# STA238 - Winter 2021

Assignment 2

Wei-Han Wang - [redacted]

February 12, 2021

## Part 1

### Step 1 (Mathematical Justification)

Assume that $X_1, ..., X_n \overset{iid}{\sim} Normal(\mu = 0, \sigma^2)$ $\mu$ represents sample finite mean; $\sigma^2$ represents sample finite variance. We have two estimators of $\sigma^2$ here: first one is $T_1$ and second one is $T_2$.

$$T_1 = S^2 = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \bar{X_n})^2$$

$$T_2 = S_*^2 = \frac{1}{n}\sum_{i=1}^{n}(X_i - \bar{X_n})^2$$

Let us begin froom finding the expectation value for $T_1$.

$$E[\bar{X}] = \frac{1}{n-1}\sum_{i=1}^{n}E[(X_i - \bar{X_n})^2]$$

We know from Chapter 13.1 of MIPS that $E[\bar{X_n}] = \mu$ and we know $E[(X_i - \bar{X_n})^2] = E[X_i] - E[\bar{X_n}] = 0$ by the property of expectation from Chapter 7 of MIPS (Dekking).

Suppoose $Y = (X_i - \bar{X_n})$.

By the alternative expression of variance, $Var(Y) = E[Y]^2 - E[Y^2] = E[Y]^2$ since E[Y] = 0.

Substitute $Y = (X_i - \bar{X_n})$ into Var(Y) then we have

$$Var(X_i - \bar{X_n})^2) = E[(X_i - \bar{X_n})^2]$$

Substitute this equation back into our $T_1$.

$Var(X_i - \bar{X_n})^2) = Var(\frac{n}{n-1}X_i - \sum_{i=1}^{n}\frac{1}{n}\bar{X_n})^2$

$Var(X_i - \bar{X_n})^2 = (\frac{n}{n-1})^2 Var(X_i) - (\frac{1}{n})^2 Var\bar{X_n}$ by the property of variance under change of units from Chapter 7.4.

$Var(X_i - \bar{X_n})^2) = [(\frac{n}{n-1})^2 + (\frac{1}{n})^2]\sigma^2$ by definition of variance of a normal distribution where $Var(X) = \sigma^2$.

$Var(X_i - \bar{X_n})^2) = \frac{n-1}{n}\sigma^2$

Substitute the result from previous line into $T_1$ formula.

$$T_1 = \frac{1}{n-1}\sum_{i=1}^{n}E[(X_i - \bar{X_n})^2] = \frac{1}{n-1}n\frac{n-1}{n}\sigma^2 = \sigma^2$$

We have reached to the answer where $T_1 = \sigma^2$ where we know $\sigma^2$ is the unbiased estimator.

Now let's calculate what answer we will get to with $T_2$. We know from the given hint of Assignment 2 Instructions that $T_2 = \frac{n-1}{n} T_1$.

We can directly substitute $T_1$ we derived from last part into this equation.

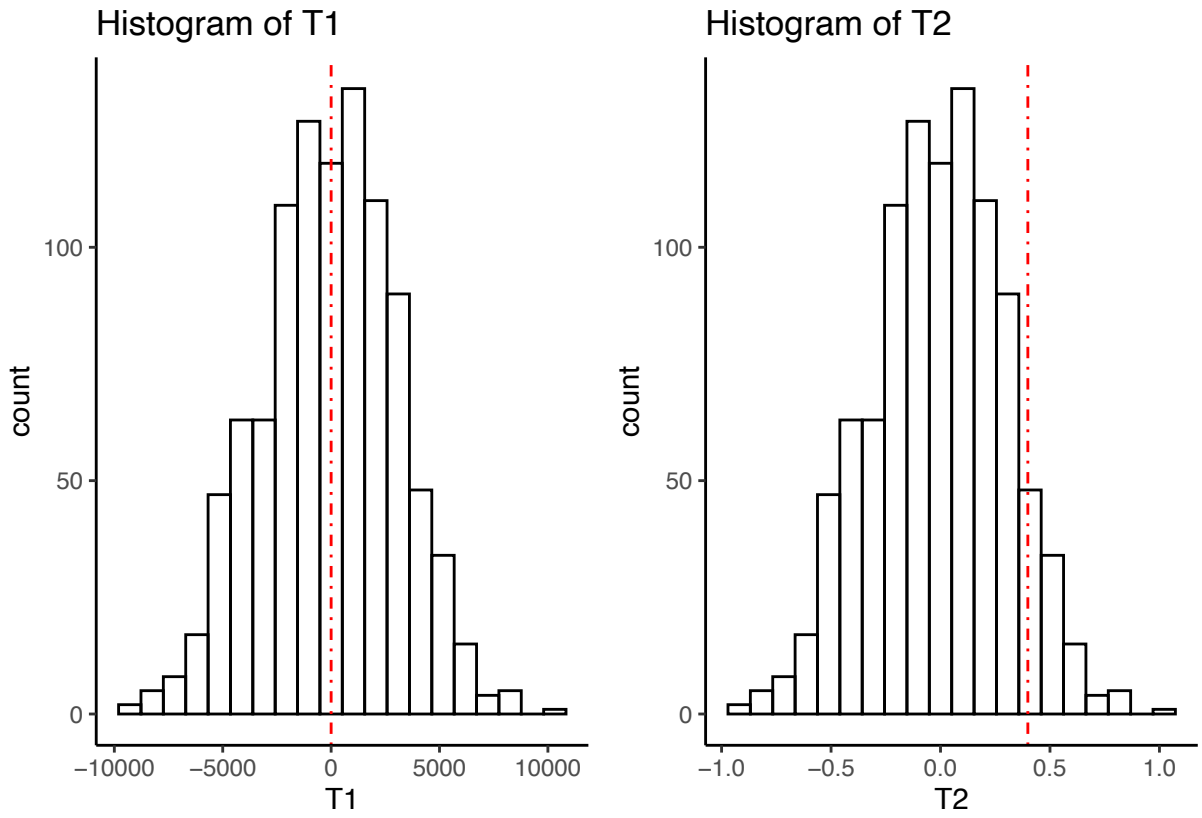
$$E[T_2] = \frac{n-1}{n} E[T_1]$$

Since we know $E[T_1] = \sigma^2$, then $E[T_2] = \frac{n-1}{n}\sigma^2$

From the result we know $T_2$ is a biased estimator.

## Step 2 (Simulation Justification)

```
set.seed(346)
# Simulate 1000 random samples of size 100
M <- 1000
n <- 100
sg <- 3
mean <- 0
p0 <- exp(mean/2*sg^2)/sqrt(2*pi) # True value of p0

#Functions of x corresponding to T1 and T2
compute_T1 <- function(x){(n-1/n)^2*(mean(x))}
compute_T2 <- function(x){((n-1)/n)*mean(x)}

# Simulate the samples and calculate the estimators for each sample
samples <- vector(mode = "list",length = M)
#Let samples be a list with length M
T1 <- T2 <- numeric(M)
for (i in 1:M) {
  samples[[i]] <- rnorm(n, 0, sg)
  T1[i] <- compute_T1(samples[[i]])
  T2[i] <- compute_T2(samples[[i]])
}

# Create the plots
plt_T1 <- tibble(T1 = T1) %>%
  ggplot(aes(x = T1)) +
  theme_classic() +
  geom_histogram(colour = "black",fill = "transparent", bins = 20) +
  geom_vline(xintercept = p0,colour = "red",linetype = "dotdash") +
  ggtitle("Histogram of T1")

plt_T2 <- tibble(T2 = T2) %>%
  ggplot(aes(x = T2)) +
  theme_classic() +
  geom_histogram(colour = "black",fill = "transparent", bins = 20) +
  geom_vline(xintercept = p0,colour = "red",linetype = "dotdash") +
  ggtitle("Histogram of T2")

plt_T1 | plt_T2
```




The goal of the side-by-side graph is to determine which function is bias. As we can see, the red dotted line represent $p_0$, the parameter. For $T_1$ graph we see the values are centered and most concentrated around the dotted line $p_0$ whereas $T_2$ graph the values are *not* concentrated around the dotted line $p_0$. Therefore, $T_1$ is the preferred estimator for the parameter $\sigma^2$ (Alison).

```
set.seed(346)
n = 100
M = 1000
T1 <- function(x){(1/n)*mean(x)}
T2 <- function(x){(n-1/n)*mean(x) - (1/n)*mean(x)}
storage <- list(
  T1 = numeric(M),
  T2 = numeric(M)
)
for (i in 1:M) {
  thesample <- sample.int(M,n,replace = FALSE)
  storage$T1[i] <- T1(thesample)
  storage$T2[i] <- T2(thesample)
}
```

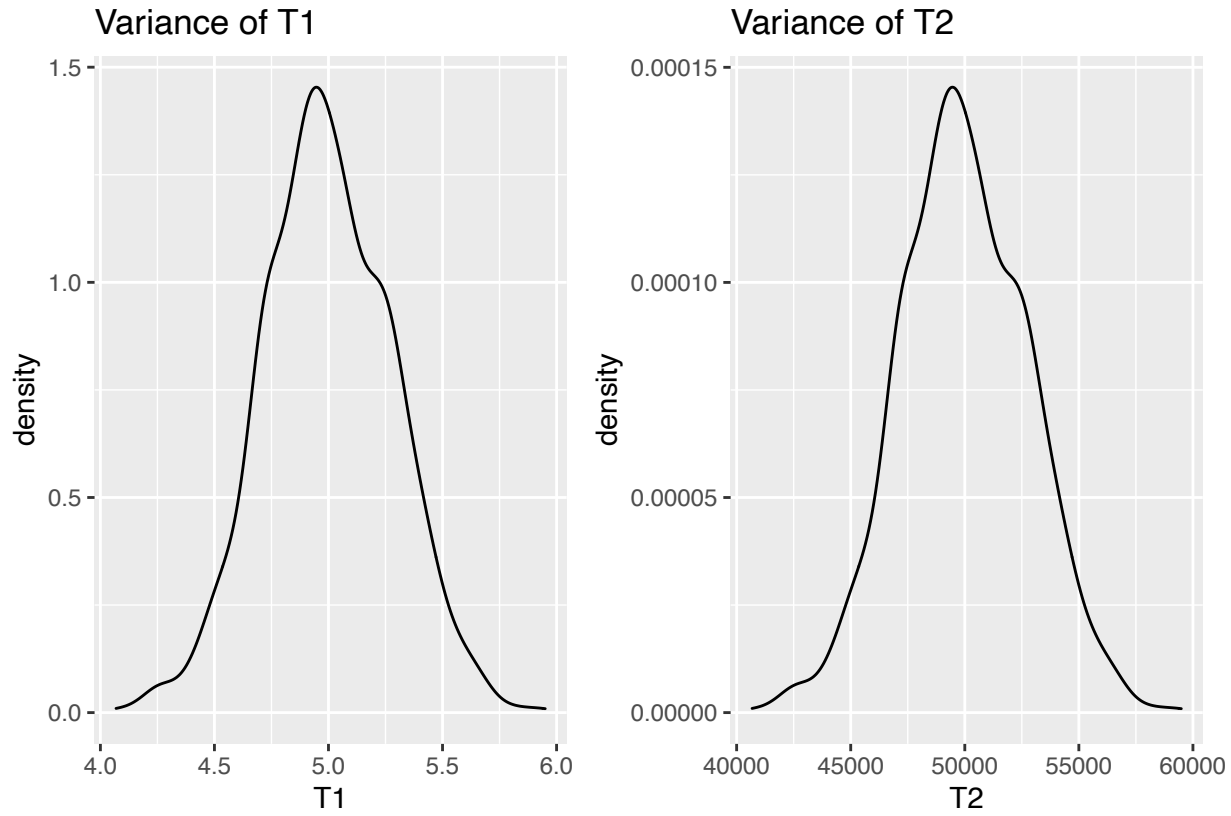
```r
# Evaluate the VAR of T1 and T2:
var(storage$T1)
```

```
## [1] 0.07853105
```

```r
var(storage$T2)
```

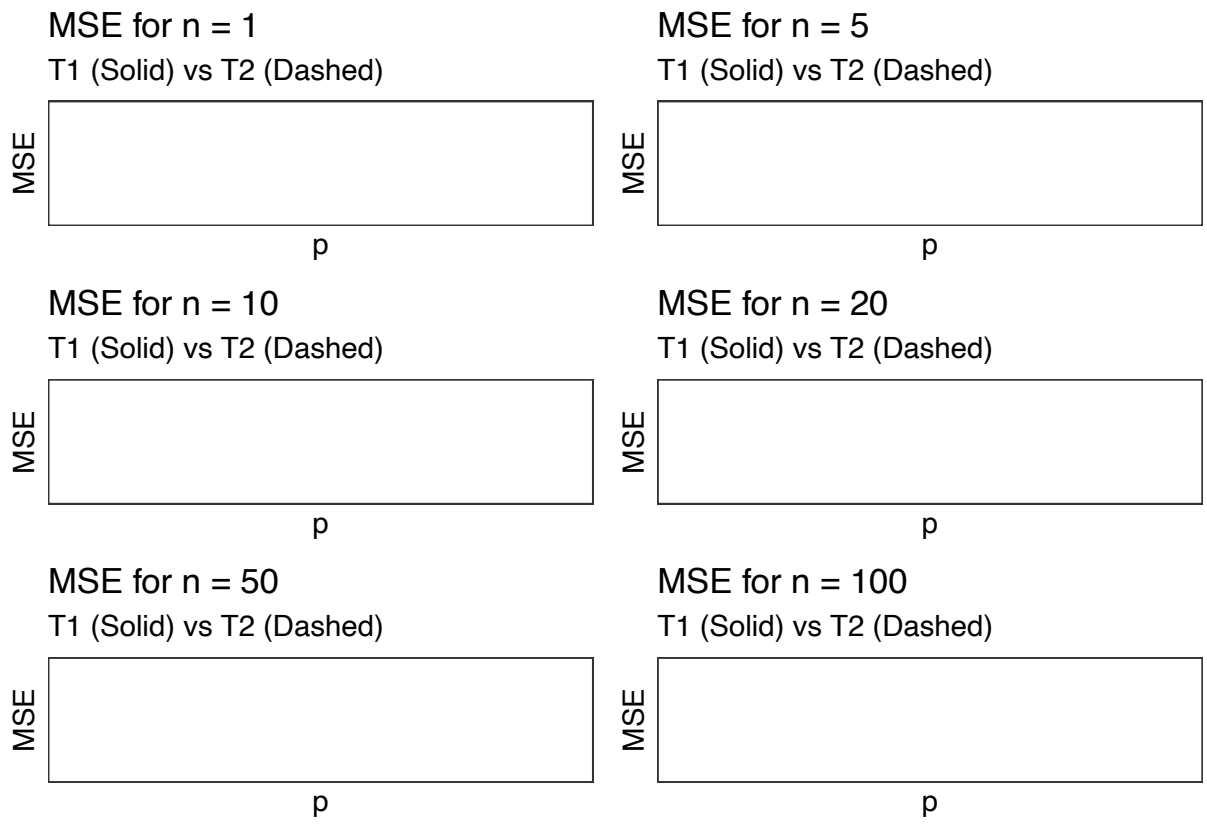
```
## [1] 7849964
```

```r
T1 <- tibble(T1 = storage$T1) %>%
  ggplot(aes(x = T1)) +
  geom_density() +
  coord_cartesian() +
  ggtitle("Variance of T1")

T2 <- tibble(T2 = storage$T2) %>%
  ggplot(aes(x = T2)) +
  geom_density() +
  coord_cartesian() +
  ggtitle("Variance of T2")

T1 | T2
```




The two graphs above demonstrate the variance of T1 and T2 estimators. However, the variance is really small and it cannot be seen clearly unless we call out the numbers. From our results, we see $Var(T_1) < Var(T_2)$ and by the definition of efficiency, $Var(T_1)$ is more efficient than $Var(T_2)$ (Alison).

```r
# Define a function for both
MSE_T1 <- function(n) sg
MSE_T2 <- function(n) -sg/n
make_plot <- function(n) {
  tibble(x = c(0,1)) %>%
    ggplot(aes(x = x)) +
    theme_bw() +
    stat_function(fun = MSE_T1,args = list(n = n)) +
    stat_function(fun = MSE_T2,args = list(n = n),linetype = 'dashed') +
    labs(title = paste0("MSE for n = ",n),
         subtitle = "T1 (Solid) vs T2 (Dashed)",
         x = "p",
         y = "MSE"
    )
}


(make_plot(1) | make_plot(5)) /
  (make_plot(10) | make_plot(20)) /
  (make_plot(50) | make_plot(100))
```




We have computed the bias, variance, and mean squared error of our estimator $T_1$ and $T_2$. Through the three methods, we've visually see from first set of graph on biasness of both estimators that $T_1$ is the unbiased estimator and $T_2$ is the biased one. Thus proving our calculation from Step 1 was correct. From the graph of variance, we used the definition of efficiency to conclude that $T_1$ can narrow down on the parameter of interest more efficiently than $T_2$ would.

# Part 2

## Model

In the world of statistics, there are many models used to analyze and visualize data. One of the models is called linear regression model. Using the linear regression and it allows us to see the relationship between two quantitative variables, which are values that are countable and measurable in a dataset. The below is a standard equation for a linear regression model.

$$Y = \beta_0 + \beta_1 X_i + \epsilon_i$$

$\beta_0$ represents the x-intercept when $X_i = 0$. $\beta_1$ is the slope of the linear regression plot. The epsilon $\epsilon_i$ represents the error between each input from the linear regression line. It is the flunctuation to the linear regression line. Y represents the dots on the linear regression plot, which are the dependent variable of this model.

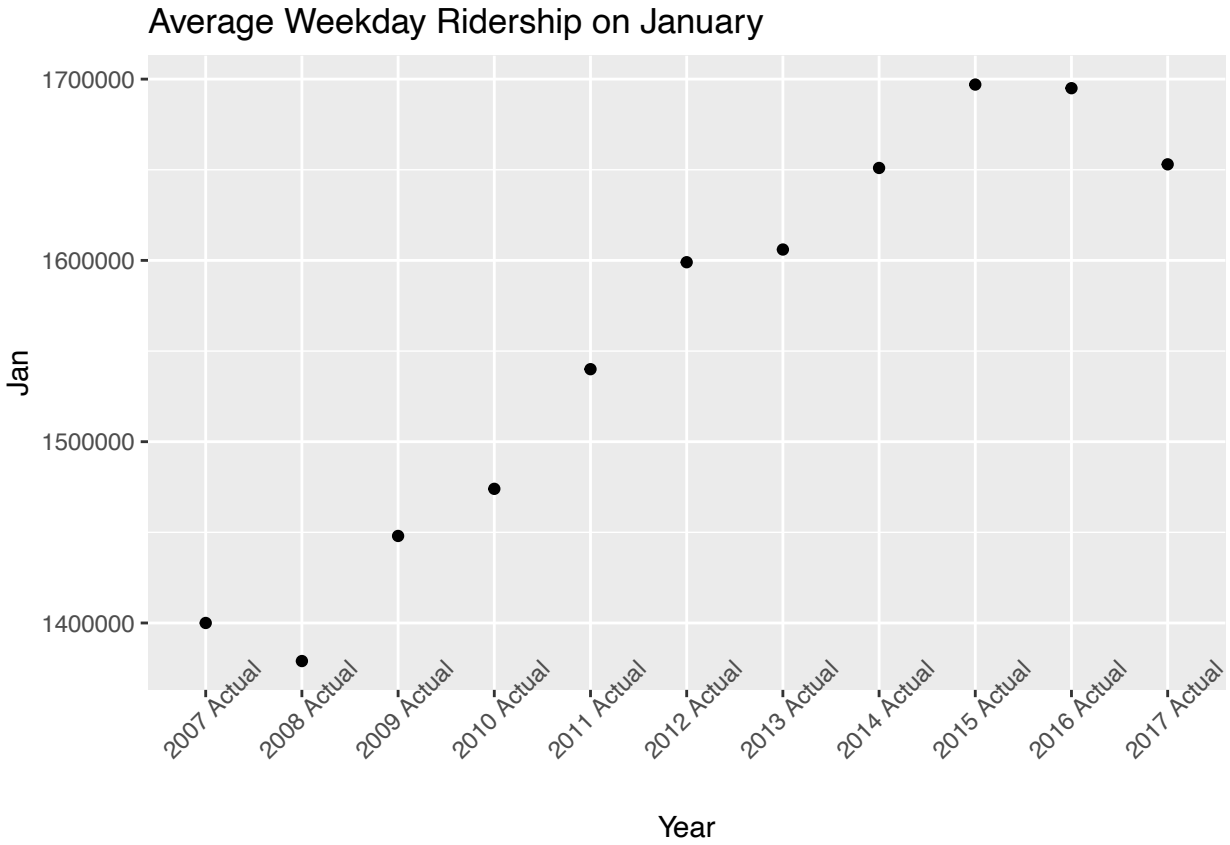
The dataset being evaluated is TTC Average Weekday Ridership 2007 to 2017 recorded by months. $\beta_0$ represents TTC predicted ridership when $X_i = 0$. $X_i$ represents the years corresponding to the monthly weekday ridership. $\epsilon_i$ shows the difference between the linear regression line from each data value. $\beta_1$ is the average increase of ridership each month. Y represents the average weekday ridership in i year.

Load and read the data of TTC Average Weekday Ridership. Then, plot a scatterplot to help see if the data is appropriate for linear regression model as we require a linear relationship between the two variables chosen.

```
## Rows: 11
## Columns: 13
## $ Year <chr> "2007 Actual", "2008 Actual", "2009 Actual", "2010 Actual", "2...
## $ Jan  <dbl> 1400000, 1379000, 1448000, 1474000, 1540000, 1599000, 1606000,...
## $ Feb  <dbl> 1486000, 1476000, 1536000, 1538000, 1595000, 1643000, 1702000,...
## $ Mar  <dbl> 1485000, 1519000, 1487000, 1489000, 1568000, 1613000, 1672000,...
## $ Apr  <dbl> 1491000, 1446000, 1499000, 1500000, 1539000, 1614000, 1650000,...
## $ May  <dbl> 1474000, 1479000, 1517000, 1530000, 1602000, 1609000, 1661000,...
## $ Jun  <dbl> 1473000, 1459000, 1467000, 1492000, 1579000, 1627000, 1670000,...
## $ Jul  <dbl> 1391000, 1444000, 1435000, 1473000, 1528000, 1528000, 1564000,...
## $ Aug  <dbl> 1356000, 1434000, 1328000, 1412000, 1449000, 1515000, 1562000,...
## $ Sep  <dbl> 1549000, 1574000, 1586000, 1615000, 1689000, 1727000, 1743000,...
## $ Oct  <dbl> 1531000, 1520000, 1578000, 1622000, 1678000, 1712000, 1760000,...
## $ Nov  <dbl> 1538000, 1600000, 1568000, 1541000, 1639000, 1716000, 1751000,...
## $ Dec  <dbl> 1396000, 1484000, 1424000, 1457000, 1583000, 1655000, 1567000,...
```

Create a scatterplot with `ggplot2` (You will need to download the package `ggplot2` first). Since we have yet to clean the data, let's pick from one column and set x to Year and y to January.

```
ttc_avg <- avg_weekday_ridership %>%
  ggplot(aes(x = Year, y = Jan)) +
  geom_point() +
  theme(axis.text.x = element_text(angle = 45)) +
  ggtitle("Average Weekday Ridership on January")
```



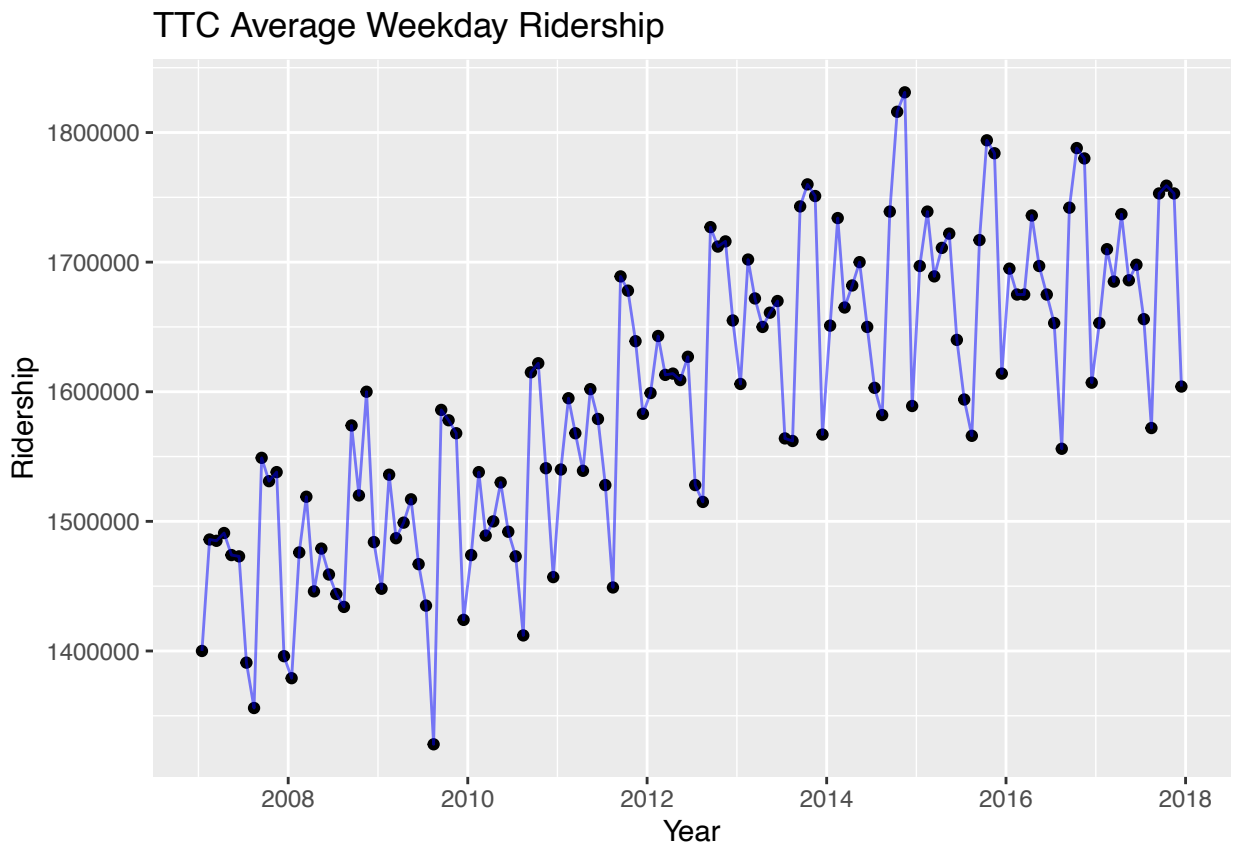

Linear regression is appropriate for this data because the two variables of the data being compared are quantitative. The second requirement for a linear regression model is variable X should be independent of variable Y. Such is the case for TTC ridership data where variable X representing years is independent of variable Y, representing the ridership numbers. As we could see from the scatterplot, there is a linear relationship between years and average ridershiip count of each month.

## Results

To begin with, when we look at the data, we see in the column "Year", there are excessive texts inside. We only need numbers to represent the years. Next, months are separated into columns but we want them all in a column so we can plot values in chronological order. We will put months into a new column and their corresponding ridership number into another one.

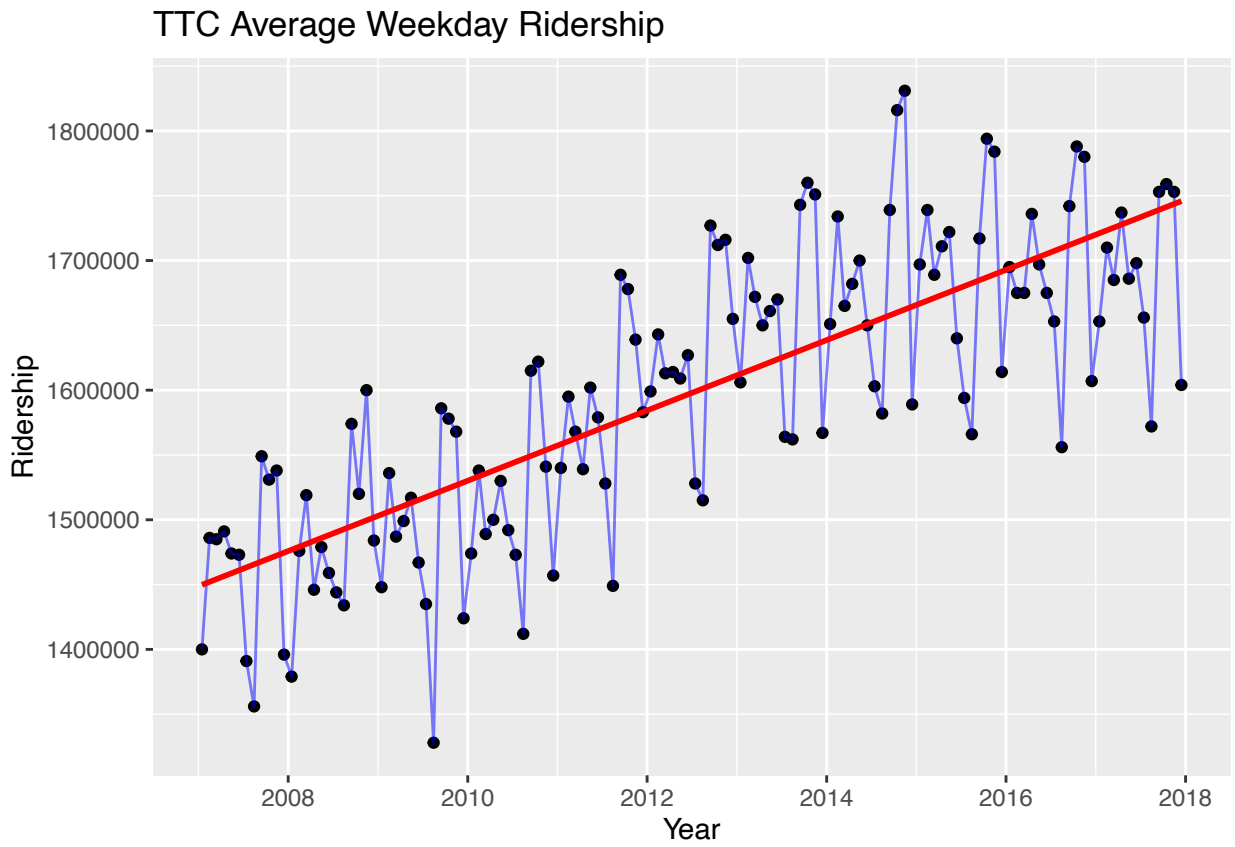
```
format_year <- "[2][0][0-9][0-9]"
avg_weekday_ridership <- avg_weekday_ridership %>%
  mutate(year = str_extract(Year, format_year)) %>%
#mutate() changes column based on given codes
#str_extract() selects first pattern matching our required pattern
  gather(month, ridership, Jan:Dec) %>%
#gather() reorganizes and turns the data into a new layout
  mutate(year = as.numeric(year))
#treat year as numbers so the linear model would not evaluate
#each year separately
```

```
a<- avg_weekday_ridership  %>%
  mutate(date = ymd(paste(year,month,"15"))) %>%
#ymd() converts strings into year/month/date format (Gimond)
#since there is no date given, we set all dates on 15th
#of each month (will not affect the graph as the gap between
#each data value is same throughout)
  ggplot(aes(x = date, y = ridership)) +
  geom_point() +
  labs(title = "TTC Average Weekday Ridership",
  x = "Year",
  y = "Ridership") +
  geom_line(colour = "Blue", alpha = 0.5)
```




From the linear regression scatterplot, we can see that there is a linear relationship between *Year* and *Ridership* from 2007 to 2017. Even though the pattern of such data appears to be cyclical, it shows a clear increase since 2007. The relationship between the two variables is positive because the general pattern of the values plotted is going upward from bottom left. When we create linear regression model, we could expect the slope of the data to be positive.

```
a + geom_smooth(method = 'lm',
                se = FALSE,
                colour = "Red",
                alpha = 0.4)
```




The best-fit linear regression is plotted onto our first graph. As we can see once again, the linear relationship between years and ridership for ttc is positive. Although there are certain months where ridership was lower but in general, the graph is increasing.

*Linear Regression Table*

```
tb <- lm(ridership~year, data = avg_weekday_ridership)
summary(tb)
```

```
##
## Call:
## lm(formula = ridership ~ year, data = avg_weekday_ridership)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -188798  -52046     703   45205  179206
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -52724680    3936786  -13.39   <2e-16 ***
## year            26999       1957   13.80   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 71090 on 130 degrees of freedom
## Multiple R-squared:  0.5943, Adjusted R-squared:  0.5911
## F-statistic: 190.4 on 1 and 130 DF,  p-value: < 2.2e-16
```

The value next to "year" and under column "estimate" is the slope of the linear regression model of TTC Average Weekday Ridership. Notice that the slope is large in the regression model, this is because first, the ridership numbers are large, and second, we treated the time, years, as numbers. The base of our dataset for both x and y axis was large therefore, the table value of our results would be large as well.

| | |
|---|---|
| $\hat{\beta}_0$ | -52724680 |
| $\hat{\beta}_1$ | 26999 |

All analysis for this report was programmed using `R version 4.0.2`.

## Bibliography


1. Grolemund, G. (2014, July 16) *Introduction to R Markdown.* RStudio. https://rmarkdown.rstudio.com/articles_intro.html. (Last Accessed: January 15, 2021)
2. Dekking, F. M., et al. (2005) *A Modern Introduction to Probability and Statistics: Understanding why and how.* Springer Science & Business Media.
3. Allaire, J.J., et. el. *References: Introduction to R Markdown.* RStudio. https://rmarkdown.rstudio.com/docs/. (Last Accessed: January 15, 2021)
4. Wickham et al., (2019). *Welcome to the tidyverse. Journal of Open Source Software*, 4(43), 1686, https://doi.org/10.21105/joss.01686
5. Hadley Wickham (2019). *stringr: Simple, Consistent Wrappers for Common String Operations.* http://stringr.tidyverse.org, https://github.com/tidyverse/stringr.
6. Hadley Wickham (2020). *tidyr: Tidy Messy Data.* https://tidyr.tidyverse.org, https://github.com/tidyverse/tidyr.
7. Garrett Grolemund, Hadley Wickham (2011). *Dates and Times Made Easy with lubridate.* Journal of Statistical Software, 40(3), 1-25. https://www.jstatsoft.org/v40/i03/.
8. Hadley Wickham, Romain François, Lionel Henry and Kirill Müller (2021). *dplyr: A Grammar of Data Manipulation.* https://dplyr.tidyverse.org, https://github.com/tidyverse/dplyr.
9. Hadley Wickham and Jim Hester (2020). *readr: Read Rectangular Text Data.* https://readr.tidyverse.org, https://github.com/tidyverse/readr.
10. Kirill Müller and Hadley Wickham (2021). *tibble: Simple Data Frames.* https://tibble.tidyverse.org/, https://github.com/tidyverse/tibble.
11. H. Wickham. *ggplot2: Elegant Graphics for Data Analysis.* Springer-Verlag New York, 2016.https://ggplot2.tidyverse.org
12. Mine Çetinkaya-Rundel, David Diez, Andrew Bray, Albert Kim, Ben Baumer, Chester Ismay and Christopher Barr (2020). *openintro: Data Sets and Supplemental Functions from 'OpenIntro' Textbooks and Labs.* R package version 2.0.0. https://github.com/OpenIntroStat/openintro
13. Makowski, Dominique (2018, August 31). *How to Cite Packages.* [https://www.r-bloggers.com/2018/08/how-to-cite-packages/] (https://www.r-bloggers.com/2018/08/how-to-cite-packages/). (Last Accessed: February 11, 2021)
14. (2020, June 08). *TTC Average Weekday Ridership.* City of Toronto. [https://www.toronto.ca/city-government/data-research-maps/toronto-progress-portal/] (https://www.toronto.ca/city-government/data-research-maps/toronto-progress-portal/). (Last Accessed: February 8, 2021)
15. Et.al. *Reshaping Your Data with tidyr · UC Business Analytics R Programming Guide* UC R Programming. [https://uc-r.github.io/tidyr] (https://uc-r.github.io/tidyr). (Last Accessed: February 11, 2021)
16. Gimond, Manny. *Working With Dates.* [https://mgimond.github.io/ES218/Week02c.html#From_complete_date_strings] (https://mgimond.github.io/ES218/Week02c.html#From_complete_date_strings). (Last Accessed: February 11, 2021)
17. Alison Gibbs, Alex Stringer (2021, January 20). *Probability, Statistics, and Data Analysis: Chapter 4 and 5.* [https://awstringer1.github.io/sta238-book/section-evaluating-estimators-efficiency-and-mean-squared-error.html#section-mean-squared-error] (https://awstringer1.github.io/sta238-book/section-evaluating-estimators-efficiency-and-mean-squared-error.html#section-mean-squared-error)
18. Thomas Lin Pedersen (2020). *patchwork: The Composer of Plots.* [https://patchwork.data-imaginist.com, https://github.com/thomasp85/patchwork.] (https://patchwork.data-imaginist.com, https://github.com/thomasp85/patchwork)