Student Exam Performance

Connor Gaudette

10/26/2021

First we will download and install the packages needed for this analysis.

```
install.packages("tidyverse")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.1'
## (as 'lib' is unspecified)
```

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.5     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.0.2     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(ggplot2)
library(dplyr)
library(tidyr)
```

Next we will load out dataset.

```
rough_data <- read_csv("StudentsPerformance.csv")
```

```
## Rows: 1000 Columns: 8

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (5): gender, race/ethnicity, parental level of education, lunch, test pr...
## dbl (3): math score, reading score, writing score

##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

After looking at a summary of the rough data, we first will ensure there are no duplicates. Next we will create a new column "total" which is the sum of all three individual math, reading, and writing scores. This total score out of 300 will be used for analysis.

```
rough_data %>%
  unique() %>%
  mutate(`math score` = as.numeric(`math score`),
         `reading score` = as.numeric(`reading score`),
         `writing score` = as.numeric(`writing score`))
```

```
## # A tibble: 1,000 x 8
##    gender `race/ethnicity` `parental level ~ lunch `test preparati~ `math score`
##    <chr>  <chr>            <chr>             <chr> <chr>                   <dbl>
##  1 female group B          bachelor's degree stan~ none                       72
##  2 female group C          some college      stan~ completed                  69
##  3 female group B          master's degree   stan~ none                       90
##  4 male   group A          associate's degr~ free~ none                       47
##  5 male   group C          some college      stan~ none                       76
##  6 female group B          associate's degr~ stan~ none                       71
##  7 female group B          some college      stan~ completed                  88
##  8 male   group B          some college      free~ none                       40
##  9 male   group D          high school       free~ completed                  64
## 10 female group B          high school       free~ none                       38
## # ... with 990 more rows, and 2 more variables: reading score <dbl>,
## #   writing score <dbl>
```

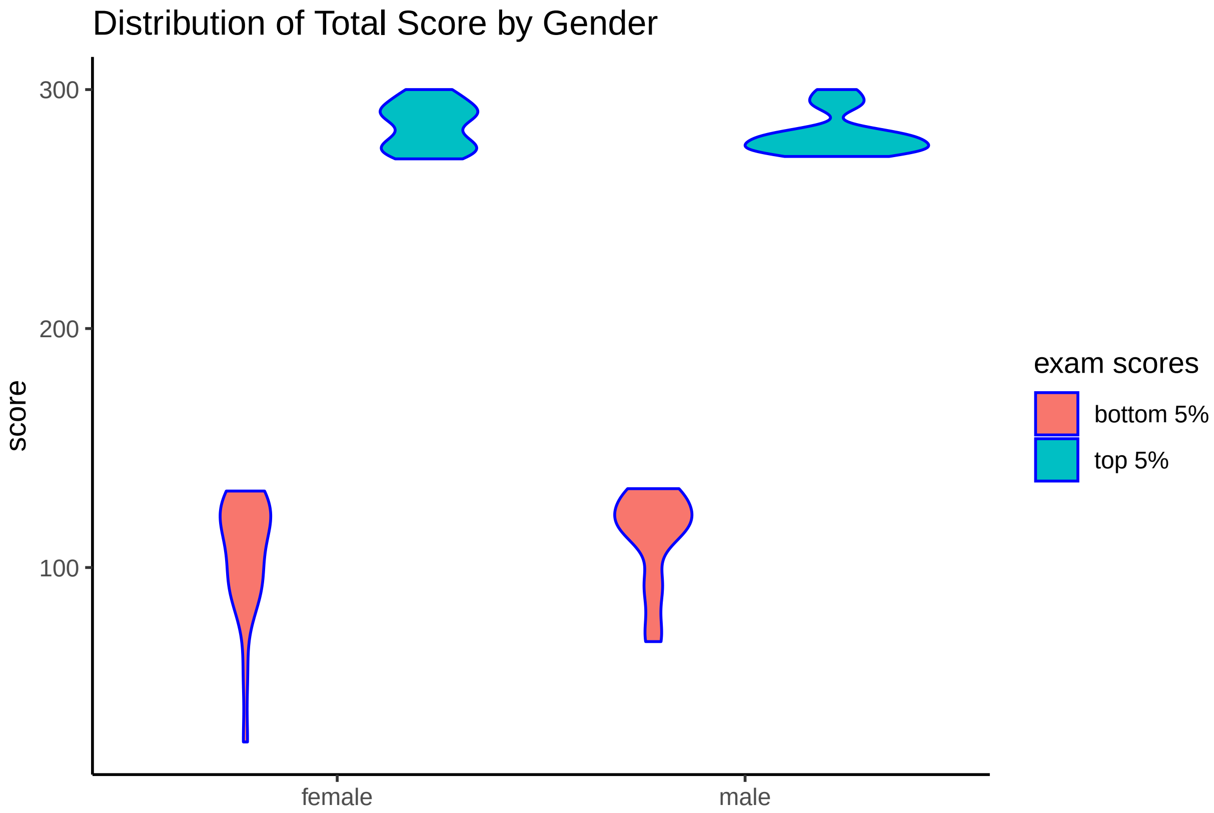
```
total_scores <- rough_data %>%
  select(`math score`, `reading score`, `writing score`)

exam_data <- rough_data %>%
  mutate(total = rowSums(total_scores, na.rm = FALSE)) %>%
  arrange(desc(total))
```

For this analysis, we will take a different approach and look only at the top 5% and the bottom 5% of reported exam scores. This analysis will allow us to directly compare the top performing students against the worst performing students to discover any significant difference between their parents' level of education, preperatory course completion, and lunch type provided to them. We begin by creating a subset of data including only the top and bottom 5% of scores.

```
top_bottom <- exam_data %>%
  arrange(desc(total)) %>%
  slice(1:50, 951:1000) %>%
  select(gender, `parental level of education`,total) %>%
  rename(parents = "parental level of education") %>%
  mutate(group = ifelse(total >= 271, "top", "bottom")) %>%
  mutate(parent_edu_level = ifelse(parents == "master's degree", "1",
                               ifelse(parents == "bachelor's degree", "2",
                                      ifelse(parents == "associate's degree", "3",
                                             ifelse(parents == "some college", "4",
                                                    ifelse(parents == "high school", "5",
                                                           ifelse(parents == "some high school", "6", "
```

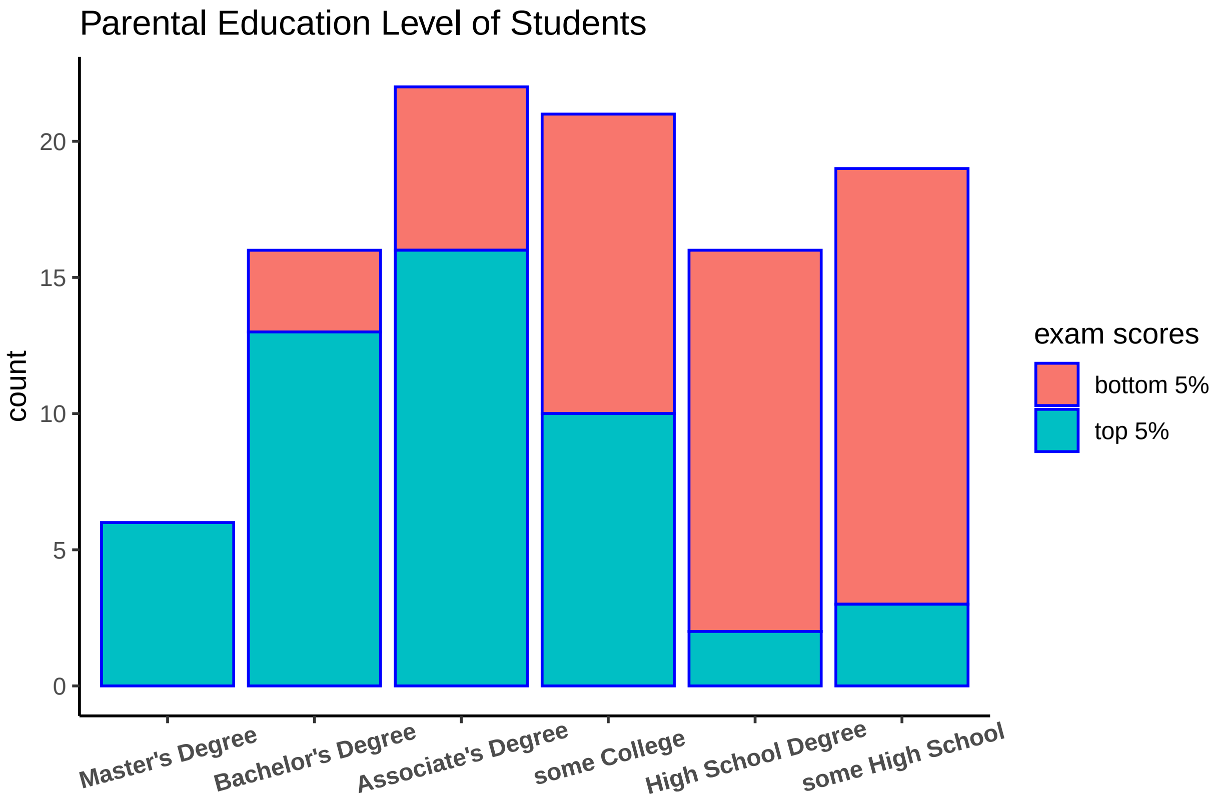
First let's determine if there is any correlation between gender and exam score amongst each group.

```
ggplot() +
  geom_violin(data = top_bottom, aes(x = gender, y = total, fill = group), color = "blue") +
  theme_classic() +
  ggtitle(label = "Distribution of Total Score by Gender") +
  labs(fill = "exam scores",
       y = "score",
       x = "") +
  scale_fill_discrete(labels = c("bottom 5%", "top 5%"))
```



Next let's compare the distribution of student's parental level of education between the top and bottom 5%.

```
ggplot() +
  geom_bar(data = top_bottom, aes(x = parent_edu_level, fill = group), stat = "count", color = "blue", 
  scale_x_discrete(labels=c("1" = "Master's Degree", "2" = "Bachelor's Degree","3" = "Associate's Degre
  theme_classic() +
  theme(axis.text.x = element_text(angle = 15, vjust = 0.7, face = "bold")) +
  ggtitle(label = "Parental Education Level of Students") +
  labs(x = "",
       fill = "exam scores") +
  scale_fill_discrete(labels = c("bottom 5%", "top 5%"))
```

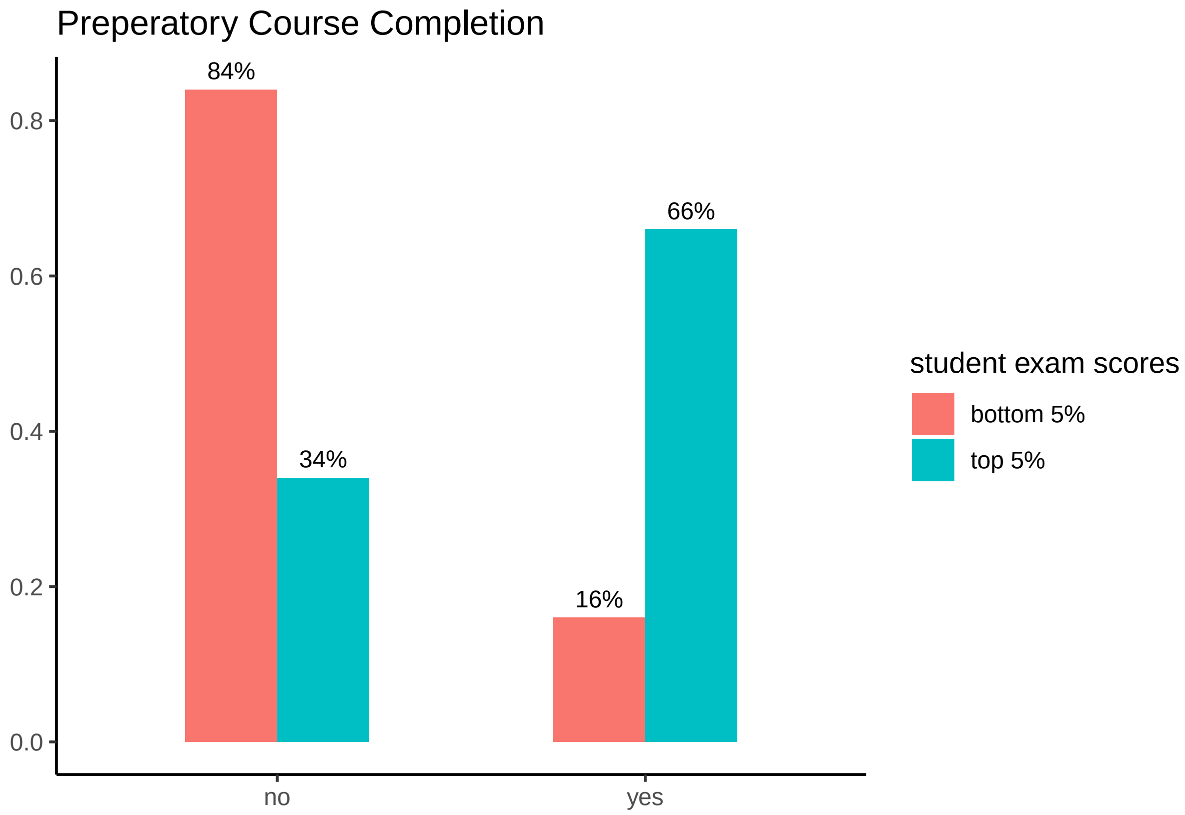


Next we will create a new subset to allow us to compare the percent of students who completed a preperatory course in both the top and bottom 5%.

```
test_prep <- exam_data %>%
  arrange(desc(total)) %>%
  slice(1:50, 951:1000) %>%
  mutate(group = ifelse(total >= 271, "top", "bottom")) %>%
  mutate(completed = ifelse(`test preparation course`== "completed", "yes", "no")) %>%
  select(group, `test preparation course`, completed)

test_prep_sum <- test_prep %>%
  group_by(group) %>%
  count(completed) %>%
  mutate(pct = (n/50))
```

A bar graph will effectively compare preperatory course completion amongst both groups.

```
ggplot(test_prep_sum, aes(x = completed, y = pct, fill = group)) +
  geom_bar(stat="identity", width=0.5, position = "dodge") +
  theme_classic() +
  ggtitle(label = "Preperatory Course Completion") +
  labs(x = "",
       y = "",
       fill = "student exam scores") +
  scale_fill_discrete(labels = c("bottom 5%", "top 5%")) +
  geom_text(aes(label=scales::percent(pct)), size = 3, position = position_dodge(width = 0.5), vjust = -
```

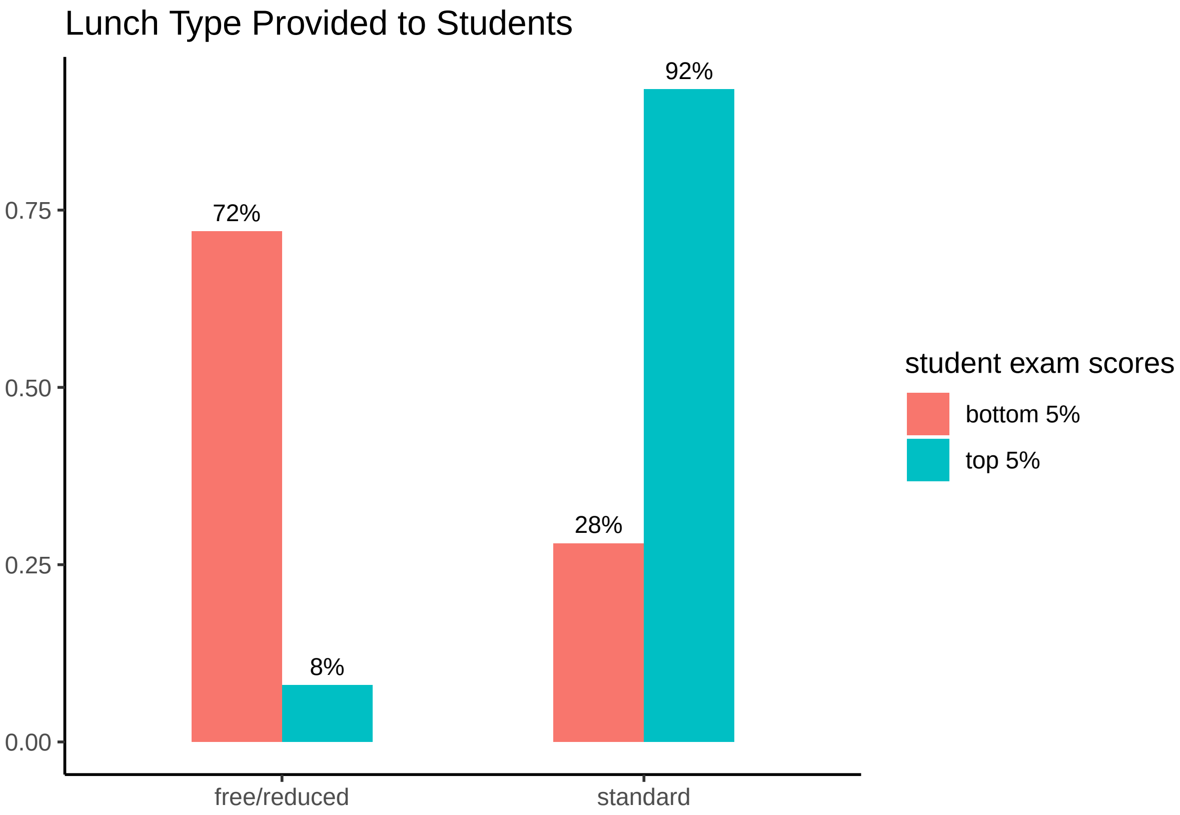


Lastly, we will repeat the last step only this time we will compare lunch type.

```
lunch <- exam_data %>%
  arrange(desc(total)) %>%
  slice(1:50, 951:1000) %>%
  mutate(group = ifelse(total >= 271, "top", "bottom")) %>%
  group_by(group) %>%
  count(lunch) %>%
  mutate(pct = (n/50)) %>%
  select(group, lunch, n, pct)
```

Again, a bar graph will effectively compare lunch type amongst both groups

```
ggplot(lunch, aes(x = lunch, y = pct, fill = group)) +
  geom_bar(stat="identity", width=0.5, position = "dodge") +
  theme_classic() +
  ggtitle(label = "Lunch Type Provided to Students") +
  labs(x = "",
       y = "",
       fill = "student exam scores") +
  scale_fill_discrete(labels = c("bottom 5%", "top 5%")) +
  geom_text(aes(label=scales::percent(pct)), size = 3, position = position_dodge(width = 0.5), vjust = -
```


Lunch Type Provided to Students
0.00
0.25
0.50
0.75
72%
8%
28%
92%
free/reduced
standard
student exam scores
bottom 5%
top 5%