POLSCI 9590: Methods I

Measures of Centre and Spread

Dave Armstrong

Videos

We covered a few different things in the videos:

1. Measures of Centre
    - Mean
    - Median
    - Mode
2. Measures of Spread
    - Range/IQR
    - Mean absolute deviation (MAD)
    - Variance/Standard Deviation
3. Z-scores (standard scores).

Videos

We covered a few different things in the videos:

1. Measures of Centre
    - Mean
    - Median
    - Mode
2. Measures of Spread
    - Range/IQR
    - Mean absolute deviation (MAD)
    - Variance/Standard Deviation
3. Z-scores (standard scores).

Questions?

Exercise 1

Question: What is the difference between people who love and hate Trudeau?

1. Make a new variable that is coded `"love"` for observations where `leader_lib` is greater than or equal to 90 and `"hate"` for observations where `leader_lib` is less than or equal to 10. All other observations should be missing.
2. Create the distribution of `educ`, `agegrp`, `market` and `relig` for these two groups.

Exercise 2

Question: What does the distribution of mental health look like for three groups of resilience.

1. Import the `gss16_can.dta` data set.
2. Use the `case_when()` function to create three groups of resilience using the $33rd$ and 67^{th} percentiles as the cutoffs.
3. Make a graph of `SRH_115` using this new resilience measure as the `facet` variable.

Setup

R Python Stata

```r
library(rio)
library(DAMisc)
library(uwo4419)
library(ggplot2)
library(dplyr)
library(tidyr)
```

Summary Statistics

R Python Stata

```
ces <- import("ces19.dta")
ces$educ <- factorize(ces$educ)
sumStats(ces, "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
```

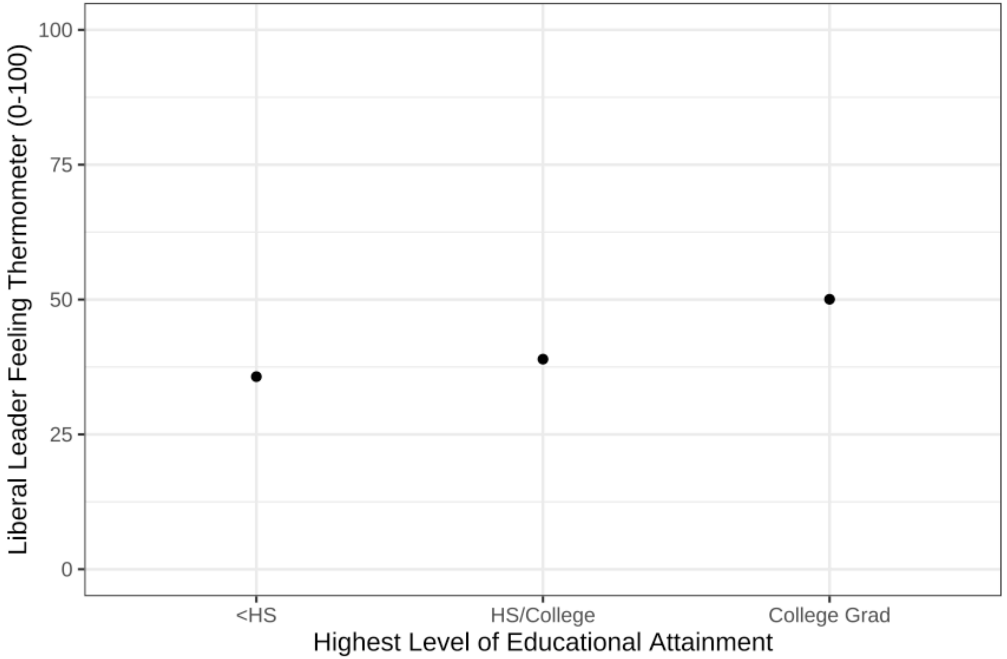
```
sumStats(ces, "leader_lib", byvar="educ")
```

```
## # 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
```

Plot

R Python Stata

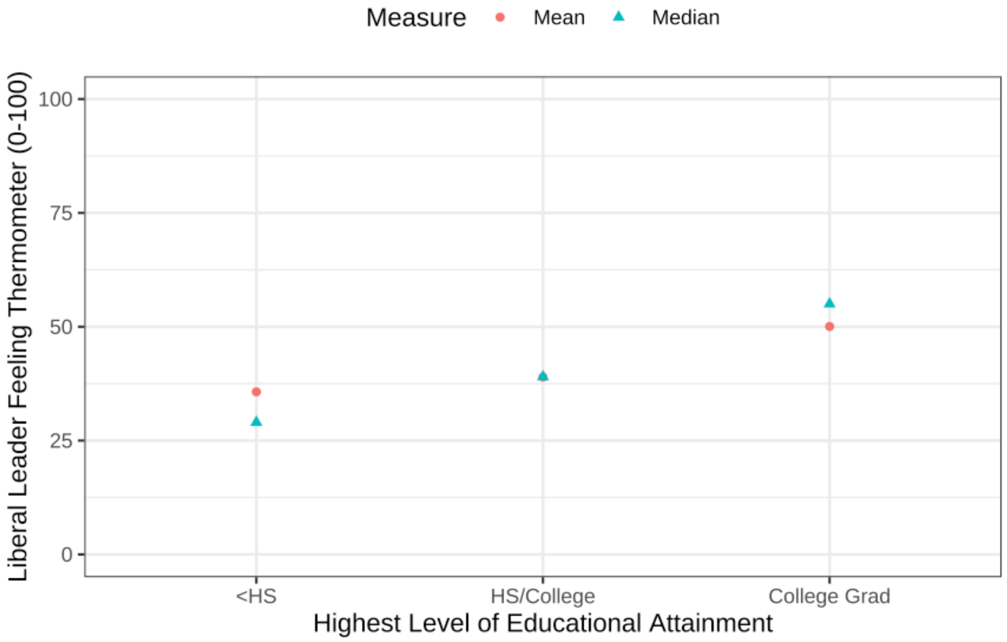
```
ces %>% filter(!is.na(educ)) %>%
ggplot(aes(x=educ, y=leader_lib)) +
  stat_summary(geom="point", fun=mean) +
  theme_bw() +
  labs(x="Highest Level of Educational Attainment",
       y = "Liberal Leader Feeling Thermometer (0-100)") +
  ylim(0,100)
```



Plot

R Python Stata

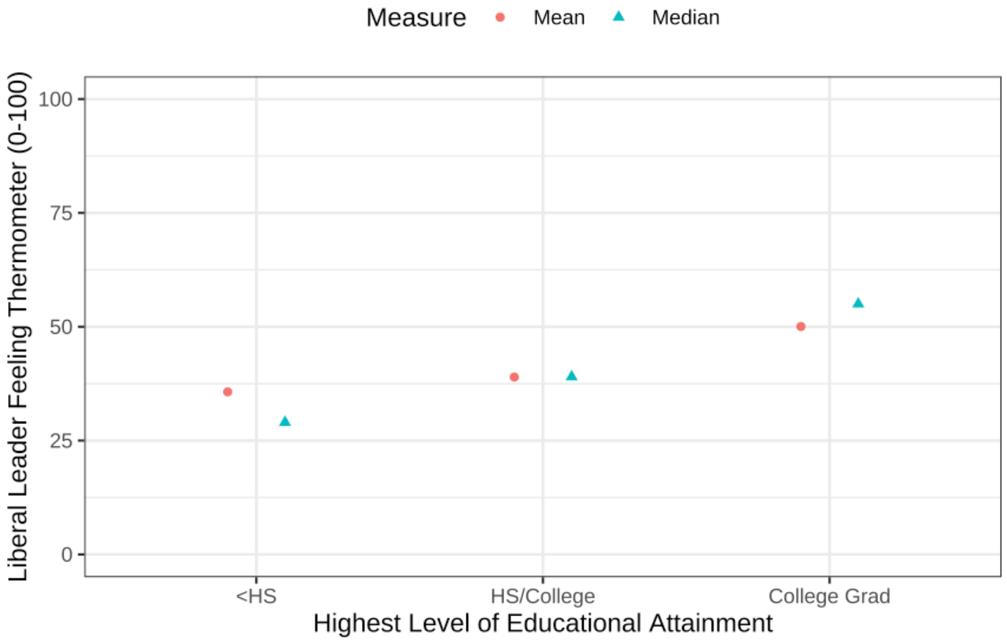
```r
ces %>% filter(!is.na(educ)) %>%
ggplot(aes(x=educ, y=leader_lib)) +
  stat_summary(aes(shape="Mean", colour="Mean"),
               geom="point",
               fun=mean) +
  stat_summary(aes(shape="Median", colour="Median"),
               geom="point",
               fun=median) +
  theme_bw() +
  theme(legend.position="top") +
  labs(x="Highest Level of Educational Attainment",
       y = "Liberal Leader Feeling Thermometer (0-100)",
       colour="Measure", shape="Measure") +
  ylim(0,100)
```



Plot

R   Python   Stata

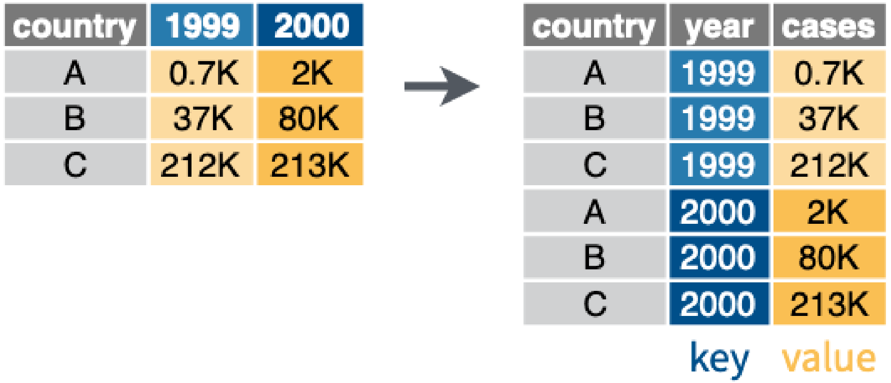
```
ces %>% filter(!is.na(educ)) %>%
ggplot(aes(x=educ, y=leader_lib)) +
  stat_summary(aes(shape="Mean", colour="Mean"),
               geom="point",
               fun=mean,
               position = position_nudge(x=-.1)) +
  stat_summary(aes(shape="Median", colour="Median"),
               geom="point",
               fun=median,
               position = position_nudge(x=.1)) +
  theme_bw() +
  theme(legend.position="top") +
  labs(x="Highest Level of Educational Attainment",
       y = "Liberal Leader Feeling Thermometer (0-100)",
       colour="Measure", shape="Measure") +
  ylim(0,100)
```



Exercise 3

Make the plot above, but for the `resilience` and the `SRH_110` variable from the GSS data we've been using.

Wide to Long



country	1999	2000
A	0.7K	2K
B	37K	80K
C	212K	213K

country	year	cases
A	1999	0.7K
B	1999	37K
C	1999	212K
A	2000	2K
B	2000	80K
C	2000	213K

R Python Stata

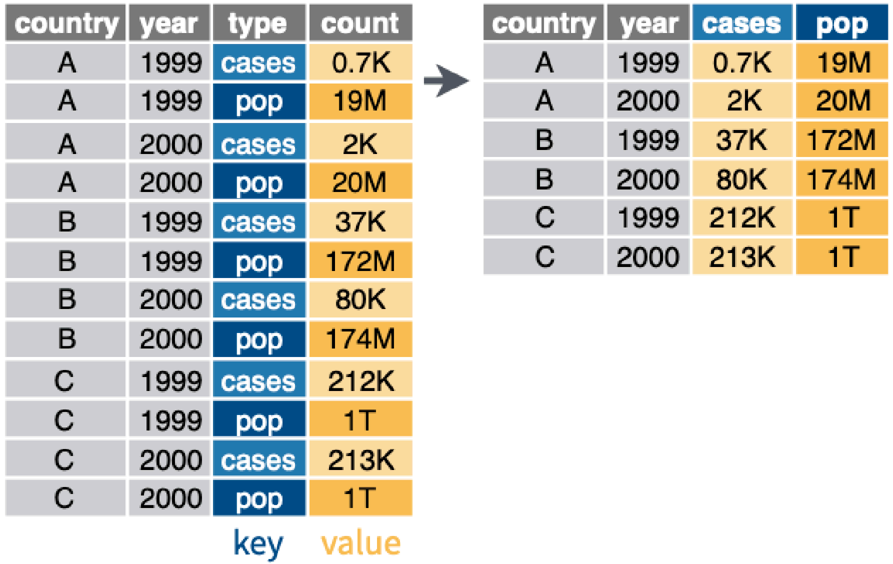
```
x <- tibble::tibble(
  country = c("A", "B", "C"),
  `1999` = 1:3,
  `2000` = 4:6)
xl <- pivot_longer(x, cols=`1999`:`2000`,
             names_to="year",
             values_to="cases")
x
```

```
## # A tibble: 3 × 3
##   country `1999` `2000`
##   <chr>    <int>  <int>
## 1 A            1      4
## 2 B            2      5
## 3 C            3      6
```

```
xl
```

```
## # A tibble: 6 × 3
##   country year  cases
##   <chr>   <chr> <int>
## 1 A       1999      1
## 2 A       2000      4
## 3 B       1999      2
## 4 B       2000      5
## 5 C       1999      3
## 6 C       2000      6
```

Long to Wide



country	year	type	count
A	1999	cases	0.7K
A	1999	pop	19M
A	2000	cases	2K
A	2000	pop	20M
B	1999	cases	37K
B	1999	pop	172M
B	2000	cases	80K
B	2000	pop	174M
C	1999	cases	212K
C	1999	pop	1T
C	2000	cases	213K
C	2000	pop	1T
		key	value

country	year	cases	pop
A	1999	0.7K	19M
A	2000	2K	20M
B	1999	37K	172M
B	2000	80K	174M
C	1999	212K	1T
C	2000	213K	1T

R

Python

Stata

```
xw <- xl %>% pivot_wider(names_from="year",
                         values_from="cases")
xl
```

```
## # A tibble: 6 × 3
##   country year  cases
##   <chr>   <chr> <int>
## 1 A       1999      1
## 2 A       2000      4
## 3 B       1999      2
## 4 B       2000      5
## 5 C       1999      3
## 6 C       2000      6
```

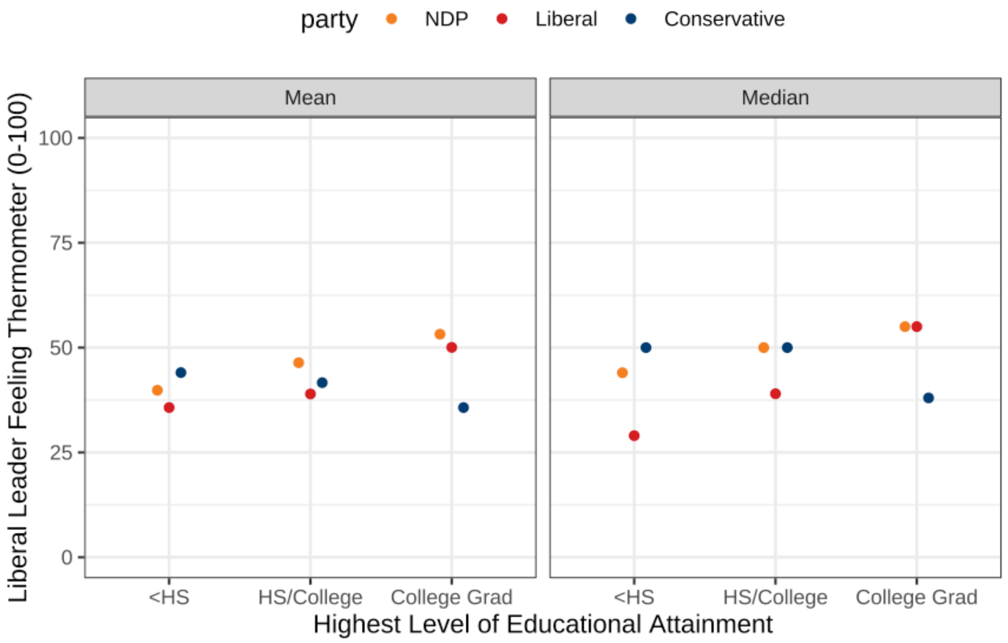
```
xw
```

```
## # A tibble: 3 × 3
##   country `1999` `2000`
##   <chr>    <int>  <int>
## 1 A            1      4
## 2 B            2      5
## 3 C            3      6
```

All Leaders

R   Python   Stata

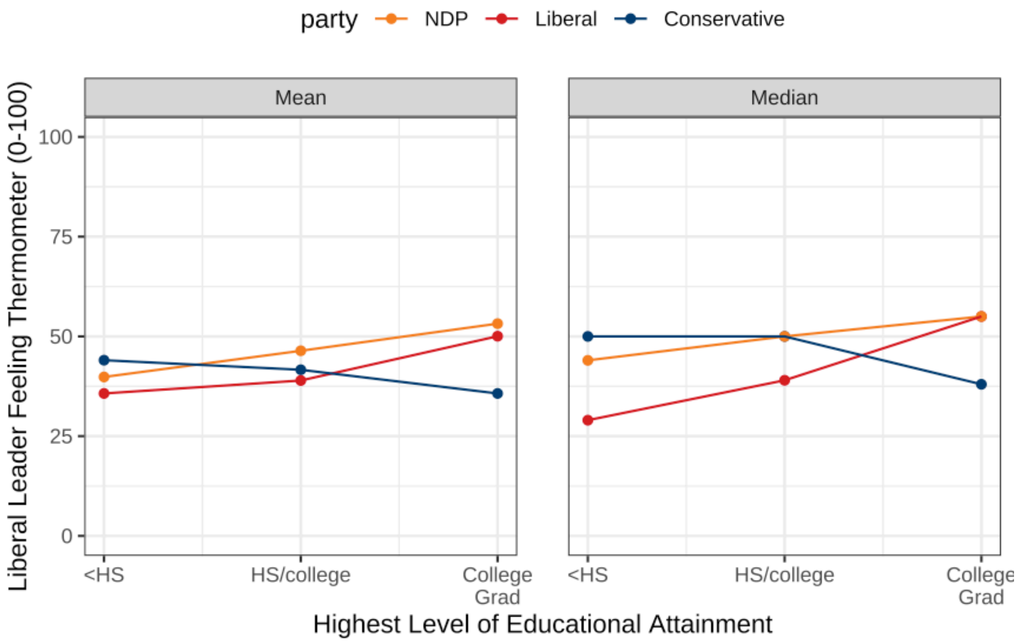
```
x <- ces %>%
  filter(!is.na(educ)) %>%
  group_by(educ) %>%
  summarise(across(starts_with("leader"),
                   list(Mean = ~mean(.x, na.rm=TRUE),
                        Median = ~median(.x, na.rm=TRUE)))) %>%
  pivot_longer(-educ,
               names_pattern="leader_(.*)_(.*)",
               names_to = c("party", "measure"),
               values_to="val") %>%
  mutate(party = factor(party,
                        levels=c("ndp", "lib", "con"),
                        labels=c("NDP", "Liberal", "Conservative'

ggplot(x, aes(x=educ, y=val,
              colour=party)) +
  geom_point(position=position_dodge(width=.25)) +
  theme_bw() +
  facet_wrap(~measure, ncol=2) +
  scale_colour_manual(values=c( "#F58220", "#d71920", "#003F72"))
  theme(legend.position="top") +
  labs(x="Highest Level of Educational Attainment",
       y = "Liberal Leader Feeling Thermometer (0-100)") +
  ylim(0,100)
```



With lines

R Python Stata

```r
ggplot(x, aes(x=as.numeric(educ), y=val,
              colour=party)) +
  geom_point() +
  geom_line() +
  theme_bw() +
  facet_wrap(~measure, ncol=2) +
  scale_colour_manual(values=c( "#F58220", "#d71920", "#003F72"))
  scale_x_continuous(breaks = 1:3, labels=c("<HS", "HS/college",
  theme(legend.position="top",
        panel.spacing=unit(1.5, "lines")) +
  labs(x="Highest Level of Educational Attainment",
       y = "Liberal Leader Feeling Thermometer (0-100)") +
  ylim(0,100)
```



Exercise 4

1. Make a two-level factor that codes bad mental health (`Fair` and `Poor` on `SRH_115`) and good mental health (`Excellent`, `Very good` and `Good` on `SRH_115`).
2. Make a graph that shows the mean and median of `resilience` for `SRH_110` with different colors for good and bad mental health.

Plot the mean and median of resilience for each of the different groups of `SRH_110` from the GSS data.