```
##### PROJECT 1 WITH R #####

#Loaded necessary libraries
library(ggplot2)# This library enables layering of data and aesthetics, making it easy to build complex visualizations by adding layers incrementally.
library(dplyr)# This library focuses on simplicity and performance for tasks like filtering, selecting, arranging, mutating, and summarizing data.
```

```
## 
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
## 
##     filter, lag
```

```
## The following objects are masked from 'package:base':
## 
##     intersect, setdiff, setequal, union
```

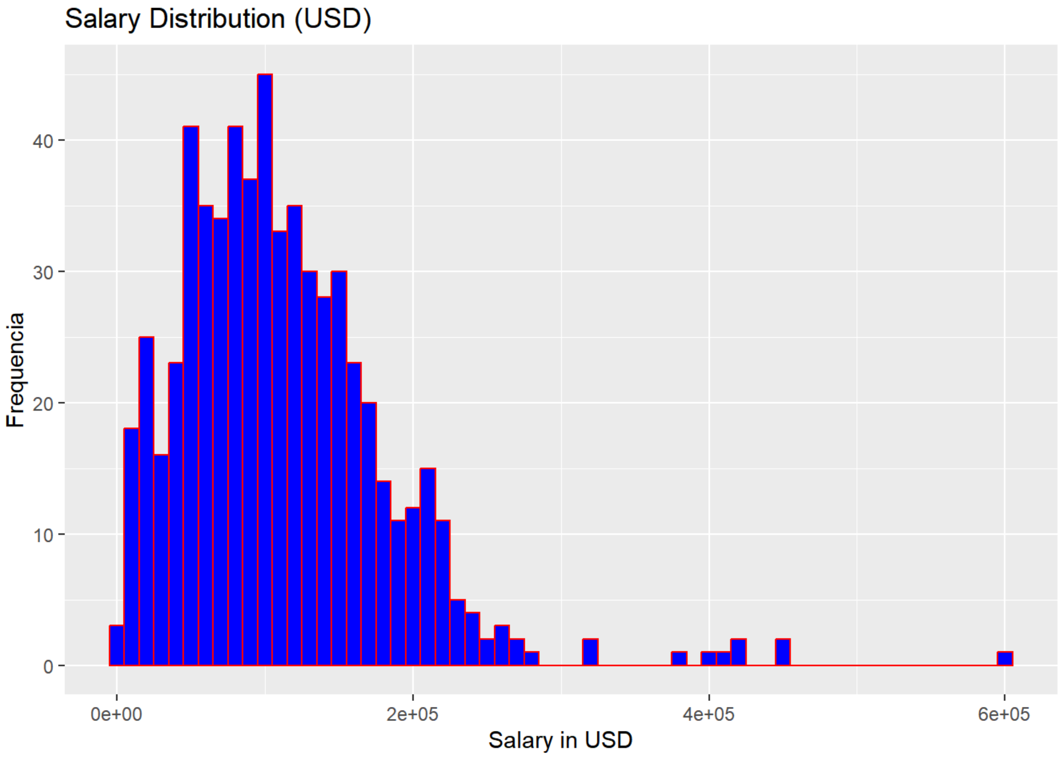
```
data=read.csv("C:\\Users\\danar\\Desktop\\week_5\\r project data_este.csv")
str(data)#Data Frame
```

```
## '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" ...
```

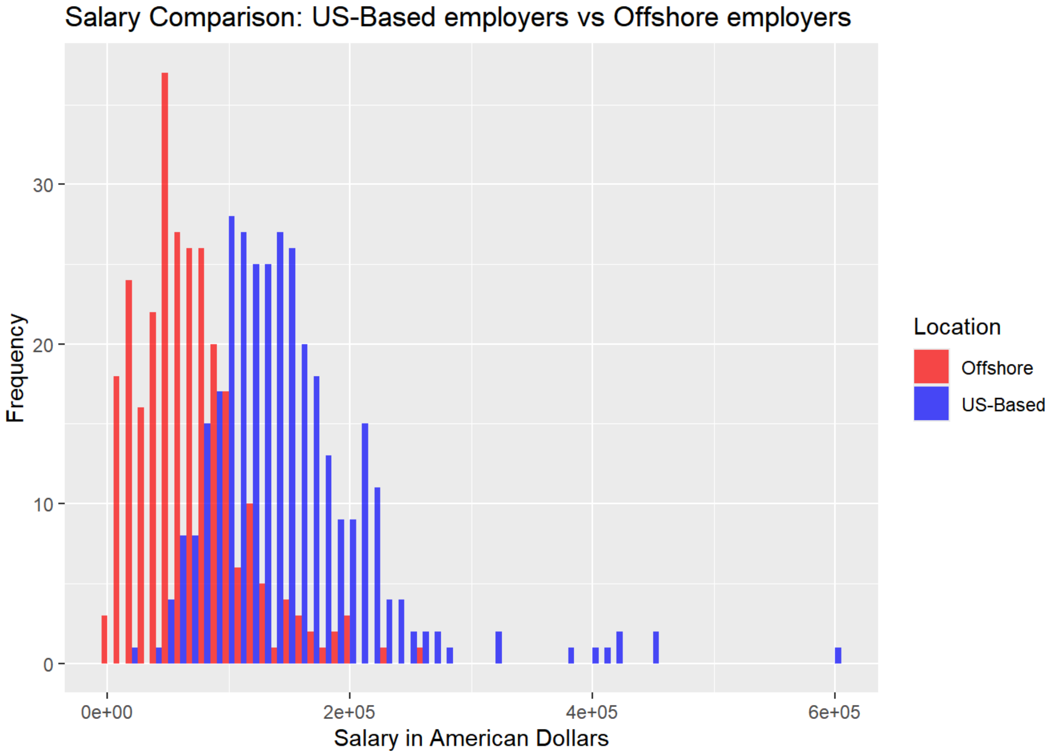
```
summary(data)#607 obs. of  12 variables
```

```
##        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
```

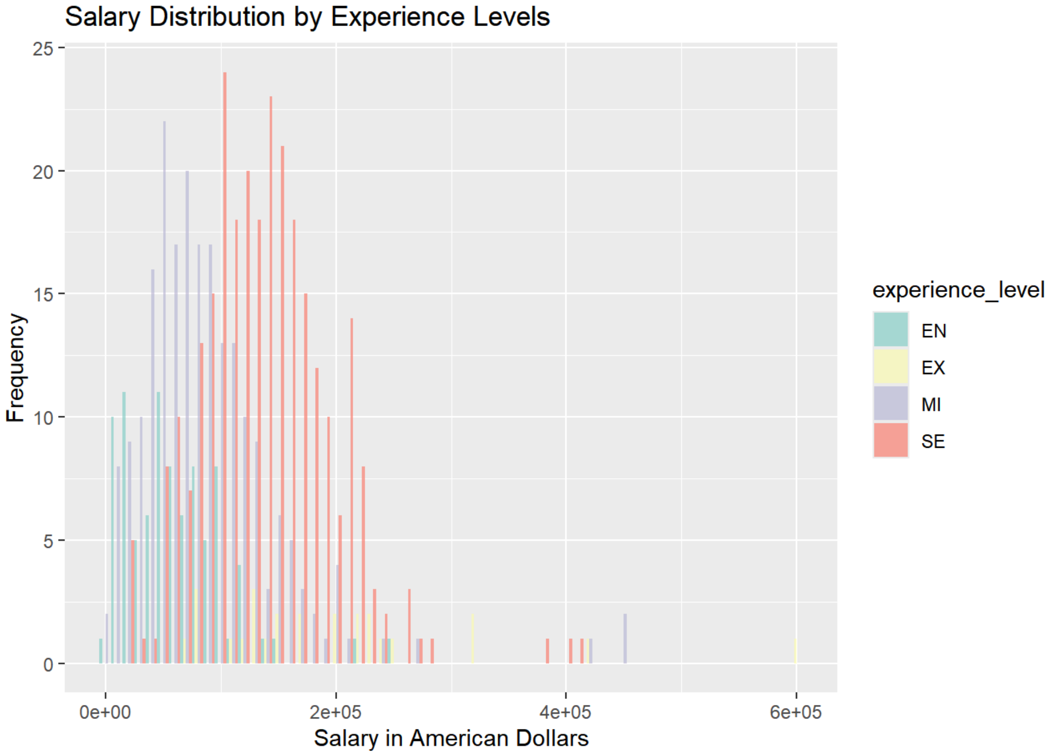
```
#First, is imortant to understand the distribution for All Employees
#This histogram show the overall distributions of salaries in USD for all employees in the dataset.
#Helps identify the general range and frequency of salaries
##### 1. Salary Distribution for All Employees
ggplot(data, aes(x=salary_in_usd)) +
geom_histogram(binwidth=10000, fill="blue", color="red") +
labs(title="Salary Distribution (USD)", x="Salary in USD", y="Frequencia")
```



```
##### 2. Salary Comparison: U.S. vs Offshore
#This histogram compares the salary distributions for employees based in the United States (US-Based) versus those offshore. Each bar is split by location
#Compare the salary ranges for U.S.-based employees against offshore employees
data= data %>%
mutate(Location=ifelse(employee_residence == "US", "US-Based", "Offshore"))
ggplot(data, aes(x=salary_in_usd, fill=Location)) +
geom_histogram(binwidth=10000, position="dodge", alpha=0.7) +
labs(title = "Salary Comparison: US-Based employers vs Offshore employers", x="Salary in American Dollars", y="Frequency") +
scale_fill_manual(values = c("US-Based" = "blue", "Offshore" = "red"))
```

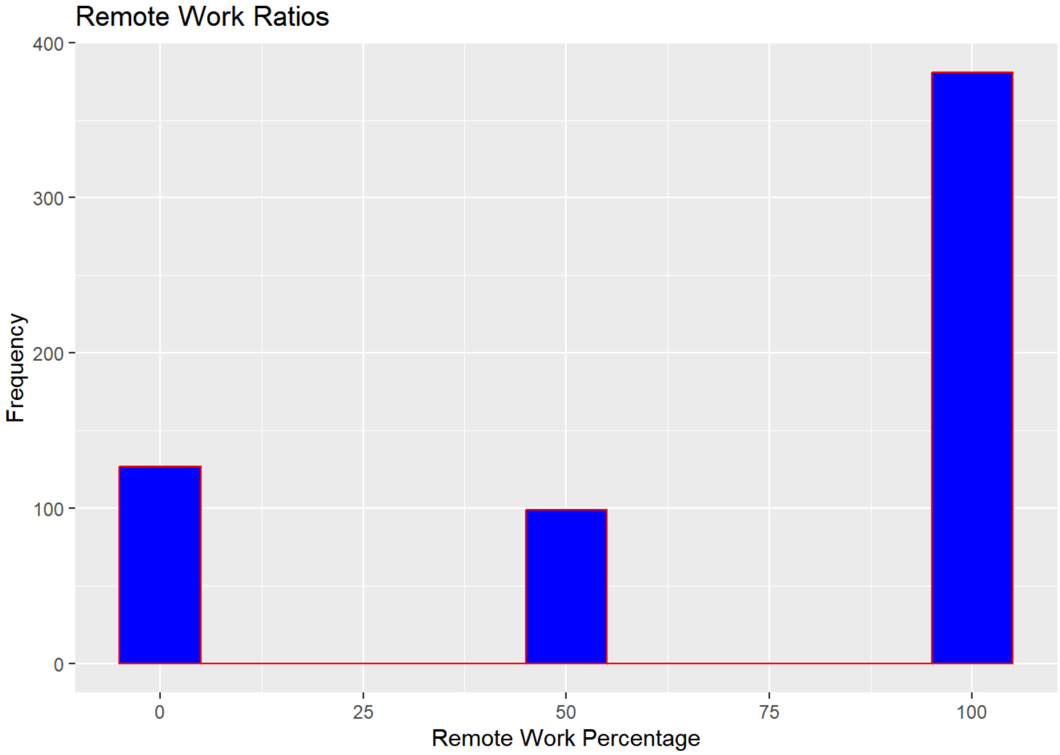


```
##### 3. Salary by Experience Level
#This histogram divides the salary data based on experience levels (Entry-Level, Mid-Level, Senior-Level, Executive-Leve
l). Each group has its own color
#This allows to see trends, such as higher salaries for Senior and Executive roles, and helps ensure that the company offe
rs competitive rates for experience levels
ggplot(data, aes(x=salary_in_usd, fill=experience_level)) +#"fill is an aesthetic that determines the color used to fill e
lements in the plot" Not sure how this part of the ggplot works, but without it, whe I try to run it it give me error
geom_histogram(binwidth=10000, position="dodge", alpha=0.7) +#"dodge" places overlapping bars, which I think make it looks
easier to read
labs(title="Salary Distribution by Experience Levels", x = "Salary in American Dollars", y = "Frequency") +
scale_fill_brewer(palette="Set3")#The other palette were hard to see the different colors
```



```
#scale_fill_brewer, I found this function when I asked to an AI, different ways to make histogrmas. An histogram, in my op
innion shows better the different informations in a graph rather a plot chart which is a little confusing
```

```
#This histogram shows the frequency of different remote work ratios: No remote work, partially remote, fully remote levels
in other words this shows correlation between remote work and salary
# 4. Remote Work Ratios
ggplot(data, aes(x=remote_ratio)) +
geom_histogram(binwidth = 10, fill="blue", color="red") +#Blue and re to show contrast
labs(title="Remote Work Ratios", x="Remote Work Percentage", y="Frequency")
```



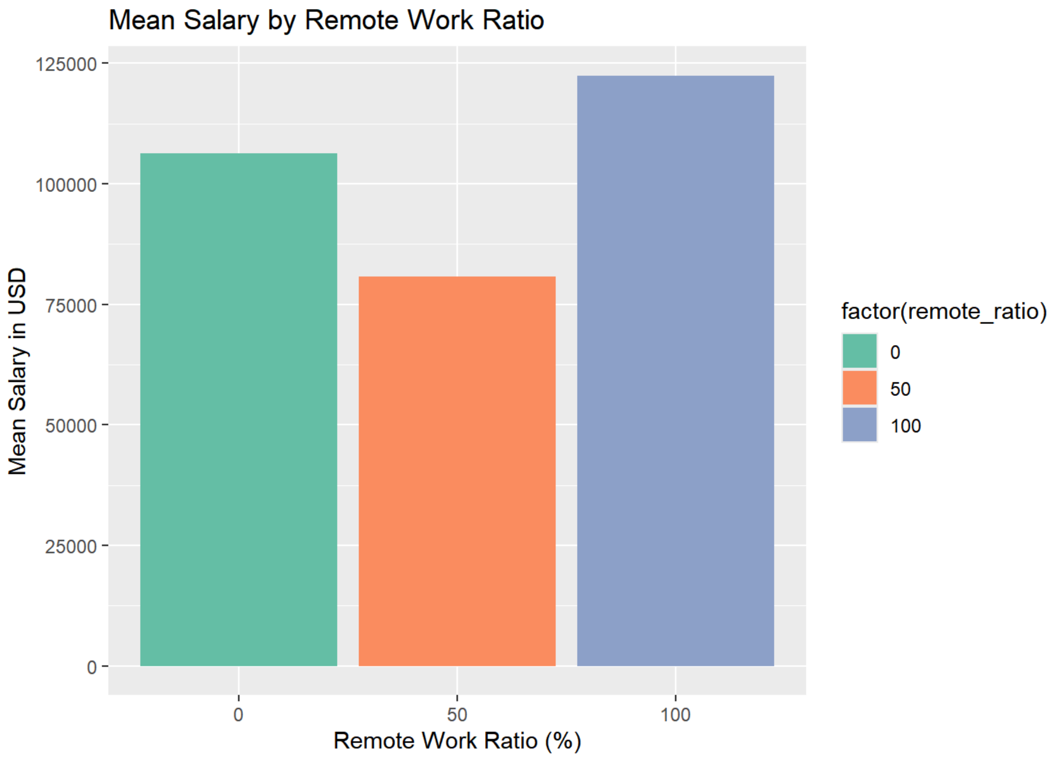
```
#It gave me error, I am reloading the package
library(ggplot2)
library(dplyr)


# Group data by job title and calculate summary statistics
job_title_summary=data %>%
group_by(job_title) %>%
summarise(
Mean_Salary=mean(salary_in_usd, na.rm = TRUE),
Median_Salary=median(salary_in_usd, na.rm = TRUE),
Min_Salary=min(salary_in_usd, na.rm = TRUE),
Max_Salary=max(salary_in_usd, na.rm = TRUE),
Count=n()
) %>%
arrange(desc(Mean_Salary))#In descend show better the difference
print(job_title_summary)
```

```
## # A tibble: 50 × 6
##    job_title               Mean_Salary Median_Salary Min_Salary Max_Salary Count
##    <chr>                         <dbl>         <dbl>      <int>      <int> <int>
##  1 Data Analytics Lead          405000        405000     405000     405000     1
##  2 Principal Data Engineer      328333.       200000     185000     600000     3
##  3 Financial Data Analyst       275000        275000     100000     450000     2
##  4 Principal Data Scienti…      215242.       173762     148261     416000     7
##  5 Director of Data Scien…      195074        168000     130026     325000     7
##  6 Data Architect               177874.       180000      90700     266400    11
##  7 Applied Data Scientist       175655        157000      54238     380000     5
##  8 Analytics Engineer           175000        179850     135000     205300     4
##  9 Data Specialist              165000        165000     165000     165000     1
## 10 Head of Data                 160163.       200000      32974     235000     5
## # ℹ 40 more rows
```

```
# Load required libraries, just in case, last part gave me error, I am reloading the package
library(ggplot2)
library(dplyr)
# Grouping data by remote work ratio and calculate summary statistics
remote_work_summary <- data %>%
group_by(remote_ratio) %>%
summarise(
Mean_Salary=mean(salary_in_usd, na.rm = TRUE),
Median_Salary=median(salary_in_usd, na.rm = TRUE),
Min_Salary=min(salary_in_usd, na.rm = TRUE),
Max_Salary=max(salary_in_usd, na.rm = TRUE),
Count=n()
) %>%
arrange(remote_ratio)#just in case to keep it neat
print(remote_work_summary)
```

```
## # A tibble: 3 × 6
##   remote_ratio Mean_Salary Median_Salary Min_Salary Max_Salary Count
##          <int>       <dbl>         <int>      <int>      <int> <int>
## 1            0     106355.         99000       2859     450000   127
## 2           50      80823.         69999       5409     423000    99
## 3          100     122457.        115000       4000     600000   381
```

```
ggplot(remote_work_summary, aes(x=factor(remote_ratio), y=Mean_Salary, fill=factor(remote_ratio))) +
geom_bar(stat="identity") +
labs(
title="Mean Salary by Remote Work Ratio",
x="Remote Work Ratio (%)",
y="Mean Salary in USD"
)+
scale_fill_brewer(palette="Set2")
```



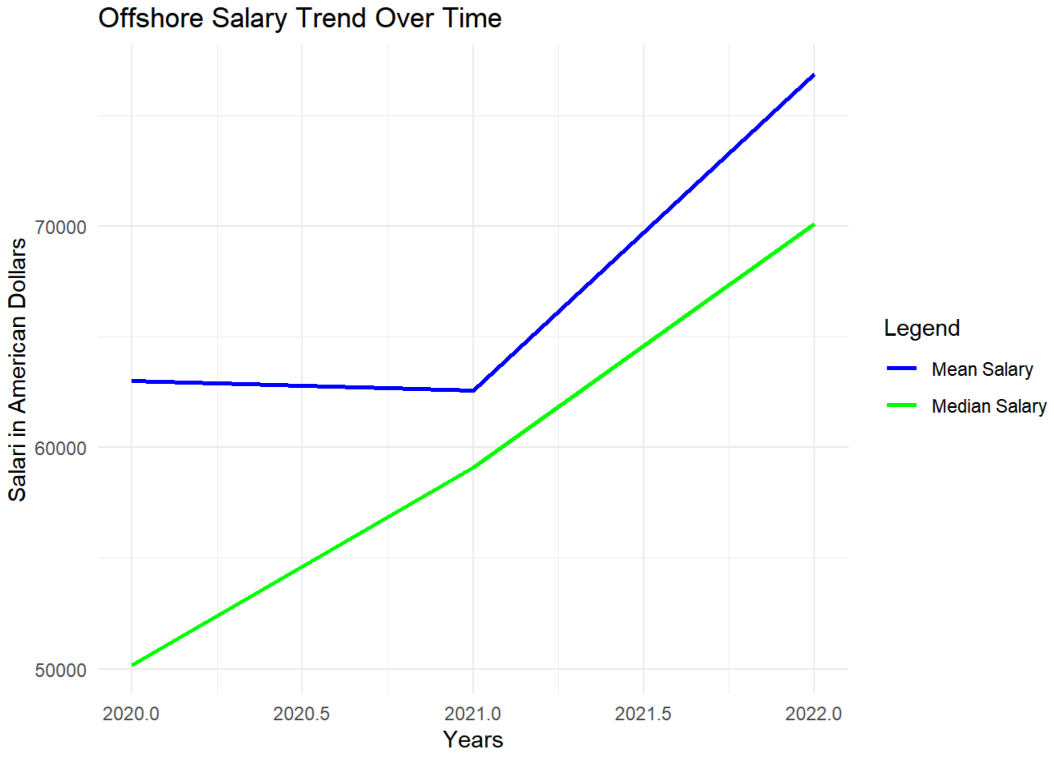
```
#Shows the average, median, minimum, and maximum salaries for each remote work category (0, 50, 100).
#Helps identify whether fully remote roles (100%) offer higher or lower salaries on averag
```

```
offshore_data=data %>%
filter(employee_residence!="US")
#Solo los del extranjero!!!
##### Group data by work year and calculate salary statistics
offshore_trend=offshore_data %>%
group_by(work_year) %>%
summarise(
Mean_Salary=mean(salary_in_usd, na.rm=TRUE),
Median_Salary=median(salary_in_usd, na.rm=TRUE),
Min_Salary=min(salary_in_usd, na.rm=TRUE),
Max_Salary=max(salary_in_usd, na.rm=TRUE),
Count=n()
)
#Para ver que tal esta
print(offshore_trend)
```

```
## # A tibble: 3 × 6
##   work_year Mean_Salary Median_Salary Min_Salary Max_Salary Count
##       <int>       <dbl>         <dbl>      <int>      <int> <int>
## 1      2020      63021.         50180       5707     260000    47
## 2      2021      62572.         59102       2859     230000   130
## 3      2022      76898.         70124      10000     200000    98
```

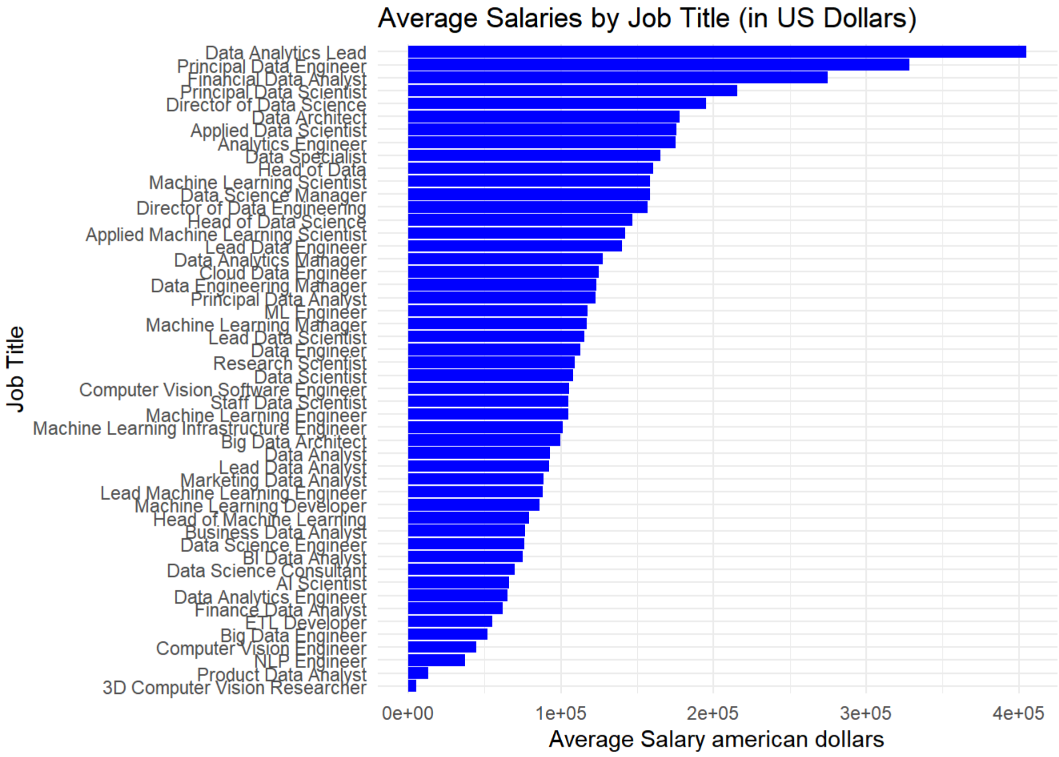
```
ggplot(offshore_trend, aes(x=work_year)) +
geom_line(aes(y=Mean_Salary, color="Mean Salary"), size=1) +
geom_line(aes(y=Median_Salary, color="Median Salary"), size=1) +
labs(
title="Offshore Salary Trend Over Time",
x="Years",
y="Salari in American Dollars",
color="Legend"
)+
scale_color_manual(values = c("Mean Salary" = "blue", "Median Salary" = "green")) +
theme_minimal()
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## ℹ Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



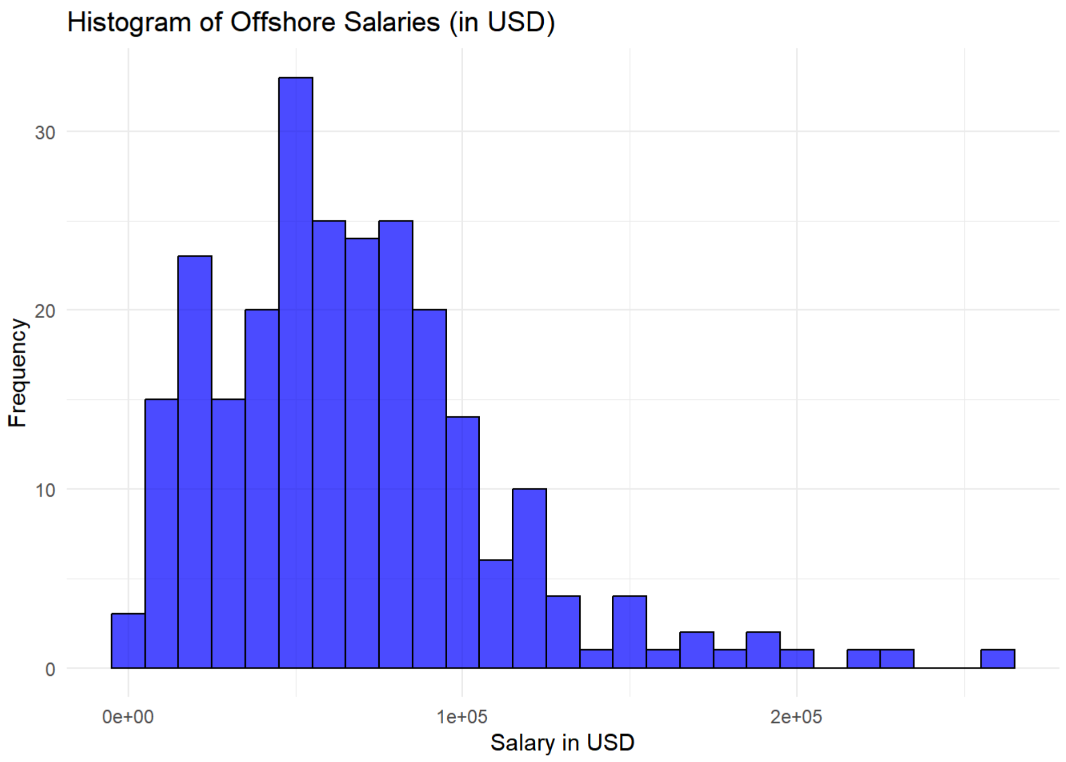
```
#Here I enter codes to analyze salary trends offshore, the code focus on employees who are not based in the United States
and examine how salaries have changed over different times. This can help understand if offshore salaries are increasing o
r stable
#If the mean salary increasing, it indicate a general upward trending in offshore salaries
```

```
library(ggplot2)#just in case!
data=read.csv("r project data_este.csv")

#Calculation of average salary by job titles, for this part I search online how to this part, it gave me many times erro after error in syntaxis
avg_salary_by_job=aggregate(salary_in_usd ~ job_title, data = data, FUN = mean)

#Bar chart (No se si deberia cambiarlo pero lo dejare asi)
ggplot(avg_salary_by_job, aes(x = reorder(job_title, salary_in_usd), y = salary_in_usd)) +
  geom_bar(stat ="identity", fill = "blue") +
  coord_flip() +
  labs(title = "Average Salaries by Job Title (in US Dollars)",
       x = "Job Title",
       y = "Average Salary american dollars") +
  theme_minimal()
```



```
#Not sure how to seperate the names
```

```
data=read.csv("r project data_este.csv")

#Filter data for offshore employees
offshore_data=subset(data,company_location!="US")
#Create a histogram for offshore salaries in USD
ggplot(offshore_data, aes(x = salary_in_usd)) +
geom_histogram(binwidth = 10000, fill = "blue", color = "black", alpha = 0.7) +
labs(
title ="Histogram of Offshore Salaries (in USD)",
x = "Salary in USD",
y = "Frequency"
) +
theme_minimal()
```


Histogram of Offshore Salaries (in USD)
30
20
10
0
Frequency
0e+00
1e+05
2e+05
Salary in USD