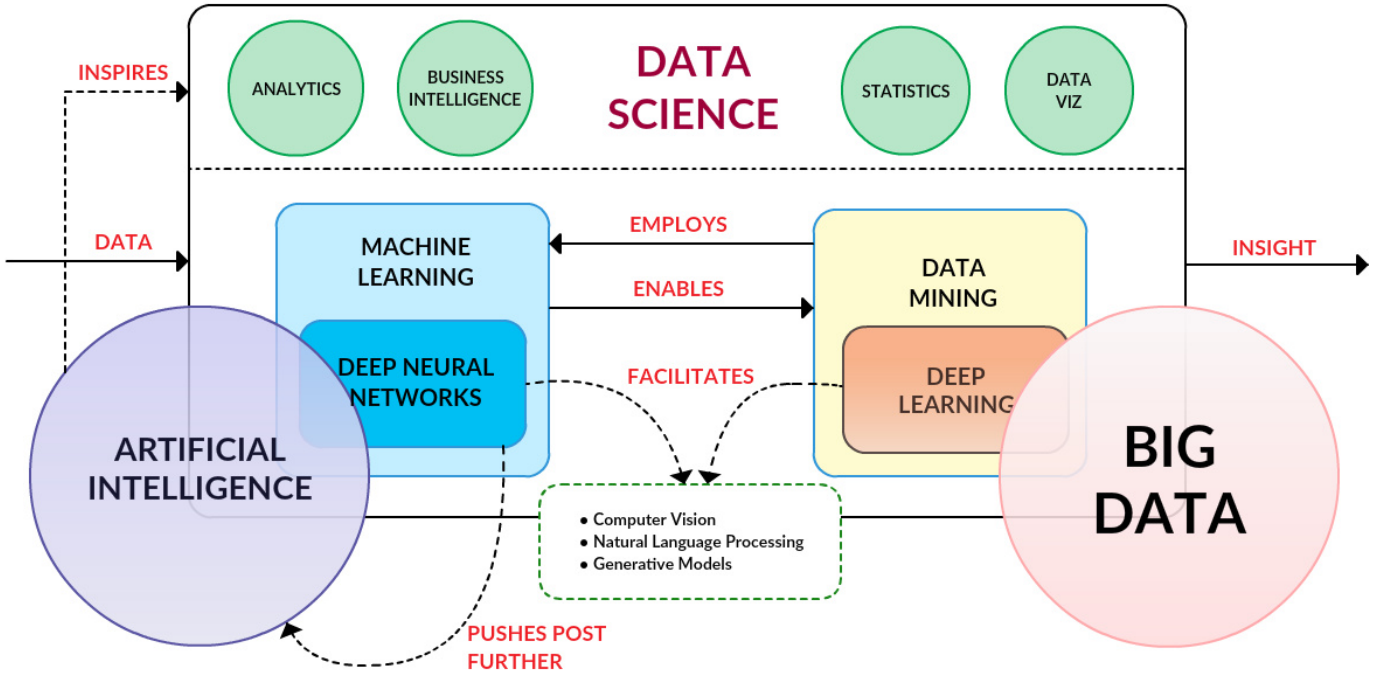
Why Data Science?



We are focusing on small-ish data

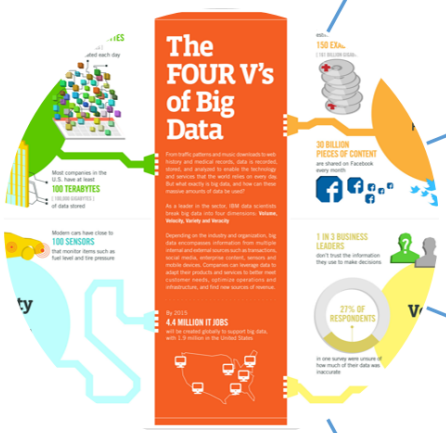
- This unit *proudly* focuses on small, in-memory datasets.
  - One can't tackle big data unless have experience with small data.
    - Small data (hundreds of Mb to 1-2Gb)
    - ~~Large data (10-100 Gb, Tb)~~ (`data.table`)
- Big data problem could be a small data problem in disguise
  - The data need to answer a specific question which might be small.
    - Find a subset, sub-sample, summary of the big data.
  - The big data problem might be decomposable into a large number of small data problems.
    - Use Hadoop or Spark to carry out embarrasingly parallel.
    - New tools such as `sparklyr`, `rhipe`, and `ddr` are needed.

Data Science, Machine Learning, AI?



Picture Credit:
{http://houseofbots.com/news-detail/2641-4-tutorial-foundations-of-machine-learning-and-data-science-for-developers}

Data Science, Machine Learning, AI?



Data Science

- Statistical Learning
- Machine Learning

Artificial Intelligence

- Symbolic Logic
- Connectionist

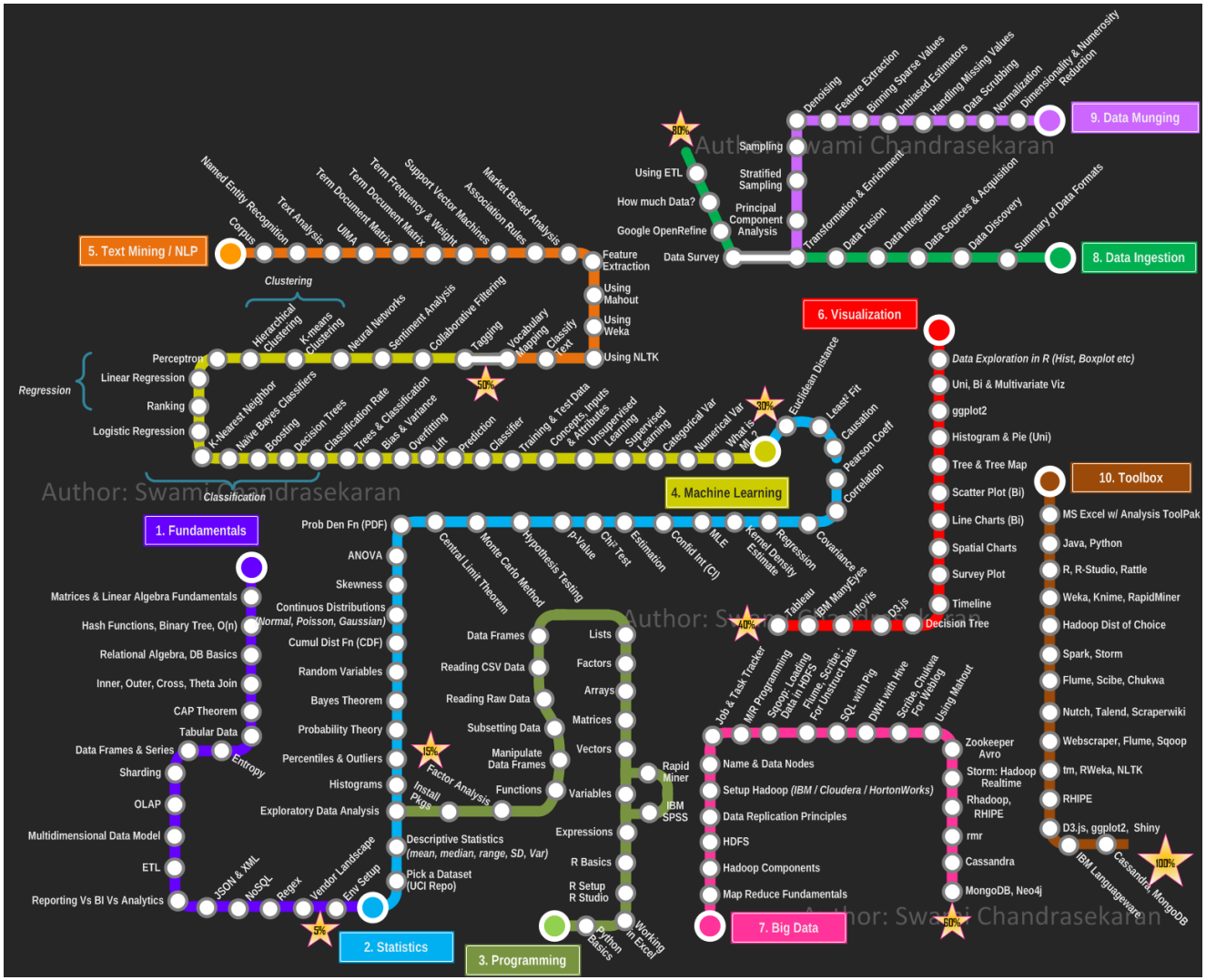
Data Mining

- Frequent Pattern Search
- Textual, Temporal, XML, Graph Data Mining

Pattern Recognition

- Video, images, speech

Data Science Curriculum Metro-Map



Picture Credit:
[http://nirvacana.com/thoughts/2013/07/08/becoming-a-data-scientist/]

Why R, not Python or Julia?

- R is a great place to start your data science journey
- R is designed from the ground up to support data science
- R is a programming language, but also an interactive environment for data science.
- R helps data scientists to focus on problems, supporting interaction between your brain and the computer.

Once you build fluency, learning other data science languages such as Python or Julia or Scala becomes easier.

In practice, most data science teams use a mix of languages, often at least R and Python.

We only deal with rectangular data

- Rectangular data are collections of values that are each associated with a variable and an observation
  - rows are observations
  - columns are variables

```
head(mpg)
```

```
## # A tibble: 6 x 11
##   manufacturer model displ  year   cyl trans      drv     cty   hwy fl
##   <chr>        <chr> <dbl> <int> <int> <chr>      <chr> <int> <int> <chr
## 1 audi         a4      1.8  1999     4 auto(l5)   f        18    29 p
## 2 audi         a4      1.8  1999     4 manual(m5) f        21    29 p
## 3 audi         a4      2    2008     4 manual(m6) f        20    31 p
## 4 audi         a4      2    2008     4 auto(av)   f        21    30 p
## 5 audi         a4      2.8  1999     6 auto(l5)   f        16    26 p
## 6 audi         a4      2.8  1999     6 manual(m5) f        18    26 p
```

- Non-rectangular data includes *images*, *sounds*, *trees*, *graphs*, *sequences*, *text*, *videos*, etc.

What's a Hypothesis

A hypothesis may be simply defined as a guess. A scientific hypothesis is an intelligent guess. – Isaac Asimov

Can you think of a hypothesis?

We focus on Hypothesis Generation

It's possible to divide data analysis into two camps:

- **Hypothesis Generation** (aka Data Exploration)
  - Look at data and armed with subject knowledge to generate interesting hypotheses to help explain why the data behaves the way it does.
  - Hypotheses are evaluated informally, using your skepticism to challenge the data in multiple ways.
- **Hypothesis Confirmation**. (aka Hypothesis Testing)
  - A precise mathematical model in order to generate falsifiable predictions.
  - You can only use an observation once to confirm a hypothesis.
  - You need to "preregister" (write out in advance) your analysis plan, and not deviate from it even when you have seen the data.

Hypothesis generation is a process beginning with an educated guess whereas hypothesis testing is a process to conclude that the educated guess is true/false or the relationship between the variables is statistically significant or not.

We focus on Hypothesis Generation

- Models are often used for exploration, and with a little care you can use visualisation for confirmation.
- The key difference is how often you look at each observation: only once, it's confirmation; more than once, it's exploration.
- Hypothesis generation helps in **comprehending the business problem** as we dive deep in inferring the various factors affecting our target variable
  - get a much better idea of what are the major factors that are responsible to solve the problem
  - guide data collection from various sources to convert business problem into a data science-based problem
  - improves domain knowledge if new to the domain when spending time understanding the problem
  - helps to approach the problem in a structured manner ?

References

- **R for Data Science (2e)**, *Hadley Wickham, Garrett Grolemund*, O'Reilly, 2nd Ed. (available online: https://r4ds.hadley.nz/) (Chapters 1)
- **Practical Data Science with R**, *Nina Zumel, John Mount*, Manning, 2nd Ed., 2020 (available in Barry J. Marshall Library, UWA: 006.312 2020 PRA) (Part 1, Chapter 1)

Data Science Life Cycle

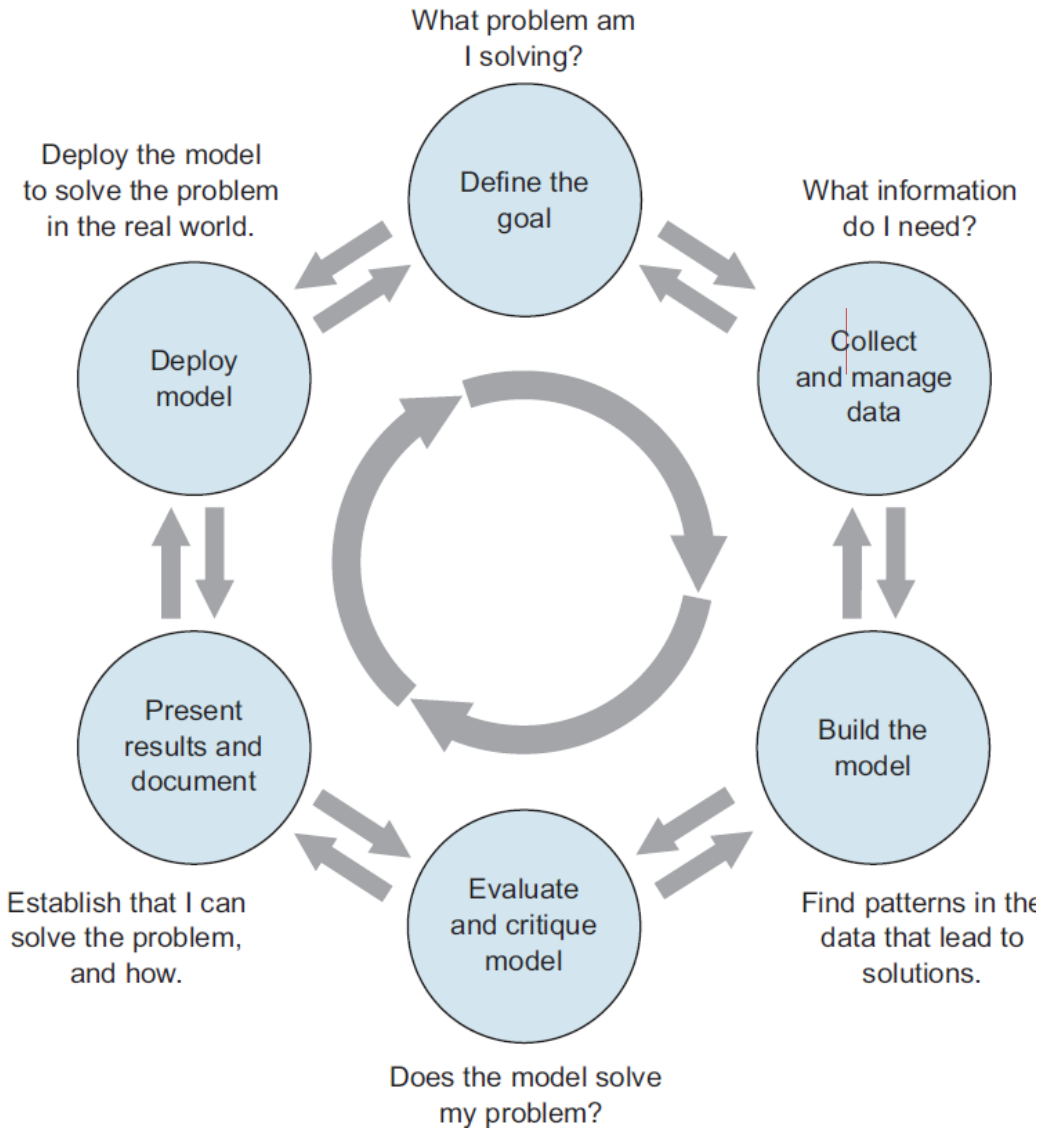
CITS4009 Computational Data Analysis

Dr. Mubashar Hassan

Department of Computer Science and Software Engineering
The University of Western Australia

Semester 2, 2024

The Data Science Life Cycle



1. Defining the goal

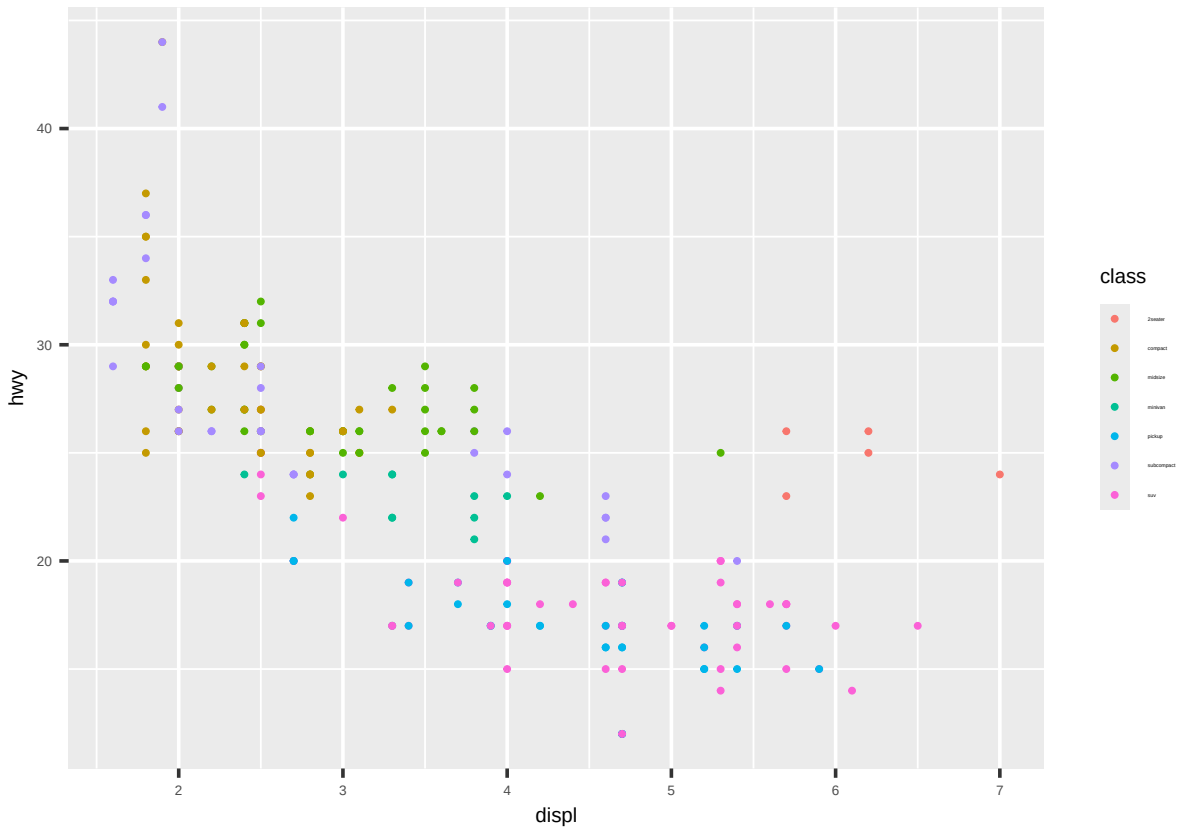
- Learn all that you can about the context of your project:
  - Why do the sponsors want the project in the first place? What do they lack, and what do they need?
  - What are they doing to solve the problem now, and why isn't that good enough?
  - What resources will you need? – what kind of data and how much staff? – Will you have domain experts to collaborate with, and – what are the computational resources?
  - How do the project sponsors plan to deploy your results? What are the constraints that have to be met for successful deployment?
- Once you have a good idea of the project's goals, you can focus on collecting data to meet those goals, or wrangling with existing data to meet those goals.

2. Identifying the data you need, exploring it, and conditioning it to be suitable for analysis.

- Most time consuming stage and most important:
  - What data is available to me?
  - Will it help me solve the problem?
  - Is it enough?
  - Is the data quality good enough?

Data Exploration - Visualisation

```
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy, color = class))
```



https://rpubs.com/shailesh/mpg-exploration

3. Modelling

- This stage is where you try to extract useful insights from the data in order to achieve your goals.
- There will be overlap and back and forth between the modelling stage and the data cleaning stage to find the best way to represent the data and the best form in which to model it.
- The most common data science modelling tasks are these:
  - **Classification** - Deciding if something belongs to one category or another.
  - **Scoring** - Predicting or estimating a numeric value, such as a price or probability
  - **Ranking** - Learning to order items by preferences
  - **Clustering** - Grouping items into most-similar groups
  - **Finding relations** - Finding correlations or potential causes of effects seen in the data
  - **Characterization** - Very general plotting and report generation from data

4. Model Evaluation and Critique

- Once you have a model, you need to determine if it meets your goals:
  - Is it accurate enough for your needs? Does it generalize well?
  - Does it perform better than "the obvious guess"? Better than whatever estimate you currently use?
  - Do the results of the model (coefficients, clusters, rules) make sense in the context of the problem domain?
- If you've answered "no" to any of these questions, it's time to loop back to the modelling step, or decide that the data doesn't support the goal you're trying to achieve.
- This might mean defining more realistic goals or gathering additional data or other resources that you need to achieve your original goal.

5. Presentation and Documentation

- After meeting the success criteria of the project, you need to present your results to your project sponsor and other stakeholders.
- Documents the model for those who need to use the model for running, using and maintaining the mode once it has been deployed.
- Document for Business-oriented audience - understand the impact of finding in terms of business metrics.
- Model Presentation for end-users should emphasize on how the model will help them do their job better.
- Model Presentation for operations staff should emphasize the impact of the model on the resources that they are responsible for.

6. Model Deployment and maintenance

- This stage comes after putting the model in the operations.
- Ensure the model run smoothly and won't make disastrous unsupervised decisions.
- Ensure model can be updated when its environment changes.

References

- **R for Data Science (2e)**, *Hadley Wickham, Garrett Grolemund*, O'Reilly, 2nd Ed. (available online: https://r4ds.hadley.nz/) (Chapters 1)
- **Practical Data Science with R**, *Nina Zumel, John Mount*, Manning, 2nd Ed., 2020 (available in Barry J. Marshall Library, UWA: 006.312 2020 PRA) (Part 1, Chapter 1)

Introduction to R

CITS4009 Computational Data Analysis

Dr. Mubashar Hassan

Department of Computer Science and Software Engineering
The University of Western Australia

Semester 2, 2024

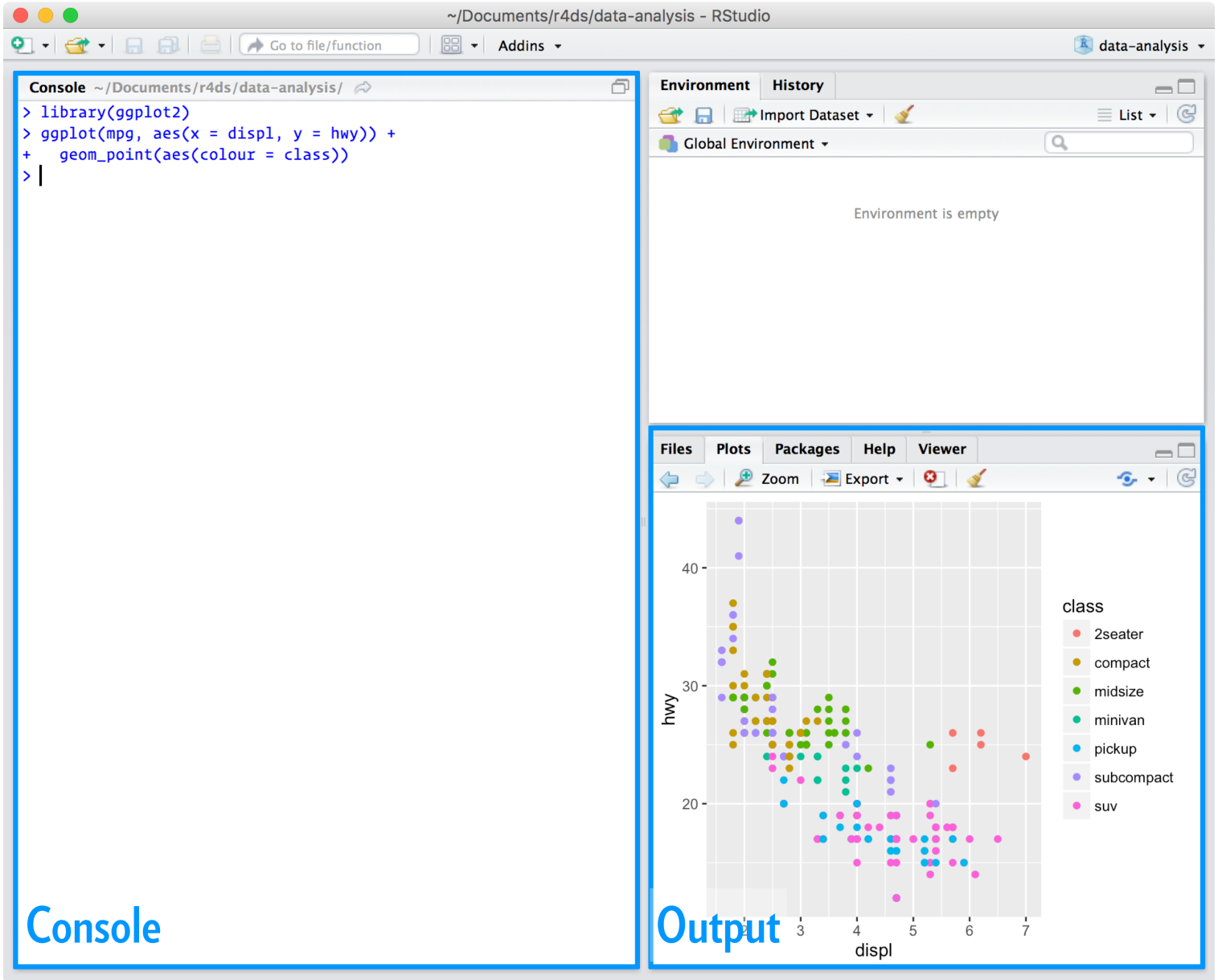
History of R

- R was originally written by Ross Ihaka and Robert Gentleman, at the University of Auckland. The project was conceived in 1992, with an initial version released in 1995 and a stable beta version in 2000.
- It is an implementation of the S language, which was principally developed by John Chambers.
- In 1998, the Association for Computing Machinery gave John Chambers its Software Award. He said:

"S has forever altered the way people analyze, visualize, and manipulate data . . . It is an elegant, widely accepted, and enduring software system, with conceptual integrity."

https://www.r-project.org/

RStudio the interactive environment



R working environment

File path is relative to working directory

```
getwd()
```

```
## [1] "C:/Users/00080148/OneDrive - UWA/CITS4009/lectures/01-
```

```
setwd("c:/R/CITS4009")
```

Install a package using `install.packages()`, e.g., to install the `tidyverse` package:

```
install.packages("tidyverse")
```

Load a package library with `library()`.

Help if know the function name

Details about a specific function:

```
help(data.frame)

## starting httpd help server ... done
? data.frame
```

Help if know the function name

See examples of a specific function:

```
example(mean)
```

```
##
## mean> x <- c(0:10, 50)
##
## mean> xm <- mean(x)
##
## mean> c(xm, mean(x, trim = 0.10))
## [1] 8.75 5.50
```

Help if know the function name

See demos of a package:

```
demo(graphics)
```

```
##
##
##  demo(graphics)
##  ---- ~~~~~~~~
##
## > #  Copyright (C) 1997-2009 The R Core Team
## >
## > require(datasets)
##
## > require(grDevices); require(graphics)
##
## > ## Here is some code which illustrates some of the differences
## > ## R and S graphics capabilities.  Note that colors are general
## > ## by a character string name (taken from the X11 rgb.txt file)
## > ## textures are given similarly.  The parameter "bg" sets the
```

Search for a function by keywords:

HTML search engine lets you search for topics with regular expressions. E.g., `help.search("csv$")` displays the links to the help pages of functions ended with `csv`.

Find commands containing a regular expression or object name:

```
apropos("var")
```

```
##  [1] ".rs.addGlobalVariable"                    ".rs.clearVar"
##  [3] ".rs.detectFreeVars_Call"                  ".rs.detectFreeVars
##  [5] ".rs.getCompletionsEnvironmentVariables" ".rs.getVar"
##  [7] ".rs.hasVar"                               ".rs.rpc.detect_fre
##  [9] ".rs.setVar"                               "all.vars"
## [11] "combine_vars"                             "estVar"
## [13] "get_all_vars"                             "globalVariables"
## [15] "pandoc_variable_arg"                      "var"
## [17] "var.test"                                 "variable.names"
## [19] "varimax"                                  "vars"
```

R as a calculator

```
log2(32)
## [1] 5
```

```
print(sqrt(2))
## [1] 1.414214
```

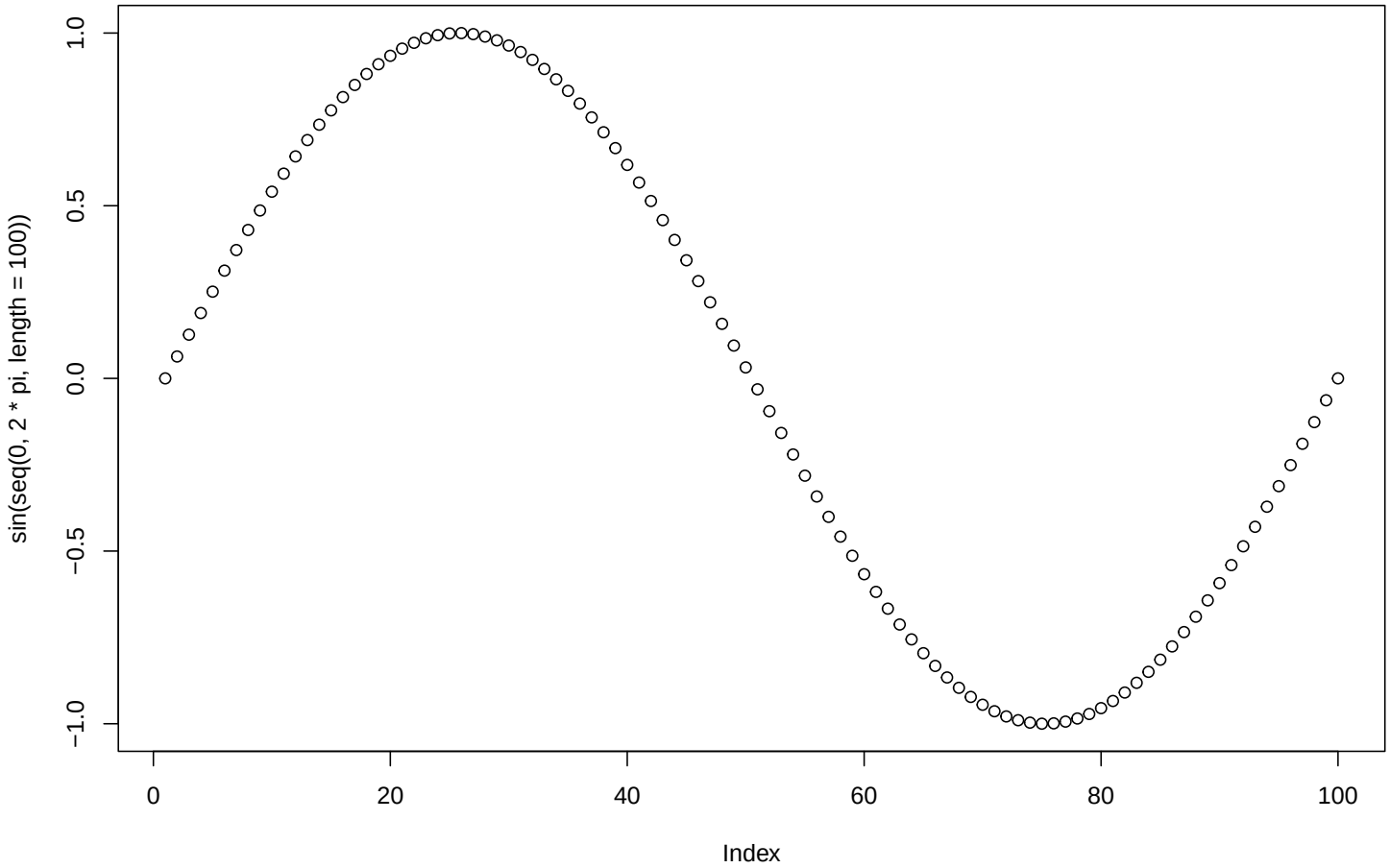
```
pi
## [1] 3.141593
```

```
seq(0, 5, length=6)
## [1] 0 1 2 3 4 5
```

```
1+1:10
##  [1]  2  3  4  5  6  7  8  9 10 11
```

R as a graphic tool

```
plot(sin(seq(0, 2*pi, length=100)))
```



Variables

Results of calculations can be stored in objects using the assignment operators:

- An arrow (<-) formed by a smaller than character and a hyphen without a space!
- The equal character (=).

```
a <- 1 + 2
```

These objects can then be used in other calculations. To print the object just enter the name of the object.

```
a
```

```
## [1] 3
```

Variable Name Restrictions

There are some restrictions when giving an object a name:

- Object names cannot contain 'strange' symbols like `!`, `+`, `-`, `#`.
- A dot (`.`) and an underscore (`_`) are allowed, also a name starting with a dot.
- Object names can contain a number but cannot start with a number.
- R is case sensitive, `X` and `x` are two different objects, as well as `temp` and `temP`.

Variable Types

Numeric

```
a <- 49  # Numeric
sqrt(a)
## [1] 7
```

```
b <- "The dog ate my homework" # Character String
sub("dog","cat",b)
## [1] "The cat ate my homework"
```

```
c <- (1+1==3) # Logical
c
## [1] FALSE
```

Vectors

In R a vector is an *ordered* collection of **data of the same type**

```
a <- c(1,2,3)
a*2
```

```
## [1] 2 4 6
```

A single number (i.e. a scalar) is the special case of a vector with 1 element.

Other vector types: character strings, logical

Matrices

matrix: rectangular table of data of the same type

```
mdat <- matrix(c(1,2,3, 11,12,13), nrow=2, ncol=3,
               byrow=TRUE, dimnames = list(c("row1", "row2"),
                                           c("C.1", "C.2", "C.3")))
mdat
```

```
##      C.1 C.2 C.3
## row1   1   2   3
## row2  11  12  13
```

```
mdat[,1:2]
```

```
##      C.1 C.2
## row1   1   2
## row2  11  12
```

Arrays

Array: higher-dimension matrix, turn a vector into multi-way dimensional matrix

```
array(1:3, c(2,4)) # recycle 1:3 "2 2/3 times"
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    1    3    2    1
## [2,]    2    1    3    2
```

Arrays (example)

Array: higher-dimension matrix, turn a vector into multi-way dimensional matrix

```
array(1:3, c(2,2,2))
```

```
## , , 1
##
##      [,1] [,2]
## [1,]    1    3
## [2,]    2    1
##
## , , 2
##
##      [,1] [,2]
## [1,]    2    1
## [2,]    3    2
```

Lists

List: *ordered* collection of **data of arbitrary types**.

```
doe <- list(name="john",age=28,married=F)
doe$name
```

```
## [1] "john"
```

Typically, vector elements are accessed by their index (an integer) and list elements by `$name` (a character string). But both types support both access methods. Slots are accessed by `@name`.

Data Frames

Data frame: rectangular table with rows and columns; - data within each column has the same type (e.g. number, text, logical), but - different columns may have different types. Represents the typical data table that researchers come up with - like a spreadsheet.

```
L3 <- LETTERS[1:3]
fac <- sample(L3, 5, replace = TRUE)
data.frame(x = 1, y = 1:5, fac = fac)
```

```
##   x y fac
## 1 1 1   C
## 2 1 2   C
## 3 1 3   A
## 4 1 4   B
## 5 1 5   B
```

Take home messages

- The data science life cycle
- Basic R syntax

References

- **R for Data Science (2e)**, *Hadley Wickham, Garrett Grolemund*, O'Reilly, 2nd Ed. (available online: https://r4ds.hadley.nz/) (Chapters 1)
- **Practical Data Science with R**, *Nina Zumel, John Mount*, Manning, 2nd Ed., 2020 (available in Barry J. Marshall Library, UWA: 006.312 2020 PRA) (Part 1, Chapter 1)
- **Introduction to the R language:** https://users.soe.ucsc.edu/~lshiue/bioc/Rintro.ppt
- **An Introduction to R:** http://csg.sph.umich.edu/abecasis/class/815.04.pdf
- (optional) **Wikipedia:** https://en.wikipedia.org/wiki/R_(programming_language)