LECTURER: DR. TAI LE QUY

ANALYTICAL SOFTWARE AND FRAMEWORKS

TOPIC OUTLINE

UNIT 3

STATISTICAL MODELING FRAMEWORKS

STUDY GOALS

- Understand the importance of statistical modeling frameworks
- Get familiar with R and Python ecosystems
- Get an overview of control structures, common packages, and their capabilities in statistical modeling in R and Python
- Compare advantages and limitations of R and Python

EXPLAIN SIMPLY

1. Why are statistical libraries and frameworks in R and Python popular and important?
2. How do solutions developed with both, R and Python, compare with regards to their capabilities and limitations?
3. What is meant by a “normal distribution”?

STATISTICAL SOFTWARE AND PROGRAMMING LANGUAGES

Today, data scientists may choose from a **huge variety of** commercial or free **programming environments and software applications for solving data-science challenges**.

Three categories: **command-line** (CLI)-**based**, **graphical user interface (GUI)-based**, and **hybrid**.

GUIs don't require upfront knowledge of commands and programming languages but **lack reproducibility**.

CLIs require knowledge of involved syntax and commands but **provide high reproducible solutions**.

Source of the text: Course book DLMBDSA02, p. 50.

THE R PROJECT FOR STATISTICAL COMPUTING

- R is an advanced **free** software environment programming for **statistical computing and graphics**.
- Supporting high-quality
  - **data manipulation,**
  - **visualization** and
  - **graphics tools**.
- Strong user **community** and extensive **documentation**
- **RStudio**: preferred IDE with GUI
- **CRAN** with numerous packages to extend core language (https://cran.r-project.org/)

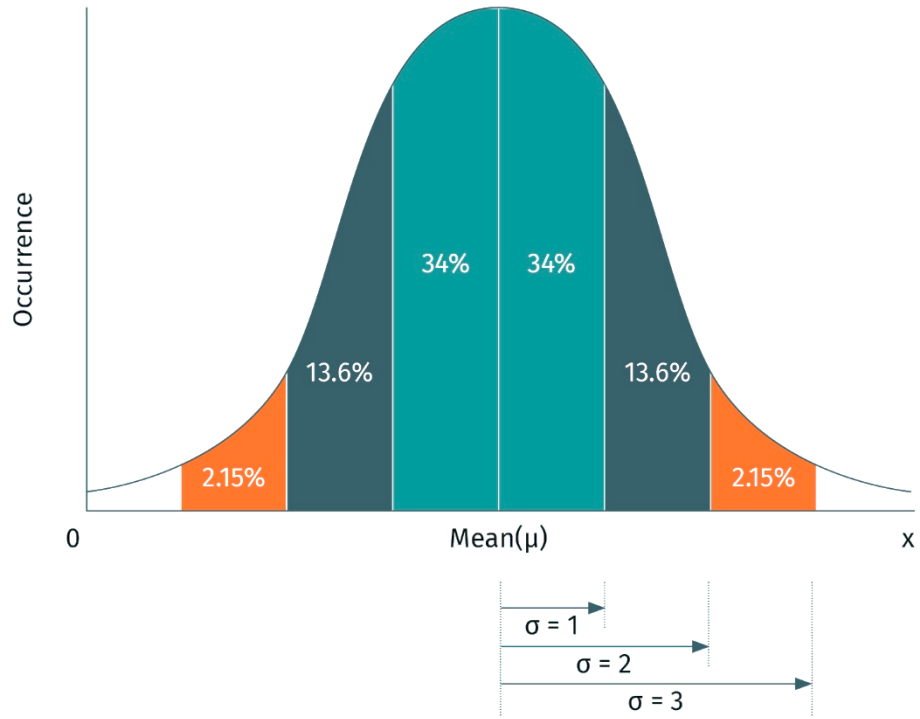
Source of the text: Course book DLMBDSA02, pp. 50—53; The R Foundation, 2021.

THE R PROJECT FOR STATISTICAL COMPUTING
FREQUENTLY USED PACKAGES FOR DATA SCIENCE

Package Name	Usage
stats	Statistical functions, distributions
Dplyr	Data manipulation
ggplot2, esquisse	Data visualization
BioConductor	Biological data
Shiny	Interactive web apps for data/viz sharing
Lubridate, Dt	Date-time data wrangling
Knitr	Dynamic report generation
mlr, mlr3, tensorflow	Machine learning
Quanteda	Text analysis
Dt	Interface to JavaScript DataTables library
Caret	Supervised learning
Janitor	Data cleaning

Source of the table: Course book DLMBDSA02, pp. 58f., 65.

The Normal Distribution



Source of the image: Course book DLMBDSA02, p. 66.

THE R PROJECT FOR STATISTICAL COMPUTING
OBJECTS, FUNCTIONS, CONTROL STRUCTURES

R is a **functional programming language** with support for **object-oriented programming** (OOP).

Functions define a combination of expressions to accomplish a task. They may accept **arguments**. Besides building own functions, **many operations are provided through built-in functions of packages**.

R provides classes of **objects**, including character, numeric, integer, complex, logical, vector, functions, etc. that can have **attributes** (metadata for the objects to describe them).

Source of the text: Adler, J., 2012; Peng, R. D., 2020; Course book DLMBDSA02, pp.56—59.

- When writing own programs or functions, **control structures** are useful to **control the flow** of execution of R expressions.
- if-else: testing a **condition** and act on it (using logical operators <, >, ==, !=, <=, >=, x&y, x|y, !x)
- for: execute a **loop** for a fixed number of times (very useful)
- Further loop constructions with **while**, **repeat**, **break**, **next**

Source of the text: Peng, R. D., 2020; Course book DLMBDSA02, p. 54f.

THE R PROJECT FOR STATISTICAL COMPUTING
CODE EXAMPLE

```r
# Note: c() is a generic function which combines its
#       arguments in a vector or a list.
x = c(1, 2, 3, 4, 5)

MeanVal = function(x) {
  return(sum(x)/length(x))
}

MeanVal(x)

# [1] 3
```

Code example derived from: Python Software Foundation, 2021.

*"Python is a **programming language** that lets you **work quickly** and **integrate systems more effectively**."*

- Free general/**multi-purpose programming language**
- **Multi-paradigm, dynamically typed, interpreted**
- **Used in in many fields of CS and SE**, e.g. Data Science, web and API development, system automation (Infrastructure as Code)
- Widely acknowledged as **easy to learn**
- Most **popular programming language** in 2021 according to PYPL

Source of the text: Python Software Foundation, 2021; Carbonnelle, P., 2021.

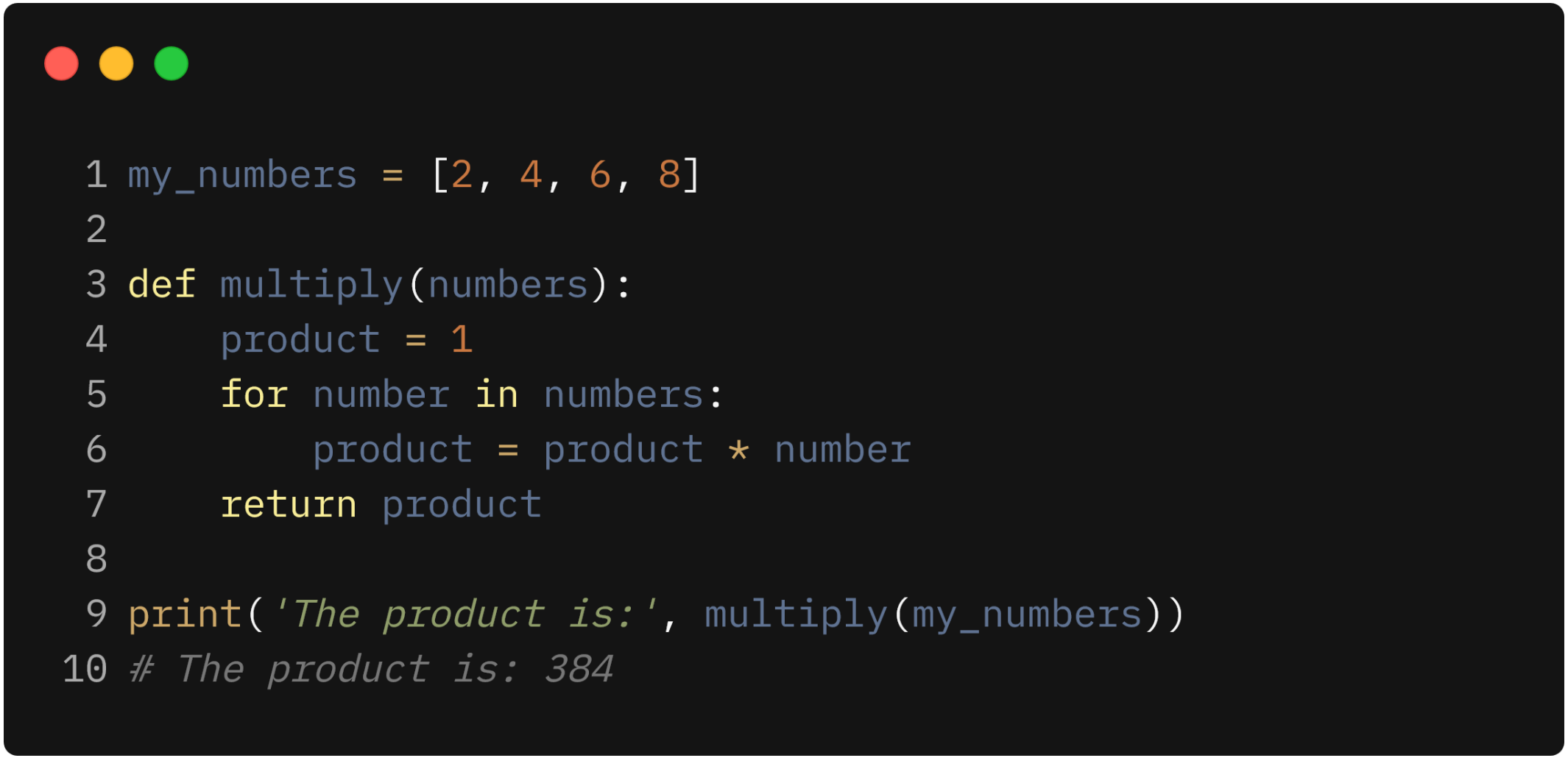
- **Quick development** due to **friendly syntax**, **built-in modules**, and **many provided libraries and frameworks**
- **Easy to integrate and to deploy at large scale**, e.g., to provide an API to a developed AI model to support business processes. No need to switch programming language/ecosystem
- Comprehensive **coverage of needs in Data Analytics and AI**
- Not built (only) for statisticians; **libraries and frameworks provide statistical and analytical features**
- Multiple high-quality IDEs are available

Source of the text: Python Software Foundation, 2021; IBM Cloud Team, 2021.

Libraries and Frameworks for Data Science	Usage
NumPy, SciPy	Scientific and technical computing
Pandas	Data manipulation and aggregation
Statsmodels	Statistical analysis
Bokeh, MatplotLib, Seaborn, Plotly	Data visualizations
Scikit-Learn, Tensorflow, Keras	Machine learning

Source of the table: Course book DLMBDSA02, p. 73.

THE PYTHON PROGRAMMING LANGUAGE
CODE EXAMPLE



```
my_numbers = [2, 4, 6, 8]

def multiply(numbers):
    product = 1
    for number in numbers:
        product = product * number
    return product

print('The product is:', multiply(my_numbers))
# The product is: 384
```

Code example derived from: Python Software Foundation, 2021.

Python offers several control structures to control the flow of the script/program.

- **if-elif-else**: testing a **condition** and acting on it (using logical operators <, >, ==, !=, <=, >=, x and y, x or y, x is y, x is not y), may be nested
- **try-except**: to **catch exceptions and avoid errors** stopping the script or application
- Multiple ways to construct **loops** and **to iterate** over collections with for, while, break, continue

Source of the text: Course book DLMBDSA02, pp. 69—80.

COMPARISON R VS. PYTHON 1/2

	R	Python
Purpose	Used **mainly** for **statistical modeling**, ideal for **data exploration** and **visualizing** your data in beautiful graphics	Used for a **variety of purposes** like web application development, data analysis, and AI
Used by	Used by statisticians and data scientists, even without programming skills	Used by developers, data engineers, and data scientists in wide range of industry, research, and engineering workflows
Suitable for	Suitable for those with no prior experience in programming	Suitable for newbies as well as experienced IT professionals
Integration	Mainly used when the data analysis tasks require standalone computing on individual computers; can be integrated via APIs to other languages to model workflows	Mainly used when the data analysis tasks need to be **integrated** with web apps **without leaving the ecosystem** (production-ready language)

Source of the table: Course book DLMBDSA02, p. 68; IBM Cloud Team, 2021; DataCamp Team, 2020.

COMPARISON R VS. PYTHON 2/2

	R	Python
Applied in	Best-known application fields: banking, finance, e-commerce, social media, and healthcare	Best-known application fields: web and internet development, business, games and 3D graphics, network programming, database access, desktop interfaces
Styleguide and Idioms	Same piece of functionality written in several **inconsistent** ways by several 3rd parties providing libraries	Any piece of functionality preferred written according to "**Pythonic" idioms** and following the **style guide** what leads to **high readability**
Is easy at	Using available libraries	Using available libraries and construction new models
Learning curve	**Considered more challenging** (for developers coming from other languages); may be easier for people with math and statistics background	**Widely considered linear and smooth**, even for novices learning their first programming language

Source of the table: Course book DLMBDSA02, p. 68; IBM Cloud Team, 2021; DataCamp Team, 2020.

PYTHON OR R — WHICH ONE TO CHOOSE? ASK SOME QUESTIONS…

- Do you have programming experience?
- What do(es) your colleagues/team/organization use preferred?
- What problems are you trying to solve?
- How important are charts and graphs?
- Do you have to integrate models in an engineering environment?

Source of the text: IBM Cloud Team, 2021; DataCamp Team, 2020.

REVIEW STUDY GOALS

- Understand the importance of statistical modeling frameworks
- Get familiar with R and Python ecosystems
- Get an overview of control structures, common packages and their capabilities in statistical modeling in R and Python
- Compare advantages and limitations of R and Python

SESSION 3

TRANSFER TASK

TRANSFER TASK

Imagine that a friend of yours is writing her thesis. She used a comprehensive questionnaire to evaluate the success of her approaches qualitatively. She'll need a thorough statistical analysis of the questionnaire's results, including hypothesis testing and high-quality visualizations. Results will come in gradually, new target groups for the survey will be acquired. The analyses will have to be carried out repeatedly. She heard of the popularity of the free R and Python ecosystems and asks you which one to choose. Justify your decision!

TRANSFER TASK

Open the provided Rstudio/Jupyter notebooks and work through the examples to provide descriptive statistics and visualizations for the provided datasets.

Please let me know if you got stuck so we can work through it together.

TRANSFER TASK
PRESENTATION OF THE RESULTS

Please present your results.

The results will be discussed in plenary.

1. One advantage of implementing **R** in data analysis is that ...

   a) it has a quick learning curve.
   b) it can be integrated into web applications.
   c) it is a free open-source language.
   d) it is quick in dealing with very big datasets.

2. A data manipulation package in **R** is called ...

a) janitor.
b) dplyr.
c) quanteda.
d) knitr.

3. The **`describe()`** function called on a **`DataFrame`** in Python's **`pandas`** library is used to ...

   a) generate descriptive statistics.
   b) show the data types.
   c) show the number of columns and rows of the dataset.
   d) return TRUE if there are no missing values.

LIST OF SOURCES

Adler, J. (2012). *R in a nutshell: A desktop quick reference* (2nd ed.). O'Reilly.

Carbonnelle, P. (2021). *PYPL PopularitY of Programming Language.* https://pypl.github.io/PYPL.html

Costa, C.D. (2020a). *Top Programming Languages for Data Science in 2020.* https://towardsdatascience.com/top-programming-languages-for-data-science-in-2020-3425d756e2a7

Costa, C.D. (2020b). *Top Programming Languages for AI Engineers in 2021.* https://towardsdatascience.com/top-programming-languages-for-ai-engineers-in-2020-33a9f16a80b0

DataCamp Team (2020). *Choosing Python or R for Data Analysis? An Infographic.* https://www.datacamp.com/community/tutorials/r-or-python-for-data-analysis

IBM Cloud Team (2021). *Python vs. R: What is the difference?* https://www.ibm.com/cloud/blog/python-vs-r

Muenchen, R.A. (2019). *The Popularity of Data Science Software.* http://r4stats.com/articles/popularity/

Statistics and Data (2021). *The Most Popular Programming Languages — 1965/2021 — New Update.* https://statisticsanddata.org/data/the-most-popular-programming-languages-1965-2021/

The R Foundation (2021). *The R Project for Statistical Computing (2021).* https://www.r-project.org/

Peng, R.D. (2020). *R Programming for Data Science.* https://bookdown.org/rdpeng/rprogdatascience/

Python Software Foundation (2021). *Welcome to Python.org.* https://www.python.org/