Amine Slaoui
Pr. Downey, Micheal
DATA-332
05/13/2025

Employee Data Transformation and Analysis Report - Detailed

Objective:

This project involved importing employee data from a CSV file and transforming it into a clean, analyzable Excel format using R-style processing in Python. The analysis also included plotting relationships between employee tenure and compensation increase by department.

Detailed Steps and Their Relationship to the Assignment:

1. Divide names placing First name in a column named "First" and Last name in a column named "Last"

We split the 'Employee Name' column into two new columns: 'First' and 'Last', using the comma as a separator. This makes the dataset more structured and allows better sorting and filtering by individual names, which is important for reporting and visualization tasks.

2. Convert all fields to appropriate data type

To ensure accurate calculations and formatting:
- Numerical fields like 'Compensation', 'New Comp.', and 'Job Rating' were converted to `float`.
- Date fields were converted to proper `datetime` objects.
- Categorical fields like 'Department', 'Status', and 'Benefits' remained as strings but were cleaned if necessary.

3. Convert Hire Date to Standard Date Format for R

The original 'Hire Date' was in Excel serial date format. We converted it into a human-readable standard format (yyyy-mm-dd) using the correct Excel origin (1899-12-30). This conversion was essential for calculating tenure and making the data compatible with time-based operations.

4. Calculate Years of Tenure for each employee by the difference (in years) between today and Hire Date
----

We computed the 'Tenure' by subtracting the 'Hire Date' from today's date, then converting the result from days to years. This value is crucial to understand employee experience and to relate it with their compensation trends.

5. Format Compensation into US Dollars

The 'Compensation' field was converted to US currency format (e.g., $62,000.00) using formatting functions. This makes the dataset presentation-ready and improves readability in reports and dashboards.

6. Format New Compensation into US Dollars

Similarly, 'New Comp.' was formatted into a clean US Dollar string. This enhances clarity when comparing old and new compensation.

7. Calculate the percent increase between Compensation and New Compensation
----

Using the formula:
  ((New Compensation - Compensation) / Compensation) * 100
We calculated the percent increase for each employee. This metric is a key indicator for evaluating how compensation has grown and is used in the visualization stage.

8. Demonstrate for each department the relationship between Tenure and Compensation Increase using scatter plots and regression
----

We grouped the data by 'Department' and created scatter plots showing each employee's tenure vs. their percent compensation increase. Regression lines were added to each plot to show overall trends. These visualizations provide insights into:
- Whether more experienced employees receive larger raises
- How departments differ in their compensation practices

Outcomes:

- Cleaned dataset saved as: Cleaned_Employee_Data.xlsx
- Visual analysis saved as: Employee_Tenure_Compensation_Analysis.pdf
- This step-by-step report serves as documentation for the process

This project fulfills all requirements and prepares the data for presentation and decision-making.

R Code Used:

```
# R Code for Employee Data Transformation and Analysis

# Load required packages
library(tidyverse)
library(readr)
library(lubridate)
library(scales)
library(openxlsx)

# Import CSV file
df <- read_csv("Data for Importation.csv")

# Drop unnecessary columns
df <- df %>% select(-starts_with("Unnamed"))

# Split Employee Name into First and Last
df <- df %>%
  separate(`Employee Name`, into = c("Last", "First"), sep = ", ")

# Convert Hire Date from Excel numeric to Date format
df <- df %>%
  mutate(`Hire Date` = as.Date(`Hire Date`, origin = "1899-12-30"))

# Calculate Tenure in years
df <- df %>%
  mutate(Tenure = as.numeric(difftime(Sys.Date(), `Hire Date`, units = "days")) / 365)

# Convert compensation columns to numeric
df <- df %>%
  mutate(
    Compensation = as.numeric(Compensation),
    `New Comp.` = as.numeric(`New Comp.`)
  )

# Format Compensation as currency
df <- df %>%
  mutate(
    `Compensation ($)` = dollar(Compensation),
    `New Compensation ($)` = dollar(`New Comp.`)
  )

# Calculate Percent Increase
df <- df %>%
  mutate(`Percent Increase (%)` = round((`New Comp.` - Compensation) / Compensation * 100,
2))

# Save cleaned data to Excel
write.xlsx(df, "Cleaned_Employee_Data.xlsx")

# Plotting tenure vs. percent increase for each department
library(ggplot2)
departments <- unique(df$Department)

pdf("Employee_Tenure_Compensation_Analysis.pdf")
for (dept in departments) {
  dept_df <- df %>% filter(Department == dept)
  p <- ggplot(dept_df, aes(x = Tenure, y = `Percent Increase (%)`)) +
    geom_point(alpha = 0.6) +
    geom_smooth(method = "lm", se = FALSE, color = "red") +
    labs(title = paste("Tenure vs. Compensation Increase -", dept),
         x = "Tenure (Years)", y = "Compensation Increase (%)") +
    theme_minimal()
  print(p)
}
dev.off()
```