

R Notebook

James Weaver

Programming Language Trends Analysis

Section 1: Load Necessary Libraries

```
library(readr)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##   filter, lag

## The following objects are masked from 'package:base':
##
##   intersect, setdiff, setequal, union
```

```
library(ggplot2)
```

Section 2: Load and Preview Dataset

```
# Load the dataset
data <- read_csv("stack_overflow_data.csv")
```

```
## Rows: 420066 Columns: 4
## -- Column specification -----
## Delimiter: ","
## chr (1): tag
## dbl (3): year, num_questions, year_total
##
## i Use 'spec()' to retrieve the full column specification for this data.
## i Specify the column types or set 'show_col_types = FALSE' to quiet this message.
```

```
# Preview the dataset
head(data, 5)
```

```
## # A tibble: 5 x 4
##   year tag          num_questions year_total
##   <dbl> <chr>          <dbl>         <dbl>
```

```
## 1 2008 treeview      69      168541
## 2 2008 scheduled-tasks 30      168541
## 3 2008 specifications 21      168541
## 4 2008 rendering     35      168541
## 5 2008 http-post      6      168541
```

Section 3: Data Transformation

```
# Calculate the percentage of questions per tag per year
data <- data %>%
  mutate(percentage = (num_questions / year_total) * 100)
```

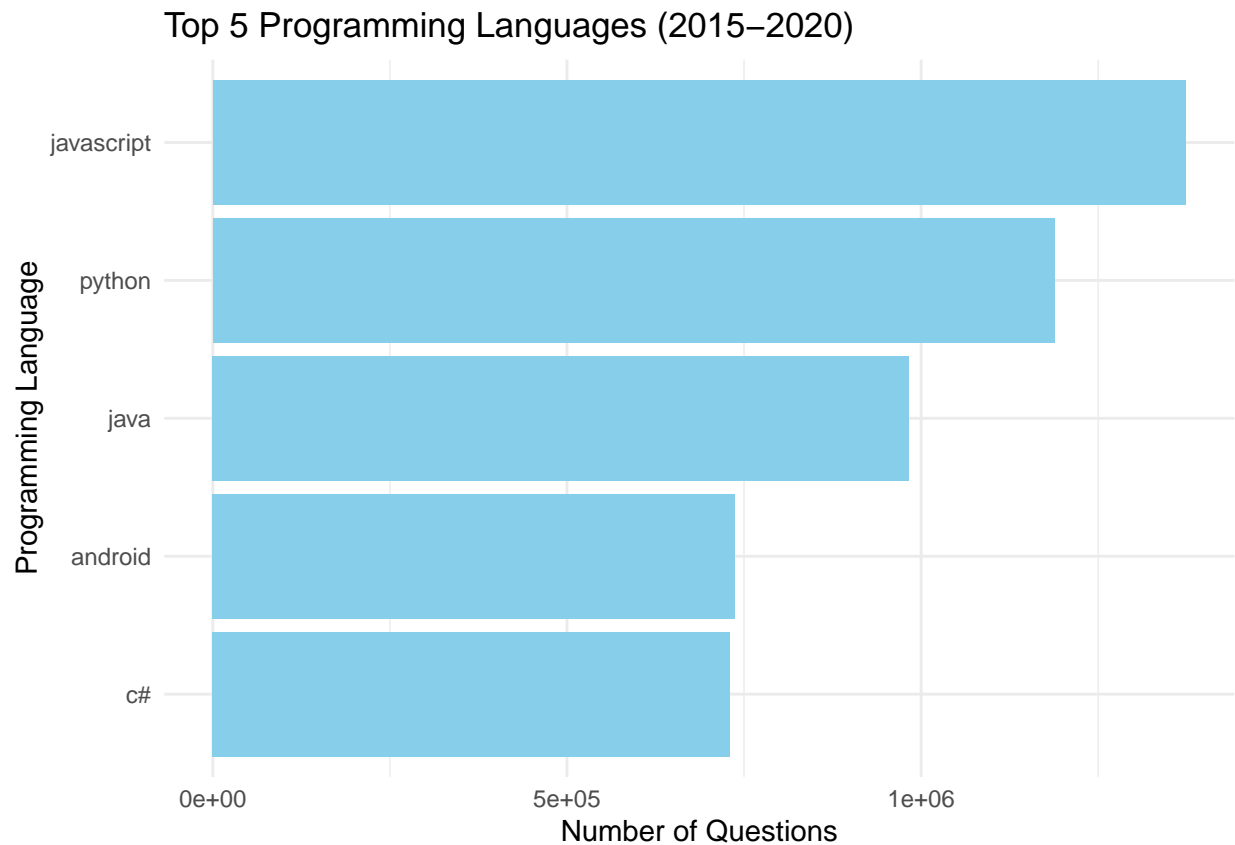
Section 4: Analyze Top Programming Languages

```
# Top Tags Analysis
top_tags <- data %>%
  filter(year %in% 2015:2020) %>%
  group_by(tag) %>%
  summarise(total = sum(num_questions)) %>%
  arrange(desc(total)) %>%
  head(5)
top_tags
```

```
## # A tibble: 5 x 2
##   tag      total
##   <chr>    <dbl>
## 1 javascript 1373634
## 2 python    1187838
## 3 java      982747
## 4 android   737330
## 5 c#        730045
```

Section 5: Visualize Top Programming Languages

```
# Visualize Top Tags for 2015-2020
ggplot(top_tags, aes(x = total, y = reorder(tag, total))) +
  geom_col(fill = "skyblue") +
  labs(
    title = "Top 5 Programming Languages (2015-2020)",
    x = "Number of Questions",
    y = "Programming Language"
  ) +
  theme_minimal()
```

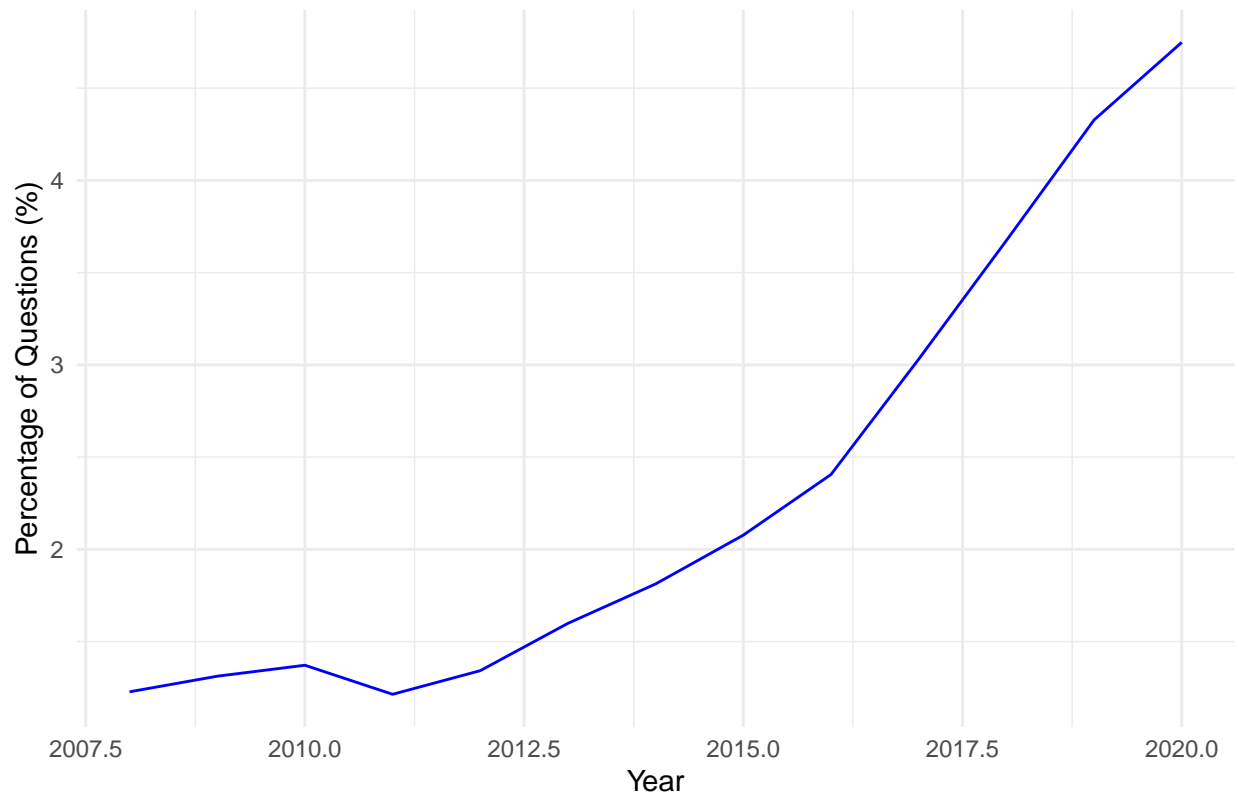


Section 6: Trend Analysis for Specific Languages

```
# Percentage Trend for a Specific Language
plot_language_trend <- function(data, language) {
  language_data <- data %>% filter(tag == language)
  ggplot(language_data, aes(x = year, y = percentage)) +
    geom_line(color = "blue") +
    labs(
      title = paste("Trend of", toupper(language), "Popularity Over Time"),
      x = "Year",
      y = "Percentage of Questions (%)"
    ) +
    theme_minimal()
}

plot_language_trend(data, "python")
```

Trend of PYTHON Popularity Over Time



Section 7: Summary Insights

```
# Summary Insights
```

```
cat("### Key Insights:\n")
```

```
## ### Key Insights:
```

```
cat("1. Python has consistently grown in popularity, surpassing many other programming languages.\n")
```

```
## 1. Python has consistently grown in popularity, surpassing many other programming languages.
```

```
cat("2. Other languages like JavaScript remain stable, while some older languages show declining trends\n")
```

```
## 2. Other languages like JavaScript remain stable, while some older languages show declining trends.
```

```
cat("3. This analysis provides insights into programming language trends to guide career and learning decisions.\n")
```

```
## 3. This analysis provides insights into programming language trends to guide career and learning decisions.
```