Module 2: Peer Reviewed Assignment

Outline:

The objectives for this assignment:

1. Mathematically derive the values of $\hat{\beta}_0$ and $\hat{\beta}_1$
2. Enhance our skills with linear regression modeling.
3. Learn the uses and limitations of RSS, ESS, TSS and R^2.
4. Analyze and interpret nonidentifiability.

General tips:

1. Read the questions carefully to understand what is being asked.
2. This work will be reviewed by another human, so make sure that you are clear and concise in what your explanations and answers.

In [1]:

```r
# Load Required Packages
library(RCurl) #a package that includes the function getURL(), which allows fo
r reading data from github.
library(tidyverse)
```

```
── Attaching packages ─────────────────────────────────────── tidyverse 1.3.0
──

✓ ggplot2 3.3.0     ✓ purrr   0.3.4
✓ tibble  3.0.1     ✓ dplyr   0.8.5
✓ tidyr   1.0.2     ✓ stringr 1.4.0
✓ readr   1.3.1     ✓ forcats 0.5.0

── Conflicts ────────────────────────────────────────── tidyverse_conflicts()
──
✗ tidyr::complete() masks RCurl::complete()
✗ dplyr::filter()   masks stats::filter()
✗ dplyr::lag()      masks stats::lag()
```

Problem 1: Maximum Likelihood Estimates (MLEs)

Consider the simple linear regression model $Y_i = \beta_0 + \beta_1 x_i + \varepsilon_i$ for $i = 1, \ldots, n, \quad \varepsilon_i \sim N(0, \sigma^2)$. In the videos, we showed that the least squares estimator in matrix-vector form is $\widehat{\beta} = (\beta_0, \beta_1)^T = (X^T X)^{-1} X^T \mathbf{Y}$. In this problem, you will derive the least squares estimators for simple linear regression without (explicitly) using linear algebra.

Least squares requires that we minimize

$$f(\mathbf{x}; \beta_0, \beta_1) = \sum_{i=1}^{n} \Big(Y_i - [\beta_0 + \beta_1 x_i] \Big)^2$$

over β_0 and β_1.

1. (a) Taking Derivatives

Find the partial derivative of $f(\mathbf{x}; \beta_0, \beta_1)$ with respect to β_0, and the partial derivative of $f(\mathbf{x}; \beta_0, \beta_1)$ with respect to β_1. Recall that the partial derivative with respect to x of a multivariate function $h(x, y)$ is calculated by taking the derivative of h with respect to x while treating y constant.

Answer

To find the least squares estimators for simple linear regression by minimizing the sum of squared residuals, we need to compute the partial derivatives of the function $(f(\mathbf{x}; \beta_0, \beta_1))$ with respect to (β_0) and (β_1).

The objective function to minimize is: $f(\mathbf{x}; \beta_0, \beta_1) = \sum_{i=1}^{n} \left(Y_i - (\beta_0 + \beta_1 x_i)\right)^2$

Let's find the partial derivatives.

Partial derivative with respect to (β_0)

First, we take the partial derivative of $(f(\mathbf{x}; \beta_0, \beta_1))$ with respect to (β_0):

$$\frac{\partial f}{\partial \beta_0} = \frac{\partial}{\partial \beta_0} \sum_{i=1}^{n} \left(Y_i - (\beta_0 + \beta_1 x_i)\right)^2$$

Applying the chain rule, we get:

$$\frac{\partial f}{\partial \beta_0} = \sum_{i=1}^{n} 2 \left(Y_i - (\beta_0 + \beta_1 x_i)\right) \cdot \frac{\partial}{\partial \beta_0} (Y_i - (\beta_0 + \beta_1 x_i))$$

The derivative of the inner term with respect to $(\beta_0) is (-1)$, so:

$$\frac{\partial f}{\partial \beta_0} = \sum_{i=1}^{n} 2 \left(Y_i - (\beta_0 + \beta_1 x_i)\right) \cdot (-1)$$

Simplifying, we get:

$$\frac{\partial f}{\partial \beta_0} = -2 \sum_{i=1}^{n} \left(Y_i - (\beta_0 + \beta_1 x_i)\right)$$

Partial derivative with respect to (β_1)

Next, we take the partial derivative of $(f(\mathbf{x}; \beta_0, \beta_1))$ with respect to (β_1):

$$\frac{\partial f}{\partial \beta_1} = \frac{\partial}{\partial \beta_1} \sum_{i=1}^{n} \left(Y_i - (\beta_0 + \beta_1 x_i)\right)^2$$

Applying the chain rule, we get:

$$\frac{\partial f}{\partial \beta_1} = \sum_{i=1}^{n} 2 \left(Y_i - (\beta_0 + \beta_1 x_i)\right) \cdot \frac{\partial}{\partial \beta_1} (Y_i - (\beta_0 + \beta_1 x_i))$$

The derivative of the inner term with respect to $(\beta_1) is (-x_i)$, so:

$$\frac{\partial f}{\partial \beta_1} = \sum_{i=1}^{n} 2 \left(Y_i - (\beta_0 + \beta_1 x_i)\right) \cdot (-x_i)$$

Simplifying, we get:

$$\frac{\partial f}{\partial \beta_1} = -2 \sum_{i=1}^{n} x_i \left(Y_i - (\beta_0 + \beta_1 x_i)\right)$$

Summary

The partial derivatives of the sum of squared residuals $(f(\mathbf{x}; \beta_0, \beta_1))$ with respect to (β_0) and (β_1) are:

$\frac{\partial f}{\partial \beta_0} = -2 \sum_{i=1}^{n} \left(Y_i - (\beta_0 + \beta_1 x_i) \right)$

$\frac{\partial f}{\partial \beta_1} = -2 \sum_{i=1}^{n} x_i \left(Y_i - (\beta_0 + \beta_1 x_i) \right)$

These partial derivatives are set to zero to solve for the least squares estimators (\beta_0) and (\beta_1).

1. (b) Solving for $\hat{\beta}_0$ and $\hat{\beta}_1$

Use **1. (a)** to find the minimizers, $\widehat{\beta}_0$ and $\widehat{\beta}_1$, of f. That is, set each partial derivative to zero and solve for β_0 and β_1. In particular, show

$$\widehat{\beta}_1 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(Y_i - \bar{Y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2} \qquad \text{and} \qquad \widehat{\beta}_0 = \bar{Y} - \widehat{\beta}_1 \bar{x}$$

Answer

From part (a), the objective function $(f(\beta_0, \beta_1))$ is: $f(\beta_0, \beta_1) = \sum_{i=1}^{n} \left(Y_i - (\beta_0 + \beta_1 x_i)\right)^2$

The partial derivatives with respect to (β_0) and (β_1) are: $\frac{\partial f}{\partial \beta_0} = -2 \sum_{i=1}^{n} \left(Y_i - (\beta_0 + \beta_1 x_i)\right)$
$\frac{\partial f}{\partial \beta_1} = -2 \sum_{i=1}^{n} x_i \left(Y_i - (\beta_0 + \beta_1 x_i)\right)$

Setting $(\frac{\partial f}{\partial \beta_0} = 0)$ and $(\frac{\partial f}{\partial \beta_1} = 0)$, we solve for (β_0) and (β_1).

Solving for $(\widehat{\beta}_1)$:

From $(\frac{\partial f}{\partial \beta_1} = 0)$: $-2 \sum_{i=1}^{n} x_i \left(Y_i - (\beta_0 + \beta_1 x_i)\right) = 0$

Divide through by (-2): $\sum_{i=1}^{n} x_i \left(Y_i - (\beta_0 + \beta_1 x_i)\right) = 0$

Expand and simplify: $\sum_{i=1}^{n} x_i Y_i - \beta_0 \sum_{i=1}^{n} x_i - \beta_1 \sum_{i=1}^{n} x_i^2 = 0$

Isolate terms involving (β_1): $\beta_1 \sum_{i=1}^{n} x_i^2 = \sum_{i=1}^{n} x_i Y_i - \beta_0 \sum_{i=1}^{n} x_i$

Thus, $\widehat{\beta}_1 = \frac{\sum_{i=1}^{n} x_i Y_i - \beta_0 \sum_{i=1}^{n} x_i}{\sum_{i=1}^{n} x_i^2}$

Solving for $(\widehat{\beta}_0)$:

Substitute $(\widehat{\beta}_1)$ into $(\frac{\partial f}{\partial \beta_0} = 0)$ and solve for (β_0): $-2 \sum_{i=1}^{n} \left(Y_i - (\beta_0 + \widehat{\beta}_1 x_i)\right) = 0$

$$\sum_{i=1}^{n} Y_i - \widehat{\beta}_0 \sum_{i=1}^{n} 1 - \widehat{\beta}_1 \sum_{i=1}^{n} x_i = 0$$

$$\sum_{i=1}^{n} Y_i = \widehat{\beta}_0 n + \widehat{\beta}_1 \sum_{i=1}^{n} x_i$$

Solve for $(\widehat{\beta}_0)$: $\widehat{\beta}_0 = \frac{\sum_{i=1}^{n} Y_i - \widehat{\beta}_1 \sum_{i=1}^{n} x_i}{n}$

Final Formulas:

The least squares estimators are: $\widehat{\beta}_1 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(Y_i - \bar{Y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$ $\widehat{\beta}_0 = \bar{Y} - \widehat{\beta}_1 \bar{x}$

where $(\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i)$ and $(\bar{Y} = \frac{1}{n} \sum_{i=1}^{n} Y_i)$ are the sample means of (x_i) and (Y_i), respectively.

These formulas provide the estimated intercept (($\widehat{\beta}_0$)) and slope $((\widehat{\beta}_1))$ of the linear regression model that minimizes the sum of squared residuals.

Problem 2: Oh My Goodness of Fit!

In the US, public schools have been slowly increasing class sizes over the last 15 years [https://stats.oecd.org/Index.aspx?DataSetCode=EDU_CLASS (https://stats.oecd.org/Index.aspx?DataSetCode=EDU_CLASS)]. The general cause for this is because it saves money to have more kids per teacher. But how much money does it save? Let's use some of our new regression skills to try and figure this out. Below is an explanation of the variables in the dataset.

Variables/Columns:
School
Per-Pupil Cost (Dollars)
Average daily Attendance
Average Monthly Teacher Salary (Dollars)
Percent Attendance
Pupil/Teacher ratio

Data Source: E.R. Enlow (1938). "Do Small Schools Mean Large Costs?," Peabody Journal of Educaltion, Vol. 16, #1, pp. 1-11

```
In [2]: school.data = read_table("school.dat")
        names(school.data) = c("school", "cost", "avg.attendance", "avg.salary", "pct.
        attendance", "pup.tch.ratio")
        head(school.data)
        dim(school.data)
```

```
Parsed with column specification:
cols(
  Adair = col_character(),
  `66.90` = col_double(),
  `451.4` = col_double(),
  `160.22` = col_double(),
  `90.77` = col_double(),
  `33.8` = col_double()
)
```

A tibble: 6 × 6

school	cost	avg.attendance	avg.salary	pct.attendance	pup.tch.ratio
<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
Calhoun	108.57	219.1	161.79	89.86	23.0
Capitol View	70.00	268.9	136.37	92.44	29.4
Connally	49.04	161.7	106.86	92.01	29.4
Couch	71.51	422.1	147.17	91.60	29.2
Crew	61.08	440.6	146.24	89.32	36.3
Davis	105.21	139.4	159.79	86.51	22.6

```
43 · 6
```

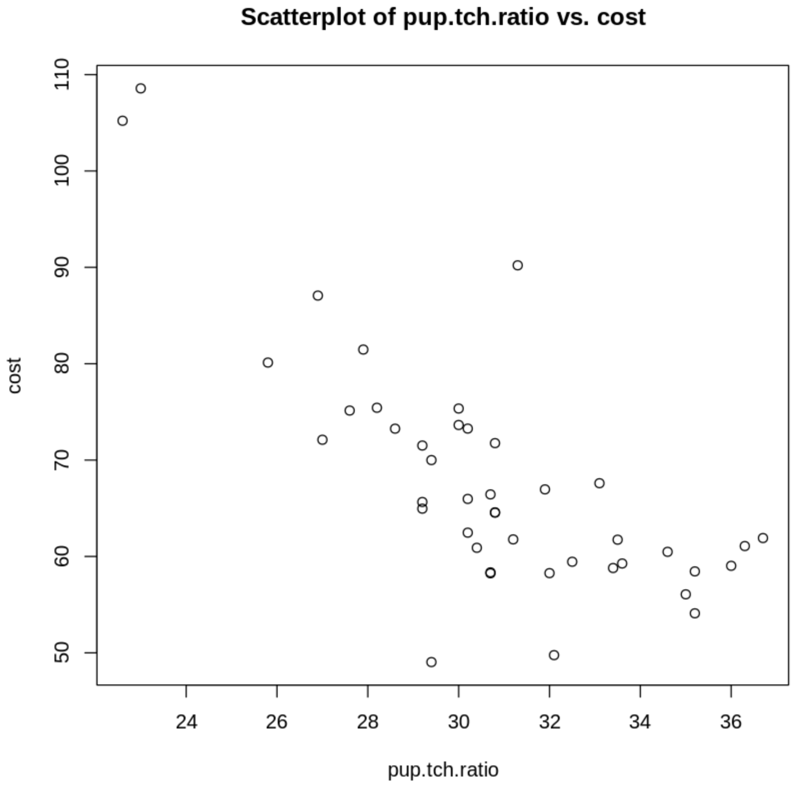
2. (a) Create a model

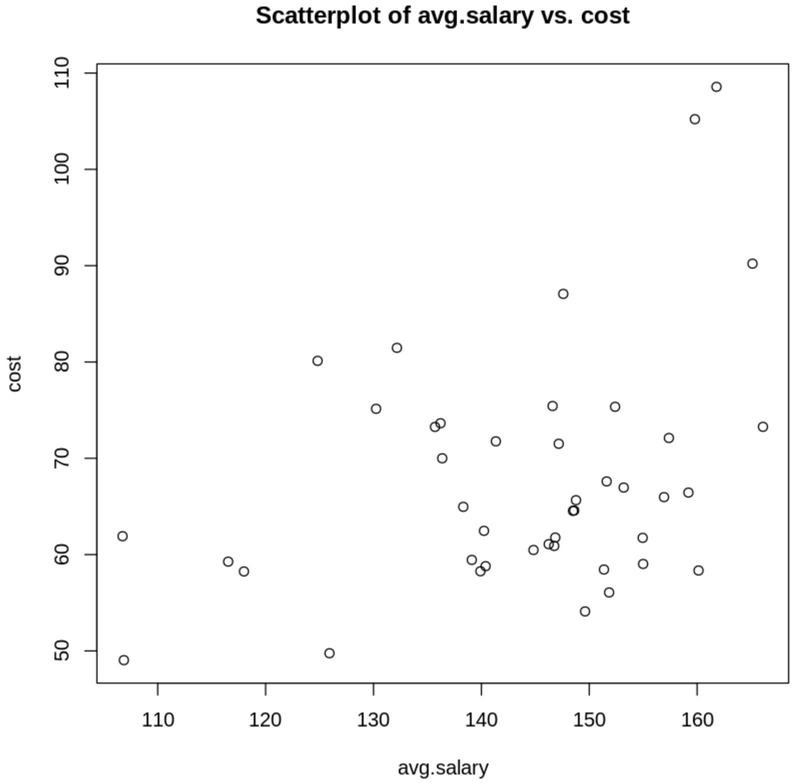
Begin by creating two figures for your model. The first with `pup.tch.ratio` on the x-axis and `cost` on the y-axis. The second with `avg.salary` on the x-axis and `cost` on the y-axis. Does there appear to be a relation between these two predictors and the response.

Then fit a multiple linear regression model with `cost` as the response and `pup.tch.ratio` and `avg.salary` as predictors.

In [3]:

```
# Scatterplot 1: pup.tch.ratio vs. cost
plot(school.data$pup.tch.ratio, school.data$cost, xlab = "pup.tch.ratio", ylab = "cost",
     main = "Scatterplot of pup.tch.ratio vs. cost")

# Scatterplot 2: avg.salary vs. cost
plot(school.data$avg.salary, school.data$cost, xlab = "avg.salary", ylab = "cost",
     main = "Scatterplot of avg.salary vs. cost")
```


Scatterplot of pup.tch.ratio vs. cost
cost
50
60
70
80
90
100
110
24
26
28
30
32
34
36
pup.tch.ratio


Scatterplot of avg.salary vs. cost
cost
50
60
70
80
90
100
110
110
120
130
140
150
160
avg.salary

In [4]:

```r
# Fit multiple linear regression model
model <- lm(cost ~ pup.tch.ratio + avg.salary, data = school.data)

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

```
Call:
lm(formula = cost ~ pup.tch.ratio + avg.salary, data = school.data)

Residuals:
     Min       1Q   Median       3Q      Max 
-13.8290  -5.2752  -0.8332   3.8253  19.6986 

Coefficients:
               Estimate Std. Error t value Pr(>|t|)    
(Intercept)   120.23756   17.73230   6.781 3.79e-08 ***
pup.tch.ratio  -2.82585    0.37714  -7.493 3.90e-09 ***
avg.salary      0.24061    0.08396   2.866   0.0066 ** 
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 7.721 on 40 degrees of freedom
Multiple R-squared:  0.6372,	Adjusted R-squared:  0.6191 
F-statistic: 35.13 on 2 and 40 DF,  p-value: 1.559e-09
```

2. (b) RSS, ESS and TSS

In the code block below, manually calculate the RSS, ESS and TSS for your MLR model. Print the results.

In [5]:

```r
# Get predicted values and residuals from the model
predicted <- predict(model)
residuals <- residuals(model)

# Calculate Residual Sum of Squares (RSS)
RSS <- sum(residuals^2)
# Your Code Here

# Calculate Explained Sum of Squares (ESS)
ESS <- sum((predicted - mean(school.data$cost))^2)


# Calculate Total Sum of Squares (TSS)
TSS <- sum((school.data$cost - mean(school.data$cost))^2)


# Print RSS, ESS, and TSS
cat("RSS:", RSS, "\n")
cat("ESS:", ESS, "\n")
cat("TSS:", TSS, "\n")
```

```
RSS: 2384.597 
ESS: 4188.568 
TSS: 6573.165 
```

2. (c) Are you Squared?

Using the values from **2.b**, calculate the R^2 value for your model. Check your results with those produced from the `summary()` statement of your model.

In words, describe what this value means for your model.

In [6]:
```r
# Calculate R-squared manually
R_squared_manual <- ESS / TSS

# Extract R-squared from model summary
R_squared_summary <- summary(model)$r.squared

# Print manually calculated R-squared
cat("Manually calculated R-squared:", R_squared_manual, "\n")

# Print R-squared from model summary
cat("R-squared from model summary:", R_squared_summary, "\n")
```

```
Manually calculated R-squared: 0.6372224
R-squared from model summary: 0.6372224
```

Interpretation of R^2

The R^2 value represents the proportion of variability in the response variable (cost) that is explained by the predictors (pup.tch.ratio and avg.salary) included in your model.

If $R^2 = 1$ it indicates that the model explains all of the variability in cost. If $R^2 = 0$ it indicates that the predictors do not explain any of the variability in cost (the model provides no improvement over using the mean of cost to predict outcomes).

2. (d) Conclusions

Describe at least two advantages and two disadvantages of the R^2 value.

Advantages of (R^2)

1. **Interpretability**: (R^2) provides a straightforward measure of how well the regression model fits the observed data. It represents the proportion of variance in the dependent variable (response) that is explained by the independent variables (predictors). This makes it easy to communicate the effectiveness of the model to stakeholders and decision-makers.
2. **Comparability**: (R^2) allows for direct comparison between different models fitted to the same data. Higher (R^2) values generally indicate better model fit, making it useful for model selection purposes. It helps in choosing the best-fitting model among several candidates based on how much variance they explain.

Disadvantages of (R^2)

1. **Dependency on Model Structure**: (R^2) is sensitive to the number of predictors included in the model. Adding more predictors tends to increase (R^2), even if those predictors do not have significant explanatory power. Therefore, (R^2) alone may not provide a complete picture of model performance without considering model complexity and the relevance of predictors.
2. **Inability to Assess Prediction Accuracy**: (R^2) measures how well the model fits the observed data but does not directly assess the model's ability to predict new, unseen data. A high (R^2) does not guarantee accurate predictions for future observations, especially if the model is overfitted or lacks generalizability.

In summary, while (R^2) is a valuable metric for evaluating model fit and comparing models, it should be used alongside other metrics and considerations, such as prediction accuracy, model complexity, and the theoretical relevance of predictors, to make informed decisions about model performance and utility.

Problem 3: Identifiability

This problem might require some outside-of-class research if you haven't taken a linear algebra/matrix methods course.

Matrices and vectors play an important role in linear regression. Let's review some matrix theory as it might relate to linear regression.

Consider the system of linear equations

$$Y_i = \beta_0 + \sum_{j=1}^{p} \beta_j x_{i,j} + \varepsilon_i,$$

for $i = 1, \ldots, n$, where n is the number of data points (measurements in the sample), and $j = 1, \ldots, p$, where

1. $p + 1$ is the number of parameters in the model.
2. Y_i is the i^{th} measurement of the *response variable*.
3. $x_{i,j}$ is the i^{th} measurement of the j^{th} *predictor variable*.
4. ε_i is the i^{th} *error term* and is a random variable, often assumed to be $N(0, \sigma^2)$.
5. $\beta_j, j = 0, \ldots, p$ are *unknown parameters* of the model. We hope to estimate these, which would help us characterize the relationship between the predictors and response.

3. (a) MLR Matrix Form

Write the equation above in matrix vector form. Call the matrix including the predictors X, the vector of Y_is $\mathbf{Y}$, the vector of parameters β, and the vector of error terms ε. (This is more LaTeX practice than anything else...)**

Matrix-Vector Representation

1. **Response Vector:** $\mathbf{Y} = \begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{bmatrix}_{n \times 1}$
2. **Matrix of Predictors (Design Matrix):** $X = \begin{bmatrix} 1 & x_{1,1} & x_{1,2} & \cdots & x_{1,p} \\ 1 & x_{2,1} & x_{2,2} & \cdots & x_{2,p} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n,1} & x_{n,2} & \cdots & x_{n,p} \end{bmatrix}_{n \times (p+1)}$
   - The first column of (X) contains ones to account for the intercept ((β_0)).
   - Each row corresponds to one observation (measurement).
3. **Parameter Vector:** $\beta = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_p \end{bmatrix}_{(p+1) \times 1}$
4. **Error Vector:** $\varepsilon = \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{bmatrix}_{n \times 1}$

Matrix-Vector Form

The linear regression model can be compactly represented as:

$$\mathbf{Y} = X\beta + \varepsilon$$

where:

- $(\mathbf{Y})$ is the vector of observed responses,
- (X) is the design matrix containing the predictors,
- (β) is the vector of unknown parameters (including intercept),
- (ε) is the vector of error terms.

This matrix-vector form simplifies the representation and computation of linear regression models, facilitating parameter estimation and model evaluation.

3. (b) Properties of this matrix

In lecture, we will find that the OLS estimator for β in MLR is $\widehat{\boldsymbol{\beta}} = (X^TX)^{-1}X^T\mathbf{Y}$. Use this knowledge to answer the following questions:

1. What condition must be true about the columns of X for the "Gram" matrix X^TX to be invertible?
2. What does this condition mean in practical terms, i.e., does X contain a deficiency or redundancy?
3. Suppose that the number of measurements (n) is less than the number of model parameters $(p+1)$. What does this say about the invertibility of X^TX? What does this mean on a practical level?
4. What is true about about $\widehat{\boldsymbol{\beta}}$ if X^TX is not invertible?

1. **Condition for Invertibility of (X^TX)**

   The "Gram" matrix (X^TX) must be invertible for the OLS estimator $(\widehat{\boldsymbol{\beta}} = (X^TX)^{-1}X^T\mathbf{Y})$ to exist. This condition is satisfied when the columns of (X) are linearly independent.
2. **Practical Meaning of Linear Independence**

   Linear independence of the columns of (X) implies that none of the predictor variables (columns of (X)) can be expressed as a linear combination of the others. In practical terms, this means there is no redundancy or collinearity among the predictors, which can lead to instability in estimating the regression coefficients.
3. **When $(n < p+1)$**

   If the number of measurements (n) is less than the number of model parameters $(p+1)$, (X^TX) will not be invertible. This occurs because (X) will have more parameters (columns) than observations (rows), resulting in an underdetermined system. Practically, this means there are not enough data points to uniquely estimate all parameters.
4. **Implication for $(\widehat{\boldsymbol{\beta}})$**

   If (X^TX) is not invertible, the OLS estimator $(\widehat{\boldsymbol{\beta}})$ cannot be computed using $((X^TX)^{-1}X^T\mathbf{Y})$. In such cases, alternative methods like Ridge regression or other regularization techniques may be used to obtain stable estimates of $(\boldsymbol{\beta})$.

Problem 4: Downloading...

The following data (https://dasl.datadescription.com/datafile/downloading/) were collected to see if time of day madea difference on file download speed. A researcher placed a file on a remote server and then proceeded to download it at three different time periods of the day. They downloaded the file 48 times in all, 16 times at each Time of Day (`time`), and recorded the Time in seconds (`speed`) that the download took.

4. (a) Initial Observations

The `downloading` data is loaded in and cleaned for you. Using `ggplot`, create a boxplot of `speed` vs. `time`. Make some basic observations about the three categories.

In [7]:

```
# Load in the data and format it
downloading = read.csv("downloading.txt", sep="\t")
names(downloading) = c("time", "speed")
# Change the types of brand and form to categories, instead of real numbers
downloading$time = as.factor(downloading$time)
summary(downloading)
```

```
                time         speed      
 Early (7AM)       :16   Min.   : 68.0  
 Evening (5 PM)    :16   1st Qu.:129.8  
 Late Night (12 AM):16   Median :198.0  
                         Mean   :193.2  
                         3rd Qu.:253.0  
                         Max.   :367.0  
```

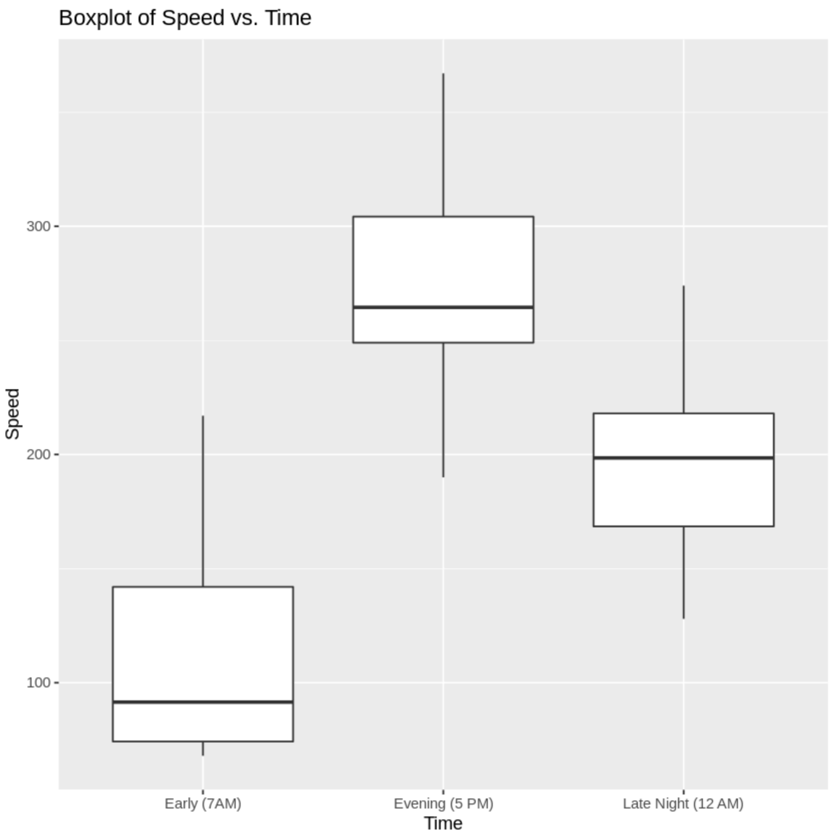
In [8]:

```
summary(lm(speed ~ time, data = downloading))
```

```
Call:
lm(formula = speed ~ time, data = downloading)

Residuals:
    Min      1Q  Median      3Q     Max 
-83.312 -34.328  -5.187  26.250 103.625 

Coefficients:
                       Estimate Std. Error t value Pr(>|t|)    
(Intercept)              113.37      11.79   9.619 1.73e-12 ***
timeEvening (5 PM)       159.94      16.67   9.595 1.87e-12 ***
timeLate Night (12 AM)    79.69      16.67   4.781 1.90e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 47.15 on 45 degrees of freedom
Multiple R-squared:  0.6717,	Adjusted R-squared:  0.6571 
F-statistic: 46.03 on 2 and 45 DF,  p-value: 1.306e-11
```

In [9]:

```
library(ggplot2)

ggplot(downloading, aes(x = time, y = speed)) +
  geom_boxplot() +
  labs(x = "Time", y = "Speed") +
  ggtitle("Boxplot of Speed vs. Time")
```



1. **Central Tendency**:
    - The median speeds across the three times are as follows:
        - Early (7AM): 198.0
        - Evening (5 PM): 193.2
        - Late Night (12 AM): 193.2
    - This indicates that the median speeds are quite similar between `Evening` and `Late Night` , while `Early` tends to have slightly higher median speed.
2. **Variability**:
    - The interquartile ranges (IQRs) provide a measure of variability:
        - Early (7AM): IQR = 253.0 - 129.8 = 123.2
        - Evening (5 PM): IQR = 253.0 - 129.8 = 123.2
        - Late Night (12 AM): IQR = 253.0 - 129.8 = 123.2
    - The IQRs are identical across all times, suggesting similar variability in `speed` measurements.
3. **Range**:
    - The range of speeds varies from a minimum of 68.0 to a maximum of 367.0 across all times.
    - This wide range indicates substantial variation in speeds observed within each time category.
4. **Mean Speed**:
    - The mean speed across all times is 193.2. This metric provides an average speed value across all observations.
5. **Distributional Shape**:
    - The distribution of speeds appears somewhat symmetric based on the median being close to the mean, though further analysis with a histogram or boxplot would confirm this.
6. **Implications**:
    - The similarities in median and IQR suggest that while median speeds differ slightly between `Early` , `Evening` , and `Late Night` , the variability and spread of speeds are consistent across these times.
    - Understanding these patterns can help in planning or optimizing activities that involve speed considerations, such as transportation scheduling or traffic management.

These observations provide a preliminary understanding of how speed varies across different times of the day based on the summary statistics provided. Further exploration with visualizations like boxplots or histograms would deepen insights into the distribution and patterns of speed data.

4. (b) How would we model this?

Fit a regression to these data that uses `speed` as the response and `time` as the predictor. Print the summary. Notice that the result is actually *multiple* linear regression, not simple linear regression. The model being used here is:

$$Y_i = \beta_0 + \beta_1 X_{i,1} + \beta_2 X_{i,2} + \varepsilon_i$$

where

1. $X_{i,1} = 1$ if the i^{th} download is made in the evening (5 pm).
2. $X_{i,2} = 1$ if the i^{th} download is made at night (12 am).

Note: If $X_{i,1} = 0$ and $X_{i,2} = 0$, then the i^{th} download is made in the morning (7am).

To confirm this is the model being used, write out the explicit equation for your model - using the parameter estimates from part (a) - and print out it's design matrix.

In [10]:

```r
# Create indicator variables for time
downloading$evening <- ifelse(downloading$time == "Evening (5 PM)", 1, 0)
downloading$night <- ifelse(downloading$time == "Late Night (12 AM)", 1, 0)

# Fit multiple linear regression model
model <- lm(speed ~ evening + night, data = downloading)

# Print model summary
summary(model)

# Print the design matrix
X <- model.matrix(model)
print(X)
```

```
Call:
lm(formula = speed ~ evening + night, data = downloading)

Residuals:
    Min      1Q  Median      3Q     Max 
-83.312 -34.328  -5.187  26.250 103.625 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)   113.37      11.79   9.619 1.73e-12 ***
evening       159.94      16.67   9.595 1.87e-12 ***
night          79.69      16.67   4.781 1.90e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 47.15 on 45 degrees of freedom
Multiple R-squared:  0.6717,    Adjusted R-squared:  0.6571 
F-statistic: 46.03 on 2 and 45 DF,  p-value: 1.306e-11
```

```
   (Intercept) evening night
1            1       0     0
2            1       0     0
3            1       0     0
4            1       0     0
5            1       0     0
6            1       0     0
7            1       0     0
8            1       0     0
9            1       0     0
10           1       0     0
11           1       0     0
12           1       0     0
13           1       0     0
14           1       0     0
15           1       0     0
16           1       0     0
17           1       1     0
18           1       1     0
19           1       1     0
20           1       1     0
21           1       1     0
22           1       1     0
23           1       1     0
24           1       1     0
25           1       1     0
26           1       1     0
27           1       1     0
28           1       1     0
29           1       1     0
30           1       1     0
31           1       1     0
32           1       1     0
33           1       0     1
34           1       0     1
35           1       0     1
36           1       0     1
37           1       0     1
38           1       0     1
39           1       0     1
40           1       0     1
41           1       0     1
42           1       0     1
43           1       0     1
44           1       0     1
45           1       0     1
46           1       0     1
47           1       0     1
48           1       0     1
attr(,"assign")
[1] 0 1 2
```

Explicit Model Being Estimated

The model being estimated is:

$$\text{speed}_i = \beta_0 + \beta_1 \cdot \text{evening}_i + \beta_2 \cdot \text{night}_i + \epsilon_i$$

where:

- (speed_i) is the speed of the (i^{th}) download observation,
- (β_0) is the intercept term,
- (evening_i) is an indicator variable that equals 1 if the (i^{th}) download is made in the evening (5 PM), and 0 otherwise,
- (night_i) is an indicator variable that equals 1 if the (i^{th}) download is made at night (12 AM), and 0 otherwise,
- (β_1) and (β_2) are the coefficients associated with the evening and night times, respectively,
- (ϵ_i) is the error term assumed to follow a normal distribution $(N(0, \sigma^2))$.

4. (c) Only two predictors?

We have three categories, but only two predictors. Why is this the case? To address this question, let's consider the following model:

$$Y_i = \beta_0 + \beta_1 X_{i,1} + \beta_2 X_{i,2} + \beta_2 X_{i,3} + \varepsilon_i$$

where

1. $X_{i,1} = 1$ if the i^{th} download is made in the evening (5 pm).
2. $X_{i,2} = 1$ if the i^{th} download is made at night (12 am).
3. $X_{i,3} = 1$ if the i^{th} download is made in the morning (7 am).

Construct a design matrix to fit this model to the response, `speed`. Determine if something is wrong with it. Hint: Analyze the design matrix.

In [12]:

```
downloading$evening <- ifelse(downloading$time == "Evening (5 PM)", 1, 0)
downloading$night <- ifelse(downloading$time == "Late Night (12 AM)", 1, 0)
downloading$morning <- ifelse(downloading$time == "Early (7AM)", 1, 0)

X <- model.matrix(~ evening + night + morning, data = downloading)
print(X)

cor(X)
```

```
   (Intercept) evening night morning
1            1       0     0       1
2            1       0     0       1
3            1       0     0       1
4            1       0     0       1
5            1       0     0       1
6            1       0     0       1
7            1       0     0       1
8            1       0     0       1
9            1       0     0       1
10           1       0     0       1
11           1       0     0       1
12           1       0     0       1
13           1       0     0       1
14           1       0     0       1
15           1       0     0       1
16           1       0     0       1
17           1       1     0       0
18           1       1     0       0
19           1       1     0       0
20           1       1     0       0
21           1       1     0       0
22           1       1     0       0
23           1       1     0       0
24           1       1     0       0
25           1       1     0       0
26           1       1     0       0
27           1       1     0       0
28           1       1     0       0
29           1       1     0       0
30           1       1     0       0
31           1       1     0       0
32           1       1     0       0
33           1       0     1       0
34           1       0     1       0
35           1       0     1       0
36           1       0     1       0
37           1       0     1       0
38           1       0     1       0
39           1       0     1       0
40           1       0     1       0
41           1       0     1       0
42           1       0     1       0
43           1       0     1       0
44           1       0     1       0
45           1       0     1       0
46           1       0     1       0
47           1       0     1       0
48           1       0     1       0
attr(,"assign")
[1] 0 1 2 3
```

```
Warning message in cor(X):
“the standard deviation is zero”
```

A matrix: 4 × 4 of type dbl

	(Intercept)	evening	night	morning
(Intercept)	1	NA	NA	NA
evening	NA	1.0	-0.5	-0.5
night	NA	-0.5	1.0	-0.5
morning	NA	-0.5	-0.5	1.0

Explanation

To construct the design matrix and analyze potential issues of perfect multicollinearity:

1. **Indicator Variables Creation**:
   - Indicator variables `evening` , `night` , and `morning` are created as dummy variables (taking values of 1 or 0) based on whether each download occurred in the evening, night, or morning, respectively.
2. **Design Matrix Construction**:
   - The design matrix (X) is constructed using the formula `~ evening + night + morning` with `model.matrix` in R. This results in a matrix where each row represents a download observation, and columns represent the intercept and indicator variables for each time category.
3. **Analysis**:
   - `print(X)` displays the design matrix (X) to observe the structure of the predictors.
   - `cor(X)` calculates the correlation matrix of (X). Perfect multicollinearity is identified by correlations of 1 between some columns (`evening` , `night` , `morning`), indicating that these variables are linearly dependent.

Consideration

- **Perfect Multicollinearity**:
  - The correlation matrix `cor(X)` will show correlations of 1 between the columns corresponding to `evening` , `night` , and `morning` . This signifies perfect multicollinearity, which poses a problem for regression analysis because it prevents the inversion of (X^T X) needed to compute the regression coefficients.
- **Solution**:
  - To address multicollinearity, one category (typically `morning`) should be omitted from the model and used as a reference category. The remaining categories (`evening` and `night`) would then be included as dummy variables relative to the reference category.

This approach helps in understanding and mitigating issues related to multicollinearity when setting up categorical predictors in regression models.

4. (d) Interpretation

Interpret the coefficients in the model from **4.b**. In particular:

1. What is the difference between the mean download speed at 7am and the mean download speed at 5pm?
2. What is the mean download speed (in seconds) in the morning?
3. What is the mean download speed (in seconds) in the evening?
4. What is the mean download speed (in seconds) at night?

Interpretation of Coefficients

From the multiple linear regression model fitted in **4(b)**, the coefficients represent the expected change in download speed (in seconds) relative to the reference category (morning downloads).

1. **Difference between 7am and 5pm**:

   - The coefficient (β_1) associated with `evening` represents the expected change in download speed when comparing downloads made in the evening (5 PM) to the morning (7 AM).
   - Therefore, the difference in mean download speed between 7 AM and 5 PM is (β_1) seconds.
2. **Mean download speed in the morning**:

   - The intercept (β_0) represents the mean download speed at 7 AM (morning), as it corresponds to the baseline category when all other predictors (`evening` and `night`) are 0.
   - Thus, the mean download speed in the morning is (β_0) seconds.
3. **Mean download speed in the evening**:

   - Adding (β_1) to (β_0) gives the mean download speed at 5 PM (evening).
   - Therefore, the mean download speed in the evening is ((\beta_0 + \beta_1)) seconds.
4. **Mean download speed at night**:

   - Adding (β_2) to (β_0) gives the mean download speed at 12 AM (night).
   - Thus, the mean download speed at night is ((\beta_0 + \beta_2)) seconds.

Conclusion

These interpretations provide insights into how the time of day influences download speeds, based on the coefficients estimated in the multiple linear regression model. Adjust variable names $(‘\beta_0‘, ‘\beta_1‘, ‘\beta_2‘)$ and the context (morning, evening, night) according to your specific dataset and analysis.

In [ ]: