HOW TO GET A DATA JOB AT

Google

Jasmijn van Hulsen

Text Analytics & NLP **February 10, 2021**

Table of Contents

Introduction

For many data lovers finding a data job at Google is the dream. However, with steep competition it might seem that it will always be this: a dream. Are there ways to increase your chances and how can you skill up in order to make that dream a reality? Lastly, with a constantly changing job title, if you have skilled up and are ready to apply, to what job should you apply?

This is what I have researched. I have taken 31 different data related job descriptions from Google's own website (Google, 2021) and divided those job descriptions into either Data Analyst roles or Data Scientists roles. In the following analysis I will compare keywords, skills and experience required.

I have also taken 10 job descriptions from Netflix's career website (Netflix, 2021) and will compare them with Google, so you can see the differences and know if Google is really the right company for you.

Token and TF_IDF analysis

It is to no-one's surprise that the word Data is the most common word among all Google jobs. But you might expect there to be a lot of technical words on top as well. As can be seen in figure 1, this is not the case. Rather business-related words are commonly used, such as team, insights, analysis, and business. This is of course a very shallow first analysis, so let's dive deeper, but it is good to keep in mind.

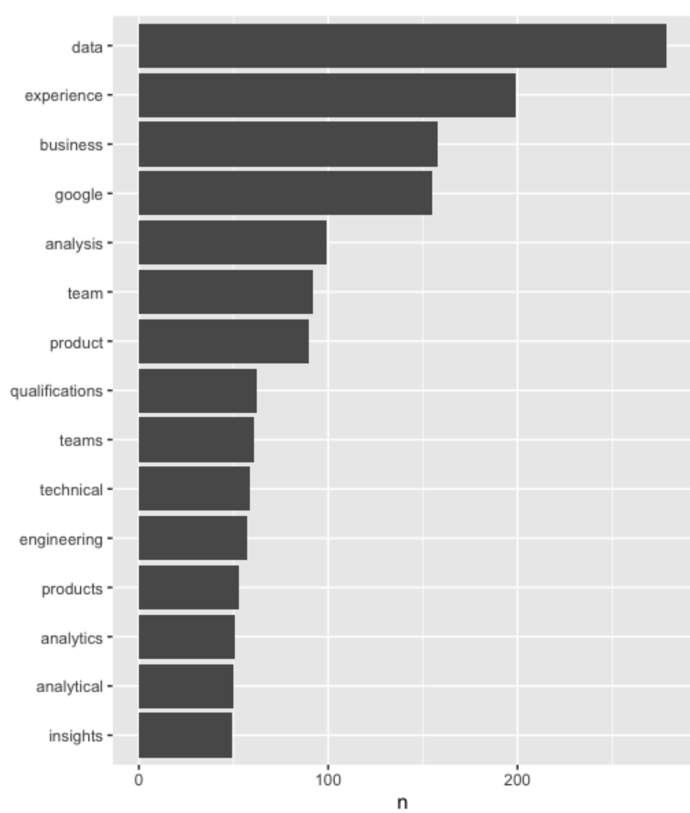

Figure 1: Word Frequency

First take-away:

Don't just focus on the technical requirements for a job when deciding to apply. Have a good look at the rest of the description and see if there are business related aspects that are commonly repeated and build your application around those words.

So, let's not only look at the frequency of words within the job descriptions, but how important these words are for the descriptions and compare these between the 2 Google jobs directions and the jobs at Netflix (figure 2).

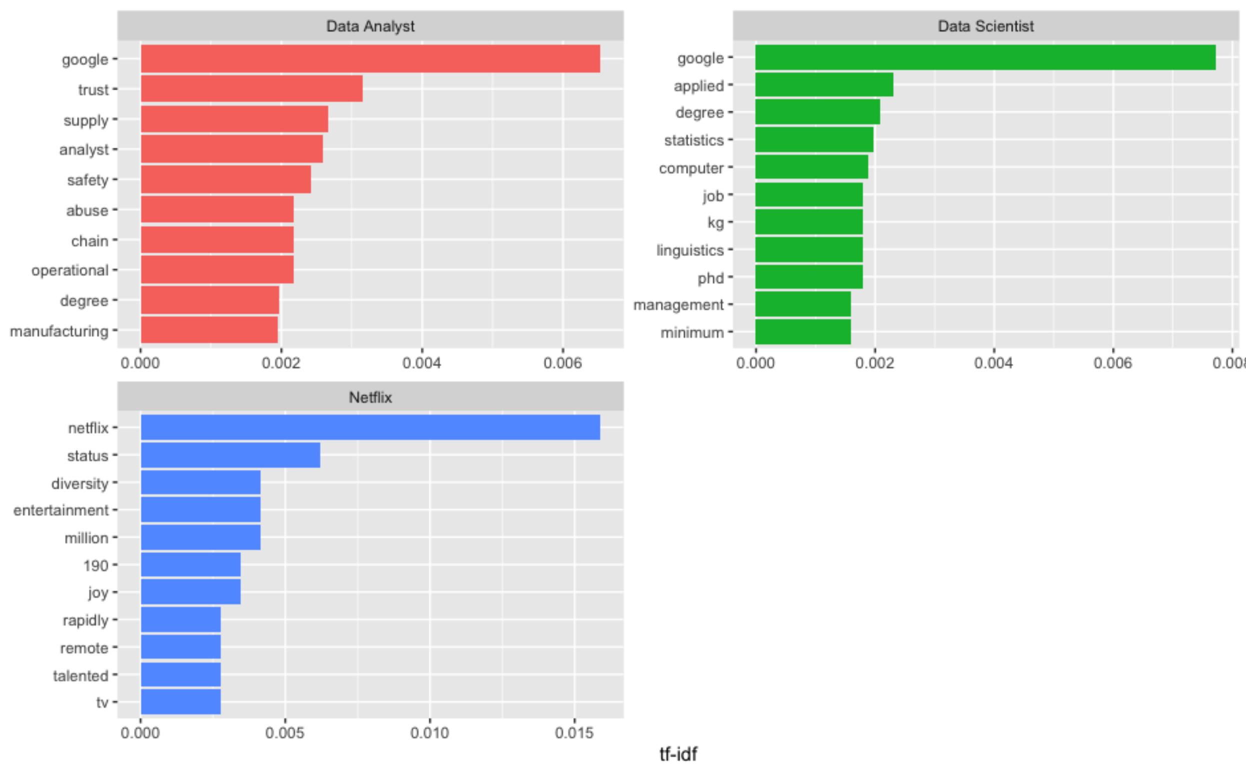

Figure 2: Most important words categorized according to job type and Netflix jobs

When we first compare Netflix with the two Google groups, we find that Netflix focusses a lot more on the company itself and work environment, whereas Google focusses more on the job responsibilities and area itself.

Comparing the Google job groups, we see that if you want to pursue a Data Analyst career, you have to focus more on processes, such as safety, supply chain or manufacturing. For example if Data Analyst at Google have a specialty make sure to apply for the analyst job that best fits your background or highlight your specialized knowledge in a portfolio.

The Data Scientist words are more focused on degrees, education and management experience. So skill up or focus on your education if becoming a Data Scientist is your goal.

Second take-away:

It seems that Netflix focusses more on their culture in their job postings. This gives the impression of an informal vibe where the connection between employees and Netflix are closer. Before applying at any company, you should consider how important this is to you.

Sentiment analysis

In order to confirm the above take-away, I took a look at the sentiments of the words. First, let's analyze the positivity vs negativity in the wording of the postings. In the word clouds (figure 3.1 and figure 3.2), you can see that both Google and Netflix use an overwhelmingly larger amount of positive (green) words than negative (red) ones. It could even be argued that some negative words such as cloud, fraud and plot aren't negative words at all, but rather describe work-related (data) words. It is to be expected that companies phrase their job descriptions positively, so no real insights here.

Figure 3.1: Bing Word cloud – Google

Figure 3.1: Bing Word cloud – Netflix

As a lot of "negative" words are actually job-related words that have no emotional sentiment. I have not conducted any additional sentiment analysis.

Bigram analysis

Loose words give a good insight into important common words, but sometimes they are taken out of context or don't mean anything at all by themselves (such as the word "business"). The following two analysis look at word groups to see if we can extract further insights from combining words. Figure 4 shows these connections, and I would like to focus on the following 3: data, business, and skills.

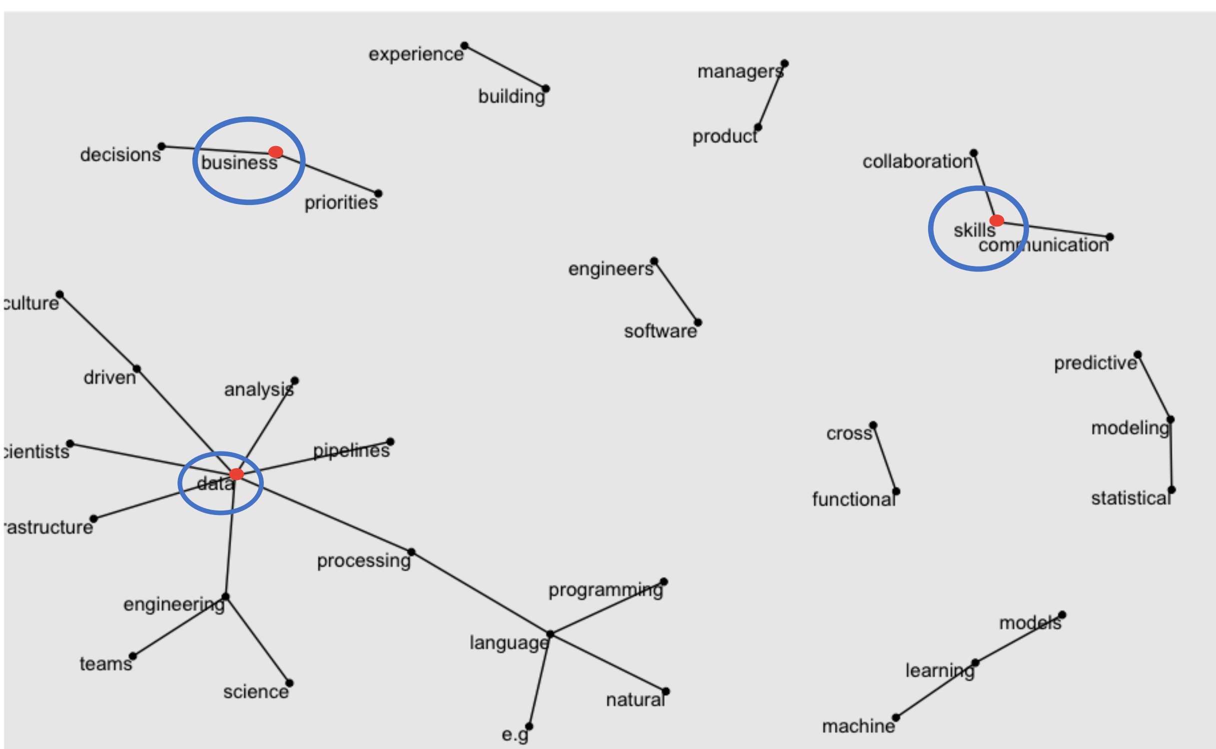

Figure 4: Bigram relationships between words

Data

Starting with the word that has the most linkages to other words – data – we can see that all three groups are similar in what they are expecting (figure 5.1). One can argue though that Netflix and Google's Data Scientist roles are slightly more focused on the data skills (groups such as data science, data engineering, data analysis, data structures), whereas the Data Analyst role describes more the tasks you will fulfil (data visualization, data feeds, data mining, data sets, data center).

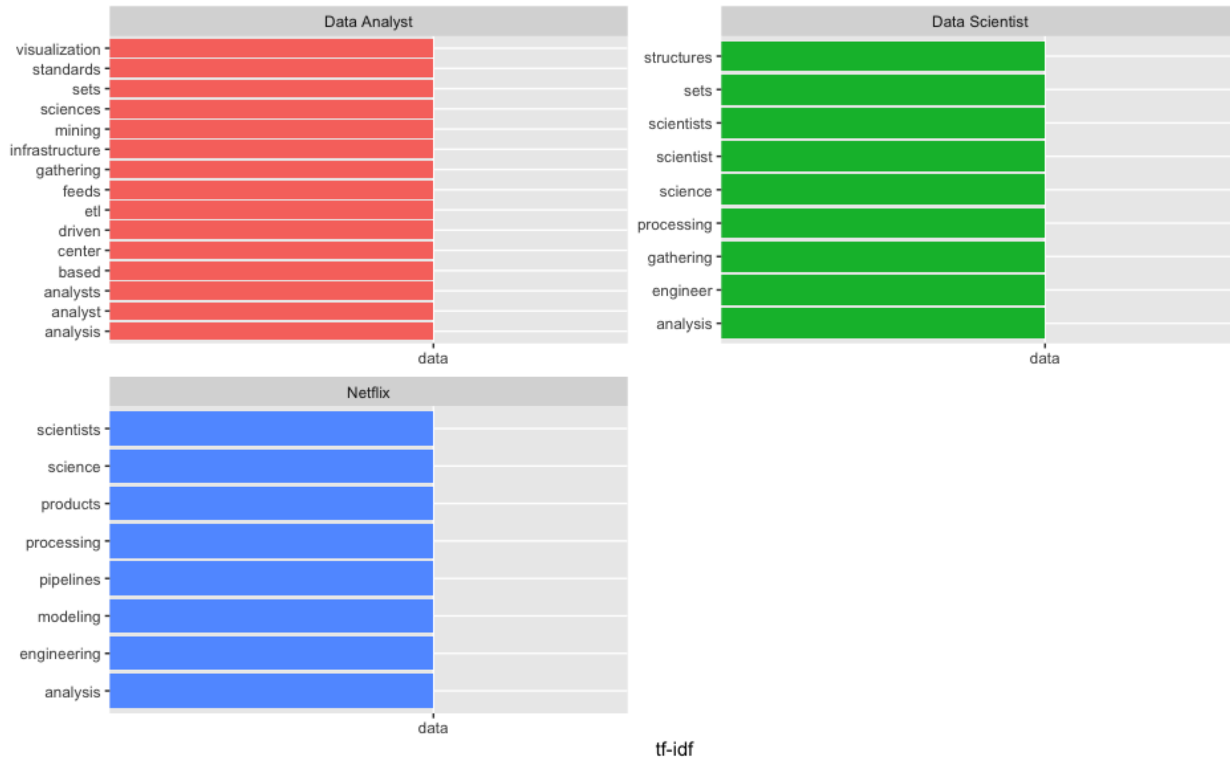

Figure 5.1: Bigram words connected to Data

Business

A more obvious difference between Google and Netflix is seen when looking at words that are connected with Business (figure 5.2). Where Google focusses more on the business requirements and recommendations and makes use of business intelligence, Netflix lookes more at the processes and stakeholders. This is not only good to know for deciding if Google is the right fit for you, but also for tailoring your application.

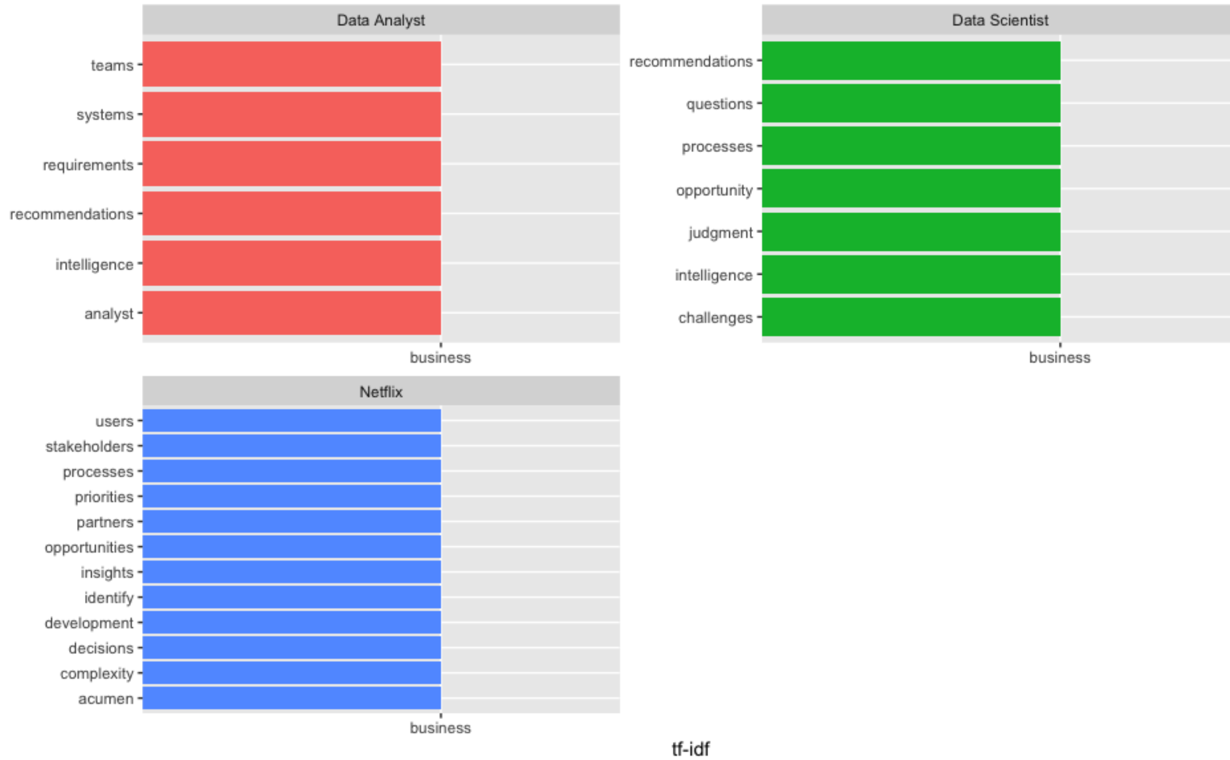

Figure 5.2: Bigram words connected to Business

Skills

Lastly, let's compare words that are related to skills (figure 5.3). Here you can see a true difference between job descriptions at Netflix vs Google. Netflix really focusses on technical skills, such as SQL, programming and modeling. Contrarily, Google focusses much more on personal and professional skills, such as (problem) solving, organization, communication and collaboration. There are not many differences between the two job categories within Google.

Third take-away:

When applying for a job, really dive into the types of skills a company is looking for. When applying for Google, don't just focus on your technical skills, but also your professional and personal ones. This reenforces the first take-away.

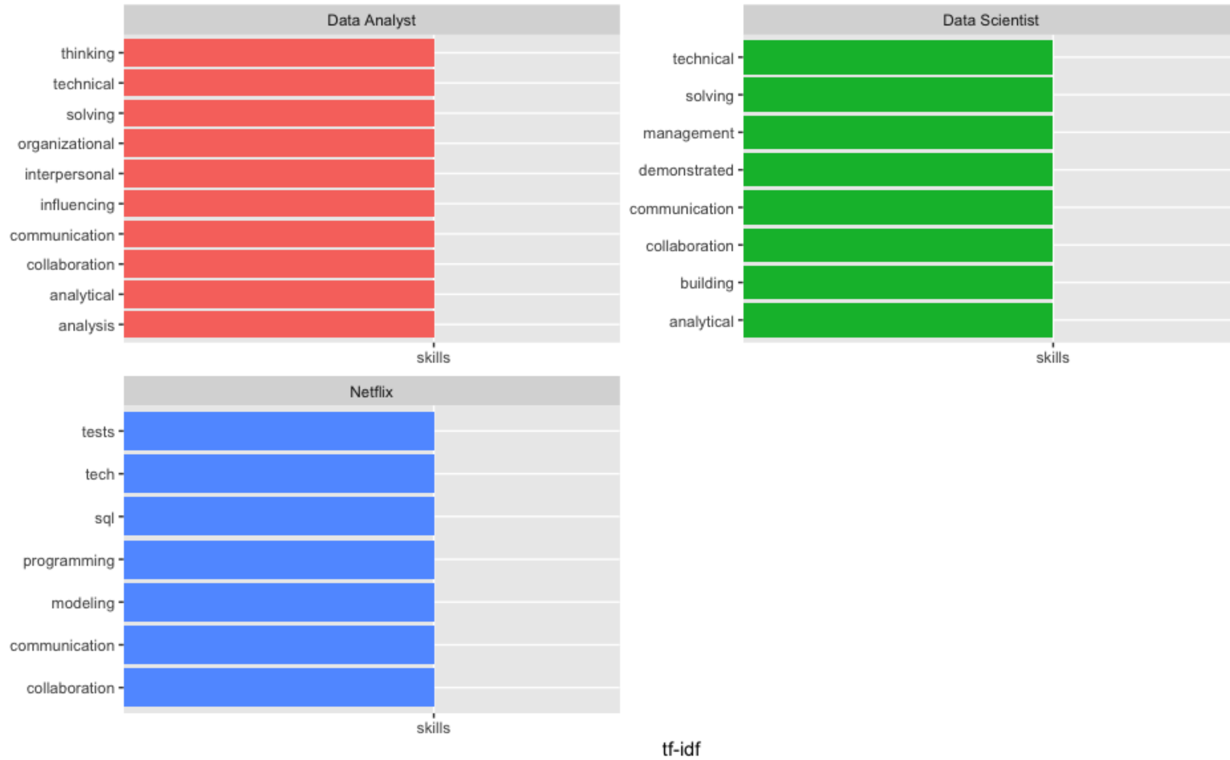

Figure 5.3: Quadrogram relationships between words

Quadrogram analysis

The second word combination is looking at combinations of 4. Hereby, I am no longer comparing Google with Netflix, but just diving deeper into the specific words that differentiate Data Analyst positions from Data Science ones.

Figure 6 looks again at the relationships between words and I want to take a better look at what the connected words show about: product, experience, and analysis (figures 7.1 to 7.3).

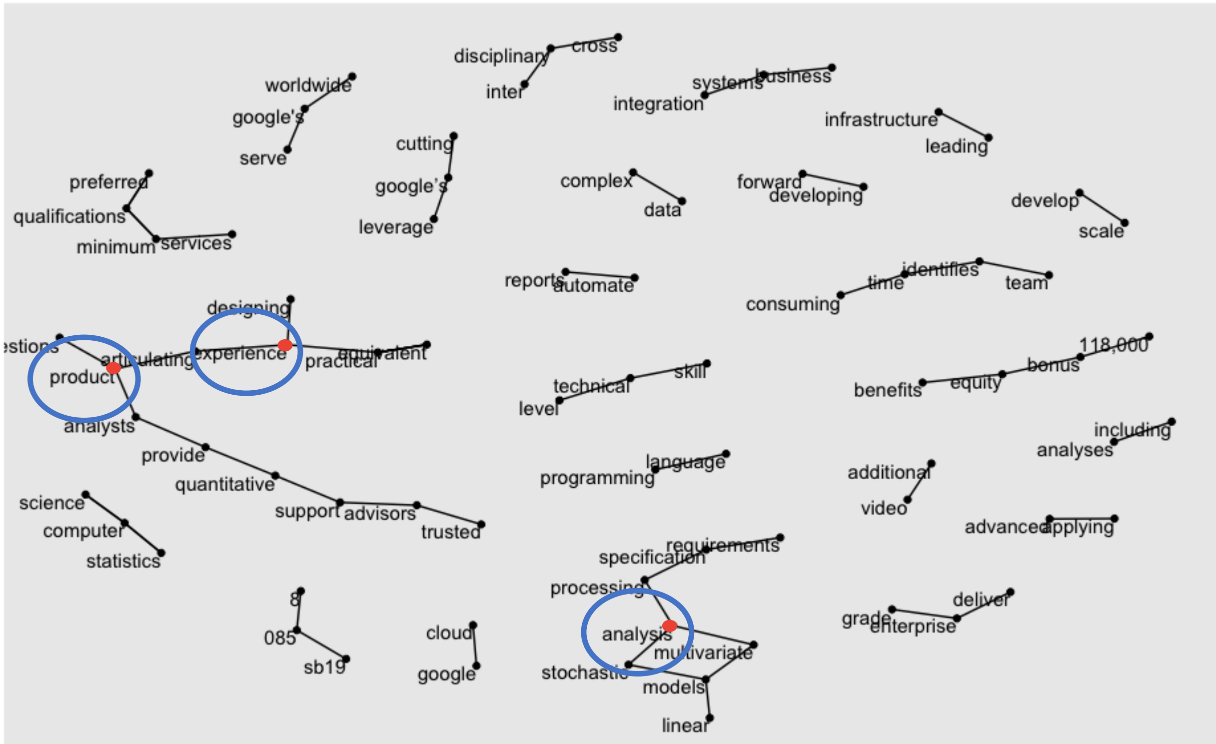

Figure 6: Quadrogram relationships between words

Product

The 2 charts in figure 7.1 show the word Product followed by 3 other words. Comparing how the word product is used in Data Analysts roles with Data Science roles, we see that in the latter it is more often used in combination with business words, such as (looking at the connecting word) managers, innovation, impact, development, etc. In Data Analyst positions, it is more connect to job specific 'day-to-day' activities, such as solutions, roadmaps, questions, and funnels.

__Fourth take-away:__

Data Analyst roles are more product focused when it comes to processes and analyzing the actual product. On the other hand, Data Science roles are more focused on the strategical aspect of products.

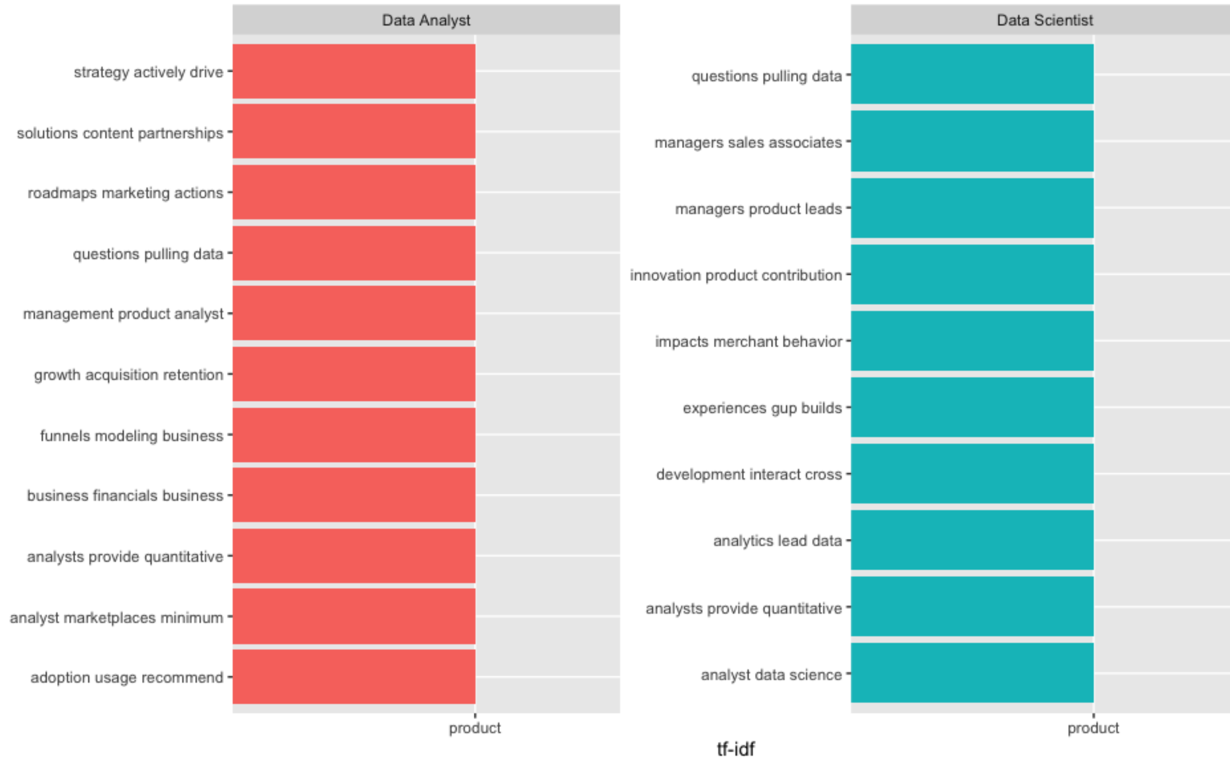

Figure 7.1: The word Product followed by 3 connected words

Experience

When it comes to experience, I looked at word combinations that end in experience. The Data Analyst as well as the Data Scientists roles require both technical and business experience. However, the insights that can be taken from the below graph (figure 7.2) are the keywords you can use in your application. If you applying for a Data Analyst job, use keywords such as “structure report dashboards”, “retail domains relevant”, or highlight your tensorflow scikit-learn experience. On the other hand, if you are planning on applying for a Data Science position, then its best to tailor your application with words such as: “statistical forecasting models”, “stakeholders preferred qualifications”, or show how you have experience with applied people management.

Fifth take-away:

Look for keywords in the description and tailor your application using those keywords. Don’t just focus on loose keywords, but also at keyword groups.

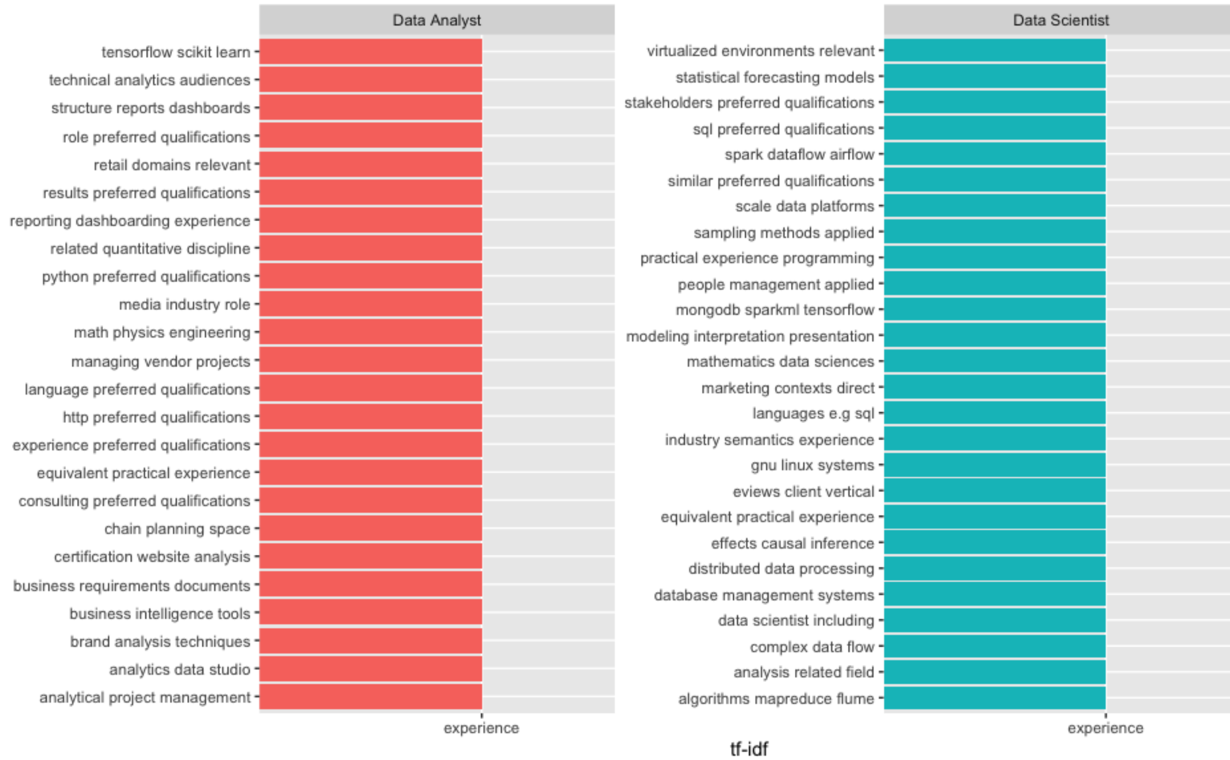

Figure 7.2: 3 connected words followed by Experience

Analysis

Lastly, let's have a look at words that are followed by the word "Analysis". Here, Data Scientist roles are more focused on professional (leadership) skills that are connected to analysis, such as word groups that start with responsibility, requirements and reliability. It really shows a deep analysis. Whereas analysis that is connected to Data Analyst jobs are more process related, such as data processing, and data gathering.

<u>Sixth take-away:</u>

Are you more of a process person or do you want to dive deep into the data and then even deeper? This also helps in deciding whether a Data Analysist or Data Scientist title fits you best. Data Science roles seems to be more strategical, whereas Data Analyst roles more process oriented.

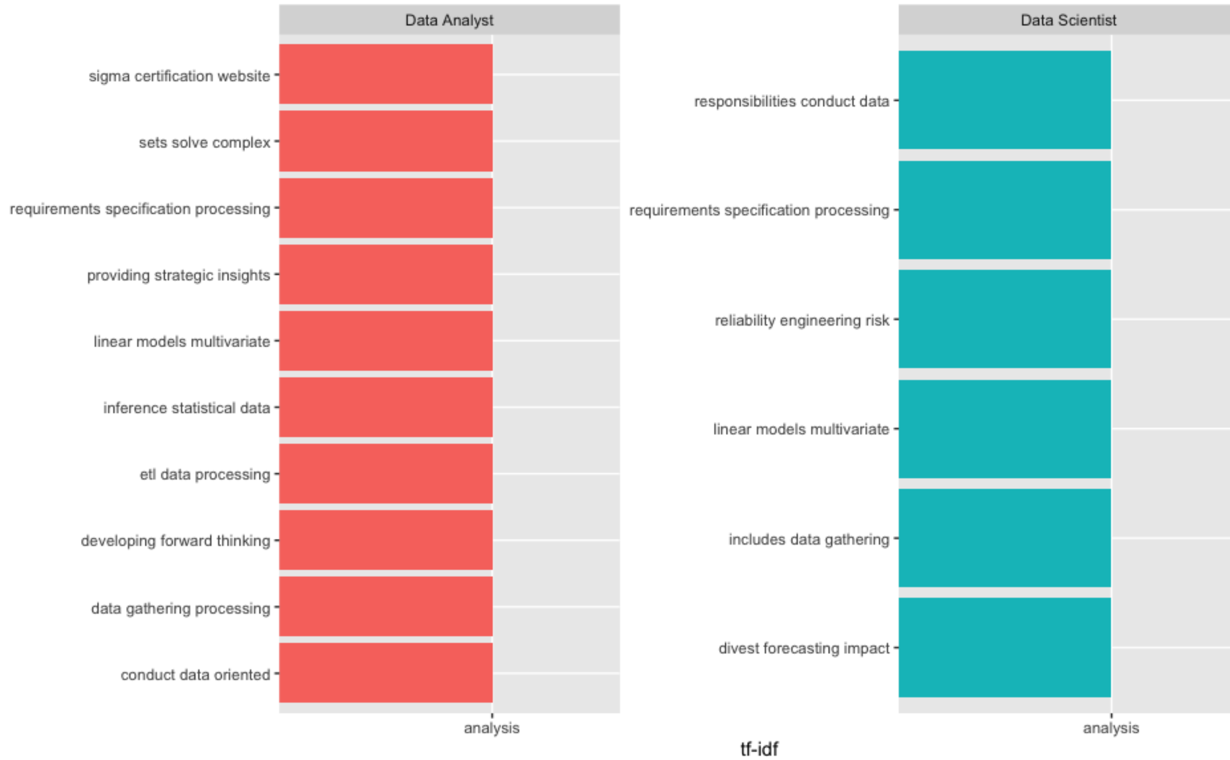

Figure 7.2: 3 connected words followed by Analysis

Conclusion

There are many ways to decide if a company and role are the right fit for you and how you can best increase your chances of getting that job! Here are some helpful tips to aid you in deciding if Google is the right fit for you and if so, how you can call yourself a Noogler in no-time!

- Don't just focus on the technical requirements for a job when deciding to apply.
- Before applying at any company look at their company culture as well.
- When applying for a job, really dive into the different types of skills a company is looking for.
- Data Analyst roles are more process oriented, where Data Science roles are more strategical.
- Look for keywords in the description and tailor your application using those keywords.
- Don't just focus on loose keywords, but also at keyword groups.

Appendices: Code & Output

Appendix 1: Importing

```
# Reading necessary documents
library(textreadr)
library(textshape)
library(dplyr)
library(stringr)
library(tidytext)
library(tidyr)
library(tidyverse)
library(tm)
library(reshape2)
library(wordcloud)
library(ggplot2)
library(igraph)
library(ggraph)

data(stop_words)

# Uploading files
setwd("/Users/jasmijnvanhulsen/Desktop/Classes/Module B/Text Analytics/Google_2")
nm <- list.files(path="/Users/jasmijnvanhulsen/Desktop/Classes/Module B/Text
Analytics/Google_2")

# Bind all the documents together
my_txt_text <- do.call(rbind, lapply(nm, function(x) paste(read_document(file=x), collapse = "
")))

# Creating a df with Google and Netflix to compare
job_all <- data_frame(title=c("Data Analyst","Data Scientist", "Netflix"), text=my_txt_text)

# Creating a df with just Google
job <- job_all[-3 ,]

# Creating a df with just Netflix
job_nf <- job_all[-c(1,2) , ]
```

Appendix 2: Tokenization

```
# Create tokens and anti_join stopwords - Google
job_tokens <- job %>%
  unnest_tokens(word, text) %>%
  anti_join(stop_words) %>%
  count(word, sort=TRUE)

job_tokens
```

```
> job_tokens
# A tibble: 1,678 x 2
   word               n
   <chr>          <int>
 1 data             279
 2 experience       199
 3 business         158
 4 google           155
 5 analysis          99
 6 team              92
 7 product           90
 8 qualifications    62
 9 teams             61
10 technical         59
# … with 1,668 more rows
```

```
# Create tokens with location information including Netflix
job_tokens_title_all <- job_all %>%
  unnest_tokens(word, text) %>%
  anti_join(stop_words) %>%
  count(title, word, sort=TRUE) %>%
  ungroup()

job_tokens_title_all
```

```
> job_tokens_title_all
# A tibble: 3,029 x 3
   title          word           n
   <chr>          <chr>      <int>
 1 Data Scientist data         173
 2 Data Analyst   data         106
 3 Data Scientist experience   104
 4 Data Analyst   business      97
 5 Data Analyst   experience    95
 6 Data Scientist google        82
 7 Data Analyst   google        73
 8 Netflix        data          70
 9 Data Scientist business      61
10 Data Analyst   team          52
# … with 3,019 more rows
```

```
# Creating graph with most frequent words - Google
freq_hist <- job_tokens %>%
  mutate(word=reorder(word, n)) %>%
  filter(n > 45) %>%
  ggplot(aes(word, n))+
  geom_col()+
  xlab(NULL)+
  coord_flip()
print(freq_hist)
```

Appendix 3: TF-IDF

```
# Create total words per article
total_words <- job_tokens_title_all %>%
  group_by(title) %>%
  summarize(total=sum(n))

# Join ai_tidy with total_words
title_words <- left_join(job_tokens_title_all, total_words)

title_words
```

```
> title_words
# A tibble: 3,029 x 4
   title          word            n total
   <chr>          <chr>       <int> <int>
 1 Data Scientist data          173  4302
 2 Data Analyst   data          106  4529
 3 Data Scientist experience    104  4302
 4 Data Analyst   business       97  4529
 5 Data Analyst   experience     95  4529
 6 Data Scientist google         82  4302
 7 Data Analyst   google         73  4529
 8 Netflix        data           70  1589
 9 Data Scientist business       61  4302
10 Data Analyst   team           52  4529
# … with 3,019 more rows
```

```
# Bind TF IDF
title_words <- title_words %>%
  bind_tf_idf(word, title, n)

title_words %>%
  arrange(desc(tf_idf))
```

```
+   arrange(desc(tf_idf))
# A tibble: 3,029 x 7
   title          word              n total      tf   idf  tf_idf
   <chr>          <chr>         <int> <int>   <dbl> <dbl>   <dbl>
 1 Netflix        netflix          23  1589 0.0145  1.10  0.0159
 2 Data Scientist google           82  4302 0.0191  0.405 0.00773
 3 Data Analyst   google           73  4529 0.0161  0.405 0.00654
 4 Netflix        status            9  1589 0.00566 1.10  0.00622
 5 Netflix        diversity         6  1589 0.00378 1.10  0.00415
 6 Netflix        entertainment     6  1589 0.00378 1.10  0.00415
 7 Netflix        million           6  1589 0.00378 1.10  0.00415
 8 Netflix        190               5  1589 0.00315 1.10  0.00346
 9 Netflix        joy               5  1589 0.00315 1.10  0.00346
10 Data Analyst   trust            13  4529 0.00287 1.10  0.00315
# … with 3,019 more rows
```

```
# Graphing most important words
title_words %>%
  arrange(desc(tf_idf)) %>%
  mutate(word=factor(word, levels=rev(unique(word)))) %>%
  group_by(title) %>%
  top_n(10) %>% # adjust for more tokens
  ungroup %>%
  ggplot(aes(word, tf_idf, fill=title))+
  geom_col(show.legend=FALSE)+
  labs(x=NULL, y="tf-idf")+
  facet_wrap(~title, ncol=2, scales="free")+
  coord_flip()
```

Appendix 4: Sentiment

```
# Sentiment Wordcloud NRC - Google
job_tokens %>%
  inner_join(get_sentiments("nrc")) %>% #lexicon_nrc
  count(word, sentiment, sort=TRUE) %>%
  acast(word ~sentiment, value.var="n", fill=0) %>%
  comparison.cloud(colors = c("grey20", "grey80"),
                   max.words=100, scale=c(1,0.1))

# Sentiment Wordcloud Bing - Google
job_tokens %>%
  inner_join(get_sentiments("bing")) %>% #lexicon_bing
  count(word, sentiment, sort=TRUE) %>%
  acast(word ~sentiment, value.var="n", fill=0) %>%
  comparison.cloud(colors = c("Red", "Dark Green"),
                   max.words=200, scale=c(1,1))
```

Appendix 5: Bigrams

```
Create bigrams with all words
job_bigrams <- job_all %>%
  group_by(title) %>%
  unnest_tokens(bigram, text, token = "ngrams", n=2) %>%
  count(bigram, sort = TRUE) %>%
  ungroup()

# Seperate words in bigrams
bigrams_separated <- job_bigrams %>%
  separate(bigram, c("word1", "word2"), sep = " ")
```

```
# Take out stopwords
bigrams_filtered <- bigrams_separated %>%
  filter(!word1 %in% stop_words$word) %>%
  filter(!word2 %in% stop_words$word)

# Creating the new bigram, "no-stop-words":
bigram_counts <- bigrams_filtered %>%
  count(word1, word2, sort = TRUE)

# See the bigrams
bigram_counts
```

```
> bigram_counts
# A tibble: 3,778 x 3
   word1         word2              n
   <chr>         <chr>          <int>
 1 business      decisions          3
 2 business      priorities         3
 3 collaboration skills             3
 4 communication skills             3
 5 cross         functional         3
 6 data          analysis           3
 7 data          driven             3
 8 data          engineering        3
 9 data          infrastructure     3
10 data          pipelines          3
# … with 3,768 more rows
```

```
# Unite words
bigram_united <- bigrams_filtered %>%
  unite(bigram, word1, word2, sep=" ")

# Create TF IDF for bigrams
bigram_tf_idf <- bigram_united %>%
  count(title, bigram) %>%
  bind_tf_idf(bigram, title, n) %>%
  arrange(desc(tf_idf))

# Create bigram relationships
bigram_graph <- bigram_counts %>%
  filter(n>2) %>% #less data, so lower n (maybe n=2)
  graph_from_data_frame()
```

```r
# Graph relationships
ggraph(bigram_graph, layout = "fr") + # fr for frequency
  geom_edge_link()+
  geom_node_point()+
  geom_node_text(aes(label=name), vjust =1, hjust=1)

# Filtering data
bigrams_filtered %>%
  filter(word1 == "data") %>%
  count(title, word2, sort = TRUE)

# Graphing data
bigrams_filtered %>%
  group_by(title) %>%
  filter(word1 == "data") %>%
  top_n(8) %>%
  ungroup %>%
  ggplot(aes(word2, word1, fill=title))+
  geom_col(show.legend=FALSE)+
  labs(x=NULL, y="tf-idf")+
  facet_wrap(~title, ncol=2, scales="free")+
  coord_flip()

# Filtering business
bigrams_filtered %>%
  filter(word1 == "business") %>%
  count(title, word2, sort = TRUE)

# Graphing business
bigrams_filtered %>%
  group_by(title) %>%
  filter(word1 == "business") %>%
  top_n(6) %>%
  ungroup %>%
  ggplot(aes(word2, word1, fill=title))+
  geom_col(show.legend=FALSE)+
  labs(x=NULL, y="tf-idf")+
  facet_wrap(~title, ncol=2, scales="free")+
  coord_flip()

# Filtering skills
bigrams_filtered %>%
  filter(word2 == "skills") %>%
  count(title, word1, sort = TRUE)
```

```
# Graphing skills
bigrams_filtered %>%
  group_by(title) %>%
  filter(word2 == "skills") %>%
  top_n(8) %>%
  ungroup %>%
  ggplot(aes(word1, word2, fill=title))+
  geom_col(show.legend=FALSE)+
  labs(x=NULL, y="tf-idf")+
  facet_wrap(~title, ncol=2, scales="free")+
  coord_flip()
```

Appendix 6: Quadrograms

```
# Create quadrogram with all words
job_quadrogram <- job %>%
  group_by(title) %>%
  unnest_tokens(quadrogram, text, token = "ngrams", n=4) %>%
  count(quadrogram, sort = TRUE) %>%
  ungroup()

# Seperate words in quadrogram
quadrogram_separated <- job_quadrogram %>%
  separate(quadrogram, c("word1", "word2", "word3", "word4"), sep = " ")

# Take out stopwords
quadrogram_filtered <- quadrogram_separated %>%
  filter(!word1 %in% stop_words$word) %>%
  filter(!word2 %in% stop_words$word) %>%
  filter(!word3 %in% stop_words$word) %>%
  filter(!word4 %in% stop_words$word)

# Creating the new quadrogram, "no-stop-words":
quadrogram_counts <- quadrogram_filtered %>%
  count(word1, word2, word3, word4, sort = TRUE)
```

```
# See the new quadrogram
quadrogram_counts
```

```
> quadrogram_counts
# A tibble: 1,303 x 5
   word1       word2      word3        word4         n
   <chr>       <chr>      <chr>        <chr>     <int>
 1 085         8          5            20            2
 2 118,000     bonus      equity       benefits      2
 3 advisors    support    customers    globally      2
 4 analyses    including  data         gathering     2
 5 analysis    stochastic models       sampling      2
 6 analysts    provide    quantitative support       2
 7 applying    advanced   analytical   methods       2
 8 articulating product   questions    pulling       2
 9 automate    reports    iteratively  build         2
10 bonus       equity     benefits     note          2
# … with 1,293 more rows
```

```
# Unite words
quadrogram_united <- quadrogram_filtered %>%
  unite(quadrogram, word1, word2, word3, word4, sep=" ") #we need to unite what we split in the previous section

# Create TF IDF for quadrogram
quadrogram_tf_idf <- quadrogram_united %>%
  count(title, quadrogram) %>%
  bind_tf_idf(quadrogram, title, n) %>%
  arrange(desc(tf_idf))

# Create quadrogram relationships
quadrogram_graph <- quadrogram_counts %>%
  filter(n>1) %>% #less data, so lower n (maybe n=2)
  graph_from_data_frame()

# Graph quadrogram relationships
ggraph(quadrogram_graph, layout = "fr") + # fr for frequency
  geom_edge_link()+
  geom_node_point()+
  geom_node_text(aes(label=name), vjust =1, hjust=1)
```

```
# Grouping first 3 words together, so I can compare to last word
quadrogram_un_last <- quadrogram_filtered %>%
  unite(quadrogram, word1, word2, word3, sep=" ")

# Grouping last 3 words together, so I can compare to fist word
quadrogram_un_first <- quadrogram_filtered %>%
  unite(quadrogram, word2, word3, word4, sep=" ")

# Check combinations with Product
quadrogram_un_first %>%
  group_by(title) %>%
  filter(word1 == "product") %>%
  top_n(5) %>% # adjust for more tokens
  ungroup %>%
  ggplot(aes(quadrogram, word1, fill=title))+
  geom_col(show.legend=FALSE)+
  labs(x=NULL, y="tf-idf")+
  facet_wrap(~title, ncol=2, scales="free")+
  coord_flip()

# Check combinations with Experience
quadrogram_un_last %>%
  group_by(title) %>%
  filter(word4 == "experience") %>%
  top_n(5) %>% # adjust for more tokens
  ungroup %>%
  ggplot(aes(quadrogram, word4, fill=title))+
  geom_col(show.legend=FALSE)+
  labs(x=NULL, y="tf-idf")+
  facet_wrap(~title, ncol=2, scales="free")+
  coord_flip()


# Check combinations with Analysis
quadrogram_un_last %>%
  group_by(title) %>%
  filter(word4 == "analysis") %>%
  top_n(5) %>% # adjust for more tokens
  ungroup %>%
  ggplot(aes(quadrogram, word4, fill=title))+
  geom_col(show.legend=FALSE)+
  labs(x=NULL, y="tf-idf")+
  facet_wrap(~title, ncol=2, scales="free")+
  coord_flip()
```

Appendix 7: Global Environment (final)

Global Environment

Data	
bigram_counts	3778 obs. of 3 variables
bigram_tf_idf	4221 obs. of 6 variables
bigram_united	4221 obs. of 3 variables
bigrams_filtered	4221 obs. of 4 variables
bigrams_separated	11936 obs. of 4 variables
freq_hist	List of 9
job	2 obs. of 2 variables
job_all	3 obs. of 2 variables
job_bigrams	11936 obs. of 3 variables
job_nf	1 obs. of 2 variables
job_quadrogram	12904 obs. of 3 variables
job_tokens	1678 obs. of 2 variables
job_tokens_title_all	3029 obs. of 3 variables
my_txt_text	chr [1:3, 1] "Data Analyst, Devices and Ser…
quadrogram_counts	1303 obs. of 5 variables
quadrogram_filtered	1364 obs. of 6 variables
quadrogram_separated	12904 obs. of 6 variables
quadrogram_tf_idf	1364 obs. of 6 variables
quadrogram_united	1364 obs. of 3 variables
stop_words	1149 obs. of 2 variables
title_words	3029 obs. of 7 variables
total_words	3 obs. of 2 variables
Values	
nm	chr [1:3] "Data Analyst.txt" "Data Scientist.…

References

Google. (2021). *Find your next job at Google*. Retrieved on https://careers.google.com

Netflix. (2021). *Netflix Jobs*. Retrieved on https://jobs.netflix.com/search