Module 1

Day 3

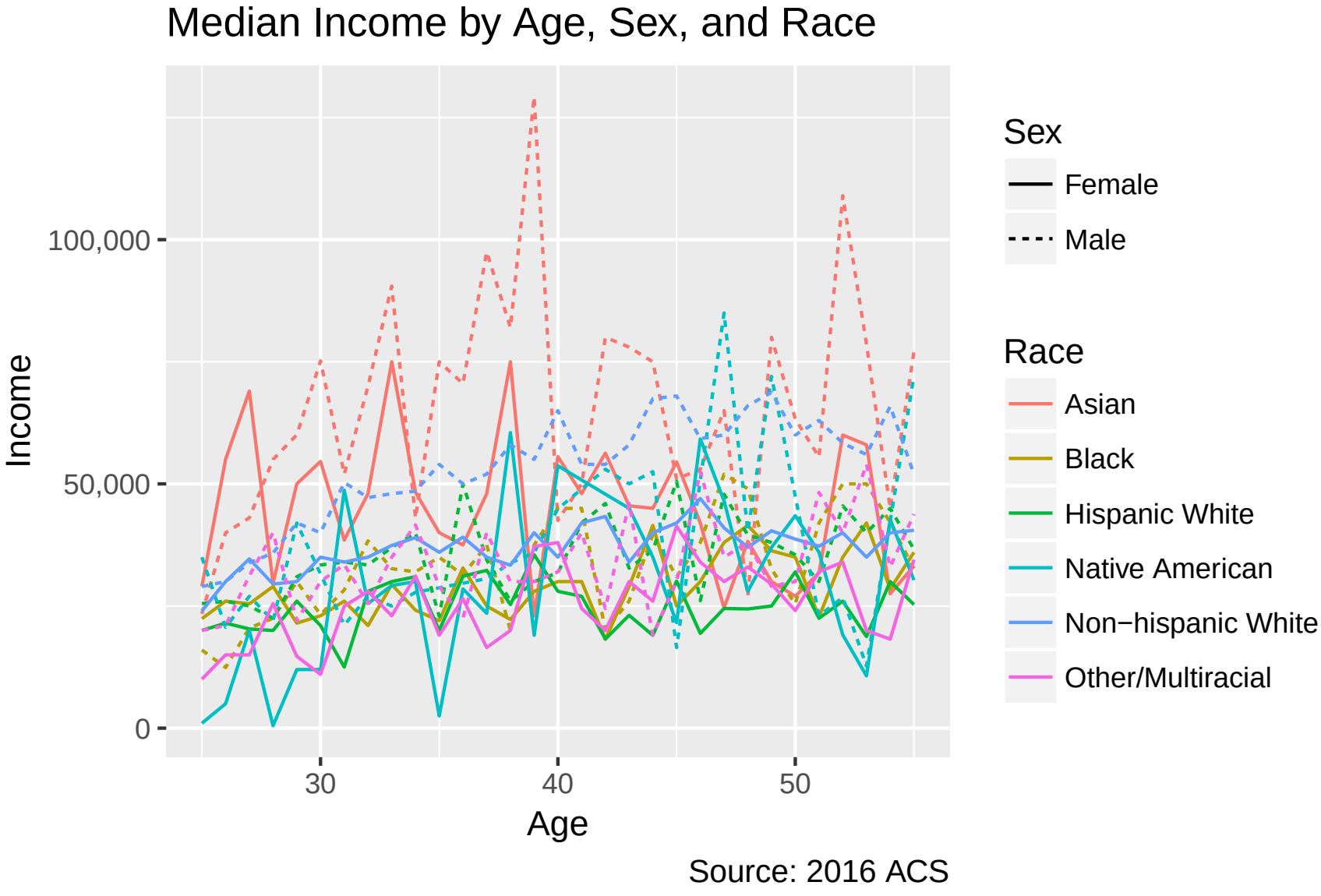
Recap last week

- `summary()`,`mean()`, `median()`, `na.rm = T`
- `seq()`
- `group_by()`, `summarise()`, `%>%`
- `case_when()`
- `ggplot()`, `geom_col()`, `geom_line()`, `geom_point()`, `geom_density()`
- `aes()`, `scale_x_discrete()`, `scale_y_continous()`, `labs()`

Recap last week

- Set your working directory
- Read in the data file "acs_2016_sample.csv"
- Calculate the median 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

Recap last week

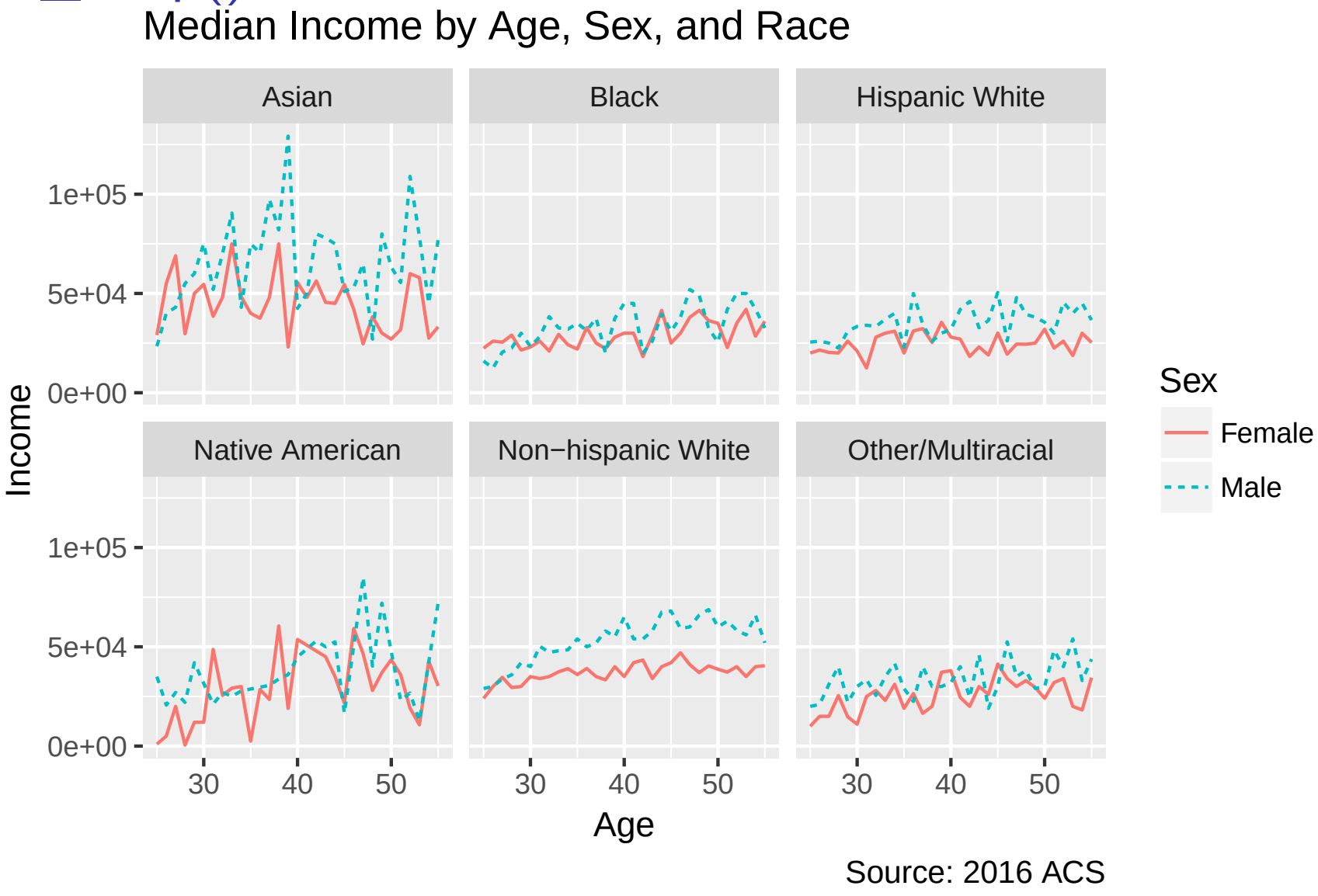


- How could we make this chart easier to read?

facet_wrap()

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

facet_wrap()

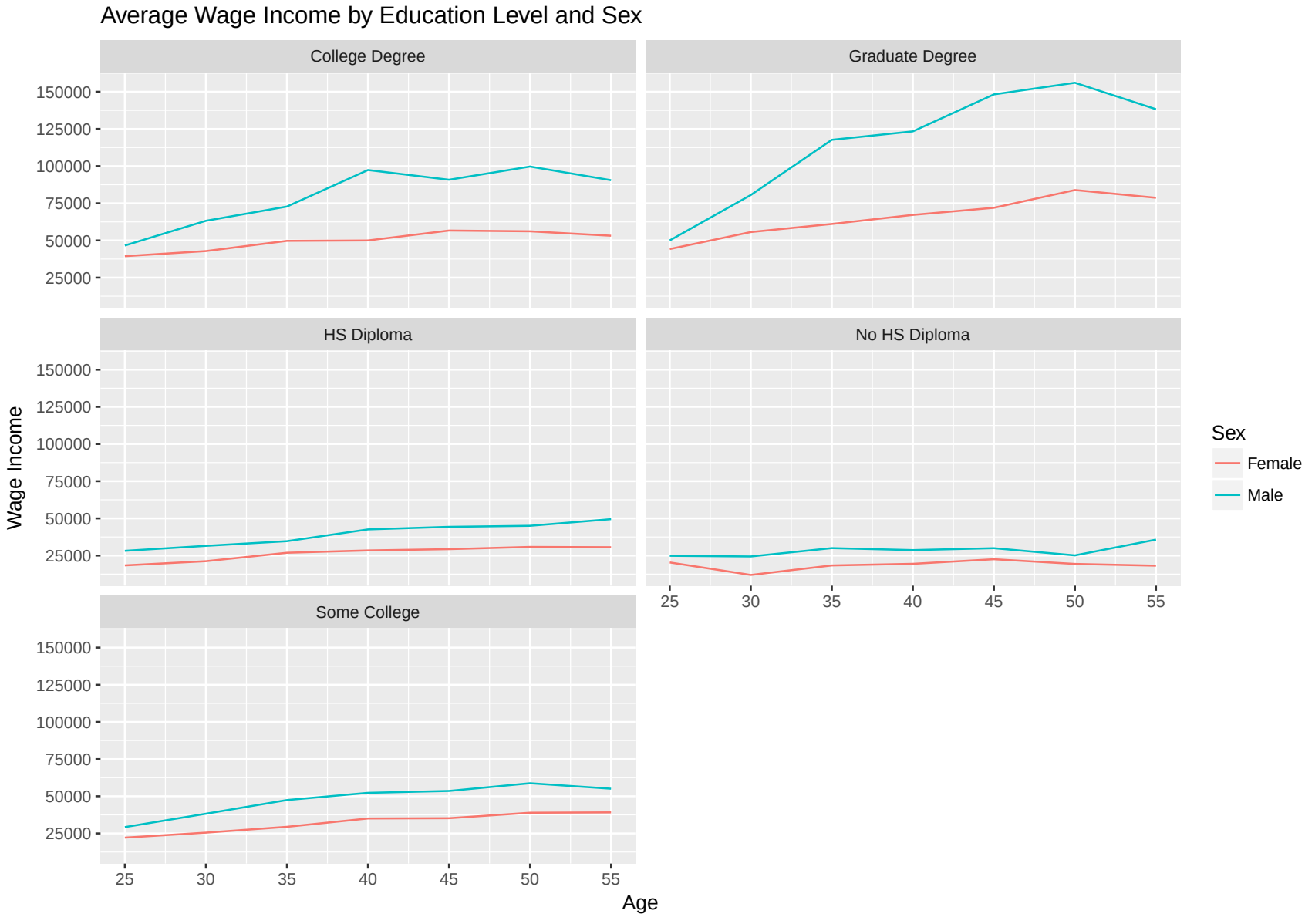


- Why do you think certain lines are noisier than others?

Recap last week

- Create a new column in acs_2016_cleaned called rounded age that is the individual 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 25,000 increments

Recap last week



Goals for Today

Economics

- Weighted Survey Data
- Weighted Summary Statistics
- Differences Between Urban and Rural Populations

Goals for Today

Programming - R

- Stargazer Package for Beautiful Tables
- Weighted Summary Statistics
- Chart Colors
- ggplot2 themes
- unique()
- Dates

Survey Data

Why do we weight survey data?

Survey Data

Why do we weight survey data? To make statistics computed from the data more representative of the population.

- Design Weight - compensate for over- or under-sampling of specific cases

Example?

- Post-Stratification or Non-response Weight - compensate for that fact that persons with certain characteristics are not as likely to respond to the survey.

Example?

Survey Data

- Weights primarily adjust means and proportions.
- May adversely affect inferential data and standard errors.
- Weights almost always increase the standard errors of your estimates.
- Introduce instability into your data.
- Very large weights (or very small ones) can also introduce instabilities (fivethirtyeight).

Calculating Weighted Summary Statistics

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

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

mean_table
```

```
## # A tibble: 2 x 3
##      sex unweighted_mean weighted_mean
##    <chr>           <dbl>         <dbl>
## 1 Female        44048.09      41896.19
## 2   Male        66555.44      63021.28
```

stargazer()

- stargazer makes it easier to create Latex, html, and text tables for data frames, summary statistics, and regressions
- 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))
```

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()

% Table created by stargazer v.5.2 by Marek Hlavac, Harvard University. E-mail: hlavac at fas.harvard.edu % Date and time: Thu, Feb 08, 2018 - 04:34:21 PM

Table 1:

Statistic	N	Mean	St. Dev.	Min	Max
age	13,161	40.466	9.099	25	55
total_income	13,161	55,783.180	65,113.930	−6,800	889,050
wage_income	13,161	51,069.900	60,103.180	0	601,000
hrs_worked	13,161	40.517	11.678	1	99
weeks_worked	13,161	5.417	1.322	1	6

stargazer()

- 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)
```

Table 2:

	sex	unweighted_mean	weighted_mean
1	Female	44048.0935069059	41896.1867103876
2	Male	66555.443310988	63021.2789973054

- What is the difference between the weighted and unweighted means?

stargazer()

- We can fix up the column names
- Show a less ridiculous 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()

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

Table 3: Mean Income by Sex

Sex	Unweighted Mean	Weighted Mean
Female	44048	41896
Male	66555	63021

stargazer()

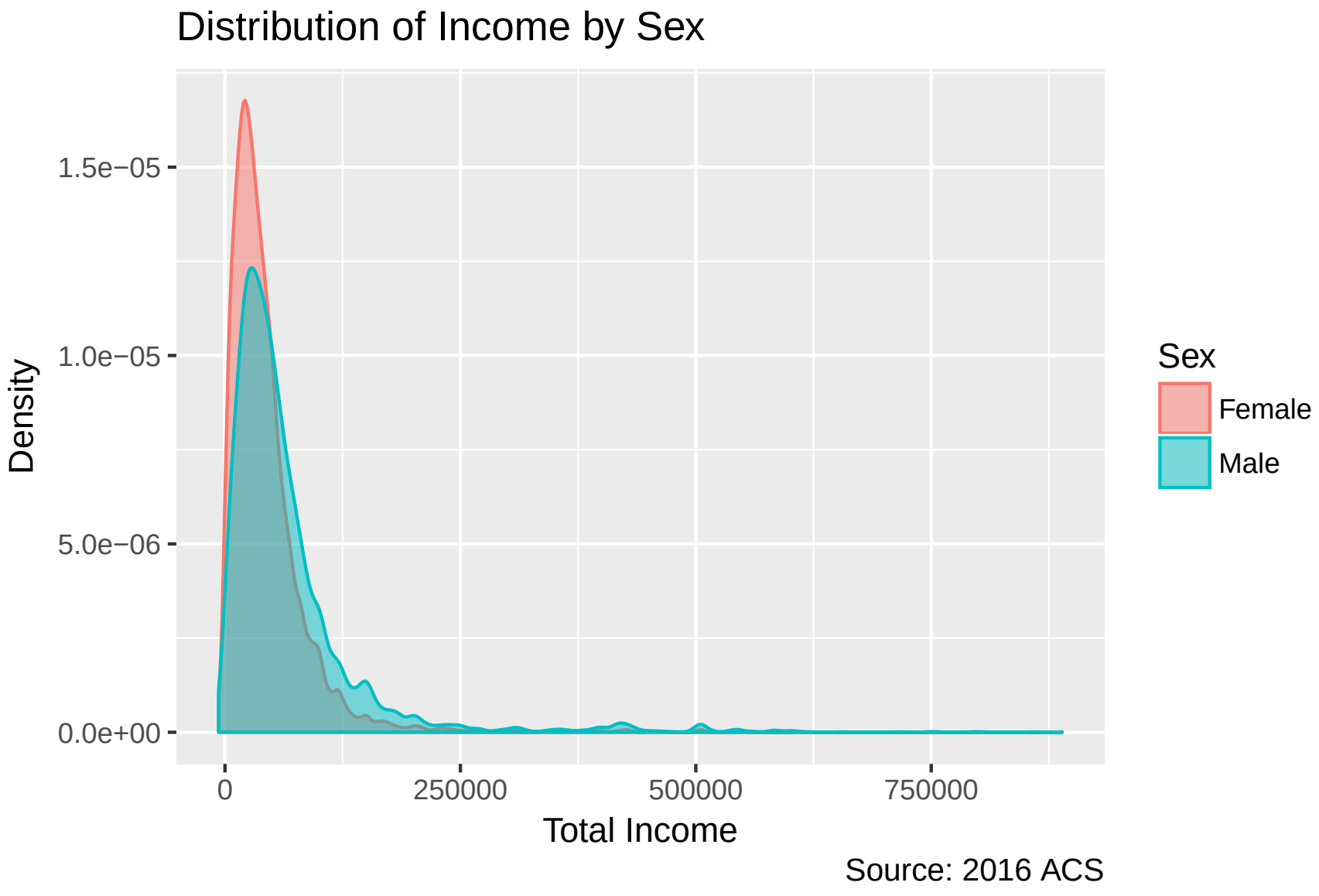
Please make a summary statistics table for the weighted median income by sex

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

Distribution of Income by Sex

- So far we have been making some pretty blah charts in terms of colors and formatting
- Let's make a quick chart of the distribution of income by sex and then make it fabulous
- What geom should you use? What are the aesthetics it needs?
- To start make a density chart of total income by sex.
- Be sure to title and label your chart appropriately

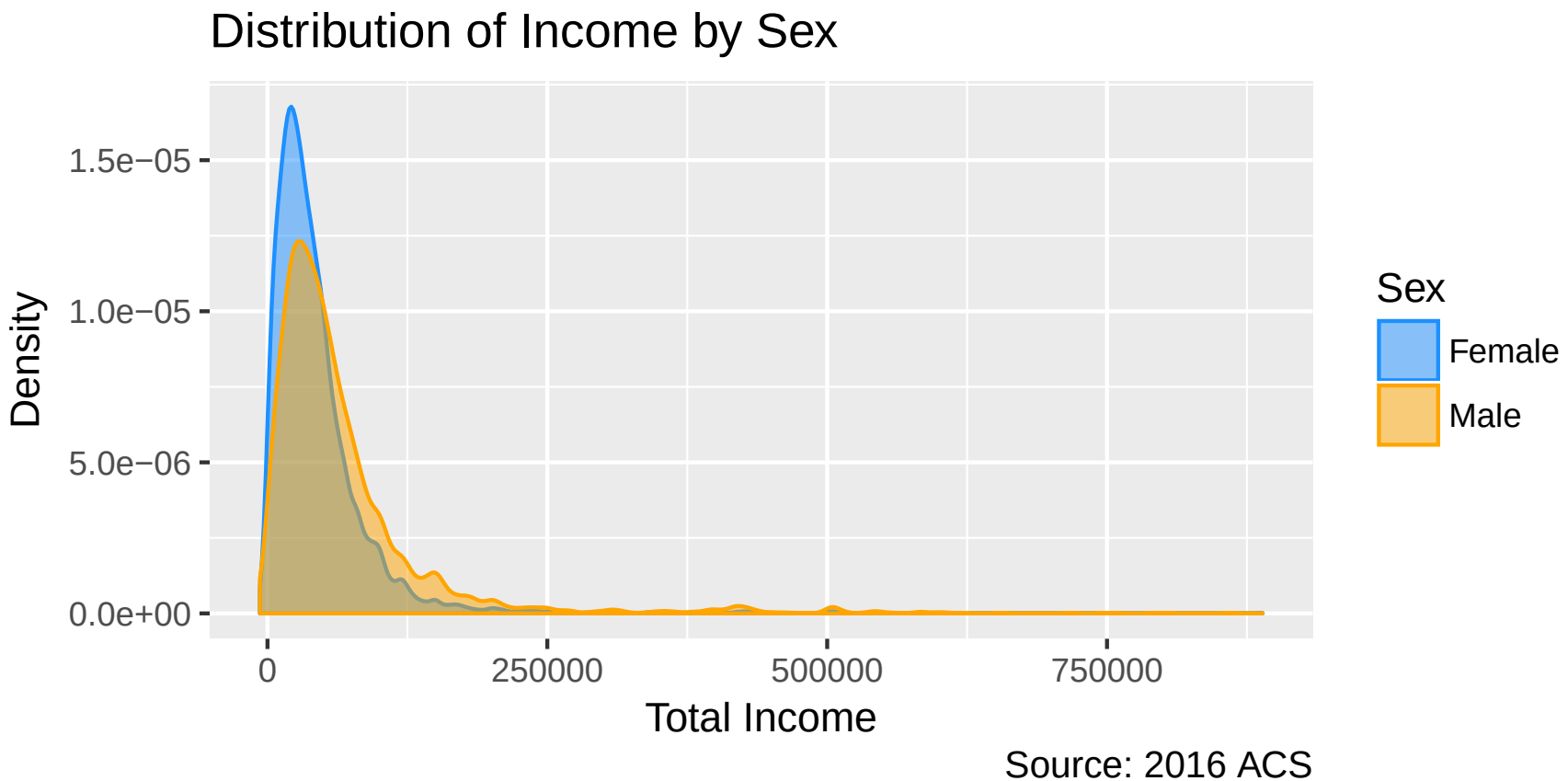
Distribution of Income by Sex



scale_color_manual()

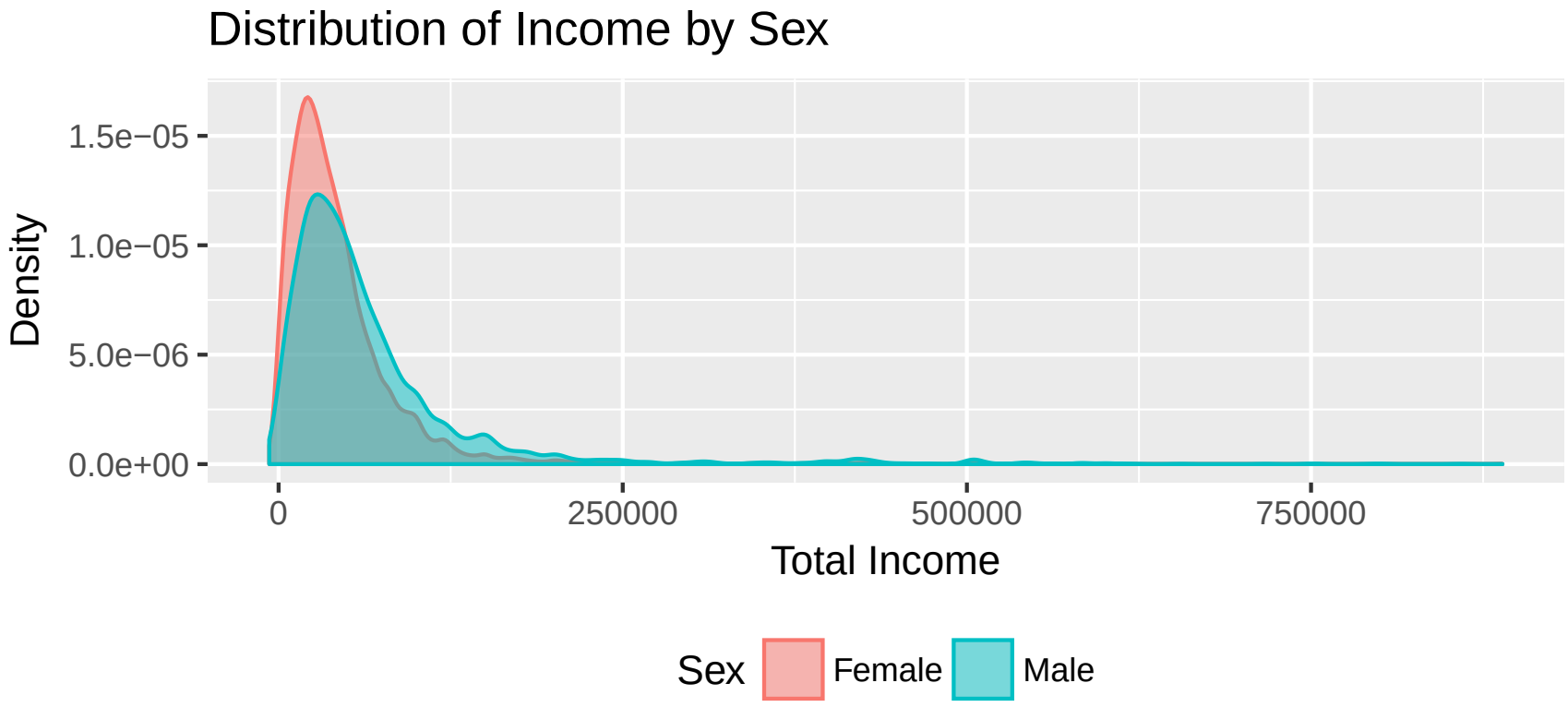
- We can use the scale_color_manual function to change the colors.

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



theme()

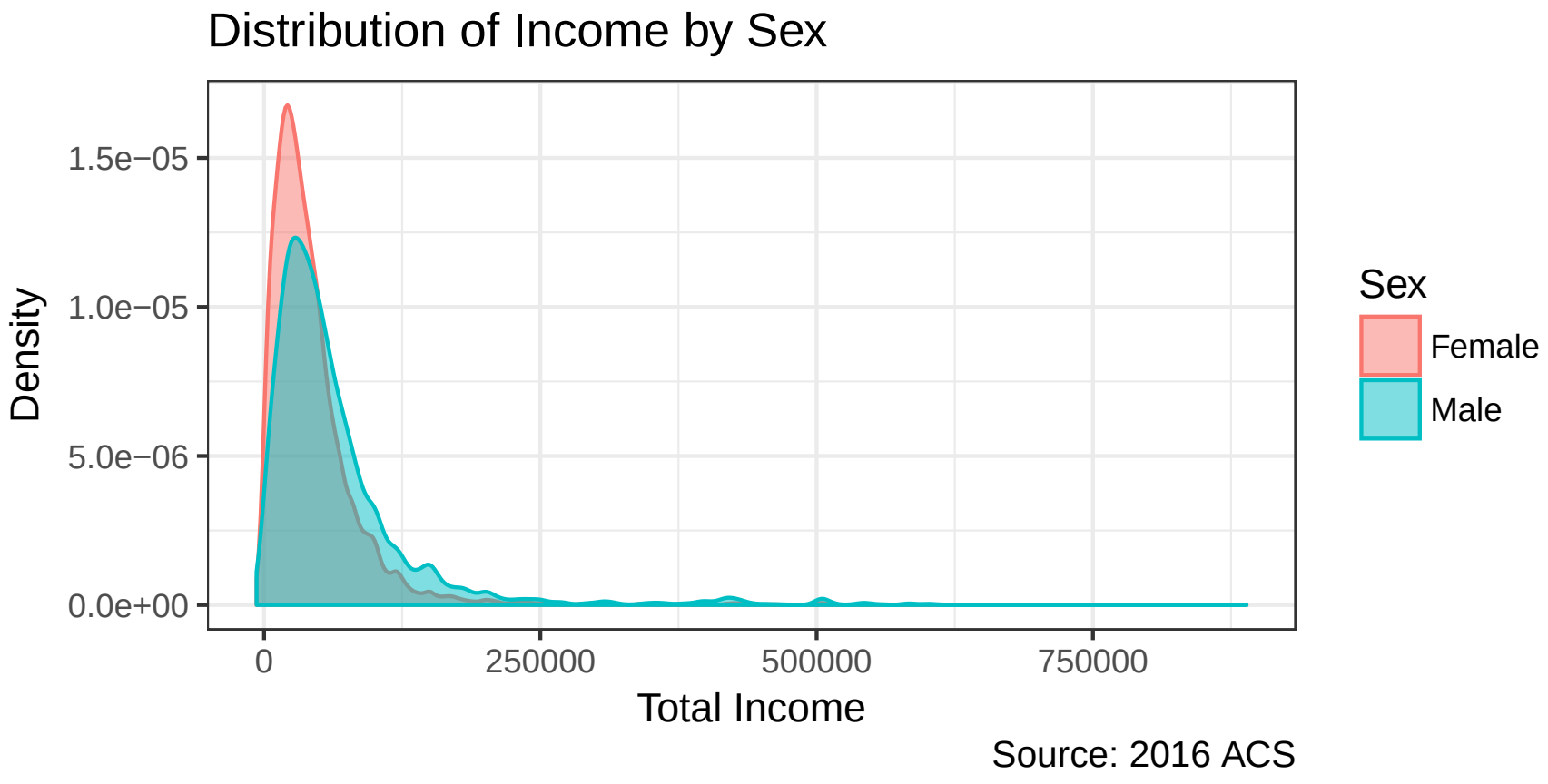
```
inc_sex_plot +
  theme(legend.position = "bottom")
```



Source: 2016 ACS

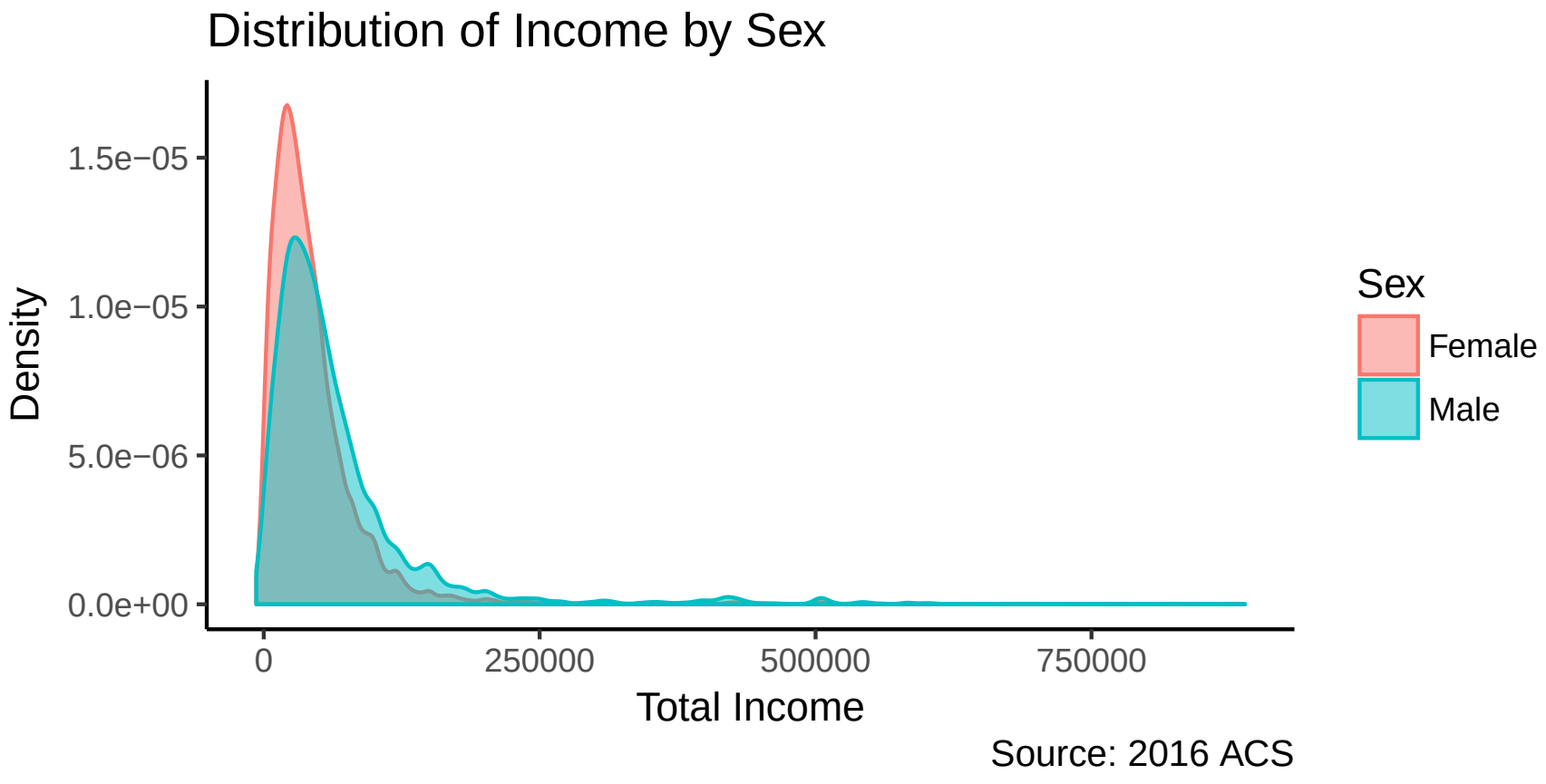
theme_bw()

```
inc_sex_plot +
  theme_bw()
```



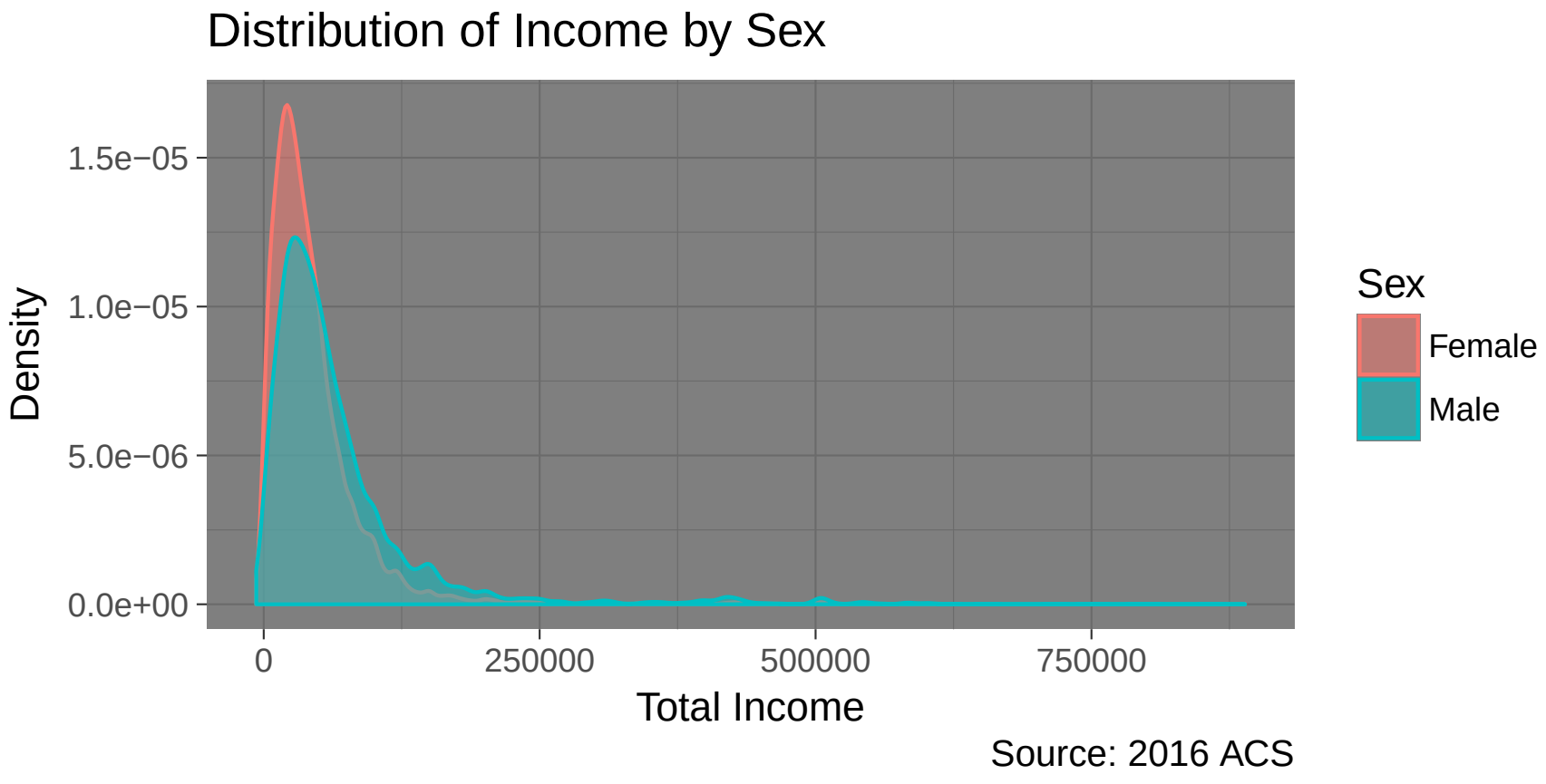
theme_classic()

```
inc_sex_plot +
  theme_classic()
```



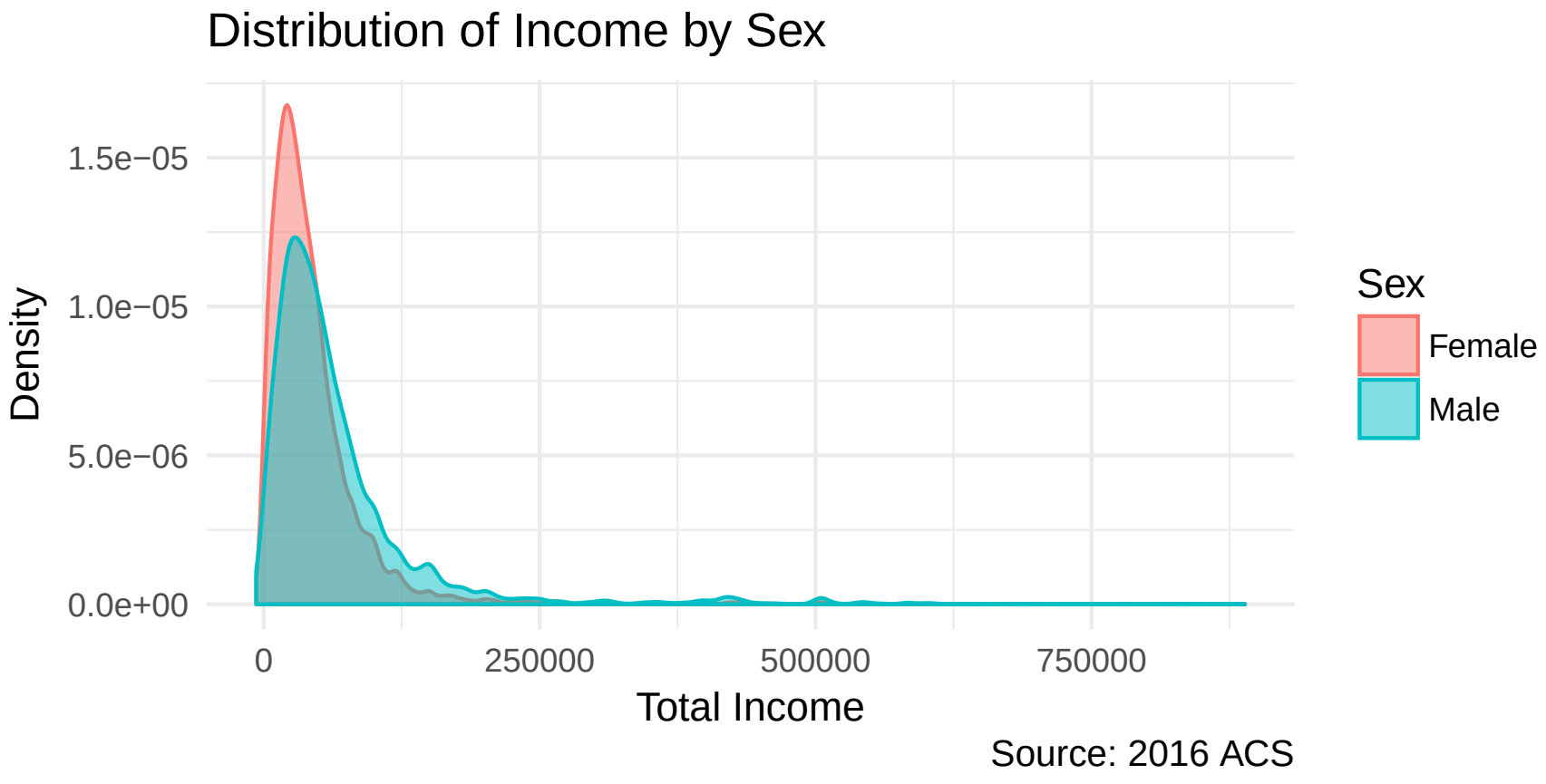
theme_dark()

```
inc_sex_plot +
  theme_dark()
```



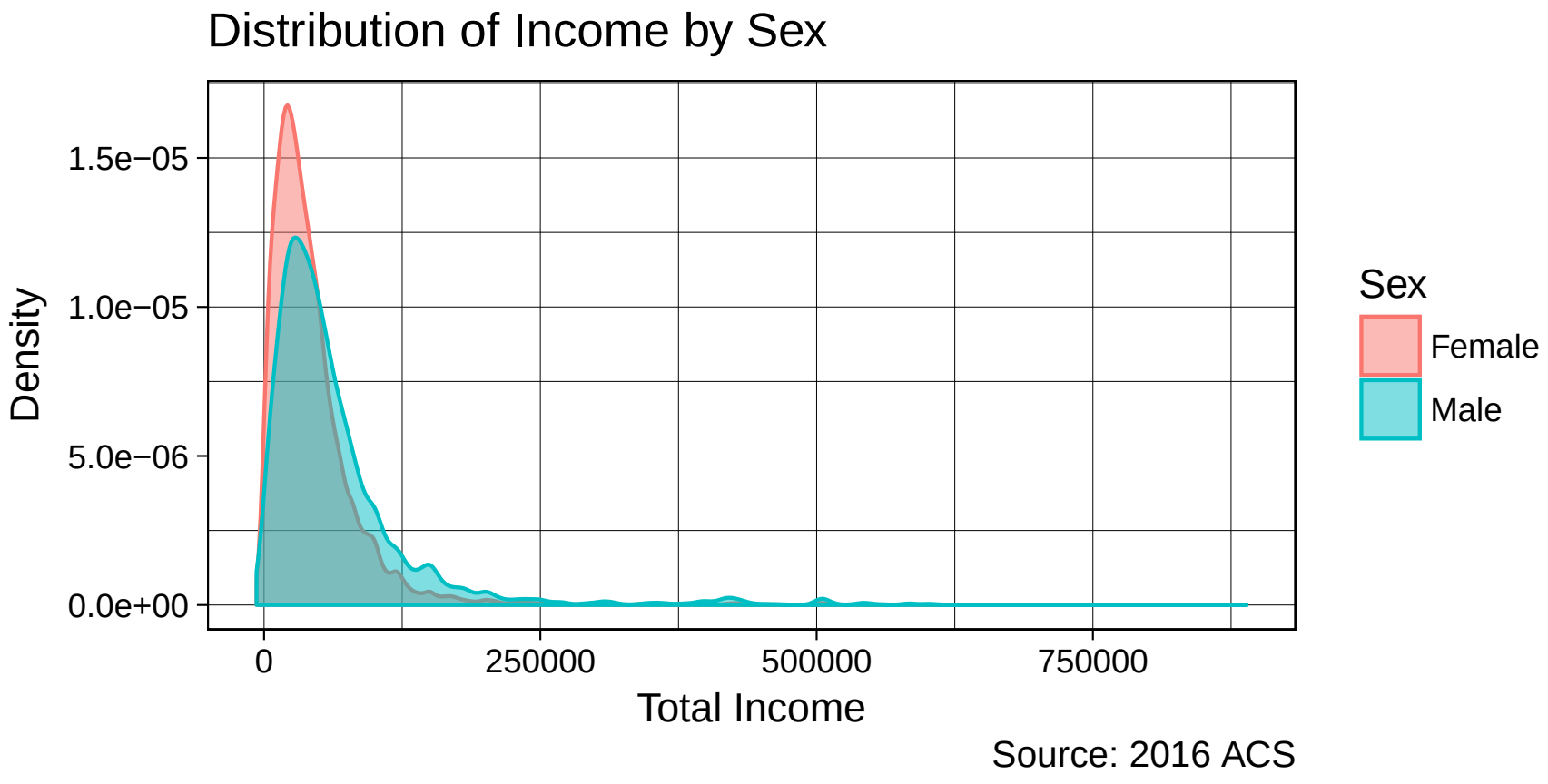
theme_minimal()

```
inc_sex_plot +
  theme_minimal()
```



theme_linedraw()

```
inc_sex_plot +
  theme_linedraw()
```



Distribution of Income by Education

- Make a fancy chart of the distribution of total income by education level
- Label your chart appropriately
- Use a non-base theme and make things colorful

Value of Tables and Charts

- When is it appropriate to show data...
  - In a chart?
  - In a table?

Urban vs. rural

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

unique()

What are the values of the metro variable?

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

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

```
## [1] "A" "B" "C" "D"
```

unique()

How many categories of metro status are there?

```
unique(acs_2016_cleaned$metro)
```

```
## [1] 4 0 1 3 2
```

- Consult the next slide to see what these mean.

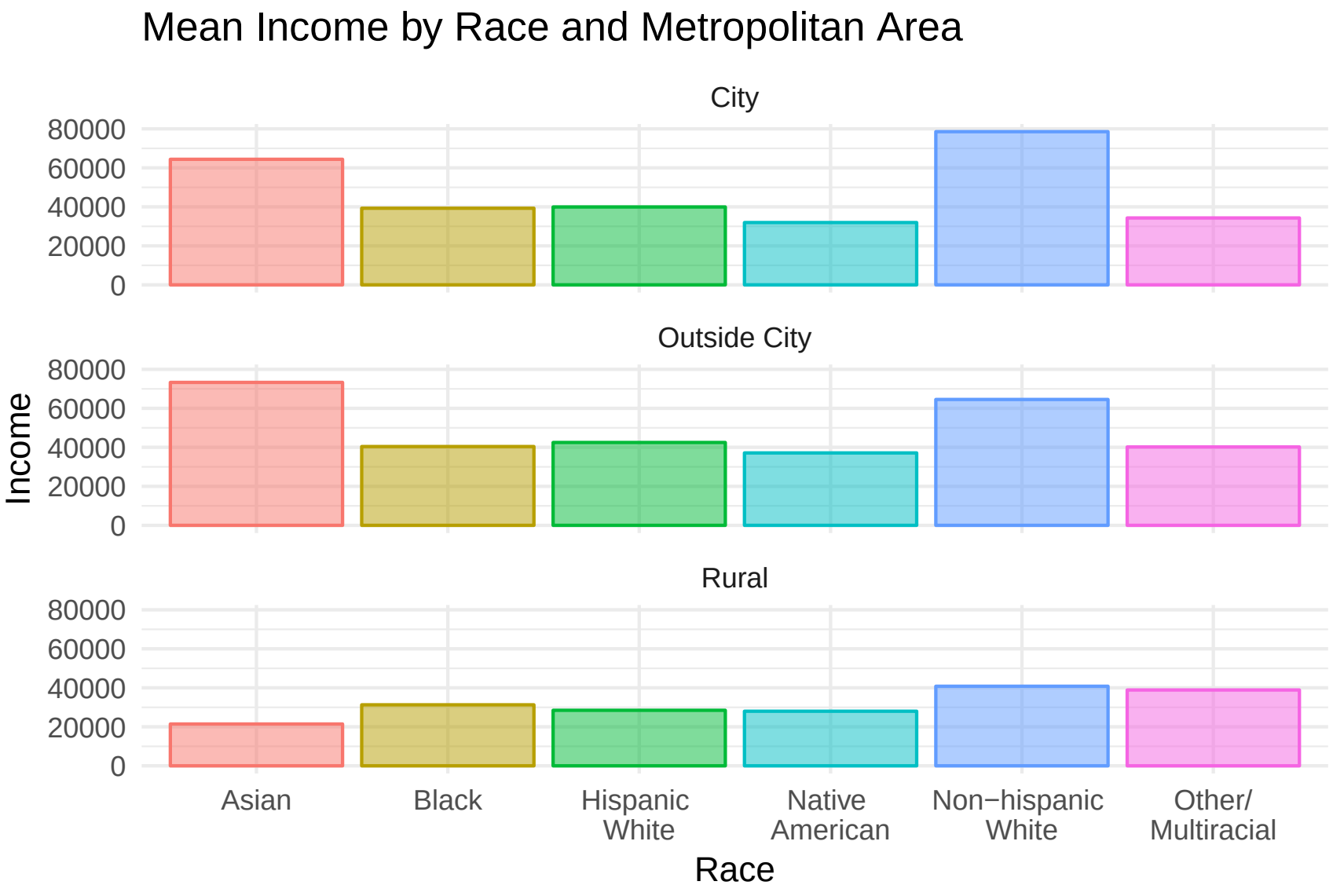
Urban vs. rural

- First filter to people we know for sure are not in a metro area or in the center or outskirts 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"

Urban vs. rural

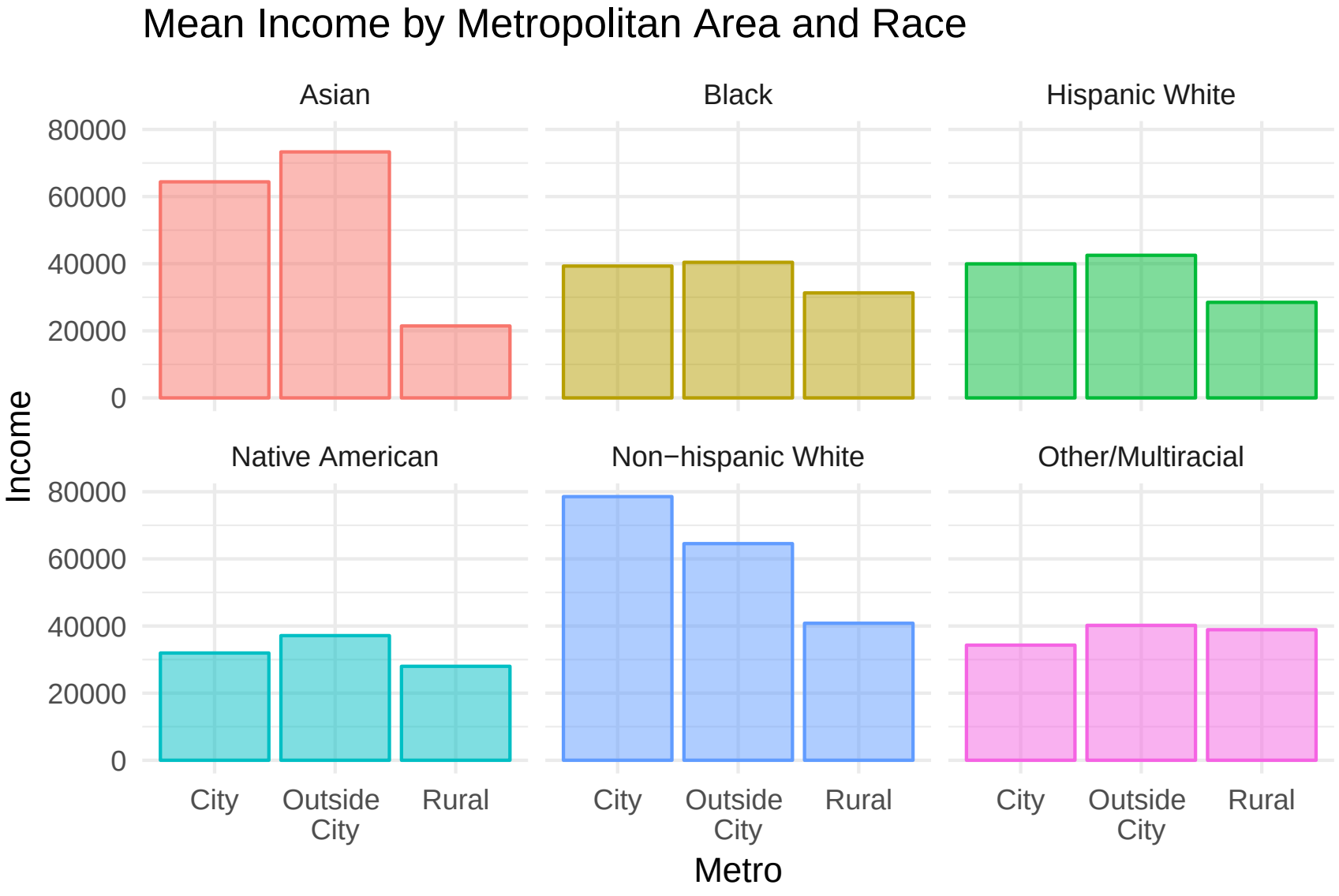
- Now calculate the average total income by race and metro status
- Be sure to take the weighted mean
- Make a faceted bar chart with race on the x axis and mean income on the y axis
- Use metro as the faceting variable
- Be sure to pick a theme and use scale_x_ to keep the labels from overlapping

Urban vs. rural



Urban vs. rural

Now what if we facet on race?

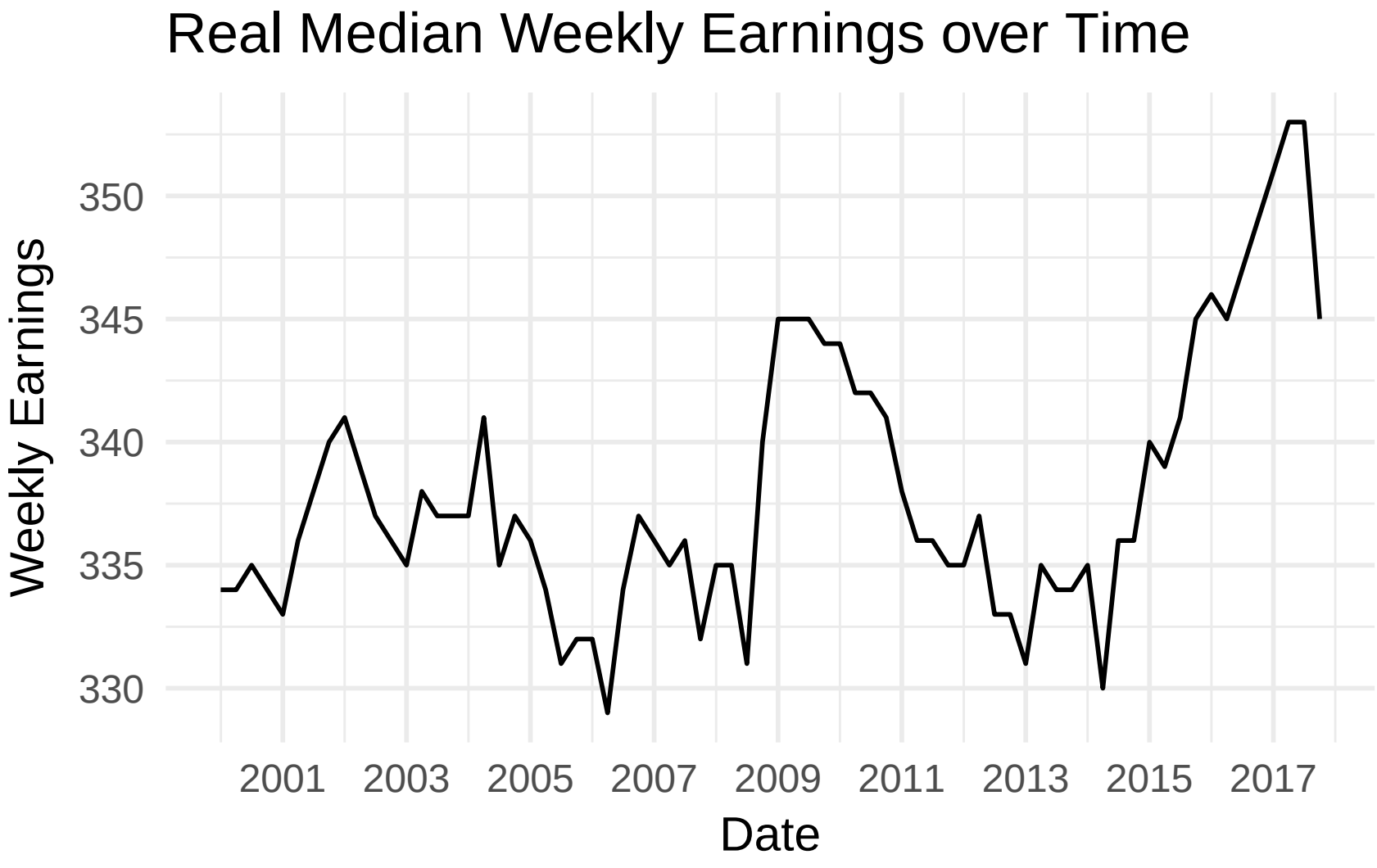


Date

- With ACS data, we're looking at a snapshot in time (2016).
- How have earnings changed over time?

Dates

We're going to learn how to make this chart.



Dates

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

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

```
## [1] "2012-08-30"
```

- We can do standard arithmetic on the dates:

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

```
## [1] "2012-08-27"
```

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

Dates

But what if our date is not in that format? What if we are importing data?

```
as.Date("08/30/2012", format = "%m/%d/%Y")
```

```
## [1] "2012-08-30"
```

```
as.Date(41149, origin = "1900-01-01")
```

```
## [1] "2012-08-30"
```

- %d -> Day
- %m -> Numeric Month
- %b -> Abbreviated Month
- %B -> Full Month
- %y -> 2-digit year
- %Y -> 4-digit year

Dates

Convert the following to date objects

- “Jan 1, 2018”
- “31/1/18”

Dates

We can also make sequences of dates

```
# First day of every month
seq(as.Date("2018-01-01"),
    as.Date("2018-04-01"),
    by = "months")
```

```
## [1] "2018-01-01" "2018-02-01" "2018-03-01" "2018-04-01"
```

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

```
## [1] "2018-01-01" "2018-01-03" "2018-01-05"
```

Dates

We can also subtract from sequences of dates!

```
# Last day of every month
seq(as.Date("2018-01-01"),
    as.Date("2018-04-01"),
    by = "months") - 1
```

```
## [1] "2017-12-31" "2018-01-31" "2018-02-28" "2018-03-31"
```

Dates

Let's look at time series of real median weekly earnings:

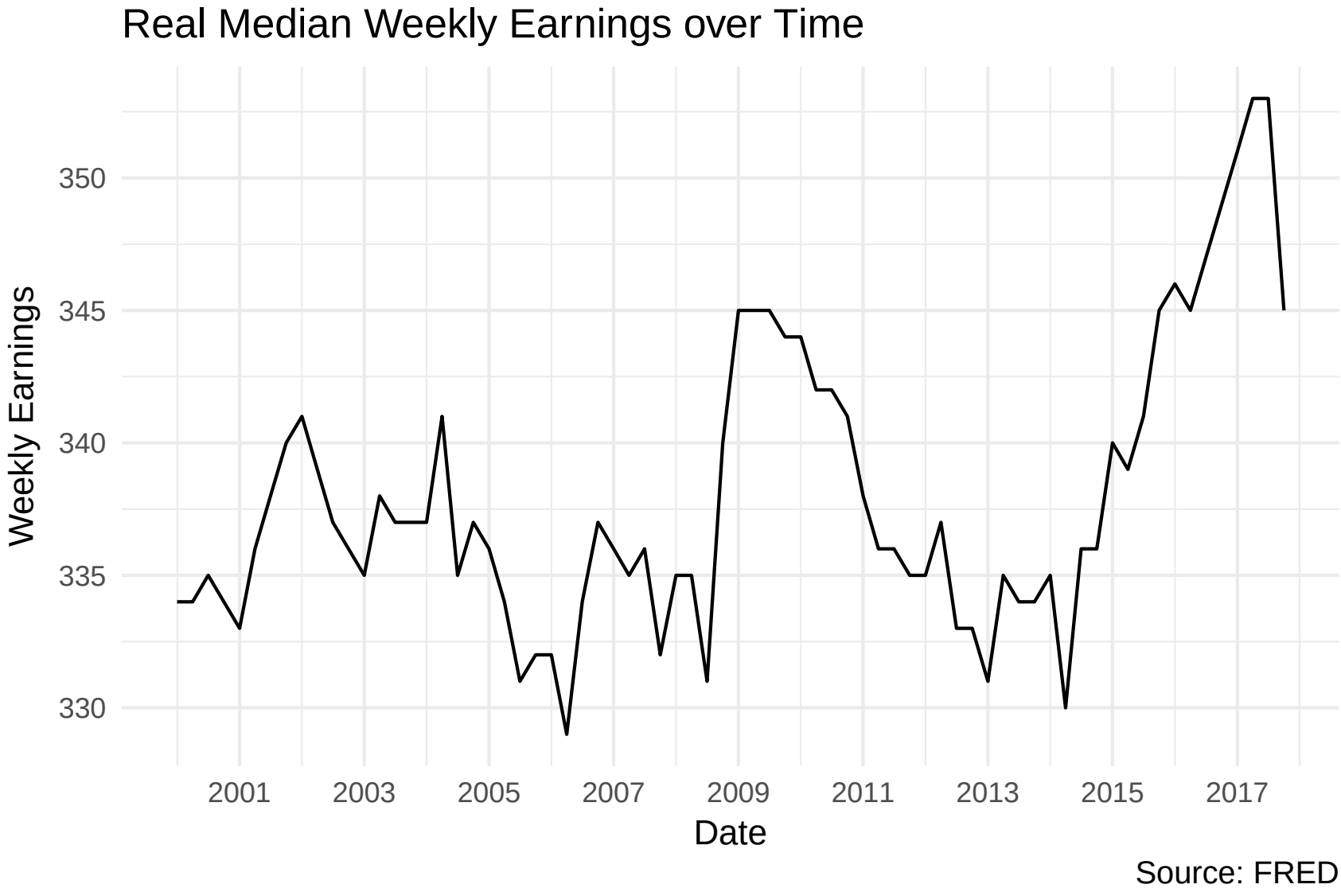
- Read in the file fred_median_income.csv
- What columns are in the dataframe?

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

Dates

- Convert the date column to a date object
- Make a line chart with the date on the x axis and Median_earnings on the y axis
- Be sure to label your chart correctly
- Use the scale_x_dates() function to label every 2 years

Dates



- How have average weekly earnings recovered since the great recession? How have they changed since 2000?

Dates: `lubridate`

- As you can see, we are currently working with quarterly data.
- Let's convert this to annual data using the package `lubridate`.
- `lubridate` makes it easy to extract parts of a date from date objects.

Dates: lubridate

- In this case, we will need to extract the year from our date column.

```
library(lubridate)
date <- as.Date("2018-02-09")

# What is the year of this date?
year(date)
```

```
## [1] 2018
```

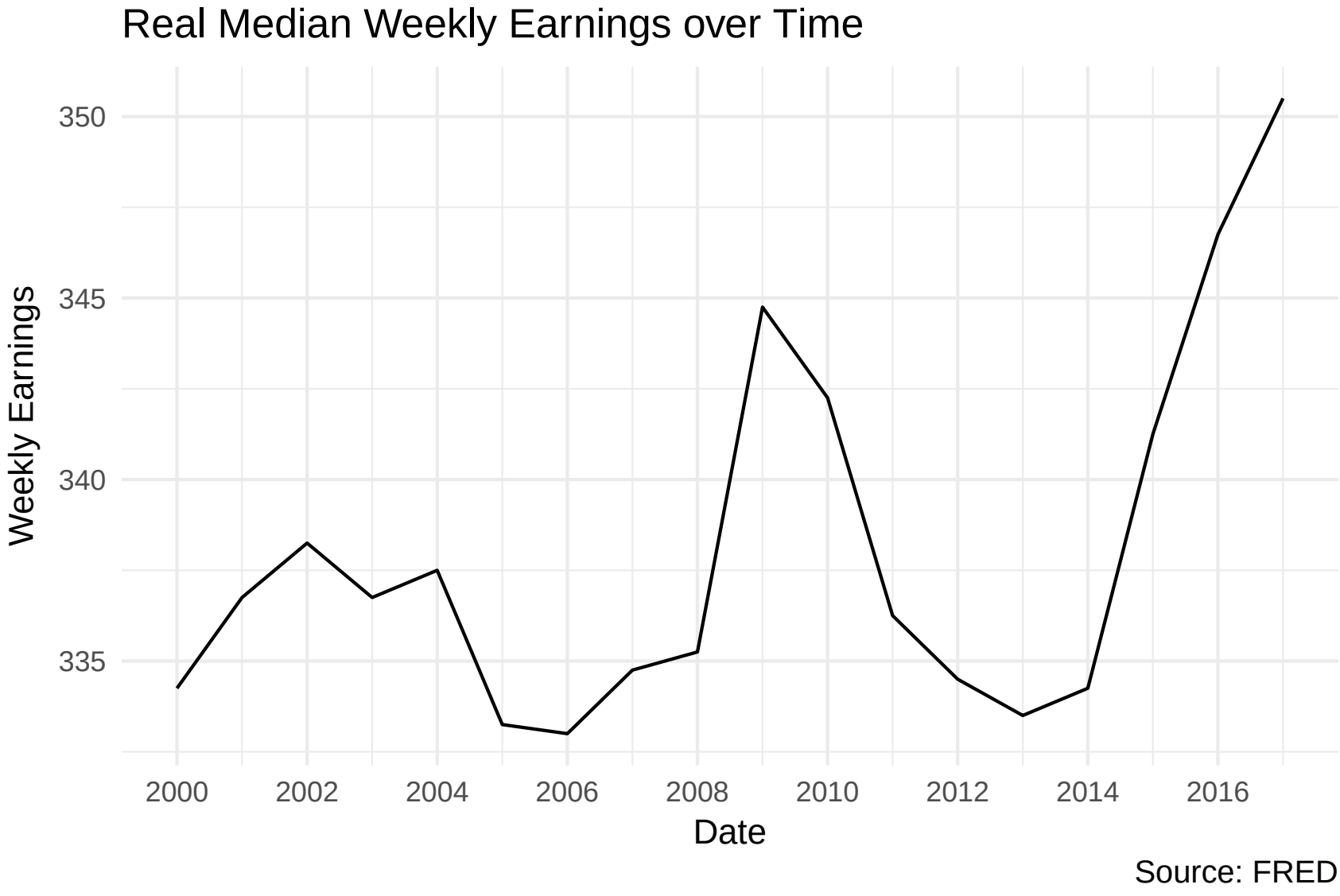
```
# What is the month of this date?
month(date)
```

```
## [1] 2
```

Dates: `lubridate`

- Create a new column of years
- Use `group_by()` and `summarise()` to create annual averages of median earnings
- Create a line chart exactly like the one before
- HINT: you won't be able to use `scale_x_date()` on the year variable

Dates: `lubridate`



- When might it be useful to aggregate data to a lower frequency?

Housekeeping

- REMINDER: your first project assignment is due next week (Friday, 2/16).
- This is your **project proposal** document.

Housekeeping

- Homework for the entire module is in one document, **Module_1_STUDENT.Rmd**.
- Look under the "Day 1", "Day 2", and "Day 3" headers.
- After today, you will work on the "Day 3" portion of the document.