DATA 607 - Project 3 - Soft Skills

Authors:Zach Alexander, Erinda Budo, Steven Ellingson, John Kellogg, Misha Kollontai, Jose Mawyin

10/14/2019

Scraping Indeed for Soft Skills Relevant to a Data Scientist

Introduction

In the field of Data Science, the mastery of "Hard Skills" (e.g. programming, math, analysis, etc) is seen as a necessity when joining the ranks of data scientists and working within the data community. Most new data scientists have studied these skills as part of the curriculum of universities and training centers where people learn about the growing field. All data scientists are expected to reach proficiency in these skills in order to be ready to meet job requirements; this is only half of what is needed to exceed in the market.

There is also a set of soft skills needed to effectively interact with other people who are not in the data science field. They make up a large part of the skillset that employees are looking for. Soft skills build better data science teams as well as present their findings to the business at large. One can not be proficient without the other.

Our group took a dive into what soft skills are most highly sought after/are most in demand.

This project contains the following Sections:

1. Data Collection
2. Data Analysis
3. Team Analysis
4. Conclusions
5. Potential Next Steps

1. Data Collection

To centralize the focus, a slack channel was created and communications started on what we would gather. We chose a method of all team members using the job search engine link to gather job descriptions focusing on "Data Science Jobs" with a geotarget based on a 15-mile radius around New York. We scraped the data from indeed.com, then manipulated and tidied the data into a data frame of 4 different columns, including "job_title", "company_name", "job_location", and "job_description". In all, the team was able to pull in jobs data from 1,412 different listings for Data Science positions. We also worked through a fair amount of string parsing, using various techniques such as a Textrank algorithm utilized by Google's Pagerank, Rapid Automatic Keyword Extraction (RAKE), Dependency parsing, and noun extraction. Although these techniques showed mild progress, there was unfortunately too much noise in the job descriptions, and it was hard to narrow down the extractions to soft skills. Nonetheless, we were set up really well with a large data frame of Data Science jobs data, with a job descriptions vector that we could use for further analysis.

1.1 Data Collection process

- Launch Indeed.com and run a search focusing the query to Data Scientist Jobs based in a 15 mile radius around New York City.
- The code below scraps through link then sorts the info from job postings into 4 different columns: job_title, company_name, job_location, job_description.

For the purpose of this paper and so none of the main values change during the life of the paper, The following is not provided as R scripting in a code chunk. It is the method utilized by the team to gain the initial data. The result of this code will be loaded straight from the CSV initially generated by the team.

```
page_result_start <- 10 # starting page
page_result_end <- 30 # last page results
page_results <- seq(from = page_result_start, to = page_result_end, by = 10)

full_df_scrape <- data.frame()
for(i in seq_along(page_results)) {

  first_page_url <- "https://www.indeed.com/jobs?q=data+scientist&l=new+york&radius=15"
  url <- paste0(first_page_url, "&start=", page_results[i])
  page <- xml2::read_html(url)
  # Sys.sleep pauses R for two seconds before it resumes
  # Putting it there avoids error messages such as "Error in
  # open.connection(con, "rb") : Timeout was reached"
  Sys.sleep(3)

  #get the job title
  job_title <- page %>%
    rvest::html_nodes("div") %>%
    rvest::html_nodes(xpath = '//a[@data-tn-element = "jobTitle"]') %>%
    rvest::html_attr("title")

  #get the company name
  company_name <- page %>%
  rvest::html_nodes(".company") %>%
  rvest::html_text() %>%
```

```
  stringi::stri_trim_both()


  #get job location
  job_location <- page %>%
  rvest::html_nodes(".location") %>%
  rvest::html_text()

  # get links
  links <- page %>%
  rvest::html_nodes("div") %>%
  rvest::html_nodes(xpath = '//*[@data-tn-element="jobTitle"]') %>%
  rvest::html_attr("href")

  job_description <- c()
  for(i in seq_along(links)) {

    url <- paste0("https://www.indeed.com/", links[i])
    page <- xml2::read_html(url)

    job_description[[i]] <- page %>%
  rvest::html_nodes("span")  %>%
  rvest::html_nodes(xpath = '//*[@class="jobsearch-JobComponent-description
                    icl-u-xs-mt--md  "]') %>%
  rvest::html_text() %>%
  stringi::stri_trim_both()
  }
  df <- data.frame(job_title, company_name, job_location, job_description)
  full_df_raw <- rbind(full_df_scrape, df)
}
```

- Export the raw scrape file to CSV since as we have a data frame containing enough information for later analysis and load into a SQL DB.
- When complete, the results are written to a csv

1.2 Upload into SQL Cloud

Using the Indeedscrap.csv an initial Database was created to enable the rest of the team to start work on a centralized data frame. Utilizing Google Cloud MYSQL, we created a schema and subsequent databases.

Process notes:

Due to restrictions in Google MySQL and to ensure the team was able to work on the cleanest data possible, a second DQS schema was required on a local machine. Using the lower environment instance, the CSV could load with no restrictions then be exported/imported into a clean database in the production Project 3 DB.

Since the data was scraped from a live website, loading into a SQL DB had to be done in two sections, the data (job_title, company_name, job_location) and the text (job_description). The text load had to take separate methods due to the delimiters generally used were present in the data already. The two tables were then joined into one db. The single DB was exported and imported into the cloud schema.

Example scripts include:

LOAD DATA infile 'C:/Users/x/Documents/CSPS - Homework/607/IndeedScrap.csv' INTO TABLE project3 FIELDS TERMINATED BY ',' ENCLOSED BY '"' LINES TERMINATED BY '̊' IGNORE 1 ROWS ;

INSERT INTO project3_all (ID, job_title, company_name, job_location, job_description) Select p.ID, p.job_title, p.company_name, p.job_location, t.job_description FROM project3 p JOIN project_text t on p.ID = t.ID;

```
# Using our centralized database, I pulled in data from our
# normalized tables and did some frequency matching
mydb = dbConnect(MySQL(), user='admin', password='project3', host='34.68.107.105', port = 3306)
full_df <- dbGetQuery(mydb, 'SELECT * FROM project3.indeed')
```

1.3 Soft Skills list

In order to separate "soft" from "hard" skills, we googled "list of soft skills" and built a list of skills from the top results.

First list: link

```
# Get list from "developgoodhabits.com"
url <- "https://www.developgoodhabits.com/soft-skills-list/"
page <- xml2::read_html(url)
skills <-page %>%
  html_nodes('body') %>%
  html_nodes( 'div') %>%
  html_nodes(".wrp.cnt")%>%
  html_nodes( 'h3') %>%
  html_text() %>%
  str_replace_all('^\\d+\\.? *','')
skills
```

```
##   [1] "Verbal Communication"
##   [2] "Non-Verbal Communication"
##   [3] "Visual Communication"
##   [4] "Written Communication"
##   [5] "Active Listening"
##   [6] "Clarity"
##   [7] "Confidence"
##   [8] "Interviewing"
##   [9] "Negotiation"
##  [10] "Personal Branding"
##  [11] "Persuasion"
##  [12] "Presentation Skills"
##  [13] "Public Speaking"
##  [14] "Storytelling"
##  [15] "Diplomacy"
##  [16] "Empathy"
##  [17] "Friendliness"
##  [18] "Humor"
##  [19] "Networking"
```

```
##  [20] "Patience"
##  [21] "Positive Reinforcement"
##  [22] "Sensitivity"
##  [23] "Tolerance"
##  [24] "Analysis"
##  [25] "Artistic Sense"
##  [26] "Brainstorming"
##  [27] "Design"
##  [28] "Design Sense"
##  [29] "Divergent Thinking"
##  [30] "Experimenting"
##  [31] "Imagination"
##  [32] "Innovation"
##  [33] "Insight"
##  [34] "Inspiration"
##  [35] "Lateral Thinking"
##  [36] "Logical Reasoning"
##  [37] "Mind Mapping"
##  [38] "Observation"
##  [39] "Persistence"
##  [40] "Questioning"
##  [41] "Reframing"
##  [42] "Troubleshooting"
##  [43] "People Management"
##  [44] "Project Management"
##  [45] "Remote Team Management"
##  [46] "Talent Management"
##  [47] "Virtual Team Management"
##  [48] "Meeting Management"
##  [49] "Agility"
##  [50] "Coaching"
##  [51] "Conflict or Dispute Resolution"
##  [52] "Cultural Intelligence"
##  [53] "Deal-Making"
##  [54] "Decision-Making"
##  [55] "Delegation"
##  [56] "Facilitating"
##  [57] "Give Clear Feedback"
##  [58] "Managing Difficult Conversations"
##  [59] "Mentoring"
##  [60] "Strategic Planning"
##  [61] "Supervising"
##  [62] "Team-Building"
##  [63] "Versatility"
##  [64] "Authenticity"
##  [65] "Encouraging"
##  [66] "Generosity"
##  [67] "Humility"
##  [68] "Inspiring"
##  [69] "Selflessness"
##  [70] "Attentive"
##  [71] "Business Ethics"
##  [72] "Calm"
##  [73] "Commitment"
```

```
##  [74] "Competitiveness"
##  [75] "Curiosity"
##  [76] "Dependability"
##  [77] "Discipline"
##  [78] "Emotion Management"
##  [79] "Highly Organized"
##  [80] "Independence"
##  [81] "Initiative"
##  [82] "Integrity"
##  [83] "Motivated"
##  [84] "Open-Minded"
##  [85] "Optimistic"
##  [86] "Perseverant"
##  [87] "Professional"
##  [88] "Punctual"
##  [89] "Reliable"
##  [90] "Resilient"
##  [91] "Responsible"
##  [92] "Results-Oriented"
##  [93] "Taking Criticism"
##  [94] "Tolerance of Change and Uncertainty"
##  [95] "Trainable"
##  [96] "Accept Feedback"
##  [97] "Collaborative"
##  [98] "Cooperation"
##  [99] "Coordination"
## [100] "Deal with Difficult Situations"
## [101] "Disability Awareness"
## [102] "Diversity Awareness"
## [103] "Emotional Intelligence"
## [104] "Idea Exchange"
## [105] "Influential"
## [106] "Intercultural Competence"
## [107] "Interpersonal Relationships Skills"
## [108] "Mediation"
## [109] "Office Politics Management"
## [110] "Personality Conflicts Management"
## [111] "Respectfulness"
## [112] "Sales Skills"
## [113] "Self-Awareness"
## [114] "Social Skills"
## [115] "Acuity"
## [116] "Allocating Resources"
## [117] "Coping"
## [118] "Critical Observation"
## [119] "Focus"
## [120] "Goal-Setting"
## [121] "Introspection"
## [122] "Memory"
## [123] "Organization"
## [124] "Personal Time Management"
## [125] "Planning"
## [126] "Prioritization"
## [127] "Recall"
```

```
## [128] "Scheduling"
## [129] "Sense of Urgency"
## [130] "Streamlining"
## [131] "Stress Management"
## [132] "Task Planning"
## [133] "Task Tracking"
## [134] "Time Awareness"
## [135] "Work-Life Balance"
```

1.3.1 Second list

We get a second list from [link] (https://training.simplicable.com)

```
url2 <- "https://training.simplicable.com/training/new/87-soft-skills"
page2 <- xml2::read_html(url2)
skills2.temp <- html_nodes(page2,'.blogy')[1] %>%
  html_text() %>%
  str_split(fixed('\r')) %>%
  unlist()
skills2 <- skills2.temp[str_detect(skills2.temp,'\\d\\.')] %>%
str_replace_all('^.*\\d+\\.? *','')
#add to other list, and only pull unique values
skills <-sort(unique(str_trim(c(skills,skills2))))
skills
```

```
##   [1] "Accept Feedback"
##   [2] "Active Listening"
##   [3] "Acuity"
##   [4] "Adaptability"
##   [5] "Agility"
##   [6] "Allocating Resources"
##   [7] "Analysis"
##   [8] "Artistic Sense"
##   [9] "Assertiveness"
##  [10] "Attentive"
##  [11] "Authenticity"
##  [12] "Body Language"
##  [13] "Brainstorming"
##  [14] "Business Ethics"
##  [15] "Business Etiquette"
##  [16] "Business Trend Awareness"
##  [17] "Calm"
##  [18] "Clarity"
##  [19] "Coaching"
##  [20] "Collaborating"
##  [21] "Collaborative"
##  [22] "Commitment"
##  [23] "Competitiveness"
##  [24] "Confidence"
##  [25] "Conflict or Dispute Resolution"
##  [26] "Conflict Resolution"
```

```
##  [27] "Cooperation"
##  [28] "Coordination"
##  [29] "Coping"
##  [30] "Crisis Management"
##  [31] "Critical Observation"
##  [32] "Critical Thinking"
##  [33] "Cultural Intelligence"
##  [34] "Curiosity"
##  [35] "Customer Service"
##  [36] "Deal-Making"
##  [37] "Deal with Difficult Situations"
##  [38] "Dealing with Difficult People"
##  [39] "Decision-Making"
##  [40] "Decision Making"
##  [41] "Delegation"
##  [42] "Dependability"
##  [43] "Design"
##  [44] "Design Sense"
##  [45] "Diplomacy"
##  [46] "Disability Awareness"
##  [47] "Discipline"
##  [48] "Dispute Resolution"
##  [49] "Divergent Thinking"
##  [50] "Diversity Awareness"
##  [51] "Emotion Management"
##  [52] "Emotional Intelligence"
##  [53] "Empathy"
##  [54] "Encouraging"
##  [55] "Enthusiasm"
##  [56] "Entrepreneurial Thinking"
##  [57] "Experimenting"
##  [58] "Facilitating"
##  [59] "Facilitation"
##  [60] "Focus"
##  [61] "Friendliness"
##  [62] "Generosity"
##  [63] "Give Clear Feedback"
##  [64] "Giving Feedback"
##  [65] "Goal-Setting"
##  [66] "Highly Organized"
##  [67] "Humility"
##  [68] "Humor"
##  [69] "Idea Exchange"
##  [70] "Imagination"
##  [71] "Independence"
##  [72] "Influential"
##  [73] "Initiative"
##  [74] "Innovation"
##  [75] "Insight"
##  [76] "Inspiration"
##  [77] "Inspiring"
##  [78] "Integrity"
##  [79] "Intercultural Competence"
##  [80] "Interpersonal Relationships"
```

```
##  [81] "Interpersonal Relationships Skills"
##  [82] "Interviewing"
##  [83] "Introspection"
##  [84] "Knowledge Management"
##  [85] "Lateral Thinking"
##  [86] "Listening"
##  [87] "Logical Reasoning"
##  [88] "Manager Management"
##  [89] "Managing"
##  [90] "Managing Difficult Conversations"
##  [91] "Managing Remote Teams"
##  [92] "Managing Virtual Teams"
##  [93] "Mediation"
##  [94] "Meeting Management"
##  [95] "Memory"
##  [96] "Mentoring"
##  [97] "Mind Mapping"
##  [98] "Motivated"
##  [99] "Motivating"
## [100] "Negotiation"
## [101] "Networking"
## [102] "Non-Verbal Communication"
## [103] "Observation"
## [104] "Office Politics"
## [105] "Office Politics Management"
## [106] "Open-Minded"
## [107] "Optimistic"
## [108] "Organization"
## [109] "Patience"
## [110] "People Management"
## [111] "Performance Management"
## [112] "Perseverant"
## [113] "Persistence"
## [114] "Personal Branding"
## [115] "Personal Time Management"
## [116] "Personality Conflicts Management"
## [117] "Persuasion"
## [118] "Physical Communication"
## [119] "Planning"
## [120] "Positive Reinforcement"
## [121] "Presentation Skills"
## [122] "Prioritization"
## [123] "Problem Solving"
## [124] "Process Improvement"
## [125] "Professional"
## [126] "Project Management"
## [127] "Public Speaking"
## [128] "Punctual"
## [129] "Questioning"
## [130] "Quick-wittedness"
## [131] "Recall"
## [132] "Reframing"
## [133] "Reliable"
## [134] "Remote Team Management"
```

```
## [135] "Research"
## [136] "Resilience"
## [137] "Resilient"
## [138] "Respectfulness"
## [139] "Responsible"
## [140] "Results-Oriented"
## [141] "Sales Skills"
## [142] "Scheduling"
## [143] "Self-Awareness"
## [144] "Self Assessment"
## [145] "Self Awareness"
## [146] "Self Confidence"
## [147] "Self Leadership"
## [148] "Selflessness"
## [149] "Selling"
## [150] "Sense of Urgency"
## [151] "Sensitivity"
## [152] "Social Skills"
## [153] "Storytelling"
## [154] "Strategic Planning"
## [155] "Streamlining"
## [156] "Stress Management"
## [157] "Supervising"
## [158] "Taking Criticism"
## [159] "Talent Management"
## [160] "Task Planning"
## [161] "Task Tracking"
## [162] "Team-Building"
## [163] "Team Building"
## [164] "Technology Savvy"
## [165] "Technology Trend Awareness"
## [166] "Time Awareness"
## [167] "Time Management"
## [168] "Tolerance"
## [169] "Tolerance of Change and Uncertainty"
## [170] "Train the Trainer"
## [171] "Trainable"
## [172] "Training"
## [173] "Troubleshooting"
## [174] "Verbal Communication"
## [175] "Versatility"
## [176] "Virtual Team Management"
## [177] "Visual Communication"
## [178] "Work-Life Balance"
## [179] "Writing"
## [180] "Writing Reports and Proposals"
## [181] "Written Communication"
```

```
write.csv(skills, 'generic_skill_list.csv')
```

This data was loaded into a SQL DB for everyone to utilize for analysis.

1.3.2 Third Skill list

We pulled a final list of *hard* skills for data scientists, so we could use this list to compare some of the most common hard skills to the soft skills we've been working with. Some of the skills were added into our larger soft-skills list since they qualified more as soft skills than hard skills. We parsed through the list and made these distinctions with group agreement before uploading the hard skill list into the database as a separate normalized table.

```
# Website with a good list of hard skills for data scientists
url_1 <- "https://towardsdatascience.com/top-skills-every-data-scientist-needs-to-master-5aba4293b88"
page_1 <- xml2::read_html(url_1)
# after reading in this html data,  manipulate it and extracted the relevant skills
skills_1 <- page_1 %>%
  rvest::html_nodes("div") %>%
  rvest::html_nodes("strong") %>%
  rvest::html_text()
skills_1 <- skills_1[c(2:10)]
# Second website with a long list of hard skills for data scientists
url_2 <- "https://www.thebalancecareers.com/list-of-data-scientist-skills-2062381"
page_2 <- xml2::read_html(url_2)
# Read HTML and extract relevant skills
skills_2 <- page_2 %>%
  rvest::html_nodes('div') %>%
  rvest::html_nodes('ul') %>%
  rvest::html_nodes('li') %>%
  rvest::html_text()
skills_2 <- skills_2[c(7:96)]
# Append into one long list
skills_fnl <- append(skills_1, skills_2)
# Convert to data and move to lowercase for easier analysis later on
skills_df <- data.frame(matrix(NA, nrow = length(skills_fnl), ncol = 2))
skills_df <- skills_df %>%
  mutate(X1 = nrow(1:length(skills_fnl)),
         X2 = skills_fnl) %>%
  mutate(X1 = seq.int(nrow(skills_df))) %>%
  select(X2) %>%
  mutate(X2 = tolower(X2)) %>%
  distinct()
# Save to csv for later upload into database
write.csv(skills_df, file = "more_skills.csv")
```

1.4 Nurse Skill list

The next section, we pulled data on a completely unrelated field, Nursing. This data was be utilized later as a method to identify skills intrinsically data driven. As above, this section is not in a chunk as the data on the job site is constantly changing. The results of this section were added to the SQL database as *nurse_indeed*

```
page_result_start <- 10 # starting page
page_result_end <- 400 # last page results
page_results <- seq(from = page_result_start, to = page_result_end, by = 10)

nurse_full_df <- data.frame()
```

```
for(i in seq_along(page_results)) {

  first_page_url <- "https://www.indeed.com/jobs?q=nurse&l=new+york"
  url <- paste0(first_page_url, "&start=", page_results[i])
  page <- xml2::read_html(url)
  # Sys.sleep pauses R for two seconds before it resumes
  # Putting it there avoids error messages such as "Error in
  # open.connection(con, "rb") : Timeout was reached"
  Sys.sleep(2)

  #get the job title
  job_title <- page %>%
    rvest::html_nodes("div") %>%
    rvest::html_nodes(xpath = '//a[@data-tn-element = "jobTitle"]') %>%
    rvest::html_attr("title")

  #get the company name
  company_name <- page %>%
  rvest::html_nodes(".company") %>%
  rvest::html_text() %>%
  stringi::stri_trim_both()


  #get job location
  job_location <- page %>%
  rvest::html_nodes(".location") %>%
  rvest::html_text()

  # get links
  links <- page %>%
  rvest::html_nodes("div") %>%
  rvest::html_nodes(xpath = '//*[@data-tn-element="jobTitle"]') %>%
  rvest::html_attr("href")

  job_description <- c()
  for(i in seq_along(links)) {

    url <- paste0("https://www.indeed.com/", links[i])
    page <- xml2::read_html(url)

    job_description[[i]] <- page %>%
  rvest::html_nodes("span")  %>%
  rvest::html_nodes(xpath = '//*[@class="jobsearch-JobComponent-description
                    icl-u-xs-mt--md  "]') %>%
  rvest::html_text() %>%
  stringi::stri_trim_both()
  }
  df <- data.frame(job_title, company_name, job_location, job_description)
  nurse_full_df_raw <- rbind(nurse_full_df, df)
}
```

```
mydb = dbConnect(MySQL(), user='admin', password='project3', host='34.68.107.105', port = 3306)
nurse_full_df <- dbGetQuery(mydb, 'SELECT * FROM project3.nurse_indeed')
```

2. Data Analysis

With the data in one place, we started breaking down what we had; manipulating it to find trends and key words. Various methods were utilized to ensure we have a comprehensive list.

Example: utilizing the direct SQL connection to find frequency of specific words, more on this later.

```
soft_skills <- dbGetQuery(mydb, 'SELECT * FROM project3.skills_text')
soft_skills <- soft_skills %>%
  mutate(Text = tolower(Text)) %>%
  mutate(Text = str_sub(Text, end = -2L))
full_df$job_description <- iconv(full_df$job_description,"WINDOWS-1252","UTF-8")
full_df <- full_df %>%
  mutate(job_description = tolower(job_description))
final_df <- data.frame(matrix(NA, nrow = length(soft_skills$Text), ncol = 2))
rows_soft_skils <- nrow(soft_skills)
for (i in 1:rows_soft_skils) {
  make_string <- soft_skills[i,2] %>% as.String()
  frequency <- stri_count_regex(full_df$job_description, make_string) %>%
    as.data.frame() %>%
    colSums()
  final_df[i,1] <- soft_skills[i,2]
  final_df[i,2] <- frequency
}
lapply(dbListConnections( dbDriver( drv = "MySQL")), dbDisconnect)
```

```
## [[1]]
## [1] TRUE
##
## [[2]]
## [1] TRUE
```

2.1 Using UDPIPE library with Indeed Data Scientist Data**

For the following analysis we used the the UDPIPE library to break up job descriptions into text units that we can more easily analyze. UDPIPE is described as "natural language processing toolkit provides language-agnostic 'tokenization', 'parts of speech tagging', 'lemmatization' and 'dependency parsing' of raw text."

```
## Downloading udpipe model from https://raw.githubusercontent.com/jwijffels/udpipe.models.ud.2.4/maste
```

```
## Visit https://github.com/jwijffels/udpipe.models.ud.2.4 for model license details
```

```
##  [1] "doc_id"        "paragraph_id"  "sentence_id"   "sentence"
##  [5] "token_id"      "token"         "lemma"         "upos"
##  [9] "xpos"          "feats"         "head_token_id" "dep_rel"
## [13] "deps"          "misc"
```

UDPIPE sentimentalizes the input text into different components. See below the growth of elements between our job descriptions data frame and the UDPIPE output.

```
## [1] 1412    5
```

```
## [1] 850786     14
```

2.2 What did not work as planned

The following three techniques (Google Textrank, Rapid Automatic Keyword Extraction (RAKE), Dependency Parsing) were used to extract the most common phrases that appeared in the Description field of Data Scientist Jobs. The idea was to find phrases common in the descriptions and based on the frequency of these phrases determine the importance as a skill.

The results as shown below were too broad to be able extract "Skills". However, the results indicate the frequency of key phrases that commonly appear in the job descriptions.

2.2.1 Textrank (word network ordered by Google Pagerank)

"Textrank is an algorithm implemented in the textrank R package. The algorithm allows to summarize text and as well allows to extract keywords. This is done by constructing a word network by looking if words are following one another. On top of that network the 'Google Pagerank' algorithm is applied to extract relevant words after which relevant words which are following one another are combined to get keywords."

```
## [1] 1349    3
```

```
##                     keyword ngram freq
## 31           data scientist     2 1100
## 40         computer science     2  915
## 42             data science     2  899
## 58            machine learn     2  692
## 68                  to work     2  635
## 70               ability to     2  623
## 78                    to be     2  588
## 97                 new york     2  520
## 109                to help     2  462
## 122         learn technique     2  429
## 128 machine learn technique     3  423
## 141                 able to     2  387
## 151           data analysis     2  369
## 152         national origin     2  368
## 178         experience work     2  325
## 179                be focus     2  324
## 193     communication skill     2  305
## 199             at deloitte     2  298
## 201      sexual orientation     2  292
## 204                to learn     2  287
```

2.2.2 Extracting Keywords using Rapid Automatic Keyword Extraction (RAKE)

"RAKE which is an acronym for Rapid Automatic Keyword Extraction. It looks for keywords by looking to a contiguous sequence of words which do not contain irrelevant words. Namely by:

1. calculating a score for each word which is part of any candidate keyword, this is done by among the words of the candidate keywords, the algorithm looks how many times each word is occurring and how many times it co-occurs with other words each word gets a score which is the ratio of the word degree (how many times it co-occurs with other words) to the word frequency
2. a RAKE score for the full candidate keyword is calculated by summing up the scores of each of the words which define the candidate keyword."

```
##                                     keyword ngram freq     rake
## 12                      partial least squares     3   75 6.025316
## 15              maximum likelihood estimates     3   75 5.937739
## 20     senior executive teamsqualifications     3   56 5.821036
## 21       microsoft excelrms powerpointlinux     3   74 5.816327
## 26                          week sprint cycles     3   78 5.673386
## 27  operationsadvertising campaign ideation     3   70 5.647186
## 30                trial protocol methodology     3   75 5.585481
## 63                  san franciscoyour skills     3   77 5.280285
## 67      non-healthcare data setsresponsible     3   74 5.256047
## 88                              new york city     3  100 5.151103
## 95             engagement campaign ideation     3   72 5.123881
## 106              persistent problem solvers     3   70 5.071512
## 109                digital asset management     3   70 5.069762
## 124              artificial neural networks     3   61 5.010637
## 137                database access packages     3   74 4.975339
## 143              equal opportunity employer     3  248 4.957026
## 144    measurable business results.eskalera     3   77 4.956919
## 170          strong performance cultureteam     3   74 4.888829
## 186              edge algorithmic solutions     3   77 4.844797
## 202                       complex data issue     3   56 4.808256
```

2.2.3 Using dependency parsing output to get the nominal subject and the adjective of it

"Dependency Parsing: When you executed the annotation using udpipe, the dep_rel field indicates how words are related to one another. A token is related to the parent using token_id and head_token_id. The dep_rel field indicates how words are linked to one another. The type of relations are defined at http://universaldependencies.org/u/dep/index.html. For this exercise we are going to take the words which have as dependency relation nsubj indicating the nominal subject and we are adding to that the adjective which is changing the nominal subject.

In this way we can combine what are people talking about with the adjective they use when they talk about the subject."

```
##                      key freq  freq_pct
## 1              proud pwc   80 5.5671538
## 2  successful individual   78 5.4279749
## 3      centic capability   78 5.4279749
## 4            centic team   78 5.4279749
## 5    suitable technology   78 5.4279749
```

```
## 6          healthy people   76 5.2887961
## 7               able data   72 5.0104384
## 8                 ny york   70 4.8712596
## 9     responsible person   57 3.9665971
## 10        able candidate   17 1.1830202
## 11      responsible team   15 1.0438413
## 12 responsible scientist   13 0.9046625
## 13     prefer experience   11 0.7654836
## 14    committed facebook   11 0.7654836
## 15          committed ibm   10 0.6958942
## 16  desirable experience    9 0.6263048
## 17 available information    8 0.5567154
## 18  responsible position    8 0.5567154
## 19    available position    7 0.4871260
## 20      responsible role    6 0.4175365
```

3. Analysis findings from the team

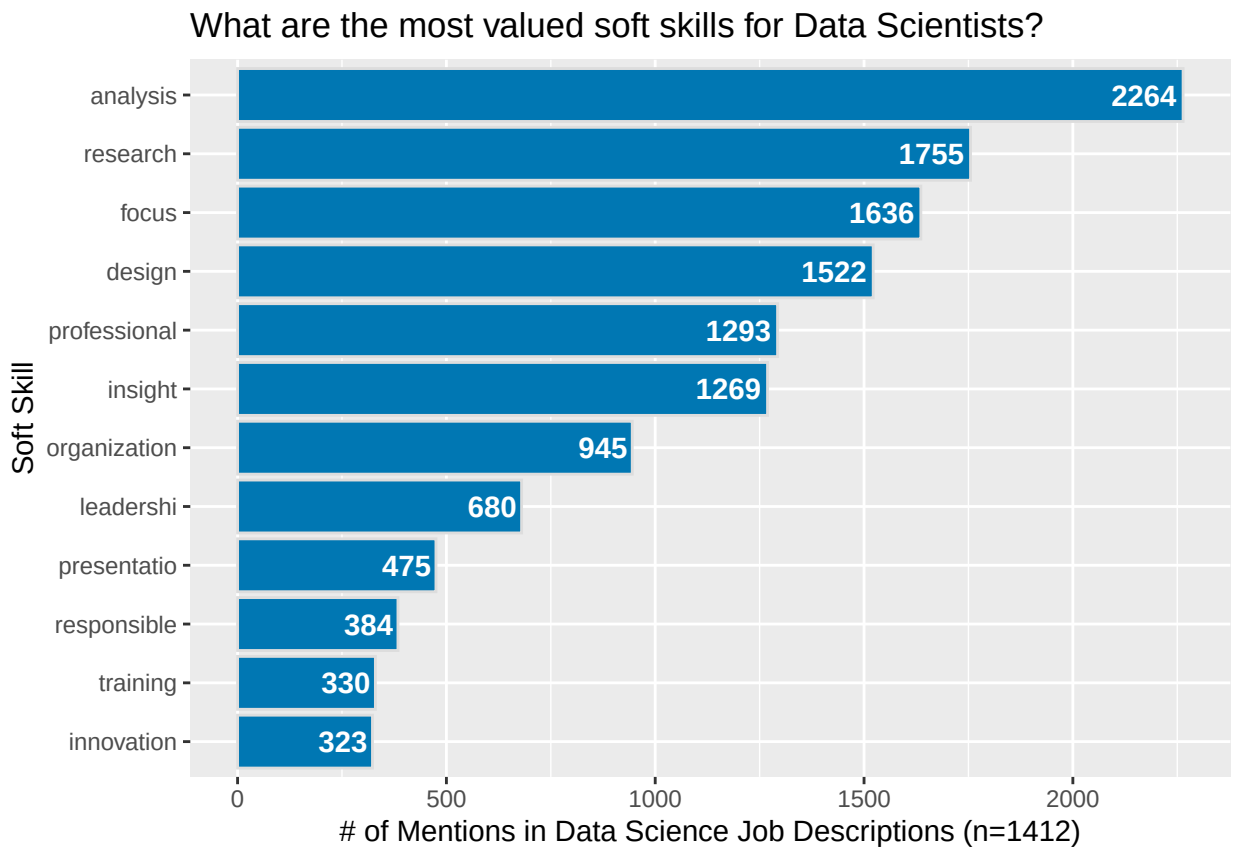
3.1 What are the most valued soft skills for Data Scientists?

```
final_df <- final_df %>%
  rename("Soft Skill" = X1,
         "Frequency" = X2)
final_df <- final_df %>%
  arrange(-Frequency)
kable(head(final_df, n = 10L), align = rep('c', 2)) %>%
  kable_styling(bootstrap_options = c("striped"), full_width = F)
```

Soft Skill	Frequency
analysis	2264
research	1755
focus	1636
design	1522
professional	1293
insight	1269
organization	945
leadershi	680
presentatio	475
responsible	384

```
final_df_sub <- final_df[c(1:12),]

ggplot(final_df_sub, aes(x=reorder(`Soft Skill`, Frequency),y=Frequency)) +
  geom_bar(position="dodge",stat="identity", fill = "#0077b3", color = "#dddddd") +
  ylab('# of Mentions in Data Science Job Descriptions (n=1412)') +
  xlab('Soft Skill') +
  coord_flip() +
  ggtitle("What are the most valued soft skills for Data Scientists?") +
  geom_text(aes(label=Frequency), vjust=0.5, hjust=1.10, position = position_dodge(width = 0.9),
            color="white", fontface="bold")
```



3.2 Comparisons across the Data Science and Nurse Job Descriptions (calculating proportions)

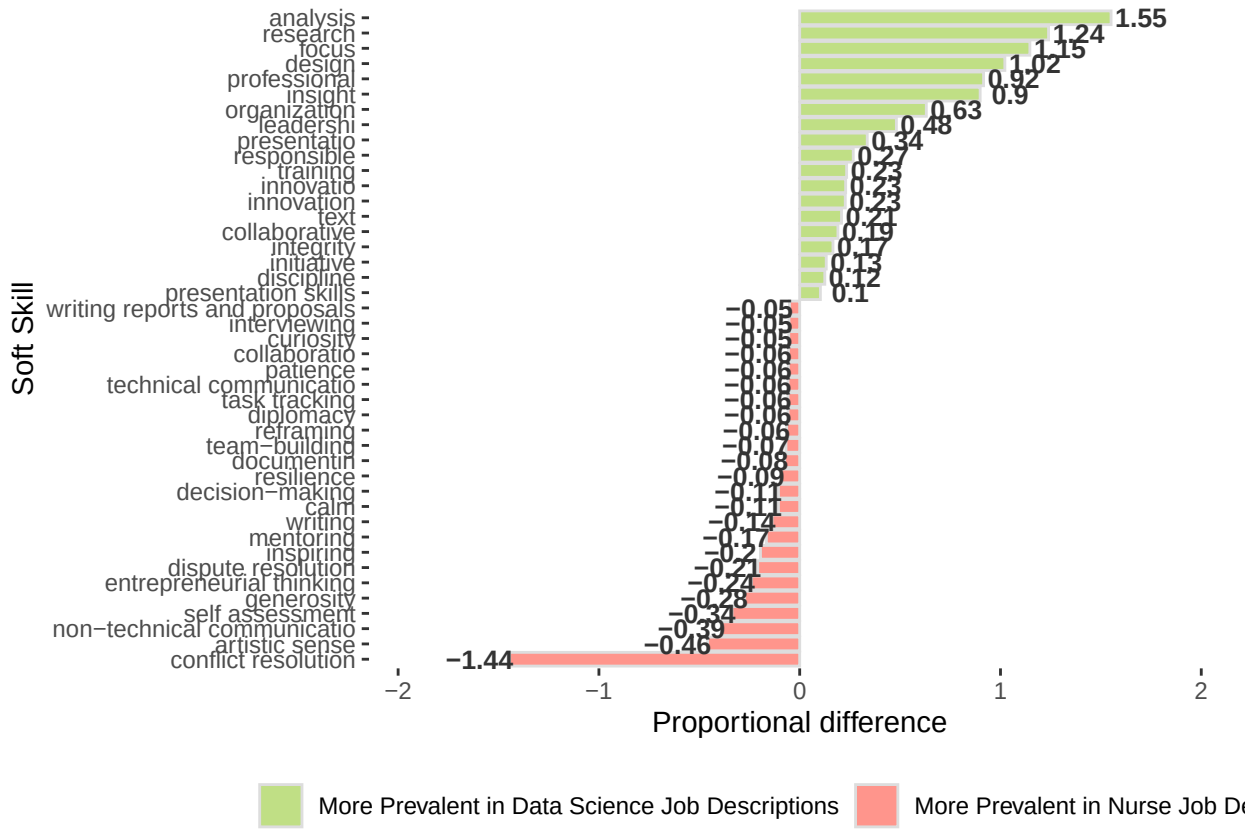
Once we had confirmed many of our frequencies, Misha took the counts from both the Data Science table and the Nurse table and combined it into one data frame. He then was able to calculate proportions of the prevalence of each soft skill out of the total soft skills in each list. Finally, he subtracted the proportions for each soft skill across the two different job descriptions to get a delta value.

The delta value is a good way to compare the soft skills prioritized by Data Science jobs compared to Nurse jobs. The higher the delta value, the more the soft skill is prioritized by Data Science jobs, the smaller the delta value, the more the soft skill is valued by Nurse jobs. Below, you can see the code used to generate the data frame and plot the most extreme deltas (both ways), on one plot for our group.

```
#Create variables for numbers of job descriptions and skills
DataN <- nrow(full_df)
NurseN <- nrow(nurse_full_df)
SkillN <- nrow(soft_skills)
#For each Soft Skill,
for (j in 1:SkillN){
  # add a zero entry under the column NurseCounts in final_df
  final_df$NurseCount[j] <- 0
  #For each Nurse Job description in the dataframe nurse_full_df
  for (i in 1:NurseN){
    #Set the value in the NurseCount column to the number of times the term in the
```

```
    #job_description column appears in the job description
    final_df$NurseCount[j] <- final_df$NurseCount[j] +
      str_count(tolower(nurse_full_df$job_description[i]),tolower(soft_skills$Text[j]))
  }
}
#Copy this to a separate dataframe (to avoid altering the total dataframe)
delta_df <- final_df
#Add columns for the frequency per job listing each skill appeared
delta_df$DataProp <- round(delta_df$Frequency / DataN, 4)
delta_df$NurseProp <- round(delta_df$NurseCount / NurseN, 4)
#Add a column that calculates the delta between the per listing
#amounts within Data Analyst listings and Nurse listings.
delta_df$Delta <- delta_df$DataProp - delta_df$NurseProp
#Rename the columns
names(delta_df) <- c("Skill", "DataCount", "NurseCount", "DataProp", "NurseProp", "Delta")
#Finally, isolate only those skills that were present in at
#least one job listing to narrow the relevant list
delta_df <- delta_df[!(delta_df$DataCount == 0 & delta_df$NurseCount == 0),]
```

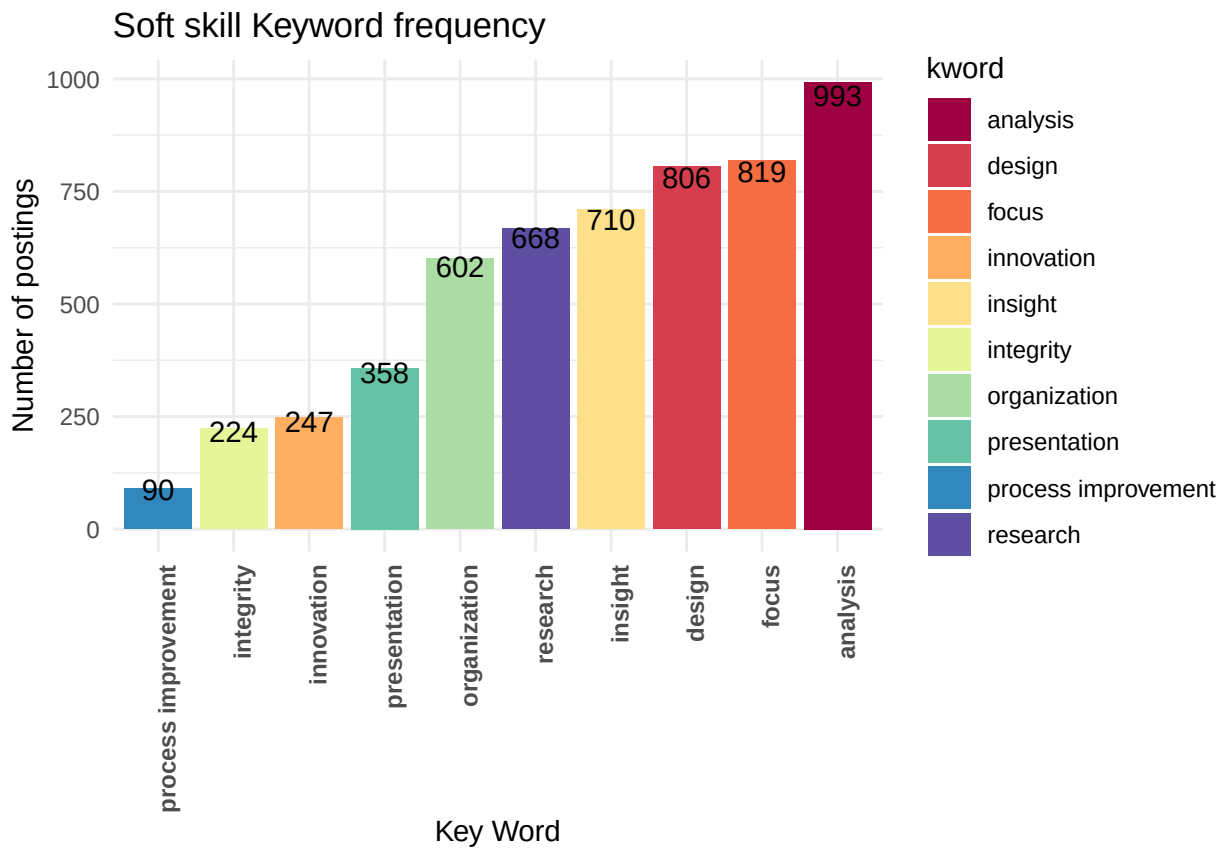
```
# frequency bar chart 1 (data)
freq_bar <- delta_df %>%
  filter(DataCount >= 160) %>%
  arrange(DataCount)
freq_bar_nurse <- delta_df %>%
  filter(NurseCount >= 40) %>%
  arrange(NurseCount)
freq_bar_delta <- delta_df %>%
  filter(Delta >= 0.1 | Delta <= -0.05) %>%
  arrange(Delta) %>%
  mutate(fillColor = ifelse(Delta > 0,
                            'More Prevalent in Data Science Job Descriptions',
                            'More Prevalent in Nurse Job Descriptions'))
ggplot(freq_bar_delta, aes(x=reorder(Skill, Delta),y=Delta, fill=fillColor)) +
  geom_bar(position="dodge",stat="identity", color = "#dddddd") +
  scale_fill_manual("Proportion of Skill",
                    values = c("More Prevalent in Data Science Job Descriptions" = "#C0DF85",
                               "More Prevalent in Nurse Job Descriptions" = "#FF958C")) +
  theme(panel.background = element_blank()) +
  theme(legend.title = element_blank()) +
  ylim(-1.95, 1.95) +
  ylab('Proportional difference') +
  xlab('Soft Skill') +
  coord_flip() +
  geom_text(aes(label=round(Delta, digits = 2), y = Delta + 0.15 * sign(Delta)),
            position = position_dodge(width = 0.5), color="#333333", fontface="bold", size=3.5) +
  theme(legend.position = "bottom")
```



As we can see from the plot, “Analysis”, “Research”, “Focus”, “Insight”, “Design”, and “Organization” are more prevalent in Data Science job descriptions than in Nurse job descriptions. Conversely, “Professional”, “Planning”, “Coordination”, “Customer Service”, and “Commitment” are more prevalent in Nurse job descriptions than Data Science job descriptions.

3.3 Number of postings with Key word occurences

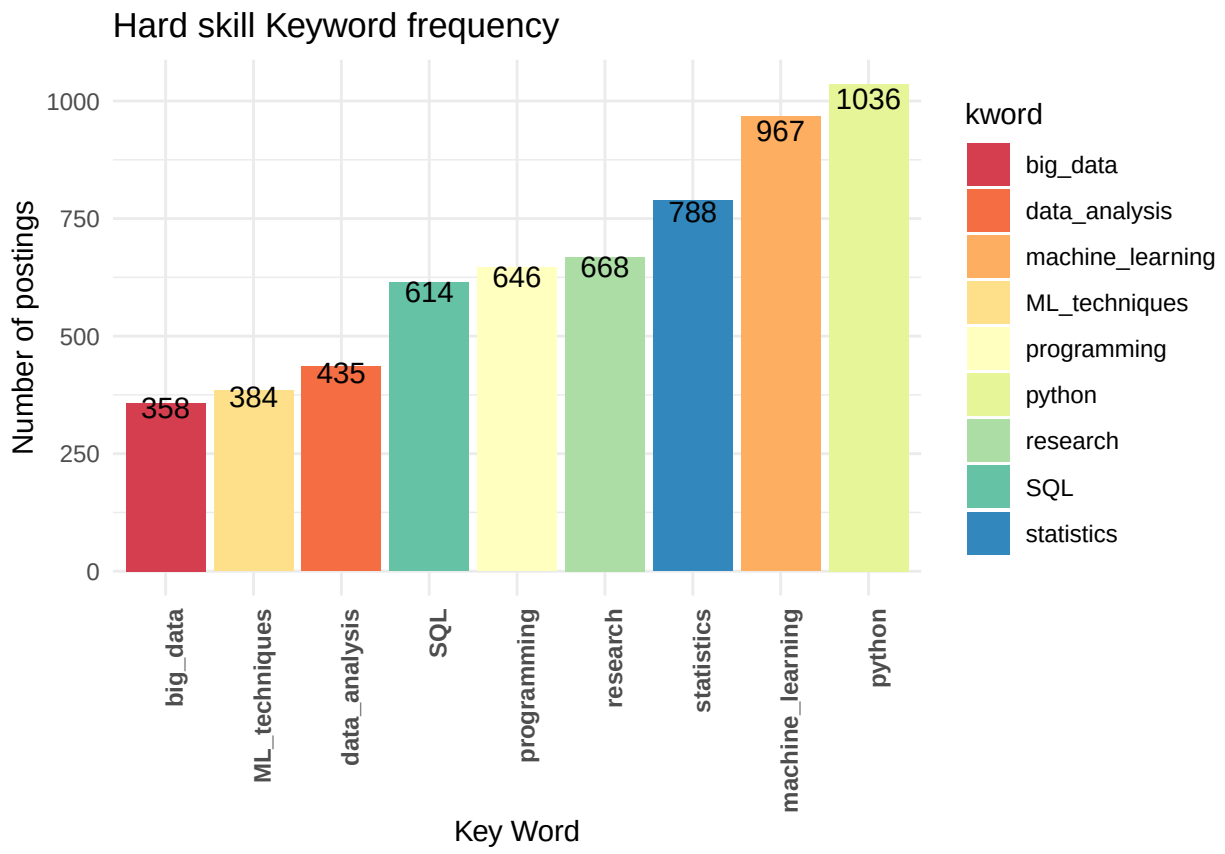
Taking this information, we were able to focus the identified key words back to the Data science dataset and see how many postings had these keywords.



3.4 Hard Skill frequency in the Technical job postings

```
top_skills <- full_df %>%
  filter(str_detect(job_description, 'machine\\s\\learning') |
         str_detect(job_description, 'research') |
         str_detect(job_description, 'python') |
         str_detect(job_description, 'statistics'))
full_df_rows <- nrow(full_df)
top_skill_rows <- nrow(top_skills)
paste0('Proportion of top hard skills in job descriptions = ',
       round(top_skill_rows/full_df_rows, digits = 4))
```

```
## [1] "Proportion of top hard skills in job descriptions = 0.9497"
```

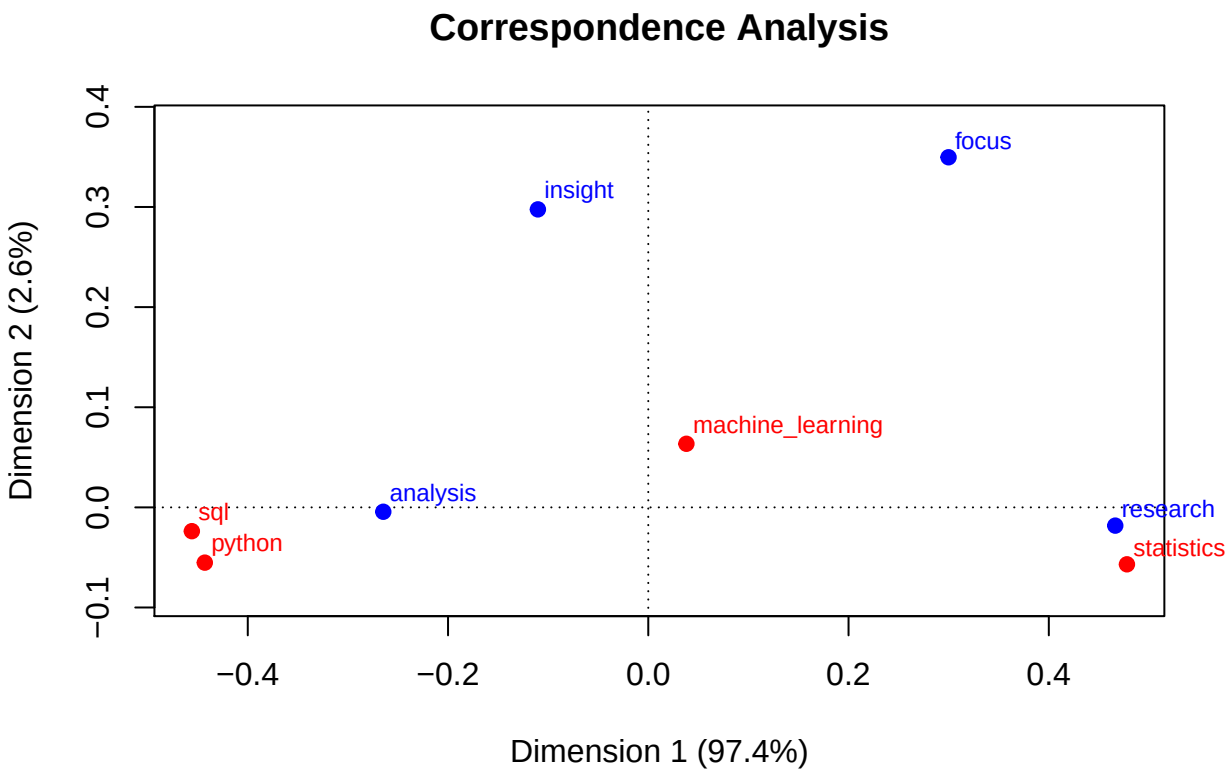


3.5 Key Word in Context (KWIC) Listings

We did some text analysis research and found R package ('corpustools'). Corpustools allowed us to find the number of hits that two words would be found within a given distance from one another in a corpus. Utilizing this method generated interesting tests on our top hard skill and top soft skill words, and see if certain hard skills frequently found closer (or were associated) with our top soft skills.

```
tc <- create_tcorpus(full_df$job_description)
hits_1_2 <- tc$search_features('"analysis machine*"~25')
hits_1_3 <- tc$search_features('"analysis sql"~25')
hits_1_4 <- tc$search_features('"analysis python"~25')
hits_1_5 <- tc$search_features('"analysis statistics"~25')
hits_2_2 <- tc$search_features('"research machine*"~25')
hits_2_3 <- tc$search_features('"research sql"~25')
hits_2_4 <- tc$search_features('"research python"~25')
hits_2_5 <- tc$search_features('"research statistics"~25')
hits_3_2 <- tc$search_features('"focus machine*"~25')
hits_3_3 <- tc$search_features('"focus sql"~25')
hits_3_4 <- tc$search_features('"focus python"~25')
hits_3_5 <- tc$search_features('"focus statistics"~25')
hits_4_2 <- tc$search_features('"insight machine*"~25')
hits_4_3 <- tc$search_features('"insight sql"~25')
hits_4_4 <- tc$search_features('"insight python"~25')
```

```
hits_4_5 <- tc$search_features('"insight statistics"~25')
kwic_1_2 <- tc$kwic(hits_1_2, ntokens = 3)
kwic_1_3 <- tc$kwic(hits_1_3, ntokens = 3)
kwic_1_4 <- tc$kwic(hits_1_4, ntokens = 3)
kwic_1_5 <- tc$kwic(hits_1_5, ntokens = 3)
kwic_2_2 <- tc$kwic(hits_2_2, ntokens = 3)
kwic_2_3 <- tc$kwic(hits_2_3, ntokens = 3)
kwic_2_4 <- tc$kwic(hits_2_4, ntokens = 3)
kwic_2_5 <- tc$kwic(hits_2_5, ntokens = 3)
kwic_3_2 <- tc$kwic(hits_3_2, ntokens = 3)
kwic_3_3 <- tc$kwic(hits_3_3, ntokens = 3)
kwic_3_4 <- tc$kwic(hits_3_4, ntokens = 3)
kwic_3_5 <- tc$kwic(hits_3_5, ntokens = 3)
kwic_4_2 <- tc$kwic(hits_4_2, ntokens = 3)
kwic_4_3 <- tc$kwic(hits_4_3, ntokens = 3)
kwic_4_4 <- tc$kwic(hits_4_4, ntokens = 3)
kwic_4_5 <- tc$kwic(hits_4_5, ntokens = 3)
k1 <- as.double(nrow(kwic_1_2))
k2 <- as.double(nrow(kwic_1_3))
k3 <- as.double(nrow(kwic_1_4))
k4 <- as.double(nrow(kwic_1_5))
k5 <- as.double(nrow(kwic_2_2))
k6 <- as.double(nrow(kwic_2_3))
k7 <- as.double(nrow(kwic_2_4))
k8 <- as.double(nrow(kwic_2_5))
k9 <- as.double(nrow(kwic_3_2))
k10 <- as.double(nrow(kwic_3_3))
k11 <- as.double(nrow(kwic_3_4))
k12 <- as.double(nrow(kwic_3_5))
k13 <- as.double(nrow(kwic_4_2))
k14 <- as.double(nrow(kwic_4_3))
k15 <- as.double(nrow(kwic_4_4))
k16 <- as.double(nrow(kwic_4_5))
k_pool <- data.frame(rbind(c(k1, k2, k3, k4),
                c(k5, k6, k7, k8),
                c(k9, k10, k11, k12),
                c(k13, k14, k15, k16)))
softskills <- c('analysis', 'research', 'focus', 'insight')
rownames(k_pool) <- softskills
k_pool <- k_pool %>%
  rename("machine_learning" = X1,
         "sql" = X2,
         "python" = X3,
         "statistics" = X4)
plot(ca(k_pool), main = "Correspondence Analysis", pch = 19)
```




```
kable(k_pool, align = rep('c', 4)) %>%
  kable_styling(bootstrap_options = c("striped"), full_width = F)
```

	machine_learning	sql	python	statistics
analysis	441	174	261	171
research	257	28	45	242
focus	23	2	2	10
insight	4	1	1	1

3.6 Visualizing keyword network and clusters

Lets try to visualize how the terms that we have recognized as the most important skills are structured inside the Job Description entries. We calculate first the co-occurrence of all the terms.

```
library(udpipe)
cooc <- cooccurrence(x = subset(x, upos %in% c("NOUN", "ADJ")),
                     term = "lemma",
                     group = c("doc_id", "paragraph_id", "sentence_id"), skipgram = 4)
dim(cooc)
```

```
## [1] 272556      3
```

3.6.1 We filter the terms so that we only keep those terms we recognized as "Soft Skills"

The co-occurrence table (cooc) above contains the frequencies of co-occurrence between each pair of words. Initally contained ALL the words in the text but we are only interested in the words that are part to the soft skills of interest. The chunk below filters out the co-occurrence table.

```
library(tidytext)
```

```
##
## Attaching package: 'tidytext'

## The following object is masked from 'package:corpustools':
##
##     get_stopwords
```

```
slill.comp <- c("analysis", "design", "focus", "innovation", "insight",
                "integrity", "organization", "presentation", "process", "improvement",
                "research", "skills", "adaptability", "supervising",
                "tolerance", "coordination", "planning", "professional",
                "training", "collaborative", "motivated", "work",
                "experience", "verbal", "responsible", "person")
cooc <- subset(cooc, term1 %in% slill.comp)
cooc <- subset(cooc, term2 %in% slill.comp)
dim(cooc)
```

```
## [1] 170   3
```

3.7 Word Network

We created a wordnetwork of all the words (chosen from our skill list) we filtered previously. The The wordnetwork graph contains only the words of interested linked by how often they appear together.

```
library(igraph)
```

```
##
## Attaching package: 'igraph'

## The following object is masked from 'package:formattable':
##
##     normalize

## The following objects are masked from 'package:purrr':
##
##     compose, simplify

## The following object is masked from 'package:tidyr':
##
##     crossing
```

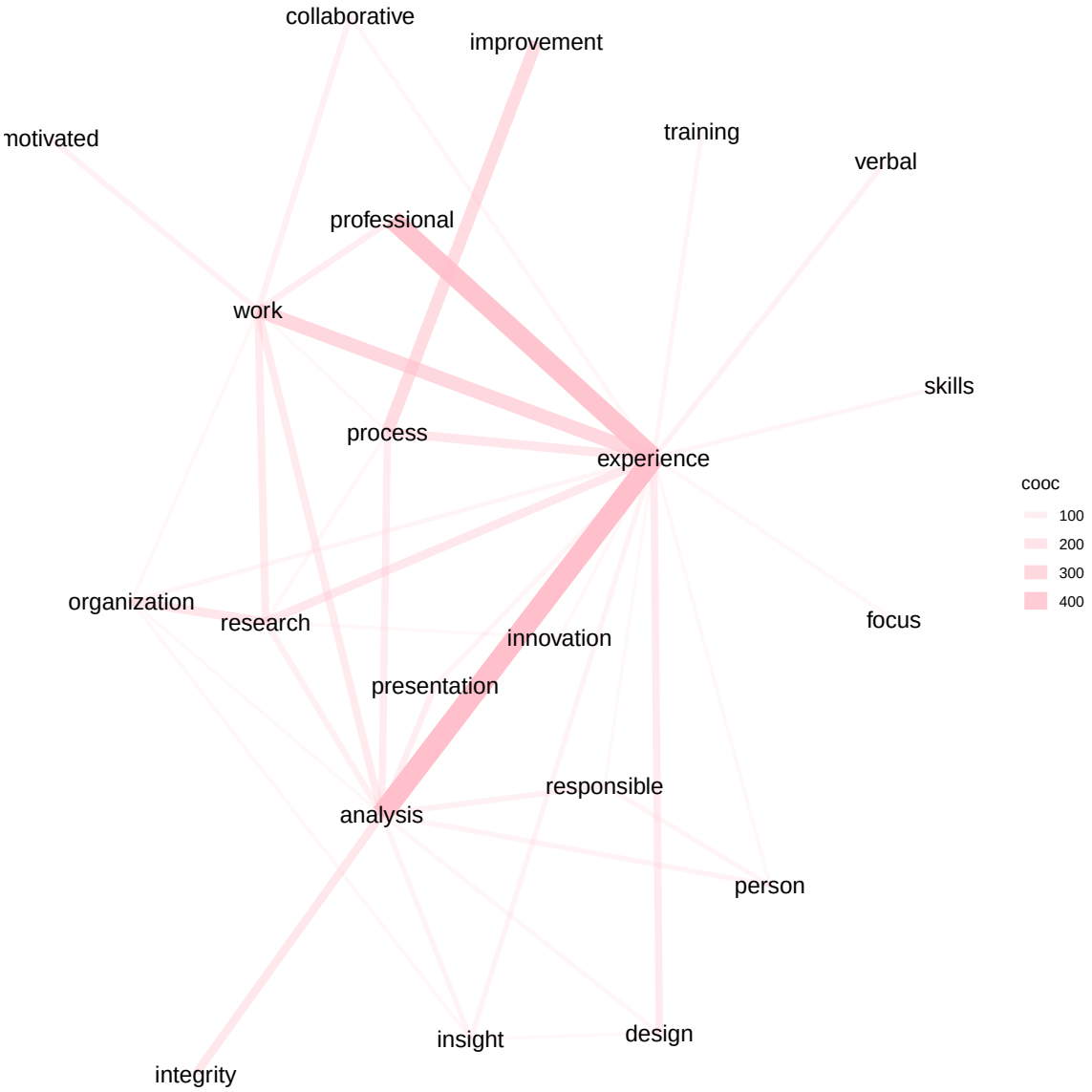
```
## The following object is masked from 'package:tibble':
##
##     as_data_frame

## The following objects are masked from 'package:dplyr':
##
##     as_data_frame, groups, union

## The following objects are masked from 'package:stats':
##
##     decompose, spectrum

## The following object is masked from 'package:base':
##
##     union
```

```
library(ggraph)
library(ggplot2)
wordnetwork <- head(cooc, 40)
wordnetwork <- graph_from_data_frame(wordnetwork)
ggraph(wordnetwork, layout = "dh") +
  geom_edge_link(aes(width = cooc, edge_alpha = cooc), edge_colour = "pink") +
  geom_node_text(aes(label = name), col = "black", size = 5) +
  theme_graph(base_family = "sans") +
  labs(title = "How Soft Skills Are Connected in the Job Description")
```



3.8 Cluster Analysis - Dendrogram

We can also analyze how the skill words are cluster together and visualize the results as a dendrogram. The dendrogram below draws a box around the four clusters of words that appear the closest together in the Job Description text.

```
### Dendrogram using community detection
# Community structure detection based on edge betweenness
# (http://igraph.org/r/doc/cluster_edge_betweenness.html)
cluster_edge_betweenness(wordnetwork, weights = E(wordnetwork)$cooc)
```

```
## Warning in cluster_edge_betweenness(wordnetwork, weights = E(wordnetwork)
## $cooc): At community.c:460 :Membership vector will be selected based on the
## lowest modularity score.
```

```
## Warning in cluster_edge_betweenness(wordnetwork, weights = E(wordnetwork)
## $cooc): At community.c:467 :Modularity calculation with weighted edge
## betweenness community detection might not make sense -- modularity treats
## edge weights as similarities while edge betwenness treats them as distances
```

```
## IGRAPH clustering edge betweenness, groups: 5, mod: 0.014
## + groups:
##   $`1`
##    [1] "analysis"     "experience"   "organization" "design"
##    [5] "person"       "insight"      "innovation"   "presentation"
##    [9] "responsible"  "verbal"       "training"     "skills"
##   [13] "focus"
##
##   $`2`
##   [1] "improvement"
##
##   $`3`
##   + ... omitted several groups/vertices
```

```
# Community detection via random walks (http://igraph.org/r/doc/cluster_walktrap.html)
cluster_walktrap(wordnetwork, weights = E(wordnetwork)$cooc, steps = 4)
```

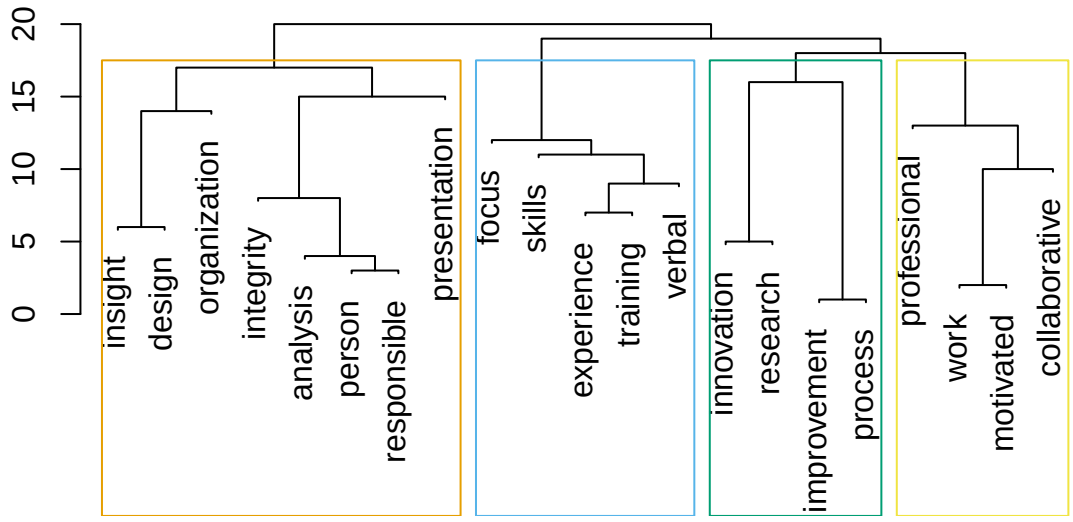
```
## IGRAPH clustering walktrap, groups: 5, mod: 0.12
## + groups:
##   $`1`
##    [1] "analysis"      "experience"    "organization"  "research"
##    [5] "design"        "professional"  "collaborative" "insight"
##    [9] "innovation"    "work"          "integrity"     "presentation"
##   [13] "verbal"        "training"      "skills"
##
##   $`2`
##   [1] "improvement" "process"
##
##   $`3`
##   + ... omitted several groups/vertices
```

```
# Community detection via optimization of modularity score
# This works for undirected graphs only
wordnetwork2 <- as.undirected(wordnetwork) # an undirected graph
cluster_fast_greedy(wordnetwork2, weights = E(wordnetwork2)$cooc)
```

```
## IGRAPH clustering fast greedy, groups: 4, mod: 0.21
```

```
## + groups:
##   $`1`
##   [1] "analysis"     "organization" "design"       "person"
##   [5] "insight"      "integrity"    "presentation" "responsible"
##
##   $`2`
##   [1] "improvement" "research"    "process"     "innovation"
##
##   $`3`
##   [1] "professional"  "collaborative" "motivated"     "work"
##
##   + ... omitted several groups/vertices
```

```
# Note that you can plot community object
comm <- cluster_fast_greedy(wordnetwork2, weights = E(wordnetwork2)$cooc)
#plot_dendrogram(comm, palette = categorical_pal(8))
plot_dendrogram(comm, mode="hclust", rect = 4, colbar = palette(rainbow(10)),
            hang = 0.01, ann = FALSE, main = "Top Keywords in Data Science Job
            Descriptions", sub = "", xlab = "", ylab = "")
```



4. Conclusion

No matter what approaches we used, we found that there are several "soft skills" that are especially prevalent in descriptions for Data Analyst job postings in the New York area (n = 1412). It should come as no surprise that the term "analysis" was the most common in these descriptions, as it is basically an extension of the job title. It only seems fitting that a Data Analyst be expected to be good at analysis. The other 11 terms that rounded out our Top-12 were Research, Focus, Design, Professional, Insight, Organization, Leadership, Presentation, Responsible, Training and Innovation.

It was especially interesting to see that 5 of the terms in our Top-12 were not found a single time within 569 job listings for Nurse positions in the New York area. None of "Professional", "Insight", "Leadership", "Presentation" or "Training" were mentioned in a single job description for these Nurse positions. This suggests that these are not simply words that appear in any generic job description - they are skills specifically sought for in Data Scientists. The remaining 7, while appearing occasionally in Nurse job descriptions, still showed incredibly high Delta values in terms of per listing frequency. The highest per Nurse listing Top-12 skill was Design with a 0.0562 per listing frequency (meaning it appeared on average in 1 out of around 18 Nurse listings), but this was dwarfed by the Data Analyst frequency of 1.0779 (meaning it appeared on average more than once a listing).

Data Analyst jobs appear to want a focused, insightful professional, who can analyze the results of research of their own design to come up with insights that they can then use to focus on training others within the organization.

5. Potential Next Steps

While working on this project we often found ourselves thinking about additional ways to improve or expand on the body of work we put together in case we found time to revisit it. Below is a list of a few of the things we considered:

- Narrowing the search of job descriptions to sections specifically outlining the skills and requirements of the job. Currently our approach looks at the entire description, which leaves us exposed to potentially matching terms not meant as required skillsets
- Accounting for the possibility that the Company Name contains a search term
- Looking at searches in different parts of the world to see if the skills sought are affected by culture/geography