```
#This data set comes from Bootstrap World. The project was created by
Marisa Laks with help from Alex Moore.

############################################################
# include Libraries we want
include shared-gdrive("Bootstrap-DataScience-v1.5.arr",
"1btFfKCcas4zkQ6-SYCYMkcDCqmduzQqB")
# include Google Sheets and Tables library
include gdrive-sheets
include tables
include image


############################################################
# Load your spreadsheet and define your table
election-2018-sheet = load-spreadsheet("1iPMuG-m-QEOnw55C5X9-
c6zY73sx2tpoZT93IJYussg")

election-table = load-table: state, population, percent-turnout,
percent-vote-dem, percent-vote-rep,
  total-seats, seats-dem, seats-rep, percent-seats-dem, percent-seats-
rep, winning-party,
  seats-match-vote
  source: election-2018-sheet.sheet-by-name("Sheet1", true)
end


############################################################
# Part 1: Look at the spreadsheet "Gerrymandering Dataset
(Bootstrap)." Choose three states and define the rows below.

#State #1:
alabama = election-table.row-n(0)

#State #2:
new-york = election-table.row-n(31)

#State #3:
florida = election-table.row-n(8)


############################################################
# Part 2: Define a table called "seats-sort" that sorts the table by
total seats starting with the most seats.

seats-sort = election-table.order-by("total-seats", false)



############################################################
# Part 3a: Write a function called "is-dem-win" that consumes a row
and produces a Boolean that returns true if the winning party is
Democratic.

# Write three examples using your rows defined above.
```

```
examples:
  is-dem-win(alabama) is alabama["winning-party"] == "Democratic"
  is-dem-win(new-york) is new-york["winning-party"] == "Democratic"
  is-dem-win(florida) is florida["winning-party"] == "Democratic"
end


# Define the function.

fun is-dem-win(row):row["winning-party"] == "Democratic" end


# Part 3b: Define a table that only contains rows of states where the
Democrats were the winning party.

dem-win = election-table.filter(is-dem-win)


############################################################
# Part 4: Repeat the process from part 3 to create a function and
table where the winning party was Republican.


examples:
  is-rep-win(alabama) is alabama["winning-party"] == "Republican"
  is-rep-win(new-york) is new-york["winning-party"] == "Republican"
  is-rep-win(florida) is florida["winning-party"] == "Republican"
end

fun is-rep-win(row): row["winning-party"] == "Republican" end

rep-win = election-table.filter(is-rep-win)


############################################################
# Part 5a: Given the function "do-seats-match-vote." Write a purpose
statement for what the function does. It is helpful to apply the
function to your defined rows first.


fun do-seats-match-vote(row):row["seats-match-vote"] == false end


# Purpose statement: The function "do-seats-match-vote" consumes a row
and returns true if the seats do not match the vote.


# Part 5b: Define a table called "seats-vote" that filters the table
by the function "do-seats-match-vote".


seats-vote = election-table.filter(do-seats-match-vote)

############################################################
# Part 6: Create at least three data displays for your defined tables.
Write the code for each in the definitions area. Explain what the
charts show.
```

```
#Chart 1:

text("Winning Party for Seats that do not match the vote", 12,
"black")
bar-chart(seats-vote, "winning-party")


#Explanation: This chart shows the number states where the seats do
not match the vote for each party.


#Chart 2:

text("Winning Party is Republican for Seats that Do Not Match the
Vote", 12, "black")
pie-chart(rep-win, "seats-match-vote")

#Explanation: This chart shows the percentage of states where the
winning party is Republican and if the seats match the vote.


#Chart 3:

text("Winning Party is Democrat for Seats the Do Not Match the Vote",
12, "black")
pie-chart(dem-win, "seats-match-vote")

#Explanation: This chart shows the percentages of states where the
winning party is Democratic and if the seats match the vote.


############################################################

# Part 7: What can you conclude about Gerrymandering from the
exploration in the previous parts?


# Conclusion: Although gerrymandering appears to happen in both
political parties, it appears to happen more frequently in states with
more Republican control.




############################################################

# Extension: Demonstrate anything else we've done in this class. For
example, you can define other tables or show different displays.
Explain what your code does.


gerrymander-dem = dem-win.filter(do-seats-match-vote)
gerrymander-rep = rep-win.filter(do-seats-match-vote)

# These tables show only the states where the seats do not match the
```

votes for each winning party.