# Load required libraries

library(dplyr)

library(tidyr)

library(lubridate)

# Load the dataset (assuming it has columns: 'Role', 'Employee\_ID', 'Start\_Date', 'End\_Date', 'Turnover')

employee\_data <- read.csv("employee\_turnover.csv")

# Convert date columns to Date type

employee\_data$Start\_Date <- as.Date(employee\_data$Start\_Date, format="%Y-%m-%d")

employee\_data$End\_Date <- as.Date(employee\_data$End\_Date, format="%Y-%m-%d")

# Create a column indicating if the employee has left (Turnover)

employee\_data$Turnover <- ifelse(is.na(employee\_data$End\_Date), 0, 1)

# Calculate turnover rate by role

turnover\_rate <- employee\_data %>%

group\_by(Role) %>%

summarise(Total\_Employees = n(),

Employees\_Left = sum(Turnover, na.rm = TRUE),

Turnover\_Rate = Employees\_Left / Total\_Employees) %>%

arrange(desc(Turnover\_Rate))

# View turnover rate by role

head(turnover\_rate)

# Predict vacancies: Assume that a higher turnover rate indicates a higher need for hiring

# Rank roles based on turnover rate

vacancy\_prediction <- turnover\_rate %>%

mutate(Predicted\_Vacancies = round(Turnover\_Rate \* Total\_Employees, 0)) %>%

arrange(desc(Predicted\_Vacancies))

# View predicted vacancies for each role

head(vacancy\_prediction)

# Save the vacancy prediction to a new CSV file

write.csv(vacancy\_prediction, "predicted\_job\_vacancies.csv", row.names = FALSE)

# Output message

cat("Job vacancy predictions have been successfully added and saved to 'predicted\_job\_vacancies.csv'.")