Aim: To analyze the relative popularity of programming languages over time based on Stack Overflow data.

Key questions to answer:

1. Which is the most popular programming language?
2. Which programming languages are growing and which ones are shrinking?
3. How has the popularity of R changed over time?
4. How has the popularity of R, ggplot2 and dplyr changed over time?
5. Exploring curiosity: How has the popularity of android, ios and windows-phones changed over time?

Key tasks:

1. Loading libraries like readr, dplyr, ggplot2.
2. Creating tables.
3. Creating new column with calculation of number of questions per tag out of total questions.
4. Using filter, piping, group_by, summarize, arrange.
5. Data visualization using line plot.

Insights:

1. Javascript is the most popular programming language.
2. C# has shrunk while Python has grown immensely in popularity.
3. R has been steadily growing over the years.
4. R is very popular in comparison to ggplot2 and dplyr.
5. The most popular has been android, then ios and windows-phone.

Following are snippets of code and the results:

1. Data on tags over time.

In [106]:
```
# Load libraries
library(readr)
library(dplyr)
library(datasets)

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

# Inspect the dataset
print(by_tag_year)
```

```
Parsed with column specification:
cols(
  year = col_double(),
  tag = col_character(),
  number = col_double(),
  year_total = col_double()
)
```

```
# 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
```

2. Adding a new column, “fraction”.

In [108]:
```
# Add fraction column
by_tag_year_fraction <- by_tag_year %>% mutate(fraction = number / year_total)

# Print the new table
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
```

click to expand output; double click to hide output

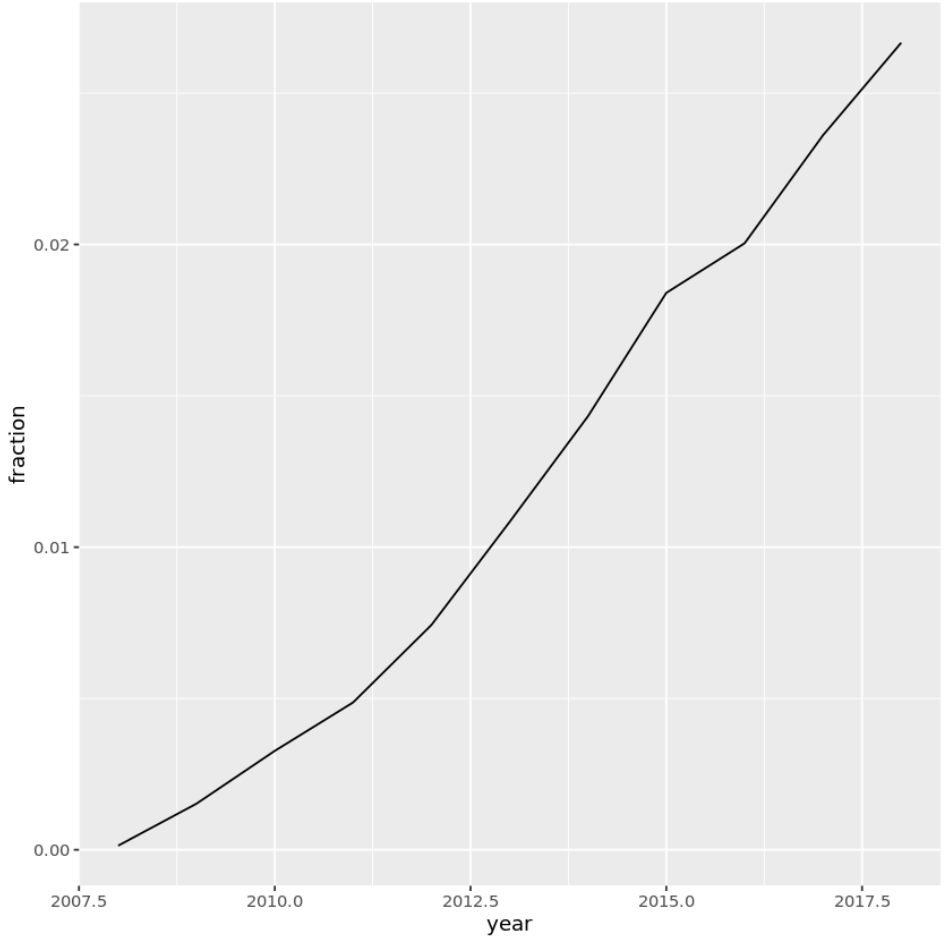
3. Has R been growing or shrinking?

In [110]:
```
# Filter for R tags
r_over_time <- filter(by_tag_year_fraction, tag == "r")

# Print the new table
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
```

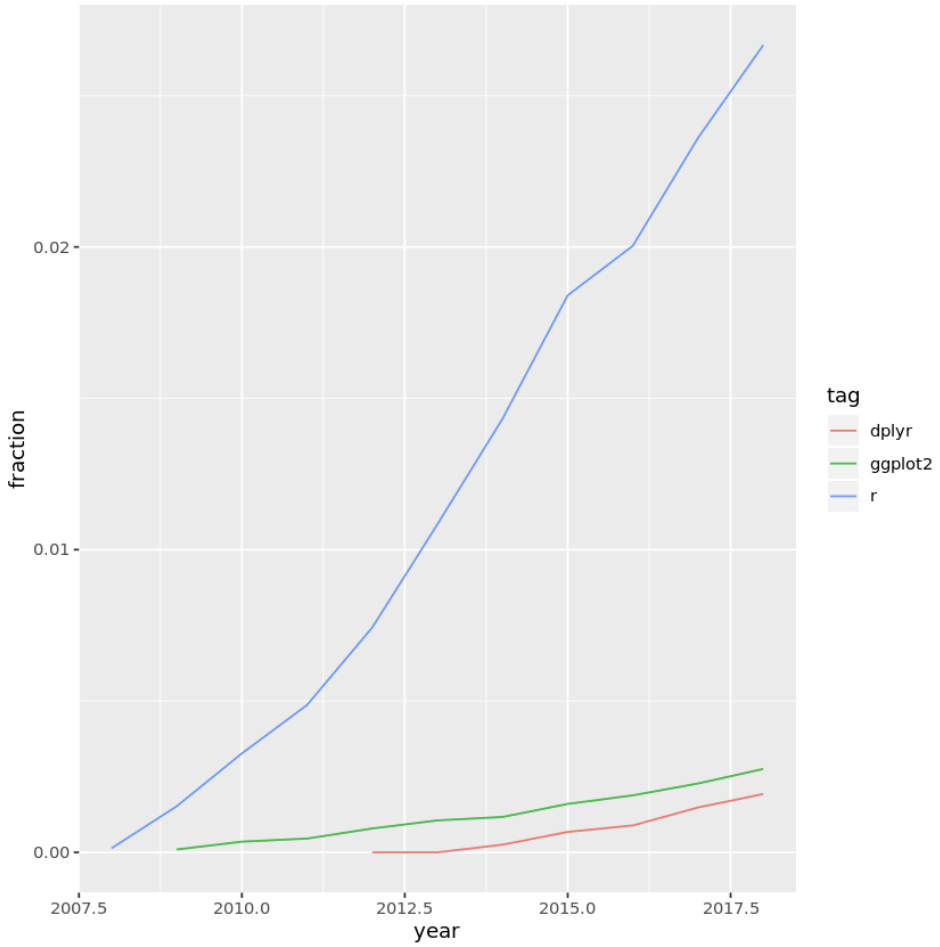
4. Visualizing change over time.

In [112]:
```
# Load ggplot2
library(ggplot2)

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



5. Popularity of R, ggplot2 and dplyr.

In [114]:
```
# A vector of selected tags
selected_tags <- c("r", "dplyr", "ggplot2")

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

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



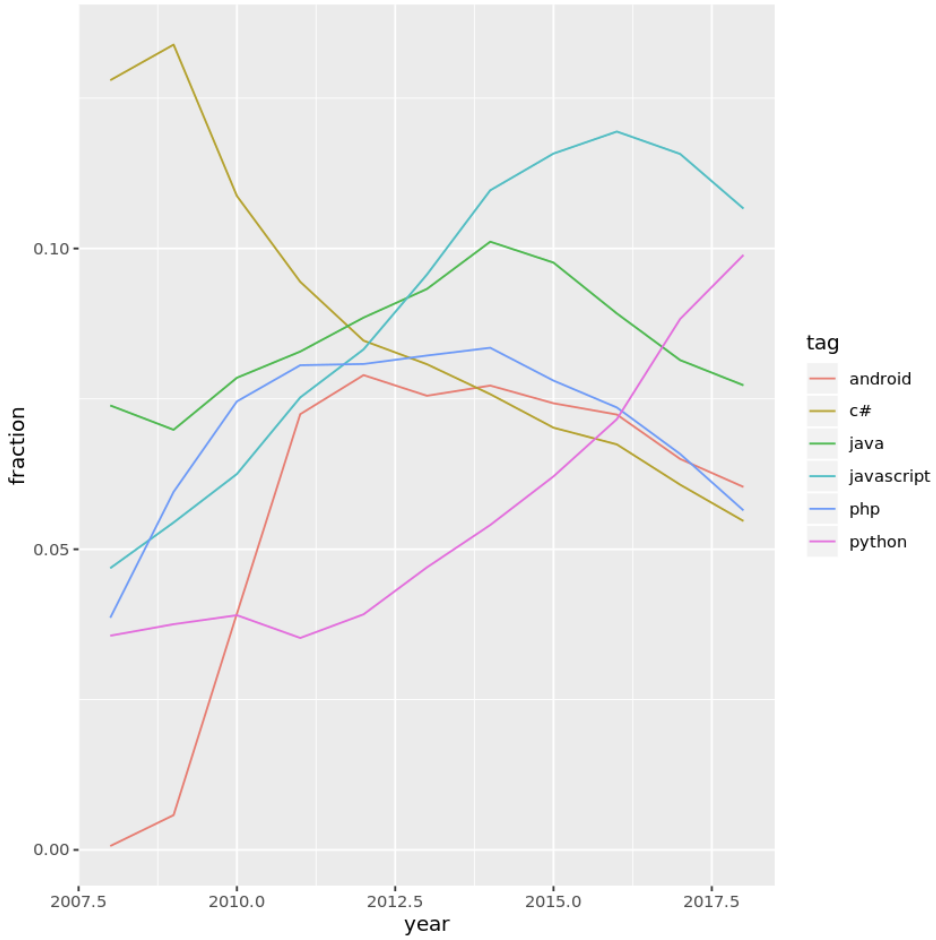
6. What are the most asked-about tags?

In [116]:
```
# Find total number of questions for each tag
sorted_tags <- by_tag_year %>% group_by(tag) %>% summarize(tag_total = sum(number)) %>% arrange(desc(tag_total))

# Print the new table
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
```

7. Popularity of programming languages over time.

In [118]:
```
# 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(data = by_tag_subset, aes(x = year, y = fraction, color = tag)) + geom_line()
```



8. Popularity of android, ios and windows-phone.

In [120]:
```
# Get tags of interest
my_tags <- c("android", "ios", "windows-phone")

# Filter for those tags
by_tag_subset <- by_tag_year_fraction %>% filter(tag %in% my_tags)

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