Module 05_Project 01_DSE 5002 R and Python Programming

Anthony V. Razzano, DHA

2024-10-11

Executive Summary

The CEO of your company has decided to hire a full-time data scientist, with the potential to build a future team. She is unsure of the appropriate salary range due to wide variations in global pay, coupled with rising wages caused by the economic recession and a competitive job market. She requests an analysis of data science salaries to establish a competitive range, particularly comparing the U.S. and offshore markets.

The company is small but rapidly growing, and the role can be remote. The CEO expects a presentation with visuals to communicate salary recommendations, and the R code should be delivered as a flat file for submission.

Metadata for the analysis includes: - Work year, experience level, employment type, job title - Salary details (total and in USD), employee residence - Remote work percentage, company location, and company size (small, medium, large)

The deliverables include: A PowerPoint presentation for the CEO with visualizations and analysis. An R script for the analysis but without showing code in the presentation.

R code for the analysis

```r
# Load necessary libraries
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
```

```r
library(ggplot2)
library(scales)
library(caret)
```

```
## Loading required package: lattice
```

```r
library(randomForest)
```

```
## randomForest 4.7-1.2

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

##
## Attaching package: 'randomForest'

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

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

```r
library(broom)
library(cluster)
library(factoextra)
```

```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```r
library(tidyr)
library(corrplot)
```

```
## corrplot 0.94 loaded
```

```r
library(e1071)  # For SVM
library(pROC)   # For ROC curve
```

```
## Type 'citation("pROC")' for a citation.

##
## Attaching package: 'pROC'

## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

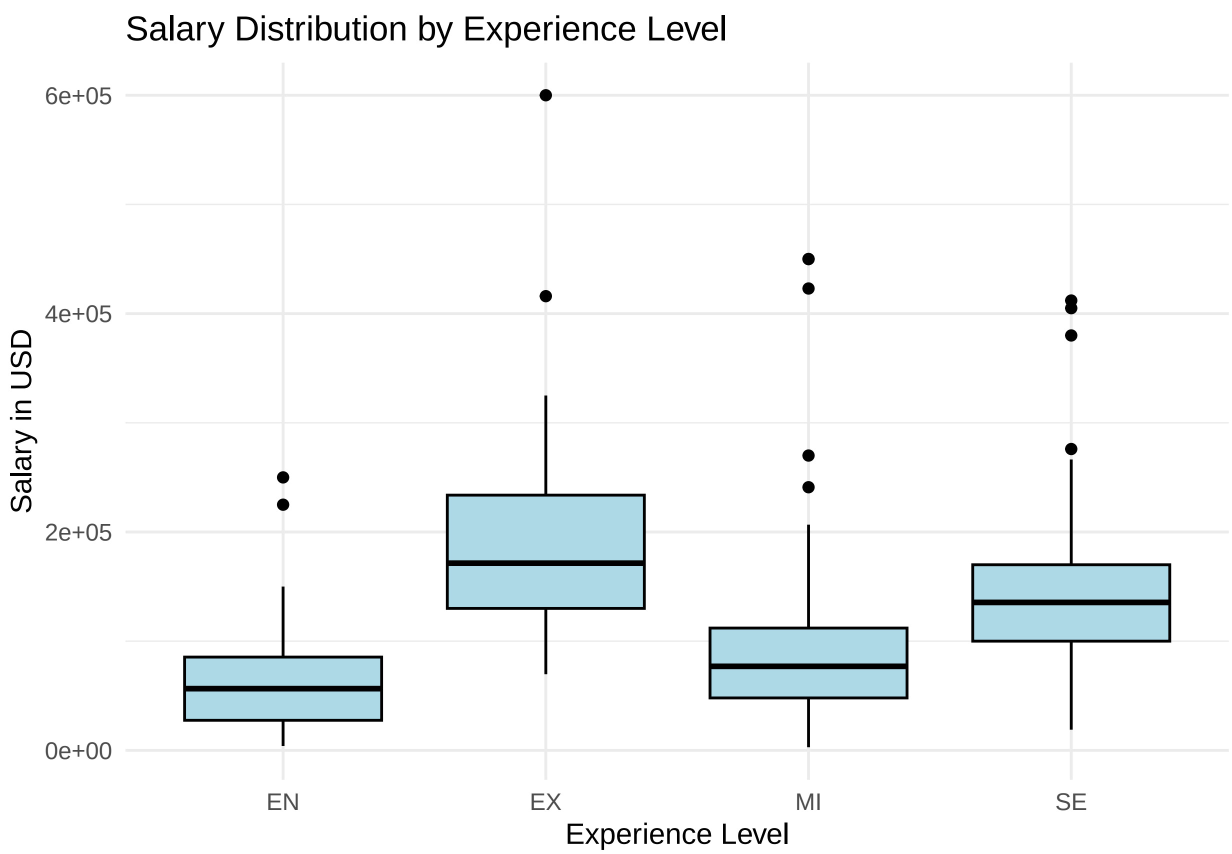
```r
# ========================
# 1. Meeting with Subject Matter Experts (Business Stakeholders)
# ========================
# Define business questions
# What salary ranges should we offer to remain competitive?
# How do U.S. salaries compare with offshore salaries?
# How does experience level influence salary?
# Should remote work impact salary decisions?
```

```r
# ========================
# 2. Data Collection & Ethics
# ========================
# Use internal CSV data file.
# Consider data privacy, anonymize personal data, and ensure ethical data handling.

# Read the CSV file
data <- read.csv("C:/Users/AVR15/Documents/r project data.csv")

# Convert relevant columns to factors
data <- data %>%
  mutate(
    experience_level = as.factor(experience_level),
    employment_type = as.factor(employment_type),
    job_title = as.factor(job_title),
    employee_residence = as.factor(employee_residence),
    company_location = as.factor(company_location),
    company_size = as.factor(company_size),
    remote_ratio = as.factor(remote_ratio)
  )

# ========================
# 3. Data Quality Check and Cleaning
# ========================

# Check for missing values and remove them
data_clean <- na.omit(data)

# Summary statistics
summary(data_clean)
```

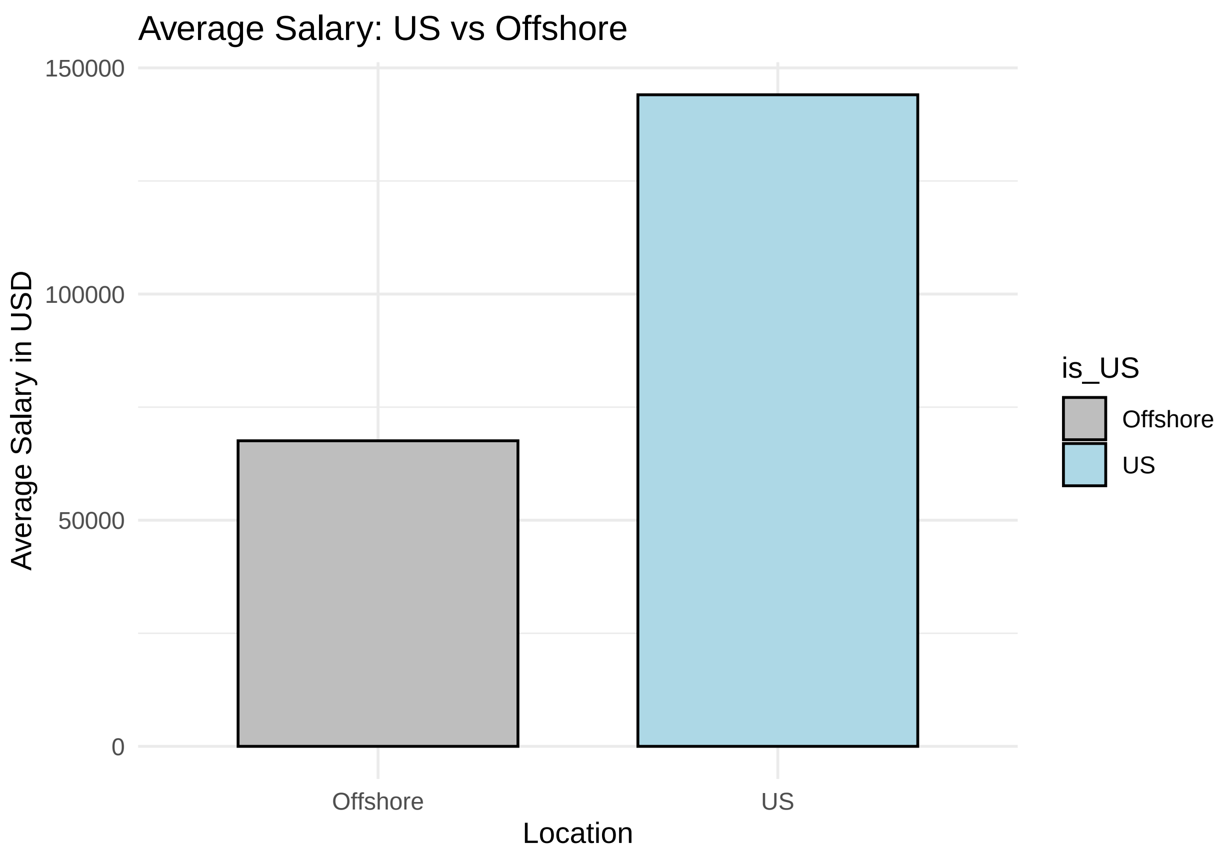
```
##      emp_id        work_year    experience_level employment_type
##  Min.   :  0.0   Min.   :2020   EN: 88           CT:  5
##  1st Qu.:151.5   1st Qu.:2021   EX: 26           FL:  4
##  Median :303.0   Median :2022   MI:213           FT:588
##  Mean   :303.0   Mean   :2021   SE:280           PT: 10
##  3rd Qu.:454.5   3rd Qu.:2022
##  Max.   :606.0   Max.   :2022
##
##                       job_title       salary         salary_currency
##  Data Scientist           :143   Min.   :    4000   Length:607
##  Data Engineer            :132   1st Qu.:   70000   Class :character
##  Data Analyst             : 97   Median :  115000   Mode  :character
##  Machine Learning Engineer: 41   Mean   :  324000
##  Research Scientist       : 16   3rd Qu.:  165000
##  Data Science Manager     : 12   Max.   :30400000
##  (Other)                  :166
##  salary_in_usd    employee_residence remote_ratio company_location company_size
##  Min.   :  2859   US     :332        0  :127      US     :355      L:198
##  1st Qu.: 62726   GB     : 44        50 : 99      GB     : 47      M:326
##  Median :101570   IN     : 30        100:381      CA     : 30      S: 83
##  Mean   :112298   CA     : 29                     DE     : 28
##  3rd Qu.:150000   DE     : 25                     IN     : 24
```

```
##  Max.   :600000   FR      : 18                       FR      : 15
##                    (Other):129                       (Other):108
```

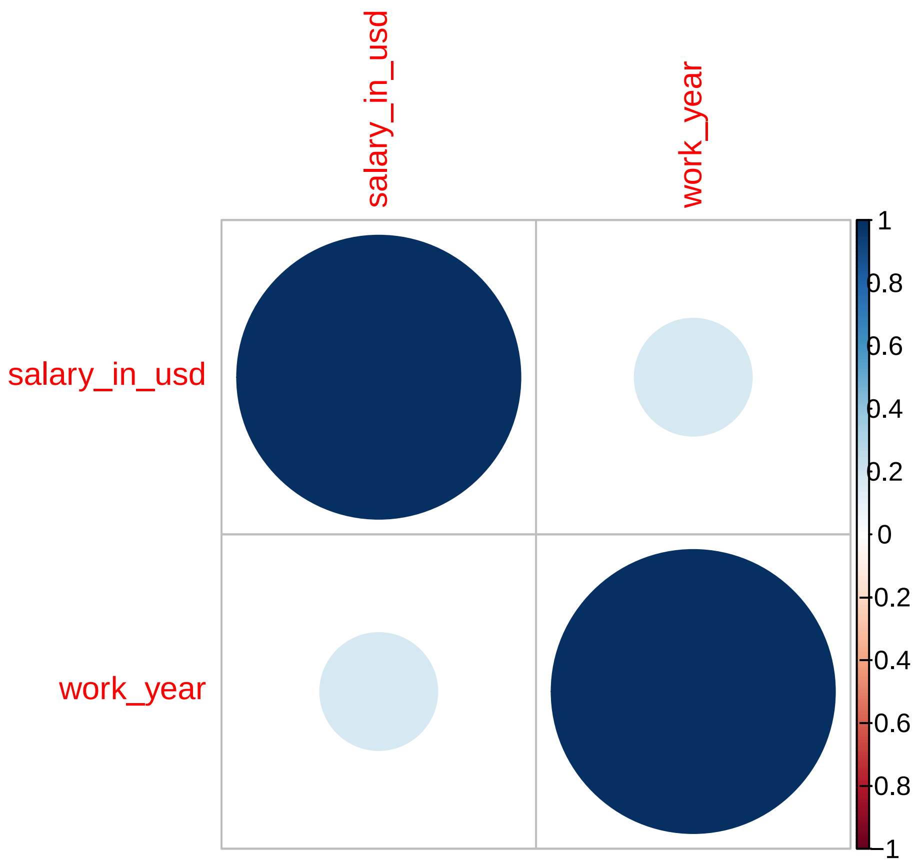
```r
# ========================
# 4. Exploratory Data Analysis (EDA)
# ========================

# Salary distribution by experience level (Boxplot)
ggplot(data_clean, aes(x = experience_level, y = salary_in_usd)) +
  geom_boxplot(fill = "lightblue", color = "black") +
  labs(title = "Salary Distribution by Experience Level", x = "Experience Level", y = "Salary in USD") +
  theme_minimal()
```



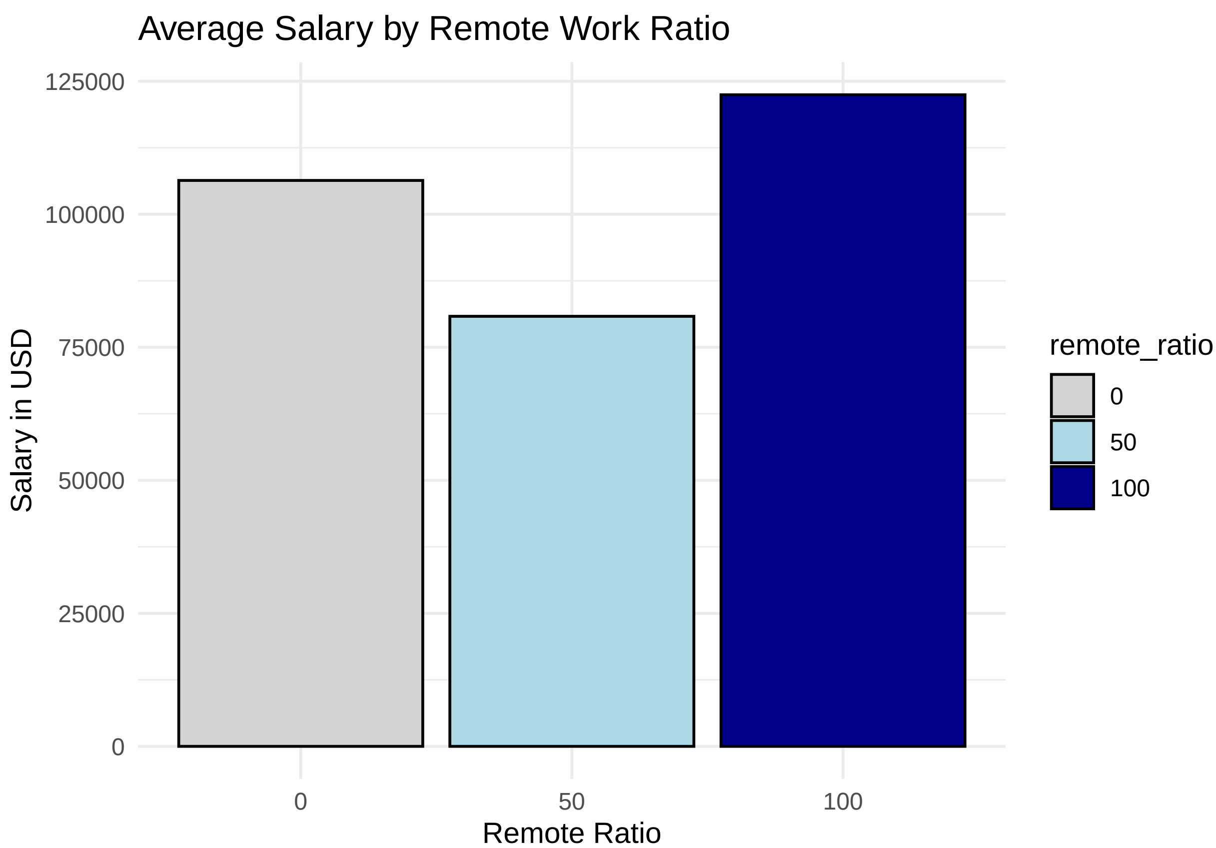

```r
# salary distribution across different experience levels visualized with boxplot. Visual reveals that s


# Average salary by company location (US vs Offshore)
data_clean <- data_clean %>%
  mutate(is_US = ifelse(company_location == "US", "US", "Offshore"))

ggplot(data_clean, aes(x = factor(is_US), y = salary_in_usd, fill = is_US)) +
  geom_bar(stat = "summary", fun = "mean", color = "black", width = 0.7) +
  labs(title = "Average Salary: US vs Offshore", x = "Location", y = "Average Salary in USD") +
  scale_fill_manual(values = c("Offshore" = "gray", "US" = "lightblue")) +
  theme_minimal()
```



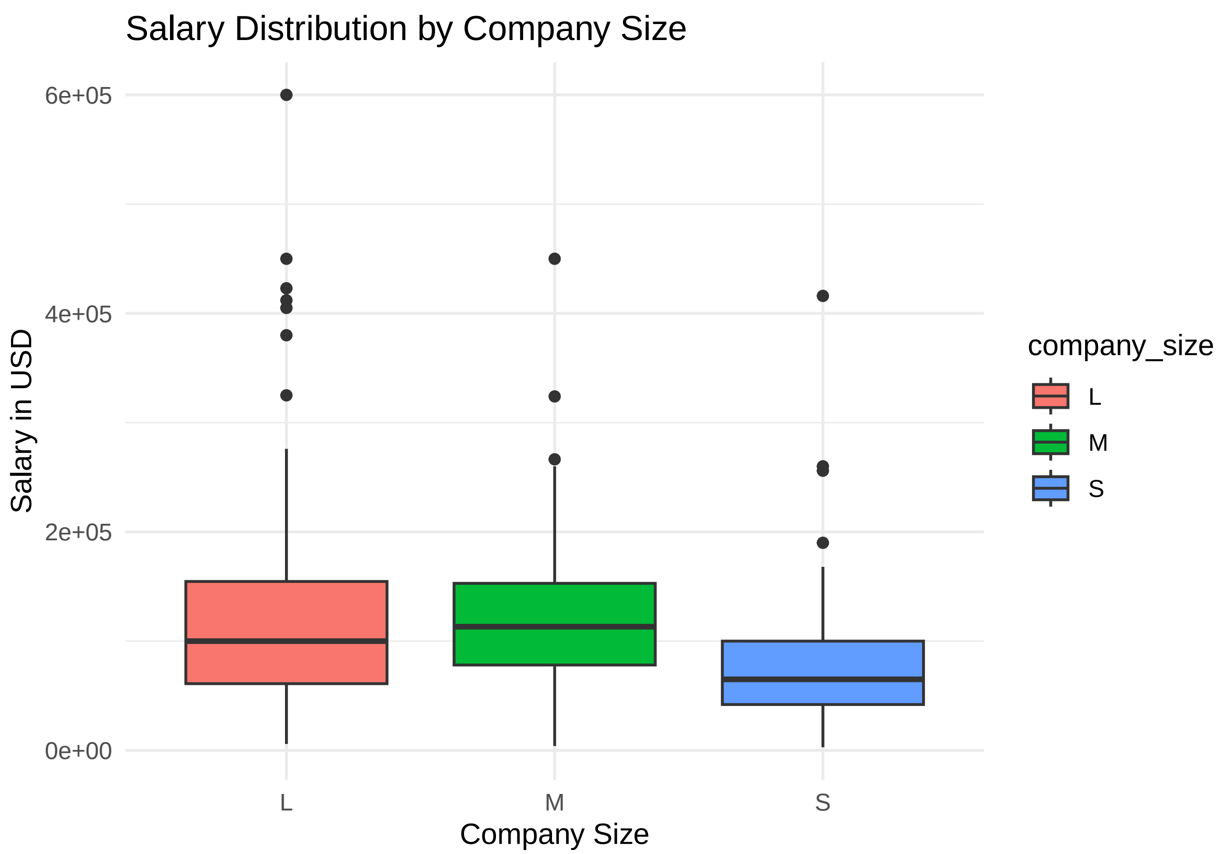

```r
# Comparison of average salaries between U.S.-based and offshore employees illustrated through a bar pl

# Correlation plot to explore relationships between numerical variables
numeric_data <- data_clean %>%
  select(salary_in_usd, work_year)

corr_matrix <- cor(numeric_data)
corrplot(corr_matrix, method = "circle")
```



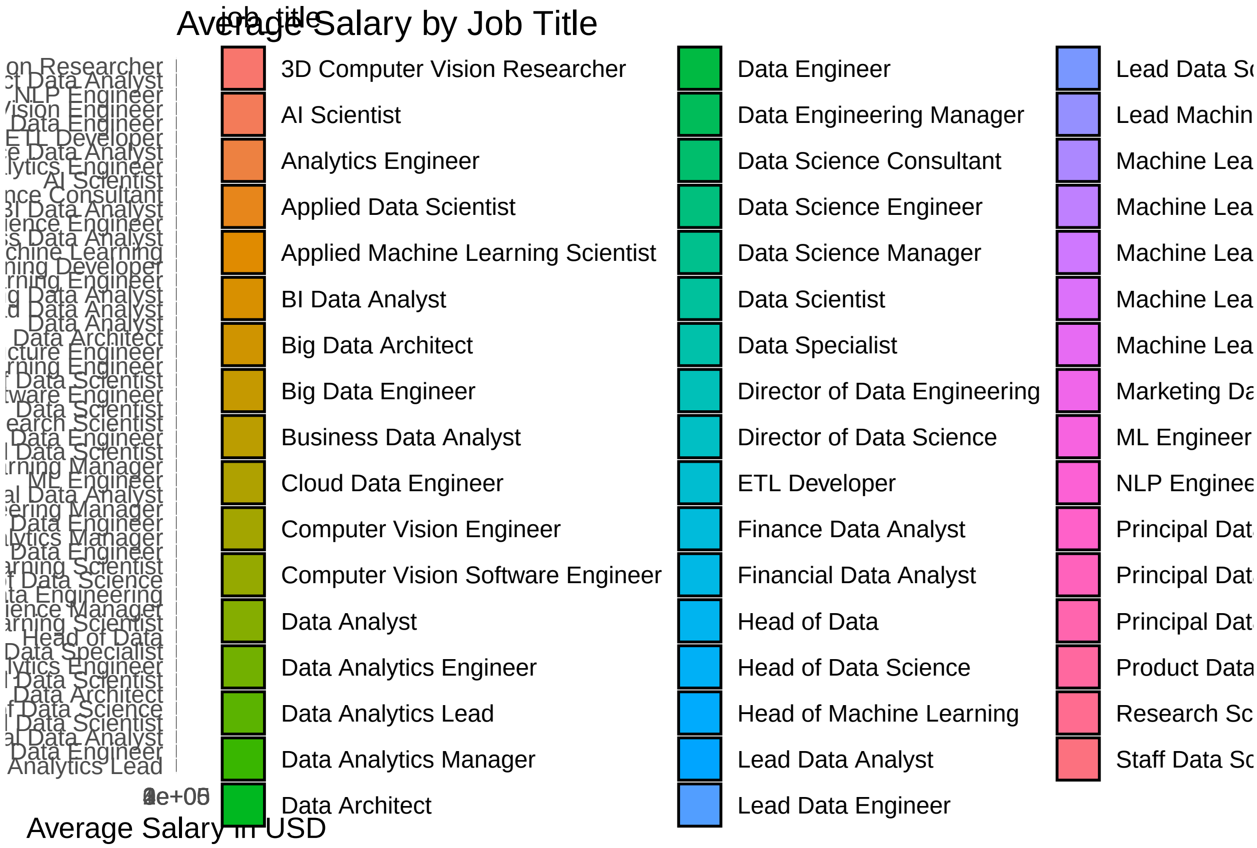

```
# Correlation plot used to examine the relationship between salary and work year. The analysis shows a 

# Remote work ratio vs Salary (Barplot)
ggplot(data_clean, aes(x = remote_ratio, y = salary_in_usd, fill = remote_ratio)) +
  geom_bar(stat = "summary", fun = "mean", color = "black") +
  labs(title = "Average Salary by Remote Work Ratio", x = "Remote Ratio", y = "Salary in USD") +
  scale_fill_manual(values = c("0" = "lightgray", "50" = "lightblue", "100" = "darkblue")) +
  theme_minimal()
```




```r
# Barplot was used to analyze the relationship between the remote work ratio and salary. The results re

# ========================
# Additional Exploratory Analyses
# ========================

# Salary distribution by company size (Boxplot)
ggplot(data_clean, aes(x = company_size, y = salary_in_usd, fill = company_size)) +
  geom_boxplot() +
  labs(title = "Salary Distribution by Company Size", x = "Company Size", y = "Salary in USD") +
  theme_minimal()
```




```
# The distribution of salaries by company size performed through a boxplot. The result revealed that la

# Average salary by job title (Barplot)
ggplot(data_clean, aes(x = reorder(job_title, -salary_in_usd), y = salary_in_usd, fill = job_title)) +
  geom_bar(stat = "summary", fun = "mean", color = "black") +
  labs(title = "Average Salary by Job Title", x = "Job Title", y = "Average Salary in USD") +
  theme_minimal() +
  coord_flip()  # Flip the coordinates for better readability
```




```
# Variation in salaries by job title was performed through a barplot with flipped coordinates for bette

# ========================
# 5. Combined Dataset for Consistent Factor Levels
# ========================

set.seed(123)  # Set a seed for reproducibility
trainIndex <- createDataPartition(data_clean$salary_in_usd, p = 0.8, list = FALSE)
train_data <- data_clean[trainIndex, ]
test_data <- data_clean[-trainIndex, ]

# ========================
# 6. Linear Regression Model
# ========================

# Create dummy variables for training data
train_dummies <- model.matrix(salary_in_usd ~ experience_level + employment_type +
                                remote_ratio + company_location + company_size,
                                data = train_data)[, -1]  # Exclude intercept

# Fit a linear regression model
lm_model <- lm(train_data$salary_in_usd ~ ., data = as.data.frame(train_dummies))

# Summary of the linear regression model
summary(lm_model)
```

```
##
## Call:
## lm(formula = train_data$salary_in_usd ~ ., data = as.data.frame(train_dummies))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -121197  -28012   -1343   19046  358209
##
## Coefficients: (9 not defined because of singularities)
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)         147432.20   43086.43   3.422 0.000681 ***
## experience_levelEX  121621.18   14283.01   8.515 2.72e-16 ***
## experience_levelMI   19926.07    8810.67   2.262 0.024215 *
## experience_levelSE   45797.95    8818.85   5.193 3.18e-07 ***
## employment_typeFL  -139122.66   41396.10  -3.361 0.000846 ***
## employment_typeFT   -67911.08   27400.28  -2.478 0.013572 *
## employment_typePT   -86600.09   35784.13  -2.420 0.015925 *
## remote_ratio50      -10139.62    9435.78  -1.075 0.283151
## remote_ratio100       3855.91    6632.37   0.581 0.561286
## company_locationAS  -36425.45   62764.52  -0.580 0.561978
## company_locationAT  -15757.31   41623.67  -0.379 0.705194
## company_locationAU         NA         NA      NA       NA
## company_locationBE   -3771.71   49973.01  -0.075 0.939871
## company_locationBR  -72144.55   49452.02  -1.459 0.145318
## company_locationCA   -7655.95   33381.53  -0.229 0.818707
## company_locationCH   22898.81   62739.91   0.365 0.715303
## company_locationCL         NA         NA      NA       NA
## company_locationCN  -36345.62   62891.59  -0.578 0.563624
## company_locationCO         NA         NA      NA       NA
## company_locationCZ  -19308.57   63064.26  -0.306 0.759619
## company_locationDE   -6994.53   33554.56  -0.208 0.834973
## company_locationDK   -9430.94   45115.21  -0.209 0.834514
## company_locationDZ   69078.08   67629.46   1.021 0.307622
## company_locationEE  -41430.52   62446.29  -0.663 0.507388
## company_locationES  -48824.97   35277.56  -1.384 0.167059
## company_locationFR  -17595.05   36994.16  -0.476 0.634585
## company_locationGB   -5857.16   33039.36  -0.177 0.859372
## company_locationGR  -34697.99   36625.36  -0.947 0.343972
## company_locationHN         NA         NA      NA       NA
## company_locationHR  -54658.40   62322.84  -0.877 0.380959
## company_locationHU         NA         NA      NA       NA
## company_locationIE  -28832.40   62322.84  -0.463 0.643861
## company_locationIL   35526.47   62841.42   0.565 0.572137
## company_locationIN  -58644.60   33958.57  -1.727 0.084886 .
## company_locationIQ   59517.08   63302.97   0.940 0.347639
## company_locationIR  -79532.53   62841.42  -1.266 0.206330
## company_locationIT  -19184.24   51187.33  -0.375 0.708003
## company_locationJP   42702.94   38511.06   1.109 0.268107
## company_locationKE  -45206.45   62764.52  -0.720 0.471754
## company_locationLU  -25188.16   44683.97  -0.564 0.573251
## company_locationMD  -52548.60   62529.03  -0.840 0.401152
## company_locationMT  -60938.57   63064.26  -0.966 0.334433
## company_locationMX  -66591.36   44025.71  -1.513 0.131117
## company_locationMY  -43377.04   63231.59  -0.686 0.493076
```

```
## company_locationNG  -48890.78   49508.61  -0.988 0.323935
## company_locationNL  -33388.90   45419.90  -0.735 0.462664
## company_locationNZ         NA         NA      NA       NA
## company_locationPK  -60529.06   50080.18  -1.209 0.227455
## company_locationPL  -59522.16   50067.13  -1.189 0.235147
## company_locationPT  -48956.36   44813.11  -1.092 0.275236
## company_locationRO         NA         NA      NA       NA
## company_locationRU  -28687.21   51342.59  -0.559 0.576625
## company_locationSG     -13.57   63064.26   0.000 0.999828
## company_locationSI         NA         NA      NA       NA
## company_locationTR  -81291.04   49741.53  -1.634 0.102924
## company_locationUA         NA         NA      NA       NA
## company_locationUS   36792.87   31790.42   1.157 0.247760
## company_locationVN  -55750.55   63457.76  -0.879 0.380131
## company_sizeM       -19770.57    6185.87  -3.196 0.001494 **
## company_sizeS       -28898.59    9213.08  -3.137 0.001824 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 53850 on 436 degrees of freedom
## Multiple R-squared:  0.4893, Adjusted R-squared:  0.4307
## F-statistic: 8.354 on 50 and 436 DF,  p-value: < 2.2e-16
```

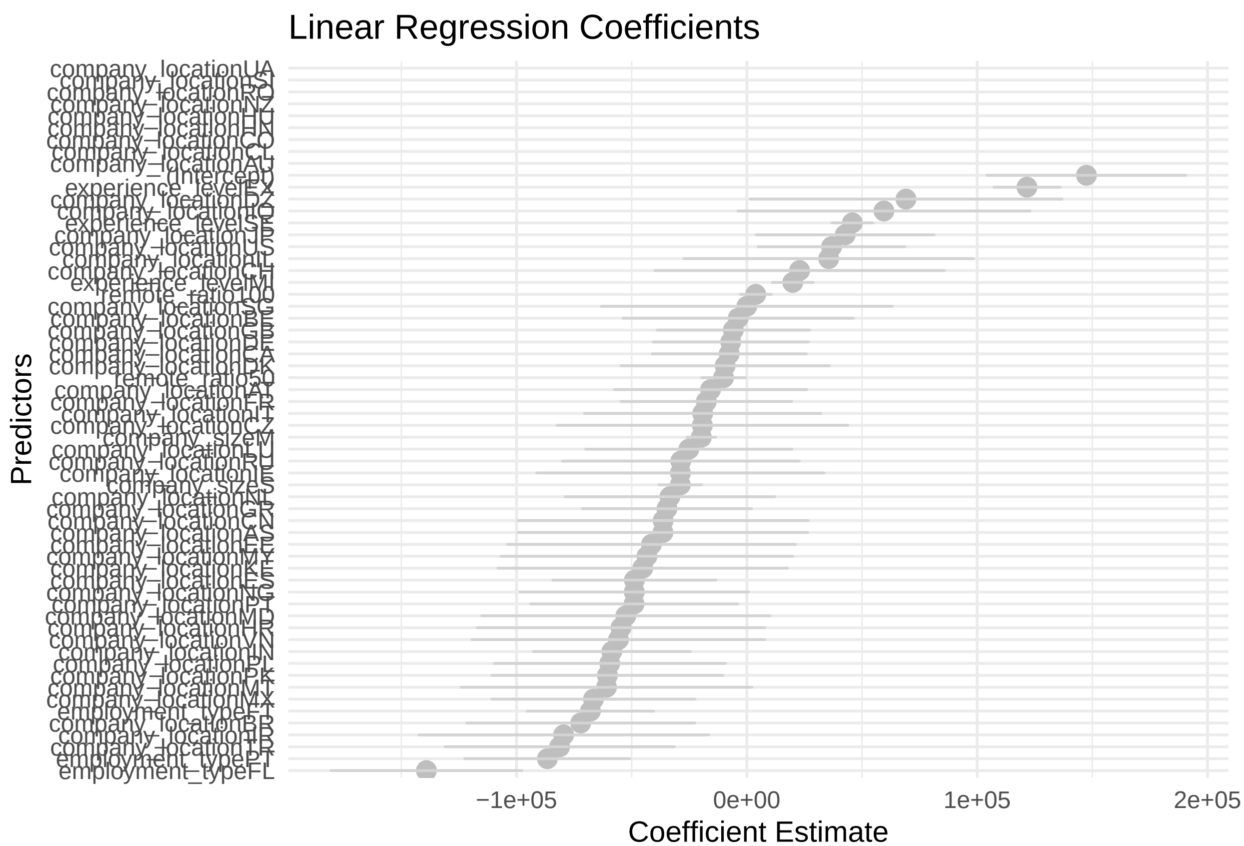
```r
# Tidy up the output for a clearer view of coefficients
tidy_lm <- tidy(lm_model)
print(tidy_lm)
```

```
## # A tibble: 60 x 5
##    term               estimate std.error statistic  p.value
##    <chr>                 <dbl>     <dbl>     <dbl>    <dbl>
##  1 (Intercept)         147432.    43086.     3.42  6.81e- 4
##  2 experience_levelEX  121621.    14283.     8.52  2.72e-16
##  3 experience_levelMI   19926.     8811.     2.26  2.42e- 2
##  4 experience_levelSE   45798.     8819.     5.19  3.18e- 7
##  5 employment_typeFL  -139123.    41396.    -3.36  8.46e- 4
##  6 employment_typeFT   -67911.    27400.    -2.48  1.36e- 2
##  7 employment_typePT   -86600.    35784.    -2.42  1.59e- 2
##  8 remote_ratio50      -10140.     9436.    -1.07  2.83e- 1
##  9 remote_ratio100       3856.     6632.     0.581 5.61e- 1
## 10 company_locationAS  -36425.    62765.    -0.580 5.62e- 1
## # i 50 more rows
```

```r
# The linear regression model illustrated experience level and employment type as significant predictor

# visualizing

ggplot(tidy_lm, aes(x = reorder(term, estimate), y = estimate)) +
  geom_point(color = "gray", size = 3) +  # Neutral gray color for points
  geom_errorbar(aes(ymin = estimate - std.error, ymax = estimate + std.error),
                width = 0.2, color = "lightgray") +  # Light gray for error bars
  labs(title = "Linear Regression Coefficients",
       x = "Predictors",
       y = "Coefficient Estimate") +
```

```r
  coord_flip() +  # Flip coordinates for better readability
  theme_minimal()  # Use a minimal theme for a clean look
```

```
## Warning: Removed 9 rows containing missing values or values outside the scale range
## (`geom_point()`).
```

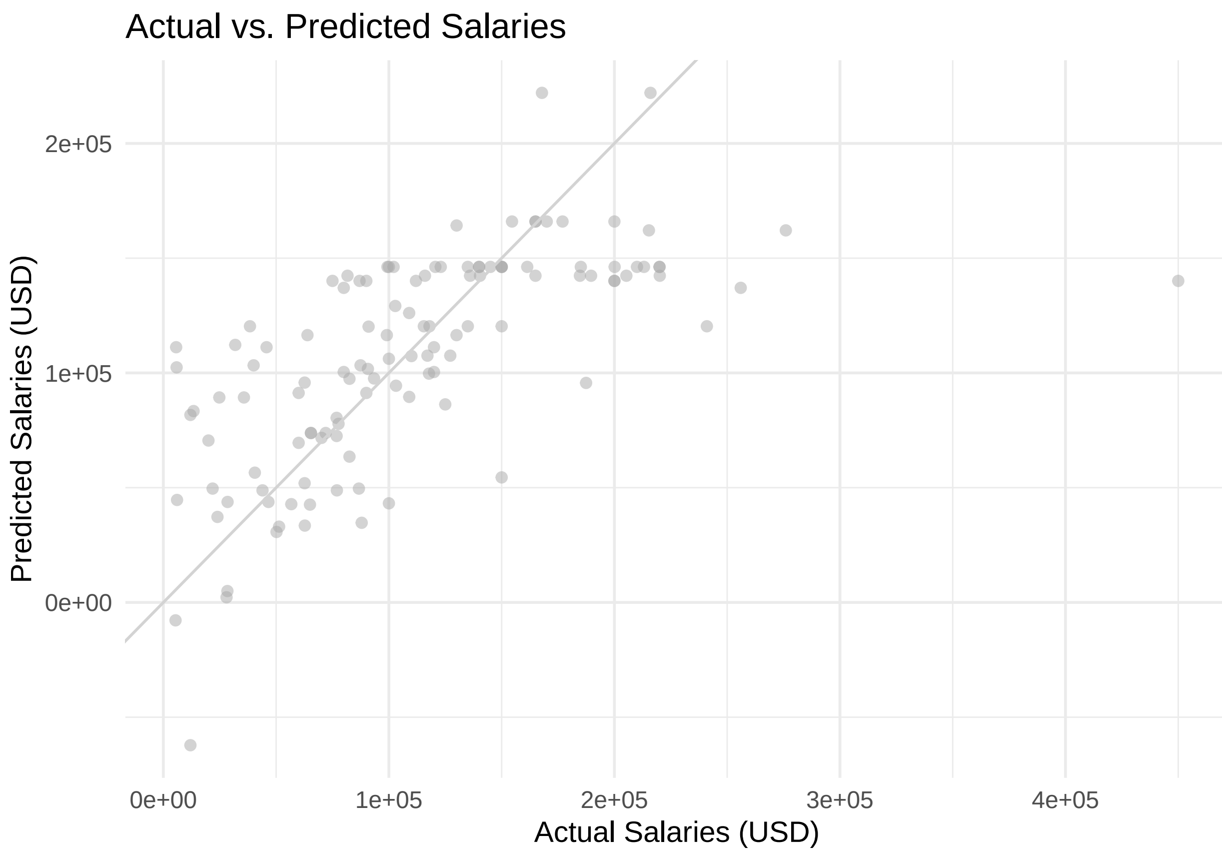



```r
# ========================
# Actual vs. Predicted Salaries
# ========================
# Predictions on test data
# Create dummy variables for the test dataset to ensure consistency with the model training data
test_dummies <- model.matrix(~ experience_level + employment_type +
                               remote_ratio + company_location + company_size,
                               data = test_data)[, -1]  # Exclude intercept

# Use the linear model to predict salaries based on the test data
lm_predictions <- predict(lm_model, newdata = as.data.frame(test_dummies))
```

```
## Warning in predict.lm(lm_model, newdata = as.data.frame(test_dummies)):
## prediction from rank-deficient fit; attr(*, "non-estim") has doubtful cases
```

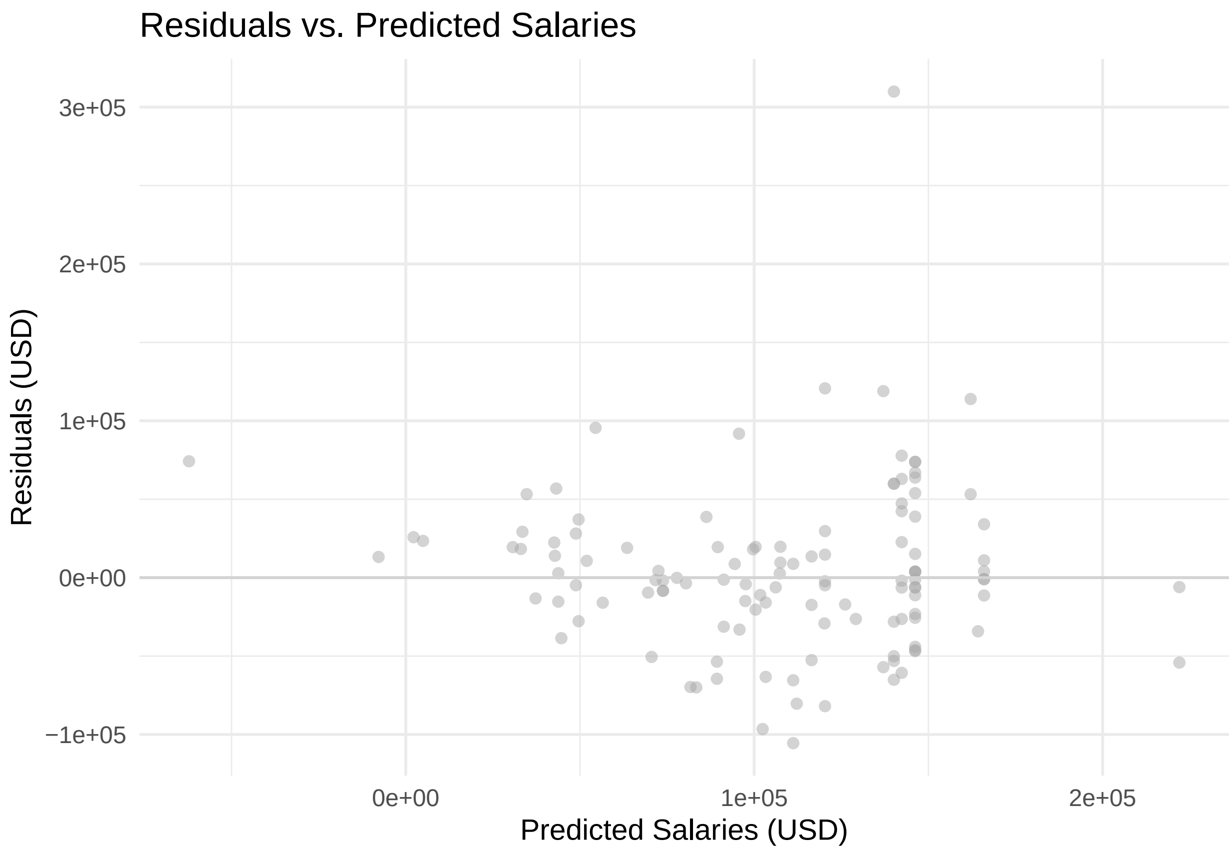
```r
# Create a data frame for plotting actual vs. predicted results
results_df <- data.frame(Actual = test_data$salary_in_usd, Predicted = lm_predictions)
```

```r
# Plot actual vs. predicted salaries
# This scatter plot helps visualize how well the model's predictions align with actual salaries.
ggplot(results_df, aes(x = Actual, y = Predicted)) +
  geom_point(color = "darkgray", alpha = 0.5) +  # Use dark gray for points with some transparency
  geom_abline(slope = 1, intercept = 0, color = "lightgray") +  # Light gray line for 45-degree referen
  labs(title = "Actual vs. Predicted Salaries",
       x = "Actual Salaries (USD)",
       y = "Predicted Salaries (USD)") +
  theme_minimal()  # Use a minimal theme for a clean look
```




```r
# ========================
# Residuals Plot
# ========================
# Calculate residuals (the difference between actual and predicted salaries)
results_df$Residuals <- results_df$Actual - results_df$Predicted

# Plot residuals vs. predicted salaries
# This plot helps assess the model's performance by visualizing the residuals.
# Ideally, the residuals should be randomly distributed around zero.
ggplot(results_df, aes(x = Predicted, y = Residuals)) +
  geom_point(color = "darkgray", alpha = 0.5) +  # Use dark gray for points with some transparency
  geom_hline(yintercept = 0, color = "lightgray") +  # Light gray horizontal line at y=0
  labs(title = "Residuals vs. Predicted Salaries",
       x = "Predicted Salaries (USD)",
       y = "Residuals (USD)") +
```

```
theme_minimal()  # Use a minimal theme for a clean look
```




```
# ========================
# 7. Random Forest Model
# ========================

# Random Forest model training
rf_model <- randomForest(salary_in_usd ~ experience_level + employment_type +
                           remote_ratio + company_location + company_size,
                           data = train_data, importance = TRUE)

# Model performance on training data
rf_predictions_train <- predict(rf_model, train_data)
rf_rmse_train <- RMSE(rf_predictions_train, train_data$salary_in_usd)
cat(paste0("Random Forest RMSE (Train): ", round(rf_rmse_train, 2), "\n"))
```

```
## Random Forest RMSE (Train): 52862.13
```

```
# Predictions on test data
rf_predictions <- predict(rf_model, test_data)
rf_rmse_test <- RMSE(rf_predictions, test_data$salary_in_usd)
rf_r2_test <- R2(rf_predictions, test_data$salary_in_usd)

# Print model performance metrics
cat(paste0("Random Forest RMSE (Test): ", round(rf_rmse_test, 2), "\n"))
```

```
## Random Forest RMSE (Test): 56236.54
```

```r
cat(paste0("Random Forest R-squared (Test): ", round(rf_r2_test, 2), "\n"))
```

```
## Random Forest R-squared (Test): 0.39
```

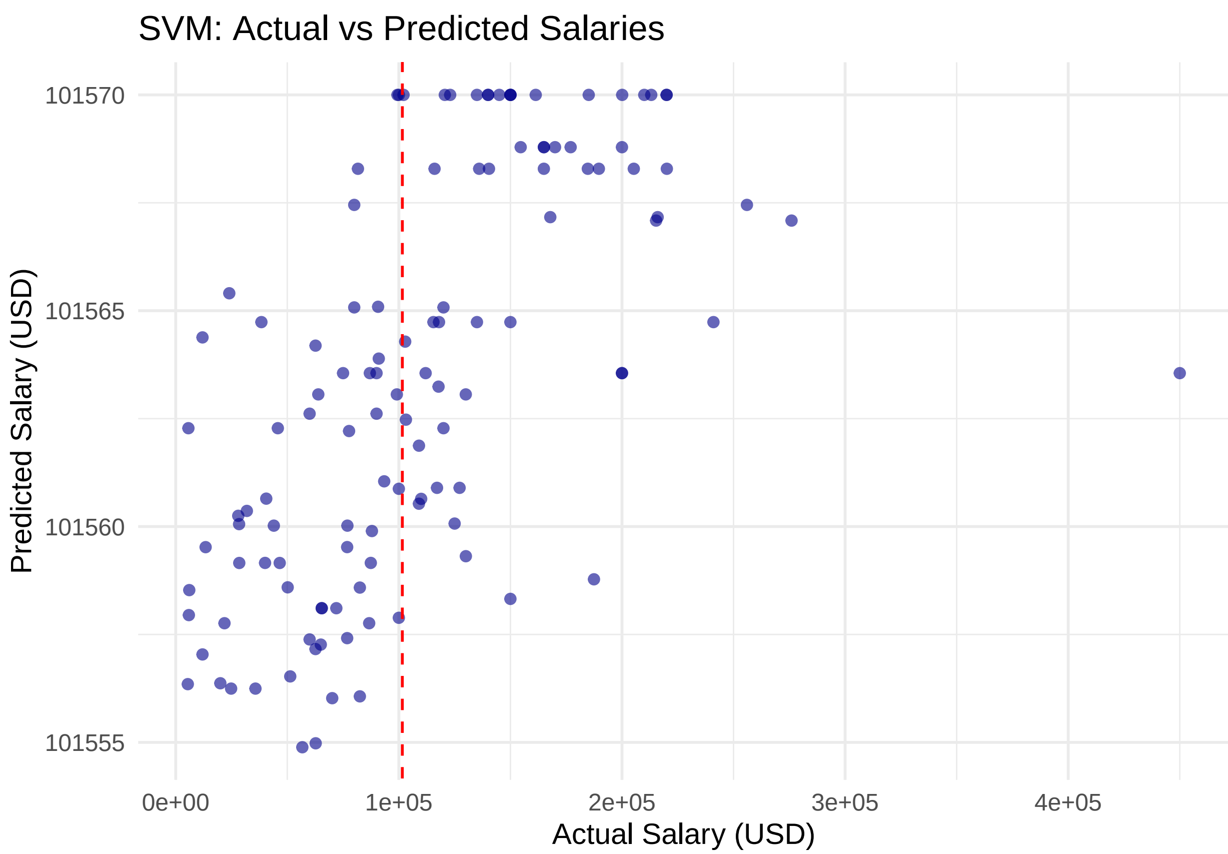
```r
# Variable importance plot
varImpPlot(rf_model)
```


```r
# The Random Forest model performed sufficiently; it achieved an RMSE of 56,236.54 on the test data, id

# ========================
# 8. Support Vector Machine Model
# ========================

# Scale the training data for SVM
train_data_scaled <- scale(train_data %>% select(salary_in_usd, work_year))

# SVM model training
svm_model <- svm(salary_in_usd ~ experience_level + employment_type +
                   remote_ratio + company_location + company_size,
                   data = train_data)

# Predictions on test data
```

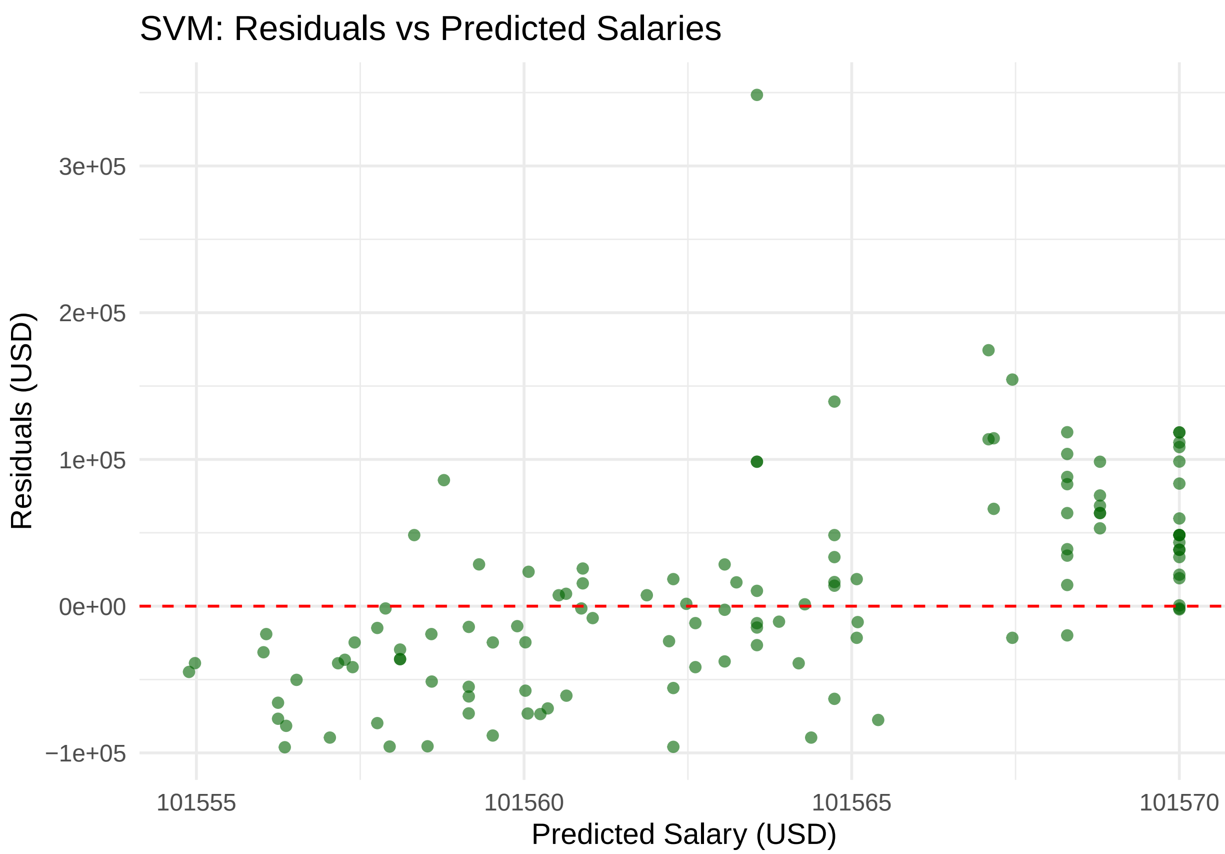
```r
svm_predictions <- predict(svm_model, test_data)
svm_rmse <- RMSE(svm_predictions, test_data$salary_in_usd)

# Print SVM model performance
cat(paste0("SVM RMSE: ", round(svm_rmse, 2), "\n"))
```

```
## SVM RMSE: 69950.19
```

```r
# Visualize for powerpoint

# Scale the training data for SVM
train_data_scaled <- scale(train_data %>% select(salary_in_usd, work_year))

# SVM model training
svm_model <- svm(salary_in_usd ~ experience_level + employment_type +
                   remote_ratio + company_location + company_size,
                 data = train_data)

# Predictions on test data
svm_predictions <- predict(svm_model, test_data)

# Calculate residuals
svm_residuals <- test_data$salary_in_usd - svm_predictions

# Plot 1: Actual vs Predicted Salaries
ggplot(data.frame(Actual = test_data$salary_in_usd, Predicted = svm_predictions),
       aes(x = Actual, y = Predicted)) +
  geom_point(color = "darkblue", alpha = 0.6) +
  geom_abline(intercept = 0, slope = 1, color = "red", linetype = "dashed") +
  labs(title = "SVM: Actual vs Predicted Salaries",
       x = "Actual Salary (USD)",
       y = "Predicted Salary (USD)") +
  theme_minimal()
```




```r
# Plot 2: Residuals vs Predicted Salaries
ggplot(data.frame(Predicted = svm_predictions, Residuals = svm_residuals),
       aes(x = Predicted, y = Residuals)) +
  geom_point(color = "darkgreen", alpha = 0.6) +
  geom_hline(yintercept = 0, color = "red", linetype = "dashed") +
  labs(title = "SVM: Residuals vs Predicted Salaries",
       x = "Predicted Salary (USD)",
       y = "Residuals (USD)") +
  theme_minimal()
```




```
# SVM RMSE output
cat(paste0("SVM RMSE: ", round(svm_rmse, 2), "\n"))
```

```
## SVM RMSE: 69950.19
```

```
# The SVM model had a higher RMSE of 69,950.19, indicating lower accuracy than the Random Forest model;

# ========================
# 9. Model Evaluation (Linear Regression)
# ========================

# Create dummy variables for test data using the same structure
test_dummies <- model.matrix(~ experience_level + employment_type +
                              remote_ratio + company_location + company_size,
                              data = test_data)[, -1]  # Exclude intercept

# Ensure the same columns are present in the test set (if necessary)
missing_cols <- setdiff(colnames(train_dummies), colnames(test_dummies))
if (length(missing_cols) > 0) {
  test_dummies <- cbind(test_dummies, matrix(0, nrow = nrow(test_dummies), ncol = length(missing_cols))
  colnames(test_dummies)[(ncol(test_dummies) - length(missing_cols) + 1):ncol(test_dummies)] <- missing
}

# Linear Regression predictions on the test set
lm_predictions <- predict(lm_model, newdata = as.data.frame(test_dummies))
```

```
## Warning in predict.lm(lm_model, newdata = as.data.frame(test_dummies)):
## prediction from rank-deficient fit; attr(*, "non-estim") has doubtful cases
```

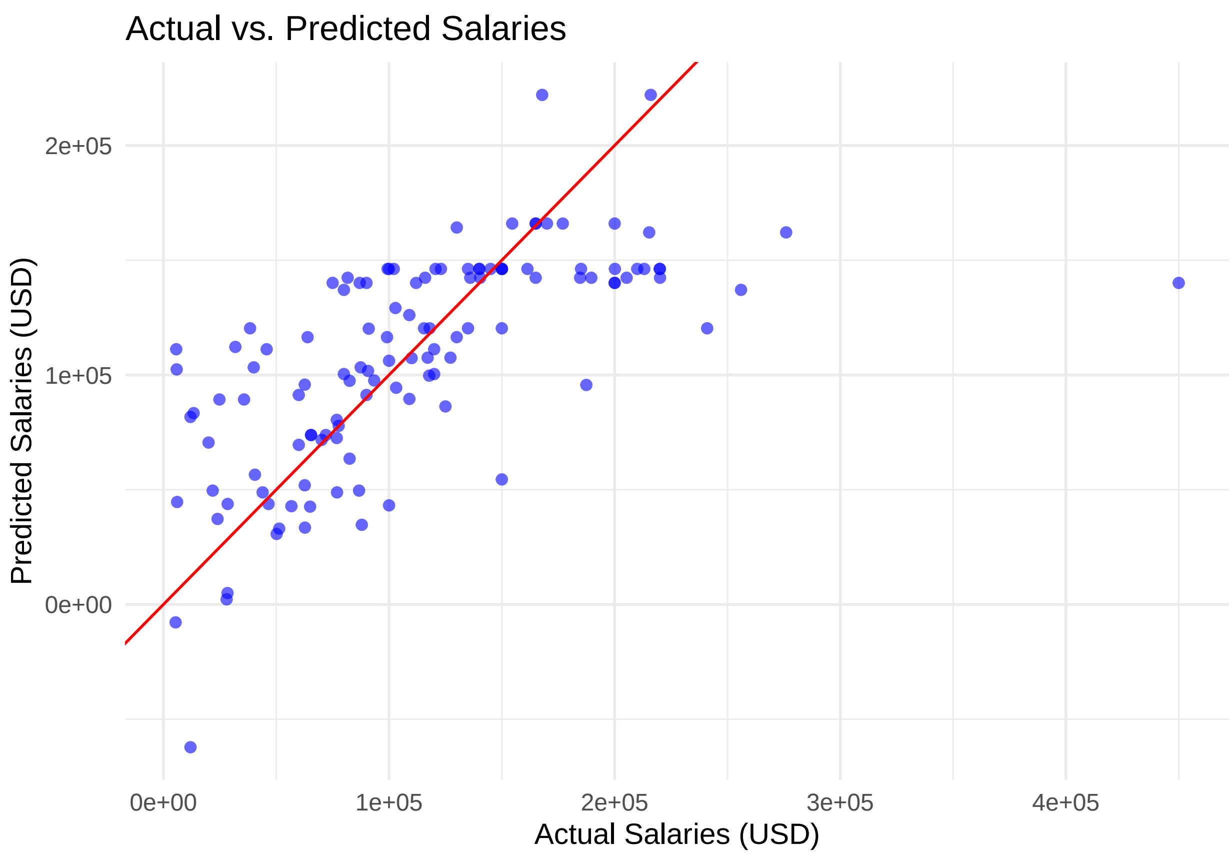
```
lm_rmse <- RMSE(lm_predictions, test_data$salary_in_usd)
lm_r2 <- R2(lm_predictions, test_data$salary_in_usd)

# Print model performance metrics
cat(paste0("Linear Regression RMSE: ", round(lm_rmse, 2), "\n"))
```

```
## Linear Regression RMSE: 52452.39
```

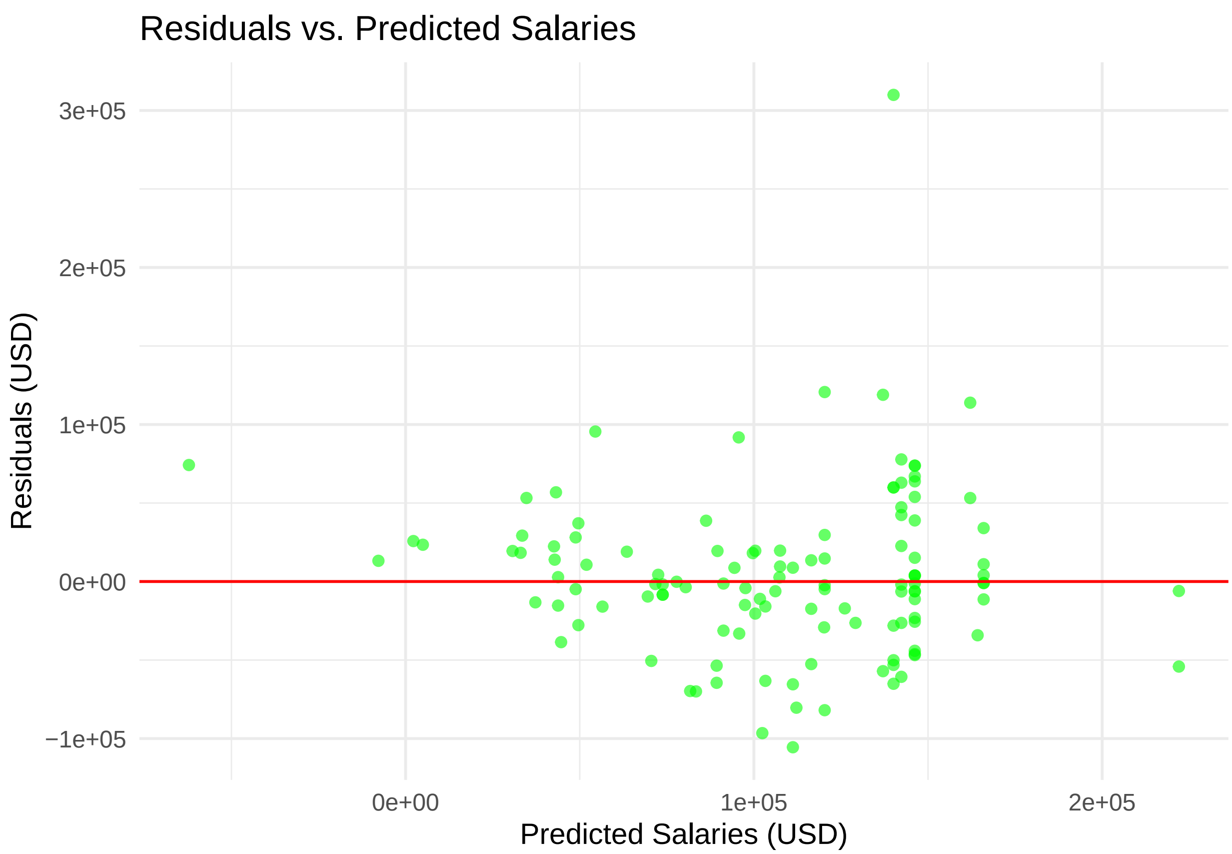
```
cat(paste0("Linear Regression R-squared: ", round(lm_r2, 2), "\n"))
```

```
## Linear Regression R-squared: 0.43
```

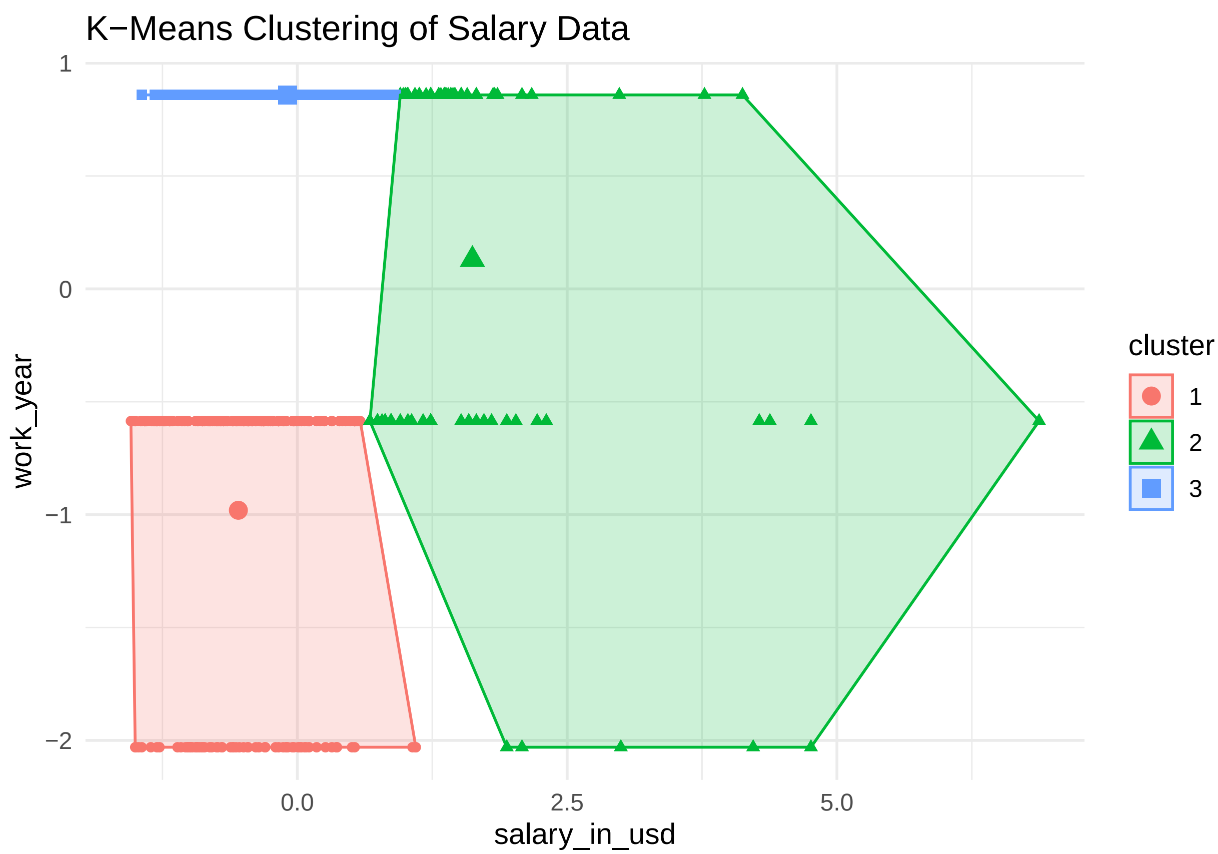
```
# Visual

# Create a data frame for actual vs predicted salaries
results_df <- data.frame(
  Actual = test_data$salary_in_usd,
  Predicted = lm_predictions,
  Residuals = test_data$salary_in_usd - lm_predictions
)

# Plot 1: Actual vs Predicted Salaries
ggplot(results_df, aes(x = Actual, y = Predicted)) +
  geom_point(alpha = 0.6, color = "blue") +  # Scatter plot of actual vs predicted
  geom_abline(slope = 1, intercept = 0, color = "red") +  # Reference line for ideal predictions
  labs(title = "Actual vs. Predicted Salaries",
       x = "Actual Salaries (USD)",
       y = "Predicted Salaries (USD)") +
  theme_minimal()
```




```r
# Plot 2: Residuals vs Predicted Salaries
ggplot(results_df, aes(x = Predicted, y = Residuals)) +
  geom_point(alpha = 0.6, color = "green") +  # Scatter plot of residuals
  geom_hline(yintercept = 0, color = "red") +  # Reference line at 0 residual
  labs(title = "Residuals vs. Predicted Salaries",
       x = "Predicted Salaries (USD)",
       y = "Residuals (USD)") +
  theme_minimal()
```



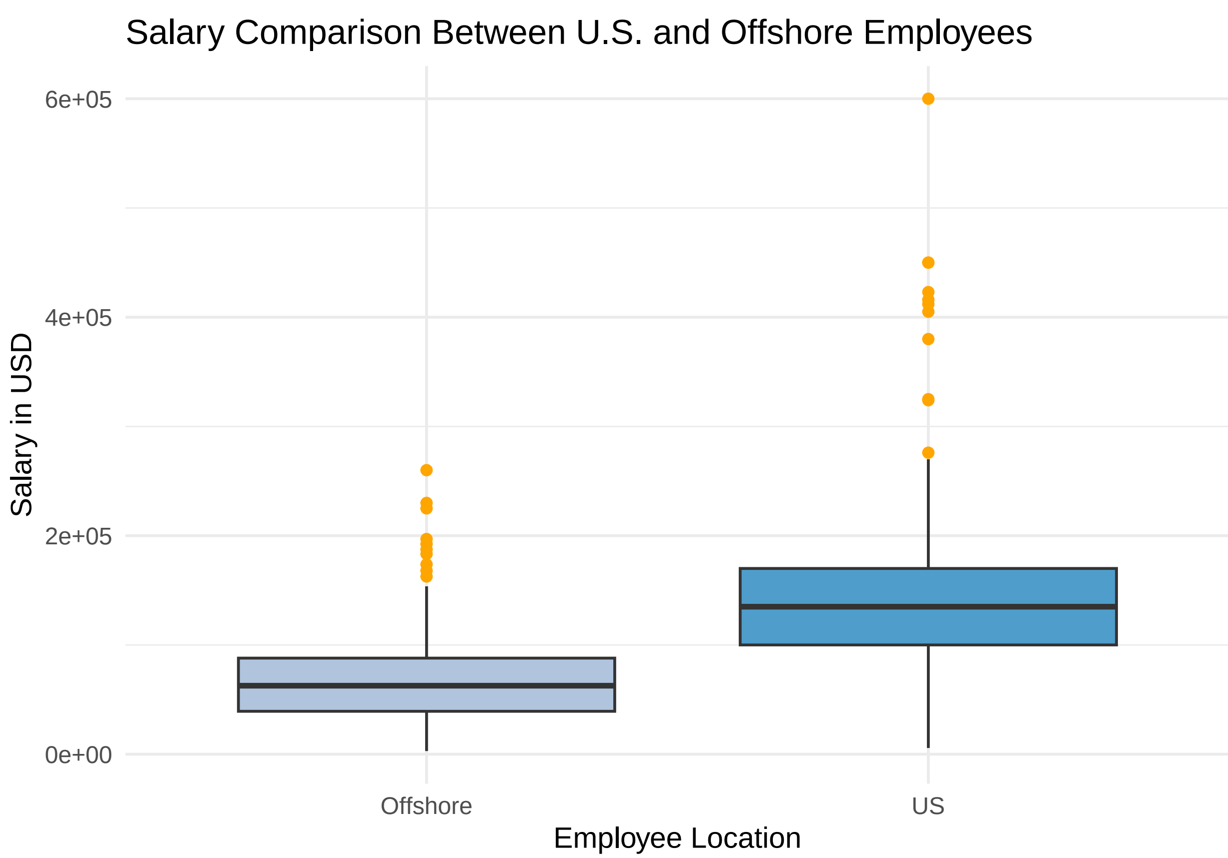

```
# =========================
# 10. Clustering (Unsupervised Learning)
# =========================

# Scale the numeric data for clustering
scaled_data <- scale(numeric_data)

# K-Means clustering with 3 clusters
kmeans_model <- kmeans(scaled_data, centers = 3, nstart = 25)

# Visualize clusters
fviz_cluster(kmeans_model, data = scaled_data, geom = "point", ellipse.type = "convex") +
  labs(title = "K-Means Clustering of Salary Data") +
  theme_minimal()
```




```
# The application of K-Means clustering was visualized through a clustering plot, which revealed the pr

# ========================
# 11. Hypothesis Testing (T-test)
# ========================
# Perform a t-test for salary difference between US and Offshore
us_salaries <- data_clean %>% filter(is_US == "US") %>% pull(salary_in_usd)
offshore_salaries <- data_clean %>% filter(is_US == "Offshore") %>% pull(salary_in_usd)

t_test_result <- t.test(us_salaries, offshore_salaries)
print(t_test_result)
```

```
##
##  Welch Two Sample t-test
##
## data:  us_salaries and offshore_salaries
## t = 16.685, df = 593.1, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  67490.78 85499.20
## sample estimates:
## mean of x mean of y
## 144055.26  67560.27
```

```r
# ========================
# Visualization of T-test Results
# ========================

# Combine salaries into a single data frame for visualization
salary_data <- data_clean %>%
  select(salary_in_usd, is_US)  # Select relevant columns

# Create a box plot to compare salaries
ggplot(salary_data, aes(x = is_US, y = salary_in_usd, fill = is_US)) +
  geom_boxplot(outlier.colour = "orange", outlier.size = 1.5) +  # Box plot with outliers
  labs(title = "Salary Comparison Between U.S. and Offshore Employees",
       x = "Employee Location",
       y = "Salary in USD") +
  scale_fill_manual(values = c("US" = "#4F9DCB", "Offshore" = "#B0C4DE")) +
  theme_minimal() +
  theme(legend.position = "none")  # Remove legend
```




```r
# t-test was used to analyze the difference in mean salaries between U.S. and offshore employees; revea

# ========================
# 12. Conclusion: Summarizing Insights
# ========================

# Bullets
```

```
# Experience level is a key determinant of salary, with more experienced professionals earning signific

# U.S.-based employees earn considerably more than offshore employees, showing a clear geographical dis

# Larger companies tend to offer higher salaries compared to small and medium-sized firms.

#Fully remote positions are associated with higher salaries, suggesting that offering remote work may h

#Random Forest model performed best in predicting salaries, highlighting experience level and employmen

#Support Vector Machine (SVM) was less effective for salary prediction, with higher error rates.

#Clustering analysis revealed three distinct groups within the salary data, offering insights into natu

#T-test confirmed a significant difference between U.S. and offshore salaries, reinforcing the need for

#Recommendations

#Prioritize experience and full-time roles in salary decisions.

#Consider offering remote work to attract top talent.

#Acknowledge and plan for higher salary costs in the U.S. market while leveraging offshore opportunitie


#Explanation

# The analysis for data science salaries provided insights for determining competitive salary ranges. R

#From the modeling techniques applied, the Random Forest model outperformed others, highlighting the im

#Given these findings, the company should prioritize experience and full-time positions when making sal
```