Data Science 1 - Midterm Project

Adam Haertter & Brennan Mulligan

The Assignment

Midterm - Open Case Studies Exploration

Open Case Studies

The Open Case Studies (OCS) project is an educational resource of experiential guides that demonstrate how to effectively derive knowledge from data in real-world challenges. This project is the result of work of a team of researchers from Johns Hopkins University and it's Principal Investigator Dr. Stephanie Hicks. The project contains 12 different case studies. You can check out the projects and case studies here: https://www.opencasestudies.org/.

Midterm Tasks

For the midterm, you are going to you are going to choose a case study that you want to work with (from the approved list that is listed) below and complete a number of tasks. Specifically,

1. You need to choose a case study to work with. Here are the list of approved case studies:

- School Shootings in the United States
- Opioids in United States
- Mental Health of American Youth
- Exploring CO2 emissions across time
- Exploring global patterns of dietary behaviors associated with health risk

2. Complete the case study. The studies are really well documented and includes all of the requisite code you will need. You will already know some of the skills from class, but each study contains some new skills. Write down your work in a Quarto document. Make sure to answer all of the questions they ask throughout. The answers are included, but try to answer the questions before you look at the key.

3. Answer the suggested homework question(s) in the same Quarto document you did #2 in.
4. Come up with your own question and answer it. Your question should be a "big picture" question, not one that can be answered with a single plot or summary table.

The Process & Analysis

(Our answers start here)

Choosing an Open Case Study

For our open case study, we have chosen to analyze the Mental Health of American Youth, chosen from the provided list of Open Case Studies.

The main questions the case study intends to follow are:

1. How have depression rates in American youth changed since 2004, according to the NSDUH data? How have rates differed between different youth subgroups (age, gender, ethnicity)?
2. Do mental health services appear to be reaching more youths? Again, how have rates differed between different youth subgroups (age, gender, ethnicity)?

Following the Case Study

Data Import

Following the case study, data should be obtained by scraping the web. Specifically, we need to scrape the NSDUH survey site.

```
# Data Import
table11.1a <- read_html("https://www.samhsa.gov/data/sites/default/files/cbhsq-reports/NSD
  html_nodes(xpath = "/html/body/div[4]/div[1]/table") %>%
  html_table() %>%
  .[[1]]

# Defining a scraper function
scraper <- function(XPATH) {
```

```
  NSDUH_url <- "https://www.opencasestudies.org/ocs-bp-youth-mental-health/data/raw/samhsa
  table <- NSDUH_url %>%
    read_html() %>%
    html_nodes(xpath = XPATH) %>%
    html_table()
  output <- table[[1]]
  output
}

# Scraping the rest of the tables
table11.1b <- scraper(XPATH = "/html/body/div[4]/div[2]/table")
table11.2a <- scraper(XPATH = '/html/body/div[4]/div[3]/table')
table11.2b <- scraper(XPATH = '/html/body/div[4]/div[4]/table')
table11.3a <- scraper(XPATH = '/html/body/div[4]/div[5]/table')
table11.3b <- scraper(XPATH = '/html/body/div[4]/div[6]/table')
table11.4a <- scraper(XPATH = '/html/body/div[4]/div[7]/table')
table11.4b <- scraper(XPATH = '/html/body/div[4]/div[8]/table')

# Saving the data locally
save(table11.1a, table11.1b, table11.2a, table11.2b,
     table11.3a, table11.3b, table11.4a, table11.4b,
     file = here("data", "imported", "imported_data.rda"))
```

Data Wrangling

```
# Data Wrangling
load(file = here("data", "imported", "imported_data.rda"))

# Separating the legend
legend <- table11.1a %>%
  as_tibble() %>%
  select(`2004`) %>%
  tail(n = 1)

table11.1a %<>%
  as_tibble() %>%
  slice(1:(n()-1))

#slice_head(table11.1a, n = (length(pull(table11.1a, `2002`))))
#pull(legend, `2004`)
```

```r
# Unifying the NA values based on the legend
table11.1a %<>%
  mutate_all(~na_if(., "nc")) %>%
  mutate_all(~na_if(., "--")) %>%
  mutate_all(~na_if(., "")) %>%
  mutate_all(~na_if(., "*"))

#head(table11.1a)
#colnames(table11.1a)

# Cleaning up the row names
table11.1a %<>%
  rename(MHS_setting = `Setting Where Mental Health ServiceWas Received`)

#head(table11.1a)
#table11.1a %>% pull(MHS_setting)

table11.1a %<>%
  mutate(MHS_setting =
           str_remove_all(string = MHS_setting,
                          pattern = "[:digit:]|\r\n|[:punct:]|")) %>%
  mutate(MHS_setting =
           str_replace_all(string = MHS_setting,
                           pattern = "[:blank:]{1,}",
                           replacement = " "))

# Cleaning up numerical values
table11.1a %<>%
  mutate(across(.cols = -MHS_setting,
                str_remove_all,
                "a|,")) %>%
  mutate(across(.cols = -MHS_setting,
                as.numeric))
```

```
Warning: There was 1 warning in `mutate()`.
i In argument: `across(.cols = -MHS_setting, str_remove_all, "a|,")`.
Caused by warning:
! The `...` argument of `across()` is deprecated as of dplyr 1.1.0.
Supply arguments directly to `.fns` through an anonymous function instead.

  # Previously
```

```
across(a:b, mean, na.rm = TRUE)

# Now
across(a:b, \(x) mean(x, na.rm = TRUE))
```

```
# Type, subtype, and short labels
table11.1a %<>%
  mutate(type = c(rep("Specialty", 9),
                  rep("Nonspecialty", 11))) %>%
  mutate(subtype = c("Specialty_total",
                     rep("Outpatient", 5),
                     rep("Inpatient", 3),
                     "Nonspecialty_total",
                     rep("Education", 3),
                     rep("General_medicine", 2),
                     rep("Juvenile_Justice", 2),
                     rep("Child_Welfare", 2),
                     "combination")) %>%
  mutate(short_label = c("Specialty total",
                         "Outpatient total",
                         "Therapist",
                         "Clinic",
                         "Day program",
                         "In-home Therapist",
                         "Inpatient total",
                         "Hospital",
                         "Residential Center",
                         "Nonspecialty total",
                         "School total",
                         "School Therapist",
                         "School Program",
                         "General Medicine",
                         "Family Dr",
                         "Justice System",
                         "Justice System",
                         "Welfare",
                         "Fostercare",
                         "Specialty Combination"))

# Filtering out empty rows
table11.1a %<>%
```

```
    filter(MHS_setting != "General_Medicine") %>%
    filter(MHS_setting != "Juvenile_Justice") %>%
    filter(MHS_setting != "Child_Welfare")

  head(table11.1a)
```

```
# A tibble: 6 x 21
  MHS_setting     `2002` `2003` `2004` `2005` `2006` `2007` `2008` `2009` `2010`
  <chr>            <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>
1 SPECIALTY MENT~   2898   3065   3348   3362   3255   3104   3129   2925   2920
2 Outpatient        2662   2795   3015   3048   2931   2787   2837   2650   2635
3 Private Therap~   2254   2347   2523   2573   2416   2365   2408   2296   2265
4 Mental Health ~    611    635    716    657    587    583    567    537    547
5 Partial Day Ho~    440    425    439    449    471    416    374    340    362
6 InHome Therapi~    693    656    762    731    719    707    716    657    674
# i 11 more variables: `2011` <dbl>, `2012` <dbl>, `2013` <dbl>, `2014` <dbl>,
#   `2015` <dbl>, `2016` <dbl>, `2017` <dbl>, `2018` <dbl>, type <chr>,
#   subtype <chr>, short_label <chr>
```

```
  dim(table11.1a)
```

```
[1] 20 21
```

```
  # Pivoting data
  table11.1a %<>%
    pivot_longer(cols = contains("20"),
                         names_to = "Year",
                         values_to = "Number") %>%
    mutate(Year = as.numeric(Year))

  dim(table11.1a)
```

```
[1] 340   6
```

```
  head(table11.1a)
```

```
# A tibble: 6 x 6
  MHS_setting                       type      subtype     short_label  Year Number
  <chr>                             <chr>     <chr>       <chr>       <dbl>  <dbl>
1 SPECIALTY MENTAL HEALTH SERVICE Specialty Specialty_~ Specialty ~  2002   2898
2 SPECIALTY MENTAL HEALTH SERVICE Specialty Specialty_~ Specialty ~  2003   3065
3 SPECIALTY MENTAL HEALTH SERVICE Specialty Specialty_~ Specialty ~  2004   3348
4 SPECIALTY MENTAL HEALTH SERVICE Specialty Specialty_~ Specialty ~  2005   3362
5 SPECIALTY MENTAL HEALTH SERVICE Specialty Specialty_~ Specialty ~  2006   3255
6 SPECIALTY MENTAL HEALTH SERVICE Specialty Specialty_~ Specialty ~  2007   3104
```

Question Opportunity:

Why do we have 340 rows now?

Answer: We now have 340 rows because we have pivoted the table to be longer. Previously, the dimensions were 20 x 21, and after the pivot the dimensions are 340 x 6. The number 340 comes from the fact that all of the columns corresponding to years (2002 to 2018) are now each their own rows, with the other columns being preserved. There are 17 of these columns, each with 20 rows before the pivot. Because each of these now becomes its own row, we can see where the new row number comes from:

$$17 * 20 = 340$$

Continuing with the data wrangling, we can wrap these previous steps into a singular function which will then be applied to Table 11.1b.

```r
# Continue Data Wrangling

# Defining a similar summary function for future
# This just requires retracing the same steps as above
data_prep_settings <- function(TABLE, new_col, pivot_col){
  # Convert to tibble
  as_tibble(TABLE) %>%
  # Remove last row
  slice(1:(n() - 1)) %>%
  # Convert bad values to NA
  mutate_all(~na_if(., "nc")) %>%
  mutate_all(~na_if(., "--")) %>%
  mutate_all(~na_if(., "")) %>%
  mutate_all(~na_if(., "*")) %>%
  # Rename column
  rename({{new_col}} := names(.)[1]) %>%
  # Clean unwanted characters out
```

```
  mutate({{new_col}} :=
        str_remove_all(string = pull(., {{new_col}}),
                       pattern = "[:digit:]|\r\n|[:punct:]|")) %>%
  mutate({{new_col}} :=
        str_replace_all(string =pull(., {{new_col}}),
                        pattern = "[:blank:]{1,}",
                        replacement = " ")) %>%
  mutate(across(.cols = -{{new_col}},
                      str_remove_all, "a|,")) %>%
  # Make columns numeric where applicable
  mutate(across(-{{new_col}}, as.numeric)) %>%
  # Add new variables
  mutate(type = c(rep("Specialty", 9), rep("Nonspecialty", 11))) %>%
  mutate(subtype = c("Specialty_total",
                      rep("Outpatient", 5),
                      rep("Inpatient", 3),
                      "Nonspecialty_total",
                      rep("Education", 3),
                      rep("General_medicine", 2),
                      rep("Juvenile_Justice", 2),
                      rep("Child_Welfare", 2),
                      "combination")) %>%
  mutate(short_label = c("Specialty total", "Outpatient total",
                         "Therapist", "Clinic", "Day program",
                         "In-home Therapist", "Inpatient total",
                         "Hospital", "Residential Center",
                         "Nonspecialty total", "School total",
                         "School Therapist", "School Program",
                         "General Medicine", "Family Dr",
                         "Justice System", "Justice System",
                         "Welfare", "Fostercare",
                         "Specialty Combination")) %>%
  # Remove majority NA rows
  filter(rowSums(is.na(select(., is.numeric))) <
           length(select(., is.numeric))) %>%
  # Reformat table
  pivot_longer(cols = contains("20"),
               names_to = "Year",
               values_to = pivot_col)%>%
  mutate(Year = as.numeric(Year))
}
```

```r
# Applying the transformation 11.1b
table11.1b <- data_prep_settings(TABLE = table11.1b,
                    new_col = "MHS_setting",
                    pivot_col = "Percent")
```

```
Warning: There was 1 warning in `filter()`.
i In argument: `rowSums(is.na(select(., is.numeric))) < length(select(.,
  is.numeric))`.
Caused by warning:
! Use of bare predicate functions was deprecated in tidyselect 1.1.0.
i Please use wrap predicates in `where()` instead.
  # Was:
  data %>% select(is.numeric)

  # Now:
  data %>% select(where(is.numeric))
```

```r
# Defining function for demographic tables
data_dem_settings <- function(TABLE){
  # Convert to tibble
  as_tibble(TABLE) %>%
  # Remove the last row
  slice(1:(n()-1)) %>%
  # Convert bad values to NA
  mutate_all(~na_if(., "nc")) %>%
  mutate_all(~na_if(., "--")) %>%
  mutate_all(~na_if(., "")) %>%
  mutate_all(~na_if(., "*")) %>%
  # Rename the first column to be "Demographic"
  rename(Demographic := names(.)[1]) %>%
  # Replace unwanted characters
  mutate(Demographic := str_replace_all(string = pull(., Demographic),
                                        pattern = "[:blank:]{1,}",
                                        replacement = " ")) %>%
  mutate(Demographic = str_replace(string = Demographic,
                                   pattern = "1",
                                   replacement = "Age: 1")) %>%
  # Create new variable
  mutate(subgroup = c("Total", rep("Age", 4),
                      rep("Gender", 3), rep("Race/Ethnicity", 9))) %>%
```

```
  # Remove "a" from numbers
  mutate(across(.cols = contains("20"),
                      str_remove_all, "a|,")) %>%
  # Convert to numeric
  mutate(across(contains("20"), as.numeric)) %>%
  # Remove majority NA rows
  filter(rowSums(is.na(select(., is.numeric))) < length(select(., is.numeric)))
}

# Applying table format to other tables
table11.2a <- data_dem_settings(TABLE = table11.2a)
table11.2b <- data_dem_settings(TABLE = table11.2b)
table11.3a <- data_dem_settings(TABLE = table11.3a)
table11.3b <- data_dem_settings(TABLE = table11.3b)
table11.4a <- data_dem_settings(TABLE = table11.4a)
table11.4b <- data_dem_settings(TABLE = table11.4b)

# Assigning data_type
table11.2a %<>% mutate(data_type = "Major_Depressive_Episode")
table11.2a %<>% mutate(data_type = "Major_Depressive_Episode")
table11.2b %<>% mutate(data_type = "Major_Depressive_Episode")
table11.3a %<>% mutate(data_type = "Severe_Major_Depressive_Episode")
table11.3b %<>% mutate(data_type = "Severe_Major_Depressive_Episode")
table11.4a %<>% mutate(data_type = "Treatment")
table11.4b %<>% mutate(data_type = "Treatment")

# Checking the data formats
data_dem_check <- function(TABLE){
  results <- tibble(
    tibble_check = case_when(is_tibble(TABLE) ~ "Good!",
                             TRUE ~ "Not a tibble"),
    legend_check = case_when(TABLE %>%
                               slice(n()) %>%
                               pull(`2018`) %>%
                               str_detect(pattern = "--")  ~ "Legend might still be there"
                             TRUE ~ "Good!"),
    NAs_check = case_when(any(str_detect(TABLE, pattern = "nc"))
                          ~ "NA not fixed",
                          TRUE ~ "Good!"),
    firstcol_check = case_when(names(TABLE)[1] == "Demographic"
                               ~ "Good!",
```

```
                                TRUE ~ "check first column"),
    white_space_check = case_when(any(str_detect(TABLE, pattern = "[:blank:]{2,}"))
                                  ~ "white spaces not fixed",
                                  TRUE ~ "Good!"),
    age_data_check = case_when(any(str_detect(pull(TABLE, Demographic), pattern = "^1"))
                                ~ "Age data not fixed!",
                                TRUE ~ "Good!"),
    subgroup_check = case_when(any(names(TABLE) == "subgroup")
                                ~ "Good!",
                                TRUE ~ "No subgroup variable!"),
    a_comma_check = case_when(TABLE%>% select(-Demographic, -subgroup, -data_type) %>%
                                map_df(~(str_detect(.x, pattern ="a|,"))) %>%
                                rowSums(na.rm = TRUE) %>%
                                sum() == 0
                               ~ "Good!",
                               TRUE ~ "There may be commas or the letter a in the year colu
    numeric_check = case_when(sum(map_dbl(TABLE, is.numeric))== sum(str_count(names(TABLE)
                                ~ "Good!",
                               TRUE ~ "Variables are not numeric!"),
    empty_row_check = case_when(nrow(TABLE %>%filter(rowSums(is.na(select(., is.numeric)))
                                                      length(select(., is.numeric)))) > 0
                                ~ "There are empty rows ",
                                TRUE ~ "Good!"))

  ifelse(all(results == "Good!"),
         "Data looks good!", glimpse(results))
}

tables_to_check <- list(table11.2a,
                        table11.2b,
                        table11.3a,
                        table11.3b,
                        table11.4a,
                        table11.4b)
tables_to_check %>% map(data_dem_check)
```

```
Warning in stri_detect_regex(string, pattern, negate = negate, opts_regex =
opts(pattern)): argument is not an atomic vector; coercing
```

```
Warning in stri_detect_regex(string, pattern, negate = negate, opts_regex =
opts(pattern)): argument is not an atomic vector; coercing
```

```
Warning in stri_detect_regex(string, pattern, negate = negate, opts_regex =
opts(pattern)): argument is not an atomic vector; coercing

Warning in stri_detect_regex(string, pattern, negate = negate, opts_regex =
opts(pattern)): argument is not an atomic vector; coercing

Warning in stri_detect_regex(string, pattern, negate = negate, opts_regex =
opts(pattern)): argument is not an atomic vector; coercing

Warning in stri_detect_regex(string, pattern, negate = negate, opts_regex =
opts(pattern)): argument is not an atomic vector; coercing

Warning in stri_detect_regex(string, pattern, negate = negate, opts_regex =
opts(pattern)): argument is not an atomic vector; coercing

Warning in stri_detect_regex(string, pattern, negate = negate, opts_regex =
opts(pattern)): argument is not an atomic vector; coercing

Warning in stri_detect_regex(string, pattern, negate = negate, opts_regex =
opts(pattern)): argument is not an atomic vector; coercing

Warning in stri_detect_regex(string, pattern, negate = negate, opts_regex =
opts(pattern)): argument is not an atomic vector; coercing

Warning in stri_detect_regex(string, pattern, negate = negate, opts_regex =
opts(pattern)): argument is not an atomic vector; coercing

Warning in stri_detect_regex(string, pattern, negate = negate, opts_regex =
opts(pattern)): argument is not an atomic vector; coercing

[[1]]
[1] "Data looks good!"

[[2]]
[1] "Data looks good!"

[[3]]
[1] "Data looks good!"

[[4]]
[1] "Data looks good!"
```

```
[[5]]
[1] "Data looks good!"

[[6]]
[1] "Data looks good!"
```

```
# Binding count data and percent data
counts <- bind_rows(table11.2a, table11.3a, table11.4a)
percents <- bind_rows(table11.2b, table11.3b, table11.4b)

counts %>% distinct(data_type)
```

```
# A tibble: 3 x 1
  data_type
  <chr>
1 Major_Depressive_Episode
2 Severe_Major_Depressive_Episode
3 Treatment
```

```
percents %>% distinct(data_type)
```

```
# A tibble: 3 x 1
  data_type
  <chr>
1 Major_Depressive_Episode
2 Severe_Major_Depressive_Episode
3 Treatment
```

```
counts %<>% pivot_longer(cols = contains("20"),
                         names_to = "Year",
                         values_to = "Number") %>%
            mutate(Year = as.numeric(Year))

percents %<>% pivot_longer(cols = contains("20"),
                           names_to = "Year",
                           values_to = "Percent") %>%
              mutate(Year = as.numeric(Year))
```

```r
glimpse(counts)
```

```
Rows: 570
Columns: 5
$ Demographic <chr> "TOTAL", "TOTAL", "TOTAL", "TOTAL", "TOTAL", "TOTAL", "TOT~
$ subgroup    <chr> "Total", "Total", "Total", "Total", "Total", "Total", "Tot~
$ data_type   <chr> "Major_Depressive_Episode", "Major_Depressive_Episode", "M~
$ Year        <dbl> 2004, 2005, 2006, 2007, 2008, 2009, 2010, 2011, 2012, 2013~
$ Number      <dbl> 2225, 2191, 1970, 2016, 2027, 1954, 1911, 2011, 2213, 2587~
```

```r
glimpse(percents)
```

```
Rows: 570
Columns: 5
$ Demographic <chr> "TOTAL", "TOTAL", "TOTAL", "TOTAL", "TOTAL", "TOTAL", "TOT~
$ subgroup    <chr> "Total", "Total", "Total", "Total", "Total", "Total", "Tot~
$ data_type   <chr> "Major_Depressive_Episode", "Major_Depressive_Episode", "M~
$ Year        <dbl> 2004, 2005, 2006, 2007, 2008, 2009, 2010, 2011, 2012, 2013~
$ Percent     <dbl> 9.0, 8.8, 7.9, 8.2, 8.3, 8.1, 8.0, 8.2, 9.1, 10.7, 11.4, 1~
```

```r
# Updating Demographics
percents %<>% mutate(Demographic = str_replace(string = Demographic,
                                               pattern = "AIAN",
                                           replacement = "American Indian and Alaska Native"

percents %<>% mutate(Demographic =  str_replace(string = Demographic,
                                               pattern = "NHOPI",
                                           replacement = "Native Hawaiian or Other Pacific I

counts %<>% mutate(Demographic = str_replace(string = Demographic,
                                               pattern = "AIAN",
                                           replacement = "American Indian and Alaska Native"

counts %<>% mutate(Demographic =  str_replace(string = Demographic,
                                               pattern = "NHOPI",
                                           replacement = "Native Hawaiian or Other Pacific I

# Saving data
```

```
save(percents, counts, table11.1a, table11.1b,
     file = here("data", "wrangled", "wrangled_data.rda"))
write_csv(percents, path = here("data", "wrangled", "percents.csv"))
```

```
Warning: The `path` argument of `write_csv()` is deprecated as of readr 1.4.0.
i Please use the `file` argument instead.
```

```
write_csv(counts, path = here("data", "wrangled", "counts.csv"))
write_csv(table11.1a, path = here("data", "wrangled", "table11.1a.csv"))
write_csv(table11.1b, path = here("data", "wrangled", "table11.1b.csv"))
```

Data Visualization

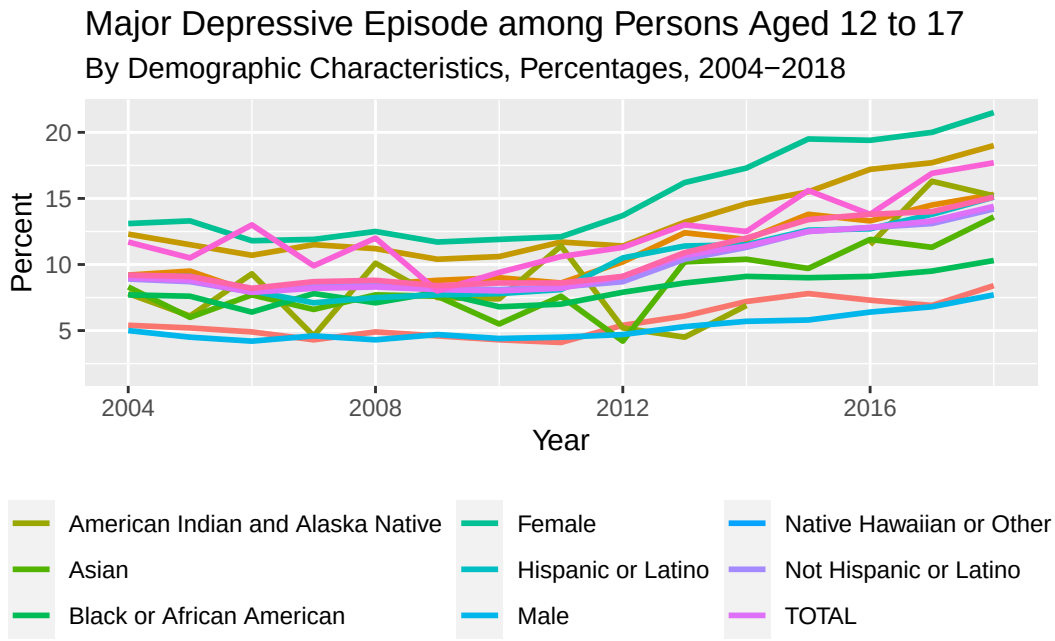
```
# Data Visualization

# Load Wrangled Data from Previous Section
load(file = here("data", "wrangled", "wrangled_data.rda"))

# Major depressive episodes among youths across time in various demographics
percents %>%
  filter(data_type == "Major_Depressive_Episode") %>%
  ggplot(aes(x = Year, y = Percent,
                      color = Demographic)) +
      geom_line(size = 1) +
      labs(title = "Major Depressive Episode among Persons Aged 12 to 17",
           subtitle = "By Demographic Characteristics, Percentages, 2004-2018") +
  theme(legend.position = "bottom")
```

```
Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
i Please use `linewidth` instead.
```

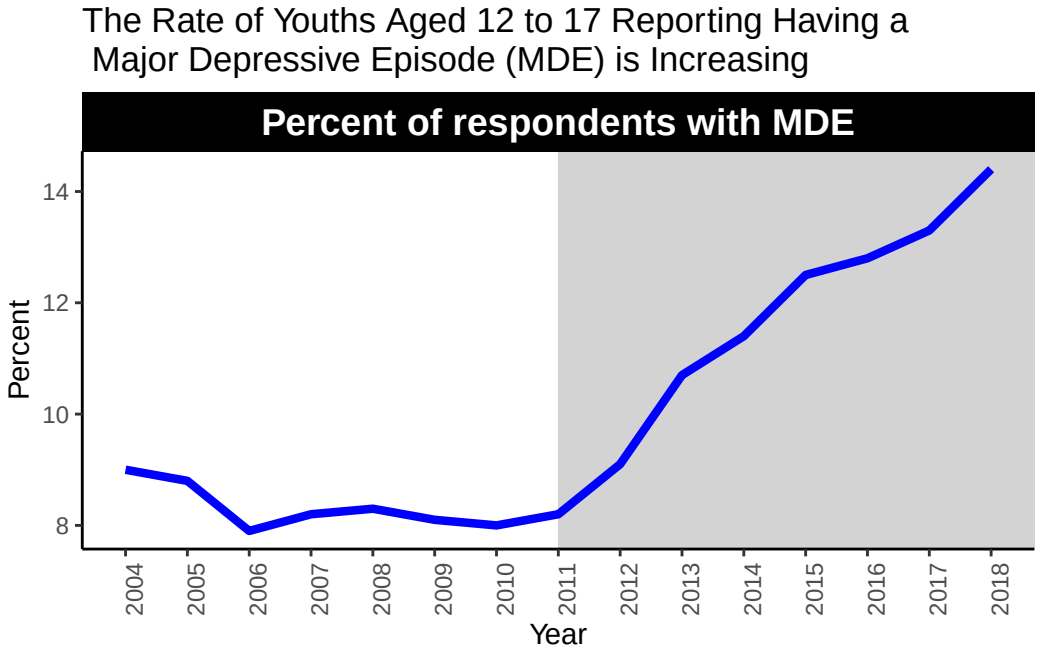
```
Warning: Removed 14 rows containing missing values (`geom_line()`).
```




```
# Major Depressive Episode Graph that emphasizes
# the increase after 2011 with a background change
MDE_total <- percents %>%
  filter(data_type == "Major_Depressive_Episode",
         Demographic == "TOTAL") %>%
  mutate(Demographic = recode(Demographic,
              "TOTAL" = "Percent of respondents with MDE")) %>%
  ggplot(aes(x = Year, y = Percent, group = Demographic)) +
    facet_wrap( ~ Demographic)+
    geom_rect(xmin = 2011, xmax = Inf,
              ymin = -Inf, ymax = Inf,
              fill = "light gray") +
    geom_line(aes(color = Demographic), size = 1.5) +
    scale_x_continuous(breaks = seq(2004, 2018, by = 1),
                       labels = seq(2004, 2018, by = 1),
                       limits = c(2004, 2018)) +
    labs(title = "The Rate of Youths Aged 12 to 17 Reporting Having a \n Major Depressive
    theme_classic() +
    theme(axis.text.x = element_text(angle = 90),
          legend.position = "none",
          strip.background = element_rect(fill ="black"),
          strip.text = element_text(face = "bold",
                                    size = 14,
```

```
                                      color = "white")) +
    scale_color_manual(values = c("blue"))

MDE_total
```

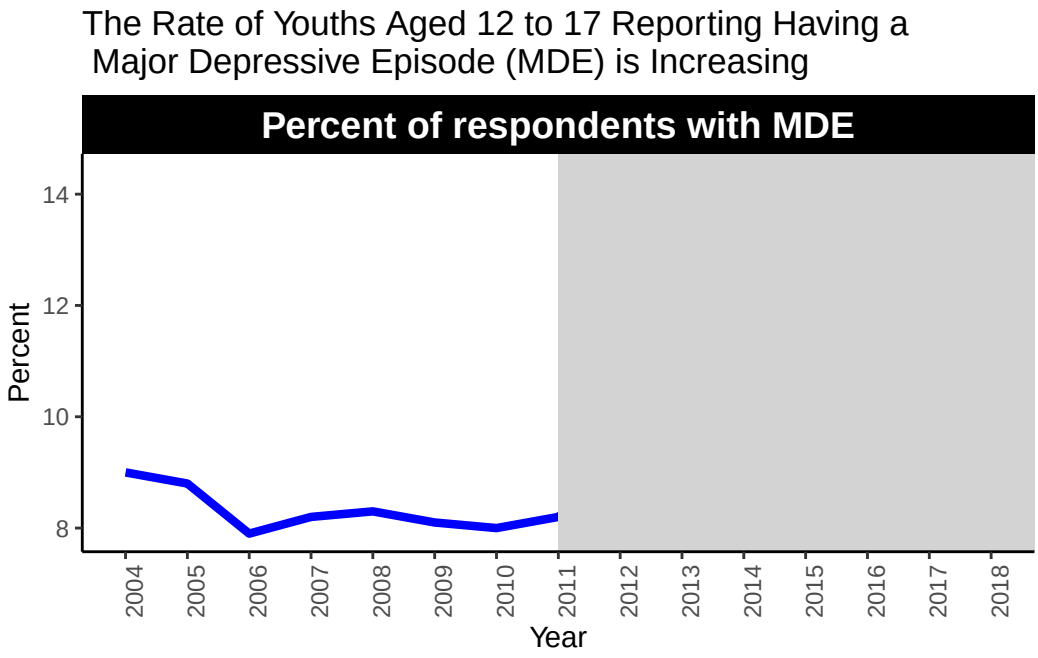



```
# Saving our graph as PNG
save(MDE_total, file = here("plots", "MDE_total.rda"))
png(here("plots", "MDE_total.png"))
MDE_total
while (!is.null(dev.list()))  dev.off()
```

Question Opportunity

What do you expect will happen when if we had used the + symbol to add the geom_rect() function with MDE_total like so? Is that what you anticipated? Why or why not?

```
MDE_total +
  geom_rect(xmin = 2011, xmax = Inf,
            ymin = -Inf, ymax = Inf,
            fill = "light gray")
```



Answer: Before running it, I expect that the rectangle will probably not act in the way that we want it to. I see now that the rectangle actually goes over the line and so now we can't actually see anything past 2011. I anticipated something like this would happen because if you're making the whole graph before adding the background, then it layers on top of what was previously created. This leads to situations like these. The code for graphs needs to act like a cake where you layer tha elements in the right order so that you can see the top as the end result in the way that you want.

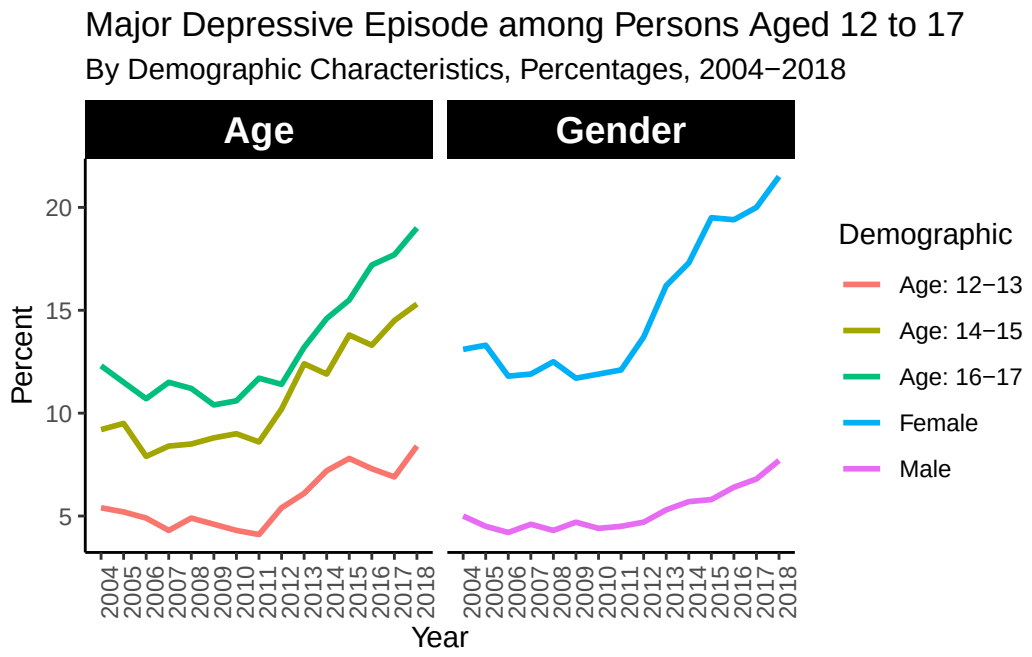
Continuing the Data Visualization Section using filters and subgroups

```
# This function helps keep a repetitive theme that we will use
ocs_theme <- function() {
  theme_classic() +
  theme(axis.text.x = element_text(angle = 90),
        strip.background = element_rect(fill = "black"),
        strip.text = element_text(face = "bold",
                                  size = 14,
                                  color = "white"))
}

#Major Depressive Episode based on Age Range and Gender Subgroups
MDE_age_gender <- percents %>%
  filter(data_type == "Major_Depressive_Episode",
```

```
         subgroup != "Race/Ethnicity",
         Demographic != "TOTAL") %>%
  ggplot(aes(x = Year, y = Percent, color = Demographic)) +
    geom_line(aes(color = Demographic), size = 1) +
    scale_x_continuous(breaks = seq(2004, 2018, by = 1),
                       labels = seq(2004, 2018, by = 1),
                       limits = c(2004, 2018)) +
    labs(title = "Major Depressive Episode among Persons Aged 12 to 17",
         subtitle = "By Demographic Characteristics, Percentages, 2004-2018") +
    facet_wrap(~ subgroup) +
    ocs_theme()

MDE_age_gender
```

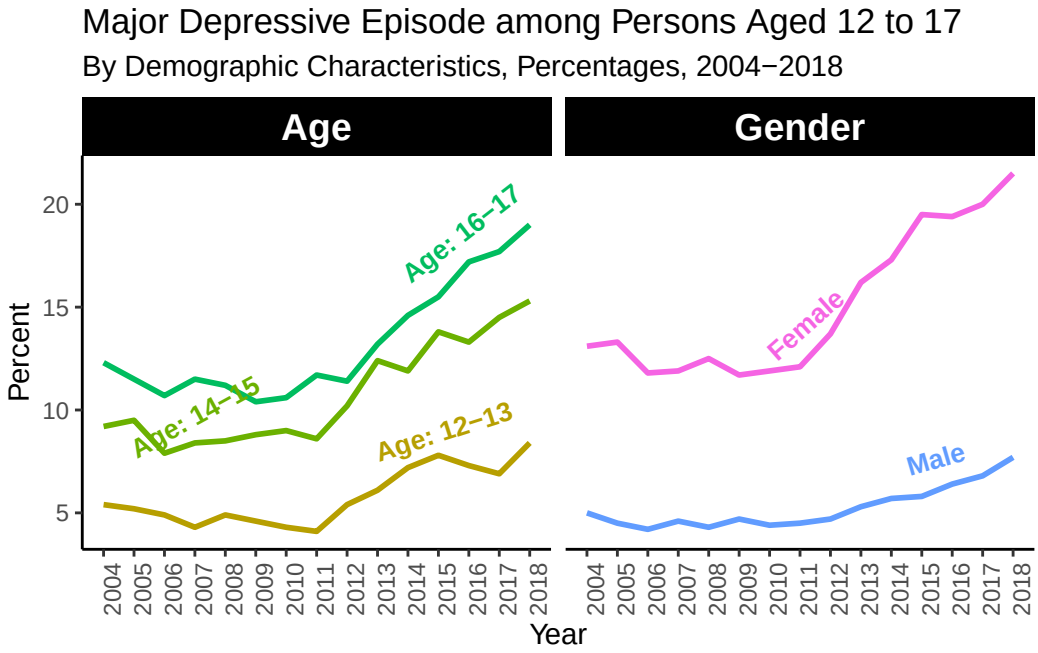



```
#Setting colors as variables
age_col_light <- c("#B79F00")
age_col<- c("#6BB100")
age_col_dark<- c("#00BD5F")
Female_col <-c("#F564E3")
Male_col <- c("#619CFF")

# Modify with different labels and font and colors
```

```
MDE_age_gender <- direct.label(
  MDE_age_gender,
  list(dl.trans(y = y + 0.38, x = x -0.1),
       "far.from.others.borders",
       cex = .8,
       fontface=c("bold"),
       dl.move("Age: 14-15", x = 2007, y = 9.7))
  ) +
  scale_color_manual(values = c(age_col_light,
                                age_col,
                                age_col_dark,
                                Female_col,
                                Male_col))

MDE_age_gender
```



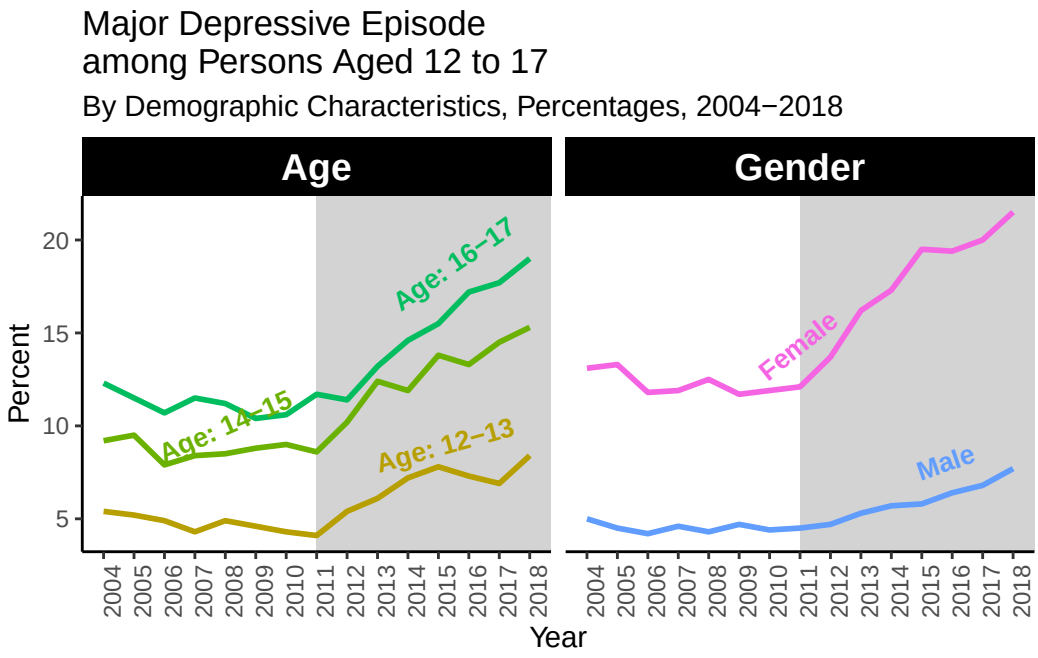
Question Opportunity:

Try to come up with the code for this plot on your own before you reveal it.

```
MDE_age_gender <-
  percents %>%
```

```
  filter(data_type == "Major_Depressive_Episode",
         subgroup != "Race/Ethnicity",
         Demographic != "TOTAL") %>%
  ggplot(aes(x = Year, y = Percent, group = Demographic)) +
    geom_rect(xmin = 2011, xmax = Inf,
              ymin = -Inf, ymax = Inf,
              fill = "light gray") +
    geom_line(aes(color = Demographic), size =1) +
    scale_x_continuous(breaks = seq(2004, 2018, by=1),
                       labels = seq(2004, 2018, by=1),
                       limits = c(2004, 2018)) +
    labs(title = "Major Depressive Episode\namong Persons Aged 12 to 17",
         subtitle = "By Demographic Characteristics, Percentages, 2004-2018") +
    facet_wrap(~ subgroup) +
    ocs_theme()

MDE_age_gender <- direct.label(
  MDE_age_gender,
  list(dl.trans(y = y +0.38, x = x -0.2),
       "far.from.others.borders",
       cex = .8,
       fontface = "bold",
       dl.move("Age: 14-15", x = 2008, y =10))
  ) +
  scale_color_manual(values = c(age_col_light,
                                age_col,
                                age_col_dark,
                                Female_col,
                                Male_col))
MDE_age_gender
```



Continuing Data Visualization

```
# Saving the graph and finishing the Data Visualization
save(MDE_age_gender, file = here("plots", "MDE_age_gender.rda"))
png(here("plots", "MDE_age_gender.png"))
MDE_age_gender
while (!is.null(dev.list()))  dev.off()

# See how it affects different Racial/Ethnic Groups
MDE_race <- percents %>%
  filter(data_type == "Major_Depressive_Episode",
         subgroup == "Race/Ethnicity") %>%
  mutate(Demographic = fct_reorder(Demographic, Percent,
                                           tail, n = 1, .desc = TRUE,
                                           .na_rm = TRUE)) %>%
  ggplot(aes(x = Year, y = Percent, group = Demographic)) +
    geom_rect(xmin = 2011, xmax = Inf,
              ymin = -Inf, ymax = Inf,
              fill = "light gray") +
    geom_line(aes(color = Demographic), size = 1) +
    facet_wrap( ~ subgroup) +
    scale_x_continuous(breaks = seq(2004, 2018, by = 1),
                       labels = seq(2004, 2018, by = 1),
```

```
                        limits = c(2004, 2018)) +
    scale_color_viridis_d() +
    labs(title = "Major Depressive Episode\namong Persons Aged 12 to 17",
         subtitle = "By Demographic Characteristics, Percentages, 2004-2018") +
    ocs_theme()

MDE_race
```

```
Warning: Removed 14 rows containing missing values (`geom_line()`).
```

Question Opportunity

How might we remove the Native Hawaiian or Other Pacific Islander group from the legend?

```
MDE_race <- percents %>%
  filter(data_type == "Major_Depressive_Episode",
         subgroup == "Race/Ethnicity",
         Demographic != "Native Hawaiian or Other Pacific Islander") %>%
  mutate(Demographic = fct_reorder(Demographic, Percent,
                                   tail, n = 1, .desc = TRUE, .na_rm = TRUE)) %>%
  ggplot(aes(x = Year, y = Percent, group = Demographic)) +
    geom_rect(xmin = 2011, xmax = Inf,
              ymin = -Inf, ymax = Inf,
              fill = "light gray") +
    geom_line(aes(color = Demographic), size = 1) +
    facet_wrap( ~ subgroup) +
    scale_x_continuous(breaks = seq(2004, 2018, by = 1),
                       labels = seq(2004, 2018, by = 1),
                       limits = c(2004, 2018)) +
    scale_color_viridis_d() +
    labs(title = "Major Depressive Episode\namong Persons Aged 12 to 17",
         subtitle = "By Demographic Characteristics, Percentages, 2004-2018") +
    ocs_theme()

save(MDE_race, file = here("plots", "MDE_race.rda"))
png(here("plots", "MDE_race.png"))
MDE_race
while (!is.null(dev.list()))  dev.off()

# Now let's take a look at how the overall rate of youths reporting having a
# major depressive episode (MDE) with severe impairment has changed over time.
```

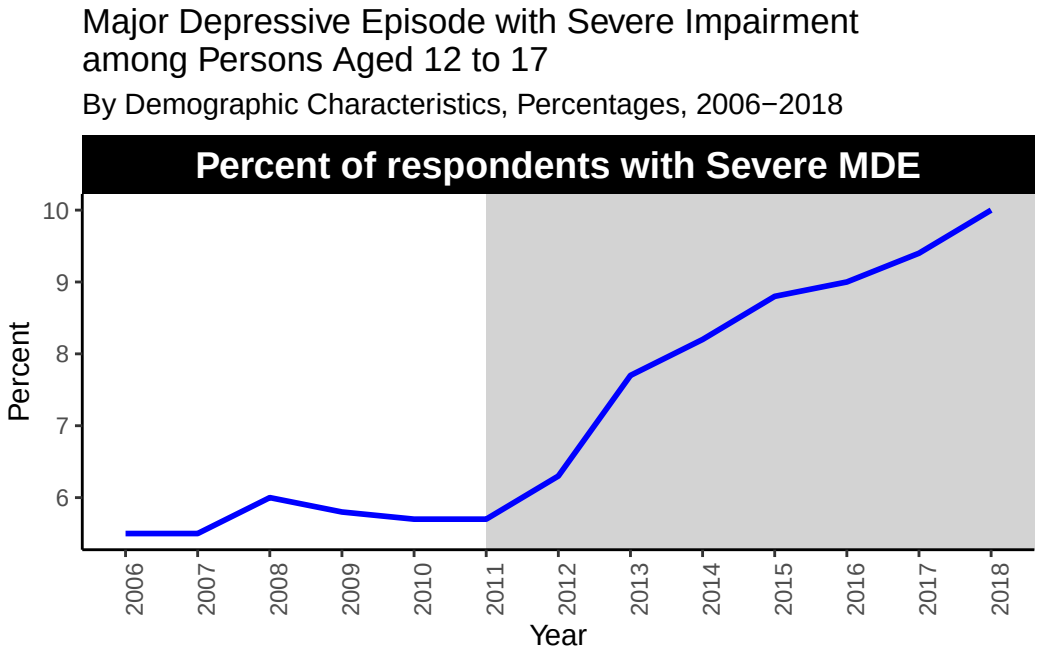
Question Opportunity

Try to come up with the code for the following two plots on your own before you reveal it. This time we will remove the legend using the theme() function.

```
MDES_total <- percents %>%
  filter(data_type == "Severe_Major_Depressive_Episode",
         Demographic == "TOTAL") %>%
  mutate(Demographic = recode(Demographic,
              "TOTAL" = "Percent of respondents with Severe MDE")) %>%
  ggplot(aes(x = Year, y = Percent, group = Demographic)) +
    geom_rect(xmin = 2011, xmax = Inf,
              ymin = -Inf, ymax = Inf,
              fill = "light gray") +
    geom_line(aes(color = Demographic), size = 1) +
    scale_x_continuous(breaks = seq(2006, 2018, by = 1),
                       labels = seq(2006, 2018, by = 1),
                       limits = c(2006, 2018)) +
    labs(title = "Major Depressive Episode with Severe Impairment\namong Persons Aged 12 t
         subtitle = "By Demographic Characteristics, Percentages, 2006-2018") +
    facet_wrap( ~ Demographic) +
    ocs_theme() +
    theme(legend.position = "none") +
    scale_color_manual(values = c("blue"))

MDES_total
```

```
Warning: Removed 2 rows containing missing values (`geom_line()`).
```



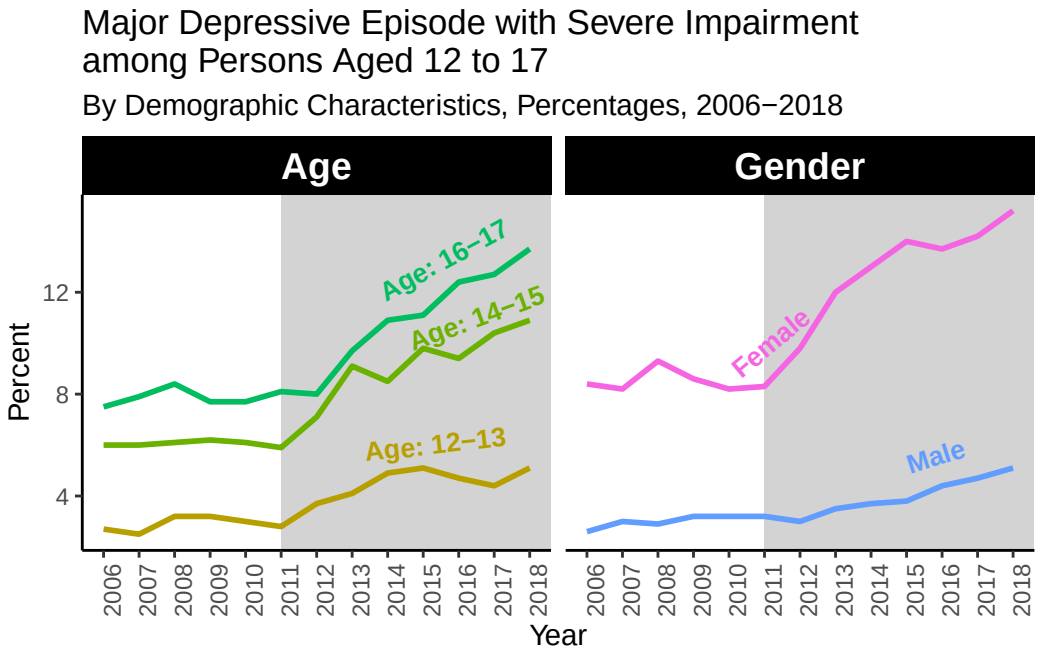

```
# Next let's look at age groups and gender differences:
MDES_age_gender <-
  percents %>%
  filter(data_type == "Severe_Major_Depressive_Episode",
         subgroup != "Race/Ethnicity",
         Demographic != "TOTAL") %>%
  ggplot(aes(x = Year, y = Percent, group = Demographic)) +
    geom_rect(xmin = 2011, xmax = Inf,
              ymin = -Inf, ymax = Inf,
              fill = "light gray") +
    geom_line(aes(color = Demographic), size = 1)+
    scale_x_continuous(breaks = seq(2006, 2018, by = 1),
                       labels = seq(2006, 2018, by = 1),
                        limits = c(2006, 2018)) +
    labs(title = "Major Depressive Episode with Severe Impairment\namong Persons Aged 12 t
         subtitle = "By Demographic Characteristics, Percentages, 2006-2018") +
    facet_wrap( ~ subgroup) +
    ocs_theme()

MDES_age_gender  <- direct.label(
  MDES_age_gender,
  list(dl.trans(y = y +0.39, x = x -0.1),
```

```
        "far.from.others.borders",
        cex = .8,
        fontface = "bold",
        dl.move("Age: 14-15", x= 2016.5, y = 11))
  ) +
  scale_color_manual(values = c(age_col_light,
                                age_col,
                                age_col_dark,
                                Female_col,
                                Male_col))
MDES_age_gender
```

```
Warning: Removed 10 rows containing missing values (`geom_line()`).
```

```
Warning: Removed 10 rows containing missing values (`geom_dl()`).
```



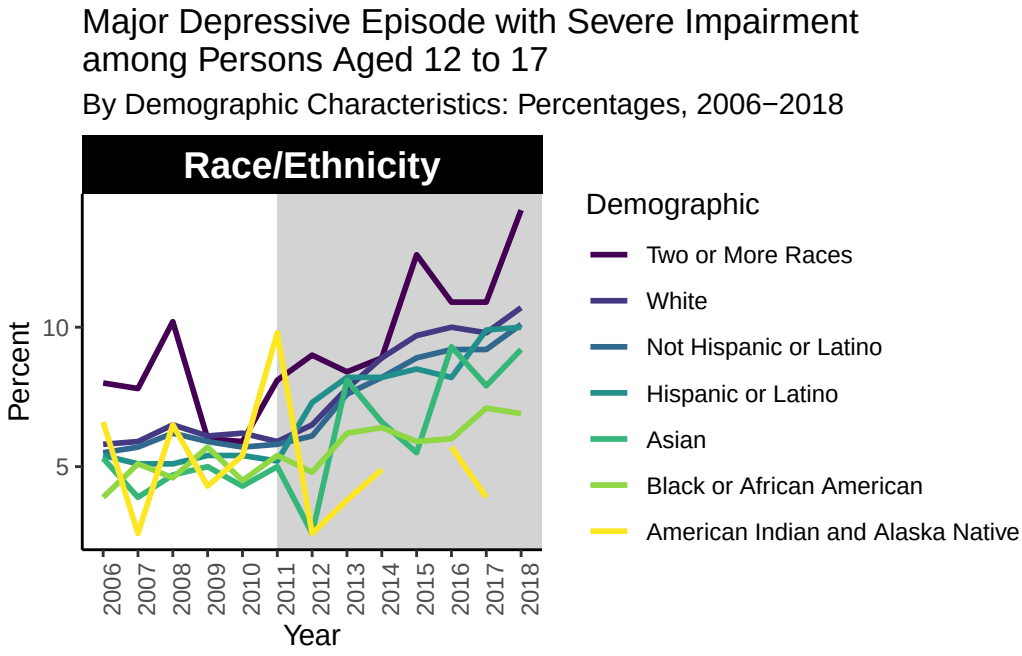
Question Opportunity

Try to come up with the code for the plot for Racial/Ethnic groups plot on your own before you reveal it.

```r
Race_MDES <- percents %>%
  filter(data_type == "Severe_Major_Depressive_Episode",
         subgroup == "Race/Ethnicity") %>%
  mutate(Demographic =
           fct_reorder(Demographic, Percent,
                       tail, n = 1, .desc = TRUE, .na_rm = TRUE)) %>%
  ggplot(aes(x = Year, y = Percent, group = Demographic)) +
    geom_rect(xmin = 2011, xmax = Inf,
              ymin = -Inf, ymax = Inf,
              fill = "light gray") +
    geom_line(aes(color = Demographic), size =1) +
    facet_wrap(~ subgroup) +
    scale_x_continuous(breaks = seq(2006, 2018, by = 1),
                       labels = seq(2006, 2018, by = 1),
                       limits = c(2006, 2018)) +
    labs(title = "Major Depressive Episode with Severe Impairment\namong Persons Aged 12 t
         subtitle = "By Demographic Characteristics: Percentages, 2006-2018") +
    ocs_theme() +
    scale_color_viridis_d()

Race_MDES
```

```
Warning: Removed 15 rows containing missing values (`geom_line()`).
```




```
# Now we will take a look at those who reported having a MDE and received
# treatment for depression.

# First, let's look overall using the Demographic == "TOTAL" group. We will
# remove the legend for this plot.
```

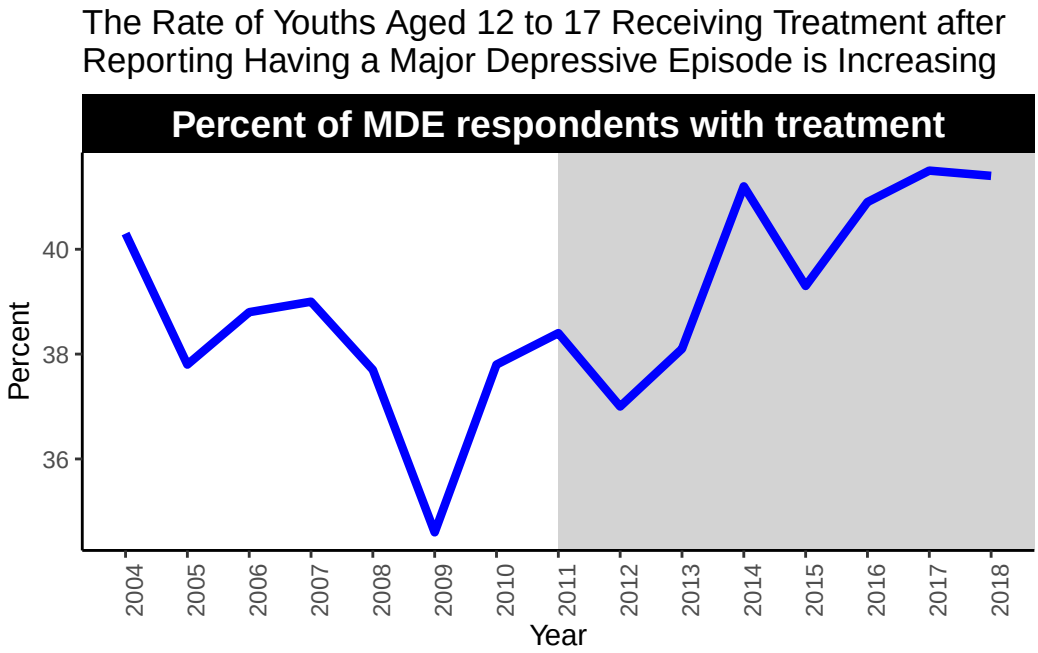
Question Opportunity

Try to come up with the code for this plot on your own before you reveal it.

```
Treat_total <- percents %>%
  filter(data_type == "Treatment",
         Demographic == "TOTAL") %>%
  mutate(Demographic = recode(Demographic,
              "TOTAL" = "Percent of MDE respondents with treatment")) %>%
  ggplot(aes(x = Year, y = Percent, group = Demographic)) +
    geom_rect(xmin = 2011, xmax = Inf,
              ymin = -Inf, ymax = Inf,
              fill = "light gray") +
    geom_line(aes(color = Demographic), size = 1.5) +
    facet_wrap(~Demographic) +
    scale_x_continuous(breaks = seq(2004, 2018, by = 1),
```

```
                        labels = seq(2004, 2018, by = 1),
                        limits = c(2004, 2018)) +
    labs(title = "The Rate of Youths Aged 12 to 17 Receiving Treatment after\nReporting Ha
    ocs_theme() +
    theme(legend.position = "none") +
    scale_color_manual(values = c("blue"))

Treat_total
```




```
save(Treat_total, file = here("plots", "Treat_total.rda"))
png(here("plots", "Treat_total.png"))
Treat_total
while (!is.null(dev.list()))  dev.off()
```

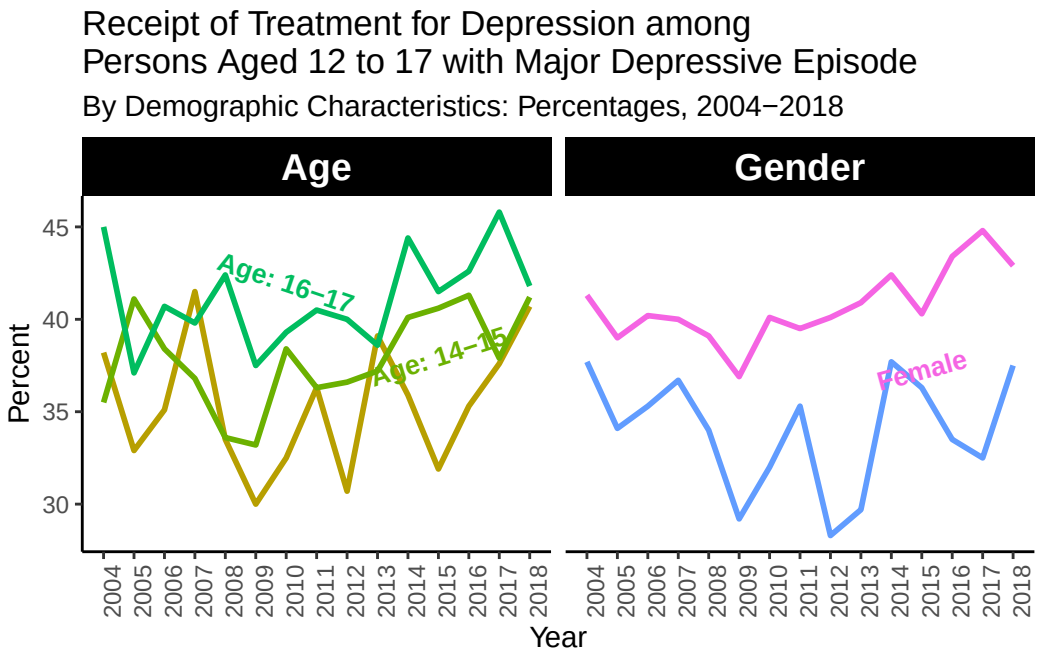
Question Opportunity

Try to come up with the code for the treat plot on your own before you reveal it.

```
treat <- percents %>%
  filter(data_type == "Treatment",
         subgroup != "Race/Ethnicity",
```

```
        subgroup != "Total") %>%
  ggplot(aes(x = Year, y = Percent, color = Demographic)) +
    geom_line(size = 1) +
    facet_wrap( ~ subgroup) +
    scale_x_continuous(breaks = seq(2004, 2018, by = 1),
                       labels = seq(2004, 2018, by = 1),
                       limits = c(2004, 2018)) +
    labs(title = "Receipt of Treatment for Depression among\nPersons Aged 12 to 17 with Ma
         subtitle = "By Demographic Characteristics: Percentages, 2004-2018") +
    ocs_theme()

treat <- direct.label(
  treat,
  list(dl.trans(y = y -1.5, x = x -0.4),
       "far.from.others.borders",
       cex = .8,
       fontface = "bold",
       dl.move("Age: 16-17", x = 2010, y = 42),
       dl.move("Age: 14-15", x = 2015, y = 38))
  ) +
  scale_color_manual(values = c(age_col_light,
                                age_col,
                                age_col_dark,
                                Female_col,
                                Male_col))
treat
```



Question Opportunity

Create a similar plot on your own for the different race / ethnic groups.

```
Race_treat <- percents %>%
  filter(data_type == "Treatment") %>%
  filter(subgroup == "Race/Ethnicity") %>%
  mutate(Demographic =
           fct_reorder(Demographic, Percent,
                       tail, n = 1, .desc = TRUE)) %>%
  ggplot(aes(x = Year, y = Percent, color = Demographic)) +
    geom_line(size = 1) +
    facet_wrap( ~ subgroup) +
    scale_x_continuous(breaks = seq(2009, 2018, by = 1),
                       labels = seq(2009, 2018, by = 1),
                       limits = c(2009, 2018)) +
    labs(title = "Receipt of Treatment for Depression among\nPersons Aged 12 to 17 with Ma
         subtitle = "By Demographic Characteristics: Percentages, 2004-2018") +
    ocs_theme() +
    scale_color_viridis_d()
```

```
Warning: There was 1 warning in `mutate()`.
i In argument: `Demographic = fct_reorder(Demographic, Percent, tail, n = 1,
```

```
  .desc = TRUE)`.
Caused by warning:
! `fct_reorder()` removing 12 missing values.
i Use `.na_rm = TRUE` to silence this message.
i Use `.na_rm = FALSE` to preserve NAs.
```

```
# Mental Health Services
Race_treat <- percents %>%
  filter(data_type == "Treatment") %>%
  filter(subgroup == "Race/Ethnicity") %>%
  mutate(Demographic =
           fct_reorder(Demographic, Percent,
                       tail, n = 1, .desc = TRUE)) %>%
  ggplot(aes(x = Year, y = Percent, color = Demographic)) +
    geom_line(size = 1) +
    facet_wrap( ~ subgroup) +
    scale_x_continuous(breaks = seq(2009, 2018, by = 1),
                       labels = seq(2009, 2018, by = 1),
                       limits = c(2009, 2018)) +
    labs(title = "Receipt of Treatment for Depression among\nPersons Aged 12 to 17 with Ma
         subtitle = "By Demographic Characteristics: Percentages, 2004-2018") +
    ocs_theme() +
    scale_color_viridis_d()
```

```
Warning: There was 1 warning in `mutate()`.
i In argument: `Demographic = fct_reorder(Demographic, Percent, tail, n = 1,
  .desc = TRUE)`.
Caused by warning:
! `fct_reorder()` removing 12 missing values.
i Use `.na_rm = TRUE` to silence this message.
i Use `.na_rm = FALSE` to preserve NAs.
```

```
# Subcategories of mental health services
plotMHSS <- table11.1b %>%
  filter(!str_detect(short_label, "total")) %>%
  ggplot(aes(x = Year, y = Percent,
         group = MHS_setting, color = short_label)) +
    geom_line(size = 1) +
    facet_wrap( ~ type) +
    scale_x_continuous(breaks = seq(2002, 2019, by = 1),
```

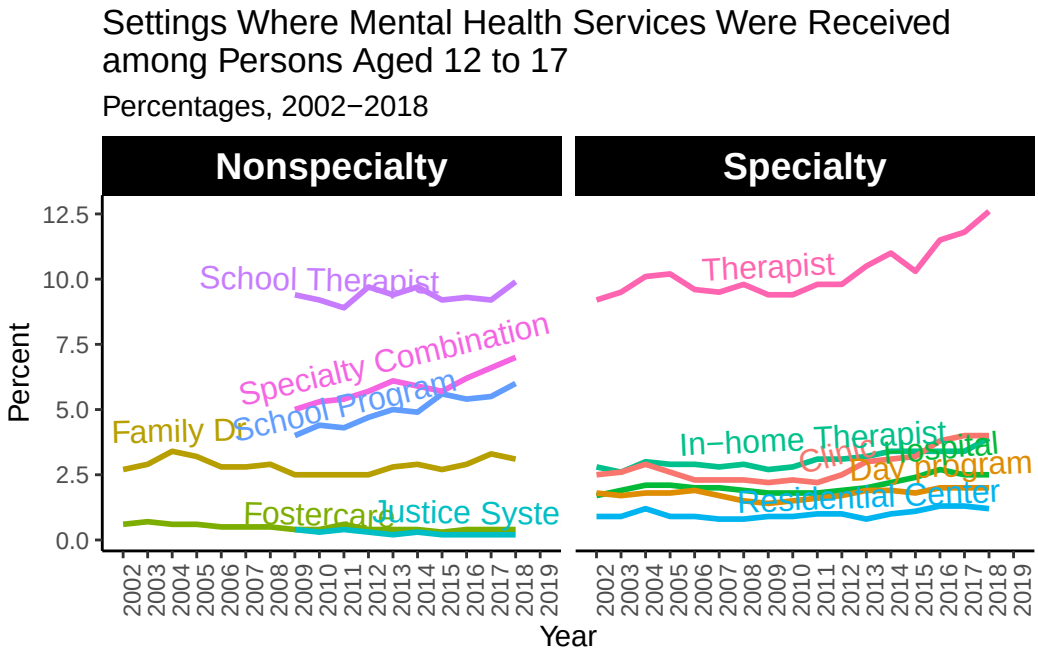
```
                        labels = seq(2002, 2019, by = 1),
                        limits = c(2002, 2019)) +
    labs(title = "Settings Where Mental Health Services Were Received\namong Persons Aged
         subtitle = "Percentages, 2002-2018") +
    ocs_theme()

plotMHSS <- direct.label(
  plotMHSS,
  list(dl.trans(y = y +0.3),
       "far.from.others.borders",
       dl.move("School Therapist", 2010, 10),
       dl.move("Fostercare", 2010, 1),
       dl.move("Therapist", x=2009, y = 10.5))
  )

plotMHSS
```

```
Warning: Removed 28 rows containing missing values (`geom_line()`).
```

```
Warning: Removed 28 rows containing missing values (`geom_dl()`).
```

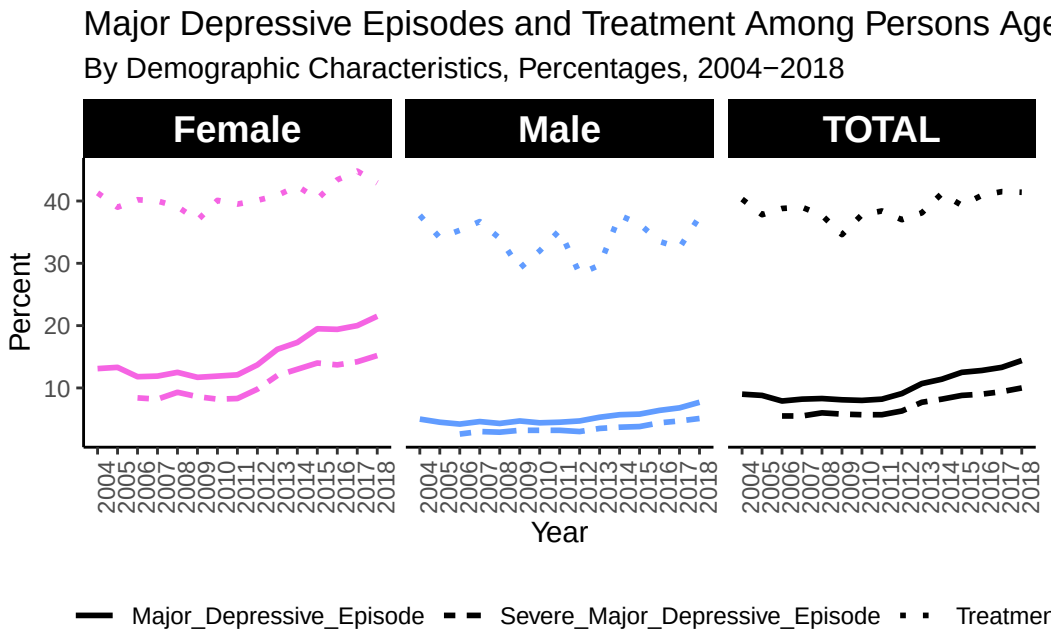


```
# Overall Outcomes by Group
gender_outcomes <-
  percents %>%
  filter(Demographic %in% c("Male", "Female", "TOTAL")) %>%
  ggplot(aes(x = Year, y = Percent, color = Demographic)) +
    geom_line(aes(linetype = data_type), size = 1) +
    scale_linetype_manual(values = c("solid", "2262", "13")) +
    scale_color_manual(values = c(Female_col, Male_col, "black")) +
    scale_x_continuous(breaks = seq(2004, 2018, by = 1),
                       labels = seq(2004, 2018, by = 1),
                       limits = c(2004, 2018)) +
    labs(title = "Major Depressive Episodes and Treatment Among Persons Aged 12 to 17",
         subtitle = "By Demographic Characteristics, Percentages, 2004-2018") +
    facet_wrap( ~ Demographic, strip.position = "top") +
  ocs_theme() +
  theme(legend.title = element_blank(),
        legend.position = "bottom") +
    guides(color = FALSE)
```

```
Warning: The `<scale>` argument of `guides()` cannot be `FALSE`. Use "none" instead as
of ggplot2 3.3.4.
```

```
gender_outcomes
```

```
Warning: Removed 6 rows containing missing values (`geom_line()`).
```



Question Opportunity

Create a similar plot on your own for different age groups.

```
age_outcomes <-percents %>%
  filter(subgroup == "Age") %>%
  ggplot(aes(x = Year, y = Percent, color = Demographic)) +
    geom_line(aes(linetype= data_type), size = 1) +
    scale_linetype_manual(values = c("solid", "2262", "13")) +
    scale_color_manual(values = c(age_col_light, age_col, age_col_dark)) +
    scale_x_continuous(breaks = seq(2004, 2018, by = 1),
                       labels = seq(2004, 2018, by = 1),
                       limits = c(2004, 2018)) +
    labs(title = "Major Depressive Episode\namong Persons Aged 12 to 17",
         subtitle = "By Demographic Characteristics, Percentages, 2004-2018") +
    facet_wrap( ~ Demographic, strip.position = "top")+
  ocs_theme() +
  theme(legend.title = element_blank(),
        legend.position = "bottom") +
    guides(color = FALSE)
```

Question Opportunity

Create a similar plot on your own for the different race / ethnic groups.

```
race_outcomes <- percents %>%
  filter(subgroup == "Race/Ethnicity",
         Demographic != "Native Hawaiian or Other Pacific Islander") %>%
  ggplot(aes(x = Year, y = Percent, color = Demographic)) +
    geom_line(aes(linetype = data_type), size=1) +
    scale_linetype_manual(values = c("solid", "2262", "13")) +
    scale_x_continuous(breaks = seq(2004, 2018, by = 1),
                       labels = seq(2004, 2018, by = 1),
                       limits = c(2004, 2018)) +
    labs(title = "Major Depressive Episode\namong Persons Aged 12 to 17",
         subtitle = "By Demographic Characteristics, Percentages, 2004-2018") +
    facet_wrap( ~ Demographic, strip.position = "top", nrow = 4) +
  ocs_theme() +
  theme(legend.title = element_blank(),
        legend.position = "bottom") +
    guides(color = FALSE)
```

Data Analysis

```
# Data Analysis
load(file = here("data", "wrangled", "wrangled_data.rda"))

# Set up chi squared data
chi_squared_11.2a <- counts %>%
  filter(data_type == "Major_Depressive_Episode") %>%
  filter(Year %in% c(2004, 2018)) %>%
  filter(Demographic %in% c("Male", "Female")) %>%
  mutate(Number = Number * 1000) # Data are in thousands
```

Question Opportunity:

Using what you just learned about pivot_wider() and select() and without scrolling up, try to come up with the code to do the wrangling for this data.

```
chi_squared_11.2a %<>%
  select(Demographic, Year, Number) %>%
  pivot_wider(names_from = Year,
```

```
                names_prefix = "Year",
                values_from = Number)
```

Continuing on following the case study, we need to continue formatting data for the chi-squared test and run the first chi-squared test.

```
# Continuing data analysis
chi_squared_11.2a %<>%
  column_to_rownames("Demographic")

chi_squared_11.2a
```

```
       Year2004 Year2018
Male     637000   946000
Female  1588000  2537000
```

```
# Run the Chi-Squared test
chisq.test(chi_squared_11.2a)
```

```
    Pearson's Chi-squared test with Yates' continuity correction

data:  chi_squared_11.2a
X-squared = 1461.2, df = 1, p-value < 2.2e-16
```

Question Opportunity:

Using what you learned about the Chi-squared test, describe the results in reference to the null hypothesis.

Answer: Since the p-value is less than 2.2^{-16}, we know that the p-value is extremely small. Without having out α significance level explicitly defined, we can assume that $\alpha = 0.05$. Our p-value is much smaller than α, so we can reject the null-hypothesis in this case. If H_0: The variables are independent, then therefore we have evidence to reject this hypothesis, and evidence suggests that the variables are not, in fact, independent.

Continuing with the data analysis, we need to set up a test to describe the size of the association present.

```r
# Set up prop_test
prop_test(chi_squared_11.2a,
          detailed = TRUE,
          correct = TRUE) %>%
  glimpse()
```

```
Rows: 1
Columns: 13
$ n           <dbl> 5708000
$ n1          <dbl> 1583000
$ n2          <dbl> 4125000
$ estimate1   <dbl> 0.4024005
$ estimate2   <dbl> 0.3849697
$ statistic   <dbl> 1461.23
$ p           <dbl> 1.040008e-319
$ df          <dbl> 1
$ conf.low    <dbl> 0.01653368
$ conf.high   <dbl> 0.01832793
$ method      <chr> "Prop test"
$ alternative <chr> "two.sided"
$ p.signif    <chr> "****"
```

```r
# Run transposed table through prop_test
t(chi_squared_11.2a) %>%
  prop_test(detailed = TRUE, correct = TRUE) %>%
  glimpse()
```

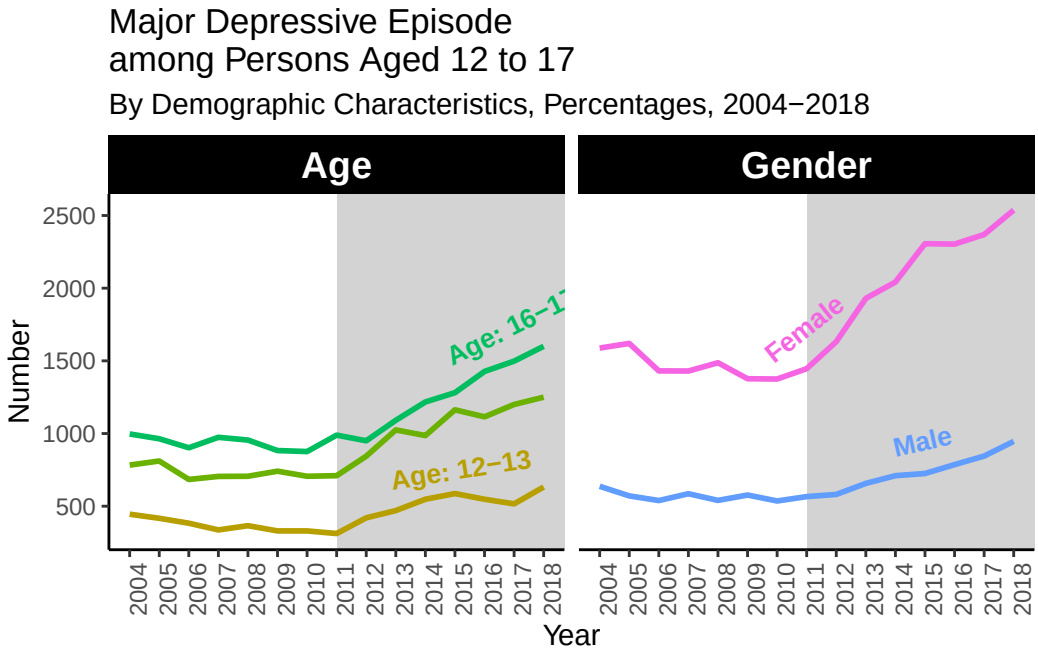
```
Rows: 1
Columns: 13
$ n           <dbl> 5708000
$ n1          <dbl> 2225000
$ n2          <dbl> 3483000
$ estimate1   <dbl> 0.2862921
$ estimate2   <dbl> 0.2716049
$ statistic   <dbl> 1461.23
$ p           <dbl> 1.040008e-319
$ df          <dbl> 1
$ conf.low    <dbl> 0.0139312
$ conf.high   <dbl> 0.01544319
```

```
$ method      <chr> "Prop test"
$ alternative <chr> "two.sided"
$ p.signif    <chr> "****"
```

```
# Cross-reference with similar plot for counts
MDE_age_gender_counts <-
  counts %>%
  filter(data_type == "Major_Depressive_Episode",
         subgroup != "Race/Ethnicity",
         Demographic != "TOTAL") %>%
  ggplot(aes(x = Year, y = Number, group = Demographic)) +
    geom_rect(xmin = 2011, xmax = Inf,
              ymin = -Inf, ymax = Inf,
              fill = "light gray") +
    geom_line(aes(color = Demographic), size =1) +
    scale_x_continuous(breaks = seq(2004, 2018, by=1),
                       labels = seq(2004, 2018, by=1),
                       limits = c(2004, 2018)) +
    labs(title = "Major Depressive Episode\namong Persons Aged 12 to 17",
         subtitle = "By Demographic Characteristics, Percentages, 2004-2018") +
    facet_wrap(~ subgroup) +
    ocs_theme()

MDE_age_gender_counts <- direct.label(
  MDE_age_gender_counts,
  list(dl.trans(y = y +0.38, x = x -0.2),
       "far.from.others.borders",
       cex = .8,
       fontface = "bold",
       dl.move("Age: 14-15", x = 2008, y =10))
  ) +
  scale_color_manual(values = c(age_col_light,
                                age_col,
                                age_col_dark,
                                Female_col,
                                Male_col))

MDE_age_gender_counts
```




```
# Analysis of Severe MDE
chi_squared_11.3a <- counts %>%
  filter(data_type == "Severe_Major_Depressive_Episode",
         Year %in% c(2006, 2018),
         Demographic %in% c("Male","Female")) %>%
  mutate(Number = Number * 1000) %>%
  select(-data_type, -subgroup) %>%
  pivot_wider(names_from = Year,
              names_prefix = "Year",
              values_from = Number) %>%
  column_to_rownames("Demographic")

chi_squared_11.3a
```

```
       Year2006 Year2018
Male     335000   628000
Female  1023000  1795000
```

```
# Running Chi-Squared test for 11.3a
chisq.test(chi_squared_11.3a)
```

```
    Pearson's Chi-squared test with Yates' continuity correction

data:  chi_squared_11.3a
X-squared = 715.87, df = 1, p-value < 2.2e-16
```

```
t(chi_squared_11.3a) %>%
  prop_test( detailed = TRUE, correct = TRUE) %>%
  glimpse()
```

```
Rows: 1
Columns: 13
$ n           <dbl> 3781000
$ n1          <dbl> 1358000
$ n2          <dbl> 2423000
$ estimate1   <dbl> 0.2466863
$ estimate2   <dbl> 0.2591828
$ statistic   <dbl> 715.8654
$ p           <dbl> 1.06e-157
$ df          <dbl> 1
$ conf.low    <dbl> -0.01340819
$ conf.high   <dbl> -0.01158486
$ method      <chr> "Prop test"
$ alternative <chr> "two.sided"
$ p.signif    <chr> "****"
```

Question Opportunity:

Try performing the same wrangling as we did for the percentage of each demographic that reported having a major depressive episode for the data about treatment.

```
# Setting up data for 11.4a

chi_squared_11.4a <- counts %>%
  filter(data_type == "Treatment",
         Year %in% c(2004, 2018),
         Demographic %in% c("Male","Female")) %>%
  mutate(Number = Number * 1000) %>%
  select(-data_type, -subgroup) %>%
  pivot_wider(names_from = Year,
              names_prefix = "Year",
```

```
                values_from = Number) %>%
    column_to_rownames("Demographic")

  chi_squared_11.4a
```

```
       Year2004 Year2018
Male     239000   351000
Female   656000  1081000
```

After this, we need to run the chi-squared test on this data for 11.4a.

```
  # Chi-Squared for 11.4a
  chisq.test(chi_squared_11.4a)
```

```
    Pearson's Chi-squared test with Yates' continuity correction

data:  chi_squared_11.4a
X-squared = 1399.1, df = 1, p-value < 2.2e-16
```

```
  t(chi_squared_11.4a) %>%
    prop_test(detailed = TRUE, correct = TRUE) %>%
    glimpse()
```

```
Rows: 1
Columns: 13
$ n           <dbl> 2327000
$ n1          <dbl> 895000
$ n2          <dbl> 1432000
$ estimate1   <dbl> 0.2670391
$ estimate2   <dbl> 0.2451117
$ statistic   <dbl> 1399.097
$ p           <dbl> 3.3e-306
$ df          <dbl> 1
$ conf.low    <dbl> 0.02077041
$ conf.high   <dbl> 0.02308434
$ method      <chr> "Prop test"
$ alternative <chr> "two.sided"
$ p.signif    <chr> "****"
```

```
# While not part of the case study, we wanted to save this new graph, too.
save(MDE_age_gender_counts, file = here("plots", "MDE_age_gender_counts.rda"))
png(here("plots", "MDE_age_gender_counts.png"))
MDE_age_gender_counts
while (!is.null(dev.list()))  dev.off()
```

Summary

```
# Summary

# Make a plot that summarizes main findings
title_plots <-
  ggdraw() +
  draw_label(
    "Self-Reported Depression Among American Youths",
    fontface = 'bold',
    size = 18,
    x = 0,
    hjust = -0.01
  )
```

Question Opportunity

What happens if we modify the hjust value?

Answer: It changes the justification of the text horizontally

Continuing the Summary

```
# Demonstrating creating a subtitle
forward <- ggdraw() +
  draw_label(
    "The percentage of youths (age 12-17) experiencing major depressive episodes (MDE) has
    size = 16,
    x = 0,
    hjust = -0.01
  )

# Load the previous plots
load(file = here("plots", "MDE_total.rda"))
```

```r
load(file = here("plots", "MDE_age_gender.rda"))
load(file = here("plots", "MDE_race.rda"))

# Using our MDE_total plot:
MDE_total_for_mp <- MDE_total +
  theme(plot.title = element_blank(),
        plot.subtitle = element_blank(),
        axis.text = element_text(color = "black"))

# Using our Treat_total plot:
treat_for_mp <-
  Treat_total +
  theme(plot.title = element_blank(),
        plot.subtitle = element_blank(),
        axis.text = element_text(color = "black"))

# Using our MDE_age_gender plot:
MDE_age_gender_for_mp <-
  MDE_age_gender +
  theme(plot.title= element_text(size = 14, color = "black"),
        plot.subtitle = element_blank(),
        axis.text = element_text(color = "black")) +
  labs(title = "Older youths and females report MDE at the highest rates\nand show the ste

# Last plot, but moving items and changing size to appear more pleasantly
MDE_race_for_mp_leg <- MDE_race +
  theme(plot.title= element_text(size = 14, color = "black"),
        plot.subtitle = element_blank(),
        axis.text = element_text(color = "black"),
        legend.position = "right",
        legend.title = element_blank(),
        legend.text = element_text(size = 14)) +
  labs(title = "All racial/ethnic groups show similar\nincreases since 2011") +
  guides(color = guide_legend(ncol = 2))

legend <- get_legend(MDE_race_for_mp_leg +
          theme(legend.justification = "right"))

#Now we will remove the legend for this plot
```

Question Opportunity

Do you remember how to do this?

Answer: Yes, in theme I can set the legend.position to equal "none"

Continuing the Summary

```
MDE_race_for_mp <- MDE_race_for_mp_leg +
  theme(legend.position = "none")

# Begin putting the plots together
row_1 <- plot_grid(MDE_total_for_mp,
                   treat_for_mp,
                   nrow = 1)

row_2 <- plot_grid(MDE_age_gender_for_mp,
                   MDE_race_for_mp,
                   nrow = 1,
                   rel_widths = c(1, 0.6))

#Turn them all into a pdf
png(filename = here("img", "mainplot_orig.png"),
         res = 300, width = 10, height = 10, units = "in")
plot_grid(title_plots,
          forward,
          row_1,
          row_2,
          legend,
          ncol = 1,
          rel_heights = c(0.1,0.2,.8, 1, 0.3))
while (!is.null(dev.list()))  dev.off()
```

Question Opportunity

Try to come up with the code for these plots on your own before you reveal it. We can use our ocs_theme() for these plots to make all the plots look similar.

```
# Answer to the Question
MDE_gender <- percents %>%
  filter(data_type == "Major_Depressive_Episode",
          subgroup == "Gender") %>%
  mutate(subgroup = recode(subgroup, "Gender" =
         "Percent of MDEs being reported from each Gender")) %>%
```

```
  ggplot(aes(x = Year, y = Percent, group = Demographic)) +
  geom_rect(xmin = 2011, xmax = Inf,
            ymin = -Inf, ymax = Inf,
            fill = "light gray") +
  geom_line(aes(color = Demographic), size = 1.5) +
  facet_wrap(~subgroup) +
  scale_y_continuous(limits = c(0, 23))+
  scale_x_continuous(breaks = seq(2004, 2018, by = 1),
                     labels = seq(2004, 2018, by = 1),
                     limits = c(2004, 2018)) +
 ocs_theme() +
 scale_color_manual(values = c(Female_col,
                               Male_col))

MDE_gender <- direct.label(
  MDE_gender,
  list(dl.trans(y = y +0.38, x = x -0.2),
       "far.from.others.borders",
       cex = 1,
       fontface=c("bold"))
  )

MDE_age <- percents %>%
  filter(data_type == "Major_Depressive_Episode",
         subgroup == "Age") %>%
  mutate(subgroup = recode(subgroup, "Age" =
       "Percent of MDEs being reported by each Age Group")) %>%
  ggplot(aes(x = Year, y = Percent, group = Demographic)) +
  geom_rect(xmin = 2011, xmax = Inf,
            ymin = -Inf, ymax = Inf,
            fill = "light gray") +
  geom_line(aes(color = Demographic), size = 1.5) +
  facet_wrap(~subgroup)+
  scale_y_continuous(limits = c(0, 23))+
  scale_x_continuous(breaks = seq(2004, 2018, by = 1),
                     labels = seq(2004, 2018, by = 1),
                     limits = c(2004, 2018)) +
  ocs_theme() +
  scale_color_manual(values = c(age_col_light,
                                age_col,
                                age_col_dark))
```

```r
MDE_age <- direct.label(
  MDE_age,
  list(dl.trans(y = y + 0.38, x = x -0.2),
       "far.from.others.borders",
       cex = 1,
       fontface=c("bold"),
       dl.move("Age: 12-13", x = 2015, y = 9))
  )

# More subtitles for gender and race plots
label <- expression(paste(
    bold("Older "),
    "youths and ",
    bold("females "),
    "report MDE at the highest rate and also show the steepest increase."), sep = "")

forward2 <- ggdraw() +
  draw_label(label,
    size = 16,
    x = 0,
    hjust = -0.01
  )

row_2 <- plot_grid(MDE_age, MDE_gender,
                   nrow = 1)

png(filename = here("img", "mainplot.png"),
         res = 300, width = 10, height = 10, units = "in")
plot_grid(title_plots,
          forward,
          row_1,
          forward2,
          row_2,
          ncol = 1,
          rel_heights = c(0.1, 0.2, 1, 0.1, 1))
while (!is.null(dev.list()))  dev.off()
```

Homework Questions

> Ask students to scrape Tables 11.5A and 11.5B from the website, which contain data about the receipt of treatment among youths who reported having a severe episode. Ask students to create plots and perform Chi-square tests to evaluate how groups compare over time.

We believe the best way to go about these Homework Questions is to run Tables 11.5A and 11.5B through the previous steps where applicable. To do this, we have split this into subsections for each of the previous stages.

Data Import

```
# Scraper
table11.5a <- scraper(XPATH = '/html/body/div[4]/div[9]/table')
table11.5b <- scraper(XPATH = '/html/body/div[4]/div[10]/table')

# Saving the data locally
save(table11.5a, table11.5b,
     file = here("data", "imported", "imported_data.rda"))
```

Data Wrangling

```
# Apply demographic transformations
table11.5a <- data_dem_settings(TABLE = table11.5a)
table11.5b <- data_dem_settings(TABLE = table11.5b)
table11.5a %<>% mutate(data_type = "Treatment_for_Severe_Major_Depressive_Episode")
table11.5b %<>% mutate(data_type = "Treatment_for_Severe_Major_Depressive_Episode")

tables_to_check_hw <- list(table11.2a,
                           table11.2b,
                           table11.3a,
                           table11.3b,
                           table11.4a,
                           table11.4b,
                           table11.5a,
                           table11.5b)
tables_to_check_hw %>% map(data_dem_check)
```

```
Warning in stri_detect_regex(string, pattern, negate = negate, opts_regex =
opts(pattern)): argument is not an atomic vector; coercing

Warning in stri_detect_regex(string, pattern, negate = negate, opts_regex =
opts(pattern)): argument is not an atomic vector; coercing

Warning in stri_detect_regex(string, pattern, negate = negate, opts_regex =
opts(pattern)): argument is not an atomic vector; coercing

Warning in stri_detect_regex(string, pattern, negate = negate, opts_regex =
opts(pattern)): argument is not an atomic vector; coercing

Warning in stri_detect_regex(string, pattern, negate = negate, opts_regex =
opts(pattern)): argument is not an atomic vector; coercing

Warning in stri_detect_regex(string, pattern, negate = negate, opts_regex =
opts(pattern)): argument is not an atomic vector; coercing

Warning in stri_detect_regex(string, pattern, negate = negate, opts_regex =
opts(pattern)): argument is not an atomic vector; coercing

Warning in stri_detect_regex(string, pattern, negate = negate, opts_regex =
opts(pattern)): argument is not an atomic vector; coercing

Warning in stri_detect_regex(string, pattern, negate = negate, opts_regex =
opts(pattern)): argument is not an atomic vector; coercing

Warning in stri_detect_regex(string, pattern, negate = negate, opts_regex =
opts(pattern)): argument is not an atomic vector; coercing

Warning in stri_detect_regex(string, pattern, negate = negate, opts_regex =
opts(pattern)): argument is not an atomic vector; coercing

Warning in stri_detect_regex(string, pattern, negate = negate, opts_regex =
opts(pattern)): argument is not an atomic vector; coercing

Warning in stri_detect_regex(string, pattern, negate = negate, opts_regex =
opts(pattern)): argument is not an atomic vector; coercing

Warning in stri_detect_regex(string, pattern, negate = negate, opts_regex =
opts(pattern)): argument is not an atomic vector; coercing

Warning in stri_detect_regex(string, pattern, negate = negate, opts_regex =
```

```
opts(pattern)): argument is not an atomic vector; coercing

Warning in stri_detect_regex(string, pattern, negate = negate, opts_regex =
opts(pattern)): argument is not an atomic vector; coercing

[[1]]
[1] "Data looks good!"

[[2]]
[1] "Data looks good!"

[[3]]
[1] "Data looks good!"

[[4]]
[1] "Data looks good!"

[[5]]
[1] "Data looks good!"

[[6]]
[1] "Data looks good!"

[[7]]
[1] "Data looks good!"

[[8]]
[1] "Data looks good!"
```

```r
# Recreate counts and percents with new tables
counts <- bind_rows(table11.2a, table11.3a, table11.4a, table11.5a)
percents <- bind_rows(table11.2b, table11.3b, table11.4b, table11.5b)
# Altering counts and percents
## This is copied from above, as the same transformations need to apply

counts %>% distinct(data_type)
```

```
# A tibble: 4 x 1
  data_type
  <chr>
1 Major_Depressive_Episode
```

```
2 Severe_Major_Depressive_Episode
3 Treatment
4 Treatment_for_Severe_Major_Depressive_Episode
```

```
percents %>% distinct(data_type)
```

```
# A tibble: 4 x 1
  data_type
  <chr>
1 Major_Depressive_Episode
2 Severe_Major_Depressive_Episode
3 Treatment
4 Treatment_for_Severe_Major_Depressive_Episode
```

```
counts %<>% pivot_longer(cols = contains("20"),
                         names_to = "Year",
                         values_to = "Number") %>%
            mutate(Year = as.numeric(Year))

percents %<>% pivot_longer(cols = contains("20"),
                           names_to = "Year",
                           values_to = "Percent") %>%
              mutate(Year = as.numeric(Year))

glimpse(counts)
```

```
Rows: 720
Columns: 5
$ Demographic <chr> "TOTAL", "TOTAL", "TOTAL", "TOTAL", "TOTAL", "TOTAL", "TOT~
$ subgroup    <chr> "Total", "Total", "Total", "Total", "Total", "Total", "Tot~
$ data_type   <chr> "Major_Depressive_Episode", "Major_Depressive_Episode", "M~
$ Year        <dbl> 2004, 2005, 2006, 2007, 2008, 2009, 2010, 2011, 2012, 2013~
$ Number      <dbl> 2225, 2191, 1970, 2016, 2027, 1954, 1911, 2011, 2213, 2587~
```

```
glimpse(percents)
```

```
Rows: 720
Columns: 5
$ Demographic <chr> "TOTAL", "TOTAL", "TOTAL", "TOTAL", "TOTAL", "TOTAL", "TOT~
$ subgroup    <chr> "Total", "Total", "Total", "Total", "Total", "Total", "Tot~
$ data_type   <chr> "Major_Depressive_Episode", "Major_Depressive_Episode", "M~
$ Year        <dbl> 2004, 2005, 2006, 2007, 2008, 2009, 2010, 2011, 2012, 2013~
$ Percent     <dbl> 9.0, 8.8, 7.9, 8.2, 8.3, 8.1, 8.0, 8.2, 9.1, 10.7, 11.4, 1~
```

```r
percents %<>% mutate(Demographic = str_replace(string = Demographic,
                                               pattern = "AIAN",
                                           replacement = "American Indian and Alaska Native"

percents %<>% mutate(Demographic =  str_replace(string = Demographic,
                                               pattern = "NHOPI",
                                           replacement = "Native Hawaiian or Other Pacific I

counts %<>% mutate(Demographic = str_replace(string = Demographic,
                                               pattern = "AIAN",
                                           replacement = "American Indian and Alaska Native"

counts %<>% mutate(Demographic =  str_replace(string = Demographic,
                                               pattern = "NHOPI",
                                           replacement = "Native Hawaiian or Other Pacific I

# Saving data
save(percents, counts, table11.5a, table11.5b,
     file = here("data", "wrangled", "wrangled_data.rda"))
write_csv(percents, path = here("data", "wrangled", "percents.csv"))
write_csv(counts, path = here("data", "wrangled", "counts.csv"))
write_csv(table11.5a, path = here("data", "wrangled", "table11.5a.csv"))
write_csv(table11.5b, path = here("data", "wrangled", "table11.5b.csv"))
```

Data Visualization

```r
# Plotting

# Load Wrangled Data from Previous Section
load(file = here("data", "wrangled", "wrangled_data.rda"))

#Increase Over Time
```

```
SMDE_total <- percents %>%
  filter(data_type == "Treatment_for_Severe_Major_Depressive_Episode",
         Demographic == "TOTAL") %>%
  mutate(Demographic = recode(Demographic,
              "TOTAL" = "Percent of respondents treated with SMDE")) %>%
  ggplot(aes(x = Year, y = Percent, group = Demographic)) +
    facet_wrap( ~ Demographic)+
    geom_rect(xmin = 2009, xmax = Inf,
              ymin = -Inf, ymax = Inf,
              fill = "light gray") +
    geom_line(aes(color = Demographic), size = 1.5) +
    scale_x_continuous(breaks = seq(2006, 2018, by = 1),
                       labels = seq(2006, 2018, by = 1),
                       limits = c(2006, 2018)) +
    labs(title = "The Rate of Youths Aged 12 to 17 Treated with \n Severe Major Depressive
    theme_classic() +
    theme(axis.text.x = element_text(angle = 90),
          legend.position = "none",
          strip.background = element_rect(fill ="black"),
          strip.text = element_text(face = "bold",
                                    size = 14,
                                    color = "white")) +
    scale_color_manual(values = c("blue"))

# Saving our graph as PNG
save(SMDE_total, file = here("plots", "SMDE_total.rda"))
png(here("plots", "SMDE_total.png"))
while (!is.null(dev.list()))  dev.off()

# Comparison of Age and Gender Groups
SMDE_age_gender <-
  percents %>%
  filter(data_type == "Treatment_for_Severe_Major_Depressive_Episode",
         subgroup != "Race/Ethnicity",
         Demographic != "TOTAL") %>%
  ggplot(aes(x = Year, y = Percent, group = Demographic)) +
    geom_rect(xmin = 2009, xmax = Inf,
              ymin = -Inf, ymax = Inf,
              fill = "light gray") +
    geom_line(aes(color = Demographic), size =1) +
    scale_x_continuous(breaks = seq(2006, 2018, by=1),
```

```
                      labels = seq(2006, 2018, by=1),
                      limits = c(2006, 2018)) +
    labs(title = "Treatment for Sevre Major Depressive Episode\namong Persons Aged 12 to 1
         subtitle = "By Demographic Characteristics, Percentages, 2006-2018") +
    facet_wrap(~ subgroup) +
    ocs_theme()

SMDE_age_gender <- direct.label(
  SMDE_age_gender,
  list(dl.trans(y = y +0.38, x = x -0.2),
       "far.from.others.borders",
       cex = .8,
       fontface = "bold",
       dl.move("Age: 12-13", x = 2015, y = 36),
       dl.move("Age: 14-15", x = 2015, y = 42),
       dl.move("Age: 16-17", x = 2008, y = 48),
       dl.move("Male", x = 2012, y = 34)
    )
  ) +
  scale_color_manual(values = c(age_col_light,
                                age_col,
                                age_col_dark,
                                Female_col,
                                Male_col))

# Saving our graph as PNG
save(SMDE_age_gender, file = here("plots", "SMDE_age_gender.rda"))
png(here("plots", "SMDE_age_gender.png"))
while (!is.null(dev.list()))  dev.off()

# Comparison of Racial/Ethnic Groups
SMDE_race <- percents %>%
  filter(data_type == "Treatment_for_Severe_Major_Depressive_Episode",
         subgroup == "Race/Ethnicity") %>%
  mutate(Demographic = fct_reorder(Demographic, Percent,
                                           tail, n = 1, .desc = TRUE,
                                           .na_rm = TRUE)) %>%
  ggplot(aes(x = Year, y = Percent, group = Demographic)) +
    geom_rect(xmin = 2011, xmax = Inf,
              ymin = -Inf, ymax = Inf,
              fill = "light gray") +
```

```
    geom_line(aes(color = Demographic), size = 1) +
    facet_wrap( ~ subgroup) +
    scale_x_continuous(breaks = seq(2004, 2018, by = 1),
                       labels = seq(2004, 2018, by = 1),
                       limits = c(2004, 2018)) +
    scale_color_viridis_d() +
    labs(title = "Major Depressive Episode\namong Persons Aged 12 to 17",
         subtitle = "By Demographic Characteristics, Percentages, 2004-2018") +
    ocs_theme()

save(SMDE_race, file = here("plots", "SMDE_race.rda"))
png(here("plots", "SMDE_race.png"))
while (!is.null(dev.list()))  dev.off()

SMDE_total
```

```
Warning: Removed 2 rows containing missing values (`geom_line()`).
```

```
SMDE_age_gender
```

```
Warning: Removed 10 rows containing missing values (`geom_line()`).
```

```
Warning: Removed 10 rows containing missing values (`geom_dl()`).
```

```
SMDE_race
```

```
Warning: Removed 8 rows containing missing values (`geom_line()`).
```

Data Analysis

```
# Setting up the data for the chi-squared test
chi_squared_11.5a <- counts %>%
  filter(data_type == "Treatment_for_Severe_Major_Depressive_Episode",
         # This is 2006 because that is the lowest year in 11.5
         Year %in% c(2006, 2018),
         Demographic %in% c("Male","Female")) %>%
```

```
    mutate(Number = Number * 1000) %>%
    select(-data_type, -subgroup) %>%
    pivot_wider(names_from = Year,
                names_prefix = "Year",
                values_from = Number) %>%
    column_to_rownames("Demographic")

  chi_squared_11.5a
```

```
       Year2006 Year2018
Male     134000   274000
Female   493000   859000
```

```
  # Running chi-squared test for 11.5a
  chisq.test(chi_squared_11.5a)
```

```
    Pearson's Chi-squared test with Yates' continuity correction

data:  chi_squared_11.5a
X-squared = 1792.1, df = 1, p-value < 2.2e-16
```

```
  t(chi_squared_11.5a) %>%
    prop_test(detailed = TRUE, correct = TRUE) %>%
    glimpse()
```

```
Rows: 1
Columns: 13
$ n           <dbl> 1760000
$ n1          <dbl> 627000
$ n2          <dbl> 1133000
$ estimate1   <dbl> 0.2137161
$ estimate2   <dbl> 0.2418358
$ statistic   <dbl> 1792.079
$ p           <dbl> 0
$ df          <dbl> 1
$ conf.low    <dbl> -0.02940596
$ conf.high   <dbl> -0.0268335
```

```
$ method      <chr> "Prop test"
$ alternative <chr> "two.sided"
$ p.signif    <chr> "****"
```

Our Own Question

For our own question, we would like to examine which age group had the largest spike in data from one year to the next. Is this group the same as the group with the highest maximum value?

We can apply this question to both Major Depressive Episode data and Major Depressive Episode with Severe Impairment.

```
# Our Own Question

# Load in data
load(file = here("data", "wrangled", "wrangled_data.rda"))

# Finding summary for largest change
max_change <- function(df, subgroup, data_type, column) {
  {{df}} %>%
  # Drop NAs to get valid data, they aren't helpful
  drop_na() %>%
  # Filter by provided parameters
  filter(subgroup == {{subgroup}}) %>%
  filter(data_type == {{data_type}}) %>%
  group_by(Demographic) %>%
  # Calculate and summarize by change
  mutate(Change = {{column}} - lag({{column}}, default = 0)) %>%
  summarize(max_change = max(Change))
}

# Apply our function
change_MDE <- max_change(counts, "Age", "Major_Depressive_Episode", Number)
change_MDES <- max_change(counts, "Age", "Severe_Major_Depressive_Episode", Number)
change_TMDE <- max_change(counts, "Age", "Treatment", Number)
change_TMDES <- max_change(counts, "Age", "Treatment_for_Severe_Major_Depressive_Episode",

glimpse(change_MDE)
```

```
Rows: 3
```

```
Columns: 2
$ Demographic <chr> "Age: 12-13", "Age: 14-15", "Age: 16-17"
$ max_change  <dbl> 445, 783, 997
```

```
glimpse(change_MDES)
```

```
Rows: 3
Columns: 2
$ Demographic <chr> "Age: 12-13", "Age: 14-15", "Age: 16-17"
$ max_change  <dbl> 211, 518, 629
```

```
glimpse(change_TMDE)
```

```
Rows: 3
Columns: 2
$ Demographic <chr> "Age: 12-13", "Age: 14-15", "Age: 16-17"
$ max_change  <dbl> 169, 278, 448
```

```
glimpse(change_TMDES)
```

```
Rows: 3
Columns: 2
$ Demographic <chr> "Age: 12-13", "Age: 14-15", "Age: 16-17"
$ max_change  <dbl> 92, 232, 304
```

From the provided data, we can see that the maximum change for each of the four data types is always associated with the age range of 17 to 18 year olds. This pattern holds true for all four data types.

Now we want to ask, how does this relate to which age groups are known to have the maximum value for each of our four categories? To do this, we must simply find the maximum value for each of our four data types. The steps here are very similar to that of the last part, but we need to modify our function to pull the new information we want.

```
# Finding summary for largest change
max_val <- function(df, subgroup, data_type, column) {
  {{df}} %>%
```

```
  # Drop NAs to get valid data, they aren't helpful
  drop_na() %>%
  # Filter by provided parameters
  filter(subgroup == {{subgroup}}) %>%
  filter(data_type == {{data_type}}) %>%
  group_by(Demographic) %>%
  # Calculate and summarize by maximum
  summarize(max_val = max({{column}}))
}

# Apply our function
val_MDE <- max_val(counts, "Age", "Major_Depressive_Episode", Number)
val_MDES <- max_val(counts, "Age", "Severe_Major_Depressive_Episode", Number)
val_TMDE <- max_val(counts, "Age", "Treatment", Number)
val_TMDES <- max_val(counts, "Age", "Treatment_for_Severe_Major_Depressive_Episode", Numbe

glimpse(val_MDE)
```

```
Rows: 3
Columns: 2
$ Demographic <chr> "Age: 12-13", "Age: 14-15", "Age: 16-17"
$ max_val     <dbl> 632, 1250, 1600
```

```
glimpse(val_MDES)
```

```
Rows: 3
Columns: 2
$ Demographic <chr> "Age: 12-13", "Age: 14-15", "Age: 16-17"
$ max_val     <dbl> 388, 891, 1150
```

```
glimpse(val_TMDE)
```

```
Rows: 3
Columns: 2
$ Demographic <chr> "Age: 12-13", "Age: 14-15", "Age: 16-17"
$ max_val     <dbl> 251, 514, 684
```

```
glimpse(val_TMDES)
```

```
Rows: 3
Columns: 2
$ Demographic <chr> "Age: 12-13", "Age: 14-15", "Age: 16-17"
$ max_val     <dbl> 177, 406, 558
```

The highest count is always for 17-18 year olds. This is not a surprise, as looking at the graphs from the Data Visualization section will give the same type of information in graph form. There, we can see that the 17-18 year old group tends to trend higher than the other age groups in general. As for relating to the changing values, we can see that these definitely seem to be related, as all 4 of the groups for maximum change are the same as they are for maximum value. The lowest of the age groups seems to be the same, as well.

We calculated this data using the `counts` data, which may be causing this unfair disparity in the data. How would this change if we used `percents` instead? Perhaps that would be a more fair representation of the changes in the data? Since we already have the functions defined, checking this will be simple.

```
# Note: We had to go back and modify the functions to be able to reuse them
# Reusing the functions was not as easy since we hard coded the column names
# Now, they are more useful, but it almost would have been easier to define new ones

# Apply our function
pct_MDE <- max_change(percents, "Age", "Major_Depressive_Episode", Percent)
pct_MDES <- max_change(percents, "Age", "Severe_Major_Depressive_Episode", Percent)
pct_TMDE <- max_change(percents, "Age", "Treatment", Percent)
pct_TMDES <- max_change(percents, "Age", "Treatment_for_Severe_Major_Depressive Episode",
```

```
Warning: There was 1 warning in `summarize()`.
i In argument: `max_change = max(Change)`.
Caused by warning in `max()`:
! no non-missing arguments to max; returning -Inf
```

```
# Apply our function
pct_MDE %<>%
  full_join(max_val(percents, "Age", "Major_Depressive_Episode", Percent),
            by = "Demographic")
pct_MDES %<>%
```

```
  full_join(max_val(percents, "Age", "Severe_Major_Depressive_Episode", Percent),
            by = "Demographic")
pct_TMDE %<>%
  full_join(max_val(percents, "Age", "Treatment", Percent),
            by = "Demographic")
pct_TMDES %<>%
  full_join(max_val(percents, "Age", "Treatment_for_Severe_Major_Depressive Episode", Perc
          by = "Demographic")
```

```
Warning: There was 1 warning in `summarize()`.
i In argument: `max_val = max(Percent)`.
Caused by warning in `max()`:
! no non-missing arguments to max; returning -Inf
```

```
glimpse(pct_MDE)
```

```
Rows: 3
Columns: 3
$ Demographic <chr> "Age: 12-13", "Age: 14-15", "Age: 16-17"
$ max_change  <dbl> 5.4, 9.2, 12.3
$ max_val     <dbl> 8.4, 15.3, 19.0
```

```
glimpse(pct_MDES)
```

```
Rows: 3
Columns: 3
$ Demographic <chr> "Age: 12-13", "Age: 14-15", "Age: 16-17"
$ max_change  <dbl> 2.7, 6.0, 7.5
$ max_val     <dbl> 5.1, 10.9, 13.7
```

```
glimpse(pct_TMDE)
```

```
Rows: 3
Columns: 3
$ Demographic <chr> "Age: 12-13", "Age: 14-15", "Age: 16-17"
$ max_change  <dbl> 38.2, 35.5, 45.0
$ max_val     <dbl> 41.5, 41.3, 45.8
```

```
glimpse(pct_TMDES)
```

```
Rows: 0
Columns: 3
$ Demographic <chr>
$ max_change  <dbl>
$ max_val     <dbl>
```

Surprisingly, this does not change the results for the maximum values for neither the change nor the strictly maximum value. However, there is an alteration in the patterns here that we can see easily. While the maximums may not have changed, the minimums have. The minimum values for each is no longer limited to the age group of 12 - 13 year olds. Here, we can see that a change has been made, and using `percents` does actually give us a better representation of the data compared to `counts`. It just so happens that both datasets produced the same age groups for maximum change and maximum value.