

STA302H1 – Final Project Descriptive Statistics

Danny Chen

August 10, 2021

Import STA302H1 Study Time and COVID Contemplation Time vs. Quiz Performance Dataset

Data Cleaning

First, I'll clean my data.

```
rearranged_data <- sta302_performance_data %>%  
  # Create a new "country" column, which is just "Country" but whose entries are factors.  
  mutate(country = as.factor(Country)) %>%  
  
  # TODO: Replace quiz grades, covid hours, and sta302h1 study hours with their  
  # TODO: median counterparts.  
  
  # Remove the "X" column: it's simply the row number, which isn't very useful.  
  # Remove the "Country" column: column "country" already exists  
  dplyr::select(-X, -Country) %>%  
  
  # Rearrange similar columns side-by-side.  
  relocate(country,  
            COVID.hours..W1., COVID.hours..W2., COVID.hours..W3., COVID.hours..W4.,  
            STA302.hours..W1., STA302.hours..W2., STA302.hours..W3., STA302.hours..W4.,  
            Quiz_1_score, Quiz_2_score, Quiz_3_score, Quiz_4_score)
```

Helper Functions

```
num_column_NAs = function(predictor_variable) {  
  sum(is.na(predictor_variable))  
}
```

```
row_nums_of_NA_columns = function(data, predictor_variable) {  
  which(is.na(predictor_variable))  
}
```

```
rows_with_num_NAs = function(data, num_NAs) {  
  return (rowSums(is.na(data)) == num_NAs)  
}
```

```
row_nums_of_NA_rows = function(data, num_NAs) {  
  return (which(rows_with_num_NAs(data, num_NAs)))  
}
```

```
display_histogram <- function(data, predictor_variable, histogram_title, x_axis_label) {  
  ggplot(data = tibble(data), mapping = aes(x = predictor_variable)) +  
    geom_histogram(col = "black", fill = "red", bins = 30) +  
    labs(title = histogram_title, y = "Frequency", x = x_axis_label) +  
    geom_vline(mapping = aes(xintercept = mean(predictor_variable, na.rm = TRUE)),  
              color = "blue", linetype = "solid") +  
    geom_vline(mapping = aes(xintercept = median(predictor_variable, na.rm = TRUE)),  
              color = "dark green", linetype = "dotted")  
}
```

```
display_boxplot <- function(data, predictor_variable, boxplot_title, y_axis_label) {  
  ggplot(mapping = aes(x = Country, y = predictor_variable)) +  
    geom_boxplot(mapping = aes(x = Country, y = predictor_variable)) +  
    labs(title = boxplot_title, x = "Country", y = y_axis_label)  
}
```

```
get_row_nums_to_exclude <- function(data) {  
  row_nums_with_3_NAs = which(rows_with_num_NAs(data, 3))  
  row_nums_with_4_NAs = which(rows_with_num_NAs(data, 4))  
  row_nums_to_exclude <- union(row_nums_with_3_NAs,  
                               row_nums_with_4_NAs)  
  return (row_nums_to_exclude)  
}
```

```
display_correlation_by_country <- function(country_data) {  
  colnames(country_data) <- c("W1COV", "W2COV", "W3COV", "W4COV",  
                             "W1302", "W2302", "W3302", "W4302",  
                             "Q1", "Q2", "Q3", "Q4")  
  round(cor(country_data, use = "pairwise.complete.obs", method = "pearson"), 2)  
}
```

Special Tables

Rows With At Least One NA

Rows with at least one NA deserve closer examination.

Some of the rows might only have 1 - 2 NAs and are therefore salvageable, which is OK.

Other rows may contain 3 or more NAs, and might indicate students who have dropped STA302H1. We'd like to exclude them from our analysis.

Here are the number of rows with 0 - 4 NAs.

```
##   nrows_0_NAs nrows_1_NAs nrows_2_NAs nrows_3_NAs nrows_4_NAs
## 1           143           9           16           19           1
```

Columns with NAs

```
##   week1_covid week2_covid week3_covid week4_covid
## 1           26           22           21           40
```

```
##   week1_sta302 week2_sta302 week3_sta302 week4_sta302
## 1           26           22           20           40
```

```
##   quiz1_score quiz2_score quiz3_score quiz4_score
## 1           13           36           31           34
```

Number of Missed Quizzes

```
##   miss_0_quizzes miss_1_quizzes miss_2_quizzes miss_3_quizzes miss_4_quizzes
## 1           176           20           3           24           4
```

Who to Exclude from the Dataset?

Identify rows with at least 3 missing quiz marks. These indicate students who have dropped STA302H1, and who should be excluded from the final data.

Notice that we didn't check the number of NAs for country of origin, COVID hours, and STA302H1 hours, since some students either forgot or abstained. So there's no reason to exclude these students from our final dataset.

```
row_nums_to_exclude <- get_row_nums_to_exclude(quiz_grades)
remaining_data = rearranged_data[-row_nums_to_exclude,]
```

Rows with Mistyped Columns

Rows whose columns are mis-typed may need to be corrected via imputation.

```
rows_with_mistyped_columns = remaining_data[c(38, 83, 84, 117),]  
# row 83: Country -> "canada" -- DONE  
# row 84: Country -> "canada" -- DONE  
  
# row 117: COVID.hours..W4. -> 0.5 hours -- DONE  
  
# row 38: STA302.hours..W3. -> 5.5<U+00A0> -- DONE  
# row 117: STA302.hours..W4. -> 7.5 hours -- DONE
```

```
# library(janitor)  
# use it to clean up data.
```

Rows Without Country Entry

Taking out the country column can come in handy for functions like `cor()` where factors aren't allowed.

```
rows_with_no_country = remaining_data %>%  
  dplyr::select(-country)
```

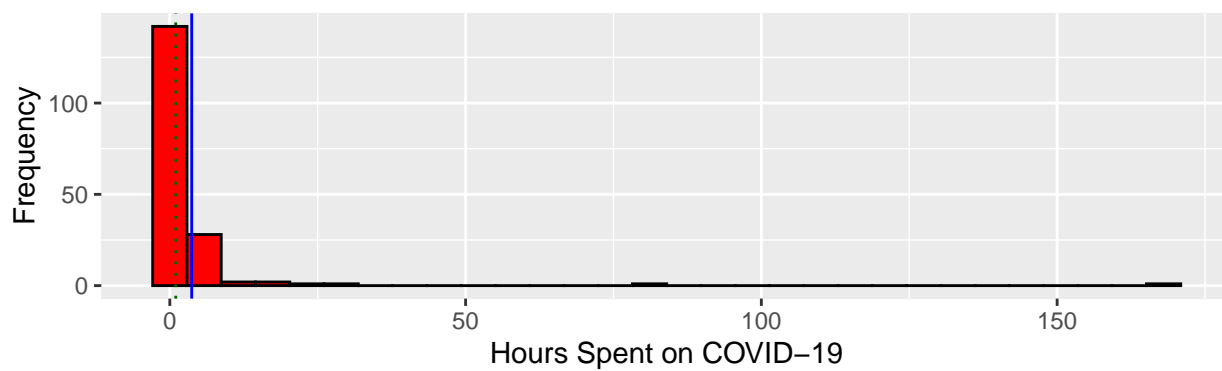
Rows Filtered by Country

This is useful if we want data for individual countries.
Only the first and last code snippets are shown.

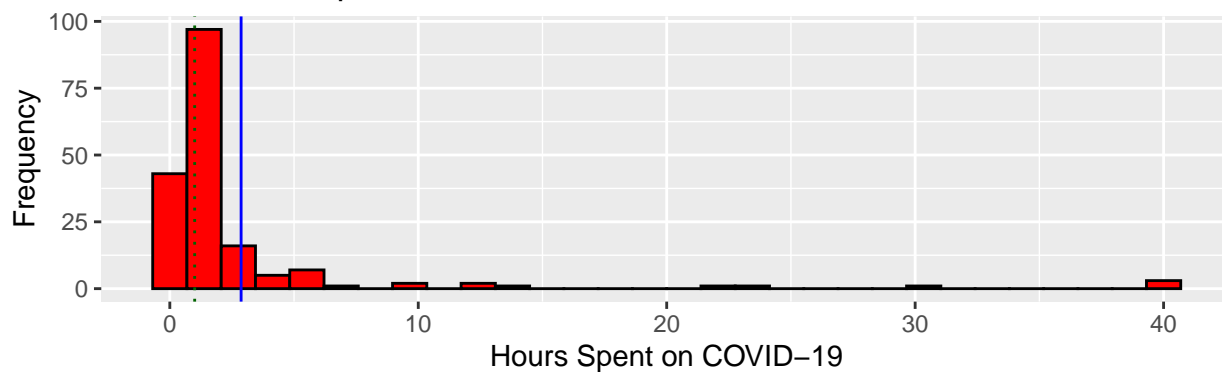
```
canada <- remaining_data %>%  
  filter(as.character(country) == "Canada") %>%  
  dplyr::select(-country)  
  
unknown <- remaining_data %>%  
  filter(is.na(as.character(country))) %>%  
  dplyr::select(-country)
```

```
##           Country  
## Canada         97  
## China          63  
## India           2  
## Japan           1  
## Mongolia        1  
## Pakistan        3  
## Singapore       2  
## South_Korea     2  
## Taiwan          3  
## UAE             2  
## USA             2  
## Unknown        21
```

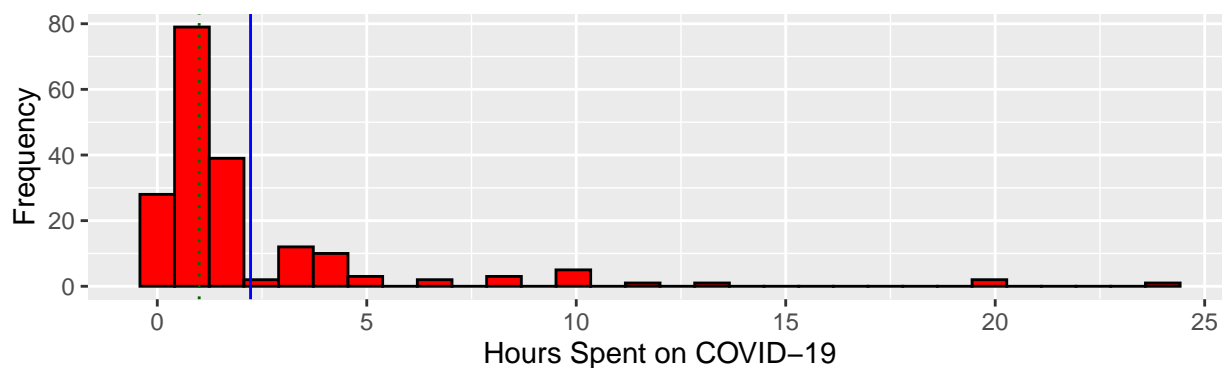
Week 1 Time Spent on COVID-19



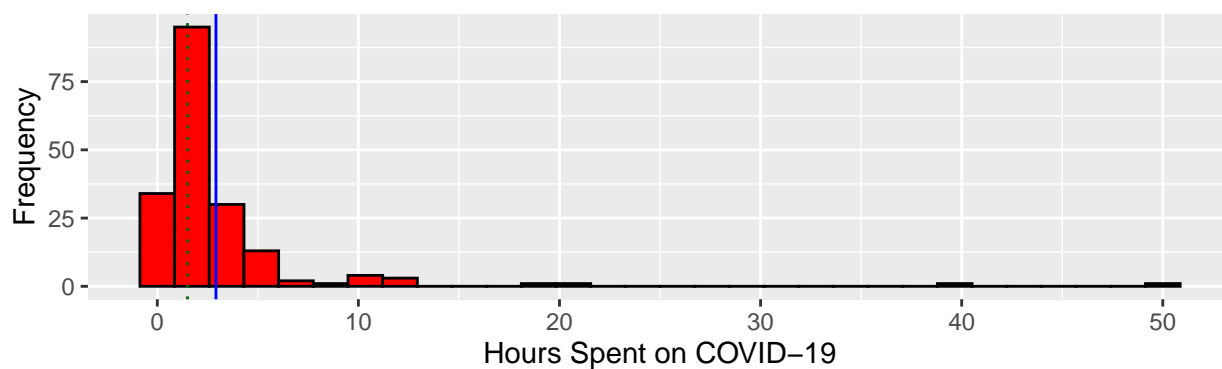
Week 2 Time Spent on COVID-19

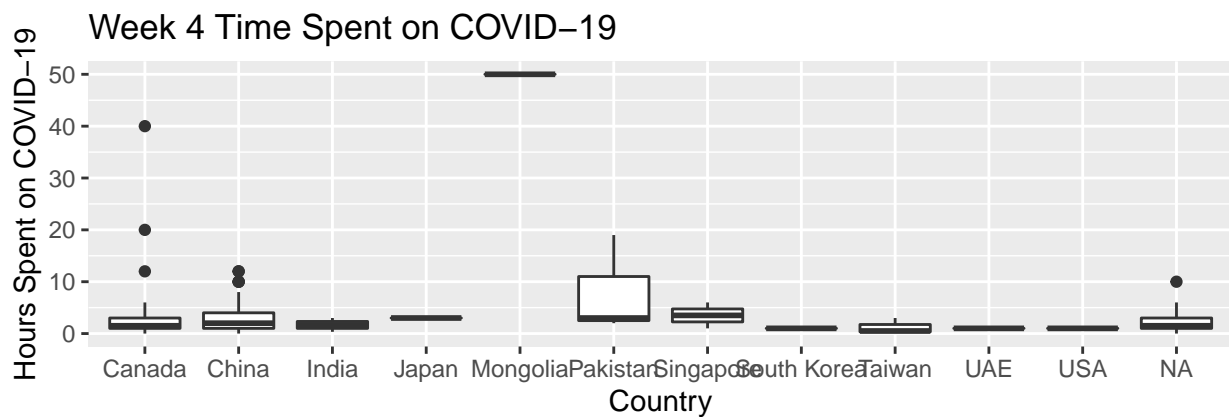
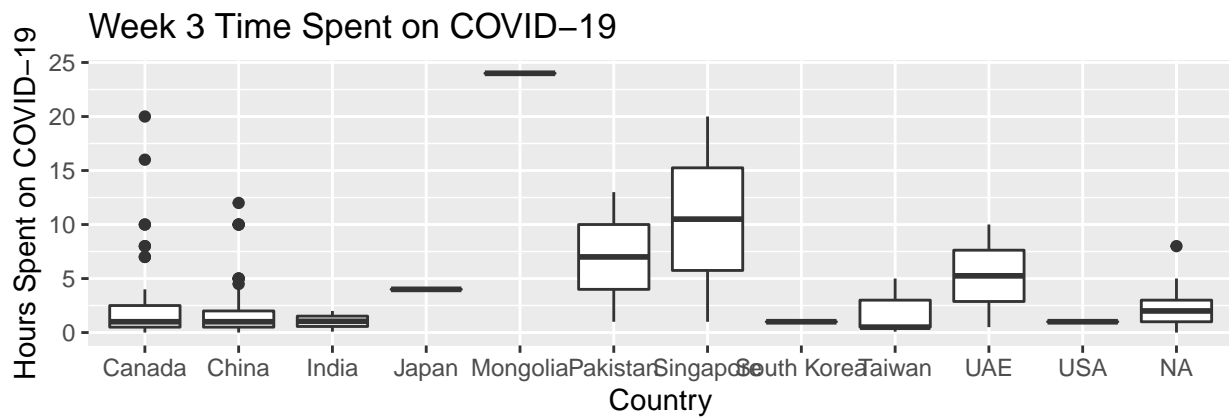
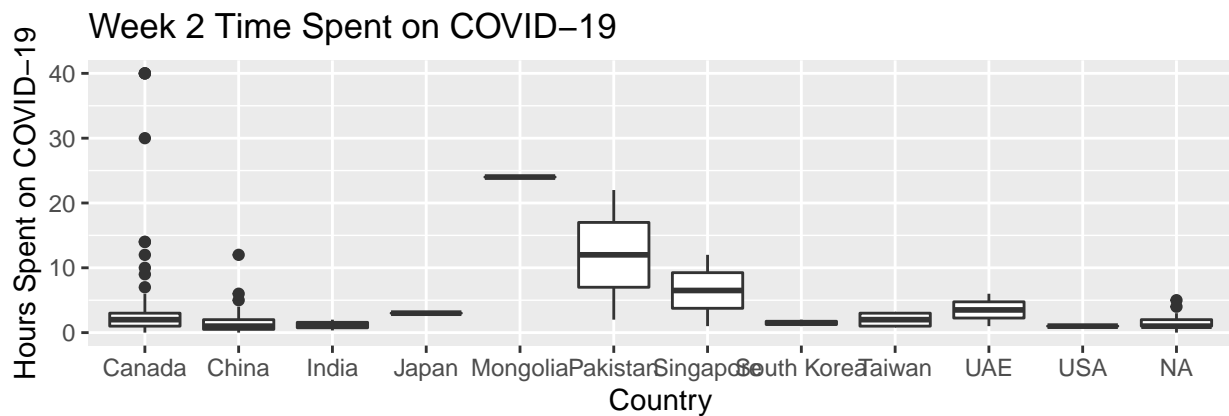
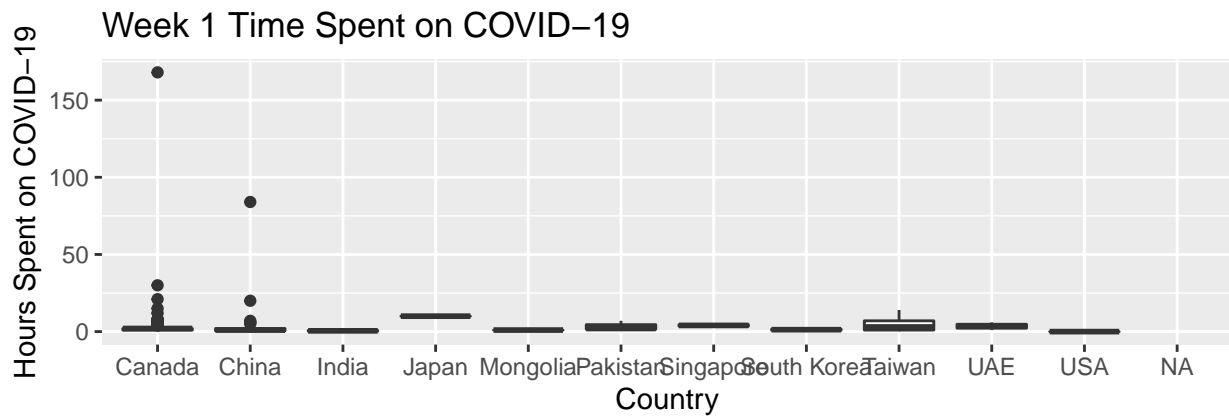


Week 3 Time Spent on COVID-19

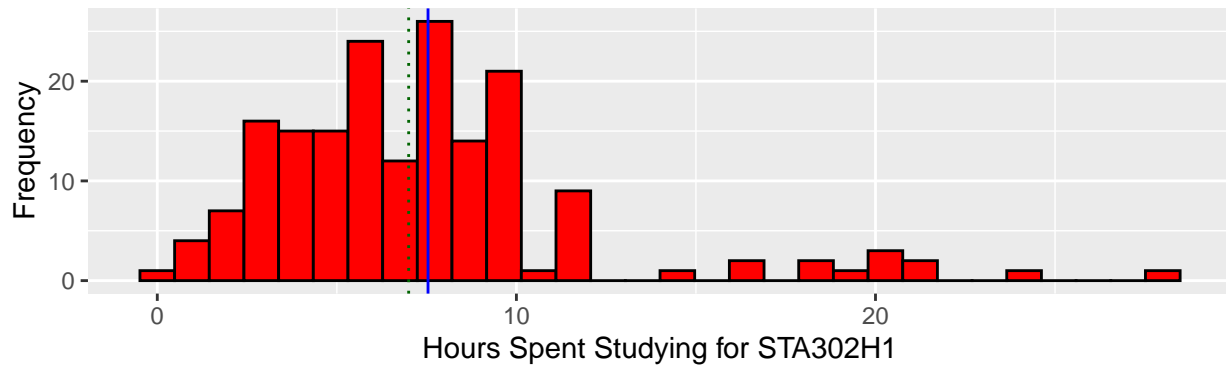


Week 4 Time Spent on COVID-19

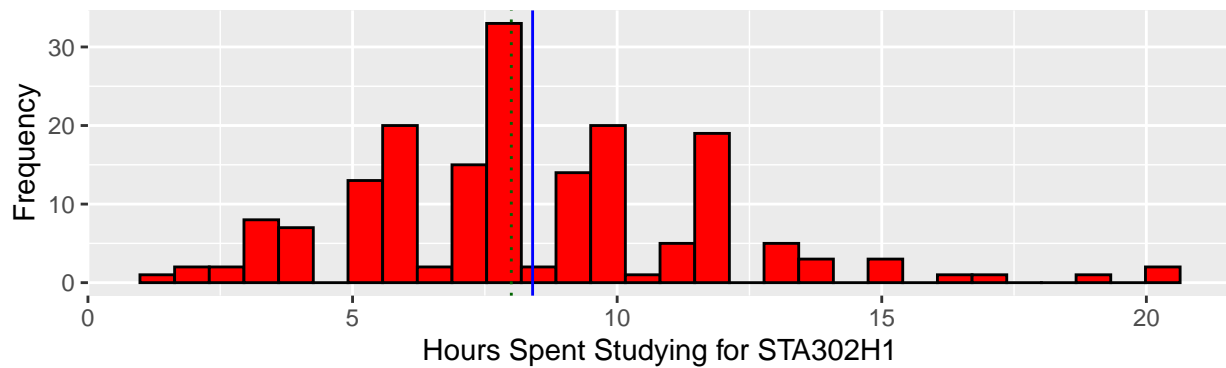




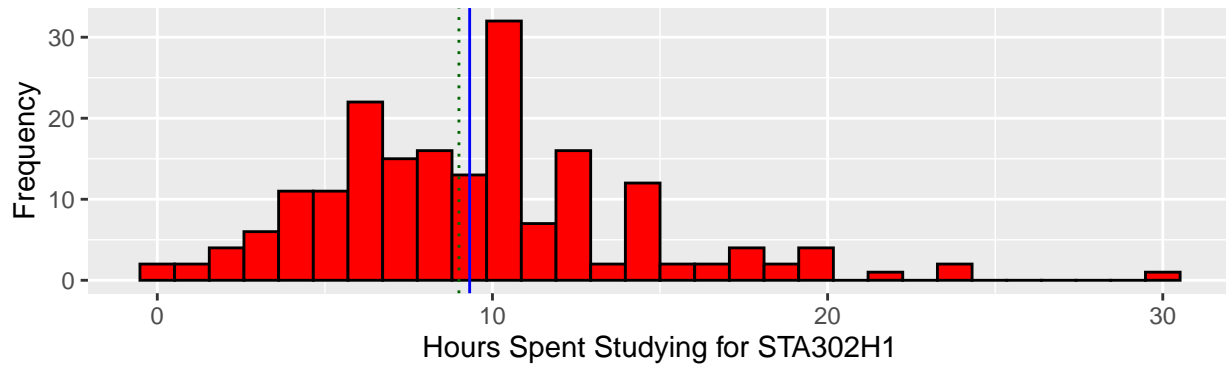
Week 1 Time Spent Studying for STA302H1



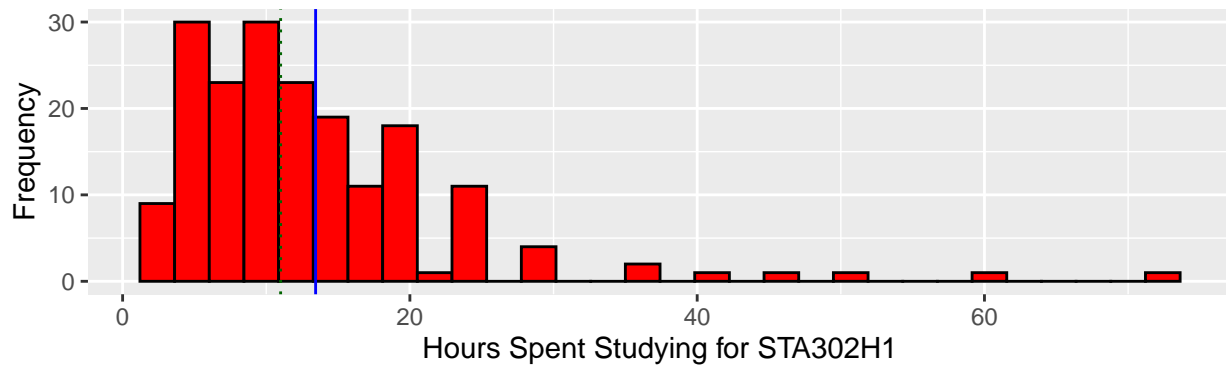
Week 2 Time Spent Studying for STA302H1



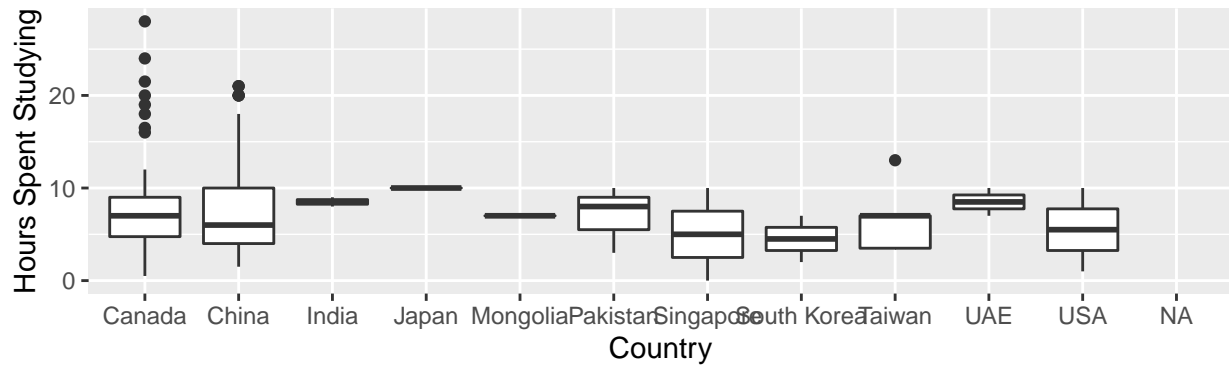
Week 3 Time Spent Studying for STA302H1



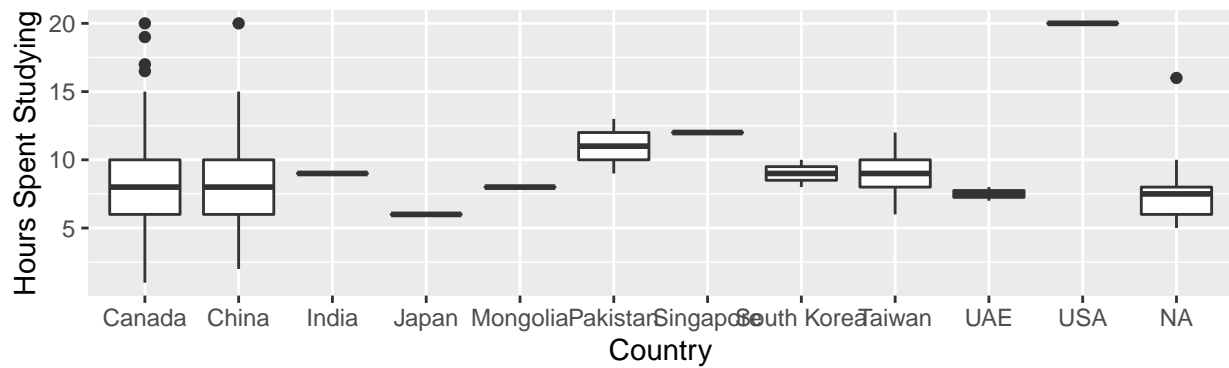
Week 4 Time Spent Studying for STA302H1



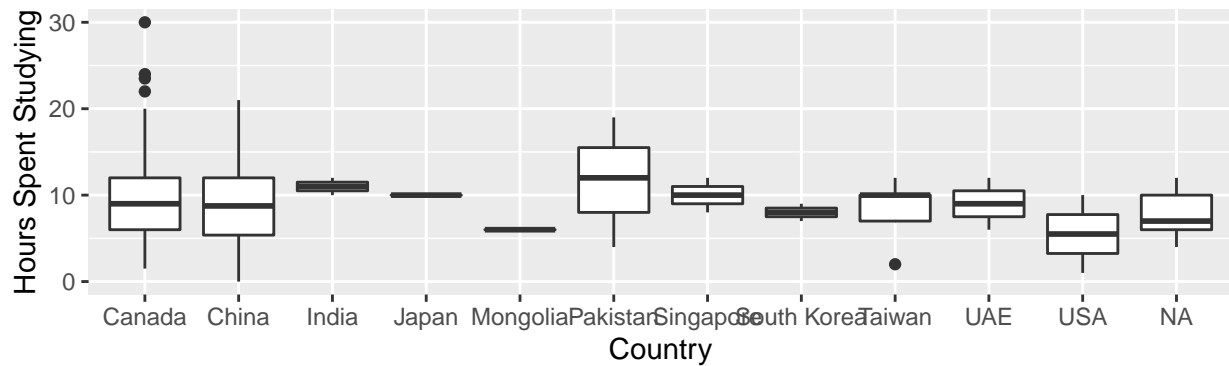
Week 1 Time Spent Studying For STA302H1



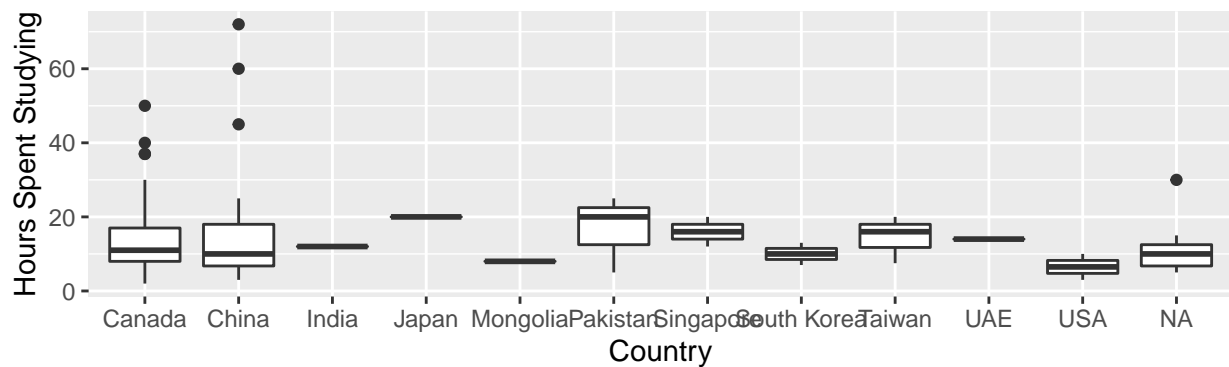
Week 2 Time Spent Studying For STA302H1

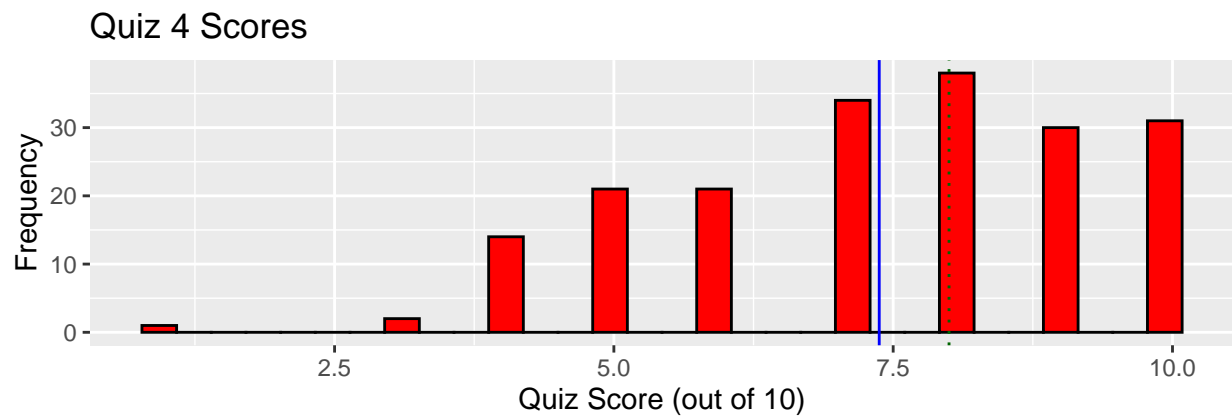
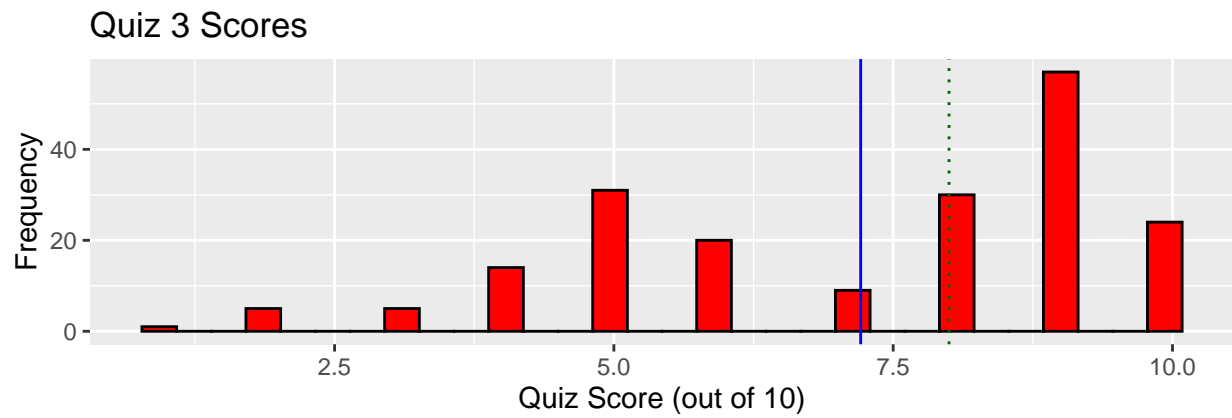
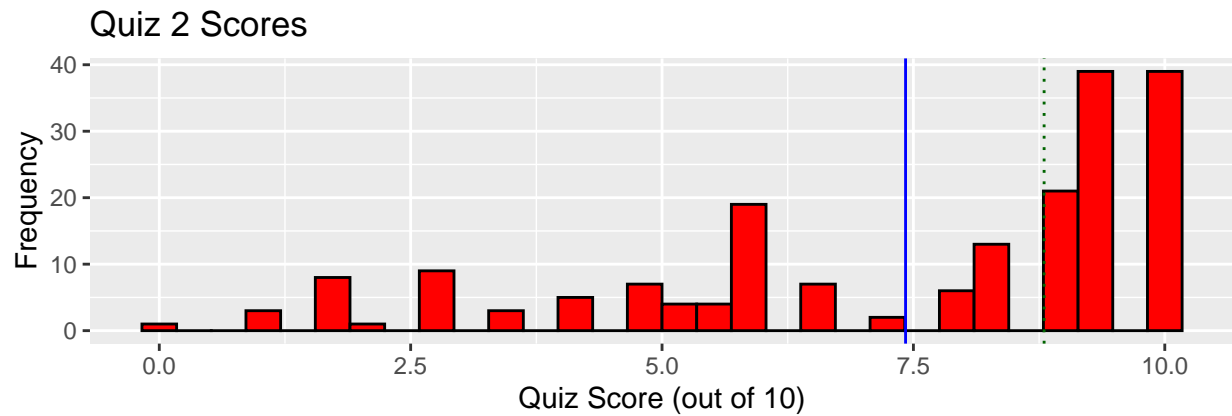
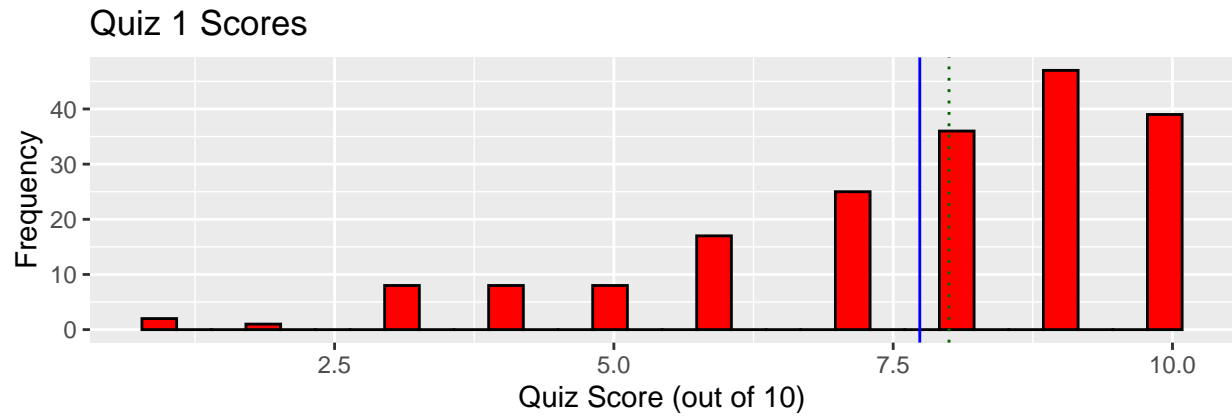


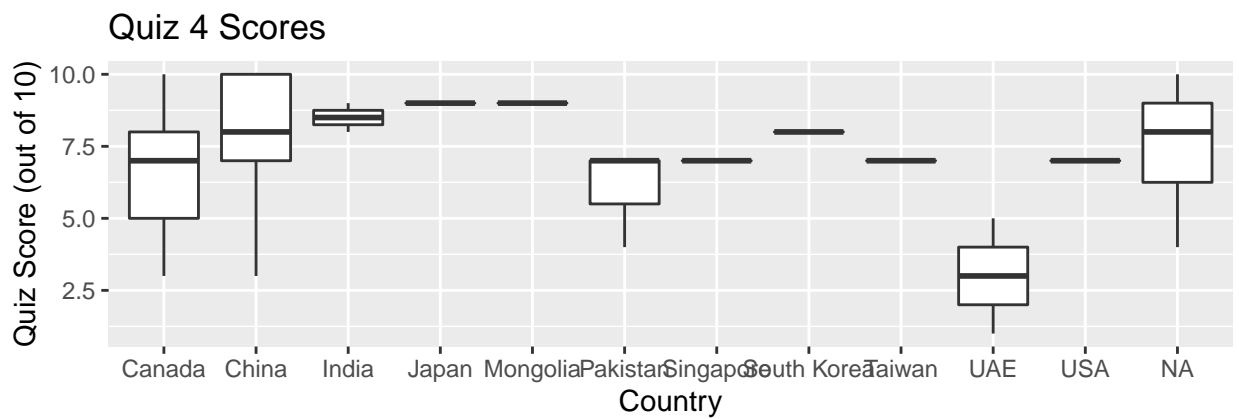
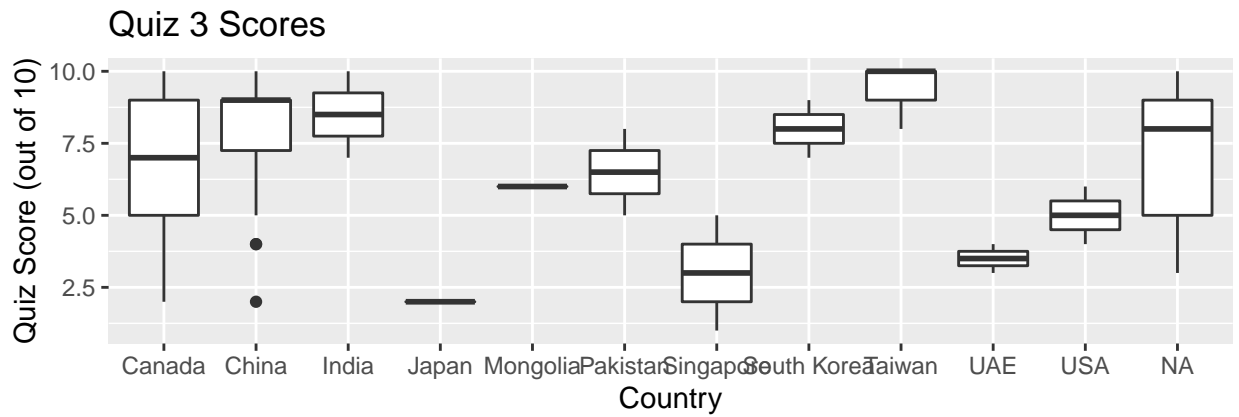
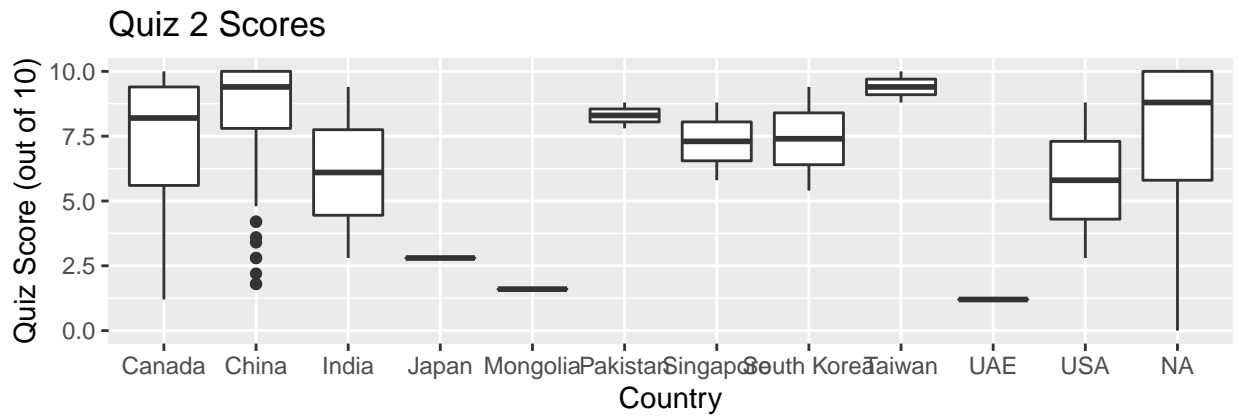
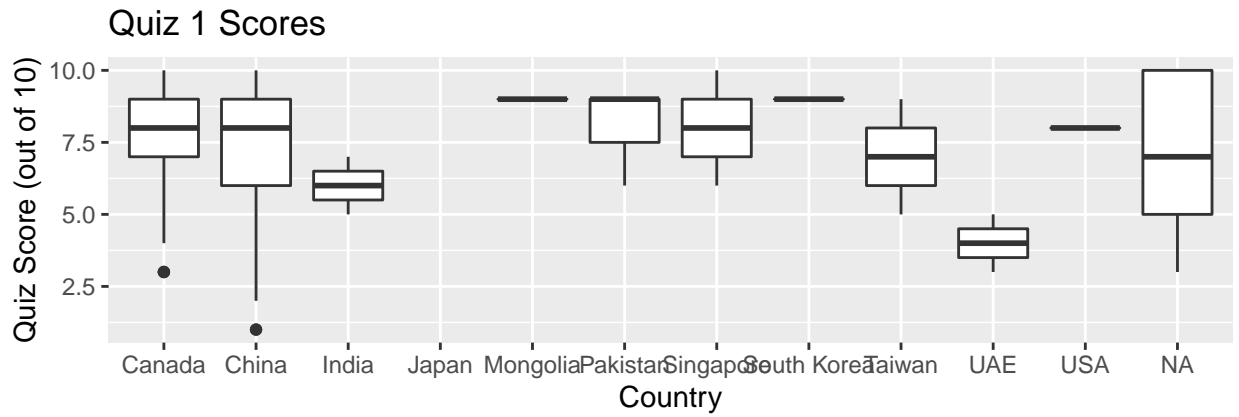
Week 3 Time Spent Studying



Week 4 Time Spent Studying







5-Number Summary Statistics

```
summary(remaining_data$COVID.hours..W1.)
```

##	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	NA's
##	0.0	1.0	1.0	3.7	2.0	168.0	21

```
summary(remaining_data$COVID.hours..W2.)
```

##	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	NA's
##	0.000	1.000	1.000	2.869	2.000	40.000	19

```
summary(remaining_data$COVID.hours..W3.)
```

##	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	NA's
##	0.000	0.500	1.000	2.227	2.000	24.000	11

```
summary(remaining_data$COVID.hours..W4.)
```

##	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	NA's
##	0.000	1.000	1.500	2.917	3.000	50.000	13

```
summary(remaining_data$STA302.hours..W1.)
```

##	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	NA's
##	0.000	5.000	7.000	7.539	9.000	28.000	21

```
summary(remaining_data$STA302.hours..W2.)
```

##	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	NA's
##	1.000	6.000	8.000	8.403	10.000	20.000	19

```
summary(remaining_data$STA302.hours..W3.)
```

##	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	NA's
##	0.00	6.00	9.00	9.32	12.00	30.00	10

```
summary(remaining_data$STA302.hours..W4.)
```

##	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	NA's
##	2.00	7.00	11.00	13.44	16.00	72.00	13

```
summary(remaining_data$Quiz_1_score)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.   NA's  
##    1.000   7.000   8.000   7.738   9.000  10.000     8
```

```
summary(remaining_data$Quiz_2_score)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.   NA's  
##    0.000   5.800   8.800   7.422   9.400  10.000     8
```

```
summary(remaining_data$Quiz_3_score)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.   NA's  
##    1.000   5.000   8.000   7.209   9.000  10.000     3
```

```
summary(remaining_data$Quiz_4_score)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.   NA's  
##    1.000   6.000   8.000   7.375   9.000  10.000     7
```

Full Model

```
# single variable per term = additive model
first_model = lm(
  quiz4 ~
    quiz1 # scatterplot seems to have no relationship
  + quiz2 # scatterplot seems to have no relationship
  + quiz3 # scatterplot looks more linear
  + covid1 # must add this linear term b/c i have a quadratic term
  + I(covid1 ^ 2) # scatterplot looks more quadratic
  + covid2 # must add this linear term b/c i have a quadratic term
  + I(covid2 ^ 2) # scatterplot looks more quadratic
  + covid3
  # + I(covid3 ^ 2) # scatterplot looks less quadratic
  + covid4 # must add this linear term b/c i have a quadratic term
  + I(covid4 ^ 2) # scatterplot looks more quadratic
  + I(covid1 * covid2) # first impressions from correlation matrix
  + I(covid2 * covid3) # correlation = 0.67
  + I(covid2 * covid4) # discard: correlation = 0.71
  + I(covid3 * covid4) # correlation = 0.72
  + I(study1 * study2) # correlation = 0.61
  + I(study1 * study3) # correlation = 0.58
  + I(study2 * study3) # correlation = 0.70
  + I(study3 * study4) # correlation = 0.62
  + country # for simplicity, but backwards process shows this term is not significant
)
summary(first_model)
```

```
stepAIC(first_model, direction = "forward")$anova
```

```
stepAIC(first_model, direction = "backward")$anova
```

```
stepAIC(first_model, direction = "both")$anova
```

Final Model

```
final_model = lm(
  quiz4 ~ quiz3
  + I(covid1 ^ 2)    # don't remove, else all other terms become insignificant
  + I(covid1 * covid2)
  + I(covid2 * covid3) # don't remove, else all other terms become insignificant
  + I(study1 * study2)
  + I(study1 * study3) # maybe don't remove?
  + I(study2 * study3)
  + I(study3 * study4)
)
summary(final_model)
```

```
stepAIC(final_model, direction = "forward")$anova
```

```
stepAIC(final_model, direction = "backward")$anova
```

```
stepAIC(final_model, direction = "both")$anova
```

Final Model with Some Terms I Pruned Myself

```
# I decide to remove more terms for simplicity.
third_model = lm(
  quiz4 ~ quiz3
  + I(covid1 ^ 2) # this lone quadratic term add a lot of complexity for negligible change in R2 and
  + I(covid1 * covid2) + I(covid2 * covid3) # these terms alone add complexity -- harder to interpret
  + I(study1 * study2)
  + I(study1 * study3) # make weeks consecutive: "want to see correlation from week to week", rather than
  + I(study2 * study3)
  + I(study3 * study4)
)
summary(third_model)
```

```
# Doing stepAIC on a well-fitted model produces the same model.
# The model is already in a "steady state."
stepAIC(third_model, direction = "both")$anova
```

```
stepAIC(third_model, direction = "forward")$anova
```

```
stepAIC(third_model, direction = "backward")$anova
```

Simplistic Model

```
fourth_model = lm(quiz4 ~ quiz3)
summary(fourth_model)
```

```
stepAIC(fourth_model, direction = "forward")$anova
```

```
stepAIC(fourth_model, direction = "backward")$anova
```

```
stepAIC(fourth_model, direction = "both")$anova
```

Linear Model Only

```
additive_model = lm(
  quiz4 ~ quiz1 + quiz2 + quiz3
  + covid1 + covid2 + covid3 + covid4
  + study1 + study2 + study3 + study4
  + country
)
summary(additive_model)
```

```
stepAIC(additive_model, direction = "forward")$anova
```

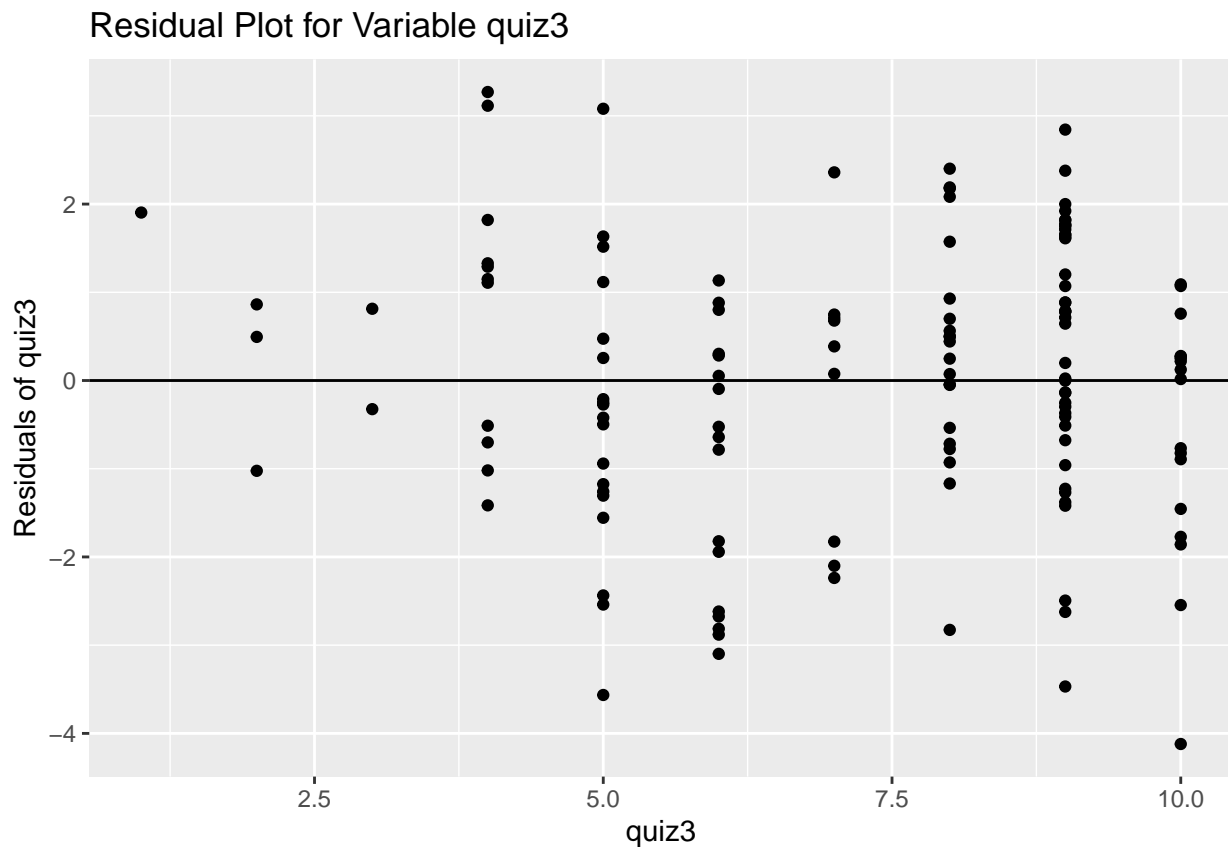
```
stepAIC(additive_model, direction = "backward")$anova
```

```
stepAIC(additive_model, direction = "both")$anova
```

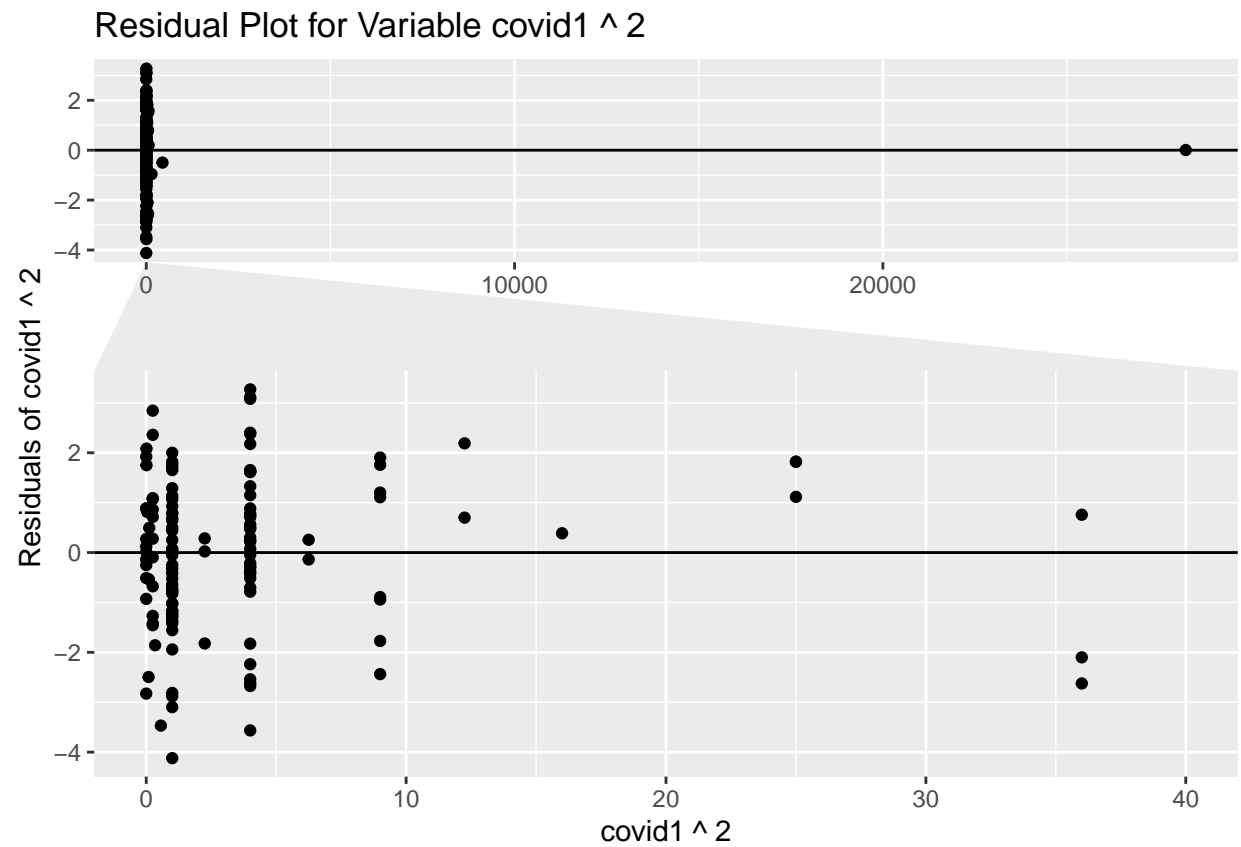
```
display_residual_plot <- function(data, model, predictor_variable, predictor_variable_name) {
  fit = fitted(model)
  residuals = resid(model)
  ggplot(data = data, aes(x = predictor_variable, y = residuals)) +
    geom_point() +
    geom_hline(yintercept = 0) +
    labs(title = paste0("Residual Plot for Variable ", predictor_variable_name),
         x = predictor_variable_name, y = paste0("Residuals of ", predictor_variable_name))
}
```

residual plot

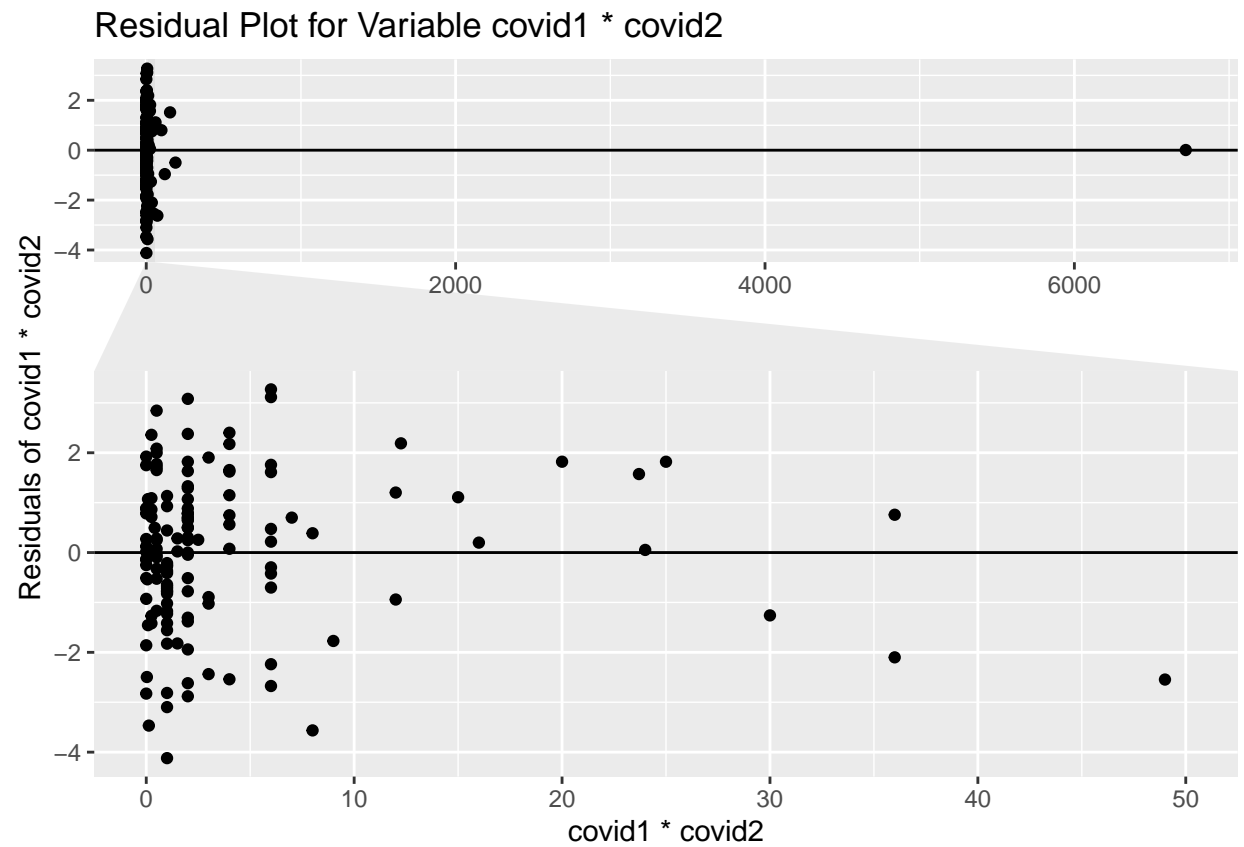
```
display_residual_plot(remaining_data_no_NAs, final_model, quiz3, "quiz3")
```



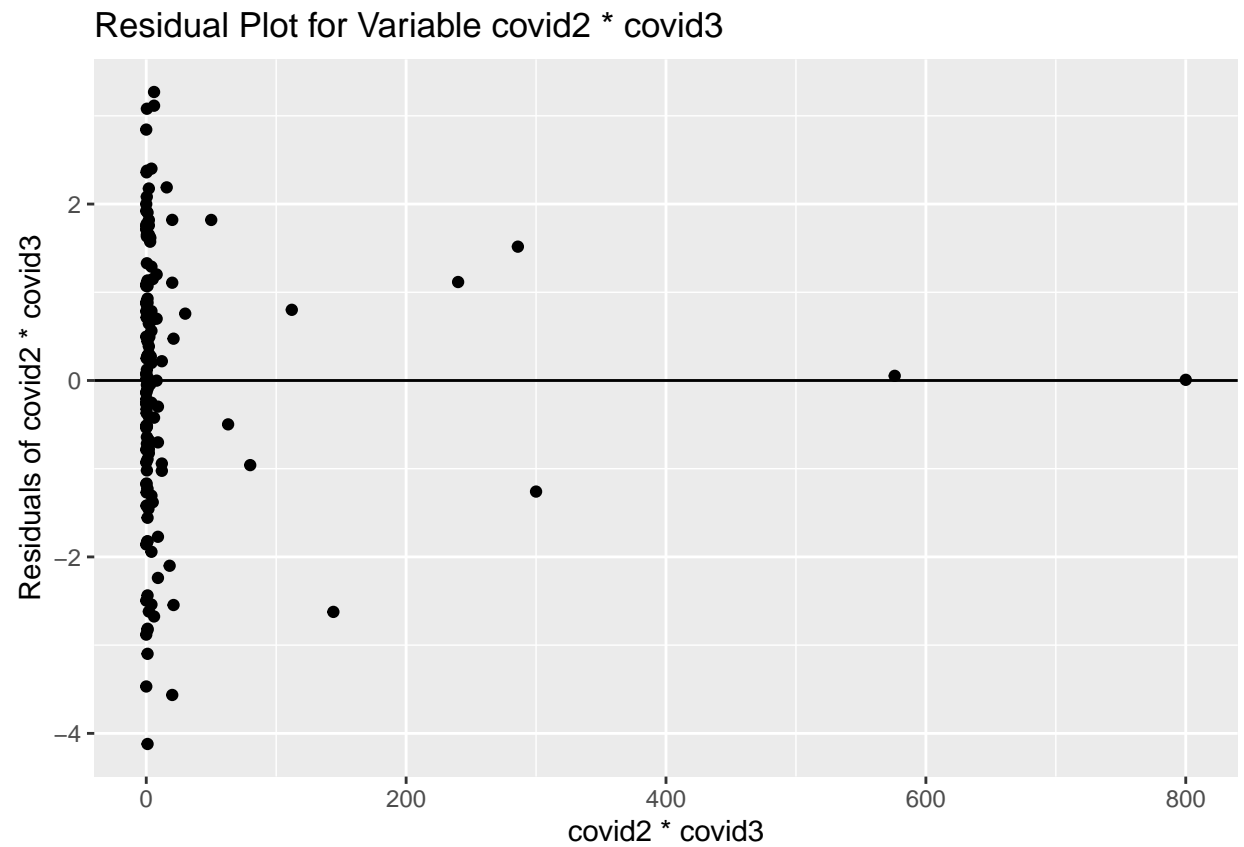
```
library(ggforce) # zooming in on plots
display_residual_plot(remaining_data_no_NAs, final_model, covid1 ~ 2, "covid1 ~ 2") +
  facet_zoom(xlim = c(0, 40)) # try upper limit = 70
```

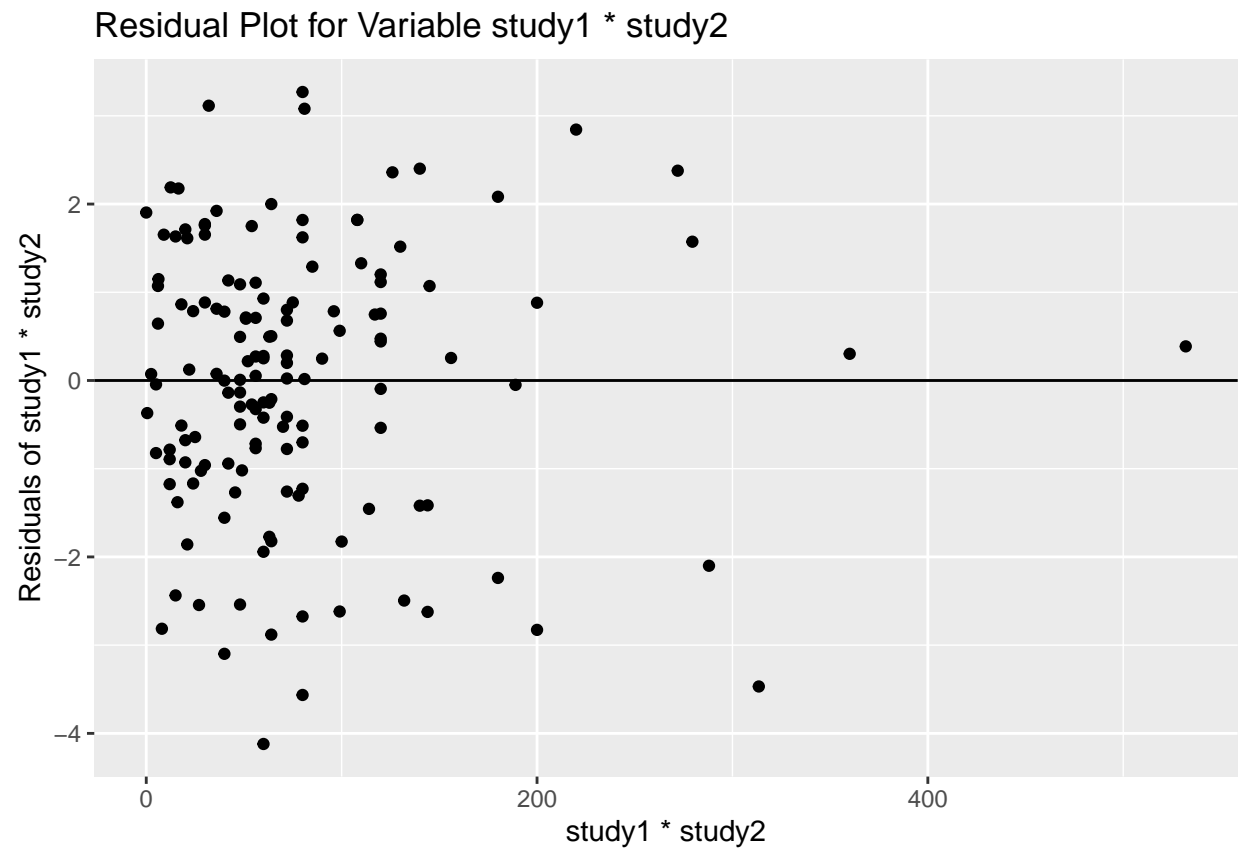
```
display_residual_plot(remaining_data_no_NAs, final_model, covid1 * covid2, "covid1 * covid2") +
  facet_zoom(xlim = c(0, 50)) # try upper limit = 100
```



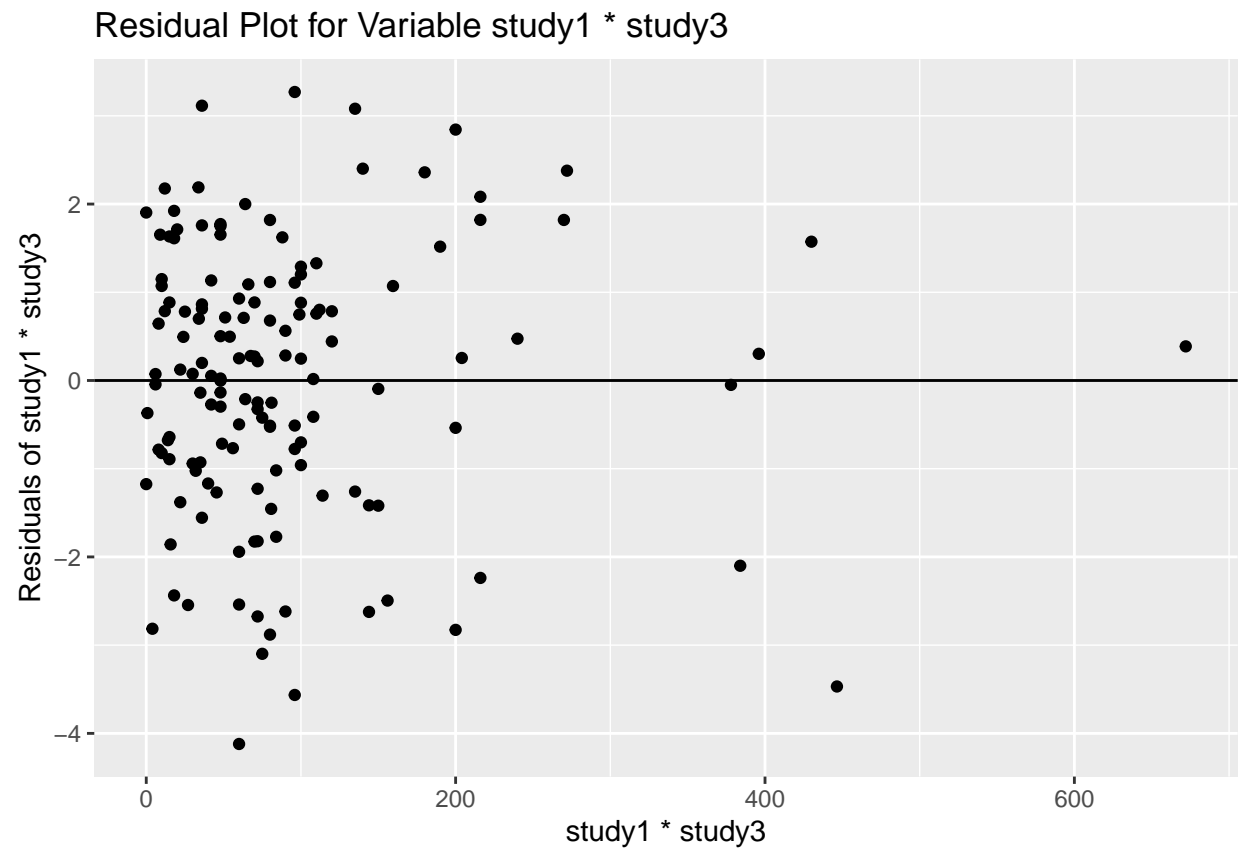
```
display_residual_plot(remaining_data_no_NAs, final_model, covid2 * covid3, "covid2 * covid3")
```



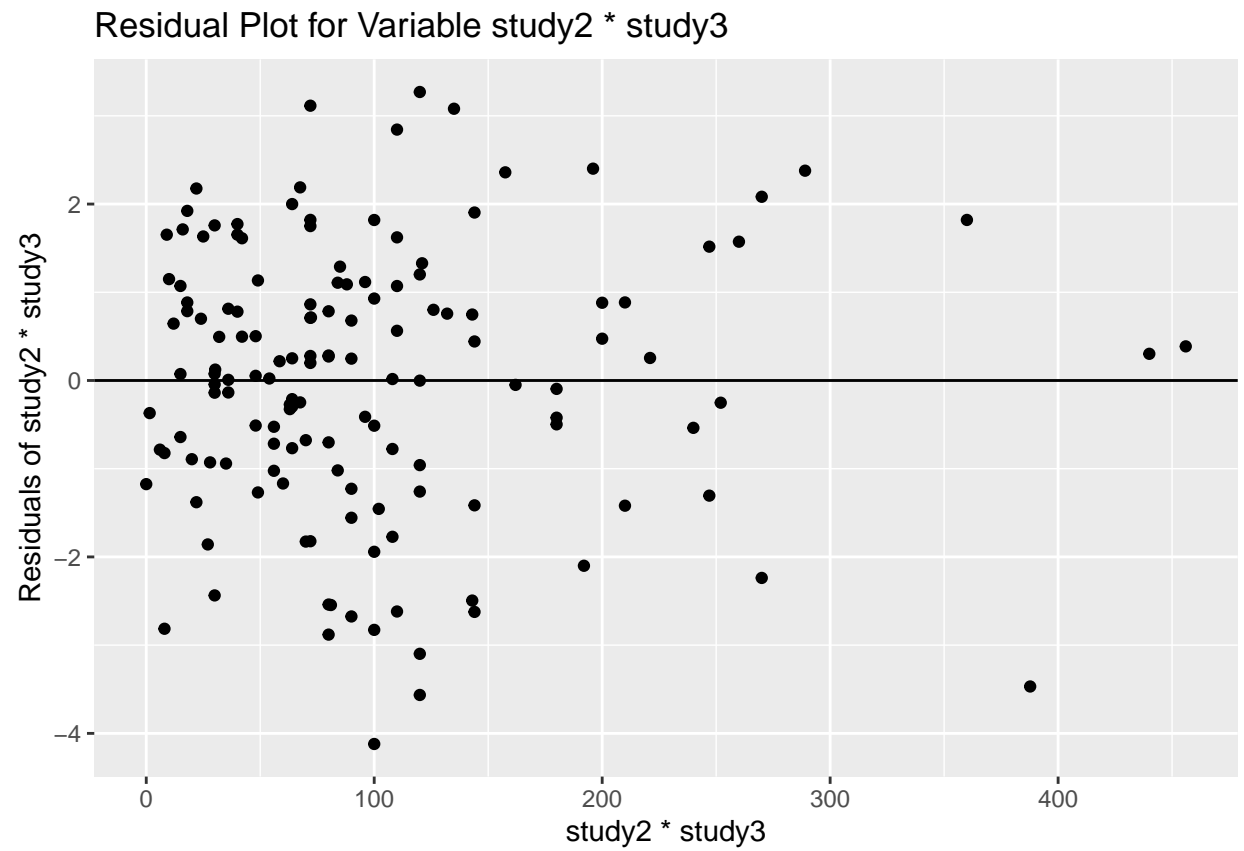
```
display_residual_plot(remaining_data_no_NAs, final_model, study1 * study2, "study1 * study2")
```



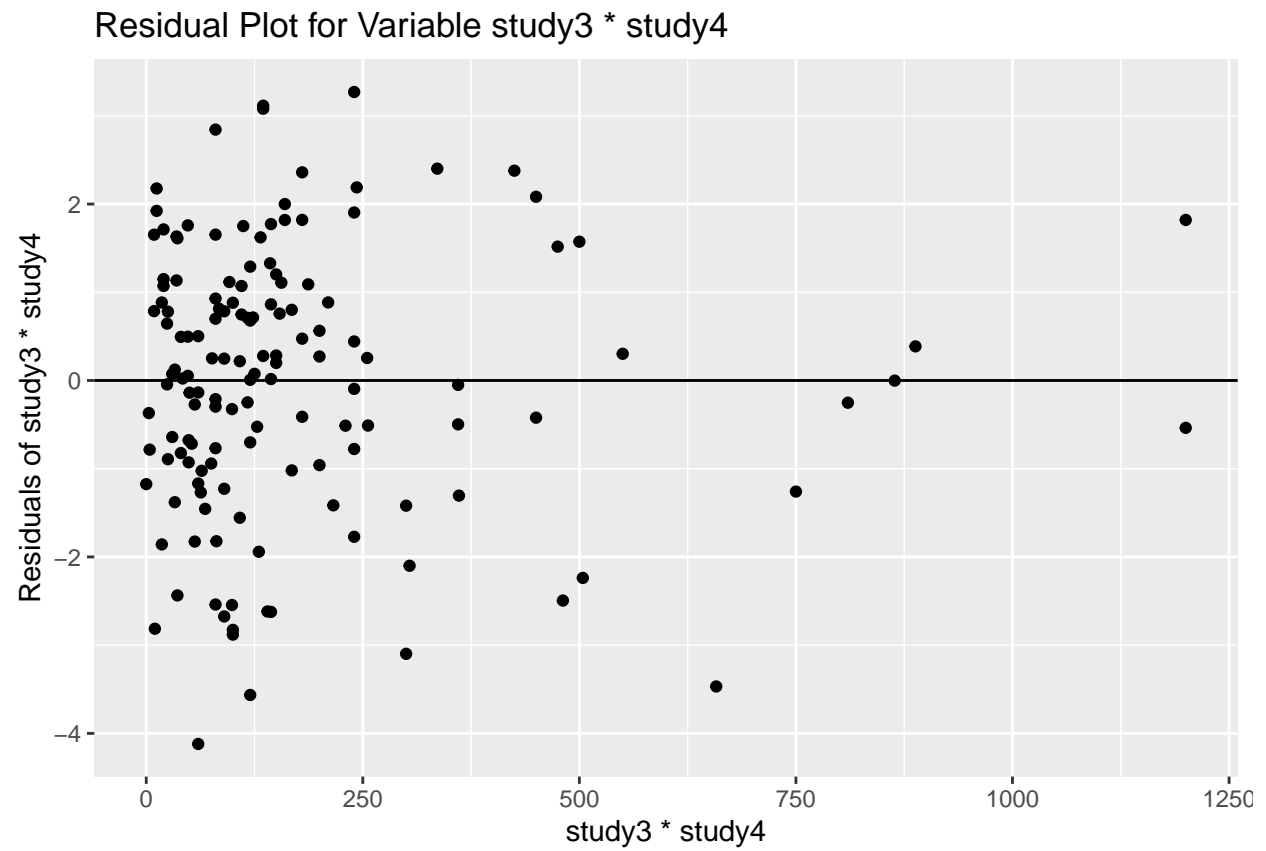
```
display_residual_plot(remaining_data_no_NAs, final_model, study1 * study3, "study1 * study3")
```



```
display_residual_plot(remaining_data_no_NAs, final_model, study2 * study3, "study2 * study3")
```

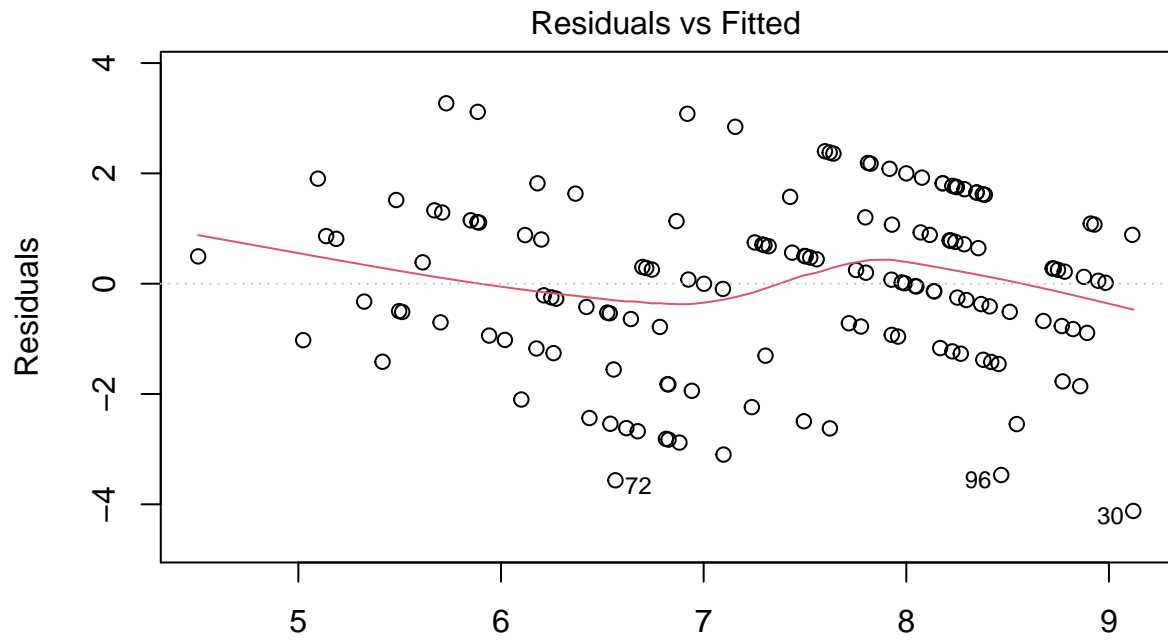


```
display_residual_plot(remaining_data_no_NAs, final_model, study3 * study4, "study3 * study4")
```

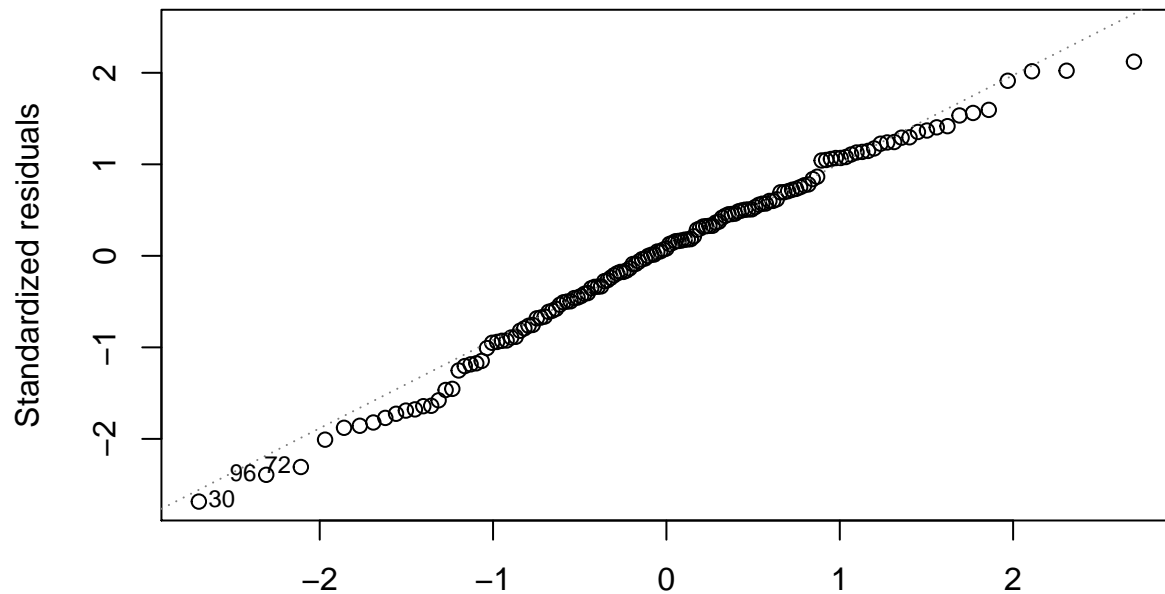


residual vs. fit, qqplot, scale-location, and residual vs. leverage

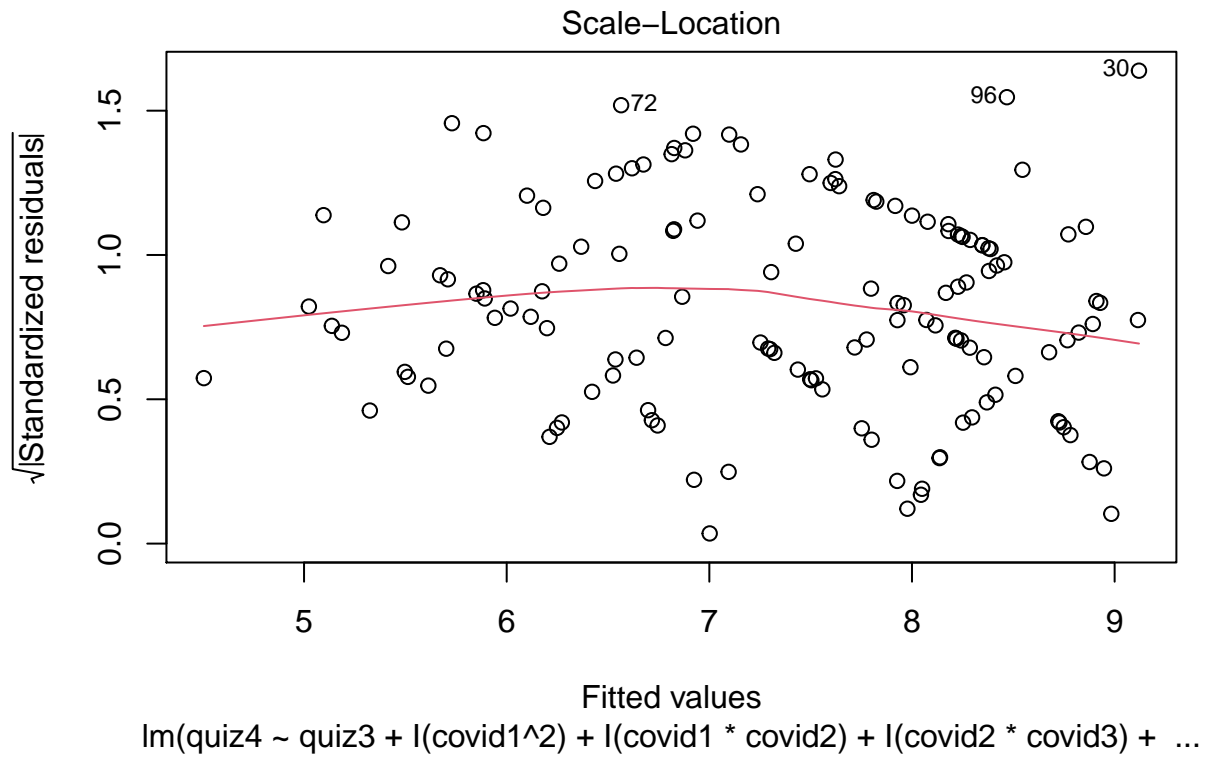
```
plot(final_model)
```



Fitted values
 $\text{lm}(\text{quiz4} \sim \text{quiz3} + \text{I}(\text{covid1}^2) + \text{I}(\text{covid1} * \text{covid2}) + \text{I}(\text{covid2} * \text{covid3}) + \dots$
 Normal Q-Q

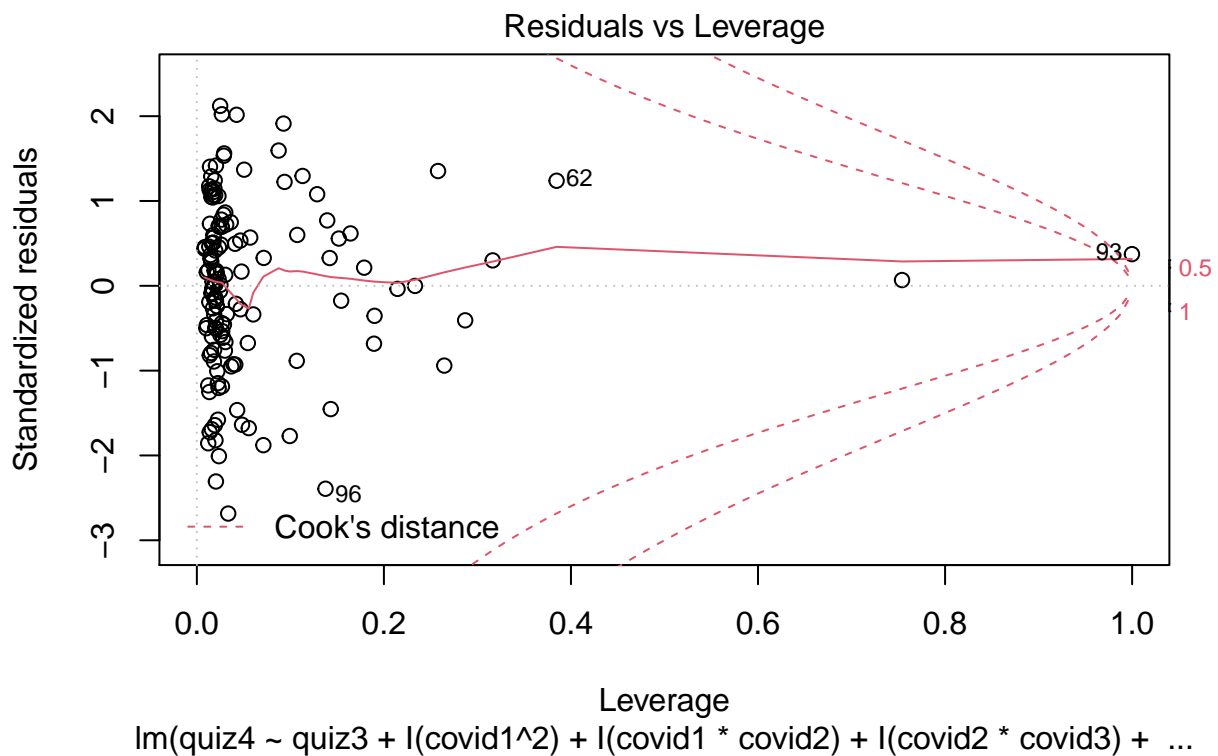


Theoretical Quantiles
 $\text{lm}(\text{quiz4} \sim \text{quiz3} + \text{I}(\text{covid1}^2) + \text{I}(\text{covid1} * \text{covid2}) + \text{I}(\text{covid2} * \text{covid3}) + \dots$

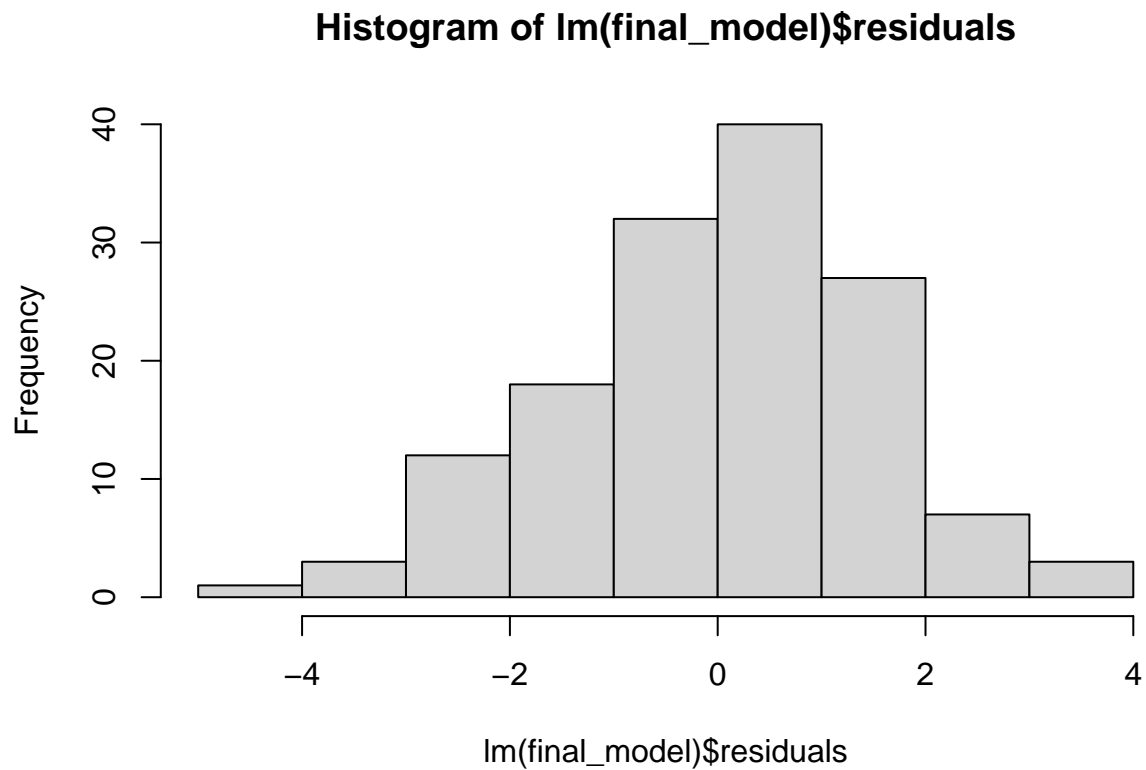


```
## Warning in sqrt(crit * p * (1 - hh)/hh): NaNs produced
```

```
## Warning in sqrt(crit * p * (1 - hh)/hh): NaNs produced
```



```
hist(lm(final_model)$residuals)
```



```
mean(lm(final_model)$residuals)
```

```
## [1] -1.651627e-17
```

```
median(lm(final_model)$residuals)
```

```
## [1] 0.07546203
```

Try Predicting on the Fitted Values

```
predicted_values = predict(final_model)
actual_values = remaining_data_no_NAs$Quiz_4_score
```

```
mean(predicted_values - actual_values)
```

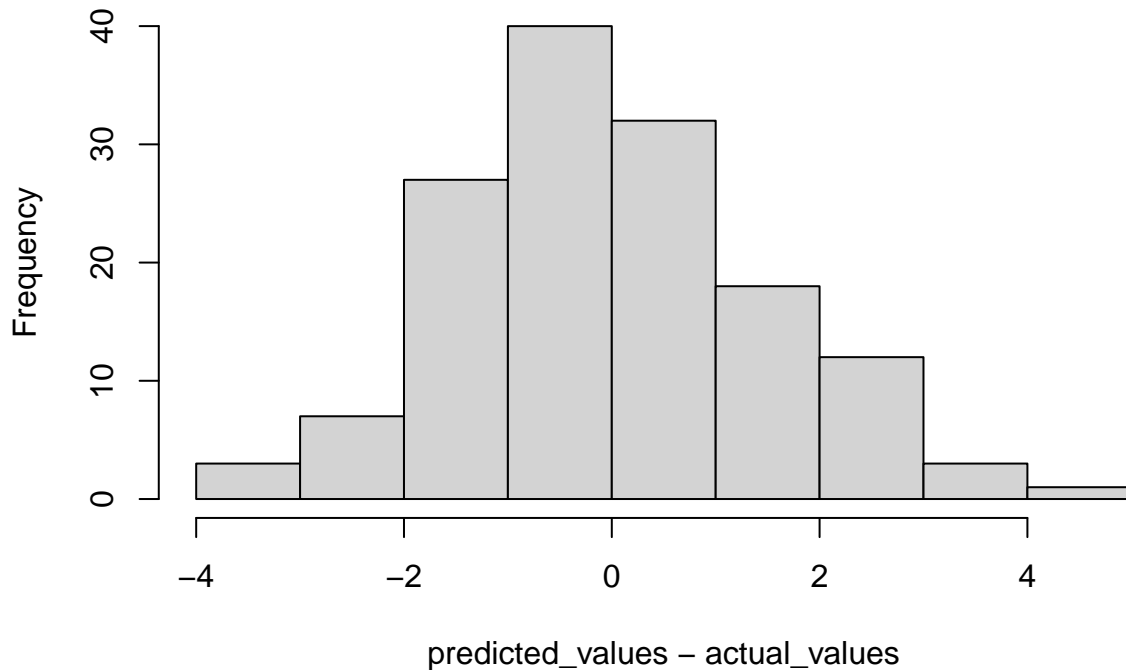
```
## [1] 5.664449e-15
```

```
median(predicted_values - actual_values)
```

```
## [1] -0.07546203
```

```
hist(predicted_values - actual_values)
```

Histogram of predicted_values – actual_values



```
t.test(predicted_values - actual_values)
```

```
##  
## One Sample t-test  
##  
## data: predicted_values - actual_values  
## t = 4.467e-14, df = 142, p-value = 1  
## alternative hypothesis: true mean is not equal to 0  
## 95 percent confidence interval:  
## -0.250671 0.250671  
## sample estimates:  
## mean of x  
## 5.664449e-15
```

50/50 Training/Testing

partitioning

```
# https://stackoverflow.com/questions/17200114/how-to-split-data-into-training-testing-sets-using-sample  
  
set.seed(888)  
library(caTools)  
sample = sample.split(remaining_data_no_NAs, SplitRatio = 0.55)
```

```
training_data = subset(remaining_data_no_NAs, sample == TRUE)
testing_data = subset(remaining_data_no_NAs, sample == FALSE)
```

training

```
training_data_no_NAs = na.omit(training_data)

quiz1 = training_data_no_NAs$Quiz_1_score
quiz2 = training_data_no_NAs$Quiz_2_score
quiz3 = training_data_no_NAs$Quiz_3_score
quiz4 = training_data_no_NAs$Quiz_4_score

covid1 = training_data_no_NAs$COVID.hours..W1.
covid2 = training_data_no_NAs$COVID.hours..W2.
covid3 = training_data_no_NAs$COVID.hours..W3.
covid4 = training_data_no_NAs$COVID.hours..W4.

study1 = training_data_no_NAs$STA302.hours..W1.
study2 = training_data_no_NAs$STA302.hours..W2.
study3 = training_data_no_NAs$STA302.hours..W3.
study4 = training_data_no_NAs$STA302.hours..W4.

country = training_data_no_NAs$country
```

testing

```
testing_data_no_NAs = na.omit(testing_data)

quiz1 = testing_data_no_NAs$Quiz_1_score
quiz2 = testing_data_no_NAs$Quiz_2_score
quiz3 = testing_data_no_NAs$Quiz_3_score
quiz4 = testing_data_no_NAs$Quiz_4_score

covid1 = testing_data_no_NAs$COVID.hours..W1.
covid2 = testing_data_no_NAs$COVID.hours..W2.
covid3 = testing_data_no_NAs$COVID.hours..W3.
covid4 = testing_data_no_NAs$COVID.hours..W4.

study1 = testing_data_no_NAs$STA302.hours..W1.
study2 = testing_data_no_NAs$STA302.hours..W2.
study3 = testing_data_no_NAs$STA302.hours..W3.
study4 = testing_data_no_NAs$STA302.hours..W4.

country = testing_data_no_NAs$country
```

```
predicted_values = predict(final_model, testing_data_no_NAs)
actual_values = testing_data$Quiz_4_score
```

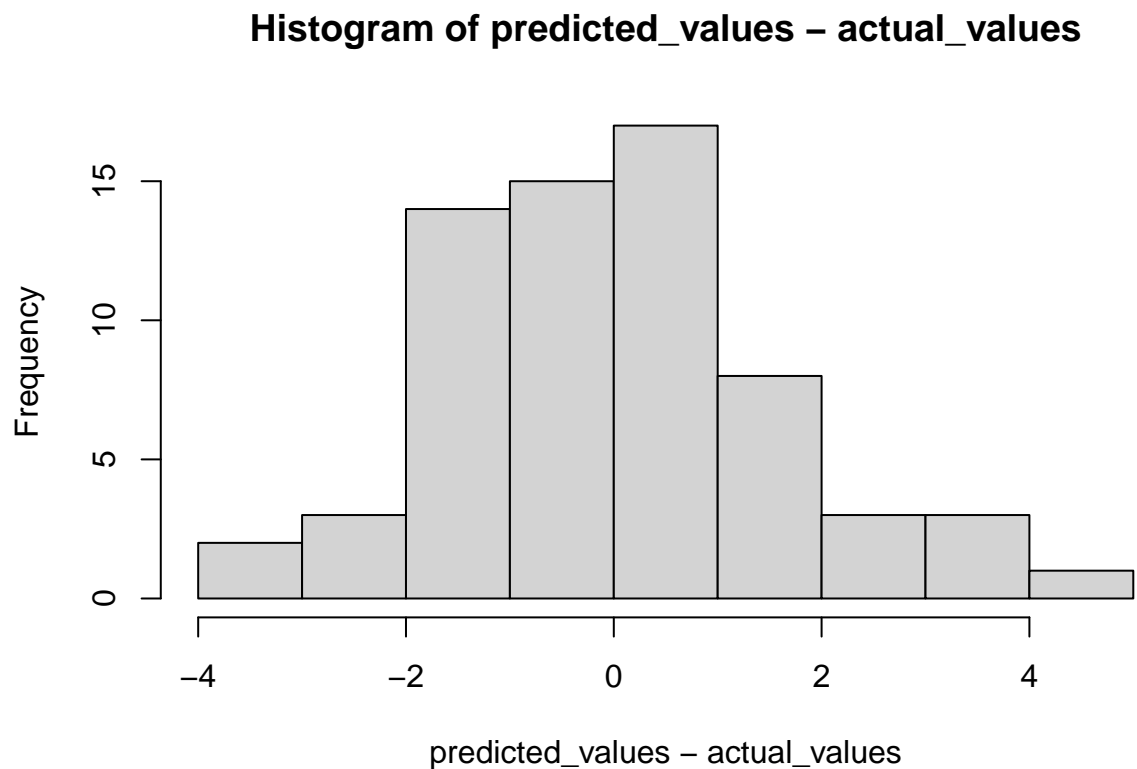
```
mean(predicted_values - actual_values)
```

```
## [1] -0.00768709
```

```
median(predicted_values - actual_values)
```

```
## [1] -0.01197368
```

```
hist(predicted_values - actual_values)
```



one-sample t-test on mean

hypothesized mean = 0

- $n = 77$, so by CLT sample mean is approximately normal

is -0.427222 statistically different from 0? the p-value should be small.

```
t.test(predicted_values - actual_values)
```

```
##
```

```
## One Sample t-test
```

```
##
```

```
## data: predicted_values - actual_values
```

```
## t = -0.038679, df = 65, p-value = 0.9693
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -0.4045976  0.3892234
## sample estimates:
##  mean of x
## -0.00768709
```

p-value = 0.7938 t = -0.26242