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.

```
cleaned_sta302_performance_data <- sta302_performance_data %>%
    # Create a new "country" column, which is just "Country" but whose entries are factors.
    mutate(country = as.factor(Country)) %>%

    # Remove the "X" column: it's simply the row number, which isn't very useful.
    # Remove the "Country" column: column "country" already exists
    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)

    # 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.
```

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.

```
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)))
}
```

```
rows_with_0_NAs = cleaned_sta302_performance_data[
  rows_with_num_NAs(cleaned_sta302_performance_data, 0),
]
rows_with_1_NAs = cleaned_sta302_performance_data[
  rows_with_num_NAs(cleaned_sta302_performance_data, 1),
]
rows_with_2_NAs = cleaned_sta302_performance_data[
  rows_with_num_NAs(cleaned_sta302_performance_data, 2),
]
rows_with_3_NAs = cleaned_sta302_performance_data[
  rows_with_num_NAs(cleaned_sta302_performance_data, 3),
]
rows_with_4_NAs = cleaned_sta302_performance_data[
  rows_with_num_NAs(cleaned_sta302_performance_data, 4),
]
```

Let's count the number of rows with 0 - 4 NAs.

```
print(as_tibble(data.frame(
  nrows_0_NAs = nrow(rows_with_0_NAs),
  nrows_1_NAs = nrow(rows_with_1_NAs),
  nrows_2_NAs = nrow(rows_with_2_NAs),
  nrows_3_NAs = nrow(rows_with_3_NAs),
  nrows_4_NAs = nrow(rows_with_4_NAs)
)))
```

```
## # A tibble: 1 x 5
##   nrows_0_NAs nrows_1_NAs nrows_2_NAs nrows_3_NAs nrows_4_NAs
##         <int>       <int>       <int>       <int>       <int>
## 1         143           9          16          19           1
```

And then we'll determine which row numbers have 0 - 4 NAs.

```
row_nums_of_NA_rows(cleaned_sta302_performance_data, 0)
```

```
##   [1]   1   2   3   4   7   8  11  13  14  15  18  20  21  22  24  25  26  27
##  [19]  29  31  32  33  35  36  37  38  42  44  45  48  50  54  55  57  60  61
##  [37]  62  63  65  66  67  68  70  71  72  73  74  75  76  77  79  81  82  83
##  [55]  84  85  86  87  88  89  92  93  94  97  99 101 103 104 105 106 107 108
##  [73] 109 110 111 112 114 115 116 118 119 122 123 124 126 127 128 129 130 134
##  [91] 135 136 137 139 140 141 142 144 146 147 149 150 151 152 153 154 155 156
## [109] 157 158 159 160 161 162 163 164 165 166 167 169 170 171 172 173 174 175
## [127] 176 177 178 179 180 183 184 185 186 187 190 191 193 196 199 200 201
```

```
row_nums_of_NA_rows(cleaned_sta302_performance_data, 1)
```

```
## [1]  34  78  80 117 132 138 143 145 197
```

```
row_nums_of_NA_rows(cleaned_sta302_performance_data, 2)
```

```
##  [1]  10  12  43  52  59  90  95  96  98 100 121 125 131 181 189 192
```

```
row_nums_of_NA_rows(cleaned_sta302_performance_data, 3)
```

```
##  [1]   5   6  28  69 113 188 195 202 203 205 207 208 209 211 215 216 217 218 221
```

```
row_nums_of_NA_rows(cleaned_sta302_performance_data, 4)
```

```
## [1] 223
```

Columns with NAs

```
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))
}
```

```
perform_data = cleaned_sta302_performance_data
print(as_tibble(data.frame(
  week1_covid = num_column_NAs(perform_data$COVID.hours..W1.),
  week2_covid = num_column_NAs(perform_data$COVID.hours..W2.),
  week3_covid = num_column_NAs(perform_data$COVID.hours..W3.),
  week4_covid = num_column_NAs(perform_data$COVID.hours..W4.)
)))
```

```
## # A tibble: 1 x 4
##   week1_covid week2_covid week3_covid week4_covid
##         <int>       <int>       <int>       <int>
## 1          26          22          21          40
```

```
print(as_tibble(data.frame(
  week1_sta302 = num_column_NAs(perform_data$STA302.hours..W1.),
  week2_sta302 = num_column_NAs(perform_data$STA302.hours..W2.),
  week3_sta302 = num_column_NAs(perform_data$STA302.hours..W3.),
  week4_sta302 = num_column_NAs(perform_data$STA302.hours..W4.)
)))
```

```
## # A tibble: 1 x 4
##   week1_sta302 week2_sta302 week3_sta302 week4_sta302
##          <int>        <int>        <int>        <int>
## 1           26           22           20           40
```

```
print(as_tibble(data.frame(
  quiz1_score = num_column_NAs(perform_data$Quiz_1_score),
  quiz2_score = num_column_NAs(perform_data$Quiz_2_score),
  quiz3_score = num_column_NAs(perform_data$Quiz_3_score),
  quiz4_score = num_column_NAs(perform_data$Quiz_4_score)
)))
```

```
## # A tibble: 1 x 4
##   quiz1_score quiz2_score quiz3_score quiz4_score
##         <int>       <int>       <int>       <int>
## 1          13          36          31          34
```

Rows with Mistyped Columns

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

```
rows_with_mistyped_columms = cleaned_sta302_performance_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 = cleaned_sta302_performance_data %>%
  select(-country)
```

Find Significance Predictor Variables, Select Predictor Variables Based on Criterion

```
# use week 5b slides -- choose model selection criterion to pick predictor variables.


# use lm() on a bunch of predictor variables to determine significant
# predictor variables.
```

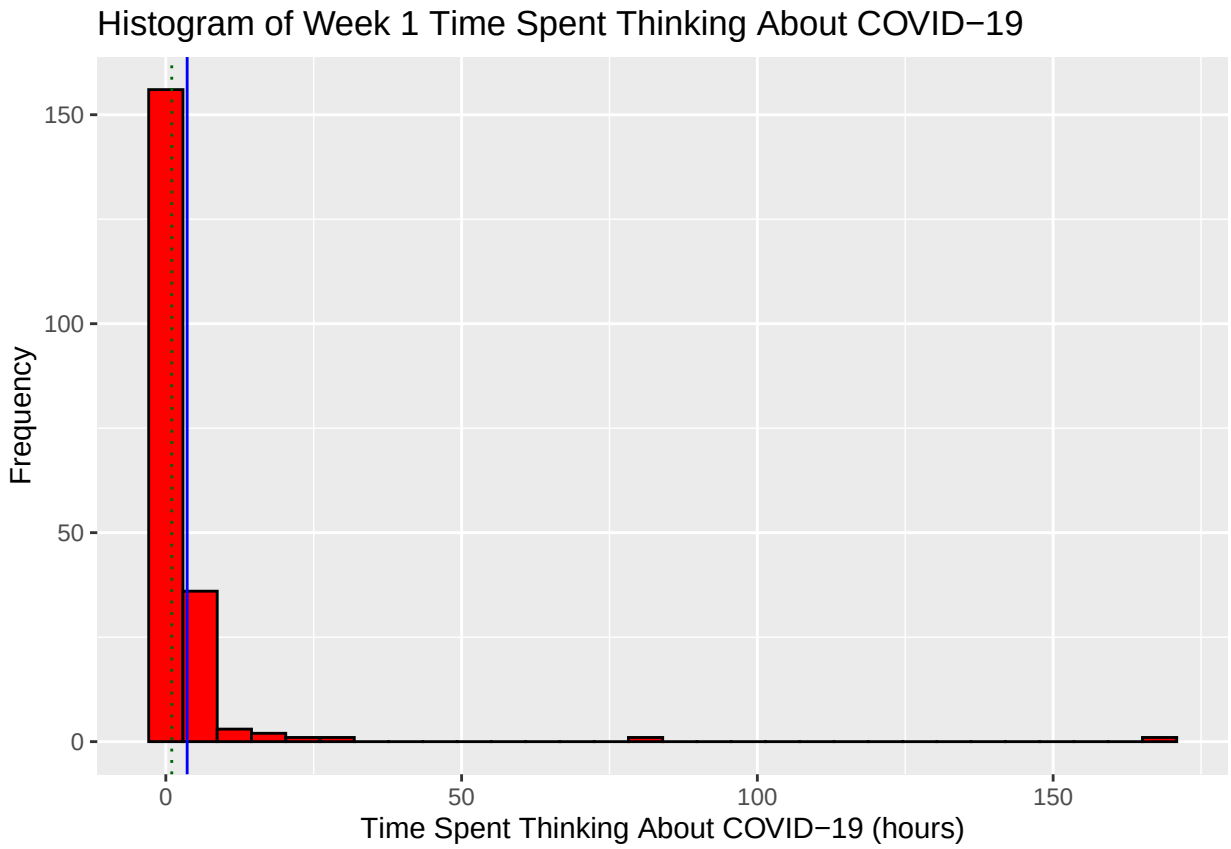
Histograms

```
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")
}
```

Histograms of COVID Hours

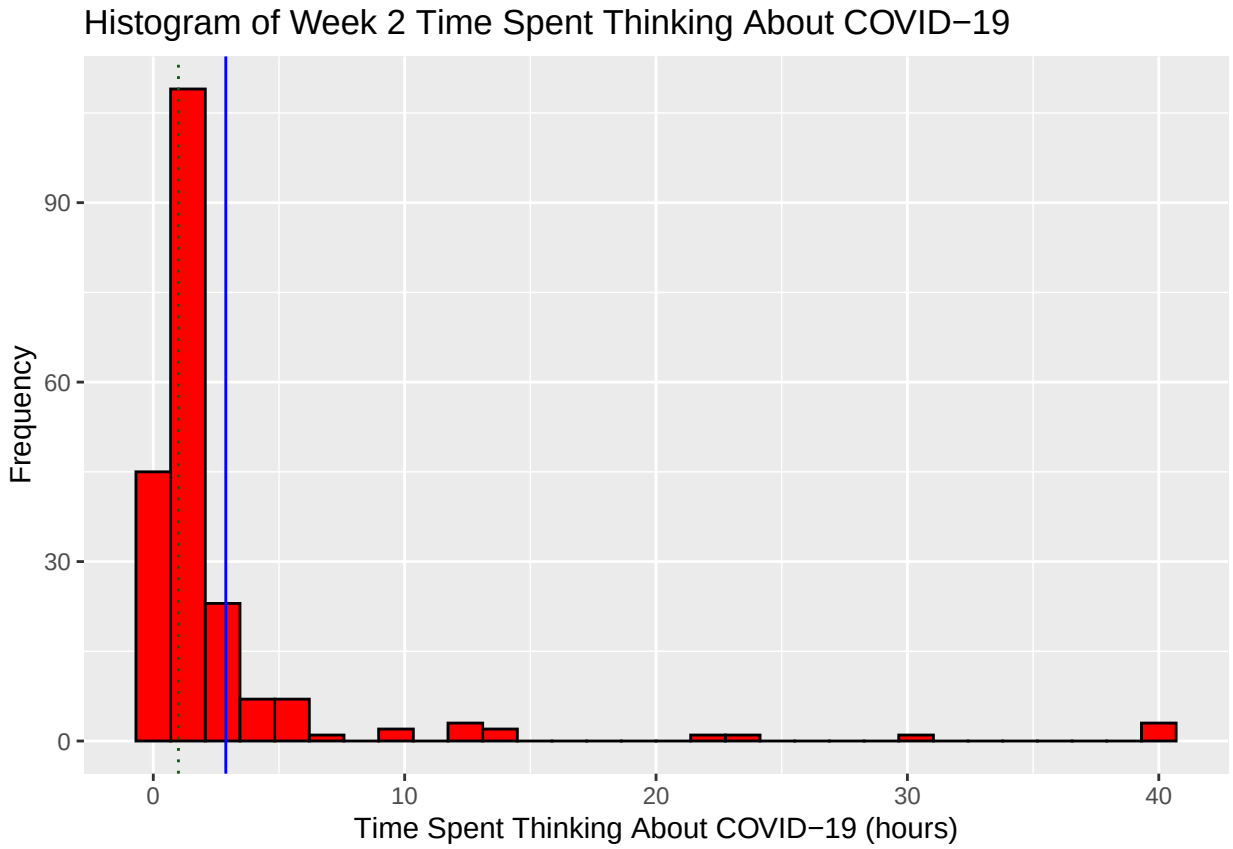
```
display_histogram(cleaned_sta302_performance_data,
                  cleaned_sta302_performance_data$COVID.hours..W1.,
                  "Histogram of Week 1 Time Spent Thinking About COVID-19",
                  "Time Spent Thinking About COVID-19 (hours)")
```

```
## Warning: Removed 26 rows containing non-finite values (stat_bin).
```



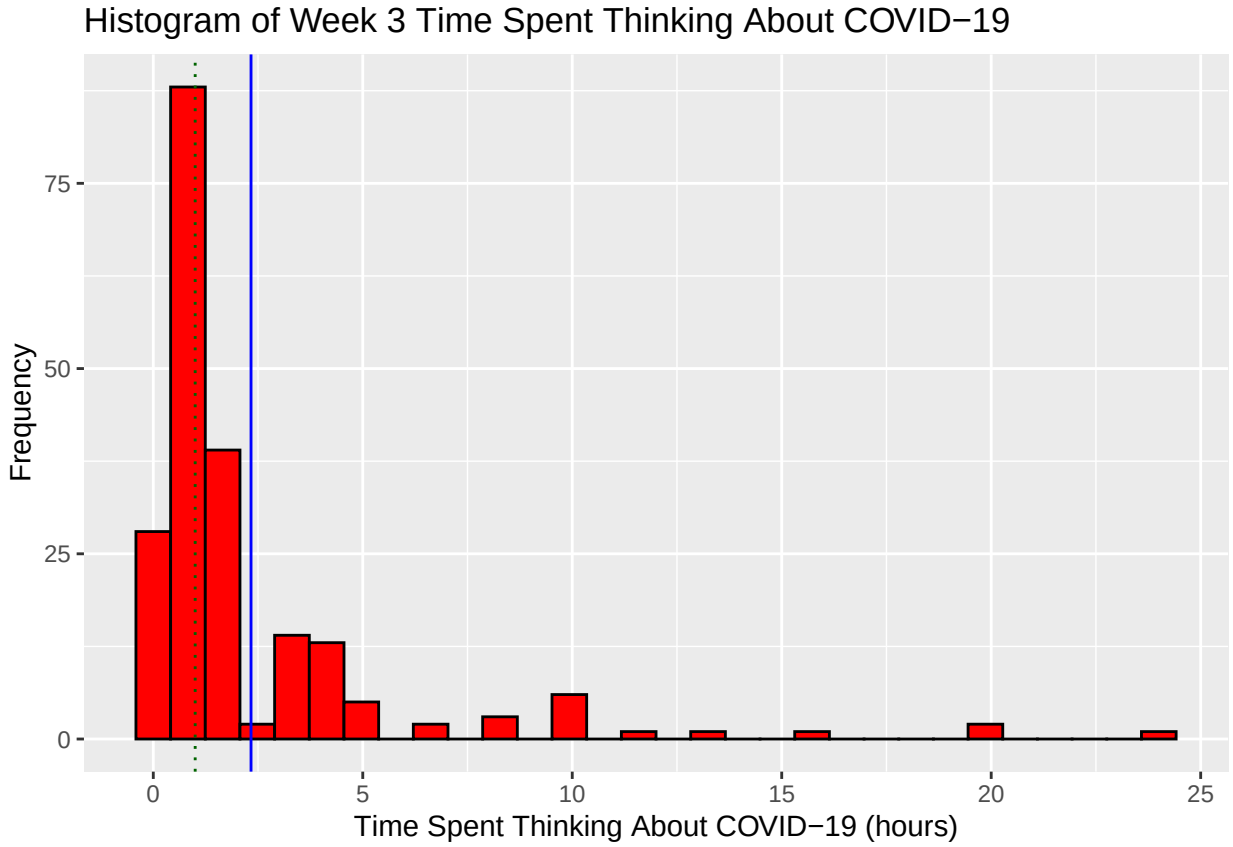
```
display_histogram(cleaned_sta302_performance_data,
                  cleaned_sta302_performance_data$COVID.hours..W2.,
                  "Histogram of Week 2 Time Spent Thinking About COVID-19",
                  "Time Spent Thinking About COVID-19 (hours)")
```

```
## Warning: Removed 22 rows containing non-finite values (stat_bin).
```



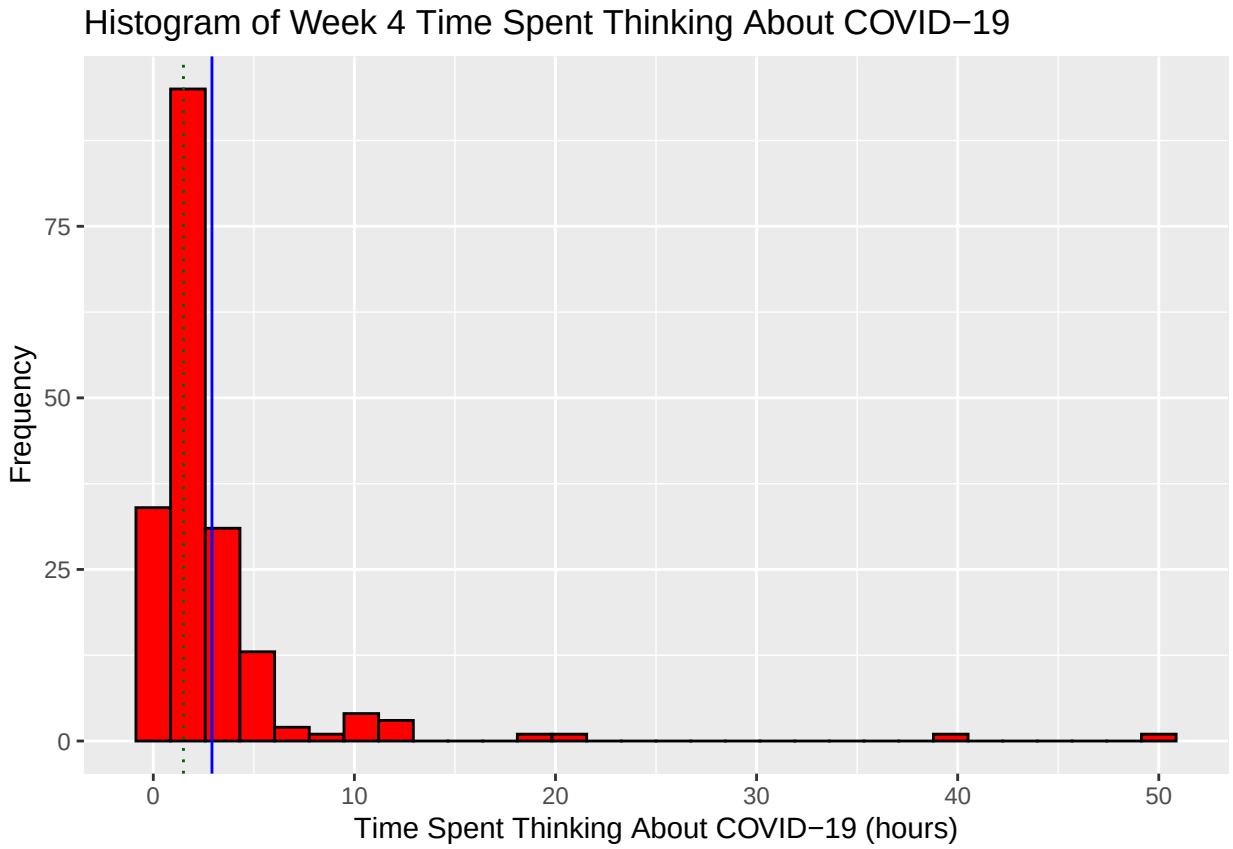
```
display_histogram(cleaned_sta302_performance_data,
                  cleaned_sta302_performance_data$COVID.hours..W3.,
                  "Histogram of Week 3 Time Spent Thinking About COVID-19",
                  "Time Spent Thinking About COVID-19 (hours)")
```

```
## Warning: Removed 21 rows containing non-finite values (stat_bin).
```



```
display_histogram(cleaned_sta302_performance_data,
                  cleaned_sta302_performance_data$COVID.hours..W4.,
                  "Histogram of Week 4 Time Spent Thinking About COVID-19",
                  "Time Spent Thinking About COVID-19 (hours)")
```

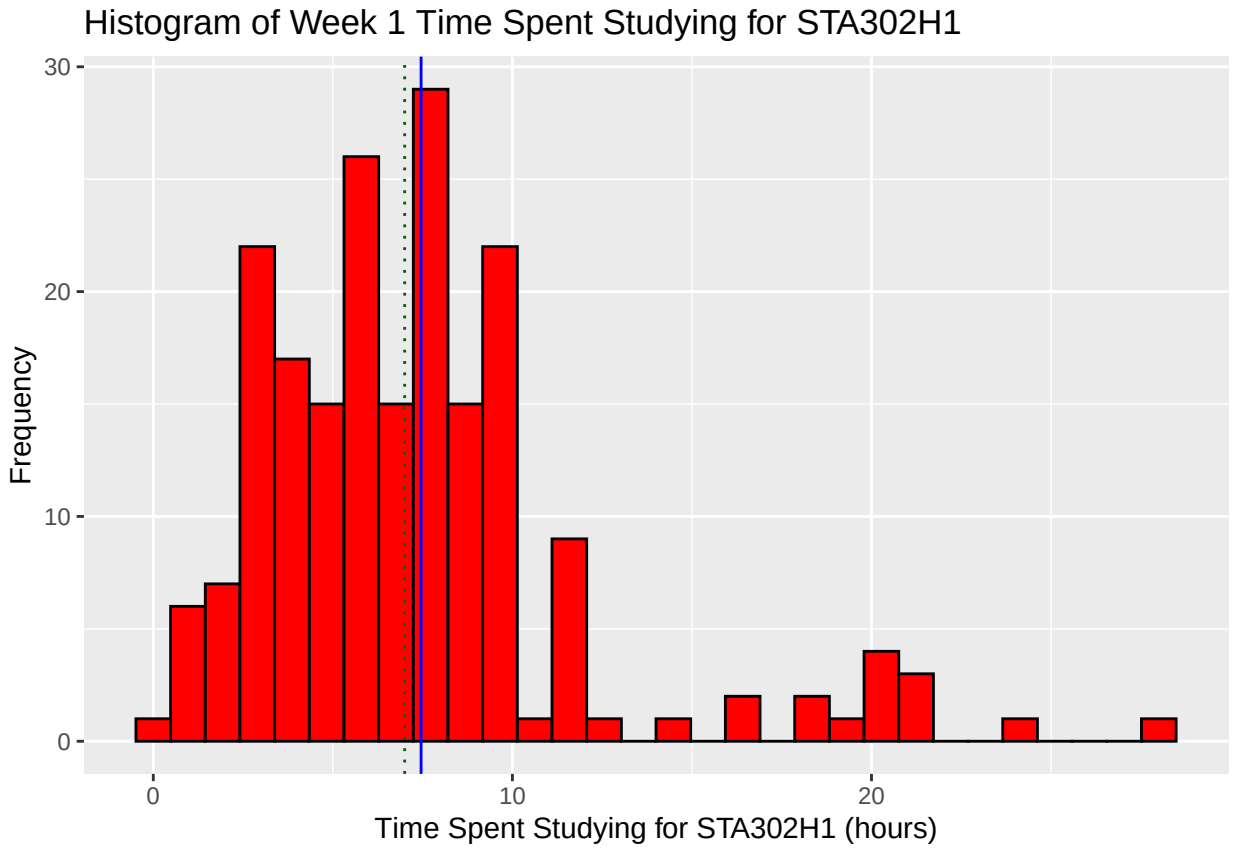
```
## Warning: Removed 40 rows containing non-finite values (stat_bin).
```



Histograms of STA302H1 Hours

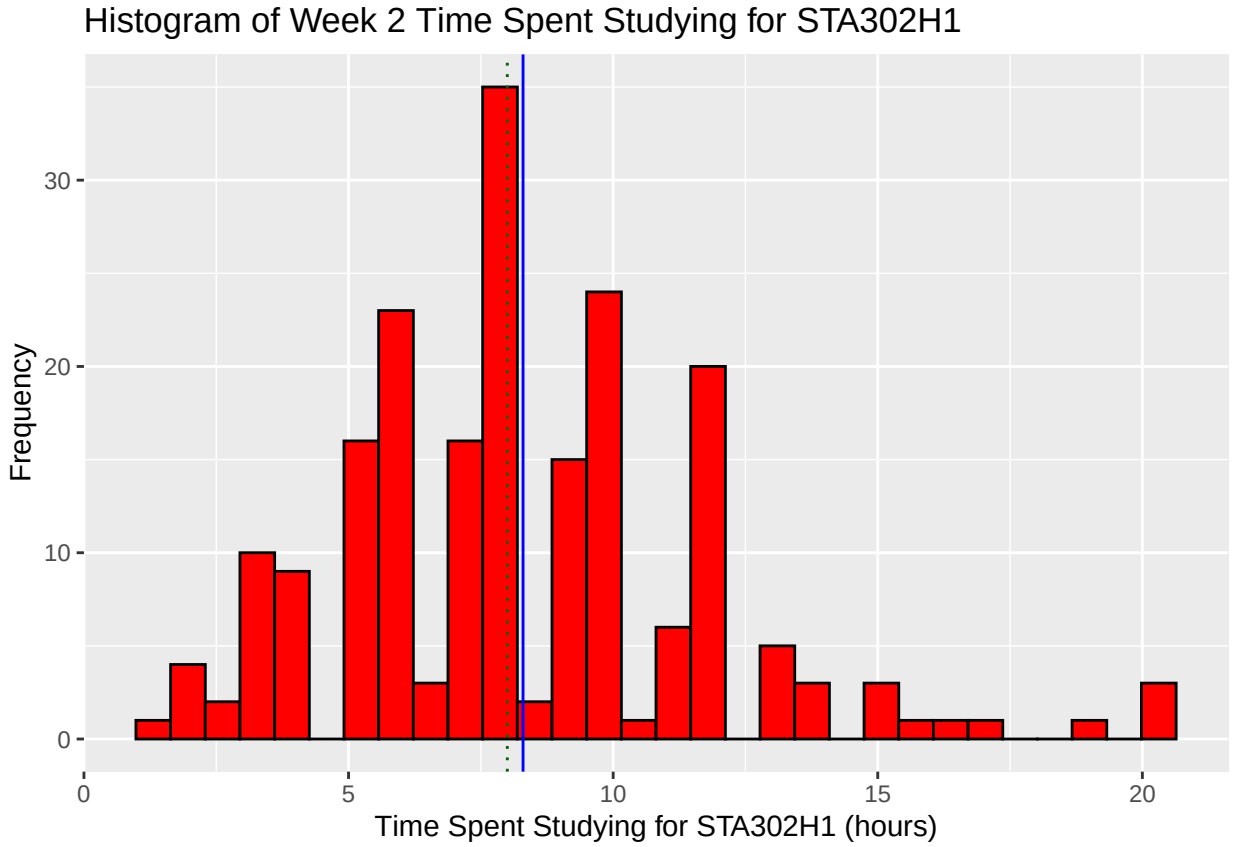
```
display_histogram(cleaned_sta302_performance_data,
                  cleaned_sta302_performance_data$STA302.hours..W1.,
                  "Histogram of Week 1 Time Spent Studying for STA302H1",
                  "Time Spent Studying for STA302H1 (hours)")
```

```
## Warning: Removed 26 rows containing non-finite values (stat_bin).
```



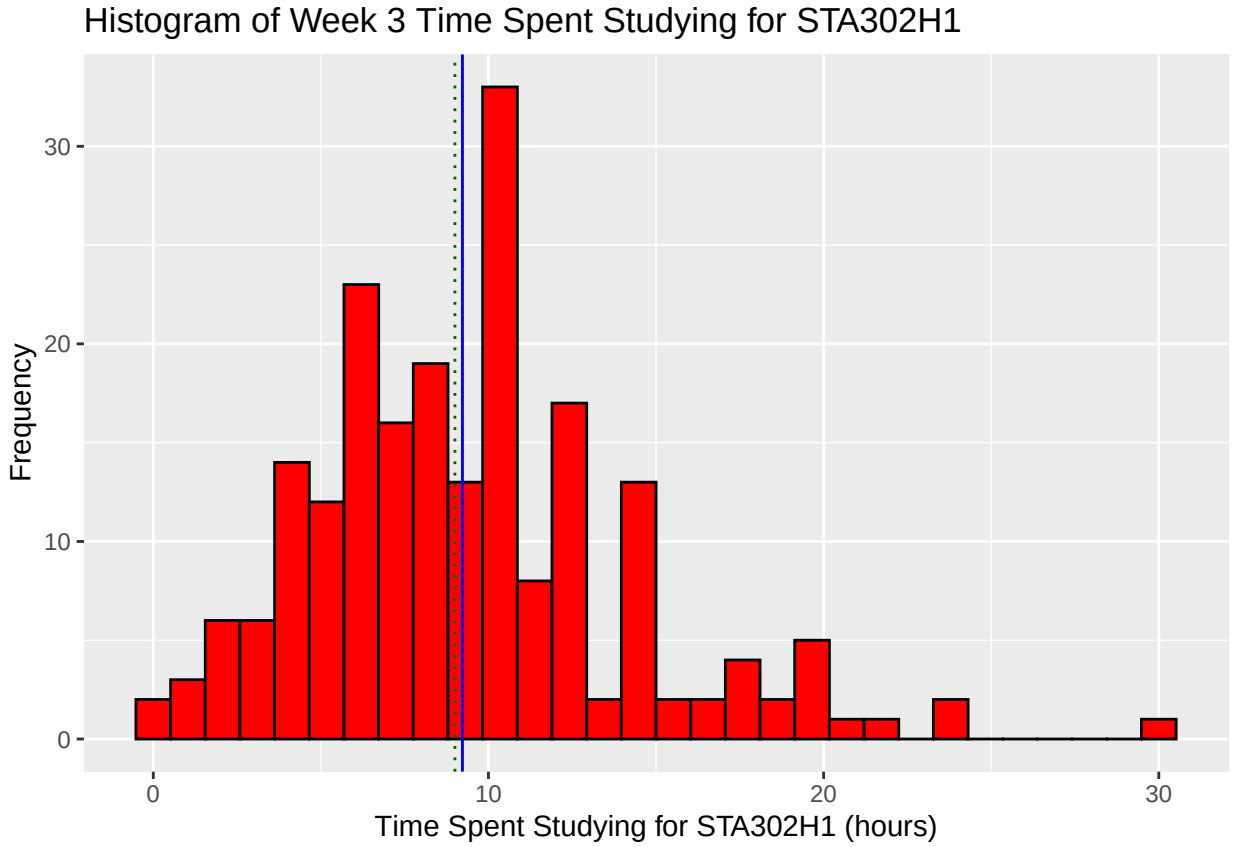
```
display_histogram(cleaned_sta302_performance_data,
                  cleaned_sta302_performance_data$STA302.hours..W2.,
                  "Histogram of Week 2 Time Spent Studying for STA302H1",
                  "Time Spent Studying for STA302H1 (hours)")
```

```
## Warning: Removed 22 rows containing non-finite values (stat_bin).
```



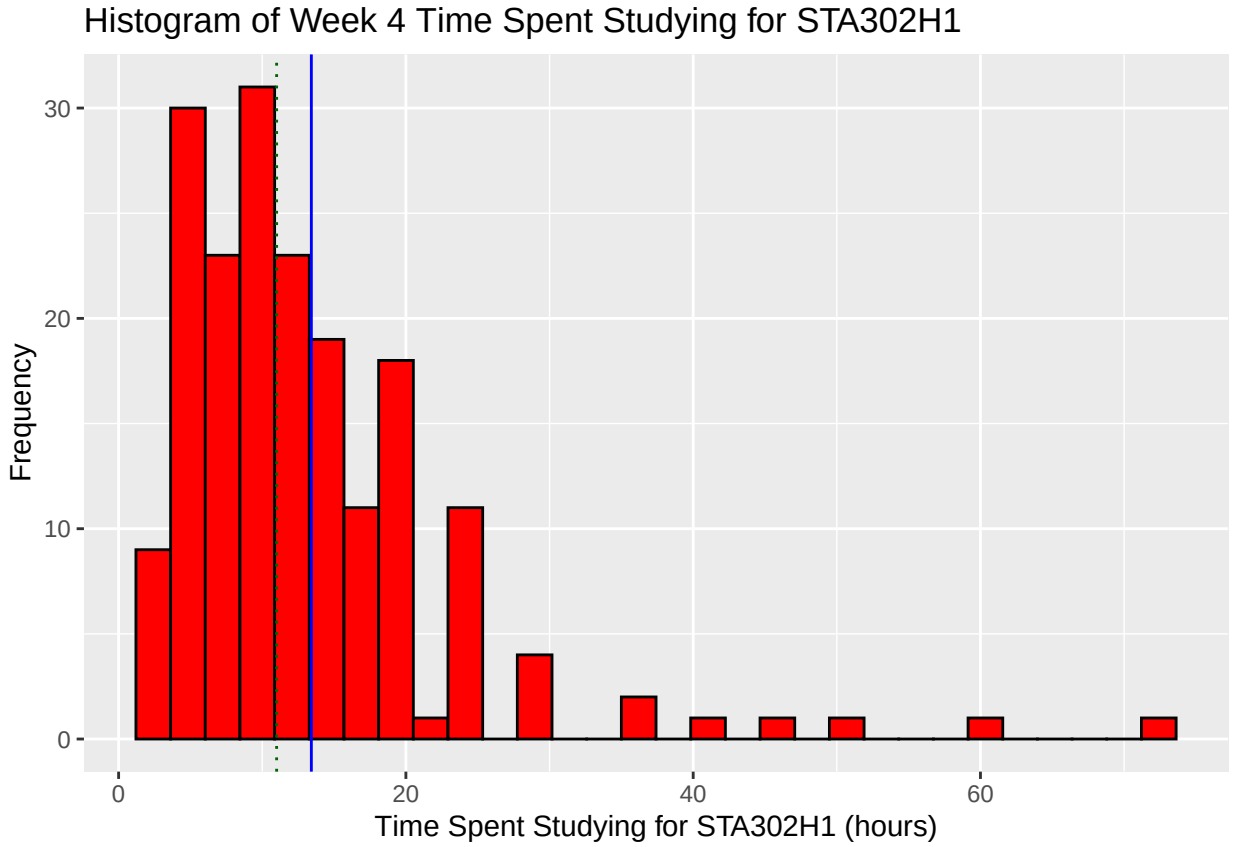
```
display_histogram(cleaned_sta302_performance_data,
                  cleaned_sta302_performance_data$STA302.hours..W3.,
                  "Histogram of Week 3 Time Spent Studying for STA302H1",
                  "Time Spent Studying for STA302H1 (hours)")
```

```
## Warning: Removed 20 rows containing non-finite values (stat_bin).
```



```
display_histogram(cleaned_sta302_performance_data,
                  cleaned_sta302_performance_data$STA302.hours..W4.,
                  "Histogram of Week 4 Time Spent Studying for STA302H1",
                  "Time Spent Studying for STA302H1 (hours)")
```

```
## Warning: Removed 40 rows containing non-finite values (stat_bin).
```



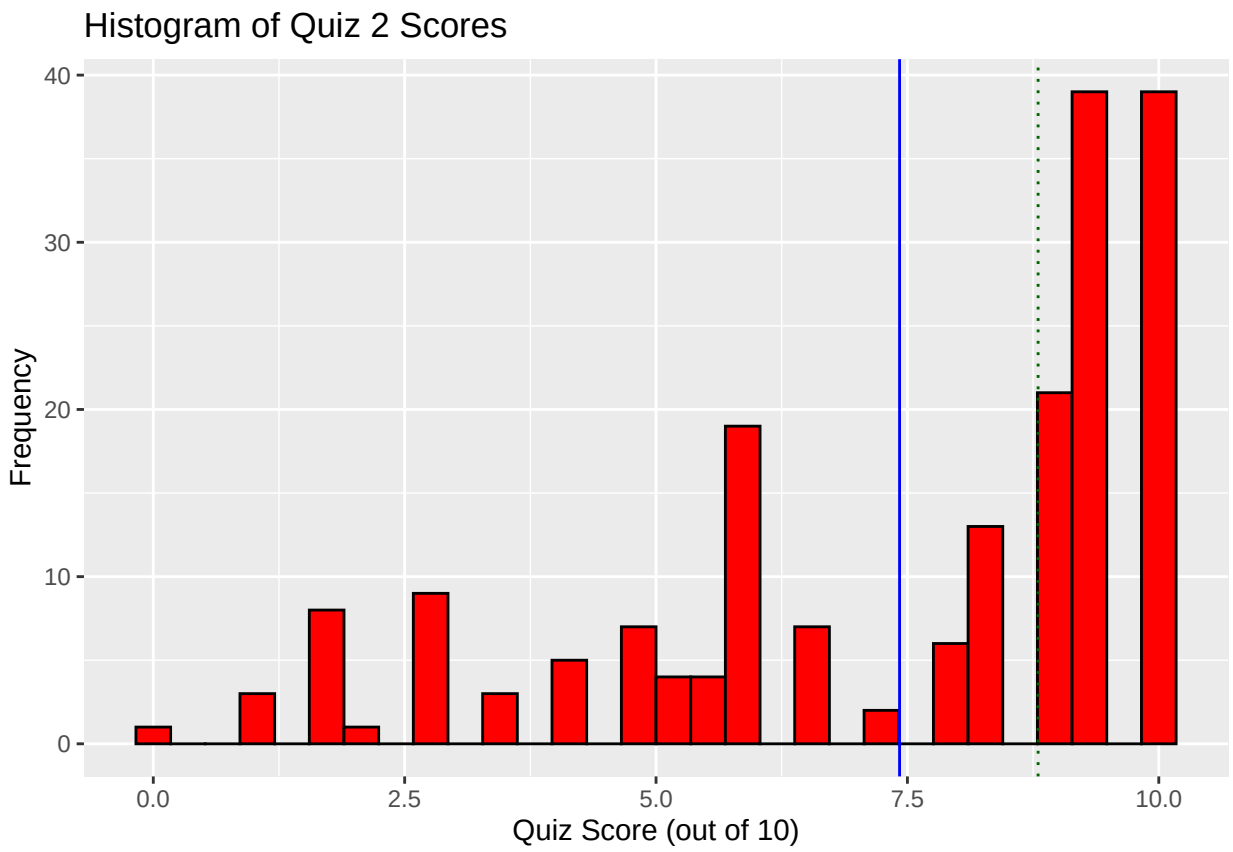
Histograms of Quiz Scores

```
display_histogram(cleaned_sta302_performance_data,
                  cleaned_sta302_performance_data$Quiz_1_score,
                  "Histogram of Quiz 1 Scores",
                  "Quiz Score (out of 10)")
```

```
## Warning: Removed 13 rows containing non-finite values (stat_bin).
```

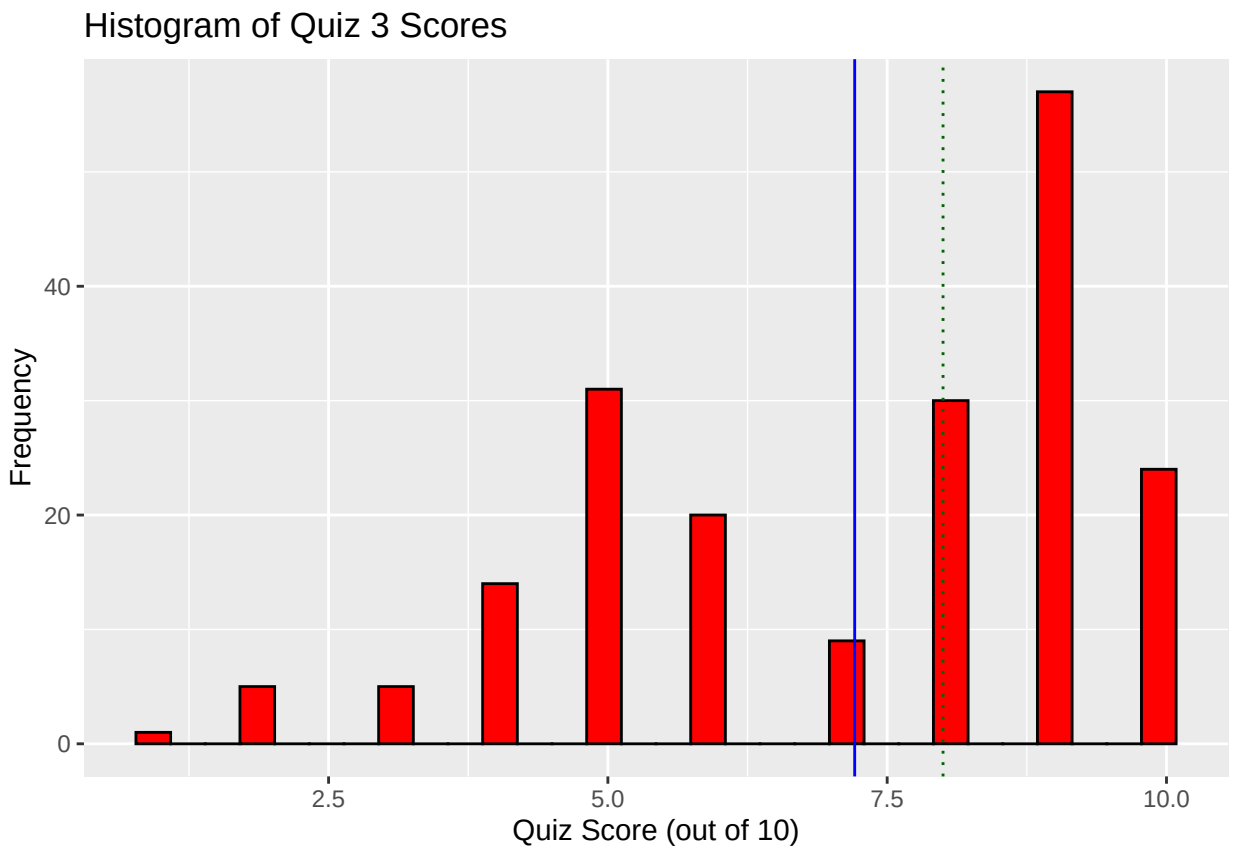
```r
display_histogram(cleaned_sta302_performance_data,
                  cleaned_sta302_performance_data$Quiz_2_score,
                  "Histogram of Quiz 2 Scores",
                  "Quiz Score (out of 10)")
```

```
## Warning: Removed 36 rows containing non-finite values (stat_bin).
```



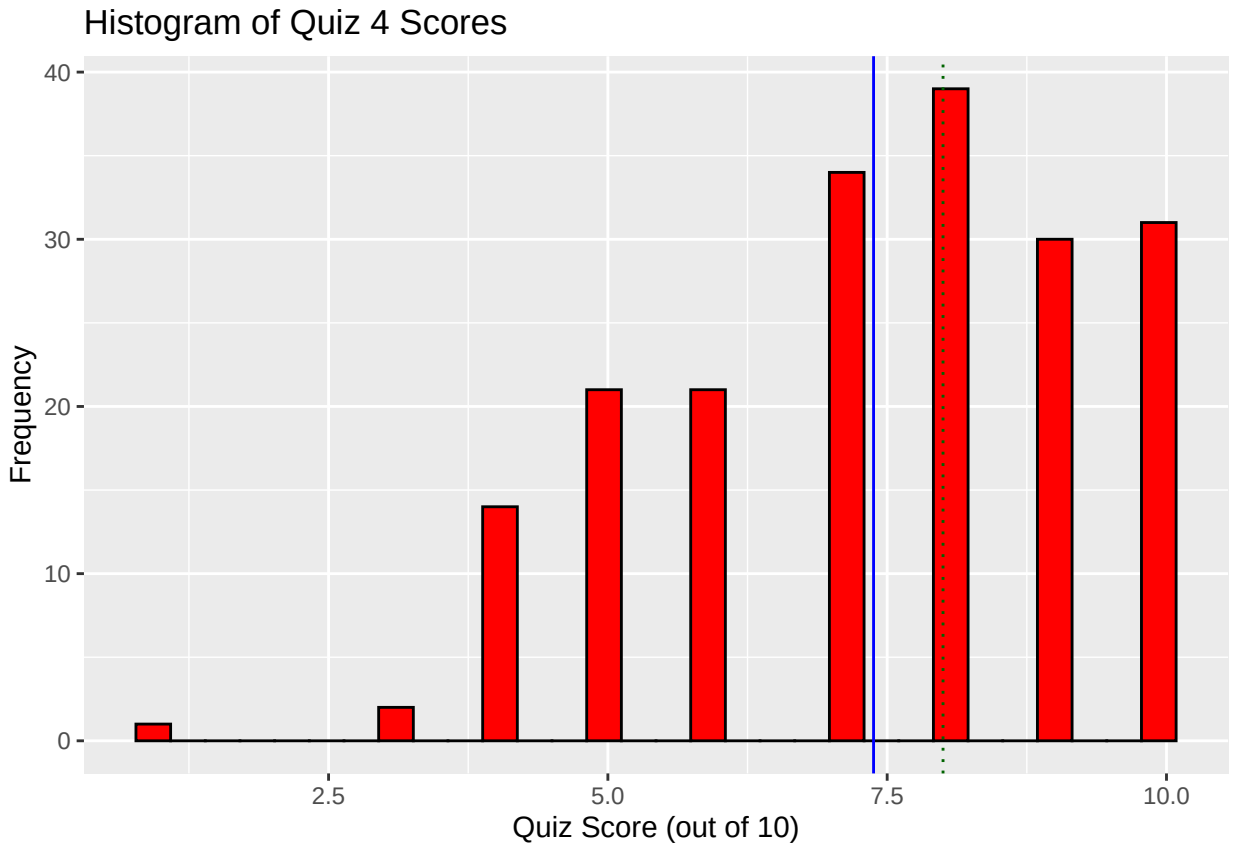
```
display_histogram(cleaned_sta302_performance_data,
                  cleaned_sta302_performance_data$Quiz_3_score,
                  "Histogram of Quiz 3 Scores",
                  "Quiz Score (out of 10)")
```

```
## Warning: Removed 31 rows containing non-finite values (stat_bin).
```



```
display_histogram(cleaned_sta302_performance_data,
                  cleaned_sta302_performance_data$Quiz_4_score,
                  "Histogram of Quiz 4 Scores",
                  "Quiz Score (out of 10)")
```

```
## Warning: Removed 34 rows containing non-finite values (stat_bin).
```



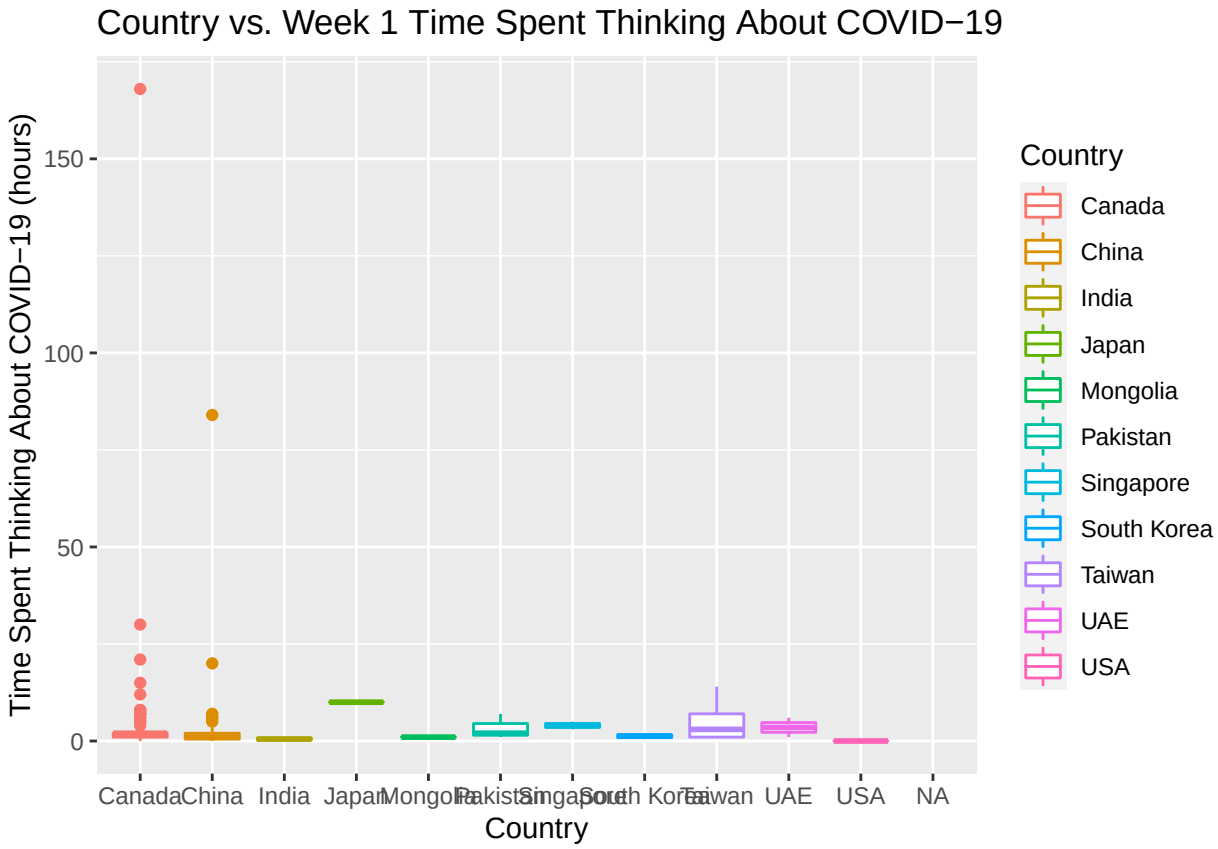
Boxplots

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

Boxplots of COVID Hours

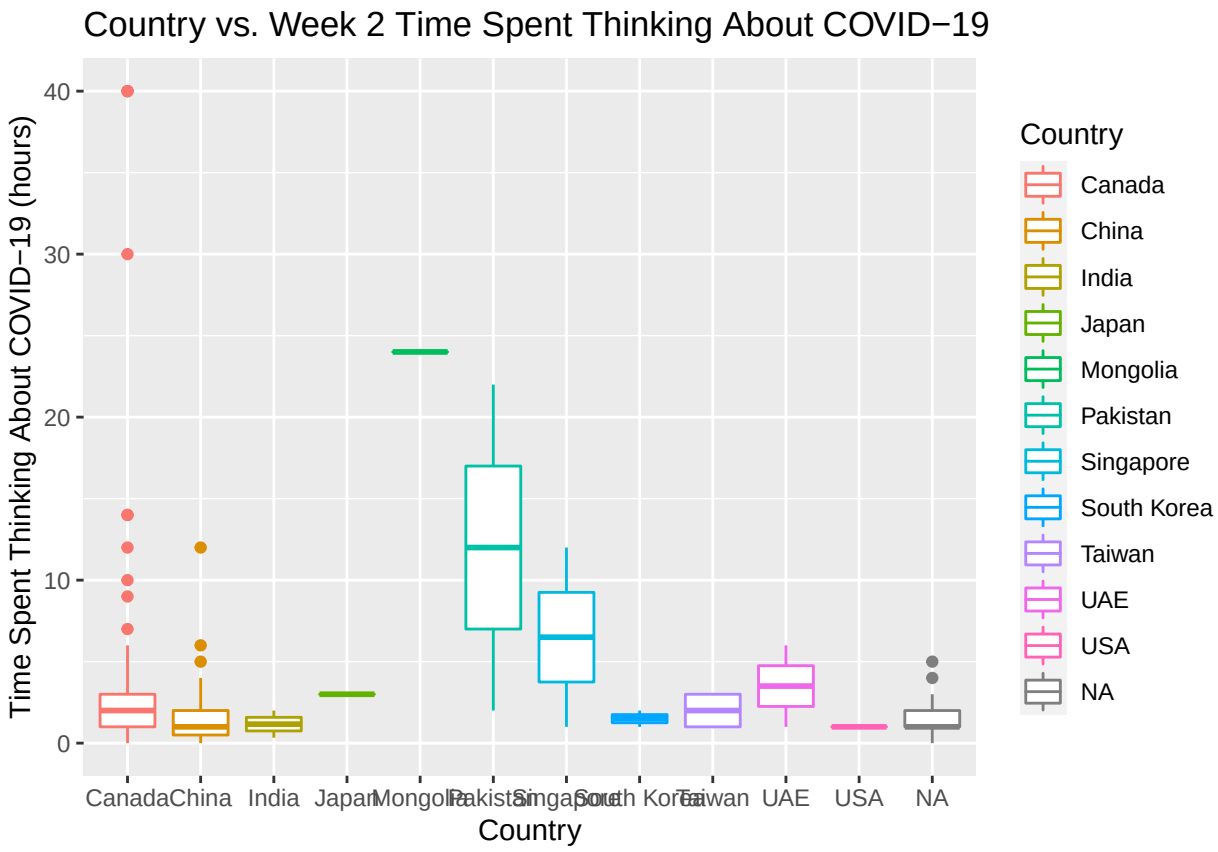
```
display_boxplot(cleaned_sta302_performance_data, COVID.hours..W1.,
                "Country vs. Week 1 Time Spent Thinking About COVID-19",
                "Time Spent Thinking About COVID-19 (hours)")
```

```
## Warning: Removed 26 rows containing non-finite values (stat_boxplot).
```



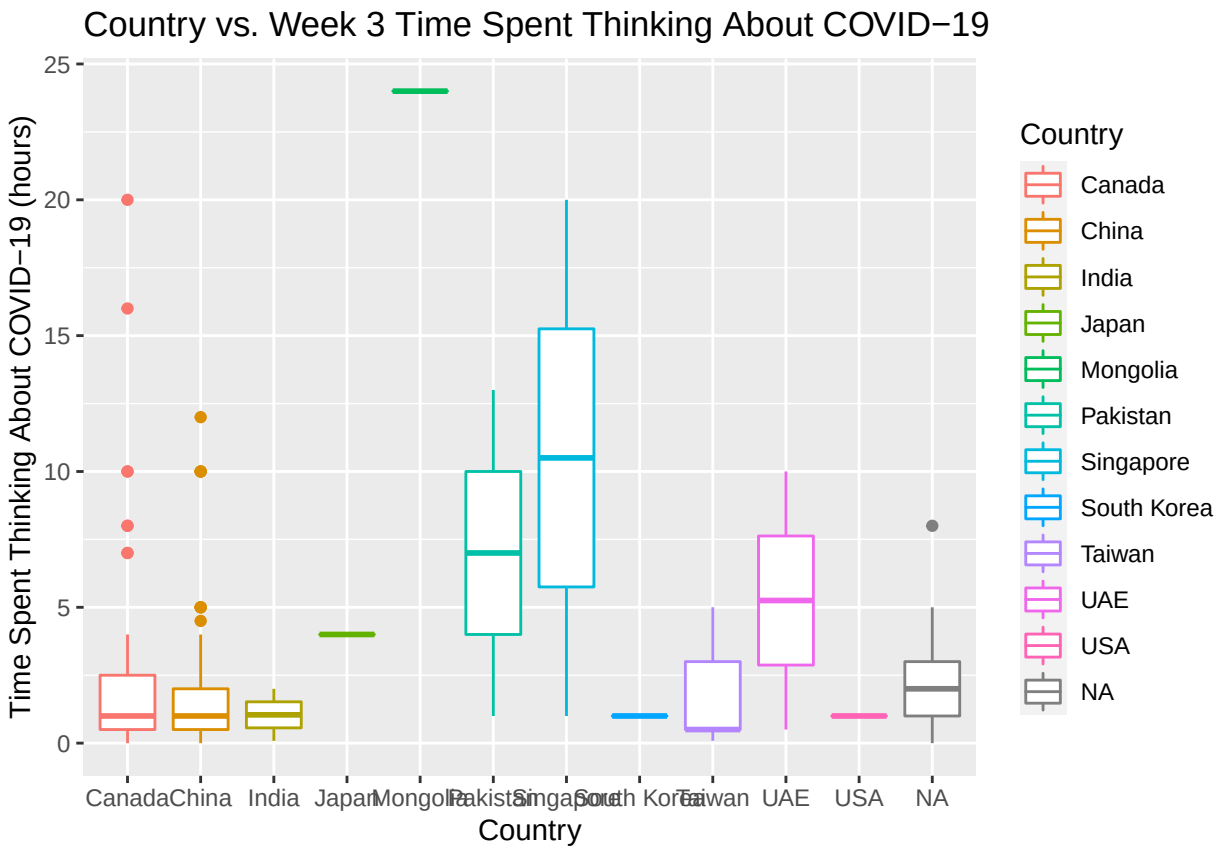
```
display_boxplot(cleaned_sta302_performance_data, COVID.hours..W2.,
                "Country vs. Week 2 Time Spent Thinking About COVID-19",
                "Time Spent Thinking About COVID-19 (hours)")
```

```
## Warning: Removed 22 rows containing non-finite values (stat_boxplot).
```



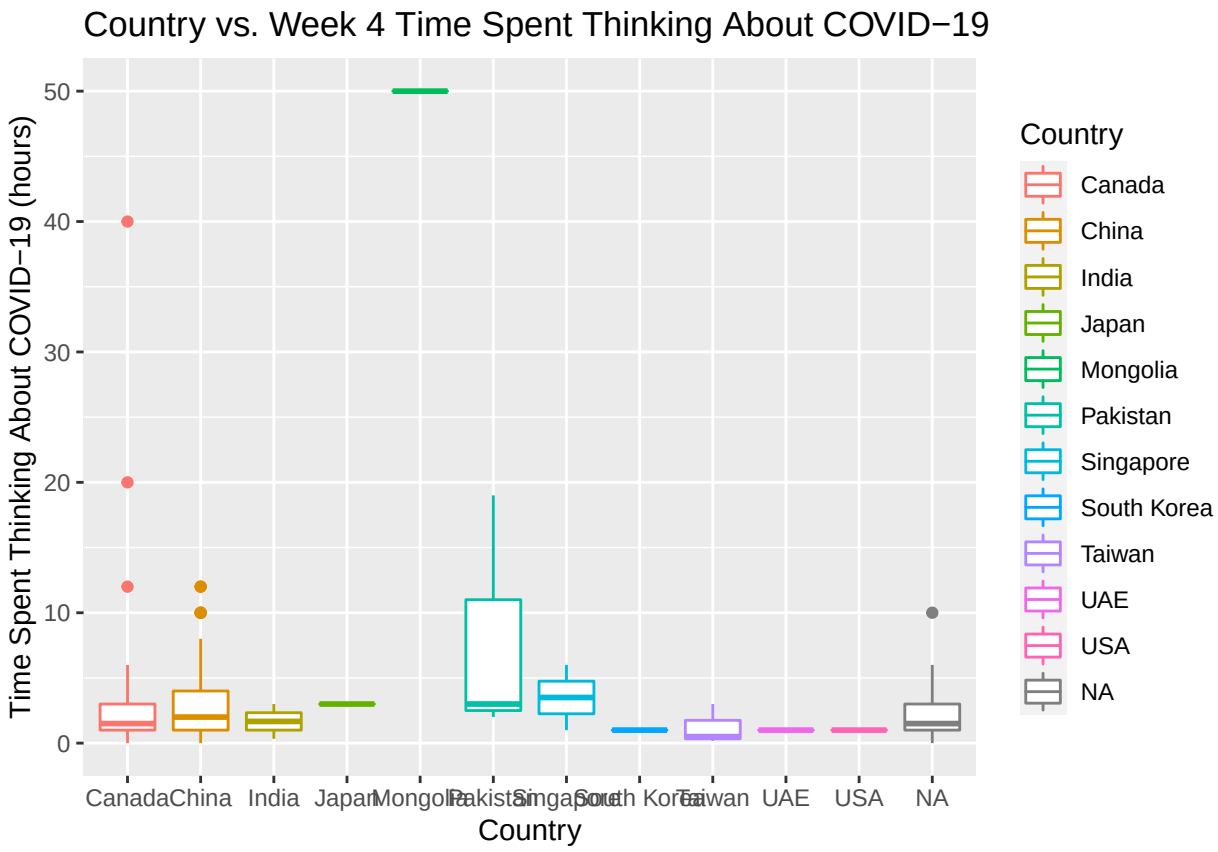
```
display_boxplot(cleaned_sta302_performance_data, COVID.hours..W3.,
                "Country vs. Week 3 Time Spent Thinking About COVID-19",
                "Time Spent Thinking About COVID-19 (hours)")
```

```
## Warning: Removed 21 rows containing non-finite values (stat_boxplot).
```



```
display_boxplot(cleaned_sta302_performance_data, COVID.hours..W4.,
                "Country vs. Week 4 Time Spent Thinking About COVID-19",
                "Time Spent Thinking About COVID-19 (hours)")
```

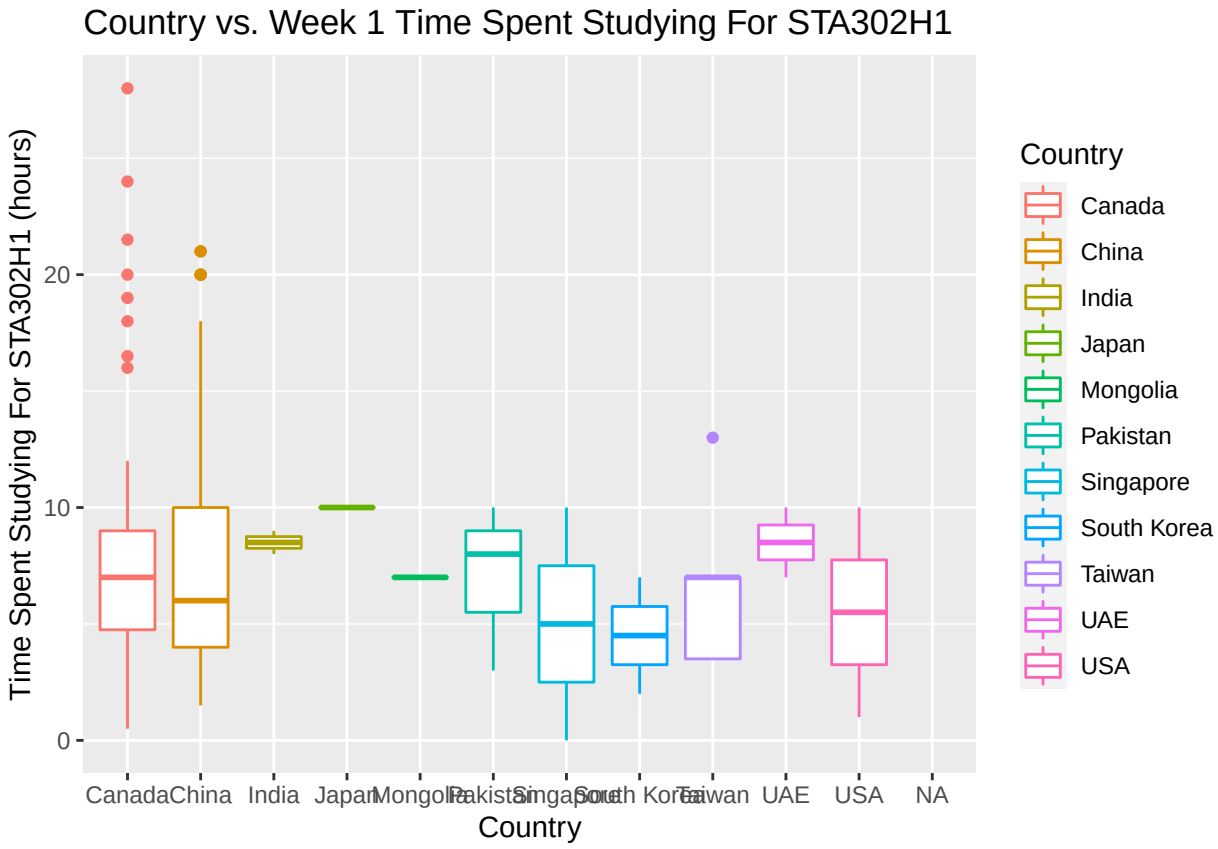
```
## Warning: Removed 40 rows containing non-finite values (stat_boxplot).
```



Boxplots of STA302H1 Hours

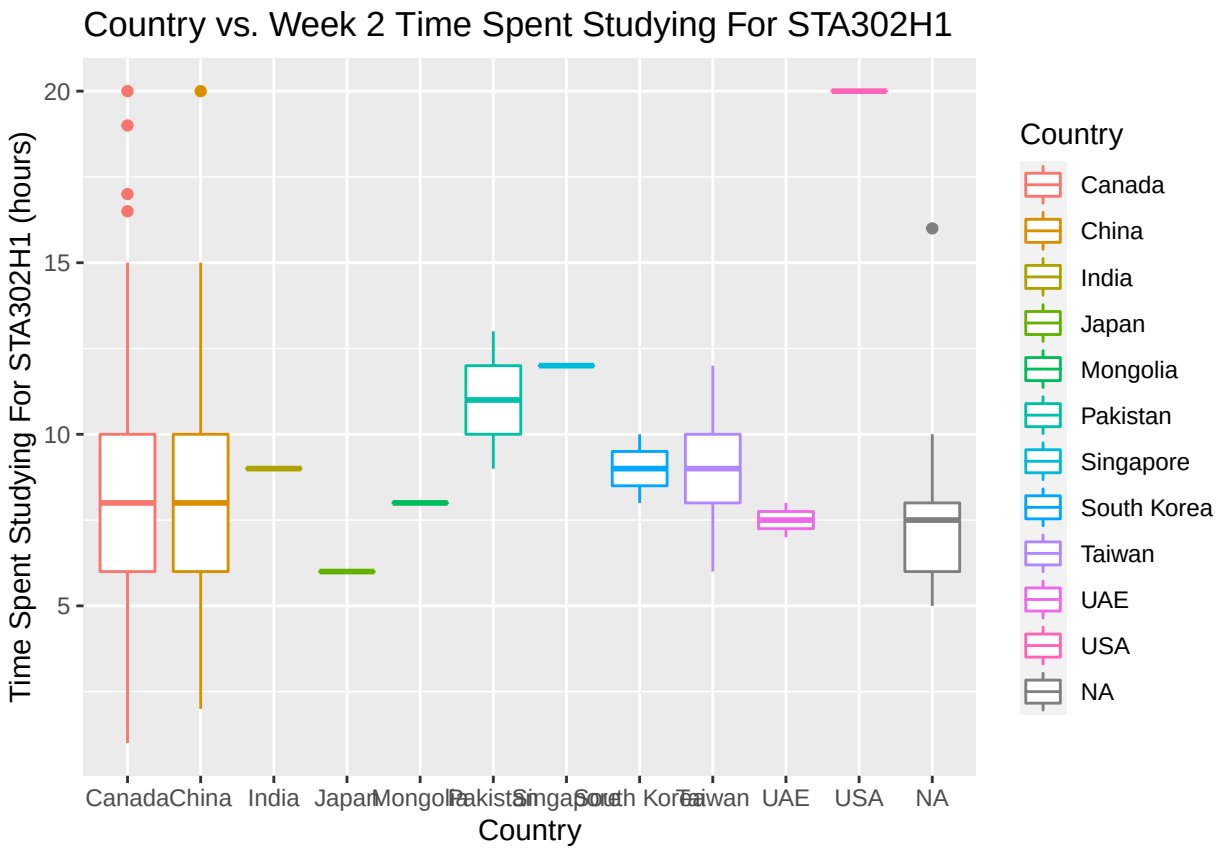
```
display_boxplot(cleaned_sta302_performance_data, STA302.hours..W1.,
                "Country vs. Week 1 Time Spent Studying For STA302H1",
                "Time Spent Studying For STA302H1 (hours)")
```

```
## Warning: Removed 26 rows containing non-finite values (stat_boxplot).
```



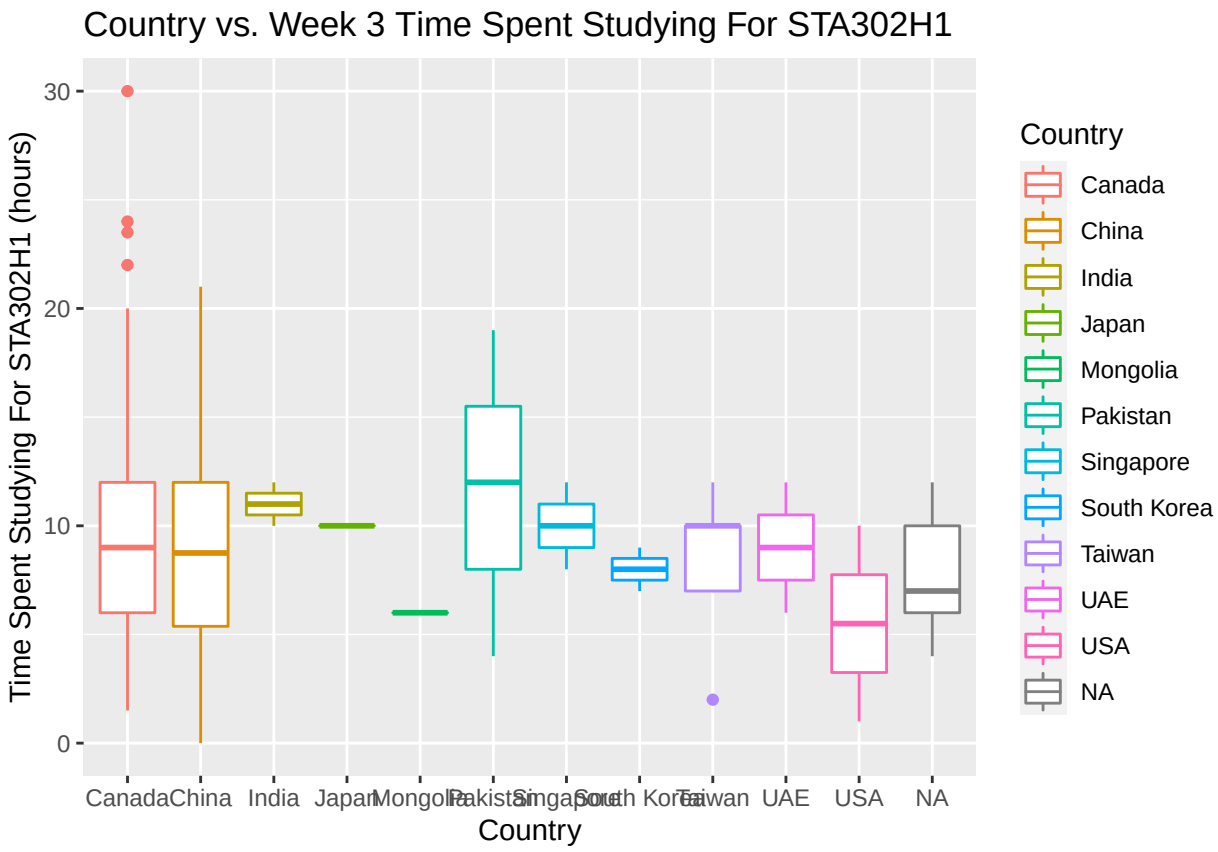
```
display_boxplot(cleaned_sta302_performance_data, STA302.hours..W2.,
                "Country vs. Week 2 Time Spent Studying For STA302H1",
                "Time Spent Studying For STA302H1 (hours)")
```

```
## Warning: Removed 22 rows containing non-finite values (stat_boxplot).
```



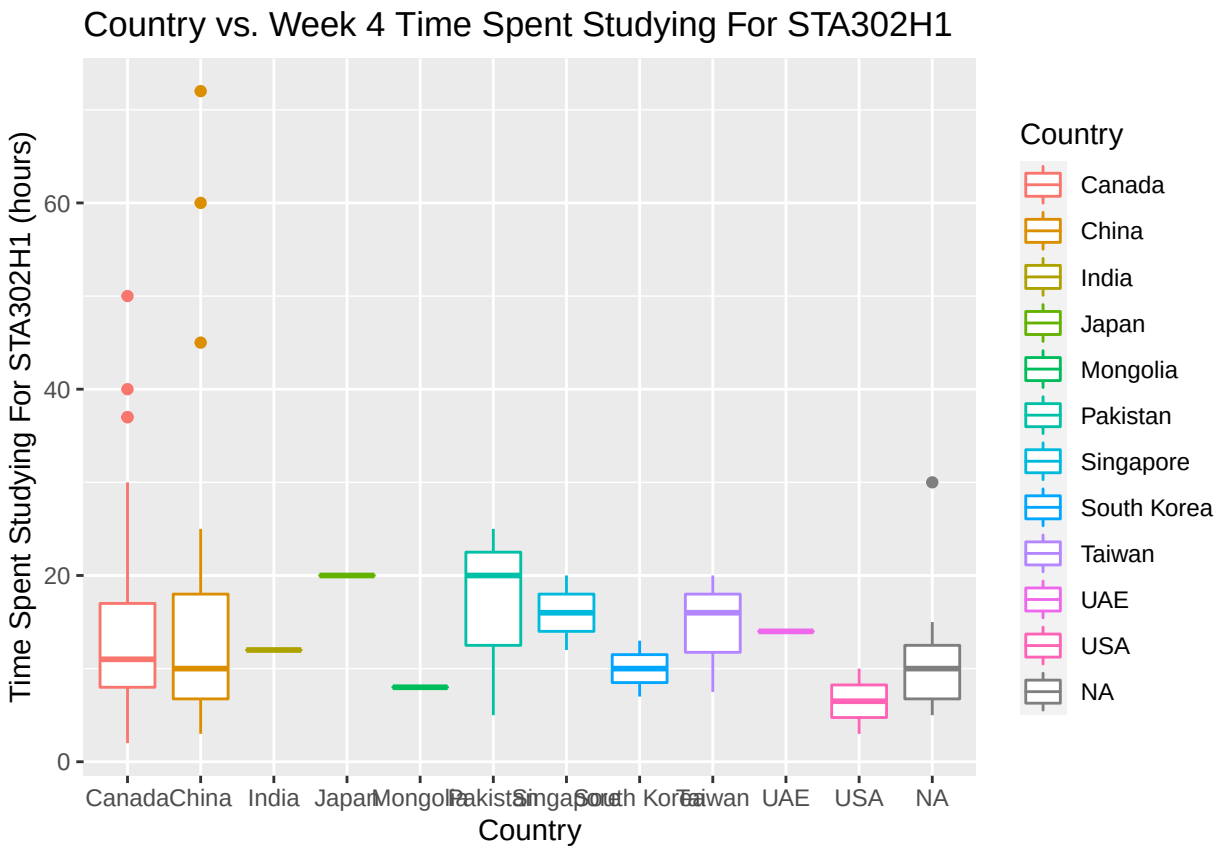
```
display_boxplot(cleaned_sta302_performance_data, STA302.hours..W3.,
                "Country vs. Week 3 Time Spent Studying For STA302H1",
                "Time Spent Studying For STA302H1 (hours)")
```

```
## Warning: Removed 20 rows containing non-finite values (stat_boxplot).
```



```
display_boxplot(cleaned_sta302_performance_data, STA302.hours..W4.,
                "Country vs. Week 4 Time Spent Studying For STA302H1",
                "Time Spent Studying For STA302H1 (hours)")
```

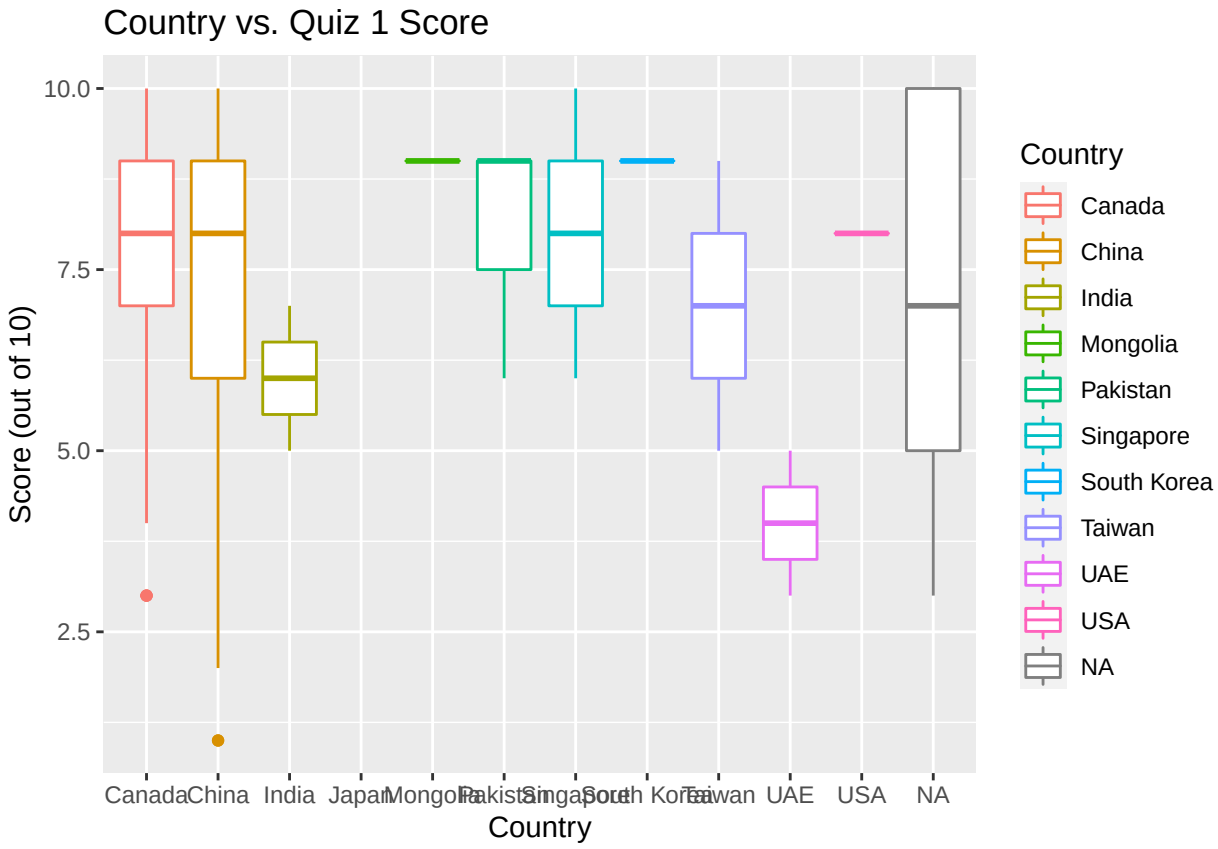
```
## Warning: Removed 40 rows containing non-finite values (stat_boxplot).
```



Boxplots of Quiz Scores

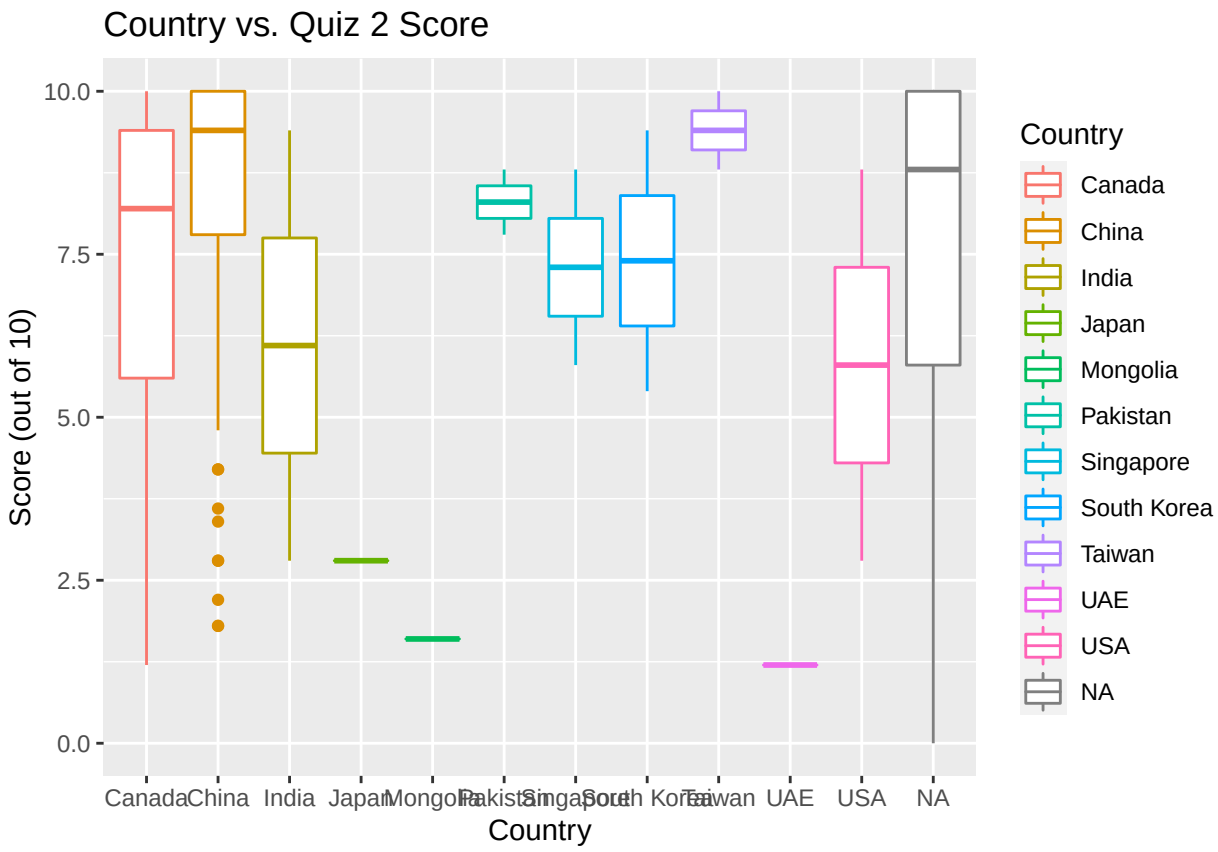
```
display_boxplot(cleaned_sta302_performance_data, Quiz_1_score,
                "Country vs. Quiz 1 Score", "Score (out of 10)")
```

```
## Warning: Removed 13 rows containing non-finite values (stat_boxplot).
```



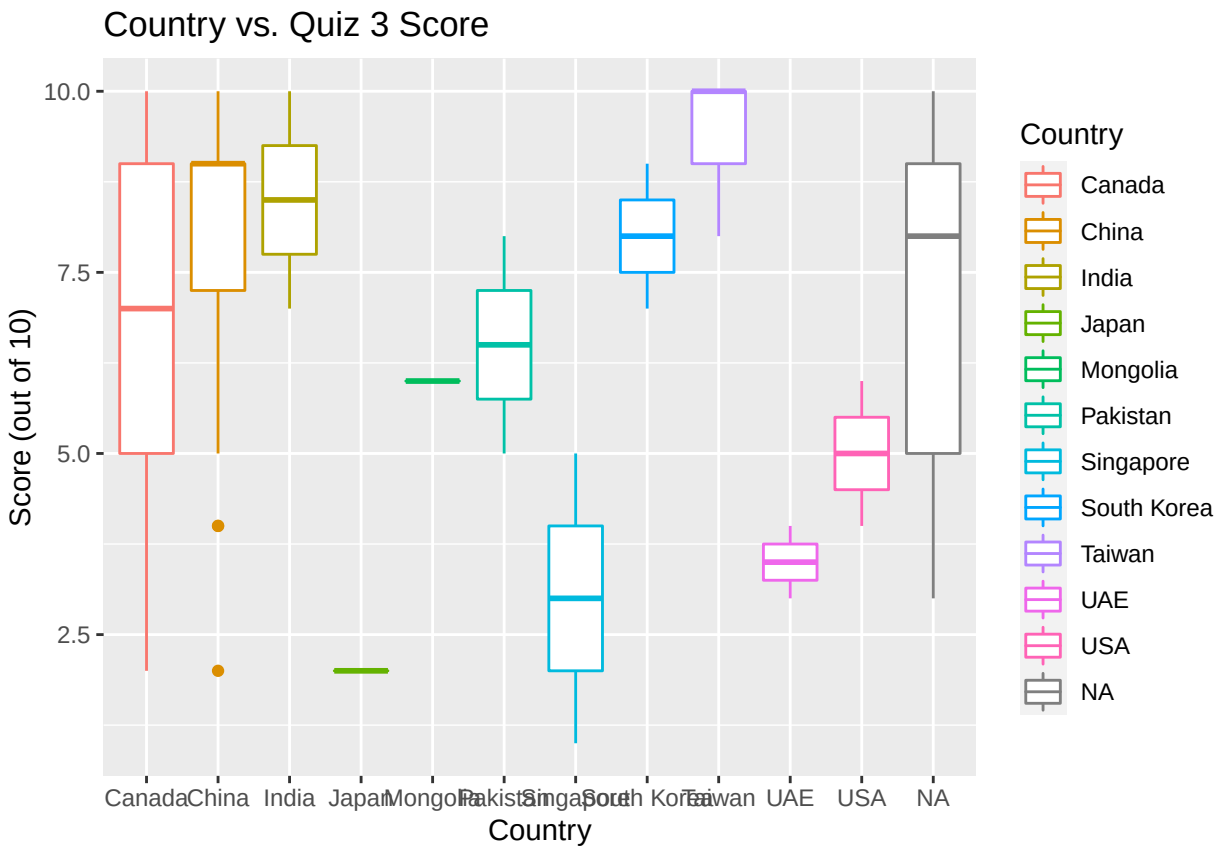
```
display_boxplot(cleaned_sta302_performance_data, Quiz_2_score,
                "Country vs. Quiz 2 Score", "Score (out of 10)")
```

```
## Warning: Removed 36 rows containing non-finite values (stat_boxplot).
```



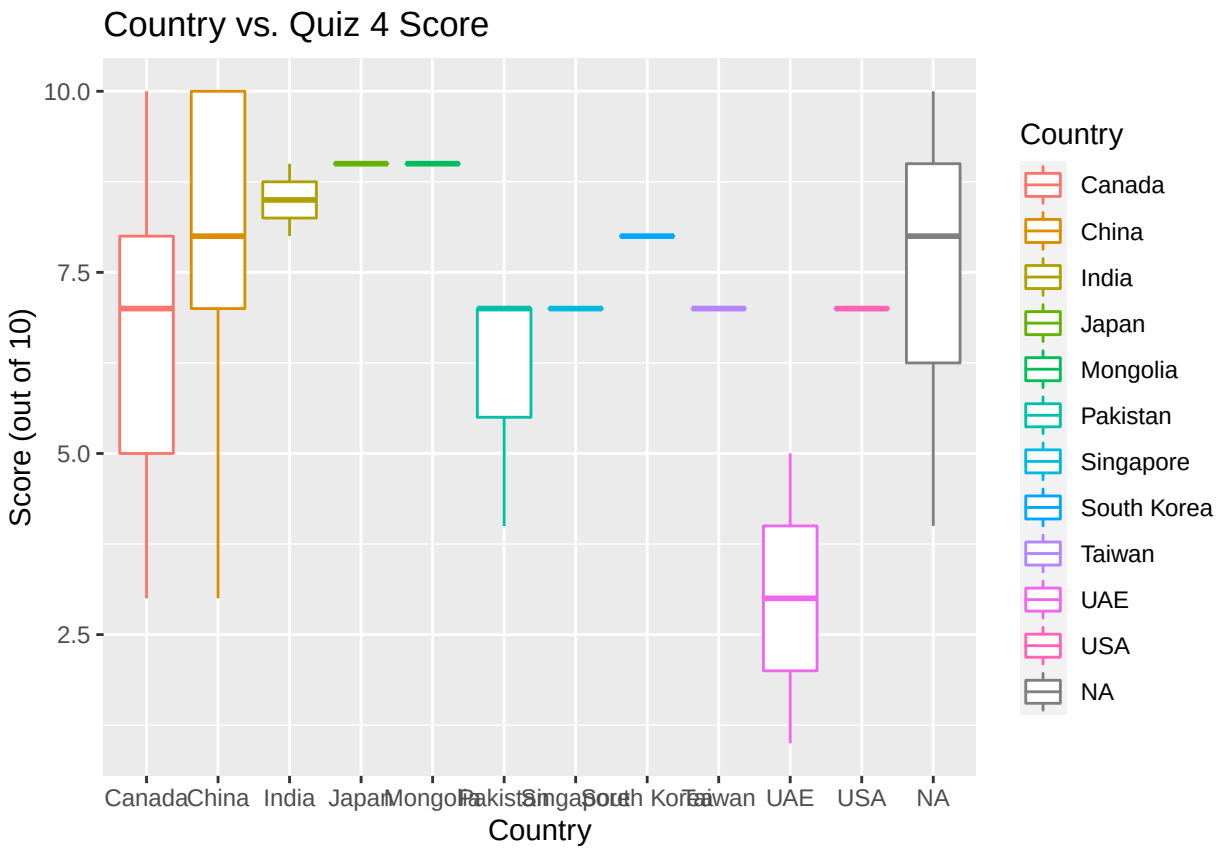
```
display_boxplot(cleaned_sta302_performance_data, Quiz_3_score,
                "Country vs. Quiz 3 Score", "Score (out of 10)")
```

```
## Warning: Removed 31 rows containing non-finite values (stat_boxplot).
```



```
display_boxplot(cleaned_sta302_performance_data, Quiz_4_score,
                "Country vs. Quiz 4 Score", "Score (out of 10)")
```

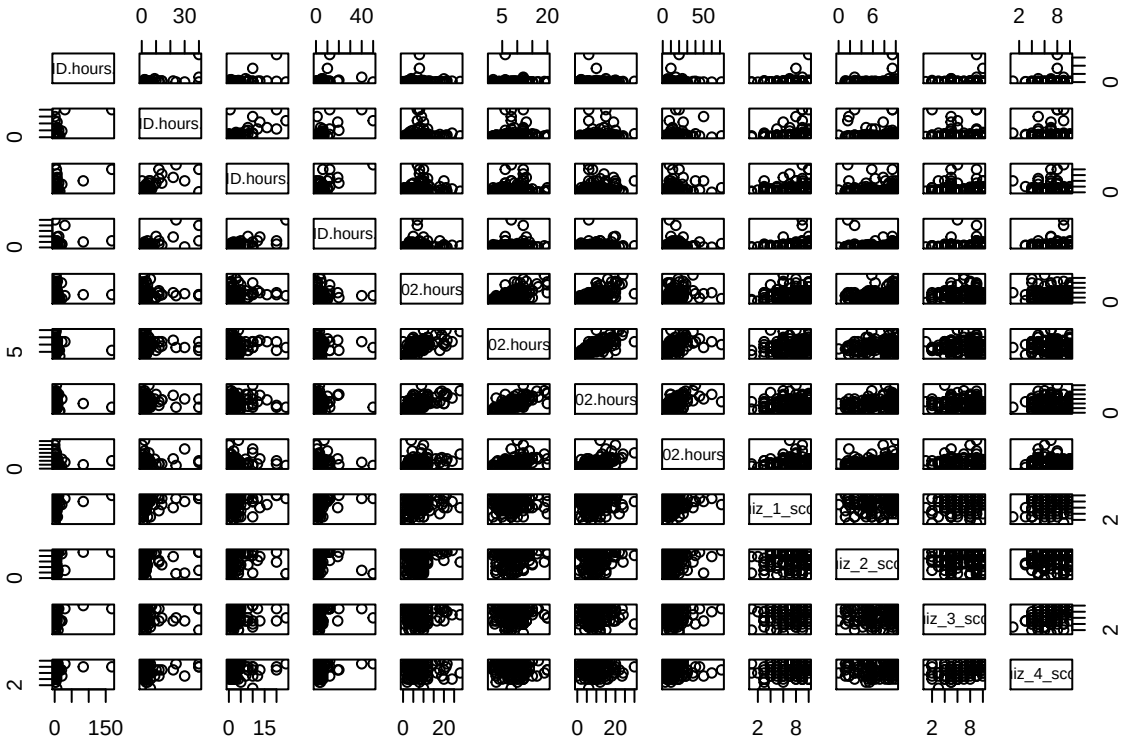
```
## Warning: Removed 34 rows containing non-finite values (stat_boxplot).
```



Scatterplots

Comprehensive pairwise scatterplot

```
pairs(~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,
      data = cleaned_sta302_performance_data)
```




```
## GGally
# ggpairs -- removes bottom half of pairs plot
# ggpairs(data = cleaned_sta302_performance_data)
```

Slightly Zoomed In Pairwise Scatterplots

```
# 1. Zoom in a bit by creating 3 - 4 pairs() functions:

# quiz4 ~ quiz 1, 2, 3
# covid4 ~ covid 1, 2, 3
# sta302h14 ~ sta302h1 1, 2, 3
```

Top 4 - 5 Intersting Scatterplots

```
# 2. Pick out 4 - 5 scatterplots that have interesting relationships


# 3. back up your choices with their correlation (R value).
```

Correlation Matrix

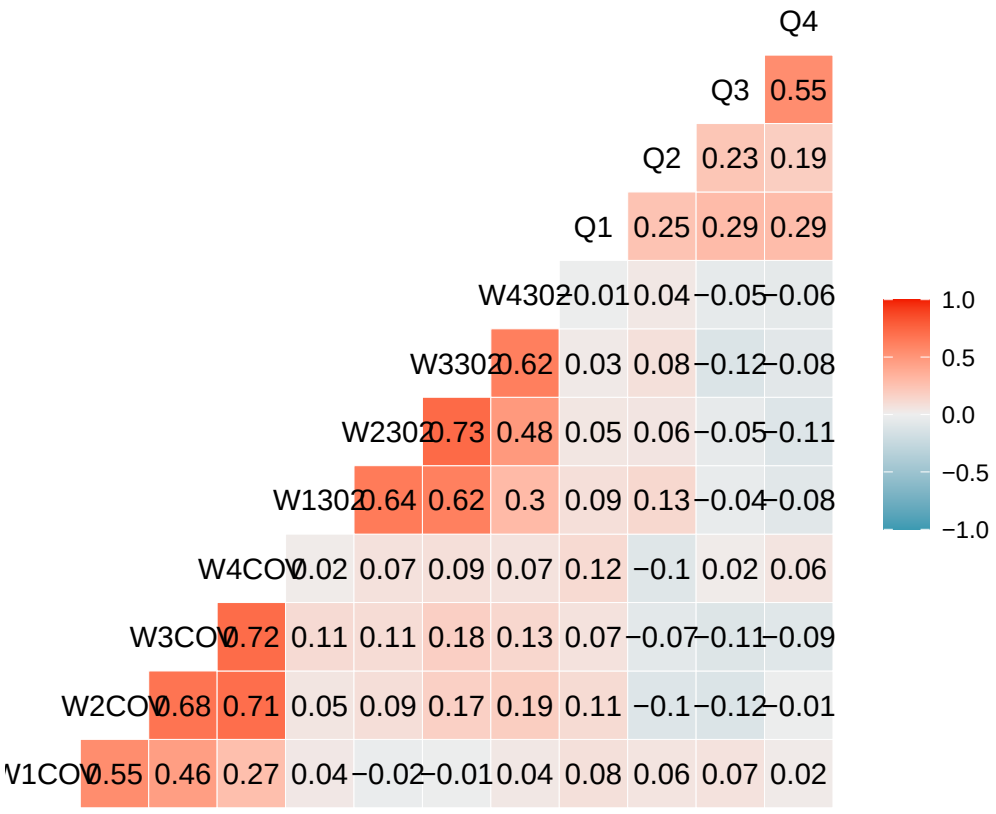
All Countries

We can find correlation matrix to determine candidate significant predictor values.

```
library(GGally)
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```
colnames(rows_with_no_country) <- c("W1COV", "W2COV", "W3COV", "W4COV",
                                    "W1302", "W2302", "W3302", "W4302",
                                    "Q1", "Q2", "Q3", "Q4")
ggcorr(rows_with_no_country, label = TRUE, label_round = 2)
```




```
round(cor(rows_with_no_country, use = "pairwise.complete.obs", method = "pearson"), 2)
```

```
##       W1COV W2COV W3COV W4COV W1302 W2302 W3302 W4302    Q1    Q2    Q3    Q4
## W1COV  1.00  0.55  0.46  0.27  0.04 -0.02 -0.01  0.04  0.08  0.06  0.07  0.02
## W2COV  0.55  1.00  0.68  0.71  0.05  0.09  0.17  0.19  0.11 -0.10 -0.12 -0.01
## W3COV  0.46  0.68  1.00  0.72  0.11  0.11  0.18  0.13  0.07 -0.07 -0.11 -0.09
## W4COV  0.27  0.71  0.72  1.00  0.02  0.07  0.09  0.07  0.12 -0.10  0.02  0.06
## W1302  0.04  0.05  0.11  0.02  1.00  0.64  0.62  0.30  0.09  0.13 -0.04 -0.08
```

```
## W2302 -0.02  0.09  0.11  0.07  0.64  1.00  0.73  0.48  0.05  0.06 -0.05 -0.11
## W3302 -0.01  0.17  0.18  0.09  0.62  0.73  1.00  0.62  0.03  0.08 -0.12 -0.08
## W4302  0.04  0.19  0.13  0.07  0.30  0.48  0.62  1.00 -0.01  0.04 -0.05 -0.06
## Q1     0.08  0.11  0.07  0.12  0.09  0.05  0.03 -0.01  1.00  0.25  0.29  0.29
## Q2     0.06 -0.10 -0.07 -0.10  0.13  0.06  0.08  0.04  0.25  1.00  0.23  0.19
## Q3     0.07 -0.12 -0.11  0.02 -0.04 -0.05 -0.12 -0.05  0.29  0.23  1.00  0.55
## Q4     0.02 -0.01 -0.09  0.06 -0.08 -0.11 -0.08 -0.06  0.29  0.19  0.55  1.00
```

By Individual Country

```
# TODO: You could also create separate correlation matrices for each country.
```

5-Number Summary Statistics

STA302H1 Hours 5-Number Summary

```
summary(sta302_performance_data$STA302.hours..W1.)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   0.000   4.000   7.000   7.458   9.000  28.000      26
```

```
summary(sta302_performance_data$STA302.hours..W2.)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   1.000   6.000   8.000   8.298  10.000  20.000      22
```

```
summary(sta302_performance_data$STA302.hours..W3.)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   0.000   6.000   9.000   9.225  11.500  30.000      20
```

```
summary(sta302_performance_data$STA302.hours..W4.)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    2.00    7.00   11.00   13.42   16.00   72.00      40
```

COVID Hours 5-Number Summary

```
summary(sta302_performance_data$COVID.hours..W1.)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   0.000   1.000   1.000   3.607   2.000 168.000      26
```

```
summary(sta302_performance_data$COVID.hours..W2.)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   0.000   1.000   1.000   2.884   2.000  40.000      22
```

```
summary(sta302_performance_data$COVID.hours..W3.)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   0.000   0.500   1.000   2.333   2.000  24.000      21
```

```
summary(sta302_performance_data$COVID.hours..W4.)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   0.000   1.000   1.500   2.918   3.000  50.000      40
```

Quiz Scores 5-Number Summary

```
summary(sta302_performance_data$Quiz_1_score)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   1.000   6.000   8.000   7.617   9.000  10.000      13
```

```
summary(sta302_performance_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      36
```

```
summary(sta302_performance_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      31
```

```
summary(sta302_performance_data$Quiz_4_score)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   1.000   6.000   8.000   7.378   9.000  10.000      34
```