IBM Data Analyst Capstone Project

Harshit Verma

18 August 2021

OUTLINE

- Executive Summary
- Introduction
- Methodology
- Results
  - Visualization – Charts
  - Dashboard
- Discussion
  - Findings & Implications
- Conclusion
- Appendix

EXECUTIVE SUMMARY

- Collecting Data using APIs and Web Scrapping
- Using a publicly available dataset from stackoverflow
  - For exploring pandas library
  - Data Wrangling
  - Exploratory Data Analysis
- Data Visualization using matplotlib and seaborn
- Building A Dashboard using IBM Cognos

INTRODUCTION

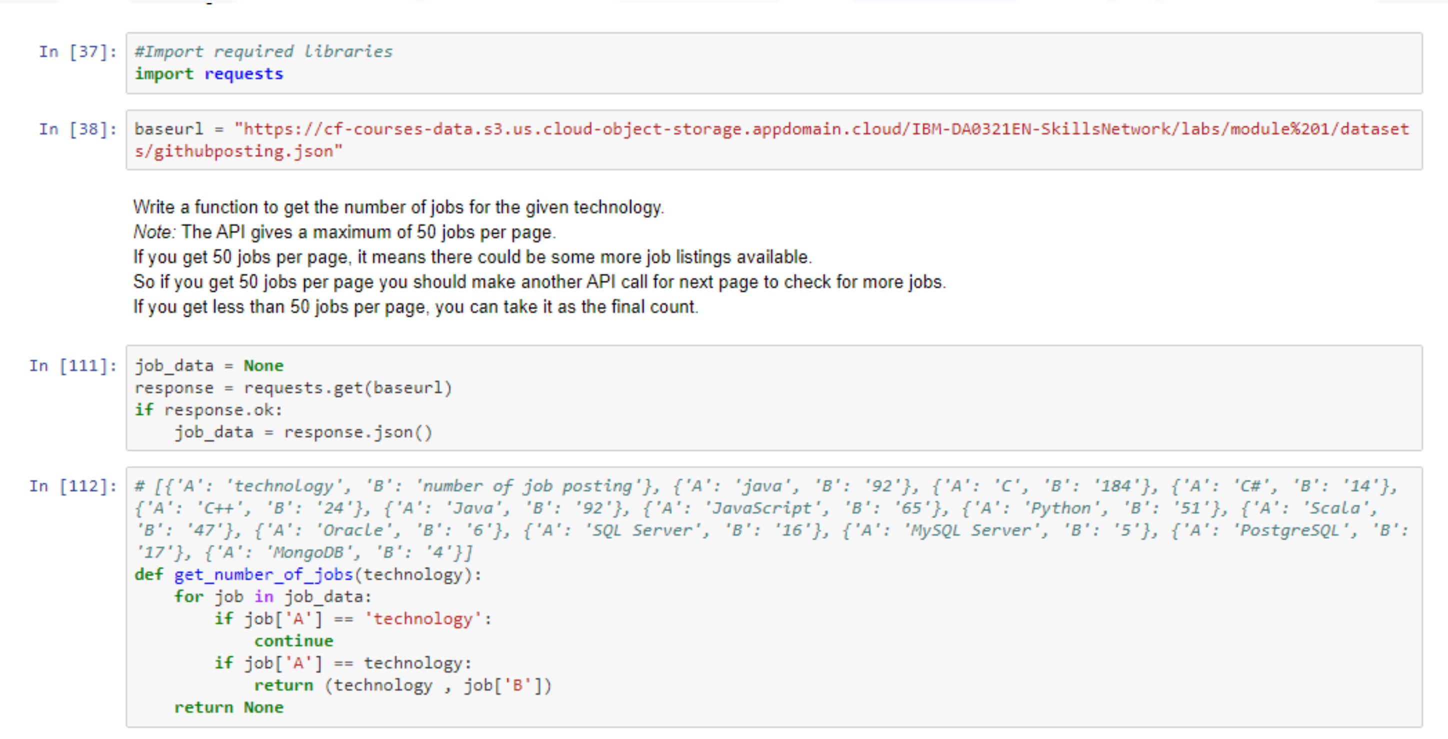
- Collecting Job data and popular language data
- Analysing and normalizing the data
- Visualizing data for future trends
- Conclusion
    - What are highest paid job profiles
    - What language, database, platform will be popular in the future

METHODOLOGY

- Using Web Scraping and publicly available dataset from different sources
- Using pandas and numpy libraries for data analysis
- Using matplotlib and seaborn for data visualization
- Using IBM Cognos for dashboard generation
    - Dashboard 3 for insights about jobs
    - Dashboard 1 and 2 for insights on popular technologies

RESULTS

Using API for extracting data



```
In [37]: #Import required libraries
         import requests
```

```
In [38]: baseurl = "https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DA0321EN-SkillsNetwork/labs/module%201/dataset
         s/githubposting.json"
```

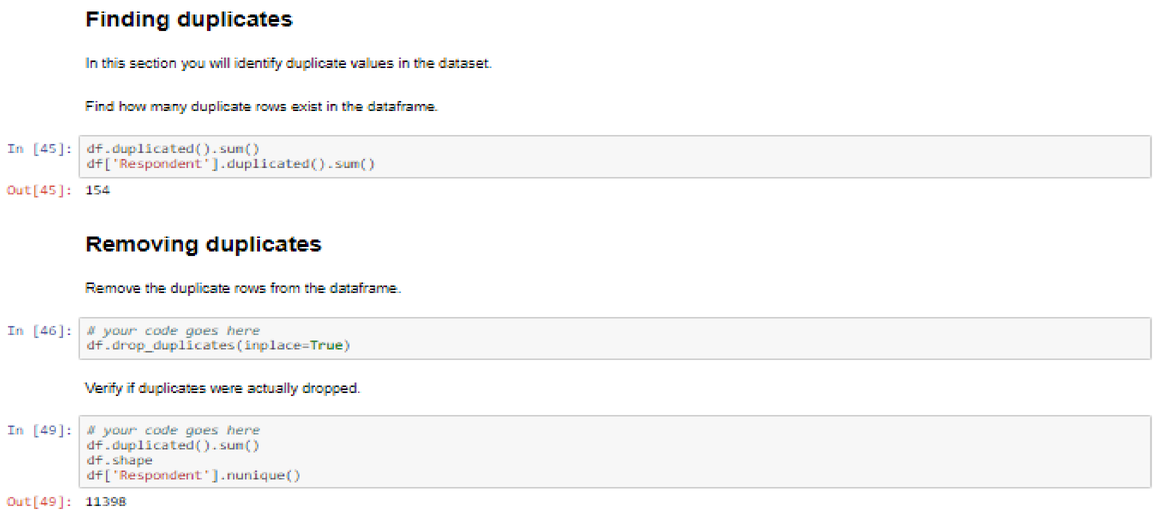
Write a function to get the number of jobs for the given technology.
Note: The API gives a maximum of 50 jobs per page.
If you get 50 jobs per page, it means there could be some more job listings available.
So if you get 50 jobs per page you should make another API call for next page to check for more jobs.
If you get less than 50 jobs per page, you can take it as the final count.

```
In [111]: job_data = None
          response = requests.get(baseurl)
          if response.ok:
              job_data = response.json()
```

```
In [112]: # [{'A': 'technology', 'B': 'number of job posting'}, {'A': 'java', 'B': '92'}, {'A': 'C', 'B': '184'}, {'A': 'C#', 'B': '14'},
          {'A': 'C++', 'B': '24'}, {'A': 'Java', 'B': '92'}, {'A': 'JavaScript', 'B': '65'}, {'A': 'Python', 'B': '51'}, {'A': 'Scala',
          'B': '47'}, {'A': 'Oracle', 'B': '6'}, {'A': 'SQL Server', 'B': '16'}, {'A': 'MySQL Server', 'B': '5'}, {'A': 'PostgreSQL', 'B':
          '17'}, {'A': 'MongoDB', 'B': '4'}]
          def get_number_of_jobs(technology):
              for job in job_data:
                  if job['A'] == 'technology':
                      continue
                  if job['A'] == technology:
                      return (technology , job['B'])
              return None
```

RESULTS

Data Wrangling



Finding duplicates

In this section you will identify duplicate values in the dataset.

Find how many duplicate rows exist in the dataframe.

```
In [45]: df.duplicated().sum()
         df['Respondent'].duplicated().sum()
Out[45]: 154
```

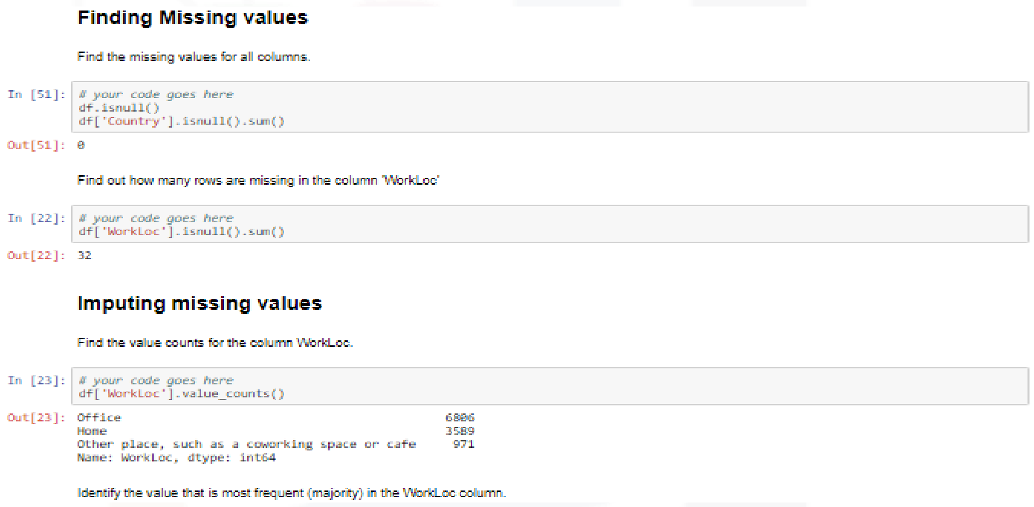
Removing duplicates

Remove the duplicate rows from the dataframe.

```
In [46]: # your code goes here
         df.drop_duplicates(inplace=True)
```

Verify if duplicates were actually dropped.

```
In [49]: # your code goes here
         df.duplicated().sum()
         df.shape
         df['Respondent'].nunique()
Out[49]: 11398
```



Finding Missing values

Find the missing values for all columns.

```
In [51]: # your code goes here
         df.isnull()
         df['Country'].isnull().sum()
Out[51]: 0
```

Find out how many rows are missing in the column 'WorkLoc'

```
In [22]: # your code goes here
         df['WorkLoc'].isnull().sum()
Out[22]: 32
```

Imputing missing values

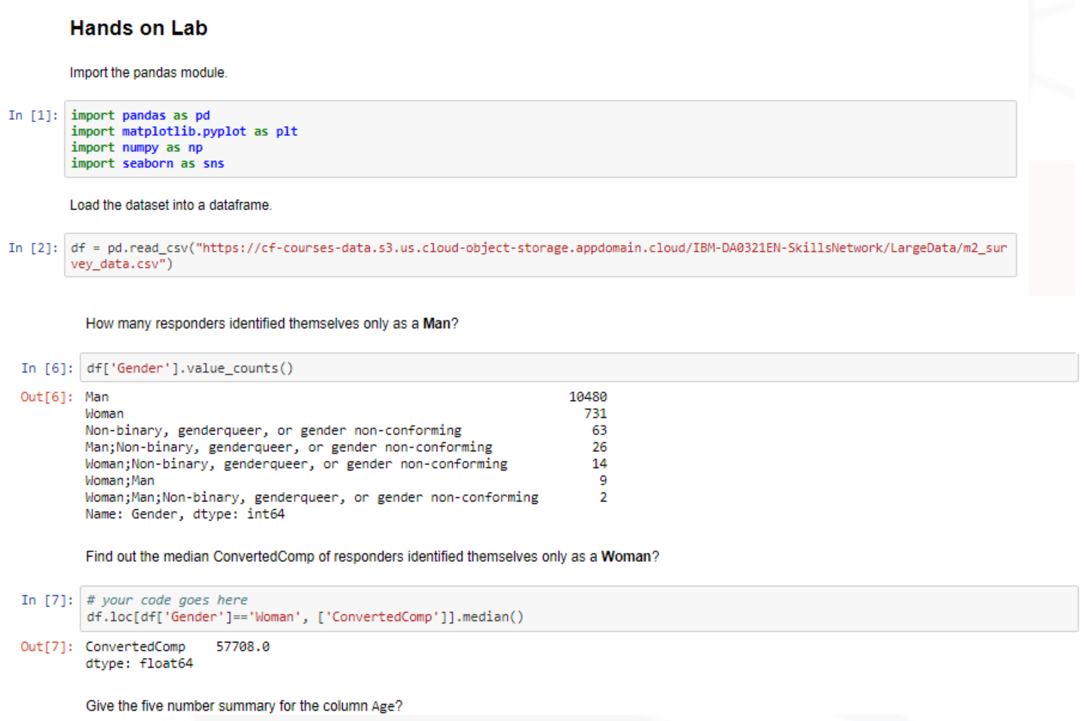
Find the value counts for the column WorkLoc.

```
In [23]: # your code goes here
         df['WorkLoc'].value_counts()
Out[23]: Office                                         6806
         Home                                           3589
         Other place, such as a coworking space or cafe  971
         Name: WorkLoc, dtype: int64
```

Identify the value that is most frequent (majority) in the WorkLoc column.

RESULTS

Exploratory Data Analysis



Hands on Lab

Import the pandas module.

```
In [1]: import pandas as pd
        import matplotlib.pyplot as plt
        import numpy as np
        import seaborn as sns
```

Load the dataset into a dataframe.

```
In [2]: df = pd.read_csv("https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DA0321EN-SkillsNetwork/LargeData/m2_sur
        vey_data.csv")
```

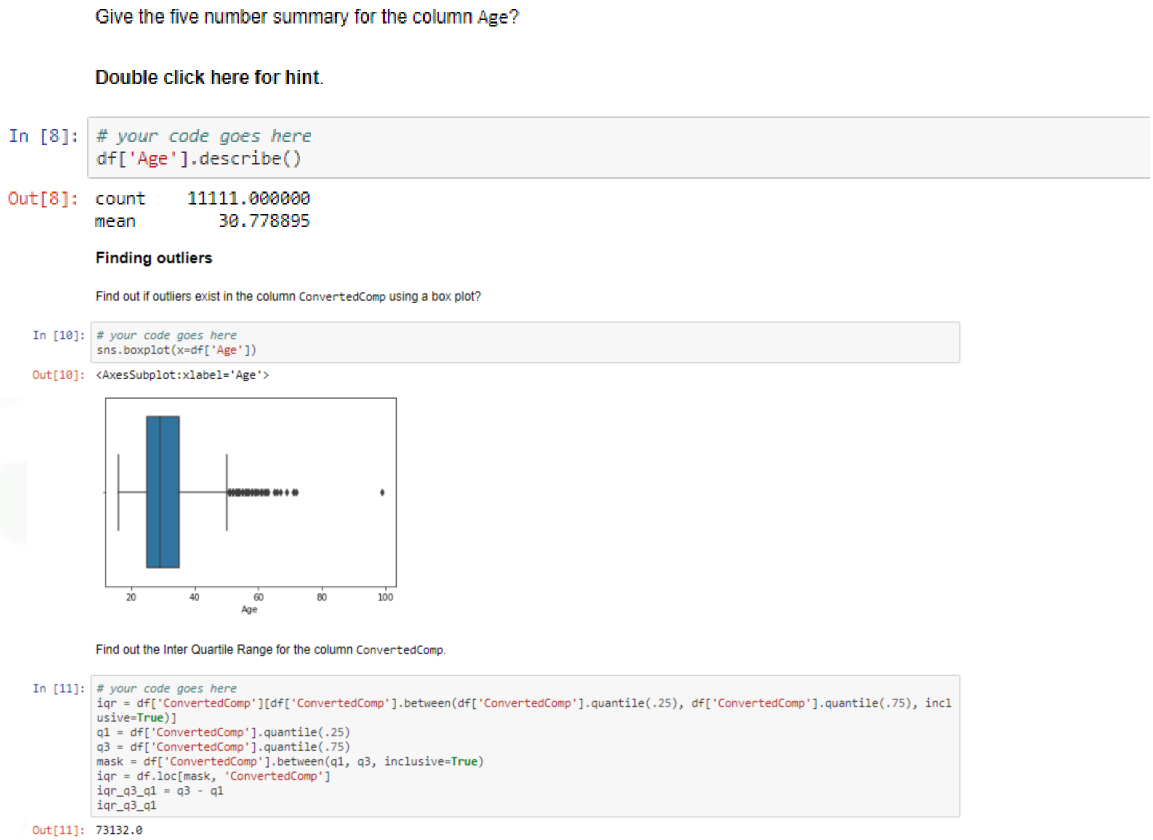
How many responders identified themselves only as a **Man**?

```
In [6]: df['Gender'].value_counts()

Out[6]: Man                                                          10480
        Woman                                                          731
        Non-binary, genderqueer, or gender non-conforming               63
        Man;Non-binary, genderqueer, or gender non-conforming           26
        Woman;Non-binary, genderqueer, or gender non-conforming         14
        Woman;Man                                                        9
        Woman;Man;Non-binary, genderqueer, or gender non-conforming      2
        Name: Gender, dtype: int64
```

Find out the median ConvertedComp of responders identified themselves only as a **Woman**?

```
In [7]: # your code goes here
        df.loc[df['Gender']=='Woman', ['ConvertedComp']].median()

Out[7]: ConvertedComp    57708.0
        dtype: float64
```

Give the five number summary for the column Age?



Give the five number summary for the column Age?

Double click here for hint.

```
In [8]: # your code goes here
        df['Age'].describe()

Out[8]: count    11111.000000
        mean        30.778895
```

Finding outliers

Find out if outliers exist in the column ConvertedComp using a box plot?

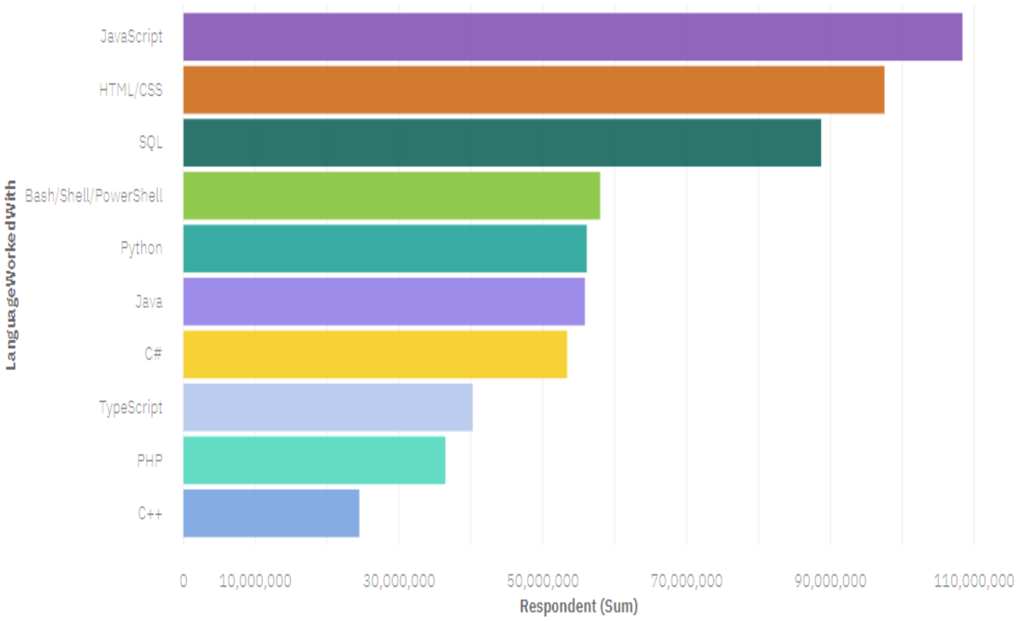
```
In [10]: # your code goes here
         sns.boxplot(x=df['Age'])

Out[10]: <AxesSubplot:xlabel='Age'>
```

Find out the Inter Quartile Range for the column ConvertedComp.

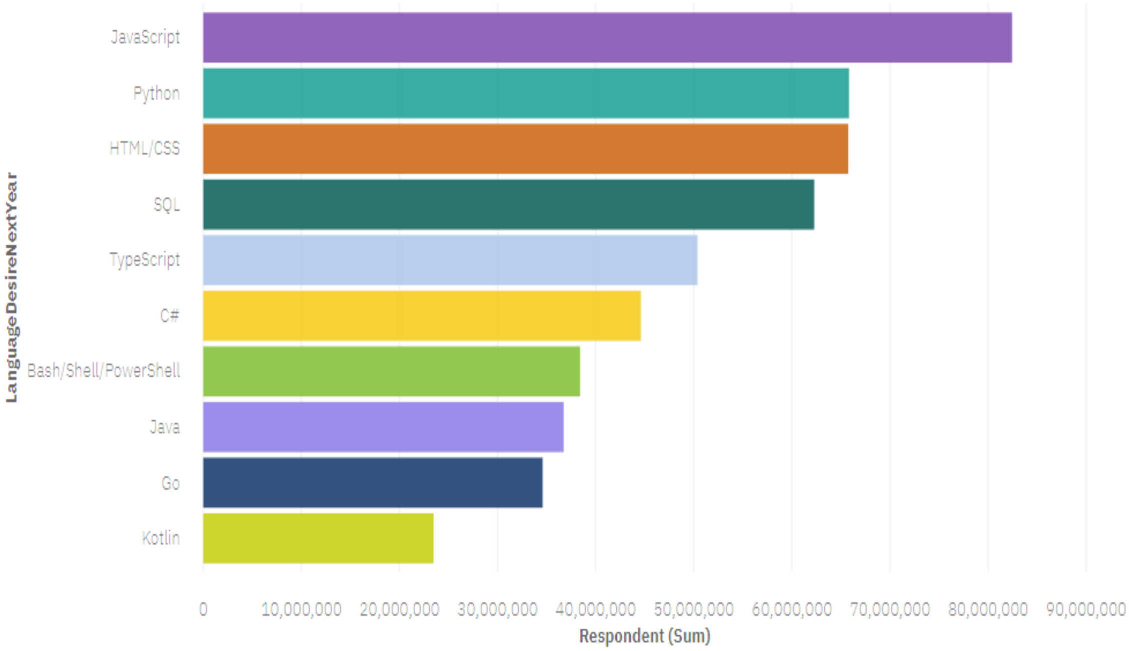
```
In [11]: # your code goes here
         iqr = df['ConvertedComp'][df['ConvertedComp'].between(df['ConvertedComp'].quantile(.25), df['ConvertedComp'].quantile(.75), incl
         usive=True)]
         q1 = df['ConvertedComp'].quantile(.25)
         q3 = df['ConvertedComp'].quantile(.75)
         mask = df['ConvertedComp'].between(q1, q3, inclusive=True)
         iqr = df.loc[mask, 'ConvertedComp']
         iqr_q3_q1 = q3 - q1
         iqr_q3_q1

Out[11]: 73132.0
```

PROGRAMMING LANGUAGE TRENDS

Current Year



Next Year



PROGRAMMING LANGUAGE TRENDS - FINDINGS & IMPLICATIONS

Findings

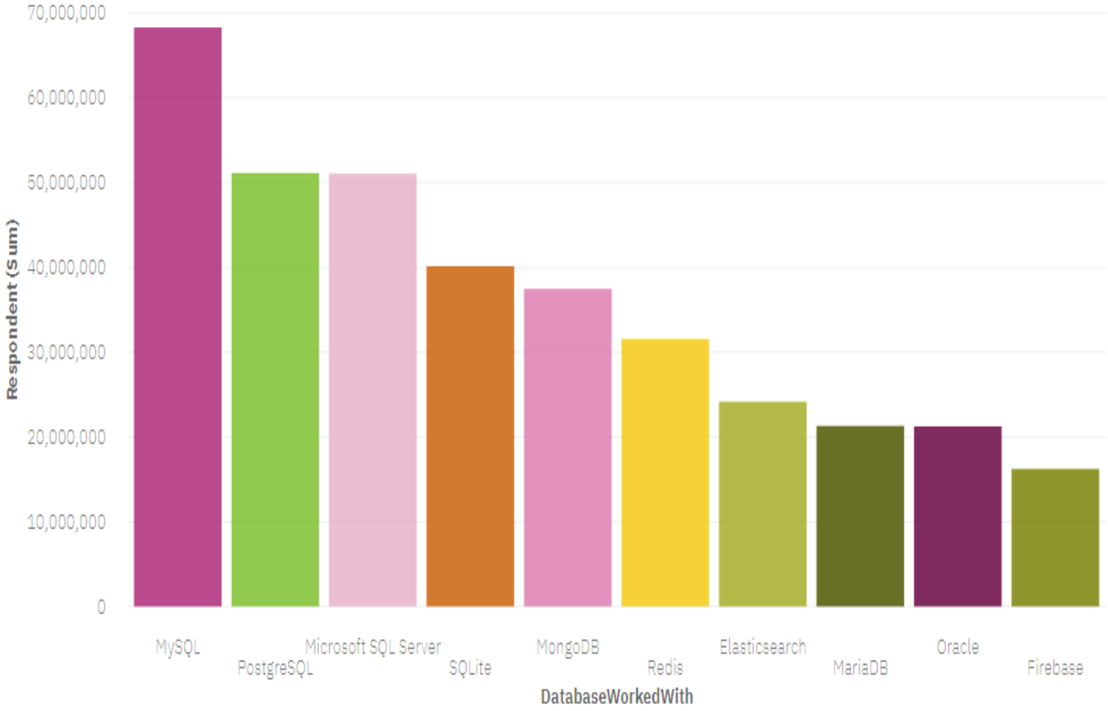
- Web development languages are most popular
- JavaScript, Html/CSS are used by majority
- Python is most popular development language

Implications

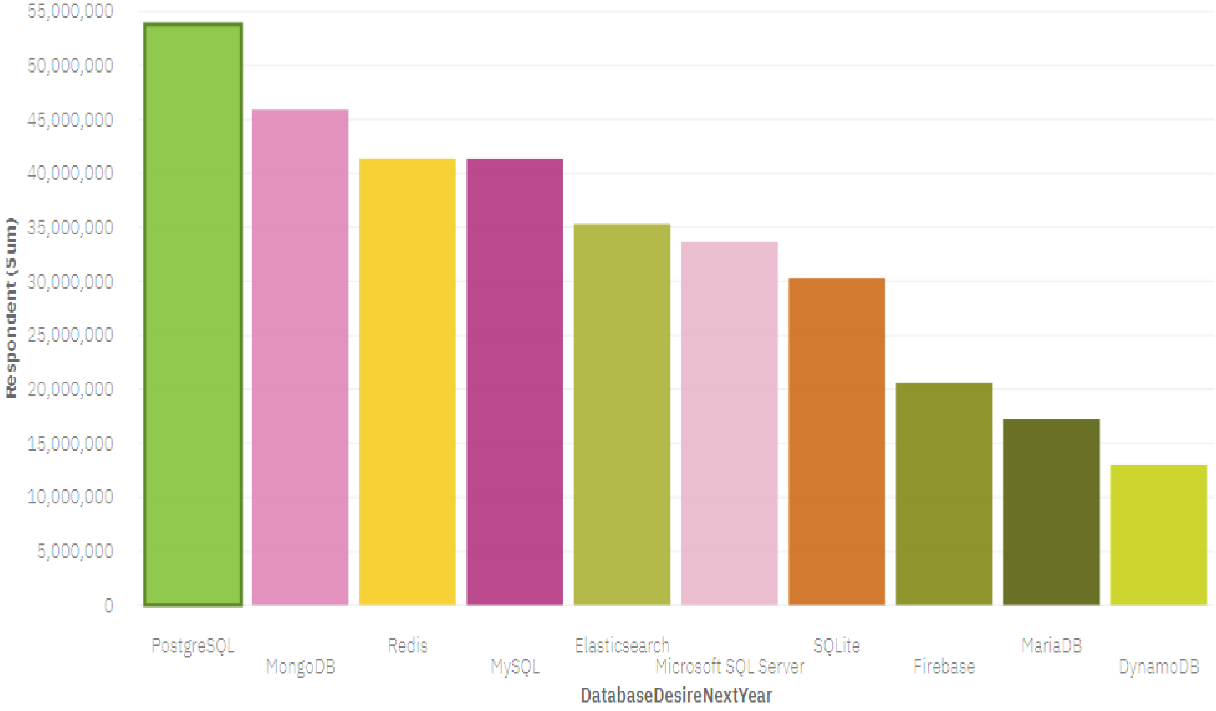
- High competition as most people work in web development
- Starting with Html/CSS, JavaScript is a good idea
- Learn Python if you are interested in development

DATABASE TRENDS

Current Year



Next Year



DATABASE TRENDS - FINDINGS & IMPLICATIONS

Findings

- MySQL is most popular language currently
- MongoDB and Redis are not most popular
- Microsoft SQL Sever is one of the most popular database

Implications

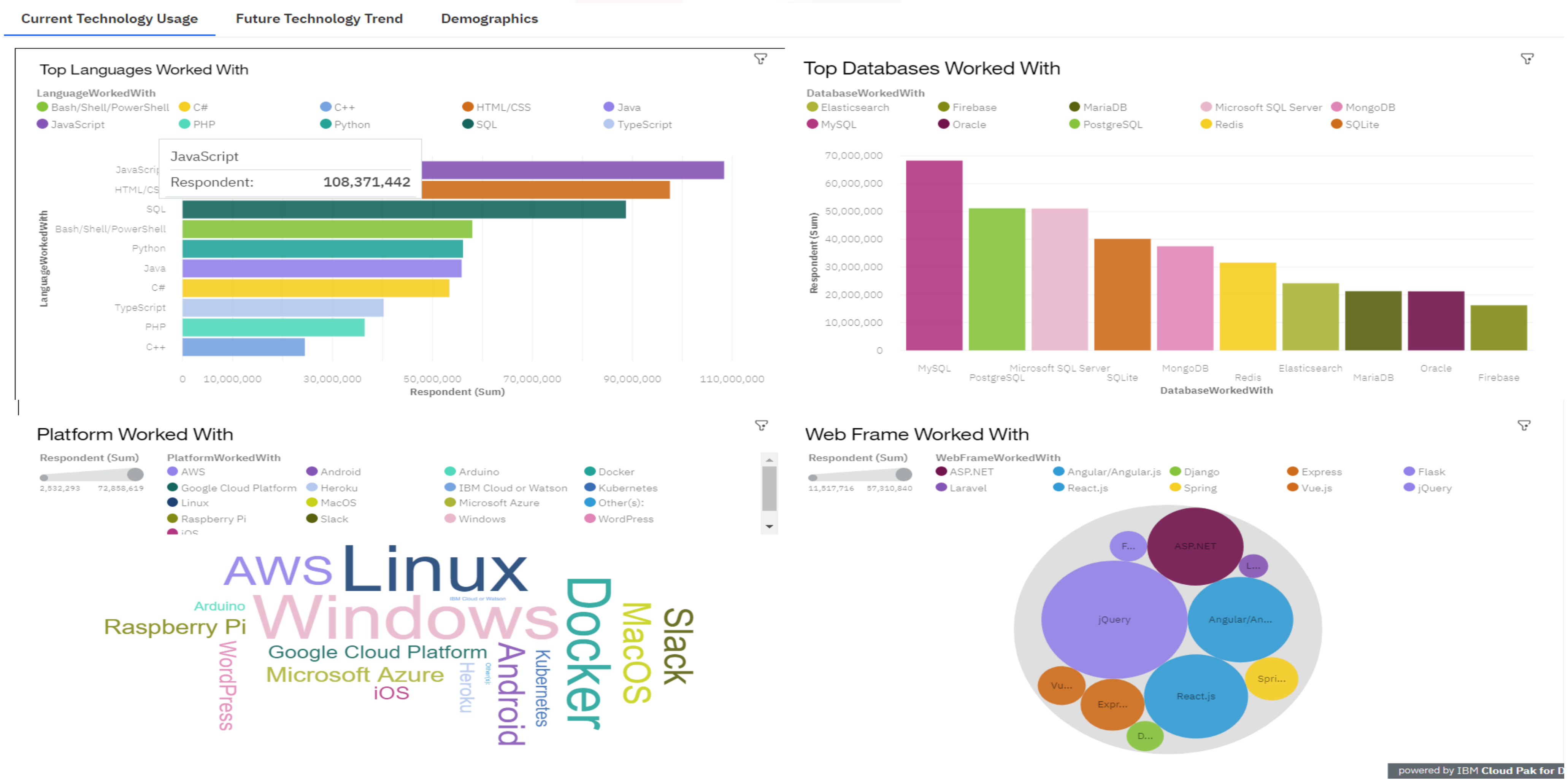
- MySQL will be replaced by PostgreSQL next year
- MongoDB and Redis will be becoming popular in the future
- Microsoft SQL will slowly lose its popularity

DASHBOARD

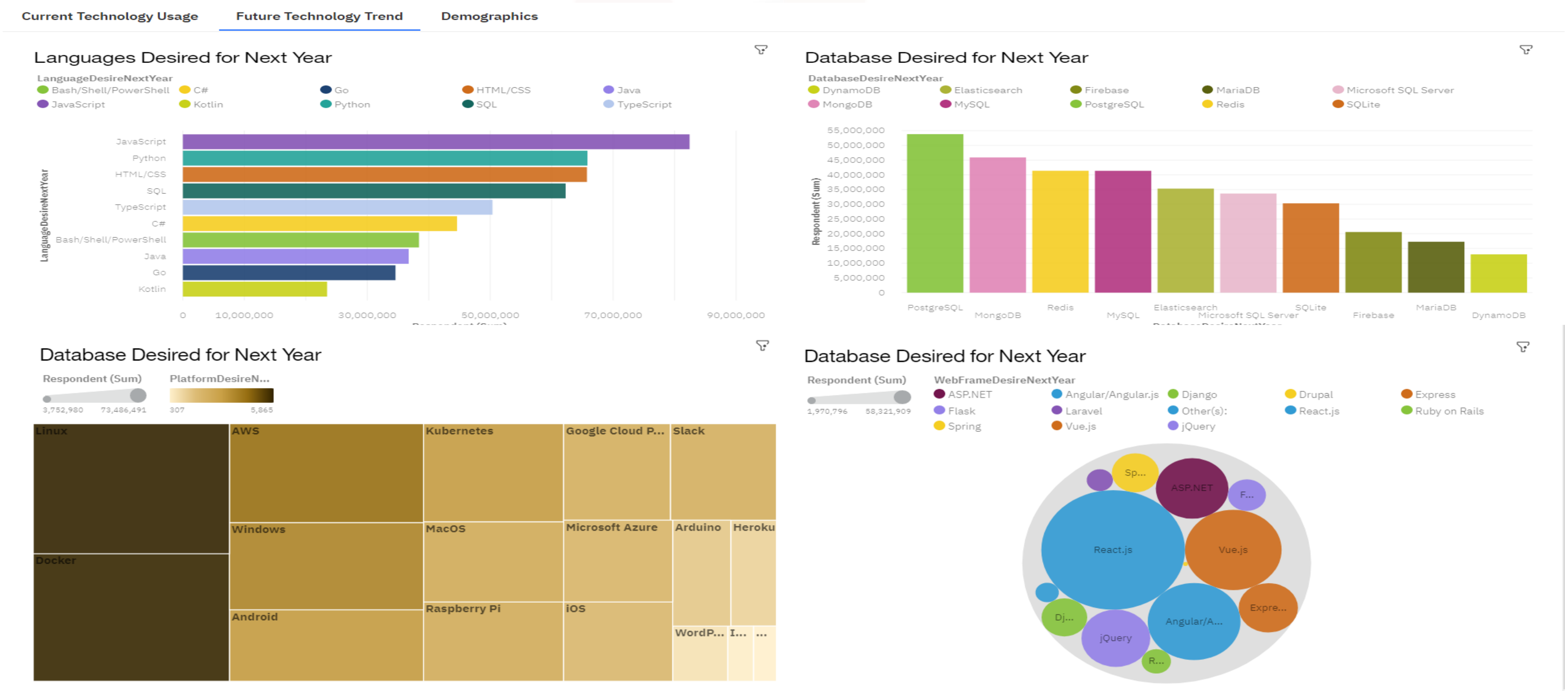
Link for the dashboard created with the help of IBM Cognos

IBM Cloud Pak for Data

DASHBOARD TAB 1



DASHBOARD TAB 2

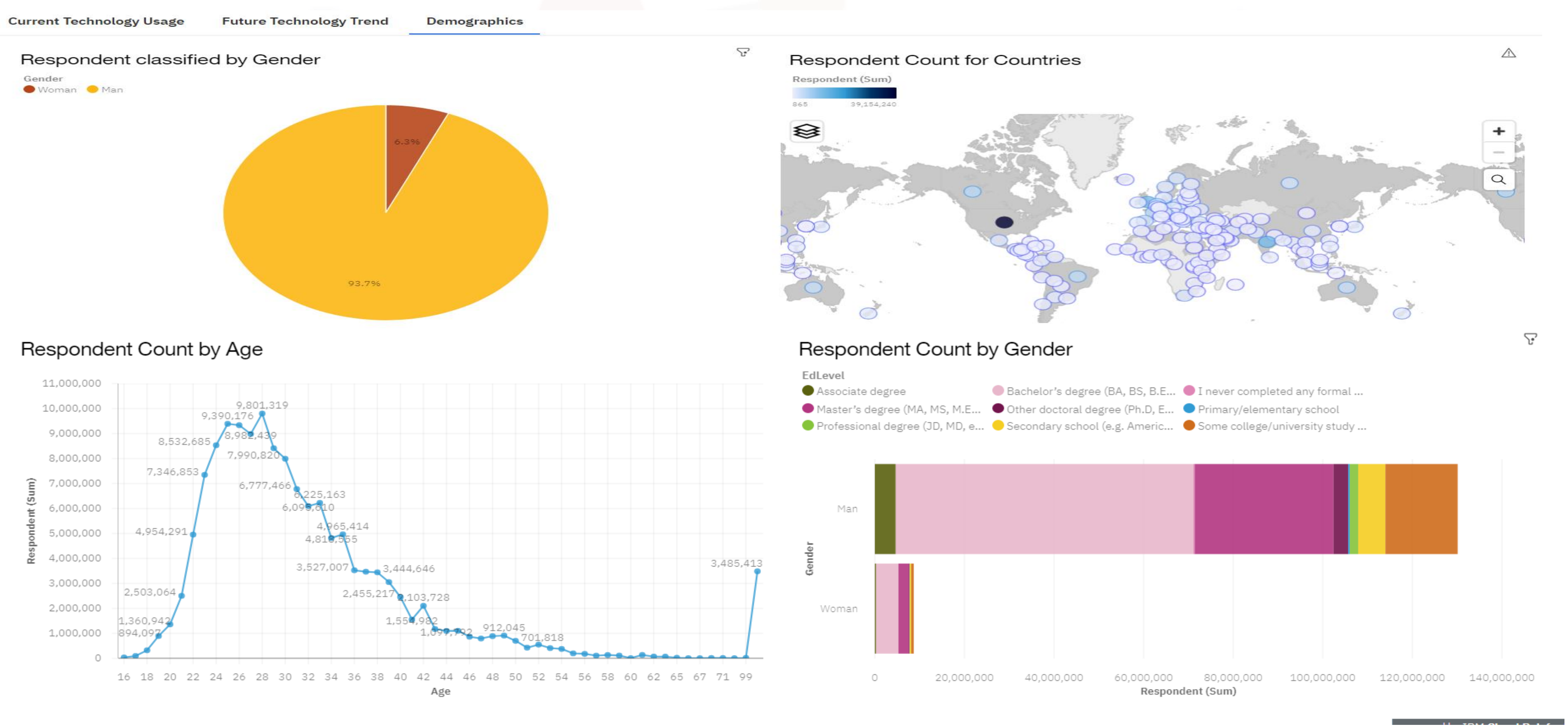


IBM Developer

powered by IBM Cloud Pak for D

SKILLS NETWORK

DASHBOARD TAB 3



DISCUSSION

- Best languages for college graduates to learn are JavaScript, HTML/CSS, Python, SQL.
- Best databases for college graduates to learn are PostgreSQL, MongoDB, Redis
- Best Platform for college graduates to learn are Linux, AWS, Windows

OVERALL FINDINGS & IMPLICATIONS

Findings

- JavaScript, HTML/CSS are widely used for web development
- Python, SQL are widely used for data analysis
- Widely used platforms for development are Linux, AWS, Windows

Implications

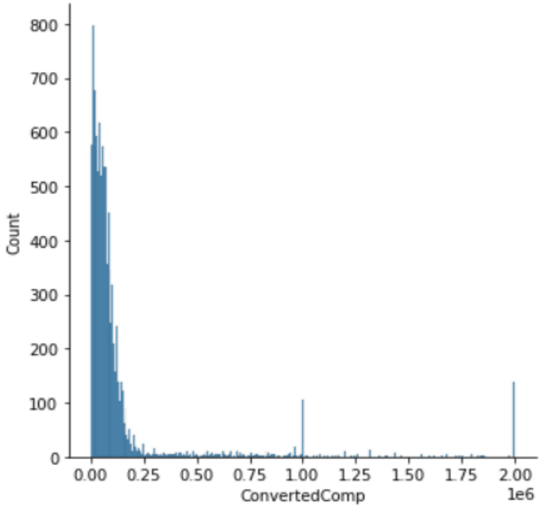
- Learning JavaScript, HTML/CSS is beneficial
- Learning Python, SQL are important for data scientist
- Having basic knowledge of Linux, AWS, Windows is important

CONCLUSION

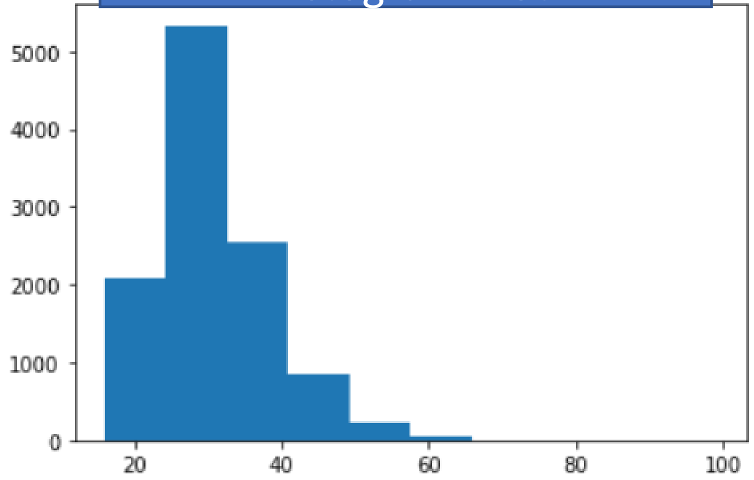
- Technology trends change rapidly with time.
- Keep learning about new technologies is important for growing in the field of technology
- Python, HTML/CSS are by far the most important languages to learn
- Learning language which is not very popular can land you a good job.

APPENDIX

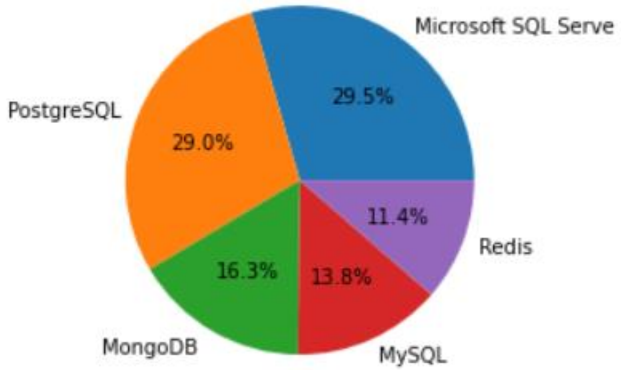
Distribution Plot



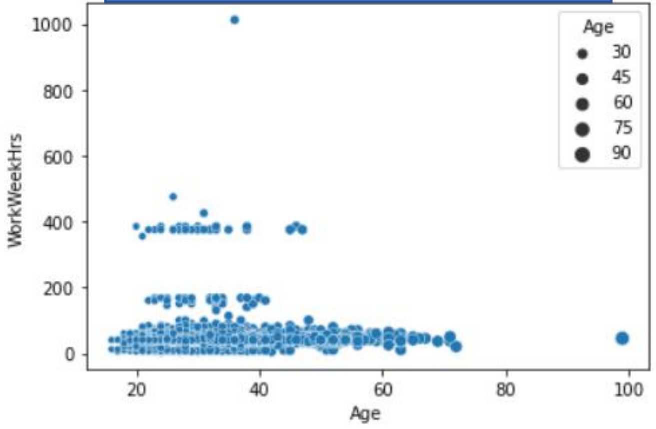
Histogram Plot



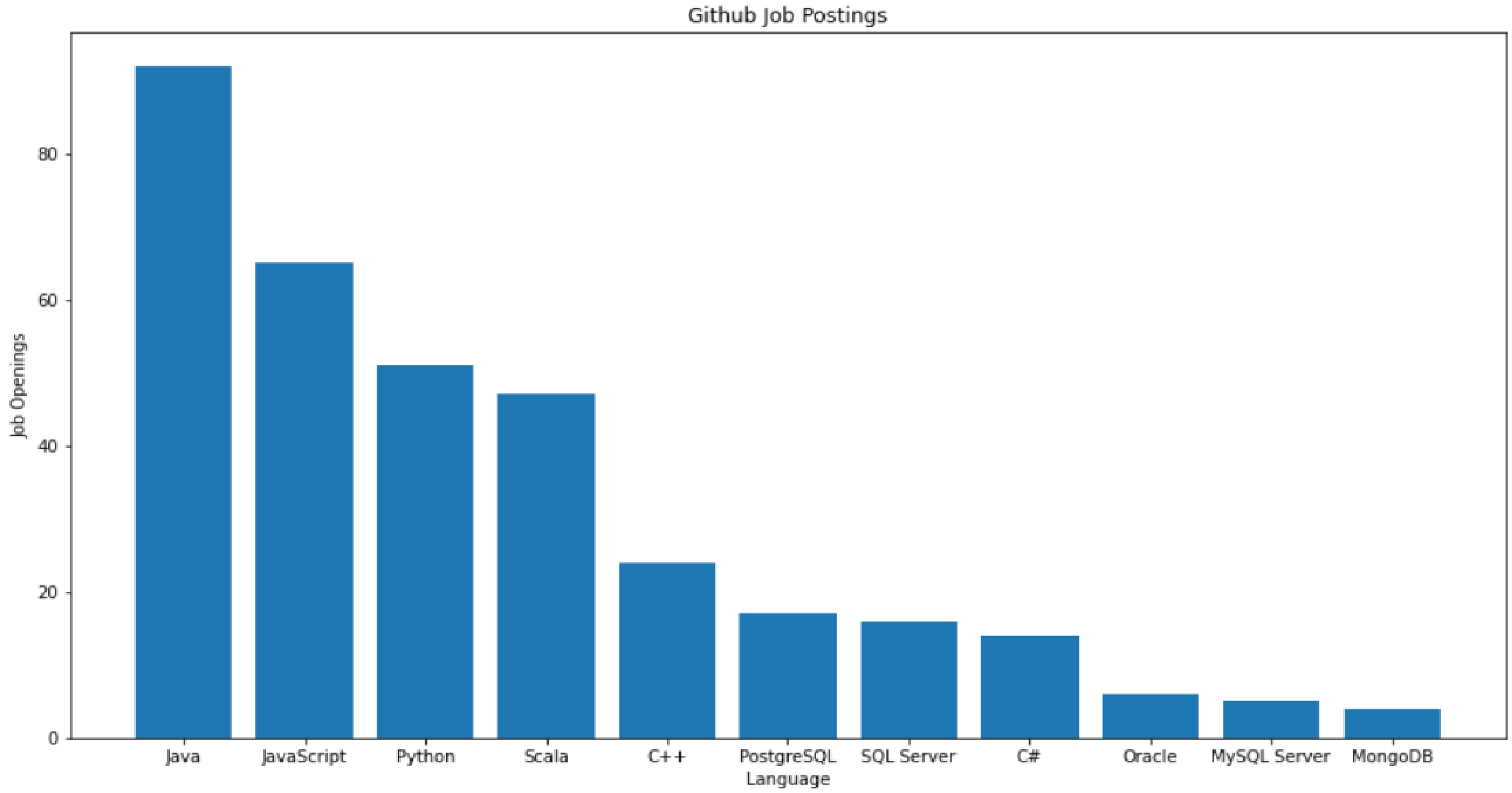
Pie Chart



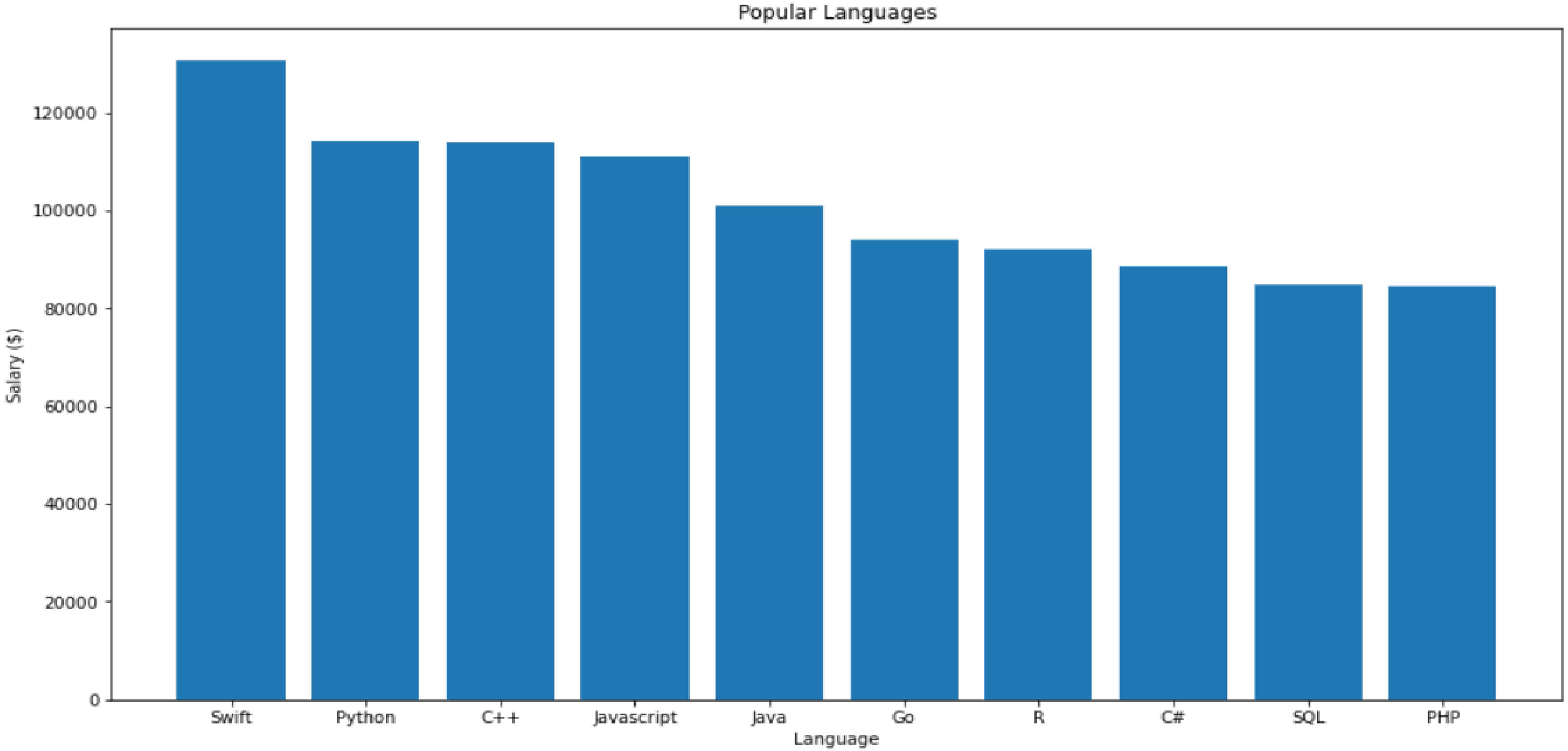
Scatter Plot



GITHUB JOB POSTINGS



POPULAR LANGUAGES


Popular Languages
Salary ($)
Swift
Python
C++
Javascript
Java
Go
R
C#
SQL
PHP
Language