Module 1 Day 1

Becca Jorgensen

```
# load libraries
library(tidyverse)
```

```
## -- Attaching packages ---------------------------- tidyverse 1.2.1 --
```

```
## ✓ ggplot2 2.2.1     ✓ purrr   0.2.4
## ✓ tibble  1.4.1     ✓ dplyr   0.7.4
## ✓ tidyr   0.7.2     ✓ stringr 1.2.0
## ✓ readr   1.1.1     ✓ forcats 0.2.0
```

```
## -- Conflicts ------------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

The American Community Survey

Use the `setwd()` function to set the working directory:

```
setwd("/fsr/home/m1raj01/Class_Materials/Spring_2018/module_1/day_1")
```

Now use the `read_csv()` function to read in the file acs_2016_educ.csv

```
acs_2016_educ <- read_csv("acs_2016_educ.csv")
```

```
## Parsed with column specification:
## cols(
##   EDUC = col_integer(),
##   INCTOT = col_double(),
##   INCWAGE = col_double()
## )
```

What are the dimensions of our data? What function should we use? 11 by 3, dim()

```
dim(acs_2016_educ)
```

```
## [1] 11  3
```

Use the `head()` function to look at the first few rows.

```
head(acs_2016_educ)
```

```
## # A tibble: 6 x 3
##    EDUC INCTOT INCWAGE
##   <int>  <dbl>   <dbl>
## 1     0  20135   12100
## 2     1  26308   12481
## 3     2  18411    8821
## 4     3  13124    7316
## 5     4  12168    6041
## 6     5  10638    5970
```

dplyr

Now try the `glimpse()` function from the dplyr library. What's the difference?

```
glimpse(acs_2016_educ)
```

```
## Observations: 11
## Variables: 3
## $ EDUC    <int> 0, 1, 2, 3, 4, 5, 6, 7, 8, 10, 11
## $ INCTOT  <dbl> 20135.44, 26307.97, 18411.05, 13123.66, 12167.85, 1063...
## $ INCWAGE <dbl> 12100.020, 12480.676, 8821.095, 7315.571, 6041.391, 59...
```

select()

The arguments to select are first the data frame followed by the columns you wish to keep. What does the following code return?

First returns only EDUC, second all but EDUC

```
select(acs_2016_educ, EDUC)
```

```
## # A tibble: 11 x 1
##     EDUC
##    <int>
##  1     0
##  2     1
##  3     2
##  4     3
##  5     4
##  6     5
##  7     6
##  8     7
##  9     8
## 10    10
## 11    11
```

```
select(acs_2016_educ, -EDUC)
```

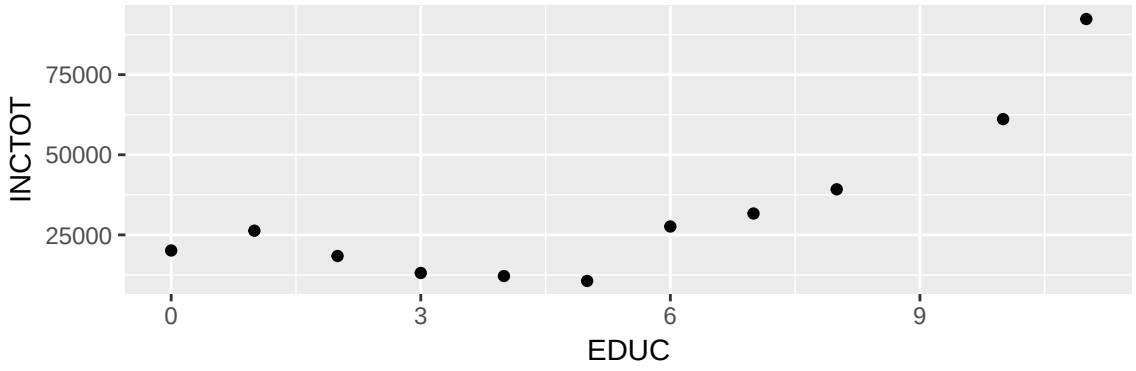
```
## # A tibble: 11 x 2
##    INCTOT INCWAGE
##     <dbl>   <dbl>
##  1  20135   12100
##  2  26308   12481
##  3  18411    8821
##  4  13124    7316
##  5  12168    6041
##  6  10638    5970
##  7  27628   17937
##  8  31673   22582
##  9  39225   30042
## 10  61134   46662
## 11  92365   68701
```

Use the select function to create a new data frame called acs_small that only has the columns EDUC and INCTOT.

ggplot2

Let's make a simple scatter plot:

```
ggplot(data = acs_small,
       aes(x = EDUC, y = INCTOT)) +
    geom_point()
```

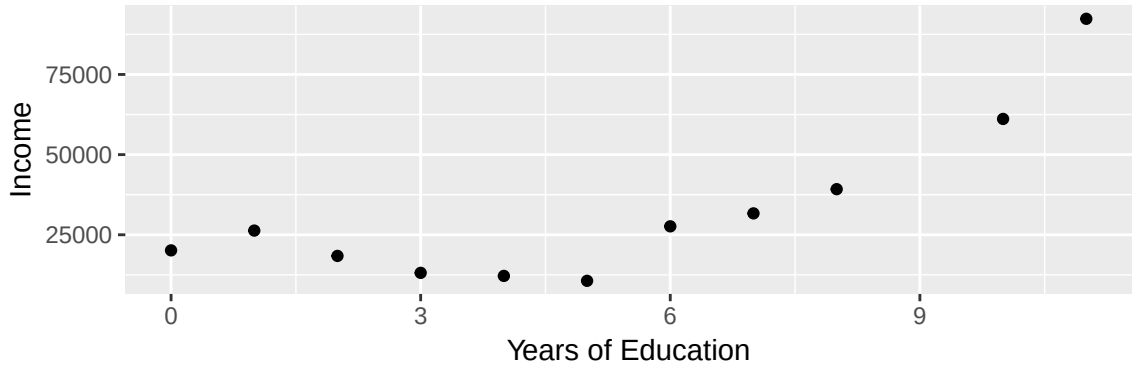


What's a major problem with this chart?

Axes don't have clear labels

Let's add axis labels

```
ggplot(data = acs_small) +
  geom_point(aes(x = EDUC, y = INCTOT)) +
  labs(x = "Years of Education",
       y = "Income")
```



What does this graph tell us about returns to education?

Generally increasing, goes up with category

Why might this chart not give us the full picture?

Other factors might matter (gender, age, experience)

What happens if we comment out our geom layer?

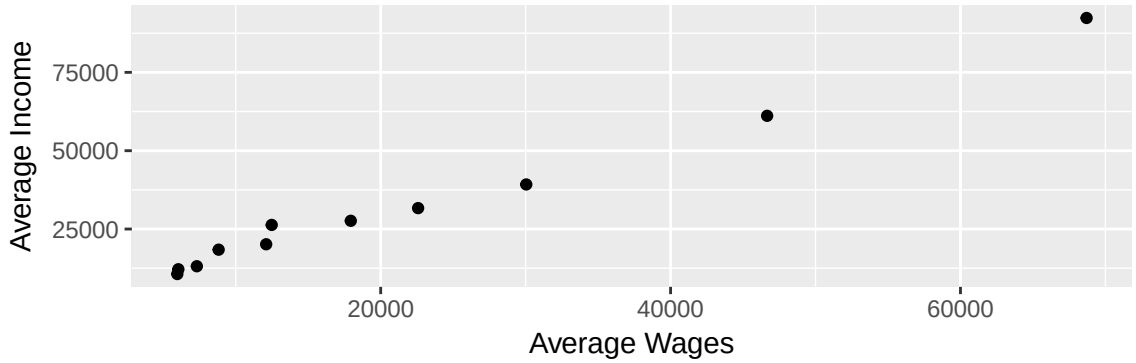
Get a blank chart

Now please create a chart that shows average wages on the x axis and average income on the y axis.

- Be sure to label your axes.
- What would you expect this to look like?
- What steps do you need to take to prepare the chart?

```
ggplot(data = acs_2016_educ,
       aes(x =INCWAGE , y =INCTOT )) +
    geom_point() +
```

```
    labs(x = "Average Wages",
         y = "Average Income")
```



Age and Income

Now read in the file acs_2016_age.csv

```
acs_2016_age <- read_csv("acs_2016_age.csv")
```

```
## Parsed with column specification:
## cols(
##   AGE = col_integer(),
##   INCTOT = col_double(),
##   INCWAGE = col_double()
## )
```

What variables are we dealing with? What functions can we use to look at the data?

```
glimpse(acs_2016_age)
```

```
## Observations: 81
## Variables: 3
## $ AGE     <int> 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29...
## $ INCTOT  <dbl> 869.3353, 1813.0587, 4535.9868, 6547.1609, 9157.6449, ...
## $ INCWAGE <dbl> 406.8056, 1186.9080, 3661.2500, 5553.2550, 8605.4019, ...
```

```
head(acs_2016_age)
```

```
## # A tibble: 6 x 3
##     AGE INCTOT INCWAGE
##   <int>  <dbl>   <dbl>
## 1    16    869     407
## 2    17   1813    1187
## 3    18   4536    3661
## 4    19   6547    5553
## 5    20   9158    8605
## 6    21  10579   10011
```

Let's make a plot showing the gap between wage and total income. First we need to mutate our data frame. Try adding a new column to the acs_2016_age dataframe that is the difference between total and wage income.

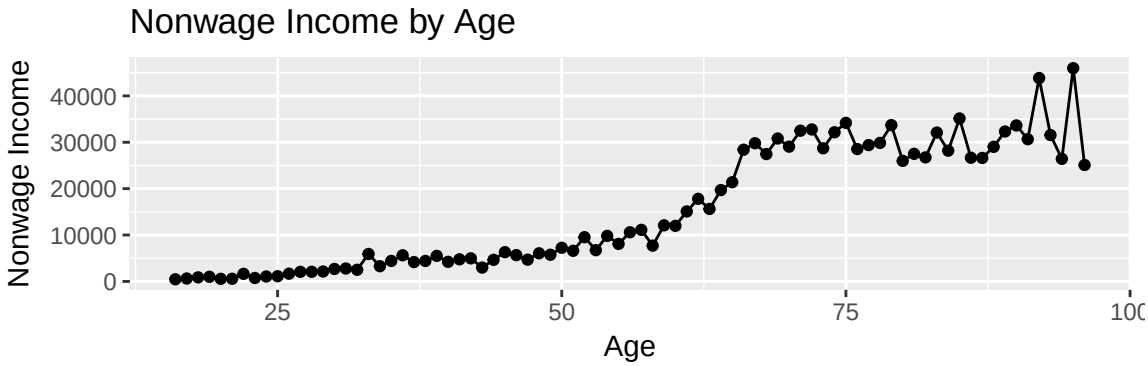
Now make a plot:

- age on the x-axis
- nonwage_income on the y-axis

- appropriate axis labels and a title
- both points and a line

Age and Non-wage Income

```
ggplot(acs_2016_age,
       aes(x = AGE, y = nonwage_income )) +
  geom_point() +
  geom_line() +
  labs(x = "Age",
       y = "Nonwage Income",
       title = "Nonwage Income by Age")
```



Why is this variable "noisy"? What's happening around age 65?

Fewer people as age increases, people are retiring

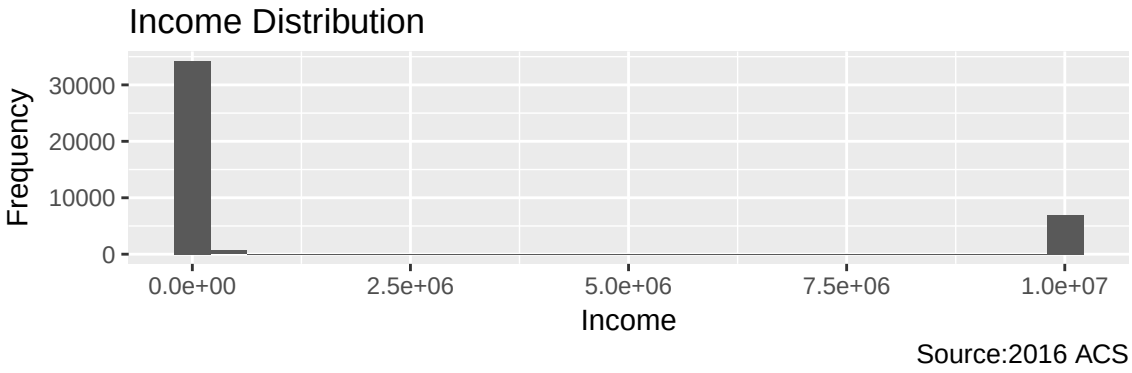
Raw ACS Data

Please read in the data file acs_2016_sample.csv. How many observations are in this dataset? How many variables?

```
## Parsed with column specification:
## cols(
##   YEAR = col_integer(),
##   STATEFIP = col_integer(),
##   METRO = col_integer(),
##   PERWT = col_double(),
##   SEX = col_integer(),
##   AGE = col_integer(),
##   RACE = col_integer(),
##   RACED = col_integer(),
##   HISPAN = col_integer(),
##   HISPAND = col_integer(),
##   EDUC = col_integer(),
##   EDUCD = col_integer(),
##   EMPSTAT = col_integer(),
##   EMPSTATD = col_integer(),
##   INCTOT = col_integer(),
##   INCWAGE = col_integer()
## )
```

Let's make a histogram of the raw ACS income variable:

```
ggplot(acs_2016_sample, aes(x =INCTOT )) +
  geom_histogram(bins =25 ) +
  labs(x = "Income",
       y = "Frequency",
       title = "Income Distribution",
       caption = "Source:2016 ACS")
```



Why does the distribution have a lot of mass at 10 million?

Missing values coded as 9999999

filter()

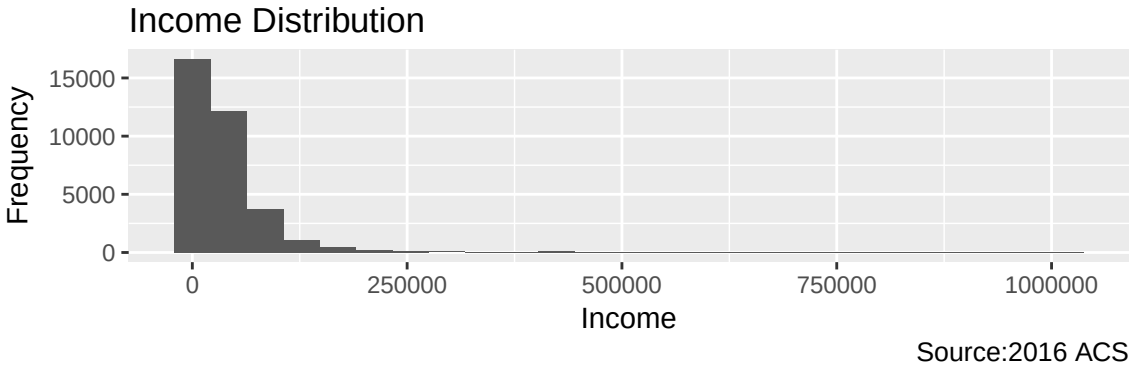
We need to remove the observations that are missing INCTOT (coded as 9999999). The appropriate dplyr verb is filter(). What logical operator could we use?

!=

```
acs_2016_sample <-
  filter(acs_2016_sample , INCTOT<9999999 )
```

Now remake the same histogram but with the filtered data:

```
ggplot(acs_2016_sample, aes(x =INCTOT )) +
  geom_histogram(bins =25 ) +
  labs(x = "Income",
       y = "Frequency",
       title = "Income Distribution",
       caption = "Source:2016 ACS")
```



Clean up the INCWAGE variable as well:

```
acs_2016_sample <-
  filter(acs_2016_sample , INCWAGE < 9999999)
```

Please created a new data frame called acs_filtered that only has people who make more than $100,000 from wages. What are the dimensions of this dataframe? How many observations did we filter out? Make a histogram of their total income

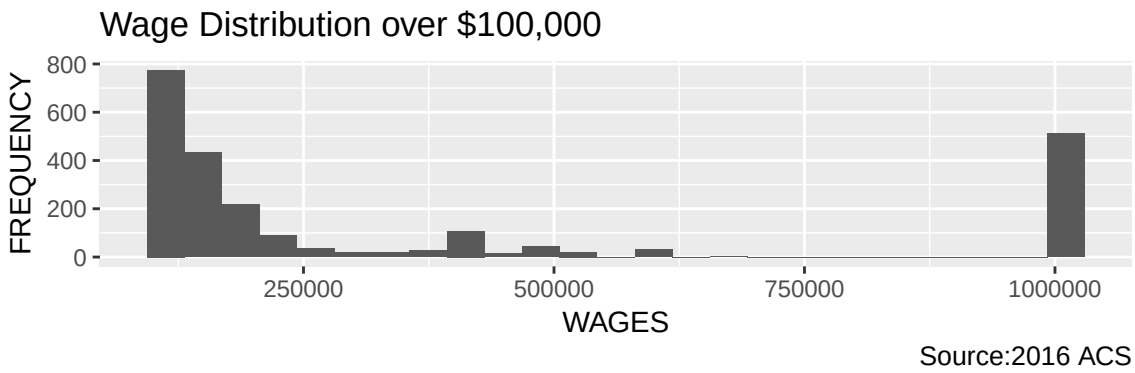
```
acs_filtered <-
  filter(acs_2016_sample ,INCWAGE>100000 )

dim(acs_2016_sample)
```

```
## [1] 34876    16
```

```
dim(acs_filtered)
```

```
## [1] 2344   16
```

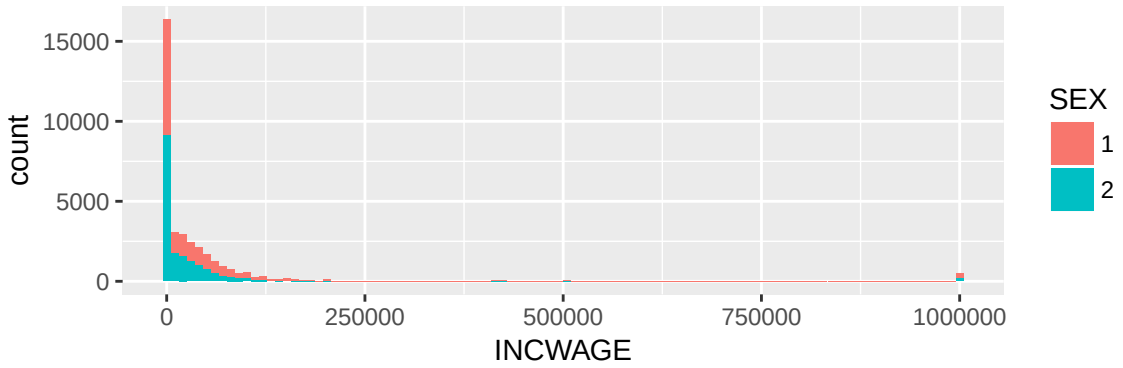
```
ggplot(acs_filtered, aes(x = INCWAGE )) +
  geom_histogram(bins = 25) +
  labs(x = "WAGES",
       y = "FREQUENCY",
       title = "Wage Distribution over $100,000",
       caption = "Source:2016 ACS")
```



Color as an Axis

Just like we can map the x and y aesthetics in ggplot2 we can map color and fill. First we need to turn the SEX variable into a categorical variable using mutate() and factor()

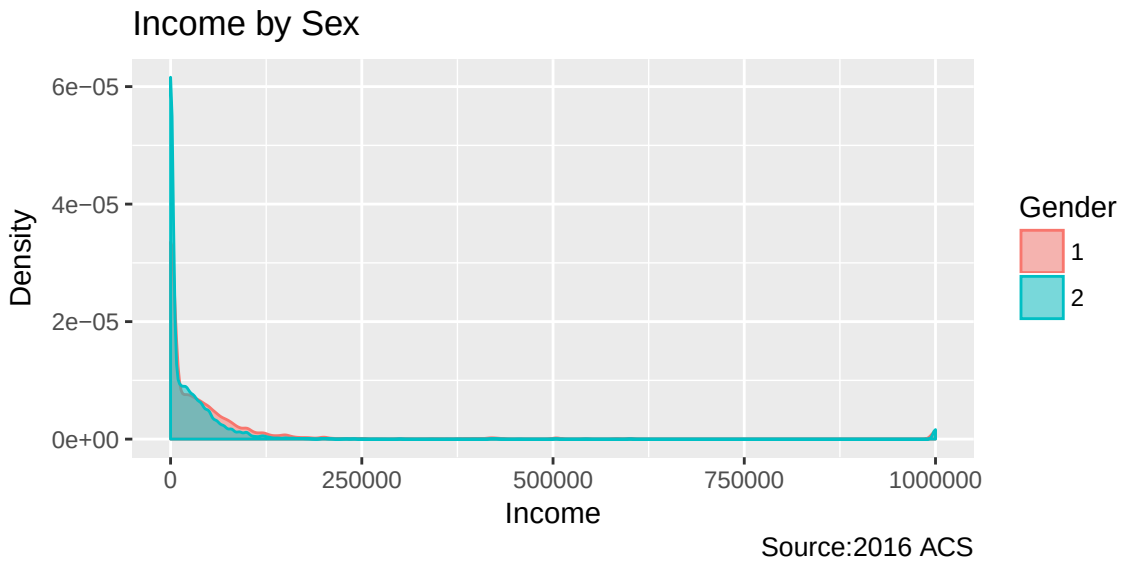
```
acs_2016_sample <-
  mutate(acs_2016_sample, SEX = factor(SEX))

ggplot(acs_2016_sample,
       aes(x = INCWAGE, fill = SEX)) +
  geom_histogram(bins = 100)
```



We can improve this chart

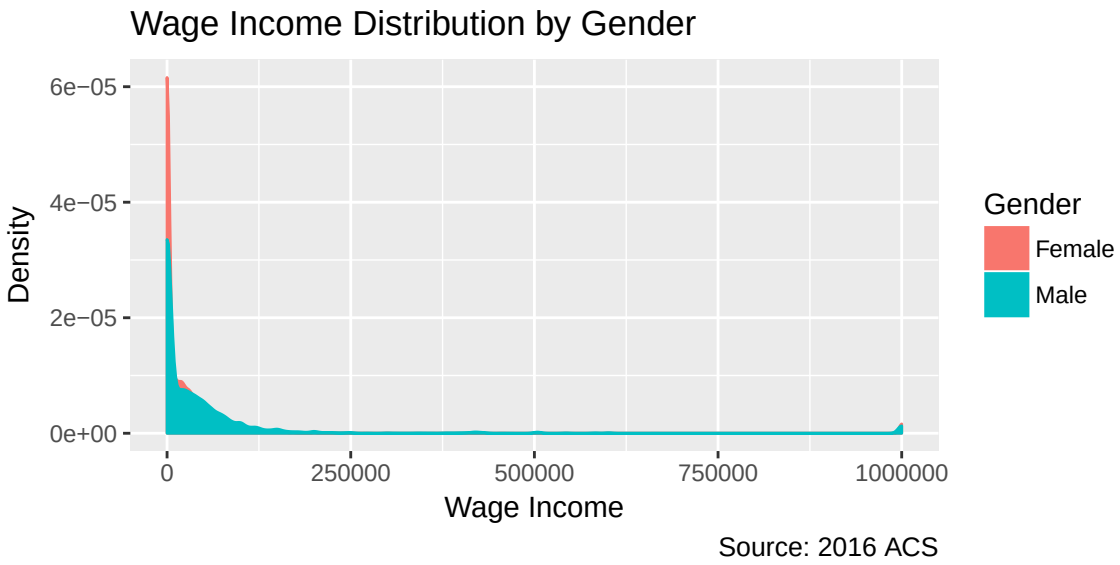
- Add axis labels and a title (be sure to label the fill axis)
- Change the alpha value to 0.5 (What's the alpha value?)
- Change the geom type from histogram to density (Why do we want to do this?)

```
ggplot(acs_2016_sample,
       aes(x = INCWAGE , fill = SEX, color =SEX  )) +
  geom_density(alpha = 0.5)+
  labs(x = "Income",
       y = "Density",
       fill = "Gender",
       color = "Gender",
       title = "Income by Sex",
       caption = "Source:2016 ACS")
```



We want to recode the SEX variable so that instead of 1,2 it is "Male", "Female". We can do this using the `ifelse()` function.

```
acs_2016_sample <-
  mutate(acs_2016_sample,
         SEX = ifelse( SEX == 1 , "Male", "Female"))
```


Wage Income Distribution by Gender
Density
6e−05
4e−05
2e−05
0e+00
0
250000
500000
750000
1000000
Wage Income
Gender
Female
Male
Source: 2016 ACS