# Install the GGally package

install.packages("GGally")

install.packages("corrplot")

install.packages("ggplot2")

install.packages("dplyr")

# Load necessary libraries

library(ggplot2)

library(dplyr)

library(GGally) # Now you can successfully load the GGally library

library(corrplot)

# Load the dataset

data <- read.csv('/content/data\_application\_energy.csv')

# Convert 'date' to Date-Time format

data$date <- as.POSIXct(data$date, format="%Y-%m-%d %H:%M:%S")

# 1. Summary statistics of the dataset

summary(data)

# 2. Distribution of energy consumption (Appliances)

ggplot(data, aes(x=Appliances)) +

geom\_histogram(bins=50, fill="skyblue", color="black") +

ggtitle("Distribution of Appliances Energy Consumption") +

xlab("Appliances Energy Consumption") +

ylab("Frequency")

# 3. Correlation matrix heatmap for numeric variables

numeric\_cols <- select\_if(data, is.numeric)

corr\_matrix <- cor(numeric\_cols)

corrplot(corr\_matrix, method="color", type="lower", tl.cex = 0.7, number.cex = 0.7, tl.col = "black")

# Linear Regression Model and Visualization

# Predicting Appliances energy consumption using a simple linear regression

model <- lm(Appliances ~ T1 + RH\_1 + T\_out + Windspeed, data = data)

summary(model)

# Residuals vs Fitted Plot

plot(model, which = 1)

# 4. Time series plot of Appliances energy consumption

ggplot(data, aes(x=date, y=Appliances)) +

geom\_line(color="darkgreen") +

ggtitle("Time Series Plot of Appliances Energy Consumption") +

xlab("Date") +

ylab("Appliances Energy Consumption")

# 5. Pair plot of selected variables

selected\_vars <- data %>% select(Appliances, T1, RH\_1, T\_out, Windspeed)

ggpairs(selected\_vars, title = "Pair Plot of Selected Variables")

# Linear Regression Model and Visualization

# Predicting Appliances energy consumption using a simple linear regression

model <- lm(Appliances ~ T1 + RH\_1 + T\_out + Windspeed, data = data)

summary(model)

# Residuals vs Fitted Plot

plot(model, which = 1)

# Normal Q-Q Plot

plot(model, which = 2)

# Scatter plot with regression line (Appliances vs T\_out)

ggplot(data, aes(x=T\_out, y=Appliances)) +

geom\_point(alpha=0.4) +

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

ggtitle("Regression Line of Appliances vs T\_out") +

xlab("T\_out") +

ylab("Appliances Energy Consumption")

# Scatter plot with regression line (Appliances vs T\_out)

# Load necessary libraries

library(readr) # For reading CSV files

library(ggplot2) # For plotting

# Load the dataset

# Replace 'data\_application\_energy.csv' with the actual file name

# If the file is not in your current directory, provide the full path

data <- read\_csv("data\_application\_energy.csv")

# Check the structure of the data to identify relevant variables

str(data)

# Fit a linear regression model using key predictors

# Example: using temperature, humidity, windspeed, and visibility as predictors

model <- lm(Appliances ~ T1 + RH\_1 + T2 + RH\_2 + T\_out + Windspeed + Visibility, data = data)

# Display summary of the regression model to see coefficients and p-values

summary(model)

# Plot residuals vs fitted values to assess model fit

ggplot(model, aes(.fitted, .resid)) +

geom\_point(color = "blue") +

geom\_hline(yintercept = 0, linetype = "dashed") +

ggtitle("Residuals vs Fitted Values") +

xlab("Fitted Values") + ylab("Residuals")

# Plot the actual vs predicted values

data$predicted <- predict(model)

ggplot(data, aes(x = predicted, y = Appliances)) +

geom\_point(color = "red") +

geom\_abline(slope = 1, intercept = 0, color = "blue", linetype = "dashed") +

ggtitle("Actual vs Predicted Appliance Consumption") +

xlab("Predicted Values") + ylab("Actual Values")

# Print your current working directory to check where R is looking for files

print(getwd())

# List files in your current working directory to see if your data file is present

print(list.files())

# Load necessary libraries

library(readr) # For reading CSV files

library(ggplot2) # For plotting

library(dplyr) # For data manipulation

# Load the dataset

# Replace 'data\_application\_energy.csv' with the actual file name

# If the file is not in the same directory as your script, provide the full path

data <- read\_csv("data\_application\_energy.csv")

# Inspect the structure of the data to choose predictors

str(data)

# Fit a linear regression model

# We will use key variables (like temperature, humidity, etc.) as predictors

# You can modify the variables as per your requirements

model <- lm(Appliances ~ T1 + RH\_1 + T2 + RH\_2 + T\_out + Windspeed + Visibility, data = data)

# Display the summary of the regression model to show coefficients, R-squared, and p-values

summary(model)

# Plot residuals vs fitted values to assess the model

ggplot(data, aes(x = model$fitted.values, y = model$residuals)) +

geom\_point(color = "blue") +

geom\_hline(yintercept = 0, linetype = "dashed") +

ggtitle("Residuals vs Fitted Values") +

xlab("Fitted Values") + ylab("Residuals")

# Plot actual vs predicted values

data$predicted <- predict(model)

ggplot(data, aes(x = predicted, y = Appliances)) +

geom\_point(color = "red") +

geom\_abline(slope = 1, intercept = 0, color = "blue", linetype = "dashed") +

ggtitle("Actual vs Predicted Appliance Energy Consumption") +

xlab("Predicted Energy Consumption") + ylab("Actual Energy Consumption")

# Load necessary libraries

library(ggplot2)

library(dplyr)

library(Metrics) # For RMSE calculation

# Load the dataset

data <- read.csv('/content/data\_application\_energy.csv')

# Convert 'date' to Date-Time format if needed

data$date <- as.POSIXct(data$date, format="%Y-%m-%d %H:%M:%S")

# Select the dependent variable (Appliances) and independent variables for the regression

model\_data <- data %>%

select(Appliances, T1, RH\_1, T\_out, Windspeed, Visibility)

# Fit the linear regression model

model <- lm(Appliances ~ ., data = model\_data)

# Display the summary of the linear regression model

summary(model)

# Predict the values based on the model

predictions <- predict(model, model\_data)

# Calculate RMSE

rmse\_value <- rmse(model\_data$Appliances, predictions)

print(paste("RMSE: ", rmse\_value))

# Calculate R-squared

r\_squared <- summary(model)$r.squared

print(paste("R-squared: ", r\_squared))

install.packages("Metrics")