Design of Calculation Engines, KPIs, and Metrics for Data Engineering

- A Comparative Study of Machine Learning Models and Statistical Metrics across different languages

Project ID: 2024-S2-68
Sponsor: Professor Julio Romero

Group ID: 11522-2S2-25

Vibhore | Rahul | Ishtiaq | Hasan | Tushar
Mentor: Yasaman Baradaran

INTRODUCTION

CHALLENGES

- Selecting the right programming language for machine learning models is challenging.
- Different languages offer varying strengths in terms of performance, accuracy, and ease of use.

OBJECTIVES

- Evaluate performance across 5 different languages.
- Languages were compared using key metrics.
- Provide insights for optimal language selection.

APPROACH

- Implemented classification and statistical methods across all languages .
- Measured different metrics for all the platforms.
- Used several datasets from diverse backgrounds.

KEY FINDINGS

- Selecting the right programming language for machine learning models is challenging.
- Different languages offer varying strengths in terms of performance, accuracy, and ease of use.

METRIC COMPARISION

Languages

- Python
- Julia
- R
- C++
- Rust

Models

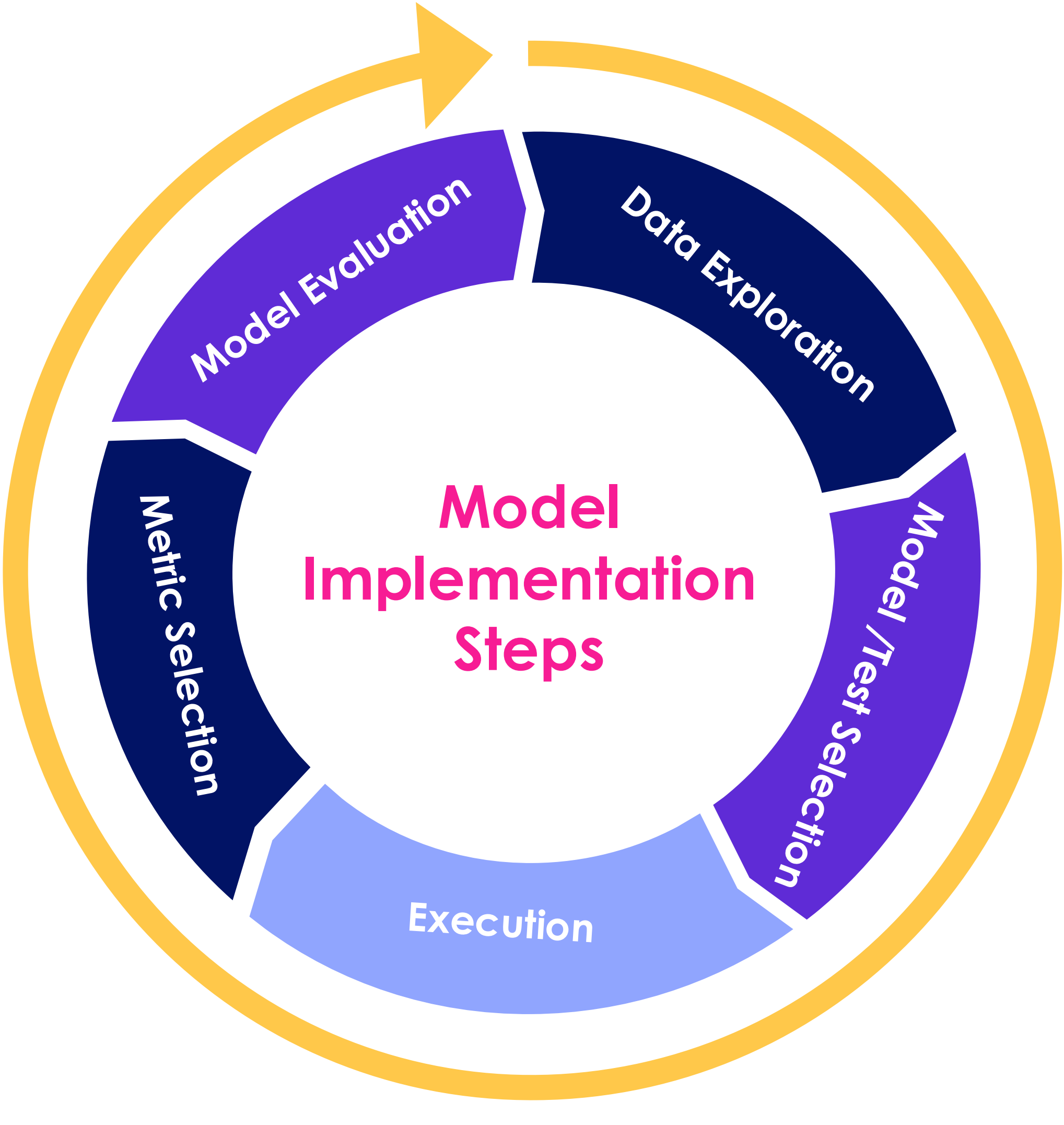
- Logistic Regression
- SVM
- Linear Regression
- ANOVA

KPIs

- Accuracy
- F1-score
- Precision
- Recall
- Rand Index
- Specificity
- P-values
- T-tests
- Tukey's HSD
- Mean Square
- Sum of Squares
- Degree of Freedom
- F value
- P value

Language	Intercept	Seeding	SNE	T-values (Seeding)	P-values (Seeding)
Julia	-0.35	15.68	0.42	3.5	0.0037
C++	-0.35	15.53	0.37	3.5	0.0037
Python	-0.35	15.68	0.42	3.5	0.0037
R	-0.35	15.68	0.42	3.5	0.0037
Rust	0.07	15.95			

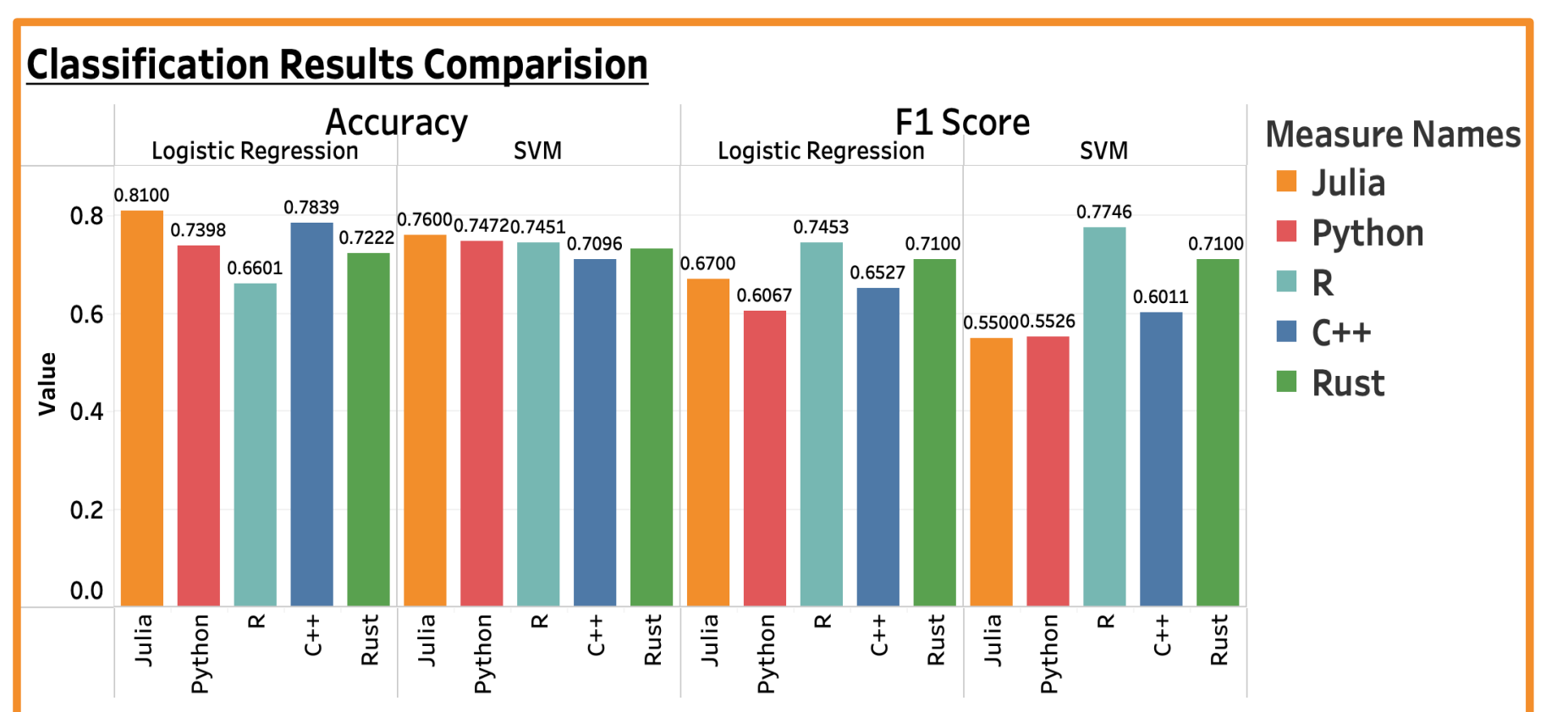
METHODS



1. Language Selection
2. Dataset Selection
3. Model Implementation
4. Metric Evaluation
6. Change Language
7. Comparative Analysis

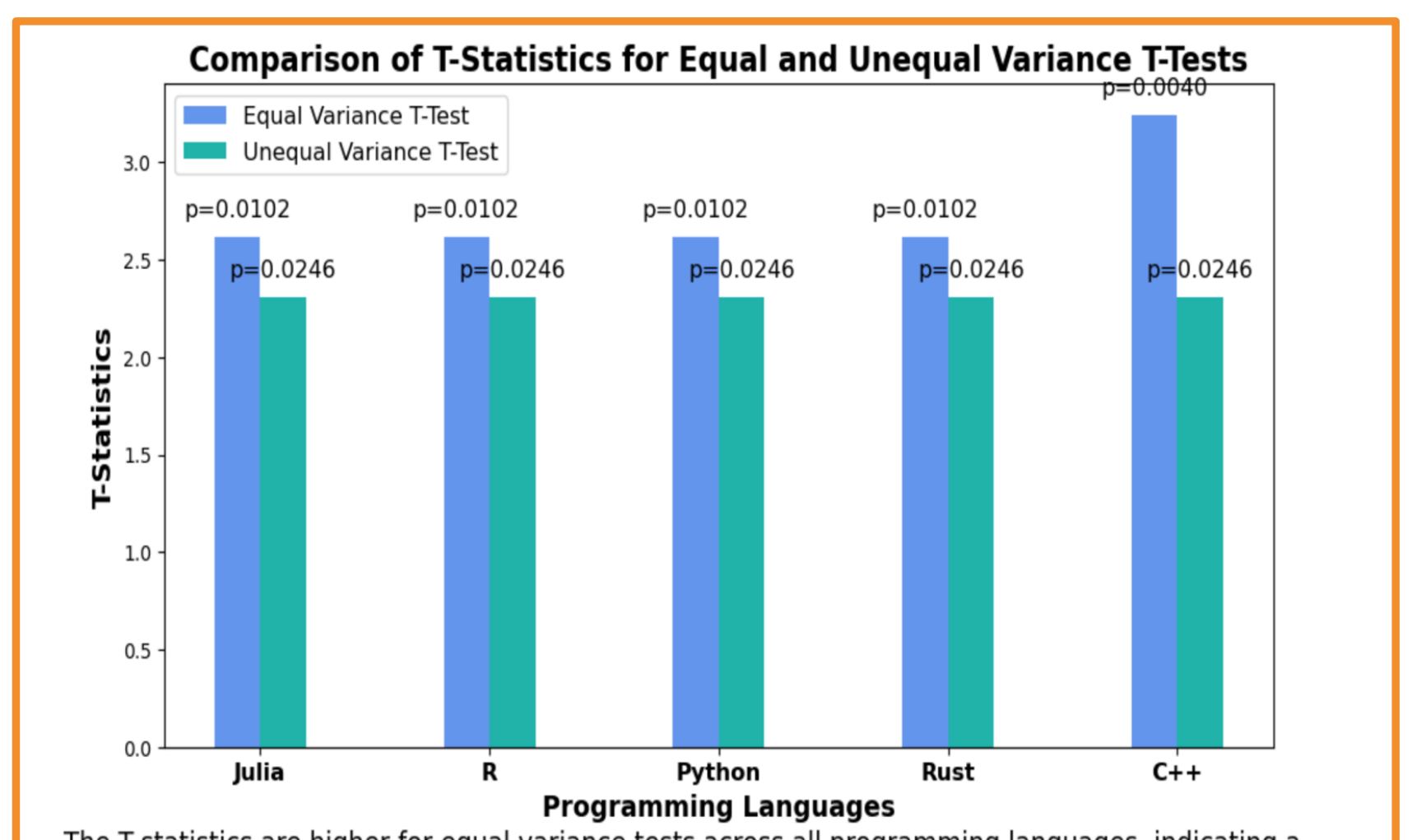
VISUALISATIONS

Classification Results Comparision



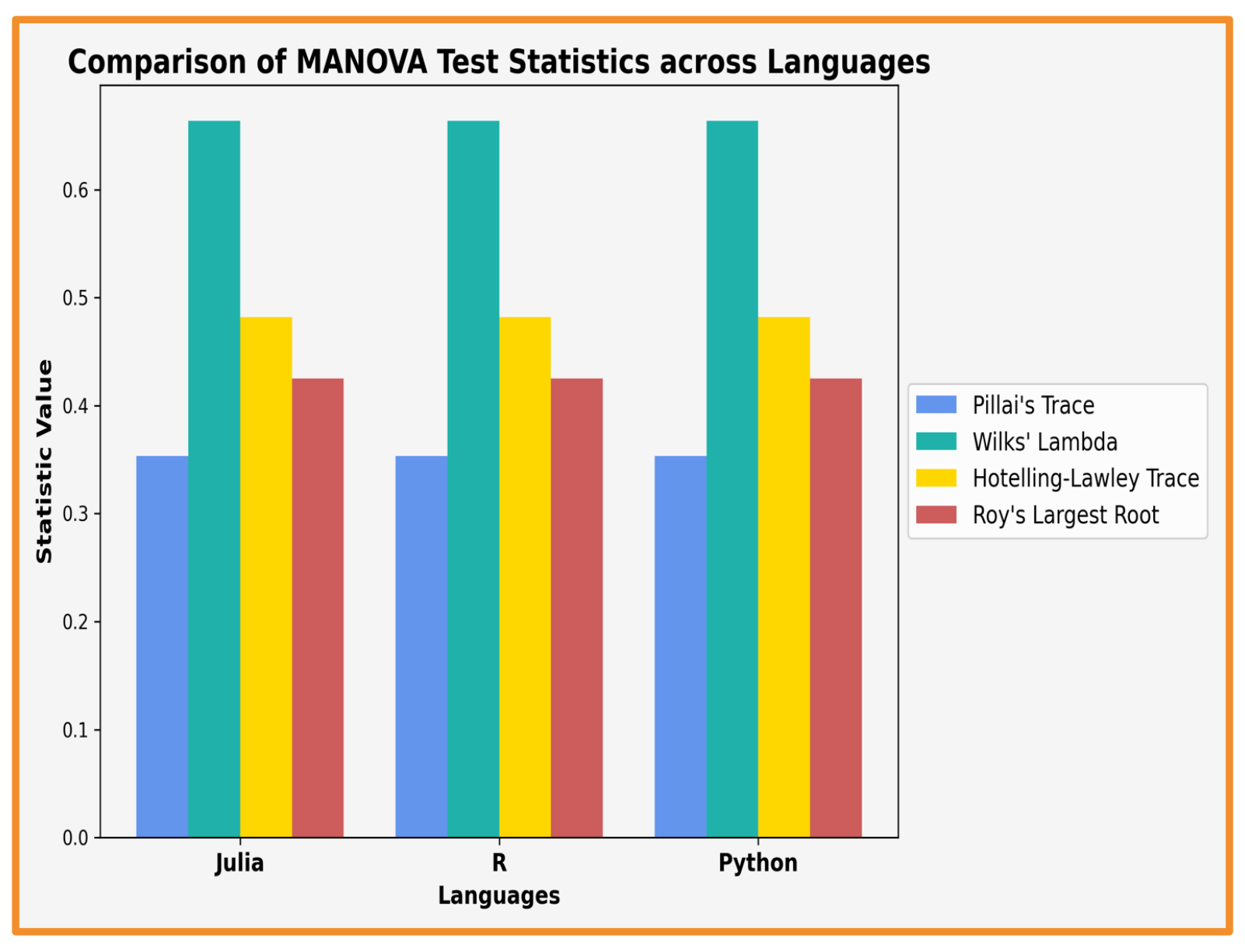
- Julia consistently shows the highest accuracy for Logistic Regression, followed closely by C++, while Python and R show lower accuracy for both models.
- In terms of F1 Score, SVM generally outperforms Logistic Regression, with Rust and R performing well for both models, while Python has the lowest F1 score.

Comparison of T-Statistics for Equal and Unequal Variance T-Tests

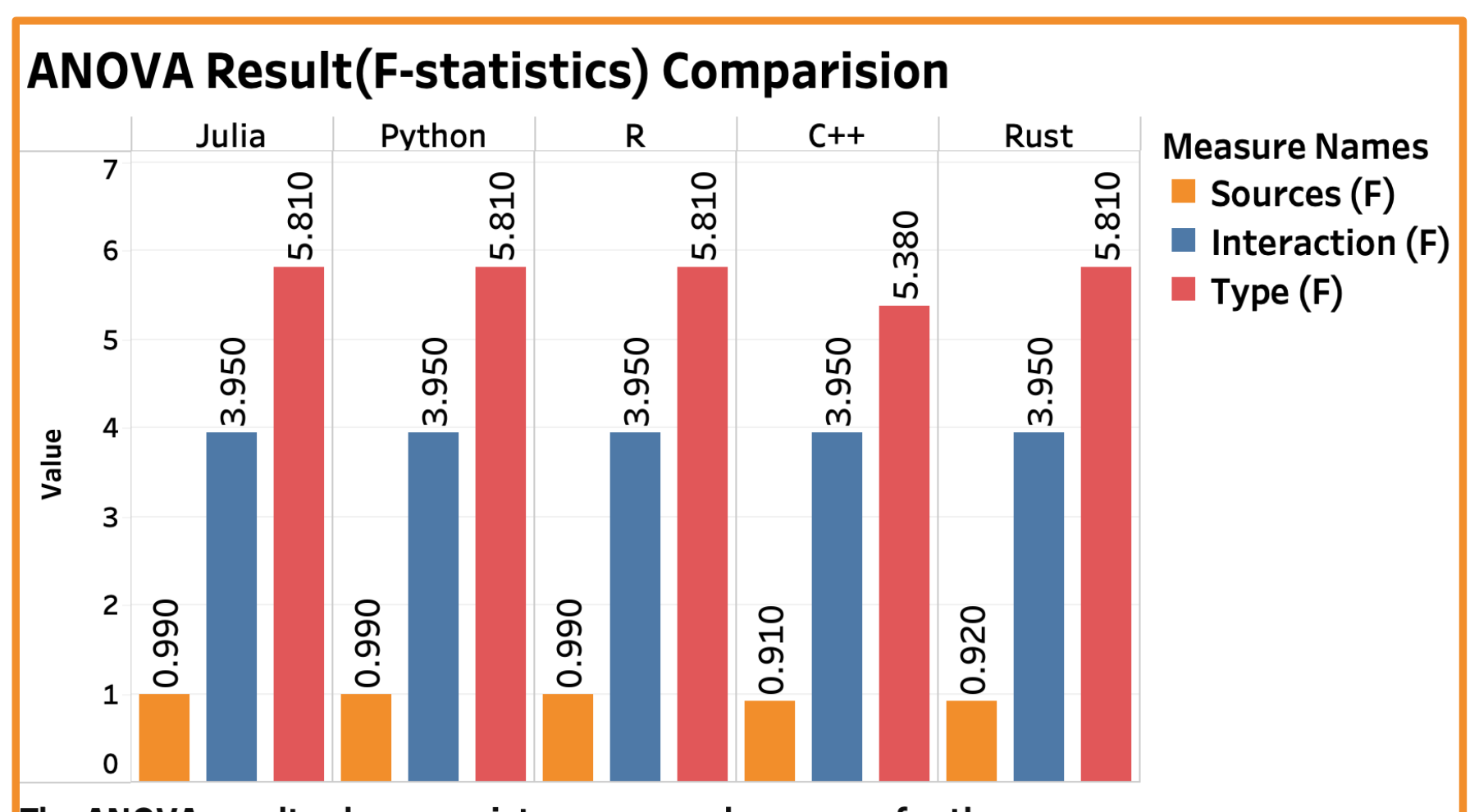


The T-statistics are higher for equal variance tests across all programming languages, indicating a stronger statistical significance under equal variance assumptions.
For unequal variance tests, the T-statistics are consistent across languages, but slightly lower than in the equal variance case, showing minor sensitivity to variance assumptions.

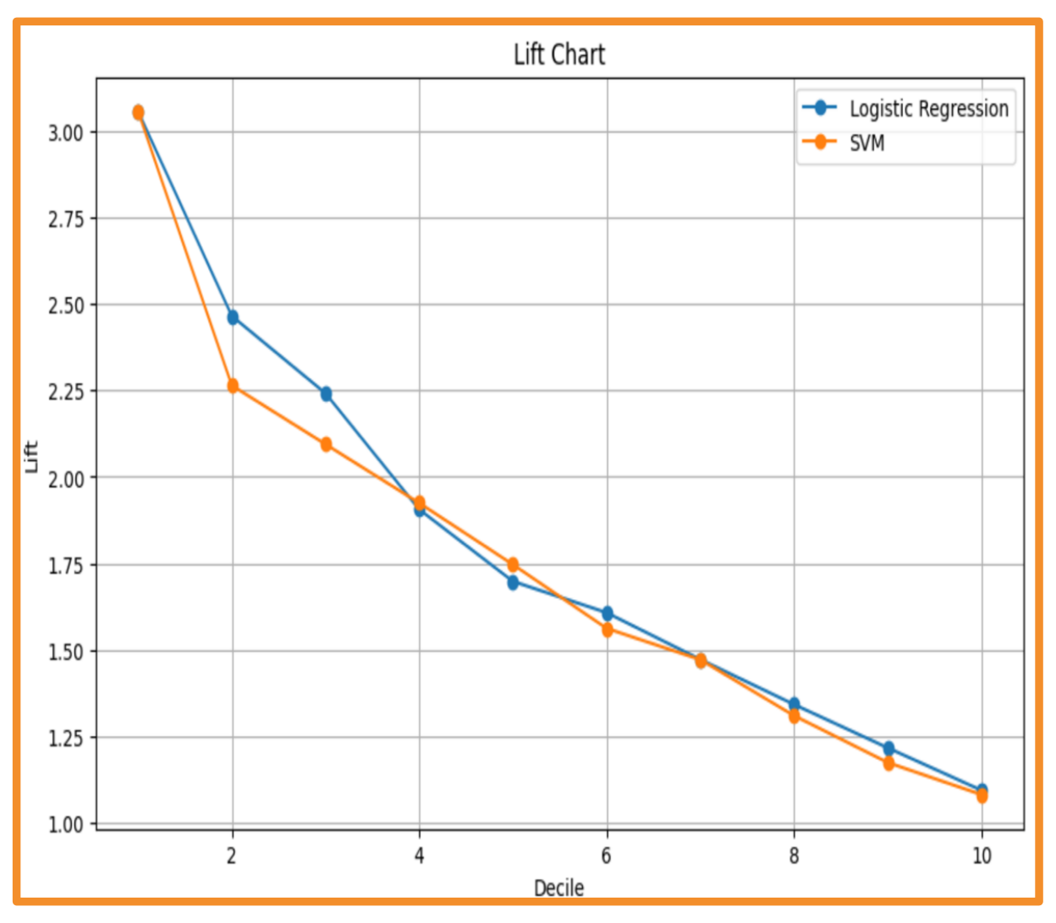
Comparison of MANOVA Test Statistics across Languages

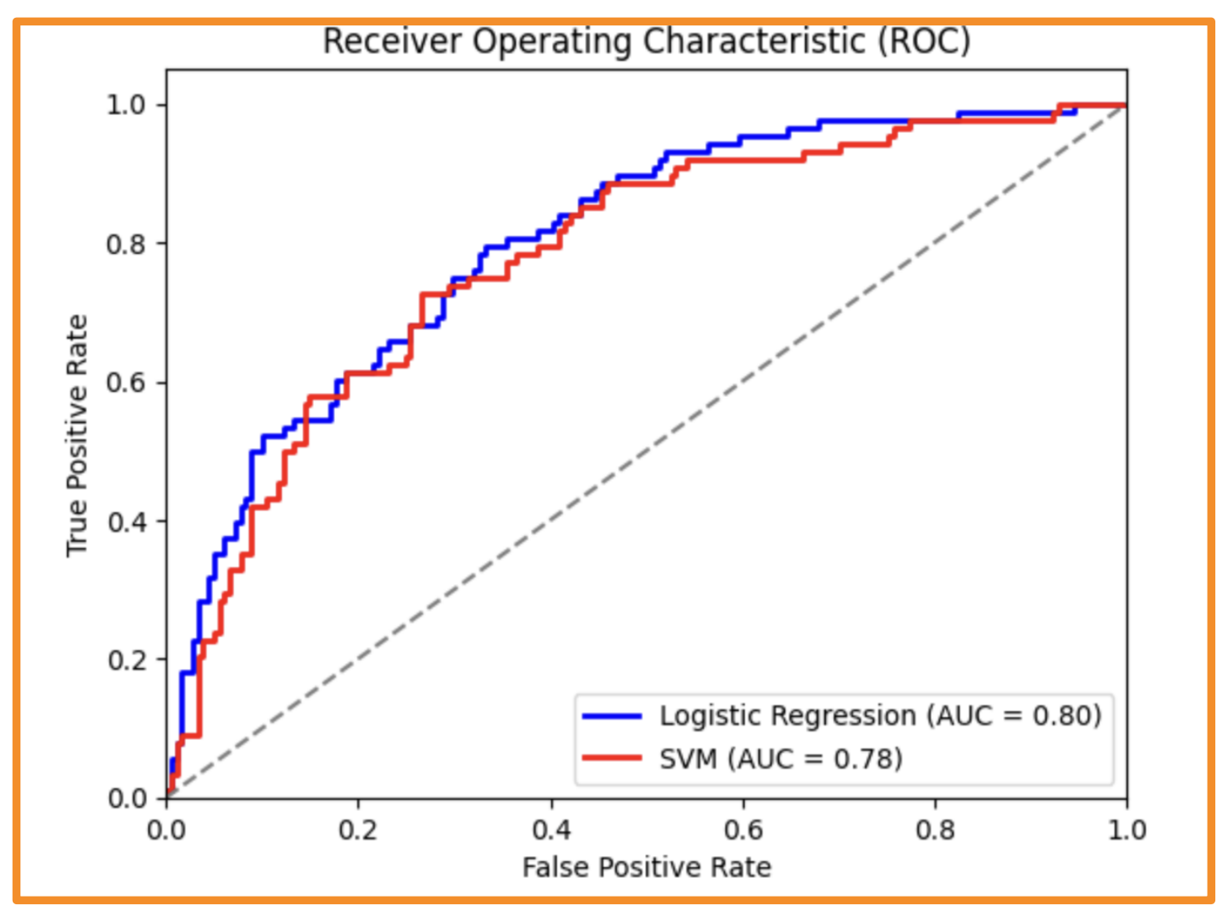


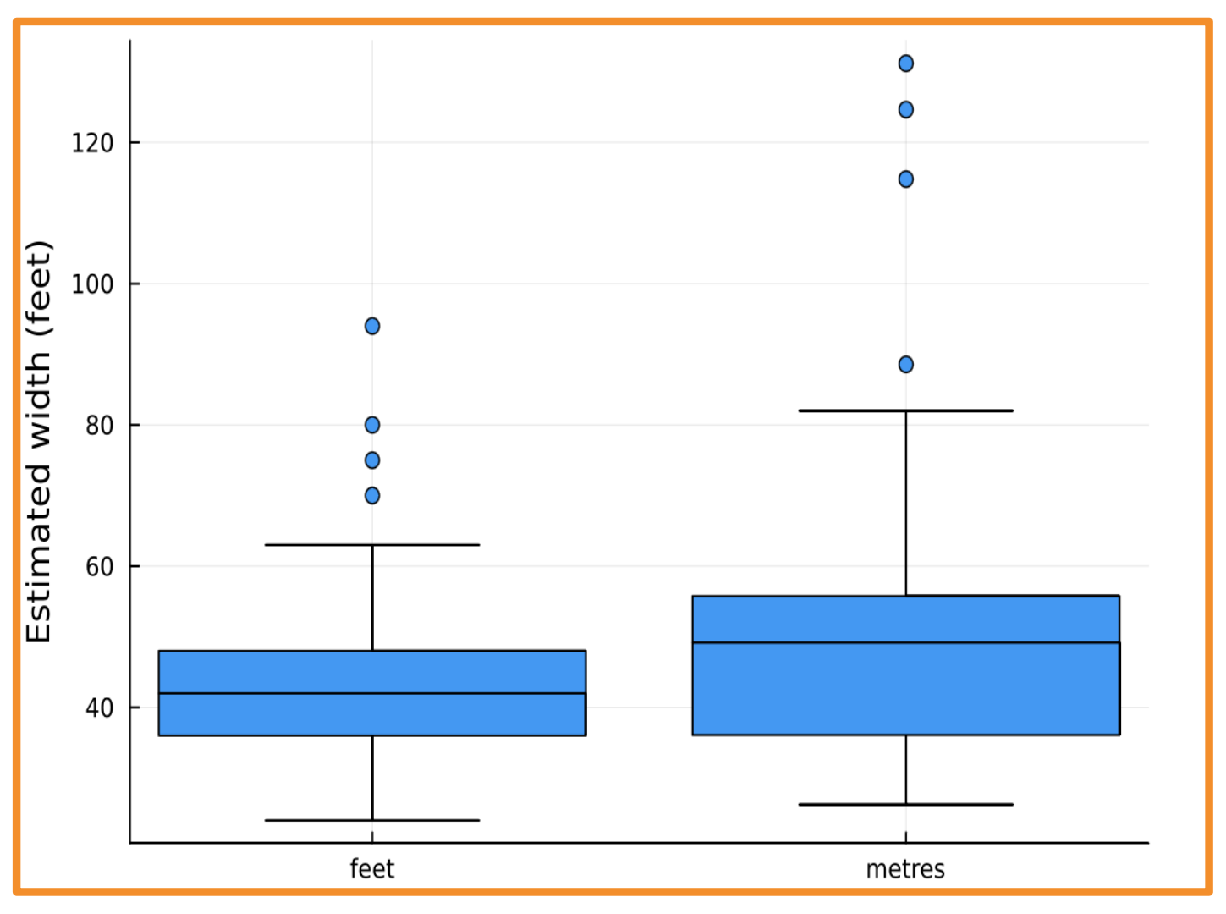
ANOVA Result(F-statistics) Comparision

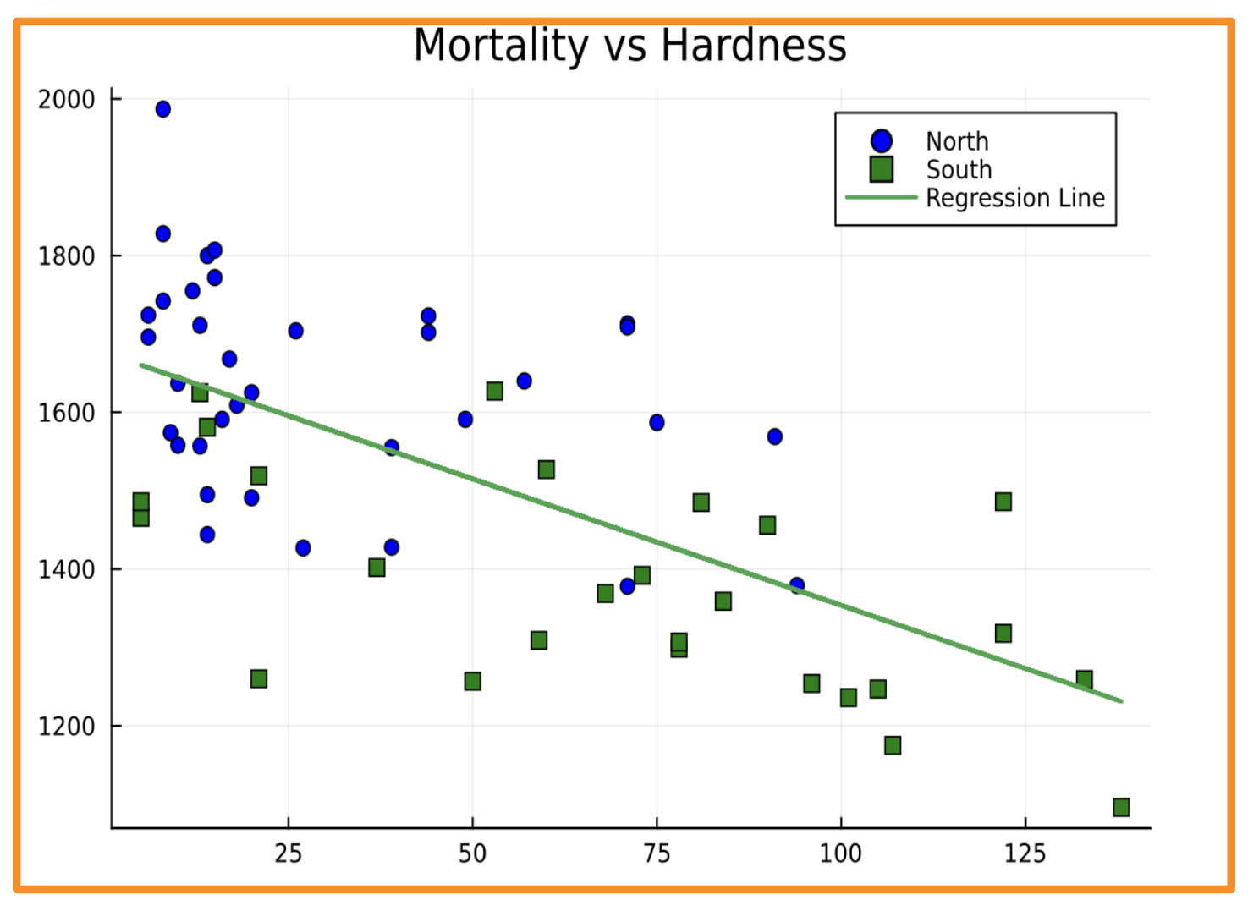


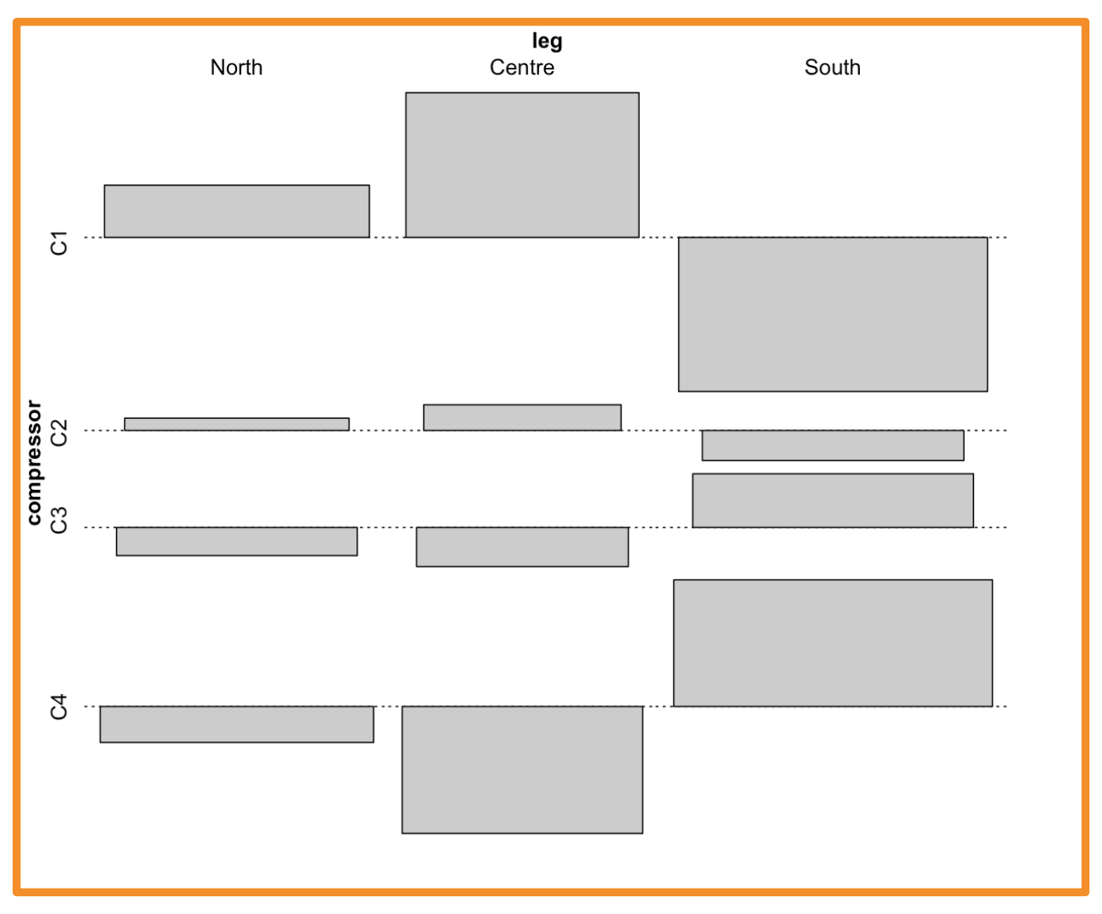
The ANOVA results show consistency across languages for the F-statistics.
Slight variations are observed in the Sources F-statistic, with C++ and Rust showing marginal differences.







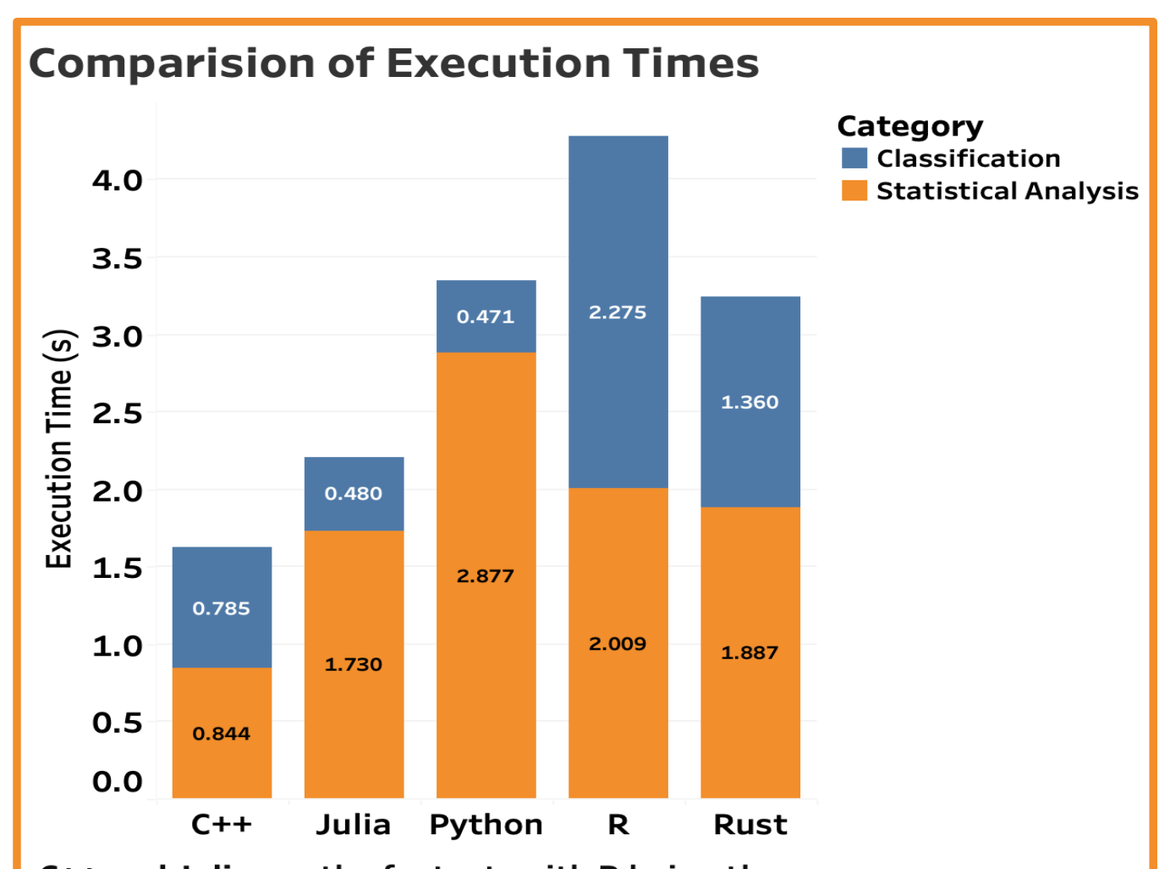




The graphs (like the ROC, box plot, scatter plot etc.) were consistently generated in Julia, Python, and R, showcasing their robust visualisation libraries (e.g., Matplotlib, Plots, ggplot2). However, C++ and Rust lack built-in libraries for statistical visualizations, making it challenging to create graphs in these languages

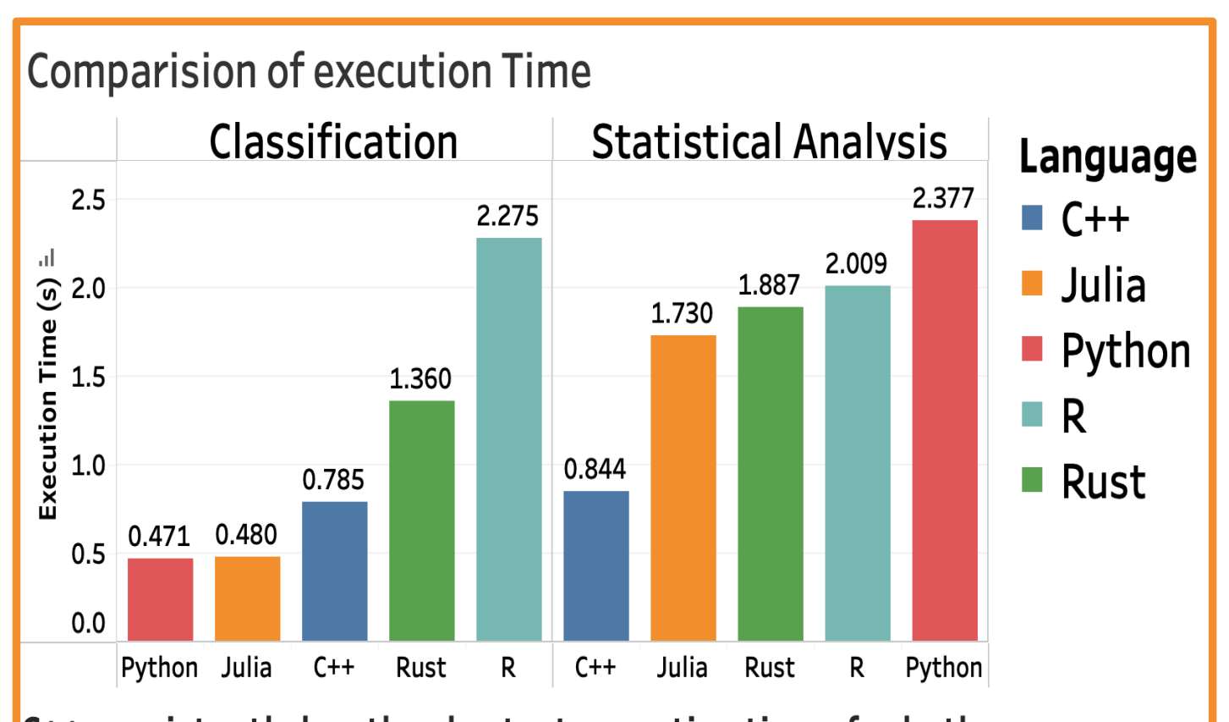
RESULTS

Comparision of Execution Times



- C++ and Julia are the fastest, with R being the slowest.
- All algorithms performed well in classification tasks, only R excelled in statistical computations
- Julia stood out as the most balanced for both classification and statistical tasks.

Comparision of execution Time



C++ consistently has the shortest execution times for both classification and statistical analysis tasks, while Rust and R exhibit the highest execution times.

Classification speeds are generally faster than statistical analysis speeds for all languages, except for R, where classification takes longer.

- **Julia** :
  - Execution time of 2.2s.
  - Balanced speed and ease of use
  - Moderate learning curve
  - High-level abstraction
- **Python**
  - Easiest development experience
  - Execution time of 2.87s
  - Lowest complexity and simplest syntax
  - Most mature library ecosystem

- **R**
  - Better for statistical tasks
  - Execution time of 4.2s.
  - Easy to learn and use
  - Strong statistical library ecosystem
- **C++**
  - Fastest with execution time of 1.7s
  - complex syntax.
  - Steep learning curve
  - Lacks the comprehensive libraries

- **Rust**
  - Slower in statistical analysis.
  - Complex memory management
  - Harder to develop with
  - Library ecosystem is still developing for data analysis.

FINDINGS

- Ease of Use: Python is the easiest to use, while C++ and Rust are more complex.
- Library and Ecosystem: Python and R have the most extensive libraries, while C++ and Rust are less suitable for data analysis.
- Performance: C++ and Julia provide the fastest execution, while R was the slowest.
- Learning Curve: Python is the easiest to learn, while C++ and Rust have the steepest learning curves.
- Balance: Julia provides an excellent balance between high performance and ease of use, making it ideal for scientific computing.
- Results and Visualizations: Julia, Python, and R offered all results and visualizations, while C++ lacked full support in this area.

LIMITATIONS

- Incomplete Results in C++ and Rust
- Execution Environment Variability
- Limited Use of Algorithms/Models
- Focus on Execution Time
- GPU models were not used.
- Cross-Language Compatibility

CONCLUSION

- This research identified C++ as the fastest, python as the easiest and Julia as the most suitable for data analysis.
- C++ and Rust lacked support for producing complete visualisations.
- This research helped identify the most efficient programming languages for data-analysis, providing insights on where performance gains can be made.
- These findings can help make informed decisions about language selection, optimizing resources, reducing costs, and improving project efficiency in data analysis workflows.

UNIVERSITY OF CANBERRA

We would like to express our sincere gratitude to Professor Julio Romero and Ms. Yasaman Baradaran for their invaluable guidance, expertise and encouragement throughout the project. Their continuous support, insightful feedback, and encouragement were crucial in guiding this project to completion.

References:
1. F. Hayat, A. U. Rehman, K. S. Arif, K. Wahab and M. Abbas, "The Influence of Agile Methodology (Scrum) on Software Project Management, (SNPD), Toyama, Japan, 2019
2. R. Kumar, A. Mankodi, A. Bhatt, B. Chaudhury and A. Amrutiya, "Cross-Platform Performance Prediction with Transfer Learning using Machine Learning," (ICCCNT), Kharagpur, India, 2020