~~~
---
title: "Project 1"
author: "Justin Williams"
date: "`r Sys.Date()`"
output: html_document
---

Data, packages, etc.

```{r}
#load libraries and data
library(tidyverse)
library(dplyr)

#load vector
scores <- c(12, 16, 18, 14, 13, 15, 12, 13, 11, 17, 10, 13, 14)

#load data
hrdata_df <- read.csv("/Users/justinwilliams/Code/9050advresearch/Project
1/HRData_New.csv")
```

# 1. I give a 10-item test of Clemson University trivia to 13 people. Their scores are as
follows: 12 16 18 14 13 15 12 13 11 17 10 13 14
## a. Calculate the mean, median, and the mode.
## b. Using the mean as your parameter estimate for your model, calculate the number of
errors, the sum of the absolute errors, and the sum of the squared errors.
## c. Now use the median as your parameter estimate and calculate the same three error
terms.
## d. Use the mode as your parameter estimate and calculate the same three error terms.

See the attached Excel file titled Project 1_Question.xslx for the questions a, b, c, and
d.

## e. Look across your answers to (b), (c), and (d) to see for each error term, which
estimate gives the lowest error? That is, which of the mean, median, and mode minimizes
the sum of squared errors and by how much? Which one minimizes the sum of absolute errors
and by how much? And which one minimizes the count of errors and by how much?

* The estimate that gives the lowest error is the mean parameter as the sum of squared
errors is 64.77 compared to 71 for both the median and mode parameter estimates.
* The parameter that minimizes the sum of absolute errors are both the median and mode,
which is 23 compared to 23.69 for the mean parameter.
* The parameter that minimizes the count of errors are both median and mode at 10.


# 2. Use RStudio or Jamovi to obtain the mean and median for the data in the previous
question and share a screenshot of the results.
```{r}
mean_score <- mean(scores)
median_score <- median(scores)
calculate_mode <- function(x) {
    ux <- unique(x)
    ux[which.max(tabulate(match(x, ux)))]
}
mode_score <- calculate_mode(scores)
```
```{r}
print(paste("Mean score:", format(round(mean_score, 2), nsmall = 2)))
print(paste("Median score:", median_score))
print(paste("Mode score:", mode_score))
```
~~~

## b. Using the mean as your parameter estimate for your model, calculate the number of errors, the sum of the absolute errors, and the sum of the squared errors.

```{r}
errors_mean <- scores - mean_score
num_errors <- sum(errors_mean != 0)
sum_abs_errors_mean <- sum(abs(errors_mean))
sum_sq_errors_mean <- sum(errors_mean^2)
```

```{r}
print(paste("Sum of absolute errors:", sum_abs_errors_mean))
print(paste("Number of errors:", num_errors))
print (paste("Sum of the Squared Errors:", sum_sq_errors_mean))
```

## c. Now use the median as your parameter estimate and calculate the same three error terms.

```{r}
errors_median <- scores - median_score
num_errors <- sum(errors_median != 0)
sum_abs_errors_median <- sum(abs(errors_median))
sum_sq_errors_median <- sum(errors_median^2)
```

```{r}
print(paste("Number of errors:", num_errors))
print(paste("Sum of absolute errors:", sum_abs_errors_median))
print (paste("Sum of the Squared Errors:", sum_sq_errors_median))
```

## d. Use the mode as your parameter estimate and calculate the same three error terms.

```{r}
errors_mode <- scores - mode_score
num_errors <- sum(errors_mode != 0)
sum_abs_errors_mode <- sum(abs(errors_mode))
sum_sq_errors_mode <- sum(errors_mode^2)
```

```{r}
print(paste("Number of errors:", num_errors))
print(paste("Sum of absolute errors:", sum_abs_errors_mode))
print (paste("Sum of the Squared Errors:", sum_sq_errors_mode))
```

## e. Look across your answers to (b), (c), and (d) to see for each error term, which estimate gives the lowest error? That is, which of the mean, median, and mode minimizes the sum of squared errors and by how much? Which one minimizes the sum of absolute errors and by how much? And which one minimizes the count of errors and by how much?

* The estimate that gives the lowest error is the mean parameter as the sum of squared errors is 64.77 compared to 71 for both the median and mode parameter estimates.
* The parameter that minimizes the sum of absolute errors are both the median and mode, which is 23 compared to 23.69 for the mean parameter.
* The parameter that minimizes the count of errors are both median and mode at 10.

# 3. “HR Data.csv” contains 311 rows and 36 columns. First, open this data file in RStudio or Jamovi. Next, obtain estimates of the mean, median, variance, and standard deviation of two variables (SALARY, ENGAGEMENTSURVEY). Report these along with a one sentence interpretation of what the values mean to you.

## Salary analysis

```{r}
hrdata_sal_mean <- mean(hrdata_df$Salary)
hrdata_sal_median <- median(hrdata_df$Salary)
hrdata_sal_var <- var(hrdata_df$Salary)
hrdata_sal_stddev <- sd(hrdata_df$Salary)
```

```{r}
```

~~~
print(paste("Mean salary:", format(round(hrdata_sal_mean, 2), nsmall = 2)))
print(paste("Median salary:", format(round(hrdata_sal_median, 2), nsmall = 2)))
print(paste("Variance of salary:", hrdata_sal_var))
print(paste("Standard deviation of salary:", hrdata_sal_stddev))
```

## Engagement statistical analysis
```{r}
hrdata_sal_mean <- mean(hrdata_df$Engagement)
hrdata_sal_median <- median(hrdata_df$Engagement)
hrdata_sal_var <- var(hrdata_df$Engagement)
hrdata_sal_stddev <- sd(hrdata_df$Engagement)
```
```{r}
print(paste("median Engagement:", hrdata_sal_median))
print(paste("Median Engagement:", hrdata_sal_median))
print(paste("Variance of Engagement:", hrdata_sal_var))
print(paste("Standard deviation of Engagement:", hrdata_sal_stddev))
```

## My interpretation of the Salary descriptive statistics and the Engagement Survey
descriptive statistic
* The variance is large which suggests that the data is spread out from the mean salary of
~$69,020.68. The standard deviation of $25,156.64 shows the average distance of each data
point from the mean salary, which seems large as well. The median salary of $62,810.00 is
lower than the mean salary of $69,020.68, indicating that the salary data is skewed to the
right, with more employees earning lower salaries than higher salaries.
* The median engagement score of 3.5 is the same as the mean engagement score, indicating
that the data is normally distributed.

# 4. Compare the mean SALARY for Men and Women. Briefly describe these results, being sure
to indicate whether it APPEARS (we don't know how to do the formal test yet) that it would
be useful to make predictions ofb SALARY conditional on employee sex (M or F)

```{r}
hrdata_sal_mean_men <- mean(hrdata_df$Salary[hrdata_df$Sex == "M"])
hrdata_sal_mean_women <- mean(hrdata_df$Salary[hrdata_df$Sex == "F"])
print(paste("Men's mean salary:", format(round(hrdata_sal_mean_men,2), nsmall = 2)))
print(paste("Women's mean salary:", format(round(hrdata_sal_mean_women, 2), nsmall = 2)))
```

## Interpretation
### The mean salary for men is $70,629.40 and $67,786.72 for women. There is a small
difference between the two means, it appears that it could  be useful to make predictions
of salary conditional on gender as the difference may be statistically significant.
~~~