Applied Statistical Programming - ggplot

2/28/2022

Patrick Edwards

Amaan Charaniya

Alex Avery

Peter Bachman

Write the R code to answer the following questions. Write the code, and then show what the computer returns when that code is run. Thoroughly comment your solutions.

You have until the beginning of class 3/2 at 10:00am to complete the assignment below. You may use R, but not any online R documentation. Submit the Rmarkdown and the knitted PDF to Canvas. Have one group member submit the activity with all group members listed at the top.

Figuring out the Competition

You've been hired by a campaign to do some data analysis during the primary stage of an election. The campaign wants to understand competitiveness of certain candidates under different general election scenarios. You will plot some summary features of the provided `primaryPolls` data using `ggplot()`.

The data is associated with 2020 Democratic primary elections. Polling results for 38 states are provided. You will create a visualization of the state of the race using this data. For three states of your choosing, generate a summary figure that visualizes the support for each candidate in that state. Each plot must include:

- a title,
- a subtitle,
- labeled axes,
- a legend for the candidates, and
- a source attribution to the GitHub URL for the data.

```
# Remove eval=FALSE to have this code block run.

# Load library dependencies
library(dplyr)
library(tidyr)
library(readr)

# Define path to the data
dataURL <- "https://jmontgomery.github.io/PDS/Datasets/president_primary_polls_feb2020.csv"
# Load the data
primaryPolls <- read_csv(dataURL)
```

```
## Rows: 16661 Columns: 33
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (21): state, pollster, sponsors, display_name, pollster_rating_name, fte...
## dbl  (8): question_id, poll_id, cycle, pollster_id, pollster_rating_id, samp...
## lgl  (3): internal, tracking, nationwide_batch
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
# Format the date
primaryPolls$start_date <- as.Date(primaryPolls$start_date, "%m/%d/%y")
```

NOTE 1: primaryPolls's unit of analysis (i.e., each observation/row) is on the candidate level. Notable variables:

- **start_date**: Survey's initiation date.
- **end_date**: Survey's termination date.
- **party**: Political party of candidate (this polling data *does* include some Republican candidates).
- **candidate_name**: name of candidate.
- **pct**: estimated proportion of the population that supports the candidate.

STEP 1: Fix errors in and consolidate the data.

- Drop Republican candidates.

```
unique(primaryPolls$party)
```

```
## [1] "DEM" "REP"
```

```
primaryPolls <- primaryPolls[which(primaryPolls$party == "DEM"), ]
unique(primaryPolls$party)
```

```
## [1] "DEM"
```

```
# Only Democratic candidates remain.
```

- Fix Julian Castro's name. I think the accents didn't load into the original dataset, so I don't include the accent.

```
unique(primaryPolls$candidate_name[primaryPolls$answer == "Castro"])
```

```
## [1] "Juli<cc><c1>n Castro"
```

```
sum(is.na(primaryPolls$answer))
```

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

```
sum(is.na(primaryPolls$candidate_name))
```

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

```
primaryPolls[primaryPolls$answer == "Castro", "candidate_name"] <- "Julian Castro"

unique(primaryPolls$candidate_name[primaryPolls$answer == "Castro"])
```

```
## [1] "Julian Castro"
```

```
# candidate_name now uses all unique values.
```

- Ensure that there are a manageable number of Democratic candidates.

```
length(unique(primaryPolls$candidate_name))
```

```
## [1] 59
```

```
# Pollsters included every notable person in some of the early polls.  We can't
# have 59 Democratic candidates on our graphs!
unique(primaryPolls$candidate_name)
```

```
##  [1] "Bernard Sanders"        "Andrew Yang"            "Pete Buttigieg"
##  [4] "Joseph R. Biden Jr."    "Tulsi Gabbard"          "Amy Klobuchar"
##  [7] "Elizabeth Warren"       "Tom Steyer"             "Michael Bloomberg"
## [10] "Michael F. Bennet"      "Deval Patrick"          "Cory A. Booker"
## [13] "Marianne Williamson"    "John K. Delaney"        "Nathan Bloxham"
## [16] "Roque De La Fuente"     "Julian Castro"          "Robert Wells"
## [19] "Kamala D. Harris"       "Beto O'Rourke"          "Steve Bullock"
## [22] "Donald Trump"           "Wayne Messam"           "Kirsten E. Gillibrand"
## [25] "Mike Gravel"            "Eric Swalwell"          "Tim Ryan"
## [28] "Hillary Rodham Clinton" "Bill de Blasio"         "Jay Robert Inslee"
## [31] "Joe Sestak"             "John Kerry"             "John Hickenlooper"
## [34] "Michelle Obama"         "Seth Moulton"           "Terry R. McAuliffe"
## [37] "Stacey Yvonne Abrams"   "Howard Schultz"         "Barack Obama"
## [40] "Dwayne Johnson"         "Margaret Wood Hassan"   "Michael Avenatti"
## [43] "Andrew Cuomo"           "Eric H. Holder"         "Sherrod Brown"
## [46] "Oprah Winfrey"          "Mark Zuckerberg"        "Tim Kaine"
## [49] "Eric Garcetti"          "Gavin Newsom"           "Nancy Pelosi"
## [52] "Gary Johnson"           "Joseph Kennedy III"     "Dennis J. Kucinich"
## [55] "Jia Lee"                "Robert C. Scott"        "Kyrsten Sinema"
## [58] "Mark R. Warner"         "Richard Neece Ojeda"
```

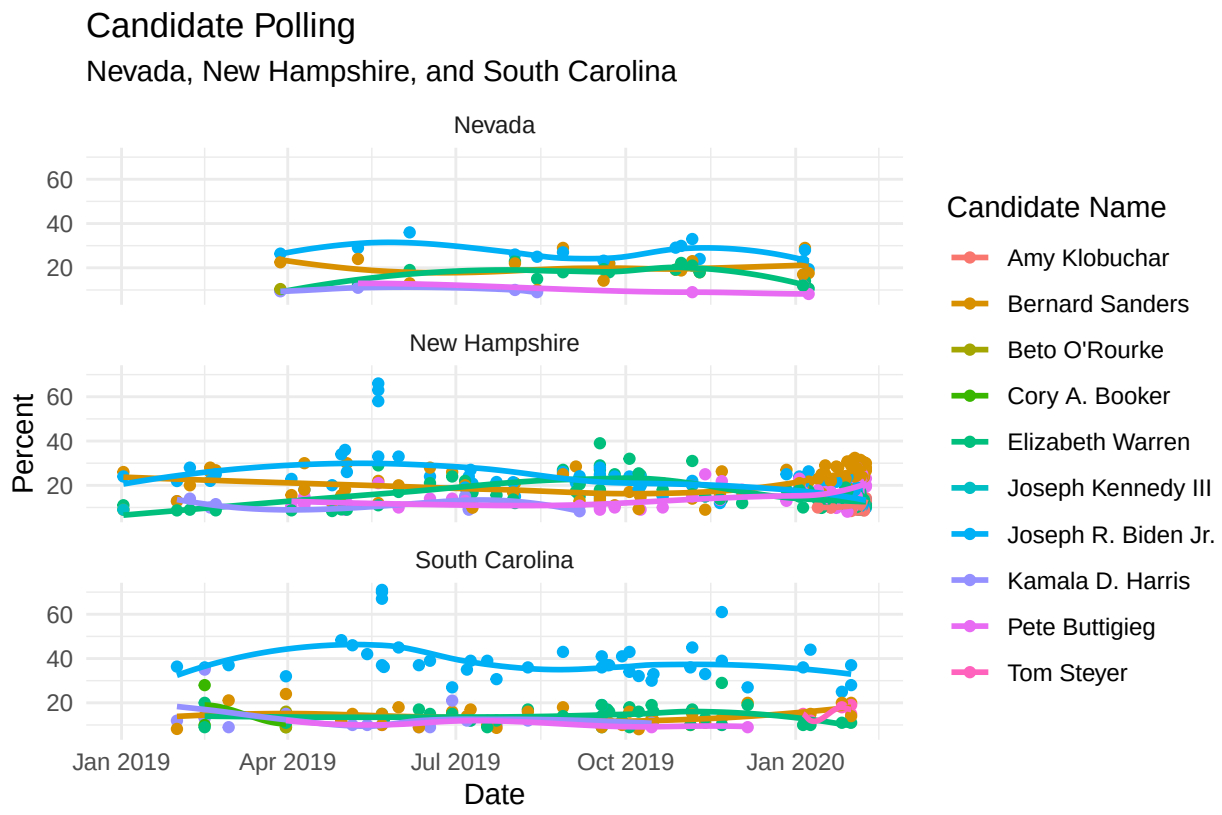
Step 2: Separate out data for each state.

* Overall graph.

```r
# We partitioned our 3 states
newhampshire <- primaryPolls[which(primaryPolls$state == "New Hampshire"), ]
nevada <- primaryPolls[which(primaryPolls$state == "Nevada"), ]
southcarolina <- primaryPolls[which(primaryPolls$state == "South Carolina"), ]
# and created one dataset
states <- rbind(newhampshire, southcarolina, nevada)

# This is a graph of each state we chose with the candidates polling percentage
# over time. We identified candidates above 8 percent and left out Michelle
# Obama because she only had one poll at this threshold back in October.
ggplot(data = states[which(states$pct > 8 & states$candidate_name != "Michelle Obama"),
    ], aes(x = start_date, y = pct, color = candidate_name)) + facet_wrap(~state,
    nrow = 3) + ggtitle("Candidate Polling", subtitle = "Nevada, New Hampshire, and South Carolina") +
    labs(color = "Candidate Name") + xlab("Date") + ylab("Percent") + geom_point() +
    geom_smooth(se = FALSE) + theme_minimal() + labs(caption = "Source: https://jmontgomery.github.io/PI
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : span too small. fewer data values than degrees of freedom.

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : pseudoinverse used at 17982

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : neighborhood radius 127.69

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : reciprocal condition number 0

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : There are other near singularities as well. 9544.3

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : span too small. fewer data values than degrees of freedom.

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : pseudoinverse used at 18024

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : neighborhood radius 182.22

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : reciprocal condition number 0

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : There are other near singularities as well. 4124.2

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : span too small. fewer data values than degrees of freedom.
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : pseudoinverse used at 17942

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : neighborhood radius 44.22

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : reciprocal condition number 0

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : There are other near singularities as well. 1955.4

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : span too small. fewer data values than degrees of freedom.

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : pseudoinverse used at 18266

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : neighborhood radius 21.13

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : reciprocal condition number 0

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : There are other near singularities as well. 26.317
```



ce: https://jmontgomery.github.io/PDS/Datasets/president_primary_polls_feb2020.csv

```
# plot.title = element_text(hjust = 0.5), plot.subtitle.title =
# element_text(hjust = 0.5), plot.caption = element_text(hjust = 0.5) )
```

* State-Level Graphs.

```
ggplot(data = nevada[which(nevada$pct > 8 & nevada$candidate_name != "Michelle Obama"),
    ], aes(x = start_date, y = pct, color = candidate_name)) + ggtitle("Candidate Polling",
    subtitle = "Nevada") + labs(color = "Candidate Name") + xlab("Date") + ylab("Percent") +
    geom_point() + geom_smooth(se = FALSE) + theme_minimal() + labs(caption = "Source: https://jmontgom
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : span too small. fewer data values than degrees of freedom.
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : pseudoinverse used at 17982
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : neighborhood radius 127.69
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : reciprocal condition number 0
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : There are other near singularities as well. 9544.3
```

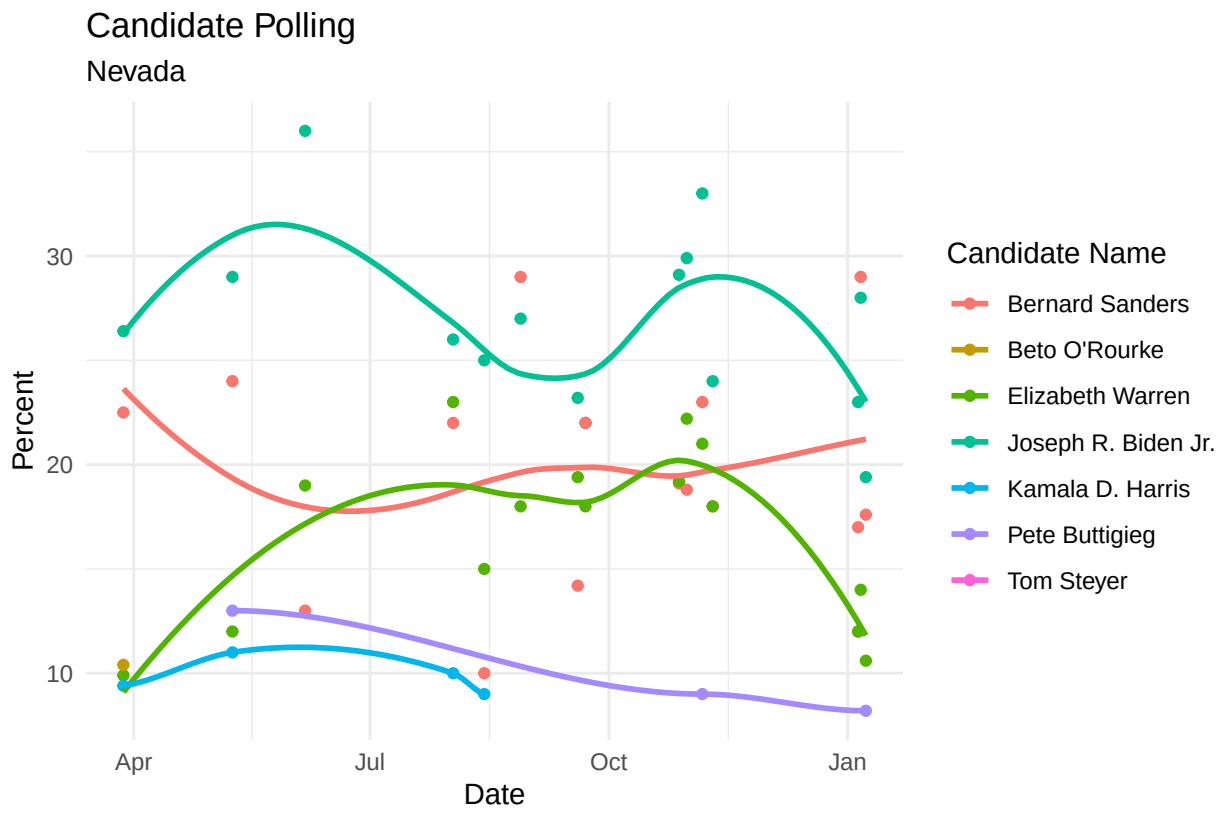
```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : span too small. fewer data values than degrees of freedom.
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : pseudoinverse used at 18024
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : neighborhood radius 182.22
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : reciprocal condition number 0
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : There are other near singularities as well. 4124.2
```



ce: https://jmontgomery.github.io/PDS/Datasets/president_primary_polls_feb2020.csv

```
ggplot(data = southcarolina[which(southcarolina$pct > 8 & southcarolina$candidate_name !=
    "Michelle Obama"), ], aes(x = start_date, y = pct, color = candidate_name)) +
    ggtitle("Candidate Polling", subtitle = "South Carolina") + labs(color = "Candidate Name") +
    xlab("Date") + ylab("Percent") + geom_point() + geom_smooth(se = FALSE) + theme_minimal() +
    labs(caption = "Source: https://jmontgomery.github.io/PDS/Datasets/president_primary_polls_feb2020.
```

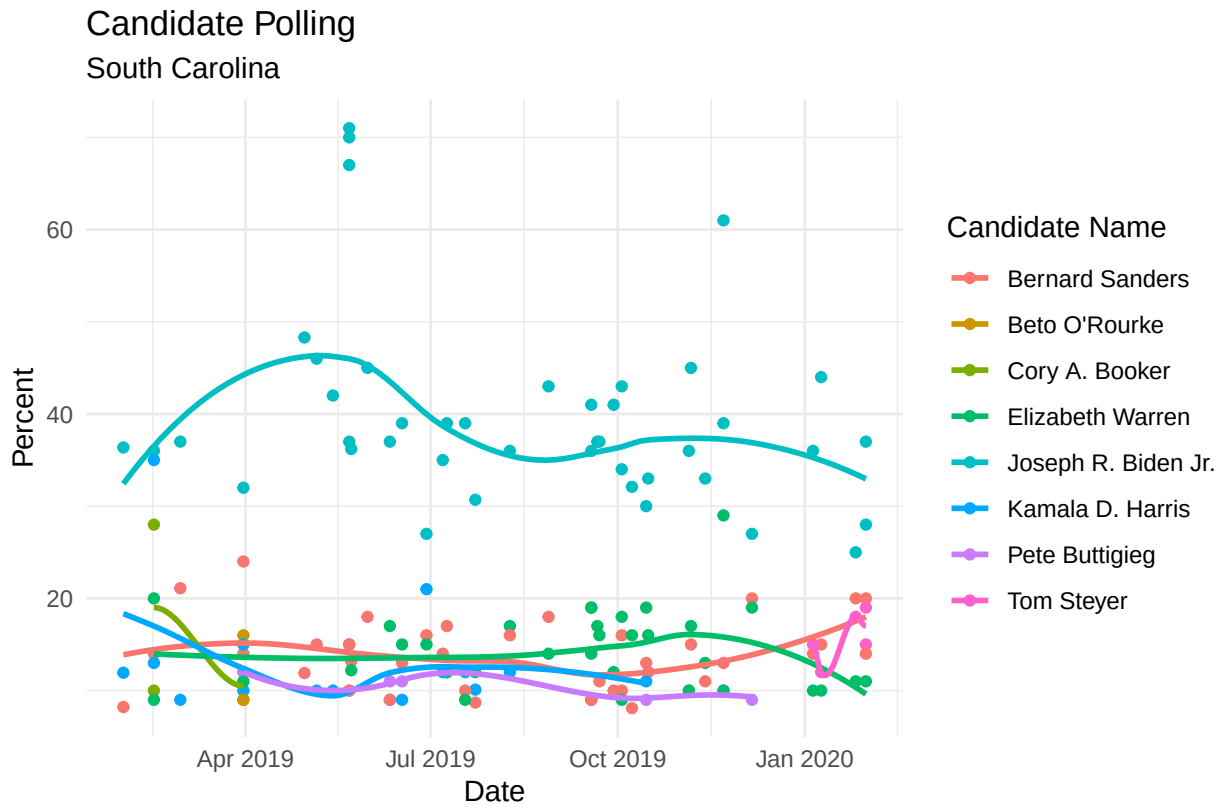
```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : span too small. fewer data values than degrees of freedom.

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : pseudoinverse used at 17942

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : neighborhood radius 44.22

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : reciprocal condition number 0

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : There are other near singularities as well. 1955.4

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : span too small. fewer data values than degrees of freedom.
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : pseudoinverse used at 18266
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : neighborhood radius 21.13
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : reciprocal condition number 0
```

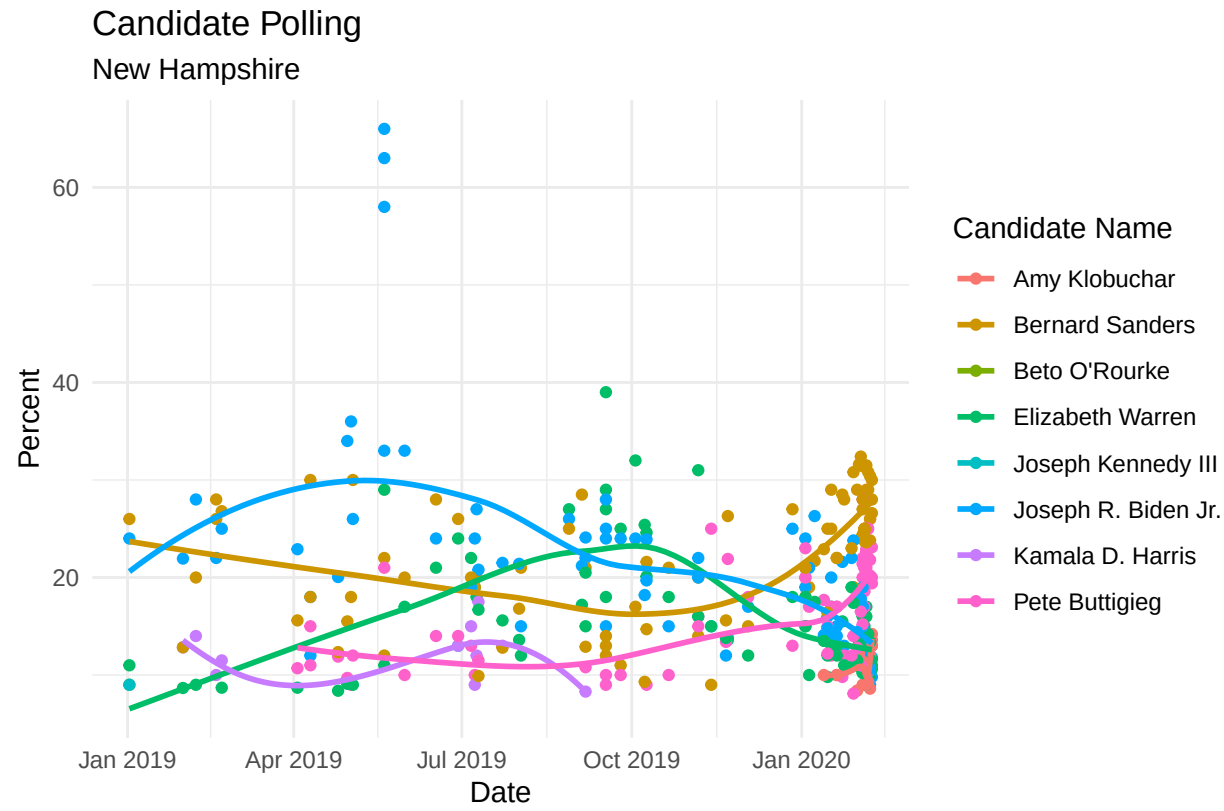
```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : There are other near singularities as well. 26.317
```



ce: https://jmontgomery.github.io/PDS/Datasets/president_primary_polls_feb2020.csv

```
ggplot(data = newhampshire[which(newhampshire$pct > 8 & newhampshire$candidate_name !=
    "Michelle Obama"), ], aes(x = start_date, y = pct, color = candidate_name)) +
    ggtitle("Candidate Polling", subtitle = "New Hampshire") + labs(color = "Candidate Name") +
    xlab("Date") + ylab("Percent") + geom_point() + geom_smooth(se = FALSE) + theme_minimal() +
    labs(caption = "Source: https://jmontgomery.github.io/PDS/Datasets/president_primary_polls_feb2020.
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



ce: https://jmontgomery.github.io/PDS/Datasets/president_primary_polls_feb2020.csv