Module 1 Day 2

```
library()
library()
```

Recap last week

Set your working directory

```
setwd()
```

Read in the data file "acs_2016_age.csv"

Create a new data frame filtered to ages between 25-55

```
acs_2016_age_filtered <- filter()
```

Make a line plot with AGE on the x axis and INCTOT on the y axis. Label your chart appropriately

```
ggplot() +
  geom_line(aes(x = , y = )) +
  labs(x = "",
       y = "",
       title = "Average Total Income by Age (25-55)",
       caption = "Source: 2016 ACS")
```

Add a variable to the original data frame that is the gap between total income and wage income

```
acs_2016_age_mutated <- mutate(  , inc_gap = )
```

Make a line plot with AGE on the x axis and inc_gap on the y axis, Label your chart appropriatly.

rename()

Start by reading in "acs_2016_sample.csv"

```
acs_2016_sample <- read_csv()
```

Using the glimpse(), head(), or str() function look at the variable names

Create a new data frame called acs_2016_cleaned and change the names:

- STATEFIP = state_code
- SEX = sex
- AGE = age
- RACE = race
- HISPAN = hispanic
- EDUC = education
- EMPSTAT = employment_status
- INCTOT = total_income
- INCWAGE = wage_income
- UHRSWORK = hrs_worked
- WKSWORK2 = weeks_worked

select()

From acs_2016_cleaned select:

- state_code
- sex
- age
- race
- hispanic
- education
- employment_status
- total_income
- wage_income
- hrs_worked
- weeks_worked

filter()

Filter to observations where:

- total_income <= 1,000,000
- wage_income <= 1,000,000
- age >= 25 & age <= 55
- hrs_worked > 0

We can begin to get an understanding of the variables in out data frame using the summary() function.

```
summary(acs_2016_cleaned)
```

mutate(), ifelse()

Using mutate() and ifelse() recode the sex variable so that

- 1 -> "Male"
- 2 -> "Female"

Mean Income by Sex

A simple question: What is the average income for men?

- Create a new data frame that only contains observations for men
- select the column total_income
- find the mean() and median()

Mean Income by Sex

```
male_data <- filter(acs_2016_cleaned, sex == ) # only rows that are male

male_data <- select(male_data, ) # we are interested in a single column: Total Income

mean(male_data$  , na.rm = T) # Our summary statistic, ignore NA values
median(male_data$  , na.rm = T)
```

What does it mean that our average income and our median income are so different? Why is the average income so much higher?

%>%

This is not an efficient way to write this code. To write more efficient code we can use nested functions f(g(x))

```
nested_data <- select(filter(acs_2016_cleaned, sex == "Male"), total_income)

mean(nested_data$total_income, na.rm = TRUE)
```

This is still not great. Luckily we have another solution, the pipe operator: %>%

```
pipe_data <-
  acs_2016_cleaned %>%
  filter(sex == "Male") %>%
  select(total_income)

mean(nested_data$total_income, na.rm = T)
```

```
seq(25, 30, 1)

25 %>% seq(30, 1) # The same as seq(25, 30, 1)

30 %>% seq(25, ., 1) # The same as seq(25, 30, 1))
```

- Using the %>% operator calculate the mean income for women

summarise()

Use the summarise() function to calculate the mean and median income for the male_data dataframe. Don't forget about (na.rm = T)

```
male_data %>%
  summarise(mean_inc = mean(total_income, na.rm = T),
            median_inc = median(total_income, na.rm = T))
```

Now do the same calculation but for the female_data data frame.

group_by(), summarise()

```
acs_2016_cleaned %>%
  group_by() %>%
  summarise(mean_inc = mean(total_income, na.rm = T),
            median_inc = median(total_income, na.rm = T))
```

- Now let's go back and clean up our data for race and education so that we can calculate some more interesting summary stats.

mutate(), ifelse(), case_when()

Nested ifelse statements get super messy very quickly! Better to use case_when:

```r
acs_2016_cleaned <-
  acs_2016_cleaned %>%
  mutate(education = case_when(education <= 5 ~ "No HS Diploma",
                               education == 6 ~ "HS Diploma",
                               education %in% c(7,8,9) ~ "Some College",
                               education == 10 ~ "College Degree",
                               education == 11 ~ "Graduate Degree"))
```

Let's use the same function to recode the race variable:

RACE Race [general version]

- 1 White
- 2 Black/African American/Negro
- 3 American Indian or Alaska Native
- 4 Chinese
- 5 Japanese
- 6 Other Asian or Pacific Islander
- 7 Other race, nec
- 8 Two major races
- 9 Three or more major races

HISPAN Hispanic origin [general version]

- 0 Not Hispanic
- 1 Mexican
- 2 Puerto Rican
- 3 Cuban
- 4 Other
- 9 Not Reported

I propose the following categories:

- race == 1 & hispanic == 0 ~ "Non-hispanic White"
- race == 1 & hispanic %in% c(1,2,3,4) ~ "Hispanic White"
- race == 2 ~ "Black"
- race == 3 ~ "Native American"
- race %in% c(4,5,6) ~ "Asian"
- race %in% c(7,8,9) ~ "Other/Multiracial"

Code up these categories using case_when

group_by(), summarise()

Now that we have our data all cleaned up that calculate some summary data and plot it:

- Create a data frame called wage_age that contains the median wage income by age and sex
- What grouping variables should you use?
- Make a line plot of the data with age on the x axis, median_wage on the y axis and sex on the color axis
- Be sure to label your chart appropriately

Now that we have our data all cleaned up that calculate some summary data and plot it:

- Create a data frame called wage_education that contains the median wage income by education level and sex
- What grouping variables should you use?

We want to make a bar plot of this data

geom_col()

```
gds_plot <- wage_eduction %>%
    ggplot(aes(x = education, y = median_income, fill = sex)) +
    geom_col(position = "dodge")

gds_plot
```

What are some things we will want to change about this chart?

scale_x_discrete()

```
gds_plot_x <-
  gds_plot +
  scale_x_discrete(labels = c("College Degree" = "College \nDegree",
                              "Graduate Degree" = "Graduate \nDegree",
                              "HS Diploma" = "HS Diploma",
                              "No HS Diploma" = "No \nHS Diploma",
                              "Some College" = "Some \nCollege"),
                   limits = c("No HS Diploma", "HS Diploma", "Some College",
                              "College Degree","Graduate Degree"))


gds_plot_x
```

scale_y_continous()

Now I want to change the y scale to go in $10,000 increments instaed of $25,000 increments. How?

- Using the scale_y_continous() function
- The argument we want to use is breaks which we want to set to a sequence

```
gds_plot_y <- gds_plot_x +
  scale_y_continuous(breaks = seq()) +
  labs()
```

- Please also add appropriate axis labels and a title

Recap Exersise

Make the same plot but instead of using sex as the color variable use race. You will need to:

- Calculate the median wage per hour for each racial group at each education level using mutate(), group_by() and summarise()
- Assume a person's hours is how much they work per week
- Use geom_col() to create a bar chart
- Use scale_x_discrete() and scale_y_continous() to fix up the axes
- Be sure to include lable and a title

Recap Exersise