# Load necessary libraries

library(readr) # For reading CSV files

library(ggplot2) # For creating plots

library(broom) # For tidying model output

library(gridExtra) # For arranging multiple plots

# Step 1: Read the data

# The file should have four columns: "calls", "duration", "work", and "school"

SimulationProject <- read\_csv("SimulationProject.csv")

# Print the first few rows of the data to verify it's loaded correctly

print(head(SimulationProject))

# Step 2: Explore the data

# Calculate summary statistics for all variables

summary(SimulationProject)

# Step 3: Perform multiple linear regression

# Fit a linear model where duration is the dependent variable and calls, work, and school are the independent variables

model <- lm(duration ~ calls + work + school, data = SimulationProject)

# Step 4: Examine the model results

# Print a summary of the model, including coefficients, R-squared, and p-values

summary(model)

# Use broom to get a tidy version of the model coefficients

tidy\_model <- tidy(model)

print(tidy\_model)

# Step 5: Visualize the regression results for each variable

# Create individual plots for calls, work, and school

plot\_calls <- ggplot(SimulationProject, aes(x = calls, y = duration)) +

geom\_point() +

geom\_smooth(method = "lm", se = FALSE, color = "red") +

labs(title = "Linear Regression: duration vs. calls",

x = "calls",

y = "duration") +

theme\_minimal()

plot\_work <- ggplot(SimulationProject, aes(x = work, y = duration)) +

geom\_point() +

geom\_smooth(method = "lm", se = FALSE, color = "red") +

labs(title = "Linear Regression: duration vs. work",

x = "work",

y = "duration") +

theme\_minimal()

plot\_school <- ggplot(SimulationProject, aes(x = school, y = duration)) +

geom\_point() +

geom\_smooth(method = "lm", se = FALSE, color = "red") +

labs(title = "Linear Regression: duration vs. school",

x = "school",

y = "duration") +

theme\_minimal()

# Step 6: Show relationships between work and calls, and work and school

# Scatter plot for work vs. calls

plot\_work\_calls <- ggplot(SimulationProject, aes(x = work, y = calls)) +

geom\_point() +

geom\_smooth(method = "lm", se = FALSE, color = "blue") +

labs(title = "Scatter Plot: work vs. calls",

x = "work",

y = "calls") +

theme\_minimal()

# Scatter plot for work vs. school

plot\_work\_school <- ggplot(SimulationProject, aes(x = work, y = school)) +

geom\_point() +

geom\_smooth(method = "lm", se = FALSE, color = "green") +

labs(title = "Scatter Plot: work vs. school",

x = "work",

y = "school") +

theme\_minimal()

# Arrange the plots vertically on the same slide (for the main regression plots)

grid.arrange(plot\_calls, plot\_work, plot\_school, ncol = 1)

# Arrange the relationship plots (work vs calls, and work vs school) on a separate slide

grid.arrange(plot\_work\_calls, plot\_work\_school, ncol = 1)

# Step 7: Model diagnostics

# Create diagnostic plots to check assumptions

par(mfrow = c(2, 2)) # Set up a 2x2 plot layout

plot(model)

# Step 8: Predictions

# Create a data frame with new values for calls, work, and school for prediction

new\_data <- data.frame(

calls = seq(min(SimulationProject$calls), max(SimulationProject$calls), length.out = 100),

work = seq(min(SimulationProject$work), max(SimulationProject$work), length.out = 100),

school = seq(min(SimulationProject$school), max(SimulationProject$school), length.out = 100)

)

# Make predictions using the model

predictions <- predict(model, newdata = new\_data, interval = "confidence")

# Combine predictions with new\_data

prediction\_data <- cbind(new\_data, predictions)

# Plot the original data, regression line, and confidence interval for each x variable

plot\_ci\_calls <- ggplot() +

geom\_point(data = SimulationProject, aes(x = calls, y = duration)) +

geom\_line(data = prediction\_data, aes(x = calls, y = fit), color = "blue") +

geom\_ribbon(data = prediction\_data, aes(x = calls, ymin = lwr, ymax = upr), alpha = 0.2) +

labs(title = "Linear Regression with Confidence Interval (calls)",

x = "calls",

y = "duration") +

theme\_minimal()

plot\_ci\_work <- ggplot() +

geom\_point(data = SimulationProject, aes(x = work, y = duration)) +

geom\_line(data = prediction\_data, aes(x = work, y = fit), color = "red") +

geom\_ribbon(data = prediction\_data, aes(x = work, ymin = lwr, ymax = upr), alpha = 0.2) +

labs(title = "Linear Regression with Confidence Interval (work)",

x = "work",

y = "duration") +

theme\_minimal()

plot\_ci\_school <- ggplot() +

geom\_point(data = SimulationProject, aes(x = school, y = duration)) +

geom\_line(data = prediction\_data, aes(x = school, y = fit), color = "green") +

geom\_ribbon(data = prediction\_data, aes(x = school, ymin = lwr, ymax = upr), alpha = 0.2) +

labs(title = "Linear Regression with Confidence Interval (school)",

x = "school",

y = "duration") +

theme\_minimal()

# Arrange the confidence interval plots vertically on the same slide

grid.arrange(plot\_ci\_calls, plot\_ci\_work, plot\_ci\_school, ncol = 1)

# Step 9: Print the R-squared value and the regression equation

cat("R-squared:", summary(model)$r.squared, "\n")

cat("Regression equation: duration =",

round(coef(model)[1], 3), "+",

round(coef(model)[2], 3), "\* calls +",

round(coef(model)[3], 3), "\* work +",

round(coef(model)[4], 3), "\* school\n")