CONSTRUCTOR UNIVERSITY

To R or to Python? Is this the question?

Adalbert F.X. Wilhelm

German Data Science Days, München
March 07, 2024

constructor.university

Content

- General Overview on R and Python
- Classical criteria
- Data Science criteria
- Educational aspects
- Conclusion/Open Questions

- programming language and environment
- initiated by Ross Ihaka and Robert Gentleman
- inspired by S programming language (“different implementation of S”) and Scheme
- GNU-project, multi-platform
- open-source environment widely used for statistical computing and graphics
- integrated suite of software facilities for data manipulation, calculation and graphical display
- “environment” = fully planned and coherent system
- for computationally intensive tasks, C, C++, and Fortran code can be linked and called at run-time
- R has its own LaTeX-like documentation format
- R distribution comes with 14 packages, more than 20k available from CRAN

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

- designed by Guido van Rossum
- high-level general purpose programming language
- multi-paradigm:
  - object-oriented
  - procedural (imperative)
  - functional
  - structured
  - reflective
- emphasising code readability
- multi-platform, open-source
- more than 500 k projects on Python Package Index

https://www.python.org/about/
https://pypi.org/

Popularity

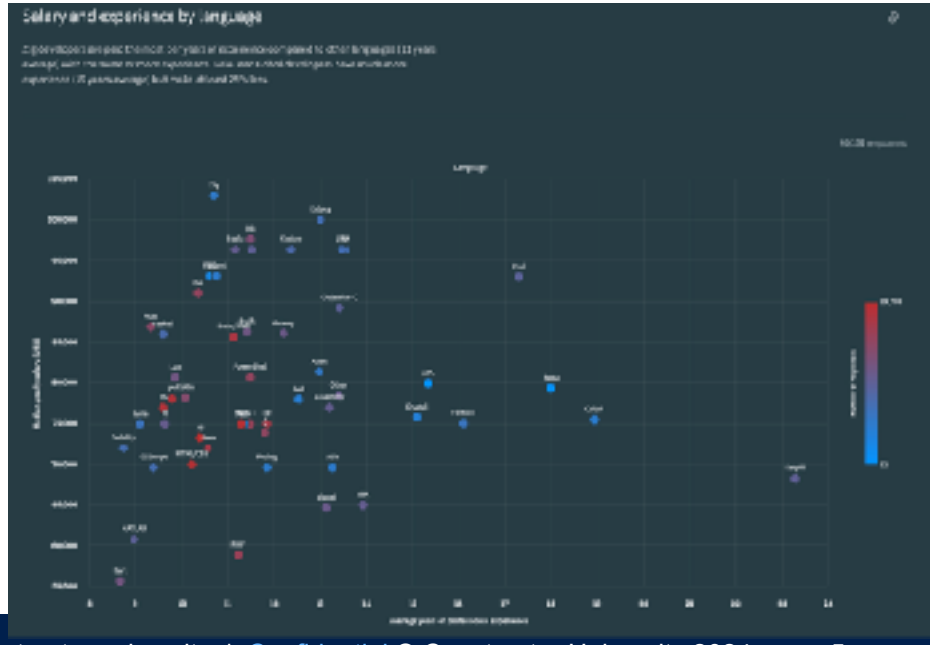
Worldwide, Mar 2024 :

Rank	Change	Language	Share	1-year trend
1		Python	28.62 %	+1.6 %
2		Java	15.79 %	-0.5 %
3		JavaScript	9.7 %	-0.6 %
4		C#	6.77 %	-0.0 %
5		C/C++	6.76 %	0.0 %
6	↑	R	4.71 %	+0.5 %
7	↓	PHP	4.5 %	-0.7 %
8		TypeScript	2.85 %	+0.1 %
9		Swift	2.74 %	+0.6 %
10		Objective-C	2.4 %	+0.1 %
11		Rust	2.36 %	+0.3 %
12	↑	Go	2.09 %	+0.2 %
13	↓	Kotlin	1.9 %	+0.0 %
14		MATLAB	1.64 %	-0.1 %
15		Ruby	1.37 %	-0.0 %
16	↑↑↑	Dart	0.97 %	+0.2 %
17		Ada	0.94 %	0.0 %
18		PowerShell	0.9 %	+0.0 %
19	↓↓↓	VBA	0.87 %	-0.1 %
20	↑↑	Lua	0.69 %	+0.1 %

TIOBE

Feb 2024	Feb 2023	Change	Programming Language	Ratings	Change
1	1		Python	15.16%	-0.32%
2	2		C	10.97%	-4.41%
3	3		C++	10.53%	-3.40%
4	4		Java	8.85%	-4.33%
5	5		C#	7.05%	+1.15%
6	7	↑	JavaScript	3.17%	+0.84%
7	6	↓	SQL	1.48%	-0.30%
8	11	↑	Go	1.35%	+0.61%
9	8	↓	Visual Basic	1.8%	-0.82%
10	10		PHP	1.5%	+0.21%

21	R	0.99%
22	SAS	0.95%
23	Classic Visual Basic	0.82%
24	Prolog	0.77%
25	Ada	0.77%



- PYPL popularity index of programming languages
- https://pypl.github.io/PYPL.html (raw data from Google Trends)
- TIOBE index
- https://www.tiobe.com/tiobe-index/
- Stackoverflow
- https://survey.stackoverflow.co/2023/#work-employment

Syntax and Readability

```
# Python example
def calculate_average(numbers):
    total = sum(numbers)
    count = len(numbers)
    average = total / count
    return average

data = [25, 30, 35, 40, 45]
result = calculate_average(data)
print("The average is:", result)
```

```
# R example
calculate_average <- function(numbers) {
  total <- sum(numbers)
  count <- length(numbers)
  average <- total / count
  return(average)
}

data <- c(25, 30, 35, 40, 45)
result <- calculate_average(data)
print(paste("The average is:", result))
```

- Python uses indentation to define code blocks, while R uses curly braces `{}`.
- Python has a more straightforward and readable syntax, with an emphasis on code readability and simplicity.
- Python uses the `.` notation for method and attribute access, and the `[ ]` notation for indexing and slicing.

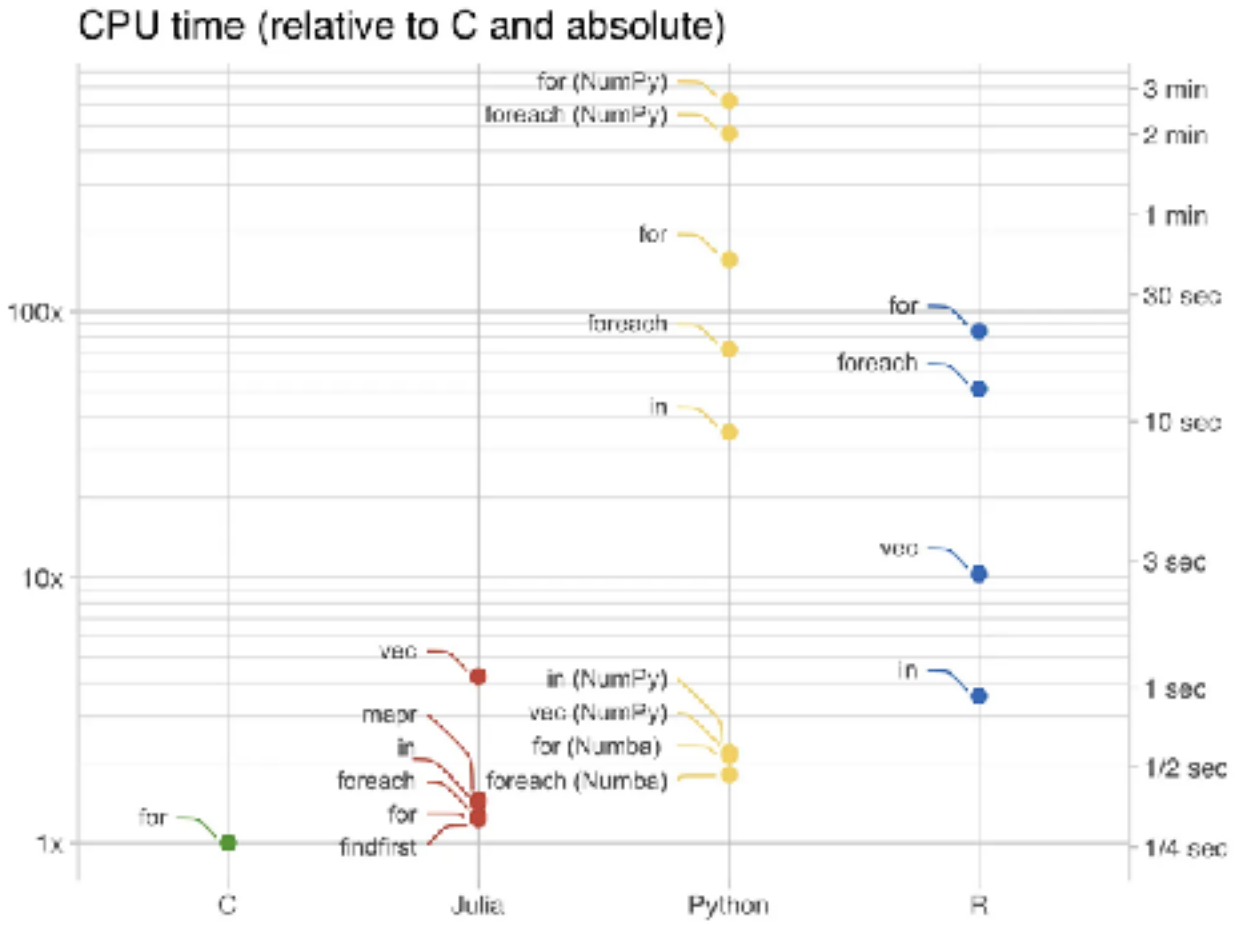
- Traditional R coders use the `<-` operator for assignment, while Python uses the = operator.
- R uses the `function_name <- function(arguments) { ... }` syntax for function definition, while Python uses `def function_name(arguments):`.
- R often uses the `print()` function to display output, while Python uses the `print` statement.
- R uses the `$` operator to access elements within a data frame, while Python uses the `.` notation for object attributes or the `[ ]` notation for dictionary and list elements.

Speed

- Membership testing on an unsorted vector of integers

https://towardsdatascience.com/r-vs-python-vs-julia-90456a2bcbab

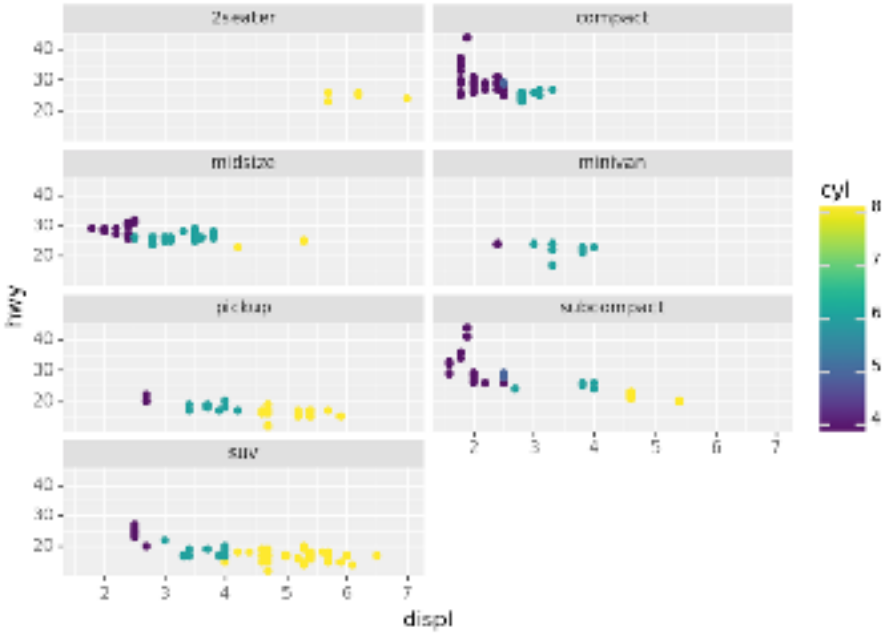
```r
# 0.93 seconds
in_search <- function(vec, x) x %in% vec

# 2.68 seconds
vec_search <- function(vec, x) any(x == vec)

# 13.33 seconds
foreach_search <- function(vec, x) {
  for (v in vec)
    if (v == x)
      return (TRUE)
  FALSE
}

# 21.94 seconds
for_search <- function(vec, x) {
  for (i in 1:length(vec))
    if (vec[i] == x)
      return (TRUE)
  FALSE
}
```



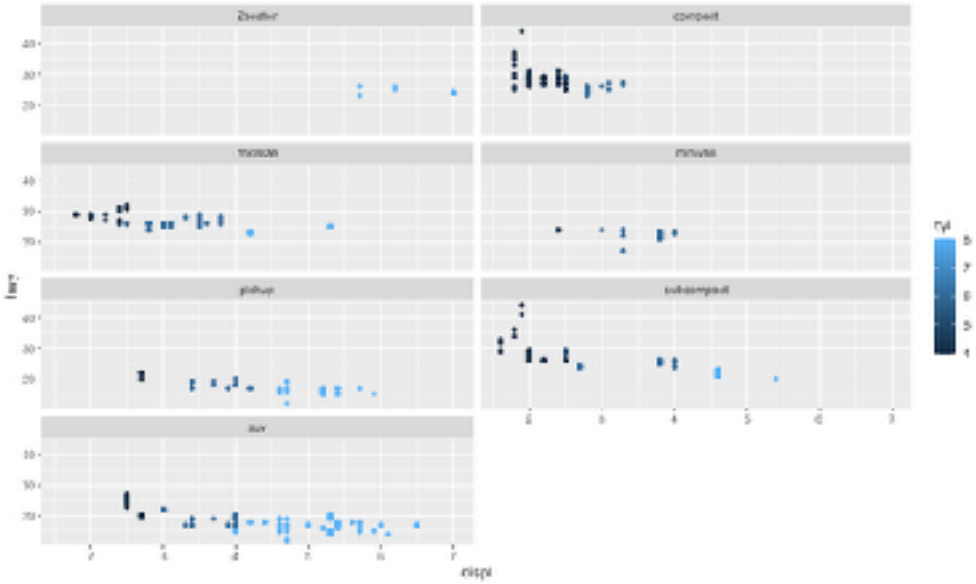
For evaluating different implementations in R, Python, and Julia, a dataset with 1.000.000 unique integers ranging from 1 to 2.000.000 was generated and 1.000 searches with all integers from 1 to 1.000 were performed.
The probability of a search being successful is ~50%, so half the times the algorithm will scan the complete vector to conclude that the search was unsuccessful.
In the remaining cases, the algorithm should require *(n+1)/2* evaluations (on average) to find the element, with *n* being the length of the vector.

Data Visualisation

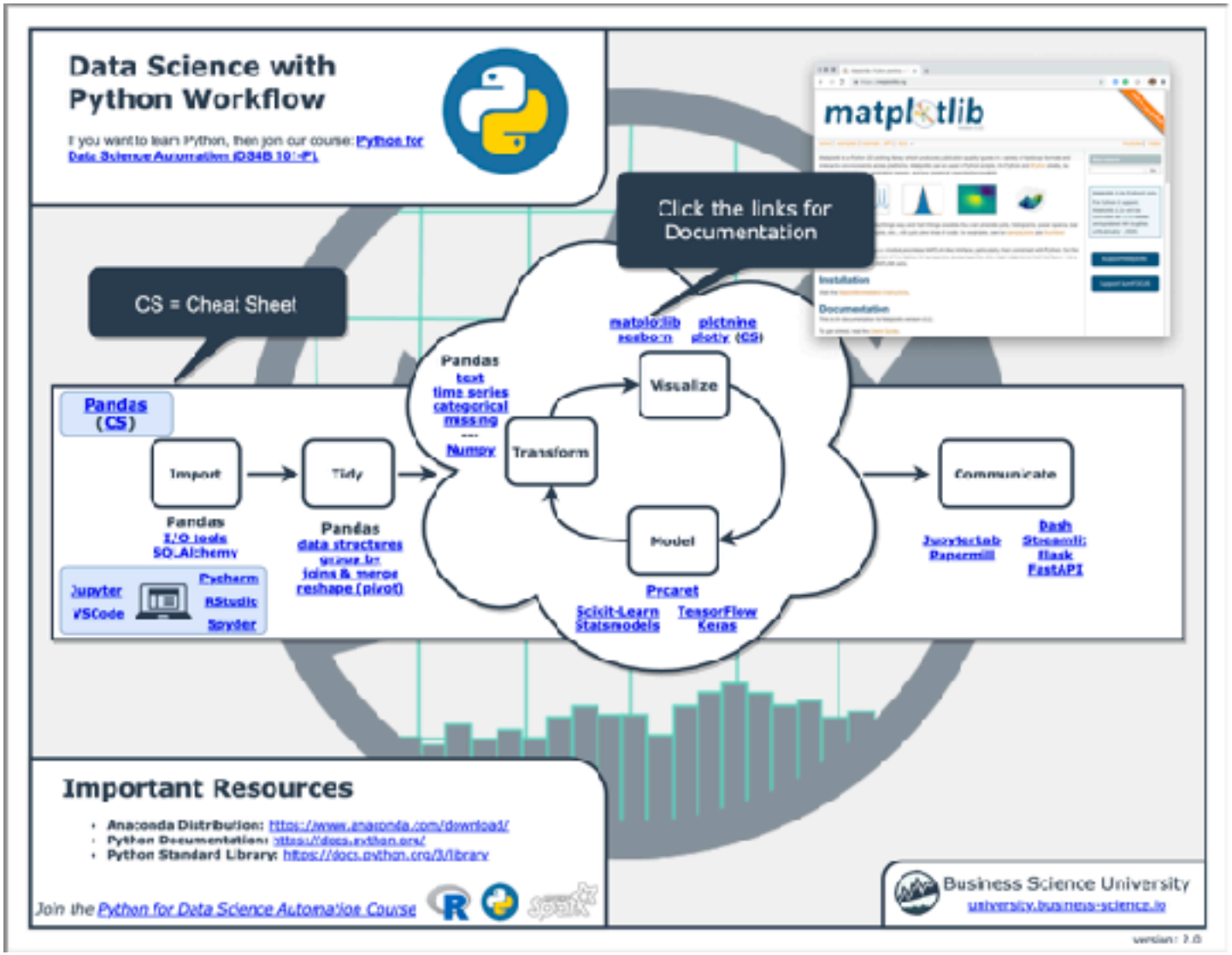
```python
# Python example using plotnine
from plotnine import *
from plotnine.data import mpg

# Create a facet plot using facet_wrap
(
    ggplot(mpg, aes(x='displ', y='hwy',
colour = 'cyl')) +
    geom_point() +
    facet_wrap('~class', ncol=2)
)
```



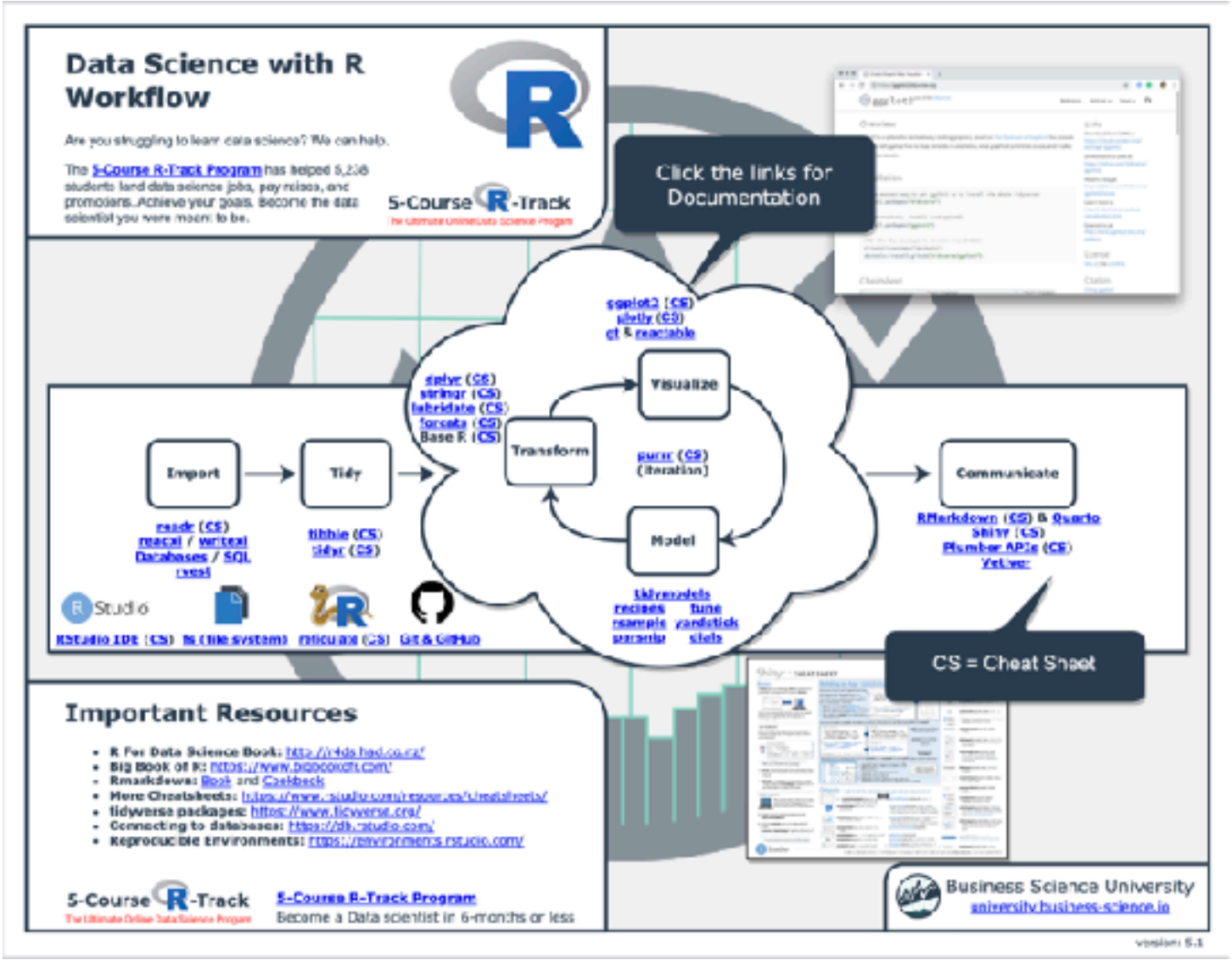

```r
# R example using ggplot2
library(ggplot2)
# Create a facet plot using facet_wrap
ggplot(mpg, aes(x = displ, y = hwy, colour=cyl)) +
geom_point() +
facet_wrap(~ class, ncol = 2)
```



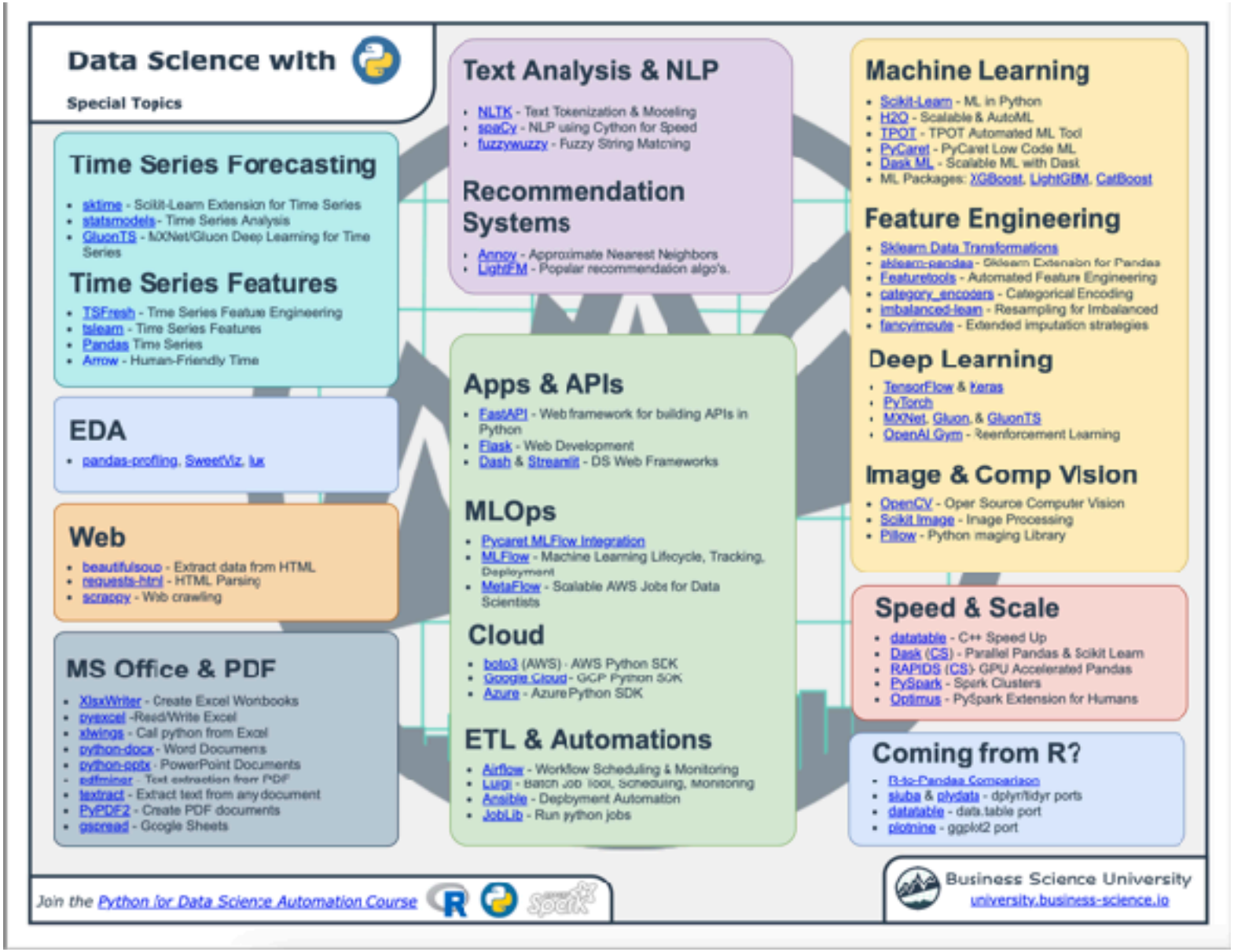
Data Science Workflow with Python



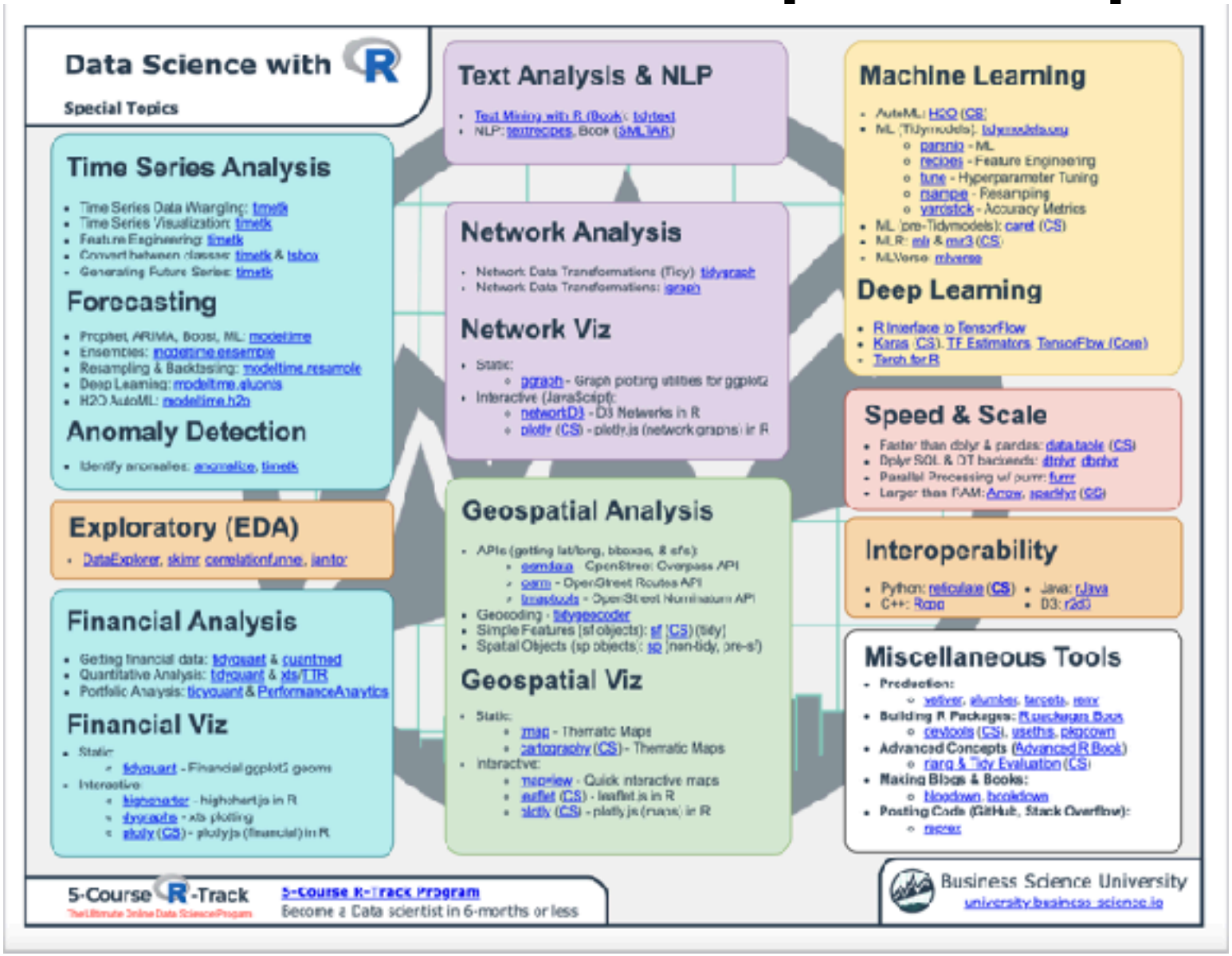
Data Science Workflow with R



Data Science with Python – Special Topics



Data Science with R – Special Topics

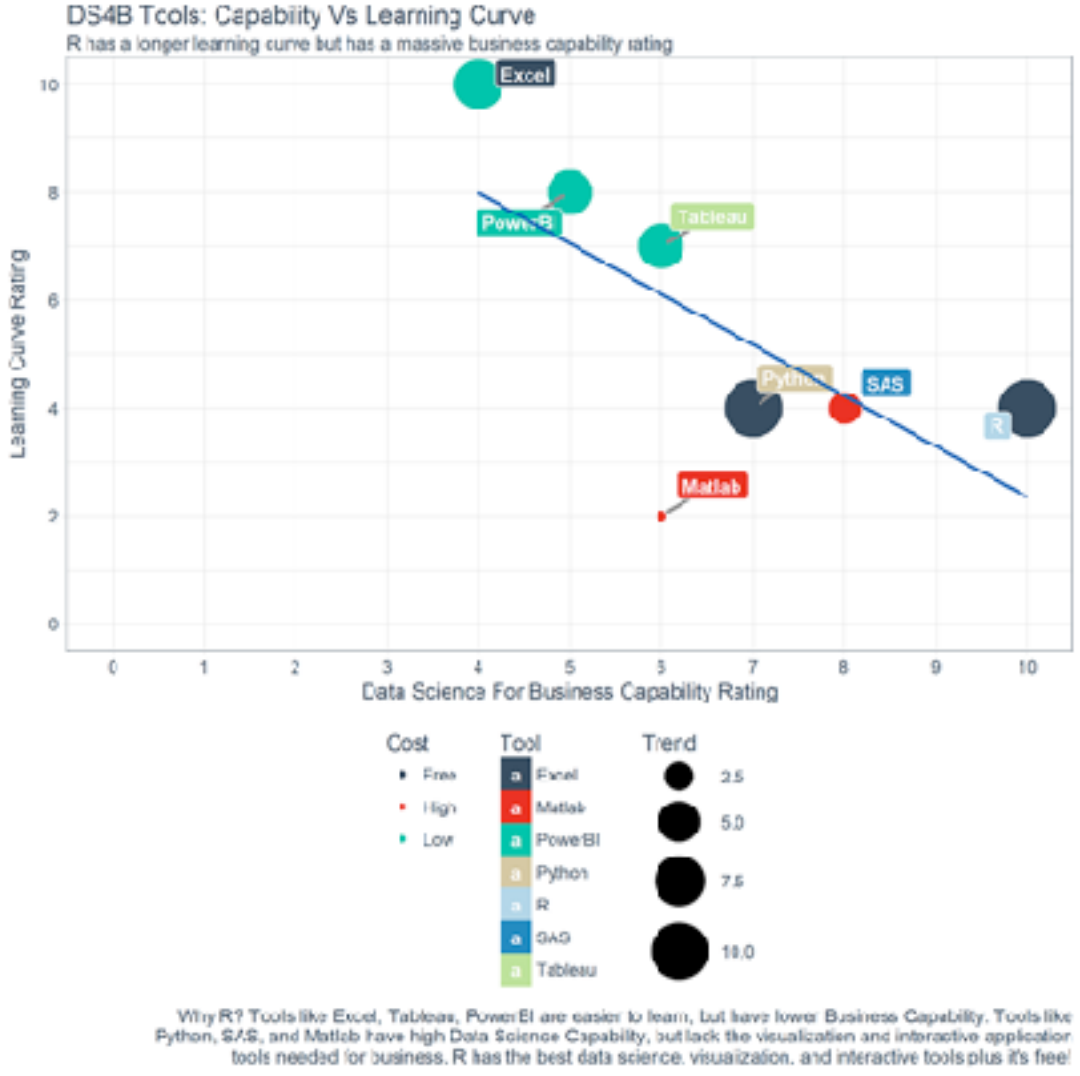


Industry Adoption

Consider your specific data science specialization when choosing between R and Python:

- For general-purpose data science, machine learning, and AI applications, Python's versatility makes it a compelling choice.
- If your focus is primarily on traditional statistical analysis, hypothesis testing, and specialized graphical techniques, R offers a rich environment for these tasks.
- In academia and research, R remains a staple for statistical research and data analysis, while Python is increasingly gaining ground, especially in machine learning research.
- In industry, Python's widespread adoption and comprehensive ecosystem give it a competitive edge, especially in sectors requiring scalable data solutions and AI integration.

https://iabac.org/blog/r-vs-python-for-data-science-a-friendly-comparison

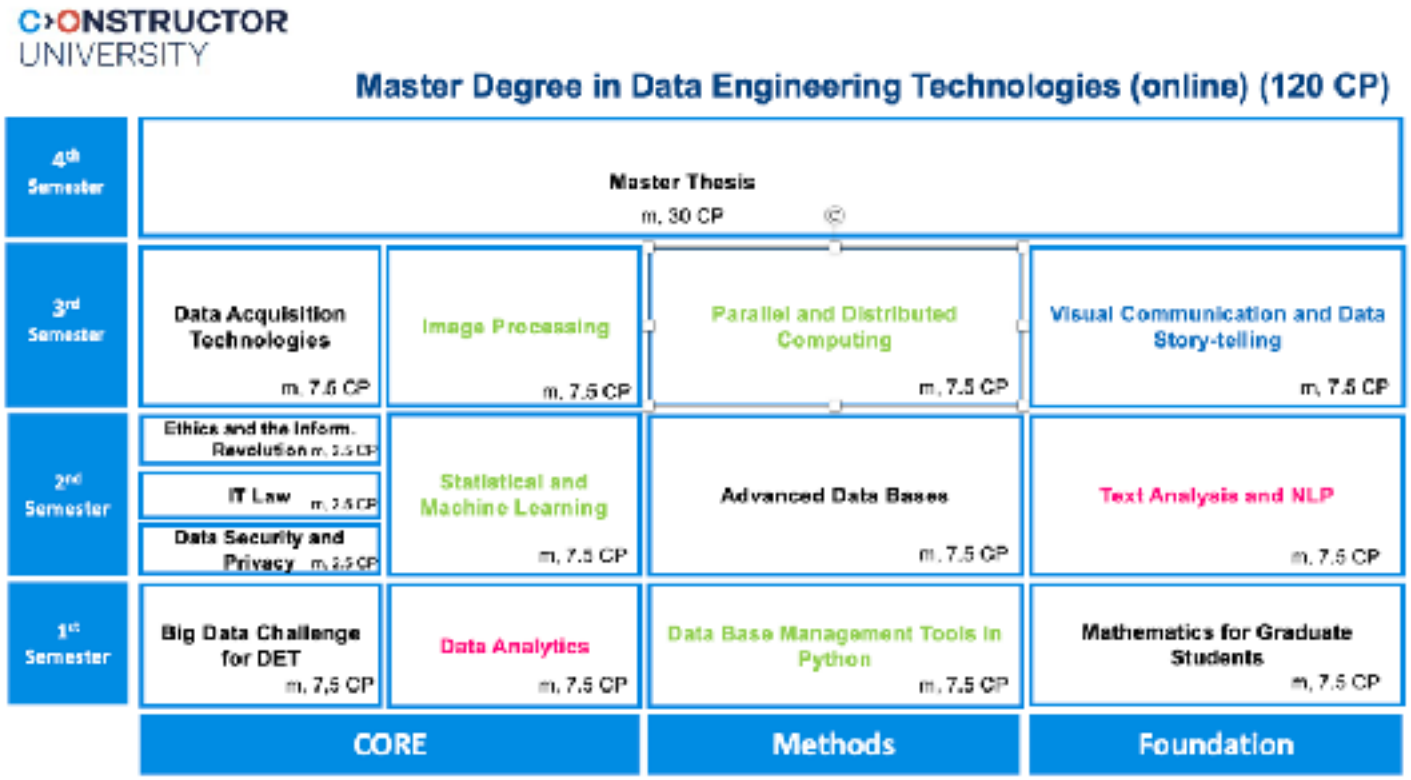


https://www.business-science.io/business/2020/12/17/six-reasons-to-use-R-for-business-2021.html

Educational Aspects

- Common Learning Goals:
  - Proficiency in Programming Languages
    - Students are expected to gain proficiency in programming languages commonly used in data science, such as **Python, R, SQL, C/C++, and Java**
    - Python is particularly emphasized due to its popularity, extensive library support, and ease of use for data science tasks
    - R is highlighted as an open-source language specifically designed for data science, focusing on statistical computing, machine learning, data manipulation, and visualization
    - C/C++ and Java are sometimes mentioned for their roles in high-performance applications, machine learning, statistical analysis, and data visualization
  - Application of Programming Languages
    - Students are expected to apply programming languages for tasks such as data manipulation, statistical analysis, machine learning, data visualization, and building machine learning models
    - Proficiency in using Python and R for machine-learning models and dealing with large datasets is emphasized.
  - Understanding of Data Science Tools and Libraries:
    - Mastery of data science tools and libraries associated with programming languages, such as Pandas, NumPy, Matplotlib in Python, and machine learning libraries like PyTorch and TensorFlow written in C/C++
  - Practical Machine Learning:
    - Acquisition of practical machine learning skills is a key component of the program.
  - Database Systems and Data Preparation:
  - Data Visualization:
    - Proficiency in data visualization using programming languages like Python and R is a crucial learning goal
  - Software Development Skills:
    - Build and automate data pipelines and analysis frameworks

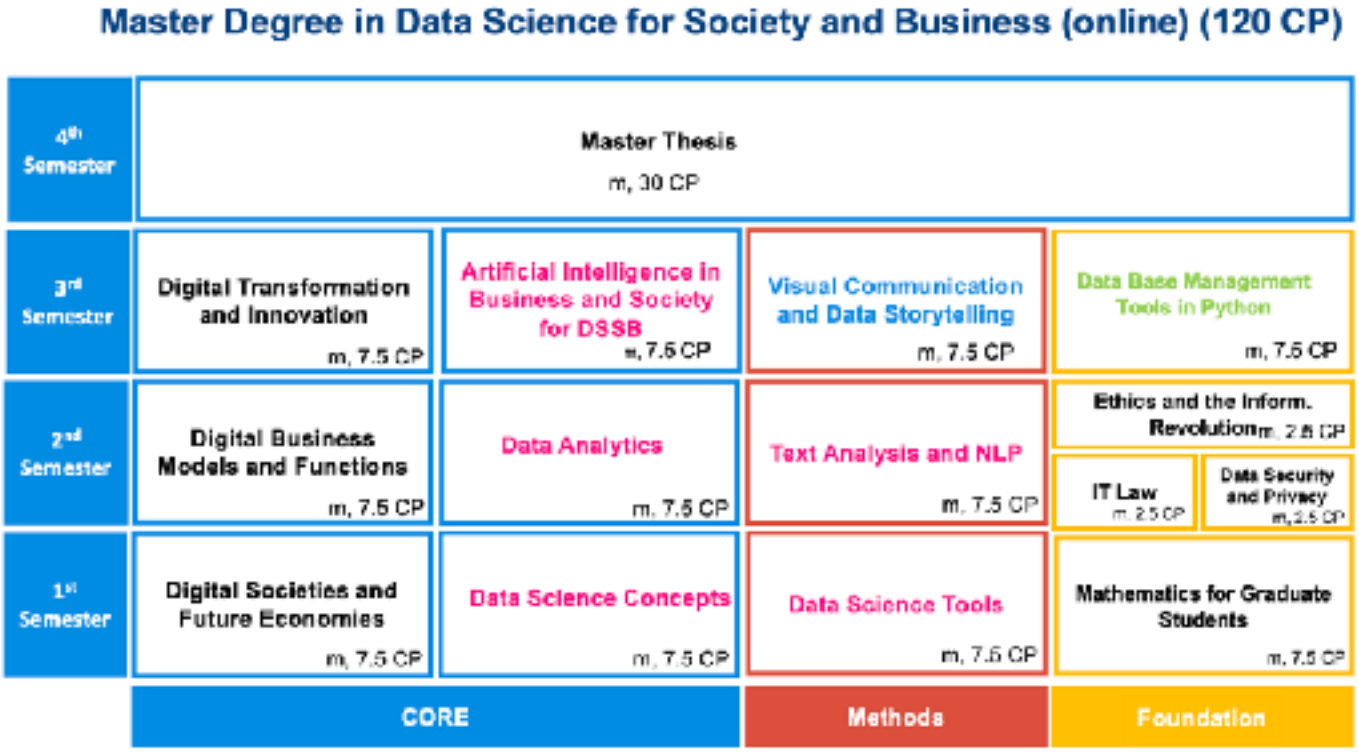
Integrating R and Python



Distribution of the R and Python programming languages across Constructor University's Master of Data Engineering Technologies program courses.

Integrating R and Python

CONSTRUCTOR UNIVERSITY



Distribution of the R and Python programming languages across Constructor University's Master of Data Science for Society and Business program courses.

Integrating R and Python

Distribution of the R and Python programming languages across the University of British Columbia's Master of Data Science program courses.

https://ubc-mds.github.io/2020-02-03-teach-python-and-r/

Pedagogical challenges of teaching R and Python concurrently

- Mixed proficiencies of previous R & Python programming skills
  - in students
  - in instructors
- Newcomers have to learn both data science concepts and tools
  - Dual (triple) task interference
- Memory decay during breaks in practice
- Standard environment for both or optimised environment for each
- Relevance of fundamental programming language specifics
- How to teach underlying paradigms through/instead of/in addition to language specifics

Some considerations

- achieve proficiency in diverse programming languages
- diverse backgrounds and expectations
- data-centric vs. model-centric vs. output-centric
- role of software development
- role of automation
- causality
- underlying paradigms and philosophies
- further differentiation in jobs and roles
- qualification and learning standards

Conclusion

Python vs R:

- *both languages are efficient for Data science*
- *provide rich tool kit for data analysis pipeline*

- ***R***
  - *excels at data visualization*
  - *has a scientific orientation*
  - *is more focused on handling data in a statistical perspective*
  - *provides a large ecosystem for data science and communication*

- ***Python***
  - ***f**ocuses on practical side of software implementation.*
  - *automation*
  - *deep learning*
  - *production or deployment*

Prof. Dr.
Adalbert F.X. Wilhelm

Professor of Statistics
Vice-Dean Bremen International Graduate School of Social Sciences

awilhelm@constructor.university
+ 49 421 200-3402

Constructor University Bremen gGmbH
Campus Ring 1
28759 Bremen
Germany