Activity Report

On

Predicting Rise and Fall Of Programming Languages And Operating systems.

Submitted by

Saptarshi Mondal (1930043)
Shreya Mishra (1930045)
Somrita Dutta (1930049)
of
School of Electronics Engineering

To
B.Tech Program in Electronics and Computer Science Engineering (5th sem)

Kalinga Institute of Industrial Technology, Deemed to be University
Bhubaneswar, India

October 2021

OBJECTIVE

Predictive analysis of popularity of programming languages and mobile operating system based on questions asked with tags on Stack Overflow.

THEORY

A data analysis process broadly involves six phases which we will be implementing in our project. The following are the six phases:

1. **ASK** questions and define the problem:

We want to analyze the popularity of various programming languages.
Is R language on the rise? Will Python still continue to be all the hype that it is now?

2. **PREPARE** data by collecting and storing information

So to gather the information that we require for our objective, we decided to target the tags used in questions asked on Stack Overflow. It would give quite accurate information on how trendy or popular or how researched about a certain tag or language is. We collected data on various tags of years 2008 through 2018.

3. **PROCESS** data by cleaning and checking the information

We verified the datasets, combined this data and cleaned it to get particular tags, number of questions asked of that tag in a year and the total number of questions asked that year. We remove any null or unacceptable values. We check for duplicate tags. We arrange the year in ascending order, and the tags in alphabetically ascending order.

4. **ANALYZE** data to find patterns, relationships and trends

Now we will first use R programming in RStudio to analyze the data that we have collected and to interpret the popularity of the languages/tags and to see how exactly we can use this information to draw conclusions. Then we will be using Tableau to visualize our interpretations and to find more patterns from our data.
We will also be predicting the fate of these languages in the coming years.

5. **SHARE** data with audience with the help of visualization like graphs and plots because putting information in the image can help people understand the analysis easily.

6. **ACT** on the data and use the analysis results

In this case, we determined what programming languages are on the rise and therefore relevant and in demand for the technology industry. So we can learn these languages and implement them in modern projects.

REQUIREMENTS

1. Tableau Public
2. RStudio

OBSERVATIONS/RESULTS

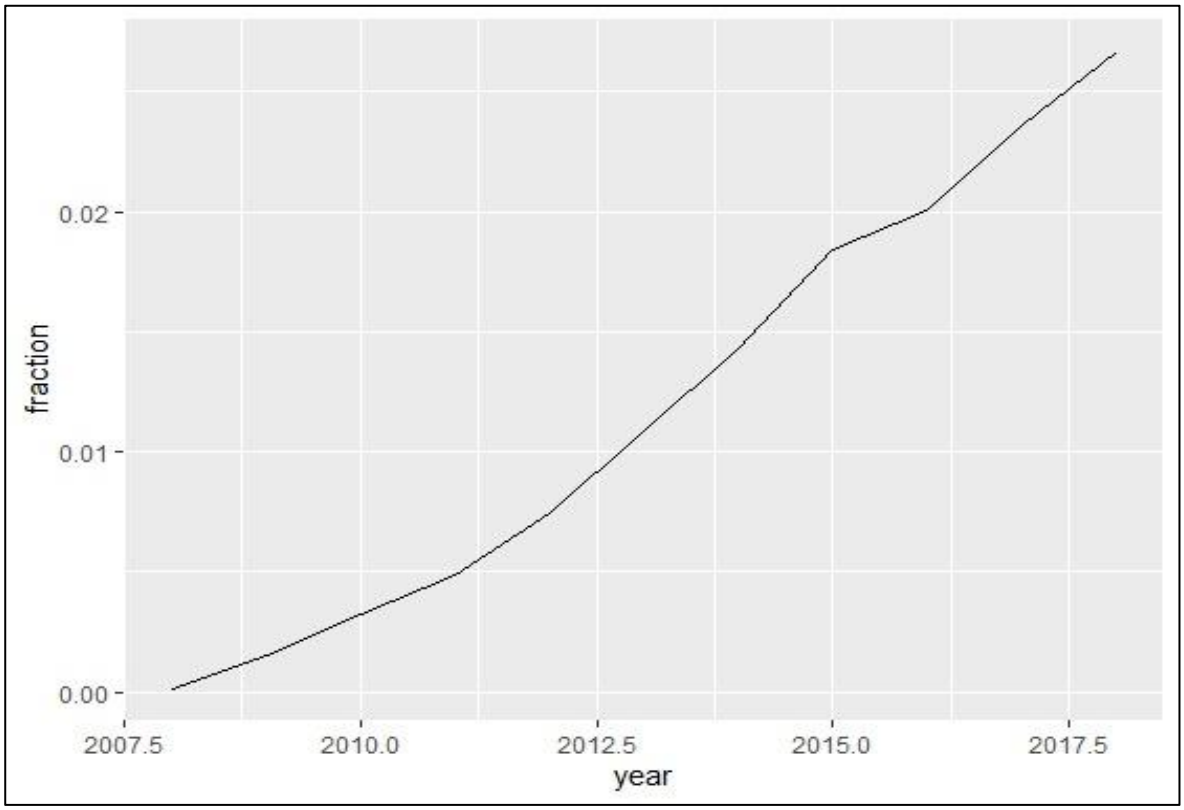


Figure 1. Popularity of R over the years

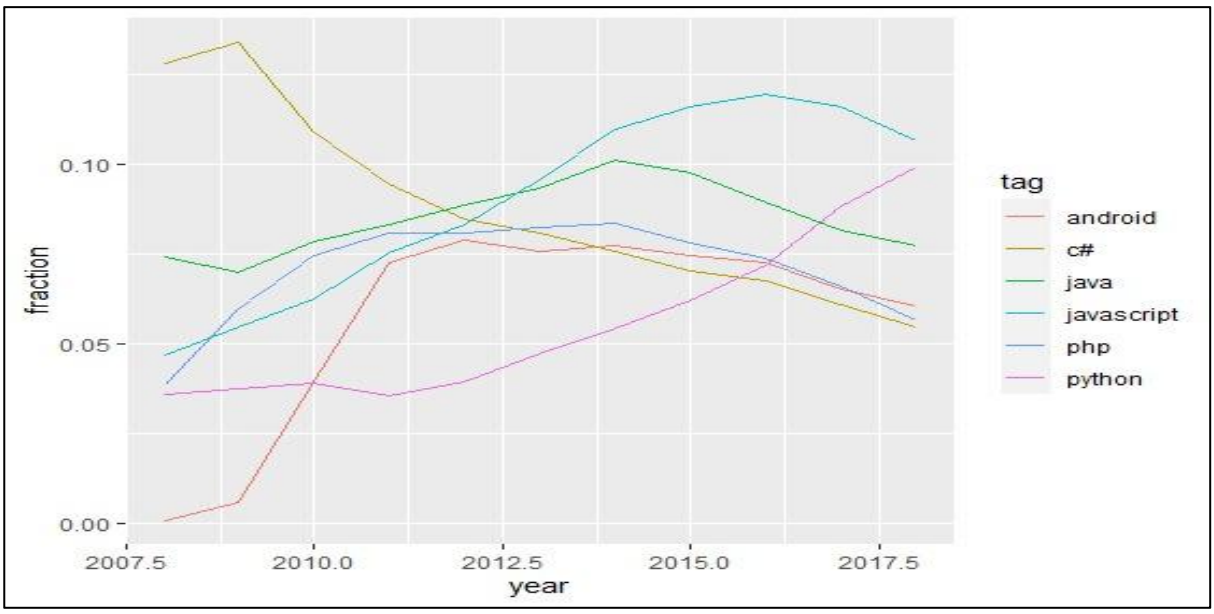


Figure 2. Analysis of the six most popular tags

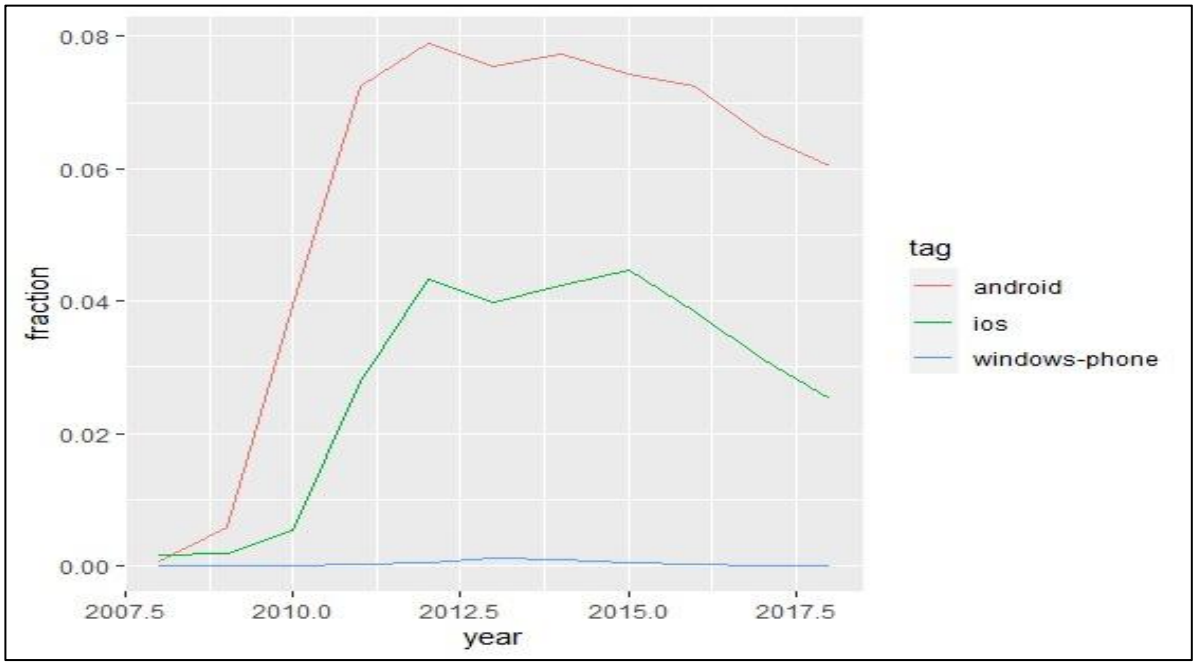


Figure 3. Analysis of R and its packages

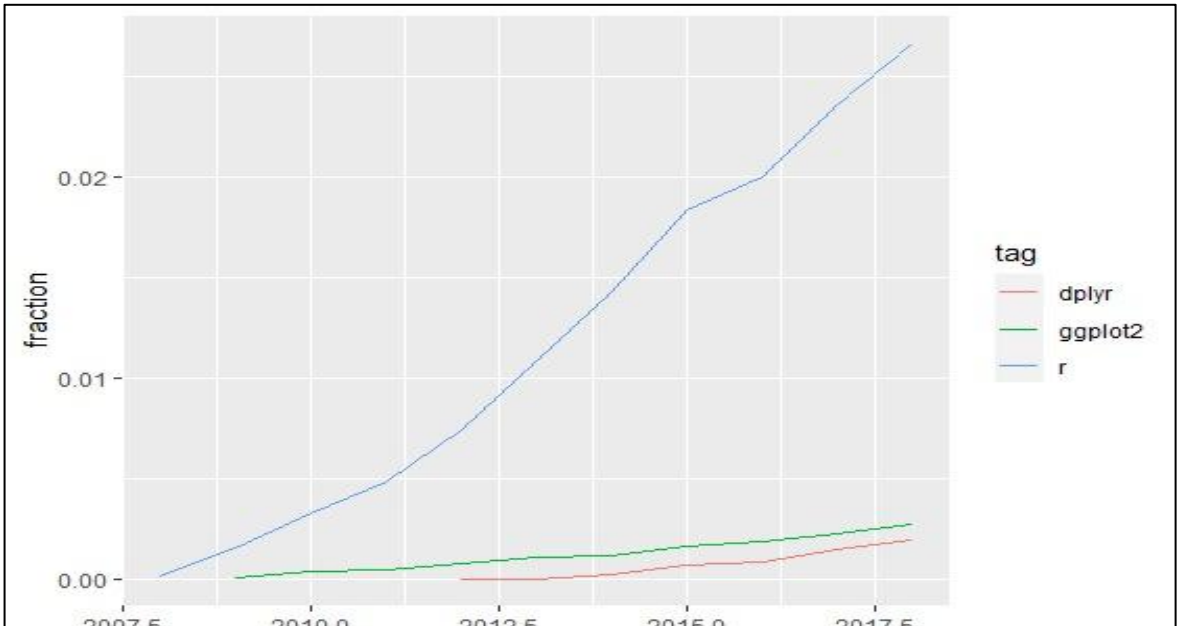


Figure 4. Analysis of mobile operating systems

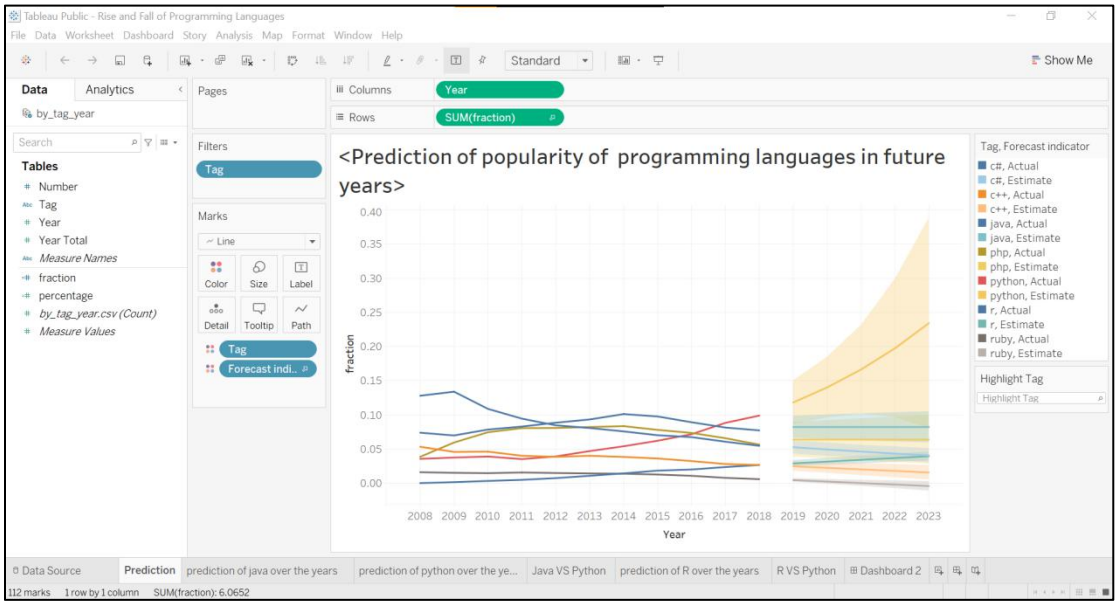


Figure 5. Prediction of popularity of programming languages in future years

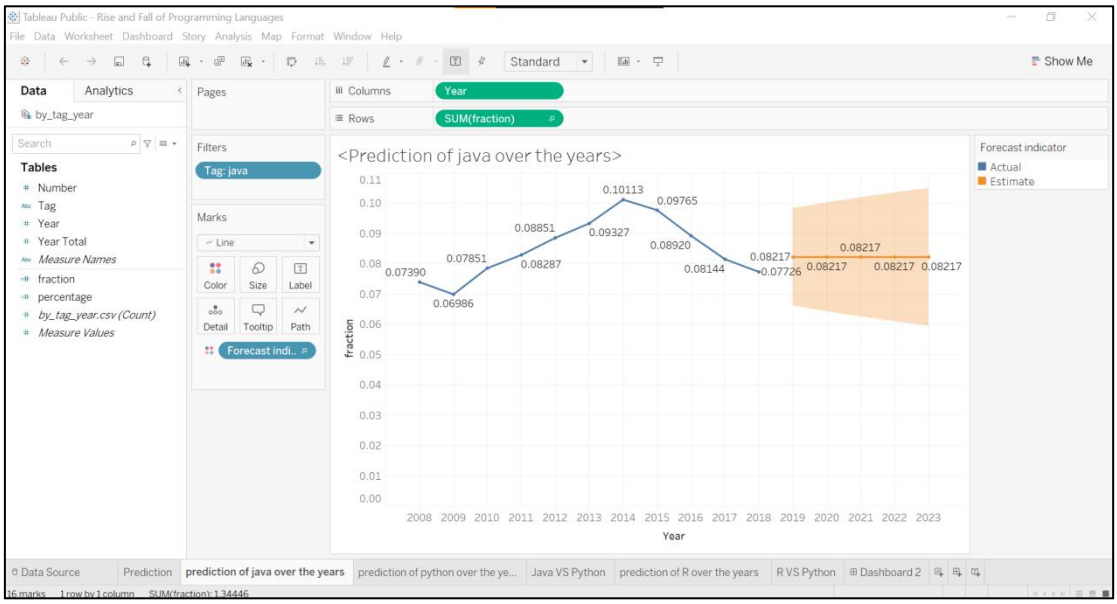


Figure 6. Prediction of popularity of Java

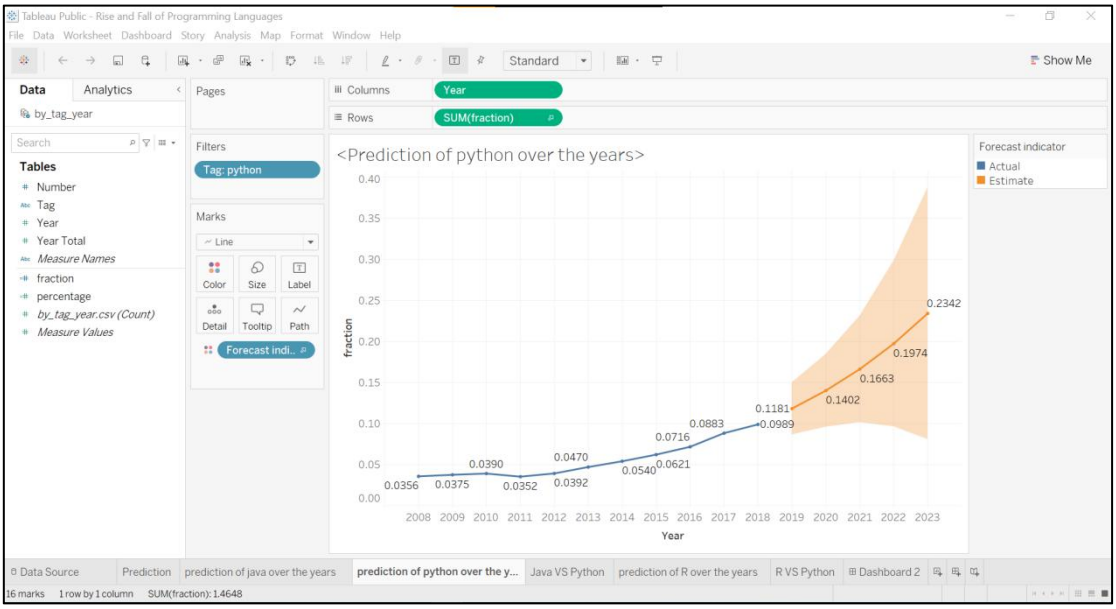


Figure 7. Prediction of popularity of Python

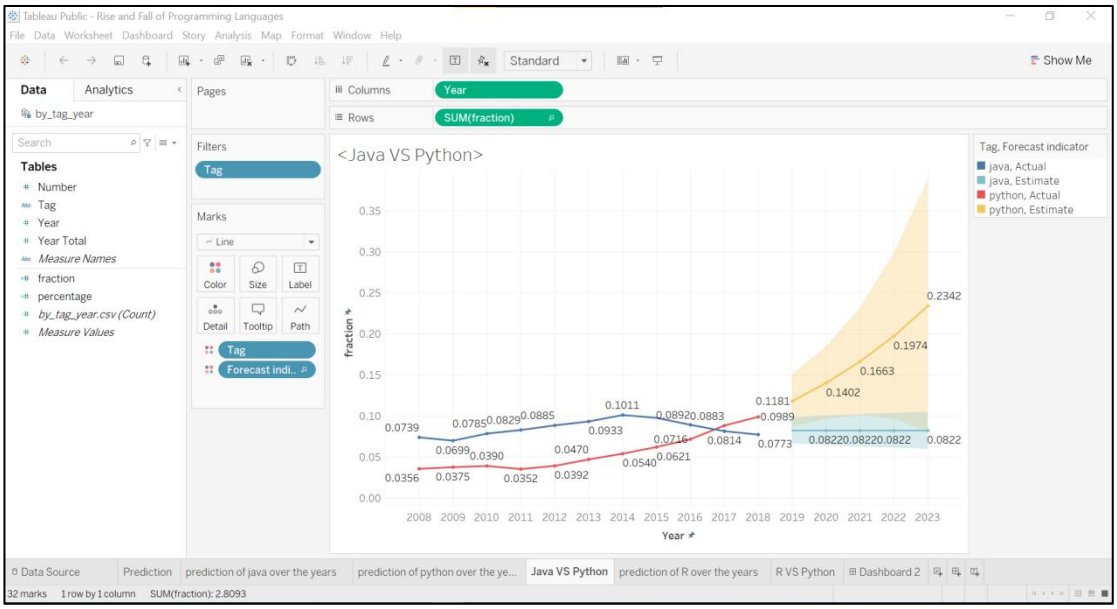


Figure 8. Prediction of popularity of Java vs Python

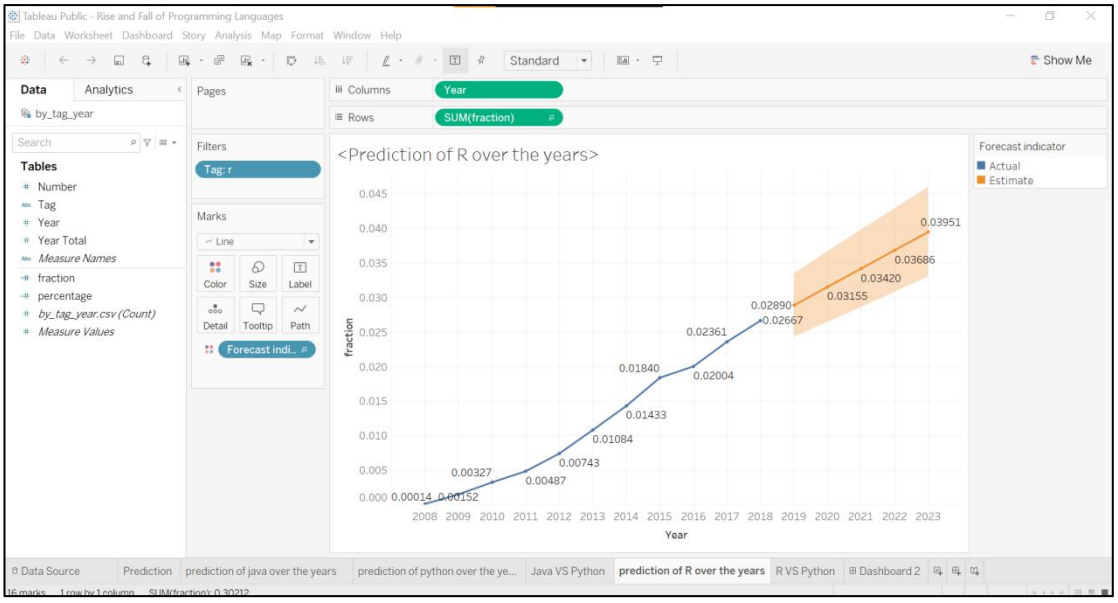


Figure 9. Prediction of popularity of R

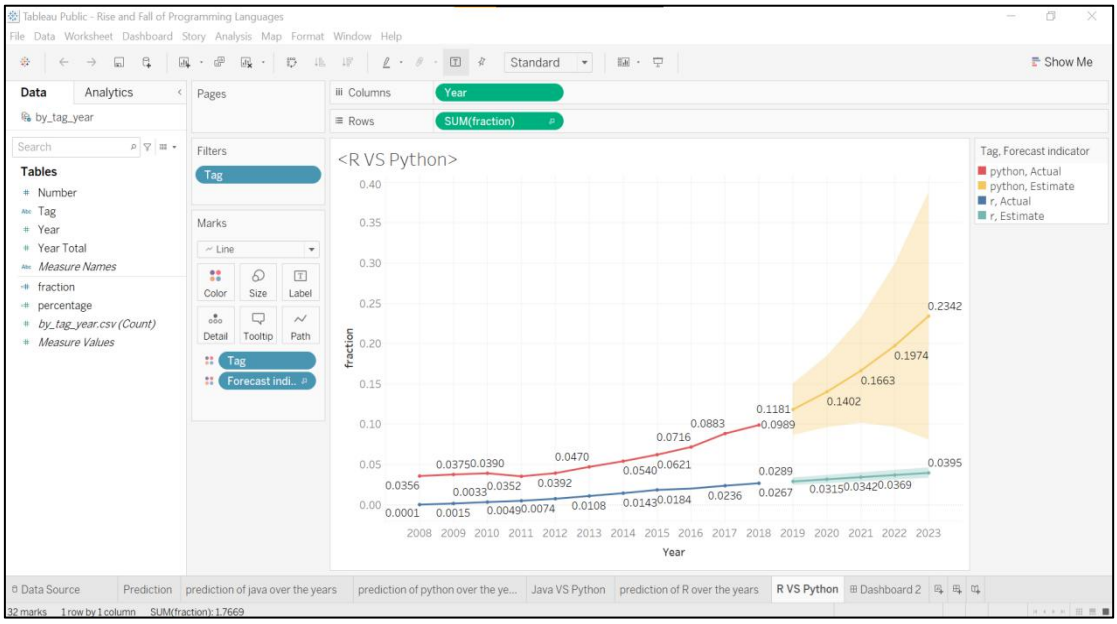


Figure 10. Prediction of popularity of R vs Python

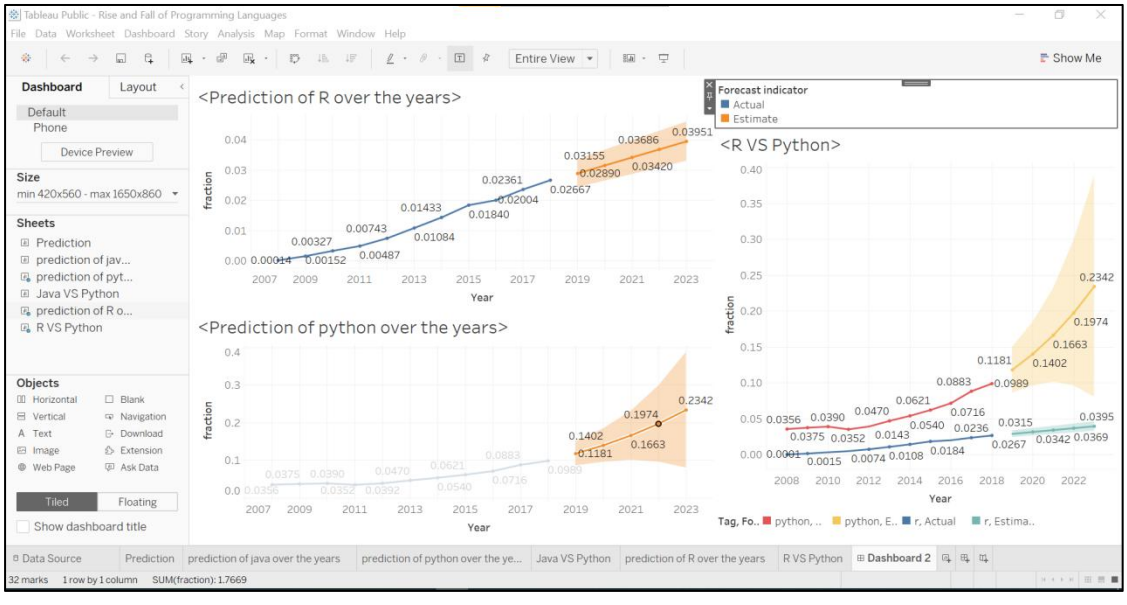


Figure 11. Dashboard

DISCUSSION OF RESULTS

- **Figure 1** indicates that R started gaining momentum in 2008 and it has been increasing ever since.
- **Figure 2** shows a lot of changes in what programming languages are most asked about. C# started getting fewer questions than it used to around 2009, and Python has grown quite impressively because it is being used in ML and AI and data analytics for applications in industrial sectors.
- In **Figure 3**, we can see the dplyr and ggplot2 tags may not have as many questions as R, but we can tell they're both growing quickly as well, clearly indicating the rise of R programming language
- In **Figure 4**, we can see how the android and ios has risen to popularity while windows operating systems for phone became obsolete and eventually discontinued in 2017.
- **Figure 5** shows the prediction of popularity of programming languages in future years. Based on our dataset we use the forecast function of Tableau to predict the popularity of programming languages in future years. Forecasting in Tableau uses a technique known as exponential smoothing. Forecast algorithms try to find a regular pattern in measures that can be continued into the future.From this graph we see that C# and C++ show a slight decline in their popularity while java and php will no change to that of now. Both R and python show an increase in their popularity. We also realize that ruby will lose its importance completely.

- **Figure 6** shows Java faced an initial backlash but soon rose to prominence and reached its maximum popularity in 2014 from where we see a steep decline in its popularity. We predict that the popularity of java will become stagnant in the upcoming years.
- In **Figure 7**, we see that Python had a stable start and took its time to gain its popularity. It is clearly evident here that demand of python will reach new heights in the upcoming years.

- If we compare Java with Python (**Figure 8**) we realize that Java loses its popularity in 2017 right about the time when python shows an increase in its fame. In our predicted future Python will be a more preferred language as compared to Java.
- With the help of our dataset we see in **Figure 9,** that R shows a steady increase in its popularity right from 2008. We predict that in future in R will continue to increase in its popularity.So, the question now is Python or R which language is going to dominate in future years?
- In **Figure 10**, we see that, even though R will show a rise in popularity, python will flourish in future years. It is clearly evident that the ratio of number of question asked to total number of question in that year is estimated to be 0.0395 for R, which is 0.2342 for python.

- **Figure 11** is the dashboard representation of comparison between the popularity of R and Python. In this case we can view all three plots simultaneously which helps in better visualization of our results.

CONCLUSION

Python based machine learning and artificial intelligence has found growing application in healthcare industries. R in data science is used to handle, store and analyze data. Its applications are found statistical modeling, in finance and social media. From our analysis and prediction, it is quite evident that Python and R are on-demand programming languages that will also have a major relevance in future and rightly so because of their numerous application.

REFERENCES

- *Rdocumentation*, "Search all R packages on CRAN and Bioconductor", 2019.[Online] Available:https://www.rdocumentation.org/ [Accessed on:Oct. 23, 2021].

- *Manosantoniou*,"Predicting the future popularity of programming languages", 2015.[Online]Available:https://www.manosantoniou.com/post/predicting-the-future-popularity-of-programming-languages/ [Accessed on:Oct. 22, 2021].

- *Coursera*,"Foundations:data,data,everywhere!",2020.[Online] Available:https://www.coursera.org/learn/foundations-data/home/welcome [Accessed on: Oct. 20, 2021].

APPENDIX:

```
library(readr)

library(dplyr)

by_tag_year <- read_csv("C:/Users/KIIT/Desktop/Rise and Fall of Programming Languages/datasets/by_tag_year.csv")

print(by_tag_year)
```

```
## # A tibble: 40,518 x 4
##     year tag           number year_total
##    <dbl> <chr>          <dbl>      <dbl>
##  1  2008 .htaccess         54      58390
##  2  2008 .net            5910      58390
##  3  2008 .net-2.0         289      58390
##  4  2008 .net-3.5         319      58390
##  5  2008 .net-4.0           6      58390
##  6  2008 .net-assembly      3      58390
##  7  2008 .net-core          1      58390
##  8  2008 2d                42      58390
##  9  2008 32-bit            19      58390
## 10  2008 32bit-64bit        4      58390
## # ... with 40,508 more rows
```

```
by_tag_year_fraction <-    by_tag_year %>%

   mutate(fraction = number / year_total)

print(by_tag_year_fraction)
```

```
## # A tibble: 40,518 x 5
##     year tag           number year_total  fraction
##    <dbl> <chr>          <dbl>      <dbl>     <dbl>
##  1  2008 .htaccess         54      58390 0.000925
##  2  2008 .net            5910      58390 0.101
##  3  2008 .net-2.0         289      58390 0.00495
##  4  2008 .net-3.5         319      58390 0.00546
##  5  2008 .net-4.0           6      58390 0.000103
##  6  2008 .net-assembly      3      58390 0.0000514
##  7  2008 .net-core          1      58390 0.0000171
##  8  2008 2d                42      58390 0.000719
##  9  2008 32-bit            19      58390 0.000325
## 10  2008 32bit-64bit        4      58390 0.0000685
## # ... with 40,508 more rows
```

```
r_over_time <- by_tag_year_fraction %>%
  filter(tag == "r")
print(r_over_time)
```

```
## # A tibble: 11 x 5
##     year tag   number year_total fraction
##    <dbl> <chr>  <dbl>      <dbl>    <dbl>
##  1  2008 r          8      58390 0.000137
##  2  2009 r        524     343868 0.00152
##  3  2010 r       2270     694391 0.00327
##  4  2011 r       5845    1200551 0.00487
##  5  2012 r      12221    1645404 0.00743
##  6  2013 r      22329    2060473 0.0108
##  7  2014 r      31011    2164701 0.0143
##  8  2015 r      40844    2219527 0.0184
##  9  2016 r      44611    2226072 0.0200
## 10  2017 r      54415    2305207 0.0236
## 11  2018 r      28938    1085170 0.0267
```

```
library(ggplot2)
ggplot(r_over_time) +
  geom_line(aes(x = year, y = fraction))
```

```
sorted_tags <- by_tag_year %>%
  group_by(tag) %>%
  summarize(tag_total = sum(number)) %>%
```

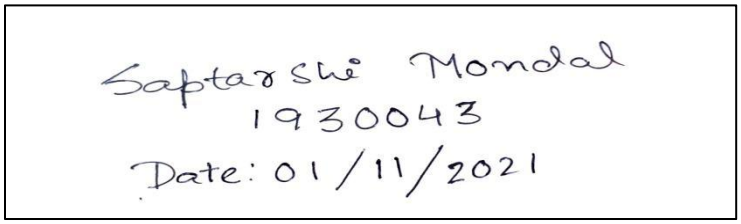
```
  arrange(desc(tag_total))
```

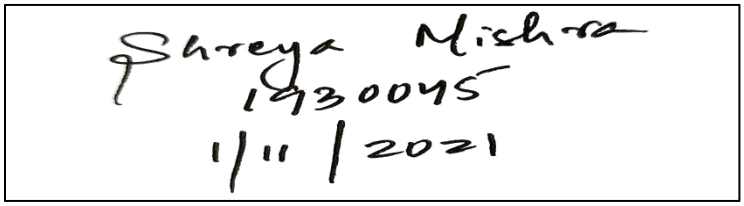
```
print(sorted_tags)
```

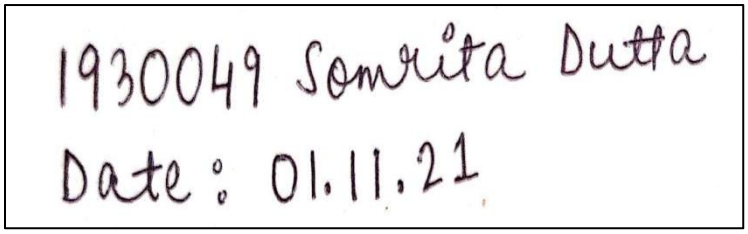
```
## # A tibble: 4,080 x 2
##    tag        tag_total
##    <chr>          <dbl>
##  1 javascript   1632049
##  2 java         1425961
##  3 c#           1217450
##  4 php          1204291
##  5 android      1110261
##  6 python        970768
##  7 jquery        915159
##  8 html          755341
##  9 c++           574263
## 10 ios           566075
## # ... with 4,070 more rows
```

```
highest_tags <- head(sorted_tags$tag)

by_tag_subset <- by_tag_year_fraction %>%
  filter(tag %in% highest_tags)

ggplot(by_tag_subset, aes(x = year,y = fraction,color = tag)) +geom_line()

selected_tags <- c("r", "dplyr", "ggplot2")

selected_tags_over_time <- by_tag_year_fraction %>%
  filter(tag %in% selected_tags)

ggplot(selected_tags_over_time, aes(x = year,y = fraction,color = tag)) + geom_line()

my_tags <- c("android", "ios", "windows-phone")

by_tag_subset <- by_tag_year_fraction %>%
  filter(tag %in% my_tags)

ggplot(by_tag_subset, aes(x = year,y = fraction,color = tag)) +geom_line()
```

STUDENT SIGNATURES







SIGNATURE OF THE CONCERNED FACULTY