POLSCI 9590: Methods I

Learning About and Visualizing Data

Dave Armstrong

Videos

We covered a few different things in the videos:

1. Frequency Distributions
2. Rates and Ratios
    - Ratios only really work for **ratio** level variables.
3. Percentages and Percentiles
4. Visualizing Distributions
    - Bar charts (or, the rightfully-maligned pie chart) for qualitative variables.
    - Histograms for quantitative variables.

Videos

We covered a few different things in the videos:

1. Frequency Distributions
2. Rates and Ratios
    - Ratios only really work for **ratio** level variables.
3. Percentages and Percentiles
4. Visualizing Distributions
    - Bar charts (or, the rightfully-maligned pie chart) for qualitative variables.
    - Histograms for quantitative variables.

Questions?

Required Packages

R | Python | Stata

For what we're doing today, you will need to install the `DAMisc` and the `uwo4419` packages from github.

```
install.packages("remotes")
remotes::install_github("davidaarmstrong/damisc")
remotes::install_github("davidaarmstrong/uwo4419")
```

```
library(ggplot2)
library(dplyr)
library(rio)
library(scales)
library(uwo4419)
library(DAMisc)
```

Sometimes when you install packages, you will be alerted to packages you currently have that have been updated since you installed them. For me, it looked like this (updating all is slightly preferred):

```
These packages have more recent versions available.
It is recommended to update all of them.
Which would you like to update?

1: All
2: CRAN packages only
3: None
4: vroom     (1.5.4   -> 1.5.5  ) [CRAN]
5: e1071     (1.7-8   -> 1.7-9  ) [CRAN]
6: DescTools (0.99.42 -> 0.99.43) [CRAN]
Enter one or more numbers, or an empty line to skip updates:
```

Import the data

R | Python | Stata

```
ces19 <- import("ces19.dta")
```

Frequency Distributions

R Python Stata

```
freqDist(ces19$educ)
```

```
##       Freq   %     Cu %
## 1      491  17.59  17.59
## 2     1029  36.86  54.44
## 3     1272  45.56 100.00
## Total 2792 100.00
```

```
ces19$educ <- factorize(ces19$educ)
freqDist(ces19$educ)
```

```
##              Freq   %     Cu %
## <HS           491  17.59  17.59
## HS/College   1029  36.86  54.44
## College Grad 1272  45.56 100.00
## Total        2792 100.00
```

Summary Statistics

R Python Stata

```
sumStats(ces19, "leader_lib")
```

```
## # A tibble: 1 × 11
##   variable    mean    sd   iqr   min   q25   q50   q75   max     n   nNA
##   <chr>      <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <int> <int>
## 1 leader_lib  43.4  30.8    61     0     9    50    70   100  2799     7
```

Summary Statistics by Group

R Python Stata

```
sumStats(ces19, "leader_lib", byvar="educ")
```

```
## # A tibble: 4 × 12
##   variable   educ     mean    sd   iqr   min   q25   q50   q75   max     n   nNA
##   <chr>      <fct>   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <int> <int>
## 1 leader_lib <HS      35.7  32.6  60        0   0      29  60     100   491     1
## 2 leader_lib HS/Col…  38.9  30.3  58.5      0   6.5    39  65     100  1029     3
## 3 leader_lib Colleg…  50.1  29.2  46        0  29      55  75     100  1272     3
## 4 leader_lib <NA>     26    29.4  32.5      0   7       9  39.5    80     7     0
```

Quantile

R | Python | Stata

You can find any percentile you want using the `quantile()` function. Below is how we would find the 62^{nd} percentile of the `leader_lib` variable.

```
quantile(ces19$leader_lib, .62, na.rm=TRUE)
```

```
## 62%
##  60
```

You can also find multiple percentiles at once.

```
quantile(ces19$leader_lib, c(.38, .62), na.rm=TRUE)
```

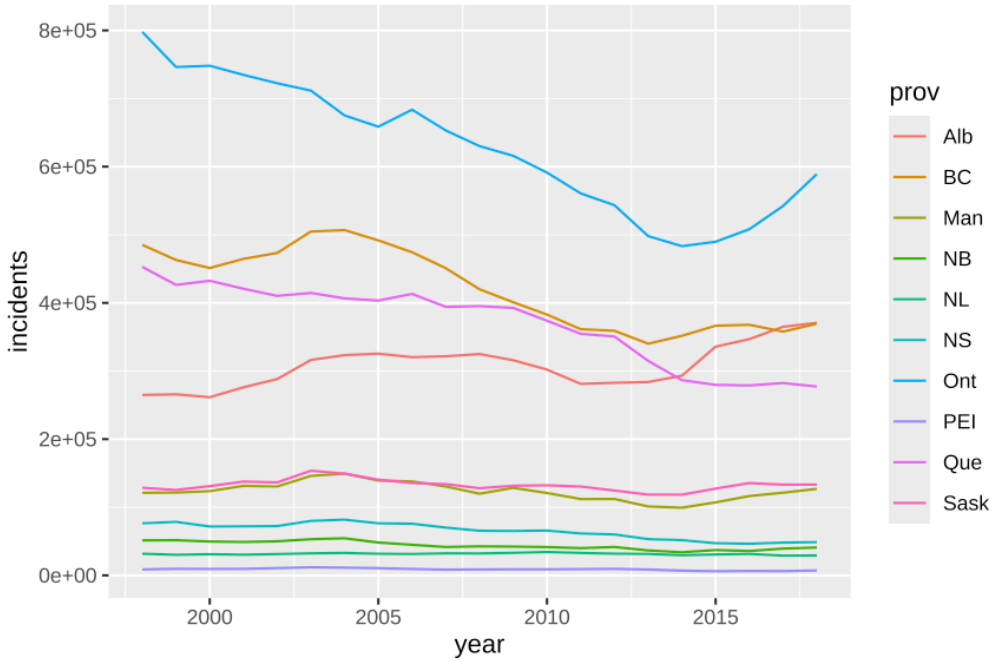
```
## 38% 62%
##  29  60
```

Line Graph

R Python Stata

You can get the crime data from the course OWL page.

```
crime <- import("crime.dta")
ggplot(crime,
       aes(x=year,
           y=incidents,
           colour=prov)) +
  geom_line()
```

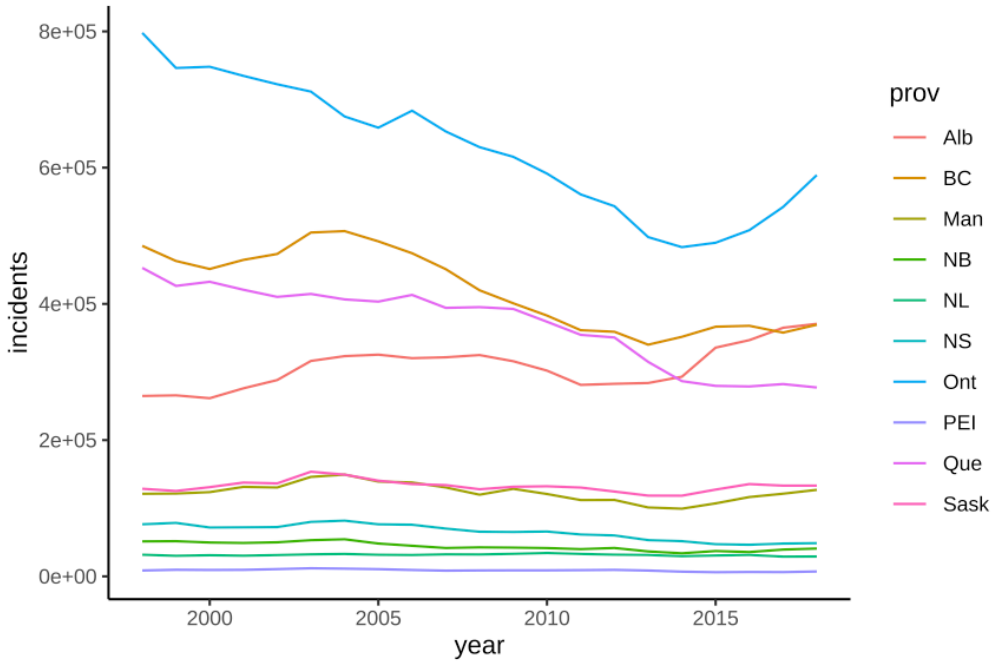


Line Graph

R Python Stata

You can get the crime data from the course OWL page.

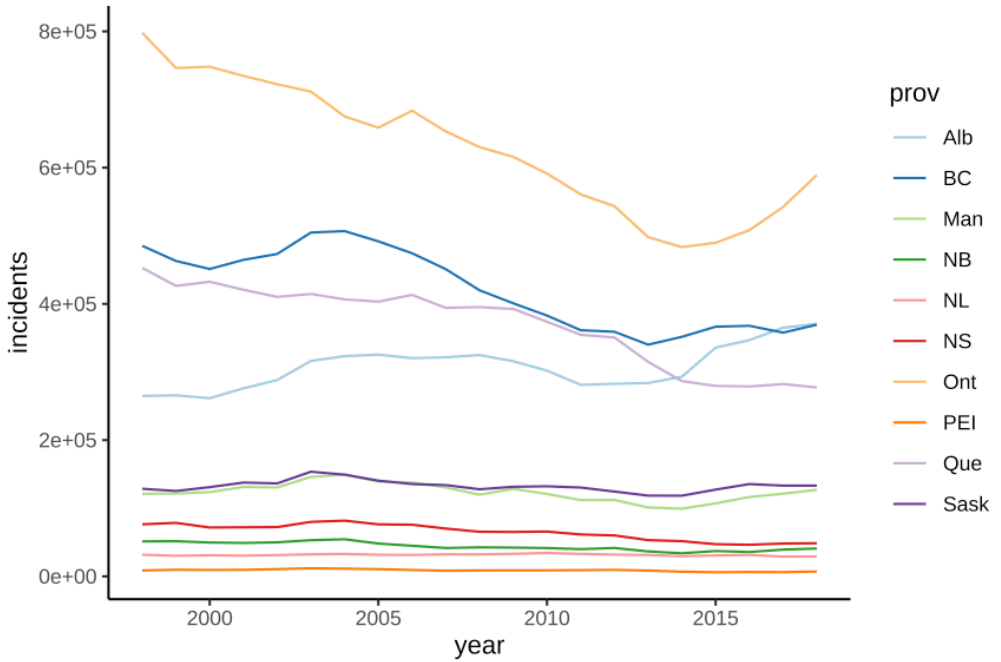
```
ggplot(crime,
       aes(x=year,
           y=incidents,
           colour=prov)) +
  geom_line() +
  theme_classic()
```



Line Graph

R Python Stata

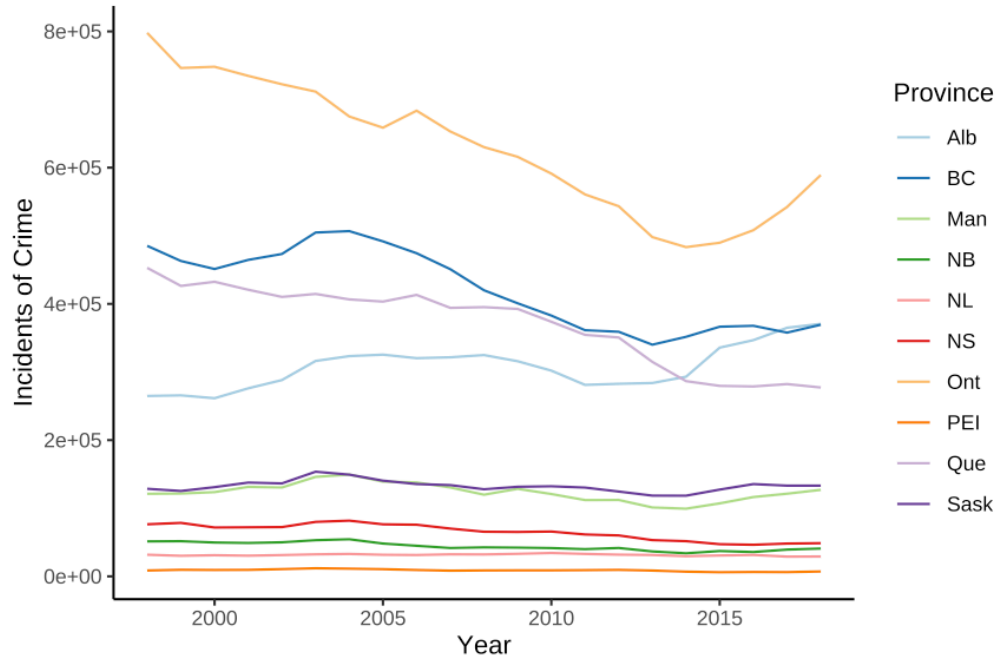
```
ggplot(crime,
       aes(x=year,
           y=incidents,
           colour=prov)) +
  geom_line() +
  theme_classic() +
  scale_color_brewer(palette="Paired")
```



Line Graph

R Python Stata

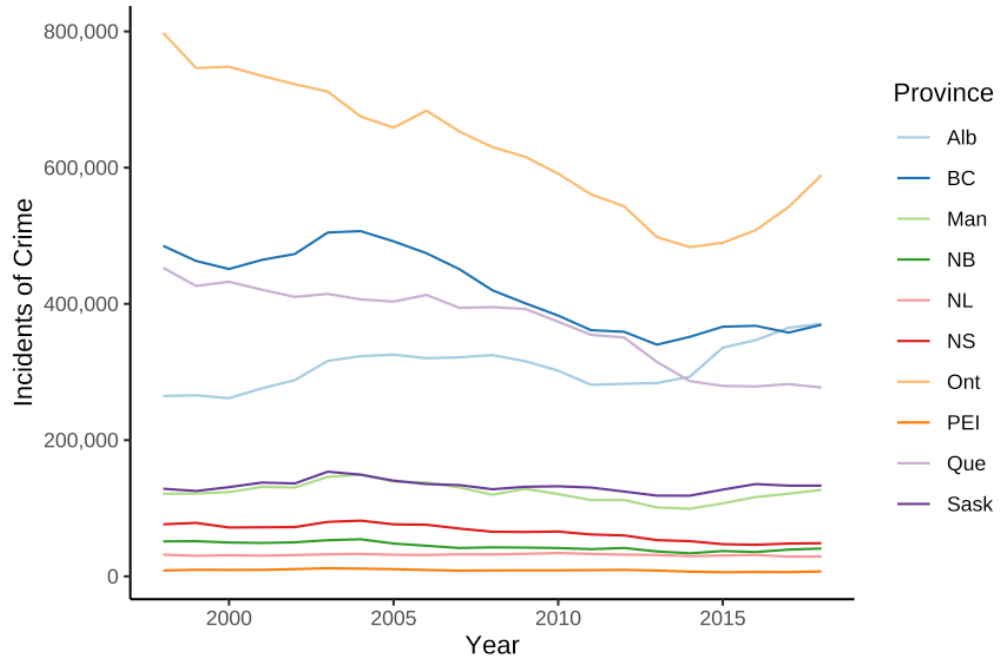
```
ggplot(crime,
       aes(x=year,
           y=incidents,
           colour=prov)) +
  geom_line() +
  theme_classic() +
  scale_color_brewer(palette="Paired") +
  labs(x="Year", y="Incidents of Crime",
       colour="Province")
```



Line Graph

R Python Stata

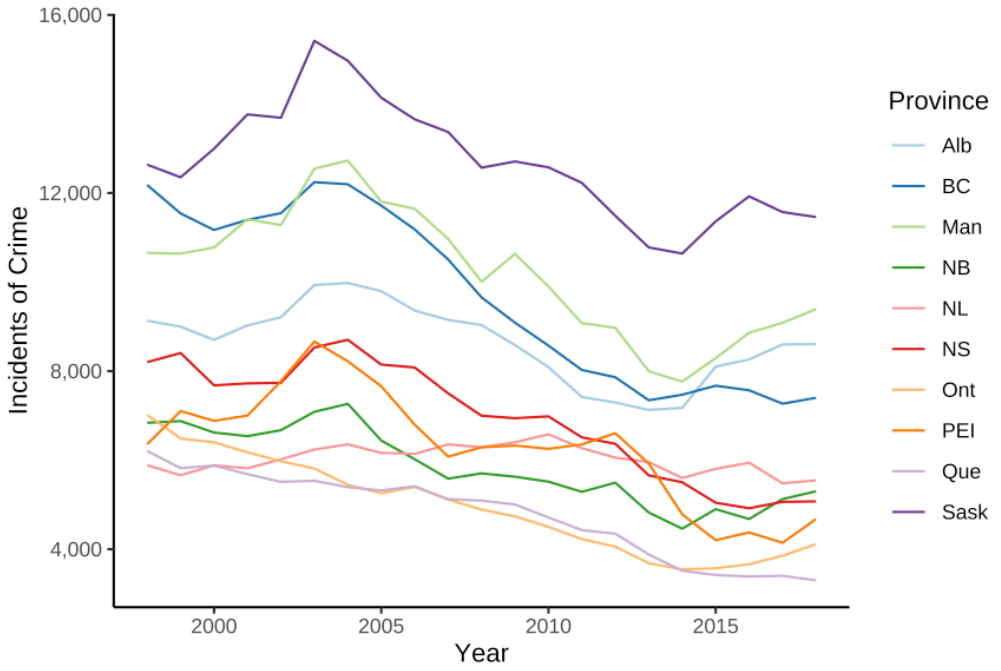
```
ggplot(crime,
       aes(x=year,
           y=incidents,
           colour=prov)) +
  geom_line() +
  theme_classic() +
  scale_colour_brewer(palette="Paired") +
  labs(x="Year", y="Incidents of Crime",
       colour="Province") +
  scale_y_continuous( label = comma)
```



Rates

R | Python | Stata

```
ggplot(crime,
       aes(x=year,
           y=rate,
           colour=prov)) +
  geom_line() +
  theme_classic() +
  scale_colour_brewer(palette="Paired") +
  labs(x="Year", y="Incidents of Crime",
       colour="Province") +
  scale_y_continuous( label = comma)
```



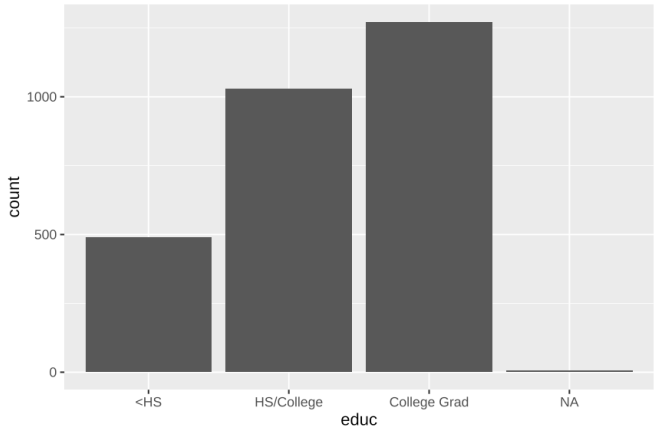
Bar Chart

R | Python | Stata

```
freqDist(ces19$educ)
```

```
##                Freq    %     Cu %
## <HS             491  17.59  17.59
## HS/College     1029  36.86  54.44
## College Grad   1272  45.56 100.00
## Total          2792 100.00
```

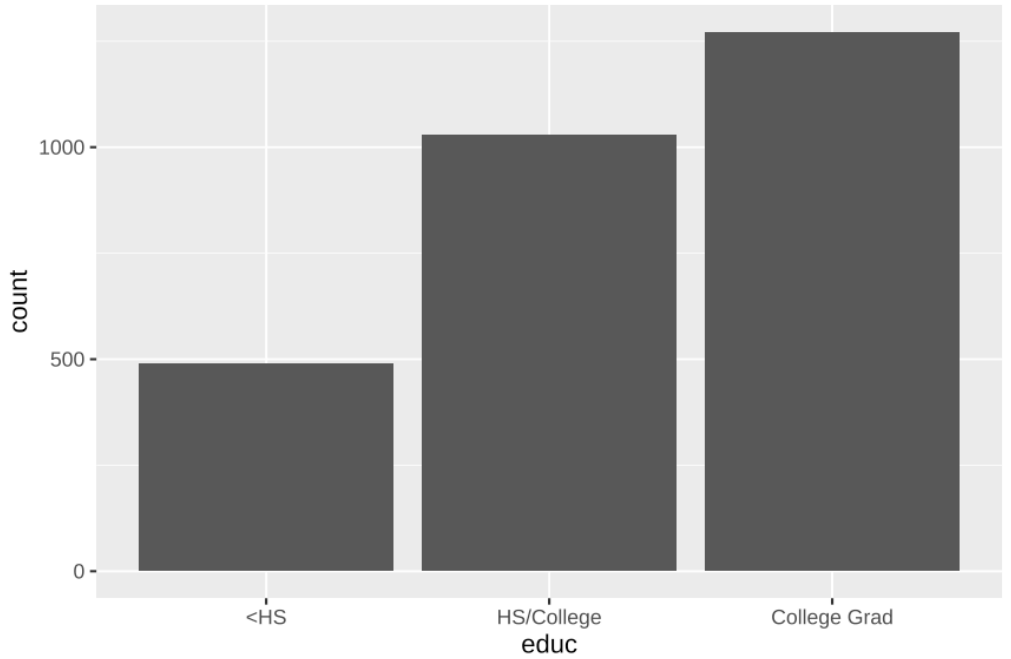
```
ggplot(ces19, aes(x=educ)) +
  geom_bar()
```



Bar Chart

R Python Stata

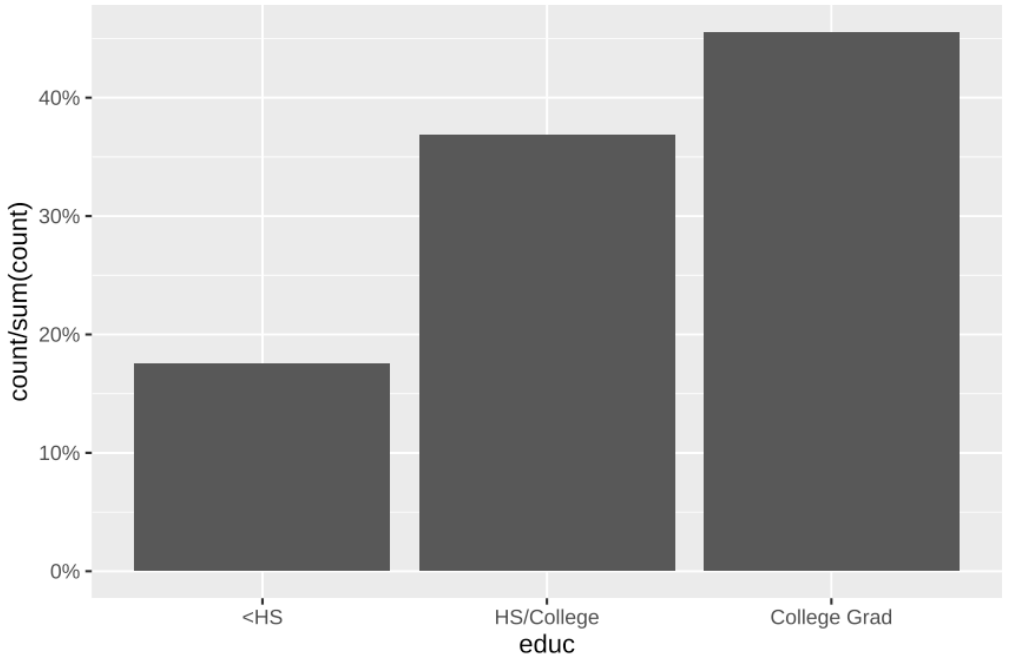
```
ces19 %>% filter(!is.na(educ)) %>%
ggplot(aes(x=educ)) +
  geom_bar()
```



Bar Chart

R | Python | Stata

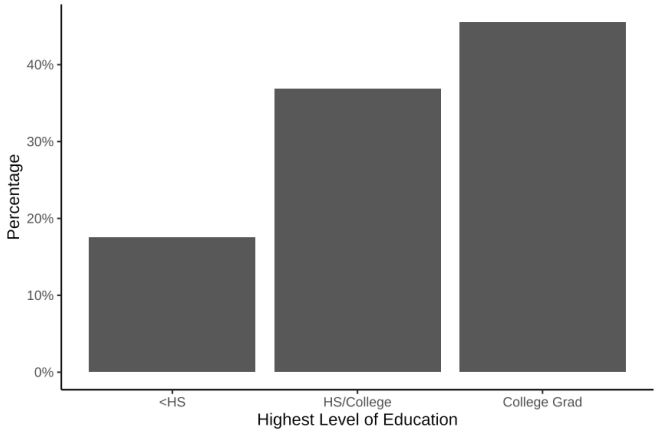
```
ces19 %>% filter(!is.na(educ)) %>%
  ggplot(aes(x=educ,
             y=after_stat(count/sum(count)))) +
    geom_bar() +
    scale_y_continuous(label=percent)
```



Bar Chart

R | Python | Stata

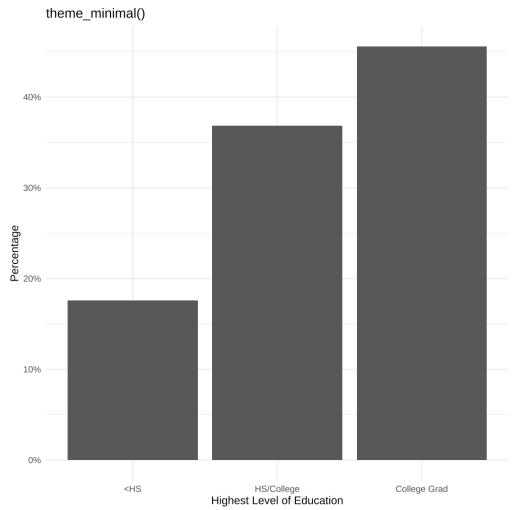
```
ces19 %>% filter(!is.na(educ)) %>%
  ggplot(aes(x=educ,
             y=after_stat(count/sum(count)))) +
    geom_bar() +
    scale_y_continuous(label=percent) +
    labs(x="Highest Level of Education",
       y="Percentage") +
    theme_classic()
```



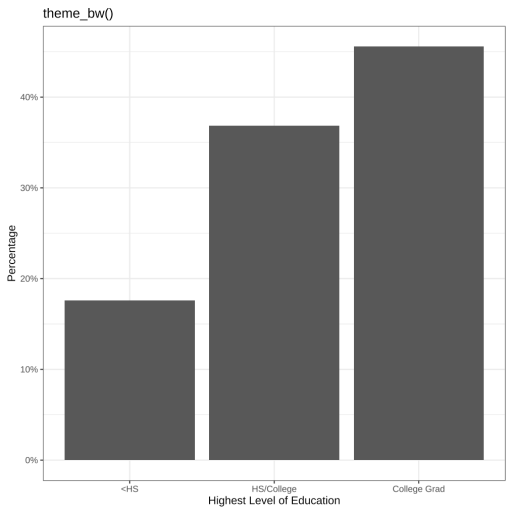
Some other themes:

```
g <- ces19 %>% filter(!is.na(educ)) %>%
  ggplot(aes(x=educ,
             y=..count../sum(..count..))) +
    geom_bar() +
    scale_y_continuous(label=
            label_percent(accuracy=2)) +
  labs(x="Highest Level of Education",
       y="Percentage")
```

```
g + theme_minimal() + ggtitle("theme_minimal()")
```



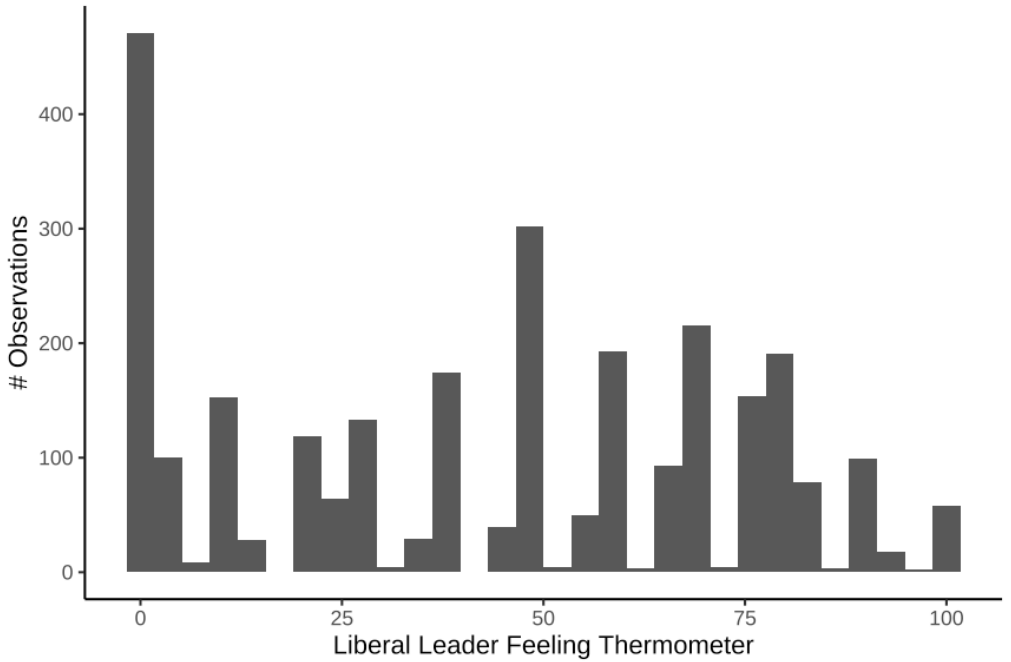

```
g + theme_bw() + ggtitle("theme_bw()")
```



Histogram

R Python Stata

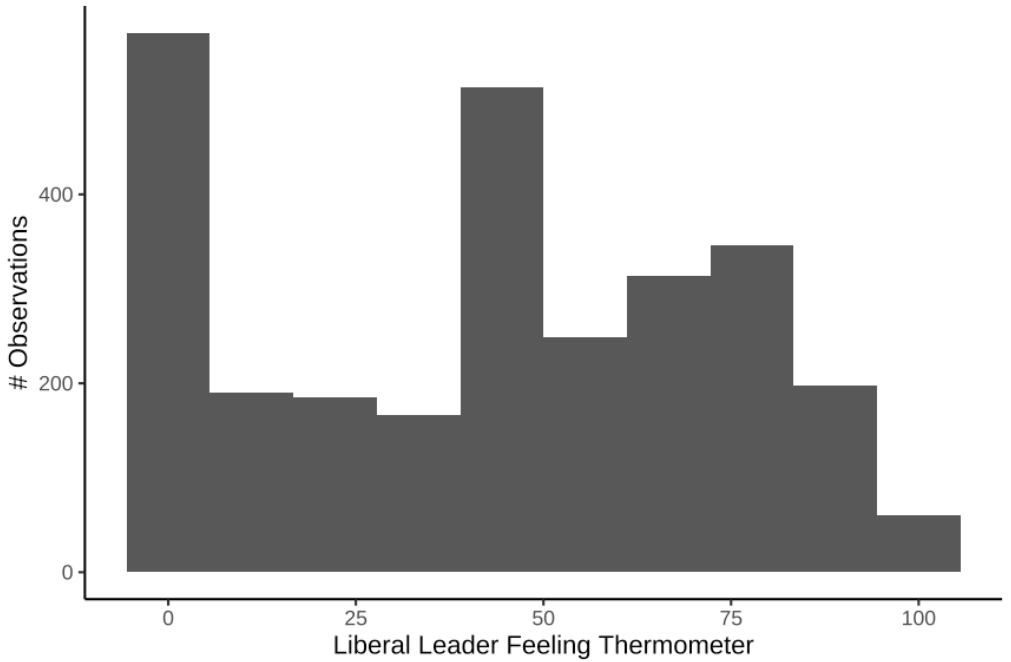
```
ggplot(ces19, aes(x=leader_lib)) +
  geom_histogram() +
  theme_classic() +
  labs(
    x="Liberal Leader Feeling Thermometer",
    y="# Observations")
```



Histogram

R Python Stata

```
ggplot(ces19, aes(x=leader_lib)) +
  geom_histogram(bins=10) +
  theme_classic() +
  labs(
    x="Liberal Leader Feeling Thermometer",
    y="# Observations")
```



Exercises

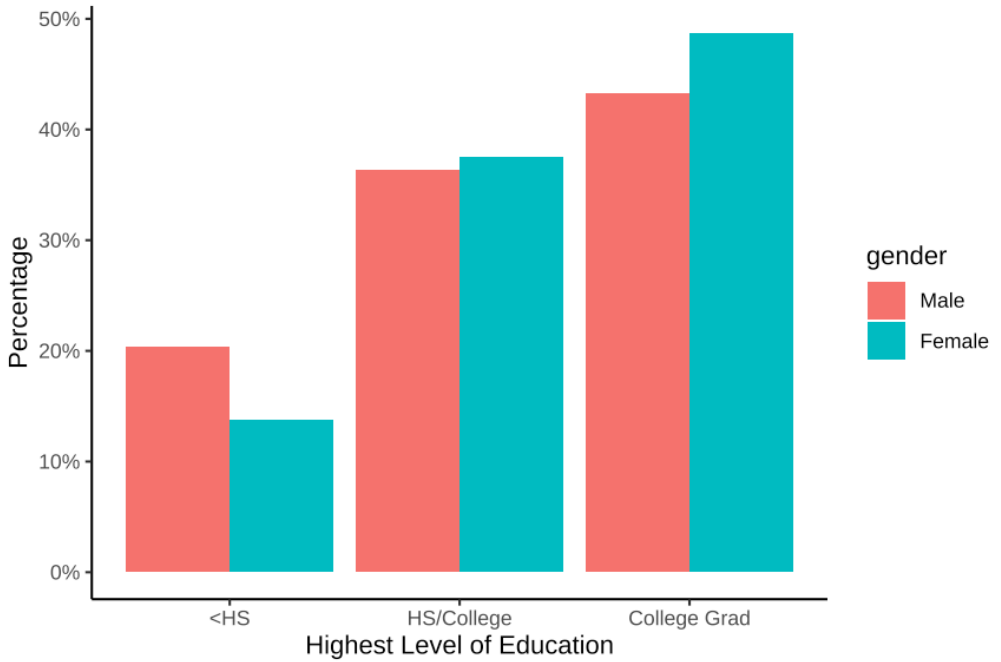
1. Using the gss data
    - Make a bar plot of `SRH_110`
    - Make a bar plot of `SRH_115`
    - Make a histogram of `resilience`

Grouped Bar Charts: Education by Gender

R | Python | Stata

```
ces19$gender <- factor(ces19$gender,
                       levels=c(1,5),
                       labels=c("Male","Female"))
```

```
ces19 %>%
  filter(!is.na(educ)) %>%
  group_by(gender, educ) %>%
  summarise(n = n()) %>%
  ungroup %>%
  group_by(gender) %>%
  mutate(prop = n/sum(n)) %>%
  ggplot(aes(x=educ,
             y=prop,
             fill=gender)) +
    geom_bar(position = position_dodge(),
             stat="identity") +
    scale_y_continuous(label=
            label_percent(accuracy=2)) +
    theme_classic() +
    labs(x="Highest Level of Education",
         y="Percentage")
```



Review

1. Reading in Data
2. Frequency Distributions/Summary Statistics
3. Graphs
    - Line Graph
    - Bar Plot
    - Histogram