

# Conducting and Presenting Multiple Linear Regression Analysis Using R

Presentation by Maria Paula Rojas 400962585

## Links

Handout (HTML website) <https://mariarojas01.github.io/data-analysis>

Handout (PDF)

Presentation (HTML)

Presentation (PDF)

## What is Multiple Linear Regression

Regression analysis is used to determine the relationships between two or more variables that exhibit cause-effect patterns, and to make predictions based on those relationships.

$$Y_i = b_0 + b_1 X_{1i} + b_2 X_{2i} + \cdots + b_n X_{ni} + u_i$$

## Basic Syntax in R

```
model <- lm(formula, data = dataset)
```

## Usefull applications

- Business: Demand forecasting, financial risk modeling, market research, etc.
- Social sciences and psychology: explore how variables such as education, income, and demographics influence life satisfaction.
- In medicine: predict patient outcomes from characteristics such as age, BMI, and biomarkers.
- Machine learning and data science: serves both as a baseline predictive model for continuous outcomes and as a tool for feature importance analysis.

## Assupmtions

1. Linearity: The model assumes a linear and additive relationship between each predictor.
2. Normality: The model errors are assumed to follow a normal distribution centered at zero.
3. Homoscedasticity: The errors are expected to have constant variance across all values of the predictors.
4. Independence: Errors must be independent of each other, meaning their co variances are zero.

## Multicollinearity

It occurs when two or more predictor variables are highly correlated, which leads to an increase in the standard errors of the estimated coefficients.

## Steps to do a multiple linear regression analysis

1. Define the research question
2. Prepare and explore the data
3. Fit the initial model with `lm()`
4. Perform diagnostics
5. Refine the model if necessary
6. Summarize results with broom or similar tools
7. Report and interpret results

## Preparing Data and Fitting the Model in R

```
# Load data
data(mtcars)
# Quick overview
summary(mtcars)
```

```
# Load data
data(mtcars)
# Quick overview
summary(mtcars)
```

mpg	cyl	disp	hp
Min. :10.40	Min. :4.000	Min. : 71.1	Min. : 52.0
1st Qu.:15.43	1st Qu.:4.000	1st Qu.:120.8	1st Qu.: 96.5
Median :19.20	Median :6.000	Median :196.3	Median :123.0
Mean :20.09	Mean :6.188	Mean :230.7	Mean :146.7
3rd Qu.:22.80	3rd Qu.:8.000	3rd Qu.:326.0	3rd Qu.:180.0
Max. :33.90	Max. :8.000	Max. :472.0	Max. :335.0

drat	wt	qsec	vs
Min. :2.760	Min. :1.513	Min. :14.50	Min. :0.0000
1st Qu.:3.080	1st Qu.:2.581	1st Qu.:16.89	1st Qu.:0.0000
Median :3.695	Median :3.325	Median :17.71	Median :0.0000
Mean :3.597	Mean :3.217	Mean :17.85	Mean :0.4375
3rd Qu.:3.920	3rd Qu.:3.610	3rd Qu.:18.90	3rd Qu.:1.0000
Max. :4.930	Max. :5.424	Max. :22.90	Max. :1.0000

am	gear	carb
Min. :0.0000	Min. :3.000	Min. :1.000
1st Qu.:0.0000	1st Qu.:3.000	1st Qu.:2.000
Median :0.0000	Median :4.000	Median :2.000
Mean :0.4062	Mean :3.688	Mean :2.812
3rd Qu.:1.0000	3rd Qu.:4.000	3rd Qu.:4.000
Max. :1.0000	Max. :5.000	Max. :8.000

## Fitting a multiple linear regression with lm()

Miles per gallon (mpg) is predicted by weight (wt), horsepower (hp), and number of cylinders (cyl).

```
# Fit the model
model <- lm(mpg ~ wt + hp + cyl, data = mtcars)
# Basic summary output
summary(model)
```

## Output

```
# Fit the model
model <- lm(mpg ~ wt + hp + cyl, data = mtcars)
# Basic summary output
summary(model)
```

Call:

```
lm(formula = mpg ~ wt + hp + cyl, data = mtcars)
```

Residuals:

	Min	1Q	Median	3Q	Max
	-3.9290	-1.5598	-0.5311	1.1850	5.8986

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	38.75179	1.78686	21.687	< 2e-16 ***
wt	-3.16697	0.74058	-4.276	0.000199 ***
hp	-0.01804	0.01188	-1.519	0.140015
cyl	-0.94162	0.55092	-1.709	0.098480 .

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.512 on 28 degrees of freedom

Multiple R-squared: 0.8431, Adjusted R-squared: 0.8263

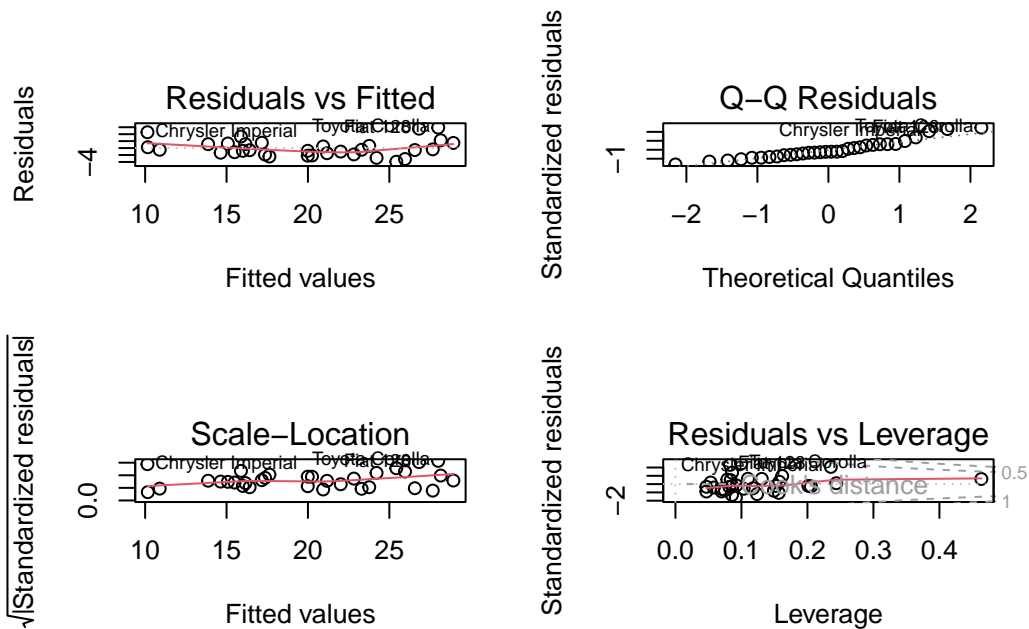
F-statistic: 50.17 on 3 and 28 DF, p-value: 2.184e-11

## Model Diagnostics in R

```
# Basic diagnostic plots (4-panel layout)
par(mfrow = c(2, 2))
```

```
plot(model)
par(mfrow = c(1, 1))
```

```
# Basic diagnostic plots (4-panel layout)
par(mfrow = c(2, 2))
plot(model)
```



```
par(mfrow = c(1, 1))
```

## Checking multicollinearity with VIF

Values above about 5 are often flagged as problematic.

```
# install.packages("car") # run once if not installed
library(car)
vif(model)
```

```
# install.packages("car") # run once if not installed
library(car)
```

Loading required package: carData

```
vif(model)
```

```
      wt      hp      cyl  
2.580486 3.258481 4.757456
```

## Creating predictions

```
# Predictions for the observed data  
mtcars$pred_mpg <- predict(model)  
  
# Create a small new data set for scenario-based predictions  
new_cars <- data.frame(  
  wt = c(2.5, 3.0, 3.5),  
  hp = c(100, 150, 200),  
  cyl = c(4, 6, 8)  
)  
  
predictions<- predict(model, newdata = new_cars)  
  
cbind(new_cars, predictions) # column bind
```

```
# Predictions for the observed data  
mtcars$pred_mpg <- predict(model)  
  
# Create a small new data set for scenario-based predictions  
new_cars <- data.frame(  
  wt = c(2.5, 3.0, 3.5),  
  hp = c(100, 150, 200),  
  cyl = c(4, 6, 8)  
)  
  
predictions<- predict(model, newdata = new_cars)  
  
cbind(new_cars, predictions) #column bind
```

```
      wt  hp cyl predictions  
1 2.5 100   4    25.26408  
2 3.0 150   6    20.89545  
3 3.5 200   8    16.52683
```

## Present predictions

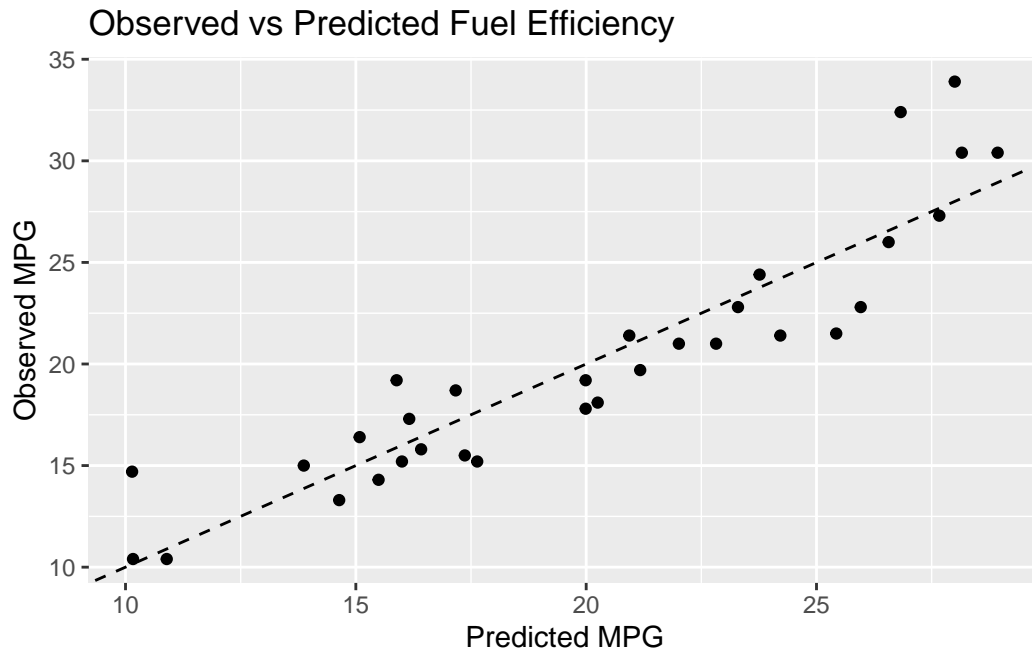
```
# install.packages("ggplot2") # run once
library(ggplot2)

ggplot(mtcars, aes(x = pred_mpg, y = mpg)) +
  geom_point() +
  geom_abline(intercept = 0, slope = 1, linetype = "dashed") +
  labs(
    x = "Predicted MPG",
    y = "Observed MPG",
    title = "Observed vs Predicted Fuel Efficiency"
  )
```

## Output

```
# install.packages("ggplot2") # run once
library(ggplot2)

ggplot(mtcars, aes(x = pred_mpg, y = mpg)) +
  geom_point() +
  geom_abline(intercept = 0, slope = 1, linetype = "dashed") +
  labs(
    x = "Predicted MPG",
    y = "Observed MPG",
    title = "Observed vs Predicted Fuel Efficiency"
  )
```



## Exercise

A small company wants to understand how advertising budget and store size affect monthly sales. For this we are running a regression and predict the outcome of a new store with advertising of 28 K EUR and Store size of 128 m<sup>2</sup>

```
sales_data <- data.frame(  
  Sales = c(120, 150, 170, 200, 220, 250, 275, 300, 320, 340),  
  Advertising = c(10, 15, 14, 20, 22, 25, 27, 30, 31, 33),  
  StoreSize = c(100, 120, 130, 150, 160, 170, 180, 190, 200, 210)  
)
```

## Solution

```
# 1. Fit a multiple linear regression model  
  
model <- lm(Sales ~ Advertising + StoreSize, data = sales_data)  
  
# 2. Show model summary
```



```
summary(model)

# 3. Predict sales for a new store

new_store <- data.frame(
  Advertising = 28,
  StoreSize = 185
)

predicted_sales <- predict(model, newdata = new_store)
predicted_sales
```

## Solution

```
# 1. Create a custom dataset
sales_data <- data.frame(
  Sales = c(120, 150, 170, 200, 220, 250, 275, 300, 320, 340),
  Advertising = c(10, 15, 14, 20, 22, 25, 27, 30, 31, 33),
  StoreSize = c(100, 120, 130, 150, 160, 170, 180, 190, 200, 210)
)

# 2. Fit a multiple linear regression model

model <- lm(Sales ~ Advertising + StoreSize, data = sales_data)

# 3. Show model summary

summary(model)
```

Call:

```
lm(formula = Sales ~ Advertising + StoreSize, data = sales_data)
```

Residuals:

	Min	1Q	Median	3Q	Max
	-11.128	-3.776	1.789	5.059	10.146

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	-65.2644	38.6401	-1.689	0.1351

Advertising	2.7007	3.0055	0.899	0.3987
StoreSize	1.4811	0.6555	2.259	0.0584 .

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 8.091 on 7 degrees of freedom

Multiple R-squared: 0.9909, Adjusted R-squared: 0.9883

F-statistic: 381.6 on 2 and 7 DF, p-value: 7.157e-08

```
# 4. Predict sales for a new store
```

```
new_store <- data.frame(  
  Advertising = 28,  
  StoreSize = 185  
)
```

```
predicted_sales <- predict(model, newdata = new_store)  
predicted_sales
```

```
1  
284.3602
```