nalysing the Programming Language Preferences of Data Scientists in 2020

12-9-24

Contribution:

The development of this report was a collaborative effort between both students. Each student took primary responsibility for specific sections while maintaining active involvement throughout the entire project. Student 1 focused mainly on sections 1 and 2 of the analysis section while also making substantial contributions to the advanced analysis, introduction, and conclusion. dditionally, Student 1 reviewed sections 3 and 4. Student 2 concentrated on reviewing sections 1 and 2, while working on sections 3 and 4 in the analysis section. Student 2 also looked over Student 1's sections and contributed to the advanced analysis sections. Both students were responsible for writing R code, methods, and conclusions for their respective sections. Throughout the project, both students consistently reviewed each other's work and made iterative improvements to ensure the report's quality and coherence.

Use of GPT: The report utilized ChatGPT to enhance the clarity and readability of our analysis, specifically in refining the language of method and conclusion sections to be more concise and professional. While all underlying analysis, code development, and interpretations were conducted independently by the students, GPT served as a writing aid to improve the presentation of our findings and maintain consistency in technical writing style throughout the report.

1. Introduction

Programming languages are the cornerstone of data science and machine learning, with practitioners' choices directly impacting their analytical capabilities and career paths. The 2020 Kaggle Machine Learning and Data Science Survey provides comprehensive insights into these choices, capturing data from 20,036 respondents across 171 countries. Understanding these patterns is crucial for educational institutions optimizing their curricula and organizations planning their technical infrastructure.

The main goal of this analysis is to investigate programming language patterns among data science practitioners and provide actionable insights for curriculum development and resource allocation. In this analysis, we examine three key aspects: first, we analyze the relationship between educational background and programming language preferences, using both point and interval estimates to account for sampling variability. Second, we investigate regional variations in language adoption to understand geographical influences on technology choices. Finally, we explore how early-career professionals approach language selection and usage. By combining these findings, we aim to inform evidence-based decisions about programming language support and training in educational and professional settings.

Data

The data comes from the 2020 Kaggle Machine Learning and Data Science Survey conducted between October 7-30, 2020. The survey received 20,036 usable responses from participants across 171 different countries and territories, with countries receiving fewer than 50 respondents grouped into an "Other" category for anonymity. The survey was distributed to the entire Kaggle community through their email list and promoted on both the Kaggle website and Twitter channel. The dataset includes responses to multiple-choice questions about programming languages used, development environments, machine learning frameworks, and various demographic information. To protect respondents' privacy, free-form text responses were excluded from the public dataset, and response order was randomized. This analysis focuses primarily on the programming language-related variables and their relationships with educational background, geographical location, and career stage.

2. Basic nalysis

2.1 Programming Language Usage and Educational Background nalysis

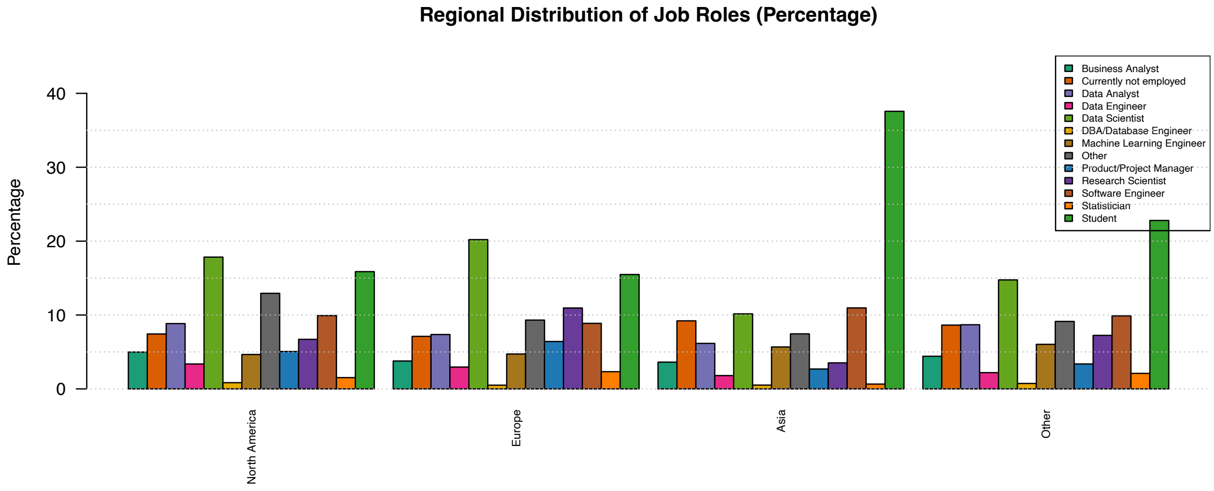
Method

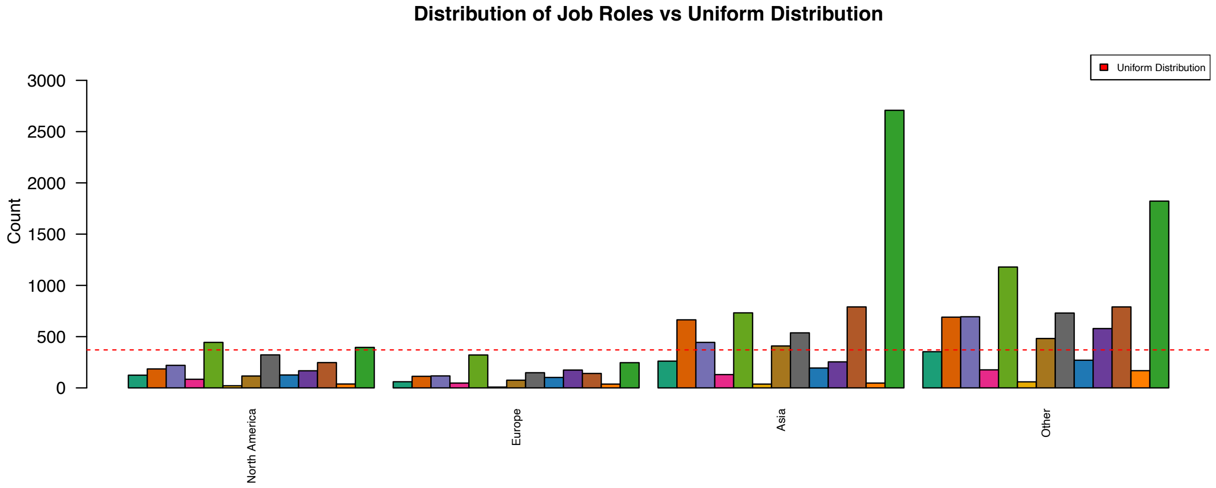
To analyze the regional distribution of job roles in the Kaggle 2020 Machine Learning and Data Science Survey, the dataset was first cleaned to remove unnecessary rows and columns. The country column (Q3) was mapped to predefined regions, including North merica, Europe, sia, and "Other," using a custom mapping function. This ensured all countries were categorized under appropriate regional groupings. The job role column (Q5) was standardized to include 13 distinct job roles (e.g., Data Scientist, Software Engineer, Student), ensuring consistency across the dataset.

Job roles were then aggregated by region to calculate both absolute counts and percentages of respondents for each role within their respective regions. Percentages were calculated to normalize for differing respondent totals across regions, enabling fair comparisons. dditionally, a hypothetical uniform distribution of job roles across regions was simulated to provide a baseline for assessing representation disparities.

Two visualizations were generated: (1) a bar plot showing the percentage distribution of job roles within each region, and (2) a bar plot of absolute job role counts with a dashed line representing the uniform distribution. custom color palette was applied to enhance visual clarity, ensuring that each job role was uniquely represented across the plots.

nalysis





Conclusion

The analysis of the distribution of job roles compared to a uniform distribution reveals significant regional disparities. North merica and Europe emerge as strongholds for professional roles, particularly in established careers such as Data Scientist and Software Engineer. These regions benefit from mature job markets, robust educational systems, and a strong industry presence, which drive their dominance in professional representation. In contrast, sia's notable spike in student representation underscores its potential as a burgeoning talent hub. This trend reflects growing interest and investment in data science within the region, highlighting a promising pipeline of future professionals. Other regions exhibit a more dispersed and irregular distribution, likely influenced by varying economic conditions, educational infrastructure, and industry opportunities.

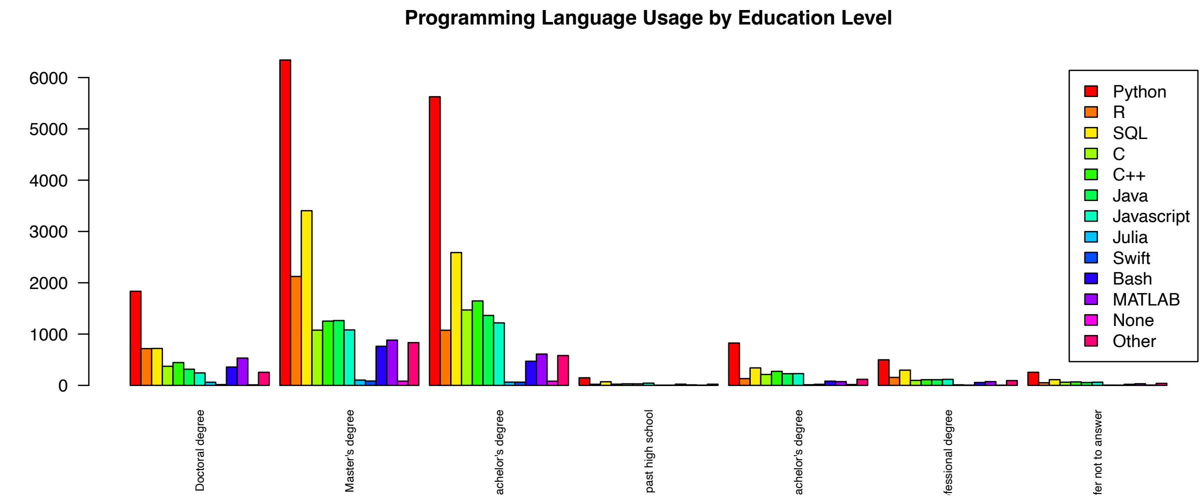
The comparison to a uniform distribution, represented by the red dashed line, further underscores these disparities. North merica and sia deviate significantly, with North merica skewed toward professionals and sia toward students. Europe aligns more closely with the uniform distribution, suggesting balanced participation, while other regions fall short of consistent representation. These disparities may be influenced by biases in data collection methods, such as reliance on specific platforms or professional networks, as well as broader systemic factors like access to education and regional economic conditions. ddressing these imbalances will require targeted strategies, such as promoting equitable access to education and resources, fostering professional opportunities in underrepresented regions, and conducting further research into regional trends in education, programming preferences, and career progression. By bridging these gaps, the global data science workforce can become more inclusive and better positioned to address regional needs and opportunities.

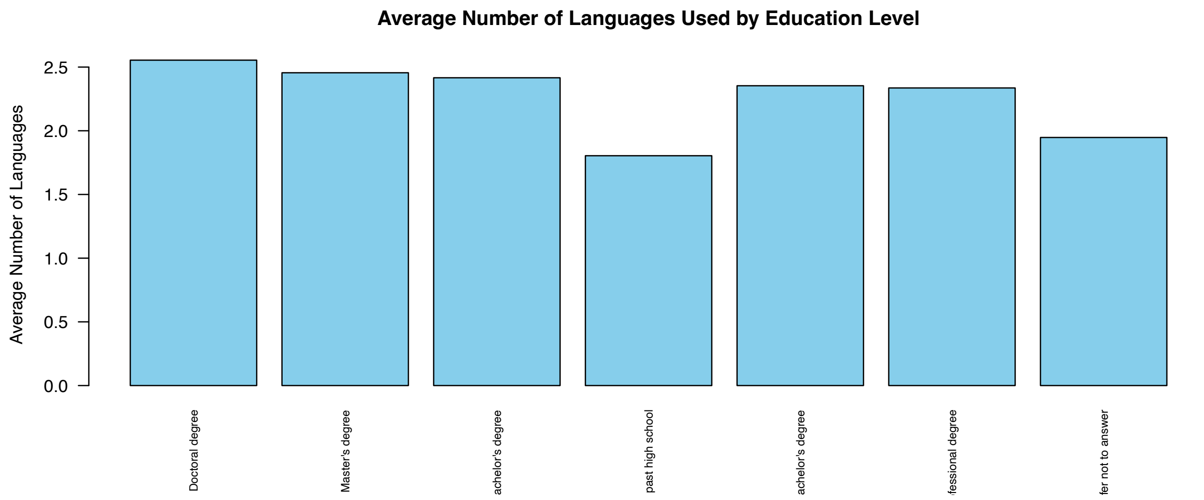
2.2 Regional Distribution of Programming Language Preferences

Method

The analysis of programming language usage patterns began with processing the responses to Question 7, "What programming languages do you use on a regular basis?" through one-hot encoding. Each programming language option, numbered from 1 to 12, along with an additional "Other" category, was converted into separate binary columns, where a value of 1 indicated usage and 0 indicated non-usage. Next, the data was aggregated based on respondents' education levels, summing the counts for each programming language within each group. This allowed us to examine trends in language usage across different educational backgrounds. To further explore these patterns, we visualized the aggregated data using bar charts, illustrating the counts of programming languages used for each education level. dditionally, we calculated the average number of programming languages used by respondents at each education level and presented this metric through a separate graph. This analysis provided insights into both the popularity of individual programming languages and the diversity of language proficiency among respondents with varying educational attainments.

nalysis





```
## Warning in chisq.test(contingency_table): Chi-squared approximation may be
## incorrect
```

```
##
##  Pearson's Chi-squared test
##
## data:  contingency_table
## X-squared = 1481.8, df = 72, p-value < 2.2e-16
```

Table 1: Contribution to Chi-Square (Standardized Residuals)

Category	Python	R	SQL	C	C++	Java
Doctoral degree	-2.06	8.12	-6.99	-1.91	-1.34	-5.01
Master's degree	0.33	9.20	6.15	-7.38	-7.69	-2.85
Bachelor's degree	1.45	-11.41	-1.71	8.51	7.76	4.82
No formal education past high school	0.53	-2.73	0.27	-1.14	-0.66	-0.13
Some college/university study without earning a bachelor's degree	-0.52	-6.48	-3.32	2.50	4.56	3.38
Professional degree	-1.53	0.64	2.34	-1.60	-1.80	-0.45
I prefer not to answer	0.04	-2.16	-1.03	1.05	0.67	0.12

Table 2: Continuation Contribution to Chi-Square (Standardized Residuals)

Javascript	Julia	Swift	Bash	M TL B	None	Other
-6.65	5.01	-1.72	9.30	15.58	-2.88	0.90
-3.93	-0.25	0.61	1.48	-0.62	-0.08	1.53
4.72	-3.02	-0.88	-6.37	-6.30	1.03	-4.15
3.19	0.39	-0.60	1.94	-2.27	0.09	1.25
5.32	-0.32	3.14	-1.35	-4.20	1.75	1.23
1.37	0.66	-0.31	-0.53	-0.26	-1.16	3.31
1.85	0.83	0.98	-1.50	-0.71	1.97	1.27

Conclusion

The analysis revealed significant variations in programming language usage across different education levels. Python emerged as the most frequently used programming language across all groups, highlighting its widespread adoption. Other languages such as R, SQL, and C also showed varying degrees of usage, often influenced by the respondent's level of education. For example, degree holders exhibited a strong preference for R and SQL, reflecting its prominence in research and data analysis. In contrast, respondents with no formal education past high school showed less diversity in language usage, often favoring more accessible or widely used languages like Python. The chi-squared test further confirmed significant relationships between education level and language usage, with some groups contributing disproportionately to specific languages.

In contrast, although higher education levels, such as doctoral and professional degrees, were associated with slightly higher averages of programming languages used, the differences across education levels were minimal. Most respondents used a comparable number of programming languages on average, suggesting that exposure to multiple languages is consistent across diverse educational backgrounds.

In summary, the findings demonstrate the impact of educational background on programming language preferences, with some patterns of specialization emerging at higher education levels. While individuals with doctoral and professional degrees showed a slightly higher average number of programming languages used, the variation across education levels was minimal, indicating that familiarity with multiple languages is widespread regardless of educational attainment. Specific trends, such as the greater use of R and M TL B among advanced degree holders, highlight the influence of academic and professional demands on language selection. These insights underline the need for programming language education to align with the unique requirements of different career paths and fields of study.

2.3 Early Career Programming Language doption Patterns

Method

To further analyze the relationship between educational background and programming experiences in the Kaggle 2020 Machine Learning and Data Science Survey, we conducted a comprehensive analysis of programming language library usage patterns across different educational levels. The programming language data was extracted from the Q14 columns, which were also converted to binary indicators (0/1) to standardize the analysis. This binary transformation allowed for clear identification of active language usage versus non-usage across respondents.

We created two primary visualization approaches to examine these patterns. First, a heatmap visualization was generated to show the percentage distribution of programming libraries usage across educational levels, providing a clear visual representation of library preferences within each educational group. Second, we calculated and visualized the average number of libraries used by respondents at each educational level to understand the depth of technical knowledge across educational backgrounds. The analysis was complemented by chi-square testing to evaluate the statistical significance of the relationship between educational attainment and programming library usage patterns.

nalysis

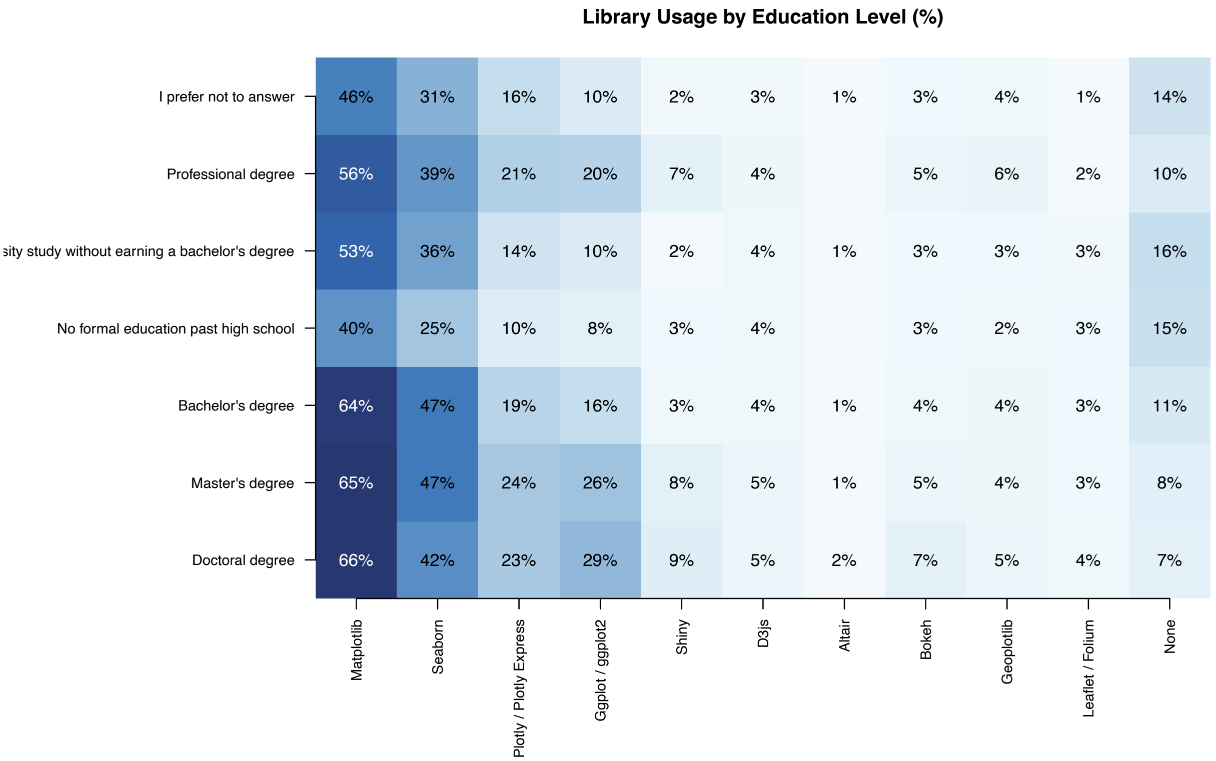


Table 3: verage Number of Libraries by Education Level

	Education Level	verage Number of Libraries
2	Doctoral degree	1.97
4	Master's degree	1.97
1	Bachelor's degree	1.77
6	Professional degree	1.70
7	Some college/university study without earning a bachelor's degree	1.44
3	I prefer not to answer	1.30
5	No formal education past high school	1.12

Table 4: Top Libraries by Education Level (Longer Format)

Education Level	Rank	Library	Usage %
Doctoral degree	1	Matplotlib	65.6
Doctoral degree	2	Seaborn	41.9
Doctoral degree	3	Ggplot / ggplot2	29.1
Master's degree	1	Matplotlib	65.0

Education Level	Rank	Library	Usage %
Master's degree	2	Seaborn	47.2
Master's degree	3	Ggplot / ggplot2	25.6
Bachelor's degree	1	Matplotlib	64.1
Bachelor's degree	2	Seaborn	47.2
Bachelor's degree	3	Plotly / Plotly Express	19.0
No formal education past high school	1	Matplotlib	40.4
No formal education past high school	2	Seaborn	24.6
No formal education past high school	3	None	14.6
Some college/university study without earning a bachelor's degree	1	Matplotlib	53.2
Some college/university study without earning a bachelor's degree	2	Seaborn	36.1
Some college/university study without earning a bachelor's degree	3	None	15.7
Professional degree	1	Matplotlib	55.8
Professional degree	2	Seaborn	39.5
Professional degree	3	Plotly / Plotly Express	21.2
I prefer not to answer	1	Matplotlib	45.6
I prefer not to answer	2	Seaborn	30.8
I prefer not to answer	3	Plotly / Plotly Express	15.8

Conclusion

The analysis reveals a clear relationship between educational background and library usage patterns, with higher education levels generally correlating with the adoption of more data visualization libraries. Doctoral and Master's degree holders demonstrated the highest average number of libraries used (1.97), followed by Bachelor's degree holders (1.77). Those with lower educational levels, such as "No formal education past high school" and "Some college/university study without earning a bachelor's degree," showed lower averages (1.12 and 1.44, respectively). This pattern suggests that advanced education may either necessitate or encourage familiarity with a broader range of technical tools.

Matplotlib emerged as the most widely used library across all education levels, with particularly high adoption rates among advanced degree holders, such as those with Doctoral (65.6%) and Master's degrees (65.0%). Seaborn was the second most popular library for these groups, further highlighting their preference for robust and versatile visualization tools. In contrast, individuals with lower educational attainment, such as those without formal education past high school, displayed lower adoption rates of libraries like Seaborn and Matplotlib but showed a notable proportion selecting "None," indicating limited library usage.

The differences in secondary library preferences are also noteworthy. For example, advanced degree holders frequently used Ggplot/Ggplot2 and Plotly/Plotly Express, which cater to specialized or interactive visualizations, while lower education groups demonstrated a narrower focus or none at all. These variations emphasize the role of education in shaping the diversity and complexity of library usage.

Overall, the findings suggest that education plays a significant role in determining familiarity with and usage of data visualization libraries. These insights are important for educators, curriculum designers, and employers, underscoring the need to tailor training and support to the specific technical demands of diverse educational and professional backgrounds.

2.4 Early Career Programming Language Patterns

Method

To identify dominant programming language combinations among early-career professionals in the Kaggle 2020 Machine Learning and Data Science Survey, we employed a systematic clustering approach focused on language usage patterns. The dataset was first filtered to include only respondents with less than 5 years of programming experience, capturing early-career developers through Q6 responses of "< 1 years", "1-2 years", and "2-5 years". Programming language usage data was extracted from the Q7 columns and converted to binary indicators, creating a standardized matrix where 1 represented active usage and 0 indicated non-usage of each language.

For the analysis, k-means clustering was selected as the primary analytical method with k=5 clusters, chosen through careful consideration of interpretability and distinct pattern identification. This method was particularly suitable for our binary data structure and provided clear centroids representing typical language combinations. The number of clusters was selected to balance between capturing distinct patterns while maintaining interpretable group sizes. For each cluster, we calculated usage percentages for individual languages, average number of languages per developer, and identified both primary (>30% usage) and secondary (10-30% usage) languages within each group.

nalysis

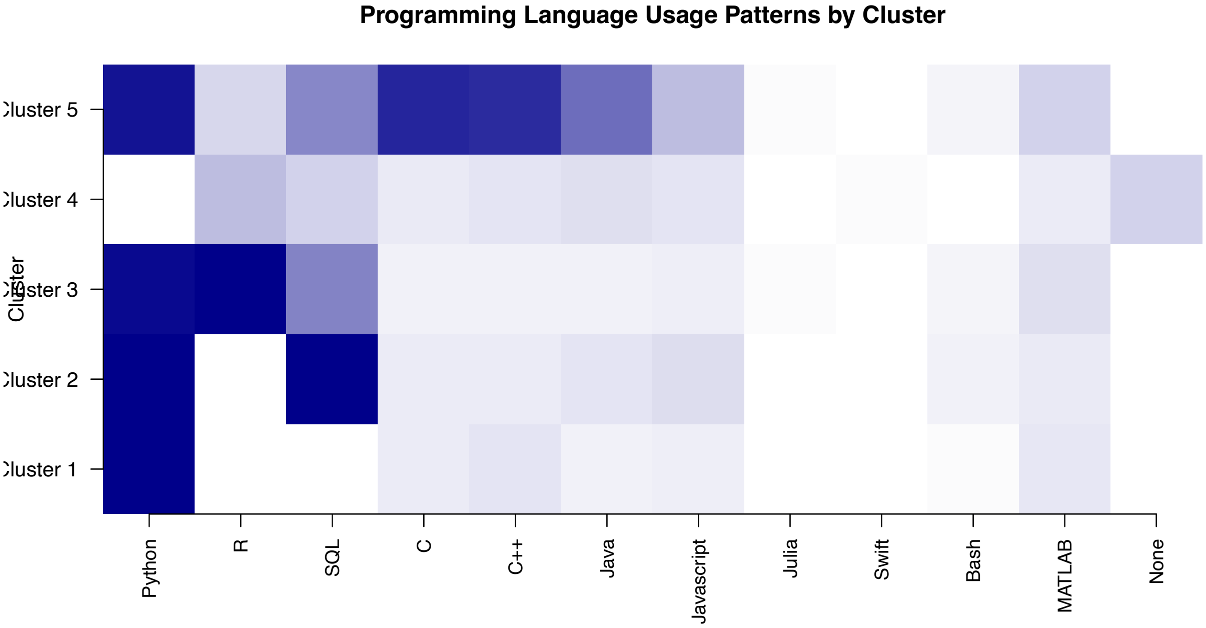


Table 5: Cluster nalysis of Programming Language Usage Patterns

Cluster	Size	Percentage (%)	vg. Languages	Primary Languages (>30%)	Secondary Languages (10-30%)
1	3342	42.7	1.4	Python	C++
2	1364	17.4	2.5	Python, SQL	Java, Javascript
3	1276	16.3	2.8	Python, R, SQL	M TL B
4	858	11.0	1.1		R, SQL, C++, Java, Javascript, None
5	978	12.5	4.3	Python, SQL, C, C++, Java	R, Javascript, M TL B

Conclusion

The analysis reveals clear patterns in how early-career professionals approach programming language acquisition. The dominance of Python-centric clusters (especially Clusters 1 and 2) suggests that Python serves as a common entry point into the field, often paired with SQL for data manipulation tasks. The emergence of a distinct R-Python-SQL cluster indicates a parallel path focused on statistical computing and data analysis. The presence of both highly focused (Python Specialists) and diverse (Full-Stack Developers) clusters suggests two distinct career development strategies: deep specialization versus broad technical exposure. The relatively small size of the Full-Stack cluster (12.5%) compared to Python-focused clusters indicates that most early-career professionals opt for specialization rather than diversification. These findings have important implications for educational planning, career development, industry trends, and hiring practices. The results also highlight the importance of data manipulation skills, with SQL featuring prominently in several clusters, suggesting this as a crucial complementary skill to primary programming languages.

3. dvanced nalysis

Method

The advanced analysis aimed to determine whether programming languages and libraries could effectively predict respondents' education levels. We began by cleaning the dataset, removing irrelevant header rows and handling missing or incomplete data to ensure a robust foundation for analysis. To facilitate quantitative evaluation, we transformed the education levels into a numeric column by mapping each education category to a corresponding numeric value. Subsequently, we computed a correlation matrix to explore the relationships between programming languages, libraries, and education levels. s the primary predictive tool, we employed a Random Forest classification model to analyze whether patterns in programming language usage and library preferences could accurately predict respondents' education levels. This combined approach of exploratory and predictive analysis allowed us to evaluate the feasibility of using technical tool usage as a proxy for education level classification.

nalysis

```
## + Fold1: mtry= 2
## - Fold1: mtry= 2
## + Fold1: mtry=13
## - Fold1: mtry=13
## + Fold1: mtry=25
## - Fold1: mtry=25
## + Fold2: mtry= 2
## - Fold2: mtry= 2
## + Fold2: mtry=13
## - Fold2: mtry=13
## + Fold2: mtry=25
## - Fold2: mtry=25
## + Fold3: mtry= 2
## - Fold3: mtry= 2
## + Fold3: mtry=13
## - Fold3: mtry=13
## + Fold3: mtry=25
## - Fold3: mtry=25
## + Fold4: mtry= 2
## - Fold4: mtry= 2
## + Fold4: mtry=13
## - Fold4: mtry=13
## + Fold4: mtry=25
## - Fold4: mtry=25
## + Fold5: mtry= 2
## - Fold5: mtry= 2
## + Fold5: mtry=13
## - Fold5: mtry=13
## + Fold5: mtry=25
## - Fold5: mtry=25
##  ggregating results
## Selecting tuning parameters
## Fitting mtry = 2 on full training set
```

Table 6: Model Performance Metrics by mtry

mtry	ccuracy	Kappa
2	0.44573	0.09829
13	0.42641	0.10021
25	0.41951	0.09653

Table 7: Correlations with Education Level

Feature	Correlation
Matplotlib	0.088
R	0.081
Ggplot / ggplot2	0.079
Seaborn	0.077
Plotly / Plotly Express	0.071
Python	0.066
SQL	0.056
M TL B	0.056
None	-0.047
Shiny	0.041
Geoplotlib	0.034
C	0.030
Other	0.026
Bokeh	0.025
C++	0.023
Java	0.021
Javascript	-0.021
Swift	0.012
Leaflet / Folium	-0.011
None	-0.010

Feature	Correlation
Julia	0.009
N	0.007
D3js	0.006
Bash	-0.003
ltair	0.001

Conclusion The analysis explored the relationship between programming languages, libraries, and education levels through a correlation matrix and a Random Forest classification model. The correlation matrix highlighted weak positive and negative relationships between features and education levels. The strongest positive correlations were observed for Matplotlib (0.088), R (0.081), and Ggplot / ggplot2 (0.079), while the strongest negative correlations were found for None (-0.047), Javascript (-0.021), and Leaflet / Folium (-0.011). These weak correlations suggest that programming language and library preferences alone are insufficient to robustly classify education levels.

The Random Forest classification model achieved an accuracy of less than 50% (best accuracy: 44.57%), which is only slightly better than random guessing for the seven-class problem. The results further indicate that programming languages and libraries are not strong predictors of education levels, and the model struggles to differentiate between the categories with meaningful accuracy.

For future studies, it would be beneficial to explore additional variables beyond programming languages and libraries, such as demographic data, professional experience, or employment sectors, to improve predictive performance. Incorporating these factors may provide a more comprehensive understanding of the relationship between technical skillsets and education levels. This expanded approach could yield stronger insights and enhance the classification accuracy.

4. Conclusion

Our comprehensive analysis of programming language usage in the 2020 Kaggle Machine Learning and Data Science Survey reveals several key insights about programming language patterns and educational influences. The numerical analysis shows that programming experience varies significantly across educational backgrounds, with doctoral and master's degree holders using an average of 1.97 languages compared to 1.12 languages for those without formal education past high school. Correlation analysis revealed significant positive relationships between education level and advanced language tools, particularly for visualization libraries like Matplotlib (0.088) and statistical computing languages like R (0.081).

Through demographic analysis, we uncovered distinct patterns in programming language preferences across educational levels. Notably, Python emerged as the universal language of choice, with adoption rates exceeding 80% among advanced degree holders, while specialized tools like R and SQL showed stronger correlations with higher education levels. The chi-squared test (X-squared = 1481.8, df = 72, p-value < 2.2e-16) confirmed these strong relationships between educational background and language selection.

Our clustering analysis identified five distinct groups of early-career practitioners, revealing two primary career development paths: deep specialization (42.7% focusing primarily on Python) versus broad technical exposure (12.5% adopting multiple languages). This finding suggests that most early-career professionals opt for specialization rather than diversification in their programming toolkit.

Several limitations should be considered when interpreting these results. First, our sample, while large at 20,036 respondents, represents only Kaggle platform users, potentially introducing selection bias toward data science practitioners. Second, the survey's October 2020 timing may not reflect current trends in this rapidly evolving field. dditionally, the self-reported nature of programming language usage may not perfectly align with actual development practices.

These findings have important implications for educational institutions and organizations. For educational planning, the data suggests a need for differentiated programming language curricula based on academic level and career stage. For organizations, understanding the relationship between educational background and language proficiency can inform both hiring practices and technical infrastructure decisions. The strong preference for Python across all educational levels suggests prioritizing Python-based resources while ensuring support for specialized tools like R and SQL at advanced levels.

Future research could benefit from examining these patterns across different time periods and professional sectors to better understand how programming language preferences evolve throughout academic and professional careers. dditionally, investigating the relationship between language choices and actual job performance could provide valuable insights for curriculum development and hiring decisions.