1. Welcome to the world of data science

Throughout the world of data science, there are many languages and tools that can be used to complete a given task. While you are often able to use whichever tool you prefer, it is often important for analysts to work with similar platforms so that they can share their code with one another. Learning what professionals in the data science industry use while at work can help you gain a better understanding of things that you may be asked to do in the future.

In this project, we are going to find out what tools and languages professionals use in their day-to-day work. Our data comes from the Kaggle Data Science Survey which includes responses from over 10,000 people that write code to analyze data in their daily work.

In [220]:

```
# Load necessary packages
library(tidyverse)

# Load the data
responses <- read_csv('datasets/kagglesurvey.csv')

# Print the first 10 rows
head(responses,10)
```

```
Parsed with column specification:
cols(
  Respondent = col_integer(),
  WorkToolsSelect = col_character(),
  LanguageRecommendationSelect = col_character(),
  EmployerIndustry = col_character(),
  WorkAlgorithmsSelect = col_character()
)
```

Respondent	WorkToolsSelect	LanguageRecommendatio
1	Amazon Web services,Oracle Data Mining/ Oracle R Enterprise,Perl	F#
2	Amazon Machine Learning,Amazon Web services,Cloudera,Hadoop/Hive/Pig,Impala,Java,Mathematica,MATLAB/Octave,Microsoft Excel Data Mining,Microsoft SQL Server Data Mining,NoSQL,Python,R,SAS Base,SAS JMP,SQL,Tableau	Python
3	C/C++,Jupyter notebooks,MATLAB/Octave,Python,R,TensorFlow	Python
4	Jupyter notebooks,Python,SQL,TensorFlow	Python
5	C/C++,Cloudera,Hadoop/Hive/Pig,Java,NoSQL,R,Unix shell / awk	R
6	SQL	Python
7	Jupyter notebooks,NoSQL,Python,R,SQL,Unix shell / awk	Python
8	Python,Spark / MLlib,Tableau,TensorFlow,Other	Python

Respondent	WorkToolsSelect	LanguageRecommendationSelect
	...,MATLAB/Octave,Python,SAS Base,SQL	Python
10	C/C++,IBM Cognos,MATLAB/Octave,Microsoft Excel Data Mining,Microsoft R Server (Formerly Revolution Analytics),Microsoft SQL Server Data Mining,Perl,Python,R,SQL,Unix shell / awk	R

In [221]:

```
library("testthat")
library('IRkernel.testthat')

run_tests({
    test_that("Read in data correctly.", {
        expect_is(responses, "tbl_df",
            info = 'You should use read_csv (with an underscore) to read
"datasets/kagglesurvey.csv" into responses')
    })

    test_that("Read in data correctly.", {
        responses_test <- read_csv('datasets/kagglesurvey.csv')
        expect_equivalent(responses, responses_test,
            info = 'responses should contain the data in "datasets/kagglesurvey.csv"')
    })

})
```

```
<ProjectReporter>
  Inherits from: <ListReporter>
  Public:
    .context: NULL
    .end_context: function (context)
    .start_context: function (context)
    add_result: function (context, test, result)
    all_tests: environment
    cat_line: function (...)
    cat_tight: function (...)
    clone: function (deep = FALSE)
    current_expectations: environment
    current_file: some name
    current_start_time: 20.352 0.34 5659.683 0.004 0.001
    dump_test: function (test)
    end_context: function (context)
    end_reporter: function ()
    end_test: function (context, test)
    get_results: function ()
    initialize: function (...)
    is_full: function ()
    out: 3
    results: environment
    rule: function (...)
    start_context: function (context)
    start_file: function (name)
    start_reporter: function ()
    start_test: function (context, test)
```

2. Using multiple tools

Now that we've loaded in the survey results, we want to focus on the tools and languages that the survey respondents use at work.

In [222]:

```
# Print the first respondent's tools and languages
responses %>%
  filter(Respondent == 1) %>%
    select(WorkToolsSelect, LanguageRecommendationSelect)
# Create a new data frame called tools
tools <- responses

# Add a new column, and unnest the new column
tools <- tools  %>%
    mutate(work_tools = strsplit(WorkToolsSelect, ","))  %>%
    unnest(work_tools)
```

```
    unnest(work_tools)

# View the first 6 rows of tools
head(tools)
```

WorkToolsSelect	LanguageRecommendationSelect
Amazon Web services,Oracle Data Mining/ Oracle R Enterprise,Perl	F#

Respondent	WorkToolsSelect	LanguageRecommendatio
1	Amazon Web services,Oracle Data Mining/ Oracle R Enterprise,Perl	F#
1	Amazon Web services,Oracle Data Mining/ Oracle R Enterprise,Perl	F#
1	Amazon Web services,Oracle Data Mining/ Oracle R Enterprise,Perl	F#
2	Amazon Machine Learning,Amazon Web services,Cloudera,Hadoop/Hive/Pig,Impala,Java,Mathematica,MATLAB/Octave,Microsoft Excel Data Mining,Microsoft SQL Server Data Mining,NoSQL,Python,R,SAS Base,SAS JMP,SQL,Tableau	Python
2	Amazon Machine Learning,Amazon Web services,Cloudera,Hadoop/Hive/Pig,Impala,Java,Mathematica,MATLAB/Octave,Microsoft Excel Data Mining,Microsoft SQL Server Data Mining,NoSQL,Python,R,SAS Base,SAS JMP,SQL,Tableau	Python
2	Amazon Machine Learning,Amazon Web services,Cloudera,Hadoop/Hive/Pig,Impala,Java,Mathematica,MATLAB/Octave,Microsoft Excel Data Mining,Microsoft SQL Server Data Mining,NoSQL,Python,R,SAS Base,SAS JMP,SQL,Tableau	Python

In [223]:

```
run_tests({
    test_that("Tools and Languages were Split and Unnested", {
        expect_true(nrow(tools) == 47409,
            info = 'Make sure that you split the tools at the commas and unnested them')
    })

    test_that("Tools and Languages were Unnested", {
        expect_is(tools$work_tools, "character",
            info = 'The work_tools column should be of class "character". Make sure that you unnest
ed the results of strsplit()')
    })

})
```

```
<ProjectReporter>
  Inherits from: <ListReporter>
  Public:
    .context: NULL
    .end_context: function (context)
    .start_context: function (context)
    add_result: function (context, test, result)
    all_tests: environment
    cat_line: function (...)
    cat_tight: function (...)
    clone: function (deep = FALSE)
    current_expectations: environment
    current_file: some name
    current_start_time: 20.499 0.34 5659.829 0.004 0.001
    dump_test: function (test)
```

```
    end_context: function (context)
    end_reporter: function ()
    end_test: function (context, test)
    get_results: function ()
    initialize: function (...)
    is_full: function ()
    out: 3
    results: environment
    rule: function (...)
    start_context: function (context)
    start_file: function (name)
    start_reporter: function ()
    start_test: function (context, test)
```

3. Counting users of each tool

Now that we've split apart all of the tools used by each respondent, we can figure out which tools are the most popular.

In [224]:

```
# Create a new data frame
tool_count <- tools

# Group the data by work_tools, summarise the counts, and arrange in descending order
tool_count <- tool_count  %>%
    group_by(work_tools)  %>%
     summarise(count = n()) %>%  #or count('work_tools')
       arrange(desc(count))

#Print the first 6 results of tool_count
head(tool_count)
```

work_tools	count
Python	6073
R	4708
SQL	4261
Jupyter notebooks	3206
TensorFlow	2256
NA	2198

In [225]:

```
run_tests({
    test_that("Tools were Grouped and Summarised", {
        expect_true(nrow(tool_count) == 50,
            info = 'Make sure that you grouped by tools and then summarised')
    })

    test_that("Values were sorted correctly", {
        expect_true(tool_count[1, 2] == 6073,
            info = 'Do not forget to sort your tool counts from largest to smallest')
    })

})
```

```
<ProjectReporter>
  Inherits from: <ListReporter>
  Public:
    .context: NULL
    .end_context: function (context)
    .start_context: function (context)
    add_result: function (context, test, result)
    all_tests: environment
    cat_line: function (...)
    cat_tight: function (...)
    clone: function (deep = FALSE)
    current_expectations: environment
    current_file: some name
```

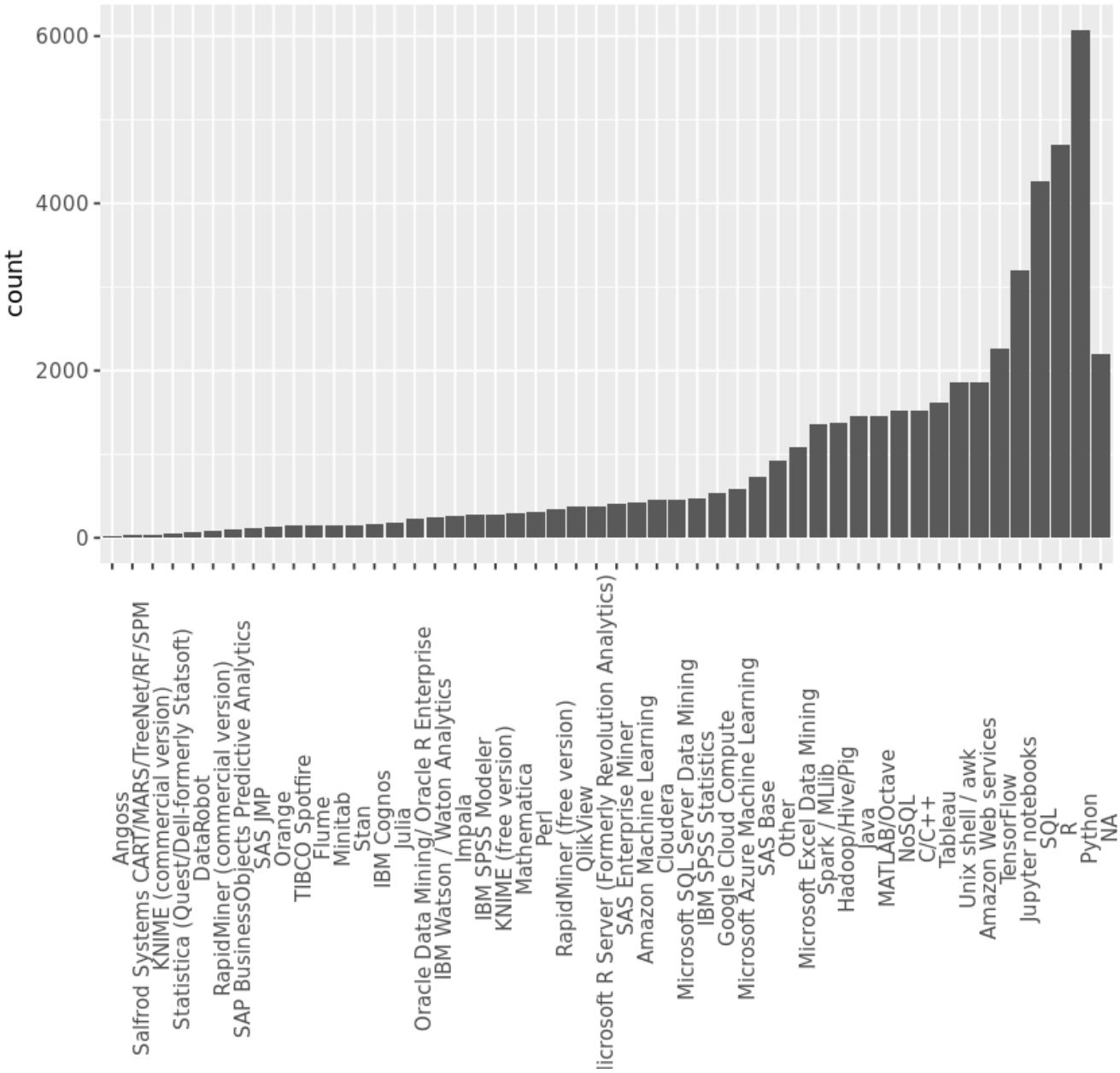
```
current_file: some name
current_start_time: 20.544 0.34 5659.873 0.004 0.001
dump_test: function (test)
end_context: function (context)
end_reporter: function ()
end_test: function (context, test)
get_results: function ()
initialize: function (...)
is_full: function ()
out: 3
results: environment
rule: function (...)
start_context: function (context)
start_file: function (name)
start_reporter: function ()
start_test: function (context, test)
```

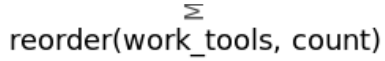
4. Plotting the most popular tools

Let's see how your favorite tools stack up against the rest.

In [226]:

```
# Create a bar chart of the work_tools column, most counts on the far right
ggplot(tool_count, aes(x = reorder(work_tools, count), y = count)) +
    geom_bar(stat = "identity") +
    theme(axis.text.x = element_text(angle = 90))
```





In [227]:

```
run_tests({
   test_that("Plot is a bar chart",{
      p <- last_plot()
      q <- p$layers[[1]]
      expect_is(q$geom, "GeomBar",
                info = "You should plot a bar chart with ggplot")
    })
})
```

```
<ProjectReporter>
  Inherits from: <ListReporter>
  Public:
    .context: NULL
    .end_context: function (context)
    .start_context: function (context)
    add_result: function (context, test, result)
    all_tests: environment
    cat_line: function (...)
    cat_tight: function (...)
    clone: function (deep = FALSE)
    current_expectations: environment
    current_file: some name
    current_start_time: 20.81 0.344 5660.142 0.004 0.001
    dump_test: function (test)
    end_context: function (context)
    end_reporter: function ()
    end_test: function (context, test)
    get_results: function ()
    initialize: function (...)
    is_full: function ()
    out: 3
    results: environment
    rule: function (...)
    start_context: function (context)
    start_file: function (name)
    start_reporter: function ()
    start_test: function (context, test)
```

5. The R vs Python debate

Within the field of data science, there is a lot of debate among professionals about whether R or Python should reign supreme. You can see from our last figure that R and Python are the two most commonly used languages, but it's possible that many respondents use both R and Python. Let's take a look at how many people use R, Python, and both tools.

In [228]:

```
# Create a new data frame called debate_tools
debate_tools <- responses

# Creat a new column called language preference
debate_tools <- debate_tools  %>%
   mutate(language_preference = case_when(
              grepl("R", WorkToolsSelect) & ! grepl("Python", WorkToolsSelect) ~ "R",
              grepl("Python", WorkToolsSelect) & ! grepl("R", WorkToolsSelect) ~ "Python",
              grepl("R", WorkToolsSelect) &  grepl("Python", WorkToolsSelect) ~ "both",
              ! grepl("R", WorkToolsSelect) & ! grepl("Python", WorkToolsSelect) ~ "neither"
   ))

# Print the first 6 rows
head(debate_tools,300)
```

Respondent	WorkToolsSelect	LanguageRecommendatio
1	Amazon Web services,Oracle Data Mining/ Oracle R Enterprise,Perl	F#
	Amazon Machine Learning,Amazon Web	

Respondent	WorkToolsSelect	LanguageRecommendation
2	services,Cloudera,Hadoop/Hive/Pig,Impala,Java,Mathematica,MATLAB/Octave,Microsoft Excel Data Mining,Microsoft SQL Server Data Mining,NoSQL,Python,R,SAS Base,SAS JMP,SQL,Tableau	Python
3	C/C++,Jupyter notebooks,MATLAB/Octave,Python,R,TensorFlow	Python
4	Jupyter notebooks,Python,SQL,TensorFlow	Python
5	C/C++,Cloudera,Hadoop/Hive/Pig,Java,NoSQL,R,Unix shell / awk	R
6	SQL	Python
7	Jupyter notebooks,NoSQL,Python,R,SQL,Unix shell / awk	Python
8	Python,Spark / MLlib,Tableau,TensorFlow,Other	Python
9	Jupyter notebooks,MATLAB/Octave,Python,SAS Base,SQL	Python
10	C/C++,IBM Cognos,MATLAB/Octave,Microsoft Excel Data Mining,Microsoft R Server (Formerly Revolution Analytics),Microsoft SQL Server Data Mining,Perl,Python,R,SQL,Unix shell / awk	R
11	C/C++,Jupyter notebooks,Python,TensorFlow	Python
12	MATLAB/Octave,Python	Matlab
13	C/C++,Java,Jupyter notebooks,MATLAB/Octave,Python,R	Python
14	Amazon Web services,Java,Jupyter notebooks,MATLAB/Octave,Microsoft Excel Data Mining,SAS Enterprise Miner,Spark / MLlib	Matlab
15	Hadoop/Hive/Pig,Java,Jupyter notebooks,NoSQL,Python,R,Spark / MLlib,SQL	Python
16	IBM Cognos,NoSQL,Unix shell / awk	R
17	Cloudera,Flume,Hadoop/Hive/Pig,Impala,Java,Jupyter notebooks,KNIME (free version),NoSQL,Python,R,SAS Enterprise Miner,Spark / MLlib,SQL	Python
18	C/C++,Cloudera,Hadoop/Hive/Pig,Impala,Jupyter notebooks,NoSQL,Python,R,Spark / MLlib,Unix shell / awk,Other	Python
19	Python,R,SQL	Python
20	NoSQL,Perl,Python,Other	Python

Respondent	WorkToolsSelected	LanguageRecommendation
21	MATLAB/Octave,Python,R,Unix shell / awk,Other	Python
22	C/C++,Jupyter notebooks,Python,TensorFlow,Unix shell / awk	Python
23	MATLAB/Octave,Python,R,SAS JMP,SQL,TIBCO Spotfire	Python
24	C/C++,Jupyter notebooks,MATLAB/Octave,Python,TensorFlow	Python
25	NA	Python
26	Python	NA
27	Amazon Web services,Hadoop/Hive/Pig,Julia,Jupyter notebooks,R,SAS Base,SAS Enterprise Miner,SQL,Unix shell / awk	Python
28	Amazon Web services,Python,R,SAP BusinessObjects Predictive Analytics,Unix shell / awk	Python
29	Jupyter notebooks,Python,R,SQL	R
30	Julia,Jupyter notebooks,Python,R,SQL,TensorFlow	Python
...	...	...
271	Amazon Web services,R	Python
272	Microsoft Excel Data Mining,SQL,Other	SQL
273	C/C++,Hadoop/Hive/Pig,Jupyter notebooks,Mathematica,MATLAB/Octave,Python,R,Spark / MLlib,TensorFlow	Python
274	C/C++,Jupyter notebooks,NoSQL,Python,R,SQL,TensorFlow,Unix shell / awk,Other	Python
275	NA	C/C++/C#
276	Microsoft R Server (Formerly Revolution Analytics),R,SAP BusinessObjects Predictive Analytics,Other	Python
277	Hadoop/Hive/Pig,Jupyter notebooks,Python,Unix shell / awk	Python
278	R	R
279	C/C++,Hadoop/Hive/Pig,Java,Python,SQL,TensorFlow	Matlab
280	C/C++	R
281	Jupyter notebooks,Python,R,SQL,Unix shell / awk	Python
282	Java,R,SQL,Tableau	SQL

Respondent	WorkToolsSelect	LanguageRecommendatio
283	Amazon Machine Learning,Julia,Jupyter notebooks,Python,R,RapidMiner (commercial version)	Python
284	KNIME (free version),NoSQL,Python,R,SQL	R
285	C/C++,Java,Jupyter notebooks,MATLAB/Octave,Perl,Python,TensorFlow,Other	Python
286	Python	Python
287	Amazon Web services,Google Cloud Compute,Unix shell / awk	Other
288	Amazon Machine Learning,Amazon Web services,Microsoft Azure Machine Learning,Microsoft Excel Data Mining,Python,SQL,Tableau,Unix shell / awk	SQL
289	Python,R,SQL	R
290	DataRobot,Jupyter notebooks,Microsoft R Server (Formerly Revolution Analytics),R,SAS Enterprise Miner,SQL	Python
291	R,SAP BusinessObjects Predictive Analytics,SQL	SQL
292	Amazon Web services,Google Cloud Compute,Hadoop/Hive/Pig,Jupyter notebooks,Microsoft Azure Machine Learning,NoSQL,Python,RapidMiner (free version),TensorFlow	Python
293	Amazon Machine Learning,Amazon Web services,C/C++,Java,Jupyter notebooks,Microsoft Azure Machine Learning,Python,TensorFlow	Python
294	Python,SQL,Tableau	Python
295	Orange,Python	Python
296	Amazon Web services,Google Cloud Compute,Hadoop/Hive/Pig,Jupyter notebooks,Python,TensorFlow,Unix shell / awk	Python
297	R,SQL,Tableau	R
298	MATLAB/Octave,Microsoft Azure Machine Learning,Python,R,SQL	R
299	Amazon Machine Learning,Amazon Web services,Cloudera,Flume,Google Cloud Compute,Hadoop/Hive/Pig,Impala,Python,Spark / MLlib,SQL,Unix shell / awk	Scala
300	R	R

In [229]:

```
run_tests({
    test_that("New column was created", {
        expect_is(debate_tools$language_preference, "character",
            info = 'The language_preference column should be of class "character". Make sure that y
```

```
ou filled this new column correctly')
    })

})
```

```
<ProjectReporter>
  Inherits from: <ListReporter>
  Public:
    .context: NULL
    .end_context: function (context)
    .start_context: function (context)
    add_result: function (context, test, result)
    all_tests: environment
    cat_line: function (...)
    cat_tight: function (...)
    clone: function (deep = FALSE)
    current_expectations: environment
    current_file: some name
    current_start_time: 20.931 0.348 5660.267 0.004 0.001
    dump_test: function (test)
    end_context: function (context)
    end_reporter: function ()
    end_test: function (context, test)
    get_results: function ()
    initialize: function (...)
    is_full: function ()
    out: 3
    results: environment
    rule: function (...)
    start_context: function (context)
    start_file: function (name)
    start_reporter: function ()
    start_test: function (context, test)
```

6. Plotting R vs Python users

Now we just need to take a closer look at how many respondents use R, Python, and both!

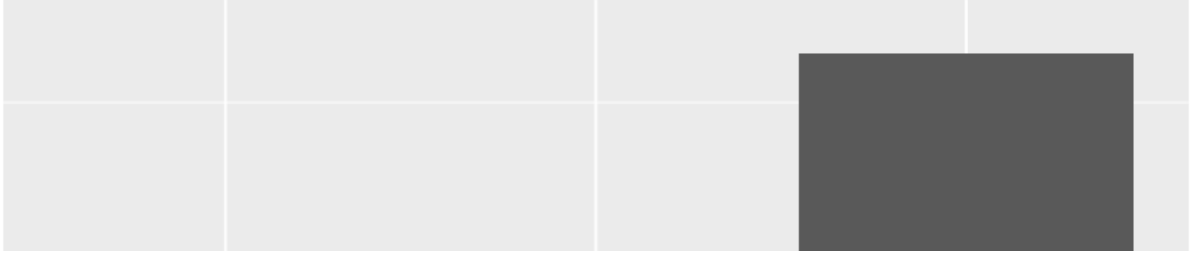
In [230]:

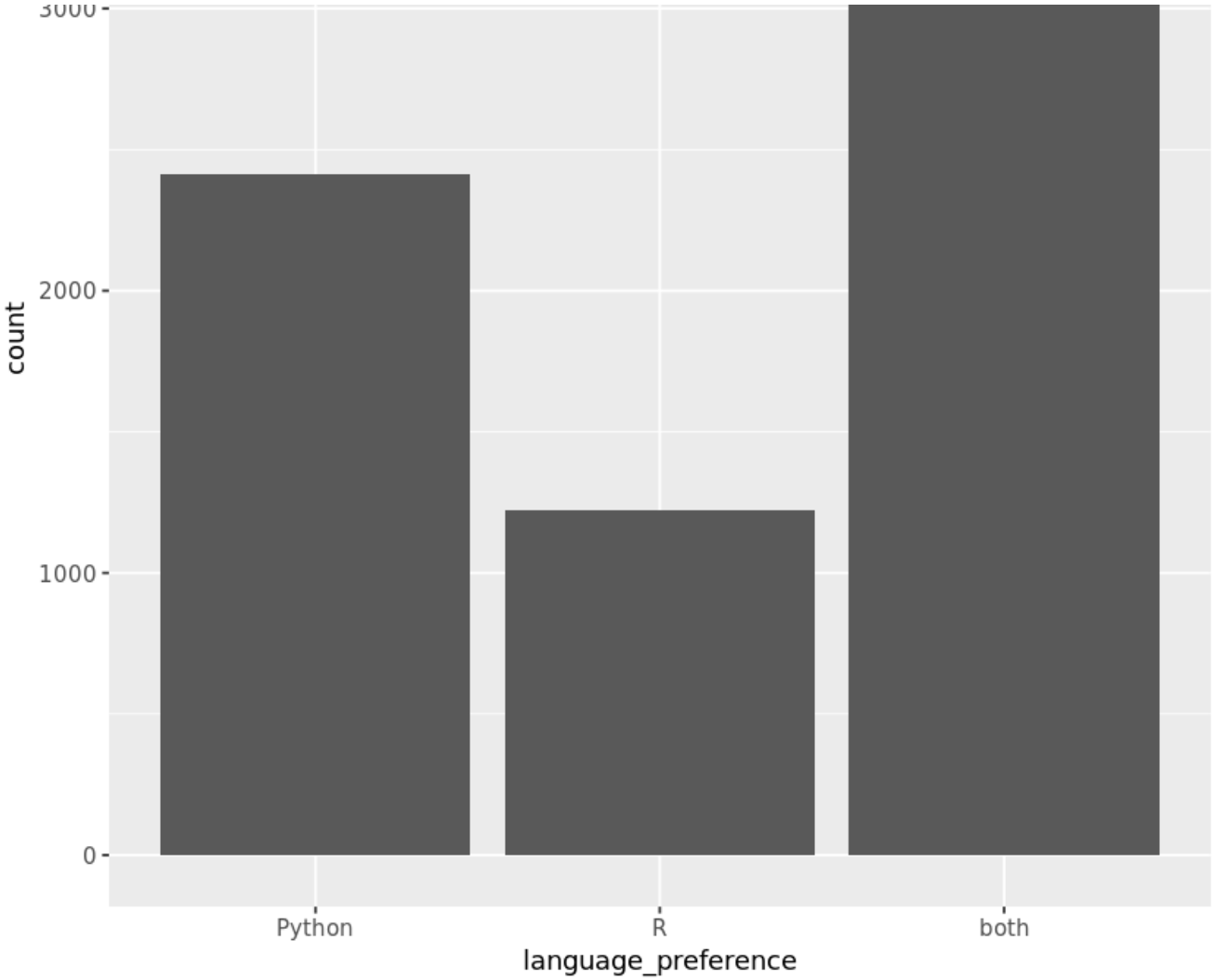
```
# Create a new data frame
debate_plot <- debate_tools

# Group by language preference, calculate number of responses, and remove "neither"
debate_plot <- debate_plot  %>%
   group_by(language_preference)  %>%
   summarise(count = n()) %>%
    filter(language_preference != "neither")
head(debate_plot)


# Create a bar chart
ggplot(debate_plot, aes(x = language_preference, y = count)) +
geom_bar(stat = 'identity')
```

language_preference	count
Python	2413
R	1220
both	3660





In [231]:

```
run_tests({
   test_that("Plot is a bar chart",{
      p <- last_plot()
      q <- p$layers[[1]]
      expect_is(q$geom, "GeomBar",
                info = "You should plot a bar chart with ggplot")
    })
})
```

```
<ProjectReporter>
  Inherits from: <ListReporter>
  Public:
    .context: NULL
    .end_context: function (context)
    .start_context: function (context)
    add_result: function (context, test, result)
    all_tests: environment
    cat_line: function (...)
    cat_tight: function (...)
    clone: function (deep = FALSE)
    current_expectations: environment
    current_file: some name
    current_start_time: 21.158 0.348 5661.808 0.004 0.001
    dump_test: function (test)
    end_context: function (context)
    end_reporter: function ()
    end_test: function (context, test)
    get_results: function ()
    initialize: function (...)
    is_full: function ()
    out: 3
    results: environment
    rule: function (...)
    start_context: function (context)
    start_file: function (name)
```

```
    start_file: function (name)
    start_reporter: function ()
    start_test: function (context, test)
```

7. Language recommendations

It looks like the largest group of professionals program in both Python and R. But what happens when they are asked which language they recommend to new learners? Do R lovers always recommend R?

In [232]:

```
# Create a new data frame
recommendations <- debate_tools

# Group by, summarise, filter, arrange, mutate, and filter
recommendations <- recommendations  %>%
    group_by(language_preference,LanguageRecommendationSelect)  %>%
    summarise(count = n())

tidy <- filter(recommendations, LanguageRecommendationSelect != 'NA' ) %>%
    arrange(language_preference, desc(count))

final <- tidy %>%
group_by(language_preference)
a = filter(final, language_preference == 'Python') %>% head(4)
b = filter(final, language_preference == 'R') %>% head(4)
c = filter(final, language_preference == 'both') %>% head(4)
d = filter(final, language_preference == 'neither') %>% head(4)

recommendations <- rbind(a,b,c,d)
recommendations
```

language_preference	LanguageRecommendationSelect	count
Python	Python	1742
Python	C/C++/C#	48
Python	Matlab	43
Python	SQL	36
R	R	632
R	Python	194
R	SQL	75
R	C/C++/C#	27
both	Python	1917
both	R	912
both	SQL	108
both	Scala	28
neither	Python	196
neither	R	94
neither	SQL	53
neither	Matlab	47

In [233]:

```
run_tests({
    test_that("Tools have been summarised", {
        expect_true(nrow(recommendations) == 16,
            info = 'Make sure that you are only keeping the top 4 responses for each language
used')
    })

})
```

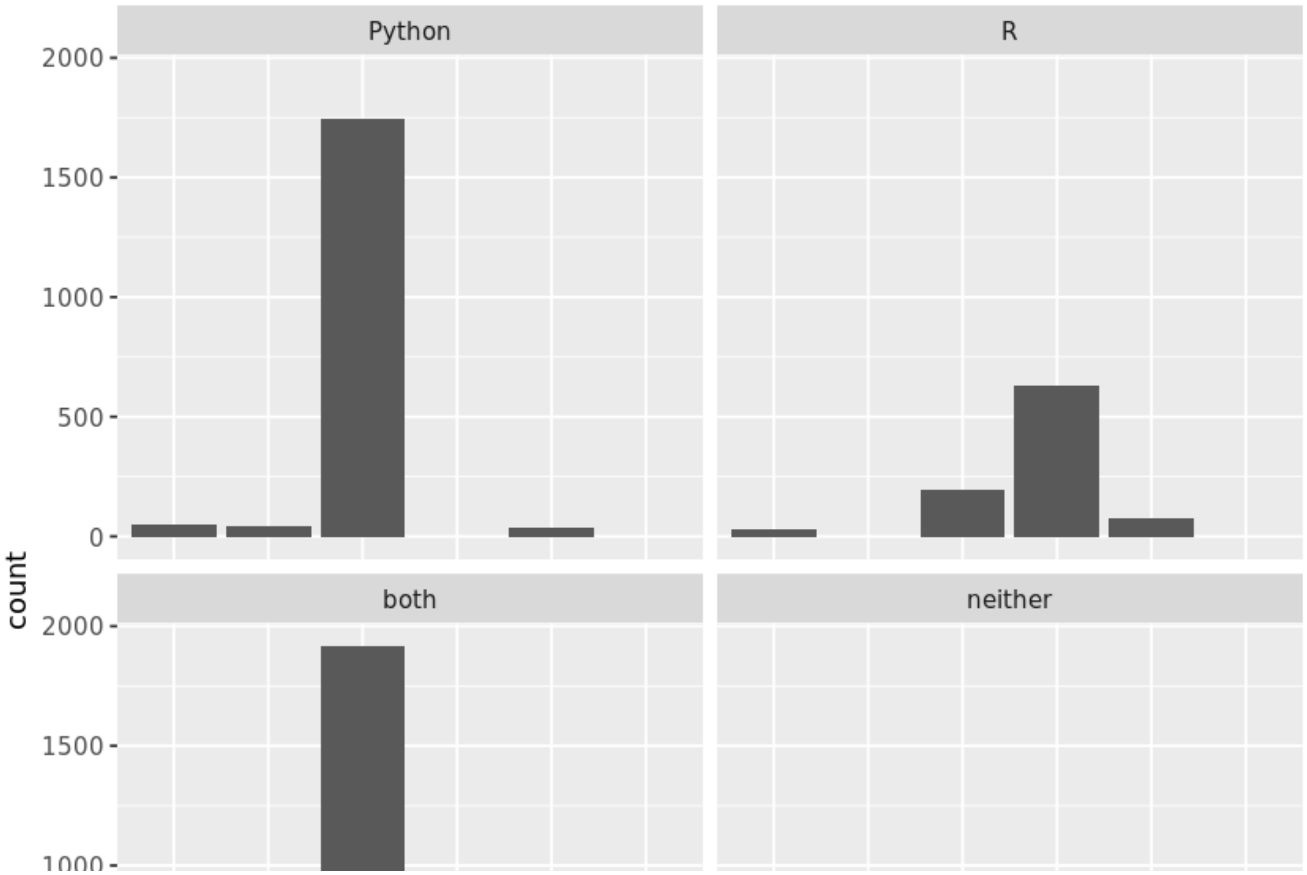
```
<ProjectReporter>
  Inherits from: <ListReporter>
  Public:
    .context: NULL
    .end_context: function (context)
    .start_context: function (context)
    add_result: function (context, test, result)
    all_tests: environment
    cat_line: function (...)
    cat_tight: function (...)
    clone: function (deep = FALSE)
    current_expectations: environment
    current_file: some name
    current_start_time: 21.318 0.352 5662.147 0.004 0.001
    dump_test: function (test)
    end_context: function (context)
    end_reporter: function ()
    end_test: function (context, test)
    get_results: function ()
    initialize: function (...)
    is_full: function ()
    out: 3
    results: environment
    rule: function (...)
    start_context: function (context)
    start_file: function (name)
    start_reporter: function ()
    start_test: function (context, test)
```

8. The most recommended language by the language used

Just one thing left. Let's graphically determine which languages are most recommended based on the language that a person uses.

In [234]:

```
# Create a faceted bar plot
ggplot(recommendations, aes(x = LanguageRecommendationSelect, y = count)) +
    geom_bar(stat = "identity") +
    facet_wrap(~language_preference) +
    theme(axis.text.x = element_text(angle = 90))
```



In [235]:

```
run_tests({
   test_that("Plot is a bar chart",{
      p <- last_plot()
      q <- p$layers[[1]]
      expect_is(q$geom, "GeomBar")
    })
})
```

```
<ProjectReporter>
  Inherits from: <ListReporter>
  Public:
    .context: NULL
    .end_context: function (context)
    .start_context: function (context)
    add_result: function (context, test, result)
    all_tests: environment
    cat_line: function (...)
    cat_tight: function (...)
    clone: function (deep = FALSE)
    current_expectations: environment
    current_file: some name
    current_start_time: 21.777 0.352 5662.605 0.004 0.001
    dump_test: function (test)
    end_context: function (context)
    end_reporter: function ()
    end_test: function (context, test)
    get_results: function ()
    initialize: function (...)
    is_full: function ()
    out: 3
    results: environment
    rule: function (...)
    start_context: function (context)
    start_file: function (name)
    start_reporter: function ()
    start_test: function (context, test)
```

9. The moral of the story

So we've made it to the end. We've found that Python is the most popular language used among Kaggle data scientists, but R users aren't far behind. And while Python users may highly recommend that new learners learn Python, would R users find the following statement `TRUE` or `FALSE`?

In [236]:

```
# Would R users find this statement TRUE or FALSE?
R_is_number_one = TRUE
```

In [237]:

```
run_tests({
    test_that("The question has been answered", {
        expect_true(R_is_number_one,
```

```
            info = 'Try again! Should R_is_number_one be set to TRUE or FALSE?')
    })

})
```

```
<ProjectReporter>
  Inherits from: <ListReporter>
  Public:
    .context: NULL
    .end_context: function (context)
    .start_context: function (context)
    add_result: function (context, test, result)
    all_tests: environment
    cat_line: function (...)
    cat_tight: function (...)
    clone: function (deep = FALSE)
    current_expectations: environment
    current_file: some name
    current_start_time: 21.807 0.352 5662.634 0.004 0.001
    dump_test: function (test)
    end_context: function (context)
    end_reporter: function ()
    end_test: function (context, test)
    get_results: function ()
    initialize: function (...)
    is_full: function ()
    out: 3
    results: environment
    rule: function (...)
    start_context: function (context)
    start_file: function (name)
    start_reporter: function ()
    start_test: function (context, test)
```