Module 1 Day 3

```
# install.packages("plyr")
library(plyr)
library(dplyr)
library(ggplot2)
# install.packages("stargazer")
library(stargazer)
# install.packages("Hmisc")
library(Hmisc)
# install.packages("lubridate")
library(lubridate)
```

Recap last week

Set your working directory

```
setwd()
```

- Read in the data file "acs_2016_sample.csv"
- Calculate the average total income by race, sex, and age
- Make a line plot with age on the x axis and total_income on the y axis
- Use race as the color axis and linetype for sex
- Label your chart appropriatly

```
acs_2016_cleaned <-
  read.csv() %>%
  rename(state_code = STATEFIP,
         sex = SEX,
         age = AGE,
         race = RACE,
         hispanic = HISPAN,
         education = EDUC,
         employment_status = EMPSTAT,
         total_income = INCTOT,
         wage_income = INCWAGE,
         hrs_worked = UHRSWORK,
         weeks_worked = WKSWORK2,
         weight = PERWT,
         metro = METRO) %>%
  select(state_code, sex, age, race, hispanic, education,
         employment_status, total_income, wage_income,
         hrs_worked, weeks_worked, weight, metro) %>%
  filter(total_income <= 1000000,
         wage_income <= 1000000,
         age >= 25,
         age <= 55,
         hrs_worked > 0) %>%
  mutate(sex = ifelse(sex == 1, "Male", "Female"),
         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"),
         race = case_when(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"))
```

```
acs_2016_cleaned %>%
  group_by() %>%
  summarise(median_income = ) %>%
  ggplot(aes(x = , y = , color = , linetype = )) +
  geom_line() +
  labs(x = "Age",
       y = "Median Wage",
       linetype = "Sex",
       color = "Race",
       title = "Median Wage by Age, Sex, and Race",
       caption = "Source: 2016 ACS")
```

facet_wrap()

```
acs_2016_cleaned %>%
  group_by() %>%
  summarise(median_income = ) %>%
  ggplot(aes(x = age,
             y = median_income,
             color = sex,
             linetype = sex)) +
  geom_line() +
  facet_wrap("race") +
  labs(x = "Age",
       y = "Median Wage",
       linetype = "Sex",
       color = "Sex",
       title = "Median Wage by Age, Sex, and Race",
       caption = "Source: 2016 ACS")
```

Why do you think certain lines are noisier than others?

Create a new column in acs_2016_cleaned called rounded age that is the indiviudal age rounded to the nearest 5. (Hint check out the function round_any()). Calculate the mean wage income by rounded_age, sex, and education level. Update the x axis so that it goes from 25-55 in 5 year increments Update the y axis so that is goes from 0-175,000 in 25,0000 increments

```
acs_2016_cleaned %>%
  mutate(rounded_age = round_any()) %>%
  group_by() %>%
  summarise(median_wage = mean()) %>%
  ggplot(aes(x = , y = , color = )) +
  geom_line() +
  facet_wrap("education", nrow = 3) +
  labs(x = "Age",
       y = "Average Wage Income",
       color = "Sex") +
  scale_x_continuous(breaks = seq()) +
  scale_y_continuous(breaks = seq())
```

Let's calculate the unweighted and weighted mean income by gender in our acs_2016_cleaned data frame

```
mean_table <-
acs_2016_cleaned %>%
  group_by() %>%
  summarise(unweighted_mean = mean(, na.rm = T),
            weighted_mean = wtd.mean(, , na.rm = T))

mean_table
```

stargazer()

It is very easy to use! It does most of the work for you

```
stargazer(acs_2016_cleaned %>%
          select(age, total_income,
                 wage_income, hrs_worked,
                 weeks_worked))
```

If you just run the code it looks like garbage (unless you are fluent in Latex). To get it to compile nicely you need to add a chunk option: `results = 'asis'`

```
stargazer(acs_2016_cleaned %>%
          select(age, total_income,
                 wage_income, hrs_worked,
                 weeks_worked))
```

We can make a table of a data frame by telling stargazer not to make it a summary table. We can turn off the annoying header using `header = FALSE`. We can also set the number of digits to use `digits = 2`

```
stargazer(mean_table, summary = FALSE, header = FALSE)
```

We can fix up the column names and show a less rediculous number of digits

```
mean_table <-
acs_2016_cleaned %>%
  group_by(sex) %>%
  summarise(unweighted_mean =
              round(mean(total_income,
                         na.rm = T),0),
            weighted_mean =
              round(wtd.mean(total_income, weight,
                             na.rm = T),0))

names(mean_table) <- c("Sex",
                       "Unweighted Mean",
                       "Weighted Mean")
```

```
stargazer(mean_table, summary = FALSE, header = FALSE,
          rownames = FALSE,
          title = "Mean Income by Sex")
```

Please make a summary statistics table for the weighted median income by gender

- The weight median function is `weighted.median()`
- Use stargazer to make it beautiful
- Give it an appropriate title

Let's make a quick chart of the distribution of income by gender and then make if fabulous

- What geom should you use? what are the aesthetics it needs?
- To start make a density chart of total income by gender.
- Be sure to title and label your chart appropriatly

We can use the scale_color_manual function to change the colors.

```
inc_sex_plot +
  scale_color_manual(values = c("dodger blue", "orange"))+
  scale_fill_manual(values = c("dodger blue", "orange"))
```

Make a fancy chart of the distribution of hourly wage by education level * label your chart appropriatly * use a non base theme and make things colorful

unique()

How many states are represented in our data?

- We can find out using the unique() function
- But what does it do?

```
unique(c("A", "B", "A", "B", "C", "C", "D"))
```

- How could we use unique() and length() to find out how many states are represented?

unique()

```
length(unique(acs_2016_cleaned$state))
```

METRO

I want to know the average total income by race for people in metro areas and outside of metro areas (rural)

- First filter to people we know for sure are not in a metro area or in the center or outskirsts of a metro area.
- 0 Not identifiable
- 4 Central / Principal city status unknown
- Then recode the metro area
- 1 ~ "Rural"
- 2 ~ "Central / Principal City"
- 3 ~ "Outside Central / Principal City"

```
acs_2016_cleaned <-
  acs_2016_cleaned %>%
  filter() %>%
  mutate(metro = case_when(metro ==  ~ "Rural",
                           metro ==  ~ "Central / Principal City",
                           metro ==  ~ "Outside Central / Principal City"))
```

- Now calculate the average total income by race and metro status
- Be sure to take the weighted mean
- Make a faceted bar chart with race on the x axis and mean income on the y axis
- Use metro as the faceting variable
- Be sure to pick a theme and use scale_x_ to keep the labels from overlapping

Dates

- Dates are very special in R
- R undstands dates of the form:

```
as.Date("2012-08-30")
```

- We can do standard arithmetic on the dates:

```
as.Date("2012-08-30") - 3
```

- What happens if we subtract two dates from each other?

Convert the following to date objects

```
"Jan 1, 2018"
"31/1/18"
```

We can also make sequences of dates

```
seq(as.Date("2018-01-01"),
    as.Date("2018-04-01"),
    by = "months")
```

```
seq(as.Date("2018-01-01"),
    as.Date("2018-01-05"),
    by = "2 days")
```

Let's look at time series of real median household income:

- Read in the file fred_median_income.csv
- What columns are in the dataframe?

```
fred_median_income <-
  read.csv("",
           stringsAsFactors = F)
```

- Convert the date column to a date object
- Make a line chart with the date on the x axis and median_hh_income on the y axis
- Be sure to label your chart correctly
- Use the scale_x_dates() function to label every 5 years

```
fred_median_income %>%
  mutate(date = as.Date(, format(""))) %>%
  ggplot(aes(x = , y = )) +
  geom_line() +
  labs(x = "Date",
       y = "Median Household Income",
       title = "Real Median Household Income over Time",
       caption = "Source: FRED") +
  scale_x_date(date_breaks = "", date_labels = "") +
  theme_minimal()
```

- How has houshold wealth recovered since the great recession?