R Notebook

This is an R Markdown Notebook. When you execute code within the notebook, the results appear beneath the code.

Try executing this chunk by clicking the *Run* button within the chunk or by placing your cursor inside it and pressing *Cmd+Shift+Enter*.

```
data <- read.csv("/Users/tehila/Downloads/auto-mpg(1).csv", header = TRUE)
head(data)
```

```
##   mpg cylinder displacement horsepower weight acceleration model.year
origin
## 1  18        8          307        130   3504         12.0         70
1
## 2  15        8          350        165   3693         11.5         70
1
## 3  18        8          318        150   3436         11.0         70
1
## 4  16        8          304        150   3433         12.0         70
1
## 5  17        8          302        140   3449         10.5         70
1
## 6  15        8          429        198   4341         10.0         70
1
##                    car.name
## 1 chevrolet chevelle malibu
## 2         buick skylark 320
## 3        plymouth satellite
## 4             amc rebel sst
## 5               ford torino
## 6          ford galaxie 500
```

```
# Subset the data for training (first 300 samples)
train_data <- data[1:300, ]

# Subset the data for testing (remaining 98 samples)
test_data <- data[301:398, ]

# Subset the data for training (first 300 samples)

# Subset the data for testing (remaining 98 samples)
test_data <- data[301:398, ]

#Convert horsepower to numeric if it's a factor
data$horsepower <- as.numeric(as.character(data$horsepower))
```

```
## Warning: NAs introduced by coercion
```

```r
train_data$horsepower <- as.numeric(as.character(train_data$horsepower))
```

```
## Warning: NAs introduced by coercion
```

```r
test_data$horsepower <- as.numeric(as.character(test_data$horsepower))
```

```
## Warning: NAs introduced by coercion
```

```r
# Check for missing values in the training and testing sets
sum(is.na(train_data$horsepower))
```

```
## [1] 2
```

```r
sum(is.na(test_data$horsepower))
```

```
## [1] 4
```

```r
#Remove rows with missing horsepower
train_data <- na.omit(train_data)
test_data <- na.omit(test_data)

# Fit simple linear regression model (predict mpg using weight)
simple_model <- lm(mpg ~ weight, data=train_data)

# View the model summary to check coefficients, R-squared, and p-value
summary(simple_model)
```

```
## 
## Call:
## lm(formula = mpg ~ weight, data = train_data)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -9.1351 -1.8947 -0.0395  1.7362 15.0939 
## 
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 40.4445554  0.6405724   63.14   <2e-16 ***
## weight      -0.0062662  0.0001965  -31.89   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 2.997 on 296 degrees of freedom
## Multiple R-squared:  0.7745, Adjusted R-squared:  0.7737 
## F-statistic:  1017 on 1 and 296 DF,  p-value: < 2.2e-16
```

```r
# Regression equation
cat("Simple Regression Equation: mpg = ", coef(simple_model)[1], "+",
coef(simple_model)[2], "* weight\n")
```

```
## Simple Regression Equation: mpg =  40.44456 + -0.006266213 * weight
```

```r
# Fit multiple linear regression model (predict mpg using weight, horsepower,
and displacement)
multiple_model <- lm(mpg ~ weight + horsepower + displacement,
data=train_data)

# View the model summary to check coefficients, R-squared, and p-value
summary(multiple_model)
```

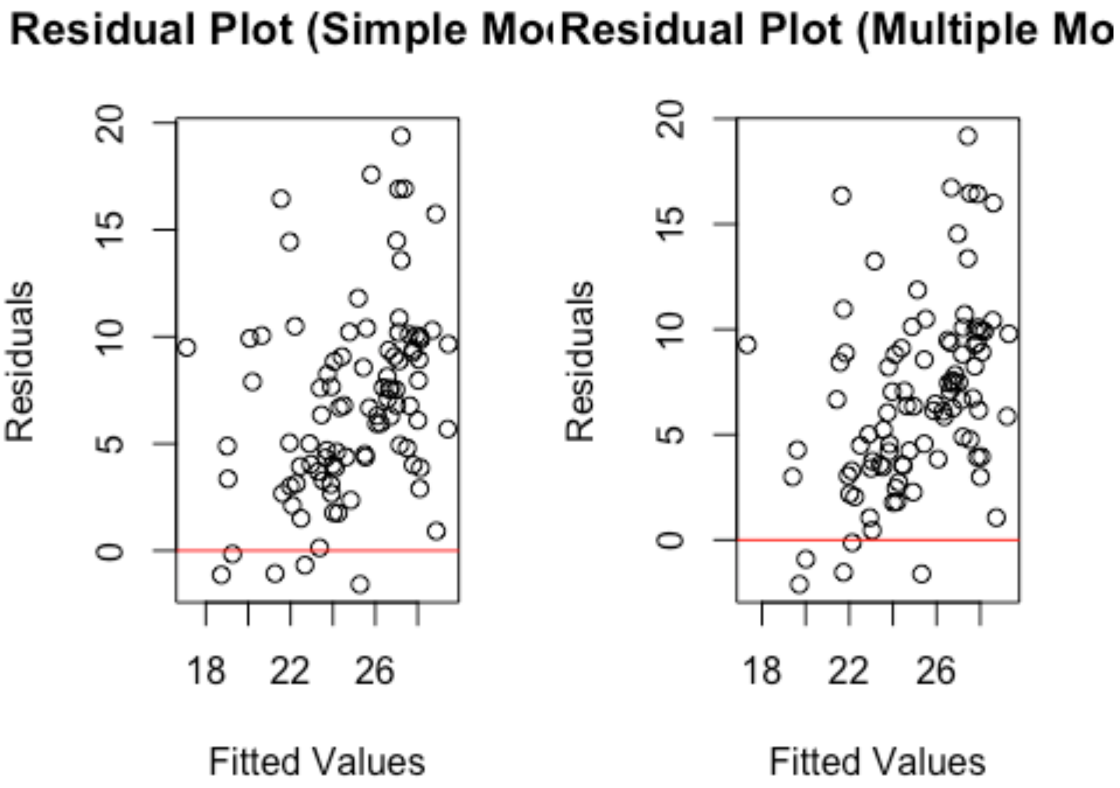
```
##
## Call:
## lm(formula = mpg ~ weight + horsepower + displacement, data = train_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.9396 -1.9036 -0.0611  1.6062 14.7474
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  39.3739544  0.9210731  42.748   <2e-16 ***
## weight       -0.0047898  0.0005328  -8.991   <2e-16 ***
## horsepower   -0.0205727  0.0096748  -2.126   0.0343 *
## displacement -0.0058457  0.0049625  -1.178   0.2398
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.95 on 294 degrees of freedom
## Multiple R-squared:  0.783,  Adjusted R-squared:  0.7808
## F-statistic: 353.6 on 3 and 294 DF,  p-value: < 2.2e-16
```

```r
# Regression equation
cat("Multiple Regression Equation: mpg = ", coef(multiple_model)[1],
          "+", coef(multiple_model)[2], "* weight +",
           coef(multiple_model)[3], "* horsepower +",
           coef(multiple_model)[4], "* displacement\n")
```

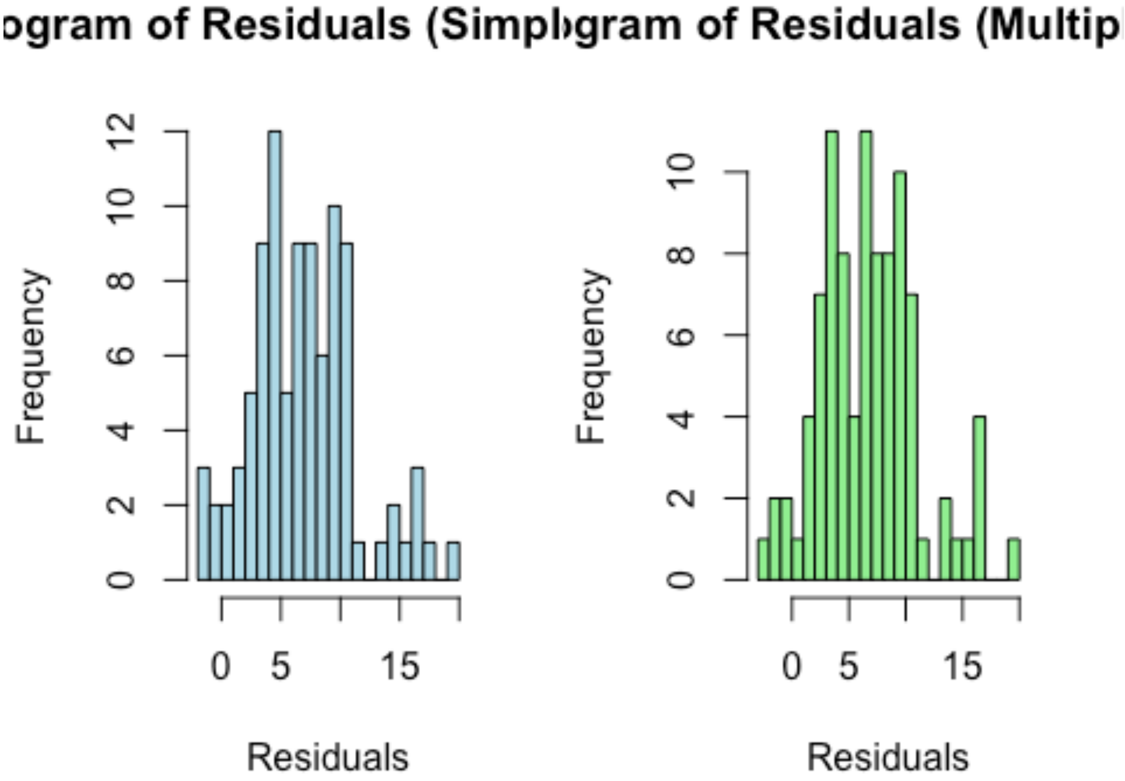
```
## Multiple Regression Equation: mpg =  39.37395 + -0.004789817 * weight + -
0.02057266 * horsepower + -0.0058457 * displacement
```

```r
# Predict mpg for the test data using the simple model
simple_predictions <- predict(simple_model, newdata=test_data)

# Predict mpg for the test data using the multiple model
multiple_predictions <- predict(multiple_model, newdata=test_data)


# Fit the multiple linear regression model
multiple_model <- lm(mpg ~ weight + horsepower + displacement,
data=train_data)

# Predict using the multiple model
```

```r
multiple_predictions <- predict(multiple_model, newdata=test_data)

# Predict mpg for the test data using the simple model
simple_predictions <- predict(simple_model, newdata=test_data)

# Predict mpg for the test data using the multiple model
multiple_predictions <- predict(multiple_model, newdata=test_data)

# Calculate residuals for both models
simple_residuals <- test_data$mpg - simple_predictions
multiple_residuals <- test_data$mpg - multiple_predictions

# Create residual plots
par(mfrow=c(1,2))  # Display side by side

# Simple Linear Regression Residual Plot
plot(simple_predictions, simple_residuals,
               main="Residual Plot (Simple Model)",
               xlab="Fitted Values",
               ylab="Residuals")
abline(h=0, col="red")

# Multiple Linear Regression Residual Plot
plot(multiple_predictions, multiple_residuals,
               main="Residual Plot (Multiple Model)",
               xlab="Fitted Values",
               ylab="Residuals")
abline(h=0, col="red")
```



```r
# Create histograms for residuals
par(mfrow=c(1,2))  # Display side by side

# Simple Model Residual Histogram
hist(simple_residuals, main="Histogram of Residuals (Simple Model)",
               xlab="Residuals", col="lightblue", breaks=20)

# Multiple Model Residual Histogram
hist(multiple_residuals, main="Histogram of Residuals (Multiple Model)",
               xlab="Residuals", col="lightgreen", breaks=20)
```



```
# For simple model
cat("Simple Model R-squared: ", summary(simple_model)$r.squared, "\n")

## Simple Model R-squared:  0.7745094

cat("Simple Model Adjusted R-squared: ", summary(simple_model)$adj.r.squared,
"\n")

## Simple Model Adjusted R-squared:  0.7737476

# For multiple model
cat("Multiple Model R-squared: ", summary(multiple_model)$r.squared, "\n")

## Multiple Model R-squared:  0.7830071

cat("Multiple Model Adjusted R-squared: ",
summary(multiple_model)$adj.r.squared, "\n")

## Multiple Model Adjusted R-squared:  0.7807928

# Calculate Mean Squared Error (MSE) for both models
simple_mse <- mean(simple_residuals^2)
multiple_mse <- mean(multiple_residuals^2)

cat("Simple Model MSE: ", simple_mse, "\n")
```

```r
## Simple Model MSE:  66.80201

cat("Multiple Model MSE: ", multiple_mse, "\n")

## Multiple Model MSE:  64.36983

# Compare predictions vs actual mpg for both models
cat("First few predictions for Simple Model:\n")

## First few predictions for Simple Model:

cat("First few predictions for Multiple Model:\n")

## First few predictions for Multiple Model:

cat("Actual values of mpg for Test Set:\n")

## Actual values of mpg for Test Set:

head(test_data$mpg)

## [1] 23.9 34.2 34.5 31.8 37.3 28.4
```