Project 3

Matthew Roland, Jean Jimenez, Kelly Eng

2023-10-17

Team Android Project #3

Introduction

The advent of data science has created a paradigm shift in how we approach problem-solving across various industries. Data science skills are increasingly becoming indispensable, but what skills are most valued in this domain? Our class was tasked with a project to explore this very question.

Our objective is to identify the most valued skills in data science and to collaborate effectively as a virtual team and share our findings in a presentation.

We used R to automate the process of data collection, transformation, and analysis.

We used various R packages like `tidyverse`, `jsonlite`, `httr`, `rvest`, and `ggplot2` for web scraping, data manipulation, and visualization.

We utilized Google Custom Search Engine API to gather data from top websites related to data science skills.

We stored the acquired data in a normalized relational database. We then conducted exploratory data analysis to classify and rank the skills based on their frequency.

Search Query

To find the top skills in data science, we used google search api to search sites for "Top Data Science Skills. An API key (`api_key`) is specified, which is used for authenticating requests to Google's API services. A query string (`query`) is defined as"top data science skills", which serves as the search keyword for the Custom Search Engine. A Custom Search Engine ID (`CSE`) is provided, which is unique to the user's programmable search engine created via Google's API. The `GET()` function is used to send an HTTP GET request to the Google API. The request is constructed with the API key, Custom Search Engine ID, and the query string as parameters. The API's response is stored in the variable `response`, which is in JSON format. The `fromJSON()` function is used to parse this JSON data into an R data structure, stored in the variable `results`. Finally, the `items` field from the `results` is extracted into the variable `search`, which contains the list of websites that match the query "top data science skills".

Web Scraping

Web scraping the top 10 sites.

Instead of directly pulling from the search API, a CSV file containing the search results is used. This is done to maintain consistency in the data, as search results can change each time the query is run. The initial search results were saved to a CSV file using the `write.csv` function. This ensures that everyone involved in the project is working with the same set of data, making the process reproducible without errors.

We first start by reading the CSV file from a GitHub URL using `read.csv()`. The `lapply()` function combined with `read_html()` is then used to scrape HTML data from each of these sites. This HTML data is stored in a list named `html_data`.

Note, some site accesses were denied. Therefore, we edited the search result links to wayback machine archived linked in order to access them.

Lastly, a function called `remove_numeric_prefix_whitespaces` is made to clean up the scraped data by removing any numeric prefixes and trailing white spaces from a given vector of strings.

```
links_to_site=read.csv(url("https://raw.githubusercontent.com/Mattr5541/DATA_607_Project_3/main/links_t

actual_links_to_site=links_to_site$x
html_data <- lapply(actual_links_to_site, read_html)


# The html data scraped from each site will be stored here
# It's only the first 9 because we get an access denied to the tenth website
# Fortunately we can use waybackmachine for the tenth website
# Google's algorithm is always updating the order so we don't know the exact order it'll be on a diffre

# html_data <- lapply(links_to_site[1:9], read_html)


# Function to remove numeric prefixes if there is any, the : post fix along with trailing white spaces
remove_numeric_prefix_whitespaces <- function(input_vector) {
  cleaned_vector <- gsub("^\\d+\\)|\\d+\\.\\s*|:$", "", input_vector)
  cleaned_vector <- trimws(cleaned_vector)
  return(cleaned_vector)
}
```

Each website was unique in the way the data was set up. To retrieve them, we manually coded scraping from each of the website. To do this, we looked at the structure of the html data and coded accordingly to extract the appropriate data.

Scraping data from the first website

```
first_site <- html_data[[1]]


# Technical skills are in an un-ordered list in this site
scraped_li_from_first_site <- first_site |>
  html_elements("article ul li") |>
  html_text2()
# The first two on the lists are not part of it, so remove them & keep the rest
technical_skills_from_first_site <- scraped_li_from_first_site[3:12]


# For the first website, the top 20+ skills are in the h3 tag nested inside an article tag after the ab
scraped_h3_from_first_site <- first_site |>
  html_elements("article h3") |>
  html_text2()

scraped_h3_from_first_site <- remove_numeric_prefix_whitespaces(scraped_h3_from_first_site[1:24])

# From the scraped h3, the first 17 are technical skills, while the rest are non-technical
technical_skills_from_first_site <- c(technical_skills_from_first_site, scraped_h3_from_first_site[1:17]
non_technical_skills_from_first_site <- scraped_h3_from_first_site[18:24]
```

```
technical_skills_from_first_site
```

```
##  [1] "Statistical analysis and computing"
##  [2] "Machine Learning"
##  [3] "Deep Learning"
##  [4] "Processing large data sets"
##  [5] "Data Visualization"
##  [6] "Data Wrangling"
##  [7] "Mathematics"
##  [8] "Programming"
##  [9] "Statistics"
## [10] "Big Data"
## [11] "Programming"
## [12] "Knowledge of SAS and Other Analytical Tools"
## [13] "Adept at Working with Unstructured Data"
## [14] "Web Scraping"
## [15] "ML with AI and DL with NLP"
## [16] "Problem-Solving Skills"
## [17] "Probability and Statistics"
## [18] "Multivariate Calculus and Linear Algebra"
## [19] "Database Management"
## [20] "Cloud Computing"
## [21] "Microsoft Excel"
## [22] "DevOps"
## [23] "Data Extraction, Transformation, and Loading"
## [24] "Business Intelligence"
## [25] "Neural Networks"
## [26] "Model Deployment"
## [27] "Data Structures and Algorithms"
```

```
non_technical_skills_from_first_site
```

```
## [1] "A Strong Business Acumen"    "Strong Communication Skills"
## [3] "Great Data Intuition"        "Analytical Mindset"
## [5] "\"Out-of-the-Box" Thinking"  "Critical Thinking"
## [7] "Decision Making"
```

Scraping Data from the Second Site:

```
# Unfortunately it requires an account to view the article, only thing you can get is the first skill
second_site <- read_html("https://towardsdatascience.com/top-10-skills-for-a-data-scientist-in-2020-2b8

second_site |>
  html_elements("h1") |>
  html_text2()
```

```
## [1] "Top 10 Skills for a Data Scientist" "1. Probability & Statistics"
```

```
# Fortunately, you can view the whole thing in wayback machine, the csv file contains the waybackmachin
second_site <- html_data[[2]]

top_10_skills_from_second_site <- second_site |>
  html_elements("h1") |>
  html_text2()
```

```r
# The top 10 skills are numbered so we need to remove the prefixes, it also starts in the second h1 tag
top_10_skills_from_second_site <- remove_numeric_prefix_whitespaces(top_10_skills_from_second_site[2:11])

top_10_skills_from_second_site
```

```
##  [1] "Probability & Statistics"
##  [2] "Multivariable Calculus & Linear Algebra"
##  [3] "Programming Skills"
##  [4] "Data Wrangling"
##  [5] "Database Management"
##  [6] "Data Visualization"
##  [7] "Machine Learning / Deep Learning"
##  [8] "Cloud Computing"
##  [9] "Microsoft Excel"
## [10] "DevOps"
```

Scraping Data from Third Website

```r
third_site <- html_data[[3]]

# The skills are in the h2 tag in this case
top_7_skills_from_third_site <- third_site |>
  html_elements("h2") |>
  html_text2()

# Remove the numeric prefixes
top_7_skills_from_third_site <- remove_numeric_prefix_whitespaces(top_7_skills_from_third_site)

top_7_skills_from_third_site
```

```
## [1] "It all Starts With the Basics - Programming Language + Database"
## [2] "Mathematics"
## [3] "Data Analysis & Visualization"
## [4] "Web Scraping"
## [5] "ML with AI & DL with NLP"
## [6] "Big Data"
## [7] "Problem-Solving Skill"
```

Scraping Data from the Fourth Website

```r
fourth_site <- html_data[[4]]

top_7_skills_from_fourth_site <- fourth_site |>
  html_elements("h3") |>
  html_text2()


top_7_skills_from_fourth_site <- top_7_skills_from_fourth_site[c(1, 3, 5, 7, 9, 11, 12)]
top_7_skills_from_fourth_site <- remove_numeric_prefix_whitespaces(top_7_skills_from_fourth_site)

top_7_skills_from_fourth_site
```

```
## [1] "Programming"
## [2] "Statistics and probability"
```

```
## [3] "Data wrangling and database management"
## [4] "Machine learning and deep learning"
## [5] "Data visualization"
## [6] "Cloud computing"
## [7] "Interpersonal skills"
```

Scraping Data from the Fifth Website

```
fifth_site <- html_data[[5]]

# This website only contains 15 skills divided into technical and non-technical
# No need to process it because there's no numeric prefixes
top_15_skills_from_fifth_site <- fifth_site |>
  html_elements("span.listed-menu h3") |>
  html_text2()

top_15_skills_from_fifth_site
```

```
##  [1] "Python Skills"                      "R Skills"
##  [3] "Statistics and Math Skills"         "SQL Skills "
##  [5] "NoSQL Skills"                       "Data Visualization Skills"
##  [7] "Machine Learning Skills "           "Deep Learning Skills"
##  [9] "Natural Language Processing Skills" "Big Data Skills"
## [11] "Cloud Computing Skills"             "Business Acumen"
## [13] "Communication Skills"               "Data Ethics Skills"
## [15] "Environmental Awareness"
```

Scraping Data from the Sixth Website

```
sixth_site <- html_data[[6]]

top_16_skills_from_sixth_site <- sixth_site |>
  html_elements('h3.wp-block-heading') |>
  html_text2()

# The stuff after the 16th skill is just telling people to go to their boot camp
top_16_skills_from_sixth_site <- top_16_skills_from_sixth_site[1:16]

top_16_skills_from_sixth_site
```

```
##  [1] "Programming Languages"
##  [2] "Mathematics, Statistical Analysis, and Probability"
##  [3] "Data Mining"
##  [4] "Machine Learning and AI"
##  [5] "Familiarity With Hadoop"
##  [6] "Data Visualization"
##  [7] "Business Strategy"
##  [8] "Cloud Computing"
##  [9] "Communication"
## [10] "Analytical Mindset"
## [11] ""Out-of-the-Box" Thinking"
## [12] "Decision Making"
## [13] "Collaboration"
## [14] "Storytelling"
```

```
## [15] "Attention to Detail"
## [16] "Continuous Learning"
```

Scraping Data from the Seventh Website

```
seventh_site <- read_html("https://www.reddit.com/r/datascience/comments/y78uss/what_technologiesskills_

# This is only letting us get things from the person that made the post on reddit & not anything from t
seventh_site |>
  html_elements('p') |>
  html_text2()
```

```
## [1] "A space for data science practitioners and professionals to engage in discussions and debates or
## [2] "OK, I know this question will make a lot of you mad. I can see the replies now: Data science mea
## [3] "Why am I asking this? In full candor, I feel like I am seriously stagnating. I am really the onl
## [4] "Many thanks in advance!"
```

```
# We have to use the old version of reddit in order to scrape results from the comments, that's why we
seventh_site <- html_data[[7]]

top_skills_with_li_tag_on_reddit <- seventh_site |>
  html_elements('li') |>
  html_text2()

# The website uses a lot of lists to access miscellaneous links to other parts of the website, what we'
top_skills_from_seventh_site <- top_skills_with_li_tag_on_reddit[c(60:68, 169:172)]

top_skills_from_seventh_site
```

```
##  [1] "Python for DS (the usuals pandas plotting etc)"
##  [2] "Modeling skills for both ML applications and data reporting."
##  [3] "SQL at least basics but by year 3 should be quite proficient with at least pulling data."
##  [4] "XGBoost, LightGBM and the ability to use them for tabular data."
##  [5] "If NLP is needed get familiar with modern transformers quickly as they are the standard now."
##  [6] "Tableau or PowerBI. It might be for data analysts but they work and business people love this
##  [7] "Familiarity with DS front ends like Streamlit or Gradio to prototype solutions."
##  [8] "How to make great PowerPoints"
##  [9] "How to ask the right question of the right people."
## [10] "Some cloud experience with a platform like AWS, Azure, etc. Ability to productionalize models
## [11] "Scripting ability in Python or similar and a basic understanding of good ML ops practices: git
## [12] "Proficient SQL, basically the ability to write whatever select statements to pull whatever data
## [13] "Ability to reason about business problems, communicating to stakeholders, focusing on what's in
```

Scraping Data from the Eighth Website

```
eighth_site <- html_data[[8]]

top_10_skills_from_eighth_site <- eighth_site |>
  html_elements('h4') |>
  html_text2()

# Remove the numeric prefixes
top_10_skills_from_eighth_site <- remove_numeric_prefix_whitespaces(top_10_skills_from_eighth_site[1:10]

top_10_skills_from_eighth_site
```

```
##  [1] "Python and R"                "Hadoop"
##  [3] "NoSQL"                       "Machine Learning"
##  [5] "Data Visualization Tools"    "Probability and Statistics"
##  [7] "3 C's-"                      "Innovation"
##  [9] "Data Intuition"              "Business Expertise"
```

Scraping Data from the Ninth Website

```
ninth_site <- html_data[[9]]

top_10_skills_from_ninth_site <- ninth_site |>
  html_elements('h2') |>
  html_text2()

top_10_skills_from_ninth_site <- remove_numeric_prefix_whitespaces(top_10_skills_from_ninth_site[1:10])

top_10_skills_from_ninth_site
```

```
##  [1] "Critical thinking"
##  [2] "Effective communication"
##  [3] "Proactive problem solving"
##  [4] "Intellectual curiosity"
##  [5] "Business sense"
##  [6] "Ability to prepare data for effective analysis"
##  [7] "Ability to make use of self-service analytics platforms"
##  [8] "Ability to write efficient and maintainable code"
##  [9] "Ability to apply maths and statistics appropriately"
## [10] "Ability to make use of machine learning and artificial intelligence (AI)"
```

Scraping Data from the Tenth Website

```
# Access is denied so we used waybackmachine version to access the top skills from the site
tenth_site <- html_data[[10]]

top_skills_from_tenth_site <- tenth_site |>
  html_elements('h3') |>
  html_text2()

top_skills_from_tenth_site <- remove_numeric_prefix_whitespaces(top_skills_from_tenth_site)

top_skills_from_tenth_site
```

```
##  [1] "Cloud computing"             "Statistics and probability"
##  [3] "Advanced mathematics"        "Machine learning"
##  [5] "Data visualization tools"    "Query languages"
##  [7] "Database management"         "Python coding"
##  [9] "Microsoft Excel"             "R programming"
## [11] "Data wrangling"              "Independence"
## [13] "Communication"               "Project management"
## [15] "Analytical thinking"
```

Cleaning Data

The data cleaning phase was an important step in refining the raw data we scraped from various websites. We began by aggregating the lists of skills from each source into a single collection, labeled `all_top_skills`. To make sure everything was uniform and to keep data analysis consistent, all skill names were converted to lowercase and any special characters were removed, making a new dataset called `cleaned_top_skills`.

To categorize the skills in a meaningful way, we defined a standard set of skill categories, which include "Statistical Analysis," "Machine Learning & AI," and "Data Visualization," among others after taking a look at `cleaned_top_skills`. These predefined categories acted as a guideline for mapping the cleaned skills. We then initiated an empty list, `mapped_skills`, to store these categorized skills.

A loop was run through each skill in `cleaned_top_skills`, using regex matching to classify the skill into one of the predefined categories. For example, skills containing terms like "statistics" or "probability" were mapped to the "Statistical Analysis" category.

After this categorization, we calculated the frequency of each skill category to understand its prevalence in the data science industry. This was stored in a table called `aggregated_skills`, which was then sorted in descending order based on frequency to create `sorted_skills_df`.

```
#combining all top skills

all_top_skills= c(technical_skills_from_first_site,top_10_skills_from_second_site,top_7_skills_from_thi


# Convert to lowercase and remove special chars
cleaned_top_skills = tolower(gsub("[^[:alnum:] ]", "", all_top_skills))

#predefined list of skills
standard_skills = c("Statistical Analysis",
                    "Machine Learning & AI",
                    "Data Processing & Wrangling",
                    "Data Visualization",
                    "Programming & Coding",
                    "Big Data & Cloud Computing",
                    "Software Tools & Technologies",
                    "Business & Communication Skills",
                    "Problem Solving & Critical Thinking",
                    "Mathematics & Computational Skills",
                    "Data Management & Transformation",
                    "Modeling & Algorithms",
                    "Soft Skills",
                    "Specialized Technologies")

# empty list
mapped_skills <- list()

# go through each skill
for(skill in cleaned_top_skills) {
  skill_lower = tolower(skill)

  if (grepl("statistics|probability", skill_lower)) {
    mapped_skills = append(mapped_skills, "Statistical Analysis")
  } else if (grepl("machine learning|deep learning|neural networks|ai|nlp", skill_lower)) {
    mapped_skills = append(mapped_skills, "Machine Learning & AI")
  } else if (grepl("data wrangling|data processing|data extraction|database management", skill_lower))
```

```
    mapped_skills = append(mapped_skills, "Data Processing & Wrangling")
  } else if (grepl("data visualization|visualization tools", skill_lower)) {
    mapped_skills = append(mapped_skills, "Data Visualization")
  } else if (grepl("programming|coding|software engineering|devops|web scraping", skill_lower)) {
    mapped_skills = append(mapped_skills, "Programming & Coding")
  } else if (grepl("big data|cloud computing|hadoop|nosql|sql|data mining", skill_lower)) {
    mapped_skills = append(mapped_skills, "Big Data & Cloud Computing")
  } else if (grepl("excel|sas|tools|technologies", skill_lower)) {
    mapped_skills = append(mapped_skills, "Software Tools & Technologies")
  } else if (grepl("business|communication|storytelling|collaboration|decision making|strategy", skill_l
    mapped_skills = append(mapped_skills, "Business & Communication Skills")
  } else if (grepl("problem solving|critical thinking|analytical|intellectual curiosity|attention to de
    mapped_skills = append(mapped_skills, "Problem Solving & Critical Thinking")
  } else if (grepl("multivariate calculus|linear algebra|mathematics", skill_lower)) {
    mapped_skills = append(mapped_skills, "Mathematics & Computational Skills")
  } else if (grepl("data wrangling|data extraction|database management|data management|transformation",
    mapped_skills = append(mapped_skills, "Data Management & Transformation")
  } else if (grepl("model deployment|neural networks|algorithms", skill_lower)) {
    mapped_skills = append(mapped_skills, "Modeling & Algorithms")
  } else if (grepl("soft skills|interpersonal|communication", skill_lower)) {
    mapped_skills = append(mapped_skills, "Soft Skills")
  } else if (grepl("sas|devops|hadoop|specialized tools|tableau|powerbi|docker", skill_lower)) {
    mapped_skills = append(mapped_skills, "Specialized Technologies")
  } else {
    mapped_skills = append(mapped_skills, "Other")
  }
}


aggregated_skills = table(unlist(mapped_skills))
sorted_skills = sort(aggregated_skills, decreasing = TRUE)
sorted_skills_df = as.data.frame(sorted_skills)
colnames(sorted_skills_df) = c("Skill Category", "Frequency")
```

Data Visualization

After the data cleaning process, we moved on to the data visualization. Initially, we used the `kable` function to display the aggregated and sorted skill categories in a tabular form. However, to offer a more intuitive understanding, we employed `ggplot2` to create a horizontal bar chart, sorting the skill categories based on their frequency of occurrence.

In addition to the bar chart, we used a word cloud to emphasize the most frequent skills visually. The word cloud allowed us to represent the skill categories as words displayed in varying sizes, which directly correlate with their frequency. This provided another angle of understanding the distribution of important skills in a less structured but more visually impactful way.

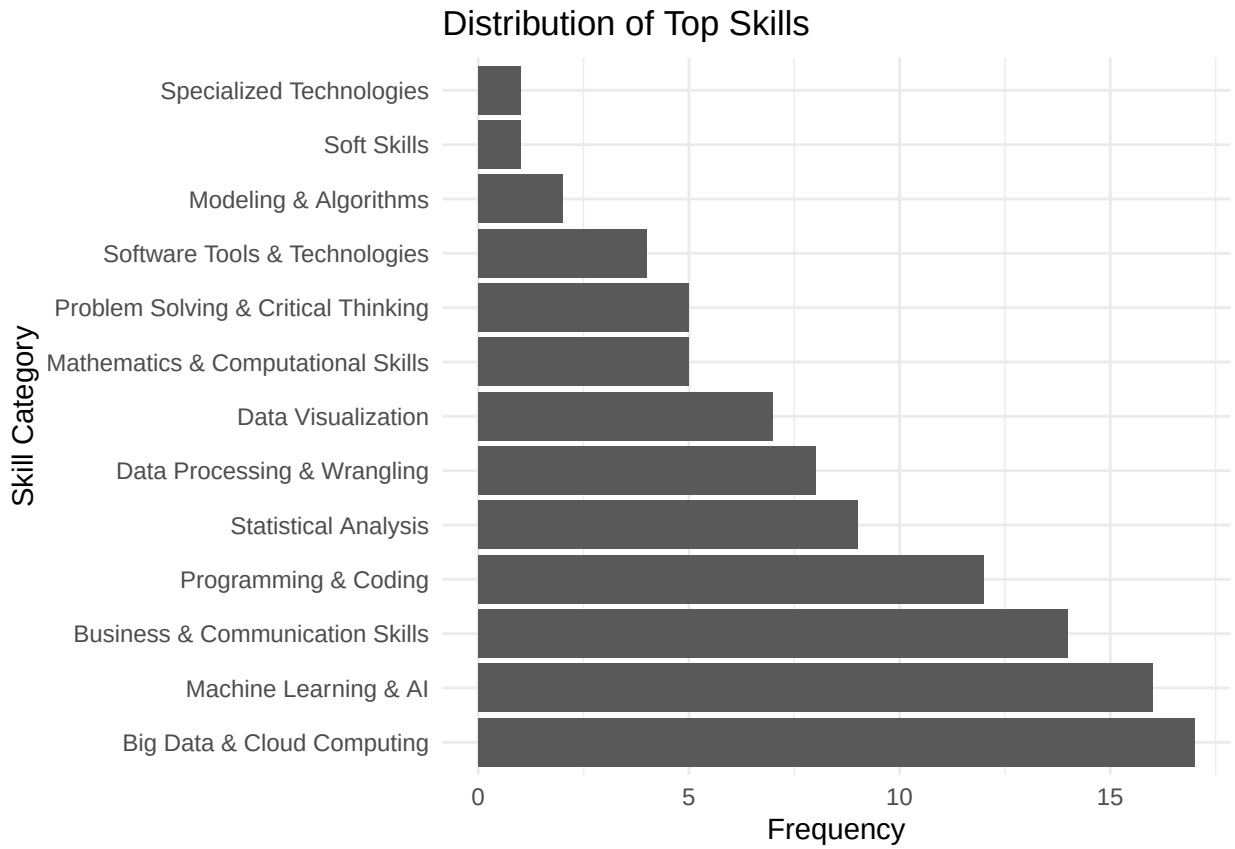
Finally, we wanted to understand the proportion of each skill category in the entire dataset. We calculated these proportions and displayed them in two ways: a pie chart and another bar plot. The pie chart provided an immediate understanding of how each skill category compared to the whole, while the bar plot offered a similar perspective in a different visual form.

```
kable(sorted_skills_df, caption = "Aggregated Skills Sorted in Descending Order")
```

Table 1: Aggregated Skills Sorted in Descending Order

Skill Category	Frequency
Other	29
Big Data & Cloud Computing	17
Machine Learning & AI	16
Business & Communication Skills	14
Programming & Coding	12
Statistical Analysis	9
Data Processing & Wrangling	8
Data Visualization	7
Mathematics & Computational Skills	5
Problem Solving & Critical Thinking	5
Software Tools & Technologies	4
Modeling & Algorithms	2
Soft Skills	1
Specialized Technologies	1

```
sorted_skills_df_filtered = subset(sorted_skills_df, sorted_skills_df$`Skill Category` != "Other")

ggplot(sorted_skills_df_filtered, aes(x=reorder(`Skill Category`, -Frequency), y=Frequency)) +
  geom_bar(stat="identity") +
  coord_flip() +
  labs(title="Distribution of Top Skills",
       x="Skill Category",
       y="Frequency") +
  theme_minimal()
```




```
sorted_skills_df = sorted_skills_df[sorted_skills_df$`Skill Category` != "Other", ]


sorted_skills_df = sorted_skills_df[!is.na(sorted_skills_df$Frequency) & sorted_skills_df$Frequency > 0
sorted_skills_df$Frequency = as.numeric(sorted_skills_df$Frequency)


colnames(sorted_skills_df) = c("Skill Category", "Frequency")



wordcloud(words = sorted_skills_df[["Skill Category"]],
          freq = sorted_skills_df$Frequency,
          min.freq = 2,
          max.words = 100,
          random.order = FALSE,
          rot.per = 0.2,
          scale = c(2, 0.2),  # Adjust the scale if needed
          colors = brewer.pal(8, "Dark2"),
          vfont = c("sans serif", "plain"))
```

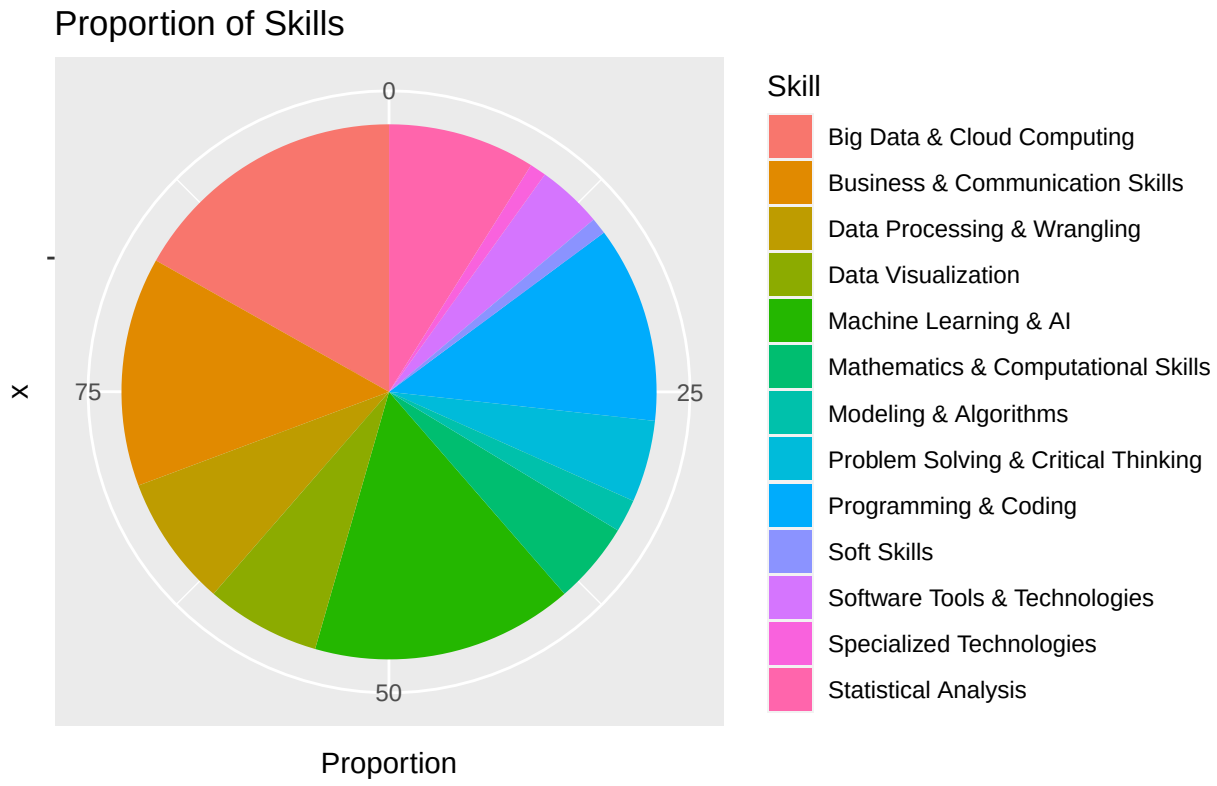
```
## Warning in wordcloud(words = sorted_skills_df[["Skill Category"]], freq =
## sorted_skills_df$Frequency, : Big Data & Cloud Computing could not be fit on
## page. It will not be plotted.
```

```
## Warning in wordcloud(words = sorted_skills_df[["Skill Category"]], freq =
## sorted_skills_df$Frequency, : Business & Communication Skills could not be fit
```

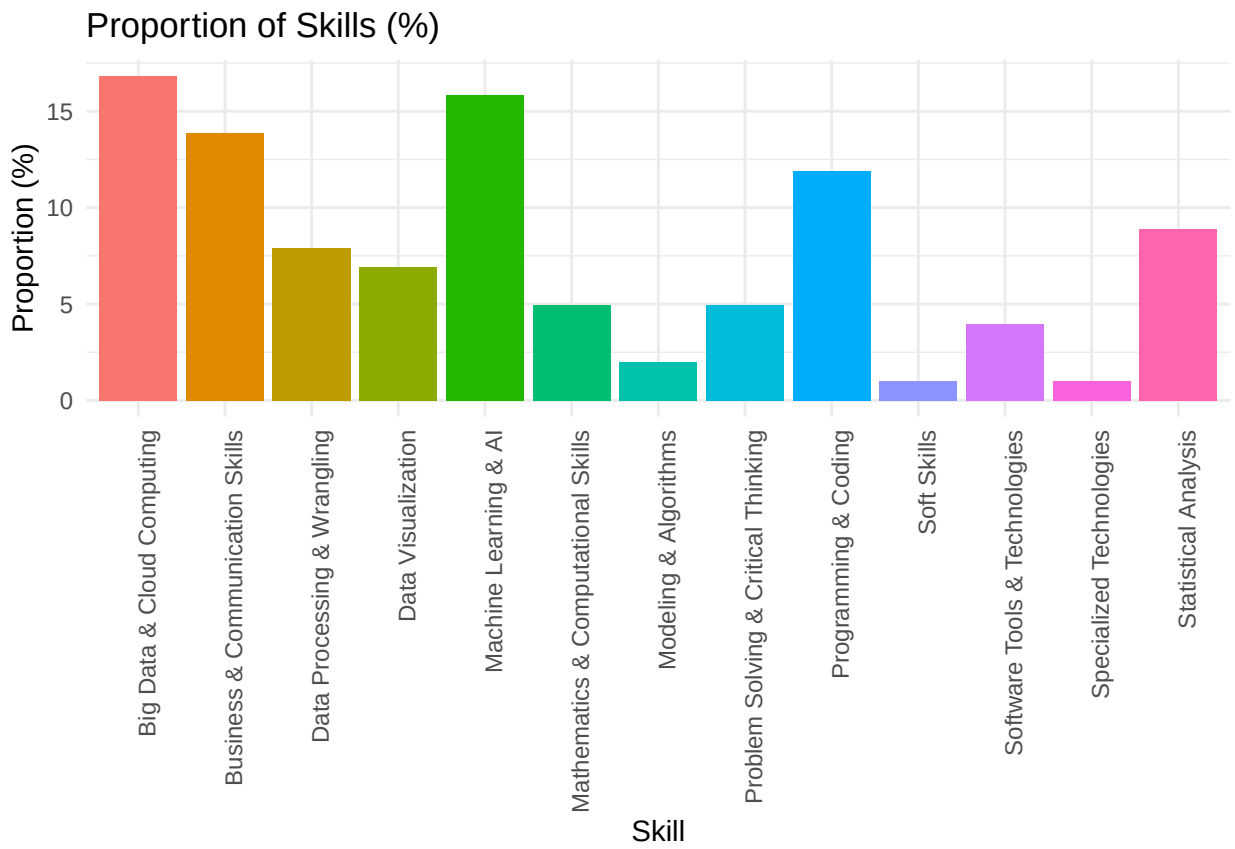
```
## on page. It will not be plotted.
```


```
#rmv other
if ('Other' %in% names(sorted_skills)) {
  sorted_skills = sorted_skills[names(sorted_skills) != 'Other']
}

#total occrences not other
total_occurrences = sum(sorted_skills)

#proportion to total
proportion_skills = round((sorted_skills / total_occurrences) * 100, 2)


new_table = data.frame("Skill" = names(proportion_skills), "Proportion (%)" = as.vector(proportion_skil

proportion_skills_tibble = tibble(Skill = names(proportion_skills), Proportion = as.vector(proportion_sk


filtered_proportion_skills_tibble = proportion_skills_tibble %>% filter(Skill != "Other")


# pie(filtered_proportion_skills_tibble$Proportion,
#     labels = paste(filtered_proportion_skills_tibble$Skill, "\n", filtered_proportion_skills_tibble$P
#     main = "Proportion of Skills",
#     col = rainbow(nrow(filtered_proportion_skills_tibble)),
#     radius = 1,
```

```
#     cex = 0.7)

ggplot(filtered_proportion_skills_tibble, aes(x = "", y = Proportion, fill = Skill)) +
  geom_bar(stat = "identity") +
  coord_polar(theta = "y") +
  labs(title = "Proportion of Skills")
```




```
#bar plot

# barplot(filtered_proportion_skills_tibble$Proportion,
#         names.arg = filtered_proportion_skills_tibble$Skill,
#         las = 2,
#         main = "Proportion of Skills (%)",
#         ylab = "Proportion (%)",
#         xlab = "Skills",
#         col = rainbow(nrow(filtered_proportion_skills_tibble)))

ggplot(filtered_proportion_skills_tibble, aes(x = Skill, y = Proportion, fill = Skill)) +
  geom_bar(stat = "identity") +
  labs(title = "Proportion of Skills (%)", y = "Proportion (%)") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 90, hjust = 1), legend.position = "none")
```



Summary and Conclusion

```
cat("------------------------------------------------------------\n")
```

```
## ------------------------------------------------------------
```

```
cat("Summary and Conclusions Based on the Skill Distribution\n")
```

```
## Summary and Conclusions Based on the Skill Distribution
```

```
cat("------------------------------------------------------------\n")
```

```
## ------------------------------------------------------------
```

```
#calc prop
proportion_skills = round((sorted_skills / total_occurrences) * 100, 2)

#sort
sorted_skills = sort(sorted_skills, decreasing = TRUE)

#top 3
top_skills = head(sorted_skills, 3)
cat("Top 3 Most Frequent Skills:\n")
```

```
## Top 3 Most Frequent Skills:
```

```
for(i in 1:3) {
  skill_name = names(top_skills)[i]
```

```
  skill_freq = top_skills[i]
  skill_prop = proportion_skills[i]
  cat(paste(i, ". ", skill_name, "(", skill_freq, "occurrences, ", skill_prop, "%)", "\n"))
}
```

```
## 1 .  Big Data & Cloud Computing ( 17 occurrences,  16.83 %)
## 2 .  Machine Learning & AI ( 16 occurrences,  15.84 %)
## 3 .  Business & Communication Skills ( 14 occurrences,  13.86 %)
```

```
#least freq
least_skills = tail(sorted_skills, 3)
least_prop=tail(proportion_skills,3)
cat("\nLeast Frequent Skills:\n")
```

```
##
## Least Frequent Skills:
```

```
for(i in 1:3) {
  skill_name = names(least_skills)[i]
  skill_freq = least_skills[i]
  skill_prop = least_prop[i]
  cat(paste(i, ". ", skill_name, "(", skill_freq, "occurrences, ", skill_prop, "%)", "\n"))
}
```

```
## 1 .  Modeling & Algorithms ( 2 occurrences,  1.98 %)
## 2 .  Soft Skills ( 1 occurrences,  0.99 %)
## 3 .  Specialized Technologies ( 1 occurrences,  0.99 %)
```

```
cat("\nThere were",total_occurrences, "skills that were sorted into categories\n")
```

```
##
## There were 101 skills that were sorted into categories
```

```
cat("\n-------------------------------------------------------------\n")
```

```
##
## -------------------------------------------------------------
```

```
# General Conclusions
cat("General Conclusions:\n")
```

```
## General Conclusions:
```

```
cat("1. Big Data & Cloud Computing has overtaken other skills in frequency, signaling a growing emphasis
```

```
## 1. Big Data & Cloud Computing has overtaken other skills in frequency, signaling a growing emphasis
```

```
cat("2. Machine Learning & AI closely follows in second, indicating a sustained, high demand for skills
```

```
## 2. Machine Learning & AI closely follows in second, indicating a sustained, high demand for skills in
```

```
cat("3. Business & Communication Skills also feature prominently, emphasizing the importance of not just
```

```
## 3. Business & Communication Skills also feature prominently, emphasizing the importance of not just
```

```
cat("4. Skills like Modeling & Algorithms, Soft Skills, and Specialized Technologies are least frequent
```

```
## 4. Skills like Modeling & Algorithms, Soft Skills, and Specialized Technologies are least frequent.
```

```
cat("-------------------------------------------------------------\n")
```

```
## -------------------------------------------------------------
```

Future Directions

While our project provides valuable insights into the most sought-after skills in data science, it's important to acknowledge some limitations and areas for future improvement. One notable issue is the inefficiency in data collection and categorization. Our methodology led to a large number of uncategorized skills falling into the 'Other' category. This is because websites use various frameworks and language to list data science skills, making it challenging to automate the categorization process effectively.

For future work, implementing advanced natural language processing techniques, such as using GPT-4 or other Large Language Models (LLMs), could significantly improve data extraction and categorization. These models can analyze the HTML data from each site, extract the skills listed, and categorize them effectively, reducing the need for manual intervention. This approach would also make the project less dependent on the structure of individual websites, mitigating the issue we faced of websites changing their structure and thereby breaking our code. In fact, using archived pages would become less necessary as LLMs could adapt to changes in website structure.

Another avenue for improvement is sourcing data from more consistent platforms or databases, perhaps academic publications or industry reports that have already aggregated this type of information. By doing so, we could reduce the variability introduced by using multiple websites with differing structures and terminologies.

In conclusion, future directions could involve leveraging advanced machine learning models for more efficient and accurate data extraction and categorization, as well as diversifying the data sources to include more standardized and reliable platforms. Both these improvements could make our analysis more robust, comprehensive, and adaptable to changes over time.