National University of Singapore
TCX2002 Introduction to Business Analytics
Tutorial 4

Lesson 6 – Predictive Analytics II - MLR, Model complexity, Generalization, and Bias-Variance Tradeoff

1. Explore the data

Scenario: Use the same data from Tutorial 3.

Create scatterplot for "Sales vs Marketing Spend" & "Sales vs Tourists".

```
# Sales vs Marketing Spend
p1 <- ggplot(df, aes(x = Marketing_Spend, y = Sales)) +
  geom_point(color = "red") +
  ggtitle("Sales vs. Marketing Spend") +
  theme_minimal() +
  theme(
    panel.grid.major = element_line(color = "lightgray", linetype = "dotted"),
    panel.grid.minor = element_line(color = "lightgray", linetype = "dotted")
  )


# Combine the two plots side by side
grid.arrange(p1, p2, ncol = 2)

# correlation between independent vars
cor(df[,c('Tourist_Arrivals', "Marketing_Spend")])

# plotting correlations
corrplot(cor(df[, sapply(df, is.numeric)],
             use="complete.obs"),
             method = "number",
             type='lower')
```

What are your comments on the correlation plot?

2. Simple Linear Regression, Multiple Linear Regression & RMSE

```
# build SLR
model1 = lm(Sales ~ Marketing_Spend, data = df)
summary(model1)
```

National University of Singapore
TCX2002 Introduction to Business Analytics
Tutorial 4

```
# R Code: Complete Assumption Check
model <- lm(Sales ~ Marketing_Spend, data = df)

# All-in-one diagnostic plots
par(mfrow = c(2, 2))
plot(model)
par(mfrow = c(1, 1))

# Interpretation
# Plot 1: Residuals vs Fitted - checks linearity & homoscedasticity
# Plot 2: Q-Q plot - checks normality
# Plot 3: Scale-Location - checks homoscedasticity
# Plot 4: Residuals vs Leverage - identifies outliers
# Formal tests

library(lmtest)
bptest(model) # Breusch-Pagan test for homoscedasticity
shapiro.test(residuals(model)) # Shapiro-Wilk test for normality
dwtest(model) # Durbin-Watson test for independence


# More than one predictors

# build MLR
model2 = lm(Sales ~ Marketing_Spend + Tourist_Arrivals, data = df)
summary(model2)

## extracting some more info from the model object
## 1. Add predicted values to original data frame
# one way

df = df%>%
  add_predictions(model2)

## 2. add residuals to original data frame
# one way

df = df%>%
   add_residuals(model2)

#Plot residues; random residue with no pattern is better!
ggplot(df, aes(pred, resid)) +
  geom_point()+geom_ref_line(h = 0)

# 3. RSqr. and Adj-RSqr.

summary(model2)$adj.r.squared
```

```
summary(model2)$r.squared


par(mfrow = c(2, 2))

plot(model2)

par(mfrow = c(1, 1))

bptest(model2) # Breusch-Pagan test for homoscedasticity

shapiro.test(residuals(model2)) # Shapiro-Wilk test for normality

dwtest(model2) # Durbin-Watson test for independence
```

Tutorial 4 Learning Outcomes	
1.	**Build and compare SLR vs MLR; interpret fit and predictions:** Compare models using R-squared and Adjusted R-squared (the latter penalizes complexity). Add predicted values and residuals to the data to visualize residual vs predicted; a random cloud around zero implies a well-specified mean structure. For performance, report RMSE (and/or MAE) on a holdout if available; lower RMSE indicates better predictive accuracy. If Adj-R^2 improves and residuals look healthier in MLR without clear multicollinearity, prefer MLR.
2.	**Check assumptions and guard generalization (bias–variance tradeoff):** Use the 4 base plots on the fitted model(s). Aim for (i) Residuals vs Fitted with no structure (linearity, constant variance), (ii) Q–Q plot close to the line (approx. normal residuals), (iii) Scale–Location flat trend (homoscedasticity), and (iv) limited high-leverage/Cook's distance points. Formal tests: Breusch–Pagan (homoscedasticity), Shapiro–Wilk (normality; interpret with caution), Durbin–Watson (independence). If violated, consider transformations (e.g., logging Sales), robust SE for inference, adding missing structure (interactions/seasonality), or time-series terms if autocorrelation exists.