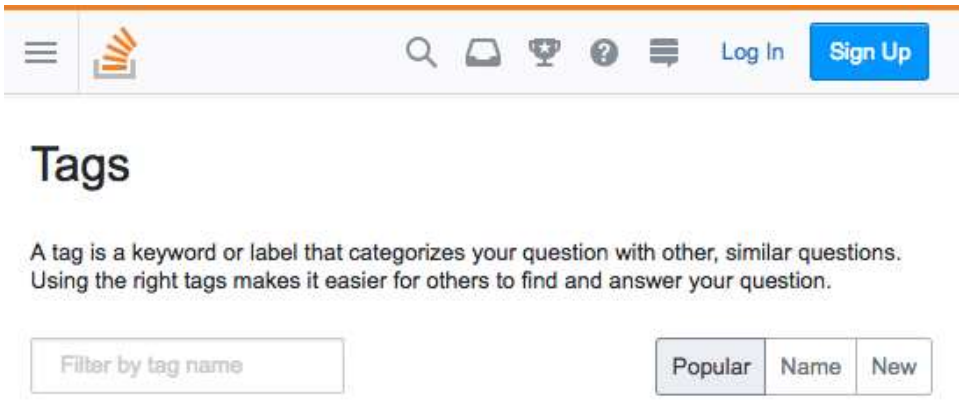
1. Data on tags over time

How can we tell what programming languages and technologies are used by the most people? How about what languages are growing and which are shrinking, so that we can tell which are most worth investing time in?

One excellent source of data is Stack Overflow, a programming question and answer site with more than 16 million questions on programming topics. By measuring the number of questions about each technology, we can get an approximate sense of how many people are using it. We're going to use open data from the Stack Exchange Data Explorer to examine the relative popularity of languages like R, Python, Java and Javascript have changed over time.

Each Stack Overflow question has a **tag**, which marks a question to describe its topic or technology. For instance, there's a tag for languages like R or Python, and for packages like ggplot2 or pandas.



We'll be working with a dataset with one observation for each tag in each year. The dataset includes both the number of questions asked in that tag in that year, and the total number of questions asked in that year.

```
# Load libraries
library(readr)
library(dplyr)

# Load dataset
by_tag_year <- read_csv("datasets/by_tag_year.csv")

# Inspect the dataset
by_tag_year
```

```
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


Rows: 40518 Columns: 4
── Column specification ────────────────────────────────────────────────────────
Delimiter: ","
chr (1): tag
dbl (3): year, number, 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.
```

in...	y...	tag	n...	year_total
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
11	2008	3d	73	58390
12	2008	64bit	149	58390
13	2008	abap	10	58390
14	2008	absolute	1	58390
15	2008	abstract	5	58390
16	2008	abstract-class	27	58390

Rows: 25,000 ⚠ truncated from 40,518 rows

2. Now in fraction format

This data has one observation for each pair of a tag and a year, showing the number of questions asked in that tag in that year and the total number of questions asked in that year. For instance, there were 54 questions asked about the `.htaccess` tag in 2008, out of a total of 58390 questions in that year.

Rather than just the counts, we're probably interested in a percentage: the fraction of questions that year that have that tag. So let's add that to the table.

```
# Add fraction column
by_tag_year_fraction <- by_tag_year %>%
    mutate(fraction = number/year_total)
```

```
# Print the new table
by_tag_year_fraction
```

i	!	tag	ı	year...	fraction
1	2008	.htaccess	54	58390	0.0009
2	2008	.net	5910	58390	0.1012
3	2008	.net-2.0	289	58390	0.0049
4	2008	.net-3.5	319	58390	0.0055
5	2008	.net-4.0	6	58390	0.0001
6	2008	.net-assembly	3	58390	0.0001
7	2008	.net-core	1	58390	0
8	2008	2d	42	58390	0.0007
9	2008	32-bit	19	58390	0.0003
10	2008	32bit-64bit	4	58390	0.0001
11	2008	3d	73	58390	0.0013
12	2008	64bit	149	58390	0.0026
13	2008	abap	10	58390	0.0002
14	2008	absolute	1	58390	0
15	2008	abstract	5	58390	0.0001
16	2008	abstract-class	27	58390	0.0005

Rows: 20,000 ⚠ truncated from 40,518 rows

3. Has R been growing or shrinking?

So far we've been learning and using the R programming language. Wouldn't we like to be sure it's a good investment for the future? Has it been keeping pace with other languages, or have people been switching out of it?

Let's look at whether the fraction of Stack Overflow questions that are about R has been increasing or decreasing over time.

```
# Filter for R tags
r_over_time <- by_tag_year_fraction %>%
    filter(tag == "r")

# Print the new table
r_over_time
```

index	year	tag	number	year_total	fraction
1	2008	r	8	58390	0.0001
2	2009	r	524	343868	0.0015
3	2010	r	2270	694391	0.0033
4	2011	r	5845	1200551	0.0049
5	2012	r	12221	1645404	0.0074
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	0.02
10	2017	r	54415	2305207	0.0236
11	2018	r	28938	1085170	0.0267

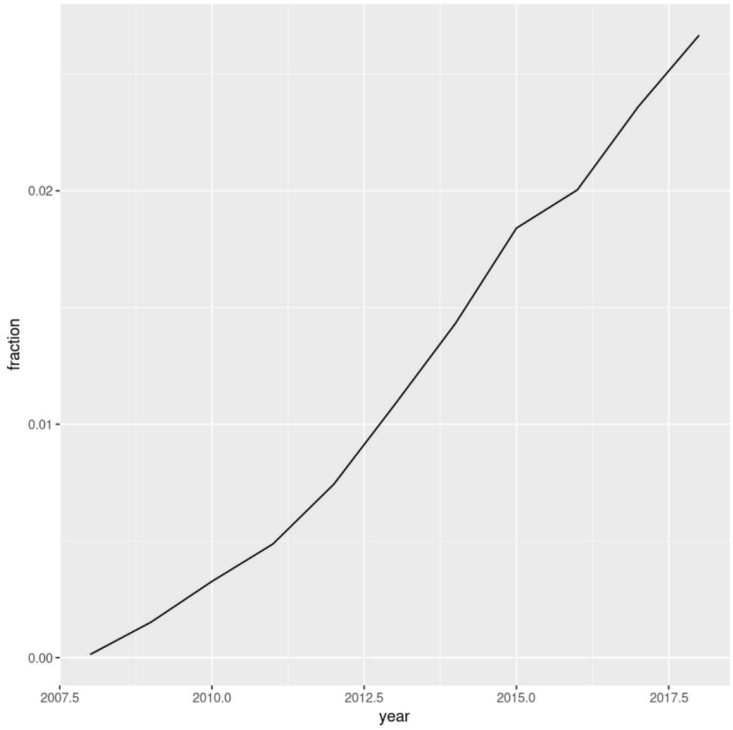
Rows: 11

4. Visualizing change over time

Rather than looking at the results in a table, we often want to create a visualization. Change over time is usually visualized with a line plot.

```
# Load ggplot2
library(ggplot2)

# Create a line plot of fraction over time
ggplot(r_over_time, aes(x=year, y=fraction)) +
    geom_line()
```



5. How about dplyr and ggplot2?

Based on that graph, it looks like R has been growing pretty fast in the last decade. Good thing we're practicing it now!

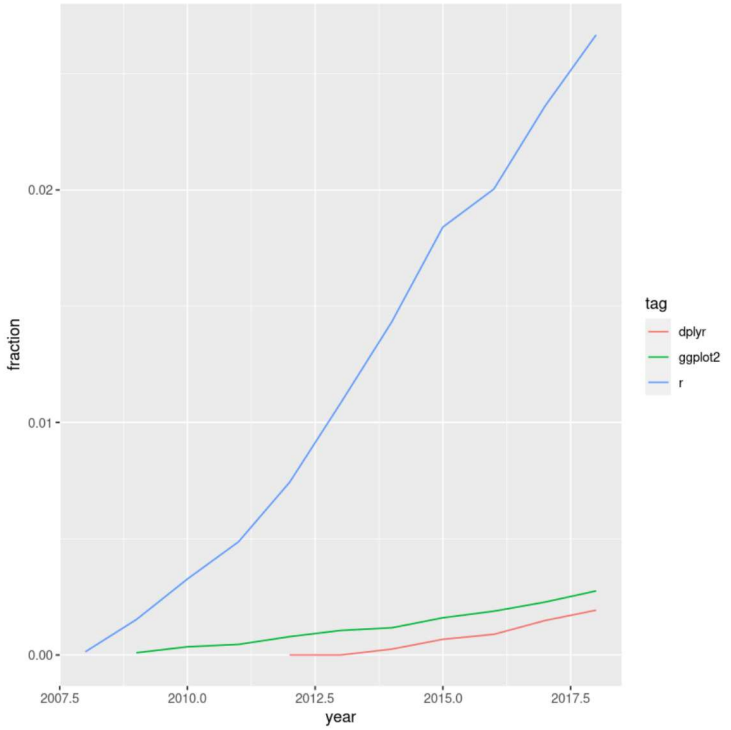
Besides R, two other interesting tags are dplyr and ggplot2, which we've already used in this analysis. They both also have Stack Overflow tags!

Instead of just looking at R, let's look at all three tags and their change over time. Are each of those tags increasing as a fraction of overall questions? Are any of them decreasing?

```
# A vector of selected tags
selected_tags <- c("r", "dplyr", "ggplot2")

# Filter for those tags
selected_tags_over_time <- by_tag_year_fraction %>%
    filter(tag %in% selected_tags)

# Plot tags over time on a line plot using color to represent tag
ggplot(selected_tags_over_time, aes(x=year, y=fraction, color = tag)) +
    geom_line()
```



6. What are the most asked-about tags?

It's sure been fun to visualize and compare tags over time. The dplyr and ggplot2 tags may not have as many questions as R, but we can tell they're both growing quickly as well.

We might like to know which tags have the most questions *overall*, not just within a particular year. Right now, we have several rows for every tag, but we'll be combining them into one. That means we want `group_by()` and `summarize()`.

Let's look at tags that have the most questions in history.

```
# Find total number of questions for each tag
sorted_tags <- by_tag_year %>%
    group_by(tag) %>%
    summarise(tag_total = sum(number)) %>%
    arrange(desc(tag_total))

# Print the new table
sorted_tags
```

	tag	t..
1	javascript	1632049
2	java	1425961

7. How have large programming languages changed over time?

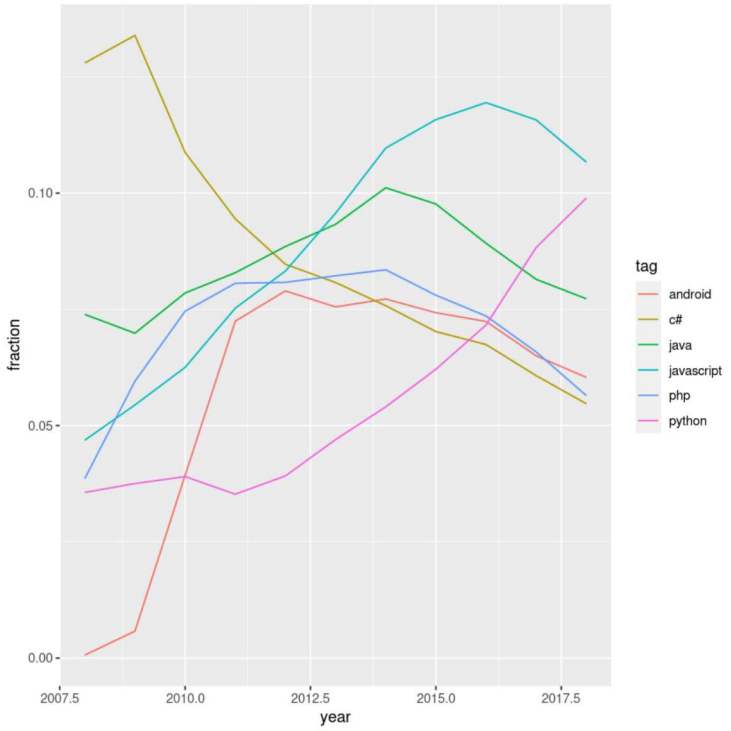
We've looked at selected tags like R, ggplot2, and dplyr, and seen that they're each growing. What tags might be *shrinking*? A good place to start is to plot the tags that we just saw that were the most-asked about of all time, including JavaScript, Java and C#.

7	jquery	915159

```
# Get the six largest tags
highest_tags <- head(sorted_tags$tag)

# Filter for the six largest tags
by_tag_subset <- by_tag_year_fraction %>%
    filter(tag %in% highest_tags)

# Plot tags over time on a line plot using color to represent tag
ggplot(by_tag_subset, aes(x = year, y = fraction, color = tag)) +
    geom_line()
```



8. Some more tags!

Wow, based on that graph we've seen a lot of changes in what programming languages are most asked about. C# gets fewer questions than it used to, and Python has grown quite impressively.

This Stack Overflow data is incredibly versatile. We can analyze *any* programming language, web framework, or tool where we'd like to see their change over time. Combined with the reproducibility of R and its libraries, we have ourselves a powerful method of uncovering insights about technology.

To demonstrate its versatility, let's check out how three big mobile operating systems (Android, iOS, and Windows Phone) have compared in popularity over time. But remember: this code can be modified simply by changing the tag names!

```
# Get tags of interest
my_tags <- c("android", "ios", "windows-phone")

# Filter for those tags
```