HW8

113078506

2025-04-09

- Gen AI Usage: I use Gen AI to refine my grammar, verify my reasoning and adjust to cleaner code.
- Students who helped: 113078505, 113078502, and 113078514, helped with conclusion reasoning.

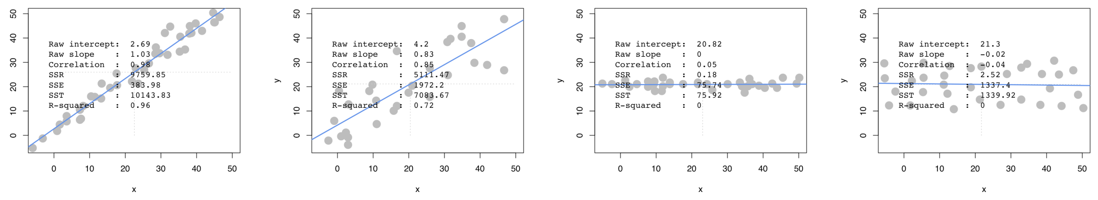
Question 1 We will use the interactive_regression() function from CompStatsLib again.

Scenario 1: Consider a very narrowly dispersed set of points that have a negative or positive steep slope

Scenario 2: Consider a widely dispersed set of points that have a negative or positive steep slope

Scenario 3: Consider a very narrowly dispersed set of points that have a negative or positive shallow slope

Scenario 4: Consider a widely dispersed set of points that have a negative or positive shallow slope



```
Raw intercept:  2.69
Raw slope    :  1.03
Correlation  :  0.98
SSR          :  9759.85
SSE          :  383.98
SST          :  10143.83
R-squared    :  0.96
```

```
Raw intercept:  4.2
Raw slope    :  0.83
Correlation  :  0.85
SSR          :  5111.47
SSE          :  1972.2
SST          :  7083.67
R-squared    :  0.72
```

```
Raw intercept:  20.82
Raw slope    :  0
Correlation  :  0.05
SSR          :  0.18
SSE          :  75.74
SST          :  75.92
R-squared    :  0
```

```
Raw intercept:  21.3
Raw slope    :  -0.02
Correlation  :  -0.04
SSR          :  2.52
SSE          :  1337.4
SST          :  1339.92
R-squared    :  0
```

a. Comparing scenarios 1 and 2, which do we expect to have a stronger R2 ?

- I expect scenario 1 to have a stronger R^2, because the data points in scenario 1 are more tightly clustered around the regression line, indicating a better fit.

b. Comparing scenarios 3 and 4, which do we expect to have a stronger R2 ?

- Similar to the previous comparison, Scenario 3 is expected to have a slightly stronger R^2 than Scenario 4 due to the tighter clustering of data points around the regression line. However, in this case, both scenarios have slopes close to zero, which leads to R^2 values that are nearly zero for both.

c. Comparing scenarios 1 and 2, which do we expect has bigger/smaller SSE, SSR, and SST? (intuitively)

- Comparing Scenario 1 and Scenario 2, we intuitively expect SSE to be smaller in Scenario 1 because the points are closer to the regression line, while Scenario 2 shows more scattered data. SSR is larger in Scenario 1, as the model explains more of the variation. SST is also larger in Scenario 1 because the overall spread of the data (y-value) is wider.

d. Comparing scenarios 3 and 4, which do we expect has bigger/smaller SSE, SSR, and SST? (intuitively)

- Intuitively I will expect Scenario 4 has larger SSE and SST due to the wider spread of the data. Scenario 3 will have a slightly higher SSR since its data shows a more consistent pattern, though both are close to zero.

Question 2

a. Use the lm() function to estimate the regression model Salary ~ Experience + Score + Degree. Show the beta coefficients, R2, and the first 5 values of y ($fitted.values)and($**residuals)**

```
pgms_salaries <- read.csv("programmer_salaries.txt", sep="\t")
model <- lm(Salary ~ Experience + Score + Degree, data = pgms_salaries)

# beta coefficients
cat("beta coefficients: ", coef(model), "\n")
```

```
## beta coefficients:  7.944849 1.147582 0.196937 2.280424
```

```
# R-squared
cat("R-squared: ",summary(model)$r.squared, "\n")
```

```
## R-squared:  0.8467961
```

```
# fitted values: y_hat
cat("fitted values top-5: ",head(fitted(model), 5), "\n")
```

```
## fitted values top-5:  27.89626 37.95204 26.02901 32.11201 36.34251
```

```
# residuals: y - y_hat
cat("residuals top-5: ",head(residuals(model), 5), "\n")
```

```
## residuals top-5:  -3.896261 5.047957 -2.329011 2.187986 -0.5425072
```

b. Use only linear algebra and the geometric view of regression to estimate the regression yourself:

```r
# i. Create an X matrix that has a first column of 1s followed by columns of the independent variables
X <- as.matrix(cbind(1, pgms_salaries$Experience, pgms_salaries$Score, pgms_salaries$Degree))

# ii. Create a y vector with the Salary values (only show the code)
y <- as.matrix(pgms_salaries$Salary)

# iii. Compute the beta_hat vector of estimated regression coefficients (show the code and values)
beta_hat <- solve(t(X) %*% X) %*% t(X) %*% y
cat("beta_hat:",beta_hat,"\n")
```

```
## beta_hat: 7.944849 1.147582 0.196937 2.280424
```

```r
# iv. Using the above, compute a y_hat vector of estimated y values, and a res vector of residuals
y_hat <- X %*% beta_hat
cat("head of y_hat: ", head(y_hat,5),"\n")
```

```
## head of y_hat:  27.89626 37.95204 26.02901 32.11201 36.34251
```

```r
res <- y-y_hat
cat("head of res: ", head(res,5),"\n")
```

```
## head of res:  -3.896261 5.047957 -2.329011 2.187986 -0.5425072
```

```r
# v. Using only the results from (i) - (iv), compute SSR, SSE and SST (show the code and values)
SSR <- sum((y_hat - mean(y))^2)
SSE <- sum((y_hat - y)^2)
SST <- sum((y - mean(y))^2)

cat(" SSR:",SSR,"\n", "SSE:",SSE,"\n","SST:",SST,"\n")
```

```
##  SSR: 507.896
##  SSE: 91.88949
##  SST: 599.7855
```

c. Compute R2 for in two ways, and confirm you get the same results (show code and values):

```r
# i. Use any combination of SSR, SSE, and SST
R2_i <- SSR / SST
cat("R2_i: ", R2_i,"\n")
```

```
## R2_i:  0.8467961
```

```r
# ii. Use the squared correlation of vectors y and y_hat
R2_ii <- cor(y,y_hat)^2
cat("R2_ii: ", R2_ii, "\n")
```

```
## R2_ii:  0.8467961
```

Question 3

```r
auto <- read.table("auto-data.txt", header=FALSE, na.strings = "?")
names(auto) <- c("mpg", "cylinders", "displacement", "horsepower", "weight",
                 "acceleration", "model_year", "origin", "car_name")
```

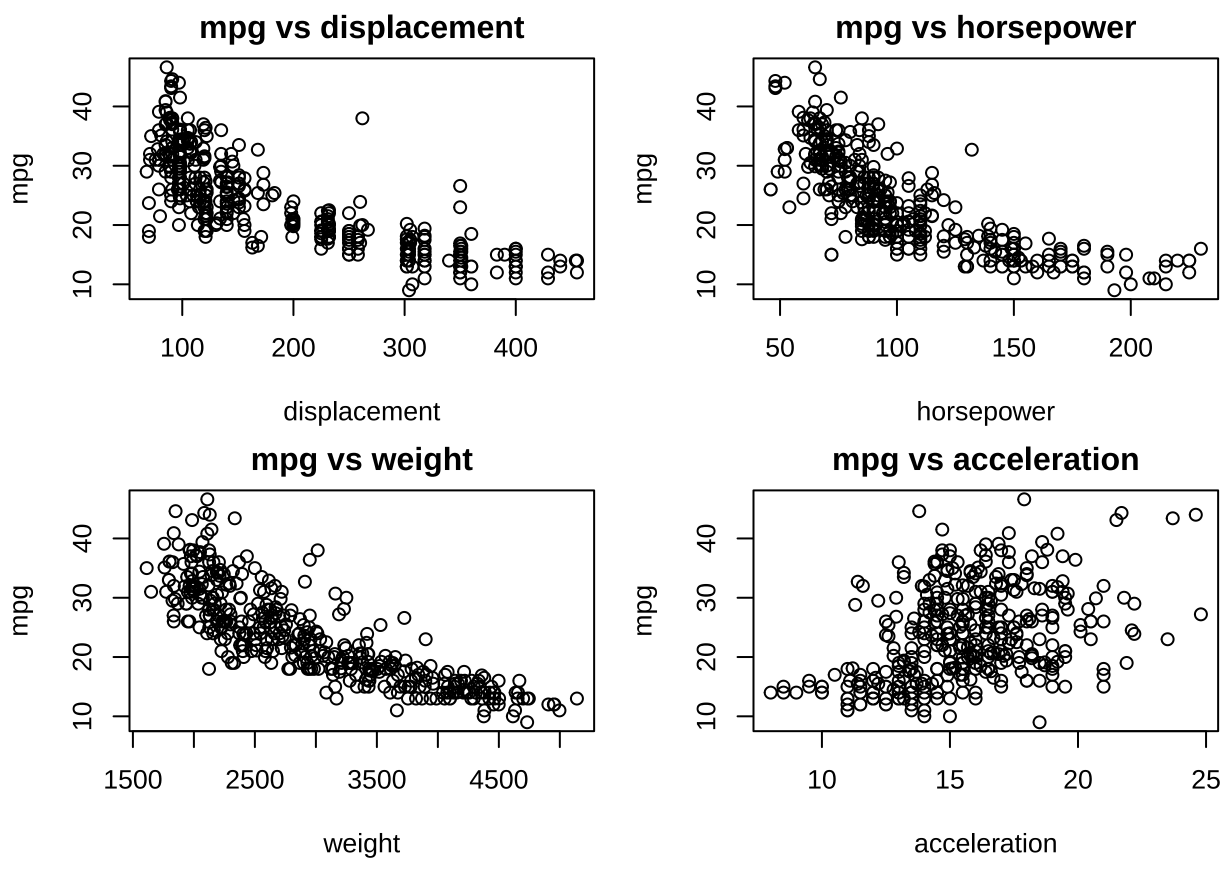
a. Let's first try exploring this data and problem

i. Visualize the data as you wish to answer the following questions (report only relevant plots)

- 1. We want to find out the variables which affect the dependent variable (mpg), so we plot the scatter plot of mpg as y-axis and other "numeric" variables as x-axis.

```r
vars <- c("displacement", "horsepower", "weight", "acceleration")

par(mfrow = c(2, 2), mar = c(4, 4, 2, 1))
for (v in vars) {
  plot(auto[[v]], auto$mpg, xlab = v, ylab = "mpg", main = paste("mpg vs", v))
}
```



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

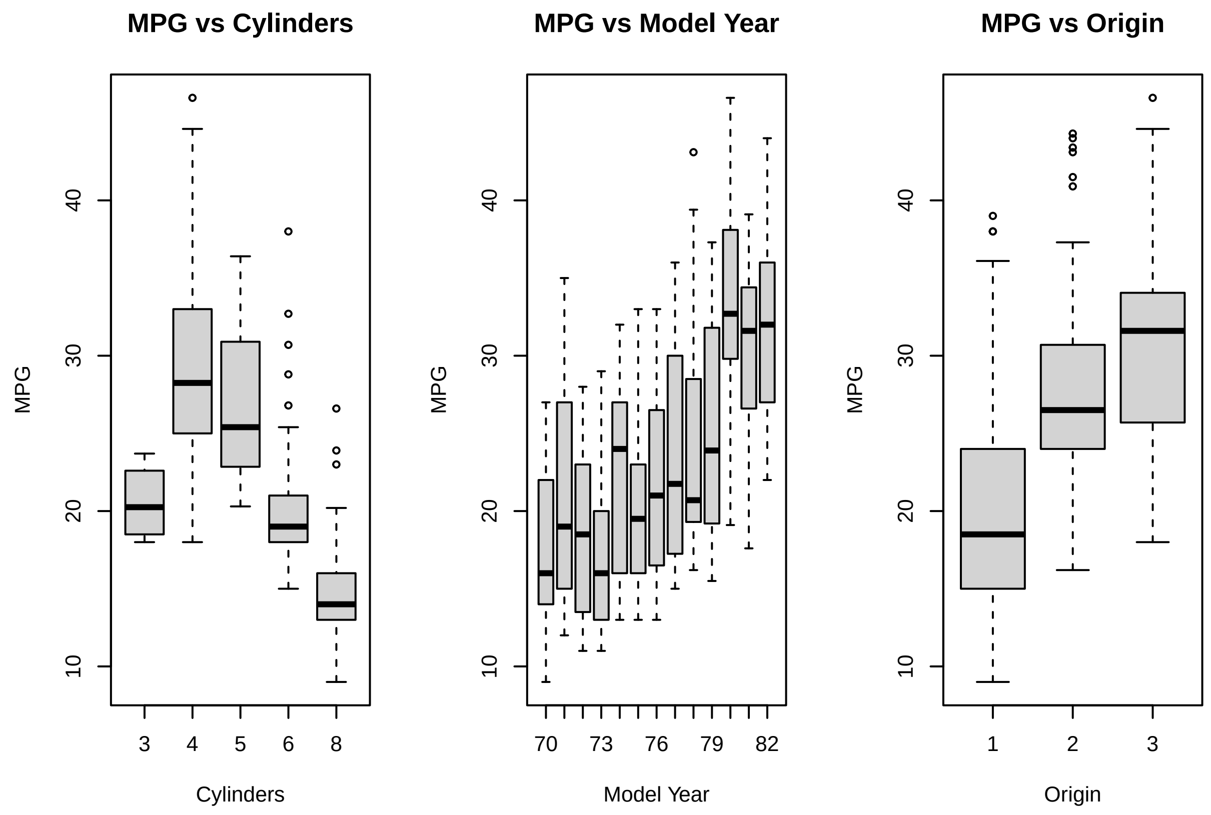
- 2. For categorical (discrete values) variables, we draw box plots to better understand if different categories matter. Here we exclude the "car_name" column as it serves more as an identifier than a meaningful predictor.

```r
par(mfrow = c(1, 3))

boxplot(mpg ~ cylinders, data = auto,
        main = "MPG vs Cylinders", xlab = "Cylinders", ylab = "MPG")

boxplot(mpg ~ model_year, data = auto,
        main = "MPG vs Model Year", xlab = "Model Year", ylab = "MPG")

boxplot(mpg ~ origin, data = auto,
        main = "MPG vs Origin", xlab = "Origin", ylab = "MPG")
```



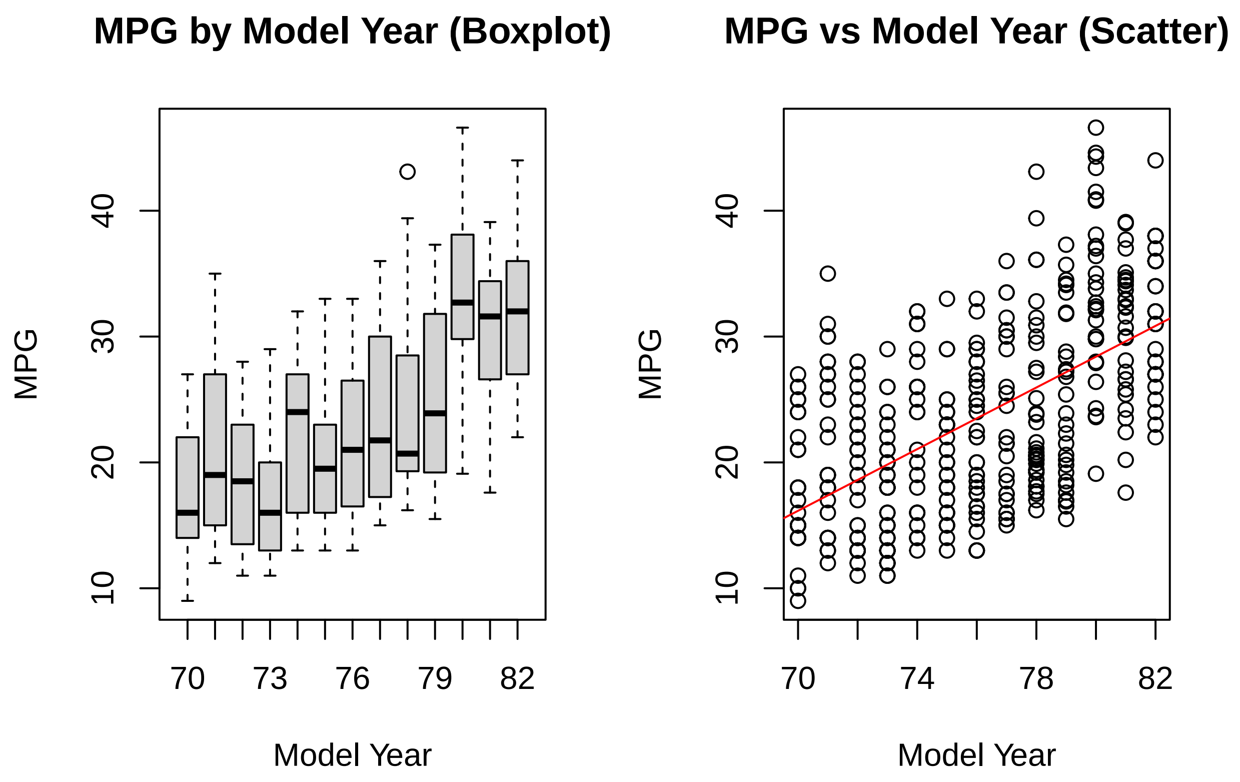
```r
par(mfrow = c(1, 1))
```

- Initially, we treated model_year as a categorical variable and used a box plot to compare average MPG across years. However, since model_year represents a time-based variable, we also considered it as a numeric variable and used a scatter plot to reveal an increasing trend in fuel efficiency over time.

```r
par(mfrow = c(1, 2))

boxplot(mpg ~ factor(model_year), data = auto,
        main = "MPG by Model Year (Boxplot)",
        xlab = "Model Year", ylab = "MPG")

plot(auto$model_year, auto$mpg,
     main = "MPG vs Model Year (Scatter)",
     xlab = "Model Year", ylab = "MPG")
abline(lm(mpg ~ model_year, data = auto), col = "red")
```



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

ii. Report a correlation table of all variables, rounding to two decimal places

```
cor_table <- cor(auto[, 1:8], use = "pairwise.complete.obs")
round(cor_table, 2)
```

```
##                mpg cylinders displacement horsepower weight acceleration
## mpg           1.00     -0.78        -0.80      -0.78  -0.83         0.42
## cylinders    -0.78      1.00         0.95       0.84   0.90        -0.51
## displacement -0.80      0.95         1.00       0.90   0.93        -0.54
## horsepower   -0.78      0.84         0.90       1.00   0.86        -0.69
## weight       -0.83      0.90         0.93       0.86   1.00        -0.42
## acceleration  0.42     -0.51        -0.54      -0.69  -0.42         1.00
## model_year    0.58     -0.35         -0.37      -0.42  -0.31         0.29
## origin        0.56     -0.56         -0.61      -0.46  -0.58         0.21
##              model_year origin
## mpg               0.58   0.56
## cylinders        -0.35  -0.56
## displacement     -0.37  -0.61
## horsepower       -0.42  -0.46
## weight           -0.31  -0.58
## acceleration      0.29   0.21
## model_year        1.00   0.18
## origin            0.18   1.00
```

iii. From the visualizations and correlations, which variables appear to relate to mpg?

- From the scatter plots, "displacement", "horsepower" , and "weight" show obvious patterns, meanwhile in box plots, all 3 variables seem have different mean of each category.
- From the correlation table, we only consider numeric variables since categorical variables are not suitable for correlation calculation. Consider the threshold of "strong relation" to |0.6|, "displacement" (-0.8), "horsepower"(-0.78), "weight"(-0.83) are having strong negative correlation to "mpg", although "model_year" has a positive correlation 0.58, it's not strongly correlated.
- In conclusion, I will consider numeric variables: "displacement", "horsepower", and "weight", categorical variables: "Cylinders" and "origin" are related to "mpg".

iv. Which relationships might not be linear? (don't worry about linearity for rest of this HW)

- From the scatter plots, some relationships appear potentially nonlinear. For example, the relationship between "acceleration" and mpg does not show a clear linear trend—it looks more flat or curved. In addition, "origin" and "cylinders" are categorical variables, so their relationship with mpg is not expected to be linear in nature.

v. Are there any pairs of independent variables that are highly correlated (r > 0.7)?

- First we draw a correlation table of all numerical independent variables.

```
auto_num <- auto[, c("displacement", "horsepower", "weight",
                     "acceleration", "model_year")]

cor_table <- cor(auto_num, use = "pairwise.complete.obs")

round(cor_table, 2)
```

```
##              displacement horsepower weight acceleration model_year
## displacement         1.00       0.90   0.93        -0.54      -0.37
## horsepower           0.90       1.00   0.86        -0.69      -0.42
## weight               0.93       0.86   1.00        -0.42      -0.31
## acceleration        -0.54      -0.69  -0.42         1.00       0.29
## model_year          -0.37      -0.42  -0.31         0.29       1.00
```

- Pairs that have strong correlation (define by surpass the threshold of |0.7|): "displacement" and "horsepower" (0.9), "displacement" and "weight" (0.93), "horsepower" and "weight" (0.86).

b. Let's create a linear regression model where mpg is dependent upon all other suitable variables

```
auto$origin <- factor(auto$origin)
model <- lm(mpg ~ cylinders + displacement + horsepower + weight + acceleration + model_year + origin, 
summary(model)
```

```
##
## Call:
## lm(formula = mpg ~ cylinders + displacement + horsepower + weight +
##     acceleration + model_year + origin, data = auto)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.0095 -2.0785 -0.0982  1.9856 13.3608
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -1.795e+01  4.677e+00  -3.839 0.000145 ***
## cylinders    -4.897e-01  3.212e-01  -1.524 0.128215
## displacement  2.398e-02  7.653e-03   3.133 0.001863 **
## horsepower   -1.818e-02  1.371e-02  -1.326 0.185488
## weight       -6.710e-03  6.551e-04 -10.243  < 2e-16 ***
## acceleration  7.910e-02  9.822e-02   0.805 0.421101
## model_year    7.770e-01  5.178e-02  15.005  < 2e-16 ***
## origin2       2.630e+00  5.664e-01   4.643 4.72e-06 ***
## origin3       2.853e+00  5.527e-01   5.162 3.93e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.307 on 383 degrees of freedom
##   (6 observations deleted due to missingness)
## Multiple R-squared:  0.8242, Adjusted R-squared:  0.8205
## F-statistic: 224.5 on 8 and 383 DF,  p-value: < 2.2e-16
```

i. Which independent variables have a 'significant' relationship with mpg at 1% significance?

- Based on the regression output, the variables that are significant at the 1% level ($p < 0.01$) are: "displacement", "weight", "model_year", "origin2", "origin3".

ii. Looking at the coefficients, is it possible to determine which independent variables are the most effective at increasing mpg? If so, which ones, and if not, why not? (hint: units!)

- We cannot directly compare the raw coefficients to determine which variable is most effective at increasing mpg, because the variables are measured in different units. For example, a one-unit increase in weight (in pounds) is not comparable to a one-unit increase in model_year. To properly compare effectiveness, we would need to standardize the variables or look at standardized coefficients.

c. Let's try to resolve some of the issues with our regression model above.

i. Create fully standardized regression results: are these slopes easier to compare? (note: consider if you should standardize origin)

```r
model_std <- lm(scale(mpg) ~ scale(cylinders) + scale(displacement) +
                scale(horsepower) + scale(weight) +
                scale(acceleration) + scale(model_year) + origin, data = auto)

summary(model_std)
```

```
##
## Call:
## lm(formula = scale(mpg) ~ scale(cylinders) + scale(displacement) +
##     scale(horsepower) + scale(weight) + scale(acceleration) +
##     scale(model_year) + origin, data = auto)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.15270 -0.26593 -0.01257  0.25404  1.70942
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)         -0.13323    0.03174  -4.198 3.35e-05 ***
## scale(cylinders)    -0.10658    0.06991  -1.524  0.12821
## scale(displacement)  0.31989    0.10210   3.133  0.00186 **
## scale(horsepower)   -0.08955    0.06751  -1.326  0.18549
## scale(weight)       -0.72705    0.07098 -10.243  < 2e-16 ***
## scale(acceleration)  0.02791    0.03465   0.805  0.42110
## scale(model_year)    0.36760    0.02450  15.005  < 2e-16 ***
## origin2              0.33649    0.07247   4.643 4.72e-06 ***
## origin3              0.36505    0.07072   5.162 3.93e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.423 on 383 degrees of freedom
##   (6 observations deleted due to missingness)
## Multiple R-squared:  0.8242, Adjusted R-squared:  0.8205
## F-statistic: 224.5 on 8 and 383 DF,  p-value: < 2.2e-16
```

- In the fully standardized regression model, all continuous variables have been transformed to have mean 0 and standard deviation 1. The variable origin is categorical and thus left unstandardized. The resulting standardized coefficients can now be meaningfully compared in magnitude. For example, weight has the largest absolute standardized coefficient, indicating it has the strongest effect on mpg among all predictors.

ii. Regress mpg over each nonsignificant independent variable, individually. Which ones become significant when we regress mpg over them individually?

```r
nonsig_vars <- c("cylinders", "horsepower", "acceleration")

for (var in nonsig_vars) {
  formula <- as.formula(paste("mpg ~", var))
  model <- lm(formula, data = auto)
  pval <- summary(model)$coefficients[2, 4]
  cat(sprintf("Variable: %-13s - p-value: %.10e\n", var, pval))
}
```

```
## Variable: cylinders     - p-value: 4.5039922462e-81
## Variable: horsepower    - p-value: 7.0319890294e-81
## Variable: acceleration  - p-value: 1.8230915351e-18
```

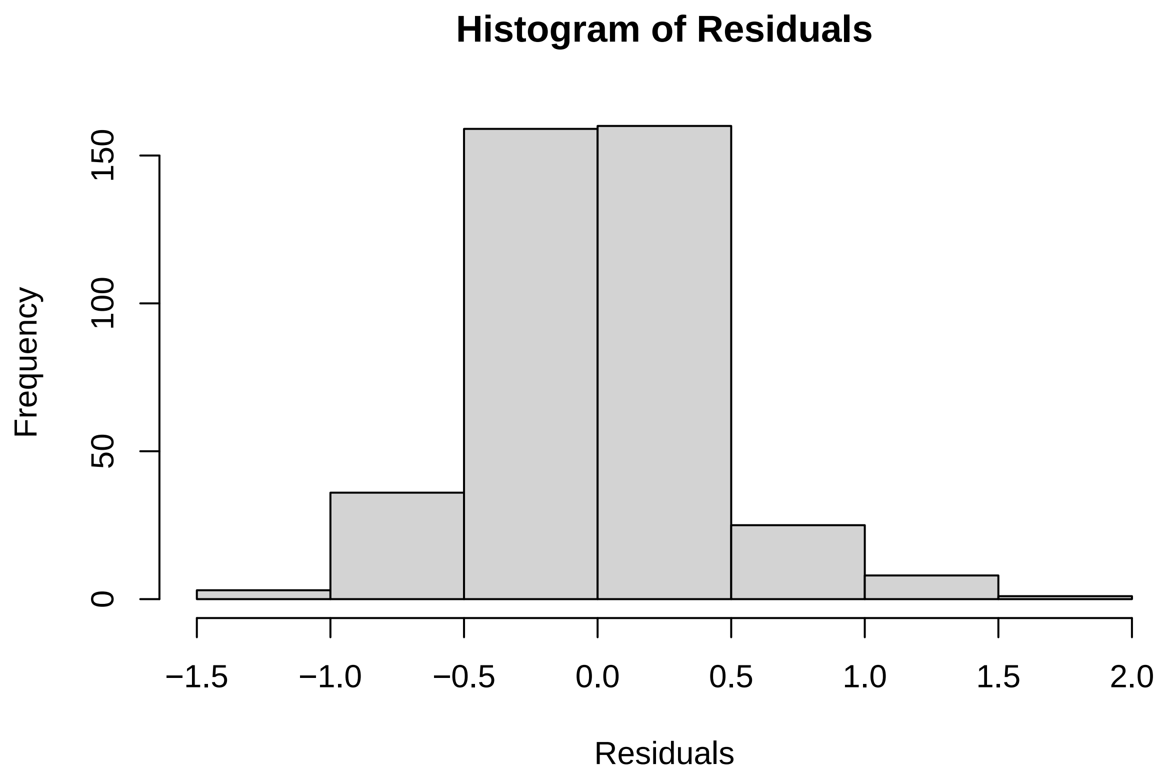
- When we regress each previously nonsignificant variable individually against mpg, we find that “cylinders”, “horsepower”, and “acceleration” become statistically significant at conventional significance levels. This suggests that these variables do have a meaningful relationship with mpg, but their effects were not detected in the full model due to multicollinearity. As a result, their unique contribution to explaining mpg was masked when included alongside highly correlated variables.

iii.Plot the distribution of the residuals: are they normally distributed and centered around zero? (get the residuals of a fitted linear model, e.g. regr <- lm(...), using regr$residuals)

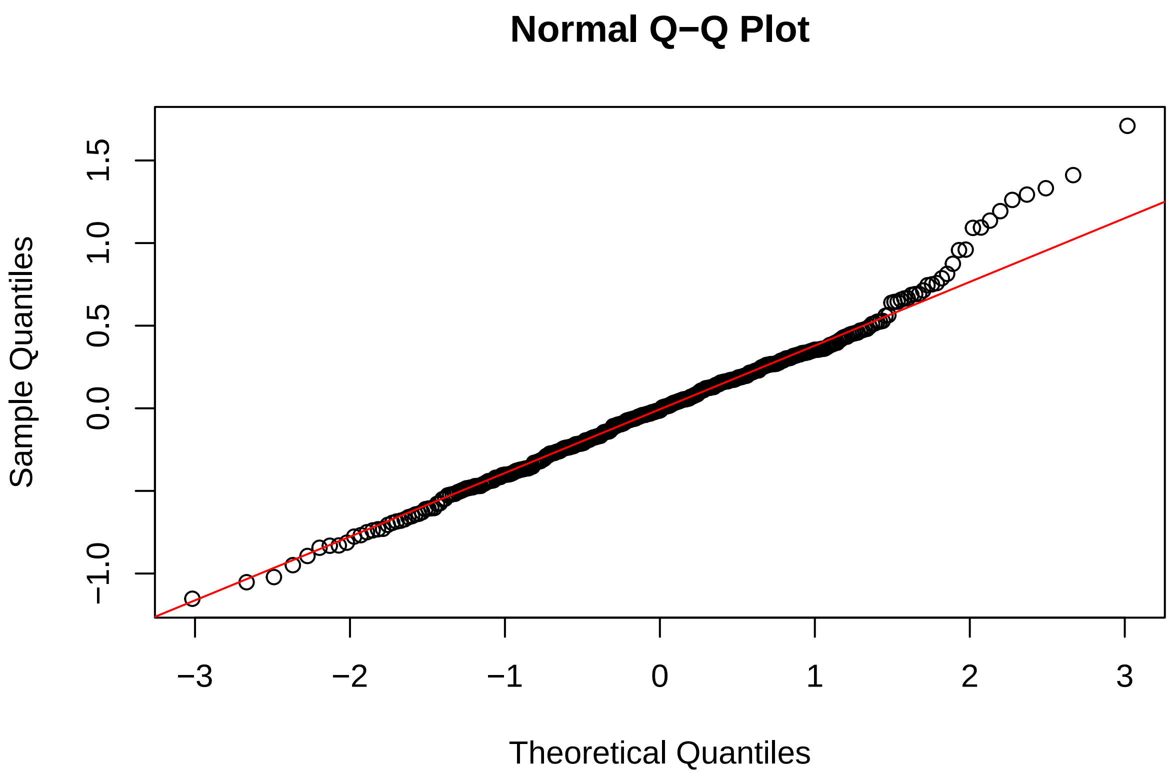
```r
regr <- lm(scale(mpg) ~ scale(cylinders) + scale(displacement) +
                scale(horsepower) + scale(weight) +
                scale(acceleration) + scale(model_year) + origin, data = auto)

res <- regr$residuals

hist(res, main = "Histogram of Residuals", xlab = "Residuals")
```



```r
qqnorm(res)
qqline(res, col = "red")
```



- The histogram shows that the residuals are roughly centered around 0, and the Q-Q plot suggests they are approximately normally distributed, though there may be some deviation in the tails.