Final-Project.R

besti

2024-12-15

```r
# Loading the data file
file_path <- "C:\\Users\\besti\\OneDrive\\Desktop\\Sarah Schenirer\\Intro Data Science\\auto-mpg.csv"
auto_data = read.csv(file_path)

# Check the structure of the data
str(auto_data)
```

```
## 'data.frame':    398 obs. of  9 variables:
##  $ mpg         : num  18 15 18 16 17 15 14 14 14 15 ...
##  $ cylinder    : int  8 8 8 8 8 8 8 8 8 8 ...
##  $ displacement: num  307 350 318 304 302 429 454 440 455 390 ...
##  $ horsepower  : chr  "130" "165" "150" "150" ...
##  $ weight      : int  3504 3693 3436 3433 3449 4341 4354 4312 4425 3850 ...
##  $ acceleration: num  12 11.5 11 12 10.5 10 9 8.5 10 8.5 ...
##  $ model.year  : int  70 70 70 70 70 70 70 70 70 70 ...
##  $ origin      : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ car.name    : chr  "chevrolet chevelle malibu" "buick skylark 320" "plymouth satellite" "amc rebel
```

```r
# Change horsepower from chr to num
auto_data$horsepower <- as.numeric(as.character(auto_data$horsepower))
```

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

```r
# Split the data into train/test
train <- auto_data[1:300, ]

test <- auto_data[301:398, ]

# Rearrange the sequence of the test data to start from 1 instead of 301
rownames(test) <- seq(length=nrow(test))

# Simple Linear Regression

# using train data
# weight as independent variable
simple_model <- lm(train$mpg ~ train$weight, data=train)
summary(simple_model)
```

```
##
## Call:
```

```
## lm(formula = train$mpg ~ train$weight, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.1077 -1.8842 -0.0333  1.7275 15.1232
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  40.3879027  0.6368804   63.41   <2e-16 ***
## train$weight -0.0062524  0.0001957  -31.96   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.992 on 298 degrees of freedom
## Multiple R-squared:  0.7741, Adjusted R-squared:  0.7733
## F-statistic:  1021 on 1 and 298 DF,  p-value: < 2.2e-16
```

```
b0_1 = simple_model$coefficients[1]
b1_1 = simple_model$coefficients[2]

# Multiple R-squared: 0.7741
# Adjusted R-squared: 0.7733
# Linear Regression Equation: y = 40.3879027 + -0.0062524 * weight

# horsepower as independent variable
simple_model2 <- lm(train$mpg ~ train$horsepower, data=train)
summary(simple_model2)
```

```
##
## Call:
## lm(formula = train$mpg ~ train$horsepower, data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.7872  -2.7817  -0.3246   2.4726  14.3103
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      34.794687   0.647855   53.71   <2e-16 ***
## train$horsepower -0.125105   0.005444  -22.98   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.783 on 296 degrees of freedom
##   (2 observations deleted due to missingness)
## Multiple R-squared:  0.6408, Adjusted R-squared:  0.6396
## F-statistic: 528.1 on 1 and 296 DF,  p-value: < 2.2e-16
```

```
b0_2 = simple_model2$coefficients[1]
b1_2 = simple_model2$coefficients[2]

# Multiple R-squared: 0.6408
# Adjusted R-squared: 0.6396
```

```r
# Linear Regression Equation: y = 34.794687 + -0.125105 * horsepower


# Multiple Linear Regression

# using train data
# weight, horsepower, displacement as independent variables
multiple_model <- lm(train$mpg ~ train$weight + train$horsepower + train$displacement, data=train)
summary(multiple_model)
```

```
##
## Call:
## lm(formula = train$mpg ~ train$weight + train$horsepower + train$displacement,
##     data = train)
##
## 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 ***
## train$weight        -0.0047898  0.0005328  -8.991   <2e-16 ***
## train$horsepower    -0.0205727  0.0096748  -2.126   0.0343 *
## train$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
##   (2 observations deleted due to missingness)
## 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
b0 <- multiple_model$coefficients[1]
b1 <- multiple_model$coefficients[2]
b2 <- multiple_model$coefficients[3]
b3 <- multiple_model$coefficients[4]

# Multiple R-squared: 0.783
# Adjusted R-squared: 0.7808
# Linear Regression Equation: y = 39.3739544 + -0.0047898 * weight + -0.0205727 * horsepower + -0.00584

# weight, horsepower as independent variables
# removed displacement because not statistically significant
multiple_model2 <- lm(train$mpg ~ train$weight + train$horsepower, data=train)
summary(multiple_model2)
```

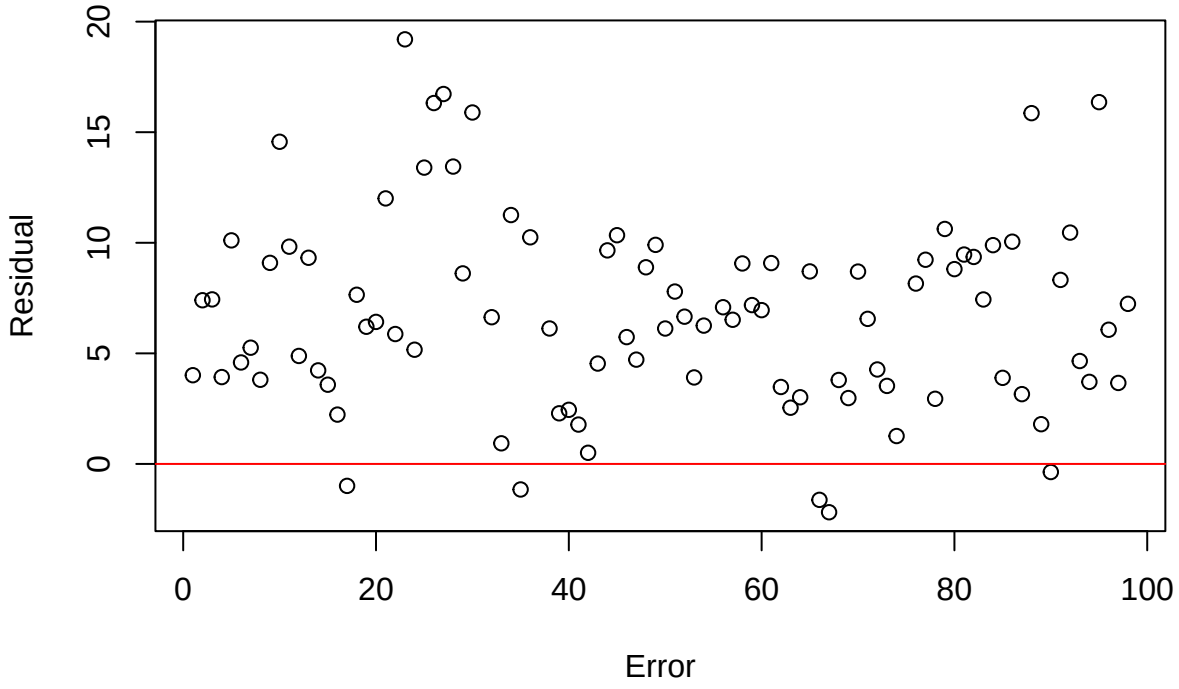
```
##
## Call:
## lm(formula = train$mpg ~ train$weight + train$horsepower, data = train)
##
## Residuals:
```

```
##     Min      1Q  Median      3Q     Max
## -8.6676 -1.8747  0.0104  1.6777 14.5954
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)      40.1577216  0.6373491   63.01  < 2e-16 ***
## train$weight     -0.0052317  0.0003785  -13.82  < 2e-16 ***
## train$horsepower -0.0264219  0.0083087   -3.18  0.00163 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.952 on 295 degrees of freedom
##   (2 observations deleted due to missingness)
## Multiple R-squared:  0.782,  Adjusted R-squared:  0.7805
## F-statistic: 529.1 on 2 and 295 DF,  p-value: < 2.2e-16
```

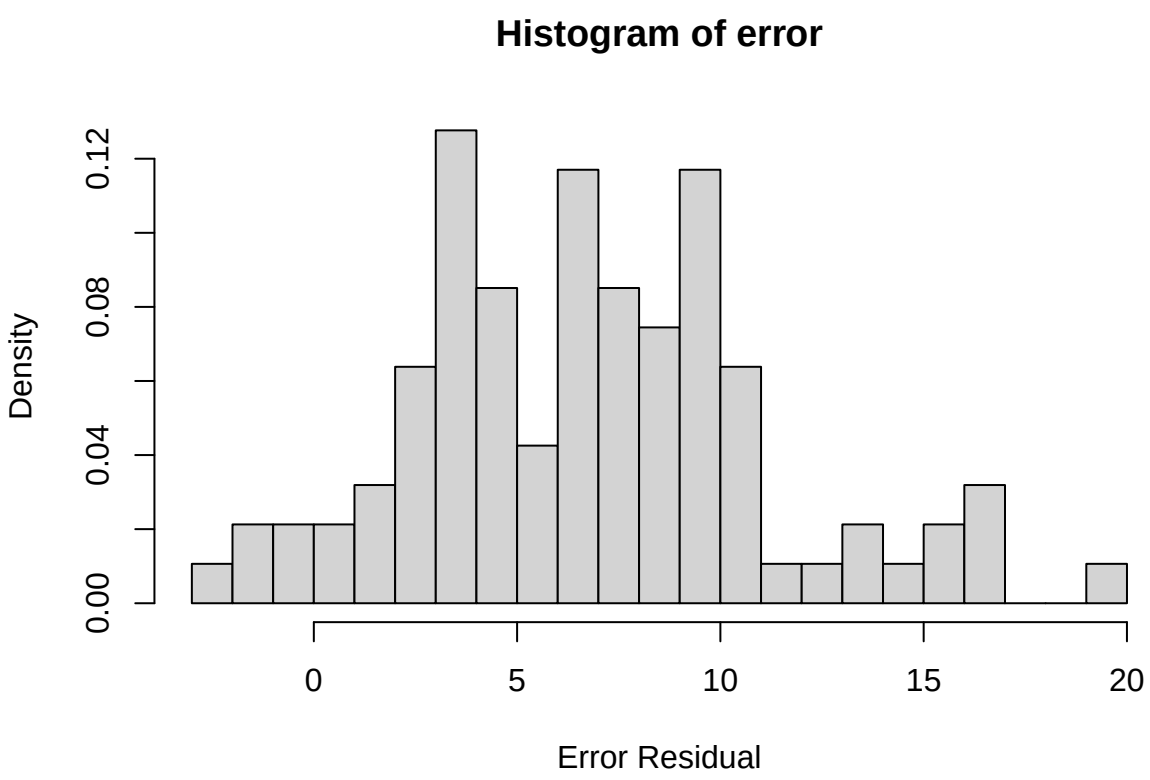
```r
B0 <- multiple_model2$coefficients[1]
B1 <- multiple_model2$coefficients[2]
B2 <- multiple_model2$coefficients[3]

# Multiple R-squared: 0.782
# Adjusted R-squared: 0.7805
# Linear Regression Equation: y = 40.1577216 + -0.0052317 * weight + -0.0264219 * horsepower

# Using multiple_model2 on test data to predict mpg
y_pred <- B0 + B1*test$weight + B2*test$horsepower
# comparing to actual mpg
y_actual <- test[, 1]
error <- y_actual - y_pred
# Residual Plot
plot(error, xlab="Error", ylab="Residual")
abline(0,0 ,col='red')
```




```
# Histogram
hist(error, prob=T, breaks=20, xlab="Error Residual", ylab="Density")
```


Histogram of error
Density
0.00
0.04
0.08
0.12
0
5
10
15
20
Error Residual