FinalProject

GROUP-11

2023-12-07

```
sal<- read.csv("salaries (1).csv")
```

```
#null values
sum(is.na(sal))
```

```
## [1] 0
```

```
#unique values in each column
sapply(sal, function(x) length(unique(x)))
```

```
##       work_year  experience_level   employment_type        job_title
##               4                 4                 4              125
##          salary   salary_currency     salary_in_usd employee_residence
##            1516                22              1799                 86
##     remote_ratio  company_location      company_size
##                3                74                 3
```

```
#label encoding for `experience_level` column INTO `experience_level_label`
sal$experience_level_label <- as.numeric(factor(sal$experience_level))
```

```
#label encoding for employment_type column INTO employment_type_label
sal$employment_type_label <- as.numeric(factor(sal$employment_type))
```

```
#label encoding for company_size column INTO company_size_label
sal$company_size_label <- as.numeric(factor(sal$company_size))
```

```
head(sal)
```

```
##   work_year experience_level employment_type                job_title
## 1      2023             SE              FT            Data Engineer
## 2      2023             SE              FT            Data Engineer
## 3      2023             SE              FT             Data Analyst
## 4      2023             SE              FT             Data Analyst
## 5      2023             SE              FT Business Intelligence Analyst
## 6      2023             SE              FT Business Intelligence Analyst
##   salary salary_currency salary_in_usd employee_residence remote_ratio
## 1 185000           USD       185000              US        100
## 2 150000           USD       150000              US        100
## 3 166000           USD       166000              US          0
## 4  73100           USD        73100              US          0
## 5 198200           USD       198200              US          0
## 6 110200           USD       110200              US          0
##   company_location company_size experience_level_label employment_type_label
## 1               US          M                       4                     3
## 2               US          M                       4                     3
## 3               US          M                       4                     3
## 4               US          M                       4                     3
## 5               US          M                       4                     3
## 6               US          M                       4                     3
##   company_size_label
## 1                  2
## 2                  2
## 3                  2
## 4                  2
## 5                  2
## 6                  2
```

```
library(readr)
library(ggplot2)
sal<- read_csv('salaries (1).csv')
```

```
## Rows: 9074 Columns: 11
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr (7): experience_level, employment_type, job_title, salary_currency, empl...
## dbl (4): work_year, salary, salary_in_usd, remote_ratio
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
head(sal)
```

```
## # A tibble: 6 × 11
##   work_year experience_level employment_type job_title    salary salary_currency
##       <dbl> <chr>            <chr>           <chr>         <dbl> <chr>
## 1      2023 SE               FT              Data Engine… 185000 USD
## 2      2023 SE               FT              Data Engine… 150000 USD
## 3      2023 SE               FT              Data Analyst 166000 USD
## 4      2023 SE               FT              Data Analyst  73100 USD
## 5      2023 SE               FT              Business In… 198200 USD
## 6      2023 SE               FT              Business In… 110200 USD
## # ℹ 5 more variables: salary_in_usd <dbl>, employee_residence <chr>,
## #   remote_ratio <dbl>, company_location <chr>, company_size <chr>
```

```
#if salary_currency is other than usd then convert salary_in_usd to USD sal$salary_in_usd <- ifelse(sal$salary_currency == "AUD", sal$salary_in_usd * 0.65, sal$salary_in_usd) sal$salary_in_usd <- ifelse(sal$salary_currency == "BRL", sal$salary_in_usd * 0.2, sal$salary_in_usd) sal$salary_in_usd <- ifelse(sal$salary_currency == "CAD", sal$salary_in_usd * 0.74, sal$salary_in_usd) sal$salary_in_usd <- ifelse(sal$salary_currency == "CHF", sal$salary_in_usd * 1.14, sal$salary_in_usd) sal$salary_in_usd <- ifelse(sal$salary_currency == "DKK", sal$salary_in_usd * 0.14, sal$salary_in_usd) sal$salary_in_usd <- ifelse(sal$salary_currency == "EUR", sal$salary_in_usd * 1.08, sal$salary_in_usd) sal$salary_in_usd <- ifelse(sal$salary_currency == "GBP", sal$salary_in_usd * 1.26, sal$salary_in_usd) sal$salary_in_usd <- ifelse(sal$salary_currency == "HKP", sal$salary_in_usd * 0.13, sal$salary_in_usd) sal$salary_in_usd <- ifelse(sal$salary_currency == "HUF", sal$salary_in_usd * 0.0028, sal$salary_in_usd) sal$salary_in_usd <- ifelse(sal$salary_currency == "ILS", sal$salary_in_usd * 0.27, sal$salary_in_usd) sal$salary_in_usd <- ifelse(sal$salary_currency == "INR", sal$salary_in_usd * 0.012, sal$salary_in_usd) sal$salary_in_usd <- ifelse(sal$salary_currency == "JPY", sal$salary_in_usd * 0.0068, sal$salary_in_usd) sal$salary_in_usd <- ifelse(sal$salary_currency == "MXN", sal$salary_in_usd * 0.058, sal$salary_in_usd) sal$salary_in_usd <- ifelse(sal$salary_currency == "NOK", sal$salary_in_usd * 0.091, sal$salary_in_usd) sal$salary_in_usd <- ifelse(sal$salary_currency == "PHP", sal$salary_in_usd * 0.018, sal$salary_in_usd) sal$salary_in_usd <- ifelse(sal$salary_currency == "PLN", sal$salary_in_usd * 0.25, sal$salary_in_usd) sal$salary_in_usd <- ifelse(sal$salary_currency == "SGD", sal$salary_in_usd * 0.75, sal$salary_in_usd) sal$salary_in_usd <- ifelse(sal$salary_currency == "THB", sal$salary_in_usd * 0.028, sal$salary_in_usd) sal$salary_in_usd <- ifelse(sal$salary_currency == "TRY", sal$salary_in_usd * 0.035, sal$salary_in_usd) sal$salary_in_usd <- ifelse(sal$salary_currency == "ZAR", sal$salary_in_usd * 0.053, sal$salary_in_usd)
```

```
#remove salary_currency column
#sal$salary_currency <- NULL
#sal$salary_in_usd<-salaries$salary_in_usd / 1000
```

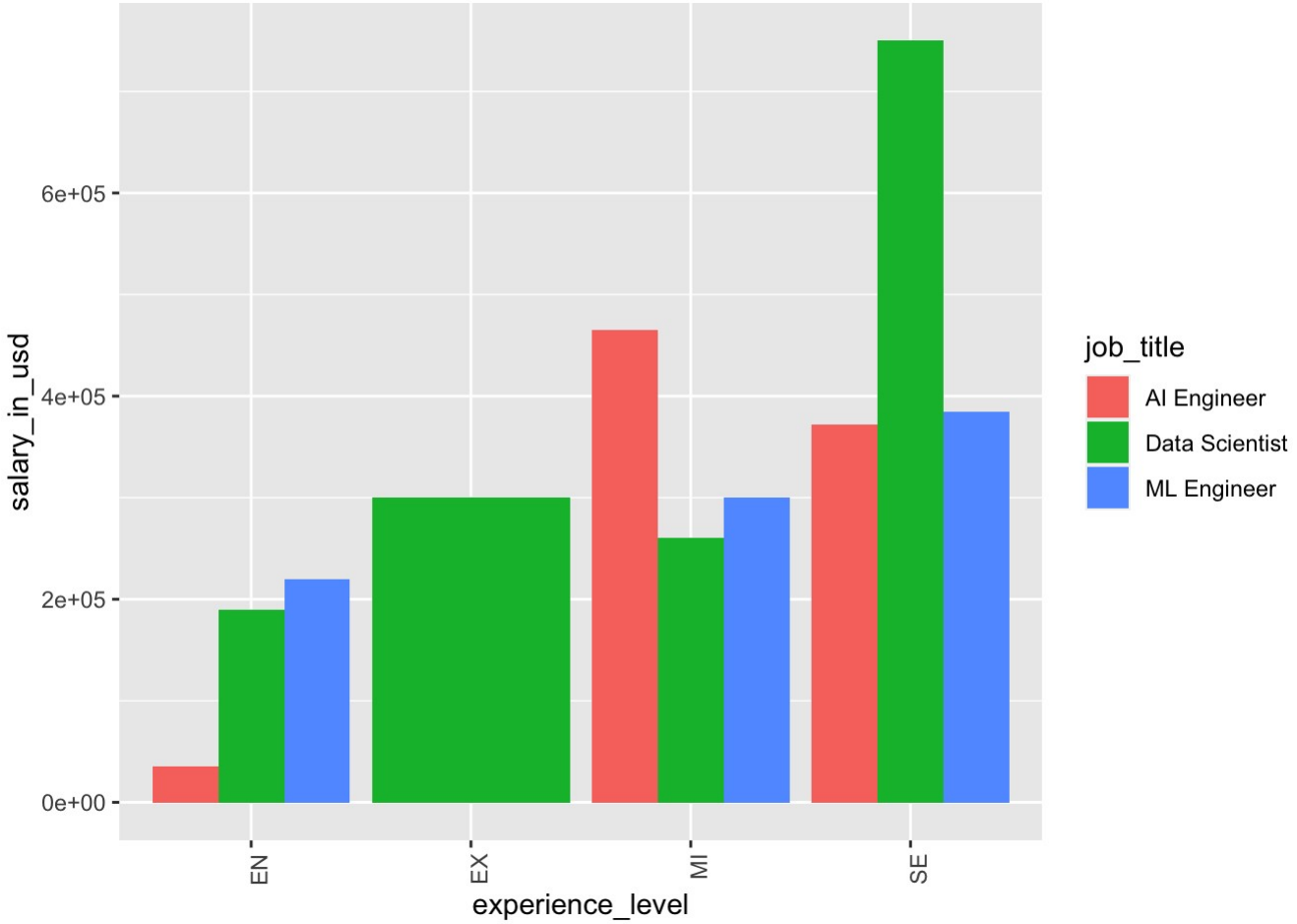
```
#access data with job_title as Data Scientist,AI Engineer,ML Engineer
salaries_1 <- subset(sal, job_title %in% c("Data Scientist", "AI Engineer", "ML Engineer"))
```

```
unique(salaries_1$job_title)
```

```
## [1] "AI Engineer"     "Data Scientist" "ML Engineer"
```

#How does the salary distribution on vary across different experience levels and job Ntles within the AI, ML, and Data Science domains globally?

```
ggplot(salaries_1, aes(x = experience_level, y = salary_in_usd, fill = job_title)) + geom_bar(stat="identity",
position = "dodge") + theme(axis.text.x = element_text(angle =
90, hjust = 1))
```




```
summary(sal)
```

```
##   work_year   experience_level   employment_type    job_title
## Min.   :2020   Length:9074        Length:9074        Length:9074
## 1st Qu.:2023   Class :character   Class :character   Class :character
## Median :2023   Mode  :character   Mode  :character   Mode  :character
## Mean   :2023
## 3rd Qu.:2023
```

```
## Max.   :2023
##    salary       salary_currency   salary_in_usd     employee_residence
## Min.   :  14000   Length:9074       Min.   :   49.5   Length:9074
## 1st Qu.:  105500   Class :character   1st Qu.:106020.0   Class :character
## Median :  144100   Mode  :character   Median :143000.0   Mode  :character
## Mean   :  174104                      Mean   :150177.0
## 3rd Qu.:  190000                      3rd Qu.:186600.0
## Max.   :30400000                      Max.   :750000.0
##  remote_ratio   company_location   company_size
## Min.   :  0.00   Length:9074       Length:9074
## 1st Qu.:  0.00   Class :character   Class :character
## Median :  0.00   Mode  :character   Mode  :character
## Mean   : 38.52
## 3rd Qu.:100.00
## Max.   :100.00
```

```
cor_matrix <- cor(sal[, c("salary_in_usd", "remote_ratio")])
print("Correlation Matrix:")
```

```
## [1] "Correlation Matrix:"
```

```
print(cor_matrix)
```

```
##                  salary_in_usd  remote_ratio
## salary_in_usd     1.00000000   -0.09717609
## remote_ratio     -0.09717609    1.00000000
```

```
unique_levels <- unique(sal$experience_level)


anova_result <- aov(salary_in_usd ~ experience_level, data = sal)


print(summary(anova_result))
```

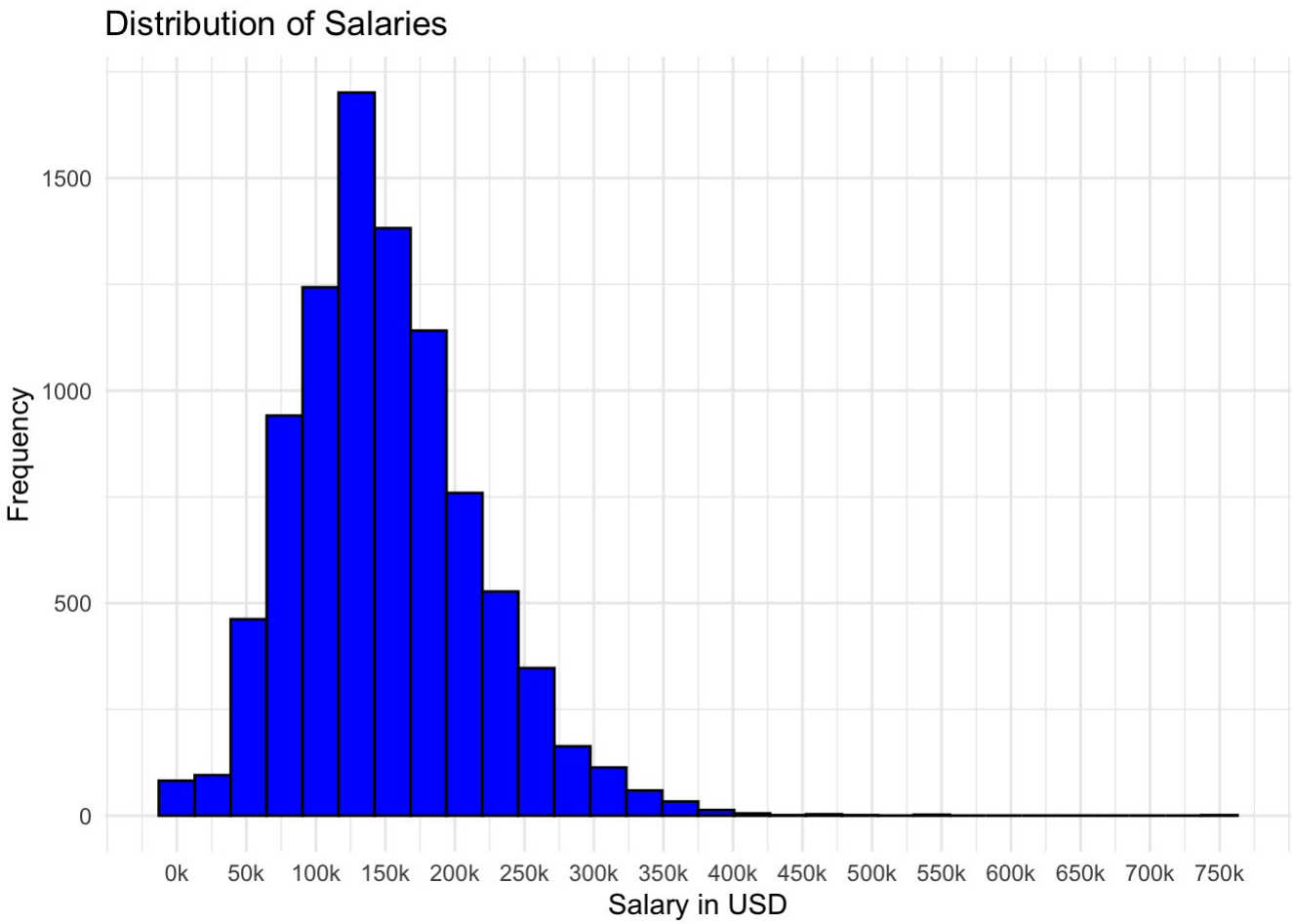
```
##                      Df    Sum Sq   Mean Sq F value Pr(>F)
## experience_level     3 5.343e+12 1.781e+12   496.9 <2e-16 ***
## Residuals         9070 3.251e+13 3.584e+09
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
posthoc_result <- TukeyHSD(anova_result)
print(posthoc_result)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = salary_in_usd ~ experience_level, data = sal)
##
## $experience_level
##              diff        lwr        upr p adj
## EX-EN 103273.96  91706.38 114841.55     0
## MI-EN  30068.09  22182.44  37953.74     0
## SE-EN  75563.60  68309.39  82817.81     0
## MI-EX -73205.88 -83105.69 -63306.06     0
## SE-EX -27710.37 -37114.96 -18305.77     0
## SE-MI  45495.51  41394.18  49596.85     0
```

here we started doing the dataanalysis for the dataset for the understanding purpose

```
ggplot(sal, aes(x = salary_in_usd)) +
  geom_histogram(bins = 30, fill = "blue", color = "black") +
  theme_minimal() +
  labs(title = "Distribution of Salaries", x = "Salary in USD", y = "Frequency") +
  scale_x_continuous(
    labels = scales::number_format(scale = 1e-3, suffix = "k"),
    breaks = seq(0, max(sal$salary_in_usd), by = 50000)
  )
```



here we can see the distribution os

salaries which are starting from 30k to 300k

Is there a significant difference in salaries based on the extent of remote work, and does this vary across countries or company sizes?

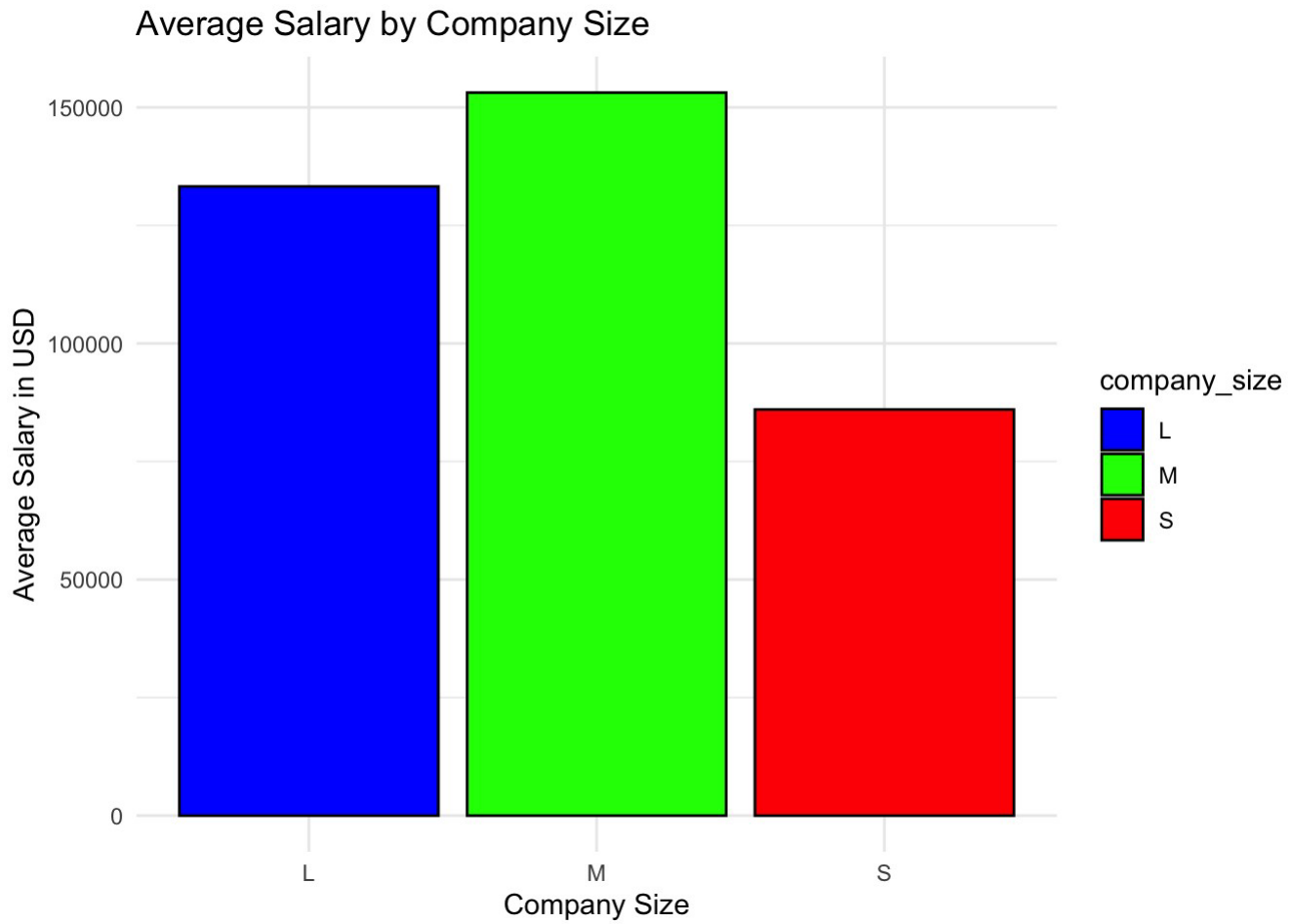
```
library(dplyr)
```

```
##
## 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
```

```
sal %>%
  group_by(company_size) %>%
  summarise(avg_salary = mean(salary_in_usd)) %>%
  ggplot(aes(x = company_size, y = avg_salary, fill = company_size)) +
  geom_bar(stat = "identity", position = "dodge", color = "black") +
  theme_minimal() +
  labs(
    title = "Average Salary by Company Size",
    x = "Company Size",
    y = "Average Salary in USD"
  ) +
  scale_fill_manual(values = c("blue", "green", "red"))  # Custom colors
```



here we started grouping the salaries by company size and avaraged for the better understanding pourpose by the compoany size for the better representation of the graphs.

Are experience levels consistently associated with higher salaries across different company sizes?

```
library(ggplot2)


sal$salary_in_k <- sal$salary_in_usd /1000

ggplot(sal, aes(x = experience_level, y = salary_in_k, color = experience_level)) +
  geom_point() +
  theme_minimal() +
  labs(
    title = "Scatter Plot of Salary in 'k' by Experience Level"
    x = "Experience Level",
    y = "Salary in 'k'"
  ) +
  scale_y_continuous(breaks = seq(0, max(sal$salary_in_k), by =100),
                     labels = scales::number_format(scale =1000e-3)) +
  scale_color_brewer(palette ="Set3")
```

from the above graph we can see that the salaries of each employe based on there experience level

```
sample_size <- length(salary_in_usd)
cat("Sample Size:", sample_size, "\n")
```

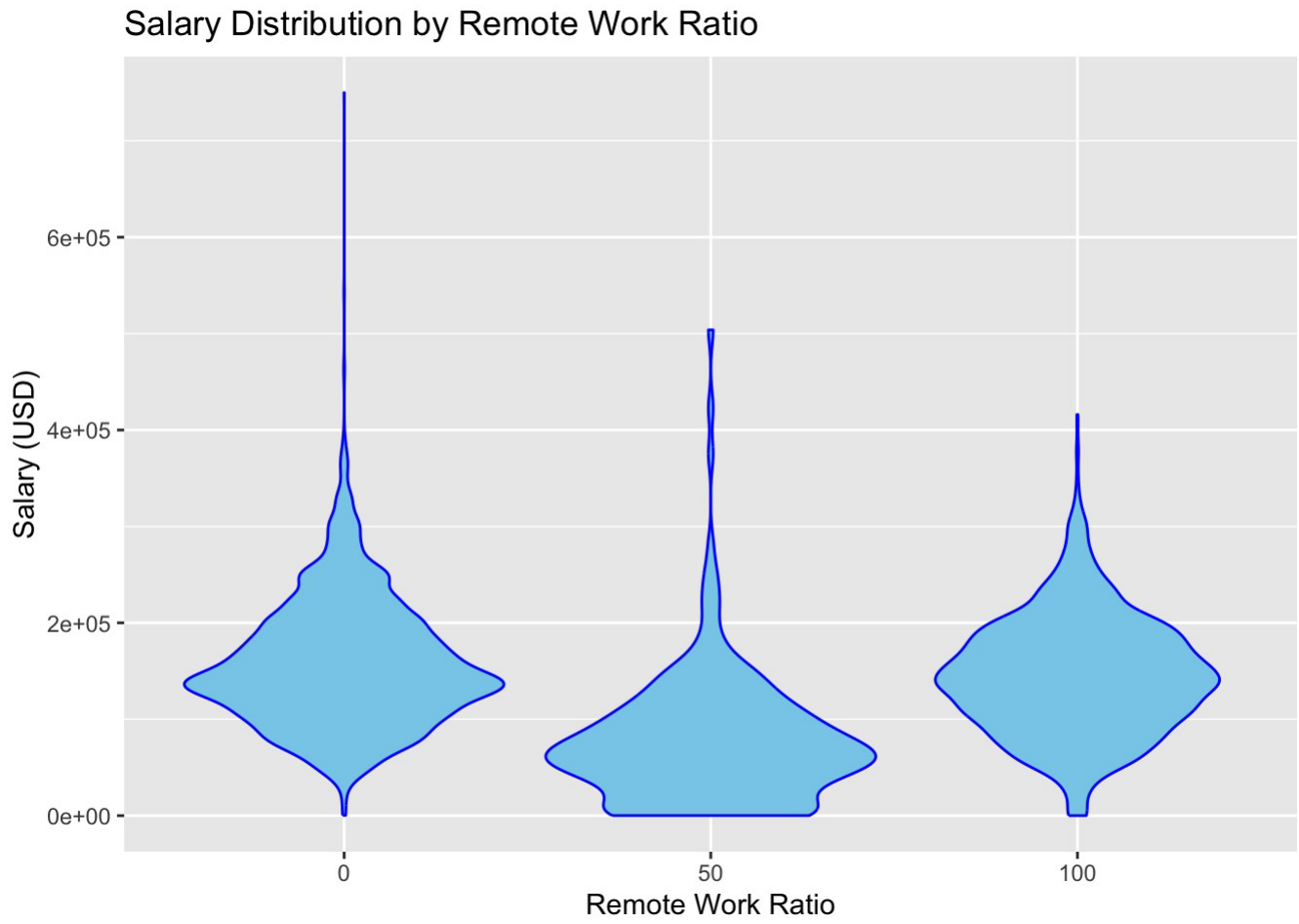
```
## Sample Size: 9074
```

```
library(dplyr)

summary_data <- sal %>%
  group_by(company_size) %>%
  summarise(avg_salary = mean(salary_in_usd))

str(summary_data)
```

```
## tibble [3 × 2] (S3: tbl_df/tbl/data.frame)
##  $ company_size: chr [1:3] "L" "M" "S"
##  $ avg_salary  : num [1:3] 133248 153132 86012
```

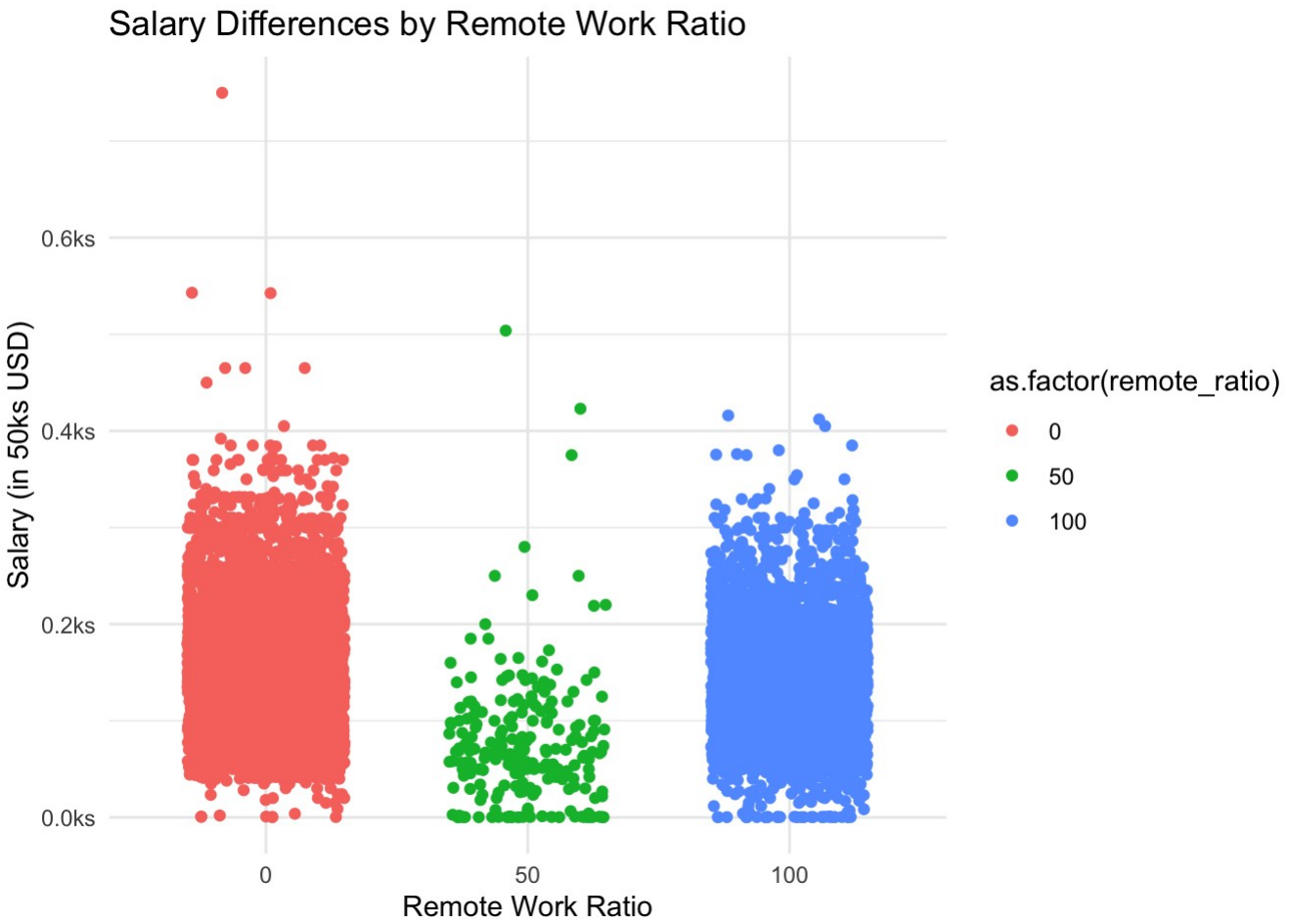
here we see that the avarage salary of large , medium and small companies salaries of employeses

```
ggplot(sal, aes(x = as.factor(remote_ratio), y = salary_in_usd)) +  geom_violin(fill = "skyblue", color = "blue") +  labs(title = "Salary Distribution by Remote Work Ratio", x = "Remote Work Ratio", y = "Salary (USD)")
```

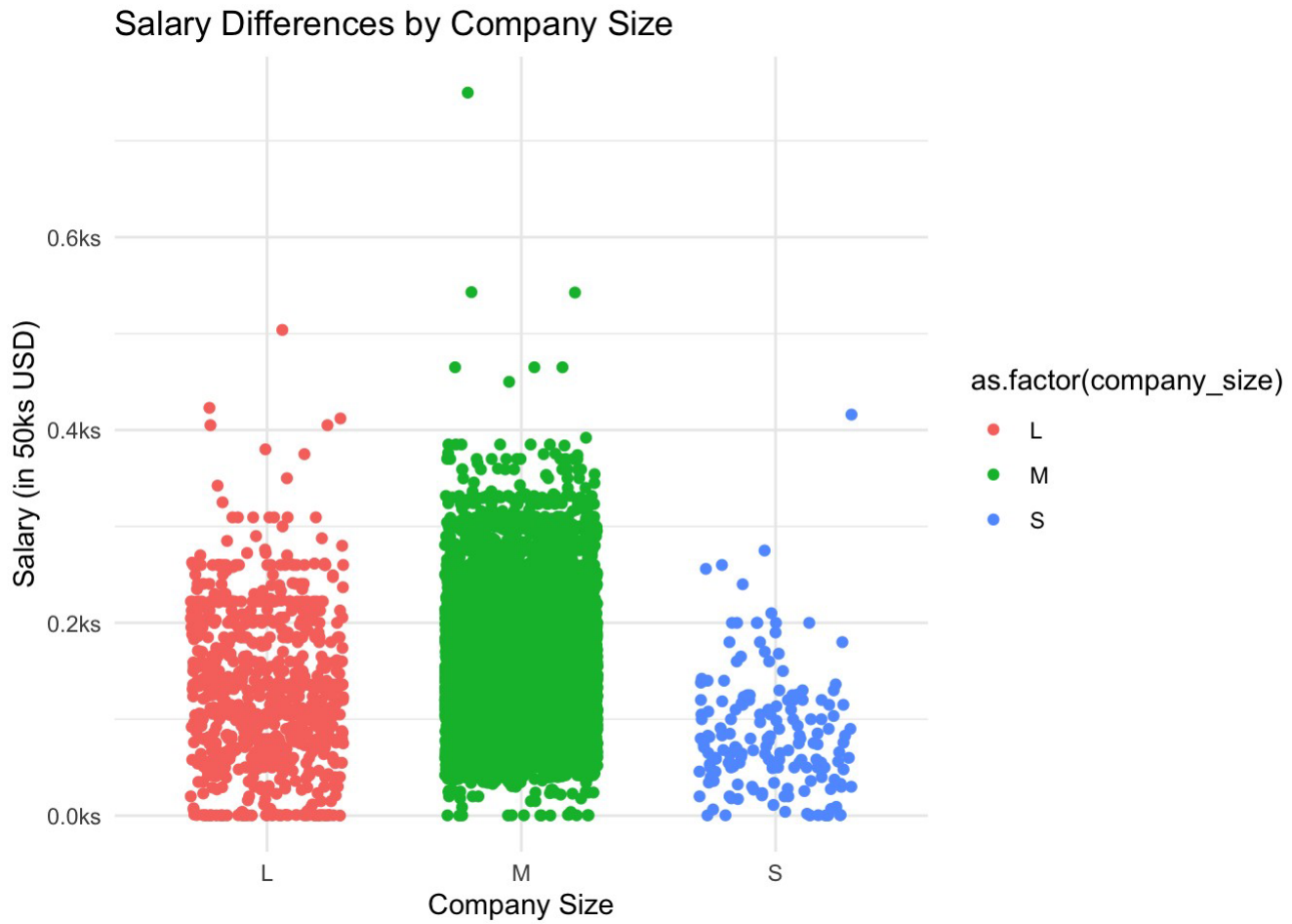



```
ggplot(sal, aes(x = as.factor(company_size), y = salary_in_usd)) +
  geom_violin(fill = "lightcoral", color = "red") +
  labs(title = "Salary Distribution by Company Size", x = "Company Size", y = "Salary (USD)")
```

```
ggplot(sal, aes(x = as.factor(remote_ratio), y = salary_in_usd/1000, color = as.factor(remote_ratio))) +
  geom_jitter(position = position_jitter(width = 0.3)) +
  labs(title = "Salary Differences by Remote Work Ratio", x = "Remote Work Ratio", y = "Salary (in 50ks USD)") +
  scale_y_continuous(labels = scales::comma_format(scale = 1e-3, suffix = "ks")) +  # Adjust y-axis labels
  theme_minimal()
```



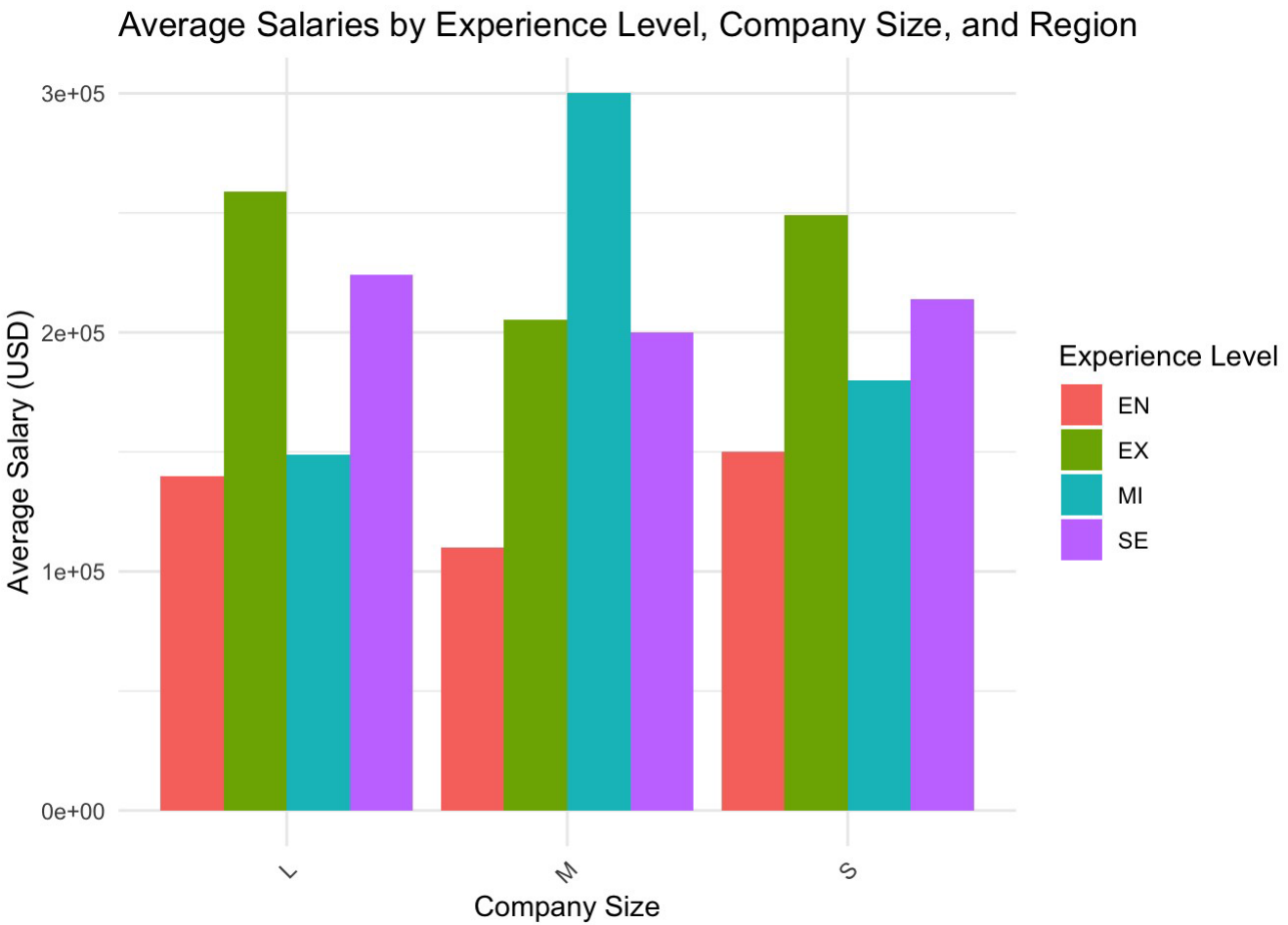

```
ggplot(sal, aes(x = as.factor(company_size), y = salary_in_usd/1000, color = as.factor(company_size))) +
  geom_jitter(position = position_jitter(width = 0.3)) +
  labs(title = "Salary Differences by Company Size", x = "Company Size", y = "Salary (in 50ks USD)") +
  scale_y_continuous(labels = scales::comma_format(scale = 1e-3, suffix = "ks")) +  # Adjust y-axis labels
  theme_minimal()
```



How does the salary distribuNon vary across different experience levels and the company size?

```
sal %>%
  group_by(experience_level, company_size, employee_residence) %>%
  summarise(avg_salary = mean(salary_in_usd)) %>%
  ggplot(aes(x = company_size, y = avg_salary, fill = experience_level)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Average Salaries by Experience Level, Company Size, and Region",
       x = "Company Size", y = "Average Salary (USD)", fill = "Experience Level") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

```
## `summarise()` has grouped output by 'experience_level', 'company_size'. You can
## override using the `.groups` argument.
```



from the above graph we can see the relation between the avarage salary , company size and the experience level

```
salary_ranges <- c(0, 50000, 100000, 150000, 200000, Inf)
salary_labels <- c("Low", "Medium-Low", "Medium", "Medium-High", "High")

sal$salary_category <- cut(sal$salary_in_usd, breaks = salary_ranges, labels = salary_labels, include.lowest = TRUE)

head(sal)
```

```
## # A tibble: 6 × 13
##   work_year experience_level employment_type job_title    salary salary_currency
##      <dbl> <chr>            <chr>           <chr>        <dbl> <chr>
## 1      2023 SE               FT              Data Engine… 185000 USD
## 2      2023 SE               FT              Data Engine… 150000 USD
## 3      2023 SE               FT              Data Analyst 166000 USD
## 4      2023 SE               FT              Data Analyst  73100 USD
## 5      2023 SE               FT              Business In… 198200 USD
## 6      2023 SE               FT              Business In… 110200 USD
## # ℹ 7 more variables: salary_in_usd <dbl>, employee_residence <chr>,
## #   remote_ratio <dbl>, company_location <chr>, company_size <chr>,
## #   salary_in_k <dbl>, salary_category <fct> here we have grouped the salaries into four groups like low ,
```

medium-low ,medium , medium-high

how can we predict the salary of Machine Learning Engineers in the United States using features such as work year, experience level, and company size?

```
sal <- sal[, c("experience_level", "company_size", "salary_in_usd")]
sal$experience_level <- as.numeric(as.factor(sal$experience_level))
sal$company_size <- as.numeric(as.factor(sal$company_size))

set.seed(123)
train_model <- sample(1:nrow(sal), 0.8 * nrow(sal))
train_data <- sal[train_model, ]
test_data <- sal[-train_model, ]

# Linear regression
lm_model <- lm(salary_in_usd ~ experience_level + company_size, data = train_data)
lm_predictions <- predict(lm_model, newdata = test_data)

# Random forest
library(randomForest)
```

```
## randomForest 4.7-1.1
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
rf_model <- randomForest(salary_in_usd ~ experience_level + company_size, data = train_data)
rf_predictions <- predict(rf_model, newdata = test_data)

# Evaluate models
lm_rmse <- sqrt(mean((lm_predictions - test_data$salary_in_usd)^2))
rf_rmse <- sqrt(mean((rf_predictions - test_data$salary_in_usd)^2))

# Evaluate the model on the test set
predictions <- predict(lm_model, newdata = test_data)

# R-squared
r_squared <- 1 - sum((test_data$salary_in_usd - predictions)^2) / sum((test_data$salary_in_usd - mean(test_data$salary_in_usd))^2)

# Mean Squared Error (MSE)
mse <- mean((test_data$salary_in_usd - predictions)^2)

cat("R-squared:", round(r_squared,4), "\n")
```

```
## R-squared: 0.0725
```

```
cat("Mean Squared Error:", round(mse,2), "\n")
```

```
## Mean Squared Error: 3751390066
```

```
cat("Linear Regression RMSE:", round(lm_rmse,4), "\n")
```

```
## Linear Regression RMSE: 61248.59
```

```
cat("Random Forest RMSE:", round(rf_rmse,4), "\n")
```

```
## Random Forest RMSE: 58588.03
```

```
#missing_values <- colSums(is.na(sal[c(experience_level, 'salary_category')]))

# Print the summary
#print(missing_values)

# Handle missing values (for example, removing rows with missing values)
sal <- na.omit(sal)

# Select relevant features
features <- sal %>%
  select(experience_level)

# Remove rows with missing values
#data <- na.omit(cbind(features, salary_category = sal$salary_category))
```

how can we predict the company level based on their experience level and the salary ?

```
library(tidyverse)
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✔ forcats   1.0.0     ✔ stringr   1.5.0
## ✔ lubridate 1.9.2     ✔ tibble    3.2.1
## ✔ purrr     1.0.2     ✔ tidyr     1.3.0
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
```

```
## ✖ randomForest::combine() masks dplyr::combine()
## ✖ dplyr::filter()     masks stats::filter()
## ✖ dplyr::lag()        masks stats::lag()
## ✖ randomForest::margin()  masks ggplot2::margin()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(caret)
```

```
## Loading required package: lattice
##
## Attaching package: 'caret'
##
## The following object is masked from 'package:purrr':
##
##     lift
```

```
library(e1071)

sal$experience_level <- as.factor(sal$experience_level)
sal$company_size <- as.factor(sal$company_size)

set.seed(123)  # for reproducible results
splitIndex <- createDataPartition(sal$company_size, p =80, list = FALSE)
training_set <- sal[splitIndex,]
test_set <- sal[-splitIndex,]

# Train an SVM model
svm_model <- svm(company_size ~ experience_level + salary_in_usd, data = training_set)

# Make predictions
predictions <- predict(svm_model, test_set)

# Evaluate the model
confusionMatrix(predictions, test_set$company_size)
```

```
## Confusion Matrix and Statistics
##
##            Reference
## Prediction     1     2     3
##          1     0     0     0
##          2   155  1625    33
##          3     0     0     0
##
## Overall Statistics
##
##                Accuracy : 0.8963
##                  95% CI : (0.8813, 0.91)
##     No Information Rate : 0.8963
##     P-Value [Acc > NIR] : 0.5194
##
##                   Kappa : 0
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                        Class: 1 Class: 2 Class: 3
## Sensitivity             0.00000   1.0000   0.0000
## Specificity             1.00000   0.0000   1.0000
## Pos Pred Value              NaN   0.8963      NaN
## Neg Pred Value          0.91451      NaN   0.9818
## Prevalence              0.08549   0.8963   0.0182
## Detection Rate          0.00000   0.8963   0.0000
## Detection Prevalence    0.00000   1.0000   0.0000
## Balanced Accuracy       0.50000   0.5000   0.5000
```

```r
library(randomForest)
library(Metrics)




# Convert factors to numeric
sal$experience_level <- as.factor(sal$experience_level)
sal$company_size <- as.factor(sal$company_size)

# Split the dataset

train_model <- createDataPartition(sal$company_size, p = 0.80, list = FALSE)
training_set <- sal[train_model,]
test_set <- sal[-train_model,]

# Train a Random Forest model
model <- randomForest(company_size ~ experience_level + salary, data = training_set)

# Make predictions
predictions <- predict(model, newdata = test_set)

# Evaluate the model
conf_matrix <- confusionMatrix(predictions, test_set$company_size)

# Calculate metrics
accuracy <- sum(predictions == test_set$company_size) / length(test_set$company_size)

# Output the results
list(confusion_matrix = conf_matrix, accuracy = accuracy)

numeric_predictions <- as.numeric(predictions)
numeric_test_labels <- as.numeric(test_set$company_size)

# Calculate MSE
mse <- mse(numeric_test_labels, numeric_predictions)

# Calculate RSE (Root Mean Squared Error)
rse <- sqrt(mse)

# Calculate R²
r_squared <- R2(numeric_test_labels, numeric_predictions)

# Output the results
list(mse = mse, rse = rse, r_squared = r_squared)
```

```
$confusion_matrix
Confusion Matrix and Statistics

          Reference
Prediction    L    M    S
         L    8    1    2
         M  147 1623   30
         S    0    1    1

Overall Statistics

               Accuracy : 0.9002
                 95% CI : (0.8854, 0.9136)
    No Information Rate : 0.8963
    P-Value [Acc > NIR] : 0.311

                  Kappa : 0.089
```

```
# Load necessary libraries
library(tidyverse)
library(caret)
library(nnet)

# Convert categorical variables to factors
data_frame$experience_level <- as.factor(data_frame$experience_level)
data_frame$company_size <- as.factor(data_frame$company_size)

# Split the data into training and testing sets

splitIndex <- createDataPartition(data_frame$company_size, p = .80, list = FALSE)
training_set <- data_frame[splitIndex,]
test_set <- data_frame[-splitIndex,]

# Train logistic regression model

multinom_model <- multinom(company_size ~ experience_level + salary_in_usd, data = training_set)

# Make predictions
predictions <- predict(multinom_model, test_set)

# Evaluate the model
confusionMatrix(predictions, test_set$company_size)
```

```
# weights:  18 (10 variable)
initial  value 7977.023828
iter  10 value 2843.345232
iter  20 value 2614.213959
iter  30 value 2613.359854
final  value 2613.359243
converged
Confusion Matrix and Statistics

          Reference
Prediction    0    1    2
         0    0    0    0
         1  155 1625   33
         2    0    0    0

Overall Statistics

               Accuracy : 0.8963
                 95% CI : (0.8813, 0.91)
    No Information Rate : 0.8963
    P-Value [Acc > NIR] : 0.5194
```

Can we predict the salary of AI, ML, and Data Science professionals based on their job title and experience level?

```r
library(dplyr)
library(fastDummies)
library(glmnet)


features <- sal[c("work_year", "experience_level", "employment_type", "job_title",
                  "salary", "salary_currency", "salary_in_usd", "employee_residence",
                  "remote_ratio", "company_location", "company_size")]


features_encoded <- fastDummies::dummy_cols(features, select_columns = c("experience_level", "job_title",  "company_size"))

# Combine encoded features with the target variable (salary)
data <- cbind(features_encoded, salary_in_usd = sal$salary_in_usd)

# splitting the dataset into train and test data , 80% is train data and 20% is test data
set.seed(123)
train_split <- sample(seq_len(nrow(data)), size = 0.8 * nrow(data))
train_data <- data[train_split, ]
test_data <- data[-train_split, ]

# parameter tuning using cross-validation for the enhancement
cv_model <- cv.glmnet(
  x = as.matrix(train_data[, -ncol(train_data)]),
  y = train_data$salary_in_usd,
  alpha = 1
)

# Find the optimal lambda for using in the lasso regression
best_lambda <- cv_model$lambda.min



# Build a final model using the optimal lambda
final_model <- glmnet(
  x = as.matrix(train_data[, -ncol(train_data)]),
  y = train_data$salary_in_usd,
  alpha = 1,
  lambda = best_lambda
)

# Make predictions on the test set
predictions <- predict(final_model, newx = as.matrix(test_data[, -ncol(test_data)]))

# Evaluate the final model
mse <- mean((predictions - test_data$salary_in_usd)^2)
rsquared <- 1 - mse / var(test_data$salary_in_usd)

cat("Mean Squared Error:", mse, "\n")
cat("R-squared:", rsquared, "\n")
```

```
> cat("Mean Squared Error:", mse, "\n")
Mean Squared Error: 3415509
> cat("R-squared:", rsquared, "\n")
R-squared: 0.9991502
>
```