```r
# ==============================================================
# R Script: BMW Regional Market Analysis
# Tasks:
#   1. Cleaned dataset summary and statistical profile
#   2. Exploratory Data Visualizations (EDA)
#   3. Predictive modeling with accuracy metrics
#   4. Visual comparison of actual vs predicted prices
#   5. Analytical discussion of results and implications
# ==============================================================

# ---- Load required packages ----
library(tidyverse)
library(caret)
library(randomForest)
library(ggplot2)
library(forcats)
library(GGally)

# ---- 1. Load and Clean Data ----
df <- read.csv("regional_bmw_data_clean.csv", stringsAsFactors = FALSE)

# Check structure
str(df)

# Basic cleaning
df <- df %>%
  drop_na(Price_USD, Year, Mileage_KM)
```

```r
# Convert data types
df$Year <- as.integer(df$Year)
df$Mileage_KM <- as.numeric(df$Mileage_KM)
df$Engine_Size_L <- as.numeric(df$Engine_Size_L)
df$Price_USD <- as.numeric(df$Price_USD)

# Fill missing categoricals
for(c in c("Model","Region","Fuel_Type","Transmission")){
  if(c %in% names(df)){
    df[[c]][is.na(df[[c]])] <- "Unknown"
  }
}

# ---- Cleaned Dataset Summary and Statistical Profile ----
cat("\n==== Summary Statistics ====\n")
print(summary(df))
cat("\n==== Missing Values ====\n")
print(colSums(is.na(df)))

# ---- 2. Exploratory Data Visualizations (EDA) ----

# Price Distribution
ggplot(df, aes(x = Price_USD)) +
  geom_histogram(bins = 60, fill = "goldenrod", color = "white") +
  labs(title = "Distribution of BMW Prices", x = "Price (USD)", y = "Count")
```

```r
# Price vs Mileage
ggplot(sample_n(df, min(3000, nrow(df))), aes(x = Mileage_KM, y = Price_USD)) +
  geom_point(alpha = 0.5, color = "steelblue") +
  labs(title = "Price vs Mileage", x = "Mileage (KM)", y = "Price (USD)")

# Price by Year
ggplot(df %>% filter(Year >= 2000), aes(x = factor(Year), y = Price_USD)) +
  geom_boxplot(fill = "lightblue") +
  labs(title = "Price by Year", x = "Model Year", y = "Price (USD)") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

# Median Price by Region
df %>%
  group_by(Region) %>%
  summarise(Median_Price = median(Price_USD, na.rm = TRUE)) %>%
  arrange(desc(Median_Price)) %>%
  slice_head(n = 12) %>%
  ggplot(aes(x = reorder(Region, Median_Price), y = Median_Price)) +
  geom_col(fill = "darkgreen") +
  coord_flip() +
  labs(title = "Top 12 Regions by Median Price", x = "Region", y = "Median Price (USD)")

# ---- 3. Predictive Modeling ----

# Reduce model categories to top 20
df$Model <- fct_lump(df$Model, n = 20, other_level = "Other")
```

```
# Prepare modeling data
model_data <- df %>%
  select(Year, Mileage_KM, Engine_Size_L, Region, Fuel_Type, Transmission, Model, 
Price_USD) %>%
  drop_na()

# Split into training and testing
set.seed(123)
train_index <- createDataPartition(model_data$Price_USD, p = 0.8, list = FALSE)
train_data <- model_data[train_index,]
test_data <- model_data[-train_index,]

# One-hot encode categorical variables
dummies <- dummyVars(Price_USD ~ ., data = train_data)
X_train <- predict(dummies, newdata = train_data)
X_test <- predict(dummies, newdata = test_data)

y_train <- train_data$Price_USD
y_test <- test_data$Price_USD

# ---- Fit Random Forest Model ----
set.seed(123)
rf_model <- randomForest(x = X_train, y = y_train, ntree = 200, importance = TRUE)

# Predictions
rf_pred <- predict(rf_model, newdata = X_test)
```

```r
# ---- Fit Linear Regression (Baseline) ----
lm_model <- train(Price_USD ~ ., data = train_data, method = "lm")
lm_pred <- predict(lm_model, newdata = test_data)

# ---- 4. Accuracy Metrics ----
rf_metrics <- postResample(pred = rf_pred, obs = y_test)
lm_metrics <- postResample(pred = lm_pred, obs = y_test)

cat("\n==== Model Performance ====\n")
metrics <- rbind(
  data.frame(Model = "Random Forest", RMSE = rf_metrics["RMSE"], R2 =
rf_metrics["Rsquared"], MAE = mean(abs(y_test - rf_pred))),
  data.frame(Model = "Linear Regression", RMSE = lm_metrics["RMSE"], R2 =
lm_metrics["Rsquared"], MAE = mean(abs(y_test - lm_pred)))
)
print(metrics)

# ---- 5. Actual vs Predicted Plot ----
comparison_df <- data.frame(
  Actual = y_test,
  Predicted_RF = rf_pred,
  Predicted_LM = lm_pred
)

ggplot(comparison_df, aes(x = Actual, y = Predicted_RF)) +
  geom_point(color = "darkorange", alpha = 0.6) +
  geom_abline(slope = 1, intercept = 0, color = "red", linetype = "dashed") +
```

```r
  labs(title = "Actual vs Predicted Prices (Random Forest)",
       x = "Actual Price (USD)",
       y = "Predicted Price (USD)")

# ---- 6. Analytical Discussion ----
cat("\n==== Analytical Discussion ====\n")
cat("
Data cleaning ensured removal of incomplete rows and standardization of numeric variables.
EDA reveals a strong negative relationship between mileage and price, and newer cars command higher prices.
Region and Model type show clear pricing segmentation — luxury or rare models priced higher.

Random Forest outperformed Linear Regression, indicating that nonlinear interactions drive pricing.
However, errors remain large due to unobserved factors (condition, trim, etc.).

Implications:
- Buyers: Use mileage and model year as negotiation levers.
- Sellers: Leverage regional and model-based pricing insights.
- Dealers: Collect additional condition and feature data to enhance pricing accuracy.

Next Steps:
- Evaluate variable importance (varImpPlot(rf_model)).
- Explore log(Price) modeling for skew reduction.
- Integrate additional vehicle attributes for richer predictions.
")
```