Project 3 Code

James Chun, Haoming Chen, Pricilla Nakyazze

Introduction

In this project, we analyzed a dataset of 12,217 data analyst job postings in the United States from Kaggle, covering the time frame 2023–2024. The primary goal was to identify the most valued data science skills in the current job market. By examining the skills mentioned in job descriptions, the project aims to provide an approximate view for data professionals and future data analysts about the industry demands.

The project workflow involved data cleaning, tidying and transformation, visualization, and database storage, along with creating an ER diagram to model the relationships in the dataset for further analysis.

Libraries

Load libraries and packages as necessary.

```
library(tidyverse)
library(DBI)
library(RCurl)
library(stringr)
library(RMySQL)
library(knitr)
```

Import Raw Data The original sources, as stated in the proposal, are found on the following sites: https://www.kaggle.com/datasets/arshkon/linkedin-job-postings/data https://www.kaggle.com/datasets/asaniczka/data-science-job-postings-and-skills/data

Note that not all provided datasets from the original sources were used. As the priority of this project are the job postings and their respective job skills, files not relevant were excluded.

As of now, the relevant datasets are stored in a github repo and accessed/imported as raw github file. Note that the postings.csv from the LinkedIn Data takes considerable time to import using this method. For better performance, suggest downloading the "posting.csv" and using read.csv

```
# "Data Science Postings (2024)" Dataset

DS_job_postings <- getURL("https://media.githubusercontent.com/media/Ryungje/DATA607/refs/heads/main/Pr
  read.csv(text = .)

DS_job_skills <- getURL("https://media.githubusercontent.com/media/Ryungje/DATA607/refs/heads/main/Proj
  read.csv(text = .)


# "LinkedIn Postings (2023-2024)" Dataset

# raw github import version
```

```
# LI_job_postings <- getURL("https://media.githubusercontent.com/media/Ryungje/DATA607/refs/heads/main/
#  read.csv(text = .)

# importing from local machine for performance
LI_job_postings <- read.csv("https://media.githubusercontent.com/media/Ryungje/DATA607/refs/heads/main/

LI_job_skills <- getURL("https://media.githubusercontent.com/media/Ryungje/DATA607/refs/heads/main/Proj
  read.csv(text = .)

LI_skill_codes <- getURL("https://media.githubusercontent.com/media/Ryungje/DATA607/refs/heads/main/Pro
  read.csv(text = .)
```

Data Tidying

A Brief Description of Datasets:

DS_job_postings contains the first set of job postings, with notable information such as *job_link*, *company*, *job_location*, *job_level*, and *job_type*.

DS_job_skills contains the respective job skills for each listing in ***job_posting***. They are related to each other through *job_link*. The skills are presented in a single string, which will require further work to parse each individual skill.

LI_job_postings contains the second set of job postings, with notable information such as *job_id*, *company_name*, *title*, *location*, *remote_allowed*,

LI_job_skills contains the skills with respect to each job presented in ***LI_job_postings***. This dataset is in proper long form, where jobs with multiple skills repeats the *job_id* and posts each skill in *skill_abr*.

LI_skill_codes contains the mappings for *skill_abr* and their respective skill names

General Tidying

Work will first be done to tidy ***DS_job_postings***. We will only keep the selected columns in ***DS_job_postings*** and merging information from ***DS_job_skills***. We will also split the skill strings and, thus, elongate ***DS_job_postings***.

```
# Keep desired columns
DS_job_postings <- DS_job_postings %>%
  select(c(job_link, company, job_location, job_level, job_type))

# Merge data from DS_job_skills into DS_job_postings
DS_job_postings <- left_join(DS_job_postings, DS_job_skills, by = "job_link")

# Convert DS_job_postings to long format
DS_job_postings <- DS_job_postings %>%
  mutate(job_skills = str_split(job_skills, ",")) %>%  # Split skill string into list
  unnest(job_skills) %>%                               # Expand list into rows
  mutate(job_skills = str_trim(job_skills)) %>%        # Trim white space
  mutate(job_skills = str_to_title(job_skills))        # Tidy skills to be capitalized

# Also convert DS_job_skills to long
DS_job_skills <- DS_job_skills %>%
  mutate(job_skills = str_split(job_skills, ",")) %>%  # Split skill string into list
```

```
  unnest(job_skills) %>%                                  # Expand list into rows
  mutate(job_skills = str_trim(job_skills)) %>%           # Trim white space
  mutate(job_skills = str_to_title(job_skills))           # Tidy skills to be capitalized
```

We will then work on tidying ***LI_job_postings***. First, we will edit ***LI_job_skills*** to represent the actual names and not *skill_abr*. Secondly, we will merge the respective skills into ***jobs_postings2*** according to *job_id*. Lastly, we will trim the unneeded columns from ***LI_job_postings***.

```
# Replace the skill_abr with actual names
LI_job_postings <- LI_job_skills %>%
  left_join(LI_skill_codes, by = "skill_abr") %>%
  select(job_id, skill_name)   %>%                  # Keep only job_id and full skill name
  right_join(LI_job_postings, by = "job_id") %>%    # merge into postings
  select(c(job_id, company_name, title, location, remote_allowed, skill_name)) %>%
  rename(job_skills = skill_name)

# Quick column rename
LI_job_skills <- LI_job_skills %>%
  left_join(LI_skill_codes, by = "skill_abr") %>%
  select(job_id, skill_name) %>%
  rename(job_skills = skill_name)
```

Note that not all entries from ***LI_job_skills*** was merged into ***LI_job_postings*** due to the latter not having an associated posting for the former.
The main datasets now are ***DS_job_postings*** and ***LI_job_postings***, with supplementary sets ***DS_job_skills*** and ***LI_job_skills***, respectively.

Data Analysis

Now that the data is tidied, we can proceed with some data analysis.

Bar Graph Function

Lots of bar graph will be made, so a function for repetitive use is now defined.

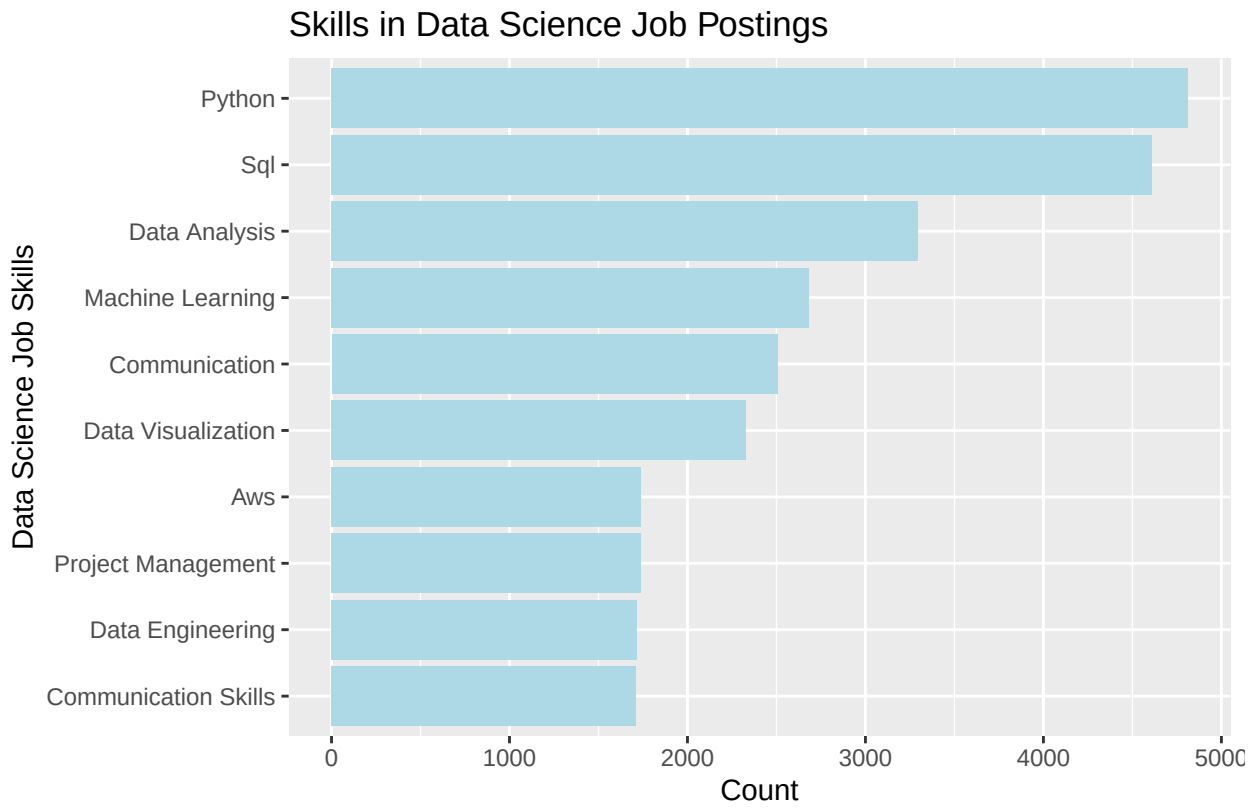
```
bar_graph <- function(data, col, xlab, title_, caption_="", color){
  data %>%
    # Count top ten most frequently used
    count({{col}}, sort = TRUE) %>%
    slice_head(n = 10) %>%

    # Make graph
    ggplot(aes(x = reorder({{col}}, n), y = n)) +
    geom_bar(stat = "identity", fill = color) +
    coord_flip() +

    # Make labels
    labs(x= xlab,
         y = "Count",
         title = title_,
         caption = caption_) +
```

```
    # Caption customization
    theme(
      plot.caption = element_text(
        hjust = 0.5,         # 0 = left, 0.5 = center, 1 = right
        face = "italic",  # style (e.g., italic, bold)
        size = 10
      )
    )
}
```

Quick Graphs

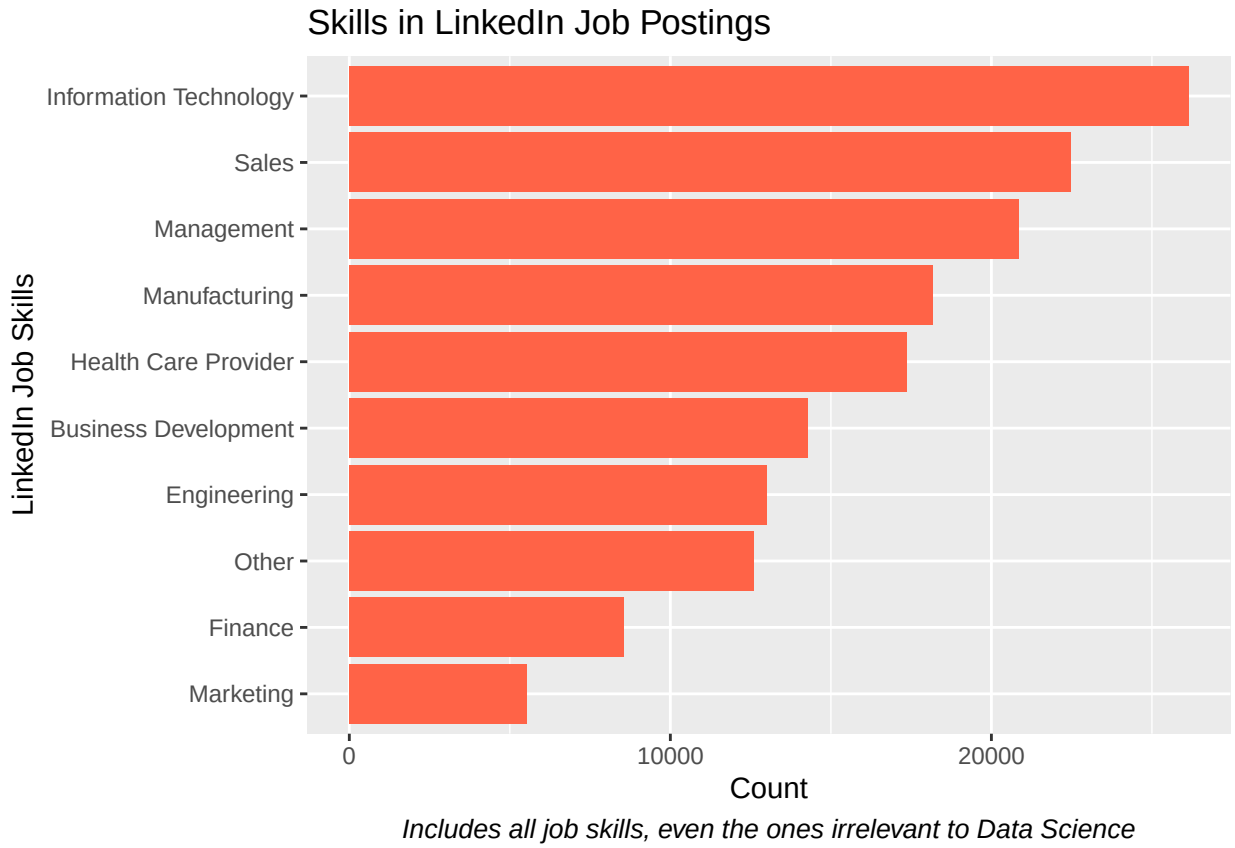
The following two graphs are simple bar graphs which display the most frequently appearing skills from each data set.

```
bar_graph(DS_job_skills, job_skills,
          xlab="Data Science Job Skills",
          title_="Skills in Data Science Job Postings",
          color="lightblue")
```



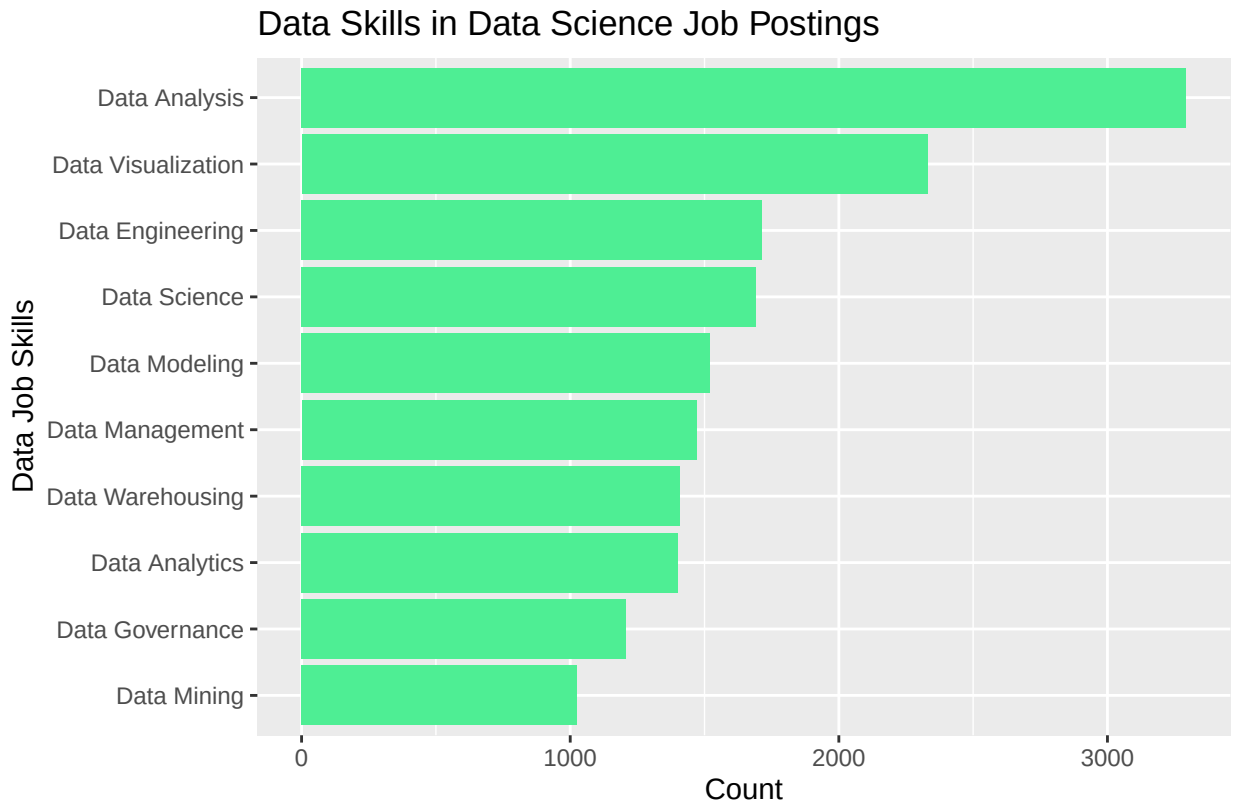

```
# LinkedIn Skills
bar_graph(LI_job_skills, job_skills,
          xlab="LinkedIn Job Skills",
          title_="Skills in LinkedIn Job Postings",
```

```
          caption_="Includes all job skills, even the ones irrelevant to Data Science",
          color="tomato")
```



In this next graph we filter job skills from ***DS_job_postings*** that specifically mention "Data" and present the top ten.

```
DS_job_postings %>%
  filter(str_detect(job_skills, "Data")) %>%

  bar_graph(job_skills,
          xlab="Data Job Skills",
          title_="Data Skills in Data Science Job Postings",
          caption_="Graph showing the job skills in DS_job_postings that specifically mention 'Data'",
          color="seagreen2")
```



Graph showing the job skills in DS_job_postings that specifically mention 'Data'

Semantics Tidying

Data Analysis?

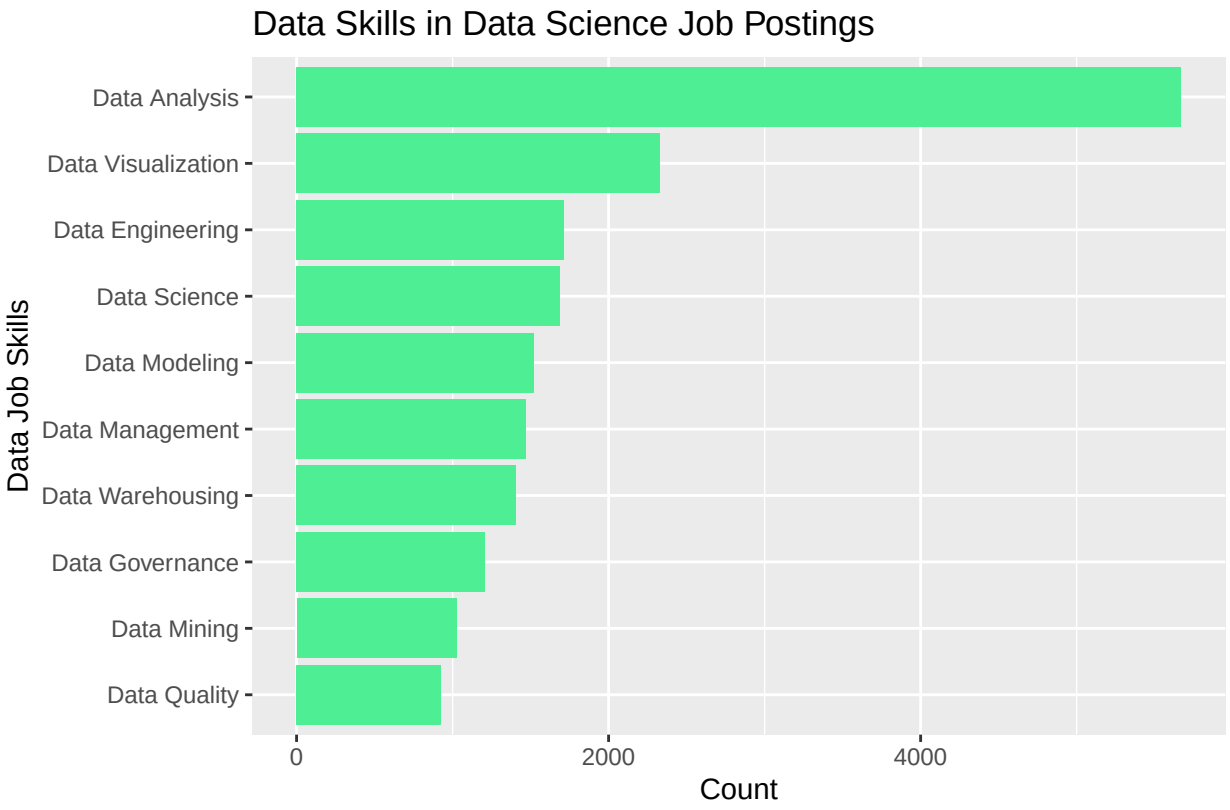
The main problem with ***DS_job_postings*** is that there are many variations for the same job skills. For example, "Data Analyst" and "Data Analysis", some others such as "Health Data Analyst", etc.
Some additional tidying to ***DS_job_postings***. Changing all instances that contains "Data Analyst" or anything similar, to say "Data Analysis." After all, these all mean the same thing: having data analytical skills.

```
DS_job_postings <- DS_job_postings %>%
  mutate(job_skills = case_when(
    str_detect(job_skills,
               regex("Data Analysis|Data Analyst|Data Analytics",
               ignore_case = TRUE)) ~ "Data Analysis",
    TRUE ~ job_skills
  ))
```

Showing the graph again after that one edit.

```
DS_job_postings %>%
  filter(str_detect(job_skills, "Data")) %>%

  bar_graph(job_skills,
```

```
          xlab="Data Job Skills",
          title_="Data Skills in Data Science Job Postings",
          caption_="Data Analysis counts increased nearly by a thousand",
          color="seagreen2")
```



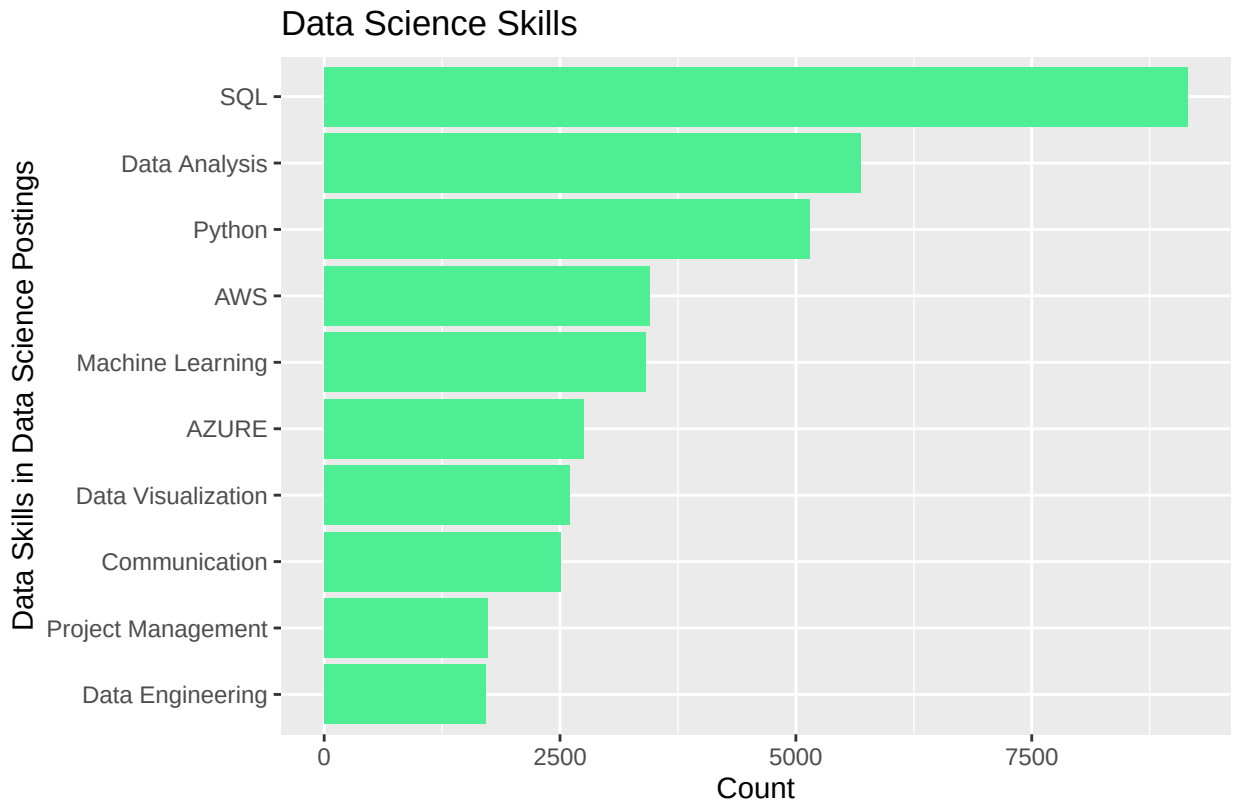
Let's now apply this idea to as many skills possible. Note that, there are some skills that are mentioned together, like "Data Visualization and Analysis" or "Artificial Intelligence/Machine Learning" which were not accounted for. Also note, ML and AI are similar but decided to keep them separate.

```
skill_var <- list(
  c("Data Analysis|Data Analyst|Data Analytics|Data Analytic|Data Analyses|Data Analytical", "Data Analy
  c("Data Visualization|Data Visualisation", "Data Visualization"),
  c("Machine Learning", "Machine Learning"),
  c("Artificial Intelligence|^AI$", "Artificial Intelligence"),
  c("Sql", "SQL"),
  c("Python", "Python"),
  c("C#", "C#"),
  c("C\\+\\+", "C++"),
  c("Aws", "AWS"),
  c("Azure", "AZURE")
  )

for (var in skill_var){
  DS_job_postings <- DS_job_postings %>%
    mutate(job_skills = case_when(
      str_detect(job_skills,
```

```
                  regex(var[1],
                  ignore_case = TRUE)) ~ var[2],
      TRUE ~ job_skills
    ))
}
```

And let's see the graph again. Keep in mind that since we were not able to apply this procedure to all the notable job skills, there may be some bias in terms of what is depicted on the graph. Clearly the bias is towards the more recognizable data science skills. We will also lift the exclusion to only skills that mention "Data"

```
bar_graph(DS_job_postings, job_skills,
          xlab="Data Skills in Data Science Postings",
          title_="Data Science Skills",
          caption_="We can see that SQL takes the lead as the most requested Data Science Skill",
          color="seagreen2"
)
```



We can see that SQL takes the lead as the most requested Data Science Skill

MySQL Database and ER Diagram

```
# connect to MySQL database
ds <- dbConnect(
  RMySQL::MySQL(),
```

```
  dbname = "ds_skills_db",
  host = "localhost",
  port = 3306,
  user = "ruser",
  password = ""
)
```

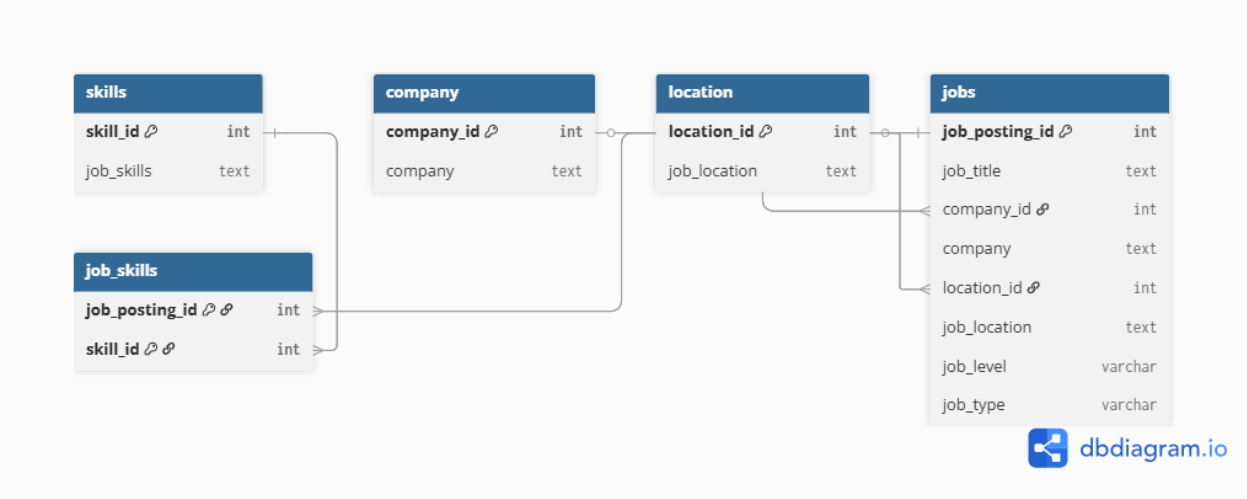
```
dbListTables(ds) #show the tables
```

```
## [1] "company"    "job_skills" "jobs"       "location"   "skills"
```

```
# read all the tables
jobs <- dbReadTable(ds, "jobs")
skills <- dbReadTable(ds, "skills")
company <- dbReadTable(ds, "company")
location <- dbReadTable(ds, "location")
job_skills <- dbReadTable(ds, "job_skills")

head(jobs)
```

```
##   job_posting_id                                  job_title company_id
## 1              1            Senior Machine Learning Engineer          1
## 2              2 Principal Software Engineer  ML Accelerators          2
## 3              3        Senior ETL Data Warehouse Specialist          3
## 4              4  Senior Data Warehouse Developer / Architect          4
## 5              5                          Lead Data Engineer          5
## 6              6                        Senior Data Engineer          6
##                 company location_id      job_location  job_level job_type
## 1     Jobs for Humanity           1     New Haven  CT Mid senior Onsite\r
## 2                Aurora           2 San Francisco  CA Mid senior Onsite\r
## 3    Adame Services LLC           3      New York  NY  Associate Onsite\r
## 4      Morph Enterprise           4    Harrisburg  PA Mid senior Onsite\r
## 5                  Dice           5         Plano  TX Mid senior Onsite\r
## 6 University of Chicago           6       Chicago  IL Mid senior Onsite\r
```

```
# https://github.com/vincent-usny/pro3/blob/main/ds.png
# needs to be downloaded and imported
include_graphics("C:/Users/vincent/OneDrive/Desktop/pro3/ds.png")
```




```
# jobs to company : many to one
# jobs to location : many to one
# job_skills to jobs: many to one
```

Scracth Code and Notes

For ***DS_job_skills*** (and the corresponding column in ***DS_job_postings***) Dataset, the listed skills are messy and there are semantic differences. For example, "Data Analysis" and "Data Analyst" appear separately, but mean the same thing.

Note that AWS and SQL are also left as Aws and Sql, respectively. Doesn't seem like there are any instances of AWS and SQL, or other variations, in the column so it should be okay.

Some problems with the way I did the type matching [the line with skill_var <- list(c("Data Analysis...]. Any skill that had a combination of skills were at risk of being classified as only one (whichever I decided to match for first in that list). Maybe this is why C++ had so many counts?

Summary

The project was processed in multiple stages:

1. Collected 12,217 job postings for data analyst roles from Kaggle, including details such as job title, company, location, job skills, job type, etc.
2. Handled missing values, transformed wide formats to long ones, and extracted skills from job descriptions.
3. Created charts to present the results in a visible way.
4. Analyzed the most frequent mentioned data skills in 2023-2024.
5. Stored the dataset in a MySQL database for structured querying and design an ER diagram to show the relationships between each entity.