Tidy and clean Linkedin Scraped Data

Anil Akyildirim, Nicholas Chung, Jai Jeffryes, Tamiko Jenkins, Joe Rovalino, Sie Siong Wong

10/20/2019

Contents

Introduction

As part of our project, we are tasked to answer the question "What are the most valued data science skills?" by working as a team, deciding what data to collect and how to collect it, use relational database and set of normalized tables and data exploration and analysis. Our team members are as follows;

- Anil Akyildirim
- Nicholas Chung
- Jai Jeffryes
- Tamiko Jenkins
- Joe Rovalino
- Sie Siong Wong

As part of project management tools, we have used Slack Private channel and Skype for Project Communication, Github for Project tracking, documentation and code collaboration, and Amazon Relational Database Service for data integration. All of our supporting code and data are on the GitHub repo, which documents branches and commits from our team.

- GitHub: https://github.com/pnojai/dskill
- Amazon Relational Database Service: msds607.ckxhi71v1dqf.us-east-1.rds.amazonaws.com

Load Data

Data Collection

We have reviewed and discussed different data types such as current job requirements around data scientists from job postings such as indeed.com or monster.com and articles around top data scientists skills in websites such as towardsdatascience and knuggets. Our approach built on the assumption that data scientists with jobs have the skills most valued by employers. We collected skills from employed data scientists.

We were inspired by the research of Jeff Hale whose article on data science skills appeared on the website, Medium.

- https://towardsdatascience.com/the-most-in-demand-skills-for-data-scientists-4a4a8db896db.

We discussed different methods of collecting the data and further how we can store it. As a result, we decided to work with usefull data within linkedin.com. We compared our findings from LinkedIn data to Mr. Hale's 2018 findings.

Load JSON files

```
# load all JSON
filenames <- list.files("data/profiles", pattern="*.json", full.names=TRUE) # this should give you a ch
example_file <- lapply(filenames[1], function(x) jsonlite::fromJSON(txt = x)) # a list in which each el
example_file
```

```
## [[1]]
## [[1]]$profileAlternative
## [[1]]$profileAlternative$name
## [1] "Aakanksha Jha"
##
## [[1]]$profileAlternative$headline
## [1] "Data Scientist at Microsoft"
##
## [[1]]$profileAlternative$location
## [1] "Greater Seattle Area"
##
## [[1]]$profileAlternative$connections
## [1] "500+"
##
## [[1]]$profileAlternative$summary
## [1] "<U+2605> Experienced Consultant with a demonstrated history of working in the information techn
##
##
## [[1]]$aboutAlternative
## [[1]]$aboutAlternative$text
## [1] "<U+2605> Experienced Consultant with a demonstrated history of working in the information techn
##
##
## [[1]]$positions
## list()
##
## [[1]]$educations
##                                                    title
## 1 Arizona State University - W. P. Carey School of Business
## 2                                       University of Mumbai
## 3                                     R. D. National College
##                            degree date1 date2
## 1         Master of Science - MS  2017  2018
## 2 Bachelor of Engineering (BEng)  2010  2014
## 3      Associate of Science - AS  2008  2010
##
## [[1]]$skills
##                                 title count
## 1                     Machine Learning    11
## 2                              Python    10
## 3                                   R    10
## 4                         Data Mining     5
## 5                  Data Visualization     6
```

```
## 6  Enterprise Resource Planning (ERP)     8
## 7                   Functional Testing     5
## 8                   Project Management     2
## 9                IT Service Management     2
## 10                     Trend Analysis     5
## 11                  Commodity Trading     8
## 12                        Credit Risk     7
## 13               Design of Experiments     3
## 14               Business Intelligence  <NA>
## 15                           Analytics  <NA>
## 16                       Data Analysis     1
## 17                                 SQL    10
## 18                             Tableau     6
## 19                                SPSS     3
## 20                             Minitab     4
## 21                                Java    10
## 22                                 C++    10
## 23            SAP Sales & Distribution     8
## 24                Microsoft PowerPoint     8
## 25                                   C     5
## 26                                HTML    10
## 27                    Microsoft Office     5
## 28                     Microsoft Azure     4
## 29                     Microsoft Excel     2
## 30                           Databases     2
## 31                      Microsoft Word     1
## 32                              Matlab  <NA>
## 33                       Data Cleaning     5
## 34                Statistical Inference     2
## 35                  Logistic Regression     3
## 36                  Linear Programming     2
## 37                  Hypothesis Testing     3
## 38                                ANOVA     4
## 39  Database Management System (DBMS)     8
## 40                                SNOW     8
## 41         Root Cause Problem Solving     4
## 42            SAP Logistics Execution     4
## 43                       Data Analytics     4
## 44     Business Continuity Management     2
## 45                              SAP TSW     9
## 46                         Product Risk     7
## 47    Data Warehousing and Management     6
## 48  Legacy System Migration Workbench     1
## 49                             StatTools     3
## 50                                   EDA     3
##
## [[1]]$recommendations
## [[1]]$recommendations$givenCount
## [1] "0"
##
## [[1]]$recommendations$receivedCount
## [1] "0"
##
## [[1]]$recommendations$given
```

```
## list()
##
## [[1]]$recommendations$received
## list()
##
##
## [[1]]$accomplishments
##   count
## 1    13
## 2    10
## 3     3
## 4     2
## 5     1
##
## 1 APL Logistics, House Prices: Advanced Regression Techniques, Profit Optimization At KOLBY'S (Data M
## 2
## 3
## 4
## 5
##
## [[1]]$peopleAlsoViewed
##                                                        user
## 1                      https://www.linkedin.com/in/ishamehra/
## 2   https://www.linkedin.com/in/shruthi-adimurthy-831b02129/
## 3       https://www.linkedin.com/in/yimei-liz-chen-6b4a267b/
## 4  https://www.linkedin.com/in/varshini-ramaseshan-3b060739/
## 5                 https://www.linkedin.com/in/rishabh-joshi/
## 6                  https://www.linkedin.com/in/priyamatnani/
## 7                 https://www.linkedin.com/in/anjalichadha1/
## 8                  https://www.linkedin.com/in/bhavanavijay/
## 9             https://www.linkedin.com/in/anmol-shrivastava/
## 10              https://www.linkedin.com/in/santoshmashetty/
##                                                        text
## 1                                  Data Scientist at Facebook
## 2  Data Analyst at Citi | Tableau Desktop Associate | #GHC19
## 3                                  Data Scientist at Facebook
## 4                                Data Scientist at Microsoft
## 5                                  Data Scientist at Facebook
## 6                                    Data Scientist at Airbnb
## 7               Business Analyst at Amazon Web Services (AWS)
## 8                                        Analytics at Google
## 9                                         Analyst at Carvana
## 10                     Data & Applied Scientist at Microsoft
##
## [[1]]$volunteerExperience
## list()
##
## [[1]]$profile
## [[1]]$profile$name
## [1] "Aakanksha Jha"
##
## [[1]]$profile$headline
## [1] "Data Scientist at Microsoft"
##
```

```
## [[1]]$profile$location
## [1] "Greater Seattle Area"
##
## [[1]]$profile$connections
## [1] "500+"
##
## [[1]]$profile$summary
## [1] "<U+2605> Experienced Consultant with a demonstrated history of working in the information techn
```

Structure Data

Bind fromJSON results

```
# apply fromJSON to read in all of the json files
# create the column (variable) title, headline, which will be populated with json file identifying info
# extract the skills data which contains the variables title and counts
# bind the results together as a data frame named raw_df
raw_df <- dplyr::bind_rows(sapply(filenames, function(x) fromJSON(x, flatten=TRUE)$skills), .id="headli
head(raw_df)
```

```
##                                headline                                title
## 1 data/profiles/aakankshajha.json.json                   Machine Learning
## 2 data/profiles/aakankshajha.json.json                            Python
## 3 data/profiles/aakankshajha.json.json                                 R
## 4 data/profiles/aakankshajha.json.json                       Data Mining
## 5 data/profiles/aakankshajha.json.json                 Data Visualization
## 6 data/profiles/aakankshajha.json.json Enterprise Resource Planning (ERP)
##   count
## 1    11
## 2    10
## 3    10
## 4     5
## 5     6
## 6     8
```

Extract Headlines

```
# apply fromJSON to read in all of the json files
# extract the headline variable from the profile data, saving each file name as the variable title
# save the mapping as  data frame headlines
headlines <- sapply(filenames, function(x) fromJSON(x, flatten=TRUE)$profile$headline, USE.NAMES = TRUE)
head(headlines)
```

```
##              data/profiles/aakankshajha.json.json
##                     "Data Scientist at Microsoft"
##              data/profiles/afshineamidi.json.json
##                                "Data Scientist at Uber"
##                    data/profiles/aj1212.json.json
##        "Data Scientist at Amazon (Audible group)"
##                data/profiles/akshay-kher.json.json
##                           "Data Scientist at Amazon"
##          data/profiles/alexandrampappas.json.json
##                     "Data Scientist and Engineer!"
## data/profiles/alice-xingwei-lu-09a1b799.json.json
##                     "Data Science Manager at Uber"
```

Map Headlines to Skills

```
#  apply a look up of the variable title specifying the filename
#  and add the headline value from the headlines  data frame
#  to the headlines variable in  data frame raw_df

raw_df$headline <- sapply(raw_df$headline, function(x) headlines[x])
head(raw_df)
```

```
##                       headline                                title count
## 1 Data Scientist at Microsoft                    Machine Learning    11
## 2 Data Scientist at Microsoft                              Python    10
## 3 Data Scientist at Microsoft                                   R    10
## 4 Data Scientist at Microsoft                         Data Mining     5
## 5 Data Scientist at Microsoft                  Data Visualization     6
## 6 Data Scientist at Microsoft Enterprise Resource Planning (ERP)     8
```

Format Data

Name and convert variables and data

```
df_conv <- raw_df
names(df_conv) <- c("title", "skills", "count")
class(df_conv)
```

```
## [1] "data.frame"
```

```
# coerce any nulls to na's
sapply(df_conv, class)
```

```
##       title      skills       count
## "character" "character" "character"
```

```
# create numeric types in counts column
df_conv$count <- as.numeric(df_conv$count)
sapply(df_conv, class)
```

```
##       title      skills       count
## "character" "character"   "numeric"
```

Remove NA's from numeric column

```
# count all rows
# 4822
nrow(df_conv)
```

```
## [1] 4778
```

```
# view a subset of rows with na's mixed with complete rows
df_conv[11:20,]
```

```
##                          title                 skills count
## 11 Data Scientist at Microsoft      Commodity Trading     8
## 12 Data Scientist at Microsoft            Credit Risk     7
## 13 Data Scientist at Microsoft Design of Experiments     3
## 14 Data Scientist at Microsoft Business Intelligence    NA
## 15 Data Scientist at Microsoft              Analytics    NA
```

```
## 16 Data Scientist at Microsoft         Data Analysis     1
## 17 Data Scientist at Microsoft                  SQL    10
## 18 Data Scientist at Microsoft              Tableau     6
## 19 Data Scientist at Microsoft                 SPSS     3
## 20 Data Scientist at Microsoft              Minitab     4
```

```
# filter for any rows with na
# count all rows with na's
# 942
df_na <- df_conv %>% filter_all(any_vars(is.na(.)))
nrow(df_na)
```

```
## [1] 916
```

```
# omit any rows with na's
# save rows without na's as a data frame names df
# count the data frame
# 4822-942 = 3880
df_omit <- na.omit(df_conv)
# view the same rows without na's
df_omit[11:20,]
```

```
##                          title               skills count
## 11 Data Scientist at Microsoft    Commodity Trading     8
## 12 Data Scientist at Microsoft          Credit Risk     7
## 13 Data Scientist at Microsoft Design of Experiments     3
## 16 Data Scientist at Microsoft         Data Analysis     1
## 17 Data Scientist at Microsoft                  SQL    10
## 18 Data Scientist at Microsoft              Tableau     6
## 19 Data Scientist at Microsoft                 SPSS     3
## 20 Data Scientist at Microsoft              Minitab     4
## 21 Data Scientist at Microsoft                 Java    10
## 22 Data Scientist at Microsoft                  C++    10
```

```
nrow(df_omit)
```

```
## [1] 3862
```

Store Data

Prepare data values for storage

```
#  Encoding(x) <- "latin1"
#  x <-  iconv(x, "latin1", "UTF-8", sub='')
#  x <- stringr::str_replace(x,",","")
#  Encoding(x) <- "UTF-8"


df_clean <- df_omit
# Remove non-ASCII character codes
test <- df_clean[256,]
df_clean$skills <- sapply(df_clean$skills, function(x) gsub('[^\x20-\x7E]', '', x))
df_clean$title <- sapply(df_clean$title, function(x) gsub('[^\x20-\x7E]', '', x))
df_clean$skills <- sapply(df_clean$skills, function(x) gsub('[@]', 'at', x))
df_clean$title <- sapply(df_clean$title, function(x) gsub('[@]', 'at', x))
df_clean$skills <- sapply(df_clean$skills, function(x) gsub('[\\|\\(\\),]', '', x))
```

```
df_clean$title <- sapply(df_clean$title, function(x) gsub('[\\|\\(\\),]', '', x))
Encoding(df_clean$skills) <- "UTF-8"
Encoding(df_clean$title) <- "UTF-8"
head(df_clean)
```

```
##                         title                          skills count
## 1 Data Scientist at Microsoft                Machine Learning    11
## 2 Data Scientist at Microsoft                          Python    10
## 3 Data Scientist at Microsoft                               R    10
## 4 Data Scientist at Microsoft                     Data Mining     5
## 5 Data Scientist at Microsoft               Data Visualization     6
## 6 Data Scientist at Microsoft Enterprise Resource Planning ERP     8
```

```
# View original
test
```

```
##                                                                                                                                                  t
## 373 Data Scientist at Conde Nast • MS in Data Science, Columbia University • IIIT-H Alumnus • Marath
##     skills count
## 373    SQL     6
```

```
# view cleaned text
df_clean[256,]
```

```
##                                                                                                                                                  title
## 373 Data Scientist at Conde Nast  MS in Data Science Columbia University  IIIT-H Alumnus  Marathoner
##     skills count
## 373    SQL     6
```

Prepare data format for storage

```
# TODO: follow df.csv convention

# Add rownames (indices) as a skill id
# to final dataframe to prepare for
# SQL-based storage and to provide option to
# remove automatic row names from write csv
# Remove depr function
#df_csv <- add_rownames(df, var = "skill_id")

# NB: these are the original row id's based on R records
# to generate skill_ids without skips for na's removed
# use a seq
df_csv <- tibble::rownames_to_column(df_clean, var = "skill_id")
df_csv$skill_id <- as.numeric(df_csv$skill_id)
head(df_csv)
```

```
##   skill_id                         title                          skills
## 1        1 Data Scientist at Microsoft                Machine Learning
## 2        2 Data Scientist at Microsoft                          Python
## 3        3 Data Scientist at Microsoft                               R
## 4        4 Data Scientist at Microsoft                     Data Mining
## 5        5 Data Scientist at Microsoft               Data Visualization
## 6        6 Data Scientist at Microsoft Enterprise Resource Planning ERP
##   count
## 1    11
```

```
## 2    10
## 3    10
## 4     5
## 5     6
## 6     8
```

```
# TODO: follow df.csv convention
# Rearrange column order with dplyr select
df_csv <- dplyr::select(df_csv, skill_id, skills, count, title)

head(df_csv)
```

```
##   skill_id                           skills count
## 1        1                 Machine Learning    11
## 2        2                           Python    10
## 3        3                                R    10
## 4        4                     Data Mining     5
## 5        5                Data Visualization     6
## 6        6 Enterprise Resource Planning ERP     8
##                          title
## 1 Data Scientist at Microsoft
## 2 Data Scientist at Microsoft
## 3 Data Scientist at Microsoft
## 4 Data Scientist at Microsoft
## 5 Data Scientist at Microsoft
## 6 Data Scientist at Microsoft
```

Write to csv

```
# write csv and upload to our mysql database
# Encoding(df_csv)
write.csv(df_csv, "results/df_alt.csv", row.names=FALSE, fileEncoding="UTF-8")
```

Load Data

Load the data from the database

```
# load the data in the database and look at 2018 Linkedin Data
user_name <- 'anil'
user_password <- "redy2rok"
database <- 'prj3'
host_name <- 'msds607.ckxhi71v1dqf.us-east-1.rds.amazonaws.com'

#connecting to the MySQL database

myDb <- dbConnect(RMariaDB::MariaDB(), user=user_name, password=user_password, dbname=database, host=hos
myDb
```

```
## <MariaDBConnection>
##   Host:    msds607.ckxhi71v1dqf.us-east-1.rds.amazonaws.com
##   Server:  5.7.22-log
##   Client:  5.5.1
```

View tables

```
#list of tables we have
dbListTables(myDb)
```

```
##  [1] "agg_linkedin"            "df"
##  [3] "df_bak"                  "ds_general_skills_clean"
##  [5] "dsmain"                  "footable"
##  [7] "just_skills"             "payscale_data"
##  [9] "rawdata"                 "sample_linkedin_tall"
## [11] "sample_linkedin_wide"    "skills_raw"
```

Exploring the Data

View 2018 Data

```
# lets load 2018 Linkedin Data
df <- dbGetQuery(myDb, "select * from df")
head(df)
```

```
##   skill_id           skills count                    title
## 1        1           Python    11 Data Scientist at Square
## 2        2                R     9 Data Scientist at Square
## 3        3              C++     7 Data Scientist at Square
## 4        4  Data Structures     5 Data Scientist at Square
## 5        5       Statistics     2 Data Scientist at Square
## 6        6 Machine Learning     2 Data Scientist at Square
```

```
nrow(df)
```

```
## [1] 3880
```

View 2019 Data

```
# There are more than 145 skills, clean to data similar to 2018 data
df <- subset(df, select = c(skills, count))
colnames(df) <- c("Skills", "Linkedin")
head(df)
```

```
##             Skills Linkedin
## 1           Python       11
## 2                R        9
## 3              C++        7
## 4  Data Structures        5
## 5       Statistics        2
## 6 Machine Learning        2
```

```
nrow(df)
```

```
## [1] 3880
```

```
# there are skills that is listed more than once. finding those
n_occur <- data.frame(table(df$Skills))
skills_more_once <- n_occur[n_occur$Freq > 1,]
head(skills_more_once)
```

```
##                 Var1 Freq
## 4        A/B Testing    3
```

```
## 8             Access    4
## 9         Accounting    2
## 11 Actuarial Science    5
## 14   Adobe Photoshop    4
## 15       Advertising    3
```

```
nrow(skills_more_once)
```

```
## [1] 349
```

```
# we need to add the count of the duplicate skills rows

df <- aggregate(Linkedin ~ Skills, dat=df, FUN=sum)
head(df)
```

```
##          Skills Linkedin
## 1          .NET        9
## 2   3D Modeling        1
## 3 8051 Assembly        3
## 4   A/B Testing        4
## 5        Abaqus       11
## 6  Ableton Live        1
```

```
# data is collected and ready to be analyzed at this point
summary(df)
```

```
##     Skills             Linkedin
##  Length:929         Min.   :   1
##  Class :character   1st Qu.:   2
##  Mode  :character   Median :   5
##                     Mean   :  33
##                     3rd Qu.:  17
##                     Max.   :2196
```

```
str(df)
```

```
## 'data.frame':    929 obs. of  2 variables:
##  $ Skills  : chr  ".NET" "3D Modeling" "8051 Assembly" "A/B Testing" ...
##  $ Linkedin:integer64 9 1 3 4 11 1 1 23 ...
```

```
df$Skills <- as.character(df$Skills)
df$Linkedin <- as.numeric(df$Linkedin)
```

Visualizing the Data

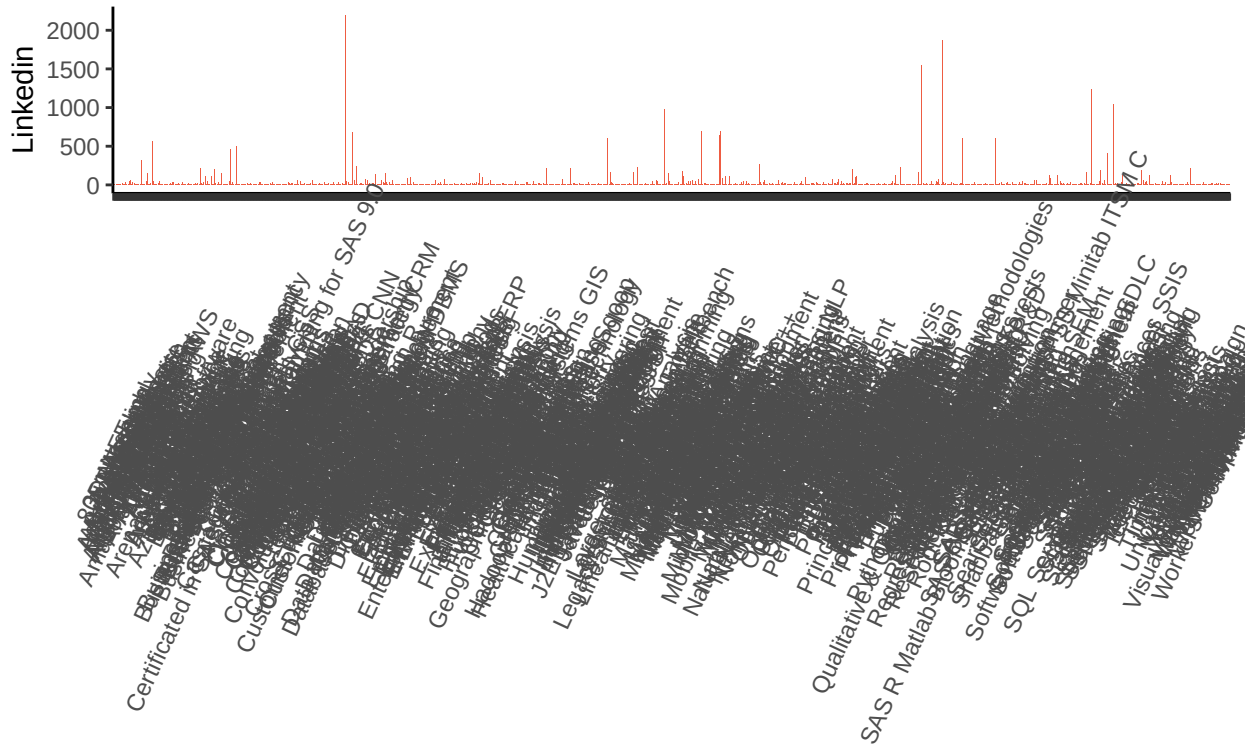
Improvements

The first exploratory pass is crowded. We'll filter the data in the next pass.

```
#We have 1157 observations (skills) that data science roles use in linkedin
#let's see the distribution

theme_set(theme_classic())

ggplot(df, aes(x=Skills, y=Linkedin))+
  geom_bar(stat="identity", width = 0.5, fill=("tomato2"))+
             theme(axis.text.x = element_text(angle = 65, vjust=0.6))
```




```
# we have way too many skills so let's only focus on the ones that has significant count.

df <- filter(df, Linkedin >100)
head(df)
```

```
##           Skills Linkedin
## 1     Algorithms      317
## 2       Analysis      148
## 3      Analytics      560
## 4       Big Data      214
## 5 Bioinformatics      106
## 6      Bloomberg      107
```

```
nrow(df)
```

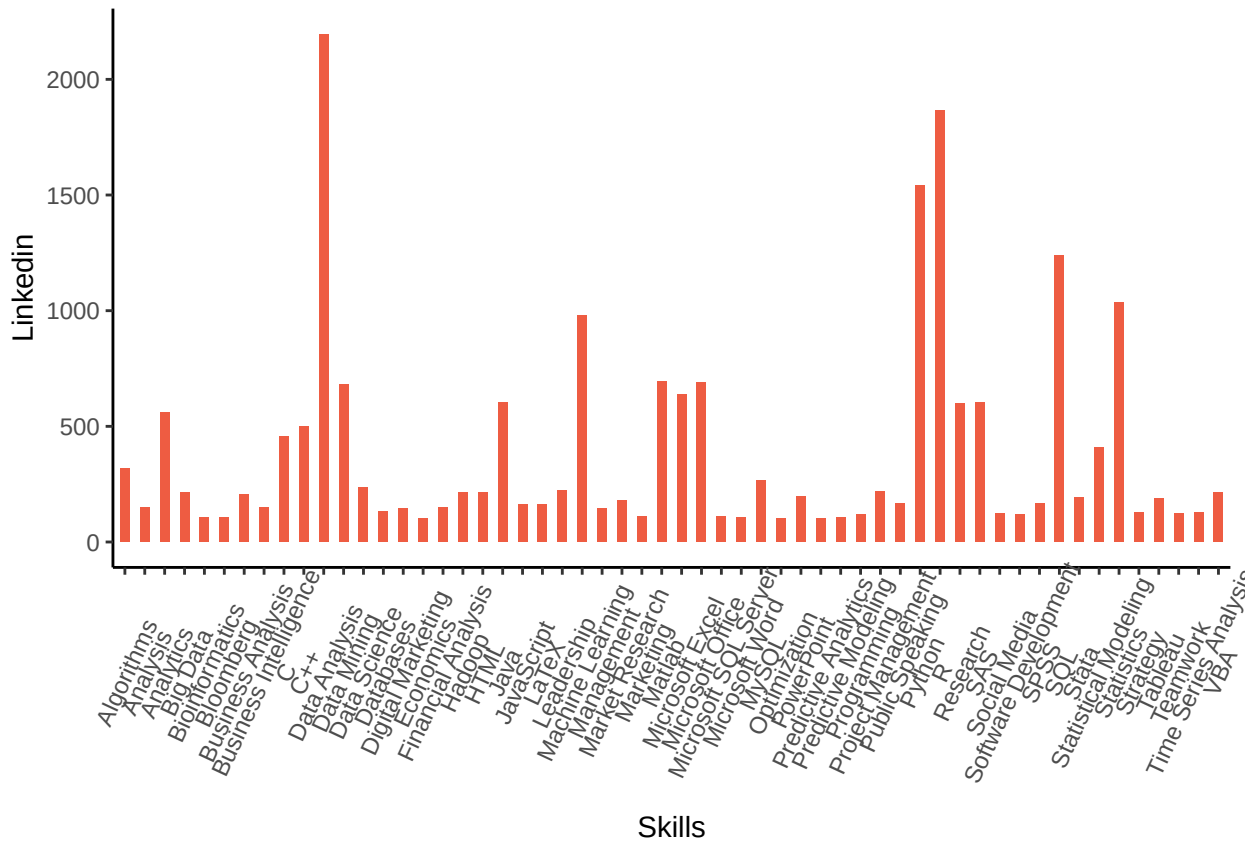
```
## [1] 56
```

Visualization

```
# we narrowed it down to 57 skills. Let's see how distribution looks like.

theme_set(theme_classic())

ggplot(df, aes(x=Skills, y=Linkedin))+
  geom_bar(stat="identity", width = 0.5, fill=("tomato2"))+
             theme(axis.text.x = element_text(angle = 65, vjust=0.6))
```




```
# let's narrow it down further.
df <- filter(df, Linkedin > 200)
head(df)
```

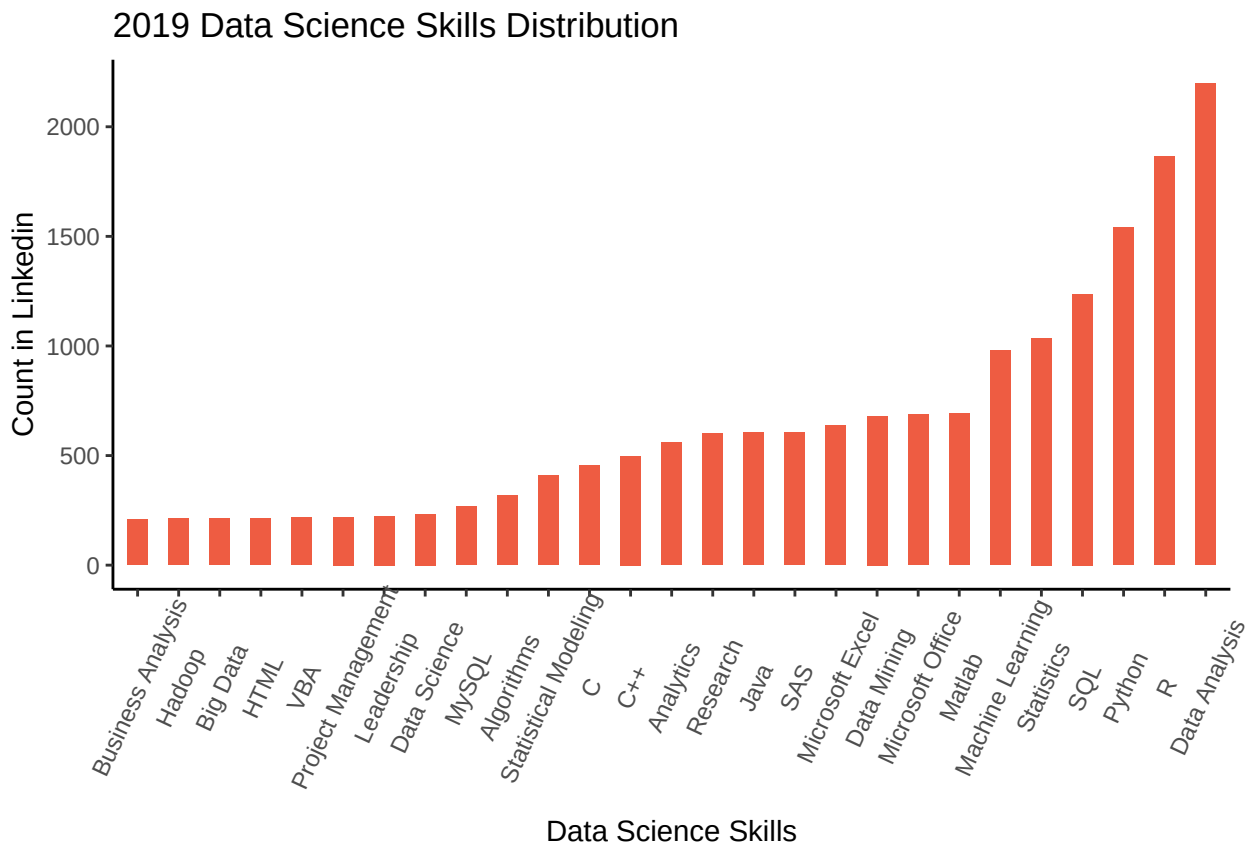
```
##              Skills Linkedin
## 1        Algorithms      317
## 2         Analytics      560
## 3          Big Data      214
## 4 Business Analysis      207
## 5                 C      456
## 6               C++      498
```

```
nrow(df)
```

```
## [1] 27
```

```
theme_set(theme_classic())

ggplot(df, aes(x=reorder(Skills, Linkedin, fun=max), y=Linkedin))+
  geom_bar(stat="identity", width = 0.5, fill=("tomato2"))+
  labs(title="2019 Data Science Skills Distribution",
       x="Data Science Skills",
       y="Count in Linkedin")+
  theme(axis.text.x = element_text(angle = 65, vjust=0.6))
```



Analyze the Data

The Data Science skills Distribution chart for 2019 shows us the most frequent data science skills that people use for their Linkedin Profiles. The results show us that, Data Analysis; as part of General Data Skills, is the most commonly used skill within Data Scientists in Linkedin. The top three programming languages used within the profiles are R, Python and SQL. Statistics and Machine Learning are in 5th and 6th place in that order. If we consider Machine Learning and Statistics, as part of General Data Science Skills and Programming Languages as part of Technical Data Science Skills, we can conclude that

Top three General Data Science Skills are Data Analysis, Statistics and Machine Learning.

Top three Technical Data Science Skills are R, Python and SQL.

```
# count for top three General and Technical Data Science Skills

data_analysis <- filter(df, df$Skills=="Data Analysis")
machine_learning <- filter(df, df$Skills=="Machine Learning")
statistics <- filter(df, df$Skills=="Statistics")
python <- filter(df, df$Skills=="Python")
r <- filter(df, df$Skills=="R")
sql <- filter(df, df$Skills=="SQL")
data_analysis
```

```
##          Skills Linkedin
## 1 Data Analysis     2196
```

```
machine_learning
```

```
##             Skills Linkedin
```

```
## 1 Machine Learning      979
```

```
statistics
```

```
##       Skills Linkedin
## 1 Statistics     1036
```

```
python
```

```
##   Skills Linkedin
## 1 Python     1539
```

```
r
```

```
##   Skills Linkedin
## 1      R     1864
```

```
sql
```

```
##   Skills Linkedin
## 1    SQL     1237
```

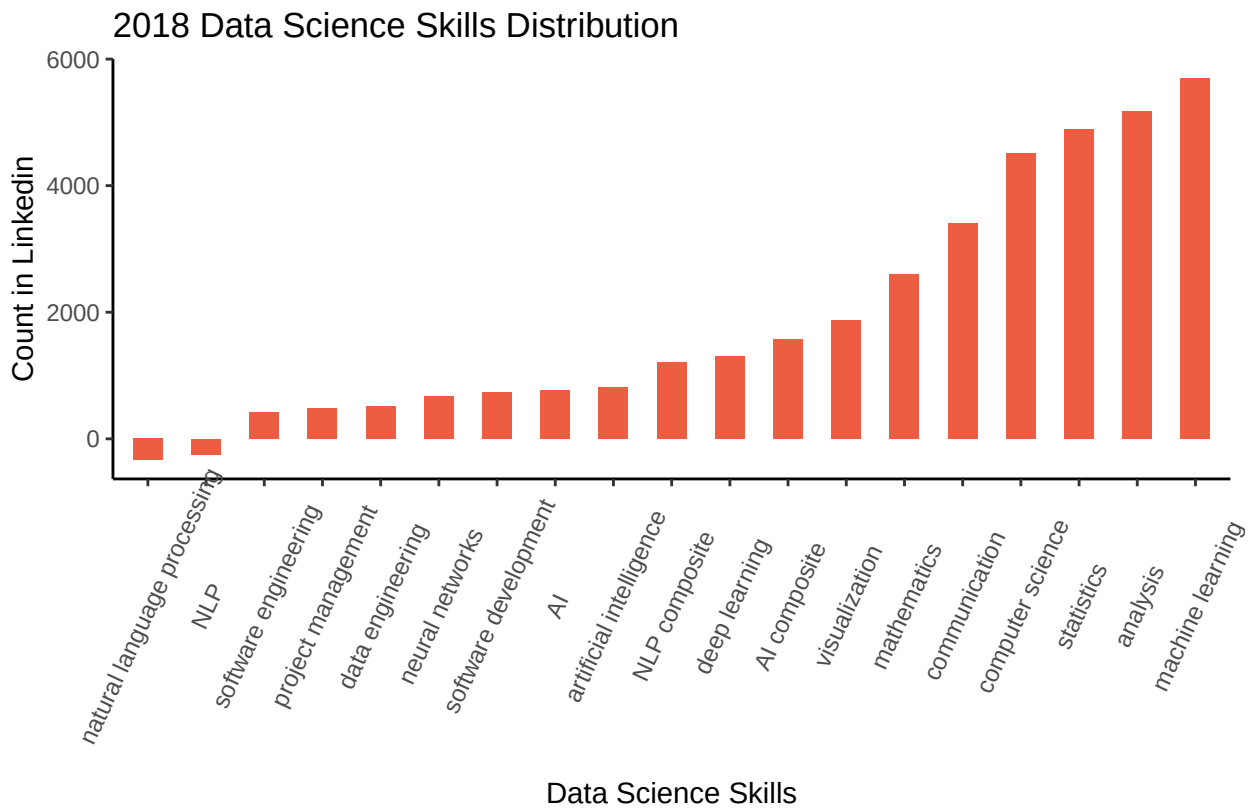
We can also look at the LinkedIn Data Set from Jeff Hale and see if they follow the same pattern.

```
# lets load 2018 Linkedin Data from Jeff Hale.
skills_2018 <- dbGetQuery(myDb, "select * from ds_general_skills_clean")
skills_2018$LinkedIn <- as.numeric(skills_2018$LinkedIn) # little cleanup
skills_2018
```

```
##                       Keyword LinkedIn Indeed SimplyHired Monster
## 1            machine learning     5701   3439        2561    2340
## 2                    analysis     5168   3500        2668    3306
## 3                  statistics     4893   2992        2308    2399
## 4            computer science     4517   2739        2093    1900
## 5               communication     3404   2344        1791    2053
## 6                 mathematics     2605   1961        1497    1815
## 7               visualization     1879   1413        1153    1207
## 8                AI composite     1568   1125         811     687
## 9               deep learning     1310    979         675     606
## 10              NLP composite     1212    910         660     582
## 11       software development      732    627         481     784
## 12            neural networks      671    485         421     305
## 13          data engineering      514    339         276     200
## 14        project management      476    397         330     348
## 15       software engineering      413    295         250     512
## 16                         AI      760    531         411     344
## 17    artificial intelligence      808    594         400     343
## 18                        NLP     -246   -192        -135    -144
## 19 natural language processing     -332   -197         -70       5
```

```
# analyze briefly to see if there are differences
theme_set(theme_classic())

ggplot(skills_2018, aes(x=reorder(Keyword, LinkedIn, fun=max),y=LinkedIn))+
  geom_bar(stat="identity", width = 0.5, fill=("tomato2"))+
  labs(title="2018 Data Science Skills Distribution",
       x="Data Science Skills",
       y="Count in Linkedin",
       caption = "Source: Jeff Hale 2018 Data Skills Analysis")+
  theme(axis.text.x = element_text(angle = 65, vjust=0.6))
```



With the assumption of computer science covering the Programming Lanaguages, we can see that the data science skills distribution for 2018 is similar to our Data Science Skills Distribution for 2018. Machine Learning , Data Analysis and Statistics leading the top General Data Science Skills. The only difference we see is that Machine Learning is slightly more used Data Science Skill than Data Analysis.

Simplify 2019 data frame

```
# subset vector v, removing the headline variable[c(-1)]
# save the new vector as v_counts
df_counts <- df_csv
head(df_counts)
```

```
##   skill_id                          skills count
## 1        1                Machine Learning    11
## 2        2                          Python    10
## 3        3                               R    10
## 4        4                     Data Mining     5
## 5        5               Data Visualization     6
## 6        6 Enterprise Resource Planning ERP     8
##                          title
## 1 Data Scientist at Microsoft
## 2 Data Scientist at Microsoft
## 3 Data Scientist at Microsoft
## 4 Data Scientist at Microsoft
## 5 Data Scientist at Microsoft
```

```
## 6 Data Scientist at Microsoft
```

View aggregate 2019 Skills counts

```
# aggregate the counts for each unique skill
# store as agg_df_counts data frame
agg_df_counts <- df_counts %>%
    group_by(skills)  %>%
    dplyr::summarise(count = n())  %>%
    arrange(desc(count))

agg_df_counts
```

```
## # A tibble: 928 x 2
##    skills           count
##    <chr>            <int>
##  1 Data Analysis      149
##  2 R                  149
##  3 Python             137
##  4 SQL                121
##  5 Machine Learning    96
##  6 Statistics          89
##  7 Microsoft Excel     82
##  8 Research            80
##  9 Microsoft Office    75
## 10 Matlab              68
## # ... with 918 more rows
```

Conclusions

The six data science skills most valued by employers in 2019 appear to be the following.

General Data Science Skills:

1- Data Analysis => 2196

2- Machine Learning => 979

3- Statistics => 1036

Technical Data Science Skills

1- R => 1864

2- Python => 1539

3- SQL => 1237

Our approach differed from Mr. Hale's. He investigated programming languages as a separate research question. Our approach commingles them. Therefore, though our high-ranking skills list includes the languages R, Python, and SQL, nothing is to be concluded from their absence from Hale's list. What we see in common are the skills of analysis, statistics, and machine learning. We believe the data tell a compelling story about investment in these disciplines.