STATS 266 Handout - Linear Regression

Qi Wang

2025-03-01

Contents

1 Introduction

Welcome to **STATS 266: Introduction to R**. This handout provides an introduction about linear regression in R. Linear regression is a fundamental statistical technique used for modeling the relationship between a dependent variable and one or more independent variables. By the end of this document, you should be able to:

- understand the mathematical formulation of linear regression
- estimate regression coefficients using the least squares method
- implement a linear regression in R
- do model diagnostics and evaluation

For this part, valuable materials to refer to include https://www.geeksforgeeks.org/ml-linear-regression/ and https://malfaro2.github.io/stat266A/lectures/EDA+REG.html.

2 Mathematical Formulation

A simple linear regression model is defined as:

$$Y = \beta_0 + \beta_1 X + \epsilon \tag{1}$$

where:

- Y is the dependent variable (response)
- X is the independent variable (predictor)
- β_0 is the intercept
- β_1 is the slope (effect of X on Y)
- ϵ is the error term, assumed to be normally distributed: $\epsilon \sim N(0, \sigma^2)$

For multiple linear regression, we extend this to:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p + \epsilon \tag{2}$$

where there are p predictor variables.

In this setting, X and Y are observed data, the other $\beta_.$ are parameters to be estimate. We also need to estimate the σ^2 in the normal assumption of the ϵ.

3 Ordinary Least Squares (OLS)

Take simple linear regression as an example, where we only have intercept and one covariate. The **Ordinary Least Squares (OLS)** method estimates β by minimizing the sum of squared residuals:

$$S(\beta) = \sum_{i=1}^{n} (Y_i - \beta_0 - \beta_1 X_i)^2 \tag{3}$$

Remember the only unknown parameter is β, so the SSE, which can be understood as a loss function here, is a function about β. We need to find which value of β lead to the smallest SSE. But how can we find it?

3.1 Taking the Derivatives

3.2 Derivatives of β_0

$$\frac{\partial S}{\partial \beta_0} = \sum_{i=1}^{n} 2\left(Y_i - \beta_0 - \beta_1 X_i\right)(-1). \tag{4}$$

Setting this derivative to zero:

$$\sum_{i=1}^{n} (Y_i - \beta_0 - \beta_1 X_i) = 0. \tag{5}$$

Rearrange:

$$\sum_{i=1}^{n} Y_i = n\beta_0 + \beta_1 \sum_{i=1}^{n} X_i. \tag{6}$$

Dividing by n, we get:

$$\hat{\beta}_0 = \bar{Y} - \hat{\beta}_1 \bar{X}. \tag{7}$$

3.3 Derivatives of β_1

$$\frac{\partial S}{\partial \beta_1} = \sum_{i=1}^{n} 2\left(Y_i - \beta_0 - \beta_1 X_i\right)(-X_i). \tag{8}$$

Setting this derivative to zero:

$$\sum_{i=1}^{n} X_i (Y_i - \beta_0 - \beta_1 X_i) = 0. \tag{9}$$

Expanding:

$$\sum_{i=1}^{n} X_i Y_i - \beta_0 \sum_{i=1}^{n} X_i - \beta_1 \sum_{i=1}^{n} X_i^2 = 0. \tag{10}$$

Substituting $\hat{\beta}_0 = \bar{Y} - \hat{\beta}_1 \bar{X}$:

$$\sum_{i=1}^{n} X_i Y_i - (\bar{Y} - \hat{\beta}_1 \bar{X}) \sum_{i=1}^{n} X_i - \hat{\beta}_1 \sum_{i=1}^{n} X_i^2 = 0. \tag{11}$$

Simplifying:

$$\sum_{i=1}^{n} X_i Y_i - \bar{Y} \sum_{i=1}^{n} X_i + \hat{\beta}_1 \bar{X} \sum_{i=1}^{n} X_i - \hat{\beta}_1 \sum_{i=1}^{n} X_i^2 = 0. \tag{12}$$

Rearrange to solve for $\hat{\beta}_1$:

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n} (X_i - \bar{X})(Y_i - \bar{Y})}{\sum_{i=1}^{n} (X_i - \bar{X})^2}. \tag{13}$$

3.4 Other Cases

In a general case, for multiple regression, the model is:

$$Y = X\beta + \epsilon, \tag{14}$$

where:

- Y is an $n \times 1$ response vector,
- X is an $n \times (p+1)$ matrix including predictors and intercept column,
- β is a $(p+1) \times 1$ coefficient vector,
- ϵ is an $n \times 1$ error vector.

The expression of the regression coefficient in a linear regression is in a closed form. The OLS estimates are obtained by solving:

$$\hat{\beta} = (X^T X)^{-1} X^T Y \tag{15}$$

In this equation, X is a $n \times p$ matrix, where n is the number of observations, and p is the number of covariates (including intercept). The length n column vector Y is the corresponding response variable.

4 Implementing in R

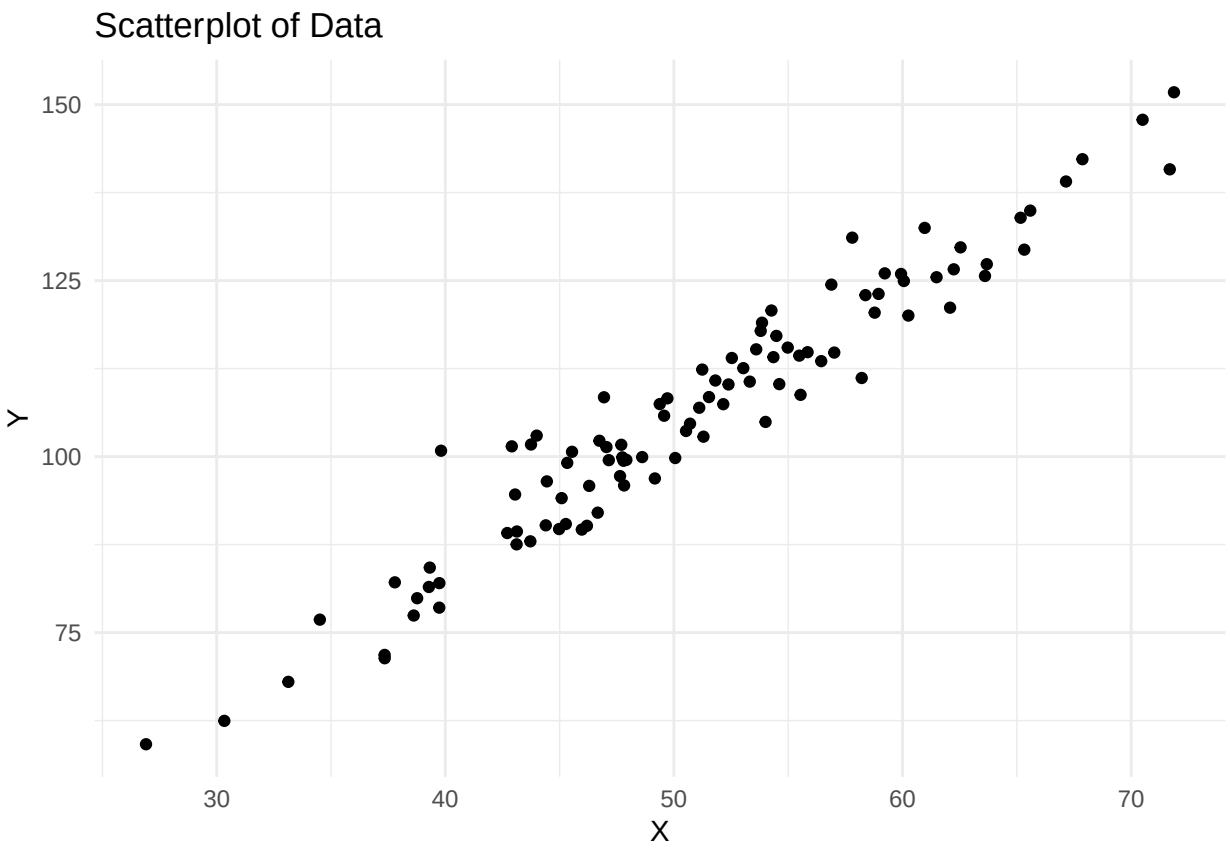
Let's fit a simple linear regression model using R.

4.1 Simulating the Data

```r
# Simulated dataset
set.seed(123)
x <- rnorm(100, mean = 50, sd = 10)  # Predictor
y <- 5 + 2*x + rnorm(100, sd = 5)  # Response with noise
data <- data.frame(x, y)

# Scatterplot of the data
library(ggplot2)
```

```r
ggplot(data, aes(x = x, y = y)) +
    geom_point() +
    labs(title = "Scatterplot of Data", x = "X", y = "Y") +
    theme_minimal()
```



4.2 Fitting a Linear Model

```r
# Fit a linear regression model
model <- lm(y ~ x, data = data)

# Summary of the model
summary(model)
```

```
##
## Call:
## lm(formula = y ~ x, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.5367 -3.4175 -0.4375  2.9032 16.4520
```

```
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5.79778    2.76324   2.098   0.0385 *
## x            1.97376    0.05344  36.935   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.854 on 98 degrees of freedom
## Multiple R-squared:  0.933,  Adjusted R-squared:  0.9323
## F-statistic:  1364 on 1 and 98 DF,  p-value: < 2.2e-16
```

The function `lm()` is for linear regression, $y \sim x$ means the response variable is y and the covariates are x.

4.3 Interpreting the Output

We see a lot information from summarizing the model:

The `summary(model)` function provides:

- Coefficients: Estimates of β_0 (Intercept) and β_1 (Slope)
- R-squared: Measure of model fit (closer to 1 means better fit)
- p-value: Tests if predictors significantly explain variation in Y

Given the estimated model:

$$\hat{Y} = \hat{\beta}_0 + \hat{\beta}_1 X$$

- Intercept interpretation: The **predicted** value of Y when $X = 0$.
- Slope interpretation: The **expected** change in Y for each additional unit of X.

5 Model Diagnostics

Model diagnostics help evaluate the **validity** of a linear regression model by checking key **assumptions**. Ensuring these assumptions hold improves the reliability of predictions and inferences. The primary diagnostics include linearity, homoscedasticity, normality, multicollinearity, and outliers. Each of these assumptions must be checked to ensure that the linear regression model provides valid and meaningful results. Let's talk about them one by one. To begin, we still use mtcars as an example, we use mpg to be the response variable, ht, and wt to be the covariates.

```r
# Load necessary libraries
library(ggplot2)
library(car)    # For VIF and diagnostics
```

```
## Loading required package: carData
```

```
library(dplyr) # For data manipulation
```

```
##
## Attaching package: 'dplyr'

## The following object is masked from 'package:car':
##
##     recode

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
data(mtcars)
model <- lm(mpg ~ hp + wt, data = mtcars)
summary(model)
```

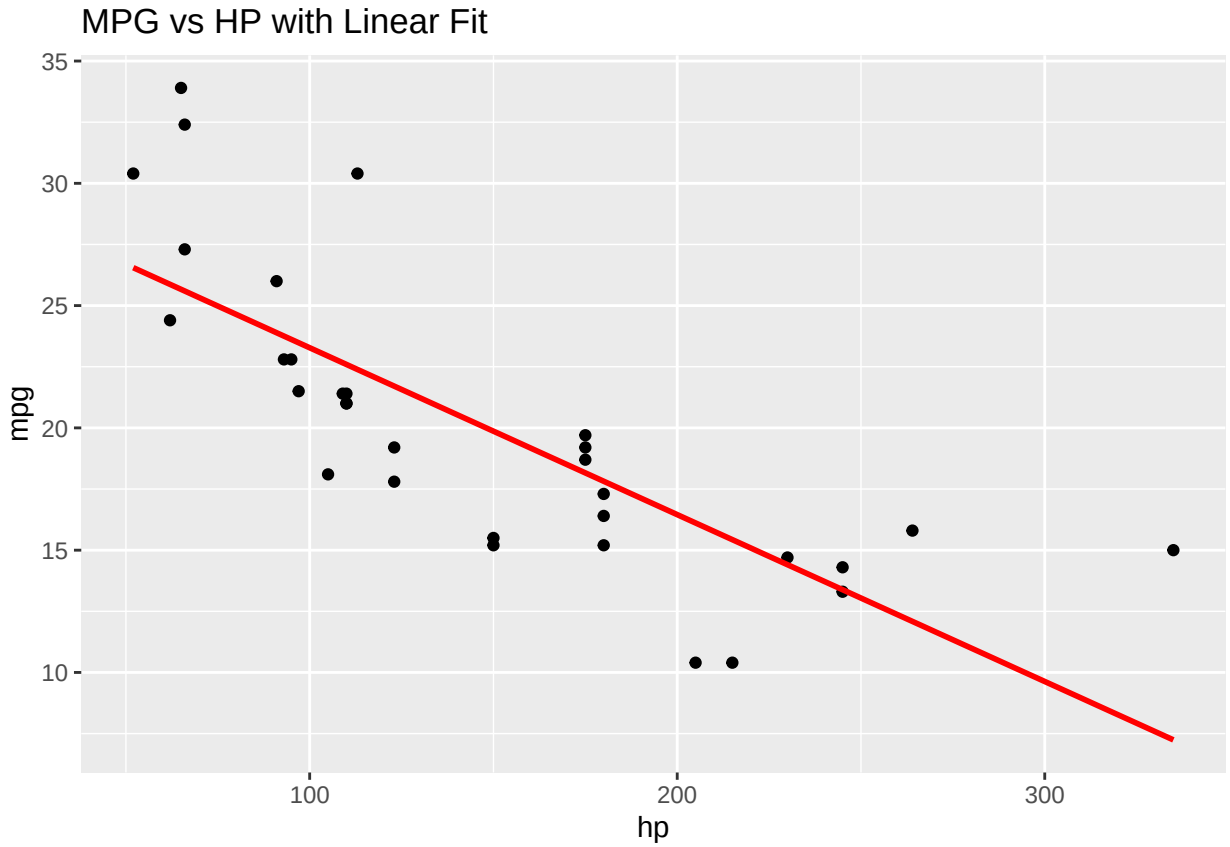
```
##
## Call:
## lm(formula = mpg ~ hp + wt, data = mtcars)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -3.941 -1.600 -0.182  1.050  5.854
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 37.22727    1.59879  23.285  < 2e-16 ***
## hp          -0.03177    0.00903  -3.519  0.00145 **
## wt          -3.87783    0.63273  -6.129 1.12e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.593 on 29 degrees of freedom
## Multiple R-squared:  0.8268, Adjusted R-squared:  0.8148
## F-statistic: 69.21 on 2 and 29 DF,  p-value: 9.109e-12
```

5.1 Linearity

The relationship between the dependent variable and independent variables should be linear. If the relationship is nonlinear, applying transformations (e.g., logarithmic or polynomial terms) might be necessary. In our example, the relationship between mpg and the predictors (hp, wt) should be linear. We use scatter plots to visualize the relationships.

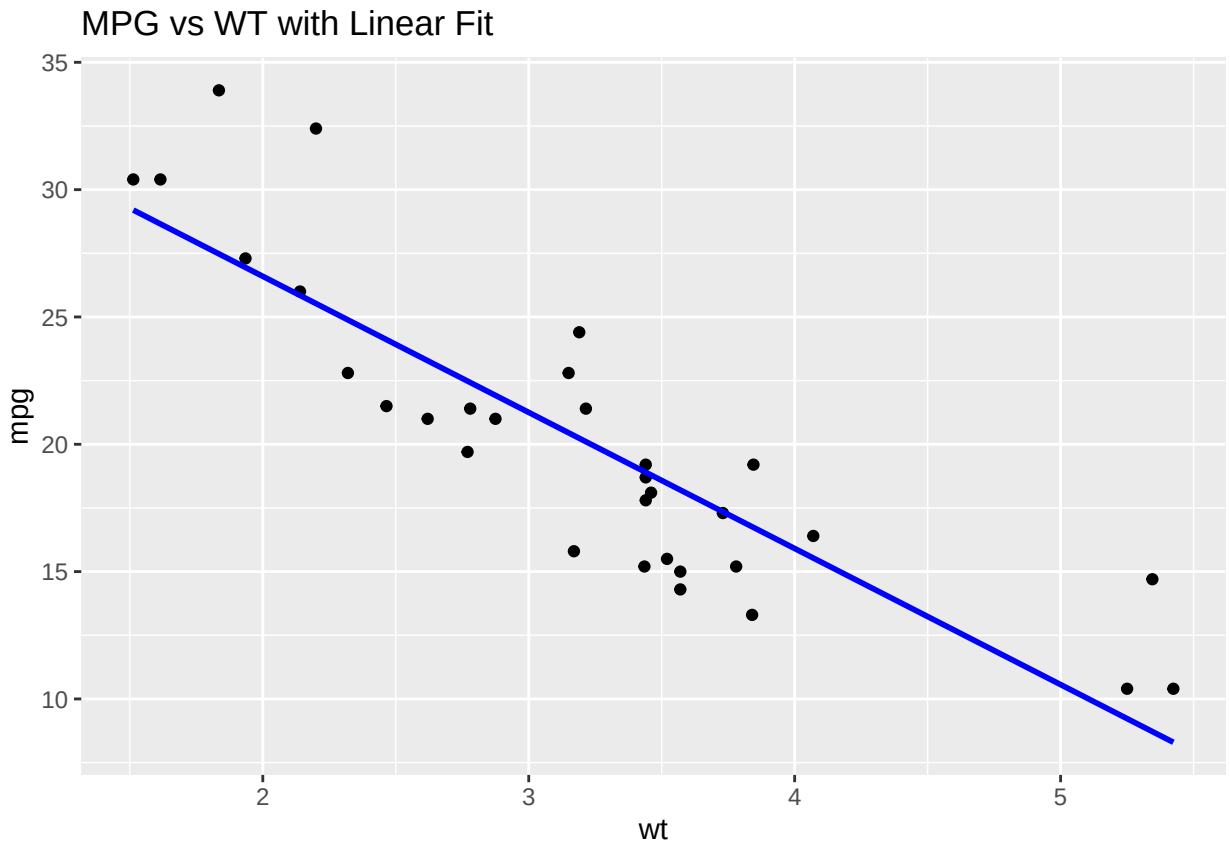
```
ggplot(mtcars, aes(x = hp, y = mpg)) +
  geom_point() +
  geom_smooth(method = "lm", color = "red", se = FALSE) +
  labs(title = "MPG vs HP with Linear Fit")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```




```
ggplot(mtcars, aes(x = wt, y = mpg)) +
  geom_point() +
  geom_smooth(method = "lm", color = "blue", se = FALSE) +
  labs(title = "MPG vs WT with Linear Fit")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



Note that the geom_smooth() function in ggplot2 is used to add a smoothed trend line to a plot, typically applied to scatter plots to visualize trends in the data. If the red or blue regression lines do not fit well, non-linearity may exist. If non-linearity is detected, transformations such as logarithm (log()), polynomial (poly()) terms, or splines may be needed.

5.2 Homoscedasticity

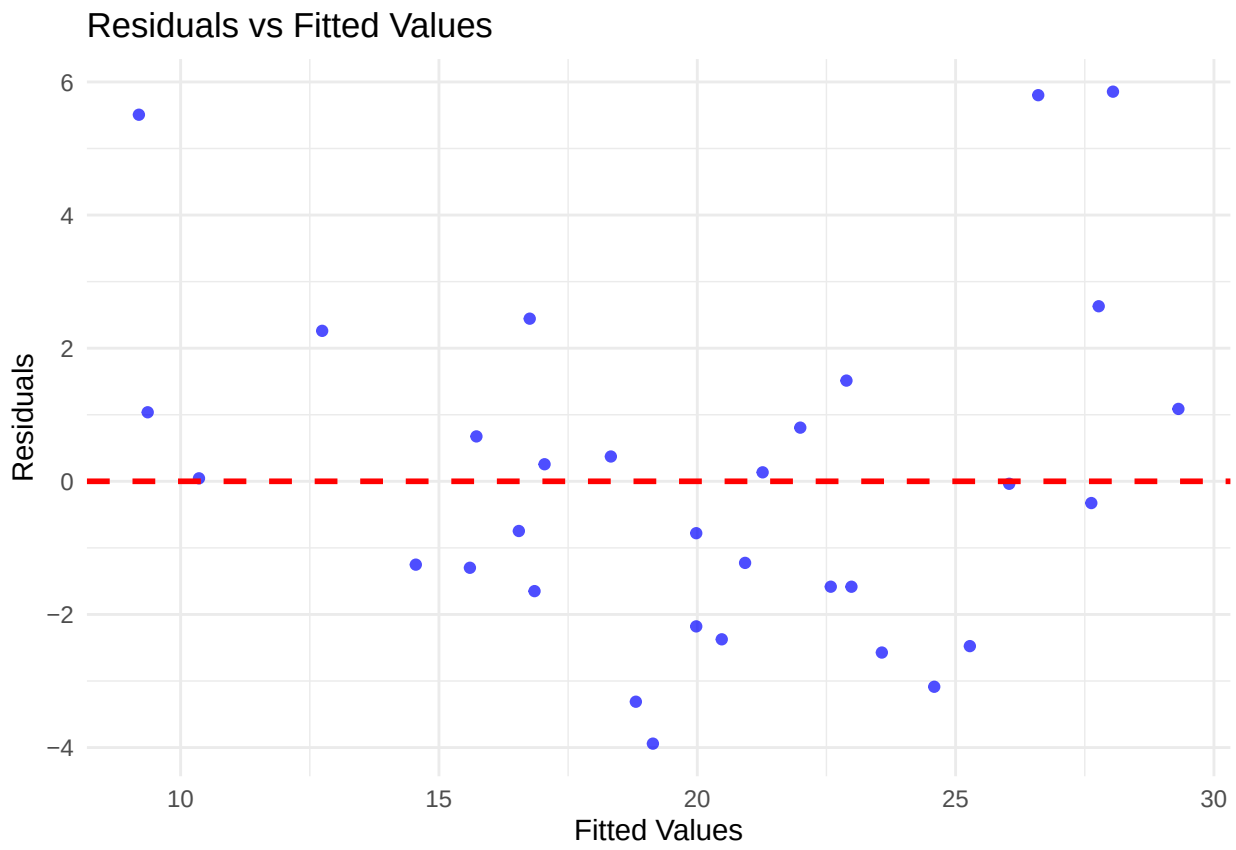
The variance of residuals should be constant across all levels of the independent variable. If residuals exhibit increasing or decreasing spread (heteroscedasticity), weighted regression or transformations can help. In other words, residuals should have constant variance across fitted values. A Residuals vs Fitted plot helps detect heteroscedasticity.

```
library(ggplot2)

# Extract fitted values and residuals
residuals_df <- data.frame(
  fitted = fitted(model),
  residuals = residuals(model)
)

# Create Residuals vs Fitted Plot
ggplot(residuals_df, aes(x = fitted, y = residuals)) +
  geom_point(color = "blue", alpha = 0.7) +  # Scatterplot of residuals
```

```r
  geom_hline(yintercept = 0, linetype = "dashed", color = "red", size = 1) +  # Reference line
  labs(
    title = "Residuals vs Fitted Values",
    x = "Fitted Values",
    y = "Residuals"
  ) +
  theme_minimal()
```



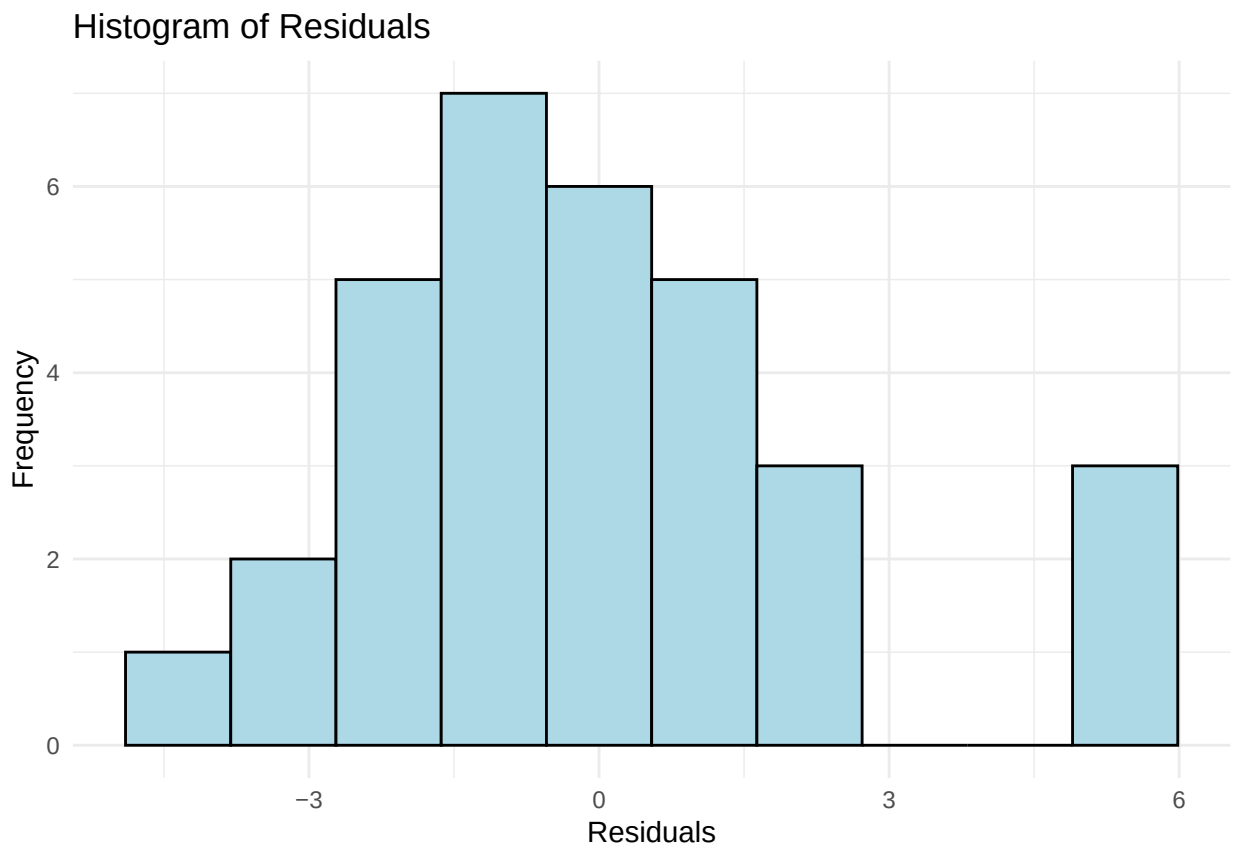
The residuals should be randomly scattered without patterns. If residuals fan out (increasing variance), heteroscedasticity is present. Possible solutions include applying log transformation to mpg (e.g., log(mpg) ~ hp + wt), or using weighted least squares (https://en.wikipedia.org/wiki/Weighted_least_squares) regression.

5.3 Normality of Residuals

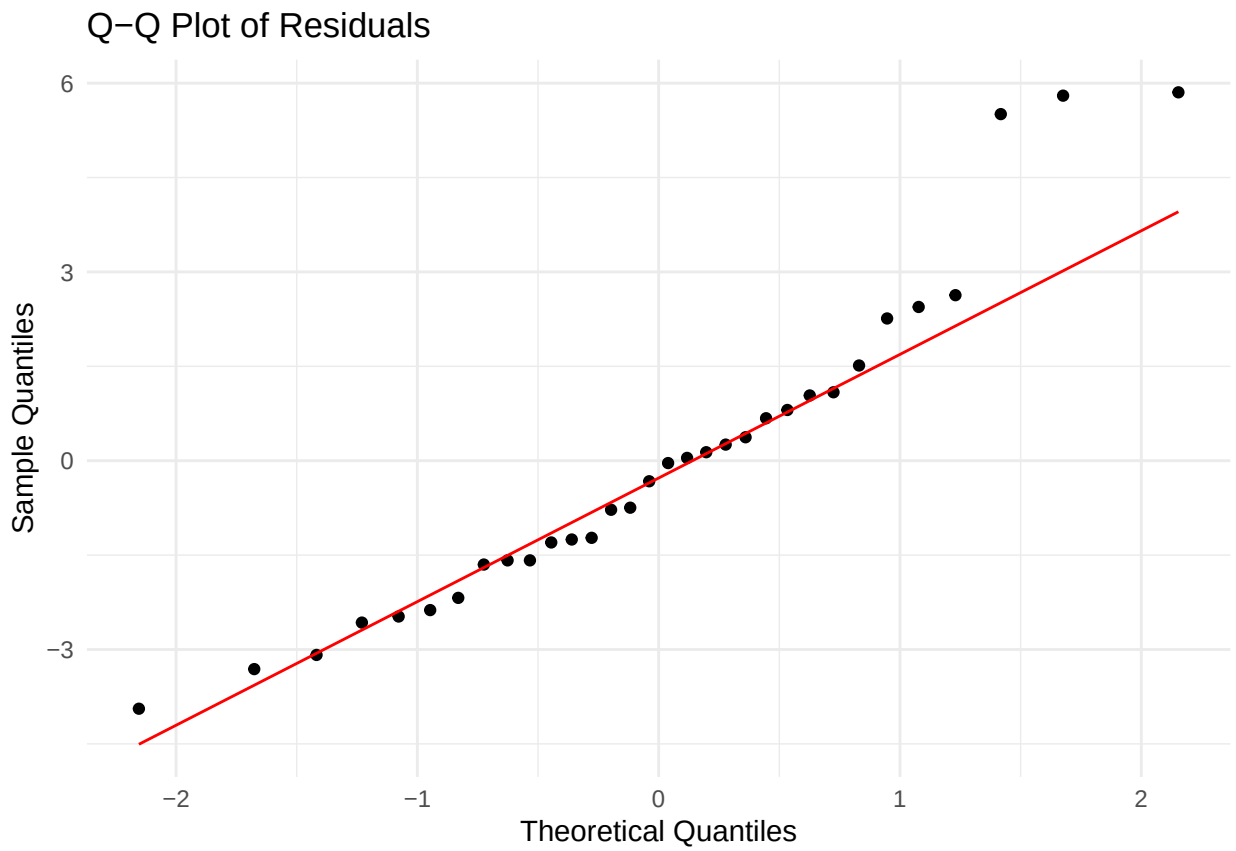
The residuals should be normally distributed to ensure valid hypothesis testing. This can be checked with a Q-Q plot or histogram. If residuals are not normal, using robust regression or bootstrapping may be useful. We can use either histogram or Q-Q plot:

```r
# Compute residuals
residuals_df <- data.frame(residuals = residuals(model))
```

```r
# Histogram of residuals using ggplot2
ggplot(residuals_df, aes(x = residuals)) +
  geom_histogram(fill = "lightblue", color = "black", bins = 10) +
  labs(title = "Histogram of Residuals", x = "Residuals", y = "Frequency") +
  theme_minimal()
```




```r
# Q-Q plot using ggplot2
ggplot(residuals_df, aes(sample = residuals)) +
  stat_qq() +
  stat_qq_line(color = "red") +
  labs(title = "Q-Q Plot of Residuals", x = "Theoretical Quantiles", y = "Sample Quantiles") +
  theme_minimal()
```



If residuals deviate strongly from the red line in the Q-Q plot, they may not be normally distributed. If the histogram is skewed, transformations like log() or sqrt() may help.

5.4 Multicollinearity

Predictor variables should not be highly correlated, as multicollinearity can distort coefficient estimates and make interpretation difficult. The Variance Inflation Factor (VIF) is commonly used to detect multicollinearity, and variables with a VIF greater than 5 should be reconsidered.

```
# Compute Variance Inflation Factor (VIF)
vif_values <- vif(model)
vif_values
```

```
##       hp       wt
## 1.766625 1.766625
```

Interpretation of VIF: General guidelines for interpreting VIF values:

- VIF < 5: No serious multicollinearity.
- VIF > 5: Multicollinearity is concerning.
- VIF > 10: Severe multicollinearity—consider removing or combining variables.

If VIF > 5, multicollinearity may be problematic, affecting the interpretability and stability of regression coefficients. If multicollinearity is present, consider these approaches:

- Remove one of the correlated predictors if they provide redundant information.
- Use Principal Component Regression (PCR) to transform predictors into uncorrelated components.
- Combine highly correlated predictors (e.g., averaging two related variables).

5.5 Outliers and Influential Points

Extreme values can disproportionately impact the regression model. Cook's Distance and leverage statistics help identify these points, and they may need to be investigated or removed based on domain knowledge. Cook's Distance for an observation i is defined as:

$$D_i = \frac{\sum_{j=1}^{n} (\hat{Y}_{j(i)} - \hat{Y}_j)^2}{p \cdot MSE}$$

where:

- $\hat{Y}_j$ is the predicted value for observation j.
- $\hat{Y}_{j(i)}$ is the predicted value when the i-th observation is removed.
- p is the number of predictors.
- MSE is the mean squared error of the model.

An observation is considered **influential** if its **Cook's Distance is greater than**:

$$\frac{4}{n - p - 2}$$

where:

- n is the number of observations.
- p is the number of predictors.

6 Ackowledgement

This teaching material is adapted from the previous material of this course made by Marcela Alfaro-Córdoba and Sheng Jiang.