Exploring the Kaggle Data Science Survey

Datacamp Project Solution

04 jul, 2021

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.

```
# Load necessary packages
library(tidyverse)
```

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

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   Respondent = col_double(),
##   WorkToolsSelect = col_character(),
##   LanguageRecommendationSelect = col_character(),
##   EmployerIndustry = col_character(),
##   WorkAlgorithmsSelect = col_character()
## )
```

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

```
## # A tibble: 10 x 5
##    Respondent WorkToolsSelect LanguageRecomme~ EmployerIndustry WorkAlgorithmsS~
##         <dbl> <chr>           <chr>            <chr>            <chr>
##  1          1 Amazon Web ser~ F#               Internet-based   Neural Networks~
##  2          2 Amazon Machine~ Python           Mix of fields    Bayesian Techni~
##  3          3 C/C++,Jupyter ~ Python           Technology       Bayesian Techni~
##  4          4 Jupyter notebo~ Python           Academic         Bayesian Techni~
##  5          5 C/C++,Cloudera~ R                Government       <NA>
##  6          6 SQL             Python           Non-profit       <NA>
##  7          7 Jupyter notebo~ Python           Internet-based   CNNs,Decision T~
##  8          8 Python,Spark /~ Python           Mix of fields    Bayesian Techni~
##  9          9 Jupyter notebo~ Python           Financial        Ensemble Method~
## 10         10 C/C++,IBM Cogn~ R                Technology       Bayesian Techni~
```

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.

```
# Print the first respondent's tools and languages
head(responses, 1)
```

```
## # A tibble: 1 x 5
##   Respondent WorkToolsSelect  LanguageRecomme~ EmployerIndustry WorkAlgorithmsS~
##        <dbl> <chr>            <chr>            <chr>            <chr>
## 1          1 Amazon Web serv~ F#               Internet-based   Neural Networks~
```

```
# 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)
```

```
## # A tibble: 6 x 6
##   Respondent WorkToolsSelect  LanguageRecomme~ EmployerIndustry WorkAlgorithmsS~
##        <dbl> <chr>            <chr>            <chr>            <chr>
## 1          1 Amazon Web serv~ F#               Internet-based   Neural Networks~
## 2          1 Amazon Web serv~ F#               Internet-based   Neural Networks~
## 3          1 Amazon Web serv~ F#               Internet-based   Neural Networks~
## 4          2 Amazon Machine ~ Python           Mix of fields    Bayesian Techni~
## 5          2 Amazon Machine ~ Python           Mix of fields    Bayesian Techni~
## 6          2 Amazon Machine ~ Python           Mix of fields    Bayesian Techni~
## # ... with 1 more variable: work_tools <chr>
```

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.

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

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

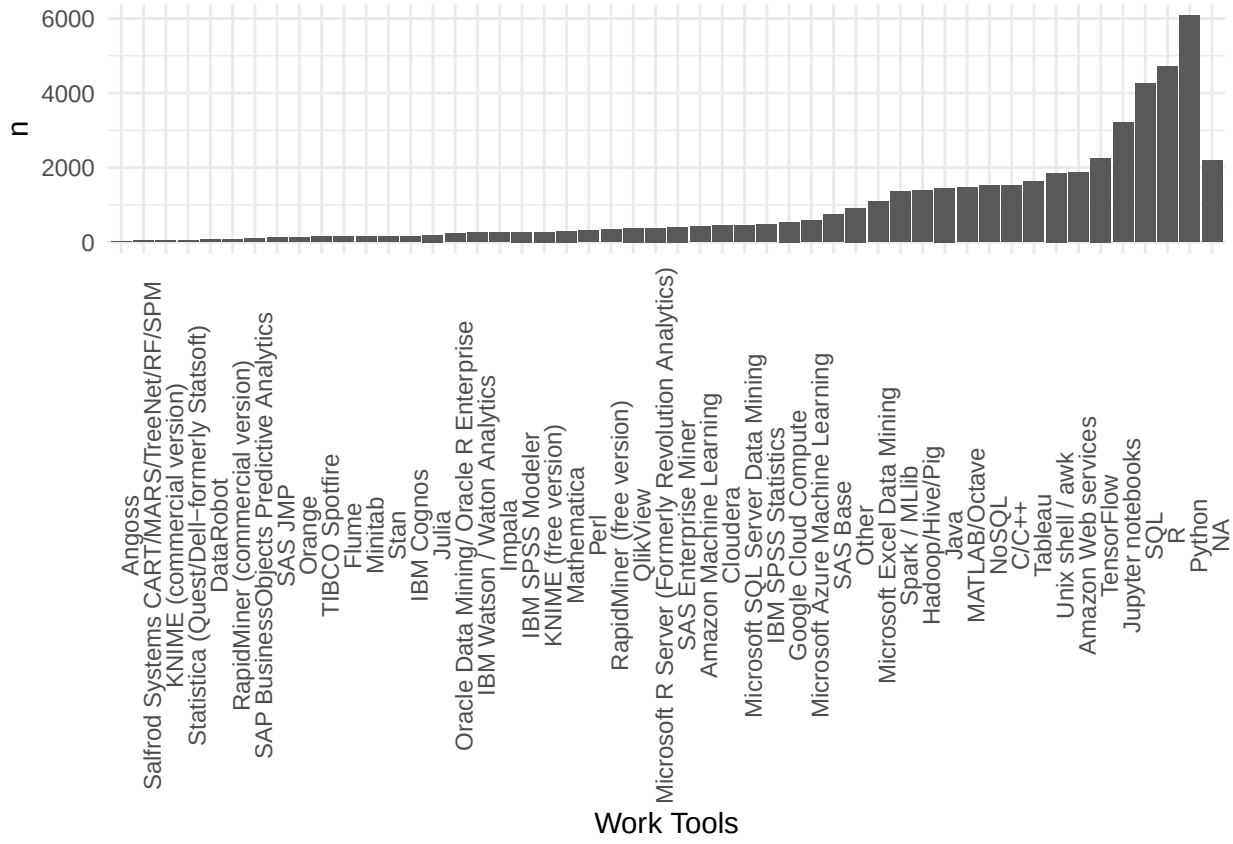
```
## # A tibble: 6 x 2
##   work_tools            n
##   <chr>             <int>
## 1 Python             6073
```

```
## 2 R                    4708
## 3 SQL                  4261
## 4 Jupyter notebooks    3206
## 5 TensorFlow           2256
## 6 <NA>                 2198
```

4 Plotting the most popular tools

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

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



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.

```
# 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)
```

```
## # A tibble: 6 x 6
##   Respondent WorkToolsSelect  LanguageRecomme~ EmployerIndustry WorkAlgorithmsS~
##        <dbl> <chr>            <chr>            <chr>            <chr>
## 1          1 Amazon Web serv~ F#               Internet-based   Neural Networks~
## 2          2 Amazon Machine ~ Python           Mix of fields    Bayesian Techni~
## 3          3 C/C++,Jupyter n~ Python           Technology       Bayesian Techni~
## 4          4 Jupyter noteboo~ Python           Academic         Bayesian Techni~
## 5          5 C/C++,Cloudera,~ R                Government       <NA>
## 6          6 SQL              Python           Non-profit       <NA>
## # ... with 1 more variable: language_preference <chr>
```

6 Plotting R vs Python users

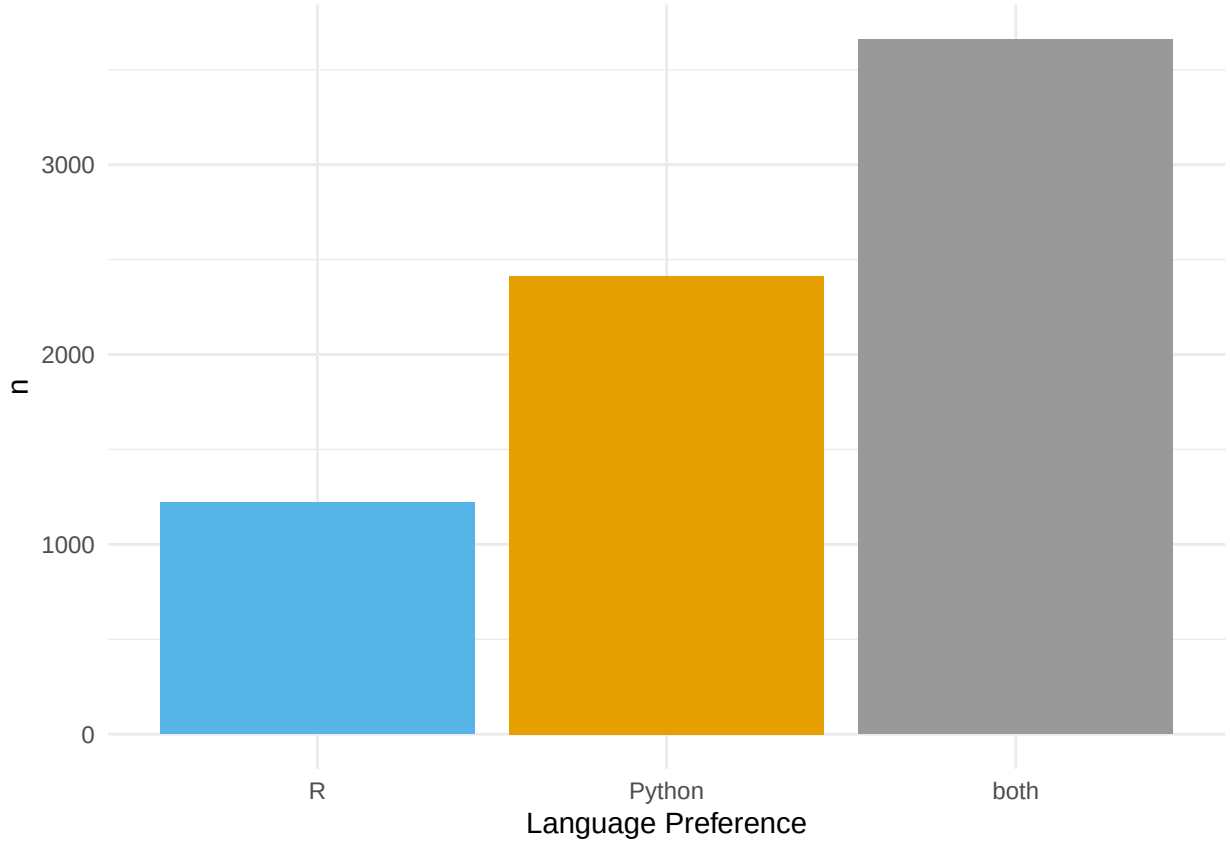
Now we just need to take a closer look at how many respondents use R, Python, and both!

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

debate_plot
```

```
## # A tibble: 3 x 2
##   language_preference     n
##   <chr>               <int>
## 1 both                 3660
## 2 Python               2413
## 3 R                    1220
```

```
# Create a bar chart
ggplot(debate_plot,
       aes(
         x = fct_reorder(language_preference, n),
         y = n,
         fill = language_preference
       )) +
  geom_bar(stat = "identity") +
  xlab("Language Preference") +
  scale_fill_manual(values = c("#999999", "#E69F00", "#56B4E9")) +
  theme_minimal() +
  theme(legend.position = "none")
```



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?

```
# Group by, summarise, arrange, mutate, and filter
recommendations <- debate_tools  %>%
  group_by(language_preference, LanguageRecommendationSelect)  %>%
  summarise(count = n()) %>%
  arrange(desc(count)) %>%
  mutate(row_numbers = row_number()) %>%
  filter(row_numbers <= 4)
```

```
## `summarise()` has grouped output by 'language_preference'. You can override using the `.groups` argu
```

```
recommendations
```

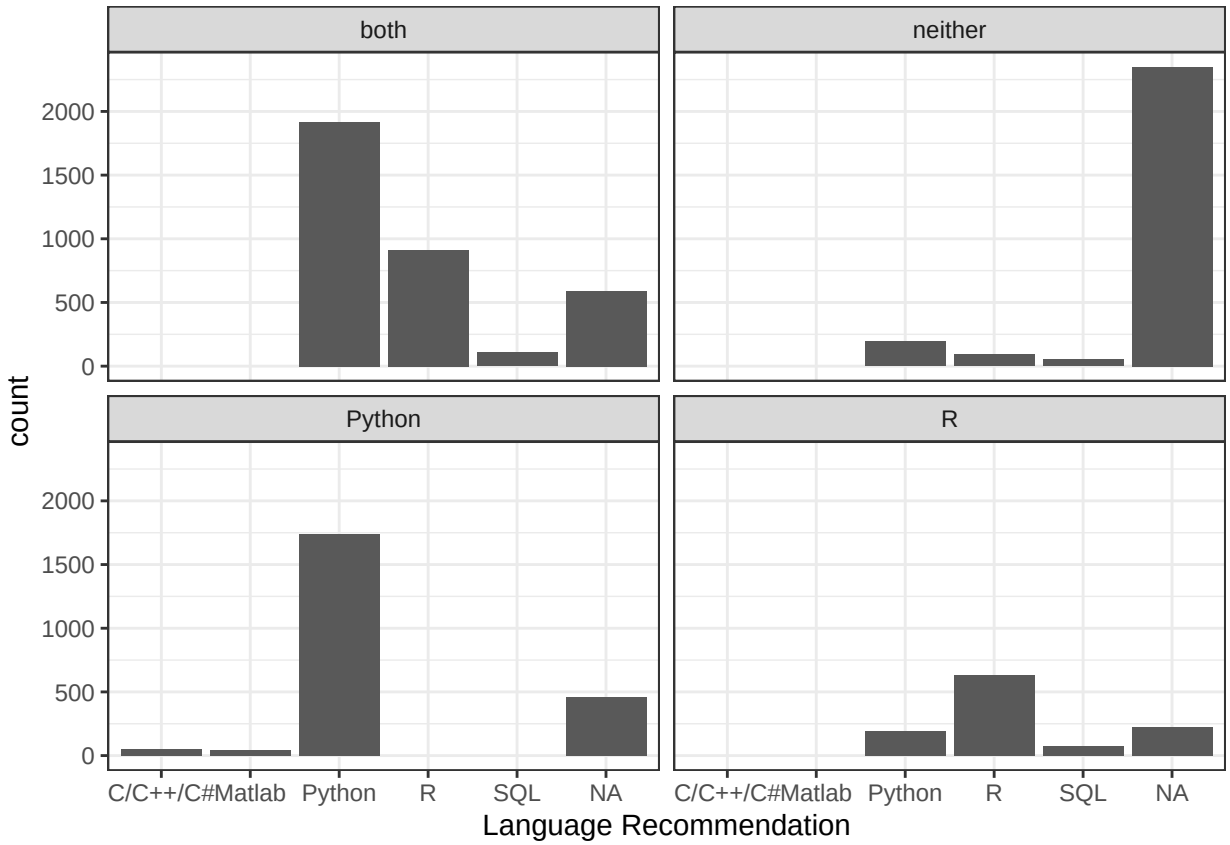
```
## # A tibble: 16 x 4
## # Groups:   language_preference [4]
##    language_preference LanguageRecommendationSelect count row_numbers
##    <chr>               <chr>                         <int>       <int>
##  1 neither             <NA>                           2348           1
##  2 both                Python                         1917           1
##  3 Python              Python                         1742           1
##  4 both                R                               912           2
##  5 R                   R                               632           1
##  6 both                <NA>                            591           3
```

```
##  7 Python              <NA>                459           2
##  8 R                   <NA>                221           2
##  9 neither             Python              196           2
## 10 R                   Python              194           3
## 11 both                SQL                 108           4
## 12 neither             R                    94           3
## 13 R                   SQL                  75           4
## 14 neither             SQL                  53           4
## 15 Python              C/C++/C#             48           3
## 16 Python              Matlab               43           4
```

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.

```
# Create a faceted bar plot
ggplot(recommendations, aes(x = LanguageRecommendationSelect, y = count)) +
  geom_bar(stat = "identity") +
  facet_wrap( ~ language_preference) +
  xlab("Language Recommendation") +
  theme_bw()
```



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`?

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