Analysis of Data Scientist Job Positions

Jung Hwa Yeom

12/14/2020

Packages that I use

```
webshot::install_phantomjs()
library(tidytext)
library(leaflet)
library(tnum)
library(httr)
library(rjson)
library(RCurl)
library(XML)
library(magrittr)
library(dplyr)
library(ggplot2)
library(tidyverse)
```

Data

Data from Json Files

```
r <- jsonlite::fromJSON("https://jobs.github.com/positions.json?description=r")
node <- jsonlite::fromJSON("https://jobs.github.com/positions.json?search=node")
stat <- jsonlite::fromJSON("https://jobs.github.com/positions.json?description=stat")
whole <- jsonlite::fromJSON("https://jobs.github.com/positions.json?")
dt <-jsonlite::fromJSON("https://jobs.github.com/positions.json?description=data")
al <-jsonlite::fromJSON("https://jobs.github.com/positions.json?description=algorithm")
ml <-jsonlite::fromJSON("https://jobs.github.com/positions.json?description=machine")
whole2 <- jsonlite::fromJSON("https://jobs.github.com/positions.json?company=Amazon")
whole3 <- jsonlite::fromJSON("https://jobs.github.com/positions.json?company=google")
muse <- jsonlite::fromJSON("https://www.themuse.com/api/public/jobs?page=10")
```

Bind data

```
# Bind data to make it into one data frame
total <- rbind(r, node, stat, whole, dt, al, ml, whole2, whole3)
total2 <- data.frame(total[1], total[2], total[5], total[6], total[7], total[8], total[9])
```

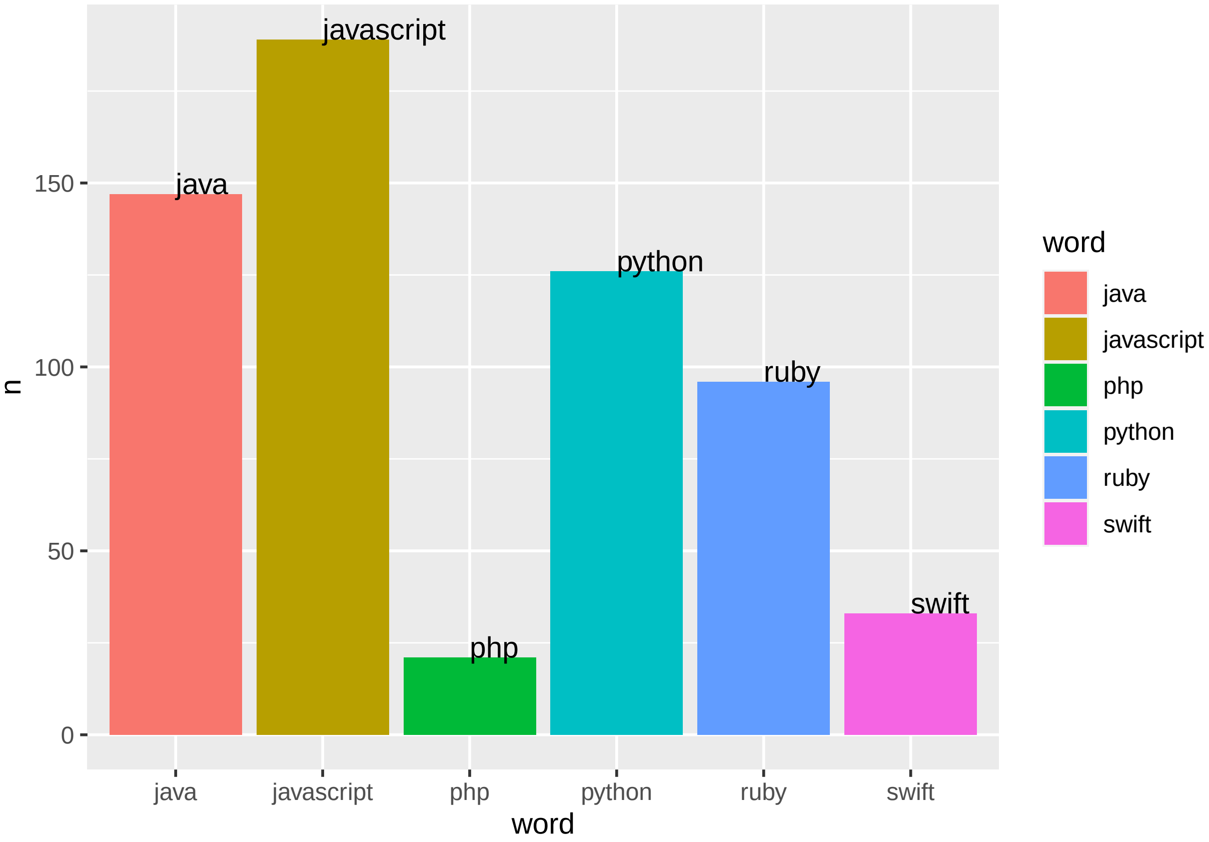
Text Analysis in Job Descriptions

```
# Remove stop-words, e.g. the, a, this, and that ...
a <- total2 %>% unnest_tokens(word, description)

data(stop_words)

a <- a %>%anti_join(stop_words)

a<- a %>%
  count(word, sort = TRUE)
```

Popular Programming Language

Javascrip, java, php, python, ruby, and swift are popular programming languages, so many companies are looking for employees who are proficient at those languages. According to https://towardsdatascience.com/top-10-in-demand-programming-languages-to-learn-in-2020-4462eb7d8d3,python, javascript, java, c#, c, c++, php, swift, go, and ruby are the most popular programming languages. Unfortunately, I could not find c#,c,c++ because they have only one character 'c' or are combined with special characters. e.g. # and ++.

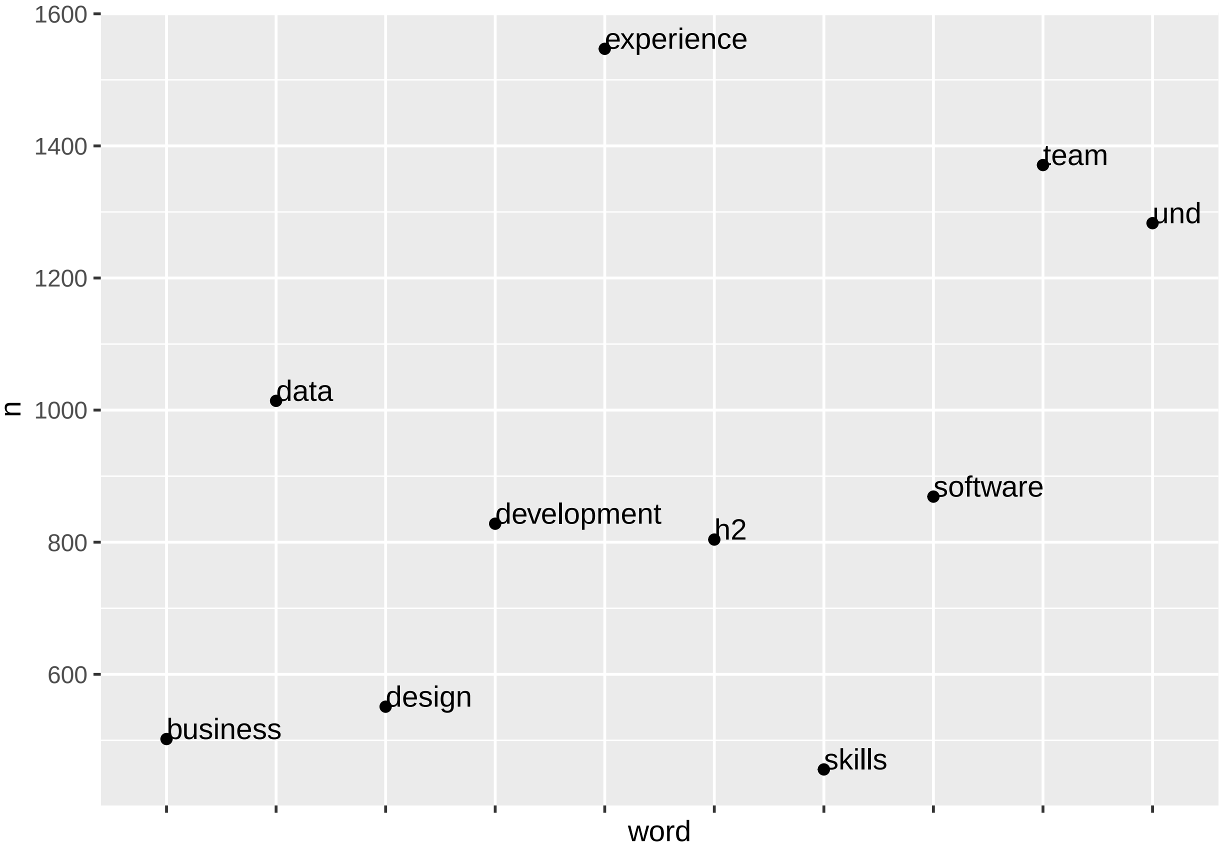
```
# Find programming languages in job description
prog_pop <- a %>% filter((word=="java")|(word=="python")|(word=="javascript")|
                           (word=="php")|(word=="swift")|(word=="go")|(word=="ruby"))

# Plot popular programming languages
ggplot(data=prog_pop, aes(x=word, y=n))+geom_bar(stat='identity', aes(fill=word))+
  geom_text(aes(label=word), hjust=0, vjust=0)
```



Word Analysis in Job Descriptions

First of all, I removed the words ‘li’ and ‘ul’ from the list. To be specific, ‘li’ means lists and ‘ul’ means unordered lists in html. Therefore, they are not related to jobs. After cleaning the list, I chose the top 10 words that are most included in job descriptions. The top 10 words are experience, team, und, data, software, development, h2, design, business, and skills. ‘und’ means and in German. Many job postings in my data are from German companies and stop_words function only works in English. Thus, I could not clean German stop words. We can interpret based on this ggplot that companies want to hire data scientists who have much experience, teamwork, software skills. Obviously, data scientists are required to fluently deal with data. We can prepare to become a data scientist by analyzing job descriptions because we can notice what skills companies want from data scientists.

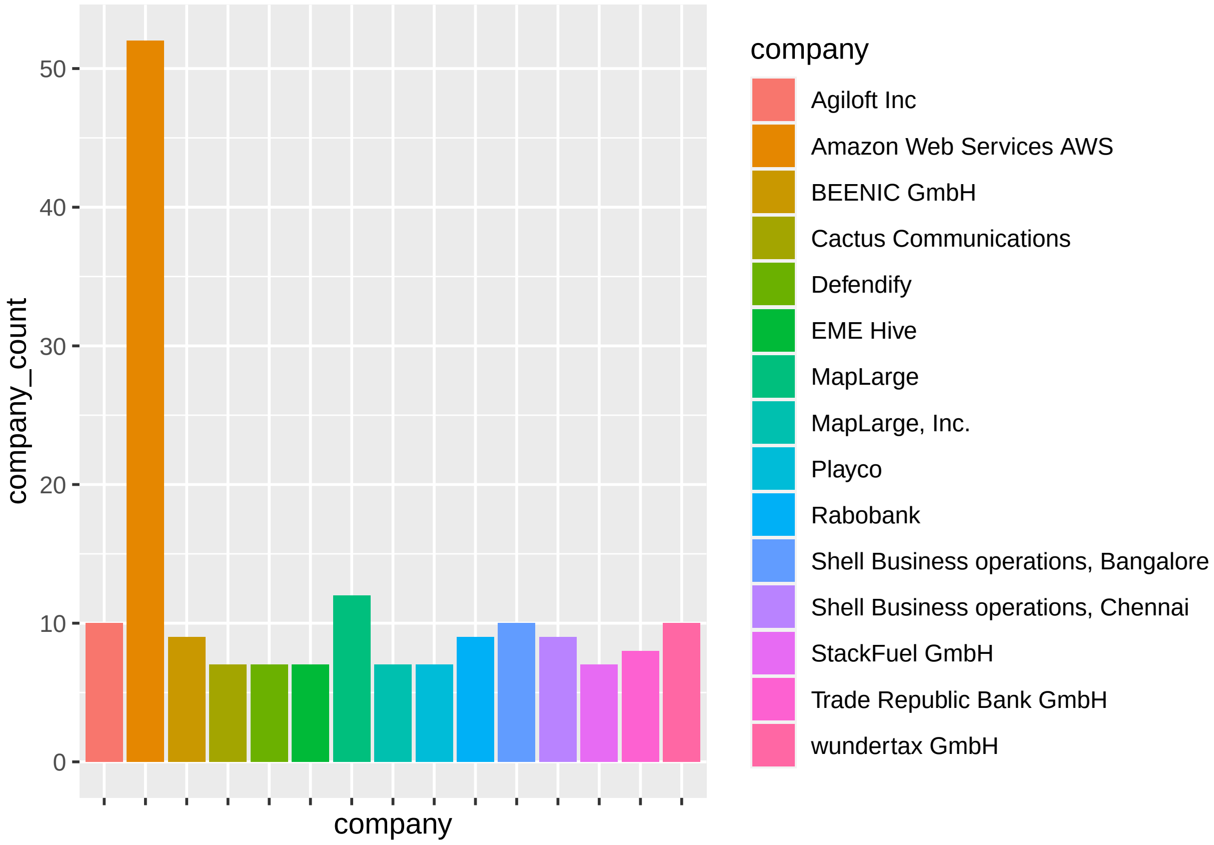
```
# Choose top 10 words that are included in the description.
b <- data.frame(a) %>% filter(n<2000) %>% top_n(10, n)

# Plot top 10 words
ggplot(data=b,aes(x=word, y=n))+geom_point()+geom_text(aes(label=word), hjust=0, vjust=0)+
  theme(axis.text.x=element_blank())
```



The Industry of Companies

I would like to figure out which industry hires a data scientist. There were many companies in the dataset and I chose top 10 companies which posted job positions most. Most of the companies are frome IT industry. For example, Agiloft is a software company in San Francisco and Amazon is a technology company in Seattle. We can conclude IT industry demands data scientists.

```r
# top 10 companies that posted job positions
total2_summary <- total2 %>% group_by(company) %>%
  summarise(company_count = n()) %>%top_n(10, company_count)

# plot top 10 companies
ggplot(total2_summary, aes(company, company_count))+
  geom_bar(stat='identity', aes(fill=company))+
  theme(axis.text.x=element_blank())
```



Location ## Data Cleaning of Location

The same location was written in many ways. For instance, Munchen, Munch, and Munich all mean Munich, so I unified them into Munich. I unified the words meaning the same location into one.

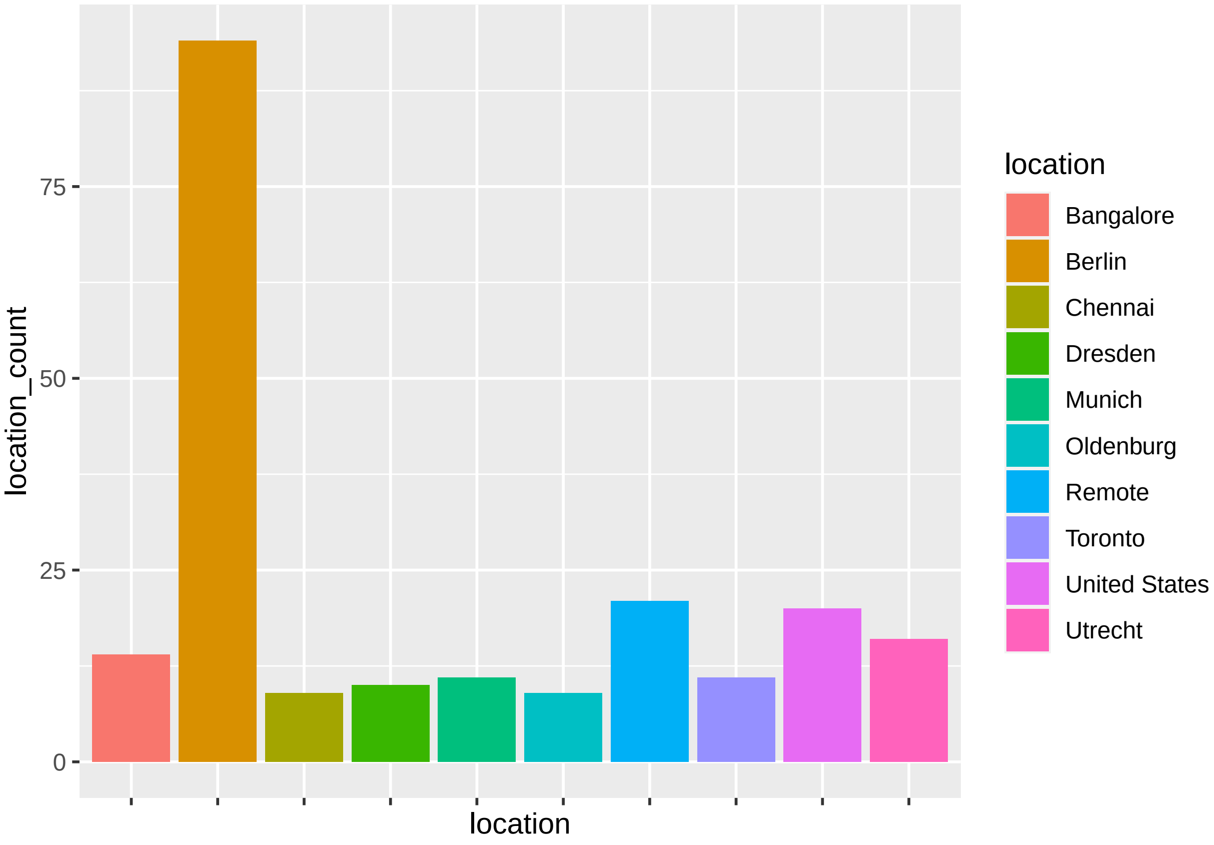
```
total2_location <- total2 %>%group_by(location)
total2_location$location[total2_location$location ==
                           "Berlin / Remote"] <- "Berlin"
total2_location$location[total2_location$location ==
                           "Berlin | Remote"] <- "Berlin"
total2_location$location[total2_location$location ==
                           "Berlin, BE, DE"] <- "Berlin"
total2_location$location[total2_location$location ==
                           "Garching, Munich"] <- "Munich"
total2_location$location[total2_location$location ==
                           "München"] <- "Munich"
total2_location$location[total2_location$location ==
                           "Munch Germany "] <- "Munich"
total2_location$location[total2_location$location ==
                           "Munich Germany"] <- "Munich"
total2_location$location[total2_location$location ==
                           "NYC / Remote"] <- "New York City"
total2_location$location[total2_location$location ==
                           "remote" ] <- "Remote"
total2_location$location[total2_location$location ==
                           "Remote (USA)" ] <- "Remote"
total2_location$location[total2_location$location ==
                           "Remote in U.S." ] <- "Remote"
```

```
total2_location$location[total2_location$location ==
                           "Remote, EU" ] <- "Remote"
total2_location$location[total2_location$location ==
                           "Utrecht (The Netherlands)" ] <- "Utrecht"
total2_location$location[total2_location$location ==
                           "Croeselaan 18, 3521CB, Utrecht" ] <- "Utrecht"
total2_location$location[total2_location$location ==
                           "Utrecht " ] <- "Utrecht"
total2_location$location[total2_location$location ==
                           "San Francisco |Remote (US/Canada)" ] <- "San Francisco"
total2_location$location[total2_location$location ==
                           "Soho, London" ] <- "London"
total2_location$location[total2_location$location ==
                           "Europe (remote)" ] <- "Remote"
total2_location$location[total2_location$location ==
                           "Seattle / Fully Remote" ] <- "Romote"
total2_location$location[total2_location$location ==
                           "Toronto, ON - REMOTE" ] <- "Toronto"
total2_location$location[total2_location$location ==
                           "Toronto, Canada (or remote within Canada)" ] <- "Toronto"
```

```
# top 10 locations
total2_location <- total2_location %>%group_by(location) %>%
  summarise(location_count =n())
location_top <- total2_location %>% slice_max(location_count, n=10)
```

Bangalore, Berlin, Chennai, Dresden, Munich, Oldenburg, Toronto, United States, and Utrecht are the most popular locations. Because of the COVID-19, many companies posted remote positions. United States is the third place on the list, but it would go up because this value did not include locations such as San Francisco, New York City, etc. Some companies wrote just United States and other companies specified the states or cities.

```
# plot top 10 locations
ggplot(location_top, aes(location, location_count))+
  geom_bar(stat='identity', aes(fill=location))+
  theme(axis.text.x=element_blank())
```



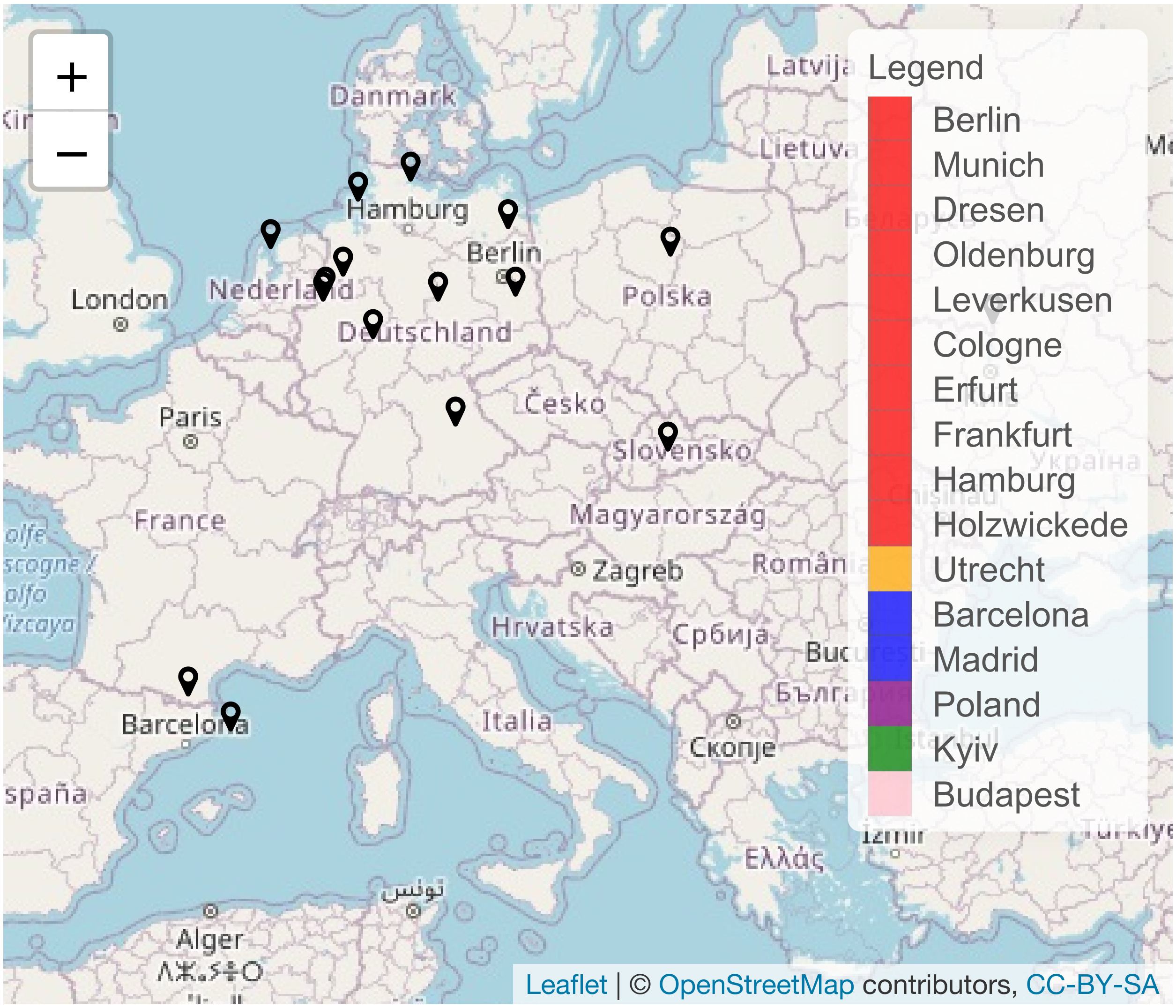
Map Locations in Europe

I would like to map locations where at least 5 positions are opened. I limited the locations in Europe. There were 16 locations in Europe and most of them are located in Germany including Berline and Munich. I put latitude and longitude to map.

```
dt.map <- data.frame(
  city = c('Berlin', 'Munich', 'Dresen', 'Oldenburg','Leverkusen',
           'Cologne', 'Erfurt', 'Frankfurt', 'Hamburg', 'Holzwickede',
           'Utrecht',
           'Barcelona', 'Madrid', 'Poland',
           'Kyiv', 'Budapest'),

  lat = c(52.52, 48.1351, 51.0504, 53.1435, 51.0459,
          50.9375, 50.9848, 50.1109, 53.5511, 51.4998,
          52.0907,
          41.3851, 40.4168, 51.9194,
          50.4501, 47.4979),

  lng = c(13.4050, 11.5820,13.7373, 8.2146,7.0192,
          6.9603, 11.0299, 8.6821, 9.9937, 7.6209,
          5.1214,
          2.1734, 3.7038, 19.1451,
          30.5234, 19.0402),
  col = c("red", "red","red","red","red",
          "red", "red","red","red","red",
          "orange",
          "blue", "blue", "purple",
          "green", "pink")
)
```

I distinguish countries by colors. For example, red color indicates Germany and blue color indicates Spain.

```
dt.map$popup <- with(dt.map, paste("<b>", city, "</b>"))

markers <- awesomeIcons(
  icon='map-marker',
  iconColor = 'black',
  markerColor = dt.map$col,
  library='fa')

map <- leaflet(data = dt.map, width = "100%" ) %>%
  addTiles() %>%  # Add default OpenStreetMap map tiles
  addAwesomeMarkers(
    lng = ~lng,
    lat = ~lat,
    popup = ~popup,
    icon = markers
  ) %>%
  addLegend(
    position='topright',
    colors= c("red","red","red","red","red",
              "red", "red","red","red","red",
              "orange",
              "blue", "blue", "purple",
              "green", "pink"),
    labels= dt.map$city,
    opacity = 0.75,
    title="Legend"
  )

map  # Show map
```



Thoughts and Future Steps

I aim to become a data scientist after graduating MSSP, so this project was interesting and helpful for me to thoroughly analyze data scientist job postings. I could notice which skills I should prepare for getting a job. Also, proficient programming skills are a basic requirement and teamwork and communication skill are important. What I most like about a data scientist as a job is that a data scientist can work all over the world. It was my first time using API data, so it took a while to figure out how to use API data. We mostly used csv file for the assignment, but this time we are required to use json file. Therefore, it was more complicated to deal with the data. As I mentioned above, I could not fully clean the data which are written in German. My future step will discover how to text mining the data which includes various different languages. Furthermore, I would like to analyze larger API data.