Introduction

Throughout data science, many languages and tools can be used to complete a given task. While you can often use whichever tool you prefer, it is usually crucial for analysts to work with similar platforms to share their code with one another. Learning what professionals in the data science industry use while at work can help you better understand things you may be asked to do in the future. In this project, we will 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.

The Data set

The dataset we use is from Kaggle (https://www.kaggle.com/kaggle/kaggle-survey-2017?utm_medium=partner&utm_source=datacamp.com&utm_campaign=ml+survey+case+study from 16,716 usable respondents and 171 countries and territories. It has the following columns:

- WorkToolsSelect: The list of tools/programming languages that users use.
- LanguageRecommendationSelect: Tools/programming language that user recommend.
- EmployerIndustry: Type of industry where users work.
- WorkAlgorithmSelect: The algorithm that the user uses.

Now we load data and package and view the first few rows.

```{r}
# Load necessary packages
library(tidyverse)

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

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

Respondent	WorkToolsSelect	LanguageRecommendationSelect	EmployerIndustry	WorkAlgorithmsSelect
1	Amazon Web services,Oracle Data Mining/ Oracle R E...	F#	Internet-based	Neural Networks,Random Forests,RNNs
2	Amazon Machine Learning,Amazon Web services,Clou...	Python	Mix of fields	Bayesian Techniques,Decision Trees,Random Forests,...
3	C/C++,Jupyter notebooks,MATLAB/Octave,Python,R,...	Python	Technology	Bayesian Techniques,CNNs,Ensemble Methods,Neural ...
4	Jupyter notebooks,Python,SQL,TensorFlow	Python	Academic	Bayesian Techniques,CNNs,Decision Trees,Gradient B...
5	C/C++,Cloudera,Hadoop/Hive/Pig,Java,NoSQL,R,Unix...	R	Government	*NA*
6	SQL	Python	Non-profit	*NA*
7	Jupyter notebooks,NoSQL,Python,R,SQL,Unix shell / awk	Python	Internet-based	CNNs,Decision Trees,Gradient Boosted Machines,Ran...
8	Python,Spark / MLlib,Tableau,TensorFlow,Other	Python	Mix of fields	Bayesian Techniques,CNNs,HMMs,Neural Networks,Ra...
9	Jupyter notebooks,MATLAB/Octave,Python,SAS Base,S...	Python	Financial	Ensemble Methods,Gradient Boosted Machines
10	C/C++,IBM Cognos,MATLAB/Octave,Microsoft Excel ...	R	Technology	Bayesian Techniques,Regression/Logistic Regression
11	C/C++,Jupyter notebooks,Python,TensorFlow	Python	Academic	CNNs,Neural Networks

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 separated by a comma. To learn

how many people use each tool, we need to separate 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 the data frame, we can use the tidyr function unnest() to separate each list item into a new row.

```{r}
# Print the first respondent's tools and languages
responses[1, 2]

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

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

Respondent	WorkToolsSelect	LanguageRecommendationSelect	EmployerIndustry	WorkAlgorithmsSelect	work_tools
32	Amazon Machine Learning	*NA*	Technology	CNNs	Amazon Machine Learning
1316	Amazon Machine Learning	*NA*	Academic	*NA*	Amazon Machine Learning
2222	Amazon Machine Learning	Matlab	Academic	Markov Logic Networks	Amazon Machine Learning
3353	Amazon Machine Learning	Stata	Financial	Other	Amazon Machine Learning
4468	Amazon Machine Learning	C/C++/C#	Academic	Bayesian Techniques	Amazon Machine Learning
7791	Amazon Machine Learning	R	Financial	Bayesian Techniques	Amazon Machine Learning
8142	Amazon Machine Learning	*NA*	Financial	*NA*	Amazon Machine Learning
8426	Amazon Machine Learning	*NA*	Technology	*NA*	Amazon Machine Learning
8723	Amazon Machine Learning	C/C++/C#	Academic	Markov Logic Networks	Amazon Machine Learning
8978	Amazon Machine Learning	*NA*	Academic	*NA*	Amazon Machine Learning
4681	Amazon Machine Learning,Amazon Web services	*NA*	Technology	Decision Trees,RNNs	Amazon Machine Learning
4681	Amazon Machine Learning,Amazon Web services	*NA*	Technology	Decision Trees,RNNs	Amazon Web services

Empirical Analysis

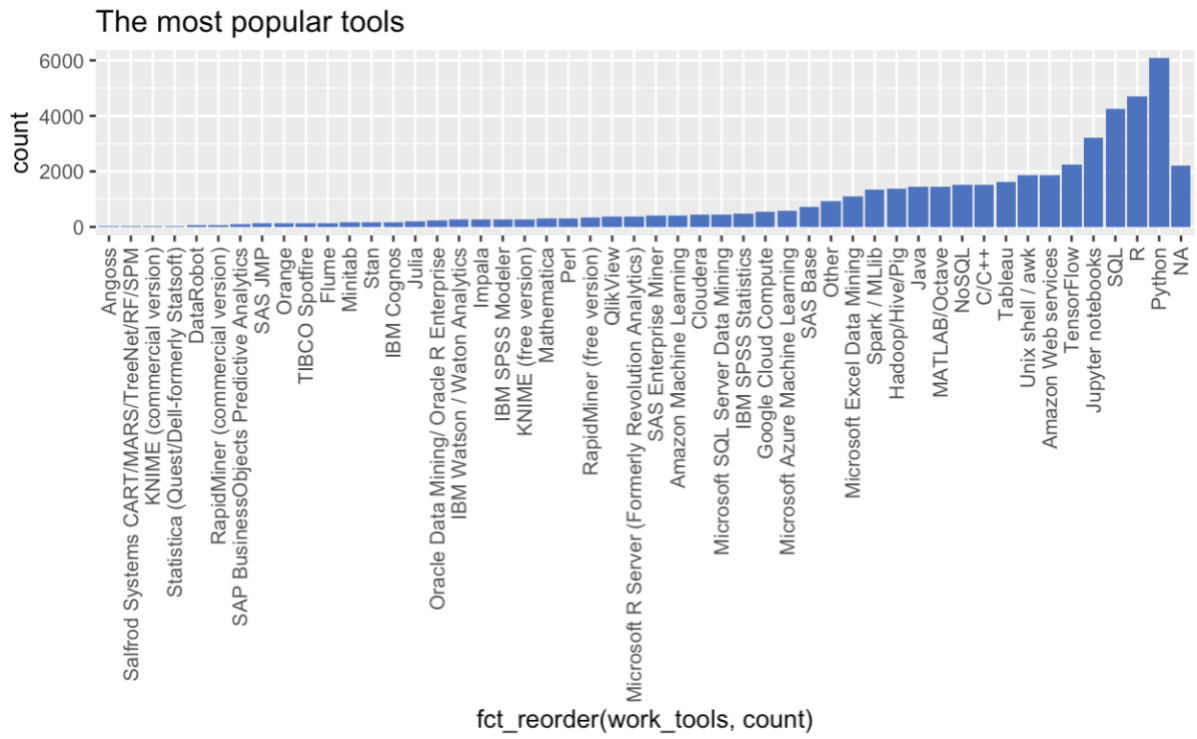
Now that we've split apart all of the tools used by each respondent, we can figure out which tools are the most popular.

```{r}
# Group the data by work_tools, summarise the counts, and arrange in descending order
tool_count <- tools %>%
  group_by(work_tools) %>%
  summarise(count = n()) %>%
  arrange(desc(count))

# Print the first six results
head(tool_count)
```

Work_tools	Count
Python	6073
R	4708
SQL	4261
Jupyter notebooks	3206
TensorFlow	2256
NA	2198

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

```{r}
# Create a bar chart of the work_tools column, most counts on the far right
ggplot(tool_count, aes(x=fct_reorder(work_tools, count), y = count)) +
  geom_bar(stat = "identity", fill = "#4D73BE") +
  theme(axis.text.x = element_text(angle=90, vjust = 0.5, hjust = 1)) +
  labs(title = "The most popular tools")
```



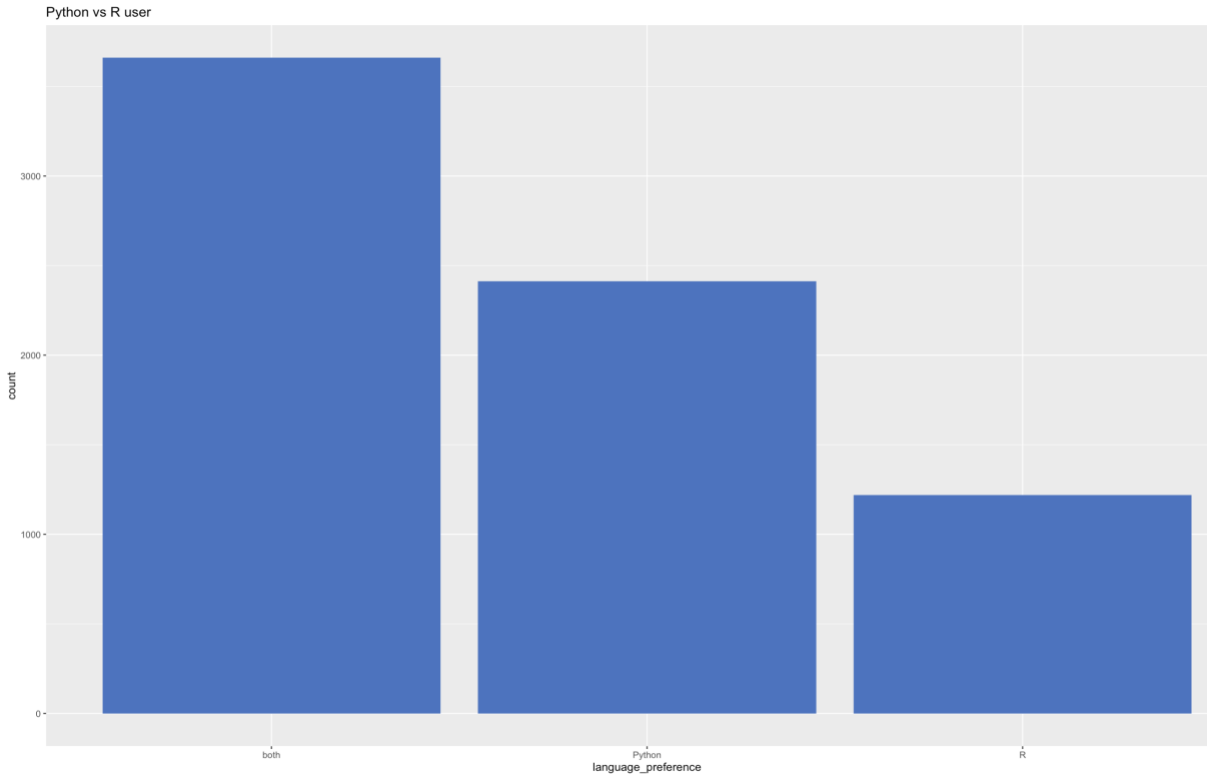
Within the field of data science, there is a lot of debate among professionals about whether R or Python should reign supreme. You can see from our last figure that R and Python are the two most commonly used languages, but it's possible that many respondents use both R and Python. Let's take a look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

```{r}
# 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(WorkToolsSelect, "R") ~ "Python",
     str_detect(WorkToolsSelect, "R") & str_detect(WorkToolsSelect, "Python") ~ "both",
    TRUE ~ "neither"
  ))

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

Respondent	WorkToolsSelect	LanguageRecommendationSelect	EmployerIndustry	WorkAlgorithmsSelect	language_preference
1	Amazon Web services,Oracle Data Mining/ Oracle R E...	F#	Internet-based	Neural Networks,Random Forests,RNNs	R
2	Amazon Machine Learning,Amazon Web services,Clou...	Python	Mix of fields	Bayesian Techniques,Decision Trees,Random Forests,...	both
3	C/C++,Jupyter notebooks,MATLAB/Octave,Python,R,...	Python	Technology	Bayesian Techniques,CNNs,Ensemble Methods,Neural ...	both
4	Jupyter notebooks,Python,SQL,TensorFlow	Python	Academic	Bayesian Techniques,CNNs,Decision Trees,Gradient B...	Python
5	C/C++,Cloudera,Hadoop/Hive/Pig,Java,NoSQL,R,Unix...	R	Government	*NA*	R
6	SQL	Python	Non-profit	*NA*	neither
7	Jupyter notebooks,NoSQL,Python,R,SQL,Unix shell / awk	Python	Internet-based	CNNs,Decision Trees,Gradient Boosted Machines,Ran...	both
8	Python,Spark / MLlib,Tableau,TensorFlow,Other	Python	Mix of fields	Bayesian Techniques,CNNs,HMMs,Neural Networks,Ra...	Python
9	Jupyter notebooks,MATLAB/Octave,Python,SAS Base,S...	Python	Financial	Ensemble Methods,Gradient Boosted Machines	Python
10	C/C++,IBM Cognos,MATLAB/Octave,Microsoft Excel ...	R	Technology	Bayesian Techniques,Regression/Logistic Regression	both
11	C/C++,Jupyter notebooks,Python,TensorFlow	Python	Academic	CNNs,Neural Networks	Python
12	MATLAB/Octave,Python	Matlab	Telecommunications	CNNs,Decision Trees,RNNs	Python

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

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

# Create a bar chart
ggplot(debate_plot, aes(x=language_preference, y = count)) +
  geom_bar(stat="identity",fill = "#4D73BE") + labs(title = "Python vs R user")
```



It looks like the largest group of professionals program in Python and R. But what happens when they are asked which language they recommend to new learners? Let's take a look at whether R users always recommend R.

```{r}
# Group by, summarise, arrange, mutate, and filter
```

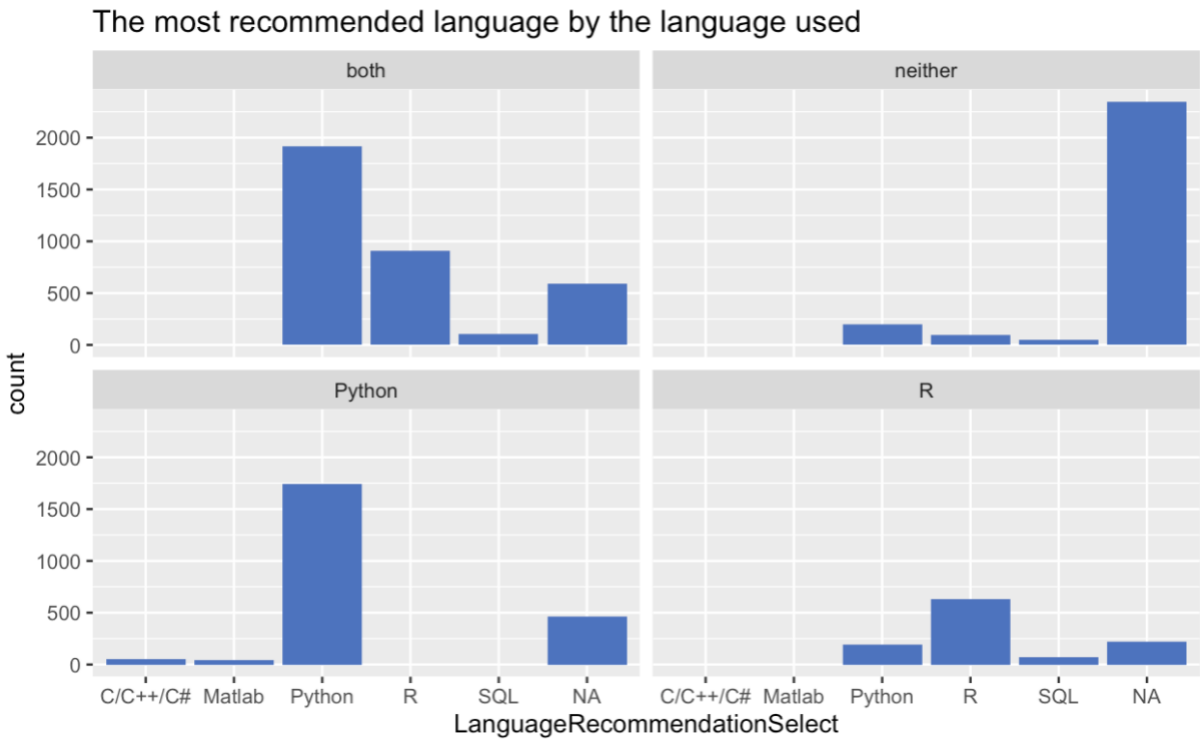
```
recommendations <- debate_tools %>%
  group_by(language_preference, LanguageRecommendationSelect) %>%
  summarise(count = n()) %>%
  arrange(language_preference, desc(count)) %>%
  mutate(row = row_number()) %>%
  filter(row<=4)
head(recommendations)
```

	language_preference	LanguageRecommendationSelect	count	row
1	both	Python	1917	1
2	both	R	912	2
3	both	*NA*	591	3
4	both	SQL	108	4
5	neither	*NA*	2348	1
6	neither	Python	196	2
7	neither	R	94	3
8	neither	SQL	53	4
9	Python	Python	1742	1
10	Python	*NA*	459	2

Conclusion

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

```{r}
# Create a faceted bar plot
ggplot(recommendations, aes(x=LanguageRecommendationSelect, y = count)) +
  geom_bar(stat="identity", fill = "#4D73BE") +
  facet_wrap(~language_preference) + labs(title="The most recommended language by the language used")
```



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.

Source

- https://www.kaggle.com/kaggle/kaggle-survey-2017?utm_medium=partner&utm_source=datacamp.com&utm_campaign=ml+survey+case+study
- https://www.guru99.com/r-vs-python.html