Emerging technologies Trends

Xinyu Zhao

06/17/2022

IBM Developer

SKILLS NETWORK

OUTLINE

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

EXECUTIVE SUMMARY

- Technology is changing the world. Data, programming, and Web Frame designing professionals are facing rapidly changing technology languages. What language is or will be popular can be an important question to answer.
- This report will analyze technology trends specifically on industry demography and technology languages.
- It is important for professionals to consistently learn new languages and this report aim to show some insights on what languages are important.

INTRODUCTION

- This Report Tries to identify technology trends
- Collect data using several APIs and Web scrapping techniques
  - Rest API
  - Beautiful Soup Web scrapping
- Data Cleaning and Processing
- Exploratory data analysis
  - SQL Query
  - Correlation, Outliers, Distribution, etc.
- Data Visualization
  - Plots, Charts
  - Cognos Dashboard
- Conclusion

METHODOLOGY

- Rest API enables us to download data online
  - Pros: Easy, fast, and organized data
  - Cons: Sometimes Not available
- Web scrapping enables data collection from any websites
  - First, Utilize “Requests” to obtain JSON file of the website
  - Utilize Beautiful Soups Python to transfer JSON to scripts
  - Iterate through the scripts to obtain Data Table
- Exploratory Data Analysis Methods
  - SQL query – Return a table of data according to our query
  - Correlation, outliers, plots – statistical understanding of data

RESULTS

```
api_url = "http://api.open-notify.org/astros.json" # this url gives use the astronaut data
```

```
response = requests.get(api_url) # Call the API using the get method and store the
                                 # output of the API call in a variable called response.
```

```
if response.ok:              # if all is well() no errors, no network timeouts)
    data = response.json()   # store the result in json format in a variable called data
                             # the variable data is of type dictionary.
```

```
print(data)   # print the data just to check the output or for debugging
```


```
{'number': 10, 'people': [{'name': 'Oleg Artemyev', 'craft': 'ISS'}, {'name': 'Denis Matveev', 'craft': 'ISS'}, {'name': 'Sergey Korsakov', 'craft': 'ISS'}, {'name': 'Kjell Lindgren', 'craft': 'ISS'}, {'name': 'Bob Hines', 'craft': 'ISS'}, {'name': 'Samantha Cristoforetti', 'craft': 'ISS'}, {'name': 'Jessica Watkins', 'craft': 'ISS'}, {'name': 'Cai Xuzhe', 'craft': 'Tiangong'}, {'name': 'Chen Dong', 'craft': 'Tiangong'}, {'name': 'Liu Yang', 'craft': 'Tiangong'}], 'message': 'success'}
```


API

	Technology	Washington DC	New York	Boston	Austin	Houston
0	C	1	0	0	0	0
1	C#	68	41	30	5	29
2	C++	55	43	38	4	42
3	Java	21	12	10	2	11
4	JavaScript	61	51	44	5	44
5	Python	258	143	129	15	151
6	Scala	3	8	1	1	5

```
<table border="1" sryle="height:150px; width:400px">
 <tbody>
  <tr>
    <td>Row 1</td>
  </tr>
  <tr>
    <td>Row 2</td>
  </tr>
  <tr>
    <td>Row 3</td>
  </tr>
  <tr>
    <td>Row 4</td>
  </tr>
 </tbody>
</table>
```

Scripts

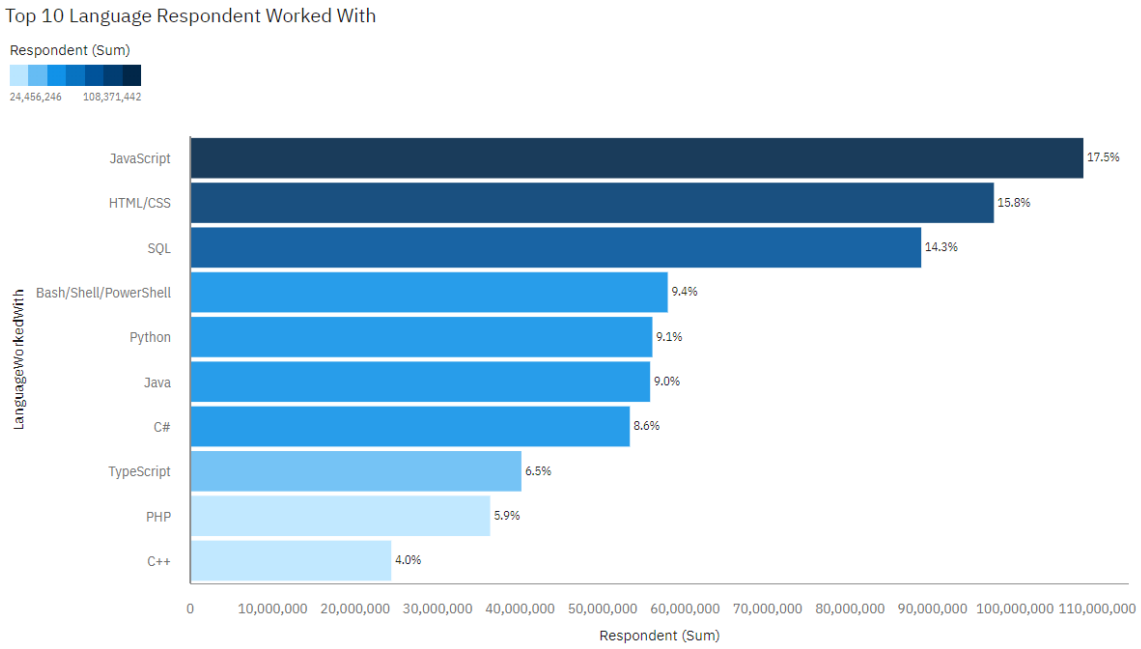
```
#your code goes here
table=soup.find('table')
def get_data(table):
    language_salary_list = []
    for row in table.find_all('tr'):
        cols=row.find_all('td')
        language_name = cols[1].getText()
        annual_average_salary = cols[3].getText()
    #     print("{}:{}".format(language_name, annual_average_salary))
        language_salary_list.append([language_name, annual_average_salary])
    language_salary_list.remove(['Language', 'Average Annual Salary'])
    return language_salary_list

data=get_data(table)
```

Iterate

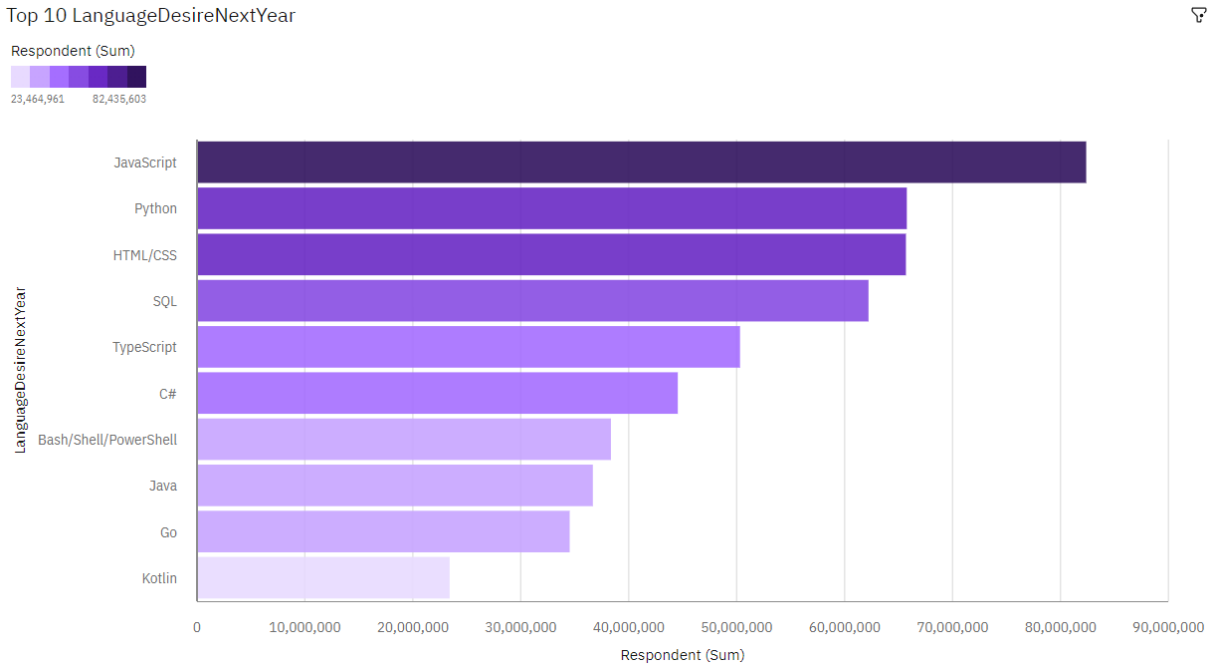
	Technology	Washington DC	New York	Boston	Austin	Houston
0	C	1	0	0	0	0
1	C#	68	41	30	5	29
2	C++	55	43	38	4	42
3	Java	21	12	10	2	11
4	JavaScript	61	51	44	5	44
5	Python	258	143	129	15	151
6	Scala	3	8	1	1	5

PROGRAMMING LANGUAGE TRENDS

Current Year



Next Year



PROGRAMMING LANGUAGE TRENDS - FINDINGS & IMPLICATIONS

Findings

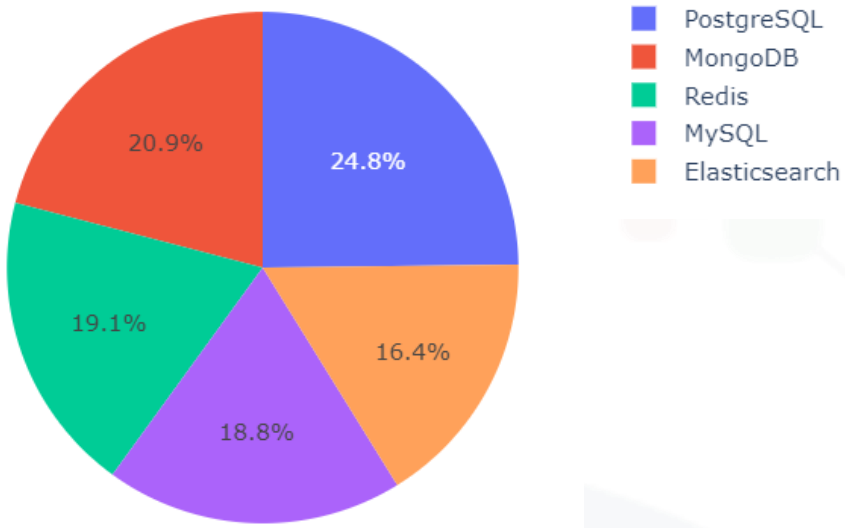
- JavaScript and HTML/CSS are two most used languages currently
- Python, Go are expected to be much more popular next year than now

Implications

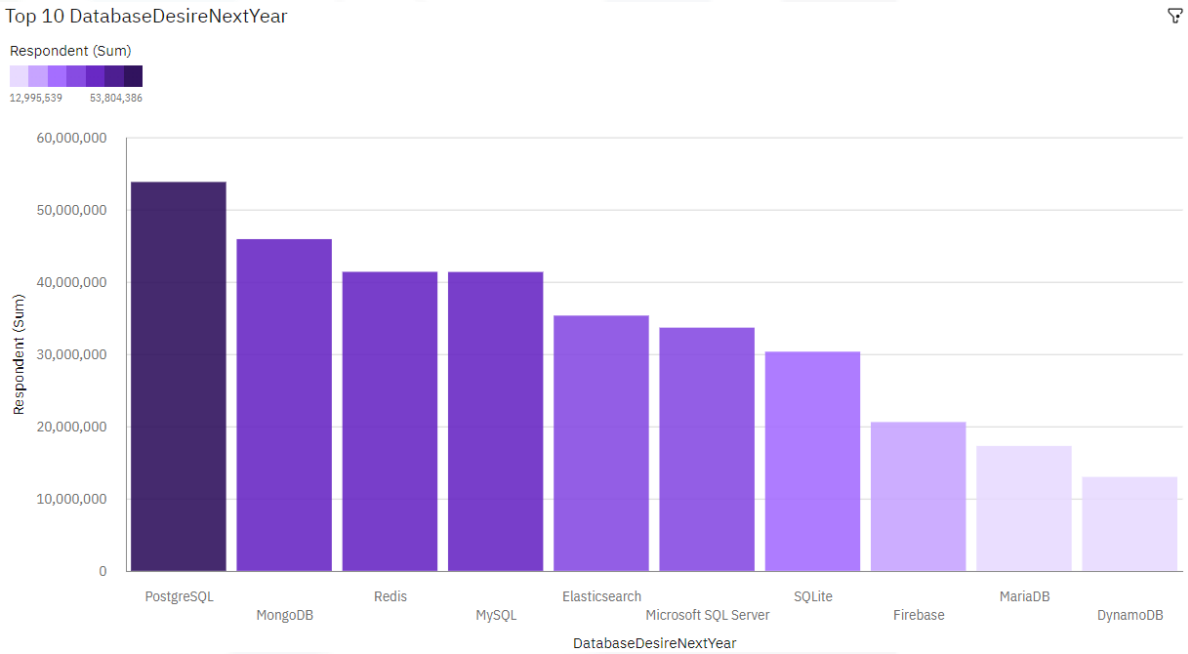
- Programming professional should consistently learn new languages
- JavaScript is a must for related career

DATABASE TRENDS

Current Year



Next Year



DATABASE TRENDS - FINDINGS & IMPLICATIONS

Findings

- There is no change for top 5 database expected next year
- Top 5 database has similar market share

Implications

- Data professional should understand multiple database languages
- There is not much change for database usage next year

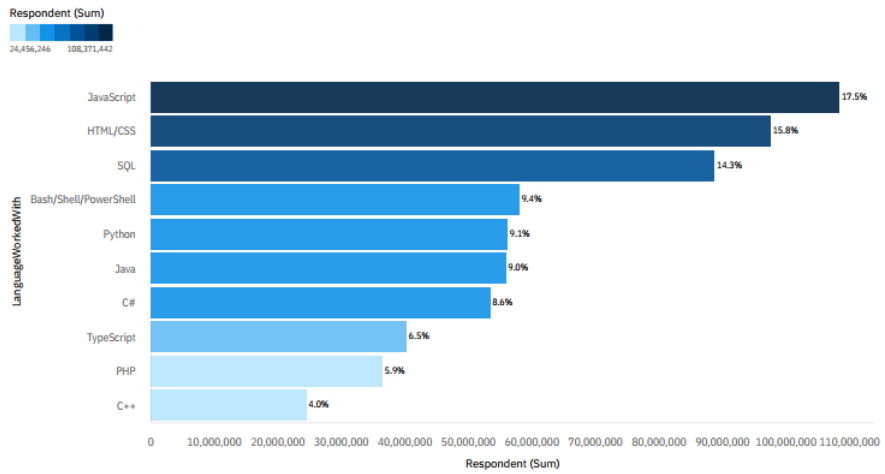
DASHBOARD

https://us1.ca.analytics.ibm.com/bi/?perspective=dashboard&pathRef=.my_folders%2FTechnology%2BDashboard_XinyuZhao&action=view&mode=dashboard&subView=model0000018173f352e0_00000003

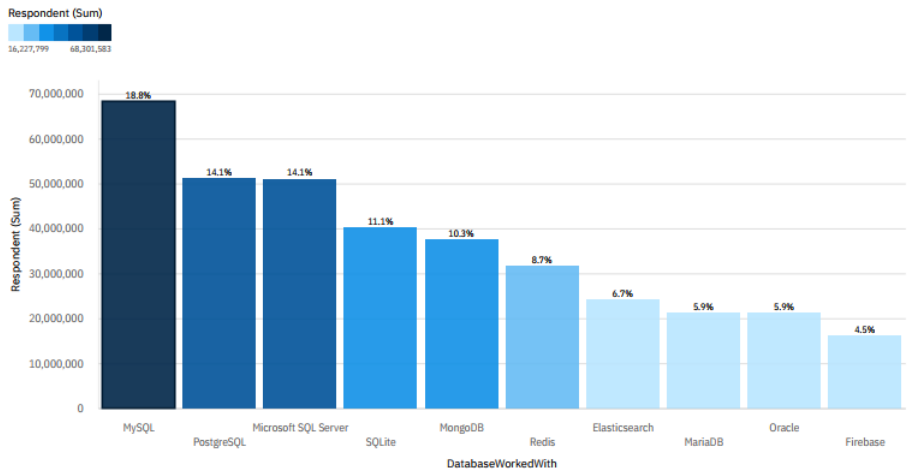
DASHBOARD-Current Technology Trends

Current Technology Usage

Top 10 Language Respondent Worked With

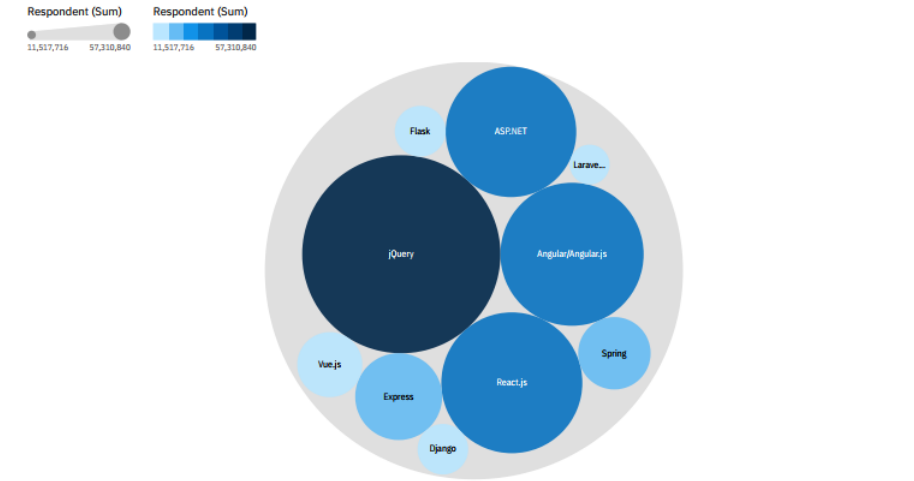


Top 10 Database Respondent Worked With



Top Platform Respondent Worked With

Top 10 WebFrame Respondent Worked With



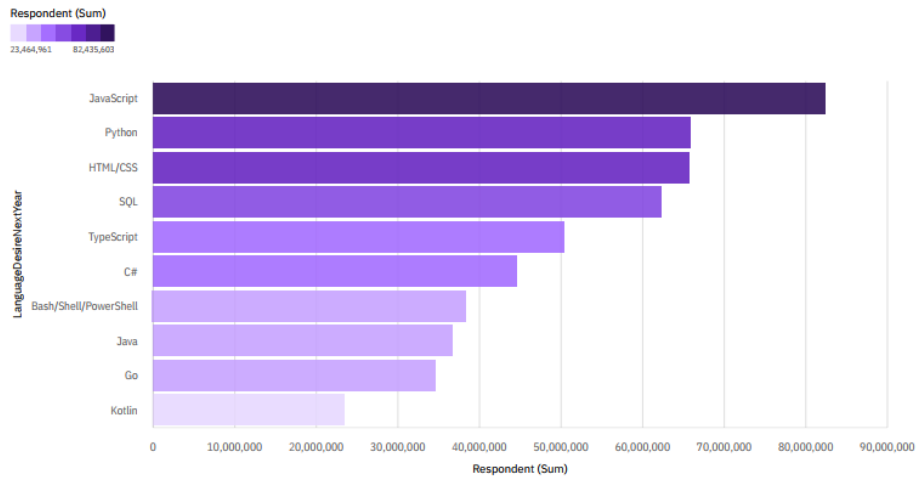
IBM De

ORK

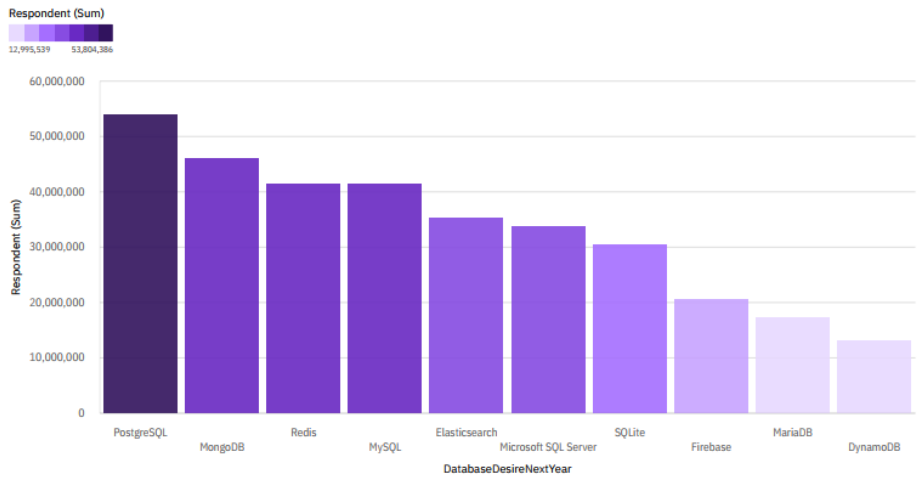
DASHBOARD-Future Tech Trend

Future Technology Trend

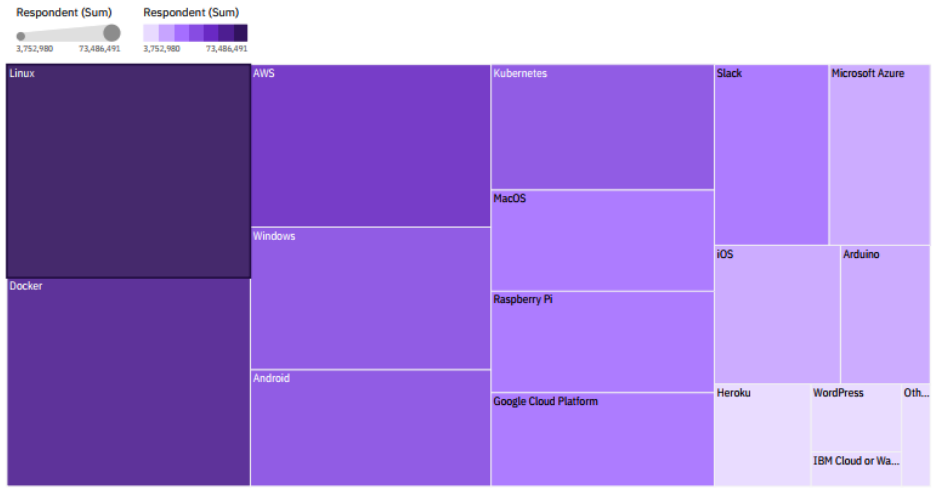
Top 10 LanguageDesireNextYear



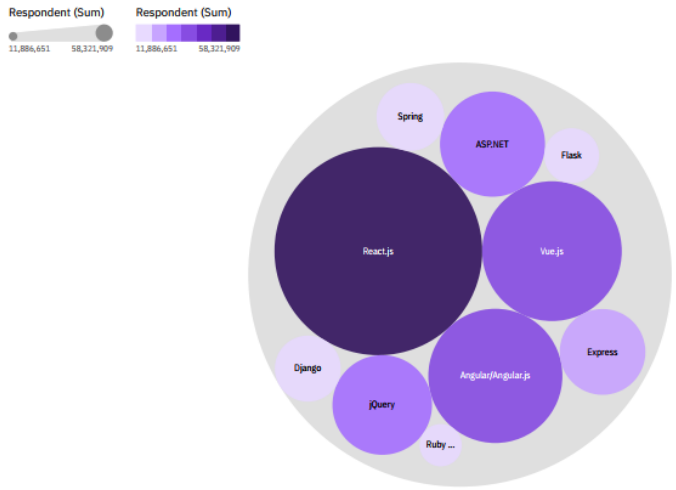
Top 10 DatabaseDesireNextYear



Top PlatformDesireNextYear



Top10 WebFrameDesireNextYear

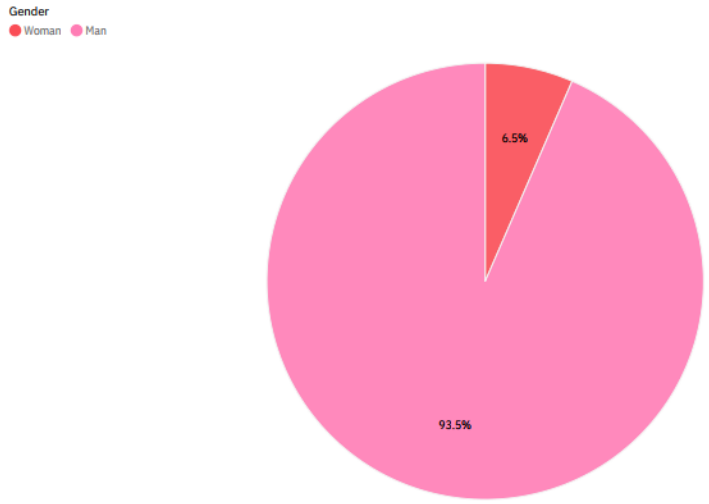


IBM De

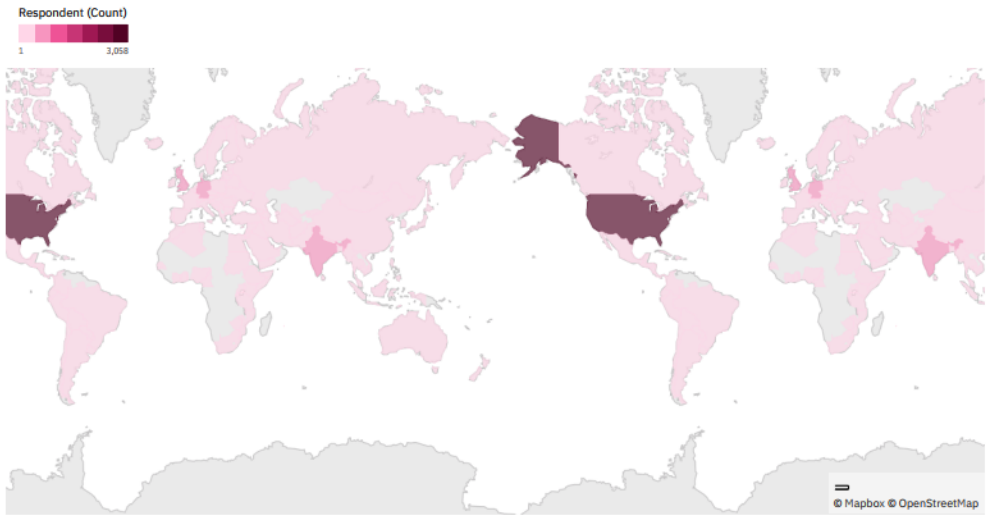
JRK

DASHBOARD-Demographic Trends

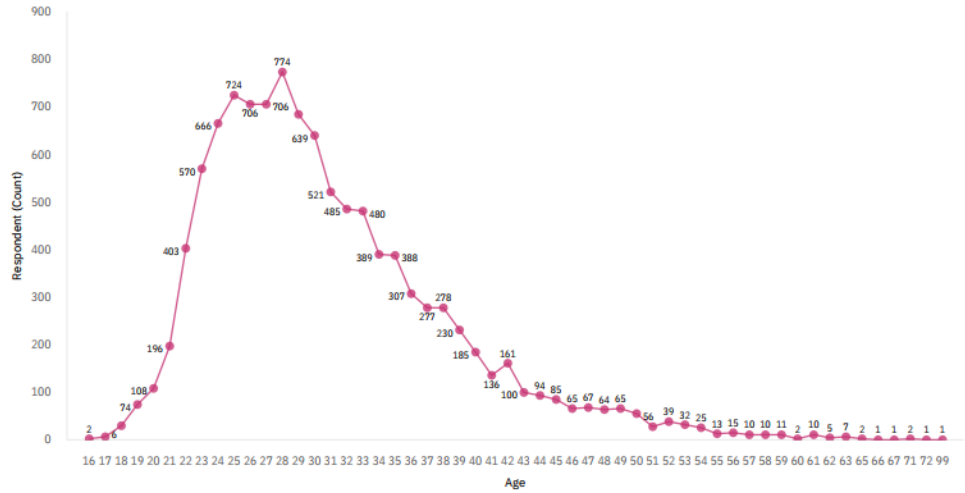
Respondent Classified by Gender



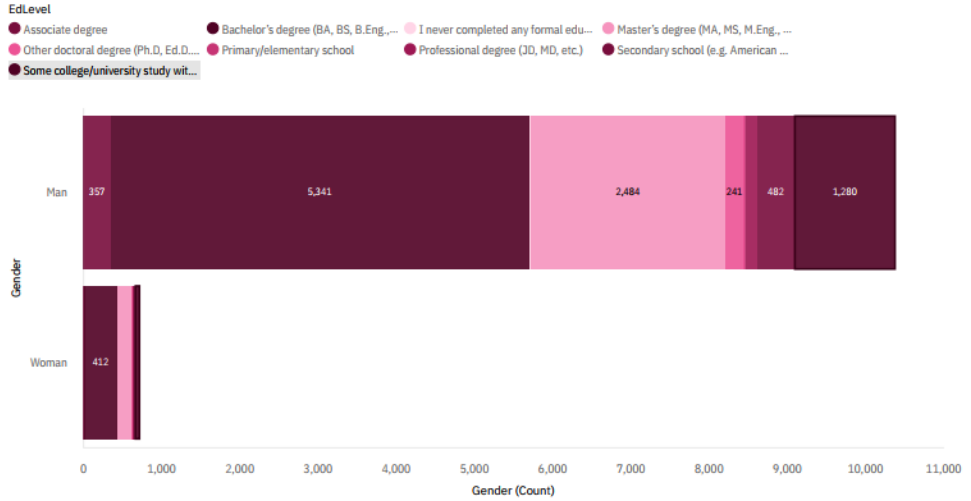
Respondent Count for Countries



Respondent Count by Age



Respondent Count by Gender, Classified by Formal Education Level



IBM I

DISCUSSION

- Most Respondents are from the US. We may need to conduct additional survey on non-US professional
- Most respondents are between 20 to 40 years old
- Too many respondents are Man. We may need to conduct further study on Women.

OVERALL FINDINGS & IMPLICATIONS

Findings

- Most Needed languages skills are PostgreSQL, JavaScript, Python, HTML/CSS, MongoDB, Linux, AWS, Docker, React.js
- Lots of programmers have Bachelor's degree
- React.js is taking over jQuery

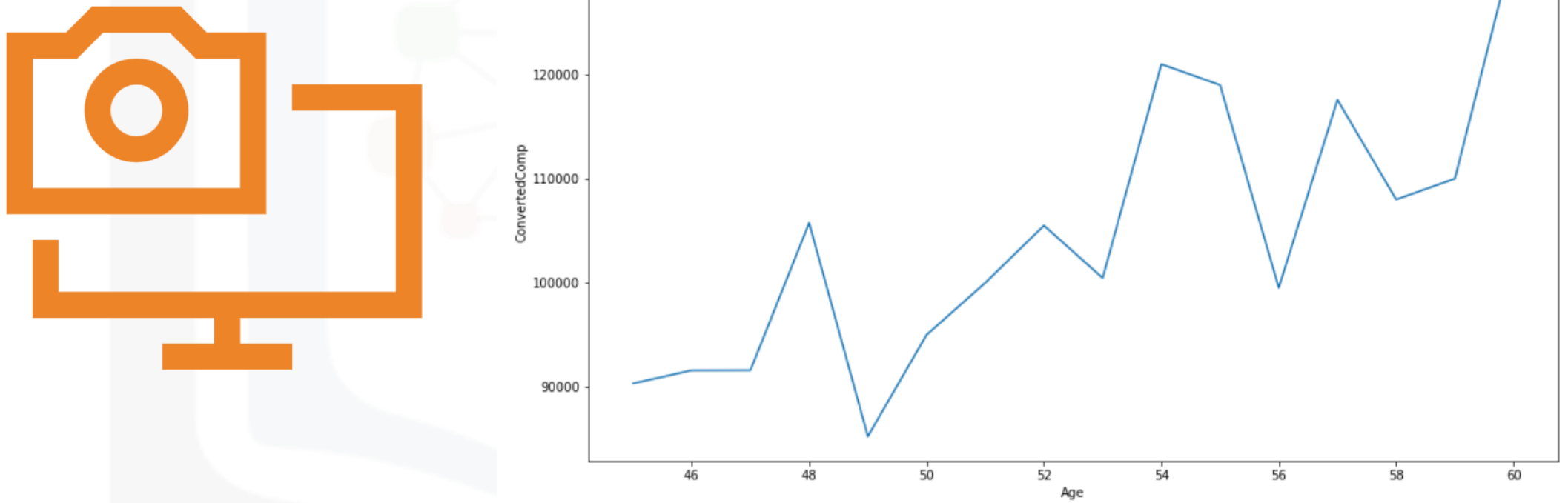
Implications

- Programming professional should understand multiple languages
- Programming professional should consistently learn new languages
- Higher education is important to learn programming
- WebFrame professionals should learn React.js

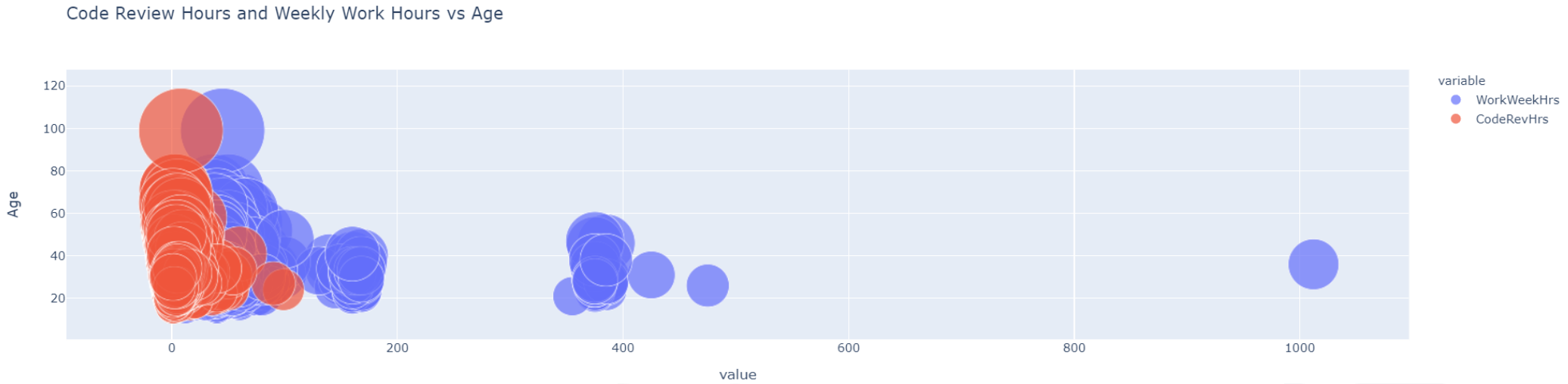
CONCLUSION

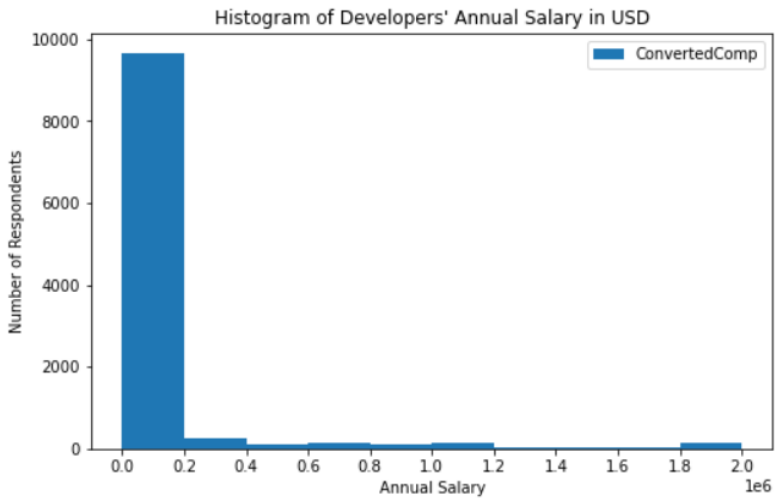
- Professional should consistently learn new languages
- Higher education is important to work in programming industry
- Database market is stabilizing
- Python, Go are expected to be much more popular next year than now

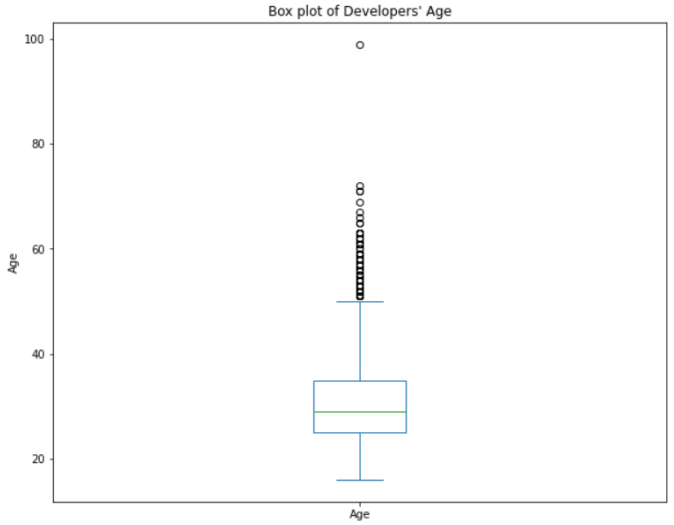
APPENDIX


Median of Developers' Annual Salary for all ages from 45 to 60
ConvertedComp
130000
120000
110000
100000
90000
ConvertedComp
46
48
50
52
54
56
58
60
Age

APPENDIX

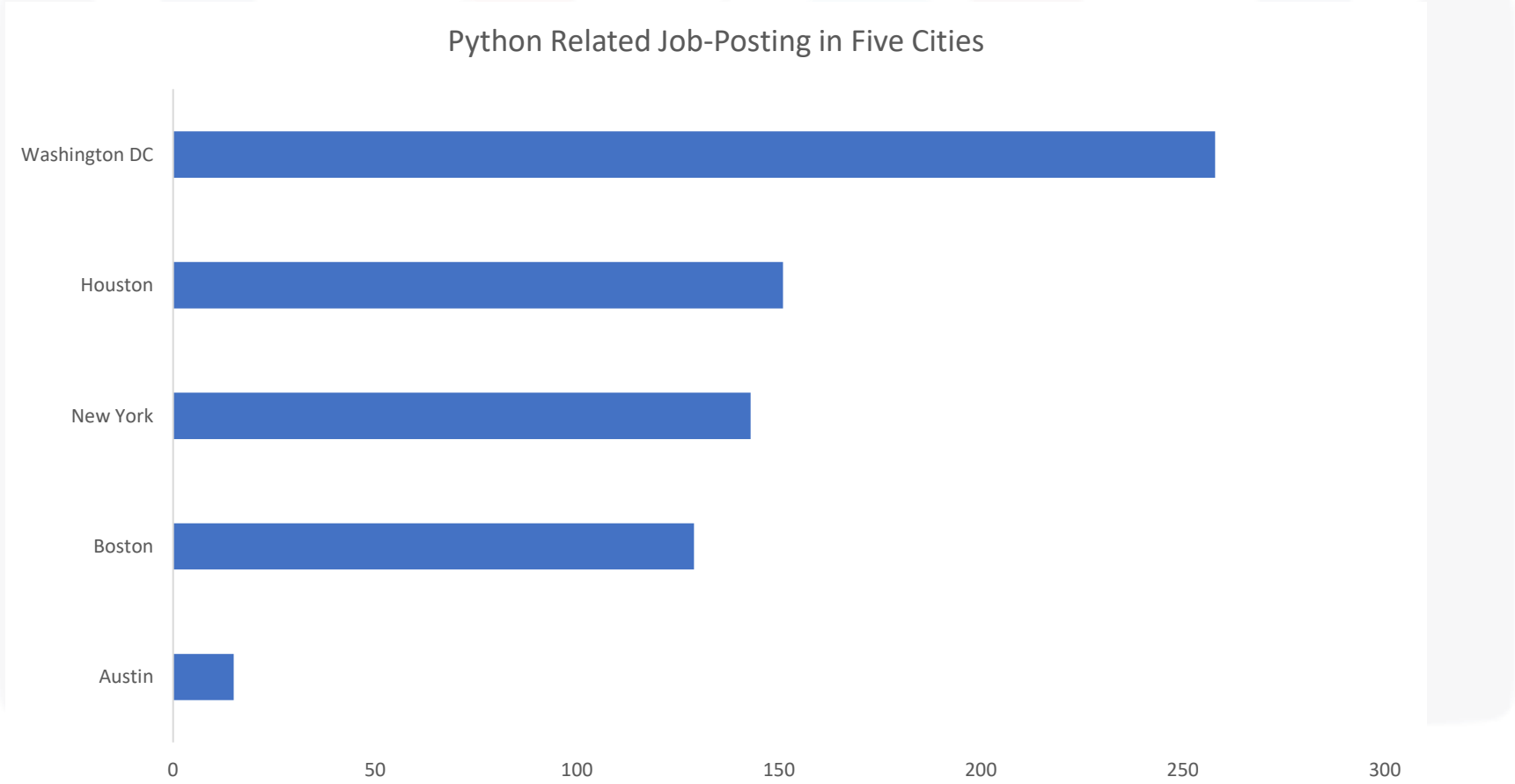

Code Review Hours and Weekly Work Hours vs Age
Age
value
variable
WorkWeekHrs
CodeRevHrs


Histogram of Developers' Annual Salary in USD
ConvertedComp
Number of Respondents
Annual Salary
1e6


Box plot of Developers' Age
Age

JOB POSTINGS

Job API - “job-postings.xlsx” - bar chart here



POPULAR LANGUAGES