Lab 1 Exercise

Theodore Landsman

8/26/2021

FIRST, SOME TIPS & HELPFUL NOTES...

```
# Place a hashtag (or several!) before text to write yourself a note as I'm doing here
#   ALWAYS LEAVE YOURSELF NOTES!
```

- Coding in R is case-sensitive!
- Run lines of code by holding down "ctrl" + "r" OR "ctrl" + "enter" on Windows; "cmd" + "return" on Mac
- There is also a button up top that says "Run" and you can use that too
- In RMarkdown, run code `chunks` by clicking the green play button at the top right of the `chunk` when you hit the blue yarn `knit` button at the top left of the file, all r code will be executed as part of generating your output file.

Datasets We Are Using Today

- Ideology score of U.S. legislators for the 117th Congress
  - `HS117_members.csv`
  - https://voteview.com/data
- Ideology score of countries using United Nations General Assembly votes
  - `IdealpointestimatesAll_Mar2021.tab`
  - https://dataverse.harvard.edu/dataset.xhtml?persistentId=hdl:1902.1/12379

Let's Describe our data

- For code chunks with data import, we typically set `echo = FALSE`, because data import doesn't convey anything that needs displaying. Dataset import will also generate a lot of annoying messages that will be displayed in your pdf unless you set `message = FALSE` in the code chunk as well
- For problem sets, I am not interested in how you are importing data UNLESS you are having a problem with data import.

Desribing the dataset:

To see a list of all of the variable names in the dataset (object) called "voteview":

```
names(voteview)
```

```
##  [1] "congress"                  "chamber"
##  [3] "icpsr"                     "state_icpsr"
##  [5] "district_code"             "state_abbrev"
##  [7] "party_code"                "occupancy"
##  [9] "last_means"                "bioname"
## [11] "bioguide_id"               "born"
```

```
## [13] "died"                           "nominate_dim1"
## [15] "nominate_dim2"                  "nominate_log_likelihood"
## [17] "nominate_geo_mean_probability" "nominate_number_of_votes"
## [19] "nominate_number_of_errors"      "conditional"
## [21] "nokken_poole_dim1"              "nokken_poole_dim2"
```

To look at a spreadsheet version of your dataset (it will open in new tab):

```
View(voteview) # note the capital V
```

Note the `eval = FALSE` which instructs R not to try and run this line of code in `pdf` generation. Running the `chunk` with the green button will still work.

To see a snapshot of your data:

```
head(UNideal) # shows you the first 6 rows
```

```
## # A tibble: 6 x 13
##   ccode session NVotesAll IdealPointAll `Q0%All` `Q5%All` `Q10%All` `Q50%All`
##   <dbl>   <dbl>     <dbl>         <dbl>    <dbl>    <dbl>     <dbl>     <dbl>
## 1     2       1        42          1.71     1.08     1.40      1.46      1.70
## 2     2       2        38          1.94     1.20     1.55      1.61      1.92
## 3     2       3       103          1.92     1.41     1.58      1.63      1.92
## 4     2       4        63          1.80     1.37     1.51      1.57      1.80
## 5     2       5        53          1.64     1.13     1.29      1.36      1.64
## 6     2       6        25          1.75     1.25     1.45      1.50      1.74
## # ... with 5 more variables: `Q90%All` <dbl>, `Q95%All` <dbl>,
## #   `Q100%All` <dbl>, iso3c <chr>, Countryname <chr>
```

Understanding check: what is our unit of observation in this dataset?

To see dimensions of dataset:

```
dim(UNideal)
```

```
## [1] 10662    13
```

To see number of rows:

```
nrow(UNideal)
```

```
## [1] 10662
```

To see number of columns:

```
ncol(UNideal)
```

```
## [1] 13
```

Understanding check: How many variables does the dataset have? How many observations?

Accessing Variables:

There are two approaches to calling up specific variables from your dataset: 1) Using "attach" function 2) Specifying location of object

1) The "attach" function seems easier, but can create problems when using multiple datasets. See https://www.r-bloggers.com/to-attach-or-not-attach-that-is-the-question/

```
attach(voteview)
summary(nominate_dim1)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.     NA's
## -0.75800 -0.37850 -0.15800  0.05735  0.50200  0.93600        1
```

```
detach(voteview)
```

2) Rather than using "attach" function, it is usually better to specify object's location. To specify a variable within a dataset, use `$`.

```
summary(voteview$nominate_dim1)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.     NA's
## -0.75800 -0.37850 -0.15800  0.05735  0.50200  0.93600        1
```

Describing Variables:

Checking variable types

```
class(UNideal$session) # example of numeric variable
```

```
## [1] "numeric"
```

```
class(UNideal$Countryname) # example of character variable
```

```
## [1] "character"
```

Describing numeric variables

First, I'm going to get rid of some annoying data so that I don't have to include `na.rm = TRUE` in every line

```
voteview <- voteview[-1,]
```

Math camp folks, what is this doing?

```
summary(voteview$nominate_dim1) # 5-number summary (min, 1st, median, 3rd, max)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -0.75800 -0.37850 -0.15800  0.05735  0.50200  0.93600
```

```
sum(is.na(voteview$nominate_dim1)) # counting number of missing values
```

```
## [1] 0
```

```
mean(voteview$nominate_dim1) # mean value
```

```
## [1] 0.05735436
```

```
min(voteview$nominate_dim1) # minimum value
```

```
## [1] -0.758
```

```
max(voteview$nominate_dim1) # maximum value
```

```
## [1] 0.936
```

```
range(voteview$nominate_dim1) # minimum and maximum
```

```
## [1] -0.758  0.936
```

```
var(voteview$nominate_dim1) # variance
```

```
## [1] 0.2157473
```

```
sd(voteview$nominate_dim1) # standard deviation
```

```
## [1] 0.4644861
```

```
  sqrt(var(voteview$nominate_dim1)) # (also standard deviation)
```

```
## [1] 0.4644861
```

```
IQR(voteview$nominate_dim1) # interquartile range (3rd-1st quartile)
```

```
## [1] 0.8805
```

- Note that base R does not have a function for finding the mode.
- To do this easily, we can use the "modeest" package.

Only run this chunk once:

```
install.packages("modeest") # installing modeest package once/R installation
```

Then you can use this function:

```
library("modeest") # activating modeest package
mfv(voteview$nominate_dim1) # mfv = "most frequent value"
```

```
## [1] -0.367
```

Other ways to describe variables

```
unique(voteview$state_abbrev) # lists unique values for a variable
```

```
##  [1] "AL" "AK" "AZ" "AR" "CA" "CO" "CT" "DE" "FL" "GA" "HI" "ID" "IL" "IN" "IA"
## [16] "KS" "KY" "LA" "ME" "MD" "MA" "MI" "MN" "MS" "MO" "MT" "NE" "NV" "NH" "NJ"
## [31] "NM" "NY" "NC" "ND" "OH" "OK" "OR" "PA" "RI" "SC" "SD" "TN" "TX" "UT" "VT"
## [46] "VA" "WA" "WV" "WI" "WY"
```

```
length(unique(voteview$state_abbrev)) # counts number of unique values
```

```
## [1] 50
```

```
table(voteview$state_abbrev) # frequency table for the number of observations for each value of region
```

```
##
## AK AL AR AZ CA CO CT DE FL GA HI IA ID IL IN KS KY LA MA MD ME MI MN MO MS MT
##  3  9  6 11 56  9  7  3 29 17  4  6  4 20 11  6  8  9 11 10  4 16 10 10  6  3
## NC ND NE NH NJ NM NV NY OH OK OR PA RI SC SD TN TX UT VA VT WA WI WV WY
## 15  3  5  4 14  6  6 29 18  7  7 20  4  9  3 11 38  6 13  3 12 10  5  3
```

```
table(voteview$nominate_dim1) # not always useful for examining numeric variables!
```

```
##
## -0.758 -0.749 -0.709 -0.698 -0.682 -0.677 -0.653 -0.628 -0.626 -0.618 -0.614
##      1      1      1      1      1      1      1      1      1      1      1
## -0.606 -0.598 -0.591 -0.587 -0.585 -0.581 -0.574 -0.571 -0.564 -0.554 -0.552
##      2      1      1      1      1      1      1      1      1      1      1
## -0.547 -0.544  -0.54 -0.539 -0.538 -0.531 -0.528 -0.518 -0.516 -0.515 -0.513
##      1      1      1      2      1      2      1      1      1      1      1
## -0.512 -0.508 -0.497 -0.492  -0.49 -0.488 -0.486 -0.485 -0.484 -0.479 -0.478
##      1      2      1      1      1      2      2      2      1      1      1
## -0.473  -0.47 -0.468 -0.467 -0.465 -0.464 -0.461  -0.46 -0.459 -0.453 -0.451
##      1      1      1      1      1      2      2      2      1      1      2
##  -0.45 -0.448 -0.447 -0.443  -0.44 -0.439 -0.438 -0.434 -0.432 -0.431 -0.428
##      1      2      2      1      1      1      2      1      2      2      2
```

```
## -0.427 -0.425 -0.424 -0.423 -0.421  -0.42 -0.417 -0.415 -0.413  -0.41 -0.408
##      2      1      1      1      1      3      2      1      1      1      1
## -0.404 -0.403 -0.402 -0.401   -0.4 -0.399 -0.398 -0.397 -0.396 -0.395 -0.394
##      1      2      4      1      3      2      1      1      1      1      4
## -0.393 -0.392 -0.391  -0.39 -0.387 -0.385 -0.384 -0.382 -0.381  -0.38 -0.379
##      2      1      2      1      1      1      2      2      1      2      1
## -0.378 -0.376 -0.375 -0.374 -0.368 -0.367 -0.365 -0.364 -0.361  -0.36 -0.359
##      1      1      2      2      2      6      1      1      2      1      1
## -0.356 -0.355 -0.352  -0.35 -0.349 -0.348 -0.344 -0.343 -0.342 -0.341 -0.335
##      1      1      1      3      2      2      1      1      2      1      2
## -0.334 -0.332 -0.331  -0.33 -0.329 -0.322 -0.319 -0.317 -0.314 -0.312 -0.311
##      1      1      1      3      1      2      1      1      1      1      1
##  -0.31 -0.309 -0.308 -0.306 -0.304 -0.303 -0.302 -0.301 -0.297 -0.295 -0.293
##      2      2      1      1      2      1      3      1      2      1      1
## -0.292 -0.286 -0.285 -0.284 -0.283 -0.282  -0.28 -0.274 -0.271  -0.27 -0.269
##      3      3      2      1      3      1      2      1      1      1      1
## -0.267 -0.264 -0.262 -0.261  -0.26 -0.258 -0.256 -0.254 -0.248 -0.246 -0.244
##      1      1      1      1      1      1      1      2      2      1      1
## -0.243 -0.241 -0.239 -0.237 -0.236 -0.234  -0.23 -0.229 -0.228 -0.224 -0.215
##      3      1      2      1      1      1      1      3      2      1      1
## -0.214 -0.203 -0.196 -0.195 -0.194 -0.192 -0.187 -0.185 -0.181 -0.177  -0.17
##      1      1      1      1      1      2      1      2      1      2      1
## -0.166 -0.163 -0.159 -0.158 -0.147  -0.13 -0.112 -0.103 -0.057  0.115  0.166
##      1      1      1      1      1      1      1      1      1      1      1
##  0.167  0.197  0.212  0.217  0.239  0.252  0.255  0.256  0.258  0.259  0.263
##      1      1      1      1      1      1      1      1      1      1      3
##  0.266  0.267  0.268  0.272  0.274  0.275  0.276  0.281  0.284  0.297    0.3
##      1      2      1      2      1      1      1      1      1      1      2
##  0.302  0.304  0.311  0.313  0.314  0.316  0.331  0.332  0.333  0.338   0.34
##      2      1      1      1      1      1      1      1      1      1      1
##  0.348   0.35  0.354  0.355  0.356  0.357  0.359   0.36  0.363  0.364   0.37
##      1      1      1      1      1      1      1      1      1      1      1
##  0.372  0.373  0.374  0.378   0.38  0.381  0.388   0.39  0.392  0.394  0.397
##      1      1      2      1      1      2      1      1      1      1      1
##  0.398  0.402  0.404  0.406  0.409  0.411  0.415  0.416  0.417   0.42  0.424
##      2      1      3      1      1      1      1      1      3      2      1
##  0.425  0.426  0.427  0.428   0.43  0.433  0.437  0.438  0.439   0.44  0.441
##      1      1      2      2      2      1      1      1      2      2      2
##  0.442  0.443  0.444  0.447   0.45  0.452  0.453  0.456  0.457  0.458  0.462
##      1      1      1      2      1      1      1      2      1      1      2
##  0.465  0.467  0.474  0.476  0.477  0.484  0.487  0.494    0.5  0.502  0.503
##      1      3      2      2      2      1      2      1      1      2      1
##  0.505  0.506  0.507  0.509  0.513  0.515  0.516  0.517  0.518  0.521  0.524
##      1      1      1      1      1      1      1      2      1      1      3
##  0.527  0.528  0.531  0.532  0.533  0.535  0.536  0.545  0.546  0.548   0.55
##      1      1      1      2      1      2      1      1      4      1      1
##  0.551  0.555  0.559  0.562  0.564  0.566  0.569   0.57  0.573  0.575   0.58
##      1      2      1      1      1      1      1      1      1      3      1
##  0.581  0.582  0.584  0.585  0.586  0.588  0.592  0.593  0.595  0.597  0.598
##      1      1      1      3      1      1      1      1      1      1      1
##  0.599    0.6  0.601  0.602  0.604  0.617  0.622  0.624  0.625  0.628  0.629
##      1      1      2      1      2      1      1      1      1      1      1
##  0.631  0.632  0.634  0.636  0.637  0.638   0.64  0.641  0.643  0.647  0.654
##      1      1      1      1      1      1      1      1      1      1      1
```

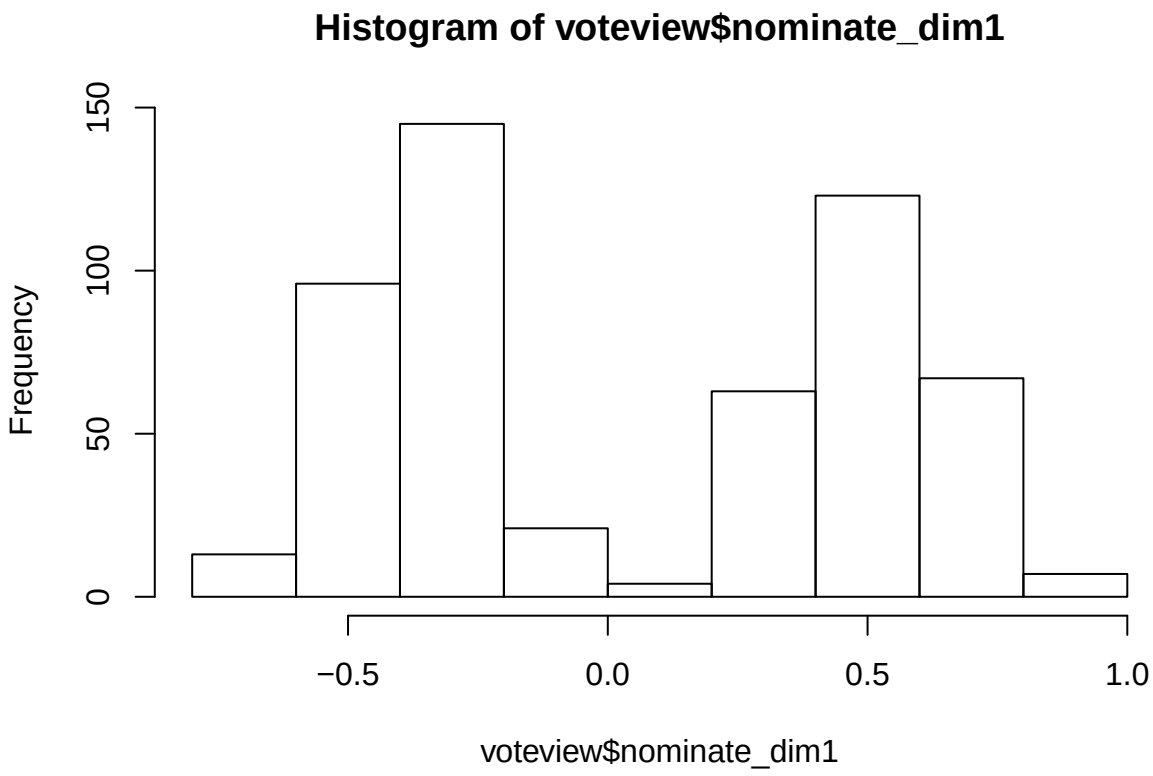
```
##  0.662  0.665  0.668   0.67  0.673  0.675  0.677  0.679  0.681  0.683  0.684
##      1      1      1      2      1      1      1      2      2      2      1
##  0.686   0.69  0.693  0.698  0.702  0.703  0.711  0.712  0.715  0.717  0.719
##      1      2      1      1      1      2      2      1      1      1      2
##  0.726   0.73  0.733  0.734  0.746  0.747  0.748  0.778  0.793  0.797  0.798
##      1      1      1      1      1      1      1      1      1      1      1
##    0.8  0.812  0.836  0.878  0.883  0.913  0.936
##      4      1      1      1      1      1      2
```

DESCRIBING VARIABLES WITH FREQUENCY AND DESCRIPTIVE PLOTS

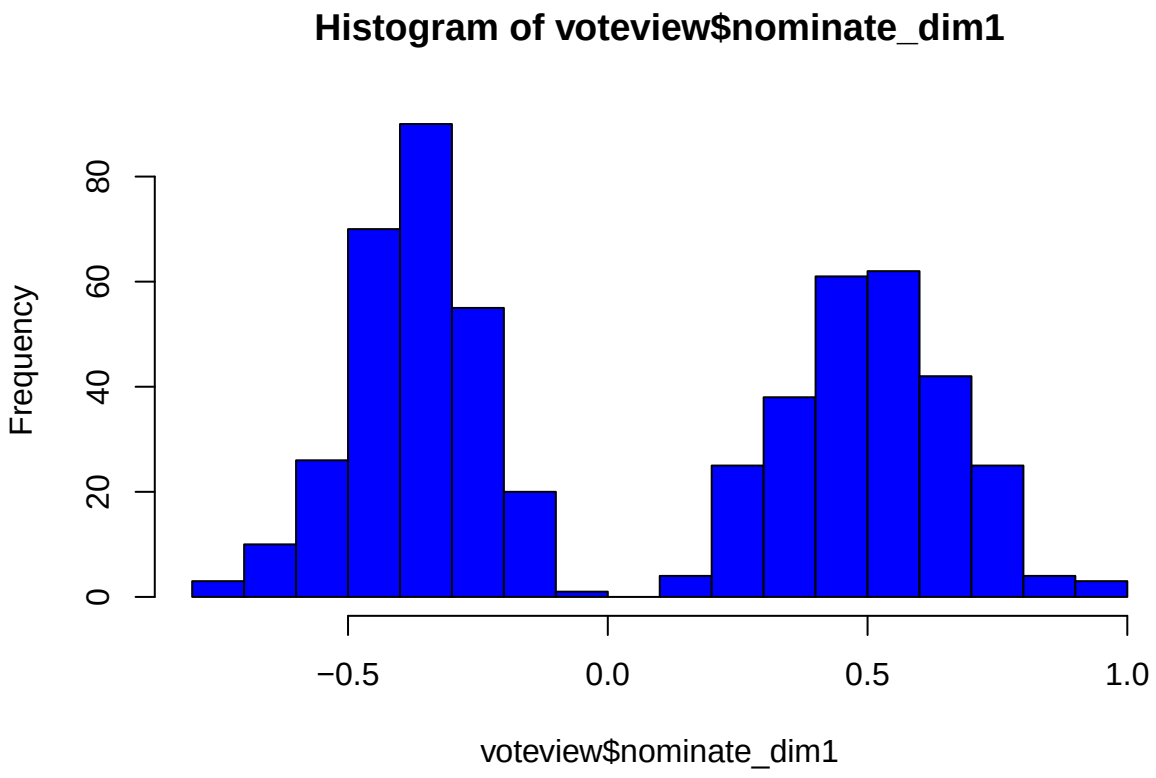
- note that you may have to set `dev.off()` to put settings back to default

Histograms

```
hist(voteview$nominate_dim1)
```

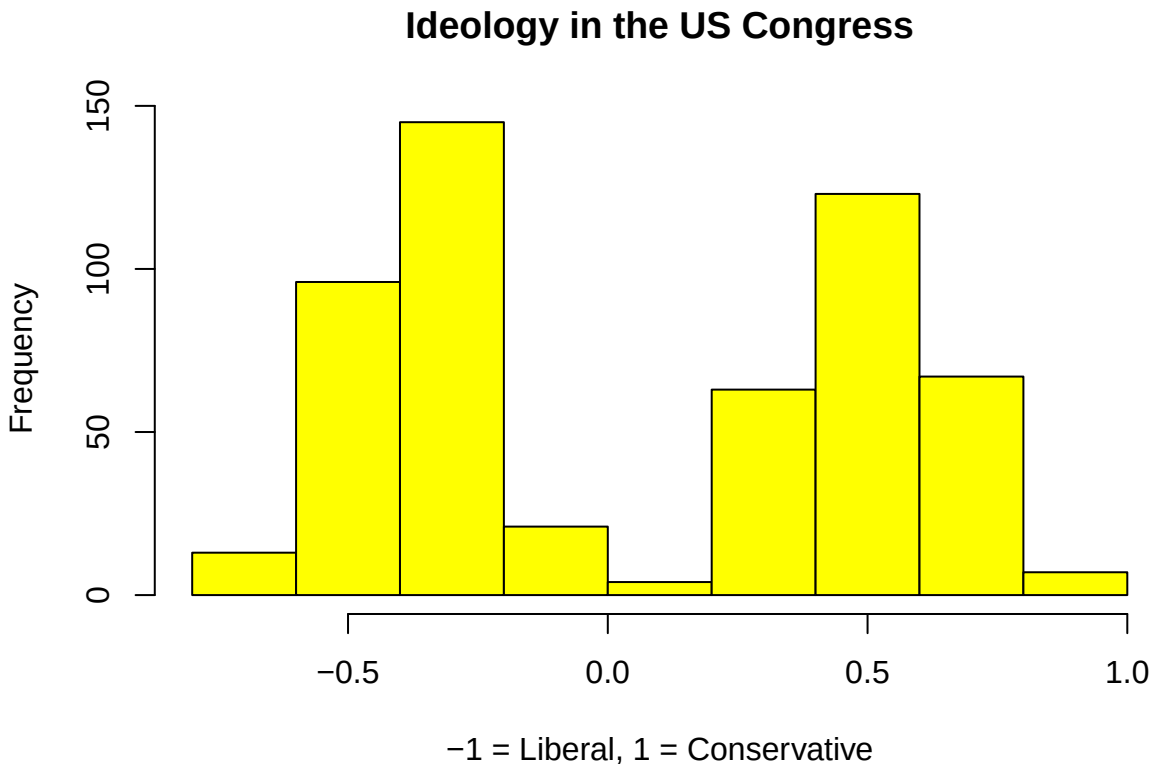



```
hist(voteview$nominate_dim1, breaks=20, col="blue") # changing number of bins and color
```



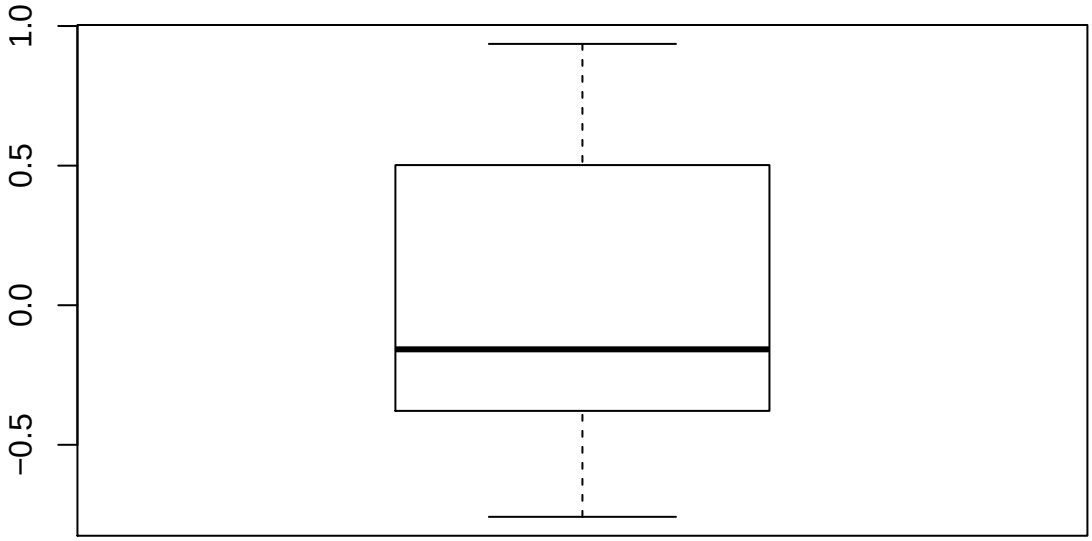

```
  # Note that R will not necessarily give you exactly the number of bins you specify

hist(voteview$nominate_dim1, breaks=8, col="yellow",
     main="Ideology in the US Congress",
     xlab="-1 = Liberal, 1 = Conservative") # adding titles and axis labels
```

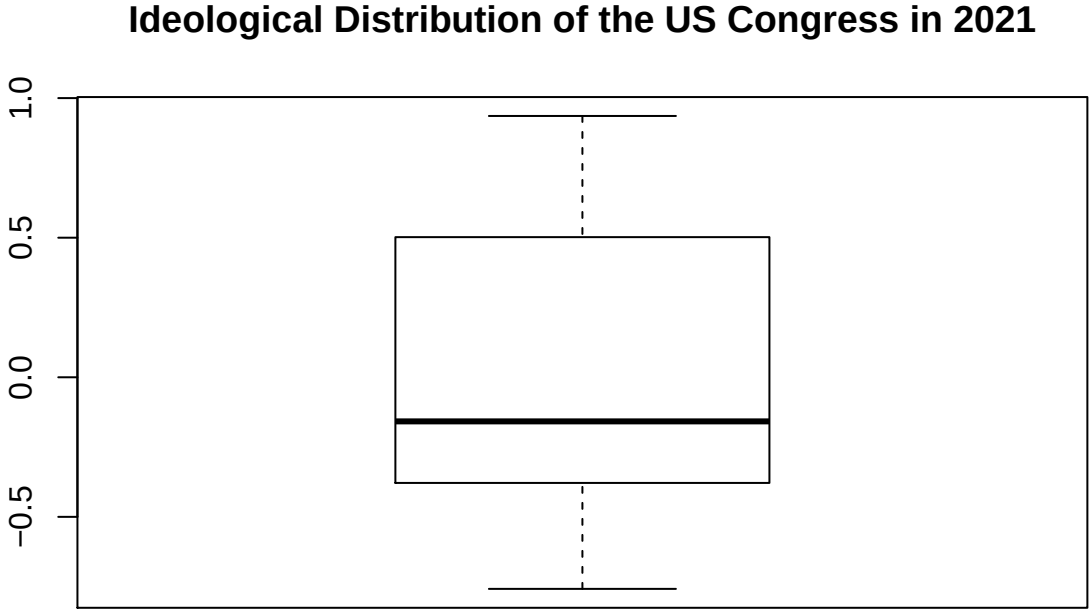


Box and whisker plots

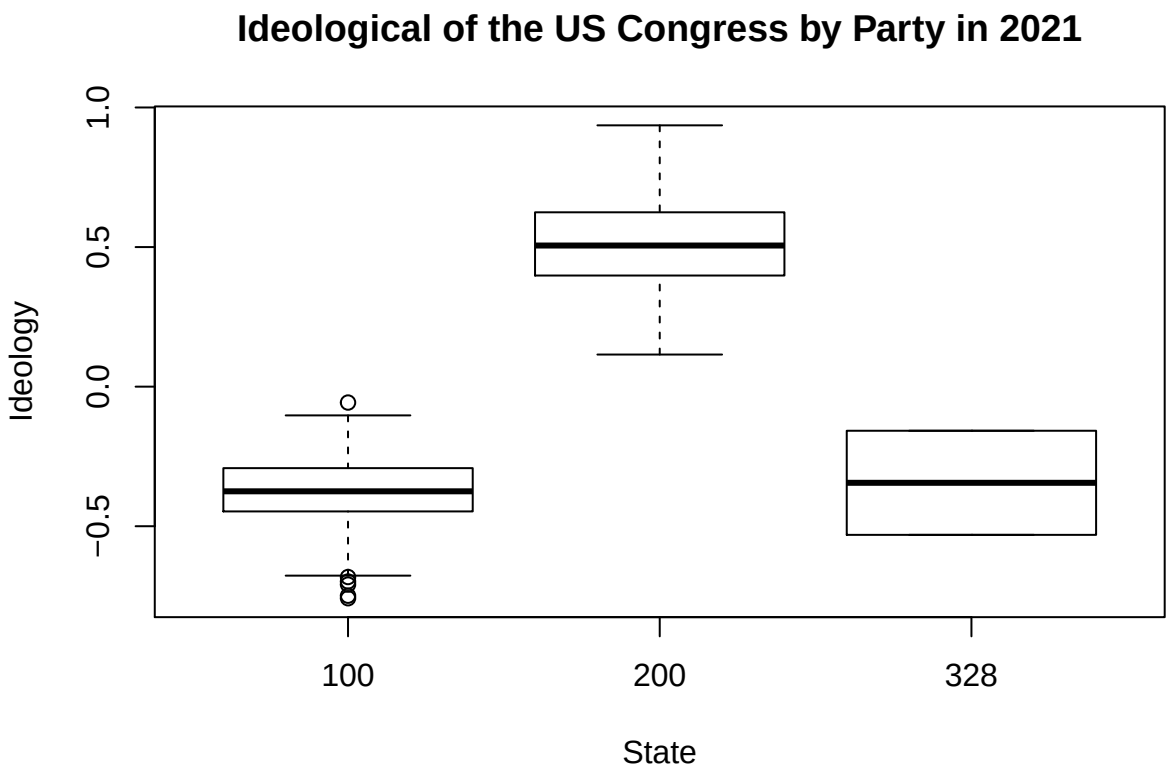
```
boxplot(voteview$nominate_dim1)
```




```
boxplot(voteview$nominate_dim1,
        main= "Ideological Distribution of the US Congress in 2021")
```




```
boxplot(nominate_dim1 ~ party_code, data=voteview,
        main= "Ideological of the US Congress by Party in 2021",
        xlab= "State", ylab="Ideology")
```



END

Note: when closing R, save your RProject file so that you can return to your datasets with all of your relevant code loaded. If you are not using an RProject, it is generally better to NOT save the "workspace image" i.e., the objects in the Environment) Starting with a clean workspace each session makes it easier to keep track of your objects and prevent programming issues. Be intentional about the project space you are working in!