

Gov 50: 9. Summarizing Data

Matthew Blackwell

Harvard University

Roadmap

1. Descriptive Statistics
2. Missing data
3. Proportion tables

1/ Descriptive Statistics

Lots of data

```
library(tidyverse)
library(gapminder)
gapminder
```

```
## # A tibble: 1,704 x 6
##   country      continent  year lifeExp      pop gdpPercap
##   <fct>        <fct>    <int> <dbl>    <int>    <dbl>
## 1 Afghanistan Asia      1952  28.8  8425333    779.
## 2 Afghanistan Asia      1957  30.3  9240934    821.
## 3 Afghanistan Asia      1962  32.0 10267083    853.
## 4 Afghanistan Asia      1967  34.0 11537966    836.
## 5 Afghanistan Asia      1972  36.1 13079460    740.
## 6 Afghanistan Asia      1977  38.4 14880372    786.
## 7 Afghanistan Asia      1982  39.9 12881816    978.
## 8 Afghanistan Asia      1987  40.8 13867957    852.
## 9 Afghanistan Asia      1992  41.7 16317921    649.
## 10 Afghanistan Asia      1997  41.8 22227415    635.
## # i 1,694 more rows
```

Lots and lots of data

```
head(gapminder$gdpPercap, n = 200)
```

```
## [1] 779 821 853 836 740 786 978 852 649
## [10] 635 727 975 1601 1942 2313 2760 3313 3533
## [19] 3631 3739 2497 3193 4604 5937 2449 3014 2551
## [28] 3247 4183 4910 5745 5681 5023 4797 5288 6223
## [37] 3521 3828 4269 5523 5473 3009 2757 2430 2628
## [46] 2277 2773 4797 5911 6857 7133 8053 9443 10079
## [55] 8998 9140 9308 10967 8798 12779 10040 10950 12217
## [64] 14526 16789 18334 19477 21889 23425 26998 30688 34435
## [73] 6137 8843 10751 12835 16662 19749 21597 23688 27042
## [82] 29096 32418 36126 9867 11636 12753 14805 18269 19340
## [91] 19211 18524 19036 20292 23404 29796 684 662 686
## [100] 721 630 660 677 752 838 973 1136 1391
## [109] 8343 9715 10991 13149 16672 19118 20980 22526 25576
## [118] 27561 30486 33693 1063 960 949 1036 1086 1029
## [127] 1278 1226 1191 1233 1373 1441 2677 2128 2181
## [136] 2587 2980 3548 3157 2754 2962 3326 3413 3822
## [145] 974 1354 1710 2172 2860 3528 4127 4314 2547
## [154] 4766 6019 7446 851 918 984 1215 2264 3215
## [163] 4551 6206 7954 8647 11004 12570 2109 2487 3337
## [172] 3430 4986 6660 7031 7807 6950 7958 8131 9066
```

How to summarize data

- How should we summarize the wages data? Many possibilities!
 - Up to now: focus on **averages** or means of variables.
- Two salient features of a variable that we want to know:
 - **Central tendency**: where is the middle/typical/average value.
 - **Spread** around the center: are all values to the center or spread out?

Center of the data

- “Center” of the data: typical/average value.
- **Mean:** sum of the values divided by the number of observations

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$$

- **Median:**

$$\text{median} = \begin{cases} \text{middle value} & \text{if number of entries is odd} \\ \frac{\text{sum of two middle values}}{2} & \text{if number of entries is even} \end{cases}$$

- In **R**: `mean()` and `median()`.

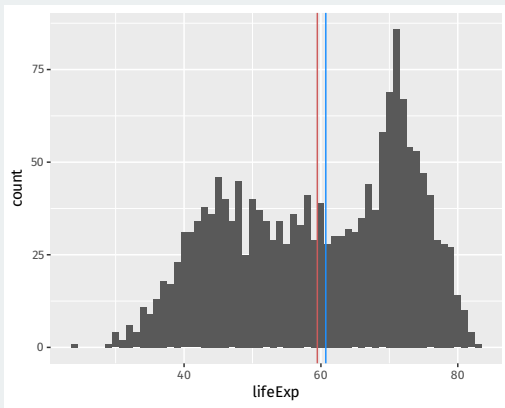
Mean vs median

- Median more robust to **outliers**:
 - Example 1: data = {0, 1, 2, 3, 5}. Mean? Median?

 - Example 2: data = {0, 1, 2, 3, 100}. Mean? Median?

- What does Mark Zuckerberg do to the mean vs median income?

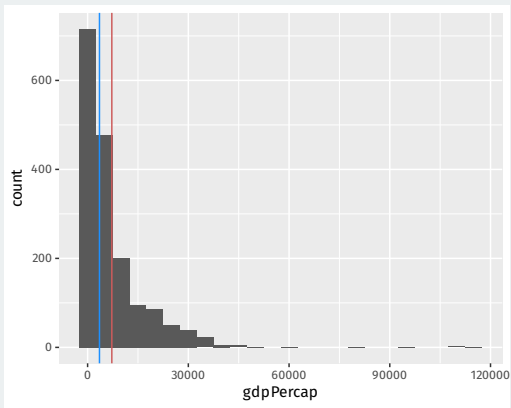

```
ggplot(gapminder, aes(x = lifeExp)) +  
  geom_histogram(binwidth = 1) +  
  geom_vline(aes(xintercept = mean(lifeExp)), color = "indianred") +  
  geom_vline(aes(xintercept = median(lifeExp)), color = "dodgerblue")
```



```
summary(gapminder$lifeExp)
```

##	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
##	23.6	48.2	60.7	59.5	70.8	82.6

```
ggplot(gapminder, aes(x = gdpPerCap)) +  
  geom_histogram(binwidth = 5000) +  
  geom_vline(aes(xintercept = mean(gdpPerCap)), color = "indianred") +  
  geom_vline(aes(xintercept = median(gdpPerCap)), color = "dodgerblue")
```

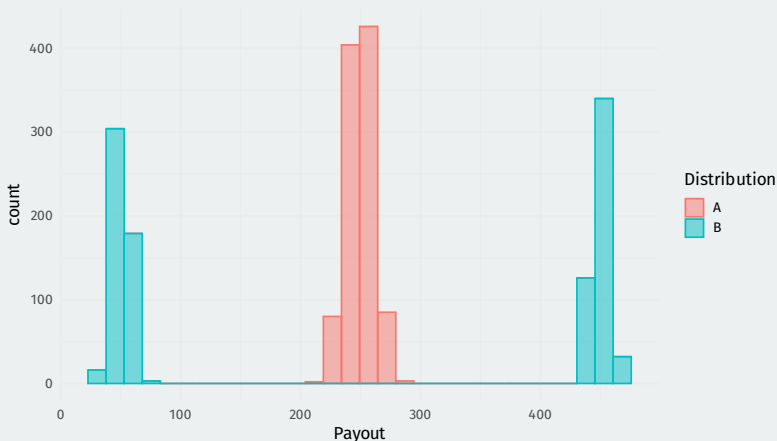


```
summary(gapminder$gdpPerCap)
```

##	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
##	241	1202	3532	7215	9325	113523

Which distribution would you prefer?

Lottery where we randomly draw one value from A or B:



They have the same mean, so why do we care about the difference? **Spread!!**

Spread of the data

- Are the values of the variable close to the center?
- **Range:** $[\min(X), \max(X)]$
- **Quantile** (quartile, percentile, etc): divide data into equal sized groups.
 - 25th percentile = lower quartile (25% of the data below this value)
 - 50th percentile = median (50% of the data below this value)
 - 75th percentile = upper quartile (75% of the data below this value)
- **Interquartile range** (IQR): a measure of variability
 - How spread out is the middle half of the data?
 - Is most of the data really close to the median or are the values spread out?
- **R** function: `range()`, `summary()`, `IQR()`

Standard deviation

- **Standard deviation:** On average, how far away are data points from the mean?

$$\text{standard deviation} = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2}$$

- Steps:
 1. Subtract each data point by the mean.
 2. Square each resulting difference.
 3. Take the sum of these values
 4. Divide by $n - 1$ (or n , doesn't matter much)
 5. Take the square root.
- **Variance** = standard deviation²
- Why not just take the average deviations from mean without squaring?

2/ Missing data

Missing data

- **Nonresponse:** respondent can't or won't answer question.
 - Sensitive questions \rightsquigarrow **social desirability bias**
 - Some countries lack official statistics like unemployment.
 - Leads to missing data.
- Missing data in R: a special value NA
- Have already seen how to use `na.rm = TRUE`

CCES data

```
library(gov50data)
cces_2020
```

```
## # A tibble: 51,551 x 6
##   gender race  educ          pid3 turnout_self pres_vote
##   <fct> <fct> <fct>          <fct>         <dbl> <fct>
## 1 Male   White 2-year      Repu~             1 Donald J~
## 2 Female White Post-grad    Demo~            NA <NA>
## 3 Female White 4-year      Inde~             1 Joe Bide~
## 4 Female White 4-year      Demo~             1 Joe Bide~
## 5 Male   White 4-year      Inde~             1 Other
## 6 Male   White Some college Repu~             1 Donald J~
## 7 Male   Black Some college Not ~            NA <NA>
## 8 Female White Some college Inde~             1 Donald J~
## 9 Female White High school gr~ Repu~             1 Donald J~
## 10 Female White 4-year      Demo~             1 Joe Bide~
## # i 51,541 more rows
```


drop_na() to remove rows with missing values

```
cces_2020 |>  
  drop_na()
```

```
## # A tibble: 45,651 x 6  
##   gender race educ pid3 turnout_self pres_vote  
##   <fct> <fct> <fct> <fct> <dbl> <fct>  
## 1 Male White 2-year Repu~ 1 Donald J~  
## 2 Female White 4-year Inde~ 1 Joe Bide~  
## 3 Female White 4-year Demo~ 1 Joe Bide~  
## 4 Male White 4-year Inde~ 1 Other  
## 5 Male White Some college Repu~ 1 Donald J~  
## 6 Female White Some college Inde~ 1 Donald J~  
## 7 Female White High school gr~ Repu~ 1 Donald J~  
## 8 Female White 4-year Demo~ 1 Joe Bide~  
## 9 Female White 4-year Demo~ 1 Joe Bide~  
## 10 Female White 4-year Demo~ 1 Joe Bide~  
## # i 45,641 more rows
```

Drop rows based on certain variables

```
cces_2020 |>  
  dim_desc()
```

```
## [1] "[51,551 x 6]"
```

```
cces_2020 |>  
  drop_na() |>  
  dim_desc()
```

```
## [1] "[45,651 x 6]"
```

```
cces_2020 |>  
  drop_na(turnout_self) |>  
  dim_desc()
```

```
## [1] "[48,462 x 6]"
```

Available-case vs complete-case analysis

Available-case analysis: use the data you have for that variable:

```
cces_2020 |>
  summarize(mean(turnout_self, na.rm = TRUE)) |>
  pull()
```

```
## [1] 0.942
```

Complete-case analysis: only use units that have data on all variables

```
cces_2020 |>
  drop_na() |>
  summarize(mean(turnout_self)) |>
  pull()
```

```
## [1] 0.999
```

(also called **listwise deletion**)

is.na() to detect missingness

Trying to detect missingness with == doesn't work:

```
c(5, 6, NA, 0) == NA
```

```
## [1] NA NA NA NA
```

Use is.na() instead:

```
is.na(c(5, 6, NA, 0))
```

```
## [1] FALSE FALSE TRUE FALSE
```

Can use sum() or mean() on this to get number/proportion missing:

```
sum(is.na(c(5, 6, NA, 0)))
```

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

Nonresponse bias

Nonresponse can create bias if lower turnout \Leftrightarrow more non-response:

```
cces_2020 |>
  group_by(pid3) |>
  summarize(
    mean_turnout = mean(turnout_self, na.rm = TRUE),
    missing_turnout = mean(is.na(turnout_self))
  )
```

```
## # A tibble: 5 x 3
##   pid3      mean_turnout missing_turnout
##   <fct>      <dbl>          <dbl>
## 1 Democrat    0.963            0.0280
## 2 Republican  0.953            0.0403
## 3 Independent 0.924            0.0718
## 4 Other       0.957            0.0709
## 5 Not sure    0.630            0.431
```

3/ Proportion tables

Review of getting counts

First, let's review how to get counts:

```
cces_2020 |>
  group_by(pres_vote) |>
  summarize(n = n())
```

```
## # A tibble: 7 x 2
##   pres_vote          n
##   <fct>          <int>
## 1 Joe Biden (Democrat) 26188
## 2 Donald J. Trump (Republican) 17702
## 3 Other              1458
## 4 I did not vote in this race   100
## 5 I did not vote             13
## 6 Not sure                 190
## 7 <NA>                    5900
```

First attempt to create proportions

```
cces_2020 |>
  group_by(pres_vote) |>
  summarize(prop = n() / sum(n()))
```

```
## # A tibble: 7 x 2
##   pres_vote                prop
##   <fct>                   <dbl>
## 1 Joe Biden (Democrat)      1
## 2 Donald J. Trump (Republican) 1
## 3 Other                    1
## 4 I did not vote in this race 1
## 5 I did not vote          1
## 6 Not sure                 1
## 7 <NA>                    1
```

Inside `summarize()` all operations are done within groups!

Mutate after summarizing

```
cces_2020 |>
  group_by(pres_vote) |>
  summarize(n = n()) |>
  mutate(prop = n / sum(n))
```

```
## # A tibble: 7 x 3
##   pres_vote          n      prop
##   <fct>          <int>   <dbl>
## 1 Joe Biden (Democrat) 26188 0.508
## 2 Donald J. Trump (Republican) 17702 0.343
## 3 Other              1458 0.0283
## 4 I did not vote in this race   100 0.00194
## 5 I did not vote                13 0.000252
## 6 Not sure                   190 0.00369
## 7 <NA>                       5900 0.114
```

Grouping is silently dropped after `summarize()`

Multiple grouping variables

What happens with multiple grouping variables

```
cces_2020 |>
  filter(pres_vote %in% c("Joe Biden (Democrat)",
                        "Donald J. Trump (Republican)")) |>
  group_by(pid3, pres_vote) |>
  summarize(n = n()) |>
  mutate(prop = n / sum(n))
```

```
## # A tibble: 10 x 4
## # Groups:   pid3 [5]
##   pid3      pres_vote          n  prop
##   <fct>    <fct>          <int> <dbl>
## 1 Democrat Joe Biden (Democrat)    17649 0.968
## 2 Democrat Donald J. Trump (Republican)   581 0.0319
## 3 Republican Joe Biden (Democrat)     856 0.0712
## 4 Republican Donald J. Trump (Republican) 11164 0.929
## 5 Independent Joe Biden (Democrat)     6601 0.571
## 6 Independent Donald J. Trump (Republican) 4951 0.429
## 7 Other     Joe Biden (Democrat)     735 0.487
## 8 Other     Donald J. Trump (Republican)   774 0.513
## 9 Not sure  Joe Biden (Democrat)     347 0.599
## 10 Not sure Donald J. Trump (Republican)   232 0.401
```

With multiple grouping variables, `summarize()` drops the last one.

Proportions are in terms of the the remaining group(s).

Dropping all groups

If we want the proportion of all rows, need to drop all groups.

```
cces_2020 |>
  filter(pres_vote %in% c("Joe Biden (Democrat)",
                        "Donald J. Trump (Republican)")) |>
  group_by(pid3, pres_vote) |>
  summarize(n = n(), .groups = "drop") |>
  mutate(prop = n / sum(n))
```

```
## # A tibble: 10 x 4
##   pid3      pres_vote          n    prop
##   <fct>    <fct>          <int> <dbl>
## 1 Democrat Joe Biden (Democrat)  17649 0.402
## 2 Democrat Donald J. Trump (Republican)   581 0.0132
## 3 Republican Joe Biden (Democrat)     856 0.0195
## 4 Republican Donald J. Trump (Republican) 11164 0.254
## 5 Independent Joe Biden (Democrat)     6601 0.150
## 6 Independent Donald J. Trump (Republican)  4951 0.113
## 7 Other      Joe Biden (Democrat)     735 0.0167
## 8 Other      Donald J. Trump (Republican)   774 0.0176
## 9 Not sure   Joe Biden (Democrat)     347 0.00791
## 10 Not sure  Donald J. Trump (Republican)  232 0.00529
```