Visualization: Part Two

Customizing the aesthetic in ggplot2 and mapmaking

Federal Reserve Board of Governors
Howard University

Introduction

- Previously:
  - `ggplot()`
  - `+`
  - `geom_`
- Today:
  - Under the hood
  - Customization
  - Large data
  - Making maps

Data

- ACS data for California
  - This data is titled `acs_large.csv`
  - The data dictionary can be found here
  - Recall that California is a large, nationally representative state

Economic Question

- How wages change based on education level
  - Ex. do Bachelor's degree holders make more than Associate's degree holders over their lifetime?

Setup

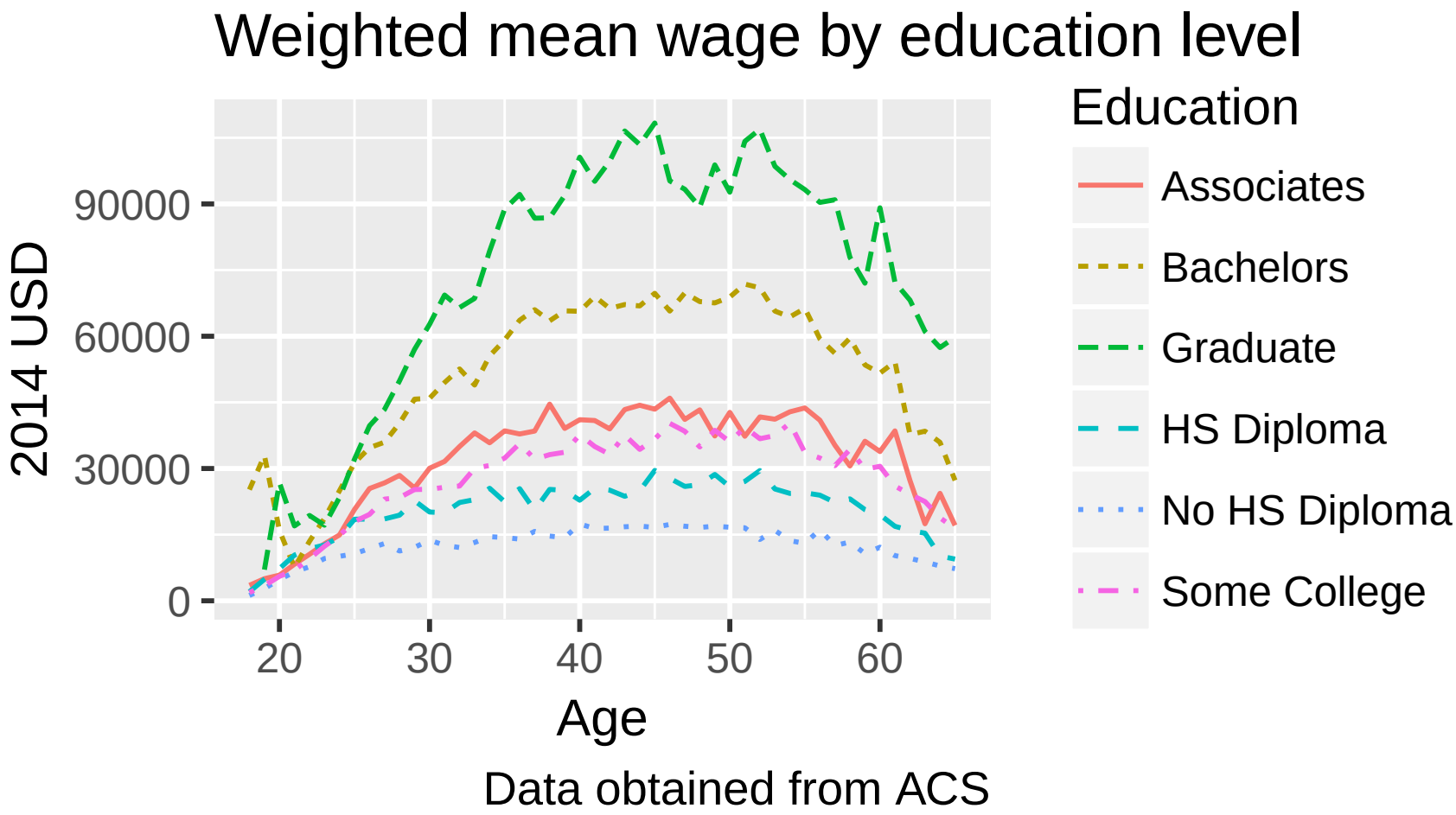
- `tidyverse` and `scales`
- Take a minute to library the needed packages and read in the dataset (`acs_large.csv`), call it "acs"

```
Out:  [1] 200000     13
```

```
Out:   [1] "PUMA00"    "PUMA10"    "AGEP"
Out:   [4] "PINCP"     "PWGTP"     "WAGP"
Out:   [7] "sex"       "eduStat"   "WKHP"
Out:  [10] "empStat"   "Occ"       "mig00Stat"
Out:  [13] "mig10Stat"
```

Review of previous plotting

- Create a plot of the weighted mean wage by education level and age, it should look like this:
  - PWGTP is the column of weights, WAGP is the wage, AGEP is age, and eduStat is the education level



Wages by Education Level

- Higher education = higher wages
- Wages level off and decline after ~50
  - Does this seem accurate? Why might this be?
- Always dig deeper
  - Full-time workers?

ggplot: a theoretical approach

- What do you need to know about a dataset in order to make a 2D line plot of it?

ggplot: example data

- So for a lineplot ggplot would need something like the following:

```
Out:      x           y          color size       linetype
Out:  1 18  3502.145     Associates    1     Associates
Out:  2 18 25210.625      Bachelors    1      Bachelors
Out:  3 18  2048.444     HS Diploma    1     HS Diploma
Out:  4 18  1211.964 No HS Diploma     1 No HS Diploma
```

- Do you always specify all of these variables when plotting a line?

ggplot: what the program sees

- For a single group, ggplot only takes x and y
- Is "Associates" an R color value?
  - How does ggplot decide what color "Associates" should be?

ggplot: scale mapping

- Scales map the data values to visuals

```
Out:      x           y color size linetype
Out:  1 18  3502.145   red    1        3
Out:  2 18 25210.625 green    1        2
Out:  3 18  2048.444  blue    1        5
Out:  4 18  1211.964 black    1        6
```

- How do we control this?

Scale functions

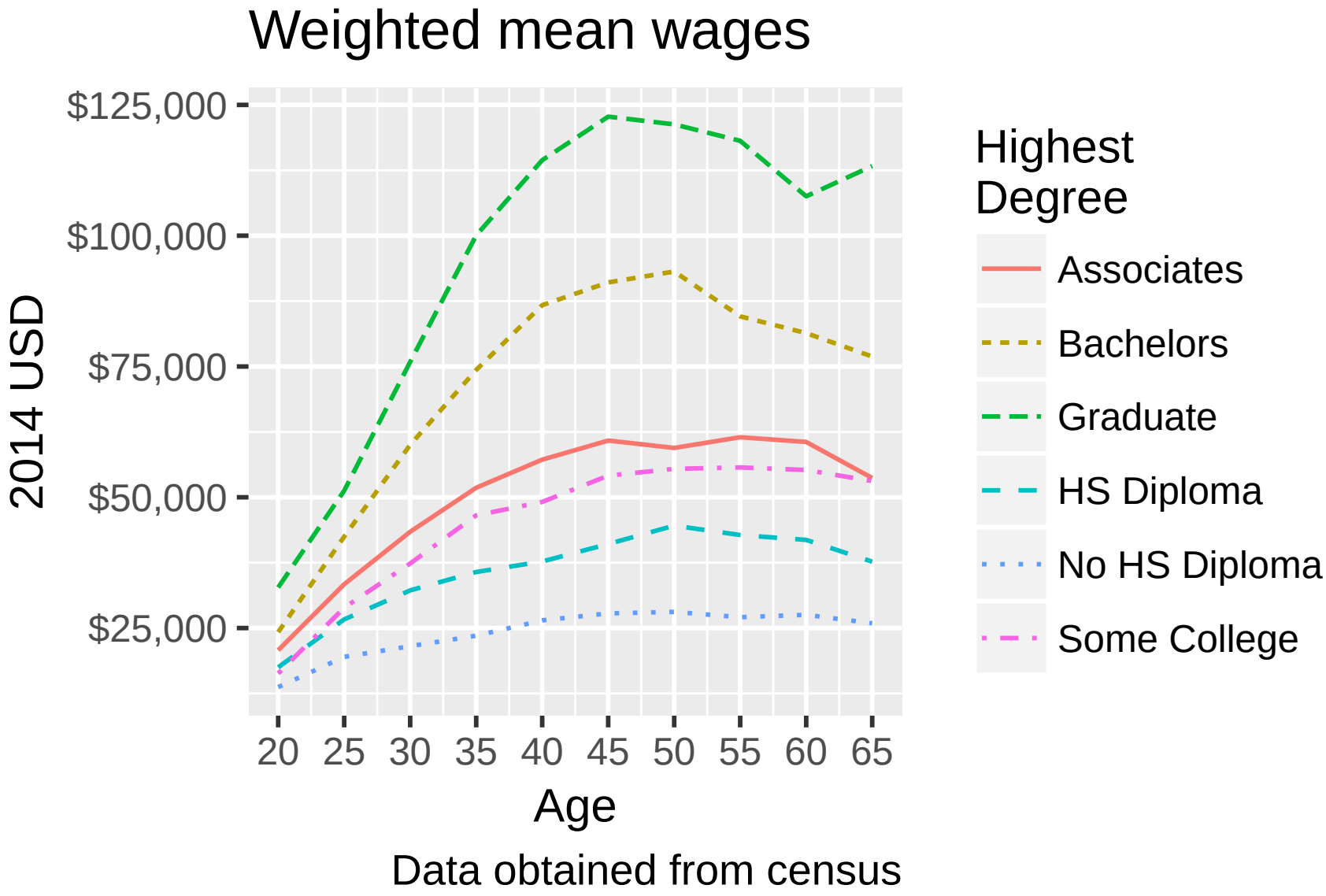
- Map a data value ("Associates"), to a displayed aesthetic ("red")
  - `scale_manual` functions gives us complete control
  - http://www.stat.columbia.edu/~tzheng/files/Rcolor.pdf
- How data are visually represented
  - On the plot itself, (color of the line),
  - Interpretation: (labels for each color).
- Can set axis limits, tick distance etc. . .

Scale Functions in Action

- Full-time Employees Only

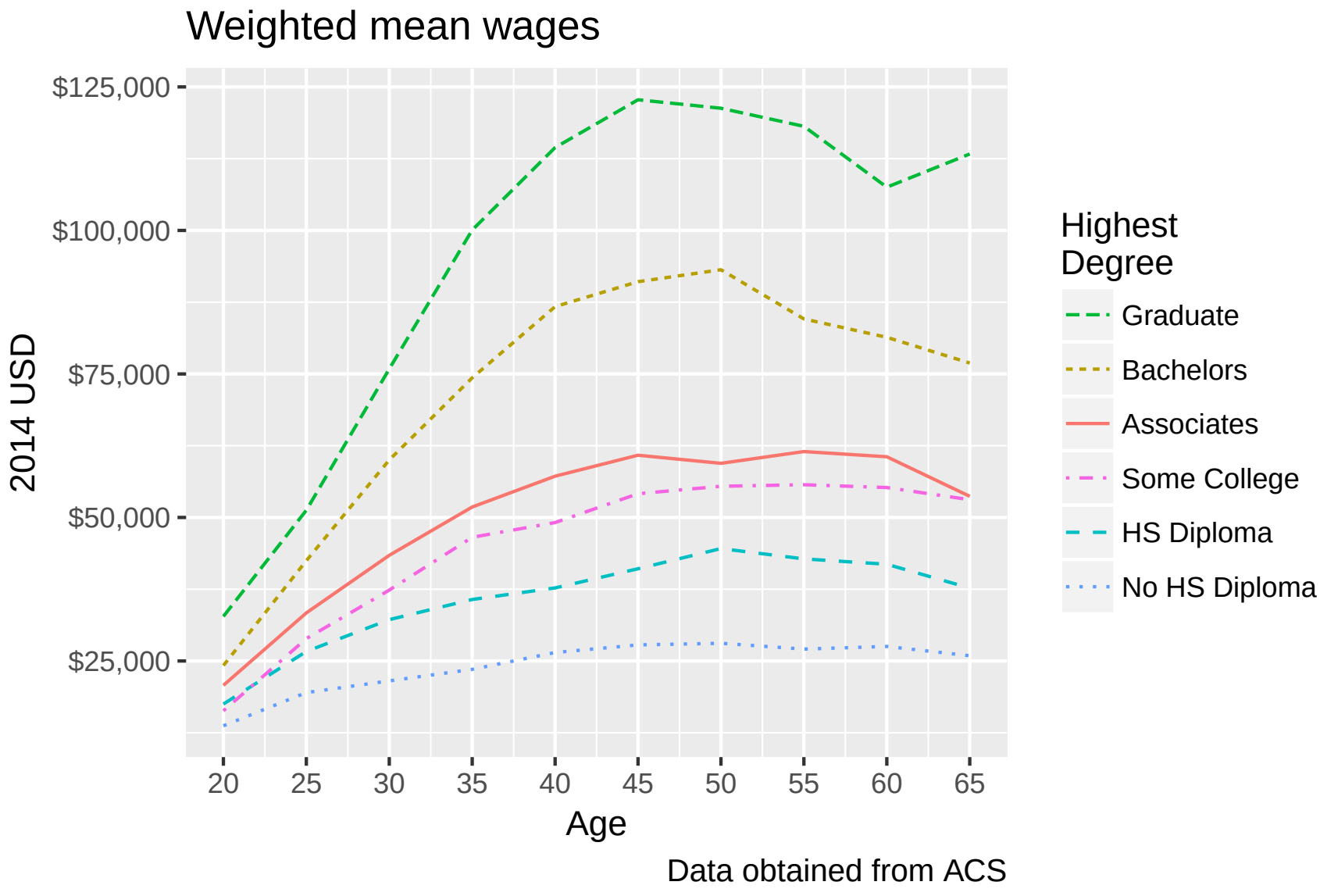
Putting dollars on the axis

- `?dollar`



In class Exercise: Color Guide

- Now change color legend
  - Order by degree
- We just saw a function used to change the tick marks on the x-axis
  - Using this information and the plots we already created, create a chart with a re-ordered color guide
- Hint: what other `scale_color_` and `scale_linetype_` functions are there?


Weighted mean wages
$125,000
$100,000
$75,000
$50,000
$25,000
2014 USD
20
25
30
35
40
45
50
55
60
65
Age
Data obtained from ACS
Highest Degree
Graduate
Bachelors
Associates
Some College
HS Diploma
No HS Diploma

Review

- Grammar of Graphics provides a framework to map data to displayed values
  - Scale functions for formatting axes, color groups, etc.
  - `manual` functions let you overwrite defaults
- Scales package also includes helpful formatting functions
  - `dollar`, `percent`, etc...

Introduction to Theming

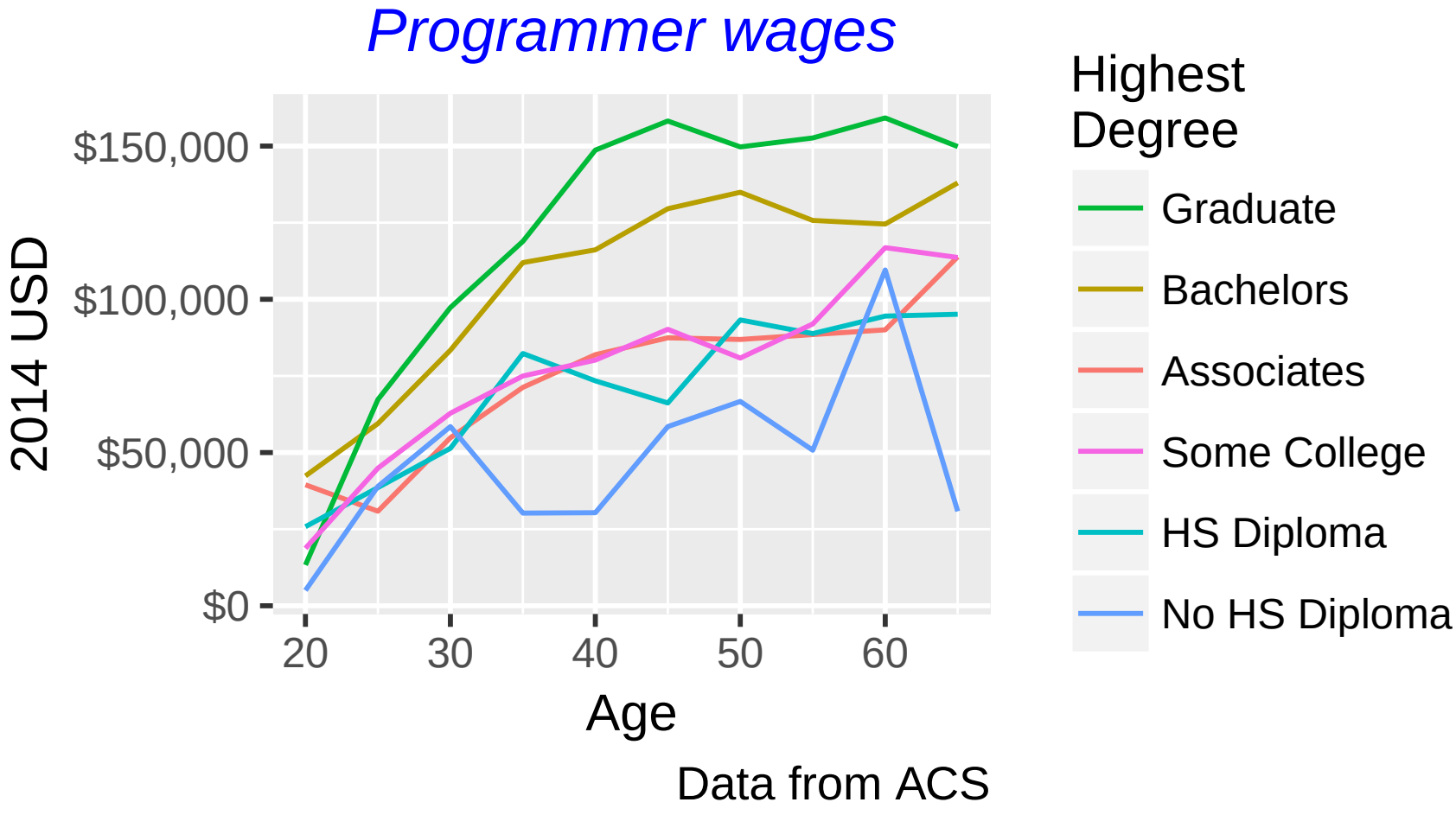
- How do I center the title?
  - Change the font?
  - Change the color?

Themes in ggplot

- `theme()`
- Two parts of visualization:
  - mapping the data to the aesthetic
  - formatting
- Scale functions control data-dependent aspects of the plot
  - Which group is which color, axis limits
- Theme functions control data-independent aspects
  - Font size of the title does not depend on the number of lines plotted

How do Coders do?

- Let's take a look how full-time workers in Computer/Math do



How theme() works

- Like a layer, preceded by +
  - Ex. `ggplot(...) + theme(...)`
- We control the plot title within the `theme()` function with plot.title, legend labels with legend.text
  - Self descriptive system
    - `legend.title`
    - `axis.title.x`, etc...

Elements

- Each argument is an `element_` or `unit`
  - Ex. `theme(plot.title = element.text(...))`
  - We'll talk about the `element` functions today
- Standardized arguments
  - Any text formatting is an `element_text()`

Element_text

- `?element_text()`
- Change all aspects of our text on the plot:
  - The font family (times, comic sans, etc)
  - The font face (bold, italic, underlined)
  - The font color
  - The font size
  - The angle (diagonal text)
  - the justification (hjust is horizontal justification, and vjust is vertical justification)

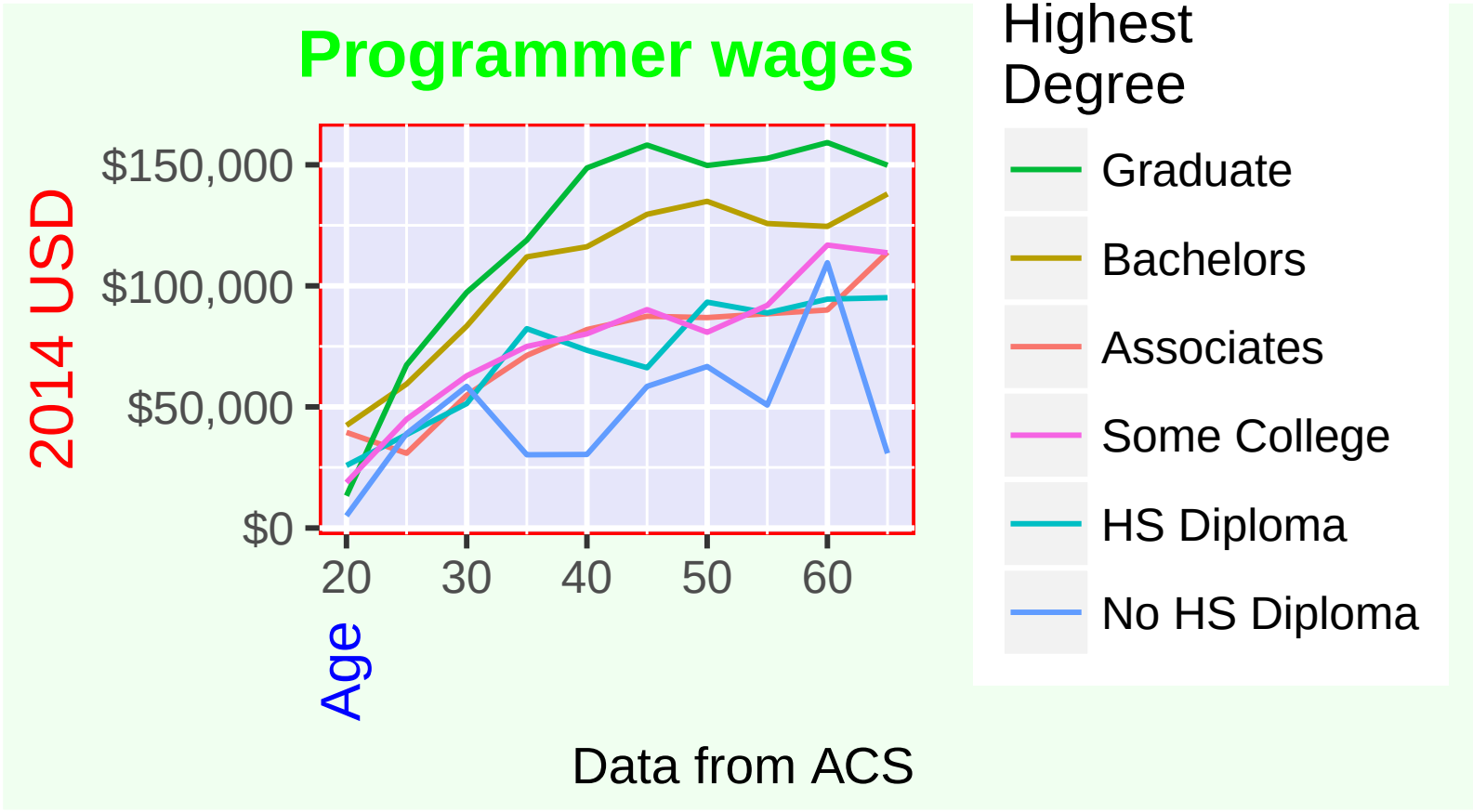
Element_rect

- There are two main rectangles on the plot:
  - The panel: the rectangle where the data is displayed
  - The plot background: the rectangle containing the panel, legend, titles, etc (usually white)

In class exercise: Element_text and Element_rect

- Ok, now it's your turn, using the theme function to update the `coder_plot` ggplot we already made:
  - make your title green, bolded, and right justified
  - make your x axis text blue and vertical
  - make your y axis text red
  - make the panel lavender with a red outline
  - make the plot background honeydew
- Check out the help for `theme`, `element_text`, and `element_rect` if you get stuck

In class exercise: Element_text and Element_rect



What do the `color` and `size` arguments of `element_rect()` control?

Review

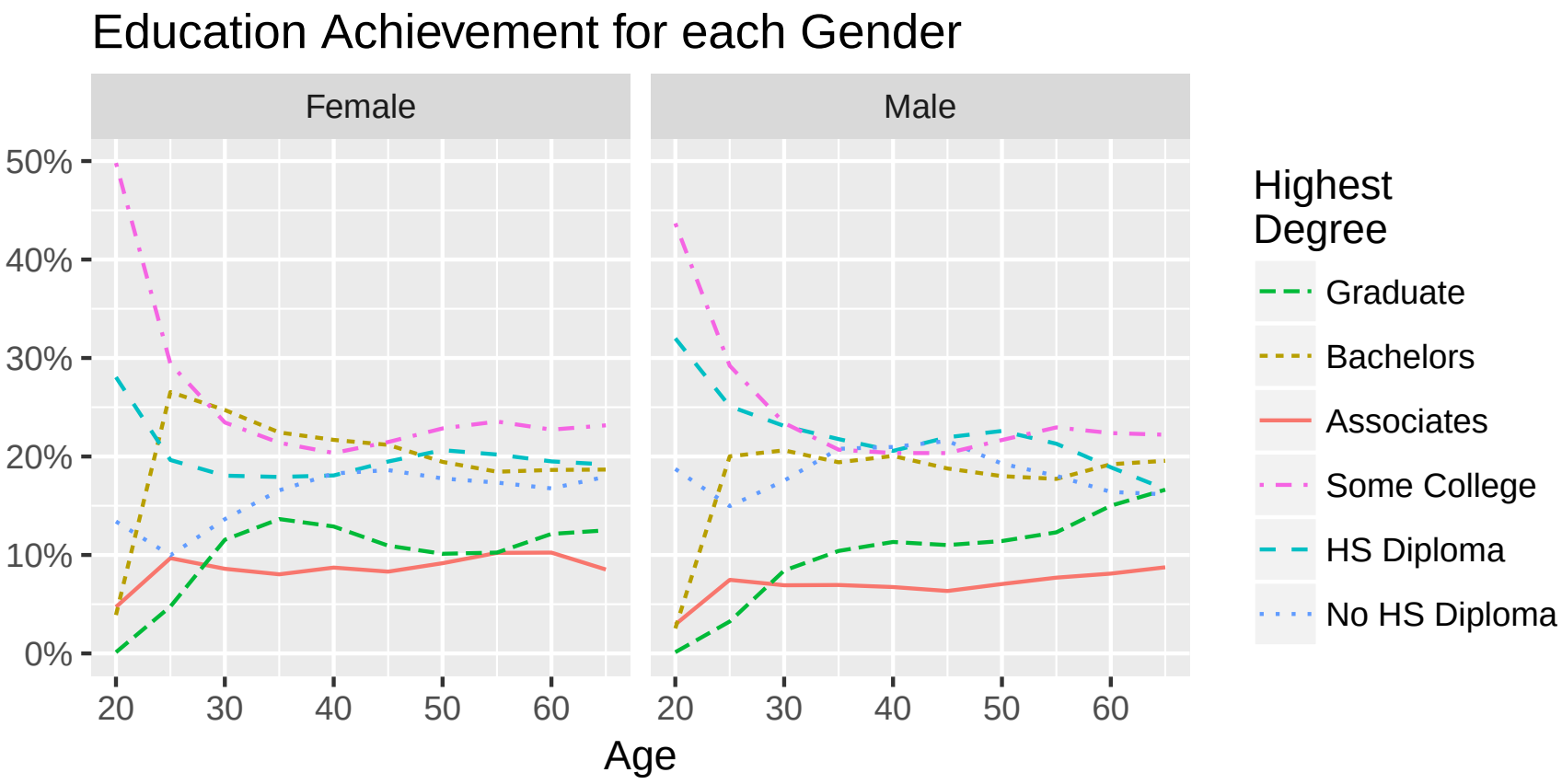
- Mapping data to visuals is done by the `scale` functions
- Formatting is controlled via the `theme()` function
  - `element_text()`
  - `element_rect()`
  - `element_line()`
  - `element_blank()`
    - Ex. `theme(panel.grid = element_blank())` for no gridlines.

Faceting

- Allows us to compare different groups (gender, education level) across the same dimensions at once
- For example, let's look at educational attainment for men and women

Faceting in Action

- Two plots side by side, `facet_wrap()`

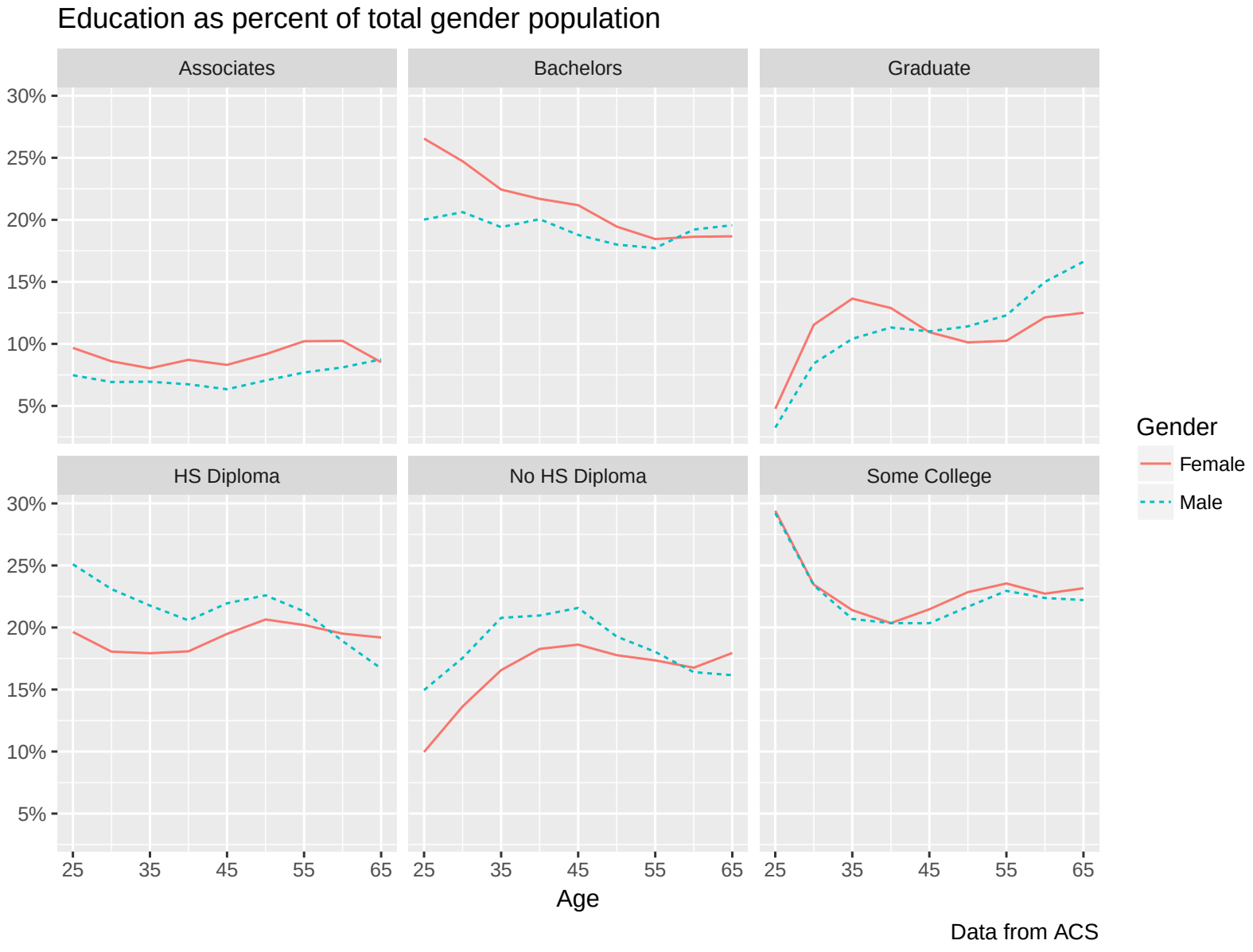


What does this plot tell us about women under 40?

How else can we improve this graph?

In Class exercise: Faceting

- What percent of the male/female populations have each degree level?
  - Use faceting to show a different plot for each level of education
- Restrict our dataset to indivudals who are at least **25 years old**


Education as percent of total gender population
Associates
Bachelors
Graduate
HS Diploma
No HS Diploma
Some College
30%
25%
20%
15%
10%
5%
25
35
45
55
65
Age
Gender
Female
Male
Data from ACS

Review: faceting

- Faceting adds another dimension to our plots
  - Allows us to compare an extra variable
- Enforces good habits due to shared axes

Making Maps with GGplot

Very Exciting!!

Making our first map

- Heatmap of California
  - Let's look at average wage by geographic location
- We need wage data by geographic region
  - What are `PUMA10` and `PUMA00`?
- We need a way to tell ggplot our geographic shapes

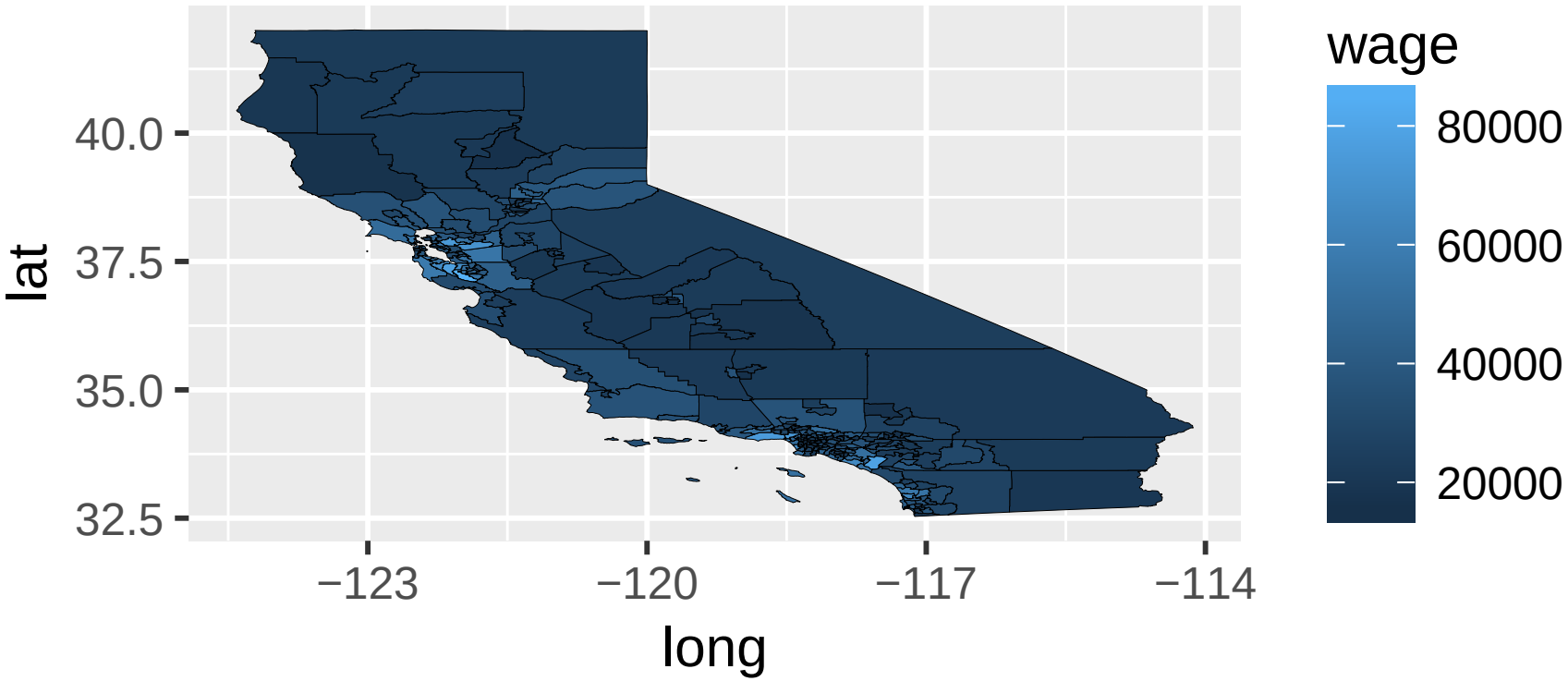
Setting up our geographic data

- We will want to use the following libraries in our maps: `maps` and `viridis`
- `library()` the packages if you haven't already and set up our `wage_data`

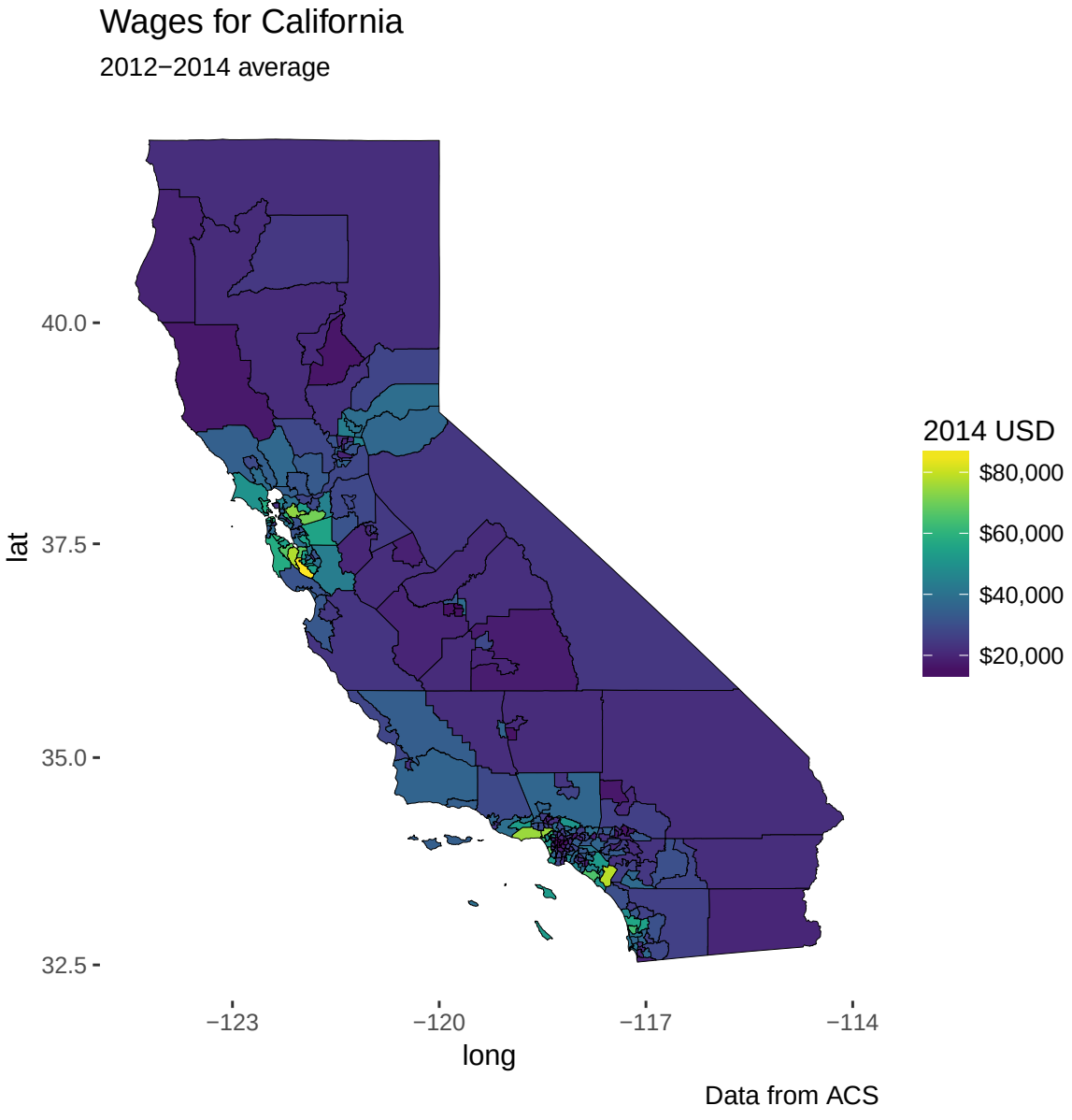
Map data file

- I've already set up a file that takes creates the correct mapping for our geographic areas, use the pumas_points csv file.
- Take a look at our `wage_map_data`: we've now added our column of wage values to dataset which had the longitude and latitudes for each PUMA.
  - Note that the `id` column has replaced `PUMA10`, as they are the same.

Making the map



Improving our Map



What did `coord_map()` do?

Concentration of Occupation

- Urban areas seem to have higher wages
- Could this be related to occupation type?
- Let's use ACS data to find percentage of people in high-wage occupations in each area

In class Exercise: Occupations

- Using our ACS data, find the 5 occupations with the highest full-time average salary
  - Aka working 35+ hours
  - Use weighted observations for count
- Call the data frame for your answer: `high_paying_occs`, it should look like this:

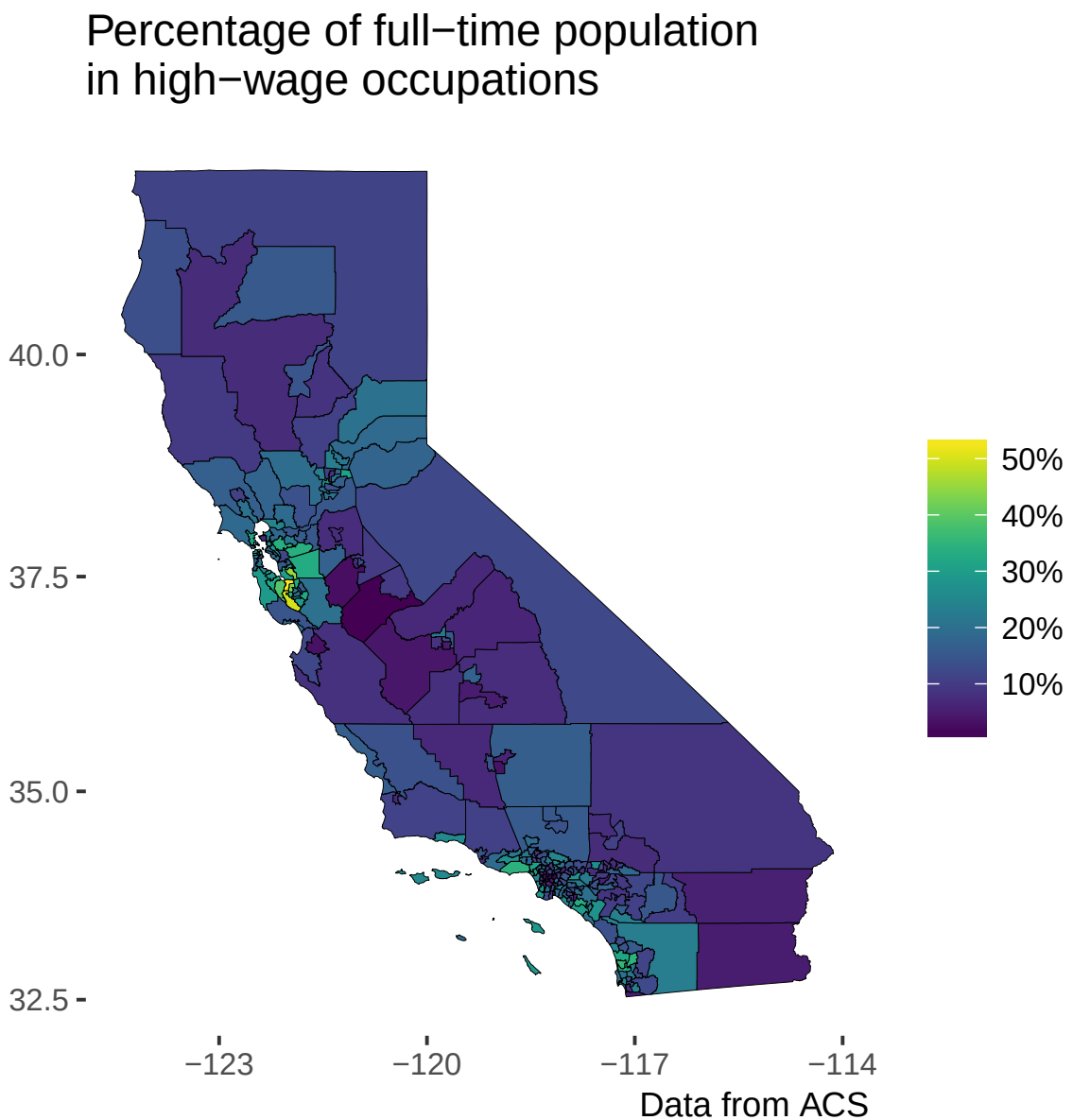
```
Out:  # A tibble: 5 x 2
Out:                          Occ     wages
Out:                        <chr>     <dbl>
Out:  1                     Legal 118211.59
Out:  2             Computer/Math 106919.68
Out:  3 Architecture/Engineering  91635.47
Out:  4                Healthcare  86769.23
Out:  5  Physical/Social Science  72618.36
```

Data prep for our Occupations

- We need the percentage of employees in high-paying jobs in each PUMA
  - Let's call the data frame `wealthy_worker_data`

In class exercise: Occupation Map

- Using the code from our previous map as a template, and the `wealthy_worker_data` just created
  - Make a heatmap showing the percentage of each PUMA's full-time employed population that works in a high-wage industry.

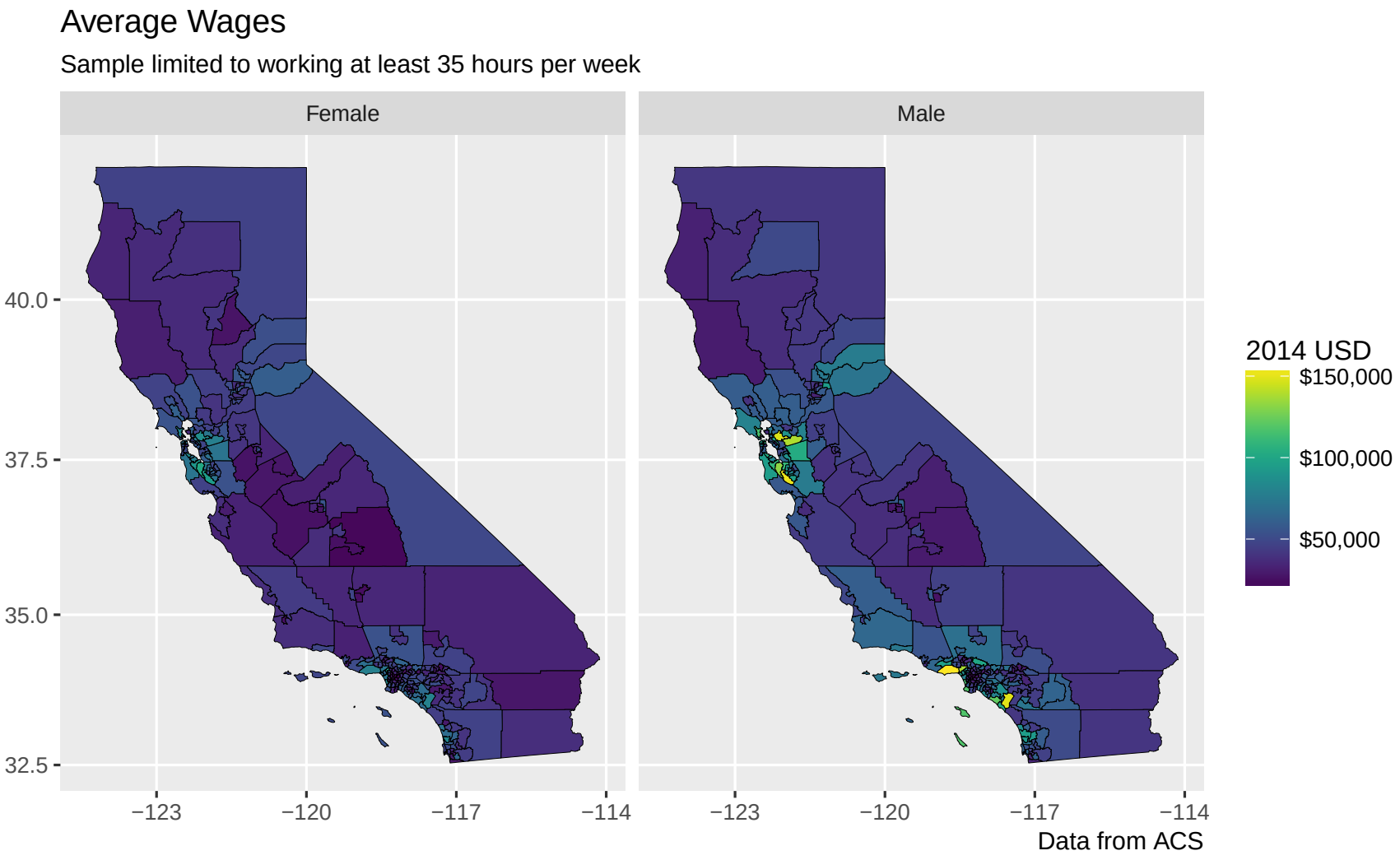


How similar does this map look to our previous map showing geographic wage levels?

Challenge Exercises

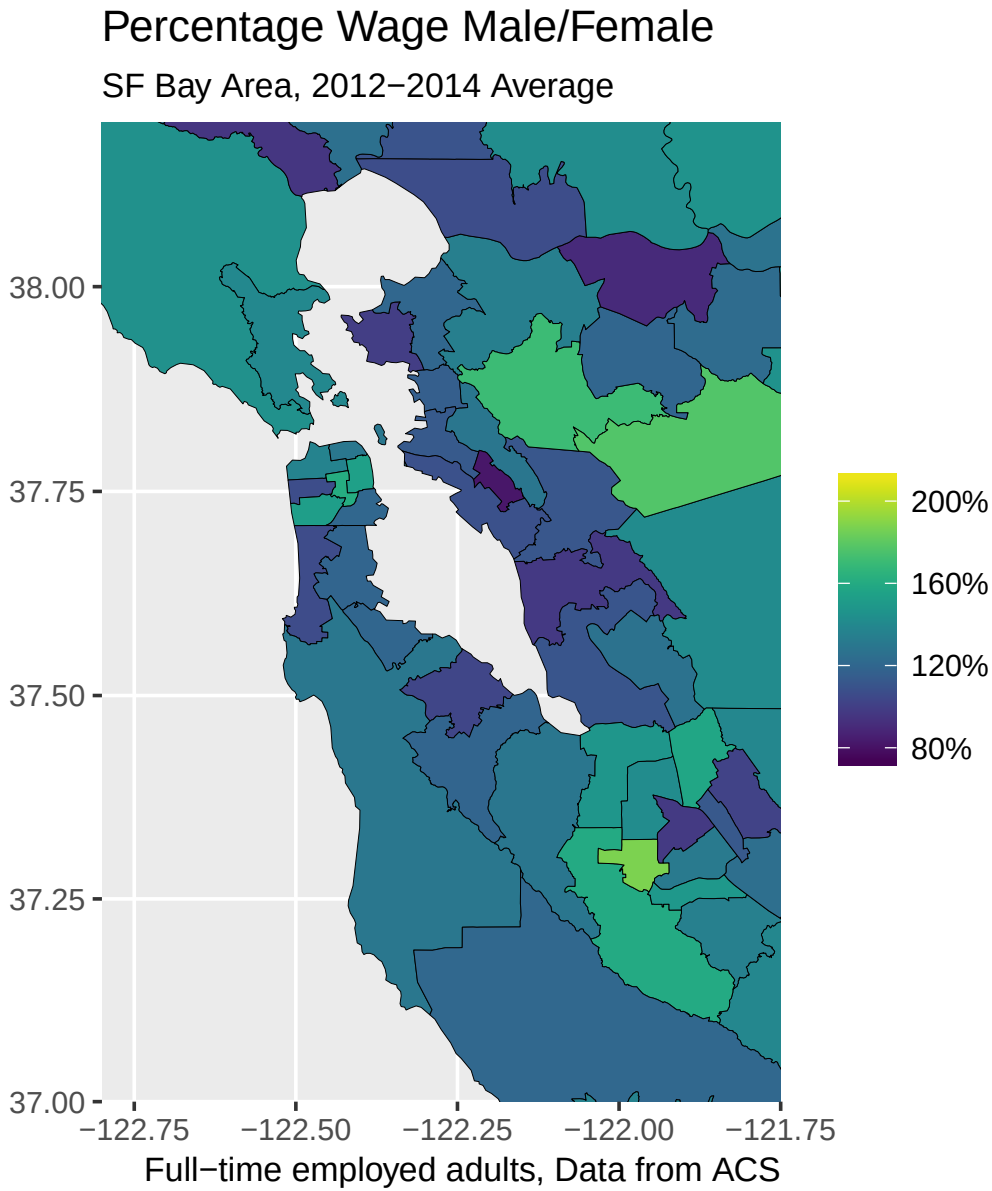
- For the rest of class take a look at creating the following infographics:
  - We want to see how wages for men and women across the state of California
  - For your first chart show the average wages for full-time employed men and women in each PUMA for the entire state, graph these two maps side by side
    - Hint: What function did we look at earlier today that allows us to show multiple charts side by side?
  - For the second chart we want to look at the San Francisco area in particular and show the wage percentage that full-time employed women make compared to full-time employed men in each PUMA

Challenge Exercise 1



What differences do you notice between the maps?

Challenge Exercise 2



Hint: You will need the `xlim` and `ylim` arguments to `coord_map()`