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 (https://www.kaggle.com/kaggle/kaggle-survey-2017?utm_medium=partner&utm_source=datacamp.com&utm_campaign=ml+survey+case+study) which includes responses from over 10,000 people that write code to analyze data in their daily work.

In [9]:
```r
# Load necessary packages
library(tidyverse)

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

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

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

A tibble: 10 x 5

Respondent	WorkToolsSelect	Lang
<dbl>	<chr>	
1	Amazon Web services,Oracle Data Mining/ Oracle R Enterprise,Perl	
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	
3	C/C++,Jupyter notebooks,MATLAB/Octave,Python,R,TensorFlow	
4	Jupyter notebooks,Python,SQL,TensorFlow	
5	C/C++,Cloudera,Hadoop/Hive/Pig,Java,NoSQL,R,Unix shell / awk	
6	SQL	
7	Jupyter notebooks,NoSQL,Python,R,SQL,Unix shell / awk	
8	Python,Spark / MLlib,Tableau,TensorFlow,Other	
9	Jupyter notebooks,MATLAB/Octave,Python,SAS Base,SQL	
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	

In [10]:
```
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 re
ad "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/kaggl
esurvey.csv".')
    })

})
```

```
Attaching package: 'testthat'

The following object is masked from 'package:dplyr':

    matches

The following object is masked from 'package:purrr':

    is_null

The following object is masked from 'package:tidyr':

    matches


2/2 tests passed
```

2. Using multiple tools

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

To get a better idea of how the data are formatted, we will look at the first respondent's tool-use and see that this survey-taker listed multiple tools that are each separated by a comma. To learn how many people use each tool, we need to separate out all of the tools used by each individual. There are several ways to complete this task, but we will use `str_split()` from `stringr` to separate the tools at each comma. Since that will create a list inside of the data frame, we can use the `tidyr` function `unnest()` to separate each list item into a new row.

In [11]:
```
# Print the first respondent's tools and languages
responses[1:1,1:3]

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

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

A tibble: 1 x 3

Respondent	WorkToolsSelect	LanguageRecommendationSelect
<dbl>	<chr>	<chr>
1	Amazon Web services,Oracle Data Mining/ Oracle R Enterprise,Perl	F#

A tibble: 6 x 6

Respondent	WorkToolsSelect	Lang
<dbl>	<chr>	
1	Amazon Web services,Oracle Data Mining/ Oracle R Enterprise,Perl	
1	Amazon Web services,Oracle Data Mining/ Oracle R Enterprise,Perl	
1	Amazon Web services,Oracle Data Mining/ Oracle R Enterprise,Perl	
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	
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	
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	

In [12]:
```
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 "characte
r". Make sure that you unnested the results of str_split().')
    })

})
```

```
2/2 tests passed
```

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 [13]:
```
# Group the data by work_tools, summarise the counts, and arrange in des
cending order
tool_count <- tools  %>%
    group_by(work_tools)  %>%
    summarise(count = n())  %>%
    arrange(desc(count))

# Print the first 6 results
head(tool_count,n=6)
```

```
`summarise()` ungrouping output (override with `.groups` argument)
```

A tibble: 6 x 2

work_tools	count
<chr>	<int>
Python	6073
R	4708
SQL	4261
Jupyter notebooks	3206
TensorFlow	2256
NA	2198

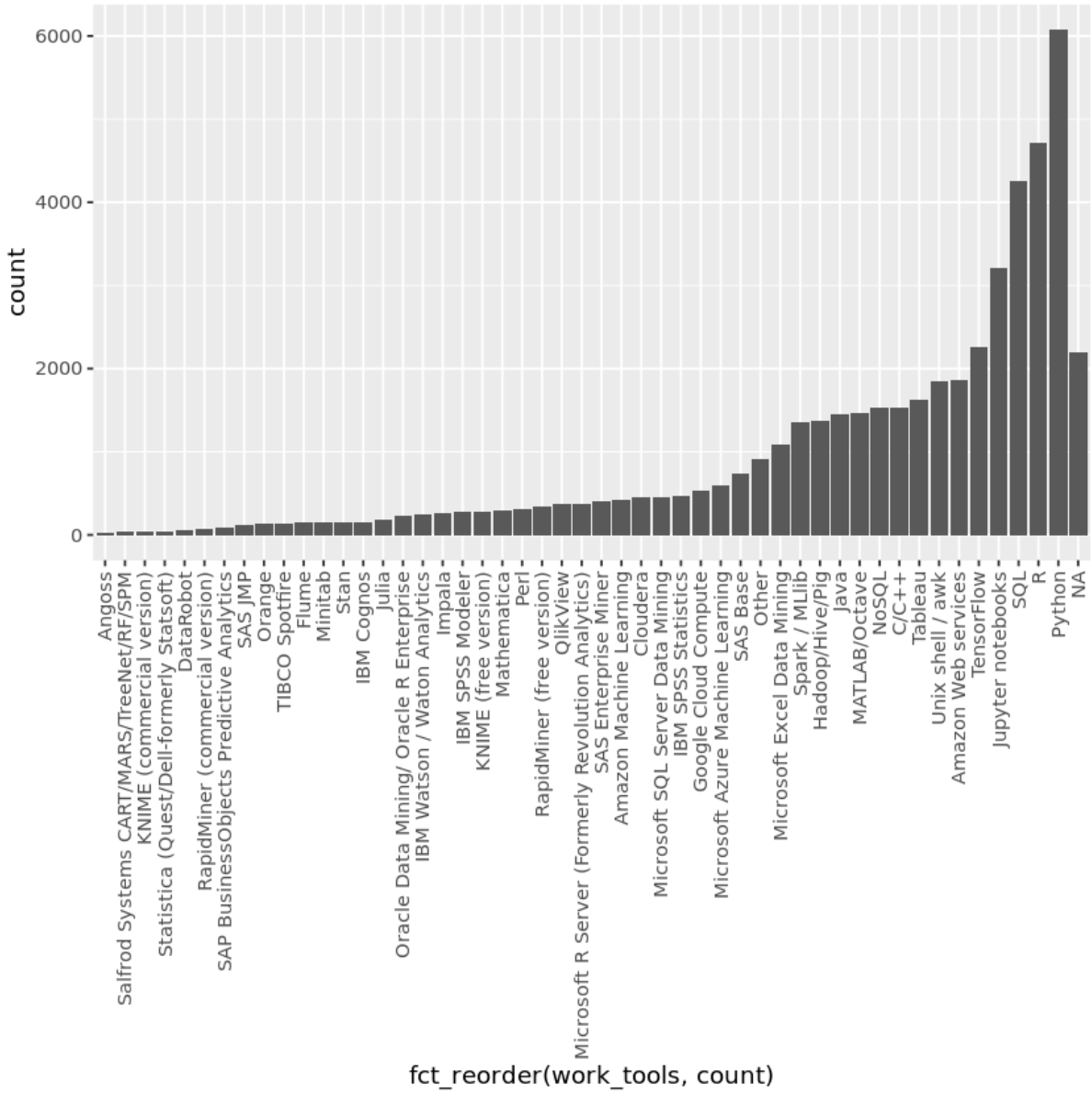
In [14]:
```
run_tests({
    test_that("Tools were Grouped and Summarised", {
        expect_true(nrow(tool_count) == 50,
            info = 'Make sure that you grouped by tools and then summari
sed the counts.')
    })

    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.')
    })

})
```

```
2/2 tests passed
```

4. Plotting the most popular tools

Let's see how the most popular tools stack up against the rest.

In [15]:
```
# Create a bar chart of the work_tools column, most counts on the far ri
ght
ggplot(tool_count, aes(x = fct_reorder(work_tools, count), y = count)) +
    geom_bar(stat = "identity") +
    theme(axis.text.x  = element_text(angle=90, vjust=0.5, hjust= 1))
```



In [16]:
```
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().")
    })
})
```

```
1/1 tests passed
```

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 [17]:
```
# Create a new column called language preference
debate_tools <- responses %>%
    mutate(language_preference = case_when(
        str_detect(WorkToolsSelect, "R") & ! str_detect(WorkToolsSelect,
"Python") ~ "R",
        str_detect(WorkToolsSelect,"Python")&! str_detect(WorkToolsSelec
t,"R")~"Python",
        str_detect(WorkToolsSelect,"R")& str_detect(WorkToolsSelect,"Pyt
hon")~"both",
        TRUE ~ "neither"
    ))

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

A tibble: 6 x 6

Respondent	WorkToolsSelect	Lang
<dbl>	<chr>	
1	Amazon Web services,Oracle Data Mining/ Oracle R Enterprise,Perl	
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	
3	C/C++,Jupyter notebooks,MATLAB/Octave,Python,R,TensorFlow	
4	Jupyter notebooks,Python,SQL,TensorFlow	
5	C/C++,Cloudera,Hadoop/Hive/Pig,Java,NoSQL,R,Unix shell / awk	
6	SQL	

In [18]:
```
debate_tools_counts <- debate_tools %>%
    count(language_preference)

run_tests({
    test_that("New column was created", {
        expect_is(debate_tools$language_preference, "character",
            info = 'The language_preference column should be of class "c
haracter". Make sure that you filled this new column correctly.')
    })
    test_that("Language preferences are correct", {
        expect_equal(filter(debate_tools_counts, language_preference ==
"both")  %>% pull(n), 3660,
            info = 'There is an incorrect amount of "both". Please check
the case_when() statements.')
        expect_equal(filter(debate_tools_counts, language_preference ==
"neither")  %>% pull(n), 2860,
            info = 'There is an incorrect amount of "neither". Please ch
eck the case_when() statements.')
        expect_equal(filter(debate_tools_counts, language_preference ==
"Python")  %>% pull(n), 2413,
            info = 'There is an incorrect amount of "Python". Please che
ck the case_when() statements.')
        expect_equal(filter(debate_tools_counts, language_preference ==
"R")  %>% pull(n), 1220,
            info = 'There is an incorrect amount of "R". Please check th
e case_when() statements.')

    })

})
```

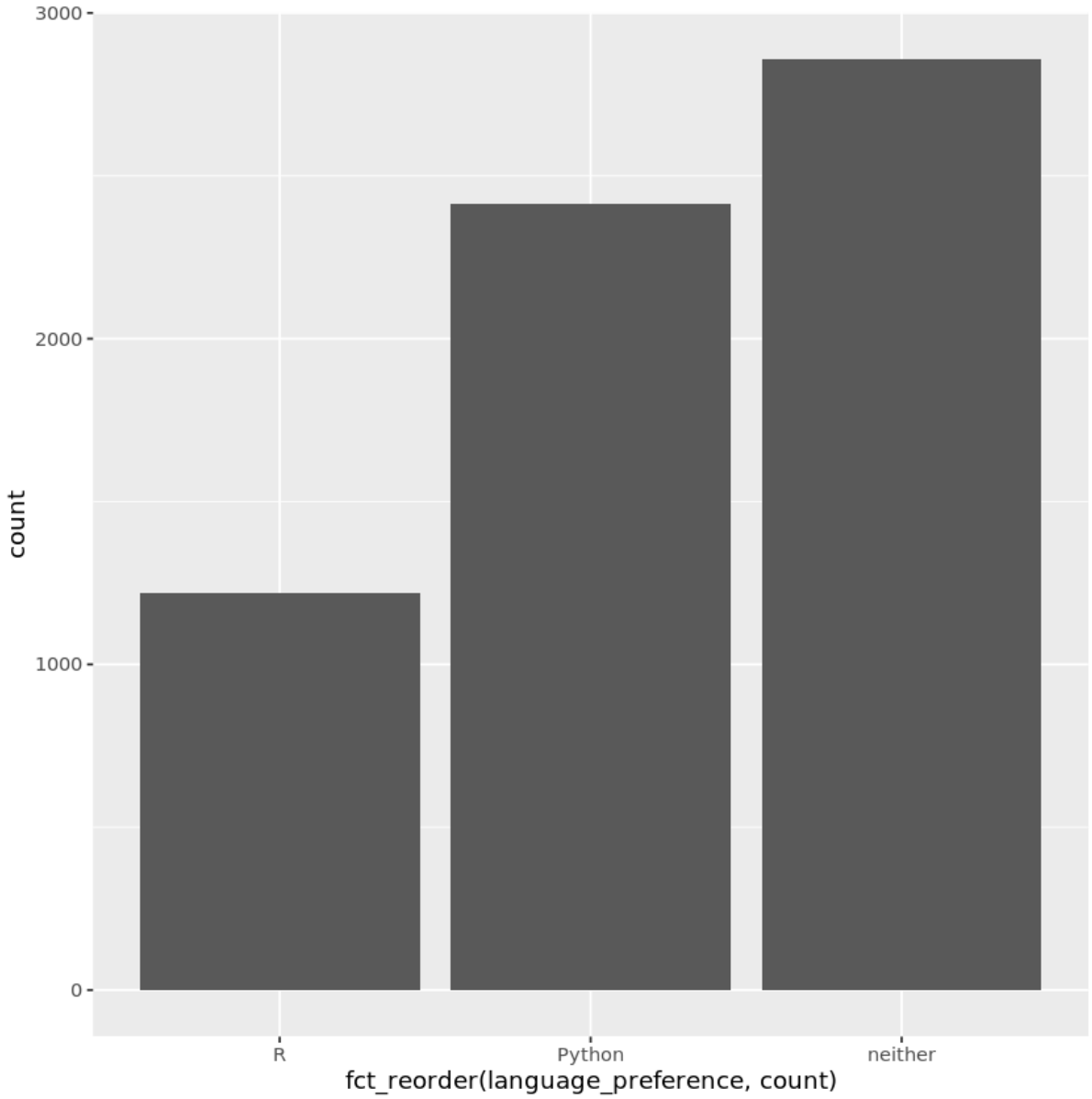
```
2/2 tests passed
```

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 [19]:
```
# Group by language preference, calculate number of responses, and remov
e "neither"
debate_plot <- debate_tools  %>%
    group_by(language_preference)  %>%
    summarise(count= n()) %>%
    filter(language_preference!= "both")

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

```
`summarise()` ungrouping output (override with `.groups` argument)
```



In [20]:
```
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().")
    })
})
```

```
1/1 tests passed
```

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 [21]:
```
# Group by, summarise, arrange, mutate, and filter
recommendations <- debate_tools  %>%
    group_by(language_preference, LanguageRecommendationSelect)  %>%
    summarise(count = n())  %>%
    arrange(language_preference, desc(count))  %>%
    mutate(row = row_number()) %>%
    filter(row <= 4)
head(recommendations)
```

```
`summarise()` regrouping output by 'language_preference' (override with
`.groups` argument)
```

A grouped_df: 6 x 4

language_preference	LanguageRecommendationSelect	count	row
<chr>	<chr>	<int>	<int>
Python	Python	1742	1
Python	NA	459	2
Python	C/C++/C#	48	3
Python	Matlab	43	4
R	R	632	1
R	NA	221	2

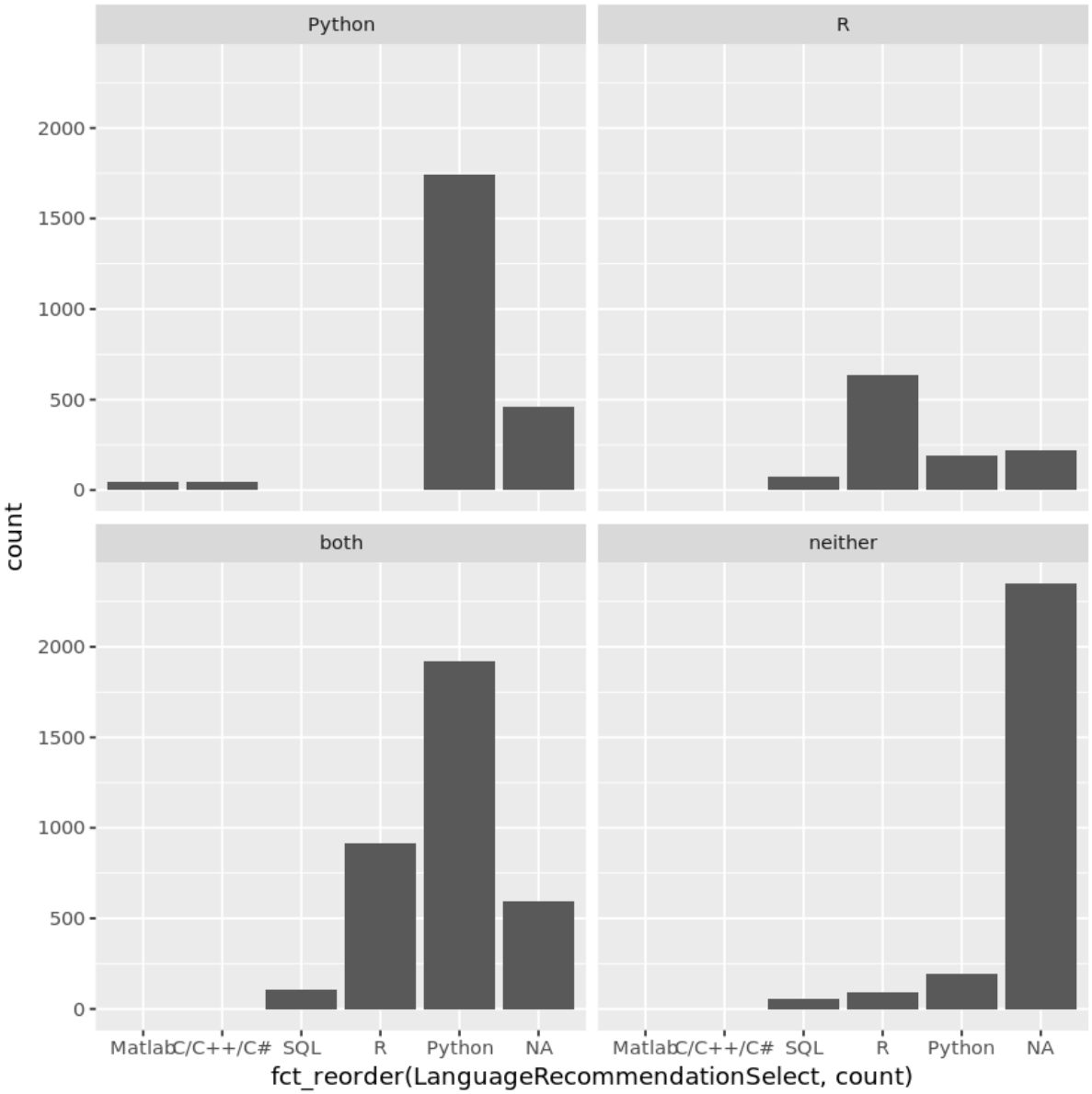
In [22]:
```
run_tests({
    test_that("Tools have been summarised", {
        expect_true(nrow(recommendations) == 16,
            info = 'Make sure that you are only keeping the top 4 respon
ses for each language used.')
    })

})
```

```
1/1 tests passed
```

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 [23]:
```
# Create a faceted bar plot
ggplot(recommendations, aes(x=fct_reorder(LanguageRecommendationSelect,c
ount),y=count)) +
geom_bar(stat="identity")+
facet_wrap(~language_preference)
```



In [24]:
```
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().")
    })
})
```

```
1/1 tests passed
```

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 [25]:
```
# Would R users find this statement TRUE or FALSE?
R_is_number_one = TRUE
```

In [26]:
```
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?')
    })

})
```

```
1/1 tests passed
```