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 [24]:
```
# Load Required Packages
library(RCurl) #a package that includes the function getURL(), which allow
s for reading data from github.
library(tidyverse)
```

Problem 1: Maximum Likelihood Estimates (MLEs)

Consider the simple linear regression model $Y_i = \beta_0 + \beta_1 x_i + \varepsilon_i$ for $i = 1, \ldots, n$, $\varepsilon_i \sim N(0, \sigma^2)$. In the videos, we showed that the least squares estimator in matrix-vector form is $\hat{\beta} = (\beta_0, \beta_1)^T = (X^T X)^{-1} X^T Y$. 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 $f(x; \beta_0, \beta_1) = n\sum i=1\left(Y_i - [\beta_0 + \beta_1 x_i]\right)^2$ over β_0 and β_1.

1. (a) Taking Derivatives

Find the partial derivative of $f(x; \beta_0, \beta_1)$ with respect to β_0, and the partial derivative of $f(x; \beta_0, \beta_1)$ 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.

First, take the partial derivative of f with respect to β_0:

$\partial\partial\beta_0 f(x; \beta_0, \beta_1) = \partial\partial\beta_0 n\sum i=1\left(Y_i - [\beta_0 + \beta_1 x_i]\right)^2 = n\sum i=1 - 2(Y_i - \beta_0 - \beta_1 x_i) = n\sum i=1(Y_i - \beta_0 - \beta_1 x_i)$

$= n\sum i=1 Y_i - n\sum i=1 \beta_0 - \beta_1 n\sum i=1 x_i = n\sum i=1 \beta_0$

Find the partial derivative of f with respect to β_1 in a similar fashion: $\partial\partial\beta_1 f(x; \beta_0, \beta_1)$

$= \partial\partial\beta_0 n\sum i=1\left(Y_i - [\beta_0 + \beta_1 x_i]\right)^2 = n\sum i=1 - 2x_i(Y_i - \beta_0 - \beta_1 x_i)$

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

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

$\hat{\beta}_1 = \sum_{ni=1}(x_i - \bar{x})(Y_i - \bar{Y})\sum_{ni=1}(x_i - \bar{x})^2$ and $\hat{\beta}_0 = \bar{Y} - \hat{\beta}_1\bar{x}$

Start with $\hat{\beta}_0$. Set partial derivative from part (a) to 0 and solve: $0 = n\sum i=1 - 2(Y_i - \beta_0 - \beta_1 x_i)$
$0 = n\sum i=1(Y_i - \beta_0 - \beta_1 x_i)$ $0 = n\sum i=1 Y_i - n\sum i=1\beta_0 - \beta_1 n\sum i=1 x_i$
Note that $\sum_{ni=1}\beta_0 = n\beta_0$. Pluggin in, we get $0 = n\sum i=1 Y_i - n\beta_0 - \beta_1 n\sum i=1 x_i$
Rearrange and solve for β_0: $n\beta_0 = n\sum i=1 Y_i - \beta_1 n\sum i=1 x_i$ $\beta_0 = \sum_{ni=1} Y_i - \beta_1\sum_{ni=1} x_i n$

The summation of Y_i and x_i divided by the number of observations is simply the mean, $\bar{Y}$ and $\bar{x}$, so $\beta_0 = \bar{Y} - \hat{\beta}_1\bar{x}$

Moving on to $\hat{\beta}_1$, solve in a similar fashion:
$0 = n\sum i=1 - 2x_i(Y_i - \beta_0 - \beta_1 x_i)$ $0 = n\sum i=1 x_i(Y_i - \beta_0 - \beta_1 x_i)$ ***Note*** ------ Dividing both sides by -2 as done at the above step produces a different form of the answer we are out to show. I only later realized this after multiple checks, but due to time constraint, left it as is.

Distribute x_i to each term: $0 = n\sum i=1(x_i Y_i - x_i\beta_0 - \beta_1 x_{2i})$
Substitute $\beta_0 = \bar{Y} - \hat{\beta}_1\bar{x}$ to get rid of β_0 and have the function in terms of x and Y only.
$0 = n\sum i=1(x_i Y_i - (\bar{Y} - \beta_1\bar{x})x_i - \beta_1 x_{2i})$ $0 = n\sum i=1(x_i Y_i - \bar{Y}x_i + \beta_1\bar{x}x_i - \beta_1 x_{2i})$
$0 = n\sum i=1(x_i Y_i - \bar{Y}x_i) + n\sum i=1(\beta_1\bar{x}x_i - \beta_1 x_{2i})$ $0 = n\sum i=1(x_i Y_i - \bar{Y}x_i) - \beta_1 n\sum i=1(x_{2i} - \bar{x}x_i)$ Isolate β_1 term and solve:
$\beta_1 = \sum_{ni=1}(x_i Y_i - \bar{Y}x_i)\sum_{ni=1}(x_{2i} - \bar{x}x_i)$ As explained in ***Note*** above, this is a different form of the answer we are supposed to show.

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]. 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 [25]:
```
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

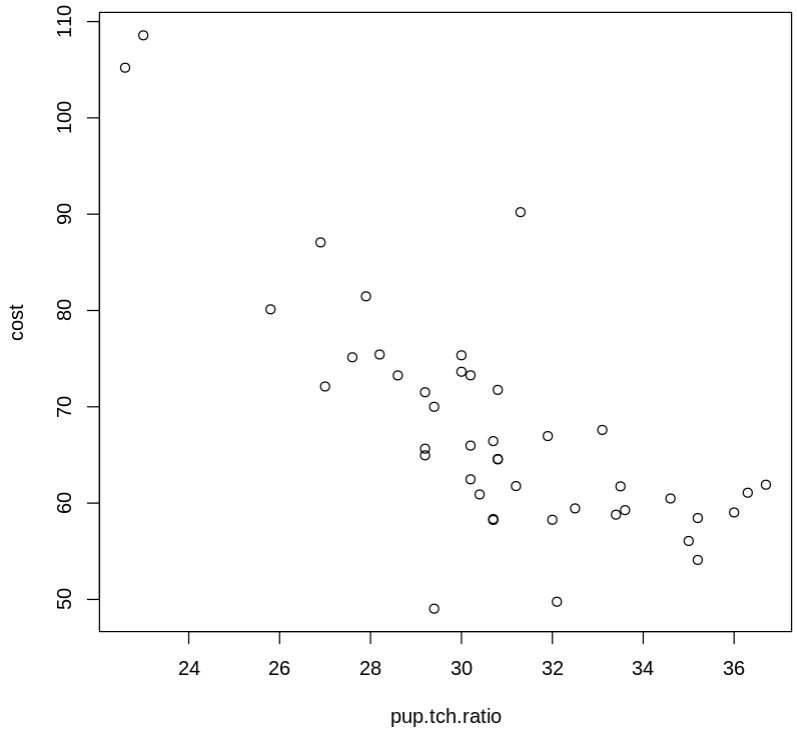
2. (a) Create a model

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.

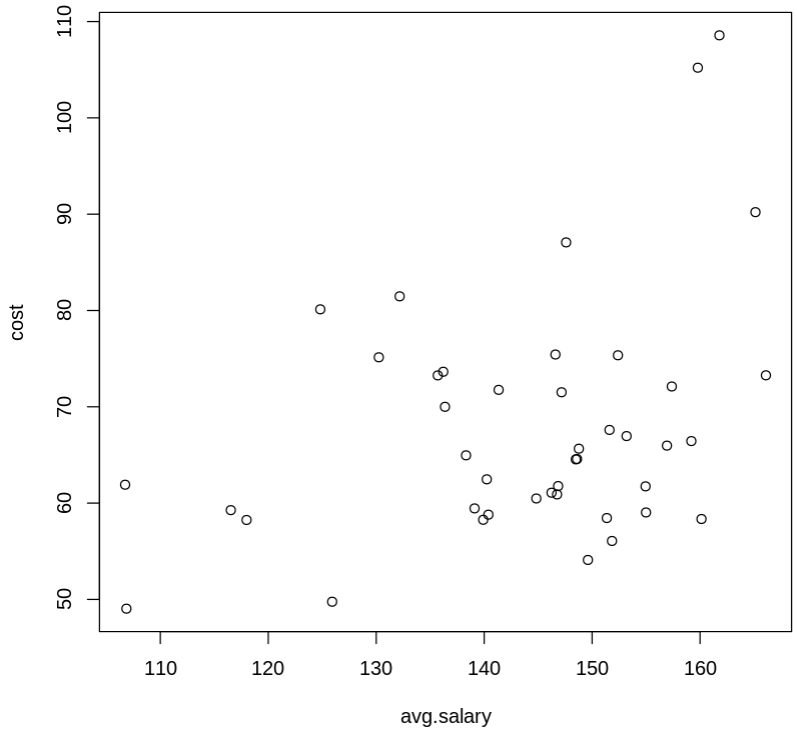
In [26]:
```
# Your Code Here
```

```r
# Create first model with pup.tch.ratio on x-axis and cost on y-axis:
with(school.data, plot(pup.tch.ratio, cost))
    # There seems to be a negative, potentially linear/quadratic relations
hip between pup.tch.ratio and cost.

# Create second model with avg.salary on x-axis and cost on y-axis:
with(school.data, plot(avg.salary, cost))
    # The plot seems to show little relationship between avg.salary and co
st.

# Fit multiple linear regression model with cost as the response and pup.t
ch.ratio and avg.salary as predictors.
fit1 = lm(cost ~ pup.tch.ratio + avg.salary, data = school.data)
summary(fit1)
    # We can see from the summary that the p-values for each predictor is
less than 0.1, which says that the expected outcome is highly likely and t
herefore replicable.

# Plot individual predictor vs cost with regression model:
ggplot(school.data, aes(y = cost, x = avg.salary)) + geom_point() + stat_s
mooth(method="lm", se=FALSE)
    # The plot shows a weak positive linear relationship between the predi
ctor and response variables, with quite a few outliers.

ggplot(school.data, aes(y = cost, x = pup.tch.ratio)) + geom_point() + sta
t_smooth(method="lm", se=FALSE)
    # The plot shows a positive linear relationship between the predictor
and response variables.

# Plot data with multiple regression:
ggplot(school.data, aes(y = cost, x = pup.tch.ratio, color = avg.salary))
+ geom_point() + stat_smooth(method="lm", se=FALSE)
    # As pup.tch.ratio increases, the cost decreases. The average salary a
ppear to be higher near the regression line, but remain quite unrelated re
lative to the ratio variable.

# Show diagnostic plots
layout(matrix(c(1,2,3,4),2,2))
plot(fit1)
    # From the first plot we can see that the relationship between the two
 predictor variables
    # and the response variable is approximately quadratic.
```

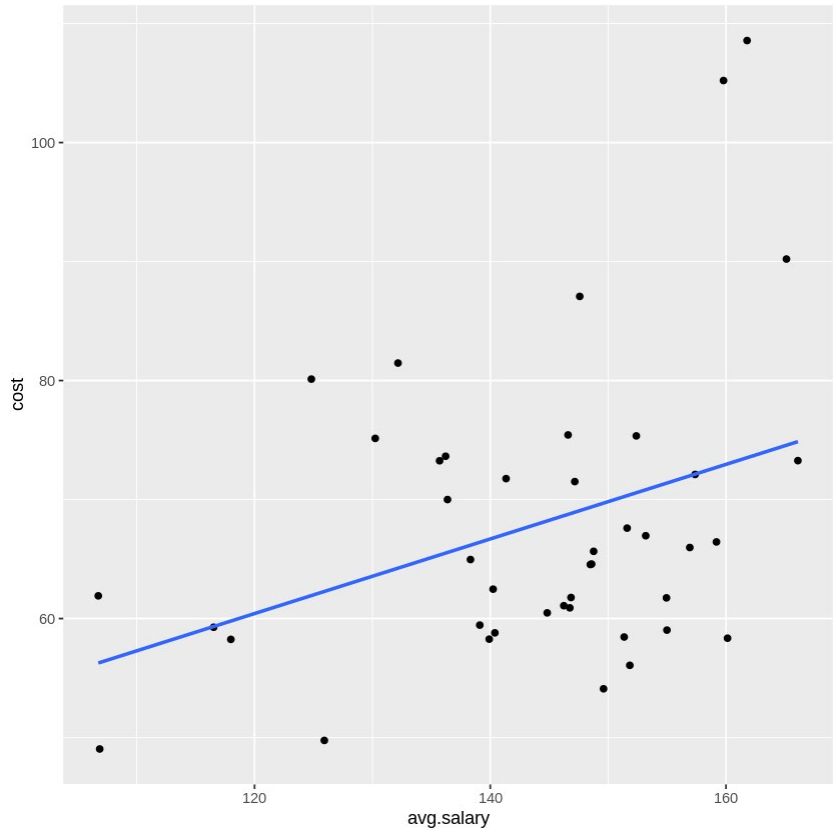



```
Call:
lm(formula = cost ~ pup.tch.ratio + avg.salary, data = school.data)

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
```

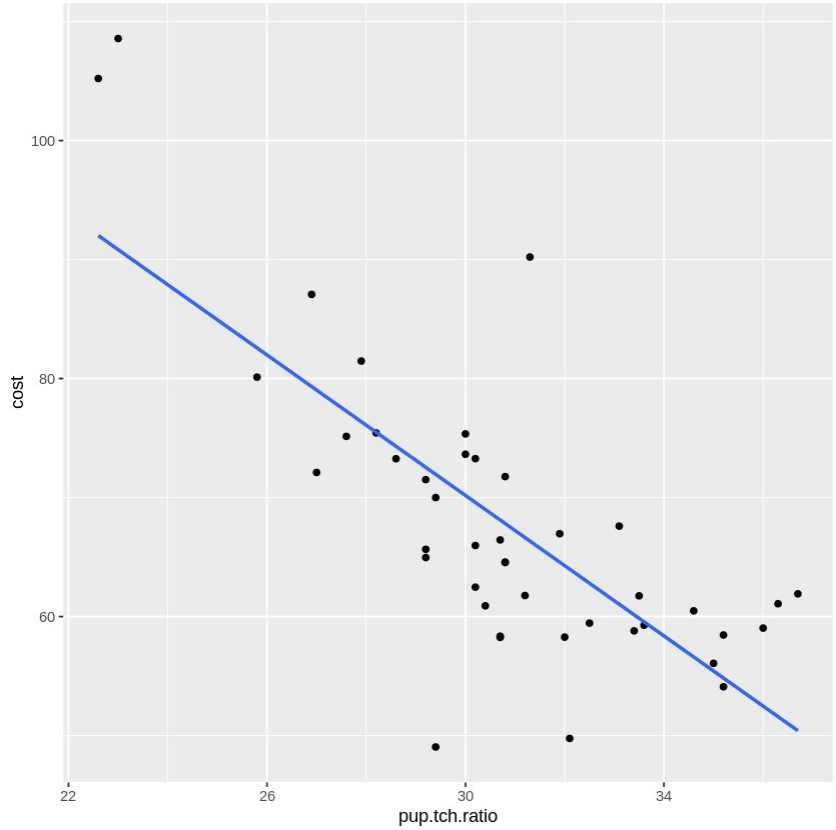
```
`geom_smooth()` using formula 'y ~ x'
```

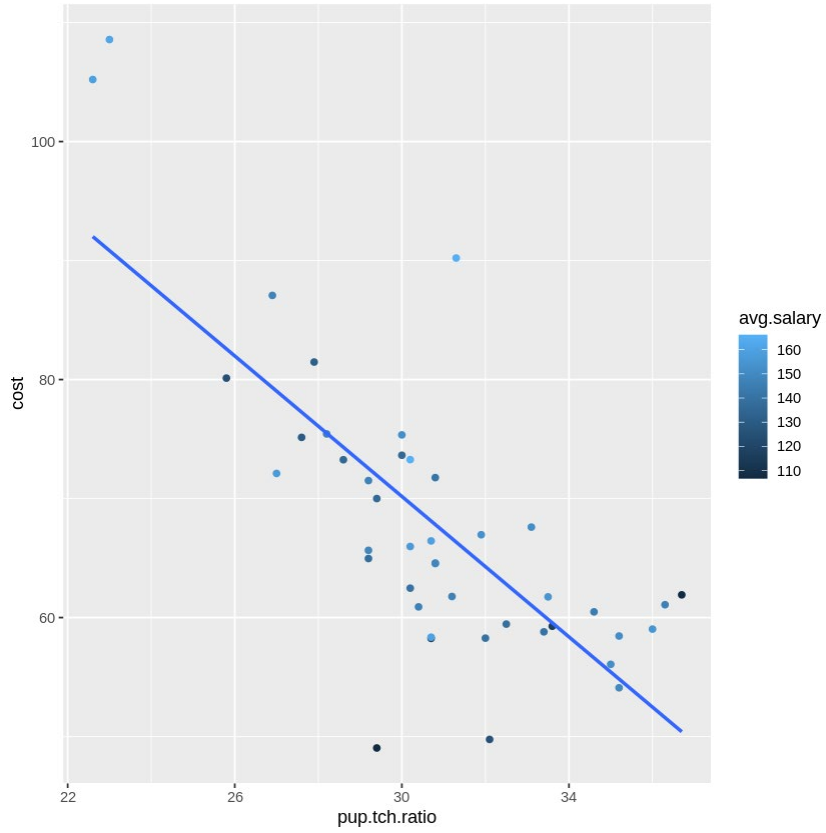


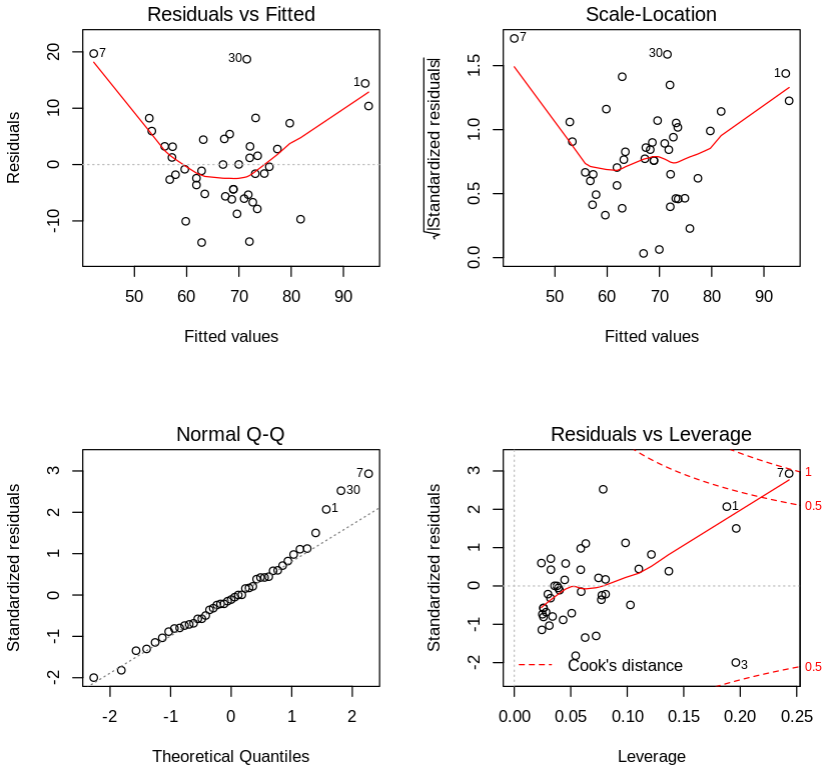

```
`geom_smooth()` using formula 'y ~ x'
```




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


100
80
60
cost
22
26
30
34
pup.tch.ratio


100
80
60
cost
22
26
30
34
pup.tch.ratio
avg.salary
160
150
140
130
120
110



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 [27]:
```
# Your Code Here
summary(fit1)

# Calculate residual sum of squares (RSS).
RSS = sum(residuals(fit1)^2)

# Calculate explained sum of squares (ESS).
ESS = sum((fitted(fit1) - mean(school.data$cost))^2)

# Calculate total sum of squares (TSS)
TSS = with(school.data, sum((cost - mean(cost))^2))

cat(paste("The RSS is", round(RSS, 2), "."),
    paste("The ESS is", round(ESS, 2), "."),
    paste("The TSS is", round(TSS, 2), "."),
    sep = "\n")
```

```
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

The RSS is 2384.6 .
The ESS is 4188.57 .
The TSS is 6573.17 .
```

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 [28]:
```
# Your Code Here
# Calculate R^2:
R_2 = 1 - RSS/TSS
R_2
summary(fit1)
```

0.637222427559588

```
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
```

In order to determine how well or poorly a regression model fits the data, we need to determine the differences between the observations and the predicted values. The smaller and unbiased these differences are, the better the fit. Essentially, R^2 is a numeric measure of the goodness of fit. The higher the coefficient of determination R^2, the smaller the differences between the observed and

fitted values.

For our case, the R^2 value came out to be $R^2 = 0.6372$. This means that about 63.7% of variation in cost can be explained by the regression model (pupil-teacher ratio, average salary).

We can see that the calculated R^2 value and the R^2 value from the `summary()` statement match.

2. (d) Conclusions

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

Advantages
1) Given that the model is correct, it gives us information about associations and correlations between the predictor and response variables.

2) We can determine how well the regression model fits the data in a simple numeric measure that represents the proportion of data that can be explained by the set of predictors.

Disadvantages
1) If the model is the wrong fit for a given dataset, the computed R^2 value can be misleading. (showing a value close to 1 in a relatively linear region of a quadratic data).

2) Without making adjustments/modifications to the original R^2 computation, we cannot use it to compare different models. This is because the computation of R^2 is designed in such a way that adding new predictors will almost always increase the R^2 value. Comparing two models with different number of predictors will produce biased/inaccurate R^2 values.

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

for i = 1, . . . , n, where n is the number of data points (measurements in the sample), and j = 1, . . . , 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. β_j, j = 0, . . . , 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 Y, the vector of parameters β, and the vector of error terms ε. (This is more LaTeX practice than anything else...)**

Vector of Y_is: $Y = \begin{pmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{pmatrix}$

Vector of parameters $\beta = \begin{pmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_p \end{pmatrix}$

Vector of predictors

$$X = \begin{pmatrix} 1 & X_{1,1} & X_{1,2} & \dots & X_{1,p} \\ 1 & X_{2,1} & X_{2,2} & \dots & X_{2,p} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & X_{n,1} & \dots & \dots & X_{n,p} \end{pmatrix}$$

Vector of error terms $\varepsilon = \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{pmatrix}$

So the system of equations $Y_i = \beta_0 + \sum_{pj=1} \beta_j x_{i,j} + \varepsilon_i$, in matrix-vector form becomes:

$$\begin{pmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{pmatrix} = \begin{pmatrix} 1 & X_{1,1} & X_{1,2} & \dots & X_{1,p} \\ 1 & X_{2,1} & X_{2,2} & \dots & X_{2,p} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & X_{n,1} & . \end{pmatrix}$$

3. (b) Properties of this matrix

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

1. What condition must be true about the columns of X for the "Gram" matrix $X^T X$ 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 $X^T X$? What does this mean on a practical level?
4. What is true about about $\hat{\beta}$ if $X^T X$ is not invertible?

1. In order for the "Gram" matrix $X^T X$ to be invertible, the columns of X (or column vectors of X) need to be linearly independent.

2. Having linearly independent column vectors in X means that there is no redundancy in the data, given that the matrix has full rank and the columns cannot be recreated using linear combinations.
3. If the number of measurements n is less than the number of model parameters, some columns in X can be recreated by linear combination. This means that X is not invertible since the columns are no longer linearly independent. In practical terms, not all columns can be used as explanatory variables in a linear model.
4. If X^TX is not invertible (with columns of X linearly dependent), then some vectors will end up being zero. This means that along the zero vectors, there is no change in the data regardless of fit with respect to the regression model. In this sense, even if the parameters (or ˆB) contribute to the model, there is no way to identify them or learn anything about the linear combinations of the parameters.

Problem 4: Downloading...

The following data 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.

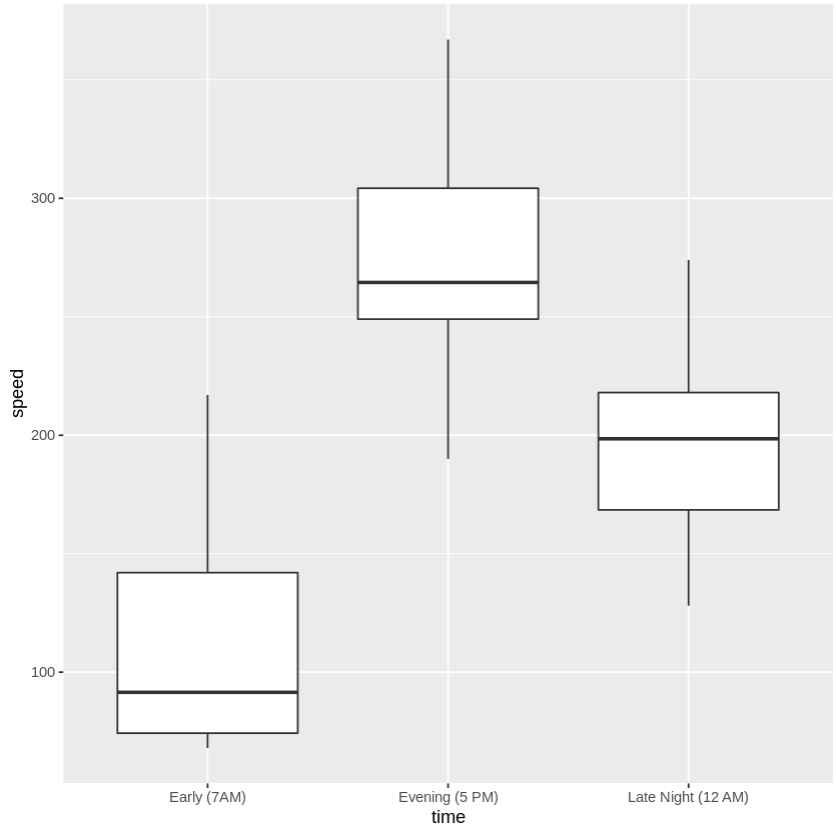
In [29]:
```
# Load in the data and format it
downloading = read.csv("downloading.txt", sep="\t")
names(downloading) = c("time", "speed")
# Change the types of brand and form to categories, instead of real number
s
downloading$time = as.factor(downloading$time)
summary(downloading)
```

```
                 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 [30]:
```
summary(lm(speed ~ time, data = downloading))
ggplot(downloading, aes(x = time, y = speed)) + geom_boxplot()
```

```
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
```



The plot shows that the download speed is fastest during the evening at 5 PM. It slows down at subsequent hours, recording an average of 200 at midnight and a little less than 100 early in the morning at 7 AM. It can also be noted that fluctuations in download speed is rather large. Varying traffic depending on locations, internet accessibility, and population density are speculations to the contributing factors.

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

th

1. $X_{i,1}$ = 1 if the i 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 [31]:
```
# Your Code Here
fit1 = lm(speed ~ time, data = downloading)
summary(fit1)
model.matrix(fit1)
```

```
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
```

A matrix: 48 × 3 of type dbl

	(Intercept)	timeEvening (5 PM)	timeLate Night (12 AM)
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

Explicit equation: $\hat{y}_i = 113.37 + 159.94X_{i,1} + 79.69X_{i,2} + \hat{\varepsilon}_i$, where $\varepsilon \sim N(0, 47.15^2)$

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 [32]:
```
# Your Code Here
# Create a column of new predictor in design matrix:
model_matrix = model.matrix(fit1)
new_matrix = cbind(model_matrix, timeMorning = c(rep(1, 16), rep(0, 32)))
new_matrix
```

A matrix: 48 × 4 of type dbl

	(Intercept)	timeEvening (5 PM)	timeLate Night (12 AM)	timeMorning
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

Looking at the design matrix, adding the new column representing download speed at 7AM creates a problem called "double dipping", where the download speed is counted twice. The original design matrix without the new column considers the download speed at 7AM as the "base case", with the other predictors as variations from that download speed.

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?

Each coefficients in the model represent the difference in means from each factor. The intercept is the mean download speed at 7AM (when all other predictors are 0). The coefficient of $X_{i,1}$ represents the difference in mean download speed between 7AM and 5PM, and similarly for 12AM. Therefore,

1. The difference between the mean download speed at 7AM and that of 5PM is approximately 160.
2. The mean download speed in the morning is approximately 113 seconds.
3. The mean download speed in the evening is approximately 273 seconds.
4. the mean download speed at night is approximately 193 seconds.

In [ ]: