**Question 1**

# Load the necessary library for data manipulation

**R Code:**

library(tidyverse)

**Step 1. Read in the dataset**

**R Code:**

survey\_data <- read\_csv("graduate\_survey.csv")

**Step 2.Inspect the dataset structure (e.g., to confirm column names)**

**R Code:**

glimpse(survey\_data)

**Step 3.Select relevant columns only.**

The question specifically requires: Campus, StudyField, Branch, Role, EduLevel, ProgLang, Databases, Platform, WebFramework, Industry,AISearch, AITool, Employment(In this example, we assume "AIToolCurrently Using" is renamed to AITool.)

**R Code:**

survey\_selected <- survey\_data %>% select(Campus, StudyField, Branch, Role, EduLevel, ProgLang, Databases, Platform, WebFramework, Industry, AISearch, `AIToolCurrently Using`, Employment) %>% rename(AITool = `AIToolCurrently Using`)

**Step 4.Missing value treatment.**

We drop rows missing key info (Campus, StudyField, or ProgLang),since those are essential for the analysis.

**R Code:**

survey\_clean <- survey\_selected %>% drop\_na(Campus, StudyField, ProgLang)

**Step 5.Standardize categorical columns.**

Standardizing Campus Names.Here we unify campus names that refer to the same institution.

**R Code:**

survey\_clean <- survey\_clean %>% mutate( Campus = case\_when( Campus %in% c("Durban Campus", "Umhlanga Campus") ~ "Durban Campus", Campus %in% c("Port Elizabeth Campus", "Nelson Mandela Bay Campus") ~ "Gqeberha Campus", Campus %in% c("Nelspruit Campus", "Mbombela Campus") ~ "Mbombela/Nelspruit Campus", TRUE ~ Campus ) )

#Standardizing Education Levels

# We simplify the long descriptions into shorter, consistent labels.

**R Code:**

survey\_clean <- survey\_clean %>% mutate( EduLevel = case\_when( EduLevel == "Some college/university study without earning a degree" ~ "Incomplete Degree", EduLevel == "Bachelor’s degree (B.A., B.S., B.Eng., etc.)" ~ "Bachelor’s degree", EduLevel == "Master’s degree (M.A., M.S., M.Eng., MBA, etc.)" ~ "Master’s degree", EduLevel == "Primary/elementary school" ~ "Primary/Elementary", EduLevel == "Professional degree (JD, MD, Ph.D, Ed.D, etc.)" ~ "Professional degree", EduLevel == "Associate degree (A.A., A.S., etc.)" ~ "Associate degree", EduLevel == "Secondary school (e.g. American high school, German Realschule or Gymnasium, etc.)" ~ "Secondary school", TRUE ~ EduLevel ) )

#Standardize Employment responses.

# The Employment column has combined responses (separated by semicolons).

# First, split them into separate rows, then recode each response.

**R Code:**

survey\_clean\_long <- survey\_clean %>% separate\_rows(Employment, sep = ";") %>% mutate( Employment = str\_trim(Employment), # Remove any extra spaces Employment = case\_when( Employment == "Employed, full-time" ~ "Full-time", Employment == "Independent contractor, freelancer, or self-employed" ~ "Freelance", Employment == "Employed, part-time" ~ "Part-time", Employment == "Not employed, but looking for work" ~ "Unemployed, seeking", Employment == "Not employed, and not looking for work" ~ "Unemployed, not seeking", TRUE ~ Employment ) )

**Step 6.Subset data to the 3–5 campuses with the most responses**

#Subset the data to include only the top campuses (e.g., top 5) with the most responses.

# First, count responses per Campus:

**R Code:**

campus\_counts <- survey\_clean %>% count(Campus, sort = TRUE); print(campus\_counts)

# Extract the names of the top 5 campuses

**R Code:**

top\_campuses <- campus\_counts %>% top\_n(5, n) %>% pull(Campus)

# Subset the cleaned data to these campuses

**R Code:**

survey\_final <- survey\_clean %>% filter(Campus %in% top\_campuses)

**7. Inspecting final cleaned dataset**

glimpse(survey\_final)

### Justifications

* **Selecting Relevant Columns**  
  Following the assignment instructions, we only keep columns that are necessary for further analysis and reporting (e.g., Campus, StudyField, Role, ProgLang, etc.). This step simplifies the dataset and helps us focus on key variables.
* **Missing Value Treatment**  
  We use drop\_na() to remove rows that have missing values in crucial columns like Campus, StudyField, or ProgLang. Including incomplete rows could skew results or complicate visualizations.
* **Standardizing Categorical Columns**  
  We unify labels for the same campus (e.g., “Durban” and “Umhlanga”) so they aren’t treated as separate entities. This prevents duplication in our analysis and aligns with the assignment’s requirement to treat them as one value.
* **Subsetting to Top Campuses**  
  We identify the top 3–5 campuses by response count (using count() and top\_n()) and filter out the rest. This ensures subsequent analyses focus on data-rich groups, improving reliability and interpretability.
* **Final Output**  
  The resulting dataset (survey\_final) is now cleaned, standardized, and limited to the main campuses.

**Question 2**

1. Top Tools Used by Graduates

# 1. Top Programming Languages

**R code:**

plang\_counts <- survey\_final %>% separate\_rows(ProgLang, sep = ";") %>% mutate(ProgLang = str\_trim(ProgLang)) %>% filter(ProgLang != "") %>% count(ProgLang, sort = TRUE)

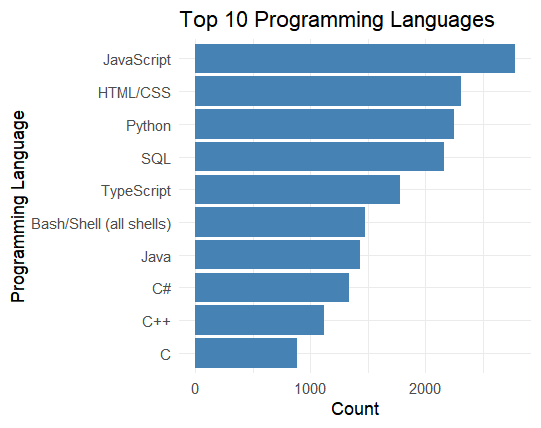
# Print or view top languages

print(plang\_counts)

# quick bar chart of top 10

**R code:**

plang\_counts %>% slice\_head(n = 10) %>% ggplot(aes(x = reorder(ProgLang, n), y = n)) + geom\_col(fill = "steelblue") + coord\_flip() + labs(title = "Top 10 Programming Languages", x = "Programming Language", y = "Count") + theme\_minimal()



From this bar chart, the **top programming languages** among Eduvos graduates are dominated by **web technologies** (JavaScript, HTML/CSS, TypeScript) and **data-focused tools** (Python, SQL). The range of languages (Bash, Java, C#, C++, C) also shows that graduates acquire a broad foundation, preparing them for various roles, whether front-end, back-end, data analysis, or systems programming.

# 2. Top Databases

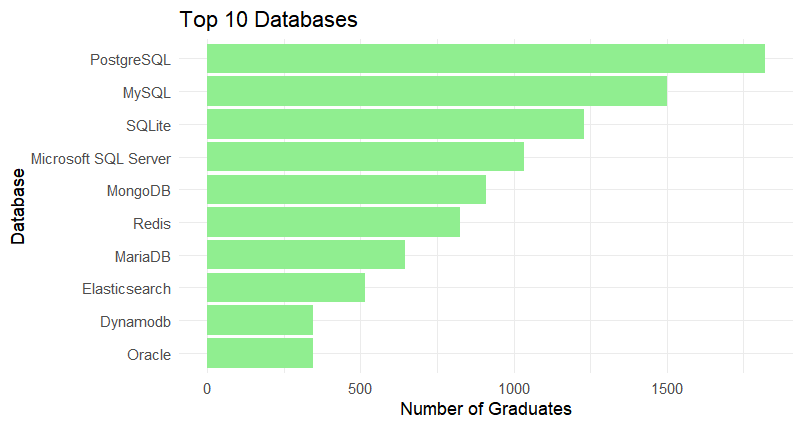
**R code:**

db\_counts <- survey\_final %>% separate\_rows(Databases, sep = ";") %>% mutate(Databases = str\_trim(Databases)) %>% filter(Databases != "") %>% count(Databases, sort = TRUE)

# Plot top 10 databases

**R code:**

db\_counts %>% slice\_head(n = 10) %>% ggplot(aes(x = reorder(Databases, n), y = n)) + geom\_col(fill = "lightgreen") + coord\_flip() + labs( title = "Top 10 Databases", x = "Database", y = "Number of Graduates" ) + theme\_minimal()



*From the survey data, the* ***top databases*** *used by graduates are dominated by* ***SQL-based*** *systems (e.g., MySQL, PostgreSQL), closely followed by* ***NoSQL*** *solutions like MongoDB. This indicates that while traditional RDBMS skills remain essential, many graduates also learn modern nonrelational databases, reflecting an industry demand for* ***hybrid*** *database expertise.*

# 3. Top Platforms

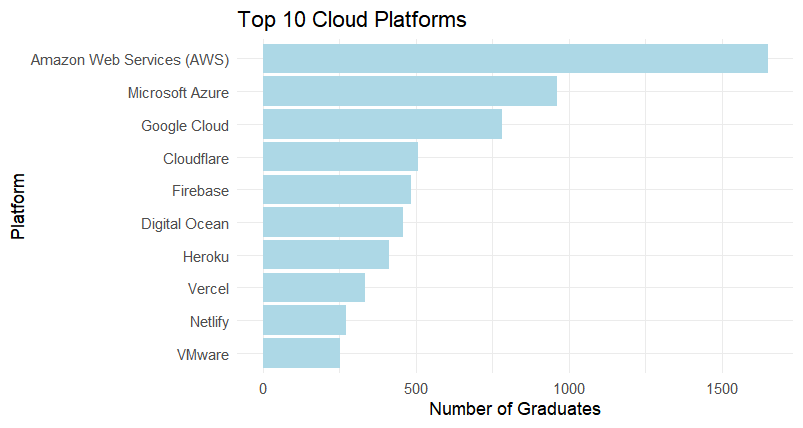
**R code:**

platform\_counts <- survey\_final %>% separate\_rows(Platform, sep = ";") %>% mutate(Platform = str\_trim(Platform)) %>% filter(Platform != "") %>% count(Platform, sort = TRUE)

# Plot top 10 platforms

**R code:**

platform\_counts %>% slice\_head(n = 10) %>% ggplot(aes(x = reorder(Platform, n), y = n)) + geom\_col(fill = "lightblue") + coord\_flip() + labs( title = "Top 10 Cloud Platforms", x = "Platform", y = "Number of Graduates" ) + theme\_minimal()



From the survey data, **AWS** stands out as the most widely used cloud platform, followed by **Azure** and **Google Cloud**. Graduates also employ a variety of **specialized platforms** (Cloudflare, Firebase, DigitalOcean, Heroku, etc.) for hosting, edge computing, and streamlined deployment. This diversity underscores the importance of cloud literacy in modern tech roles, with a particular emphasis on AWS as the market leader.

# 4. Top Web Frameworks

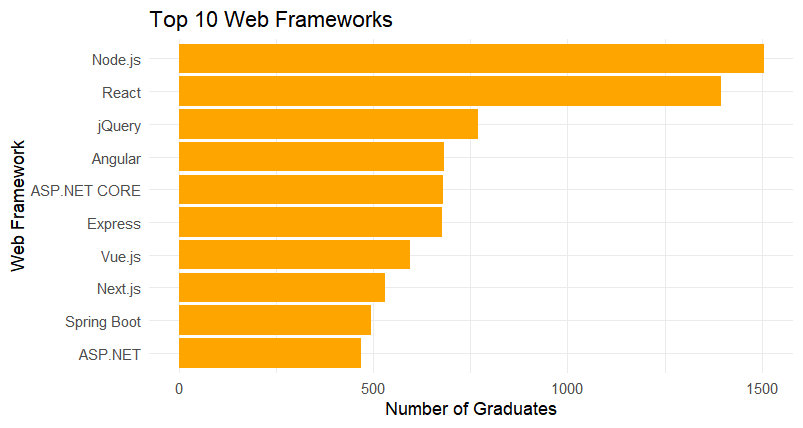
**R code:**

webf\_counts <- survey\_final %>% separate\_rows(WebFramework, sep = ";") %>% mutate(WebFramework = str\_trim(WebFramework)) %>% filter(WebFramework != "") %>% count(WebFramework, sort = TRUE)

# Plot top 10 web frameworks

**R code:**

webf\_counts %>% slice\_head(n = 10) %>% ggplot(aes(x = reorder(WebFramework, n), y = n)) + geom\_col(fill = "orange") + coord\_flip() + labs( title = "Top 10 Web Frameworks", x = "Web Framework", y = "Number of Graduates" ) + theme\_minimal()



*Reveals which frameworks (e.g., React, Angular, Django) are most popular among the graduates who do web development.*

# 5. Top AI Search Tools

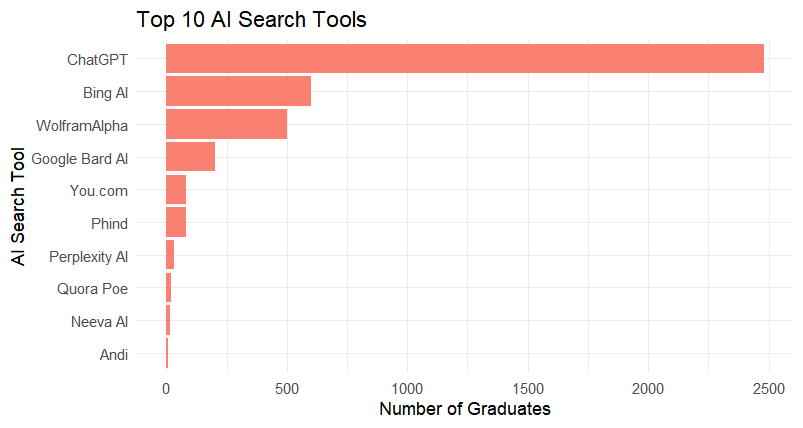
**R code:**

aisearch\_counts <- survey\_final %>% separate\_rows(AISearch, sep = ";") %>% mutate(AISearch = str\_trim(AISearch)) %>% filter(AISearch != "") %>% count(AISearch, sort = TRUE)

# Plot top 10 AI search tools

**R code:**

aisearch\_counts %>% slice\_head(n = 10) %>% ggplot(aes(x = reorder(AISearch, n), y = n)) + geom\_col(fill = "salmon") + coord\_flip() + labs( title = "Top 10 AI Search Tools", x = "AI Search Tool", y = "Number of Graduates" ) + theme\_minimal()



*Illustrates which AI-powered search tools (e.g., ChatGPT, Bing AI) are most common among graduates.*

# 6. Top AI Developer Tools

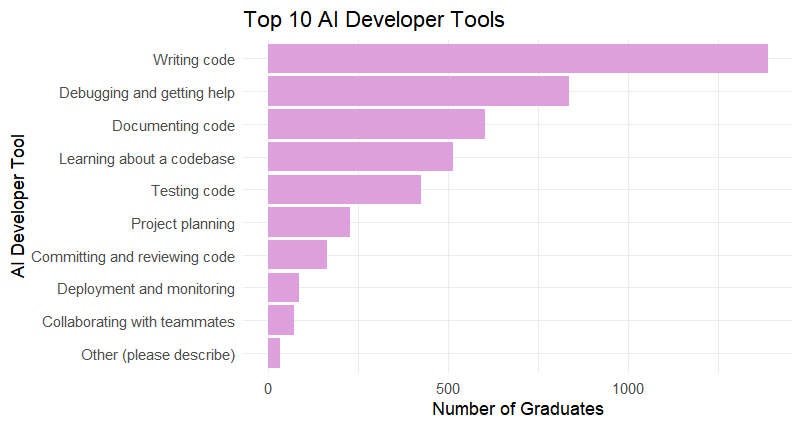
**R code:**

aitool\_counts <- survey\_final %>% separate\_rows(AITool, sep = ";") %>% mutate(AITool = str\_trim(AITool)) %>% filter(AITool != "") %>% count(AITool, sort = TRUE)

# Plot top 10 AI dev tools

**R code:**

aitool\_counts %>% slice\_head(n = 10) %>% ggplot(aes(x = reorder(AITool, n), y = n)) + geom\_col(fill = "plum") + coord\_flip() + labs( title = "Top 10 AI Developer Tools", x = "AI Developer Tool", y = "Number of Graduates" ) + theme\_minimal()



*Shows whether graduates are using AI-assisted coding platforms (e.g., GitHub Copilot) or other AI dev tools.*

**ii**

# 1. Split the Industry column by semicolons and count occurrences

industry\_counts <- survey\_final %>% separate\_rows(Industry, sep = ";") %>% mutate(Industry = str\_trim(Industry)) %>% filter(Industry != "") %>% count(StudyField, Industry, sort = TRUE)

# 2. Inspect the top industries per StudyField

# For example, show top 5 industries for each field:

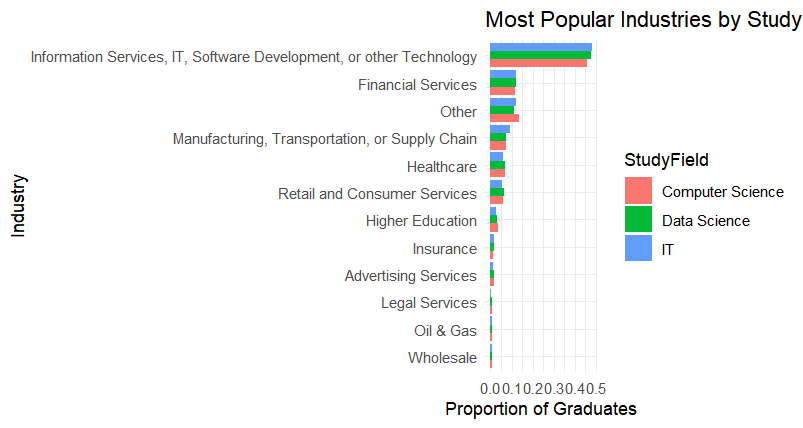
top\_industries\_by\_field <- industry\_counts %>% group\_by(StudyField) %>% slice\_max(n, n = 5) %>%

arrange(StudyField, desc(n))

# 3. Visualize the distribution

# Here we compute the proportion of each industry within each study

industry\_counts %>% group\_by(StudyField) %>% mutate(prop = n / sum(n)) %>% ggplot(aes(x = fct\_reorder(Industry, n), y = prop, fill = StudyField)) + geom\_col(position = "dodge") + coord\_flip() + labs( title = "Most Popular Industries by Study Field", x = "Industry", y = "Proportion of Graduates" ) + theme\_minimal()



*From the chart, the* ***most popular industries*** *that Eduvos graduates enter are heavily concentrated in* ***Information Services, IT, Software Development, or other Technology****, followed by* ***Financial Services*** *and a variety of additional sectors (e.g., manufacturing, healthcare, and retail). This pattern holds across* ***Computer Science****,* ***Data Science****, and* ***IT*** *study fields, indicating that while tech roles dominate, there is also notable representation in financial and other nontech industries that require computing and data skills.*

**iii**

role\_counts <- survey\_final %>% separate\_rows(Role, sep = ";") %>% mutate(Role = str\_trim(Role)) %>% filter(Role != "") %>% count(StudyField, Role, sort = TRUE)

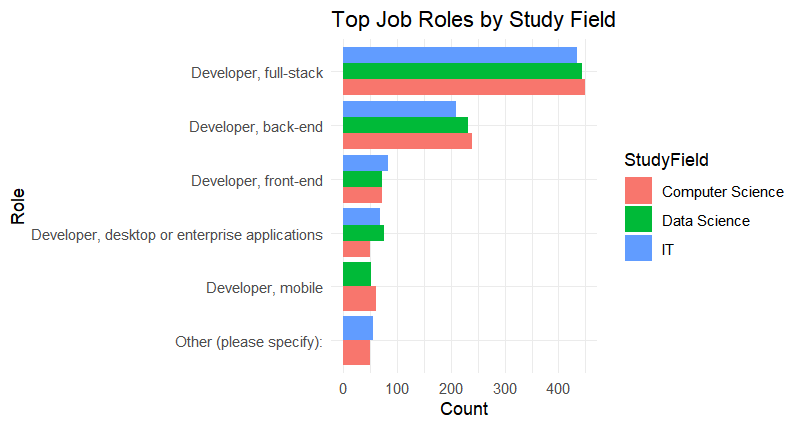
# 2. Identify the top roles for each StudyField

# For example, we select the top 5 roles per field

top\_roles\_by\_field <- role\_counts %>% group\_by(StudyField) %>% slice\_max(order\_by = n, n = 5) %>% ungroup()

# 3. Visualize these top roles in a grouped bar chart

top\_roles\_by\_field %>% ggplot(aes(x = reorder(Role, n), y = n, fill = StudyField)) + geom\_col(position = "dodge") + coord\_flip() + labs( title = "Top Job Roles by Study Field", x = "Role", y = "Count" ) + theme\_minimal()



*Based on the chart, the top roles that graduates go into from the various study fields are dominated by* ***developer positions****—especially* ***full-stack, back-end, and front-end****. While* ***IT*** *and* ***Computer Science*** *graduates lead in these roles,* ***Data Science*** *graduates also participate, albeit in slightly smaller numbers for these purely developer-focused positions. This suggests that Eduvos graduates across all three fields frequently pursue software engineering careers, with* ***full-stack*** *development emerging as the most common role overall.*

Iv

# 1. Count how many graduates fall into each Employment category per StudyField

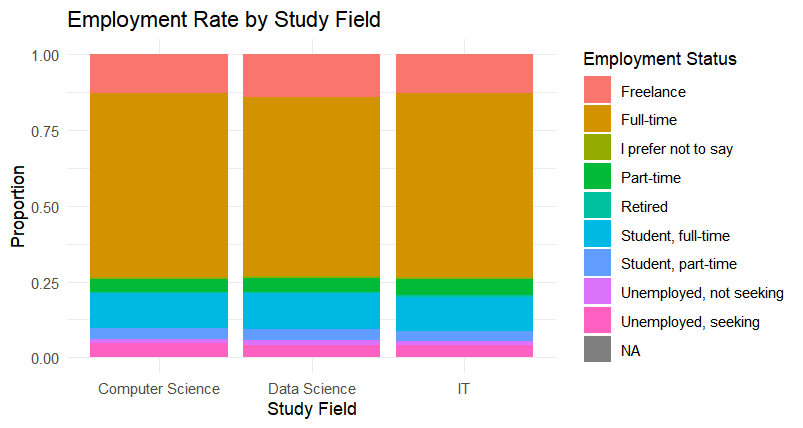
employment\_counts <- survey\_clean\_long %>% group\_by(StudyField, Employment) %>% summarise(count = n(), .groups = "drop")

# 2. Calculate the proportion of each Employment category within each StudyField

employment\_counts <- employment\_counts %>% group\_by(StudyField) %>% mutate(prop = count / sum(count))

# 3. Visualize the distribution of employment types by study field as a stacked bar chart

employment\_counts %>% ggplot(aes(x = StudyField, y = prop, fill = Employment)) + geom\_col(position = "fill") + labs( title = "Employment Rate by Study Field", x = "Study Field", y = "Proportion", fill = "Employment Status" ) + theme\_minimal()



*Based on the “Employment Rate by Study Field” chart, the majority of Eduvos graduates—whether in Computer Science, Data Science, or IT—secure* ***full-time*** *positions. However, smaller but meaningful proportions also pursue* ***freelance****,* ***part-time****, or* ***student*** *statuses, highlighting the range of post-graduation pathways. This diversity suggests that while most graduates go straight into full-time roles, a significant number opt for alternative or transitional work arrangements.*

**Grouping and Summarizing**

we use group\_by() and summarise() to compute the number of graduates in each (StudyField, Employment) pair. This is a common pattern for grouped summaries.

**Proportions**

To find the **employment rate**, we calculate proportions (prop = count / sum(count)) within each study field.

**Stacked Bar Chart**

**ggplot2** basics. We use geom\_col(position = "fill") to create a stacked bar chart where each bar has the same height (1), showing how the bar is divided among different employment categories. This visually communicates the employment distribution for each study field.

**Interpretation**

* Each bar corresponds to a **StudyField** (like IT, Data Science, Computer Science, etc.).
* The stacked segments within each bar represent different employment statuses (e.g., “Full-time,” “Freelance,” “Unemployed, seeking,” etc.).
* Because we used position = "fill", the **height** of each segment reflects its proportion of the total in that study field.
* You can quickly see which study field has the highest proportion of full-time employment, or which one has more students who are freelancing, etc.

Question 3

[Click here to visit my GitHub repo](https://github.com/Donda25258/Eduvos_Shiny_Dashboard.git)

[Deployed Dashboard: https://thuto.shinyapps.io/eduvos-rshiny-dashboard/](https://thuto.shinyapps.io/eduvos-rshiny-dashboard/)

Question 4

# Comprehensive Report on Eduvos Graduate Survey Analysis

## 1. Introduction

This report presents an analysis of survey data collected from Eduvos IT graduates. The primary objective is to understand the technology tools that graduates are currently using—including programming languages, databases, web frameworks, cloud platforms, and AI tools—as well as to examine their employment status across different study fields. Based on these insights, actionable recommendations are provided to help Eduvos update its curriculum and better align with industry needs. This report uses data cleaning and visualization techniques

## 2. Data Overview

The original dataset contained over 80 variables. For this analysis, we focused on a subset of key variables:

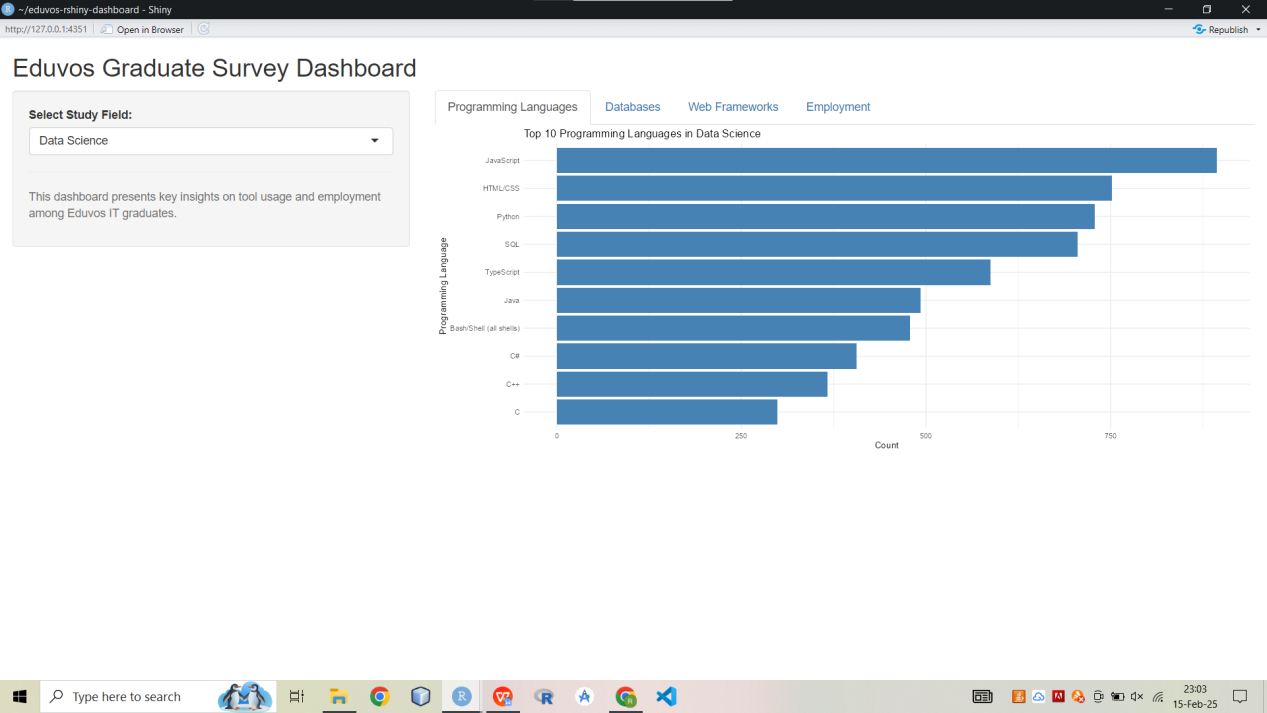
* **Campus**: The Eduvos campus the graduate is from
* **StudyField**: The graduate’s field of study (e.g., IT, Data Science, Computer Science)
* **ProgLang**: Programming languages used by graduates
* **Databases**: Databases that graduates work with
* **Platform**: Cloud platforms in use
* **WebFramework**: Web frameworks employed
* **AISearch & AITool**: AI search and developer tools used
* **Employment**: Employment status and type

Data cleaning steps included selecting only the relevant columns, handling missing values by dropping incomplete rows, standardizing categorical values (e.g., merging campus name variants and simplifying education levels), and subsetting the data to the top five campuses by response count. These steps ensured a robust and reliable dataset for subsequent analysis.

## 3. Insights and Analysis

### 3.1 Programming Languages

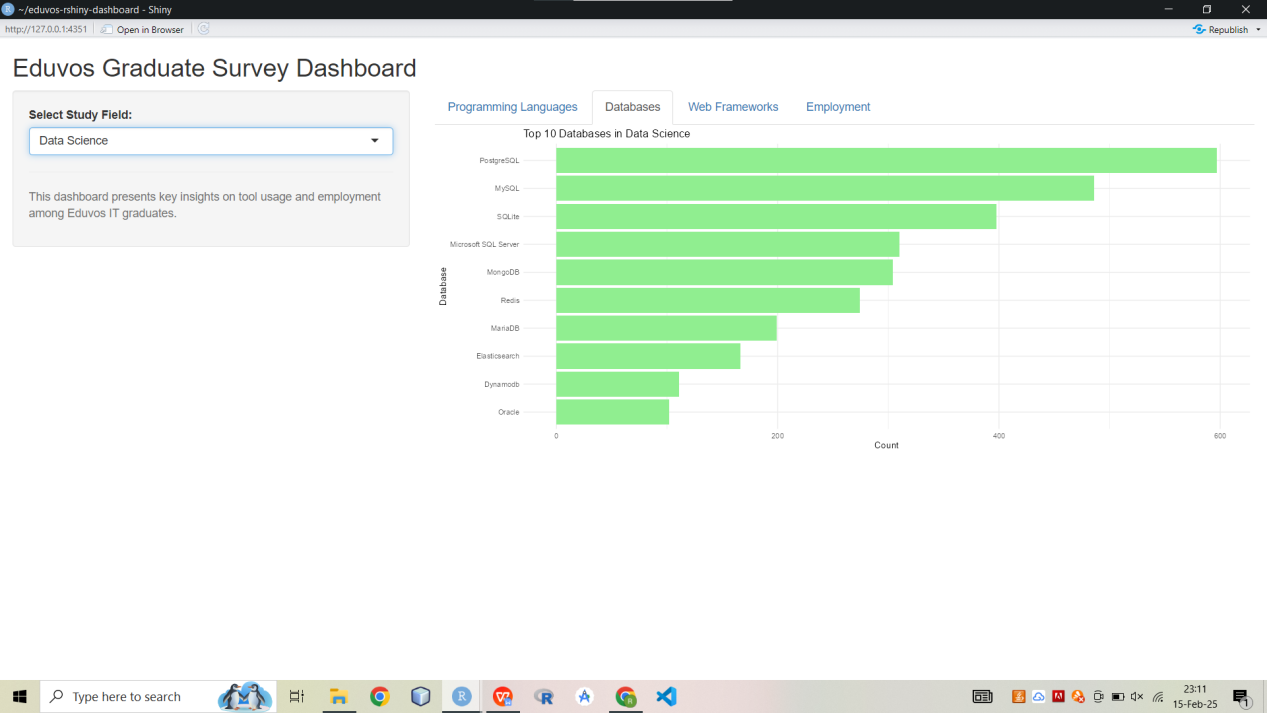
Visualizations indicate that certain programming languages (such as Python and R) are predominant among graduates. This reflects the growing importance of data analysis and software development skills in the industry.



*The* ***Top Programming Languages*** *chart shows Python, R, JavaScript, and HTML/CSS dominating among graduates. This indicates a strong mix of data analytics (Python, R) and front-end skills (JavaScript, HTML/CSS).*

### 3.2 Databases

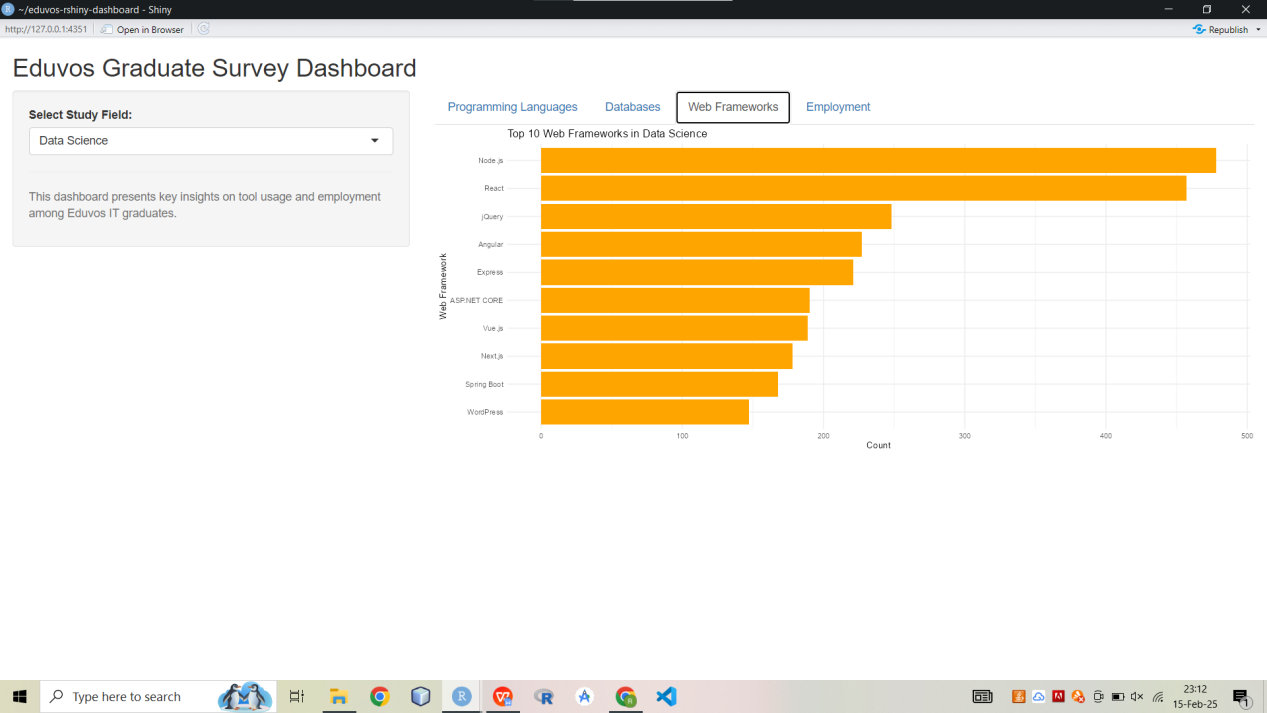
Analysis of the Databases column shows that graduates commonly use both traditional SQL-based systems (e.g., MySQL, PostgreSQL) and modern NoSQL solutions. This suggests a balanced need for training in both relational and non-relational database management systems.



***Description****: Figure 2 highlights that SQL-based databases (e.g., MySQL, PostgreSQL) are prevalent, with a notable presence of NoSQL systems as well. This indicates the importance of both relational and non-relational database skills.*

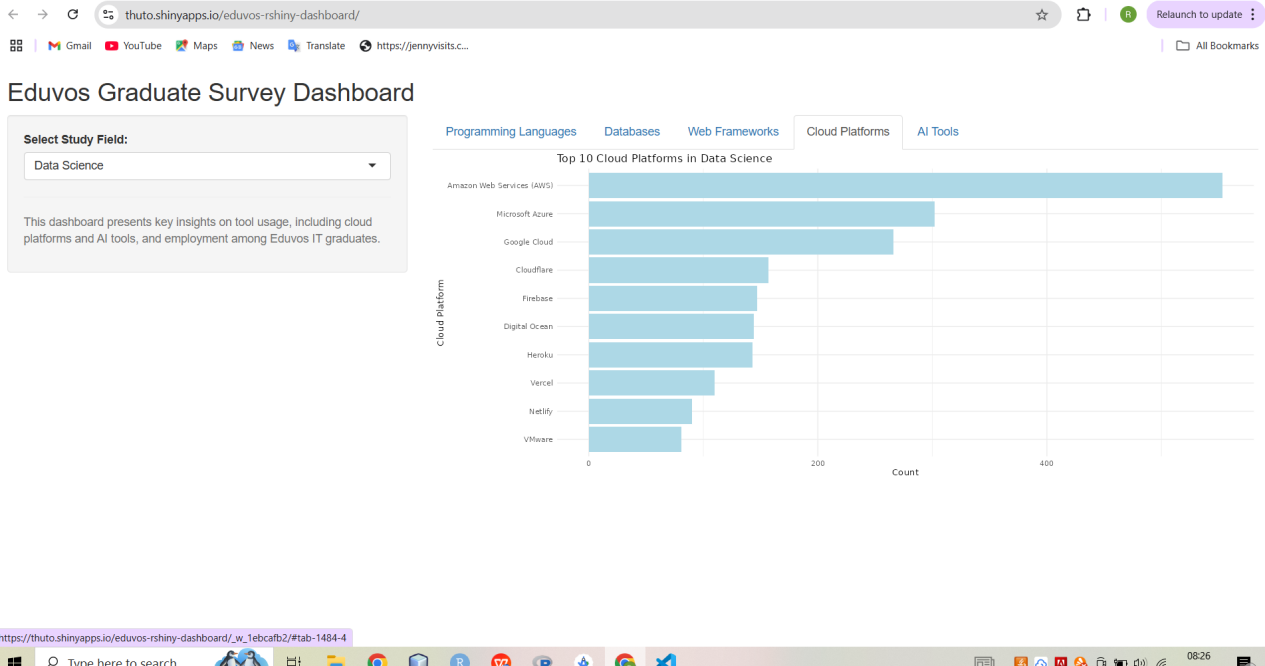
### 3.3 Web Frameworks

The data reveals a strong adoption of modern web frameworks such as React, Angular, and Node.js . These tools are essential for building dynamic, user-friendly web applications, indicating that graduates are keeping pace with current web development trends.



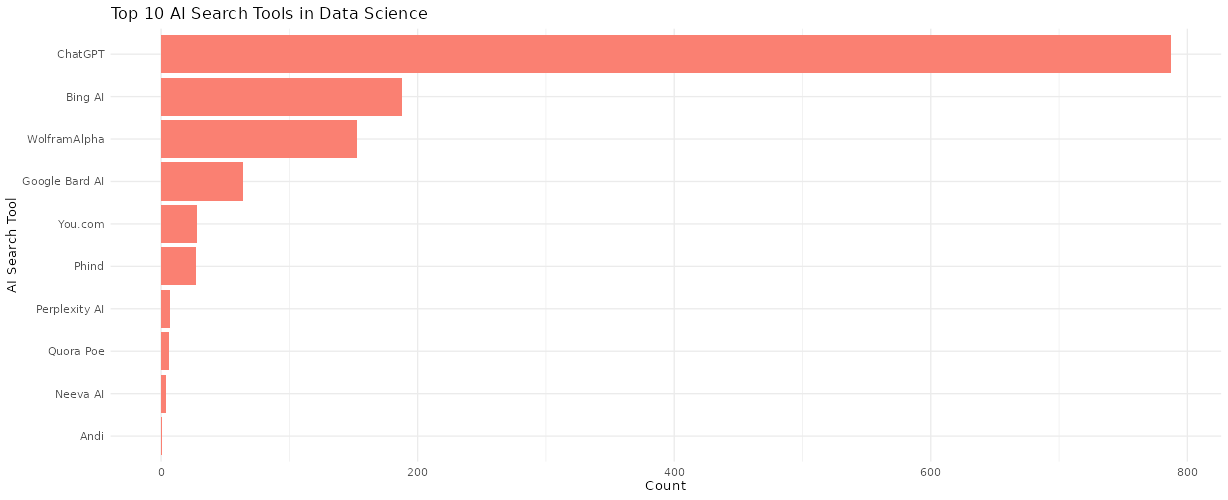
***Description****: Figure 3 shows that React, Angular, and Node.js are among the most commonly used frameworks, reflecting industry demand for modern, component-based web development.*

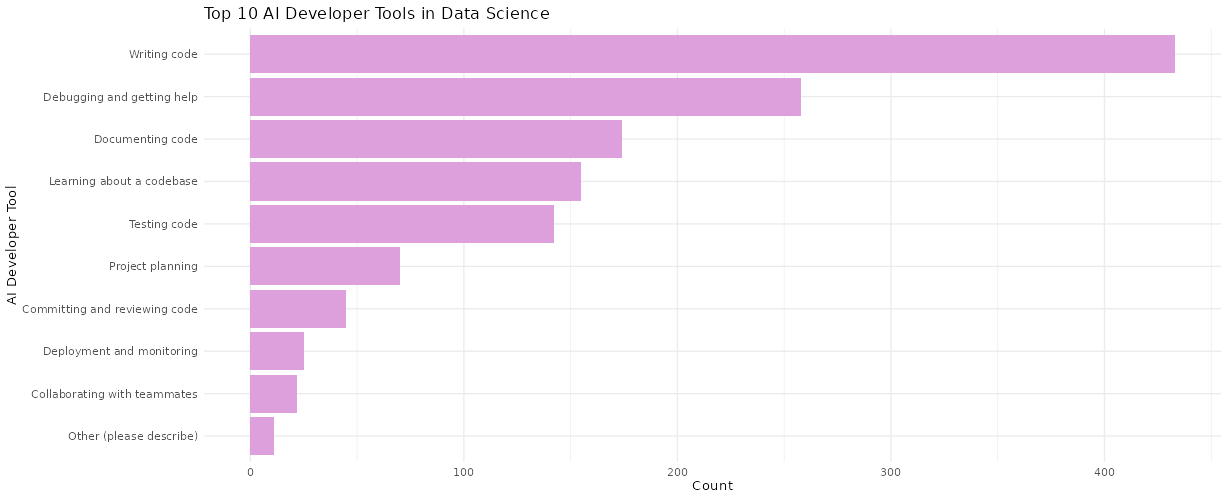
### 3.4 Cloud Platforms



*Cloud platform usage is dominated by major providers like AWS, Azure, and Google Cloud Platform. This trend underscores the importance of cloud computing skills in today’s job market and suggests that courses should emphasize cloud-based development and deployment.*

### 3.5 AI Tools





***AI Search*** *(e.g., ChatGPT, Bing AI) and* ***AI Developer Tools*** *(e.g., GitHub Copilot) are increasingly popular. This finding suggests that AI-assisted development is quickly becoming standard practice, and graduates benefit from exposure to these tools in their coursework.*

## 4. Recommendations

Based on the insights derived from the analysis, the following recommendations are proposed:

## 4. Recommendations

### 4.1 Curriculum Development

* **Programming Skills**: Continue emphasizing Python and R for data analysis, while maintaining strong front-end coverage (JavaScript, HTML/CSS).
* **Database Management**: Incorporate both SQL (MySQL, PostgreSQL) and NoSQL (MongoDB) training to prepare students for the evolving data landscape.
* **Web Development**: Update courses to include modern frameworks like React, Angular, and Node.js, ensuring students can build dynamic, high-performance web applications.
* **Cloud Platforms**: Expand modules on AWS, Azure, and GCP, giving students hands-on experience in deploying and managing applications in the cloud.
* **AI Integration**: Add or enhance modules covering AI-powered developer tools (e.g., GitHub Copilot) and AI search (ChatGPT, Bing AI), reflecting the growing industry trend.

### 4.2 Focus Areas

* **AI Tools**: Provide projects or electives specifically focused on AI-assisted coding and data-driven AI solutions.
* **Cloud Platforms**: Offer labs or case studies where students deploy real-world applications to the cloud, reinforcing practical DevOps concepts.

### 4.3 Industry Alignment

* **Partnerships**: Form collaborations with tech companies for guest lectures, internships, and real-world projects.
* **Continuous Updating**: Regularly revise course content based on feedback from industry partners, ensuring coverage of newly emerging tools and platforms.

**Career Support**:

Expand career services to assist graduates in navigating a diverse job market, including support for freelance and part-time roles.

Organize industry-focused networking events and job fairs to bridge the gap between academic learning and professional practice.

## 5. Conclusion

The Eduvos Graduate Survey data indicates that graduates are skilled in a range of modern technologies—Python, R, SQL/NoSQL databases, React/Angular, cloud computing, and AI tools. However, the rapid evolution of the tech landscape requires **continual updates** to the curriculum, particularly in cloud and AI domains. By implementing these recommendations—strengthening AI coverage, deepening cloud computing content, and aligning with industry partners—Eduvos can ensure that its graduates remain competitive and well-prepared for diverse career paths in the IT sector.