All Tools Data Analysis Project

During our training at

National Telecommunication Insitute And ITIDA

Full Data Analysis Process with Python

Full Data Analysis Process with SQL

Full Data Analysis Process with Excel

Our Dataset Talk About

This dataset provides a comprehensive overview of global student migration patterns, capturing the educational and professional journeys of 5,000 international students. Each record represents an individual student and includes detailed information about their academic background, mobility decisions, and career outcomes.

Key dimensions covered in the dataset include:

- **Demographics of Movement: Origin and destination countries/cities of students.**
- **Academic Pathways: University name, course of study, field of specialization, year of enrollment, graduation year, and academic performance (GPA/score).**
- **Motivations and Support: Reasons for enrollment abroad (e.g., higher rankings, scholarships, job opportunities, quality of life, political stability) and whether financial scholarships were received.**
- **Language and Visa Information: Language proficiency tests and scores, visa status during study, and post-graduation visa types.**
- **Career Outcomes: Placement status upon graduation, placement country, employing company, and starting salary in USD.**

Our Full Data Analysis Process with Python

We carried out a comprehensive *end-to-end data analysis using Python* , transforming raw student migration records into clear insights and recommendations.

Process Overview:

1. **Data Loading & Initial Inspection**
   - **Imported essential Python libraries: pandas, numpy, matplotlib, seaborn.**
   - **Uploaded and read the dataset (5,000 student records across 20 variables).**
   - **Performed initial checks: structure, data types, summary statistics.**
2. **Data Cleaning**
   - **Removed duplicates and standardized inconsistent categorical entries.**
   - **Handled missing values logically:**
     - **Students marked "Not Placed" → placement company & country set to NaN, starting salary corrected to NaN instead of zero.**
     - **Language proficiency missing → test score also marked as NaN.**
   - **Normalized text fields (title case, no extra spaces).**
   - **Replaced unknown categories with "Unknown" for clarity.**
3. **Feature Engineering**
   - **Added scholarship_flag and placement_flag for easier filtering and analysis.**
   - **Created new analytical dimensions such as average salary by scholarship status, placement rates by language test, etc.**

```
[ ] # Replace "None" or empty strings with NaN
    df = df.replace(["None", ""], np.nan)
```

```
[ ] # Handle students who are not placed
    mask_not_placed = df["placement_status"].str.lower() == "not place
    df.loc[mask_not_placed, ["placement_country", "placement_company"
```

```
[ ] # Replace starting salary 0 with NaN for not placed students
    df.loc[(df["placement_status"].str.lower() == "not placed") &
           (df["starting_salary_usd"] == 0), "starting_salary_usd"] =
```

```
[ ] # Replace test_score 0 with NaN if no language test
    df.loc[(df["language_proficiency_test"].isna()) &
           (df["test_score"] == 0), "test_score"] = np.nan
```

Insights

```
[ ] insights = []

    if placement_rate < 70:
        insights.append(f"Placement rate is relatively low ({placem

    if salary_difference > 0:
        insights.append("Students with scholarships have higher ave

    if placement_by_language.max() - placement_by_language.min() >
        insights.append("Language proficiency test results strongly

    print("Insights:")
```

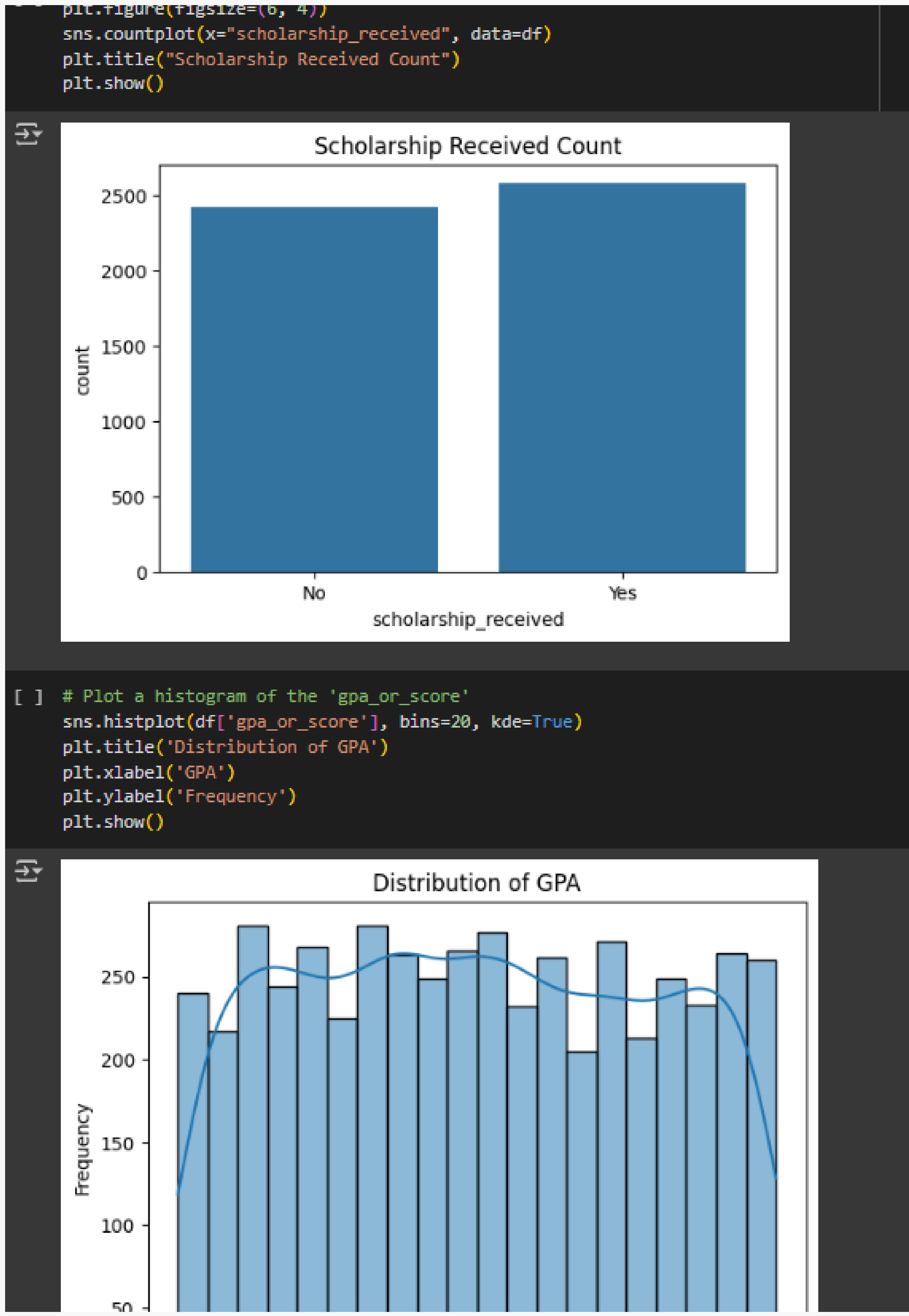
Our Full Data Analysis Process with Python

4-**Exploratory Data Analysis (EDA)**

- **Identified top destination countries and most popular fields of study.**
- **Compared scholarship vs. non-scholarship students in terms of GPA, salary, and placement success.**
- **Investigated salary distributions, GPA performance, and graduation trends.**
- **Detected correlations between GPA, test scores, and starting salaries.**

5-**Data Visualization**

- **Bar charts and pie charts for scholarships, placements, and destination countries.**
- **Histograms and boxplots to explore GPA and salary distributions.**
- **Correlation heatmap to highlight relationships between academic and employment factors.**
- **Line graphs showing enrollment growth and migration patterns over time.**



Our Full Data Analysis Process with Python

6-**Key Performance Indicators (KPIs)**

- **Placement Rate (% of students employed after graduation).**
- **Average Starting Salary (overall, with scholarship, without scholarship).**
- **Placement Success by Language Test (TOEFL, IELTS, PTE, etc.).**
- **Salary difference between scholarship holders and non-holders.**

7-**Insights & Recommendations**

- **Scholarships are strongly linked to higher placement rates and better salaries.**
- **Language proficiency significantly impacts job opportunities in destination countries.**
- **Certain fields of study (e.g., Engineering, Data Science) consistently yield higher starting salaries.**
- **Employers like Microsoft, Google, Apple, and IBM are among the top recruiters, highlighting strong industry connections.**
- **Recommended actions: expand scholarships, strengthen partnerships with employers, and support students with language training programs.**

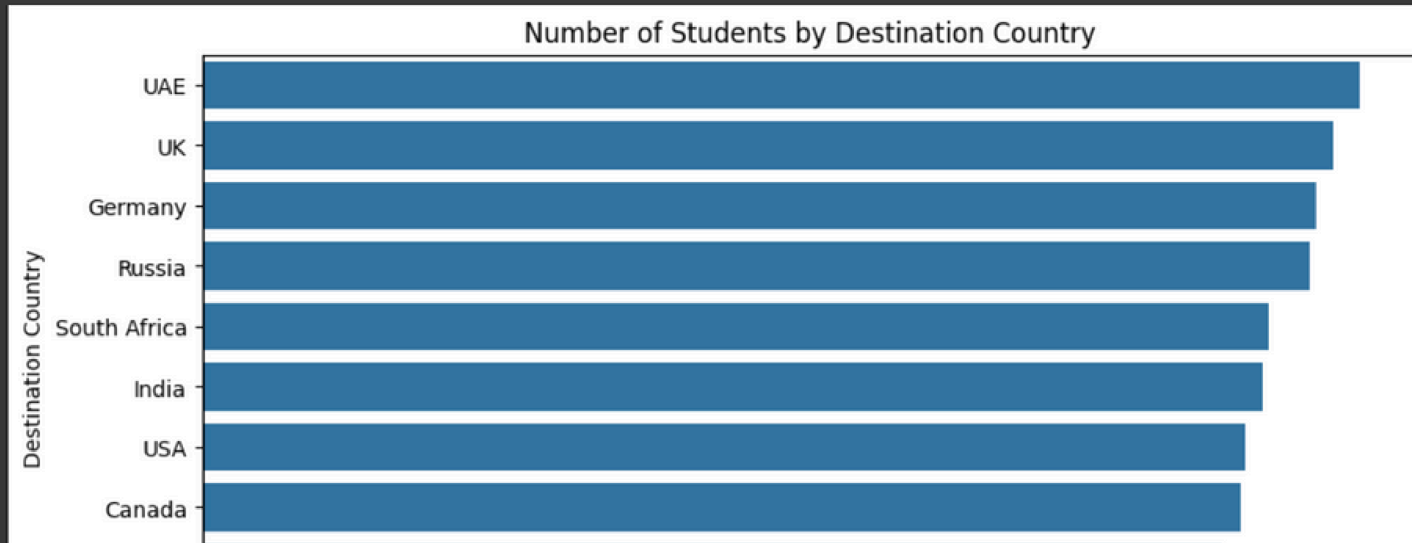
KPIs

```
# Calculate some key metrics for student placement and salaries
placement_rate = df['placed_flag'].mean() * 100
average_salary = df['starting_salary_usd'].mean()
average_salary_scholarship = df[df['scholarship_flag']]['starting_salary_usd']
average_salary_no_scholarship = df[~df['scholarship_flag']]['starting_salary_us
salary_difference = average_salary_scholarship - average_salary_no_scholarship

# Placement rate based on language proficiency test
placement_by_language = df.groupby('language_proficiency_test')['placed_flag']

# Display the KPIs
print("Key Performance Indicators (KPIs)")
print(f"Placement Rate: {placement_rate:.2f}%")
print(f"Average Starting Salary: ${average_salary:,.2f}")
print(f"Average Salary (Scholarship): ${average_salary_scholarship:,.2f}")
print(f"Average Salary (No Scholarship): ${average_salary_no_scholarship:,.2f}
print(f"Salary Difference: ${salary_difference:,.2f}")
print("Placement Rate by Language Test:")
print(placement_by_language)
```

```
# Bar chart: Number of students by destination country
plt.figure(figsize=(10,5))
sns.countplot(y='destination_country', data=df, order=df['destination_country'].value_counts().index)
plt.title("Number of Students by Destination Country")
plt.xlabel("Count")
plt.ylabel("Destination Country")
plt.show()
```

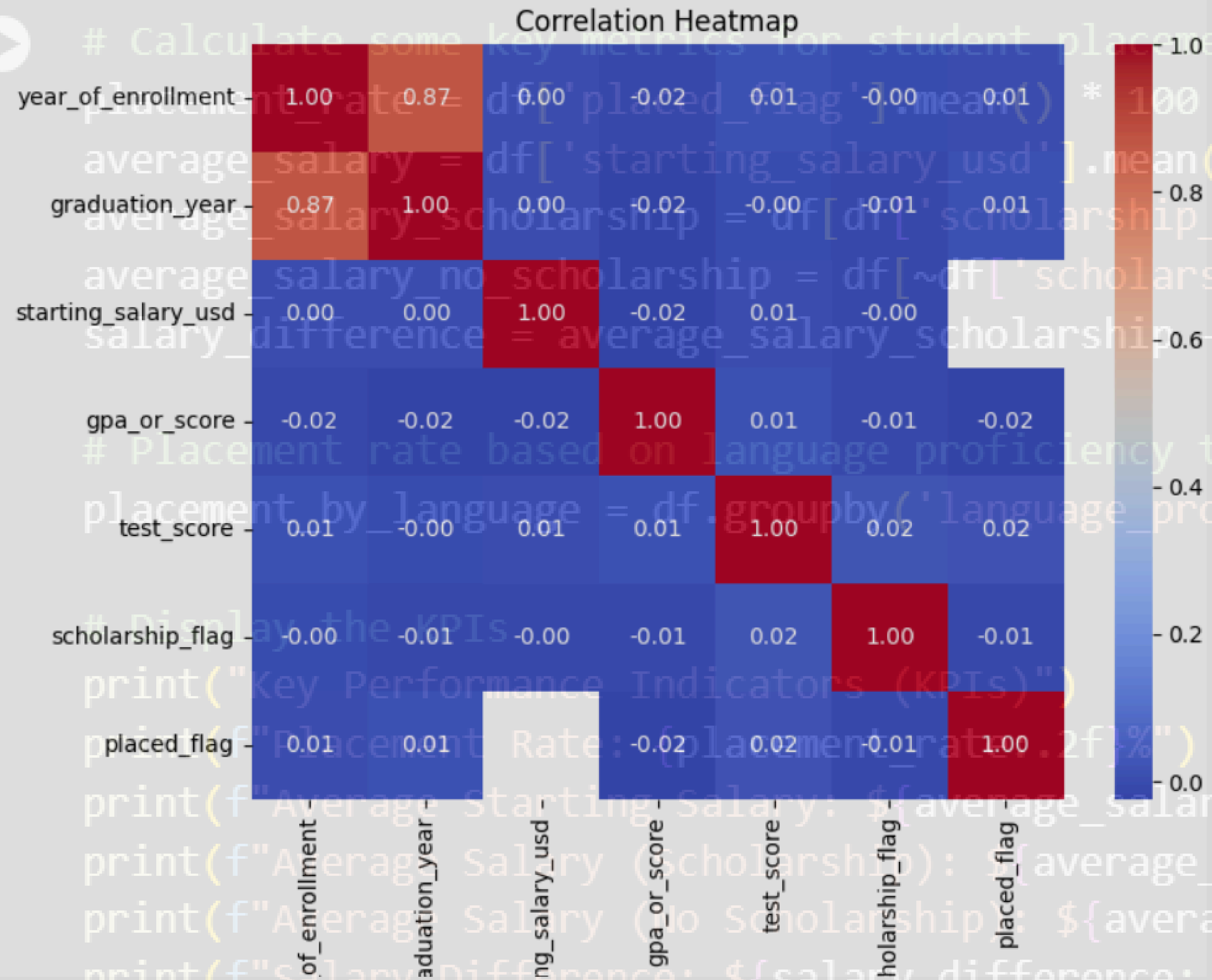


KPIs

```
# Calculate some key metrics for student placement and salaries
placement_rate = df['placed_flag'].mean() * 100
average_salary = df['starting_salary_usd'].mean()
average_salary_scholarship = df[df['scholarship_flag']]['starting_salary_usd'].mean()
average_salary_no_scholarship = df[~df['scholarship_flag']]['starting_salary_usd'].mean()
salary_difference = average_salary_scholarship - average_salary_no_scholarship

# Placement rate based on language proficiency test
placement_by_language = df.groupby('language_proficiency_test')['placed_flag'].mean() * 100

# Display the KPIs
print("Key Performance Indicators (KPIs)")
print(f"Placement Rate: {placement_rate:.2f}%")
print(f"Average Starting Salary: ${average_salary:,.2f}")
print(f"Average Salary (Scholarship): ${average_salary_scholarship:,.2f}")
print(f"Average Salary (No Scholarship): ${average_salary_no_scholarship:,.2f}")
print(f"Salary Difference: ${salary_difference:,.2f}")
print("Placement Rate by Language Test:")
print(placement_by_language)
```



Extract key insights

```
# Top 5 countries receiving most students
top_destinations = df['destination_country'].value_counts().head(5)
print("Top 5 Destination Countries:\n", top_destinations)
```

```
Top 5 Destination Countries:
 destination_country
UAE             538
UK              526
Germany         518
Russia          515
South Africa    496
Name: count, dtype: int64
```

```
# Average starting salary by field of study
avg_salary_by_field = df.groupby('field_of_study')['starting_salary_usd'].mean().sort_values(ascending=False)
print("\nAverage Starting Salary by Field of Study:\n", avg_salary_by_field)
```

```
Average Starting Salary by Field of Study:
 field_of_study
Medicine            92317.744526
Business            91072.245098
Law                 90966.996865
Social Sciences     88660.177177
Engineering         87801.701149
Computer Science    87459.235484
Arts                87277.836066
Natural Sciences    85255.031847
Name: starting_salary_usd, dtype: float64
```

Recommendations

```
recommendations = []

if placement_rate < 70:
    recommendations.append("Expand job partnerships to improve placement rate.")
if salary_difference > 0:
    recommendations.append("Increase the number of scholarships to boost salary outcomes.")
if placement_by_language.max() - placement_by_language.min() > 10:
    recommendations.append("Offer language training programs to improve placement opportuni

print("Recommendations:")
for r in recommendations:
    print(f"- {r}")
```

```
Recommendations:
- Expand job partnerships to improve placement rate.
```

Our Full Data Analysis Process with SQL

We also performed a complete data analysis process using *SQL* to ensure robust validation, efficient querying, and powerful insights from the global student migration dataset.

Steps We Followed:

1. **Data Quality & Cleaning with SQL**
    - **Missing Data Check: Used aggregate queries (COUNT, SUM CASE) to detect missing values in key columns (student ID, origin, destination, university, field of study, placement status, etc.).**
    - **Duplicates Check: Grouped by student_id to identify students with duplicate entries.**
    - **Data Standardization: For example, we updated records where destination_country was listed as "UK", "Britain", or "England" into a unified format: "United Kingdom".**
    - **Validation of GPA & Scores: Ensured GPA and test scores were positive values and checked their min, max, and average.**

```
    cast(sum(case when placement_status = 'placed' then 1 else 0 end) *
    avg(starting_salary_usd) as avg_salary
from global_student_migration
where starting_salary_usd > 0
group by case when field_of_study in ('computer science', 'engineering',
              then 'stem' else 'non-stem' end;
--------------------------------
select
    destination_country,
    language_proficiency_test,
    count(*) as test_count,
    cast(count(*) * 100.0 / sum(count(*)) over (partition by destinatio
from global_student_migration
where language_proficiency_test <> 'none'
  and destination_country in ('united kingdom', 'usa')
group by destination_country, language_proficiency_test
order by destination_country, test_count desc;
-----------------------------------
select
    avg(case when placement_status = 'placed' then gpa_or_score end) as
    avg(case when placement_status = 'not placed' then gpa_or_score end)
    avg(case when placement_status = 'placed' then test_score end) as av
    avg(case when placement_status = 'not placed' then test_score end) a
from global_student_migration
where gpa_or_score > 0 and test_score > 0;
------------------------------------
select
    university_name,
    count(*) as total,
    sum(case when placement_status = 'placed' then 1 else 0 end) as plac
    cast(sum(case when placement_status = 'placed' then 1 else 0 end) *
from global_student_migration
group by university_name
having count(*) > 5
```

Our Full Data Analysis Process with SQL

2-Exploratory Data Analysis (EDA) with SQL

wrote multiple SQL queries to explore the dataset from different perspectives:

- **Placement Analysis: Counted students by placement status (Placed vs. Not Placed) and calculated placement percentages.**
- **Enrollment Trends: Measured reasons for enrollment, enrollment growth by year, and study duration from enrollment to graduation.**
- **Origin vs. Destination: Aggregated student counts by origin countries, destination countries, and their flows.**
- **Universities & Fields of Study: Identified the top universities, most popular fields of study, and calculated placement rates per field.**

3-Key Insights & Business Intelligence with SQL

- **Scholarships & Outcomes: Compared scholarship holders vs. non-holders in terms of placement success and salaries.**
- **Placement Companies: Extracted the top hiring companies (e.g., Microsoft, Google, Apple, IBM) and calculated their average salaries offered.**
- **Field Performance: Ranked fields of study by GPA, placement rate, and starting salary.**
- **Visa Impact: Measured how different post-graduation visa types affected placement success.**
- **Language Proficiency: Compared TOEFL, IELTS, PTE, etc. and found that higher test scores correlated with higher placement rates.**

Our Full Data Analysis Process with SQL

4-KPI Calculations with SQL

Using SQL, we directly computed key metrics:

- **Placement Rate (%) = (Placed ÷ Total Students) × 100.**
- **Average Starting Salary (by field, university, company, country).**
- **Scholarship Impact on placement success and salary.**
- **STEM vs. Non-STEM Analysis showing higher placement rates and salaries in STEM fields.**
- **High Achievers Check (students with GPA ≥ 3.95) and their placement outcomes.**

5-Advanced Comparative Analysis with SQL

- **Tracked outbound vs. inbound flows for specific countries (e.g., UAE).**
- **Compared India vs. Germany or India vs. Russia in terms of destination preferences.**
- **Analyzed how enrollment reasons and graduation years influenced student outcomes.**
- **Evaluated university-level performance (placement success, average salaries).**

```sql
select
    count(*) as total_records,
    sum(case when student_id is null then 1 else 0 end
    sum(case when origin_country is null then 1 else 0
    sum(case when destination_country is null then 1 e
    sum(case when university_name is null then 1 else
    sum(case when field_of_study is null then 1 else 0
    sum(case when year_of_enrollment is null then 1 el
    sum(case when placement_status is null then 1 else
from global_student_migration;
---------------
select student_id, count(*) as count
from global_student_migration
group by student_id
having count(*) > 1;
-----------------------
update global_student_migration
set destination_country = 'united kingdom'
where destination_country in ('uk', 'england', 'britai
------------------
select min(gpa_or_score) as min_gpa, max(gpa_or_score)
from global_student_migration
where gpa_or_score > 0;
------------------------------
select
    placement_status,
    count(*) as count,
    cast(count(*) * 100.0 / (select count(*) from glob
from global_student_migration
group by placement_status;
-----------------------
select
    enrollment_reason,
    count(*) as count,
    cast(count(*) * 100.0 / (select count(*) from glob
from global_student_migration
group by enrollment_reason
order by count desc;
```

```sql
    cast(sum(case when placement_status = 'placed' then 1 else 0 end) * 100.0 / count(*) as
from global_student_migration
where test_score > 0
group by case when test_score >= 7.5 then 'high score' else 'lower score' end;
-----------------------------------
select
    field_of_study,
    destination_country,
    count(*) as student_count
from global_student_migration
where field_of_study in ('computer science', 'business')
group by field_of_study, destination_country
order by field_of_study, student_count desc;
----------------------------------
select
    origin_country,
    field_of_study,
    destination_country,
    count(*) as student_count
from global_student_migration
where origin_country in ('india', 'russia')
  and field_of_study in ('computer science', 'business')
group by origin_country, field_of_study, destination_country
order by origin_country, field_of_study, student_count desc;
```

```sql
    destination_country,
    language_proficiency_test,
    count(*) as test_count,
    cast(count(*) * 100.0 / sum(count(*)) over (partition by desti
from global_student_migration
where language_proficiency_test <> 'none'
  and destination_country in ('united kingdom', 'usa')
group by destination_country, language_proficiency_test
order by destination_country, test_count desc;
-----------------------------------
select
    avg(case when placement_status = 'placed' then gpa_or_score en
    avg(case when placement_status = 'not placed' then gpa_or_scor
    avg(case when placement_status = 'placed' then test_score end)
    avg(case when placement_status = 'not placed' then test_score
from global_student_migration
where gpa_or_score > 0 and test_score > 0;
```

```sql
---------------------
t
ost_graduation_visa,
vg(starting_salary_usd) as avg_salary
global_student_migration
 placement_status = 'placed' and starting_salary_usd > 0
 by post_graduation_visa
 by avg_salary desc;
-----------------------
t
raduation_year,
vg(starting_salary_usd) as avg_salary
global_student_migration
 placement_status = 'placed' and starting_salary_usd > 0
 by graduation_year
 by graduation_year;
------------------------
t
ase when field_of_study in ('computer science', 'engineering', 'medicine', 'natural sciences')
    then 'stem' else 'non-stem' end as field_category,
ount(*) as total,
```

```sql
from global_student_migration
where gpa_or_score > 0 and test_score > 0;
-----------------------------------
select
    university_name,
    count(*) as total,
    sum(case when placement_status = 'placed' then 1 else 0 end) as placed,
    cast(sum(case when placement_status = 'placed' then 1 else 0 end) * 100.0 / count(*) as
from global_student_migration
group by university_name
having count(*) > 5
order by placement_rate desc;
-----------------------------
select count(*) as high_achievers
from global_student_migration
where gpa_or_score >= 3.95 and placement_status = 'placed';
-----------------------
select *
from global_student_migration
where gpa_or_score > 3.8 and placement_status = 'not placed';
-----------------------------
```

Our Full Data Analysis Process with Excel

We also performed a complete data analysis process using Excel, leveraging pivot tables, slicers, and dashboards to ensure data cleaning, interactive exploration, and clear visualization of insights from the global student migration dataset.

Steps We Followed:

1-Data Cleaning & Preparation

- **Removed duplicates and standardized values.**
- **Cleaned missing fields: salaries, GPA, test scores, visa status.**
- **Converted numeric fields into proper types.**
- **Standardized placement status.**
- **Created structured tables ready for Pivot Tables & Dashboard.**

Result: A clean dataset ready for analysis.

Our Full Data Analysis Process with Excel

We also performed a complete data analysis process using Excel, leveraging pivot tables, slicers, and dashboards to ensure data cleaning, interactive exploration, and clear visualization of insights from the global student migration dataset.

2-**Pivot Tables & Slicers**

- **Pivot Tables Built For:**
  - **Scholarships by field of study.**
  - **Student distribution by country & city.**
  - **Placement outcomes by company.**
  - **Student numbers trend (2019–2023).**
- **Slicers Added:**
  - **Year of enrollment**
  - **Field of study**
  - **Placement country**

Our Full Data Analysis Process with Excel

We also performed a complete data analysis process using Excel, leveraging pivot tables, slicers, and dashboards to ensure data cleaning, interactive exploration, and clear visualization of insights from the global student migration dataset.

3-**Dashboard KPIs**

At the top of the dashboard, we displayed Key Performance Indicators:

- **Average Starting Salary → $88,618**
- **Scholarship Rate → 52%**
- **Average GPA/Score → 3.24**
- **Placement Rate → 50.18% placed, 49.82% not placed**

KPIs give a quick summary of overall student outcomes.

4-**Dashboard Visuals**

The dashboard includes:

- **Placements by City (area chart).**
- **Student Numbers Trend (line chart showing growth 2019–2023).**
- **Scholarships by Field (bar chart).**
- **Student Distribution by Country (bar chart).**
- **Enrollment Reasons by City (clustered bar).**
- **Company-wise Student Distribution (bar chart of top recruiters).**

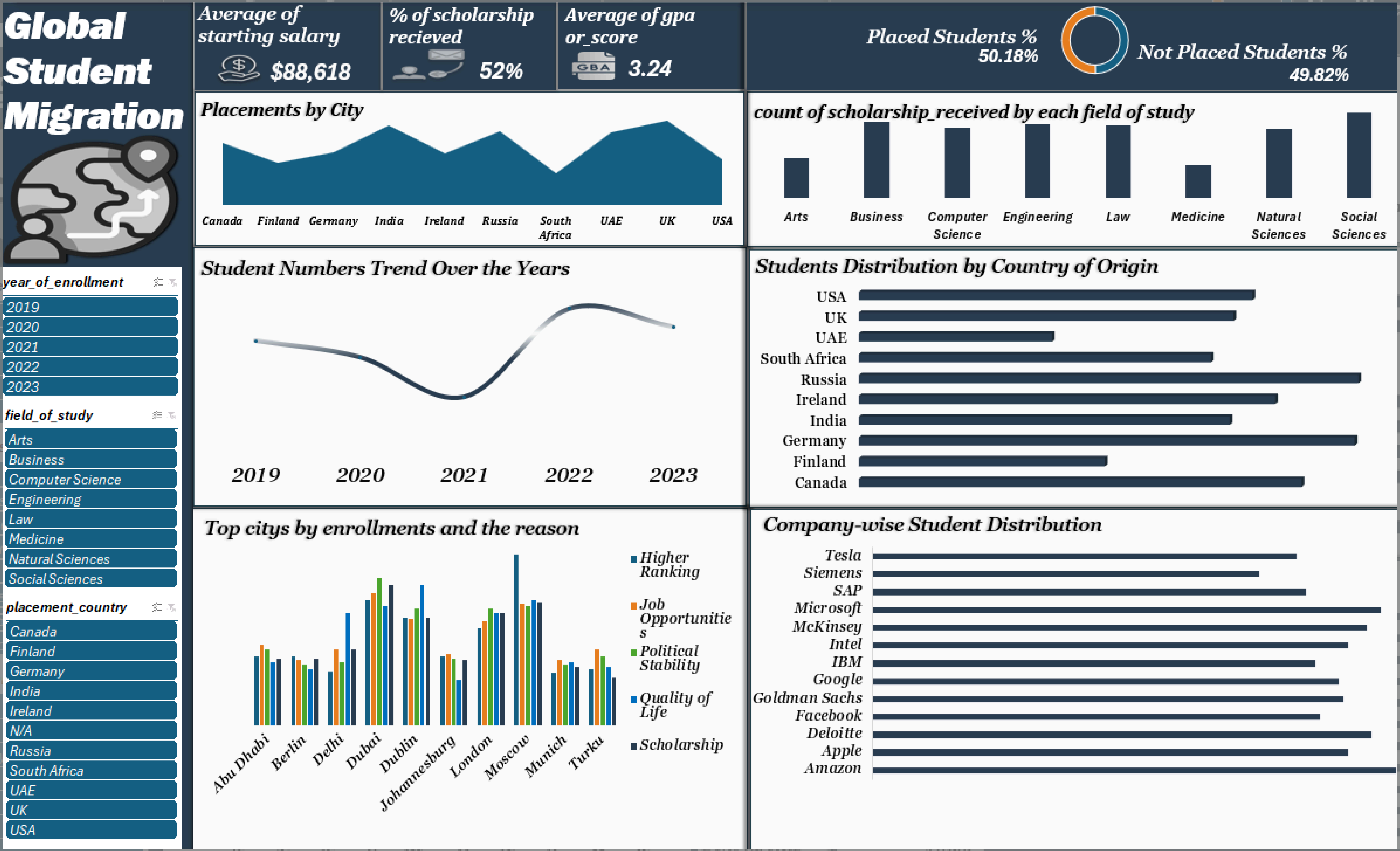
Each visual shows a different perspective: fields, cities, countries, companies.

Our Full Data Analysis Process with Excel

We also performed a complete data analysis process using Excel, leveraging pivot tables, slicers, and dashboards to ensure data cleaning, interactive exploration, and clear visualization of insights from the global student migration dataset.

5 – Insights & Key Findings

- **Top Sending Countries → Russia, Canada, South Africa.**
- **Top Cities for Enrollment → Dubai, Moscow, London (due to job opportunities & stability).**
- **Scholarships are evenly spread (~600–645 per field).**
- **Top Companies Hiring → Microsoft, Apple, Amazon, Deloitte.**
- **Student Numbers remained stable with a peak in 2022 (1027 students).**
- **Placements → Half of students successfully placed; tech and business students domina**



Global Student Migration
year_of_enrollment
2019
2020
2021
2022
2023
field_of_study
Arts
Business
Computer Science
Engineering
Law
Medicine
Natural Sciences
Social Sciences
placement_country
Canada
Finland
Germany
India
Ireland
N/A
Russia
South Africa
UAE
UK
USA
Average of starting salary
$88,618
% of scholarship recieved
52%
Average of gpa or_score
GBA
3.24
Placed Students %
50.18%
Not Placed Students %
49.82%
Placements by City
Canada
Finland
Germany
India
Ireland
Russia
South Africa
UAE
UK
USA
count of scholarship_received by each field of study
Arts
Business
Computer Science
Engineering
Law
Medicine
Natural Sciences
Social Sciences
Student Numbers Trend Over the Years
2019
2020
2021
2022
2023
Students Distribution by Country of Origin
USA
UK
UAE
South Africa
Russia
Ireland
India
Germany
Finland
Canada
Top citys by enrollments and the reason
Higher Ranking
Job Opportunities
Political Stability
Quality of Life
Scholarship
Abu Dhabi
Berlin
Delhi
Dubai
Dublin
Johannesburg
London
Moscow
Munich
Turku
Company-wise Student Distribution
Tesla
Siemens
SAP
Microsoft
McKinsey
Intel
IBM
Google
Goldman Sachs
Facebook
Deloitte
Apple
Amazon

Conclusion

"Data is not just about charts and numbers; it's about making things clearer, easier, and more meaningful. I hope this presentation didn't just share insights, but also helped you see how data can actually support decisions in a practical and simple way. Because when data is understood, it truly becomes power."

“I’ll be happy to answer any questions you may have?

Thank you