Analysis of Salaries in 2024

Souhaila El kadaoui

2024-08-30

Table of Contents

# 1.Introduction

The project aims to provide an overview of jobs and salaries for 2024. The dataset focuses on trends in jobs and salaries for 2024 and includes information on internship titles, company names, locations, estimated salaries, and company ratings. The columns in the dataset are as follows:

-**Company:** Name of the hiring company.  
-**Company Score:** Average rating of the company.  
-**Job Title:** Title of the internship position.  
-**Location:** Job location, including city and state (or if remote).  
-**Salary:** Estimated salary range for the internship position, as listed by the employer or Glassdoor.

# 2.Objectives of the Project

-Comparison of salaries between different companies.  
-Evaluation of employee reviews for companies with high and low salaries.  
-Average salary determination for each job.

# 3.Project Steps and Processes

### 3.1 Project’s tools

-R as a programming language -dplyr & ggplot2,knitr stringr,RColorBrewer R’s Packages for manipulation and visualization of data -R markdown ### 3.2 Import necessary packages

#Suppress messages when loading the dplyr package  
suppressMessages(library(knitr))

## Warning: le package 'knitr' a été compilé avec la version R 4.4.1

suppressWarnings(library(knitr))  
suppressMessages(library(dplyr))

## Warning: le package 'dplyr' a été compilé avec la version R 4.4.1

suppressWarnings(library(dplyr))  
suppressMessages(library(ggplot2))

## Warning: le package 'ggplot2' a été compilé avec la version R 4.4.1

suppressWarnings(library(ggplot2))  
suppressMessages(library(stringr))

## Warning: le package 'stringr' a été compilé avec la version R 4.4.1

suppressWarnings(library(stringr))  
suppressMessages(library(RColorBrewer))  
suppressWarnings(library(RColorBrewer))  
suppressMessages(library(gridExtra))

## Warning: le package 'gridExtra' a été compilé avec la version R 4.4.1

suppressWarnings(library(gridExtra))  
library(dplyr)  
library(ggplot2)  
library(knitr)  
library(stringr)  
library(RColorBrewer)  
library(gridExtra)

### 3.3 Data Cleaning

#### load data

suppressMessages({  
 suppressWarnings({  
 data <- read.csv(file.choose(), header = TRUE, sep = ",")  
 df <- data.frame(data) #make sure to have data frame structure which is more flexible for using  
 invisible(head(df)) #to hide console output  
 })  
})

#### Display the first 6 rows as a table

kable(head(df), caption = "First 6 Rows of the dataframe")

First 6 Rows of the dataframe

| Company | Company.Score | Job.Title | Location | Salary |
| --- | --- | --- | --- | --- |
| PepsiCo | 3.9 | 2025 Summer Intern: R&D | Plano, TX | $21.50 - $55.40 Per Hour (Employer est.) |
| ghSMART | 4.8 | Data Engineer Intern | Remote | $20.00 - $30.00 Per Hour (Employer est.) |
| PepsiCo | 3.9 | 2025 Summer Intern: eCommerce Data Science | United States | $21.50 - $40.19 Per Hour (Employer est.) |
| U.S. Bank National Association | 3.6 | 2025 Data Analytics Summer Intern | Minneapolis, MN | $20.63 - $30.25 Per Hour (Employer est.) |
| Plymouth Rock Assurance | 3.4 | Fall 2024 Data Science/Predictive Modeler Intern | Woodbridge, NJ | $77K - $130K (Glassdoor est.) |
| Brookfield Properties | 3.8 | Data Analyst Intern | Charleston, SC | $36K - $50K (Glassdoor est.) |

#### correct columns names to avoid problems (related to “.”)

df=rename(df,Company\_Score=Company.Score,Job\_Title=Job.Title)  
#verify the correction  
colnames(df)

## [1] "Company" "Company\_Score" "Job\_Title" "Location"   
## [5] "Salary"

#### Verify duplicated rows

cat("the number of duplicated rows is:",sum(duplicated(df)))

## the number of duplicated rows is: 0

#### verify missing values in cases of columns

missing\_values <- sapply(df, function(x) sum(is.na(x)))  
print("Number of missing values for each columns")

## [1] "Number of missing values for each columns"

missing\_values

## Company Company\_Score Job\_Title Location Salary   
## 0 98 0 0 0

n=nrow(df)  
#remove rows with missing cases  
df <- df[!is.na(df$Company) & !is.na(df$Company\_Score) & !is.na(df$Job\_Title) & !is.na(df$Location) & !is.na(df$Salary), ]  
#ignore empty cases(don't respect rows format), which dosn't remove if the condition above  
df=df[1:500,]  
n1=nrow(df)  
cat("the rows number with missing values is ",n,".After correction will be ",n1," rows")

## the rows number with missing values is 649 .After correction will be 500 rows

#### verify data types of columns

print("Data types:")

## [1] "Data types:"

for (i in 1:ncol(df)) {  
 column\_name <- names(df)[i]  
 column\_data <- df[[i]]  
   
 # Check if all elements are numeric  
 if (all(sapply(column\_data, is.numeric))) {  
 cat(column\_name, "elements are numeric\n")  
 }   
 # Check if all elements are character  
 else if (all(sapply(column\_data, is.character))) {  
 cat(column\_name, "elements are character\n")  
 }   
 # If elements are neither numeric nor character  
 else {  
 cat(column\_name, "elements aren't numeric or character\n")  
 }  
}

## Company elements are character  
## Company\_Score elements are numeric  
## Job\_Title elements are character  
## Location elements are character  
## Salary elements are character

#### correct Salary column data format

# Initialize vectors  
reference <- vector("character", length = nrow(df))  
categories\_salary <- vector("character", length = nrow(df))  
  
  
for (i in 1:nrow(df)) {  
 # extract refrence of informations  
 extracted <- str\_extract(df$Salary[i], "(?<=\\().+?(?=\\))")  
 df$Salary[i] <- gsub("\\s\*\\([^\\)]+\\)", "", df$Salary[i])  
 if (!is.na(extracted)) {  
 reference[i] <- gsub("est", "", extracted)  
 } else {  
 reference[i] <- "Unknown reference"  
 }  
   
}  
# Add the vectors to the data frame  
df$Reference <- reference  
# create salary columns without others character data  
# Initialize vectors to store the results  
numeric\_range <- vector("character", length = nrow(df))  
category <- vector("character", length = nrow(df))  
for (i in 1:nrow(df)) {  
 salary <- df$Salary[i]  
   
 if (str\_detect(salary, "^\\$\\d+\\.\\d{2}\\s\*-\\s\*\\$\\d+\\.\\d{2}\\s+.\*")) { # verify the format :"$20.63 - $30.25 Per Hour "  
 result <- str\_match(salary, "(\\$\\d+\\.\\d{2}\\s\*-\\s\*\\$\\d+\\.\\d{2})\\s\*(.\*)") #extract sub part   
   
 # extract numeric part  
 numeric\_range[i] <- result[2] # "$20.63 - $30.25 "  
 category[i] <- "Per Hour" # "Per Hour"   
 }  
 else if(str\_detect(salary,"^\\$\\d{1,3}K\\s\*-\\s\*\\$\\d{1,3}K\\s\*$")){ #"$20.63 - $30.25 "  
 numeric\_range[i]=salary  
 category[i]="Annual"  
   
 }  
 else if(str\_detect(salary,"^\\$\\d+\\.\\d{2}\\s+Per Hour\\s\*$")){ #format:"$22.86"  
 result <- str\_match(salary, "(\\$\\d+\\.\\d{2})\\s+Per Hour")  
   
 numeric\_range[i] <- result[2] # "$22.86"  
 category[i] <- "Per Hour"   
 }  
 else if(str\_detect(salary,"^\\$\\d+(\\.\\d{1,2})?K\\s\*$")) { #format "$22.86K"  
 numeric\_range[i]=salary  
 category[i]="Annual"  
   
 }  
   
}  
#add result to dataframe  
df$Salary\_Range <- numeric\_range  
df$Category <- category  
#remove original salary column  
df=df[1:nrow(df),-5]  
#create Mean\_Salary Column to make sure of data homogenity  
  
pattern\_dollars\_range <-"^\\$\\d+\\.\\d{2}\\s\*-\\s\*\\$\\d+\\.\\d{2}$" #"$77 - $130"  
df$Mean\_SalaryUSD=NA  
# correct salary data type: convert to numeric  
for (i in 1:nrow(df)) {  
 salary <- df$Salary[i]  
 # Vérifier si la chaîne correspond au format avec une plage de valeurs  
 if (str\_detect(salary, pattern\_dollars\_range)) { #extract salary with format "$44.99-$7.99"  
   
 pattern <- "\\$([\\d\\.]+)\\s\*-\\s\*\\$([\\d\\.]+)"  
 result <- str\_match(salary, pattern)#split $ symbole to numbers  
   
 if (!is.na(result[1])) {  
 min\_salary <- as.numeric(result[2])  
 max\_salary <- as.numeric(result[3])  
 Moy=c(min\_salary,max\_salary)  
   
 df$Mean\_SalaryUSD[i] <- mean(Moy)  
 }  
 }  
 else if(str\_detect(salary,"^\\$([\\d]+)K\\s\*-\\s\*\\$([\\d]+)K\\s\*$")){ #extract salary from "$44.99K-$7.99K"  
   
 result <- str\_match(salary, "^\\$([\\d]+)K\\s\*-\\s\*\\$([\\d]+)K\\s\*$")  
  
# Vérifier si le résultat n'est pas NA  
if (!is.na(result[1])) {  
 # Convertir les montants en numériques, supprimer le "K" et utiliser les valeurs telles quelles  
 min\_salary <- as.numeric(result[2])  
 max\_salary <- as.numeric(result[3])  
 Moy=c(min\_salary\*1000,max\_salary\*1000)  
   
 df$Mean\_SalaryUSD[i] <- mean(Moy)  
}  
 }  
 else if (str\_detect(salary,"^\\$\\d+\\.\\d{2}$")){ #extract salary from "$44.99"  
 salary\_numeric <- gsub("\\$", "", salary)  
salary\_value <- as.numeric(salary\_numeric)  
df$Mean\_SalaryUSD[i]=salary\_value  
  
 }  
 else if (str\_detect(salary, "^\\$\\d+(\\.\\d{1,2})?K\\s\*$")) { #extract salary from "$44.99K"  
 # Extraire la partie numérique sans le '$' et 'K'  
 number <- str\_extract(salary, "\\d+(\\.\\d{1,2})?")  
 # Convertir en numérique et multiplier par 1000  
 numeric\_value <- as.numeric(number) \* 1000  
 df$Mean\_SalaryUSD[i]=numeric\_value  
 }}

#### reverify Mean\_Salary\_USD data

column\_name <- names(df)[8]  
column\_data <- df[[8]]  
  
# Check if all elements are numeric, ignoring NAs  
check=all(sapply(column\_data, function(x) is.numeric(x) ))   
cat(column\_name, " are all elements numeric?",check)

## Mean\_SalaryUSD are all elements numeric? TRUE

### 3.4 statistic Summarize

# Calculate the mean salary for each category (Annual and Per Hour)  
mean\_salaries <- tapply(df$Mean\_SalaryUSD, df$Category, mean, na.rm = TRUE)  
# Display the results  
cat("The mean salary in both cases (Annual or Per Hour in $) for all selected Jobs is:\n")

## The mean salary in both cases (Annual or Per Hour in $) for all selected Jobs is:

cat(" Annual Salary :", mean\_salaries["Annual"], "$\t")

## Annual Salary : 66118.64 $

cat("Per Hour Salary:", mean\_salaries["Per Hour"], "$\n")

## Per Hour Salary: 24.80636 $

#specify if we want to calculate max salary of each job,or mean salary ,if annual or per hour ,type are:(min\_,max\_,mean\_)  
summarise\_job=function(df,type,cat){  
 if(type=="max\_"){  
 X=data.frame(Job\_vs\_Per\_hour\_Salary=tapply(df$Mean\_SalaryUSD[df$Category == cat], df$Job\_Title[df$Category == cat],max))  
 }  
 if(type=="min\_"){  
 X=data.frame(Job\_vs\_Per\_hour\_Salary=tapply(df$Mean\_SalaryUSD[df$Category == cat], df$Job\_Title[df$Category == cat],min))  
 }  
 if(type=="mean\_"){  
 X=data.frame(Job\_vs\_Per\_hour\_Salary=tapply(df$Mean\_SalaryUSD[df$Category == cat], df$Job\_Title[df$Category == cat],mean))  
 }  
 return(X)  
}  
#mean annualy of each job  
result\_An=summarise\_job(df,"mean\_","Annual")  
result\_PerH=summarise\_job(df,"mean\_","Per Hour")  
max\_An=summarise\_job(df,"max\_","Annual")  
max\_PH=summarise\_job(df,"max\_","Per Hour")  
min\_An=summarise\_job(df,"min\_","Annual")  
min\_PH=summarise\_job(df,"min\_","Per Hour")

#### the mean annual salary($) of some data intern/job selected

for (i in seq(1:40)) {  
 # Check if the job title contains the word "data"  
 if (grepl("data", rownames(result\_An)[i], ignore.case = TRUE)) {  
 # Print the job title and corresponding mean annual salary  
 cat(rownames(result\_An)[i], ":", result\_An[i, 1], "\n")  
 }  
}

## 2025 Land Data Management Internship Program : 61000   
## 2025 Summer Intern: Technology Data & Analytics, Data Engineer & Data Science : 71500   
## Academy Data Analyst - Part-time, Remote : 120000   
## Ad Data Solutions Intern (Global Business Solutions - Data Analytics) - 2025 Summer/Fall (BS/MS/MBA/PhD) : 91000   
## Analytics & Data Intern : 81500   
## Campus Undergraduate – 2025 US Consumer Services Data Analytics Summer Internship New York, NY : 89500   
## Customer Data Intern : 61000   
## Data & AI Intern : 71500   
## Data Analyst Intern : 59333.33

#### the min annual salary($) of some data intern/job selected

for (i in seq(1:40)) {  
 # Check if the job title contains the word "data"  
 if (grepl("data", rownames(min\_An)[i], ignore.case = TRUE)) {  
 # Print the job title and corresponding mean annual salary  
 cat(rownames(min\_An)[i], ":", min\_An[i, 1], "\n")  
 }  
}

## 2025 Land Data Management Internship Program : 61000   
## 2025 Summer Intern: Technology Data & Analytics, Data Engineer & Data Science : 71500   
## Academy Data Analyst - Part-time, Remote : 120000   
## Ad Data Solutions Intern (Global Business Solutions - Data Analytics) - 2025 Summer/Fall (BS/MS/MBA/PhD) : 91000   
## Analytics & Data Intern : 81500   
## Campus Undergraduate – 2025 US Consumer Services Data Analytics Summer Internship New York, NY : 89500   
## Customer Data Intern : 61000   
## Data & AI Intern : 71500   
## Data Analyst Intern : 43000

#### the max annual salary($) of some data intern/job selected

for (i in seq(1:40)) {  
 # Check if the job title contains the word "data"  
 if (grepl("data", rownames(max\_An)[i], ignore.case = TRUE)) {  
 # Print the job title and corresponding mean annual salary  
 cat(rownames(max\_An)[i], ":", max\_An[i, 1], "\n")  
 }  
}

## 2025 Land Data Management Internship Program : 61000   
## 2025 Summer Intern: Technology Data & Analytics, Data Engineer & Data Science : 71500   
## Academy Data Analyst - Part-time, Remote : 120000   
## Ad Data Solutions Intern (Global Business Solutions - Data Analytics) - 2025 Summer/Fall (BS/MS/MBA/PhD) : 91000   
## Analytics & Data Intern : 81500   
## Campus Undergraduate – 2025 US Consumer Services Data Analytics Summer Internship New York, NY : 89500   
## Customer Data Intern : 61000   
## Data & AI Intern : 71500   
## Data Analyst Intern : 76500

#### the of Per\_Hour salary($) of some data intern/job selected

for (i in 1:nrow(result\_PerH)) {  
 # Check if the job title contains the word "data"  
 if (grepl("data", rownames(result\_PerH)[i], ignore.case = TRUE)) {  
 # Print the job title and corresponding mean annual salary  
 cat(rownames(result\_PerH)[i], ":", result\_PerH[i, 1], "\n")  
 }  
}

## 2025 BNY Summer Internship Program - Engineering (Data Science) : 22.5   
## 2025 Data Analytics Summer Intern : 25.44   
## 2025 Summer Intern: Commerce Data & Technology : 30.845   
## 2025 Summer Intern: eCommerce Data Science : 30.845   
## Backend Software Engineer Intern (TikTok Data Ecosystem, Data Lake) - 2025 Summer (BS/MS) : 17.5   
## Billing Data Analytics Intern : 55   
## College to Corporate Internship – Data Analyst Advanced Development Program : 15   
## Current PhD - Data Science Internship - Summer 2025 : 23.05   
## Data Analyst Intern : 23.5   
## Data Analyst Intern - Must Live in St. Louis, MO! : 16.52   
## Data Analytics Intern : 24.9   
## Data Analytics Management Consulting Intern : 37   
## Data Analytics/Procurement Intern : 21   
## Data Discovery ETL Developer Intern : 17.5   
## Data Engineer Intern : 25   
## Data Engineer Intern - Advanced Analytics : 19.5   
## Data Engineer Intern/Co-op : 31.5   
## Data Engineering Internship (Summer 2025) : 20   
## Data Entry Intern : 18.25   
## Data Reporting and Analytics Intern : 24.5   
## Data Science Fall Internship : 21   
## Data Science Intern - Fall 2024 : 25   
## Data Scientist Intern - Jonesboro, AR : 42.5   
## Digital Health Data Engineer Intern - Hybrid, Part-Time : 27.5   
## Engineering Intern - Data Analytics : 19.22   
## EV Strategy Data Science Intern- Fall 2024- Franklin, TN : 20   
## Fall Data Analytics Internship - National : 22   
## Geospatial Data Analyst Internship : 15   
## Intern - Controls and Data Systems Engineering : 22   
## Intern - Yield Enhancement, Data Analysis : 17.5   
## Intern, Commercial Credit and Data Analytics : 35.75   
## Internship - Payload Satellite Engineering: Big Data Sets : 17.5   
## Internship and Data Coordinator : 20.5   
## Machine Learning Engineer Intern (Risk Data Mining) - 2025 Summer (PhD) : 17   
## Manufacturing Engineering Data Science Intern (Fall 2024) : 35   
## Museum Data Collection Intern : 42   
## Product Support Data Interns : 17   
## Professional Development Program (PDP) Internship - Data Science Summer 2025 : 23.49   
## Prudential: 2025 Corporate, Internal Audit Data Analytics Internship Program : 19.5   
## Prudential: 2025 Technology & Data, Software Engineering Internship Program : 28.5   
## RA Data Analytics Apprentice : 27.25   
## Relational Database Intern (Data Tech Infra) - 2025 Summer (MS) : 18   
## SAP Ariba iXp Intern - Business Intelligence and Data Analytics : 30   
## Sapna NYC Data, Research, & Communications Intern : 67.5   
## Security Data Scientist Intern (Security Execution and Performance) - 2025 Summer (BS/MS) : 18.5   
## Software Engineer Intern (Data Management Suite-Globalization) - 2025 Summer/Fall (BS/MS) : 45   
## Student Opportunities - Data Science Intern (PhD candidate) - Fall 2024 (Hybrid) : 18   
## Synergy Data Operator (2025) : 16   
## Temporary Data Engineer/Data Analyst Intern : 22.86   
## Trading Data & Analytics Intern Summer 2025 - Minneapolis, MN : 33   
## Virginia Talent + Opportunity Partnership Data Analytics Intern (#W0018) : 17   
## Volunteer Database Management Intern (Fall 2024) : 20

#### The variance of annual/Per Hour salary from mean Salary

#Group by categories  
std=vector(length = 2)  
  
std[1]=sd(df$Mean\_SalaryUSD[ df$Category=="Annual"])  
std[2]=sd(df$Mean\_SalaryUSD[ df$Category=="Per Hour" ])  
cat("The (Annual,Per Hour) Salary variation between jobs can attend :(",std[1],"$,",std[2],"$ )")

## The (Annual,Per Hour) Salary variation between jobs can attend :( 30178.7 $, 10.14544 $ )

#### the maximum and minimum salary in 2024

min\_max=function(df,category){  
 max\_salary=max(df$Mean\_Salary[df$Category == category])  
 min\_salary <- min(df$Mean\_Salary[df$Category == category])  
 max\_index <- which.max(df$Mean\_Salary[df$Category == category])  
 min\_index <- which.min(df$Mean\_Salary[df$Category == category])  
 return (c(max\_salary,max\_index,min\_salary,min\_index))  
}  
Annual\_max\_min=vector(length=4)  
Per\_Hour\_max\_min=vector(length=4)  
Annual\_max\_min=min\_max(df,"Annual")  
Per\_Hour\_max\_min=min\_max(df,"Per Hour")  
index\_min=as.numeric(Per\_Hour\_max\_min[4])  
index\_max=as.numeric(Per\_Hour\_max\_min[2])  
index\_min1=as.numeric(Annual\_max\_min[4])  
index\_max1=as.numeric(Annual\_max\_min[2])  
cat("The (minimum, maximum) salary in Per Hour are (",   
 Per\_Hour\_max\_min[3], "$,", Per\_Hour\_max\_min[1], "$) for",   
 df$Job\_Title[index\_min], " & ", df$Job\_Title[index\_max],   
 "jobs in", df$Company[index\_min], "and", df$Company[index\_max], " respectfully\n")

## The (minimum, maximum) salary in Per Hour are ( 8 $, 78.5 $) for Data Science- Internship opportunity (UT Austin/ University of Texas-San Marcos/ Texas A& M) -Onsite- Austin Tx. & Intern - Marketing, Play-Doh (Sept - Dec 2024) jobs in EPMA and Hasbro respectfully

cat("The (minimum, maximum) Annual salary are (",   
 Annual\_max\_min[3], "$,", Annual\_max\_min[1], "$) for",   
 df$Job\_Title[index\_min1], " & ", df$Job\_Title[index\_max1],   
 "jobs in", df$Company[index\_min1], "and", df$Company[index\_max1], " respectfully\n")

## The (minimum, maximum) Annual salary are ( 1000 $, 162500 $) for Research Intern & Data Analyst Intern jobs in California Institute of Genetics and InComm Payments respectfully

#### distribution of salary

quantiles\_salary=function(df,cat){  
 q1=quantile(df$Mean\_Salary[df$Category==cat],0.25)  
 q2=quantile(df$Mean\_Salary[df$Category==cat],0.5)  
 q3=quantile(df$Mean\_Salary[df$Category==cat],0.75)  
 return(c(q1,q2,q3))  
}  
Annualy=quantiles\_salary(df,"Annual")  
per\_hour=quantiles\_salary(df,"Per Hour")  
cat("25% of Annual salaries are under ",Annualy[1],"$.Median: is ",Annualy[2],"$,and 75% are equal or more then",Annualy[3],"$.")

## 25% of Annual salaries are under 49000 $.Median: is 62000 $,and 75% are equal or more then 81125 $.

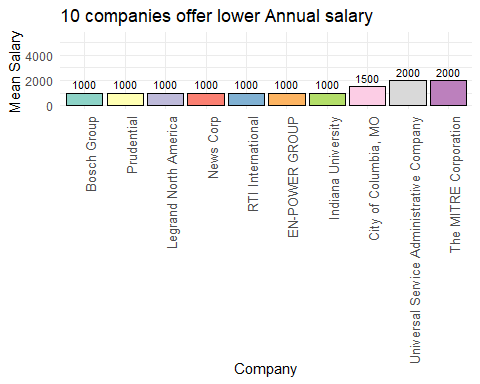
cat("25% of salaries in Per hour are under ",per\_hour[1],"$. Median: is ",per\_hour[2],"$, and 75% are equal or more then",per\_hour[3],"$.")

## 25% of salaries in Per hour are under 18 $. Median: is 21.875 $, and 75% are equal or more then 28.5 $.

### 3.5 Data Visualisation

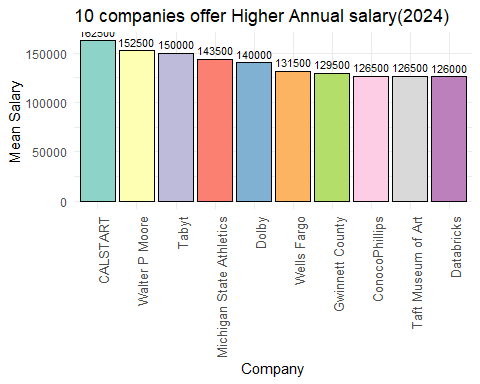
#### the 10 companies offer lower Annual salary

df1 <- df[order(df$Mean\_SalaryUSD),]  
lower\_sl <- df1$Mean\_SalaryUSD[df1$Category == "Annual"][1:10]  
label\_X <- vector(length = 10)  
for (i in seq\_len(10)) {  
 label\_X[i] <- paste(df1$Company[df1$Category == "Annual"][i], sep = "")  
}  
  
# Create a data frame for plotting  
plot\_data <- data.frame(  
 Mean\_Salary = lower\_sl,  
 Label = factor(label\_X, levels = label\_X) # Convert labels to factor to ensure correct order  
)  
options(repr.plot.width=45, repr.plot.height=5)  
# Create the bar plot with labels  
  
bar\_colors <- brewer.pal(length(plot\_data$Mean\_Salary), "Set3")  
ggplot(plot\_data, aes(x = Label, y = Mean\_Salary)) +  
 geom\_bar(stat = "identity", fill = bar\_colors, color = "black") +  
 geom\_text(aes(label = Mean\_Salary), vjust = -0.5, size = 3) +  
 labs(x = "Company", y = "Mean Salary", title = "10 companies offer lower Annual salary") +  
 ylim(0, 5500) + # Set the y-axis limits  
 theme\_minimal() +  
 theme(axis.text.x = element\_text(angle = 90, hjust = 1, size = 10))



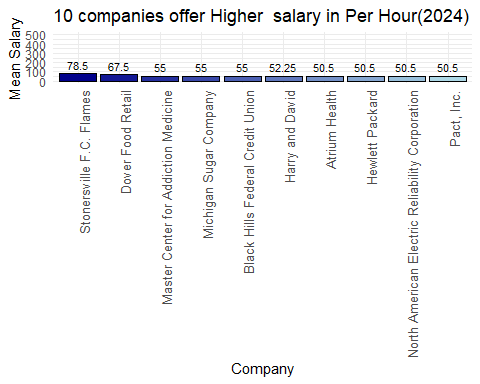
#### the 10 companies offer higher Annual salary

data=df[order(df$Mean\_SalaryUSD, decreasing = TRUE), ]  
higher\_sl <- data$Mean\_Salary[data$Category == "Annual"][1:10]  
label\_X1 <- vector(length = 10)  
for (i in seq\_len(10)) {  
 label\_X1[i] <- paste(data$Company[data$Category == "Annual"][i], sep = "")  
}  
  
# Create a data frame for plotting  
plot\_data1 <- data.frame(  
 Mean\_Salary = higher\_sl,  
 Label = factor(label\_X1, levels = label\_X1) # Convert labels to factor to ensure correct order  
)  
options(repr.plot.width=45, repr.plot.height=5)  
# Create the bar plot with labels  
  
bar\_colors <- brewer.pal(length(plot\_data1$Mean\_Salary), "Set3")  
ggplot(plot\_data1, aes(x = Label, y = Mean\_Salary)) +  
 geom\_bar(stat = "identity", fill = bar\_colors, color = "black") +  
 geom\_text(aes(label = Mean\_Salary), vjust = -0.5, size = 3) +  
 labs(x = "Company", y = "Mean Salary", title = "10 companies offer Higher Annual salary(2024)") +  
   
 theme\_minimal() +  
 theme(axis.text.x = element\_text(angle = 90, hjust = 1, size = 10))



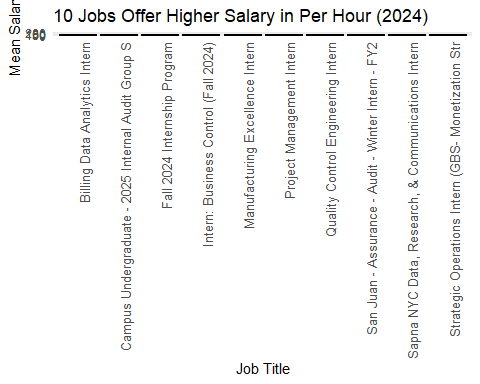
#### the 10 companies offer higher Per Hour salary

higher\_sl1 <- data$Mean\_SalaryUSD[data$Category == "Per Hour"][1:10]  
label\_X12 <- vector(length = 10)  
for (i in seq\_len(10)) {  
 label\_X12[i] <- paste(data$Company[data$Category == "Per Hour"][i], sep = "")  
}  
  
# Create a data frame for plotting  
plot\_data2 <- data.frame(  
 Mean\_Salary = higher\_sl1,  
 Label = factor(label\_X12, levels = label\_X12) # Convert labels to factor to ensure correct order  
)  
options(repr.plot.width=45, repr.plot.height=5)  
# Create the bar plot with labels  
  
num\_colors <- length(plot\_data2$Mean\_Salary)  
blue\_shades <- colorRampPalette(c( "darkblue","lightblue"))(num\_colors)  
ggplot(plot\_data2, aes(x = Label, y = Mean\_Salary)) +  
 geom\_bar(stat = "identity", fill = blue\_shades, color = "black") +  
 geom\_text(aes(label = Mean\_Salary), vjust = -0.5, size = 3) +  
 labs(x = "Company", y = "Mean Salary", title = "10 companies offer Higher salary in Per Hour(2024)") +  
 ylim(0, 500)+  
 theme\_minimal() +  
 theme(axis.text.x = element\_text(angle = 90, hjust = 1, size = 10))



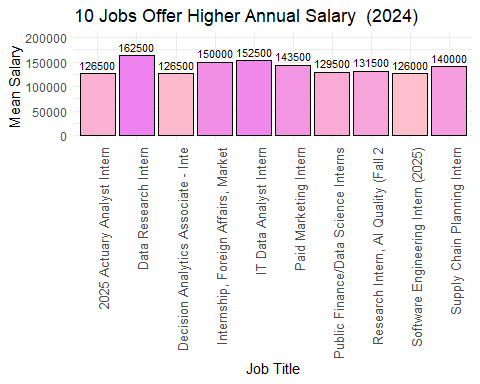
#### Jobs offer higher salary in Per Hour

data21 <- df[order(df$Mean\_SalaryUSD, decreasing = TRUE), c(3, 7, 8)]  
data2 <- data21[data21$Category == "Per Hour", ]  
data2\_ <- data2[1:10, ]  
data2\_$Job\_Title <- substr(data2\_$Job\_Title, 1, 50)  
# Define shades of blue  
num\_colors <- length(data2\_$Mean\_SalaryUSD)  
#blue\_shades <- colorRampPalette(c("orange", "yellow"))(num\_colors)  
blue\_shades1 <- colorRampPalette(c( "darkblue","lightblue"))(num\_colors)  
# Create the bar plot with shades of blue  
ggplot(data2\_, aes(x = Job\_Title, y = Mean\_SalaryUSD)) +  
 geom\_bar(stat = "identity", color = "black",fill=blue\_shades1) +  
 geom\_text(aes(label = Mean\_SalaryUSD), vjust = -0.5, size = 3) +  
 scale\_fill\_manual(values = blue\_shades1) + # Apply custom shades of blue  
 labs(x = "Job Title", y = "Mean Salary", title = "10 Jobs Offer Higher Salary in Per Hour (2024)") +  
 coord\_cartesian(ylim = c(0, 200)) + # Set y-axis limits  
 theme\_minimal() +  
 theme(axis.text.x = element\_text(angle = 90, hjust = 1, size = 10))



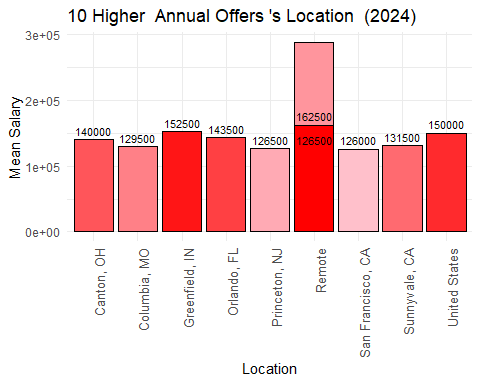
#### Jobs offer higher Annual salary

data3 <- data21[data21$Category == "Annual", ]  
data3\_ <- data3[1:10, ]  
data3\_$Job\_Title <- substr(data3\_$Job\_Title, 1, 35)  
# Define shades of blue  
num\_colors <- length(data3\_$Mean\_SalaryUSD)  
blue\_shades <- colorRampPalette(c("skyblue", "blue"))(num\_colors)  
  
  
blue\_shades7 <- colorRampPalette(c("violet", "pink"))(num\_colors)  
ggplot(data3\_, aes(x = Job\_Title, y = Mean\_SalaryUSD)) +  
 geom\_bar(stat = "identity", color = "black",fill=blue\_shades7) +  
 geom\_text(aes(label = Mean\_SalaryUSD), vjust = -0.5, size = 3) +  
 scale\_fill\_manual(values = blue\_shades) + # Apply custom shades of blue  
 labs(x = "Job Title", y = "Mean Salary", title = "10 Jobs Offer Higher Annual Salary (2024)") +  
 ylim(0, 200000) +  
 # Set y-axis limits  
 theme\_minimal() +  
 theme(axis.text.x = element\_text(angle = 90, hjust = 1, size = 10))



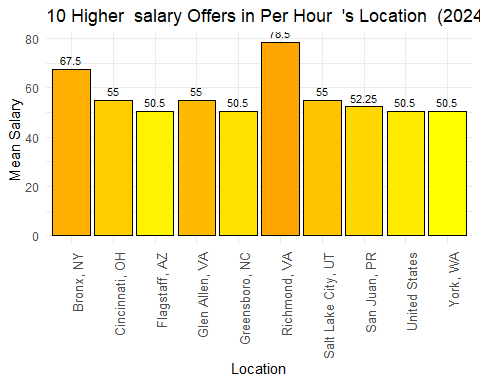
#### Higher Annual Offers ’s Location

data\_4=data[data$Category=="Annual",]  
data\_4=data\_4[1:10,]  
num\_colors <- length(data\_4$Mean\_SalaryUSD)  
blue\_shades2 <- colorRampPalette(c("red", "pink"))(num\_colors)  
  
# Create the bar plot with shades of blue  
ggplot(data\_4, aes(x = Location, y = Mean\_SalaryUSD)) +  
 geom\_bar(stat = "identity", color = "black",fill=blue\_shades2) +  
 geom\_text(aes(label = Mean\_SalaryUSD), vjust = -0.5, size = 3) +  
 scale\_fill\_manual(values = blue\_shades) + # Apply custom shades of blue  
 labs(x = "Location", y = "Mean Salary", title = "10 Higher Annual Offers 's Location (2024)") +  
 # Set y-axis limits  
 theme\_minimal() +  
 theme(axis.text.x = element\_text(angle = 90, hjust = 1, size = 10))



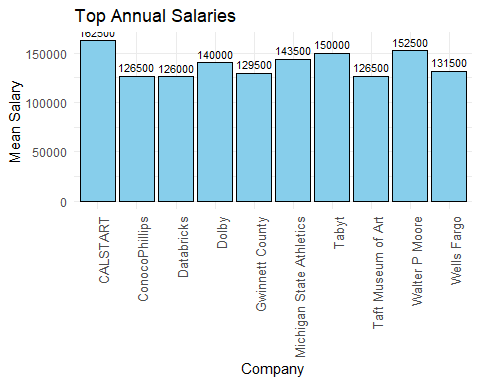
#### Higher Offers salary in Per Hour ’s Location

data\_5=data[data$Category=="Per Hour",]  
data\_5=data\_5[1:10,]  
num\_colors <- length(data\_5$Mean\_SalaryUSD)  
blue\_shades5 <- colorRampPalette(c("orange", "yellow"))(num\_colors)  
  
ggplot(data\_5, aes(x = Location, y = Mean\_SalaryUSD)) +  
 geom\_bar(stat = "identity", color = "black",fill=blue\_shades5 ) +  
 geom\_text(aes(label = Mean\_SalaryUSD), vjust = -0.5, size = 3) +  
 scale\_fill\_manual(values = blue\_shades) + # Apply custom shades of blue  
 labs(x = "Location", y = "Mean Salary", title = "10 Higher salary Offers in Per Hour 's Location (2024)") +  
 # Set y-axis limits  
 theme\_minimal() +  
 theme(axis.text.x = element\_text(angle = 90, hjust = 1, size = 10))

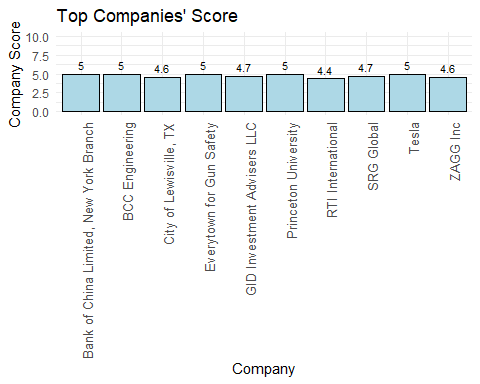


#### Analysis company\_Score

# Filter and prepare data for the second plot  
data6 <- df[order(df$Company\_Score, decreasing = TRUE), ]  
data6 <- data6[data6$Category == "Annual", ]  
data6 <- data6[1:10, ]  
  
  
# First plot  
 ggplot(data\_4, aes(x = Company, y = Mean\_SalaryUSD)) +  
 geom\_bar(stat = "identity", color = "black", fill = "skyblue") +  
 geom\_text(aes(label = Mean\_SalaryUSD), vjust = -0.5, size = 3) +  
 labs(x = "Company", y = "Mean Salary", title = "Top Annual Salaries") +  
 theme\_minimal() +  
 theme(axis.text.x = element\_text(angle = 90, hjust = 1, size = 10))



# Second plot  
ggplot(data6, aes(x = Company, y = Company\_Score)) +  
 geom\_bar(stat = "identity", color = "black", fill = "lightblue") +  
 geom\_text(aes(label = Company\_Score), vjust = -0.5, size = 3) +  
 labs(x = "Company", y = "Company Score", title = "Top Companies' Score") +  
 ylim(0, 10) +  
 theme\_minimal() +  
 theme(axis.text.x = element\_text(angle = 90, hjust = 1, size = 10))



# 4.Conclusion

The data provided valuable insights, indicating that remote work has firmly established itself as a significant trend, with many companies increasingly adopting this model. Notably, the salary associated with remote positions is substantial. In parallel, the dataset shows that this highlights the growing importance of data in decision-making processes, as companies seek to leverage their data to attract skilled professionals. Data-related jobs are in high demand, with these professionals commanding some of the highest salaries.

Moreover, the analysis reveals that a company’s success and high ratings are not solely determined by salary. Other factors, such as employee well-being, psychological comfort, and a supportive work environment, play crucial roles. This is evident when examining the top 10 companies by score, where high salaries alone did not guarantee the highest rankings. For example, although CaLSTART offers competitive salaries, it was not among the top 10 companies with the highest scores, underscoring the importance of a holistic approach to employee satisfaction.