

Working with Data in R

Sam Frederick
sdf2128@columbia.edu

Data Sources for Final Projects

- Harvard Dataverse
- Cooperative Election Study
- American National Election Study
- Voteview (NOMINATE Scores for Congress)
- Center for Effective Lawmaking (Effective Lawmaking in Congress)
- Countless others
- Can also make your own

Reading Different File Types

File Type	R Function
.csv	<code>read.csv()</code>
.RData	<code>load()</code>
.txt	<code>read.delim()</code>
.dta	<code>read_dta()</code> from haven package
.xlsx	<code>read_excel()</code> from readxl package

Downloading Data

Attention: We recently changed the way that we present the vote totals for each roll call. We no longer include paired votes or the pseudo-votes of presidents in our reported totals. The new totals should now match those officially reported. Paired votes and the pseudo-votes continue to be included in the plots, maps, and enumerations of the voting on each roll call.



Vote and Member Search

[advanced search](#)

Search tip: Search for names in the news: [Nancy Pelosi](#)

109,912 votes found.

Sort by [Newest](#) / [Oldest](#)

[117th Congress](#) > [House](#) > [Vote 924 on 2022-09-30](#)

Bill number: HR8987

Vote: 400-31 (Passed)

Question: On Passage

Description: Fairness for 9/11 Families Act

Downloading Data

Attention: We recently changed the way that we present the vote totals for each roll call. We no longer include paired votes or the pseudo-votes of presidents in our reported totals. The new totals should now match those officially reported. Paired votes and the pseudo-votes continue to be included in the plots, maps, and enumerations of the voting on each roll call.



Vote and Member Search

advanced search

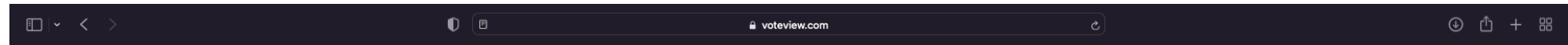
Search tip: Look up Senate confirmation votes: attorney general confirmation

109,912 votes found.

Sort by Newest / Oldest

117th Congress > House > Vote 924 on 2022-09-30	<div></div> <div></div>
Bill number: HR8987	
Vote: 400-31 (Passed)	
Question: On Passage	
Description: Fairness for 9/11 Families Act	
117th Congress > House > Vote 923 on 2022-09-30	<div></div> <div></div>

Downloading Data



voteview.com beta

[search](#) [chamber](#) [party](#) [geography](#) [data](#) [about](#)

Attention: We recently changed the way that we present the vote totals for each roll call. We no longer include paired votes or the pseudo-votes of presidents in our reported totals. The new totals should now match those officially reported. Paired votes and the pseudo-votes continue to be included in the plots, maps, and enumerations of the voting on each roll call.

Realtime NOMINATE Ideology and Related Data

This section contains download links for NOMINATE scores and other data that we make available to the public, in addition to tutorial articles explaining how to generate popular ancillary data from our data exports. Please continue by choosing the data you wish to download.

Data is updated live, as new votes are taken. If you are an institutional user of our data and wish to be notified before major or breaking changes are made to the data, please see the [About](#) page.

Please cite the dataset as:

Lewis, Jeffrey B., Keith Poole, Howard Rosenthal, Adam Boche, Aaron Rudkin, and Luke Sonnet (2022). *Voteview: Congressional Roll-Call Votes Database*. <https://voteview.com/>

Data Type:

Chamber:

Congress:

File Format:

[Download Data](#)

Member Ideology

This data includes basic biographical information (state, district, party, name) and ideological scores for members of the selected congresses.

[Click here for help using this data](#)

Ancillary Data and Analyses

We are pleased to present a collection of articles discussing data and analyses that make use of NOMINATE / voteview.com, along with the source code used to produce the analyses. We hope these will be of use to scholars, journalists, and students interested in producing analysis using our data:

[Attendance of Senators and House Members Running for President:](#) Track the rate of participation in roll call votes of Members of Congress who are running for President in 2020

[Clausen, Peltzman, and Issue codes for 1st to 113th Congresses:](#) This short article provides descriptions for the numeric Clausen, Peltzman, and Issue codes supplied with our data.

[Joe Biden's Party Loyalty in the Senate:](#) This article calculates Joe Biden's career rate of party loyalty in the Senate and compares

Downloading Data

The screenshot shows the MIT Election Lab website. The header includes navigation links: RESEARCH, DATA, FIND AN EXPERT, MIT ELECTION DATA + SCIENCE LAB, ABOUT, ENGAGE, NEWS, and the MIT logo. The main banner features the word 'RESEARCH' in large white letters on a blue background with a dotted pattern. Below the banner, there are two featured articles:

OCT 5, 2022 MEDSL

Announcing our Newest Grant Recipients

We are pleased to announce that we are funding 18 research teams around the country with our new Evolving Election Administration Landscape Grants! Click here to read more & see the full list of recipients and projects.

NEW GRANT PROJECTS

OCT 5, 2022 MEDSL

Register for our 2022 Post-Election Webinar

On December 8, we will be hosting a public webinar all about the 2022 election, featuring our own takes on what happened as well as highlighting other researchers' work and what they saw. Register today to hold your spot and receive more details about the event!

UPCOMING EVENT

Downloading Data

The screenshot shows the MIT Election Data + Science Lab website. The header includes navigation links: RESEARCH, DATA, FIND AN EXPERT, MIT ELECTION DATA + SCIENCE LAB, ABOUT, ENGAGE, NEWS, and the MIT logo. The main banner features the word 'RESEARCH' in large white letters on a blue background with a dotted pattern. Below the banner, there are two featured articles:

OCT 5, 2022 MEDSL

Announcing our Newest Grant Recipients

We are pleased to announce that we are funding 18 research teams around the country with our new Evolving Election Administration Landscape Grants! Click here to read more & see the full list of recipients and projects.

NEW GRANT PROJECTS

OCT 5, 2022 MEDSL

Register for our 2022 Post-Election Webinar

On December 8, we will be hosting a public webinar all about the 2022 election, featuring our own takes on what happened as well as highlighting other researchers' work and what they saw. Register today to hold your spot and receive more details about the event!

UPCOMING EVENT

Downloading Data

RESEARCH

DATA

FIND AN EXPERT

MIT ELECTION DATA
+ SCIENCE LAB

ABOUT

ENGAGE

NEWS

MIT

JAN 1, 2019

MIT Election Data and Science Lab

State Constituency-Level Returns 2018

This repository contains official constituency (state-level) for the 2018 midterm elections, including data for the U.S. House, U.S. Senate, state offices and county-level returns.

FEDERAL ELECTIONS

BY STATE

GENERAL ELECTION

MAR 31, 2022

MIT Election Data and Science Lab

U.S. House 1976–2020

This data file contains constituency (district) returns for elections to the U.S. House of Representatives from 1976 to 2020.

FEDERAL ELECTIONS

BY DISTRICT

GENERAL ELECTION

NOV 7, 2018

MIT Election Data and Science Lab

U.S. General Elections 2018 - Analysis Dataset

This repository includes demographic and past election data that can easily be merged with

U.S. Senate 1976–2020

This data file contains constituency (state-level) returns for elections to the U.S. Senate from 1976 to 2020.

FEDERAL ELECTIONS

BY STATE

GENERAL ELECTION

JAN 13, 2021

MIT Election Data and Science Lab

U.S. President 1976–2020

This data file contains constituency (state-level) returns for elections to the U.S. presidency from 1976 to 2020.

FEDERAL ELECTIONS

BY STATE

GENERAL ELECTION

AUG 15, 2018

MIT Election Data and Science Lab

U.S. Primary Elections 2018

This is a repository for unofficial 2018 primary election returns.

FEDERAL ELECTIONS

BY COUNTY

PRIMARY ELECTION

Downloading Data


dataverse.harvard.edu

Data | MIT Election Lab

U.S. House 1976–2020 - U.S. House Elections

HARVARD
Dataverse

Add Data ▾ Search ▾ About User Guide Support Log In



MIT Election Data and Science Lab, 2017, "U.S. House 1976–2020", <https://doi.org/10.7910/DVN/IG0UN2>, Harvard Dataverse, V11, UNF:6:ry6R0P1KRBhWkIfZzKiM8A== [fileUNF]
[Cite Dataset ▾](#) Learn about [Data Citation Standards](#).

Access Dataset ▾


Contact Owner Share

Dataset Metrics ⓘ
24,825 Downloads ⓘ

Description ⓘ
This data file contains constituency (district) returns for elections to the U.S. House of Representatives from 1976 to 2020.

Subject ⓘ
Social Sciences

Keyword ⓘ
Elections

License/Data Use Agreement
 CC0 1.0

Files Metadata Terms Versions

Search this dataset... 🔍


Filter by
File Type: All ▾ Access: All ▾

Sort ▾



1 to 2 of 2 Files


Download ▾

Access File




[1976-2020-house.tab](#)
Tabular Data - 4.4 MB
Published Mar 31, 2022
1,866 Downloads
20 Variables, 31103 Observations UNF:6:ry6R...M8A== 📄





[codebook-us-house-1976-2020.md](#)
Markdown Text - 5.3 KB
Published Jul 13, 2021
1,491 Downloads
MD5: c71...a78 📄



Downloading Data


dataverse.harvard.edu

Data | MIT Election Lab

U.S. House 1976–2020 - U.S. House Elections

HARVARD
Dataverse

Add Data ▾ Search ▾ About User Guide Support Log In



MIT Election Data and Science Lab, 2017, "U.S. House 1976–2020", <https://doi.org/10.7910/DVN/IG0UN2>, Harvard Dataverse, V11, UNF:6:ry6R0P1KRBhWkIfZzKiM8A== [fileUNF]
[Cite Dataset ▾](#) Learn about [Data Citation Standards](#).

Access Dataset ▾
Contact Owner Share

Dataset Metrics ⓘ
24,825 Downloads ⓘ


Description ⓘ This data file contains constituency (district) returns for elections to the U.S. House of Representatives from 1976 to 2020.

Dataset Terms

✕

This dataset is made available under the following terms. Please confirm and/or complete the information needed below in order to continue.

License/Data Use Agreement

Our [Community Norms](#) as well as good scientific practices expect that proper credit is given via citation. Please use the data citation shown on the dataset page.
 CC0 1.0

Name *

Email *


Institution *

Position *

Accept Cancel

MD5: c71...a78

Copyright © 2022, The President & Fellows of Harvard College | [Privacy Policy](#)

Powered by  v. 5.12 build pirate-(eye)-patch

Feedback

Building a Dataset

1. Clean and Standardize Data
2. Merge Data
3. Check Merge
4. Fix Errors (if any)
5. Repeat until no errors remain

Cleaning Data

```
1 nominate <- read.csv("H116_members.csv")
2 elections <- read.csv("1976-2020-house.csv")

1 elections <- subset(elections, subset = year ==
2     2018 & writein == FALSE & special ==
3     FALSE & is.na(runoff) & stage == "GEN")
4 nominate <- subset(nominate, subset = chamber ==
5     "House" & state_abbrev %in% state.abb)
```

Cleaning Data: Elections

```
1 elections <- aggregate(candidatevotes ~ state_po +  
2     district + candidate + totalvotes, data = elections,  
3     FUN = sum)  
4 elections$pct_vote <- (elections$candidatevotes/elections$totalvotes) *  
5     100  
6 elections$candidate_lastname <- gsub(" $|\\\"|'",  
7     "", gsub(" +", " ", gsub("JR$| I$| II$| III$| IV$",  
8     "", elections$candidate)))  
9 elections$candidate_lastname <- unlist(lapply(elections$candidate_lastname,  
10     FUN = function(x) tail(unlist(strsplit(x,  
11     " ")), n = 1)))
```

Cleaning Data: Elections

What does this code do?

```
1 elections <- aggregate(candidatevotes ~ state_po +  
2   district + candidate + totalvotes, data = elections,  
3   FUN = sum)  
4 elections$pct_vote <- (elections$candidatevotes/elections$totalvotes) *  
5   100  
6 elections$candidate_lastname <- gsub(" $|\\\"'",  
7   "", gsub(" +", " ", gsub("JR$| I$| II$| III$| IV$",  
8   "", elections$candidate)))  
9 elections$candidate_lastname <- unlist(lapply(elections$candidate_lastname,  
10   FUN = function(x) tail(unlist(strsplit(x,  
11   " ")), n = 1)))
```

Cleaning Data: Elections

What does this code do?

```
1 elections <- aggregate(candidatevotes ~ state_po +  
2     district + candidate + totalvotes, data = elections,  
3     FUN = sum)  
4 elections$pct_vote <- (elections$candidatevotes/elections$totalvotes) *  
5     100  
6 elections$candidate_lastname <- gsub(" $|\\\'",  
7     "", gsub(" +", " ", gsub("JR$| I$| II$| III$| IV$",  
8     "", elections$candidate)))  
9 elections$candidate_lastname <- unlist(lapply(elections$candidate_lastname,  
10     FUN = function(x) tail(unlist(strsplit(x,  
11     " ")), n = 1)))
```


Cleaning Data: Elections

```
1 elections <- elections[, colnames(elections) %in%  
2   c("state_po", "district", "candidate_lastname",  
3     "pct_vote")]  
4 elections$district <- ifelse(elections$district ==  
5   0, 1, elections$district)  
6 colnames(elections)[1] <- "state_abbrev"  
7 colnames(elections)[2] <- "district_code"
```

Cleaning Data: Elections

What does this code do?

```
1 elections <- elections[, colnames(elections) %in%  
2   c("state_po", "district", "candidate_lastname",  
3     "pct_vote")]  
4 elections$district <- ifelse(elections$district ==  
5   0, 1, elections$district)  
6 colnames(elections)[1] <- "state_abbrev"  
7 colnames(elections)[2] <- "district_code"
```

Cleaning Data: Elections

What does this code do?

```
1 elections <- elections[, colnames(elections) %in%  
2   c("state_po", "district", "candidate_lastname",  
3     "pct_vote")]  
4 elections$district <- ifelse(elections$district ==  
5   0, 1, elections$district)  
6 colnames(elections)[1] <- "state_abbrev"  
7 colnames(elections)[2] <- "district_code"
```

Cleaning Data: Helpful Functions

```
1 aggregate(y ~ g1 + g2 + g3..., data = data,  
2          FUN = ...)
```

- Groups y variable by g1, g2, g3,...
- data: Include your data
- FUN: apply this function to grouped y variable within each group

Cleaning Data: Helpful Functions

```
1 x %in% y
```

- Checks whether x is in vector y

```
1 1 %in% c(1, 2, 3)
```

```
[1] TRUE
```

```
1 1 %in% c(2, 3, 4)
```

```
[1] FALSE
```

```
1 "a" %in% c("a", "b", "c")
```

```
[1] TRUE
```

Cleaning Data: Helpful Functions

```
1 ifelse(condition, x, y)
```

- If the condition is TRUE, then ifelse() returns x
- If the condition is FALSE, then ifelse() returns y

```
1 ifelse(1 %in% c(1, 2, 3), "In Vector", "Not In Vector")
```

```
[1] "In Vector"
```

```
1 ifelse(5 %in% c(1, 2, 3), "In Vector", "Not In Vector")
```

```
[1] "Not In Vector"
```

Cleaning Data: Helpful Functions

```
1 lapply(x, FUN = ...)
```

- Applies function specified by FUN to every element of x
- Returns a list

```
1 lapply(1:3, FUN = function(x) x * 2)
```

```
[[1]]  
[1] 2
```

```
[[2]]  
[1] 4
```

```
[[3]]  
[1] 6
```

Cleaning Data: Helpful Functions

```
1 unlist(x)
```

- Turns list into vector

```
1 unlist(lapply(1:3, FUN = function(x) x *  
2          2))
```

```
[1] 2 4 6
```


Cleaning Data: Strings

```
1 toupper(x)  
2 tolower(x)
```

- Changes strings to all uppercase or all lowercase

```
1 toupper("abc")
```

```
[1] "ABC"
```

```
1 tolower("AbCD")
```

```
[1] "abcd"
```

Cleaning Data: Strings

```
1 strsplit(x, split = s)
```

- Splits string at s into a vector within a list

```
1 strsplit("BIDEN, Joseph R.", ", ")
```

```
[[1]]
```

```
[1] "BIDEN"      "Joseph R."
```

```
1 unlist(strsplit("BIDEN, Joseph R.", ", "))
```

```
[1] "BIDEN"      "Joseph R."
```

Cleaning Data: Strings

```
1 iconv(x, to = "ASCII//TRANSLIT")
```

- Converts text x to ASCII

```
1 iconv("äé", to = "ASCII//TRANSLIT")
```

```
[1] "\"a'e"
```

Regular Expressions (Regex)

- Often working with texts that vary in content but have predictable formats
- Regex can help with this
- Regex matches patterns in text

```
1 grepl(pattern = patt, x = x)
```

- Is there a match for pattern patt in string x?

```
1 grepl(pattern = "A", x = "BCD")
```

```
[1] FALSE
```

```
1 grepl(pattern = "9", x = "2019")
```

```
[1] TRUE
```

Regular Expressions (Regex)

```
1 gsub(pattern = patt, replacement = rep, x = x)
```

- Replaces occurrences of patt with rep in string x

```
1 gsub(pattern = "A", replacement = "Z", x = "ABACAD")
```

```
[1] "ZBZCZD"
```

```
1 gsub(pattern = "9", replacement = "X", x = "1999999999")
```

```
[1] "1XXXXXXXXX"
```

Regular Expressions (Regex)

- Brackets [] create groups of characters to be matched
 - “[A-Z]” matches all uppercase letters from A to Z
 - “[0-9]” matches all numbers
 - “[^0-9]” matches everything except numbers
 - Can customize content in brackets

```
1 gsub("[0-9]", "", "2019AL01")
```

```
[1] "AL"
```

Regular Expressions (Regex)

- How many matches do we want?
 - “s*” signifies 0 or more matches of s
 - “s+” signifies 1 or more matches of s
 - “s{num}” signifies num matches of s
 - “s{min,max}” signifies between min and max matches of s
- How could we get only state and district from the string “2019AL01”?

```
1 gsub(pattern, "", "2019AL01")
```

```
1 gsub("[0-9]{4}", "", "2019AL01")
```

```
[1] "AL01"
```

Regular Expressions (Regex)

- “`^[A-Z]`” matches only at the start of the string

```
1 grepl("^[A-Z]", "2019AL01")
```

```
[1] FALSE
```

- “`[A-Z]$`” matches only at the end of the string

```
1 grepl("[A-Z]$", "2019AL01")
```

```
[1] FALSE
```

```
1 grepl("[A-Z]$", "AL201901A")
```

```
[1] TRUE
```


Regular Expressions (Regex)

- Matching special characters
 - Need to escape with two “\”

```
1 grepl("\\?", "How's it going?")
```

```
[1] TRUE
```

Regular Expressions (Regex)

- Great reference at this [link](#)

Cleaning Data: NOMINATE

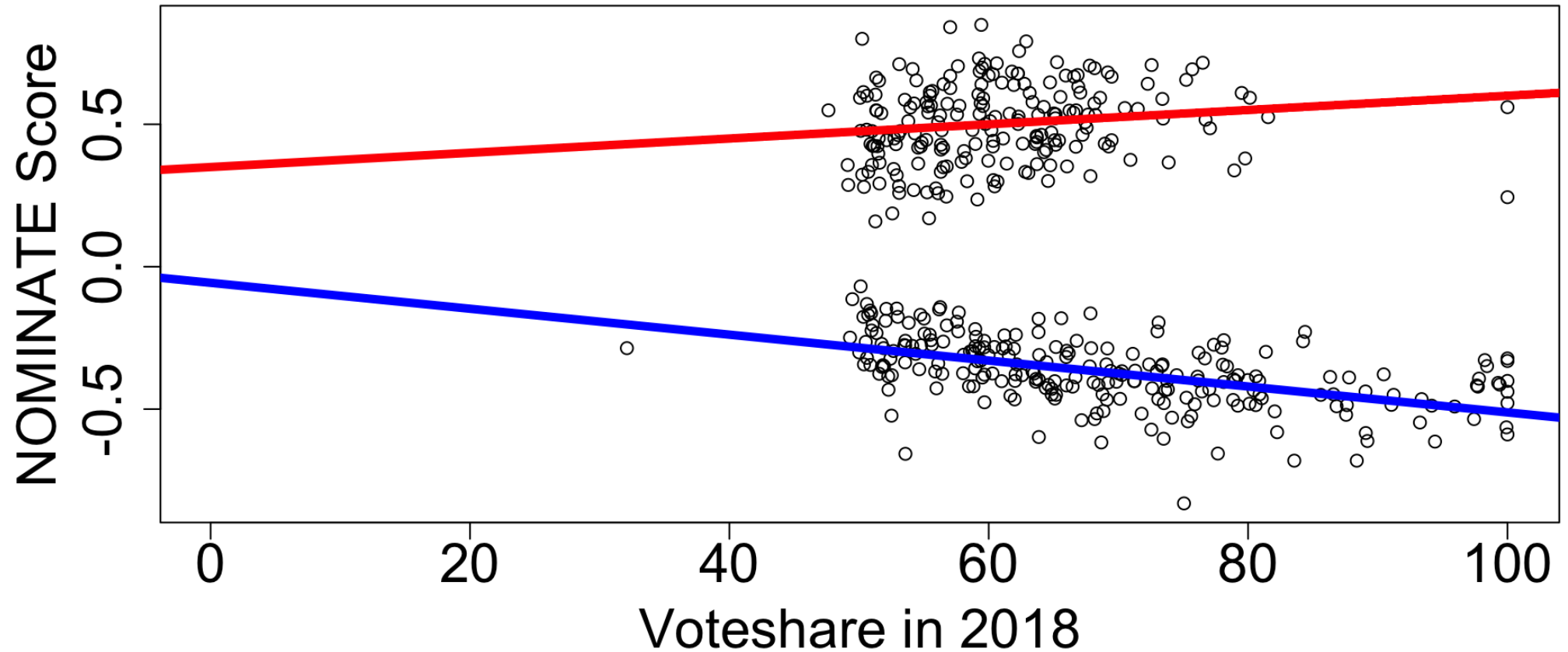
```
1 nominate <- nominate[, colnames(nominate) %in%
2   c("district_code", "state_abbrev", "party_code",
3     "nominate_dim1", "bioname")]
4 nominate$candidate_lastname <- unlist(lapply(nominate$bioname,
5   function(x) head(unlist(strsplit(x, ",")),
6     n = 1)))
7 nominate$candidate_lastname <- toupper(nominate$candidate_lastname)
8 nominate$candidate_lastname <- gsub("\\'",
9   "", iconv(nominate$candidate_lastname,
10     to = "ASCII//TRANSLIT"))
11 nominate$candidate_lastname <- unlist(lapply(nominate$candidate_lastname,
12   function(x) tail(unlist(strsplit(x, " ")),
13     n = 1)))
14 nominate$party <- ifelse(nominate$party_code ==
15   200, "REPUBLICAN", ifelse(nominate$party_code ==
16   100, "DEMOCRAT", "INDEPENDENT"))
17 nominate <- subset(nominate, subset = party !=
18   "INDEPENDENT")
```

Merging Data

```
1 full <- merge(elections, nominate, by = c("district_code",  
2   "state_abbrev", "candidate_lastname"),  
3   all = TRUE)  
4 plot(full$pct_vote, full$nominate_dim1, xlab = "Voteshare in 2018",  
5   ylab = "NOMINATE Score", main = "NOMINATE vs. Voteshare")  
6 fit <- lm(nominate_dim1 ~ pct_vote * party,  
7   data = full)  
8 abline(a = coef(fit)[1], b = coef(fit)[2],  
9   col = "blue", lwd = 5)  
10 abline(a = coef(fit)[1] + coef(fit)[3], b = coef(fit)[2] +  
11   coef(fit)[4], col = "red", lwd = 5)
```

Merging Data

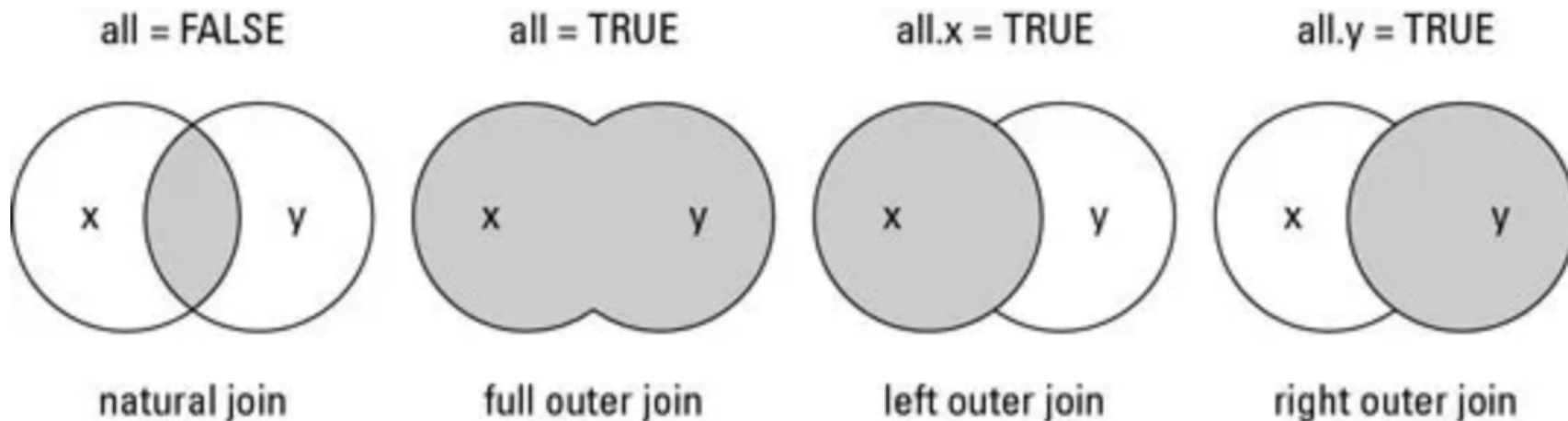
NOMINATE vs. Voteshare



Merging Data

```
1 merge(x, y, by = c(colname1, colname2, ...),  
2       all = , all.x = , all.y = )
```

- Merge dataframe x with dataframe y by matches in specified columns



Source

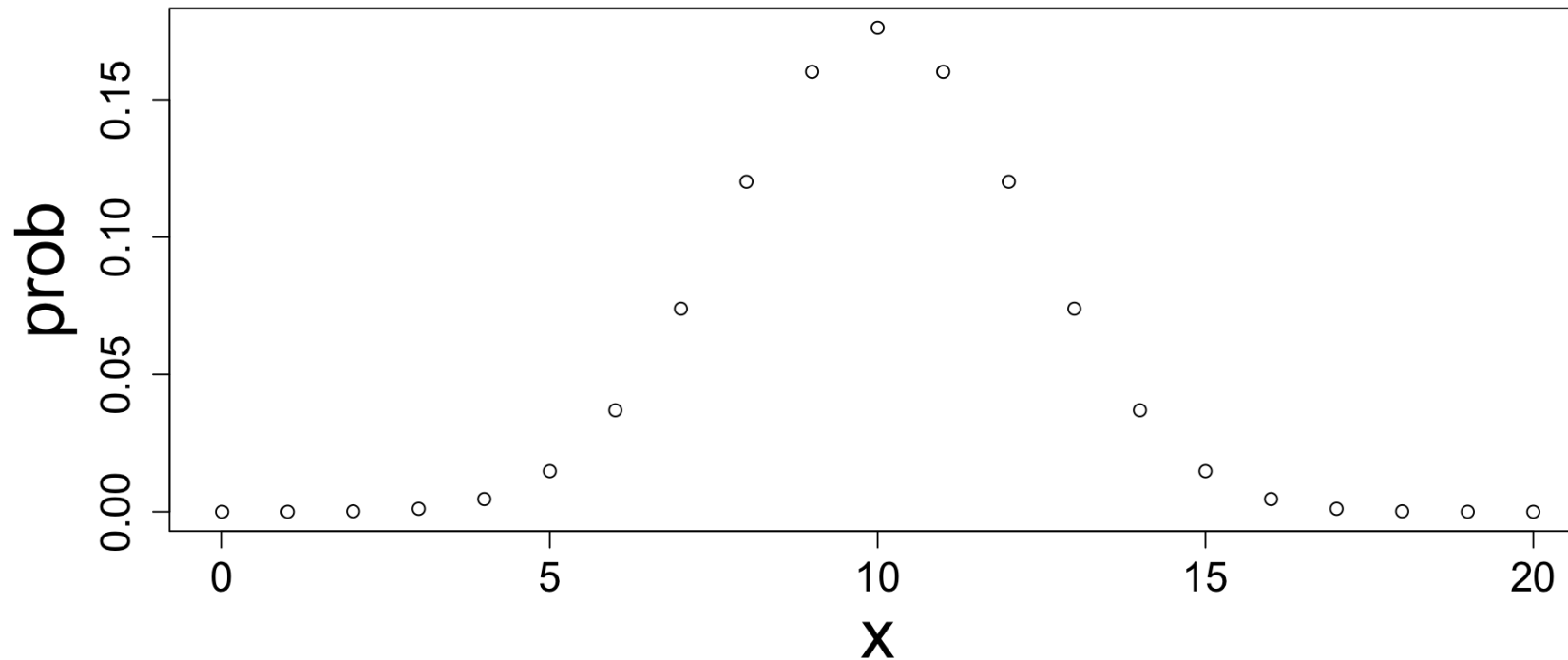
Probability: Binomial Distribution

Probability Mass Function

$$\begin{aligned} Pr(X = x) &= \binom{n}{x} p^x (1 - p)^{n-x} \\ &= \frac{n!}{x!(n-x)!} p^x (1 - p)^{n-x} \end{aligned}$$

Probability: Binomial Distribution

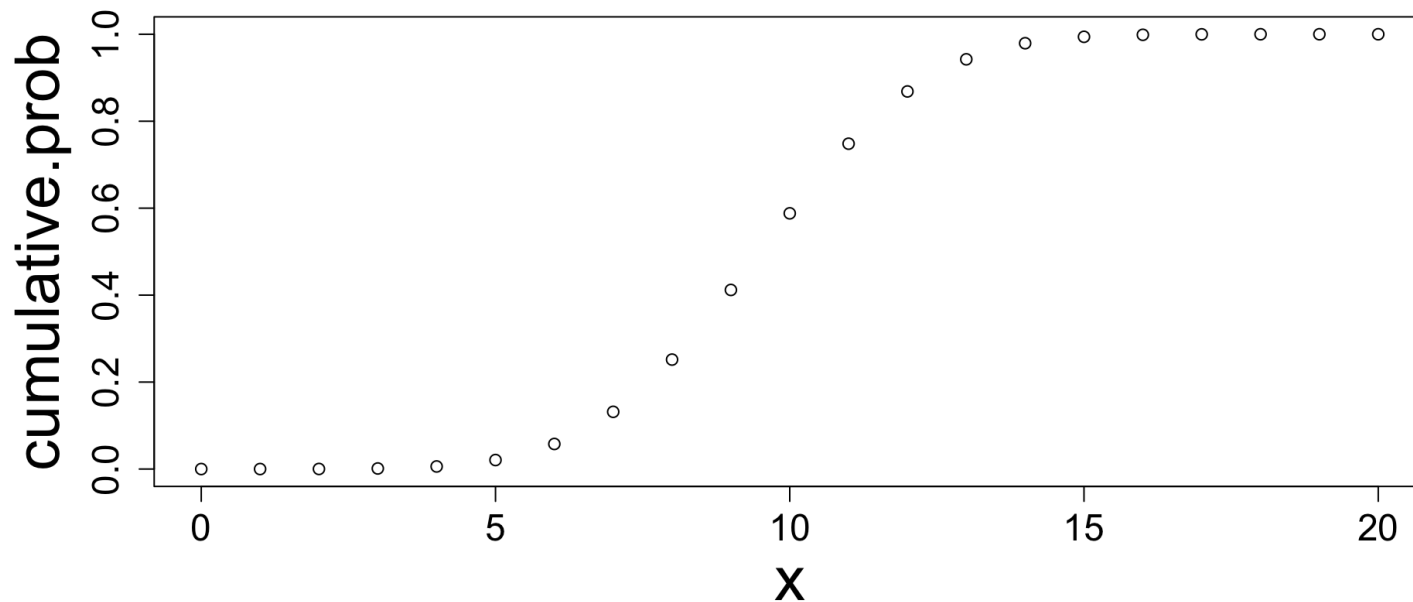
```
1 n <- 20
2 x <- seq(0, 20, 1)
3 prob <- dbinom(x, size = n, prob = 0.5)
4 plot(x, prob)
```



Probability: Binomial Distribution

Cumulative Distribution Function

```
1 n <- 20
2 x <- seq(0, 20, 1)
3 cumulative.prob <- pbinom(x, n, prob = 0.5)
4 plot(x, cumulative.prob)
```



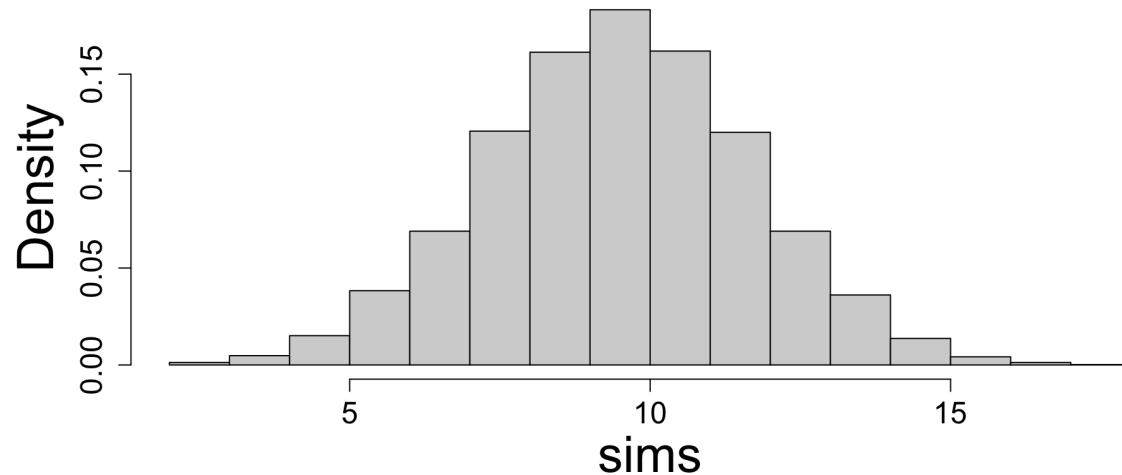
Probability: Binomial Distribution

Random Number Generation

- Simulate random numbers from distribution

```
1 set.seed(123)
2 sims <- rbinom(10000, 20, 0.5)
3 hist(sims, freq = F)
```

Histogram of sims



Probability

- Why simulate random numbers?
 - Challenging computations
 - Fake data with known distribution to check our intuition