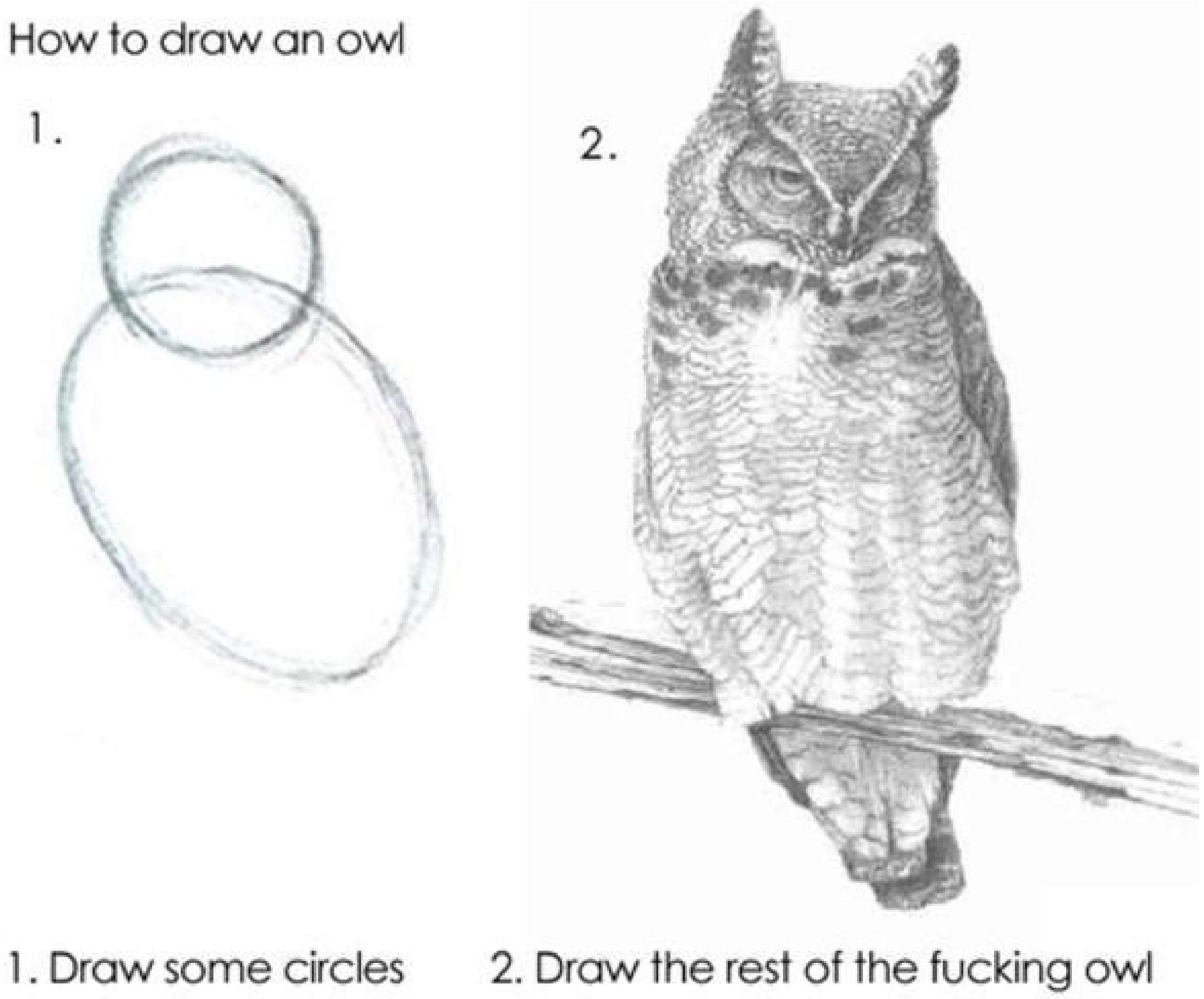
Averages, Expectation, Aggregation

Understanding Political Numbers

Feb 18, 2019

Learning ggplot



(Spooky voice) *Statistiiiiiiics*

This week: "the signal and the noise"

- Today: Means
- Wednesday: Variance

In section: major tidyverse functions

Questions about exercise 1?

Major functions in the tidyverse

a.k.a. "verbs." They modify and return *data frames*

Function	Operation
arrange()	Sort data frame along variable(s)
select()	Choose variables (columns) from a data frame
filter()	Choose cases (rows) from a data frame
mutate()	Create or modify variables
count()	Tabulate variable(s) in a data frame
summarize()	Calculate summary statistics from a data frame
group_by()	Implicitly partition a data frame along variable(s)

Arrange

```r
# load tidyverse and gapminder data
library("tidyverse")
library("gapminder")

# With tidyverse verbs, the first argument is the data frame
# Sort by year and then by continent.
arrange(gapminder, year, continent)
```

```
## # A tibble: 1,704 x 6
##    country                  continent  year lifeExp     pop gdpPercap
##    <fct>                    <fct>     <int>   <dbl>   <int>     <dbl>
##  1 Algeria                  Africa     1952    43.1 9279525     2449.
##  2 Angola                   Africa     1952    30.0 4232095     3521.
##  3 Benin                    Africa     1952    38.2 1738315     1063.
##  4 Botswana                 Africa     1952    47.6  442308      851.
##  5 Burkina Faso             Africa     1952    32.0 4469979      543.
##  6 Burundi                  Africa     1952    39.0 2445618      339.
##  7 Cameroon                 Africa     1952    38.5 5009067     1173.
##  8 Central African Republic Africa     1952    35.5 1291695     1071.
##  9 Chad                     Africa     1952    38.1 2682462     1179.
## 10 Comoros                  Africa     1952    40.7  153936     1103.
## # … with 1,694 more rows
```

Select variables

```r
# Which variables do I want to keep?
# Again, first arg is the data frame name
# (notice lack of $)
select(gapminder, country, year, gdpPercap)
```

```
## # A tibble: 1,704 x 3
##    country      year gdpPercap
##    <fct>       <int>     <dbl>
##  1 Afghanistan  1952      779.
##  2 Afghanistan  1957      821.
##  3 Afghanistan  1962      853.
##  4 Afghanistan  1967      836.
##  5 Afghanistan  1972      740.
##  6 Afghanistan  1977      786.
##  7 Afghanistan  1982      978.
##  8 Afghanistan  1987      852.
##  9 Afghanistan  1992      649.
## 10 Afghanistan  1997      635.
## # … with 1,694 more rows
```

Filter observations

```r
# Which cases (rows) do I want to keep?
# filter(dataset, logical test)
# keep rows where test result is TRUE
filter(gapminder, country == "United States")
```

Logical operators:

- `==` means "is equal to"
- `!=` means "not equal to"
- `>` and `<` mean "greater/less than"
- `>=` and `<=` are "greater than/less than or equal to"

Combine logical tests with `&` (and) or `|` (or)

- `filter(gapminder, country == "United States" & year > 2000)`

```
## # A tibble: 12 x 6
##    country       continent  year lifeExp       pop gdpPercap
##    <fct>         <fct>     <int>   <dbl>     <int>     <dbl>
##  1 United States Americas   1952    68.4 157553000    13990.
##  2 United States Americas   1957    69.5 171984000    14847.
##  3 United States Americas   1962    70.2 186538000    16173.
##  4 United States Americas   1967    70.8 198712000    19530.
##  5 United States Americas   1972    71.3 209896000    21806.
##  6 United States Americas   1977    73.4 220239000    24073.
##  7 United States Americas   1982    74.6 232187835    25010.
##  8 United States Americas   1987    75.0 242803533    29884.
##  9 United States Americas   1992    76.1 256894189    32004.
## 10 United States Americas   1997    76.8 272911760    35767.
## 11 United States Americas   2002    77.3 287675526    39097.
## 12 United States Americas   2007    78.2 301139947    42952.
```

Create variables with "mutate"

```
# mutate(dataframe, new_variable = (whatever you want))
mutate(gapminder,
       gdp = gdpPercap * pop)
```

```
## # A tibble: 1,704 x 7
##    country     continent  year lifeExp      pop gdpPercap          gdp
##    <fct>       <fct>     <int>   <dbl>    <int>     <dbl>        <dbl>
##  1 Afghanistan Asia       1952    28.8  8425333      779.  6567086330.
##  2 Afghanistan Asia       1957    30.3  9240934      821.  7585448670.
##  3 Afghanistan Asia       1962    32.0 10267083      853.  8758855797.
##  4 Afghanistan Asia       1967    34.0 11537966      836.  9648014150.
##  5 Afghanistan Asia       1972    36.1 13079460      740.  9678553274.
##  6 Afghanistan Asia       1977    38.4 14880372      786. 11697659231.
##  7 Afghanistan Asia       1982    39.9 12881816      978. 12598563401.
##  8 Afghanistan Asia       1987    40.8 13867957      852. 11820990309.
##  9 Afghanistan Asia       1992    41.7 16317921      649. 10595901589.
## 10 Afghanistan Asia       1997    41.8 22227415      635. 14121995875.
## # … with 1,694 more rows
```

Count (or tabulate)

```
# tabulate variable(s) with count().
# Again... result is a DATA FRAME
count(gapminder, continent, year)
```

```
## # A tibble: 60 x 3
##    continent  year     n
##    <fct>     <int> <int>
##  1 Africa     1952    52
##  2 Africa     1957    52
##  3 Africa     1962    52
##  4 Africa     1967    52
##  5 Africa     1972    52
##  6 Africa     1977    52
##  7 Africa     1982    52
##  8 Africa     1987    52
##  9 Africa     1992    52
## 10 Africa     1997    52
## # … with 50 more rows
```

Summarize variables

```r
# New data frame of summary calculations
# Use na.rm = TRUE to skip missing values when calculating summary stats
summarize(gapminder,
          mean_lifeexp = mean(lifeExp),
          min_lifeexp = min(lifeExp),
          max_lifeexp = max(lifeExp, na.rm = TRUE))
```

```
## # A tibble: 1 x 3
##   mean_lifeexp min_lifeexp max_lifeexp
##          <dbl>       <dbl>       <dbl>
## 1         59.5        23.6        82.6
```

Group data by variables

```
# partition data into groups. Pretty benign when used alone
group_by(gapminder, continent)
```

```
## # A tibble: 1,704 x 6
## # Groups:   continent [5]
##    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.
## # … with 1,694 more rows
```

Group and summarize

```r
# the `<-` scans the next line
gap_by_continent <-
  group_by(gapminder, continent)

summarize(gap_by_continent,
          mean_life = mean(lifeExp))
```

```
## # A tibble: 5 x 2
##   continent mean_life
##   <fct>         <dbl>
## 1 Africa         48.9
## 2 Americas       64.7
## 3 Asia           60.1
## 4 Europe         71.9
## 5 Oceania        74.3
```

```r
# Because result of group_by() is a data frame,
#  you could pass result directly to summarize
summarize(group_by(gapminder, continent),
          mean_life = mean(lifeExp))
```

```
## # A tibble: 5 x 2
##   continent mean_life
##   <fct>         <dbl>
## 1 Africa         48.9
## 2 Americas       64.7
## 3 Asia           60.1
## 4 Europe         71.9
## 5 Oceania        74.3
```

Averages

Question: do women vote for Democrats more than men do?

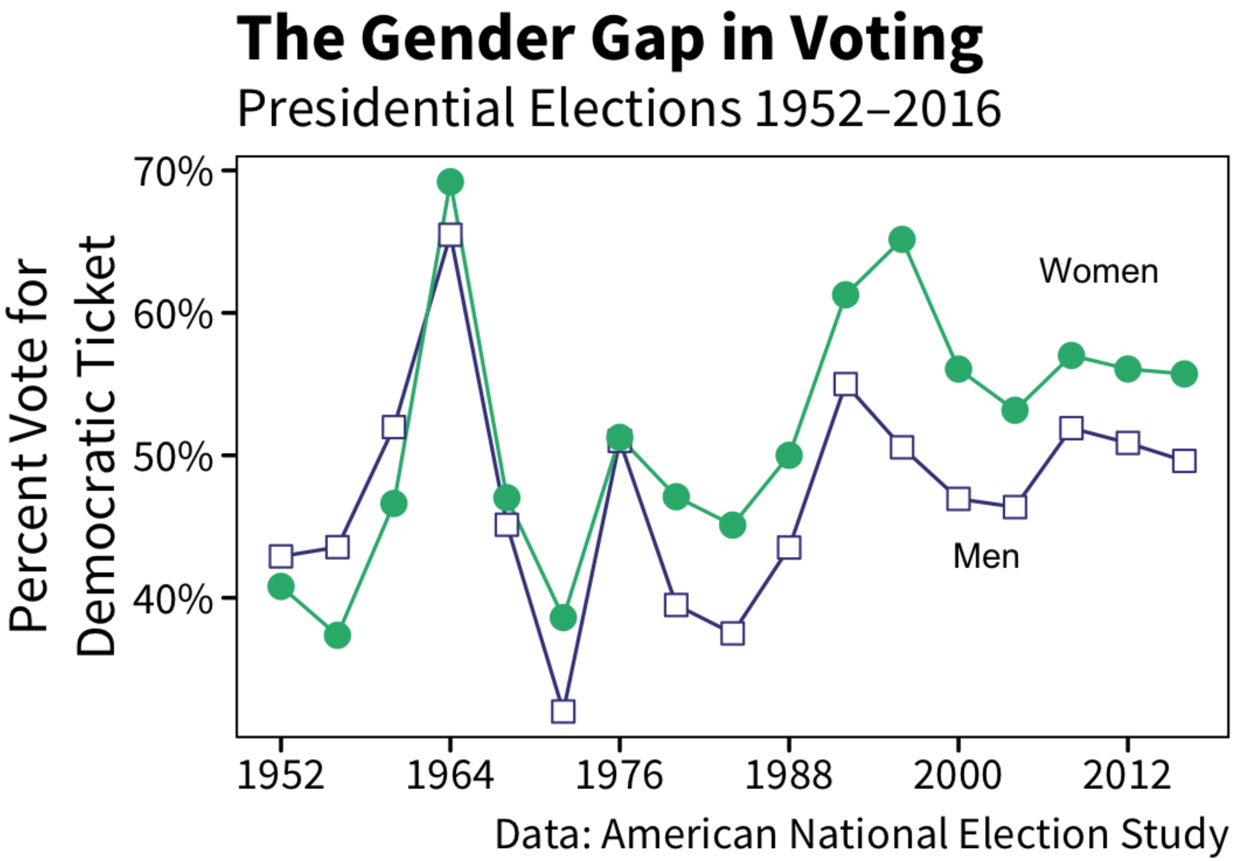
Break the question down:

1. What's the *average* rate of Democratic voting among women?
2. Among men?
3. How different are they?

Question: do women vote for Democrats more than men do?

Break the question down:

1. What's the *average* rate of Democratic voting among women?
2. Among men?
3. How different are they?



Question: is voter turnout higher among older voters?

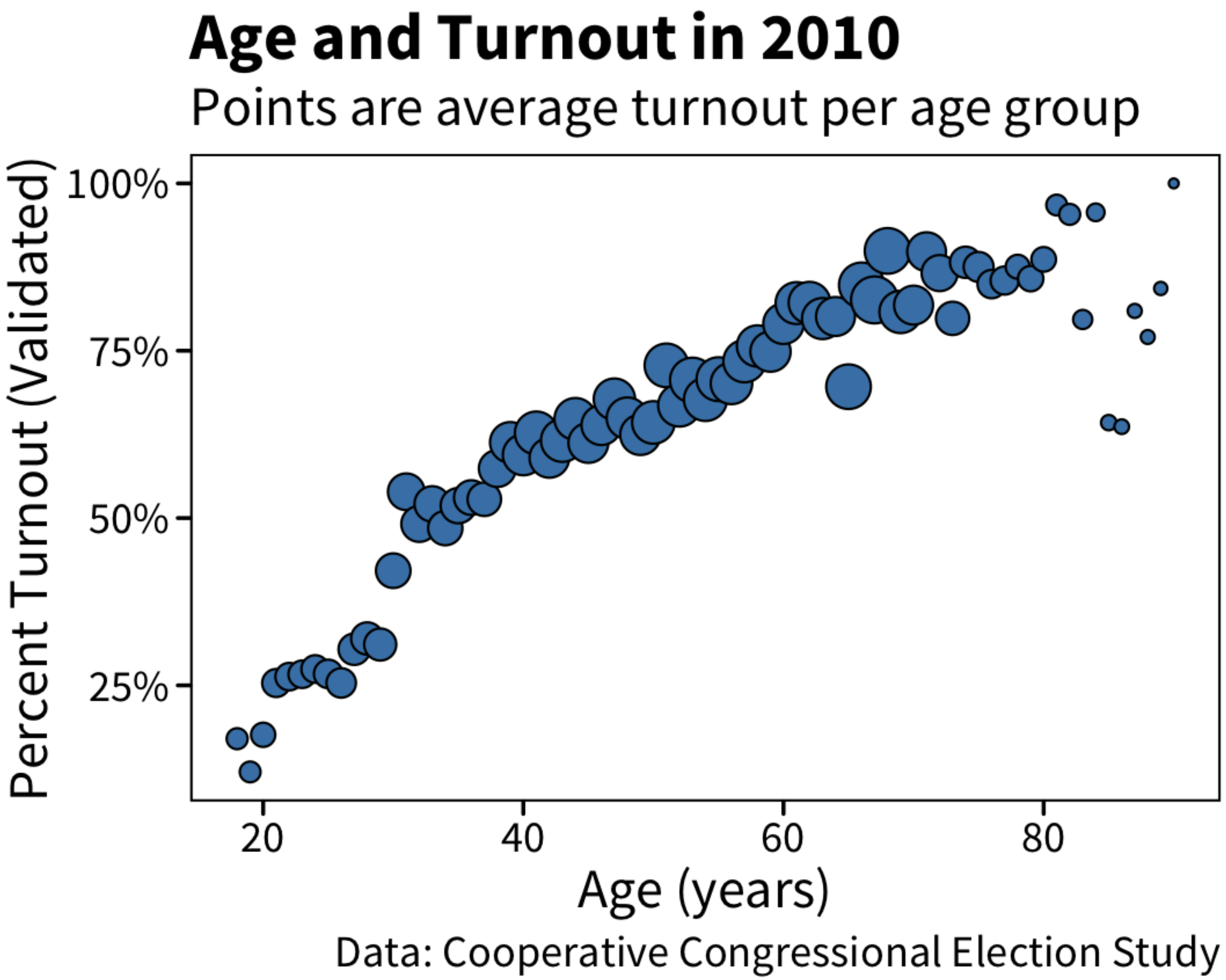
As a comparison of averages:

- Average turnout among older voters
- Among younger voters
- with a twist: age is continuous(ish)

Question: is voter turnout higher among older voters?

As a comparison of averages:

- Average turnout among older voters
- Among younger voters
- with a twist: age is continuous(ish)



Averages are useful because they tell us about the *typical* behavior in the data

Practically & Ethically: individuals $\neq$ averages

Averaging (the math)

$$x = \begin{bmatrix} 6 & 15 & 8 & 16 & 17 \end{bmatrix}.$$

The average of x, we call $\bar{x}$.

Averaging (the math)

$$x = \begin{bmatrix} 6 & 15 & 8 & 16 & 17 \end{bmatrix}.$$

The average of x, we call $\bar{x}$.

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

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

Averaging (the math)

$x = \begin{bmatrix} 6 & 15 & 8 & 16 & 17 \end{bmatrix}.$

The average of x, we call $\bar{x}$.

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

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

```
mean(x)
```

```
## [1] 12.4
```

```
sum(x) / length(x)
```

```
## [1] 12.4
```

Strategies for averaging different data types

Strategies for averaging different data types

Quantitative (interval and ratio) data

```
summarize(gapminder,
          avg_lifeexp = mean(lifeExp),
          avg_gdpPercap = mean(gdpPercap))
```

```
## # A tibble: 1 x 2
##   avg_lifeexp avg_gdpPercap
##         <dbl>         <dbl>
## 1        59.5         7215.
```

Strategies for averaging different data types

Quantitative (interval and ratio) data

```
summarize(gapminder,
          avg_lifeexp = mean(lifeExp),
          avg_gdpPercap = mean(gdpPercap))
```

```
## # A tibble: 1 x 2
##   avg_lifeexp avg_gdpPercap
##         <dbl>         <dbl>
## 1        59.5         7215.
```

Categorical (nominal and ordinal) data

```
# Proportion of data in continents.
summarize(gapminder,
          pr_afr = mean(continent == "Africa"),
          pr_euro = mean(continent == "Europe"))
```

```
## # A tibble: 1 x 2
##   pr_afr pr_euro
##    <dbl>   <dbl>
## 1  0.366   0.211
```

If we have a vector of 1s and 0s ("successes" and "failures"), the mean is equal to the proportion of 1s (successes)

Why we like averages: noise canceling out

We flip a coin.

```r
# make a coin vector
coin <- c("Heads", "Tails")

# "flip" the coin
sample(coin, 1)
```

```
## [1] "Heads"
```

Why we like averages: noise canceling out

We flip a coin.

```r
# make a coin vector
coin <- c("Heads", "Tails")

# "flip" the coin
sample(coin, 1)
```

```
## [1] "Heads"
```

We flip it 5 times.

```r
# 'replace' means we put the coin back each time
flips <- sample(coin, 5, replace = TRUE)

# what's the proportion of heads?
mean(flips == "Heads")
```

```
## [1] 0.2
```

Why we like averages: noise canceling out

Flip 100 times. After each flip, find proportion of heads *up to that point*

Eventually this "running average" should approach what number?

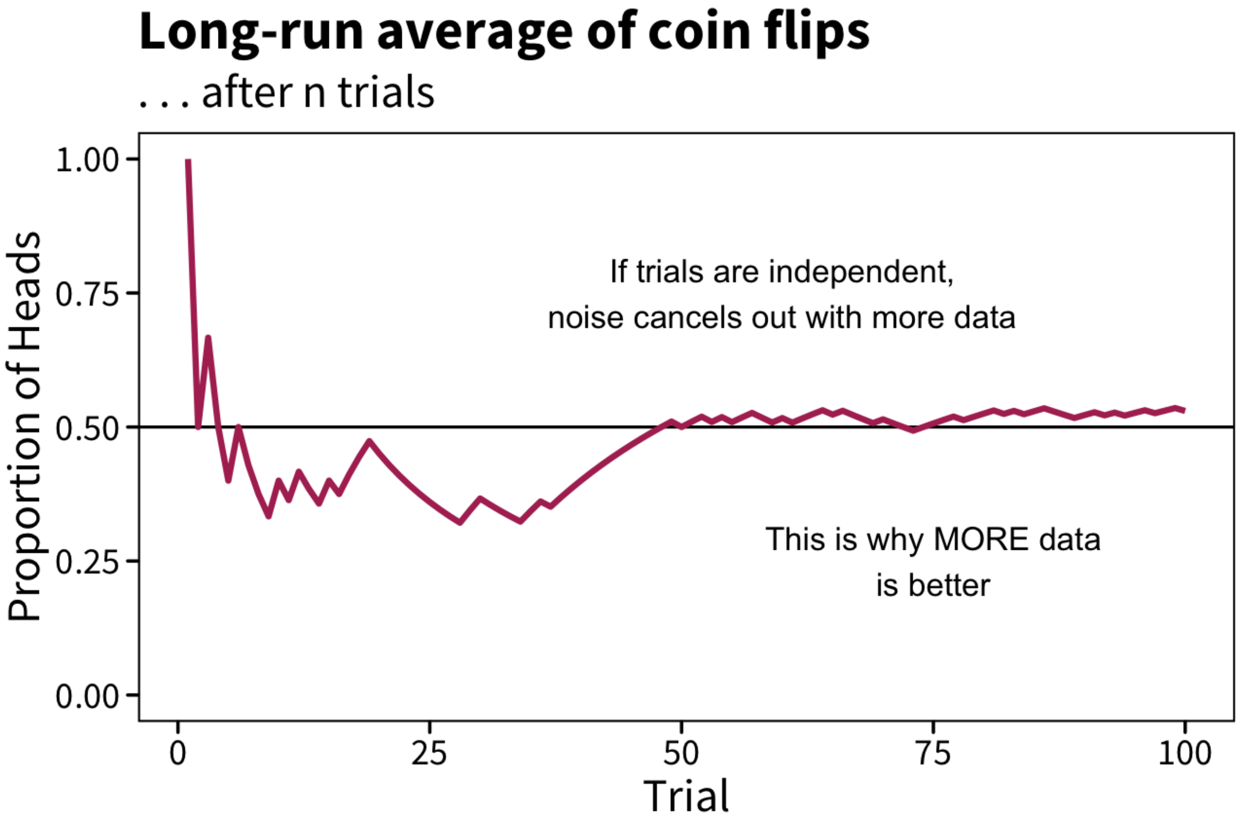
```
## # A tibble: 100 x 3
##    trial flip  running_mean
##    <int> <chr>        <dbl>
##  1     1 Heads        1
##  2     2 Tails        0.5
##  3     3 Heads        0.667
##  4     4 Tails        0.5
##  5     5 Tails        0.4
##  6     6 Heads        0.5
##  7     7 Tails        0.429
##  8     8 Tails        0.375
##  9     9 Tails        0.333
## 10    10 Heads        0.4
## # … with 90 more rows
```

Why we like averages: noise canceling out

Flip 100 times. After each flip, find proportion of heads *up to that point*

Eventually this "running average" should approach what number?

```
## # A tibble: 100 x 3
##    trial flip  running_mean
##    <int> <chr>        <dbl>
##  1     1 Heads        1
##  2     2 Tails        0.5
##  3     3 Heads        0.667
##  4     4 Tails        0.5
##  5     5 Tails        0.4
##  6     6 Heads        0.5
##  7     7 Tails        0.429
##  8     8 Tails        0.375
##  9     9 Tails        0.333
## 10    10 Heads        0.4
## # … with 90 more rows
```



Expectation

Expectation

The true / theoretical / long-run average

Expectation

The true / theoretical / long-run average

Example: more coins

Suppose that the variable $\mathbf{X}$ contains an *arbitrary number* of coin flips (1 = "Heads", 0 = "Tails").

As the number of trials approaches $+\infty$, the mean of $\mathbf{X}$ approaches what value?

Expectation

The true / theoretical / long-run average

Example: more coins

Suppose that the variable $\mathbf{X}$ contains an *arbitrary number* of coin flips (1 = "Heads", 0 = "Tails").

As the number of trials approaches $+\infty$, the mean of $\mathbf{X}$ approaches what value?

$$\mathrm{E}[\mathbf{X}] = 0.5$$

Probabilities are one type of *expectation*, but not the only kind

Another toy example: rolling a die

Roll a six-sided die. What's the expected value?

Another toy example: rolling a die

Roll a six-sided die. What's the expected value? It's 3.5, why?

Another toy example: rolling a die

Roll a six-sided die. What's the expected value? It's 3.5, why?

It's the "theoretical average."

Another toy example: rolling a die

Roll a six-sided die. What's the expected value? It's 3.5, why?

It's the "theoretical average."

$$
\begin{aligned}
\mathrm{E}[X] &= \sum_{k=1}^{K} x_k p_k \\
&= x_1 p_1 + x_2 p_2 + \ldots + x_K p_K
\end{aligned}
$$

- x_k represents a possible outcome
- p_k is the probability of outcome k
- K is the total number of possibilities

Another toy example: rolling a die

Roll a six-sided die. What's the expected value? It's 3.5, why?

It's the "theoretical average."

$$\begin{aligned} \mathrm{E}[X] &= \sum_{k=1}^{K} x_k p_k \\ &= x_1 p_1 + x_2 p_2 + \ldots + x_K p_K \end{aligned}$$

- x_k represents a possible outcome
- p_k is the probability of outcome k
- K is the total number of possibilities

$$\mathrm{E}[\text{Die}] = \left(1 \times \frac{1}{6}\right) + \left(2 \times \frac{1}{6}\right) + \ldots + \left(6 \times \frac{1}{6}\right) = 3.5$$

Expectation is a weighted average of all possible outcomes (each outcome weighted by its probability of occurrence)

Why does this matter?

The *theoretical average* influences the data we collect, but our data don't *perfectly reflect* the true expectation

Why does this matter?

The *theoretical average* influences the data we collect, but our data don't *perfectly reflect* the true expectation

- Candidate A and B are running for Senate
- TRUE support for Candidate A is 54%
- "the population parameter" ($\mu = 0.54$)

Why does this matter?

The *theoretical average* influences the data we collect, but our data don't *perfectly reflect* the true expectation

- Candidate A and B are running for Senate
- TRUE support for Candidate A is 54%
- "the population parameter" ($\mu = 0.54$)

We take a survey of 500 voters

```
# Voters support A or B randomly with given probabilities.
# repeat 500x with replacement
voters <- sample(c("A", "B"), prob = c(0.54, 1 - 0.54),
                 size = 500, replace = TRUE)

# what proportion of the sample is voting for A?
mean(voters == "A")
```

```
## [1] 0.56
```

Why does this matter?

The *theoretical average* influences the data we collect, but our data don't *perfectly reflect* the true expectation

- Candidate A and B are running for Senate
- TRUE support for Candidate A is 54%
- "the population parameter" ($\mu = 0.54$)

We take a survey of 500 voters

```
# Voters support A or B randomly with given probabilities.
# repeat 500x with replacement
voters <- sample(c("A", "B"), prob = c(0.54, 1 - 0.54),
                 size = 500, replace = TRUE)

# what proportion of the sample is voting for A?
mean(voters == "A")
```

```
## [1] 0.56
```

Expected value (a.k.a. "population mean") is $\mu = 0.54$

Estimated value (a.k.a. "sample mean") is $\bar{x} = 0.56$

Taking a sample

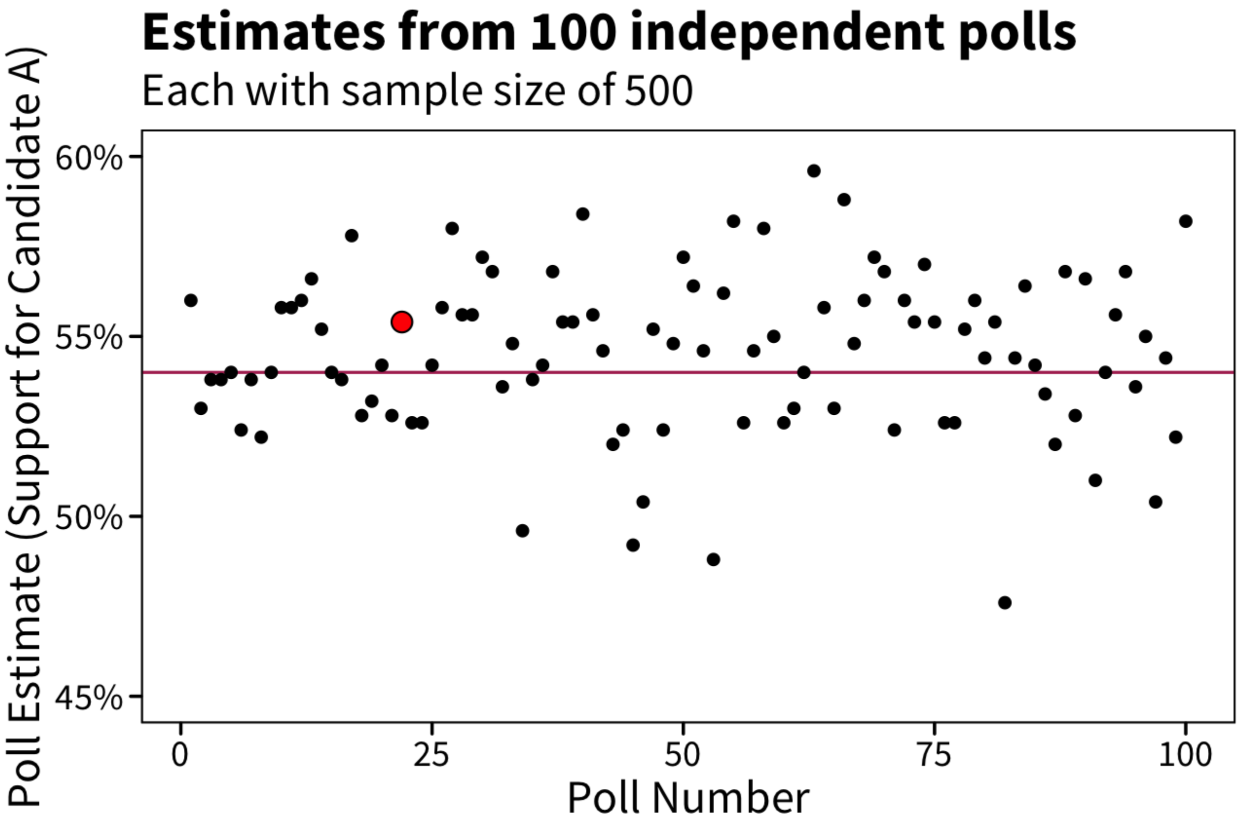
We will never know the *true mean* (μ) with certainty.

But we can take samples of data and calculate the *sample mean* ($\bar{x}$) within the sample.

Taking a sample

We will never know the *true mean* (μ) with certainty.

But we can take samples of data and calculate the *sample mean* ($\bar{x}$) within the sample.



The whole point of statistics

is to figure out how confident we can be about the *real truth*

given that we only can observe our imperfect sample of data

Aggregation

Aggregation

We have data at some level of analysis, and we want to summarize it at a higher level of analysis

Aggregation

We have data at some level of analysis, and we want to summarize it at a higher level of analysis

Life expectancy *aggregated* by year

```
## # A tibble: 12 x 2
##     year lifeExp_mean
##    <int>        <dbl>
##  1  1952         49.1
##  2  1957         51.5
##  3  1962         53.6
##  4  1967         55.7
##  5  1972         57.6
##  6  1977         59.6
##  7  1982         61.5
##  8  1987         63.2
##  9  1992         64.2
## 10  1997         65.0
## 11  2002         65.7
## 12  2007         67.0
```

Aggregation

We have data at some level of analysis, and we want to summarize it at a higher level of analysis

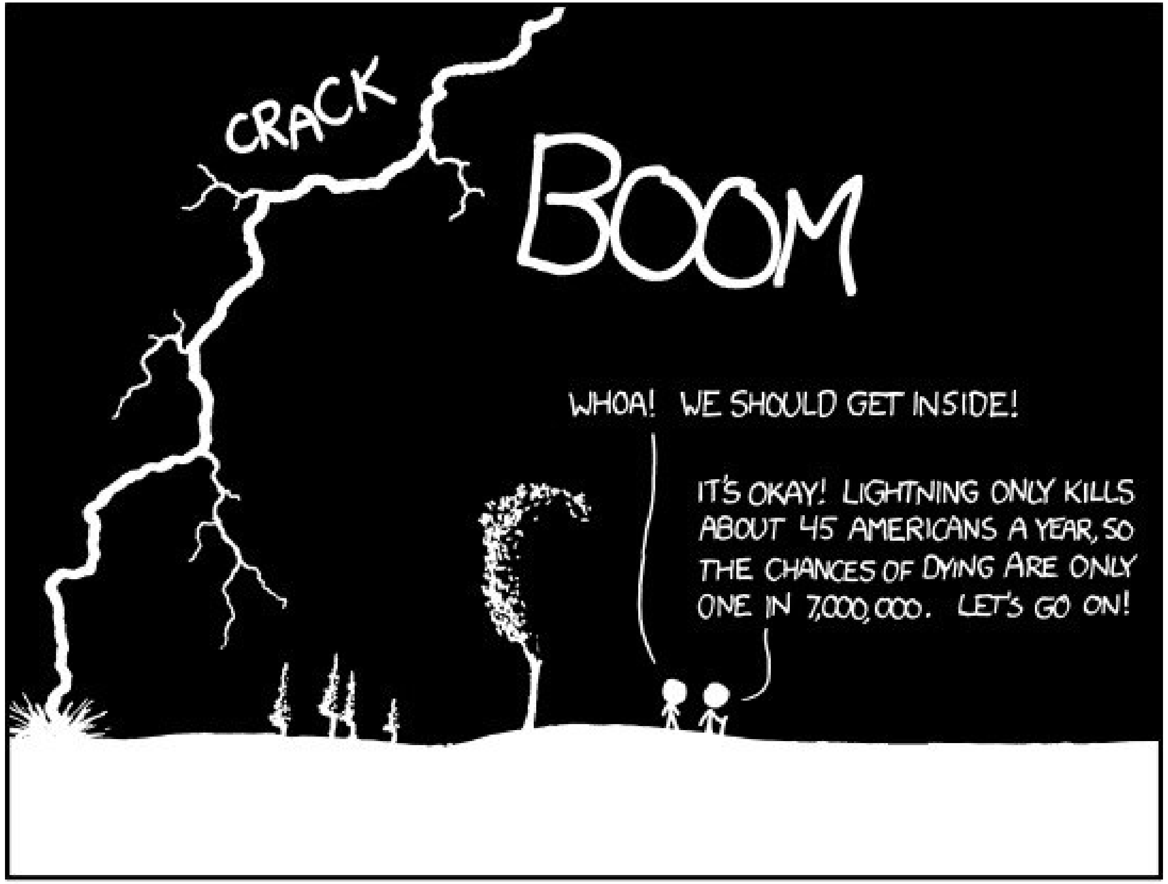
Life expectancy *aggregated* by year

```
## # A tibble: 12 x 2
##     year lifeExp_mean
##    <int>        <dbl>
##  1  1952         49.1
##  2  1957         51.5
##  3  1962         53.6
##  4  1967         55.7
##  5  1972         57.6
##  6  1977         59.6
##  7  1982         61.5
##  8  1987         63.2
##  9  1992         64.2
## 10  1997         65.0
## 11  2002         65.7
## 12  2007         67.0
```

Life expectancy *aggregated* by continent-year

```
## # A tibble: 60 x 3
## # Groups:   continent [5]
##    continent  year lifeExp_mean
##    <fct>     <int>        <dbl>
##  1 Africa     1952         39.1
##  2 Africa     1957         41.3
##  3 Africa     1962         43.3
##  4 Africa     1967         45.3
##  5 Africa     1972         47.5
##  6 Africa     1977         49.6
##  7 Africa     1982         51.6
##  8 Africa     1987         53.3
##  9 Africa     1992         53.6
## 10 Africa     1997         53.6
## # … with 50 more rows
```

"Conditional" averages and "conditional" probabilities

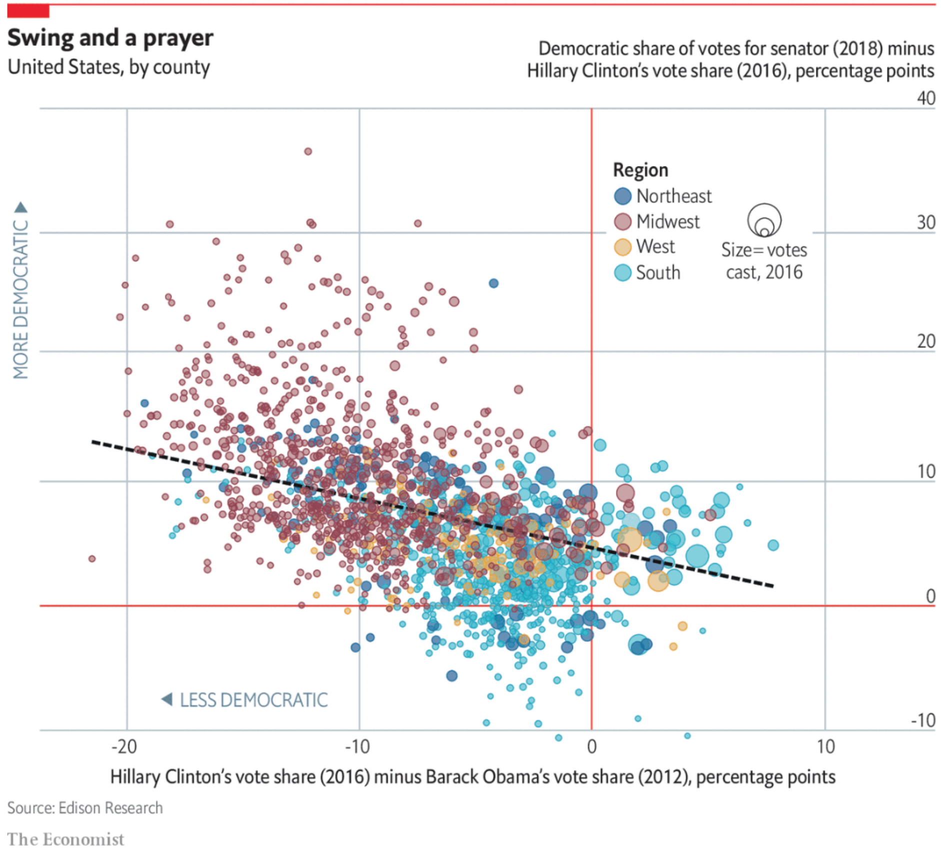


THE ANNUAL DEATH RATE AMONG PEOPLE WHO KNOW THAT STATISTIC IS ONE IN SIX.

Ecological Fallacy:
Assuming that group-level patterns apply to individuals within the group

Obama-Trump voters turn back to Democrats

Senate Democrats did especially well where Donald Trump had gained the most ground



See ya

Wednesday is (spooky voice) randomnesssssss