Data Science Salary Analysis

Sue Susman, MEd, BSN, RN

2023-07-05

**Table of Contents**

[Project Overview: 1](#_Toc139413881)

# Project Overview:

The CEO of the small, but rapidly expanding company I work for is interested in hiring a full-time data scientist. While the company is located in the US, the position can be remote for the right person. My research will include salary differences between hiring someone in the US vs. someone outside of the US. I will analyze data to find the salary distribution of employee residence, job title, and experience level, among other factors in order to determine a reasonable salary range that will allow us to hire the best talent. Factors to consider include rising salaries in the recession, along with a highly competitive job market.

options(repos = "https://cran.rstudio.com/")

install.packages("tidyverse")  
## The downloaded binary packages are in  
## /var/folders/7w/vf\_3wx5d6kq9bm2666hl7yy80000gn/T//RtmpJH2Hox/downloaded\_packages

install.packages("ggwordcloud")  
## The downloaded binary packages are in  
## /var/folders/7w/vf\_3wx5d6kq9bm2666hl7yy80000gn/T//RtmpJH2Hox/downloaded\_packages

install.packages("viridis")  
## The downloaded binary packages are in  
## /var/folders/7w/vf\_3wx5d6kq9bm2666hl7yy80000gn/T//RtmpJH2Hox/downloaded\_packages

install.packages("scales")  
## The downloaded binary packages are in  
## /var/folders/7w/vf\_3wx5d6kq9bm2666hl7yy80000gn/T//RtmpJH2Hox/downloaded\_packages

install.packages("dplyr")  
## The downloaded binary packages are in  
## /var/folders/7w/vf\_3wx5d6kq9bm2666hl7yy80000gn/T//RtmpJH2Hox/downloaded\_packages

library(tidyverse)

## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
## ✔ dplyr 1.1.2 ✔ readr 2.1.4  
## ✔ forcats 1.0.0 ✔ stringr 1.5.0  
## ✔ ggplot2 3.4.2 ✔ tibble 3.2.1  
## ✔ lubridate 1.9.2 ✔ tidyr 1.3.0  
## ✔ purrr 1.0.1   
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()  
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

library(ggwordcloud)  
 library(viridis)

## Loading required package: viridisLite

library(scales)

## Attaching package: 'scales'

## The following object is masked from 'package:viridis':  
## viridis\_pal

## The following object is masked from 'package:purrr':  
## discard

## The following object is masked from 'package:readr':  
## col\_factor

library(dplyr)  
  
 options(repr.plot.width = 15, repr.plot.height = 15)

#Main Data set  
salary\_df <- read.csv("data.csv")  
  
#Country codes Data set, used for cleaning later  
iso\_df <- read.csv("wikipedia-iso-country-codes.csv")

Checking the dimensions of a dataframe can be useful for understanding the size and structure of your data and for performing various operations and analyses based on the number of rows and columns present.

t(t(names(salary\_df))) # In summary, the code transposes the column names of the data\_df dataframe and returns them as a column matrix to display the column names in a user friendly manner.

## [,1]   
## [1,] "X"   
## [2,] "work\_year"   
## [3,] "experience\_level"   
## [4,] "employment\_type"   
## [5,] "job\_title"   
## [6,] "salary"   
## [7,] "salary\_currency"   
## [8,] "salary\_in\_usd"   
## [9,] "employee\_residence"  
## [10,] "remote\_ratio"   
## [11,] "company\_location"   
## [12,] "company\_size"

#Dimensions of the data frame  
data\_dimensions <- dim(salary\_df)  
num\_rows <- data\_dimensions[1]  
num\_cols <- data\_dimensions[2]  
  
text <- paste("There are", num\_rows,"Rows", "and", num\_cols, "Columns", "in this dataset")

print(text)

## [1] "There are 607 Rows and 12 Columns in this dataset"

head(salary\_df) #Displays the first couple of rows of the data frame

## X work\_year experience\_level employment\_type job\_title  
## 1 0 2020 MI FT Data Scientist  
## 2 1 2020 SE FT Machine Learning Scientist  
## 3 2 2020 SE FT Big Data Engineer  
## 4 3 2020 MI FT Product Data Analyst  
## 5 4 2020 SE FT Machine Learning Engineer  
## 6 5 2020 EN FT Data Analyst  
## salary salary\_currency salary\_in\_usd employee\_residence remote\_ratio  
## 1 70000 EUR 79833 DE 0  
## 2 260000 USD 260000 JP 0  
## 3 85000 GBP 109024 GB 50  
## 4 20000 USD 20000 HN 0  
## 5 150000 USD 150000 US 50  
## 6 72000 USD 72000 US 100  
## company\_location company\_size  
## 1 DE L  
## 2 JP S  
## 3 GB M  
## 4 HN S  
## 5 US L  
## 6 US L

str(salary\_df) #Display list of columns and data types

## 'data.frame': 607 obs. of 12 variables:  
## $ X : int 0 1 2 3 4 5 6 7 8 9 ...  
## $ work\_year : int 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 ...  
## $ experience\_level : chr "MI" "SE" "SE" "MI" ...  
## $ employment\_type : chr "FT" "FT" "FT" "FT" ...  
## $ job\_title : chr "Data Scientist" "Machine Learning Scientist" "Big Data Engineer" "Product Data Analyst" ...  
## $ salary : int 70000 260000 85000 20000 150000 72000 190000 11000000 135000 125000 ...  
## $ salary\_currency : chr "EUR" "USD" "GBP" "USD" ...  
## $ salary\_in\_usd : int 79833 260000 109024 20000 150000 72000 190000 35735 135000 125000 ...  
## $ employee\_residence: chr "DE" "JP" "GB" "HN" ...  
## $ remote\_ratio : int 0 0 50 0 50 100 100 50 100 50 ...  
## $ company\_location : chr "DE" "JP" "GB" "HN" ...  
## $ company\_size : chr "L" "S" "M" "S" ...

summary(salary\_df) #Statistical summary of data

## X work\_year experience\_level employment\_type   
## Min. : 0.0 Min. :2020 Length:607 Length:607   
## 1st Qu.:151.5 1st Qu.:2021 Class :character Class :character   
## Median :303.0 Median :2022 Mode :character Mode :character   
## Mean :303.0 Mean :2021   
## 3rd Qu.:454.5 3rd Qu.:2022   
## Max. :606.0 Max. :2022   
## job\_title salary salary\_currency salary\_in\_usd   
## Length:607 Min. : 4000 Length:607 Min. : 2859   
## Class :character 1st Qu.: 70000 Class :character 1st Qu.: 62726   
## Mode :character Median : 115000 Mode :character Median :101570   
## Mean : 324000 Mean :112298   
## 3rd Qu.: 165000 3rd Qu.:150000   
## Max. :30400000 Max. :600000   
## employee\_residence remote\_ratio company\_location company\_size   
## Length:607 Min. : 0.00 Length:607 Length:607   
## Class :character 1st Qu.: 50.00 Class :character Class :character   
## Mode :character Median :100.00 Mode :character Mode :character   
## Mean : 70.92   
## 3rd Qu.:100.00   
## Max. :100.00

# Checking for any NA values  
any(is.na(salary\_df))

## [1] FALSE

#Checking for unique values per column  
library(dplyr)  
salary\_df %>%   
 summarise(  
 work\_year = n\_distinct(work\_year),  
 experience\_level = n\_distinct(experience\_level),  
 employment\_type = n\_distinct(employment\_type),  
 job\_title = n\_distinct(job\_title),  
 salary = n\_distinct(salary),  
 salary\_currency = n\_distinct(salary\_currency),  
 salary\_in\_usd = n\_distinct(salary\_in\_usd),  
 employee\_residence = n\_distinct(employee\_residence),  
 remote\_ratio = n\_distinct(remote\_ratio),  
 company\_location = n\_distinct(company\_location),  
 company\_size = n\_distinct(company\_size)  
 )

## work\_year experience\_level employment\_type job\_title salary salary\_currency  
## 1 3 4 4 50 272 17  
## salary\_in\_usd employee\_residence remote\_ratio company\_location company\_size  
## 1 369 57 3 50 3

Cleaning the Data There are a few problems we need to fix: 1. The columns experience\_level, employment\_type, employee\_residence, company\_size and company\_location contain abbreviations that may not be understood to some, we will rename the data contained in those columns to more descriptive labels.¶

#Renamed abbreviations for the experience\_level column  
salary\_df$experience\_level[salary\_df$experience\_level == "SE"] <- "Senior"  
salary\_df$experience\_level[salary\_df$experience\_level == "MI"] <- "Indermediate"  
salary\_df$experience\_level[salary\_df$experience\_level == "EN"] <- "Junior/Entry"  
salary\_df$experience\_level[salary\_df$experience\_level == "EX"] <- "Director/Executive"

#Renamed abbreviations for the employment\_type column  
salary\_df$employment\_type[salary\_df$employment\_type == "FT"] <- "FullTime"  
salary\_df$employment\_type[salary\_df$employment\_type == "PT"] <- "PartTime"  
salary\_df$employment\_type[salary\_df$employment\_type == "CT"] <- "Contract"  
salary\_df$employment\_type[salary\_df$employment\_type == "FL"] <- "FreeLance"

#Renamed abbreviations for the company\_size column  
salary\_df$company\_size[salary\_df$company\_size == "S"] <- "Small"  
salary\_df$company\_size[salary\_df$company\_size == "M"] <- "Medium"  
salary\_df$company\_size[salary\_df$company\_size == "L"] <- "Large"

#Pulling columns from salary\_df and iso into vectors that we can iterate from.  
employee\_residence <- salary\_df %>% pull(employee\_residence)  
  
Alpha.2.code <- iso\_df %>% pull(Alpha.2.code)  
  
Country\_name <- iso\_df %>% pull(English.short.name.lower.case)  
  
index <- 0  
new\_employee\_residence <- c()  
  
#Looping through each item in new\_employee\_residence  
for (item in employee\_residence) {  
   
#check where the item exists in Alpha.2.code, then assign the index found in Alpha.2.code to index  
 index <- which(Alpha.2.code == item)[1]  
   
#use the index to find the corresponding Country\_name, then append that Country\_name to new\_employee\_residence, save to new\_employee\_residence  
 new\_employee\_residence <- append(new\_employee\_residence, Country\_name[index])  
}  
  
# assign new\_employee\_residence to the employee\_residence column  
salary\_df$employee\_residence <- new\_employee\_residence

str(salary\_df$experience\_level)

## chr [1:607] "Indermediate" "Senior" "Senior" "Indermediate" "Senior" ...

str(salary\_df$employment\_type)

## chr [1:607] "FullTime" "FullTime" "FullTime" "FullTime" "FullTime" ...

str(salary\_df$employee\_residence)

## chr [1:607] "Germany" "Japan" "United Kingdom" "Honduras" ...

str(salary\_df$company\_size)

## chr [1:607] "Large" "Small" "Medium" "Small" "Large" "Large" "Small" ...

str(salary\_df$company\_location)

## chr [1:607] "DE" "JP" "GB" "HN" "US" "US" "US" "HU" "US" "NZ" "FR" "IN" ...

1. We do not need the salary or salary\_currency columns for our analysis.¶

salary\_df <- salary\_df %>%  
 select(-c(X, salary, salary\_currency))  
  
str(salary\_df)

## 'data.frame': 607 obs. of 9 variables:  
## $ work\_year : int 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 ...  
## $ experience\_level : chr "Indermediate" "Senior" "Senior" "Indermediate" ...  
## $ employment\_type : chr "FullTime" "FullTime" "FullTime" "FullTime" ...  
## $ job\_title : chr "Data Scientist" "Machine Learning Scientist" "Big Data Engineer" "Product Data Analyst" ...  
## $ salary\_in\_usd : int 79833 260000 109024 20000 150000 72000 190000 35735 135000 125000 ...  
## $ employee\_residence: chr "Germany" "Japan" "United Kingdom" "Honduras" ...  
## $ remote\_ratio : int 0 0 50 0 50 100 100 50 100 50 ...  
## $ company\_location : chr "DE" "JP" "GB" "HN" ...  
## $ company\_size : chr "Large" "Small" "Medium" "Small" ...

1. Since remote\_ratio contains ratios they all should have % at the end.¶

salary\_df$remote\_ratio <- gsub("%", "", salary\_df$remote\_ratio)

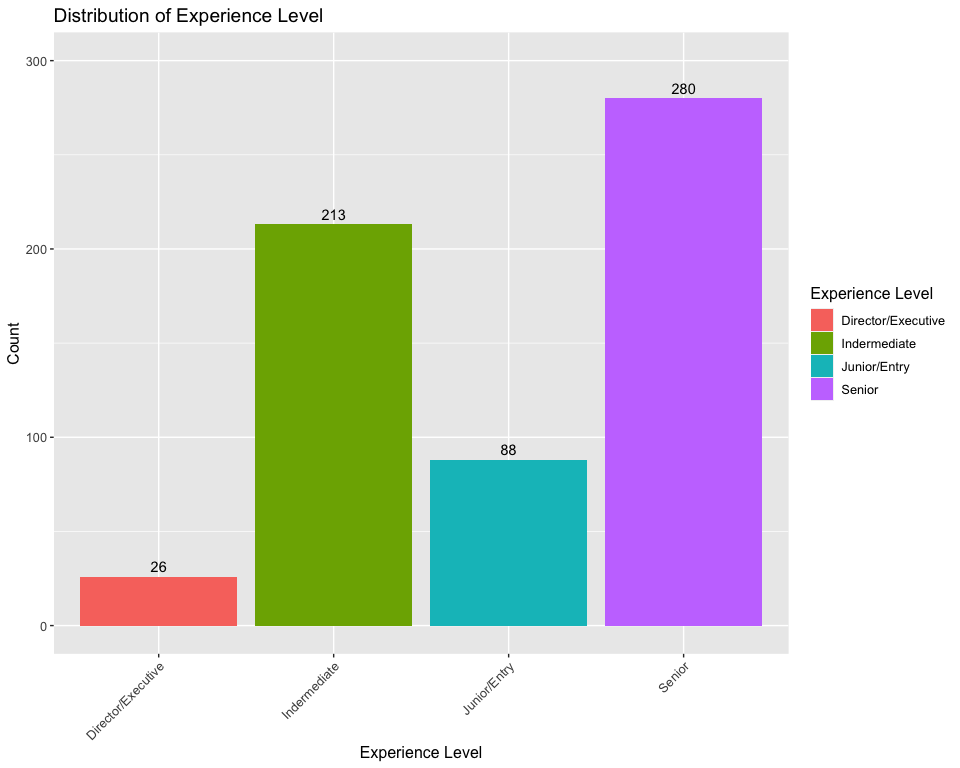
#Add a % at the end of every element in remote\_ratio  
remote\_ratio <- salary\_df %>% pull(remote\_ratio)  
new\_remote\_ratio <- paste0(remote\_ratio, "%")  
  
salary\_df$remote\_ratio <- new\_remote\_ratio

str(salary\_df$remote\_ratio)

## chr [1:607] "0%" "0%" "50%" "0%" "50%" "100%" "100%" "50%" "100%" "50%" ...

Analysis

library(ggplot2)  
library(dplyr)  
  
experience\_level <- data.frame(experience\_level = salary\_df$experience\_level)  
experience\_counts <- experience\_level %>% count(experience\_level)  
  
ggplot(data = experience\_level) +  
 geom\_bar(mapping = aes(x = experience\_level, fill = experience\_level)) +  
 geom\_text(data = experience\_counts, aes(x = experience\_level, y = n, label = n), vjust = -0.5) +  
 theme(axis.text.x = element\_text(angle = 45, hjust = 1), text = element\_text(size = 12)) +  
 labs(title = "Distribution of Experience Level",  
 x = "Experience Level",  
 y = "Count",  
 fill = "Experience Level") +  
 coord\_cartesian(ylim = c(0, 300))



Most people with data science related jobs are Senior level.

library(ggplot2)  
  
query <- salary\_df %>%  
 select(job\_title, experience\_level) %>%  
 group\_by(job\_title, experience\_level) %>%  
 summarise(Count = n()) %>%  
 arrange(desc(Count)) %>%  
 filter(job\_title %in% c("Data Engineer", "Data Scientist", "Data Analyst", "Machine Learning Engineer", "Analytics Engineer", "Data Architect", "Research Scientist", "Applied Scientist", "Data Science Manager", "Research Engineer"))

## `summarise()` has grouped output by 'job\_title'. You can override using the  
## `.groups` argument.

ggplot(query, aes(x = job\_title, y = Count, fill = experience\_level)) +  
 geom\_bar(stat = "identity", position = "dodge") +  
 geom\_text(aes(label = Count, group = experience\_level), position = position\_dodge(width = 0.9), vjust = -0.5, size = 2, color = "black") +  
 theme(axis.text.x = element\_text(angle = 50, hjust = 1), text = element\_text(size = 12)) +  
 labs(title = "Experience Level / Job Title", subtitle = "For the 10 most popular data science jobs", x = "Job Title")

A graph of different colored bars

Description automatically generated The ratio of Seniors vs other experience levels is roughly the same per job title.

library(ggplot2)  
  
employment\_type <- data.frame(employment\_type = salary\_df$employment\_type)  
  
ggplot(data = employment\_type, aes(x = employment\_type)) +  
 geom\_bar(fill = "orange") +  
 geom\_text(stat = "count", aes(label = after\_stat(count)), vjust = -0.5, color = "black") +  
 labs(title = "Employment Type Distribution", x = "Employment Type", y = "Count") +  
 theme(axis.text.x = element\_text(angle = 45, hjust = 1), text = element\_text(size = 14)) +   
 scale\_y\_continuous(limits = c(0, 700))

A graph of a number of employment type distribution

Description automatically generatedVirtually all employees are full time.

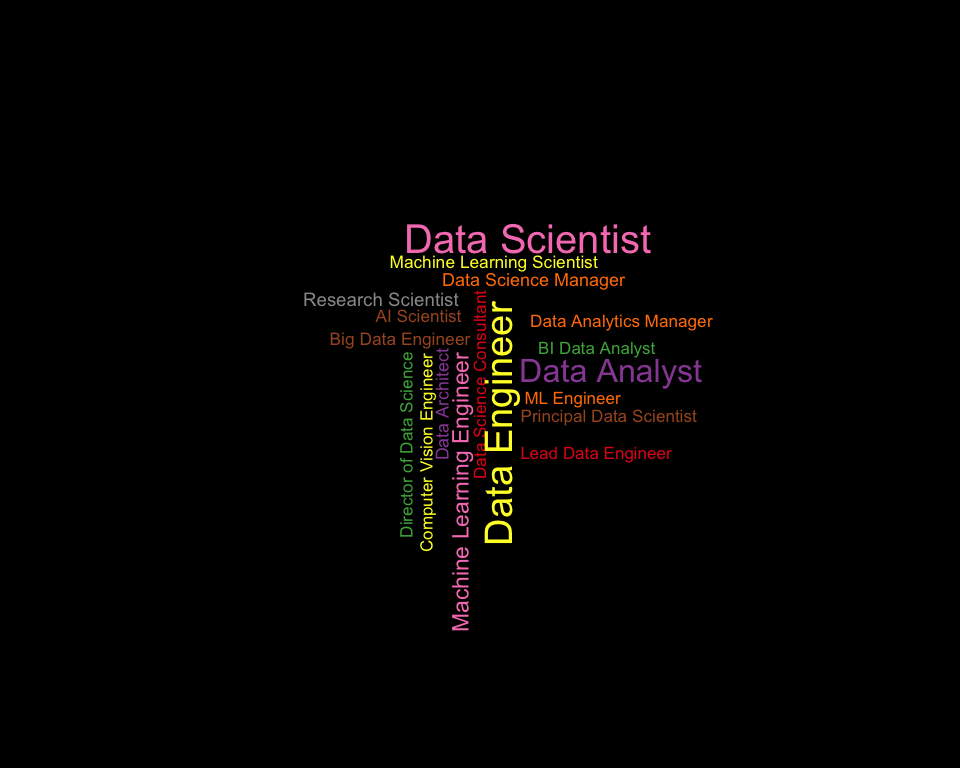
job\_title\_counts <- table(salary\_df$job\_title)  
  
# Order the job titles in decreasing order  
ordered\_counts <- sort(job\_title\_counts, decreasing = TRUE)  
  
# Display the ordered job title counts  
ordered\_counts

##   
## Data Scientist   
## 143   
## Data Engineer   
## 132   
## Data Analyst   
## 97   
## Machine Learning Engineer   
## 41   
## Research Scientist   
## 16   
## Data Science Manager   
## 12   
## Data Architect   
## 11   
## Big Data Engineer   
## 8   
## Machine Learning Scientist   
## 8   
## AI Scientist   
## 7   
## Data Analytics Manager   
## 7   
## Data Science Consultant   
## 7   
## Director of Data Science   
## 7   
## Principal Data Scientist   
## 7   
## BI Data Analyst   
## 6   
## Computer Vision Engineer   
## 6   
## Lead Data Engineer   
## 6   
## ML Engineer   
## 6   
## Applied Data Scientist   
## 5   
## Business Data Analyst   
## 5   
## Data Engineering Manager   
## 5   
## Head of Data   
## 5   
## Analytics Engineer   
## 4   
## Applied Machine Learning Scientist   
## 4   
## Data Analytics Engineer   
## 4   
## Head of Data Science   
## 4   
## Computer Vision Software Engineer   
## 3   
## Data Science Engineer   
## 3   
## Lead Data Analyst   
## 3   
## Lead Data Scientist   
## 3   
## Machine Learning Developer   
## 3   
## Machine Learning Infrastructure Engineer   
## 3   
## Principal Data Engineer   
## 3   
## Cloud Data Engineer   
## 2   
## Director of Data Engineering   
## 2   
## ETL Developer   
## 2   
## Financial Data Analyst   
## 2   
## Principal Data Analyst   
## 2   
## Product Data Analyst   
## 2   
## 3D Computer Vision Researcher   
## 1   
## Big Data Architect   
## 1   
## Data Analytics Lead   
## 1   
## Data Specialist   
## 1   
## Finance Data Analyst   
## 1   
## Head of Machine Learning   
## 1   
## Lead Machine Learning Engineer   
## 1   
## Machine Learning Manager   
## 1   
## Marketing Data Analyst   
## 1   
## NLP Engineer   
## 1   
## Staff Data Scientist   
## 1

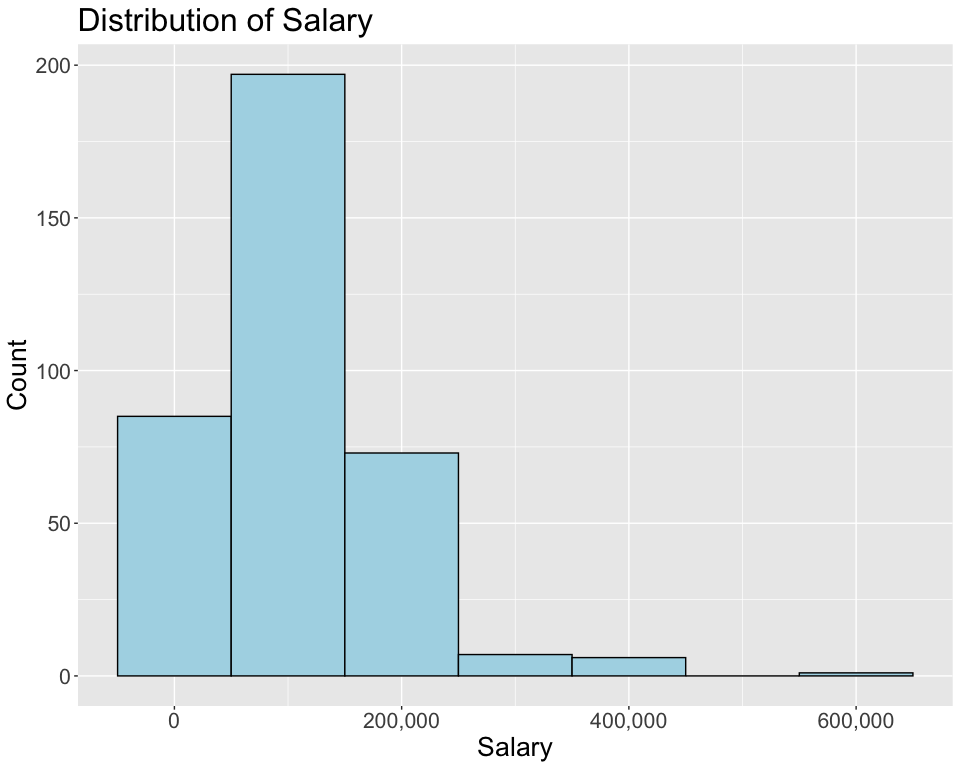
library(wordcloud)

## Loading required package: RColorBrewer

library(RColorBrewer)  
  
# Convert job\_title to character  
salary\_df$job\_title <- as.character(salary\_df$job\_title)  
  
# Filter the data based on job titles with count > 5  
filtered\_counts <- job\_title\_counts[job\_title\_counts > 5]  
  
# Generate 20 bright colors by repeating the "Set1" palette  
num\_colors <- 20  
colors <- rep(brewer.pal(9, "Set1"), length.out = num\_colors)  
  
# Create a data frame with word frequencies  
word\_data <- data.frame(word = names(filtered\_counts),  
 freq = as.numeric(filtered\_counts))  
  
# Set a dark background for the word cloud  
par(bg = "black")  
  
# Generate the word cloud using wordcloud  
set.seed(123) # For reproducibility of colors and angles  
wordcloud(word\_data$word, word\_data$freq, scale = c(2.5, 1), random.order = TRUE,  
 colors = colors, rot.per = 0.2, random.color = TRUE)



library(ggplot2)  
library(dplyr)  
  
# Create a data frame with salary counts  
salary <- salary\_df %>%  
 group\_by(salary\_in\_usd) %>%  
 summarise(count = n())  
  
# Convert salary\_in\_usd to numeric  
salary$salary\_in\_usd <- as.numeric(as.character(salary$salary\_in\_usd))  
  
# Sort the data frame by salary\_in\_usd  
salary <- salary[order(salary$salary\_in\_usd), ]  
  
# Set plot dimensions  
options(repr.plot.width = 15, repr.plot.height = 15)  
  
# Create the histogram plot with binwidth of 100000  
ggplot(salary, aes(x = salary\_in\_usd)) +  
 geom\_histogram(binwidth = 100000, fill = "lightblue", color = "black") +  
 scale\_x\_continuous(labels = scales::comma) +  
 theme(text = element\_text(size = 20)) +  
 labs(title = "Distribution of Salary", x = "Salary", y = "Count")

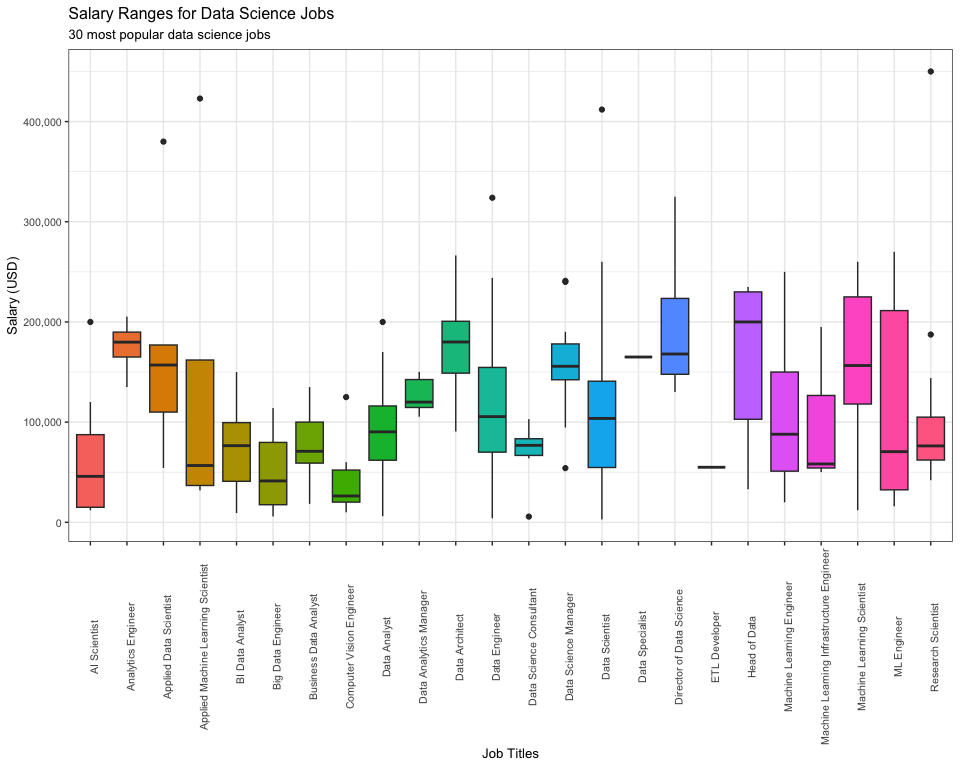


# Print summary statistics  
print(summary(salary))

## salary\_in\_usd count   
## Min. : 2859 Min. : 1.000   
## 1st Qu.: 53192 1st Qu.: 1.000   
## Median : 93000 Median : 1.000   
## Mean :107471 Mean : 1.645   
## 3rd Qu.:145000 3rd Qu.: 2.000   
## Max. :600000 Max. :15.000

The average salary for all Data Science related jobs is 130,000 and 180,000$

library(ggplot2)  
library(scales)  
  
salary\_df$salary\_in\_usd <- as.numeric(salary\_df$salary\_in\_usd)  
  
temp <- data.frame(table(salary\_df$job\_title))  
  
temp <- temp %>%  
 arrange(desc(Freq)) %>%  
 head(n = 70)  
  
data <- salary\_df %>%  
 select(job\_title, salary\_in\_usd) %>%  
 filter(job\_title == "Data Engineer" | job\_title == "Data Scientist" | job\_title == "Data Analyst" | job\_title == "Machine Learning Engineer" | job\_title == "Analytics Engineer" | job\_title == "Data Architect" | job\_title == "Research Scientist" | job\_title == "Applied Scientist" | job\_title == "Data Science Manager" | job\_title == "Research Engineer" | job\_title == "ML Engineer" | job\_title == "Data Manager" | job\_title == "Machine Learning Scientist" | job\_title == "Data Science Consultant" | job\_title == "Data Analytics Manager" | job\_title == "Computer Vision Engineer" | job\_title == "AI Scientist" | job\_title == "BI Data Analyst" | job\_title == "Business Data Analyst" | job\_title == "Data Specialist" | job\_title == "BI Developer" | job\_title == "Applied Machine Learning Scientist" | job\_title == "AI Developer" | job\_title == "Big Data Engineer" | job\_title == "Director of Data Science" | job\_title == "Machine Learning Infrastructure Engineer" | job\_title == "Applied Data Scientist" | job\_title == "Data Operations Engineer" | job\_title == "ETL Developer" | job\_title == "Head of Data")  
  
options(repr.plot.width = 15, repr.plot.height = 15)  
  
data %>%  
 ggplot(aes(job\_title, salary\_in\_usd, fill = job\_title))+  
 geom\_boxplot()+  
 theme\_bw()+  
 theme(axis.text.x = element\_text(angle = 90), legend.position = "none", text = element\_text(size = 10))+  
 scale\_y\_continuous(labels = comma)+  
 labs(title = "Salary Ranges for Data Science Jobs", subtitle = "30 most popular data science jobs", x = "Job Titles", y = "Salary (USD)")



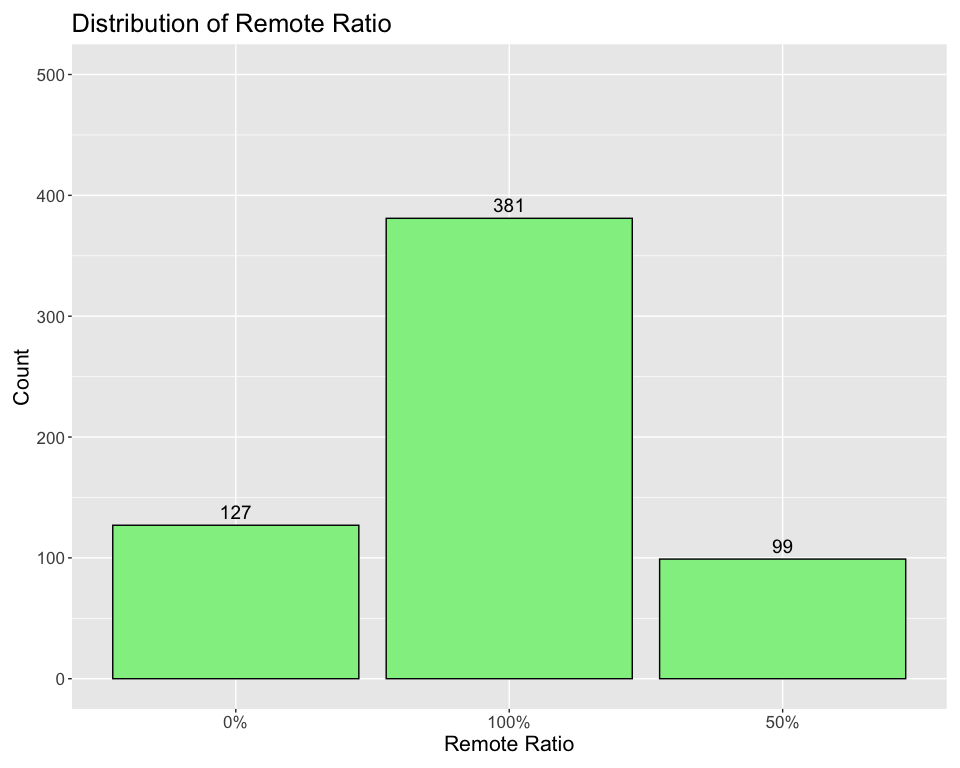
library(ggplot2)  
library(magrittr)

##   
## Attaching package: 'magrittr'

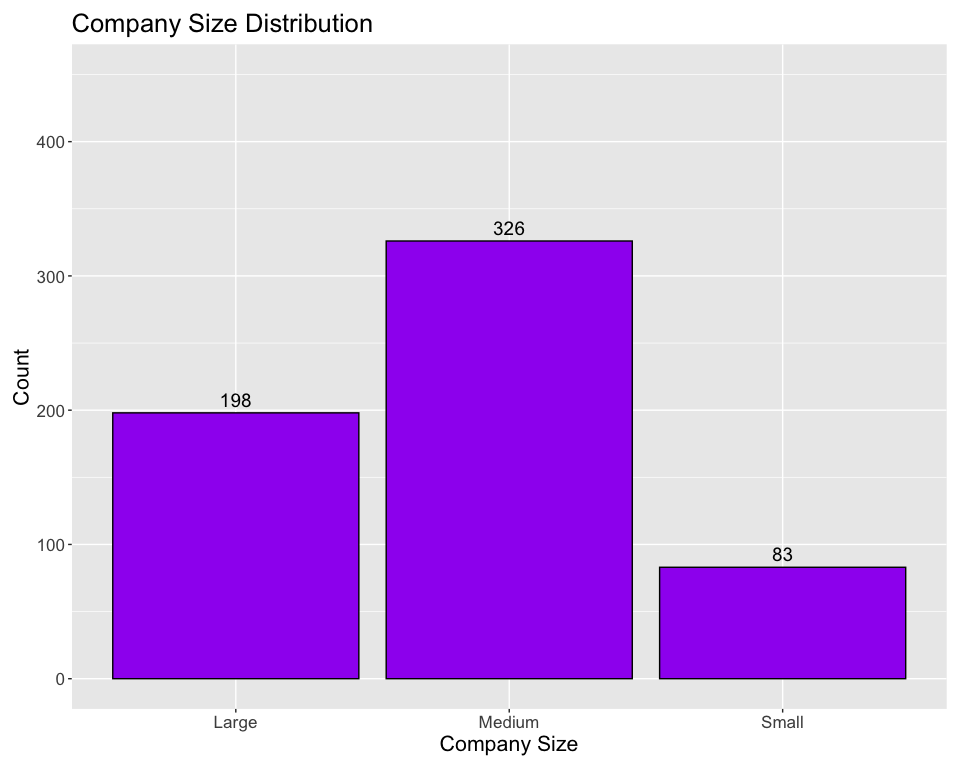
## The following object is masked from 'package:purrr':  
##   
## set\_names

## The following object is masked from 'package:tidyr':  
##   
## extract

remote\_ratio <- data.frame(table(salary\_df$remote\_ratio))  
  
options(repr.plot.width = 15, repr.plot.height = 15)  
  
remote\_ratio %>%  
 ggplot() +  
 aes(x = Var1, y = Freq) +  
 geom\_col(fill = "lightgreen", color = "black") +  
 geom\_text(aes(label = Freq), vjust = -0.5, size = 5) +  
 xlab("Remote Ratio") +  
 ylab("Count") +  
 ylim(c(0, 500)) + # Set y-axis limits (adjust the upper limit as desired)  
 theme(text = element\_text(size = 16)) +  
 labs(title = "Distribution of Remote Ratio") +  
 theme(plot.margin = margin(10, 10, 10, 10, "pt"))



company\_size <- data.frame(table(salary\_df$company\_size))  
  
options(repr.plot.width = 15, repr.plot.height = 15)  
  
 company\_size %>%  
 ggplot() +  
 aes(x = Var1, y = Freq) +  
 geom\_col(fill = "purple", color = "black") +  
 geom\_text(aes(label = Freq), vjust = -0.5, size = 5) +  
 labs(title = "Company Size Distribution", x = "Company Size", y = "Count") +  
 ylim(c(0, 450)) + # Set y-axis limits (adjust the upper limit as desired)  
 theme(text = element\_text(size = 16)) +  
 theme(plot.margin = margin(10, 10, 10, 10, "pt"))



Conclusions In this analysis, we explored the salaries of data science jobs in the US using data from Kaggle. We found that the average salary of all data related jobs is 130,000 and 180,000$.We also identified some trends and patterns in the data science job market, such as most employee’s experience being senior-level and most data science companies are located in the US with company sizes ranging from 50-250 employees.

In conclusion, our analysis provides a valuable snapshot of the current state and trends of data science salaries in the US, which can help data scientists and employers make informed decisions and plan their careers.