



How can you determine which programming languages and technologies are most widely used? Which languages are gaining or losing popularity, helping you decide where to focus your efforts?

One excellent data source is Stack Overflow, a programming question-and-answer site with more than 16 million questions on programming topics. Each Stack Overflow question is tagged with a label identifying its topic or technology. By counting the number of questions related to each technology, you can estimate the popularity of different programming languages.

In this project, you will use data from the Stack Exchange Data Explorer to examine the relative popularity of R compared to other programming languages.

You'll work with a dataset containing one observation per tag per year, including the number of questions for that tag and the total number of questions that year.

stack_overflow_data.csv

Column	Description
year	The year the question was asked (2008-2020)
tag	A word or phrase that describes the topic of the question, such as the programming language
num_questions	The number of questions with a certain tag in that year
year_total	The total number of questions asked in that year

```
# Load necessary packages
library(readr)
library(dplyr)
library(ggplot2)

#import pandas as pd
#import matplotlib.pyplot as plt
```

Hidden output

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

#data=pd.read_csv("stack_overflow_data.csv")
```

Hidden output

View the dataset
head(data)

#data.head()

index	...	↑↓	year	...	↑↓	tag	...	↑↓	num_questions	...	↑↓	year_total	...
1					2008	treeview					69		168
2					2008	scheduled-tasks					30		168
3					2008	specifications					21		168
4					2008	rendering					35		168
5					2008	http-post					6		168
6					2008	static-assert					1		168

Rows: 6  Expand

```
df <- subset(data, tag %in% c("r"))
df$percentage <- (df$num_questions / df$year_total)*100
r_2020 <- subset(df, year %in% c(2020))

#df=data[data['tag'].isin(['r'])]
#df['percentage']=df['num_questions']/df['year_total']
#r_2020=df[df['year'].isin([2020])]
```

```
df2 <- subset(data, year %in% c(2015, 2016, 2017, 2018, 2019, 2020))
df2 <- aggregate(num_questions ~ tag, data = df2, sum)
df2 <- df2[order(-df2$num_questions), ]
highest_tags <- head(df2, 5) # top 5 rows

#df2=data[data['year'].isin([2015, 2016, 2017, 2018, 2019, 2020])]
#df2=df2.groupby('tag')['num_questions'].sum()
#df2=df2.sort_values(ascending=False)
#highest_tags=df2.head().index
```