

Sentiment Analysis of Antidepressant Reviews: Unveiling Gender Dynamics in SSRIs and SNRIs

Gavriel Steinmetz-Silber, Noori Selina, Zainab Oketokoun, and Haig Bedros

Part A: National Center for Health Statistics

Data Source: <https://www.cdc.gov/nchs/hus/data-finder.htm?year=2019&table=Table%20009>

#Introduction: Since COVID, various news outlets have been reporting on what they are calling a mental health crisis in the United States in individuals of all age groups. The CDC reports that 1 in 25 US Adults lives with serious mental health conditions e.g. major depression. The NIMH reports that mental disorders cost the U.S. 193 billion dollars alone annually in lost earnings.

We wanted to investigate if there were patterns relating to mental health over years to determine if there is a mental health crisis occurring and if so, is it an one that is on the rise or are we, the American public, just noticing an existing condition. We looked to the CDC to get annual data on mental health in the U.S. After looking at variety of variables, we decided that death by suicide rates were a traceable phenotype. The initial struggle was having to load the file as an excel and then having to rename columns, changing class types repeatedly and eliminating “N/A” before we could even transform the data. We fixed this issue by locating an API key. We loaded the data using an API key by first defining the URL, and reading it in as a csv file. The next challenge was splitting variables. For example we had a column that had combined Sex (alone) and Sex + Ethnicity together in one column. We use dplyr, tidyverse and stringr, to remove ethnicity from genders and combine Race and Ethnicity related information to one column. We simplified all female related information to the variable = “female”, did the same for the males and combined it all into one column. We removed and rewrote column names. We aggregated variables like e.g. year and sex to create a plot showing the trend of mental health over the years for the different sexes, ethnicities and age_range groups.

We lead our statistical analysis by having a null hypothesis stating there would be no differences seen for death by suicide rates between sex, between ethnic groups and between age_range groups. First we use dplyr to determine summary stats, and followed this up by using ggplot to compare center(mean) and spread of males rates to female rates, center and spread of the rates between the different ethnic groups and center and spread of the rates between the different age range groups. Then we used the Levene test(under the package “car”) to determine if the variance was equal between the groups. We use LearnBayes package to determine correlation value between death rates and Sex and lastly, we used a one.way test for an ANOVA test to identify the statistical significance of the difference seen in the age groups.

1. Data Loading

Death rates for suicide, by sex, race, Hispanic origin, and age: United States, selected years 1950-2018

```
# Define the URL of the API
cdc_api <- "https://data.cdc.gov/resource/9j2v-jamp.csv"

# Read the CSV data from the URL
```

```
suicide_rates <- read.csv(cdc_api)
```

```
# Glimpse of our data
head(suicide_rates)
```

```
##               indicator                                     unit
## 1 Death rates for suicide Deaths per 100,000 resident population, age-adjusted
## 2 Death rates for suicide Deaths per 100,000 resident population, age-adjusted
## 3 Death rates for suicide Deaths per 100,000 resident population, age-adjusted
## 4 Death rates for suicide Deaths per 100,000 resident population, age-adjusted
## 5 Death rates for suicide Deaths per 100,000 resident population, age-adjusted
## 6 Death rates for suicide Deaths per 100,000 resident population, age-adjusted
##   unit_num stub_name stub_name_num  stub_label stub_label_num year year_num
## 1      1      Total              0 All persons              0 1950         1
## 2      1      Total              0 All persons              0 1960         2
## 3      1      Total              0 All persons              0 1970         3
## 4      1      Total              0 All persons              0 1980         4
## 5      1      Total              0 All persons              0 1981         5
## 6      1      Total              0 All persons              0 1982         6
##   age age_num estimate flag
## 1 All ages      0    13.2
## 2 All ages      0    12.5
## 3 All ages      0    13.1
## 4 All ages      0    12.2
## 5 All ages      0    12.3
## 6 All ages      0    12.5
```

2. Data cleaning and tidying

```
library(stringr)
library(dplyr)
library(tidyverse)
```

```
# Extract the different characteristic variables from the column stub_label
```

```
suicide_rates <- suicide_rates %>%
```

```
  mutate(
```

```
    # Extract Sex
```

```
    Sex = case_when(
```

```
      str_detect(stub_label, "^Male") ~ "Male",
```

```
      str_detect(stub_label, "^Female") ~ "Female",
```

```
      TRUE ~ NA_character_ # Assign NA to entries that do not start with Male or Female
```

```
    ),
```

```
    # Combine Race and Ethnicity into one column, ignore entries that are just age groups
```

```
    RaceEthnicity = case_when(
```

```
      str_detect(stub_label, "White$") ~ "White",
```

```
      str_detect(stub_label, "Black or African American$") ~ "Black or African American",
```

```
      str_detect(stub_label, "American Indian or Alaska Native$") ~ "American Indian or Alaska Native",
```

```
      str_detect(stub_label, "Asian or Pacific Islander$") ~ "Asian or Pacific Islander",
```

```
      str_detect(stub_label, "Asian$") ~ "Asian",
```

```
      str_detect(stub_label, "Native Hawaiian or Other Pacific Islander$") ~ "Native Hawaiian or Other Pacific Islander"
```

```

    str_detect(stub_label, "Not Hispanic or Latino") ~ "Not Hispanic or Latino",
    str_detect(stub_label, "Hispanic or Latino") ~ "Hispanic or Latino",
    str_detect(stub_label, "years$") ~ NA_character_, # Assign NA to entries that are just age groups
    TRUE ~ "Other/Unknown" # Use this to catch all other entries that don't match previous patterns
  )
)

# Now we can remove the original `stub_label` column if it's no longer needed
suicide_rates <- suicide_rates %>%
  select(-stub_label)

# Replace NA values in the 'Sex' column with "All"
suicide_rates <- suicide_rates %>%
  mutate(Sex = replace_na(Sex, "All"))

# Replace NA values in the 'RaceEthnicity' column with "Other/Unknown"
suicide_rates <- suicide_rates %>%
  mutate(RaceEthnicity = replace_na(RaceEthnicity, "Other/Unknown"))

# Renaming 'estimate' to 'death_rate_est' for readability and drop specified columns
suicide_rates <- suicide_rates %>%
  select(-unit_num, -stub_name_num, -stub_name, -stub_label_num, -year_num, -age_num, -flag) %>%
  rename(death_rate_est = estimate) %>%
  select(year, everything())

# We'll use the 'year' as a continuous x-axis for a line plot
# Making sure the 'year' column is numeric
suicide_rates$year <- as.numeric(as.character(suicide_rates$year))

# Aggregate data by year and calculate the average death rate
suicide_rates_yearly <- suicide_rates %>%
  group_by(year, Sex) %>%
  summarise(death_rate_est = mean(death_rate_est, na.rm = TRUE))

# Plot 1: Trends of suicide rates throughout the years where sex = All
plot1 <- ggplot(data = suicide_rates_yearly, aes(x = year, y = death_rate_est, color = Sex)) +
  geom_line() +
  theme_minimal() +
  labs(
    title = "Trends of Suicide Rates Throughout the Years (Combined Sexes)",
    x = "Year",
    y = "Average Death Rate (per 100,000 individuals)",
    color = "Sex"
  ) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

# Plot 2: Trends of suicide rates throughout the years of the RaceEthnicity
plot2 <- suicide_rates %>%
  group_by(year, RaceEthnicity) %>%
  summarise(death_rate_est = mean(death_rate_est, na.rm = TRUE)) %>%
  ggplot(aes(x = year, y = death_rate_est, color = RaceEthnicity)) +
  geom_line() +

```

```

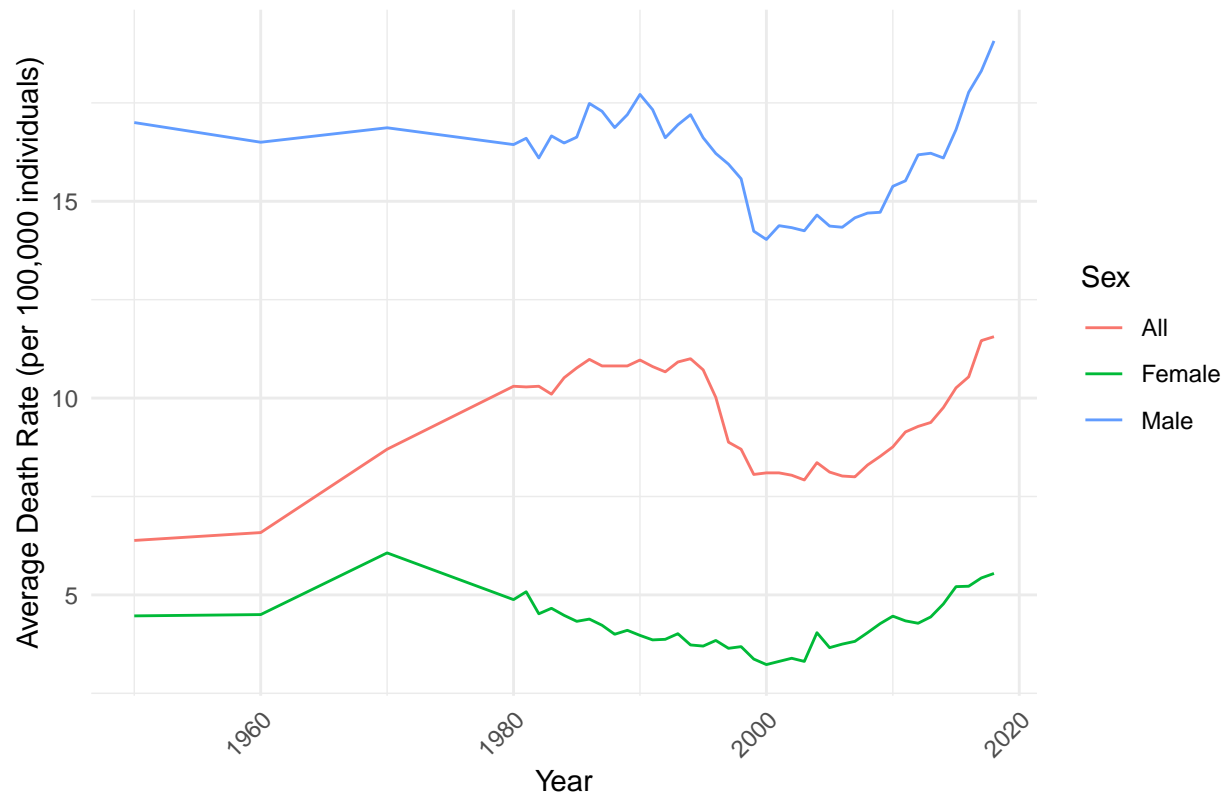
theme_minimal() +
labs(
  title = "Trends of Suicide Rates Throughout the Years by Race/Ethnicity",
  x = "Year",
  y = "Average Death Rate (per 100,000 individuals)",
  color = "Race/Ethnicity"
) +
theme(axis.text.x = element_text(angle = 45, hjust = 1))

# Plot 3: Trends of suicide rates throughout the years of sex = Female or Male and age != All Ages
plot3 <- suicide_rates %>%
  filter(Sex != "All", age != "All Ages") %>%
  group_by(year, Sex, age) %>%
  summarise(death_rate_est = mean(death_rate_est, na.rm = TRUE)) %>%
  ggplot(aes(x = year, y = death_rate_est, color = Sex, group = interaction(Sex, age))) +
  geom_line() +
  facet_wrap(~age, scales = "free_y") +
  theme_minimal() +
  labs(
    title = "Trends of Suicide Rates Throughout the Years by Sex and Age Group",
    x = "Year",
    y = "Average Death Rate (per 100,000 individuals)",
    color = "Sex"
  ) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

# Printing the plots
print(plot1)

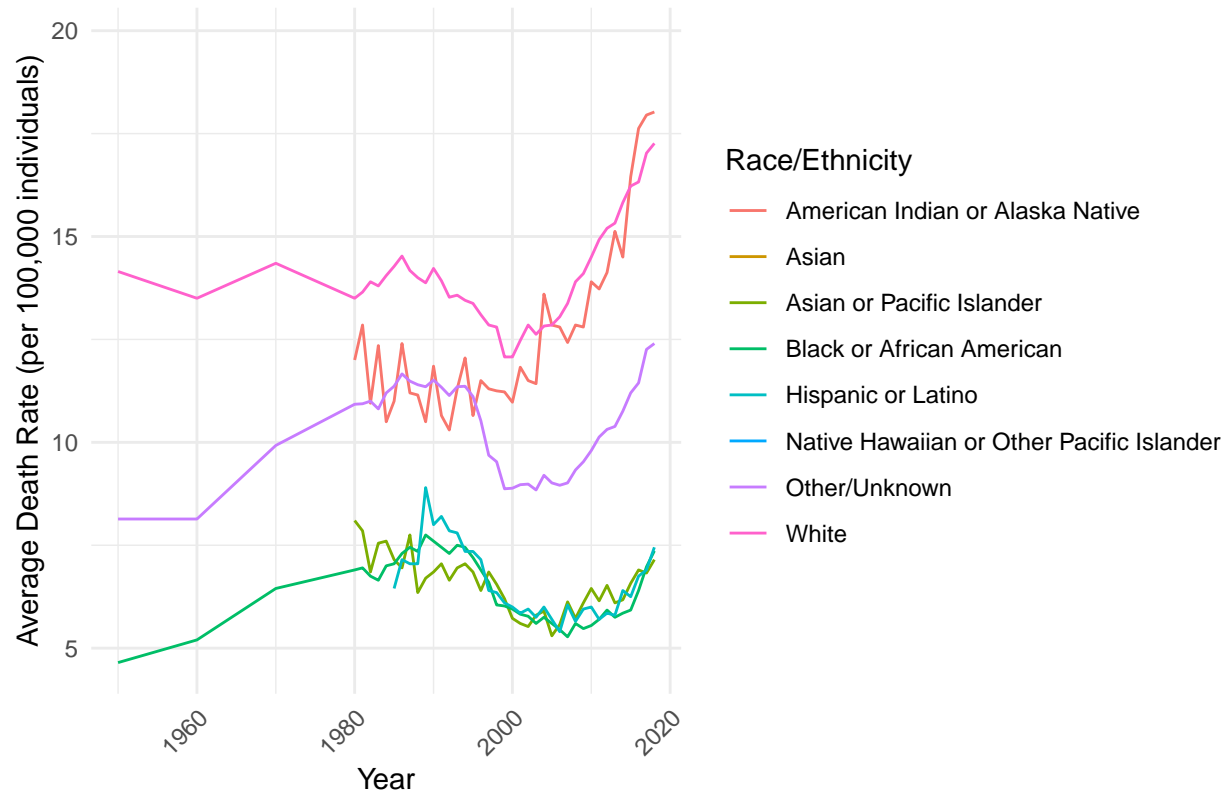
```

Trends of Suicide Rates Throughout the Years (Combined Sexes)



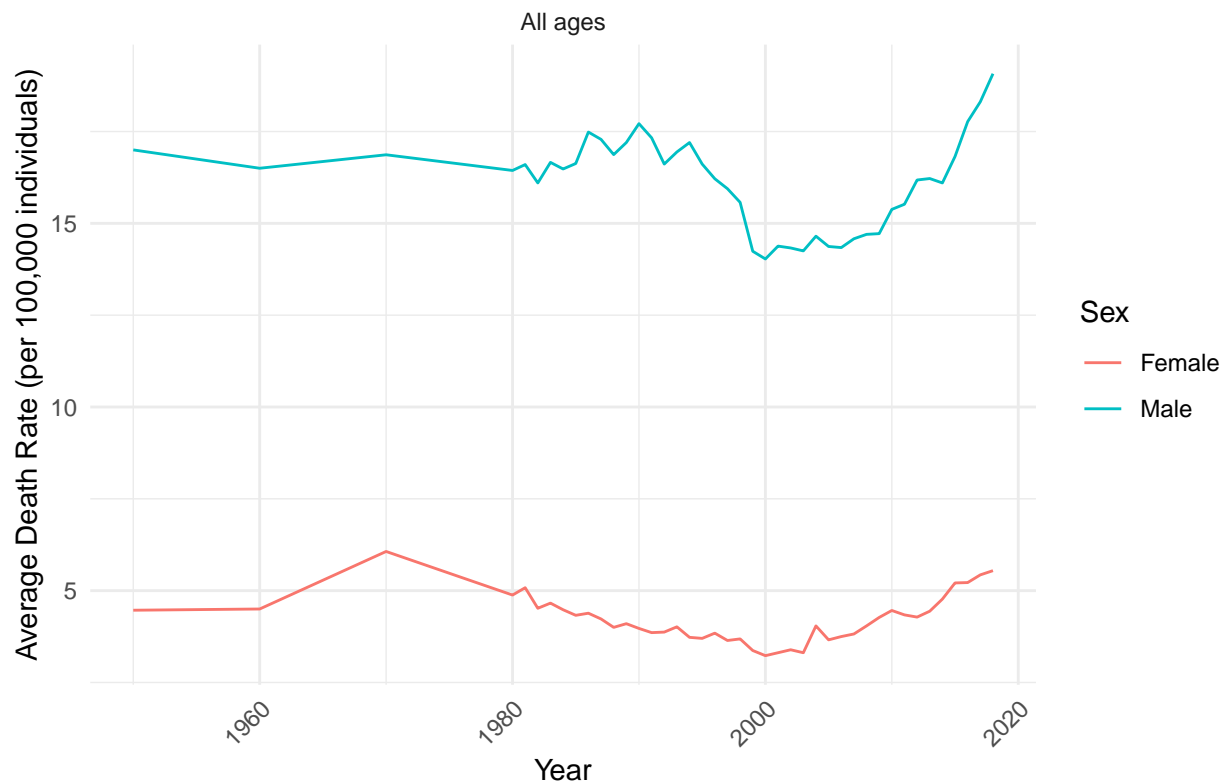
```
print(plot2)
```

Trends of Suicide Rates Throughout the Years by Race/Ethnicity



```
print(plot3)
```

Trends of Suicide Rates Throughout the Years by Sex and Age Group



Notes about our dataset:

- In our dataset, the age column denotes groups for which the death rates are age-adjusted, expressed per 100,000 individuals, allowing for consistent comparisons by accounting for age distribution differences within the population.
- The 'All ages' value in the age column category represents an age-adjusted rate, providing a summary measure of the suicide death rate across the entire population, standardized to account for variations in age distribution.

3. Data transformation

```
# Transforming our data to include a new column for decade
suicide_rates_transformed <- suicide_rates %>%
  mutate(Decade = floor(year / 10) * 10) %>% # Create a new column for Decade
  group_by(Decade, RaceEthnicity) %>% # Group by Decade and RaceEthnicity
  summarize(MeanDeathRate = mean(death_rate_est, na.rm = TRUE)) # Calculate mean death rate

# View the transformed data
suicide_rates_transformed
```

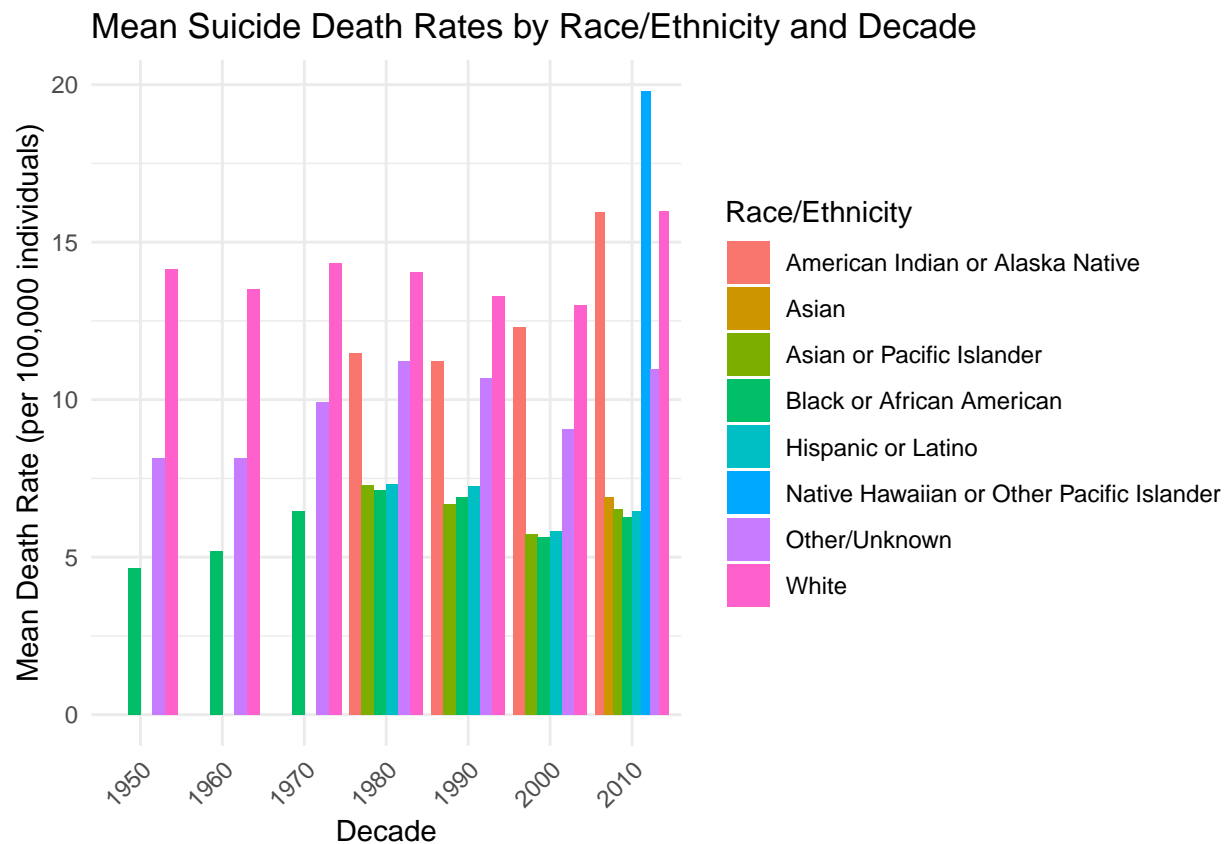
```
## # A tibble: 44 x 3
## # Groups:   Decade [7]
##   Decade RaceEthnicity MeanDeathRate
##   <dbl> <chr>           <dbl>
```

```
## 1 1950 American Indian or Alaska Native      NaN
## 2 1950 Asian or Pacific Islander             NaN
## 3 1950 Black or African American             4.65
## 4 1950 Hispanic or Latino                   NaN
## 5 1950 Other/Unknown                        8.14
## 6 1950 White                               14.2
## 7 1960 American Indian or Alaska Native      NaN
## 8 1960 Asian or Pacific Islander             NaN
## 9 1960 Black or African American             5.2
## 10 1960 Hispanic or Latino                   NaN
## # i 34 more rows
```

```
# Plotting our transformed data
```

```
plot4 <- ggplot(suicide_rates_transformed, aes(x = as.factor(Decade), y = MeanDeathRate, fill = RaceEthnicity)) +
  geom_bar(stat = "identity", position = position_dodge()) +
  theme_minimal() +
  labs(
    title = "Mean Suicide Death Rates by Race/Ethnicity and Decade",
    x = "Decade",
    y = "Mean Death Rate (per 100,000 individuals)",
    fill = "Race/Ethnicity"
  ) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) # Rotate x labels for readability

print(plot4)
```




```
head(suicide_rates_yearly)
```

Suicide Yearly Rates

```
## # A tibble: 6 x 3
## # Groups:   year [2]
##   year Sex    death_rate_est
##   <dbl> <chr>      <dbl>
## 1  1950 All      6.38
## 2  1950 Female  4.47
## 3  1950 Male    17
## 4  1960 All      6.58
## 5  1960 Female  4.5
## 6  1960 Male    16.5
```

```
head(suicide_rates)
```

Suicide Rates

```
##   year                indicator
## 1 1950 Death rates for suicide
## 2 1960 Death rates for suicide
## 3 1970 Death rates for suicide
## 4 1980 Death rates for suicide
## 5 1981 Death rates for suicide
## 6 1982 Death rates for suicide
##                                     unit    age death_rate_est
## 1 Deaths per 100,000 resident population, age-adjusted All ages      13.2
## 2 Deaths per 100,000 resident population, age-adjusted All ages      12.5
## 3 Deaths per 100,000 resident population, age-adjusted All ages      13.1
## 4 Deaths per 100,000 resident population, age-adjusted All ages      12.2
## 5 Deaths per 100,000 resident population, age-adjusted All ages      12.3
## 6 Deaths per 100,000 resident population, age-adjusted All ages      12.5
##   Sex RaceEthnicity
## 1 All Other/Unknown
## 2 All Other/Unknown
## 3 All Other/Unknown
## 4 All Other/Unknown
## 5 All Other/Unknown
## 6 All Other/Unknown
```

```
head(suicide_rates_transformed)
```

```
## # A tibble: 6 x 3
## # Groups:   Decade [1]
##   Decade RaceEthnicity      MeanDeathRate
##   <dbl> <chr>              <dbl>
## 1  1950 American Indian or Alaska Native      NaN
```

```
## 2    1950 Asian or Pacific Islander      NaN
## 3    1950 Black or African American      4.65
## 4    1950 Hispanic or Latino            NaN
## 5    1950 Other/Unknown                 8.14
## 6    1950 White                        14.2
```

4. Statistical Analysis

Question : IS there a difference in between the death rate of males and females In plot __, we see a big discrepancy between the death rate of male and females though the years. Is that significant? The hypothesis we are moving on is that there is no significant difference between the death rate of males to that of females

Statistical Analysis

Descriptive Stats : We grouped by Sex and apply tapply() function to calculate the descriptive stats of the death rates

```
tapply(suicide_rates_yearly$death_rate_est,suicide_rates_yearly$Sex,summary,na.rm=TRUE)
```

```
## $All
##      Min. 1st Qu.  Median      Mean 3rd Qu.     Max.
##  6.383   8.315   9.888   9.517  10.754  11.560
##
## $Female
##      Min. 1st Qu.  Median      Mean 3rd Qu.     Max.
##  3.230   3.768   4.164   4.236   4.495   6.067
##
## $Male
##      Min. 1st Qu.  Median      Mean 3rd Qu.     Max.
##  14.03  14.88   16.46   16.15  16.93  19.07
```

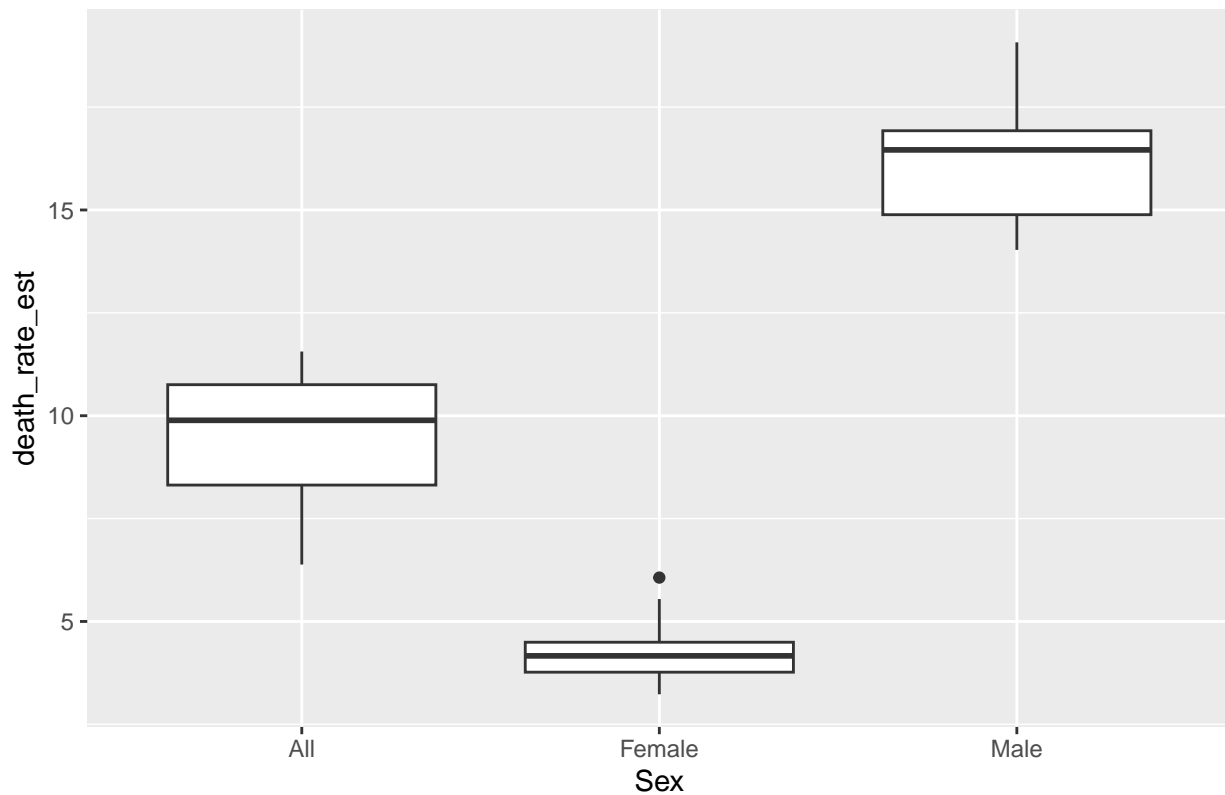
```
tapply(suicide_rates_yearly$death_rate_est,suicide_rates_yearly$Sex,sd,na.rm=TRUE)
```

```
##           All      Female      Male
## 1.3488072 0.6483008 1.2504841
```

OBSERVATION: The center and spread for death rates males is a lot higher than females alone and with the combined sexes. Infact the center or average death rate for males is around 4-fold of that of the females

```
ggplot(data = suicide_rates_yearly, mapping = aes(x = Sex,y = death_rate_est))+
  geom_boxplot()+
  ggtitle("Checking the means and std of Males to Females to All")
```

Checking the means and std of Males to Females to All



OBSERVATION: We see that the avg death rate of males is higher than the avg death rate of the sexes together and the avg death rate of females. We see the avg death rate of females is lower than avg for the group (rate where the sexes are together).

Is this significant?

```
library(car)
# Levene's test of equal variances.
# Low p-value means the variances are not equal.
leveneTest(suicide_rates_yearly$death_rate_est, suicide_rates_yearly$Sex)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##      Df F value    Pr(>F)
## group  2 11.241 3.283e-05 ***
##      123
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

5. Results:

The p-value is less than 0.001, so we reject the null hypothesis that there is no difference between the death rates of male to death rates of females.

```

library(LearnBayes)

dfA = na.omit(suicide_rates_yearly)

# Code 'Female' as 0 and 'Male' as 1 in the 'Sex' column
# This assumes that dfA$Sex is a factor or character vector
dfA$Sex <- as.numeric(dfA$Sex == 'Male')

# Remove rows where 'Sex' is "All"
dfA <- dfA[dfA$Sex != "All", , drop = FALSE]

# Ensure all columns are numeric for correlation; if not, convert them
dfA_numeric <- data.frame(lapply(dfA, function(x) if(is.numeric(x)) x else as.numeric(as.character(x))))

# Remove any NA values that could cause cor() to fail
dfA_numeric <- na.omit(dfA_numeric)

# Calculate the correlation matrix for numeric data frame
cor_matrix <- cor(dfA_numeric)

# Print the correlation matrix
print(cor_matrix)

```

```

##              year              Sex death_rate_est
## year          1.000000e+00 -7.818384e-21  -0.009065921
## Sex           -7.818384e-21  1.000000e+00   0.874333761
## death_rate_est -9.065921e-03  8.743338e-01  1.000000000

```

```

#tapply(suicide_rates$death_rate_est,suicide_rates$ages,summary,na.rm=TRUE)
tapply(suicide_rates$death_rate_est,suicide_rates$age,summary,na.rm=TRUE)

```

Looking at Ages:

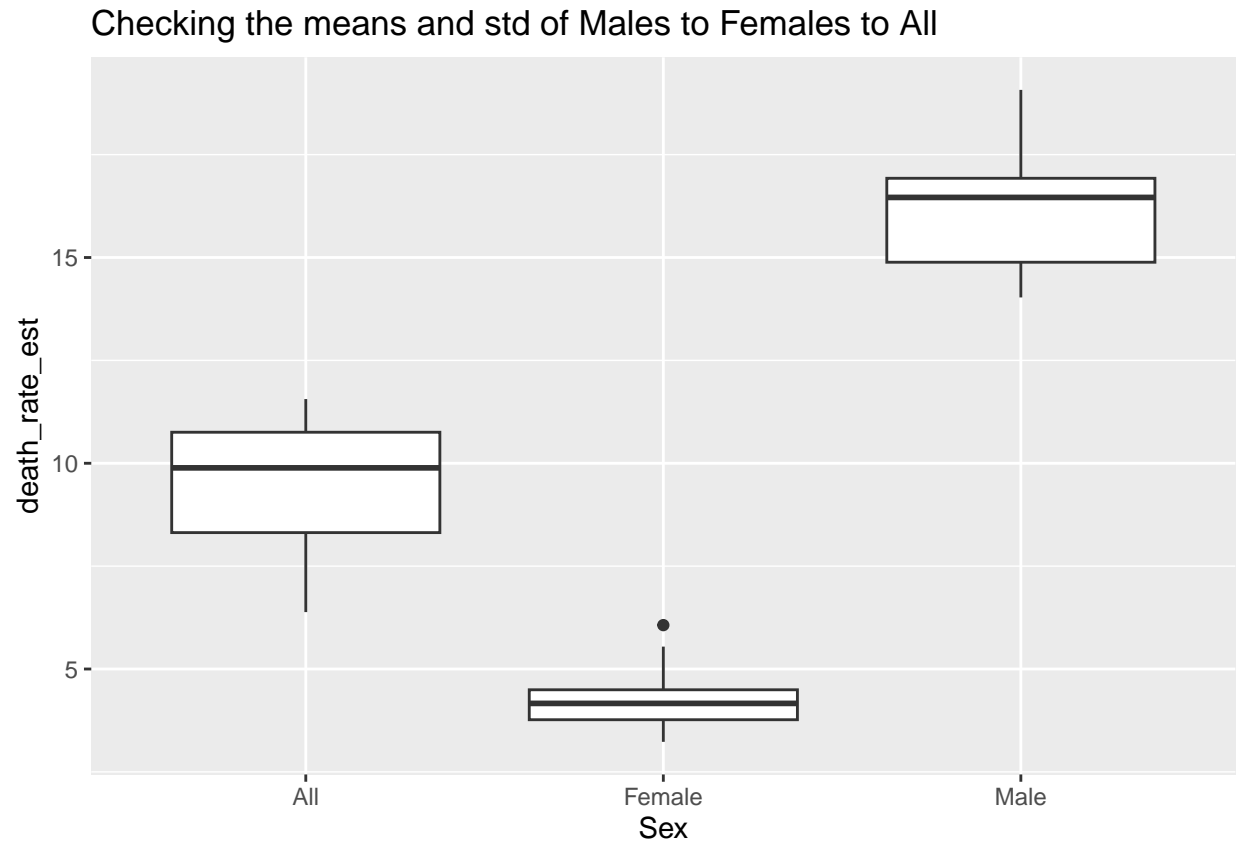
```

## $'10-14 years'
##   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.300  1.125   1.400   1.410  1.600   2.900
##
## $'15-19 years'
##   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   2.700  7.600   8.650   8.738 10.175  11.800
##
## $'15-24 years'
##   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   4.50   10.03   11.70   11.32  12.90   14.50
##
## $'20-24 years'
##   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   6.20   14.60   15.05   14.22  15.53   16.10
##
## $'All ages'

```

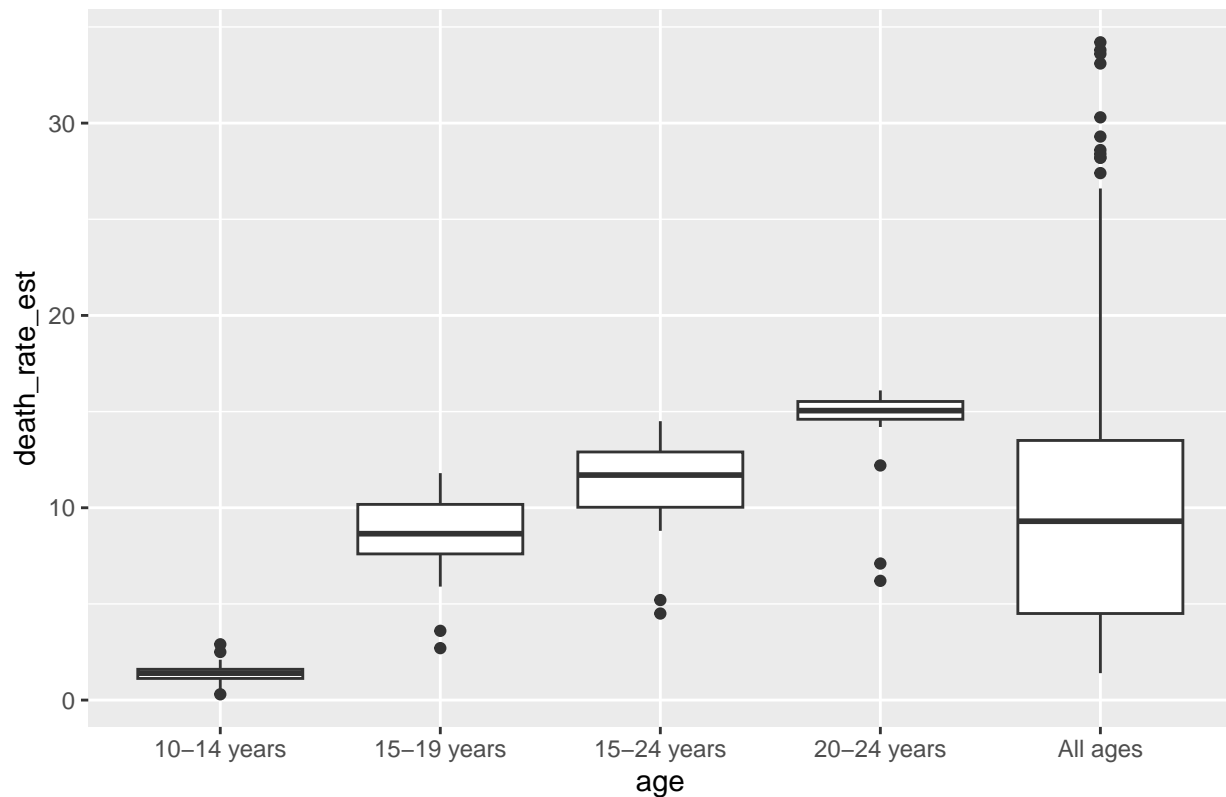
##	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	NA's
##	1.40	4.50	9.30	10.37	13.50	34.20	89

```
ggplot(data = suicide_rates_yearly, mapping = aes(x = Sex, y = death_rate_est))+
  geom_boxplot()+
  ggtitle("Checking the means and std of Males to Females to All")
```



```
ggplot(data = suicide_rates, mapping = aes(x = age, y = death_rate_est))+
  geom_boxplot()+
  ggtitle("Center and Spread between the Age_ranges")
```

Center and Spread between the Age_ranges



#OBSERVATION: The center for the death rates of suicides for ages in the data increase as the ages ranges increase. There is trend that as the age increase so does the average death rates of suicides . Each respective box plot is tight, thus not a lot of spread in each age range but they do have outliers of very low values.

```
library(car)
# Levene's test of equal variances.
leveneTest(suicide_rates$death_rate_est,suicide_rates$age)
```

Checking variances between age groups:

```
## Levene's Test for Homogeneity of Variance (center = median)
##      Df F value    Pr(>F)
## group  4  38.67 < 2.2e-16 ***
##      906
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Low p-value means the variances are not equal.

We reject the null hypothesis , we reject the idea that the death rate has equal variances between all the age_ranges

```
#install.packages("LearnBayes")
```

```
library(LearnBayes)
```

```
fit = lm(death_rate_est ~ age, data = suicide_rates, x=TRUE, y=TRUE)
```

```
posterior_sims = blinreg(fit$y, fit$x, 5000)
```

```
fit$coefficients
```

```
##      (Intercept) age15-19 years age15-24 years age20-24 years    ageAll ages
##      1.409524      7.328571      9.914286      12.810476      8.958842
```

```
oneway.test(death_rate_est ~ age, data = suicide_rates, var.equal = FALSE)
```

```
##
```

```
## One-way analysis of means (not assuming equal variances)
```

```
##
```

```
## data: death_rate_est and age
```

```
## F = 647.47, num df = 4.000, denom df = 82.156, p-value < 2.2e-16
```

###RESULTS We reject the null hypothesis, so we reject the idea that all the averages of the rates (of the suicide rates) are equal between the age_ranges

```
tapply(suicide_rates_transformed$MeanDeathRate, suicide_rates_transformed$RaceEthnicity, summary, na.rm=TRUE)
```

```
## $'American Indian or Alaska Native'
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.      NA's
##      11.21   11.42   11.90   12.74   13.21   15.95         3
```

```
##
```

```
## $Asian
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      6.9      6.9      6.9      6.9      6.9      6.9
```

```
##
```

```
## $'Asian or Pacific Islander'
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.      NA's
##      5.737   6.339   6.615   6.563   6.839   7.285         3
```

```
##
```

```
## $'Black or African American'
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      4.650   5.415   6.280   6.035   6.684   7.115
```

```
##
```

```
## $'Hispanic or Latino'
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.      NA's
##      5.830   6.299   6.855   6.715   7.271   7.320         3
```

```
##
```

```
## $'Native Hawaiian or Other Pacific Islander'
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      19.8   19.8   19.8   19.8   19.8   19.8
```

```
##
```

```
## $'Other/Unknown'
```

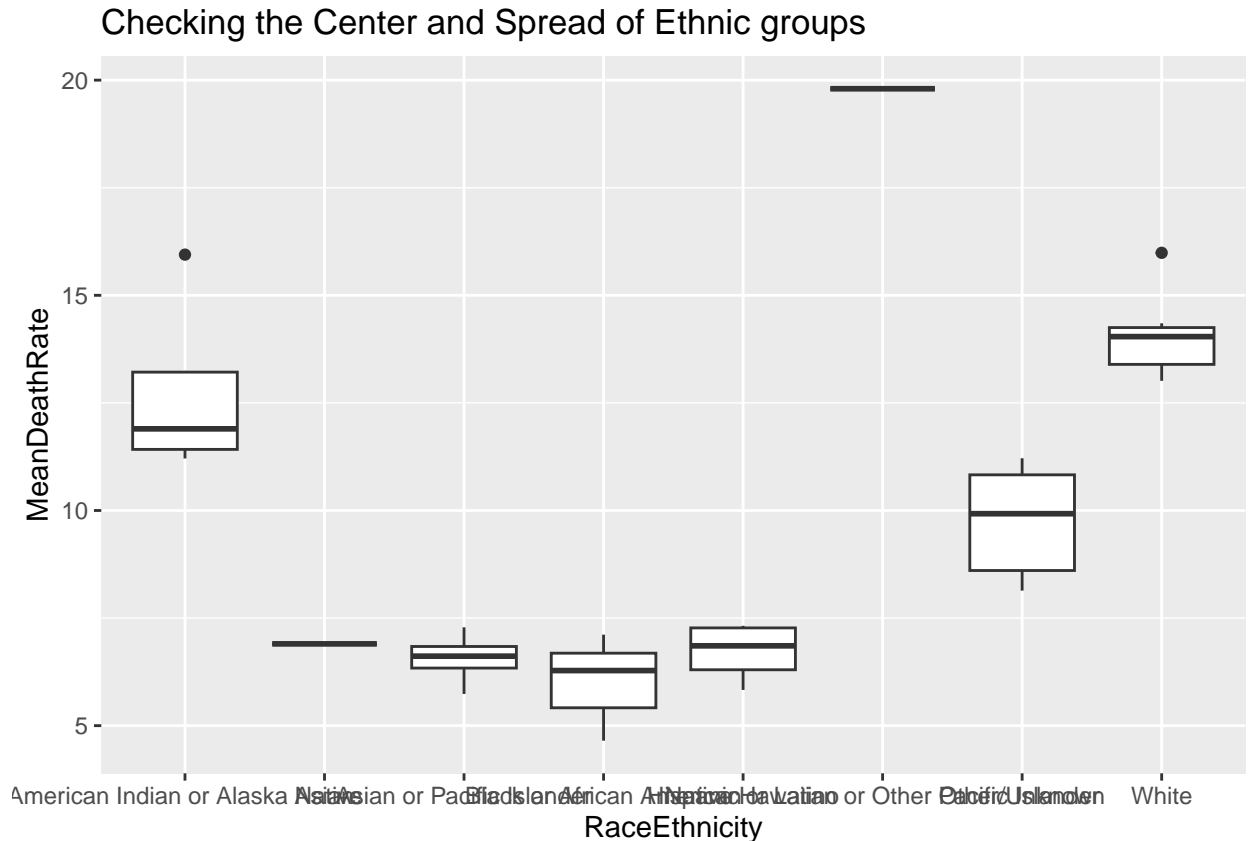
```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      8.137   8.605   9.925   9.735  10.829  11.214
```

```
##
```

```
## $White
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      13.01  13.39   14.04   14.05  14.25   15.99
```

```
library(ggplot2)
ggplot(data = suicide_rates_transformed, mapping = aes(x = RaceEthnicity ,y = MeanDeathRate))+
  geom_boxplot()+
  ggtitle("Checking the Center and Spread of Ethnic groups")
```



The two highest death rates are observed in American Indians and Whites. The lowest averages of death rates is observed in African-Americans.

CONCLUSION

We observed a difference in rates between different races/ethnic groups, however we believe that since there is a large amount of missing data , the analysis needed heavy transformation to not calculate a heavily skew analysis. We do recognize that it was not until 1995 that the NIH mandated that clinical and field work -NIH funded- projects must include participants from POCs and marginalized groups. This may account for the low values in the POC groups in comparison to the Whites.

Lastly, the increase of the death rates as the years increase, suggest that the country is uncovering an existing mental health problem or that there is one on the rise.

We decided to further explore Sex differences and engage with data that reflect real-life patient experiences and investigate if there is a pattern indicating a difference in experience between male and female patients.

Part B: Sentiment analysis

1. Web Scraping

To gather the reviews, we will scrape from WebMD (e.g. <https://reviews.webmd.com/drugs/drugreview-63990-lexapro-oral>). All the pages have a similar format. There appear to be boxes, with some basic user info on top and then the content of the review below. There are also some star ratings, but we will rely exclusively on the text for this project.

Inspecting the webpages further, we see that each review is under the headers of review sections are marked as “card-header”). So we can use this as part of a loop. Further, within each review, we can find the user informatoin by looking for the class .details. We can find the date by looking for the class .date. And we can find the text of the review by looking for the immediately following class .description. that contains a paragraph of the class “description-text.”

Now, one struggle is that there are only 20 reviews on a page but each drug has many, many reviews. For Lexapro, I see there are 215 pages (so presumably around 215 x 19 to 20 total reviews). However, for each drug there’s also a consistent format. Taking Lexapro again, the format is: “[https://reviews.webmd.com/drugs/drugreview-63990-lexapro-oral?conditionid=&sortval=1&page=%5Bpage number%5D&next_page=true](https://reviews.webmd.com/drugs/drugreview-63990-lexapro-oral?conditionid=&sortval=1&page=%5Bpage%20number%5D&next_page=true)”

In fact, for all the drugs, the latter part of the website URL is the same. So, we will define a function and have two arguments: the start of the URL and the number of pages:

```
library(tidyverse)
library(rvest)

scrape_reviews = function(url_beginning, num_pages) {
  reviews_data = list()
  for (page in 1:num_pages) {
    url = paste0(url_beginning, page, "&next_page=true") #this will make up the full URL
    webpage = read_html(url)

    review_headers = html_nodes(webpage, ".card-header")
    reviews_data_page = list()

    for (header in review_headers) {
      user_info = html_text(html_nodes(header, ".details"))
      date = html_text(html_nodes(header, ".date"))
      review_text = html_text(html_nodes(header, xpath = "./following-sibling::div[@class='description']"))
      page_data = list(user_info = user_info, date = date, review_text = review_text)
      reviews_data_page = append(reviews_data_page, list(page_data)) #adding this review to the page reviews
    }

    reviews_data = append(reviews_data, reviews_data_page) # Adding this page's reviews to the list of reviews
  }

  reviews_df = data.frame(do.call(rbind, reviews_data))
  colnames(reviews_df) = c("reviewer_info", "Date", "Review")
  return(reviews_df)
}
```

Now, we *could* just pass the different drugs into the function (e.g. `cymbalta_reviews_df = scrape_reviews("https://reviews.webmd.com/drugs/drugreview-91491-cymbalta-oral?conditionid=&sortval=1&page=", 234)`).

However, this will all take a while so we can use parallel computing to hurry matters up. We start by defining the tasks for the workers. We define tasks a list of all the main tasks (i.e. one for cymbalta, one for effexor, and so on). We also have a list for each drug-that contains the two relevant paramaters.

```
library(parallel)
library(doParallel)

# the tasks are a list of
tasks = list(
  list(url_beginning = "https://reviews.webmd.com/drugs/drugreview-91491-cymbalta-oral?conditionid=&sort=",
        url_end = "&page="),
  list(url_beginning = "https://reviews.webmd.com/drugs/drugreview-4896-effexor-xr-oral?conditionid=&sort=",
        url_end = "&page="),
  list(url_beginning = "https://reviews.webmd.com/drugs/drugreview-6997-prozac-oral?conditionid=&sort=",
        url_end = "&page="),
  list(url_beginning = "https://reviews.webmd.com/drugs/drugreview-63990-lexapro-oral?conditionid=&sort=",
        url_end = "&page=")
)
```

Now we set up th parallel backend and use foreach and %dopar% to loop in parallel. We also have to ensure that all of these new environments have the rvest package since it's necessary for the scrape_reviews function.

```
n.cores = parallel::detectCores() - 1
my.cluster = parallel::makeCluster(n.cores, type = "FORK")
doParallel::registerDoParallel(cl = my.cluster)

results = foreach(task = tasks, .packages = c("rvest")) %dopar% {
  scrape_reviews(task$url_beginning, task$num_pages)
}

parallel::stopCluster(my.cluster)

names(results) = c("cymbalta_reviews_df", "effexor_reviews_df", "prozac_reviews_df", "lexapro_df")
cymbalta_df = results$cymbalta_reviews_df
effexor_df = results$effexor_reviews_df
prozac_df = results$prozac_reviews_df
lexapro_df = results$lexapro_df
```

For processing and publishing purposes on RPubS, we marked the above scraping code with {r, eval=FALSE} to prevent it from running. The if statement was used solely to demonstrate our data acquisition method.

We generated CSV files from the above code and imported them into GitHub using write.csv. By uploading the CSVs to GitHub, we ensure reproducibility. We now proceed to the data cleaning and tidying phase.

2. Data loading, cleaning, and tidying

```
library(httr)

# Reading a CSV file from a raw GitHub link - WEBMD Scrapped Data
raw_cymbalta_reviews <- read.csv(url("https://raw.githubusercontent.com/hbedros/SSRI-SNRI-Gender-Sentiment-analysis/master/data/raw_cymbalta_reviews.csv"))
raw_effexor_reviews <- read.csv(url("https://raw.githubusercontent.com/hbedros/SSRI-SNRI-Gender-Sentiment-analysis/master/data/raw_effexor_reviews.csv"))
raw_lexapro_reviews <- read.csv(url("https://raw.githubusercontent.com/hbedros/SSRI-SNRI-Gender-Sentiment-analysis/master/data/raw_lexapro_reviews.csv"))
raw_prozac_reviews <- read.csv(url("https://raw.githubusercontent.com/hbedros/SSRI-SNRI-Gender-Sentiment-analysis/master/data/raw_prozac_reviews.csv"))
```

```

library(tidyr)
library(dplyr)

# Rename the first column of raw_prozac_reviews to 'reviewer_info'
names(raw_prozac_reviews)[1] <- "reviewer_info"

# Define the cleanup function
cleanup_data <- function(raw_cymbalta_reviews) {

# Separate the 'reviewer_info' columns into separate columns
data_separated <- raw_cymbalta_reviews %>%
  separate(reviewer_info, into = paste0("column", 1:5), sep = " \\| ", fill = "right", extra = "merge")

# Define the 'reassign_columns' function to handle reassignment
reassign_columns <- function(df) {
  df %>%
    mutate(
      # Assign the first column as Name
      Name = column1,
      # Extract the first age range number as the Age using RegEx. If not present, return NA.
      Age = case_when(
        grepl("^\\d{2}-\\d{2}", column2) ~ sub("^((\\d{2})-\\d{2})", "\\1", column2),
        grepl("^\\d{2}-\\d{2}", column3) ~ sub("^((\\d{2})-\\d{2})", "\\1", column3),
        grepl("75 or over", column2) ~ "75+",
        grepl("75 or over", column3) ~ "75+",
        TRUE ~ NA_character_
      ),
      # Identify and assign the Gender. If not present, return NA.
      Gender = case_when(
        grepl("Male|Female|Nonbinary|Transgender", column3) ~ column3,
        grepl("Male|Female|Nonbinary|Transgender", column4) ~ column4,
        TRUE ~ NA_character_
      ),
      # Check for medication duration and assign it MedDur_Months If not present, return NA.
      MedDur_Months = case_when(
        grepl("On medication for", column4) ~ column4,
        grepl("On medication for", column5) ~ column5,
        TRUE ~ NA_character_
      ),
      # Assign Role based on the content or default to 'Patient'.
      Role = ifelse(grepl("Patient|Caregiver", column5), column5,
        ifelse(grepl("Patient|Caregiver", column4), column4, "Patient"))
    ) %>%

  select(Name, Age, Gender, MedDur_Months, Role)
}

# Create 'reassign_columns' function to the separated data
data_cleaned <- reassign_columns(data_separated)

# Remove the first column from 'raw_cymbalta_reviews'
data_dropped_column <- select(raw_cymbalta_reviews, -reviewer_info)

```

```

# Merge 'data_dropped_column' with 'data11_cleaned'
# If there is no unique ID to merge by, we'll assume that rows align and we can bind by row number.
cymbalta_reviews <- bind_cols(data_cleaned, data_dropped_column)

# Custom function to convert medication duration strings to a range in months
convert_to_month_range <- function(duration) {
  # Remove the leading space and "On medication for" part
  duration <- gsub(" On medication for ", "", duration)
  # Define the conversion pattern
  pattern <- c("less than 1 month" = "0-1",
               "1 to 6 months" = "1-6",
               "6 months to less than 1 year" = "6-12",
               "1 to less than 2 years" = "12-24",
               "2 to less than 5 years" = "24-60",
               "5 to less than 10 years" = "60-120",
               "10 years or more" = "120+")
  # Match the pattern and return the corresponding range
  return(pattern[duration])
}

# Apply the custom function to the 'Medication Duration' column
cymbalta_reviews <- cymbalta_reviews %>%
  mutate(MedDur_Months = ifelse(is.na(MedDur_Months), NA, convert_to_month_range(MedDur_Months)))

# Remove spaces from column 'Role'
cymbalta_reviews <- cymbalta_reviews %>%
  mutate(Role = trimws(Role, which = "left"))

cymbalta_reviews <- cymbalta_reviews %>%
  mutate(Gender = trimws(Gender, which = "left"))
}

# List of our raw dataset names
raw_dataset_names <- c("raw_cymbalta_reviews", "raw_effexor_reviews", "raw_lexapro_reviews", "raw_proza")

# Apply the cleanup function to each dataset and assign the results to separate variables
for (dataset_name in raw_dataset_names) {
  # Use get() to retrieve the value of the variable by name
  dataset <- get(dataset_name)

  # Apply the cleanup_data function
  cleaned_data <- cleanup_data(dataset)

  # Remove 'raw_' from the name to create the new variable name
  cleaned_name <- gsub("raw_", "", dataset_name)

  # Use assign() to assign the cleaned data to a new variable in the global environment
  assign(cleaned_name, cleaned_data, envir = .GlobalEnv)
}

# Function to categorize age into specified ranges
categorize_age <- function(age) {
  case_when(

```

```

age >= 7 & age <= 12 ~ "7 - 12",
age >= 13 & age <= 18 ~ "13 - 18",
age >= 19 & age <= 24 ~ "19 - 24",
age >= 25 & age <= 34 ~ "25 - 34",
age >= 35 & age <= 44 ~ "35 - 44",
age >= 45 & age <= 54 ~ "45 - 54",
age >= 55 & age <= 64 ~ "55 - 64",
age >= 65 & age <= 74 ~ "65 - 74",
age >= 75 | age == "75+" ~ "75+",
TRUE ~ NA_character_ # For NA or unclassified ages
)
}

# Apply the function to each cleaned dataset
list_datasets <- list(cymbalta_reviews, effexor_reviews, lexapro_reviews, prozac_reviews)
names(list_datasets) <- c("cymbalta_reviews", "effexor_reviews", "lexapro_reviews", "prozac_reviews")

list_datasets <- lapply(list_datasets, function(dataset) {
  dataset %>%
    mutate(Age = ifelse(!is.na(Age) & Age != "75+", as.numeric(Age), Age), # Convert to numeric, but keep "75+"
           Age = categorize_age(Age)) # Apply the age categorization function
})

# Extracting the datasets back to their respective variables
list2env(list_datasets, envir = .GlobalEnv)

```

```
## <environment: R_GlobalEnv>
```

3. Unnesting

We start by unnesting. We need to separate the words of a review such that each row of a dataframe contains a single word.

```

library(tidytext)

cymbalta_reviews$Date = as.Date(cymbalta_reviews$Date, format = "%m/%d/%Y")
tidy_cymbalta = cymbalta_reviews %>%
  arrange(Date) %>%
  mutate(review_id = row_number()) %>%
  unnest_tokens(word, Review)

effexor_reviews$Date = as.Date(effexor_reviews$Date, format = "%m/%d/%Y") #Just in case we'd like to sort over Date
tidy_effexor = effexor_reviews %>%
  arrange(Date) %>%
  mutate(review_id = row_number()) %>%
  unnest_tokens(word, Review)

lexapro_reviews$Date = as.Date(lexapro_reviews$Date, format = "%m/%d/%Y") # in case we want to sort over Date
tidy_lexapro = lexapro_reviews %>%
  arrange(Date) %>%
  mutate(review_id = row_number()) %>%
  unnest_tokens(word, Review)

```

```

prozac_reviews = prozac_reviews %>% rename(Review = review_text, Date = date)

prozac_reviews$Date = as.Date(prozac_reviews$Date, format = "%m/%d/%Y") # in case we want to sort over
tidy_prozac = prozac_reviews %>%
  arrange(Date) %>%
  mutate(review_id = row_number()) %>%
  unnest_tokens(word, Review)

head(tidy_cymbalta)

```

##	Name	Age	Gender	MedDur_Months	Role	Date	review_id	word
## 1	Painless	25 - 34	Female	1-6	Patient	2007-09-18	1	the
## 2	Painless	25 - 34	Female	1-6	Patient	2007-09-18	1	medication
## 3	Painless	25 - 34	Female	1-6	Patient	2007-09-18	1	has
## 4	Painless	25 - 34	Female	1-6	Patient	2007-09-18	1	saved
## 5	Painless	25 - 34	Female	1-6	Patient	2007-09-18	1	my
## 6	Painless	25 - 34	Female	1-6	Patient	2007-09-18	1	life

The reviews are now in tidy format. Each row represents a single word (which later we will score). We also added a column for review_id; this will enable us to sum up the scores of reviews.

1: Unnesting.

We start by unnesting. We need to separate the words of a review such that each row of a dataframe contains a single word.

```

cymbalta_reviews$Date = as.Date(cymbalta_reviews$Date, format = "%m/%d/%Y")
tidy_cymbalta = cymbalta_reviews %>%
  arrange(Date) %>%
  mutate(review_id = row_number()) %>%
  unnest_tokens(word, Review)

effexor_reviews$Date = as.Date(effexor_reviews$Date, format = "%m/%d/%Y") #Just in case we'd like to so
tidy_effexor = effexor_reviews %>%
  arrange(Date) %>%
  mutate(review_id = row_number()) %>%
  unnest_tokens(word, Review)

lexapro_reviews$Date = as.Date(lexapro_reviews$Date, format = "%m/%d/%Y") # in case we want to sort over
tidy_lexapro = lexapro_reviews %>%
  arrange(Date) %>%
  mutate(review_id = row_number()) %>%
  unnest_tokens(word, Review)

prozac_reviews$Date = as.Date(prozac_reviews$Date, format = "%m/%d/%Y") # in case we want to sort over
tidy_prozac = prozac_reviews %>%
  arrange(Date) %>%
  mutate(review_id = row_number()) %>%
  unnest_tokens(word, Review)

head(tidy_cymbalta)

```

##	Name	Age	Gender	MedDur_Months	Role	Date	review_id	word
----	------	-----	--------	---------------	------	------	-----------	------

## 1 Painless 25 - 34 Female	1-6 Patient 2007-09-18	1	the
## 2 Painless 25 - 34 Female	1-6 Patient 2007-09-18	1	medication
## 3 Painless 25 - 34 Female	1-6 Patient 2007-09-18	1	has
## 4 Painless 25 - 34 Female	1-6 Patient 2007-09-18	1	saved
## 5 Painless 25 - 34 Female	1-6 Patient 2007-09-18	1	my
## 6 Painless 25 - 34 Female	1-6 Patient 2007-09-18	1	life

The reviews are now in tidy format. Each row represents a single word (which later we will score). We also added a column for `review_id`; this will enable us to sum up the scores of reviews.

4. Scoring

We will use the AFINN lexicon to assign each row a score. We'll also remove the words depression and anxiety from the calculations, since a review could be mentioning how their experience with depression has improved due to the drug.

```
if (!require("textdata")) {
  install.packages("textdata", repos = "http://cran.us.r-project.org", dependencies = TRUE)
  library(afinn)
}

library(textdata)

afinn = get_sentiments("afinn")

#essentially making "depression" and "anxiety" stop words:
tidy_effexor = tidy_effexor %>%
  filter(!(word %in% c("depression", "anxiety"))) %>%
  inner_join(afinn, by = "word")

effexor_scores = tidy_effexor %>%
  group_by(review_id, Gender, Age, MedDur_Months, Role, Date) %>%
  summarize(total_score = sum(value, na.rm = TRUE))

# And the same for the other drugs...

tidy_cymbalta = tidy_cymbalta %>%
  filter(!(word %in% c("depression", "anxiety"))) %>%
  inner_join(afinn, by = "word")

cymbalta_scores = tidy_cymbalta %>%
  group_by(review_id, Gender, Age, MedDur_Months, Role, Date) %>%
  summarize(total_score = sum(value, na.rm = TRUE))

tidy_lexapro = tidy_lexapro %>%
  filter(!(word %in% c("depression", "anxiety"))) %>%
  inner_join(afinn, by = "word")

lexapro_scores = tidy_lexapro %>%
  group_by(review_id, Gender, Age, MedDur_Months, Role, Date) %>%
  summarize(total_score = sum(value, na.rm = TRUE))

tidy_prozac = tidy_prozac %>%
```

```

filter(!(word %in% c("depression", "anxiety"))) %>%
inner_join(afinn, by = "word")

prozac_scores = tidy_prozac %>%
  group_by(review_id, Gender, Age, MedDur_Months, Role, Date) %>%
  summarize(total_score = sum(value, na.rm = TRUE))

head(cymbalta_scores)

```

```

## # A tibble: 6 x 7
## # Groups:   review_id, Gender, Age, MedDur_Months, Role [6]
##   review_id Gender Age      MedDur_Months Role    Date      total_score
##   <int> <chr> <chr>      <chr>      <chr> <date>      <dbl>
## 1         1 Female 25 - 34 1-6        Patient 2007-09-18      -5
## 2         2 Male   25 - 34 0-1        Patient 2007-09-18       3
## 3         3 Female 25 - 34 12-24      Patient 2007-09-18      -6
## 4         4 Male   75+     0-1        Patient 2007-09-19      -3
## 5         6 Female 45 - 54 60-120     Patient 2007-09-20       2
## 6         8 Female 45 - 54 24-60     Patient 2007-09-20       7

```

```

head(effexor_scores)

```

```

## # A tibble: 6 x 7
## # Groups:   review_id, Gender, Age, MedDur_Months, Role [6]
##   review_id Gender Age      MedDur_Months Role    Date      total_score
##   <int> <chr> <chr>      <chr>      <chr> <date>      <dbl>
## 1         1 <NA> <NA>      <NA>      Patient 2007-09-18      -4
## 2         2 Male   35 - 44 12-24      Patient 2007-09-18       1
## 3         3 Female 35 - 44 60-120     Patient 2007-09-19       2
## 4         4 Female 35 - 44 24-60      Patient 2007-09-30       2
## 5         5 Female 19 - 24 24-60      Patient 2007-10-08     -18
## 6         6 Female 25 - 34 1-6        Patient 2007-11-06      15

```

```

head(lexapro_scores)

```

```

## # A tibble: 6 x 7
## # Groups:   review_id, Gender, Age, MedDur_Months, Role [6]
##   review_id Gender Age      MedDur_Months Role    Date      total_score
##   <int> <chr> <chr>      <chr>      <chr> <date>      <dbl>
## 1         1 Female 13 - 18 1-6        Patient 2007-09-20      -4
## 2         3 Female 35 - 44 24-60      Patient 2007-09-26       1
## 3         4 Female 25 - 34 12-24      Patient 2007-11-04      -7
## 4         5 Female 25 - 34 24-60      Patient 2007-11-08       1
## 5         6 Female 13 - 18 <NA>     Patient 2008-02-26      -4
## 6         7 Female 13 - 18 0-1        Patient 2008-03-18     -10

```

```

head(prozac_scores)

```

```

## # A tibble: 6 x 7
## # Groups:   review_id, Gender, Age, MedDur_Months, Role [6]
##   review_id Gender Age      MedDur_Months Role    Date      total_score

```


##	<int>	<chr>	<chr>	<chr>	<chr>	<date>	<dbl>
## 1	3	Female	25 - 34	0-1	Patient	2007-09-19	-4
## 2	4	Female	19 - 24	24-60	Patient	2007-09-19	-3
## 3	5	Female	45 - 54	1-6	Patient	2007-09-21	2
## 4	6	Female	25 - 34	12-24	Patient	2007-09-22	-3
## 5	7	Male	45 - 54	60-120	Patient	2007-09-22	2
## 6	8	Female	19 - 24	6-12	Patient	2007-09-22	-2

So now we're in great shape, with dataframes containing the scores of each review for each drug. Furthermore, the dataframes have gender information, so we're able to analyze the reviews by gender.

5. Analysis

Step 3 is a major one. We analyze the mainframes we've created. We begin the analysis by performing some visualizations

6. Visualizations

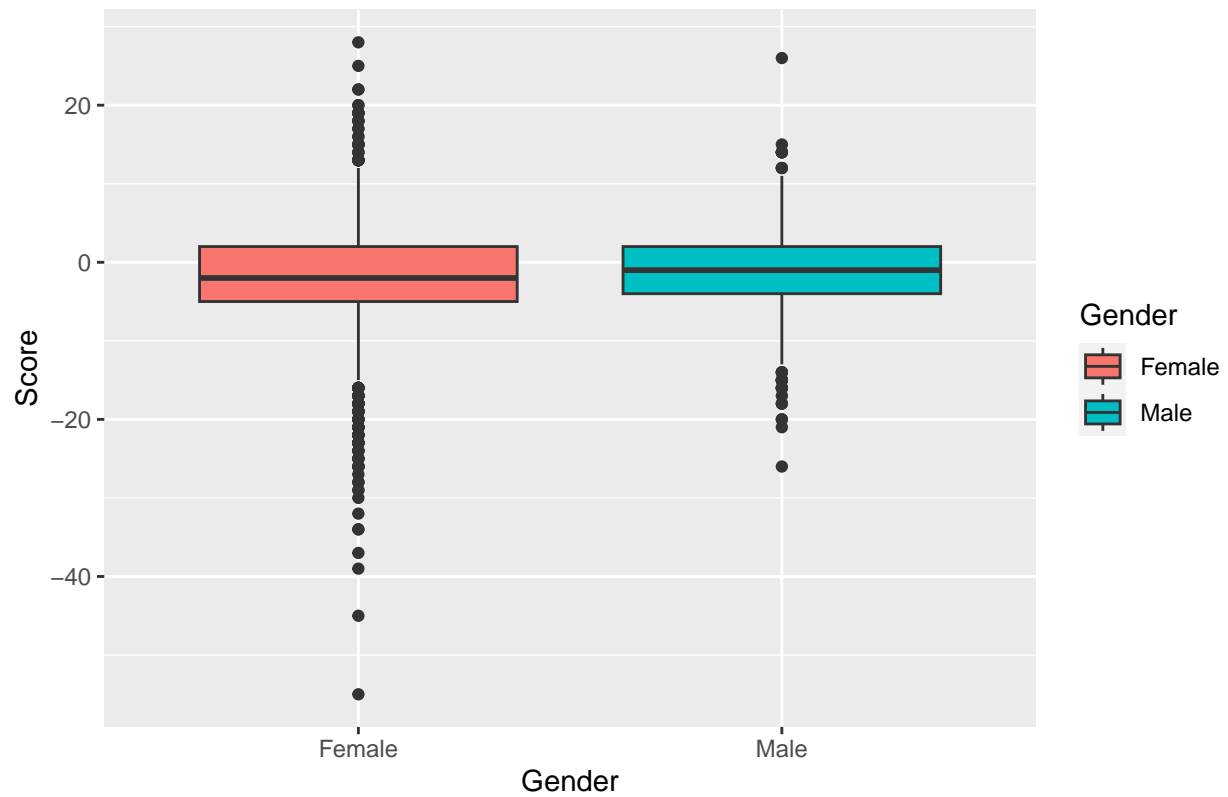
Again, we are primarily interested in how male and female reviews of these medications may differ. We start by creating box plots for all of the four drugs, filling by gender. We evaluate only male and female since those represent the vast majority of reviews. Similar analysis of other gender identities is warranted; unfortunately we lack the data at this time. We first visualize the SNRIs.

```
library(ggplot2)

cymbalta_scores = cymbalta_scores %>%
  filter(Gender %in% c("Male", "Female"))

cymbalta_scores %>% ggplot(aes(x = Gender, y = total_score, fill = Gender)) +
  geom_boxplot() +
  labs(title = "Distribution of Cymbalta Review Scores by Gender", x = "Gender", y = "Score")
```

Distribution of Cymbalta Review Scores by Gender



```
effexor_scores = effexor_scores %>%  
  filter(Gender %in% c("Male", "Female"))  
  
effexor_scores %>% ggplot(aes(x = Gender, y = total_score, fill = Gender)) +  
  geom_boxplot() +  
  labs(title = "Distribution of Effexor Review Scores by Gender", x = "Gender", y = "Score")
```

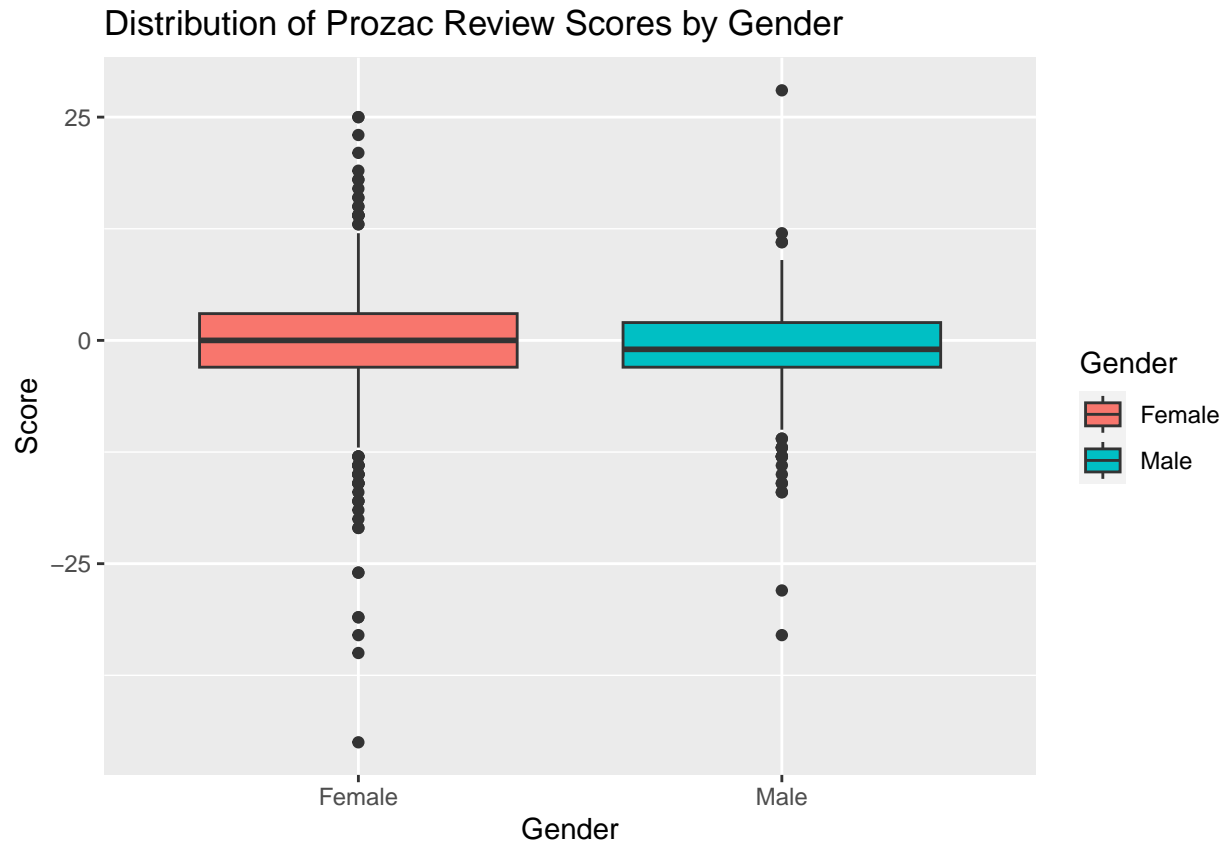


Interestingly, the results look remarkably similar for male and female reviews of Cymbalta. The IQRs for Effexor are also similar, although the male median is noticeably lower than the female median. We turn now to quickly visualize the SSRIs:

```
lexapro_scores = lexapro_scores %>%  
  filter(Gender %in% c("Male", "Female"))  
  
lexapro_scores %>% ggplot(aes(x = Gender, y = total_score, fill = Gender)) +  
  geom_boxplot() +  
  labs(title = "Distribution of Lexapro Review Scores by Gender", x = "Gender", y = "Score")
```



```
prozac_scores = prozac_scores %>%  
  filter(Gender %in% c("Male", "Female"))  
  
prozac_scores %>% ggplot(aes(x = Gender, y = total_score, fill = Gender)) +  
  geom_boxplot() +  
  labs(title = "Distribution of Prozac Review Scores by Gender", x = "Gender", y = "Score")
```



Interestingly, the IQRs for both drugs are quite small—especially male Lexapro reviews. This makes it a bit hard to analyze differences. The male and female medians look close for both drugs, although the male median looks marginally lower for both. At this time, we turn to make a more precise calculation, namely whether the differences in means between male and female reviewers is significant for both SSRIs and SNRIs:

7. ANOVA

There are three basic conditions for ANOVA: **1. Independence within and across groups.** This condition is met. **2. Nearly equal variability across the groups.** Granted, we looked at the individual drugs, but the boxplots indicate that this condition is met for both sets of drugs (SNRIs and SSRIs). **3. Nearly normal data** I'll quickly check that the data is nearly normal. First, I have to create the two datasets:

```
snri = rbind(cymbalta_scores, effexor_scores)
ssri = rbind(lexapro_scores, prozac_scores)
```

```
head(snri)
```

```
## # A tibble: 6 x 7
## # Groups:   review_id, Gender, Age, MedDur_Months, Role [6]
##   review_id Gender Age    MedDur_Months Role    Date    total_score
##       <int> <chr>  <chr>      <chr>      <chr>  <date>      <dbl>
## 1         1 Female 25 - 34 1-6      Patient 2007-09-18        -5
## 2         2 Male   25 - 34 0-1      Patient 2007-09-18         3
## 3         3 Female 25 - 34 12-24    Patient 2007-09-18        -6
```

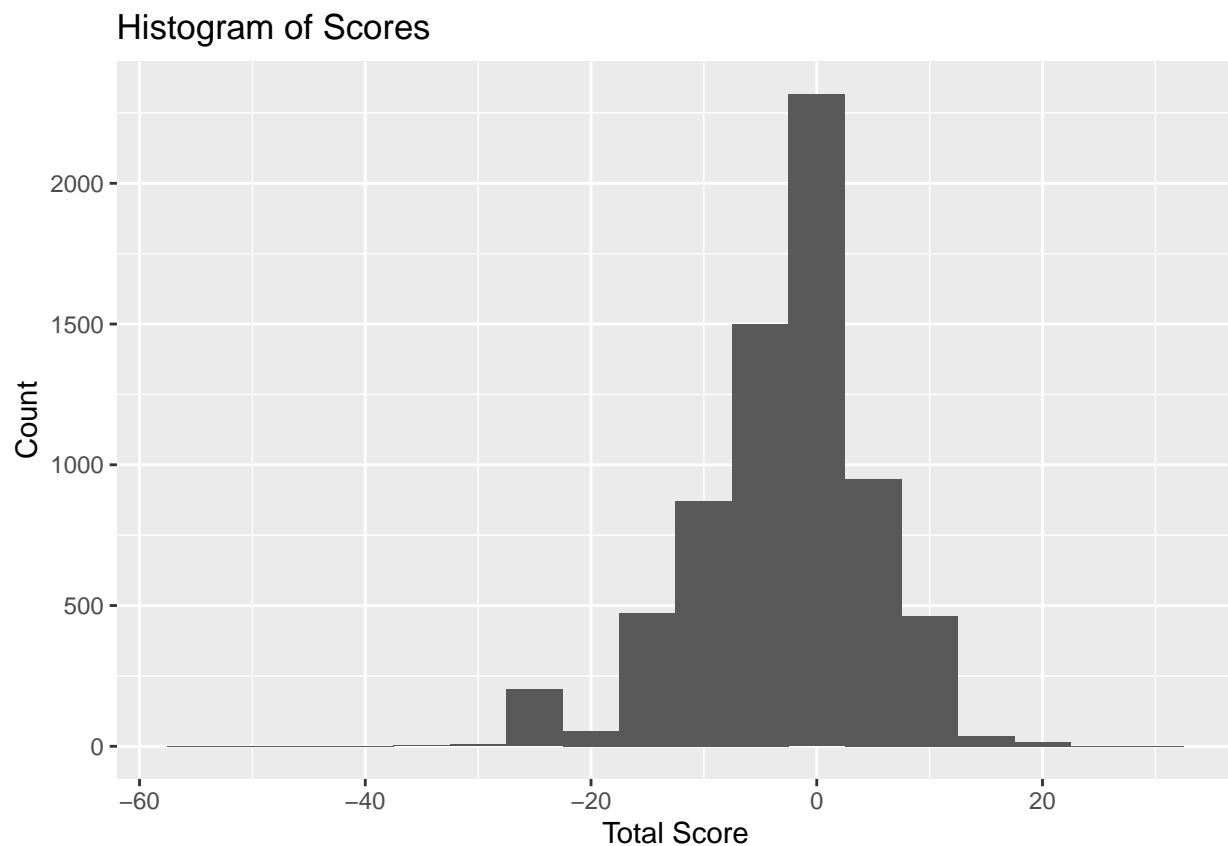
```
## 4      4 Male  75+    0-1      Patient 2007-09-19      -3
## 5      6 Female 45 - 54 60-120 Patient 2007-09-20       2
## 6      8 Female 45 - 54 24-60 Patient 2007-09-20       7
```

```
head(ssri)
```

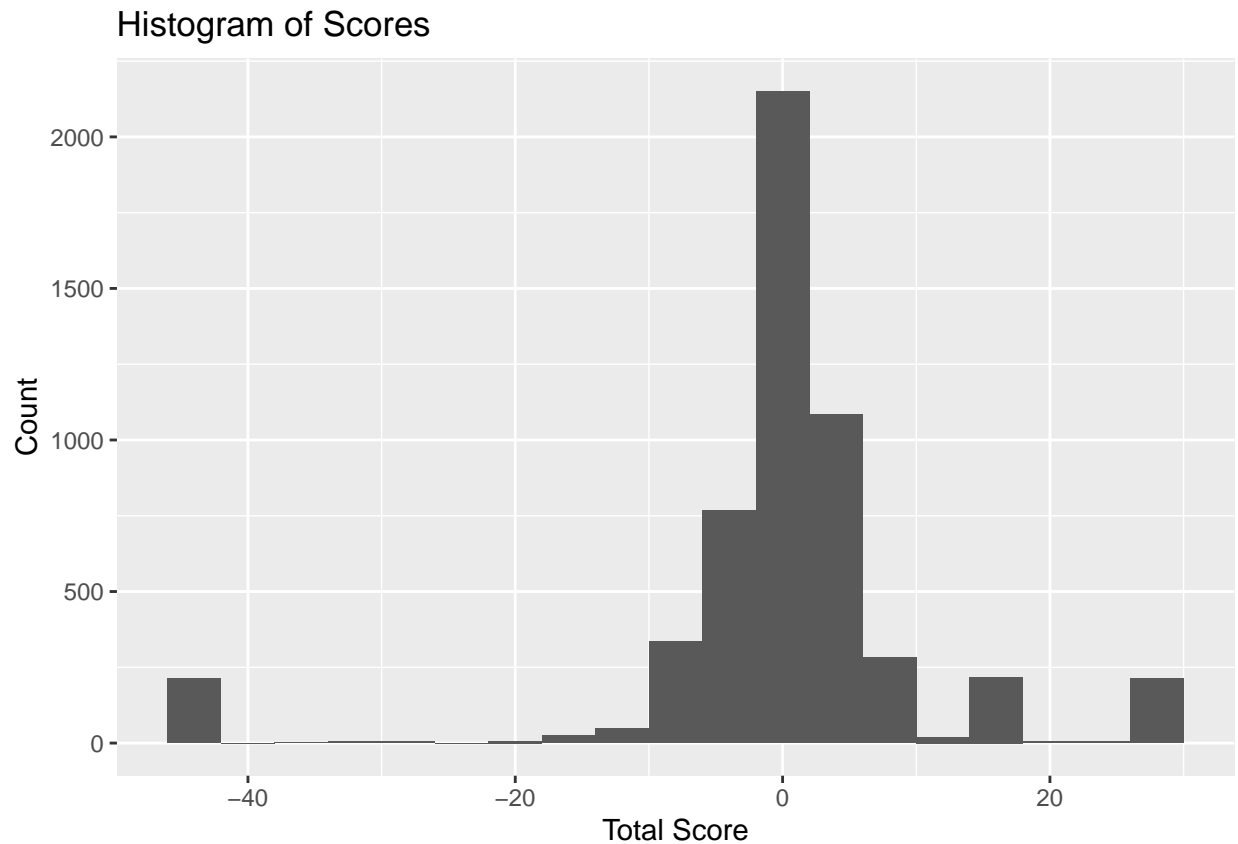
```
## # A tibble: 6 x 7
## # Groups:   review_id, Gender, Age, MedDur_Months, Role [6]
##   review_id Gender Age    MedDur_Months Role    Date    total_score
##     <int> <chr> <chr>    <chr>      <chr> <date>      <dbl>
## 1         1 Female 13 - 18 1-6        Patient 2007-09-20      -4
## 2         3 Female 35 - 44 24-60      Patient 2007-09-26       1
## 3         4 Female 25 - 34 12-24      Patient 2007-11-04     -7
## 4         5 Female 25 - 34 24-60      Patient 2007-11-08       1
## 5         6 Female 13 - 18 <NA>      Patient 2008-02-26     -4
## 6         7 Female 13 - 18 0-1        Patient 2008-03-18    -10
```

Then I'll use histograms to evaluate (near) normality.

```
snri %>% ggplot(aes(x = total_score)) +
  geom_histogram(binwidth = 5) +
  labs(title = "Histogram of Scores",
       x = "Total Score",
       y = "Count")
```



```
ssri %>% ggplot(aes(x = total_score)) +
  geom_histogram(binwidth = 4) +
  labs(title = "Histogram of Scores",
        x = "Total Score",
        y = "Count")
```



They're not perfectly normal, but frankly they're close enough to proceed with ANOVA:

We now do ANOVA for each:

```
snri_anova = aov(total_score ~ Gender, data = snri)
summary(snri_anova)
```

```
##              Df Sum Sq Mean Sq F value Pr(>F)
## Gender         1    136   136.19    2.301  0.129
## Residuals    6896 408094    59.18
```

```
ssri_anova = aov(total_score ~ Gender, data = ssri)
summary(ssri_anova)
```

```
##              Df Sum Sq Mean Sq F value Pr(>F)
## Gender         1  14593   14593    98.74 <2e-16 ***
## Residuals    5377 794666    148
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Immediately we can see that we do not reject the null hypothesis for the SNRI data. In other words, there is no statistically significant difference in male and female scores of reviews for the SNRI drugs.

On the other hand, there's an extremely low p-value for the SSRI ANOVA. We can reject the null hypothesis and conclude that there is indeed a statistically significant difference in male and female scores of reviews for the SSRI drugs.

The SSRI data is therefore especially interesting to us. Let's gather some statistics by gender:

```
gender_stats = ssri %>%
  group_by(Gender) %>%
  summarize(
    Mean = mean(total_score, na.rm = TRUE),
    Median = median(total_score, na.rm = TRUE),
    Q1 = quantile(total_score, 0.25, na.rm = TRUE),
    Q3 = quantile(total_score, 0.75, na.rm = TRUE),
    IQR = IQR(total_score, na.rm = TRUE),
    Min = min(total_score, na.rm = TRUE),
    Max = max(total_score, na.rm = TRUE)
  )

gender_stats
```

```
## # A tibble: 2 x 8
##   Gender Mean Median    Q1    Q3   IQR   Min   Max
##   <chr>   <dbl>   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 Female -1.30     2    -2     4     6   -46    25
## 2 Male   2.07     0    -1     0     1  -33    30
```

Male reviews have a lower mean and median score. However, the minimum review score and maximum review score are both lower for female reviews than they are for male reviews.

Now, let's consider the question from another angle. Suppose we know someone gave a negative review. Is that person more likely male or female? At first pass, we might be inclined to say male since the male reviews have a lower mean. However, we can define "negative reviews" in different fashions. As such, we can use a shiny app to answer this question (by allowing the user to interactively establish a threshold for what's considered a negative review). By the way, I'm new to shiny apps and I asked my friend (not in the course) to help me with this component.

```
library(shiny)
library(DT)

ui = fluidPage(
  titlePanel("Dynamic Threshold for SSRI Ratings"), #title (do we like?)
  #w will do a side and a main...in the side is the slider
  sidebarLayout(
    sidebarPanel(
      sliderInput("threshold", "Threshold for Negative Rating:",
                  min = -50, max = 0, value = -30)
    ),
    #in the main is the table
    mainPanel(
      DTOutput("tableOutput")
    )
  )
)
```



```

)

# server logic, we have inputs from UI and outputs to UI
server = function(input, output) {
  output$tableOutput = renderDT({ #inside these curly brackets is what we'll see in the table
    threshold = input$threshold

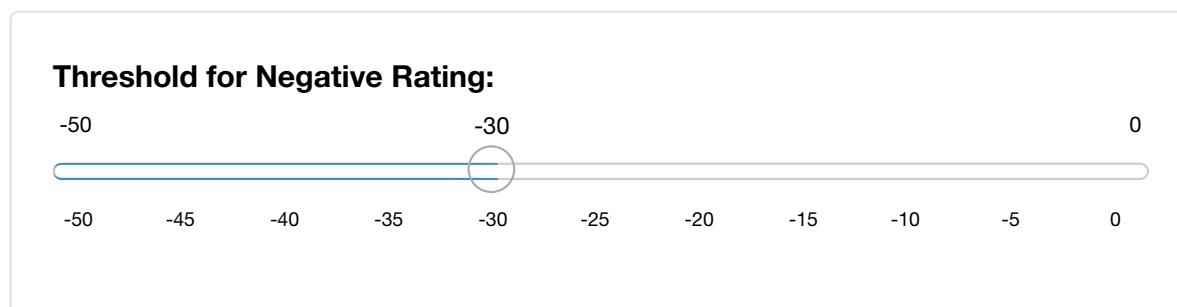
probabilities = ssri %>%
  group_by(Gender) %>%
  summarize(
    TotalRatings = n(),
    NegativeRatings = sum(total_score < threshold, na.rm = TRUE),
    Probability = NegativeRatings / TotalRatings
  )

  datatable(probabilities, options = list(pageLength = 2))
})
}

# run
shinyApp(ui = ui, server = server)

```

Dynamic Threshold for SSRI Ratings



Show entries

Search:

	Gender	TotalRatings	NegativeRatings	Probability
1	Female	3246	216	0.066543438077634
2	Male	2133	1	0.0004688232536333802

Showing 1 to 2 of 2 entries

Previous

1

Next

As it turns out, if we define a negative review as anything 0 or lower, or -3 or lower, then it is more likely that the negative reviewer is male. However, if we define it as -1 or lower, -2 or lower, -4 or lower, and so on, then the reviewer is more likely to be female. This provides quite important information as it suggests that while, overall, males may have a worse experience with these drugs, the really bad experiences may belong to female patients.

Thus far, we've assigned scores to reviews of four drugs, visualized these scores, and conducted ANOVA on SSRI and SNRI drugs to conclude that there is a statistically significant difference between the means of male and female SSRI drug review scores. We also considered the points at which a negative review is more likely to be male or female. We now turn to a more qualitative look at the different drug reviews.

Common words

```
#Split data based on gender
effexor_male <- effexor_scores %>% filter(Gender == "Male")
effexor_female <- effexor_scores %>% filter(Gender == "Female")
cymbalta_male <- cymbalta_scores %>% filter(Gender == "Male")
```

```

cymbalta_female <- cymbalta_scores %>% filter(Gender == "Female")
lexapro_male <- lexapro_scores %>% filter(Gender == "Male")
lexapro_female <- lexapro_scores %>% filter(Gender == "Female")
prozac_male <- prozac_scores %>% filter(Gender == "Male")
prozac_female <- prozac_scores %>% filter(Gender == "Female")

```

```

# Merge SSRI datasets (tidy_lexapro, tidy_prozac)
ssri_merged <- bind_rows(tidy_lexapro %>% mutate(drug = "Lexapro"),
                        tidy_prozac %>% mutate(drug = "Prozac"))
# Merge SNRI datasets (tidy_effexor, tidy_cymbalta)
snri_merged <- bind_rows(tidy_effexor %>% mutate(drug = "Effexor"),
                        tidy_cymbalta %>% mutate(drug = "Cymbalta"))

```

Combining the SSRI Datasets (prozac and lexapro)

```

ssri_positive_male <- ssri_merged %>%
  filter(Gender == "Male" & value > 0) %>%
  group_by(word) %>%
  summarise(count = n()) %>%
  arrange(desc(count)) %>%
  slice_head(n = 10)
ssri_positive_female <- ssri_merged %>%
  filter(Gender == "Female" & value > 0) %>%
  group_by(word) %>%
  summarise(count = n()) %>%
  arrange(desc(count)) %>%
  slice_head(n = 10)

```

Positive words for SSRI's based on Male and Female

```

library(wordcloud)
# Male
wordcloud(ssri_positive_male$word, ssri_positive_male$count, max.words = 10, scale = c(3, 0.5),
          main = "SSRI Positive Words - Male")

```

loved
wan
gre
li

Word cloud for SSRI's Positive words based on Male and Female

```
# Female  
wordcloud(ssri_positive_female$word, ssri_positive_female$count, max.words = 10, scale = c(3, 0.5),  
          main = "SSRI Positive Words - Female")
```



```
ssri_negative_male <- ssri_merged %>%
  filter(Gender == "Male" & value < 0) %>%
  group_by(word) %>%
  summarise(count = n()) %>%
  arrange(desc(count)) %>%
  slice_head(n = 10)
ssri_negative_female <- ssri_merged %>%
  filter(Gender == "Female" & value < 0) %>%
  group_by(word) %>%
  summarise(count = n()) %>%
  arrange(desc(count)) %>%
  slice_head(n = 10)
```

Negative words for SSRI's based on Male and Female

```
library(wordcloud) #Selina, I changed it to wordcloud (from wordcloud2 because it wasn't working)
# Male
wordcloud(ssri_negative_male$word, ssri_negative_male$count, max.words = 10, scale = c(3, 0.5),
  main = "SSRI Negative Words - Male")
```

g
n
a
horrible
los

Word cloud for SSRI's Negative words based on Male and Female

```
# Female  
wordcloud(ssri_negative_female$word, ssri_negative_female$count, max.words = 10, scale = c(3, 0.5),  
          main = "SSRI Negative Words - Female")
```



```
snri_positive_male <- snri_merged %>%
  filter(Gender == "Male" & value > 0) %>%
  group_by(word) %>%
  summarise(count = n()) %>%
  arrange(desc(count)) %>%
  slice_head(n = 10)
snri_positive_female <- snri_merged %>%
  filter(Gender == "Female" & value > 0) %>%
  group_by(word) %>%
  summarise(count = n()) %>%
  arrange(desc(count)) %>%
  slice_head(n = 10)
```

Positive words for SNRI's based on Male and Female

```
# Male
wordcloud(snri_positive_male$word, snri_positive_male$count, max.words = 10, scale = c(3, 0.5),
  main = "SNRI Positive Words - Male")
```

fe
great
better S
go

Word cloud for SNRI's Positive words based on Male and Female

```
# Female  
wordcloud(snri_positive_female$word, snri_positive_female$count, max.words = 10, scale = c(3, 0.5),  
          main = "SNRI Positive Words - Female")
```




```
snri_negative_male <- snri_merged %>%
  filter(Gender == "Male" & value < 0) %>%
  group_by(word) %>%
  summarise(count = n()) %>%
  arrange(desc(count)) %>%
  slice_head(n = 10)
snri_negative_female <- snri_merged %>%
  filter(Gender == "Female" & value < 0) %>%
  group_by(word) %>%
  summarise(count = n()) %>%
  arrange(desc(count)) %>%
  slice_head(n = 10)
```

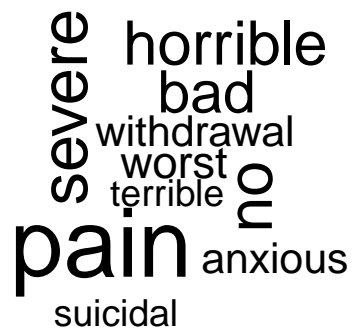
Negative words for SNRI's based on Male and Female

```
# Male
wordcloud(snri_negative_male$word, snri_negative_male$count, max.words = 10, scale = c(3, 0.5),
  main = "SNRI Negative Words - Male")
```

cover
I
stop

Word cloud for SNRI's Negative words based on Male and Female

```
# Female
wordcloud(snri_negative_female$word, snri_negative_female$count, max.words = 10, scale = c(3, 0.5),
          main = "SNRI Negative Words - Female")
```



A word cloud featuring the following terms: severe, horrible, bad, withdrawal, worst, terrible, no, pain, anxious, and suicidal. The word 'pain' is the largest and most prominent, positioned centrally. Other words are arranged around it in various sizes and orientations, with 'severe' and 'horrible' being the next largest.

Conclusion

@Zainab please update

Then I will add a graphic that represents our conclusion