

investigation-US-election-2016

```
library(conflicted)
  conflicts_prefer(dplyr::filter, dplyr::lag)
library(tidyverse)
library(R6)
library(formatR)
library(lubridate)
library(rmarkdown)
```

```
allstates = polls$state |>
  unique() |>
  sort()
polls = polls |>
  mutate(
    interval = startdate %--% enddate,
    middate = ymd(startdate + (enddate - startdate)/2),
    .keep = "unused",
    .after = "state"
  ) |>
  mutate(
    grade = grade |>
      factor(
        levels = c(
          "A+",
          "A",
          "A-",
          "B+",
          "B",
          "B-",
          "C+",
          "C",
          "C-",
          "D+",
          "D",
          "D-",
          NA
        )
      ),
    .keep = "unused",
    .before = "samplesize"
  )
```

```
finalresults = tibble(
  state = allstates,
  clinton = c(34.36, 36.55, 45.13, 33.65, 61.73, 48.16, 54.57, 53.09, 90.86, 47.82, 45.64, 62.22, 27.
  trump = c(62.08, 51.28, 48.67, 60.57, 31.62, 43.25, 40.93, 41.71, 4.09, 49.02, 50.77, 30.04, 59.26,
```

```

johnson = c(2.09, 5.88, 4.13, 2.65, 3.37, 5.18, 2.96, 3.33, 1.58, 2.20, 3.05, 3.72, 4.10, 3.79, 4.9
mcmullin = c(NA, NA, 0.68, 1.17, 0.28, 1.04, 0.13, 0.16, NA, NA, 0.32, NA, 6.73, 0.21, NA, 0.79, 0.
)

```

```

# helps filter for candidate name in results
candidatenames = c("clinton", "trump", "johnson", "mcmullin")

candidate = R6Class(
  classname = "candidate",
  public = list(
    name = "character",
    opponents = "character",
    colour = "character",
    polls = "tbl_df",
    finalresults = "tbl_df",

    initialize = function(name, colour){
      self$name = name
      self$colour = colour
      self$opponents = candidatenames[candidatenames != self$name] # the list of other candidates
      self$polls = polls |>
        select(!contains(self$opponents)) |> # filter polls to only this candidate
        rename(
          rawpolls = starts_with("rawpoll"),
          adjpolls = starts_with("adjpoll")
        ) # rename columns for use in candidateplot functions
      self$finalresults = finalresults |>
        select(!contains(self$opponents))
    } # end of initialize
  ) # end of list
)

```

```

clinton = candidate$new(name = "clinton", colour = "blue")
trump = candidate$new(name = "trump", colour = "red")
johnson = candidate$new(name = "johnson", colour = "orange")
mcmullin = candidate$new(name = "mcmullin", colour = "purple")

```

```

candidates = list(clinton, trump, johnson, mcmullin)
rm(candidatenames) # only needed for candidate construction

```

```

densepollsbegin = function(thestate, thedate) { # for graphing purposes
  # calculate the earliest date after thedate (inclusive) where there are three
# consecutive polls in short succession (within two months)
  statepolls = polls |>
    filter(state == thestate, middate >= thedate) |>
    arrange(middate)
  earlydate = statepolls$middate |>
    min()
  while (nrow(statepolls) > 2) {
    statepolls = statepolls[-1,]
    mos = interval(
      start = earlydate,
      end = earlydate %m+% months(2, abbreviate = FALSE)
    )
  }
}

```

```

    )
    if (
      mos |>
      int_overlaps(
        statepolls$interval[which.min(sapply(statepolls$interval, int_start))]
      )
    ) {
      temppolls = statepolls[-1,]
      if (
        mos |>
        int_overlaps(
          temppolls$interval[which.min(sapply(temppolls$interval, int_start))]
        )
      ) {
        return(earlydate)
      }
    }
    earlydate = statepolls$midddate |>
      min()
  }
  return(earliestdate) # failed to find three such polls; just graph all polls
}

```

```

earliestdate = ymd("2016-08-01")
finaldate = ymd("2016-11-08")
dayafter = finaldate + days(1) # used just for technical graphing purposes

```

```

candidateplotraw = function(thestate, thecandidate, thefirstdate) {
  statepolls = thecandidate$polls |>
    filter(state == thestate)

  # methods to avoid plotting Johnson's numbers where his polls are insignificant
  # check whether Johnson obtained more than 13% raw in any poll conducted after Sep. 1st
  if (thecandidate$name == "johnson" && !(thestate %in% congdistricts)) {
    statepolls = statepolls |>
      filter(midddate >= ymd("2016-09-01"))
    if (max(statepolls$rawpolls, na.rm = TRUE) <= 13.0) {
      return(last_plot())
    }
  }

  if ( # in this state, not all polls for this candidate are NA
    any(!is.na(statepolls$rawpolls))
  ) {
    datevsraw = aes(
      x = statepolls$midddate,
      y = statepolls$rawpolls
    )
    voterresult = filter(thecandidate$finalresults, state == thestate)[1,2] |>
      as.numeric()
    return(
      last_plot() + geom_point(
        mapping = datevsraw,

```

```

        colour = thecandidate$colour,
        alpha = 0.8,
        na.rm = TRUE
    ) + geom_smooth(
        mapping = datevsraw,
        colour = thecandidate$colour,
        fill = thecandidate$colour,
        alpha = 0.4,
        na.rm = TRUE
    ) + geom_segment(
        mapping = aes(
            x = ymd("2016-11-01"),
            y = voterresult,
            xend = dayafter,
            yend = voterresult
        ),
        colour = thecandidate$colour,
        linewidth = 1.0
    )
    ) # end of return
} # end of if
else return(last_plot()) # catch the case where the candidate was not polled in the given state
}

candidateplotadj = function(theestate, thecandidate, thefirstdate) {

    statepolls = thecandidate$polls |>
        filter(state == theestate) # candidate's polling numbers in the given state

    # methods to avoid plotting Johnson's numbers where his polls are insignificant
    # check whether Johnson obtained more than 13% (raw! for comparison to plots of raw polls) in any p
    if (thecandidate$name == "johnson") {
        statepolls = statepolls |>
            filter(midddate >= ymd("2016-09-01"))
        if (max(statepolls$rawpolls, na.rm = TRUE) <= 13.0) {
            return(last_plot())
        }
    }

    if ( # in this state, not all polls for this candidate are NA
        any(!is.na(statepolls$adjpolls))
    ) { # so we can display this candidate's polls here
        datevsadj = aes(
            x = statepolls$midddate,
            y = statepolls$adjpolls
        )
        voterresult = filter(thecandidate$finalresults, state == theestate)[1,2] |>
            as.numeric()
        return(
            last_plot() + geom_point(
                mapping = datevsadj,
                colour = thecandidate$colour,
                alpha = 0.8,

```

```

        na.rm = TRUE
      ) + geom_smooth(
        mapping = datevsadj,
        colour = thecandidate$colour,
        fill = thecandidate$colour,
        alpha = 0.4,
        na.rm = TRUE
      ) + geom_segment(
        mapping = aes(
          x = ymd("2016-11-01"),
          y = voterresult,
          xend = dayafter,
          yend = voterresult
        ),
        colour = thecandidate$colour,
        linewidth = 1.0
      )
    ) # end of return
  } # end of if
  else return(last_plot()) # catch the case where the candidate was not polled in the given state
}

```

```

stateplotraw = function(theystate, firstdatetoplot) {
  plot = ggplot()
  for (cand in candidates){
    # McMullin was not a factor outside of Utah
    # don't attempt to plot his polls in other states
    if(cand$name != "mcmullin" || theystate == "Utah"){
      plot = candidateplotraw(theystate, cand, firstdatetoplot)
    }
  } # end of for
  stateraw = theystate |>
    paste("Raw Polls", sep = " - ")

  return(
    plot + geom_vline(xintercept = finaldate) + labs(
      title = stateraw,
      x = "Date",
      y = "Voting Intent (%)"
    ) + xlim(firstdatetoplot, dayafter)
  )
}

```

```

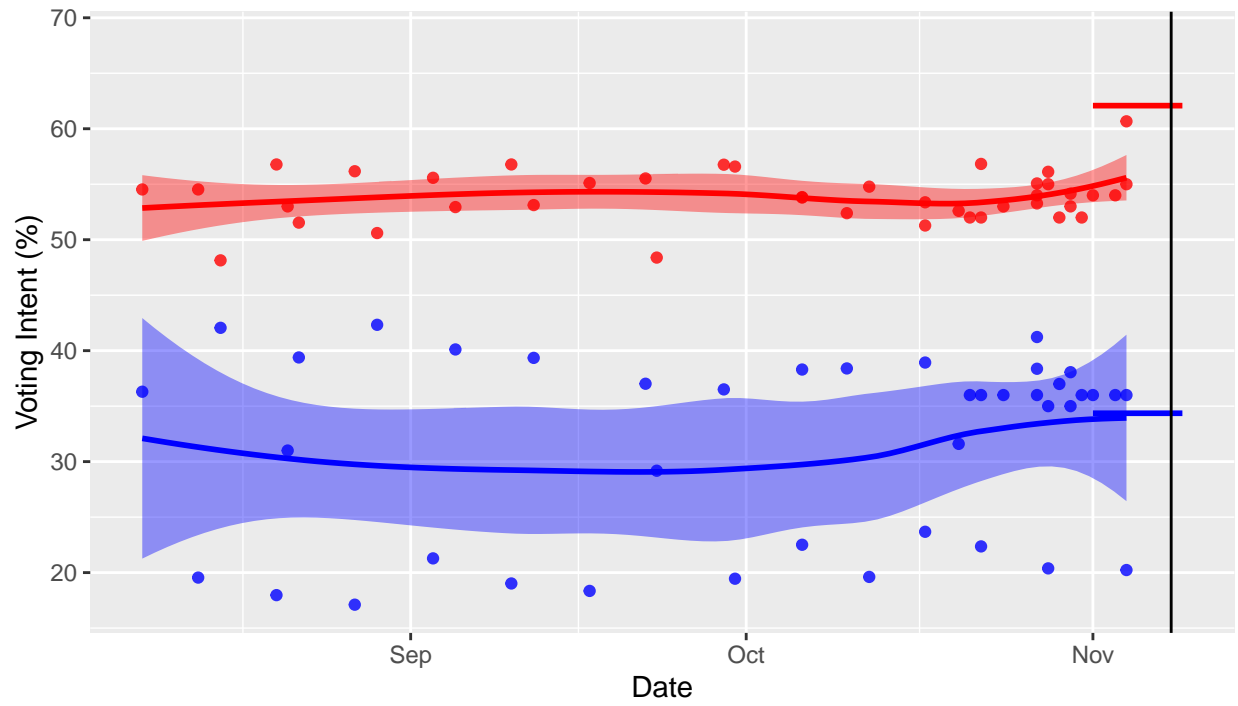
stateplotadj = function(theystate, firstdatetoplot) {
  plot = ggplot()
  for (cand in candidates){
    plot = candidateplotadj(theystate, cand, firstdatetoplot)
  }
  stateadj = theystate |>
    paste("Adjusted Polls", sep = " - ")

  return(
    plot + geom_vline(xintercept = finaldate) + labs(

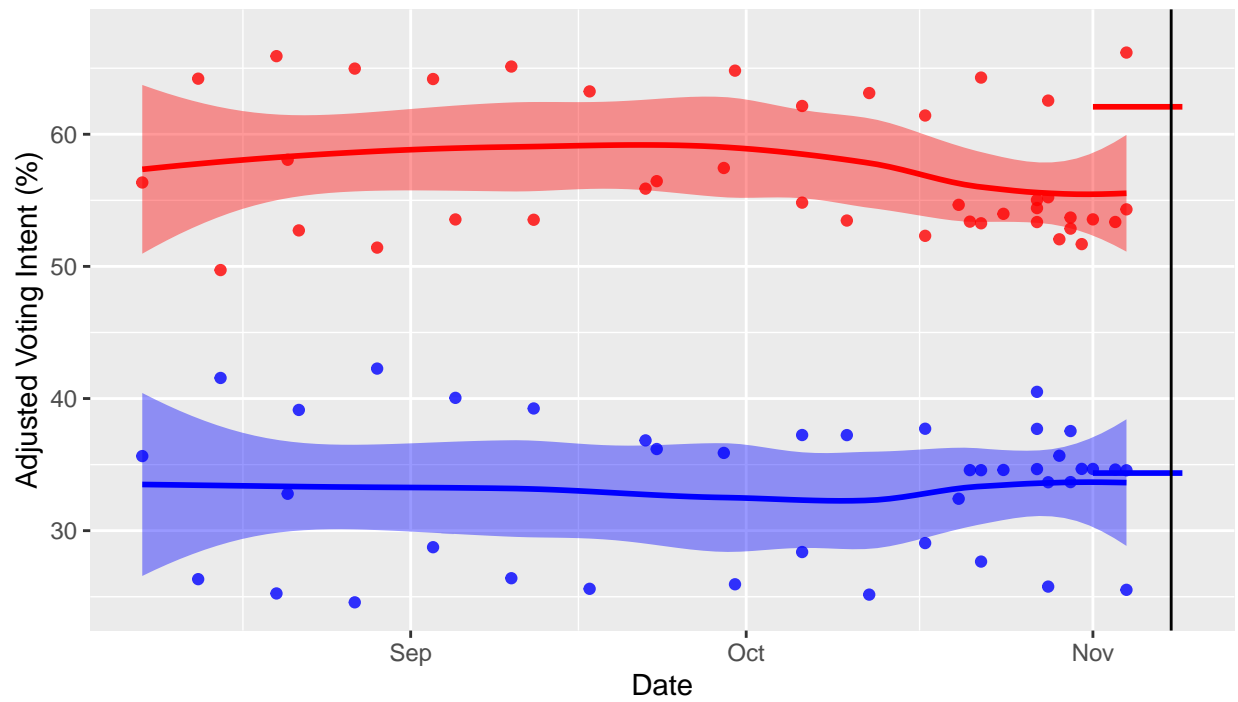
```

```
        title = stateadj,  
        x = "Date",  
        y = "Adjusted Voting Intent (%)"  
    ) + xlim(firstdate, topdate)  
}
```

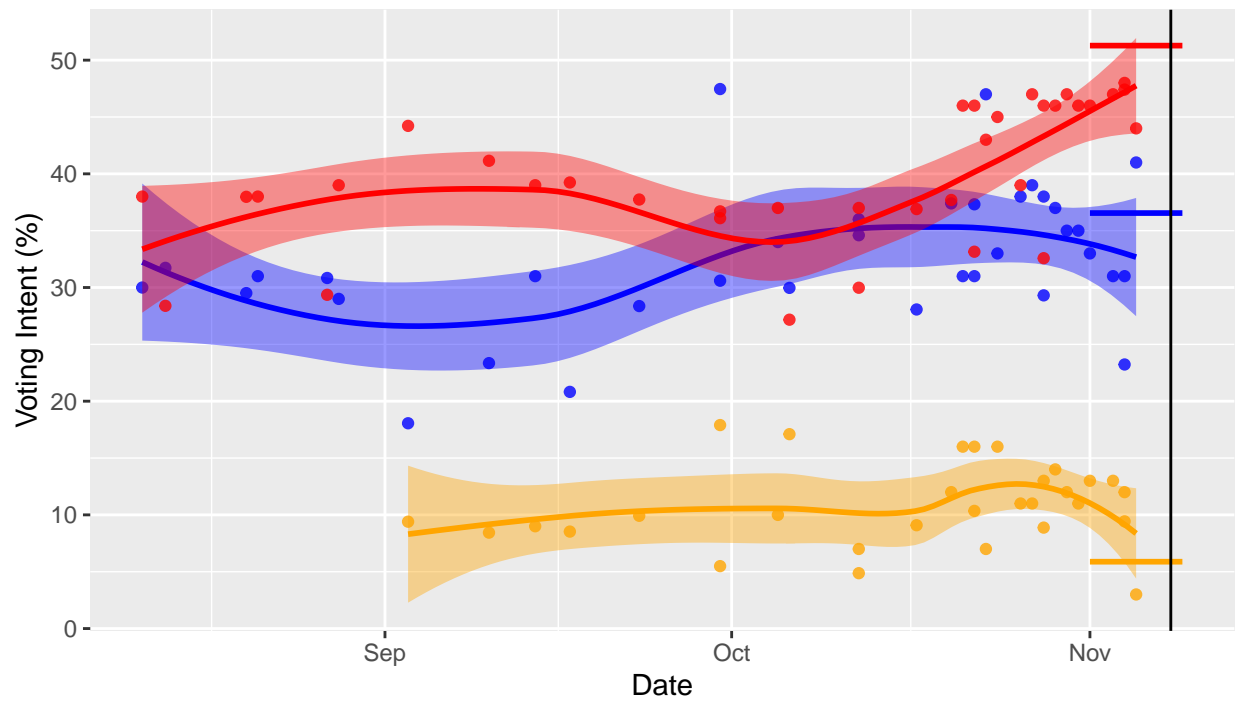
Alabama – Raw Polls



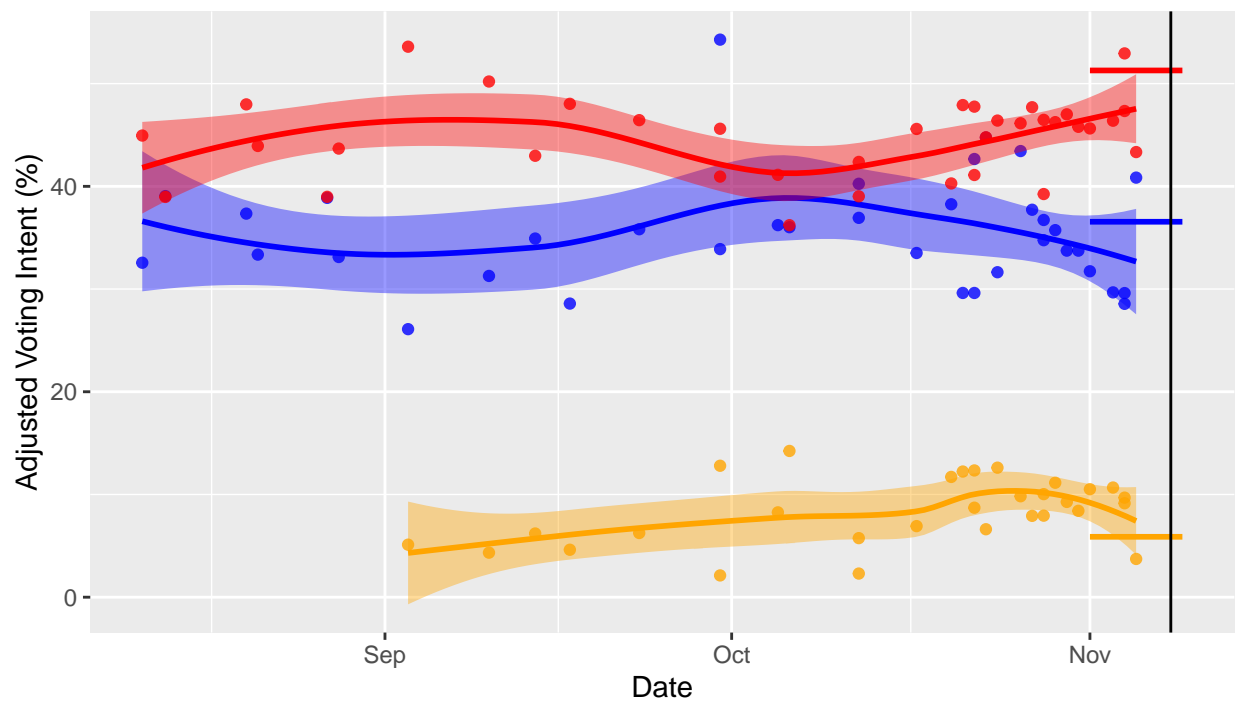
Alabama – Adjusted Polls



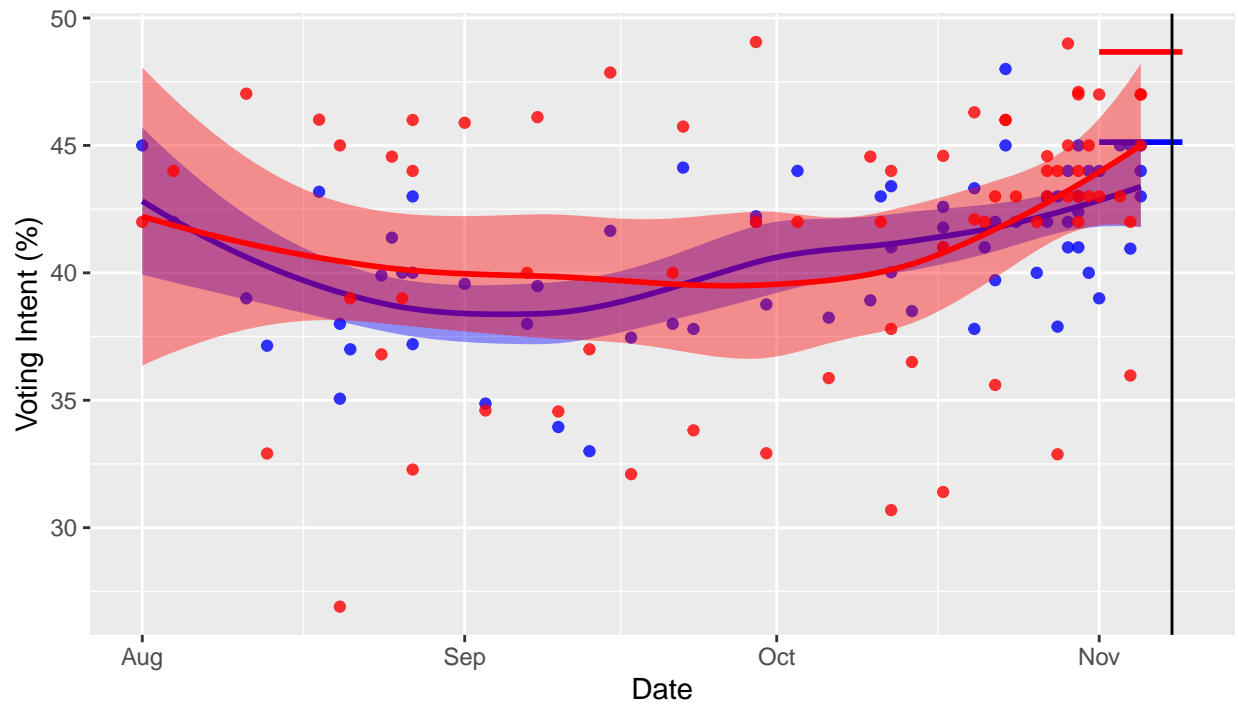
Alaska – Raw Polls



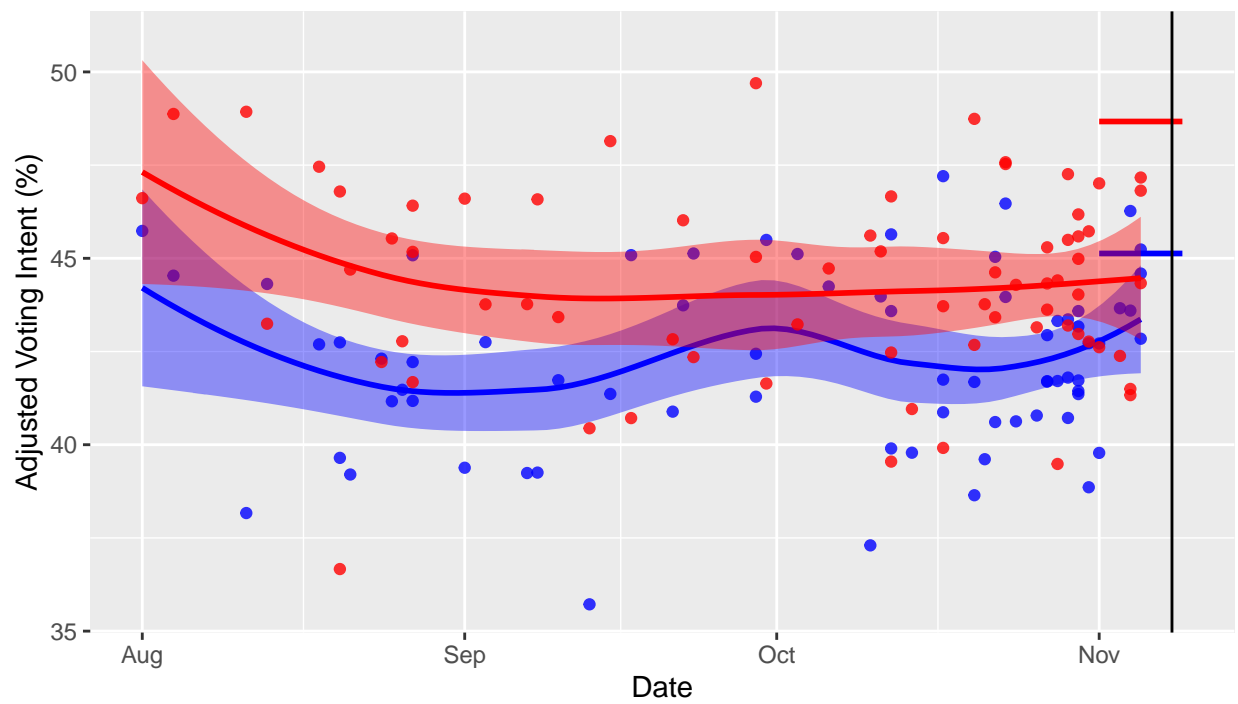
Alaska – Adjusted Polls



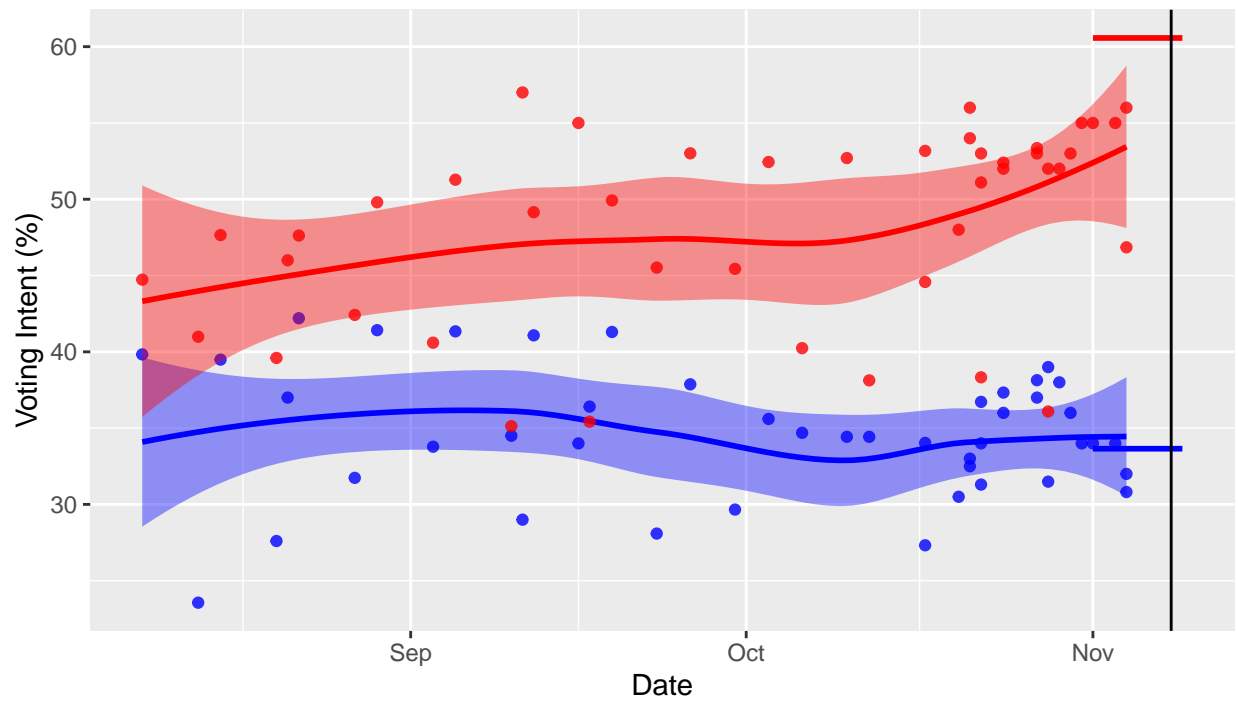
Arizona – Raw Polls



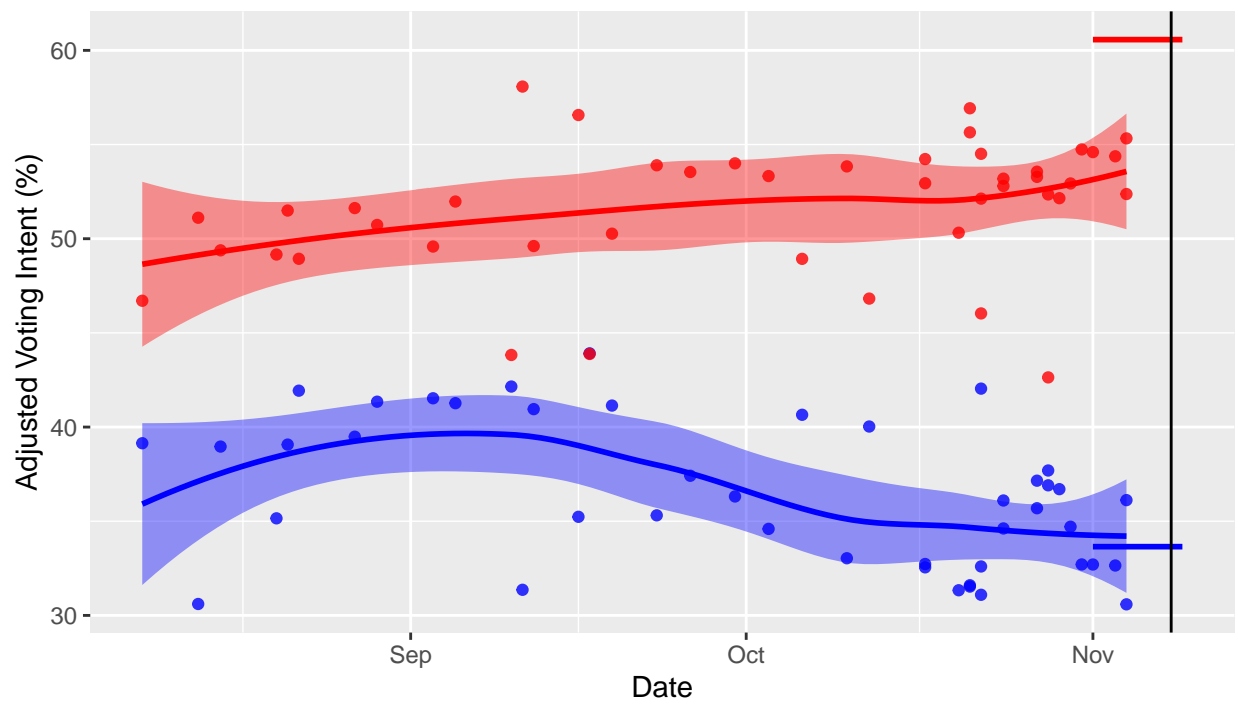
Arizona – Adjusted Polls



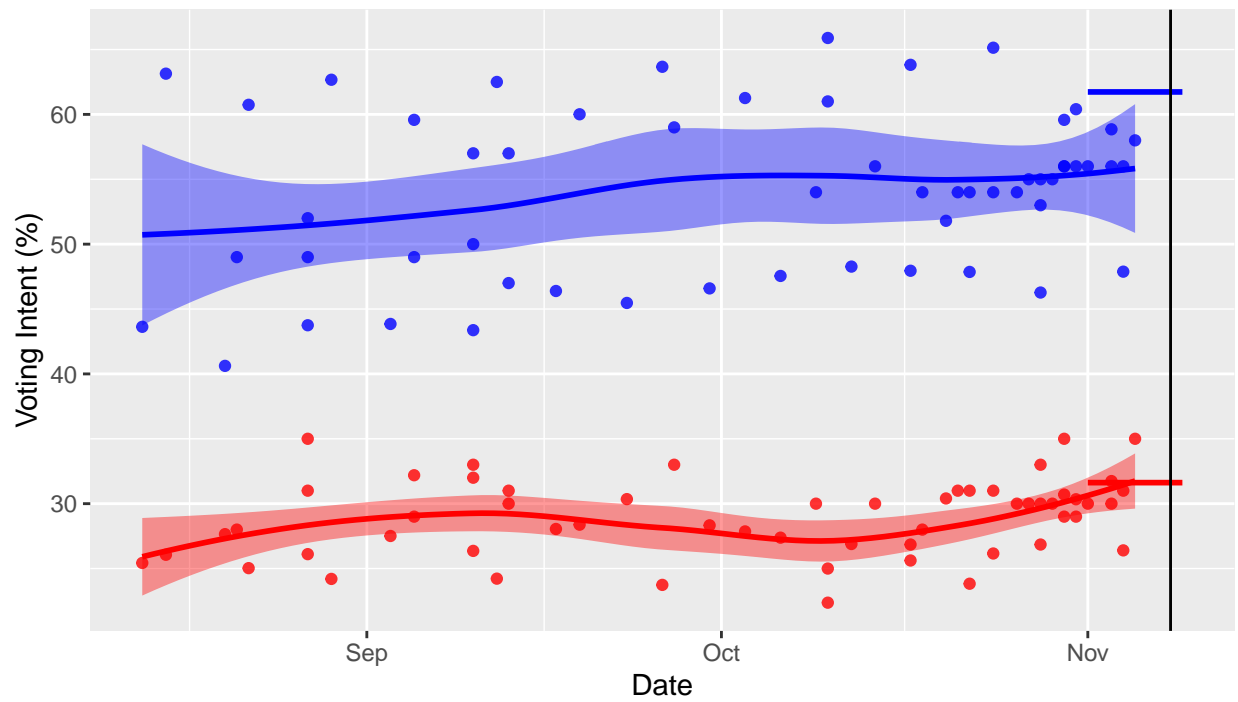
Arkansas – Raw Polls



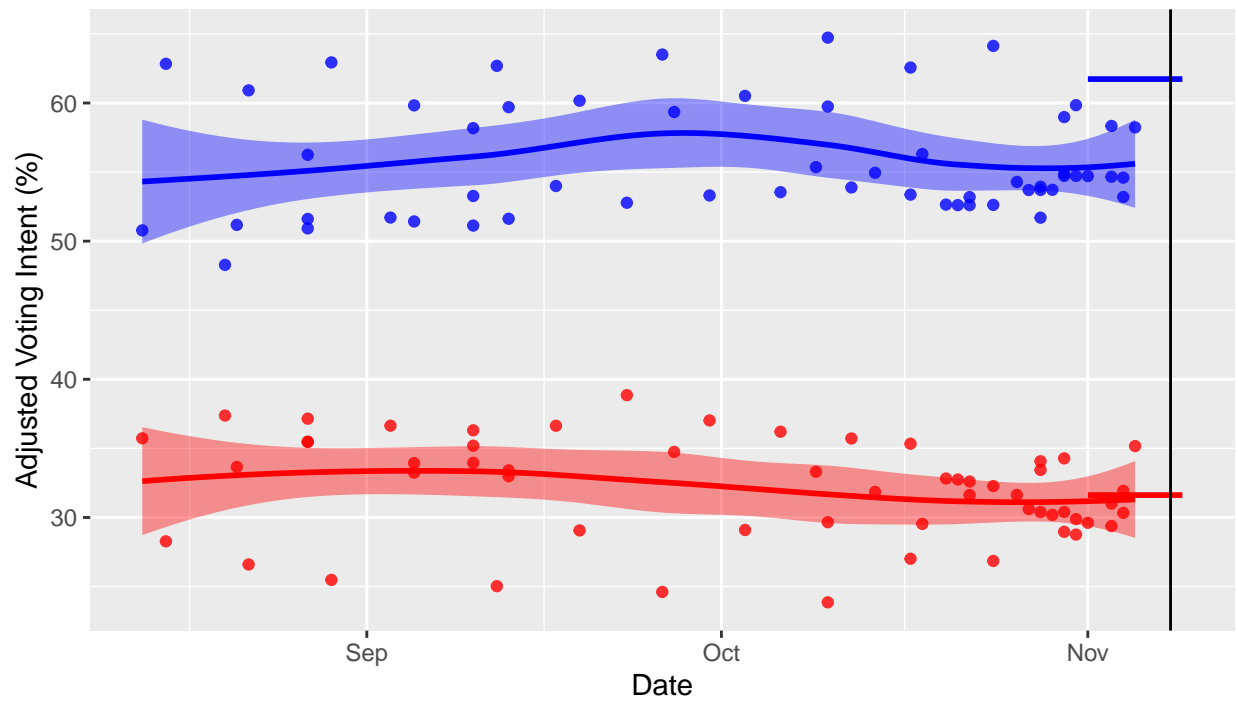
Arkansas – Adjusted Polls



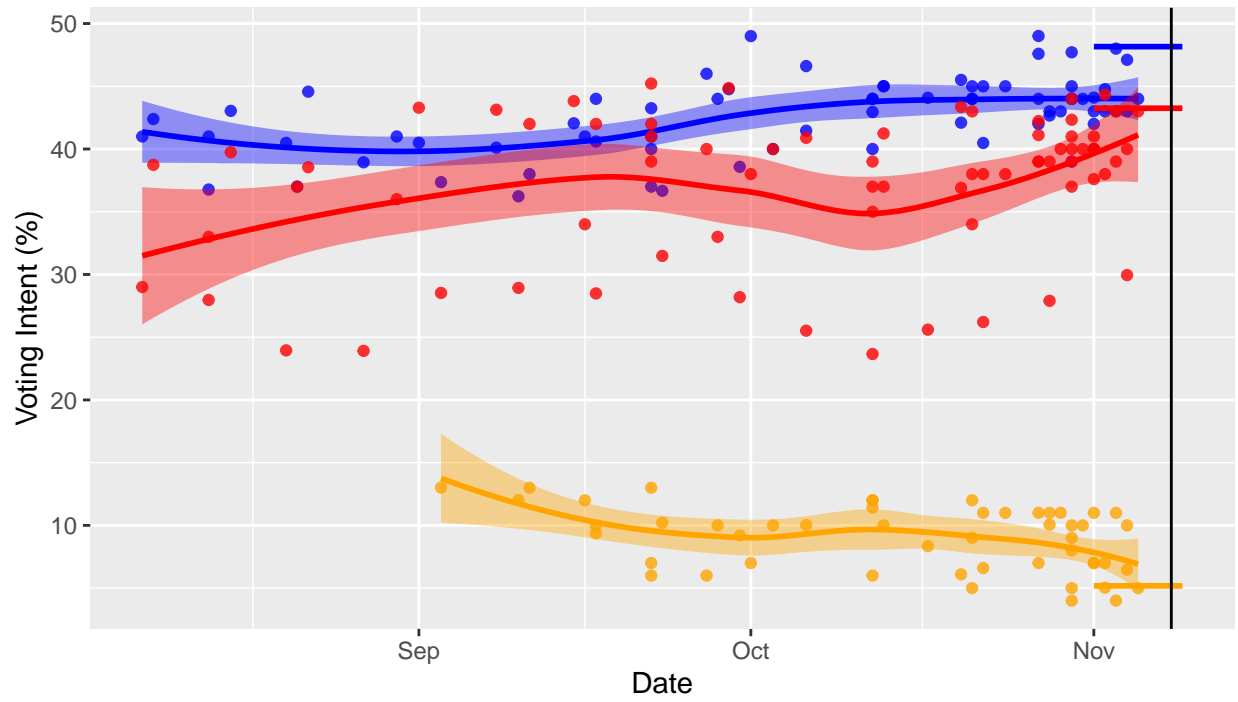
California – Raw Polls



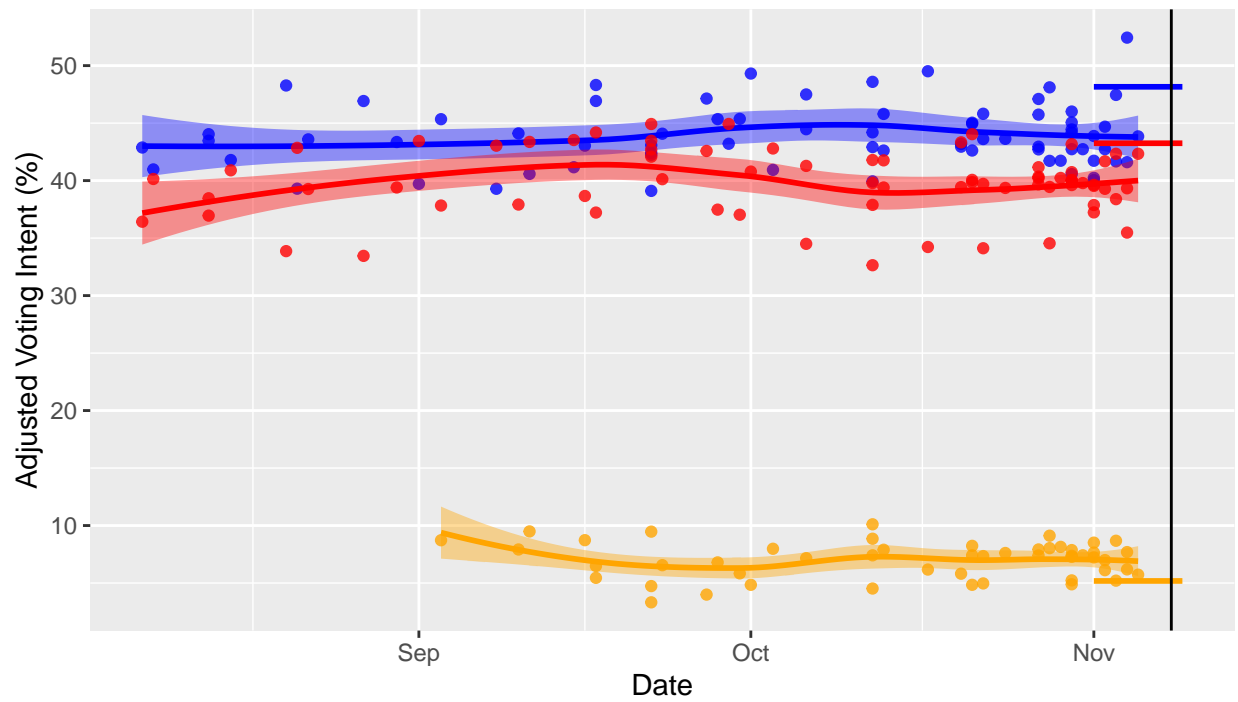
California – Adjusted Polls



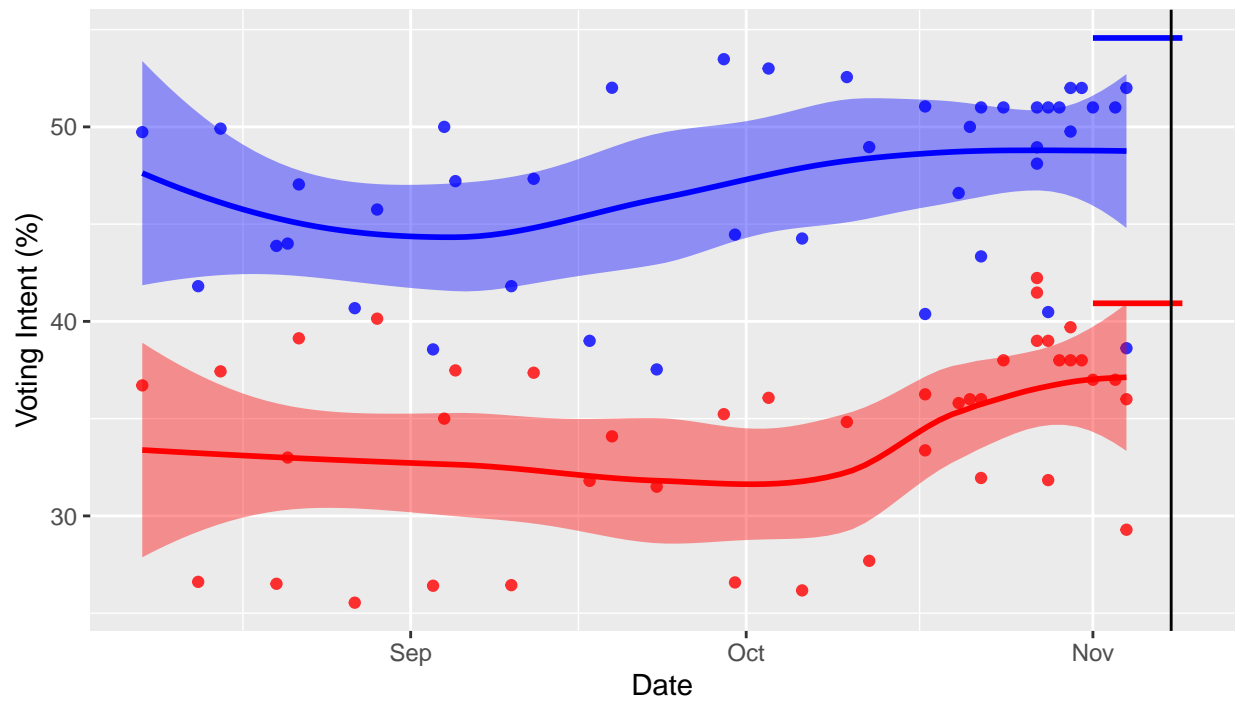
Colorado – Raw Polls



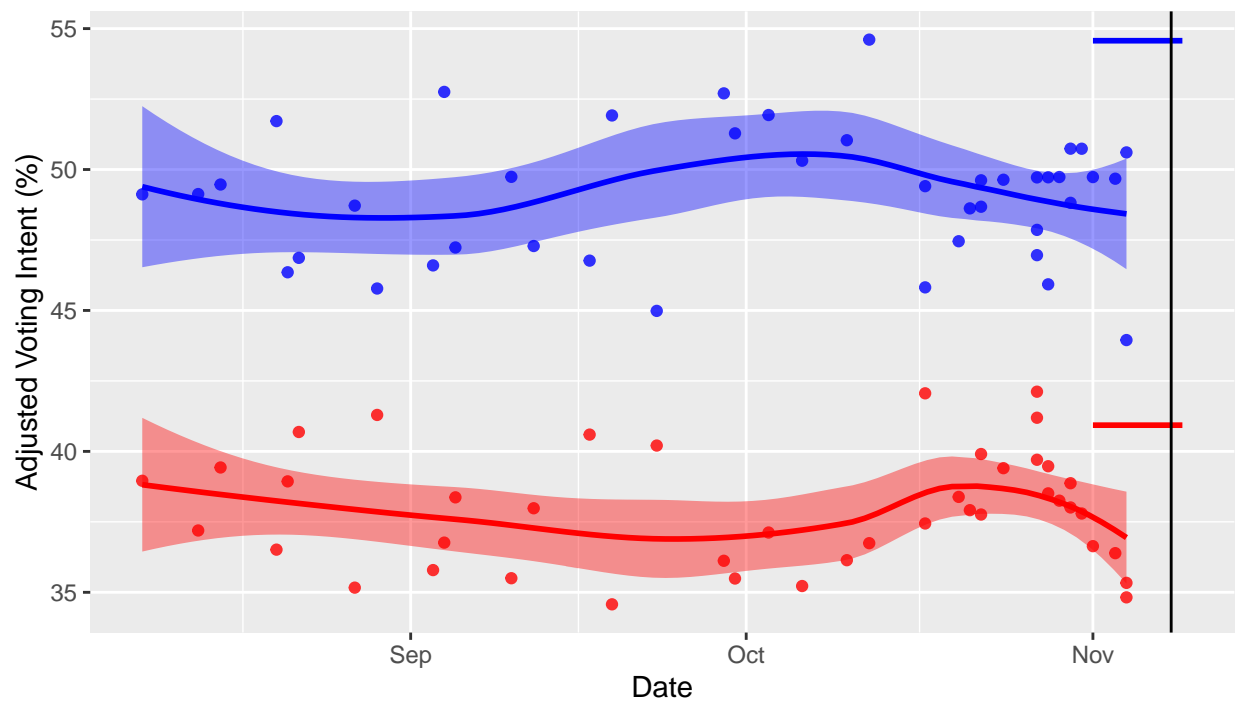
Colorado – Adjusted Polls



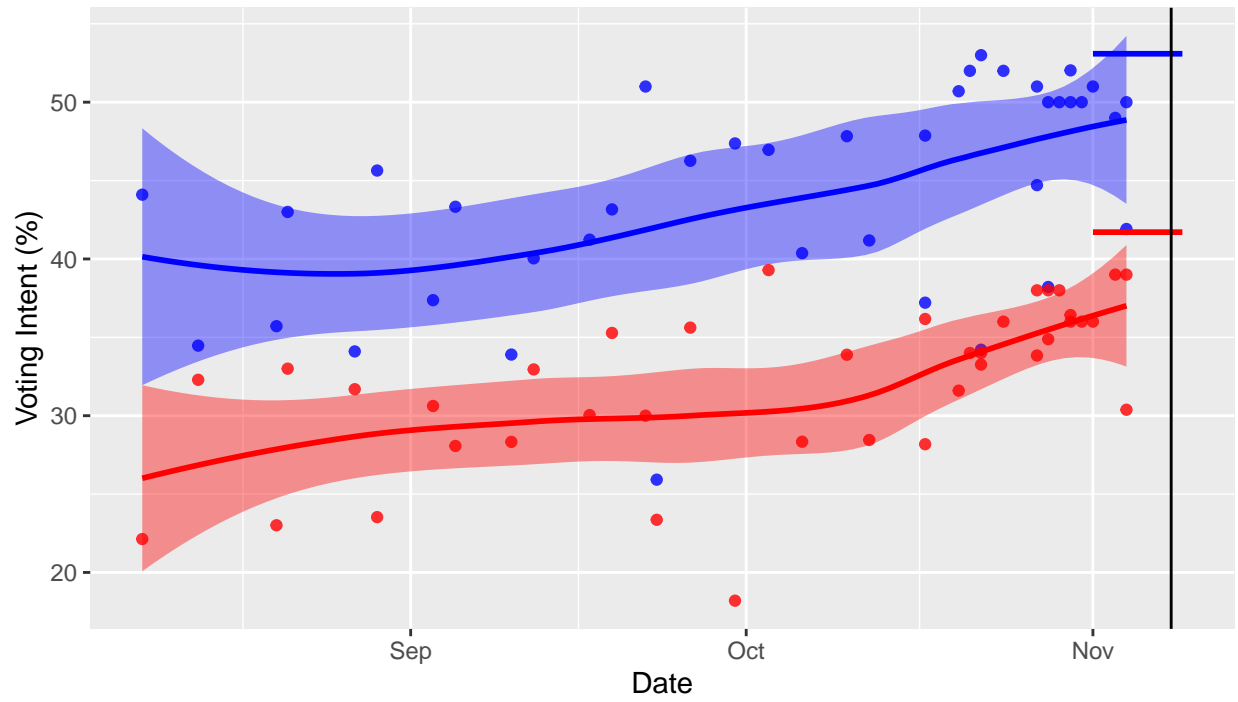
Connecticut – Raw Polls



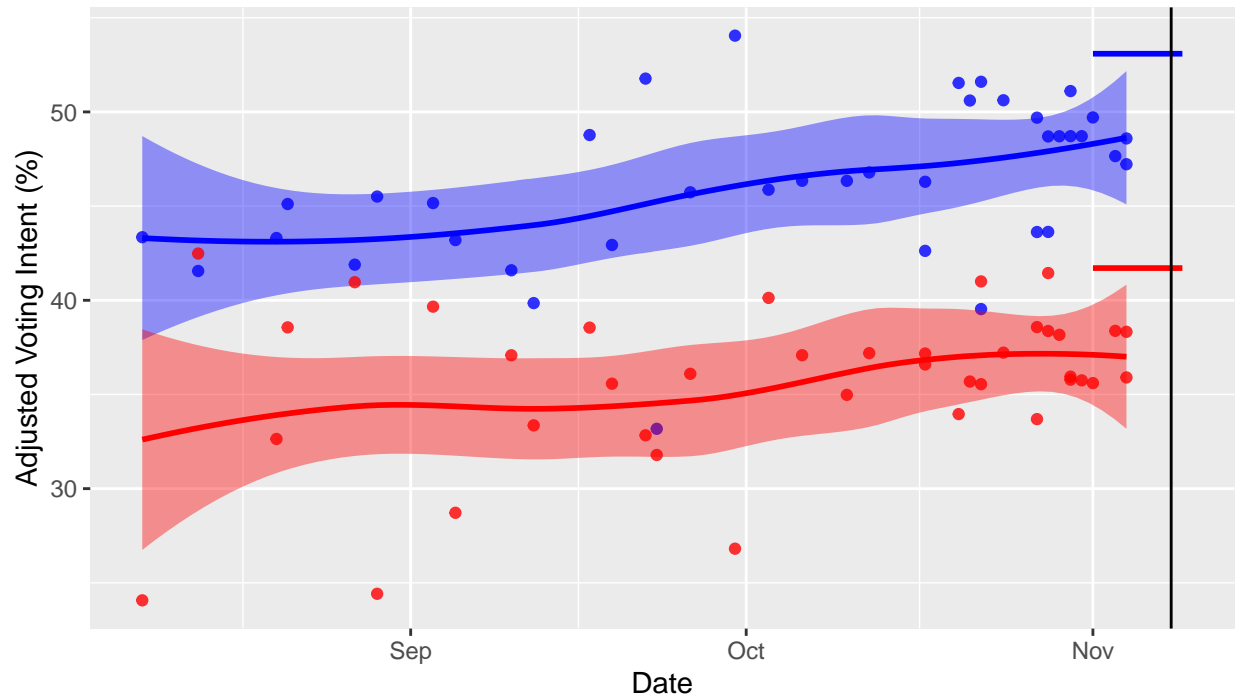
Connecticut – Adjusted Polls



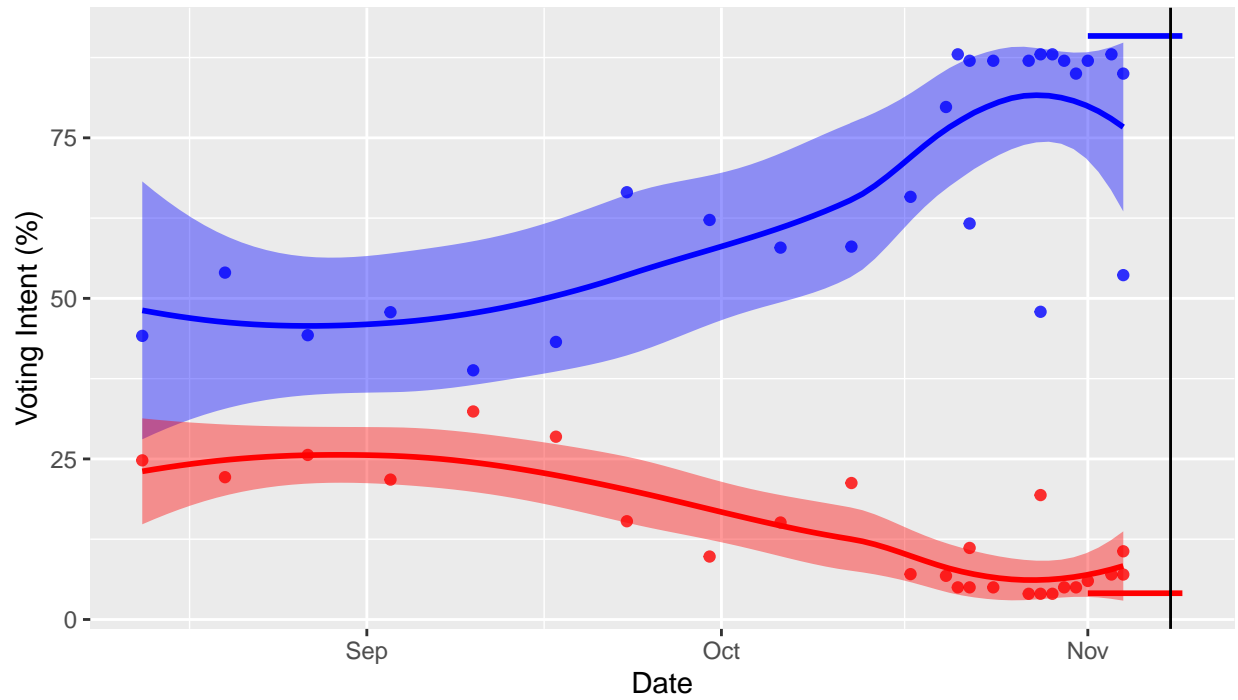
Delaware – Raw Polls



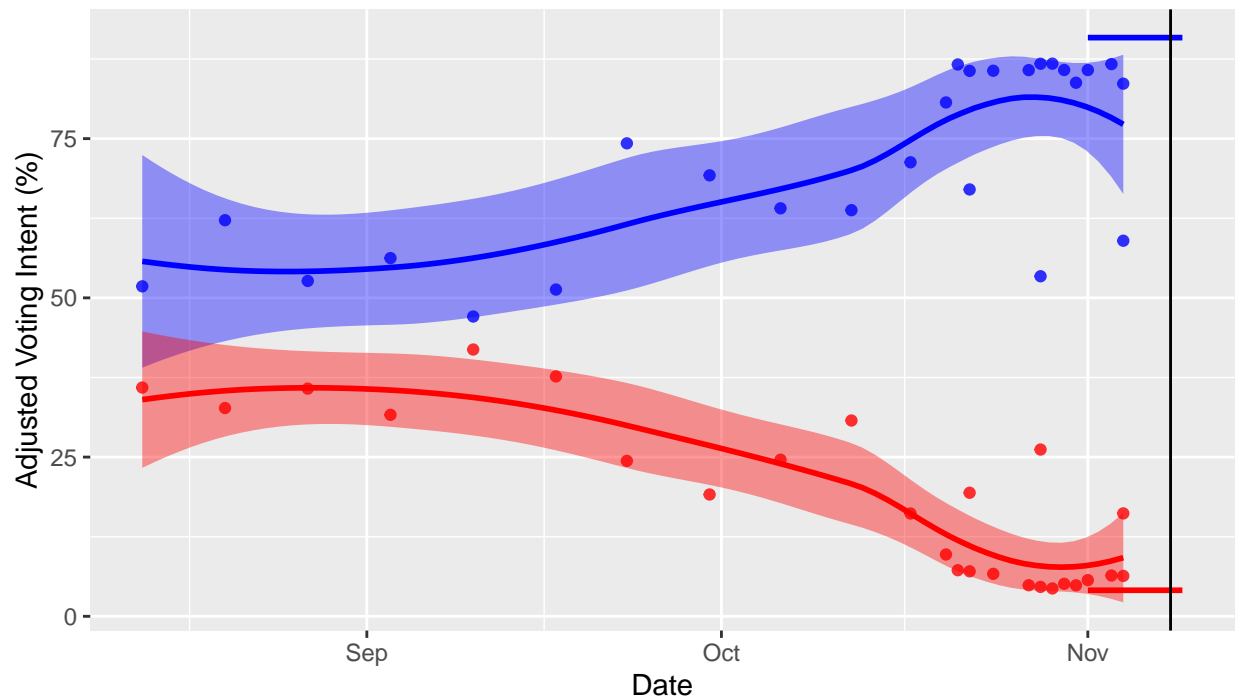
Delaware – Adjusted Polls



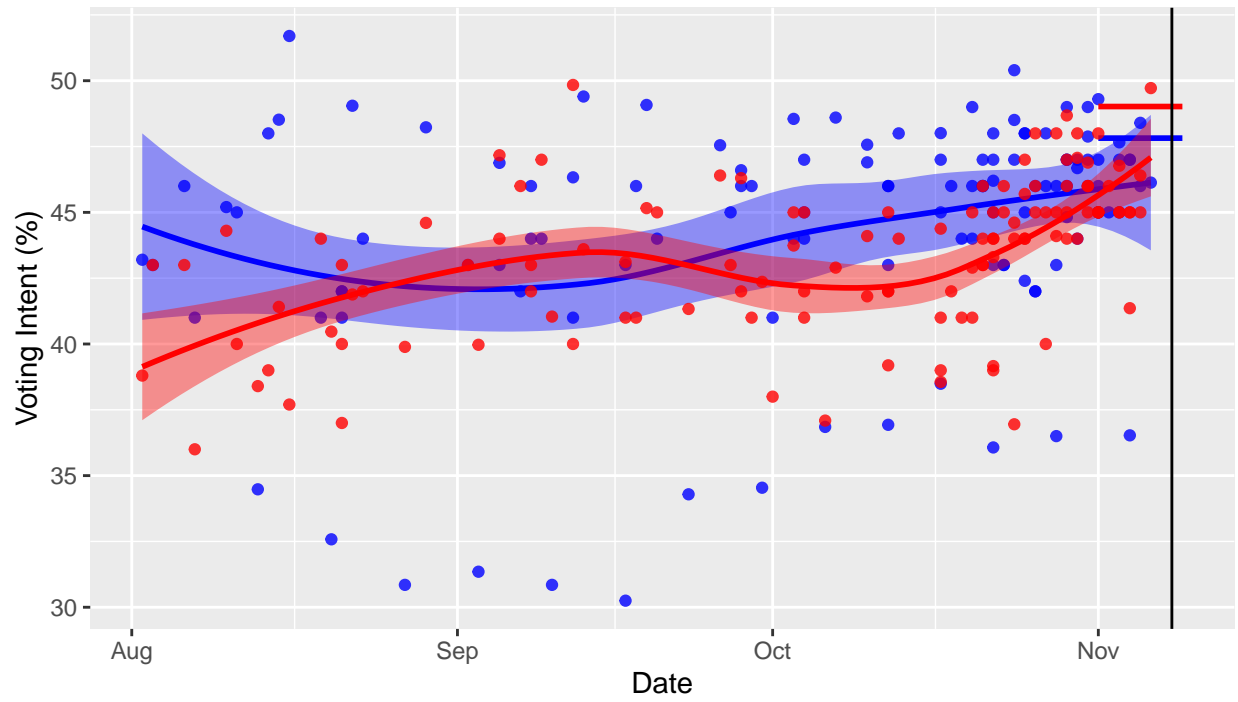
District of Columbia – Raw Polls



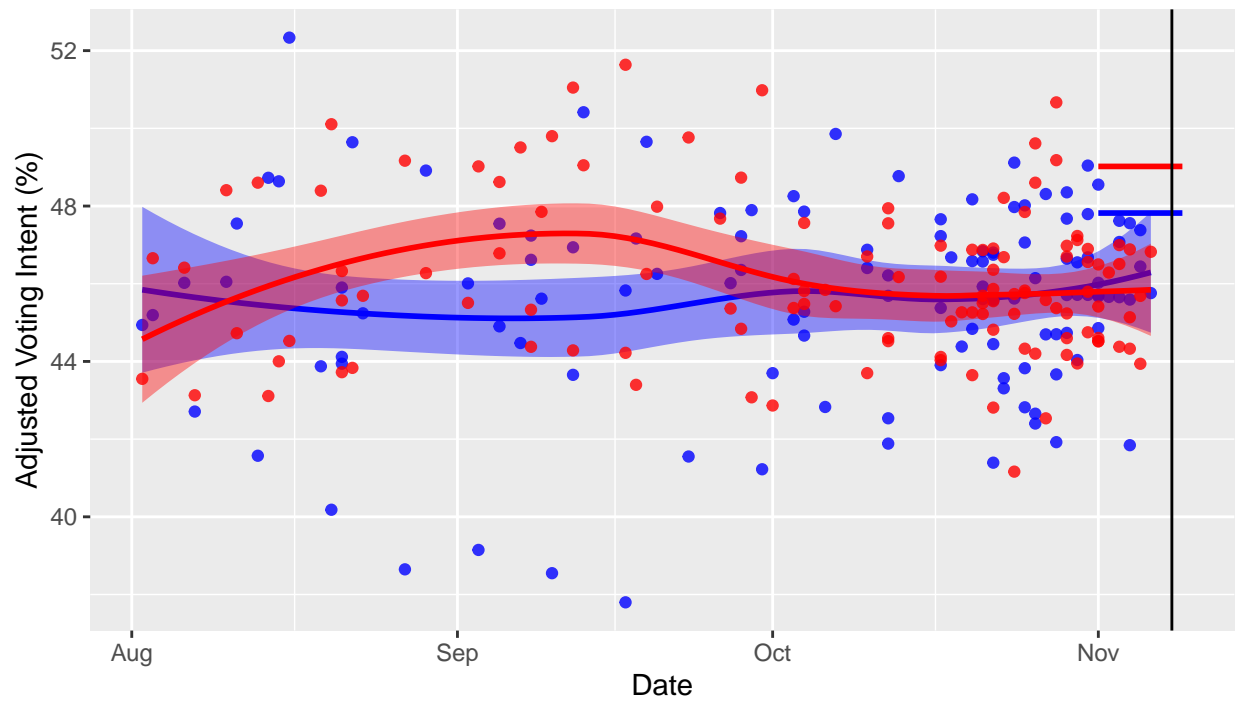
District of Columbia – Adjusted Polls



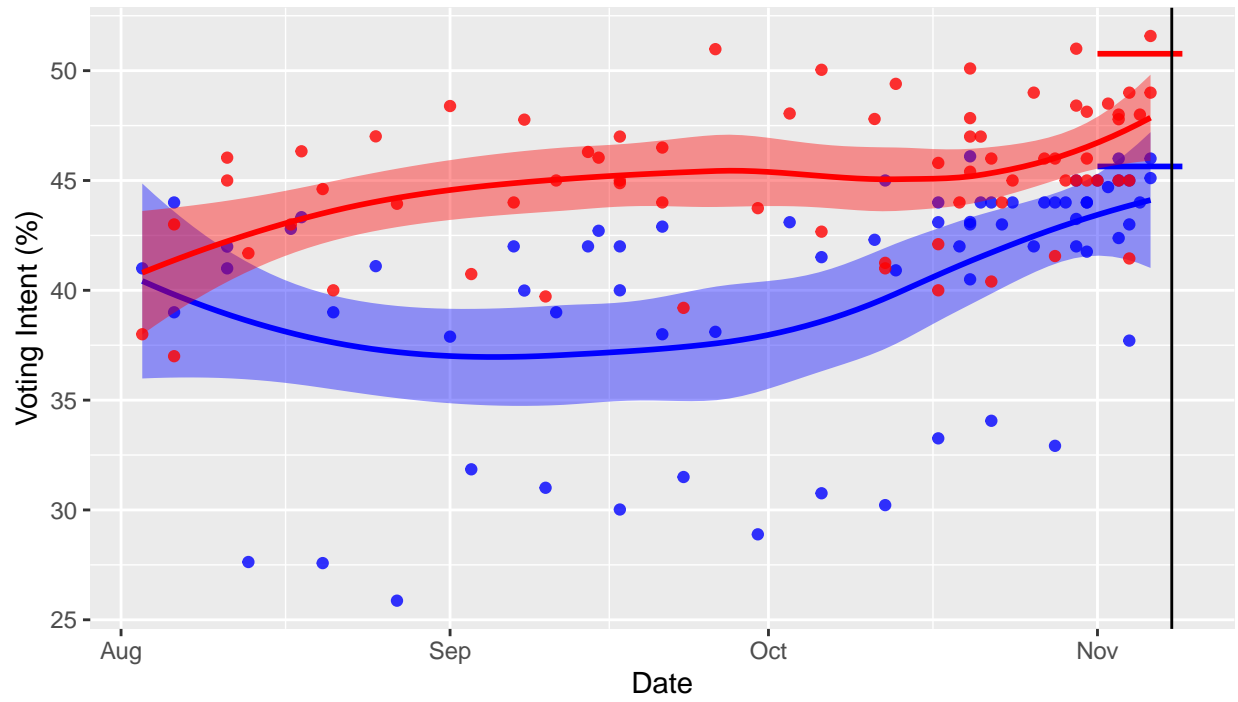
Florida – Raw Polls



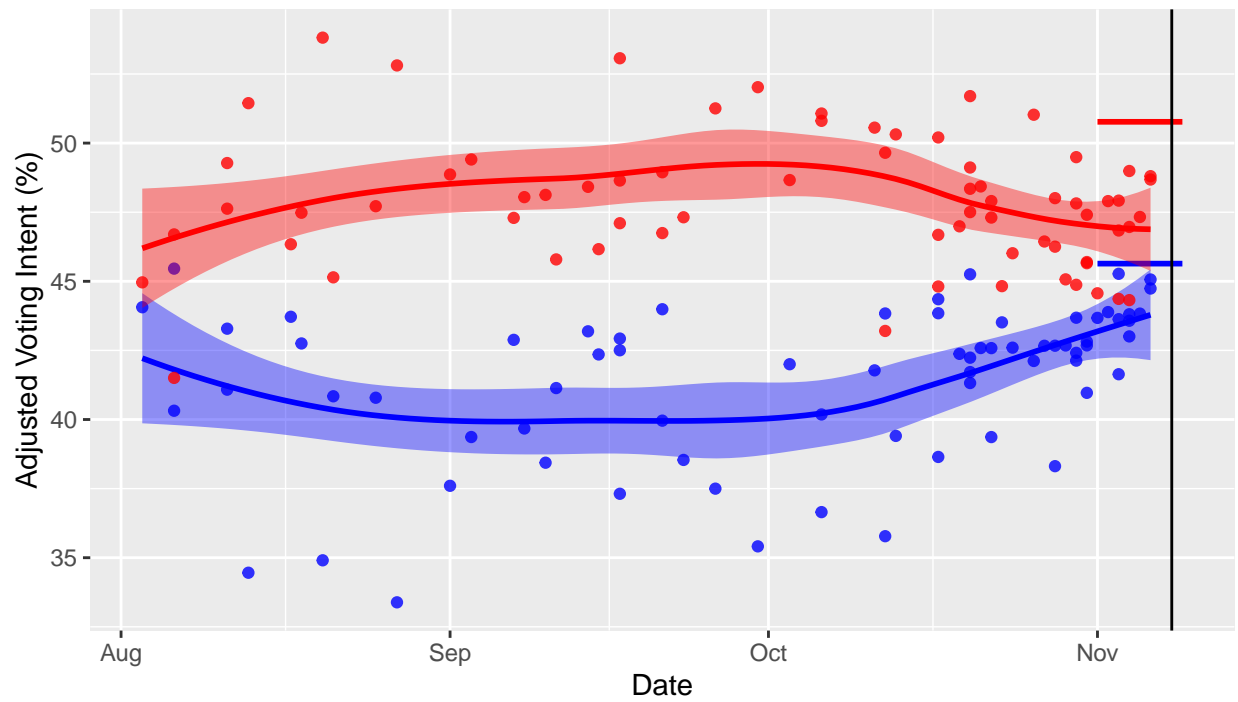
Florida – Adjusted Polls



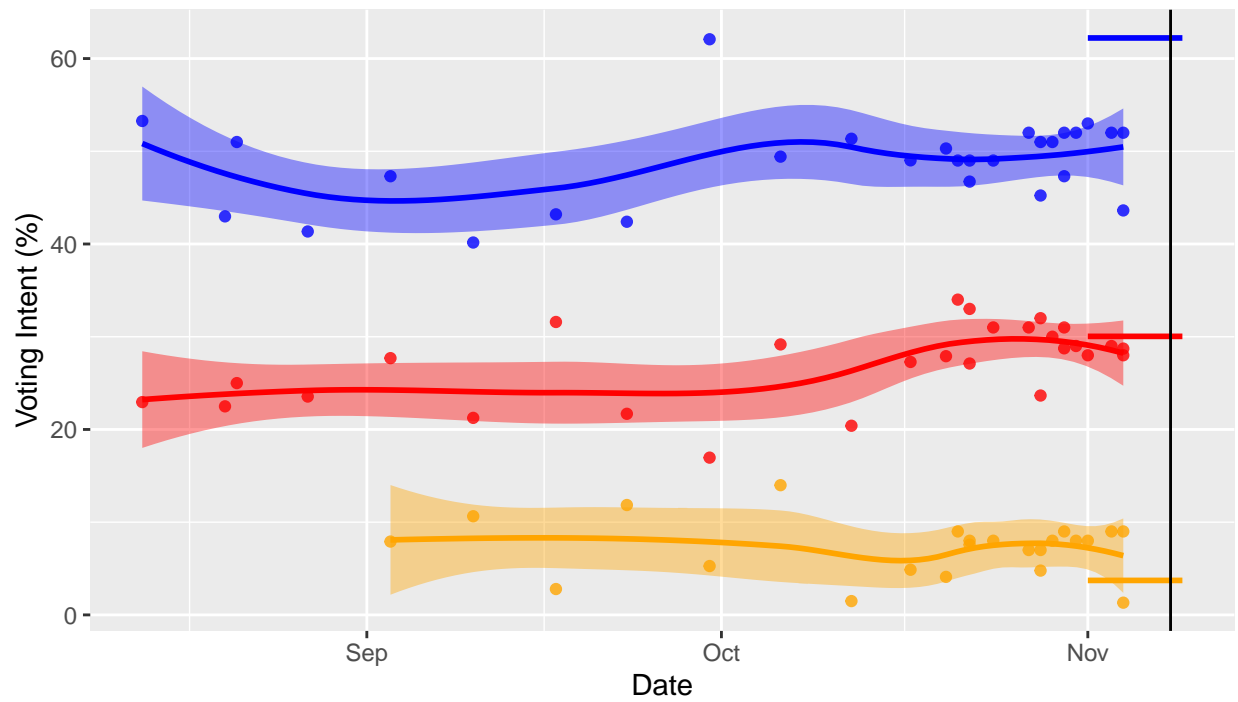
Georgia – Raw Polls



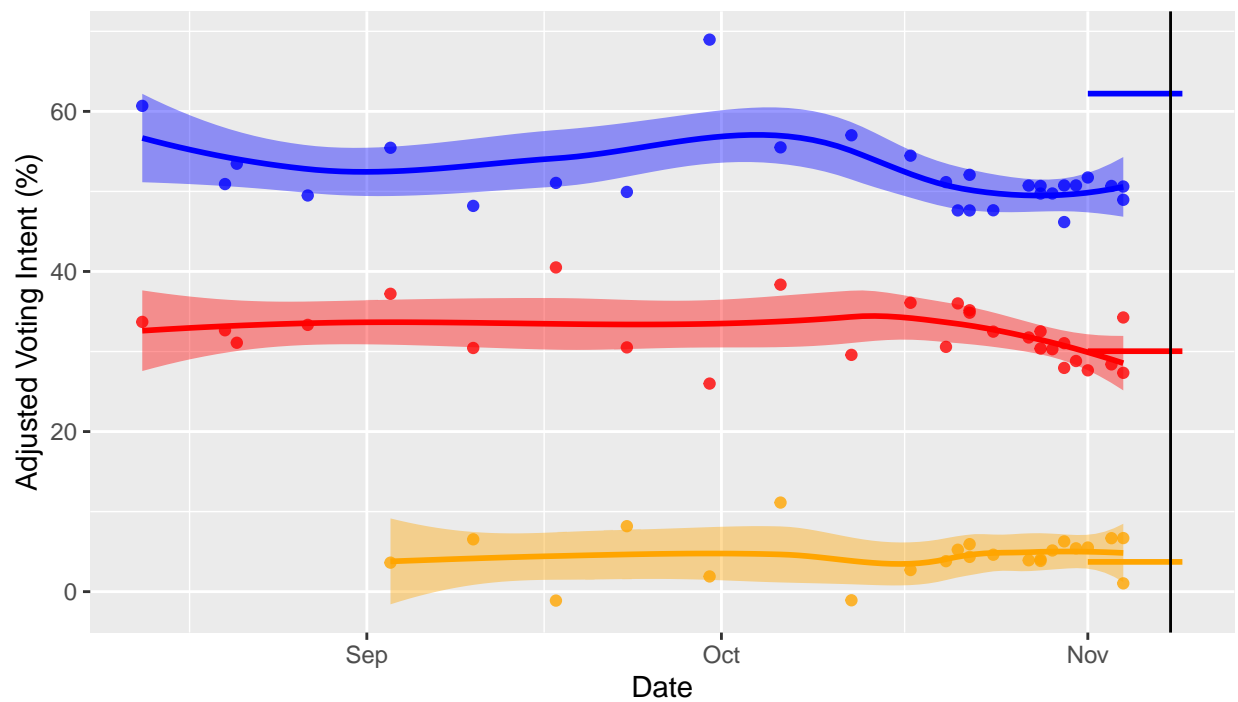
Georgia – Adjusted Polls



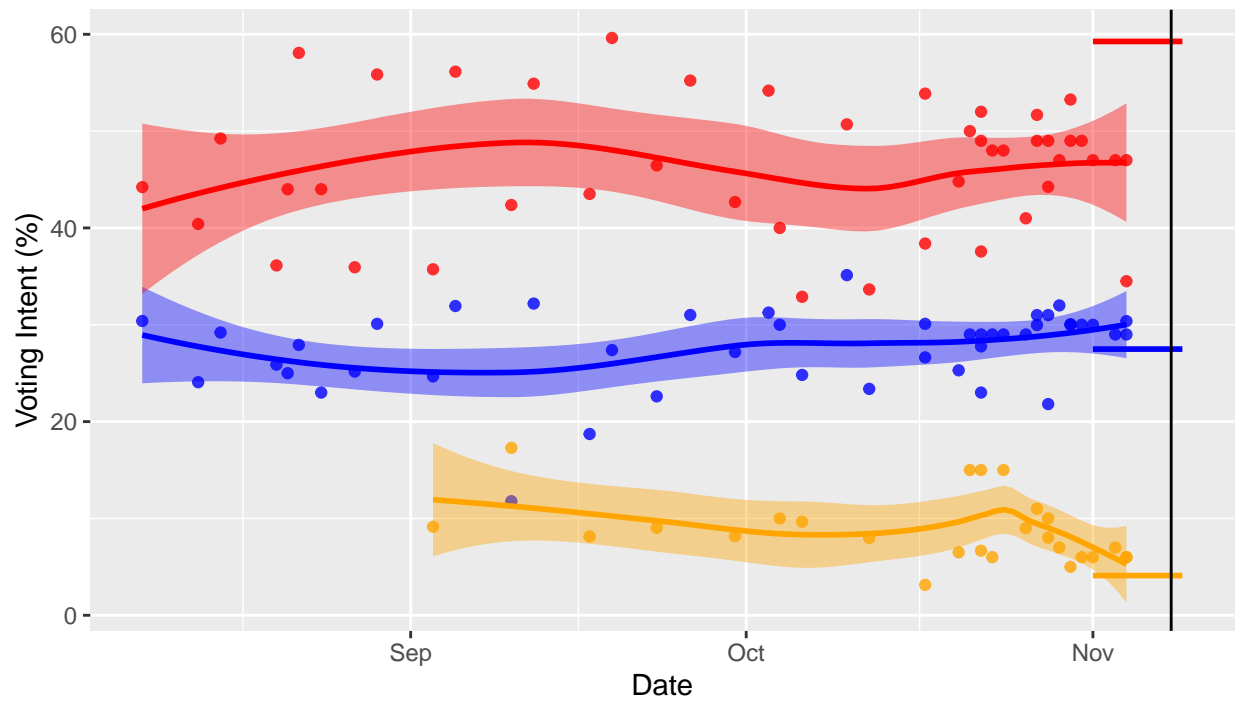
Hawaii – Raw Polls



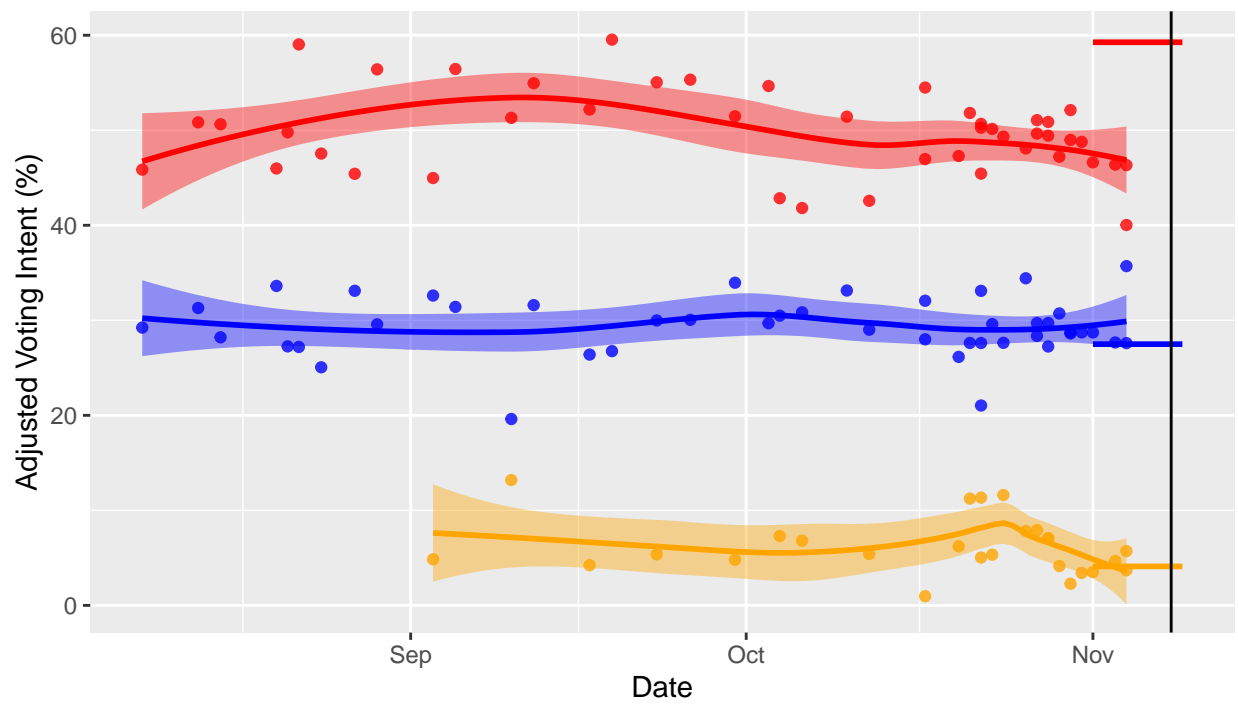
Hawaii – Adjusted Polls



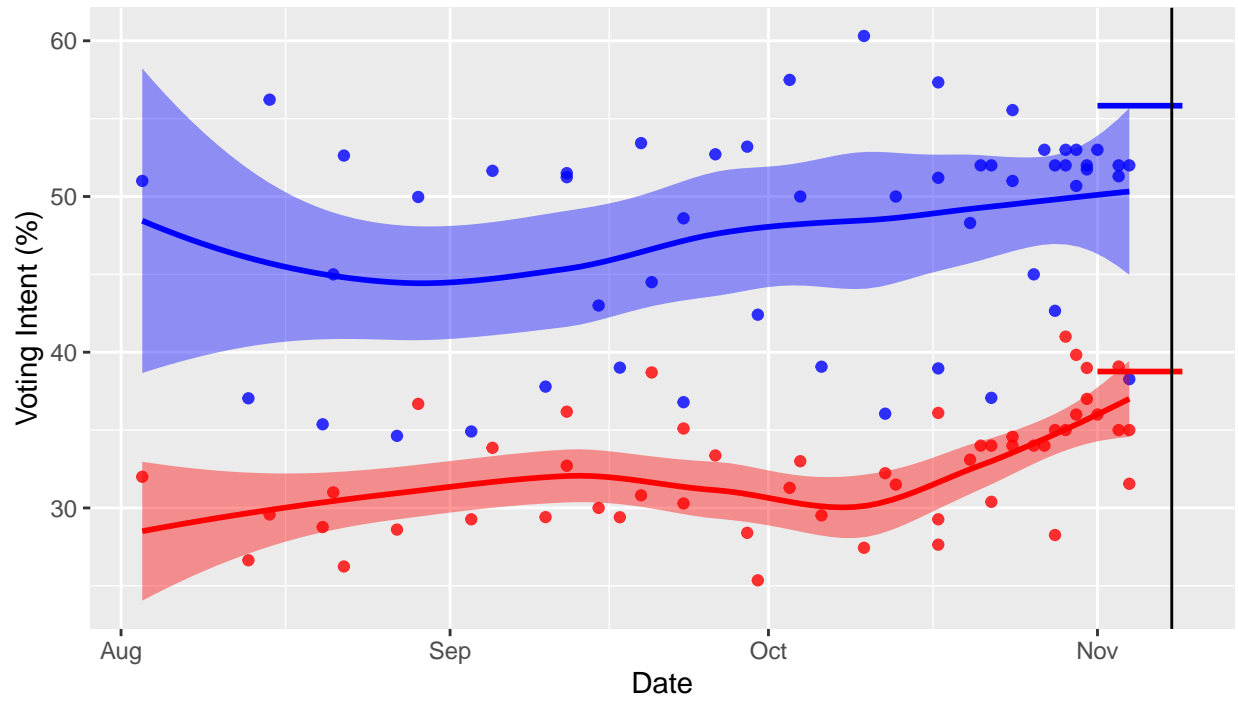
Idaho – Raw Polls



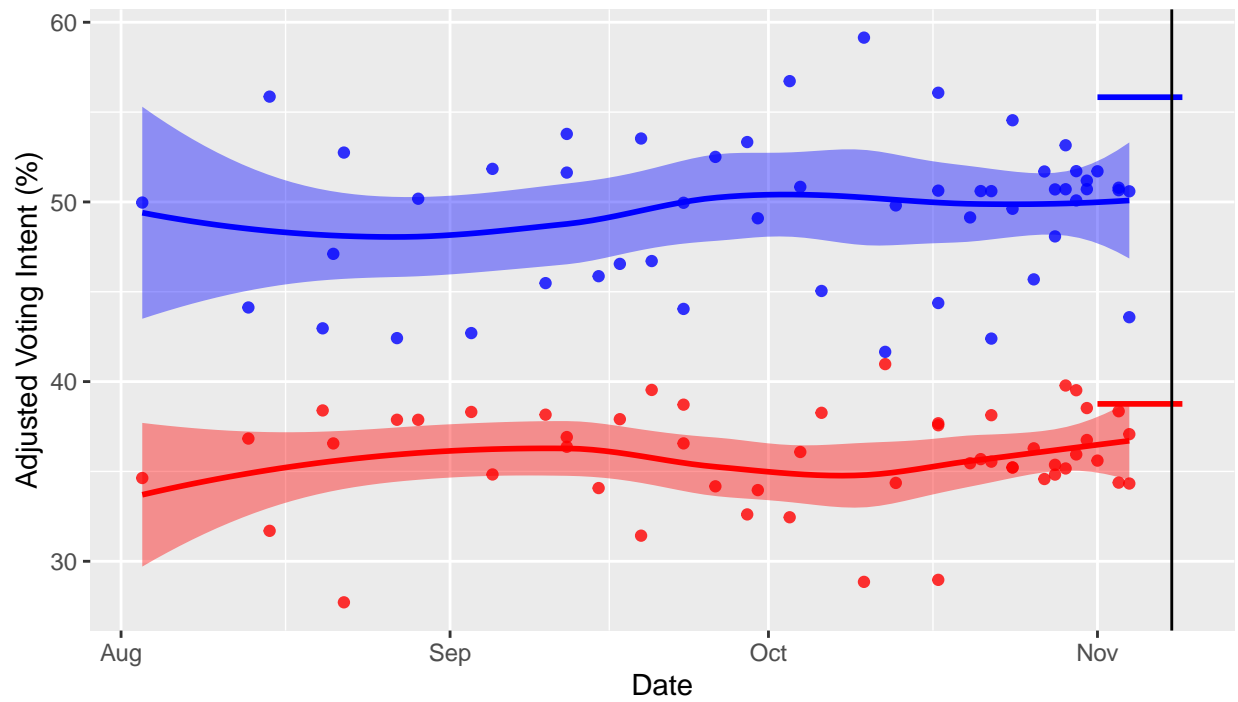
Idaho – Adjusted Polls



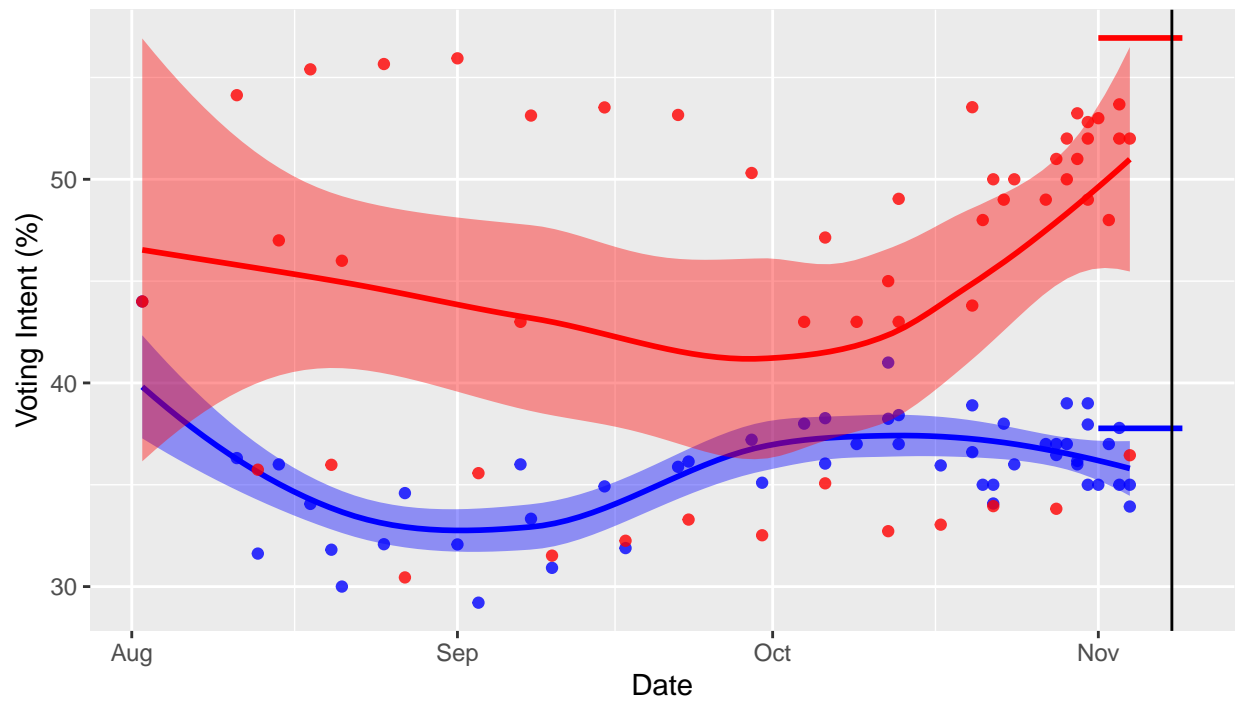
Illinois – Raw Polls



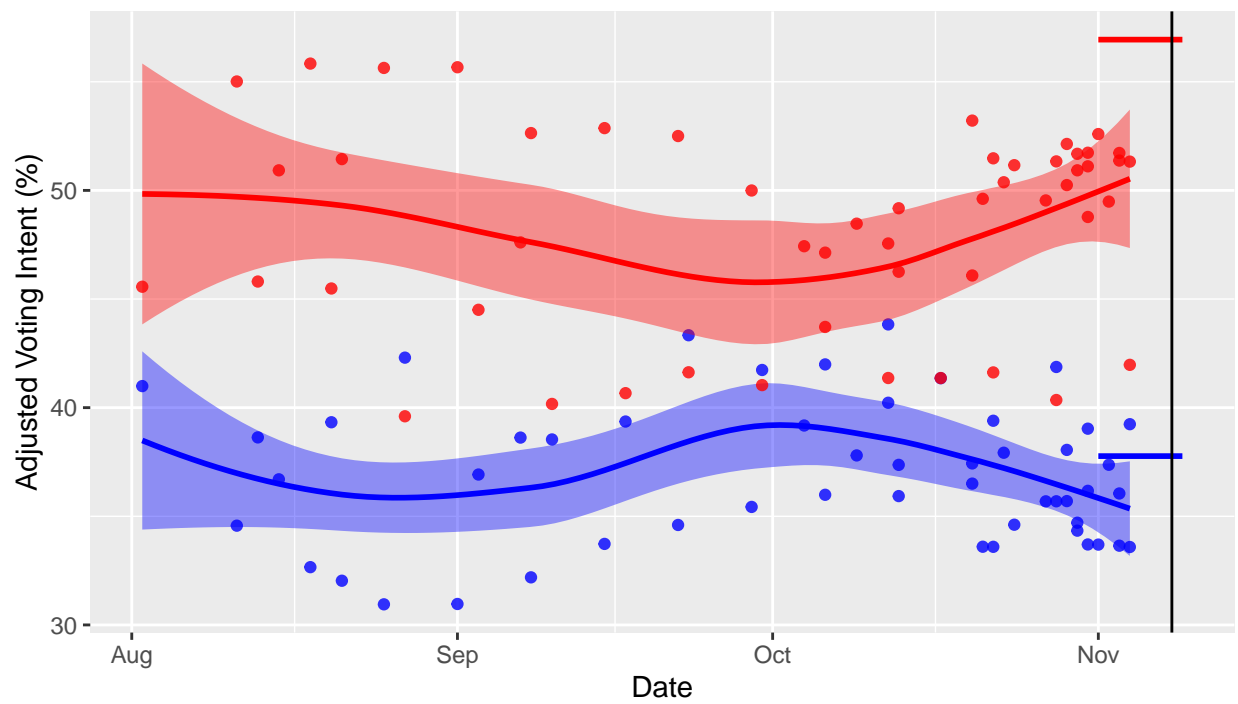
Illinois – Adjusted Polls



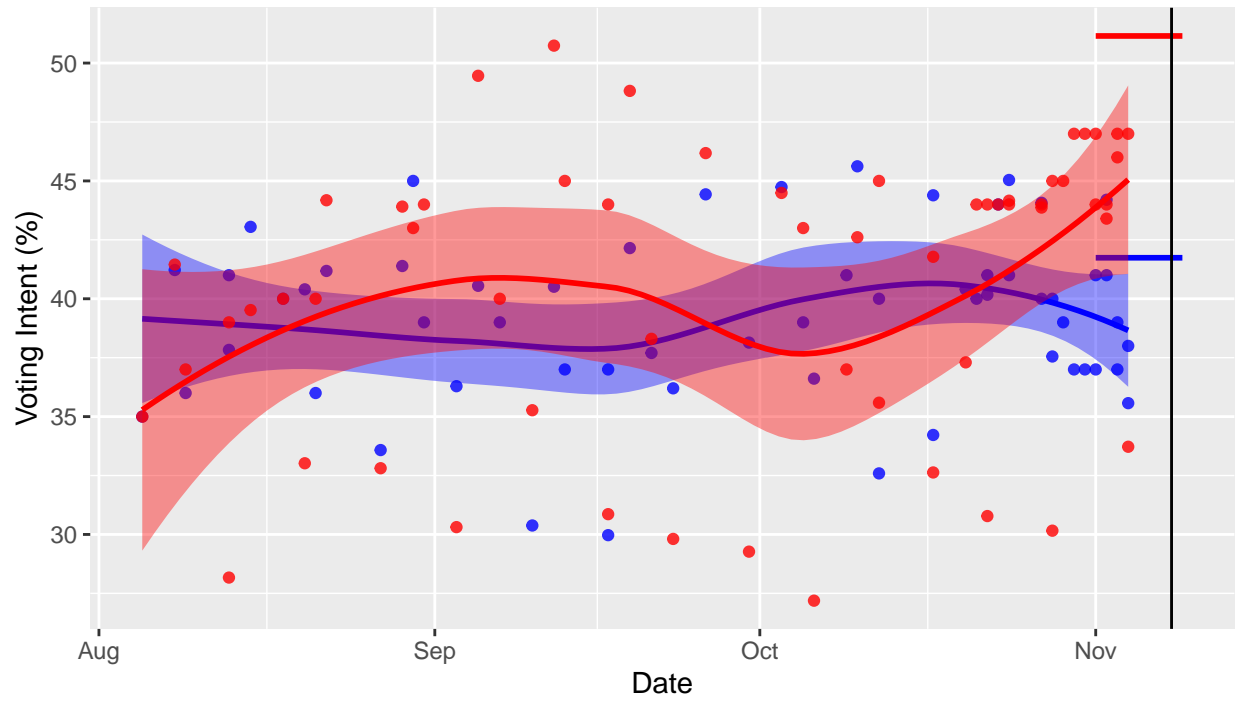
Indiana – Raw Polls



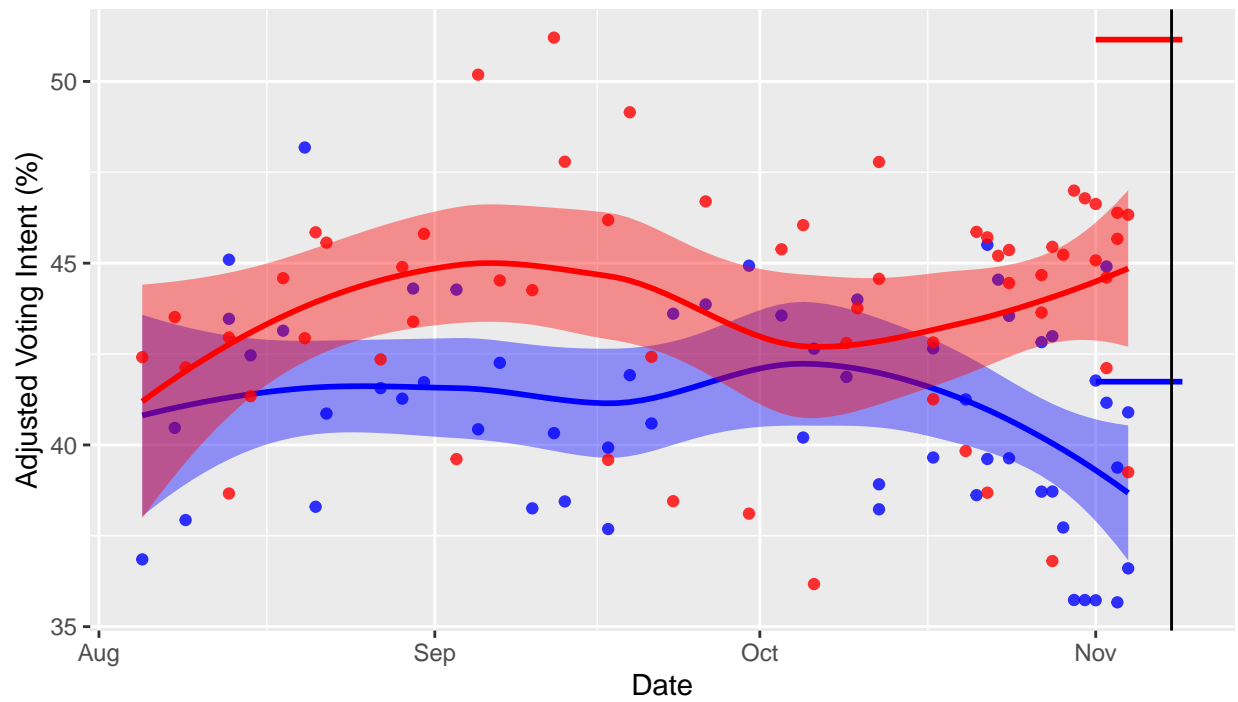
Indiana – Adjusted Polls



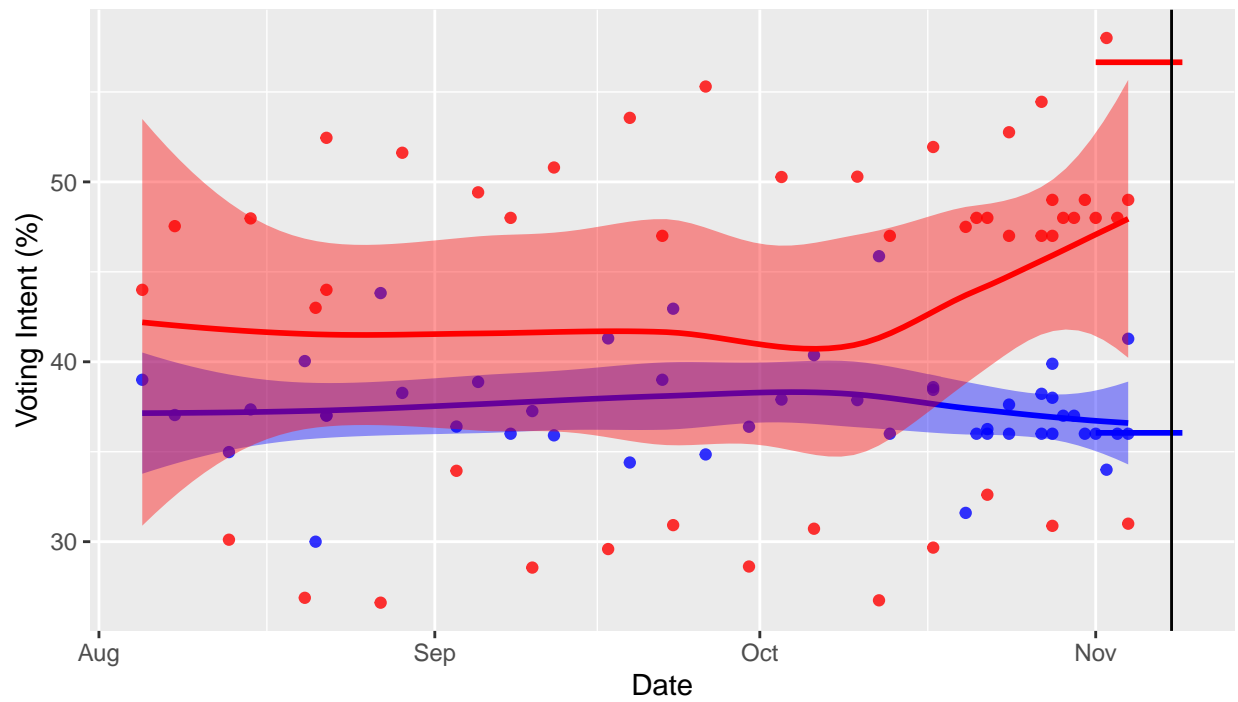
Iowa – Raw Polls



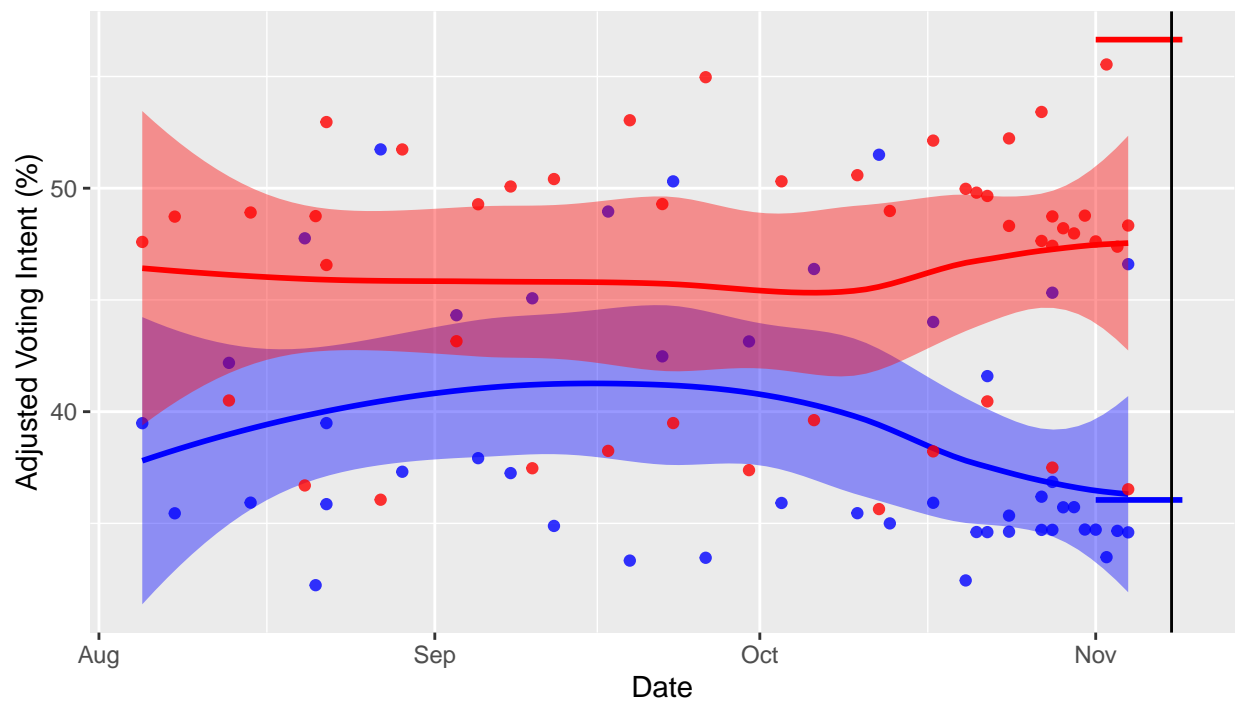
Iowa – Adjusted Polls



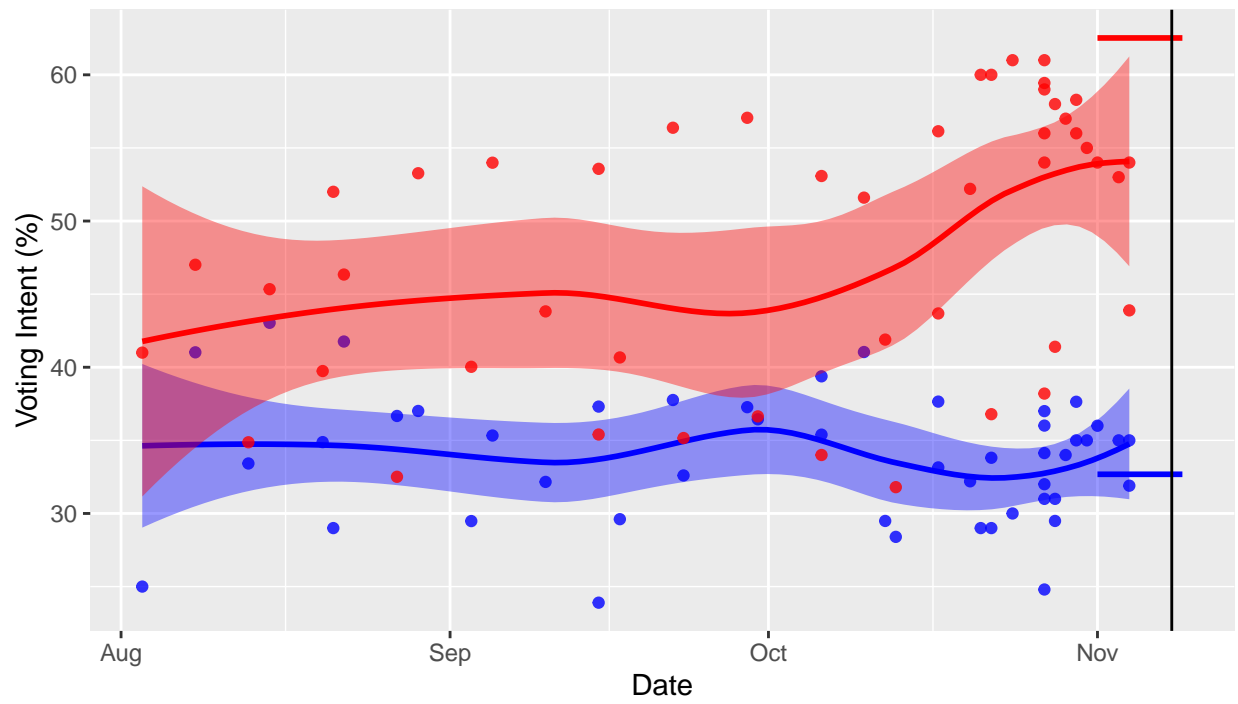
Kansas – Raw Polls



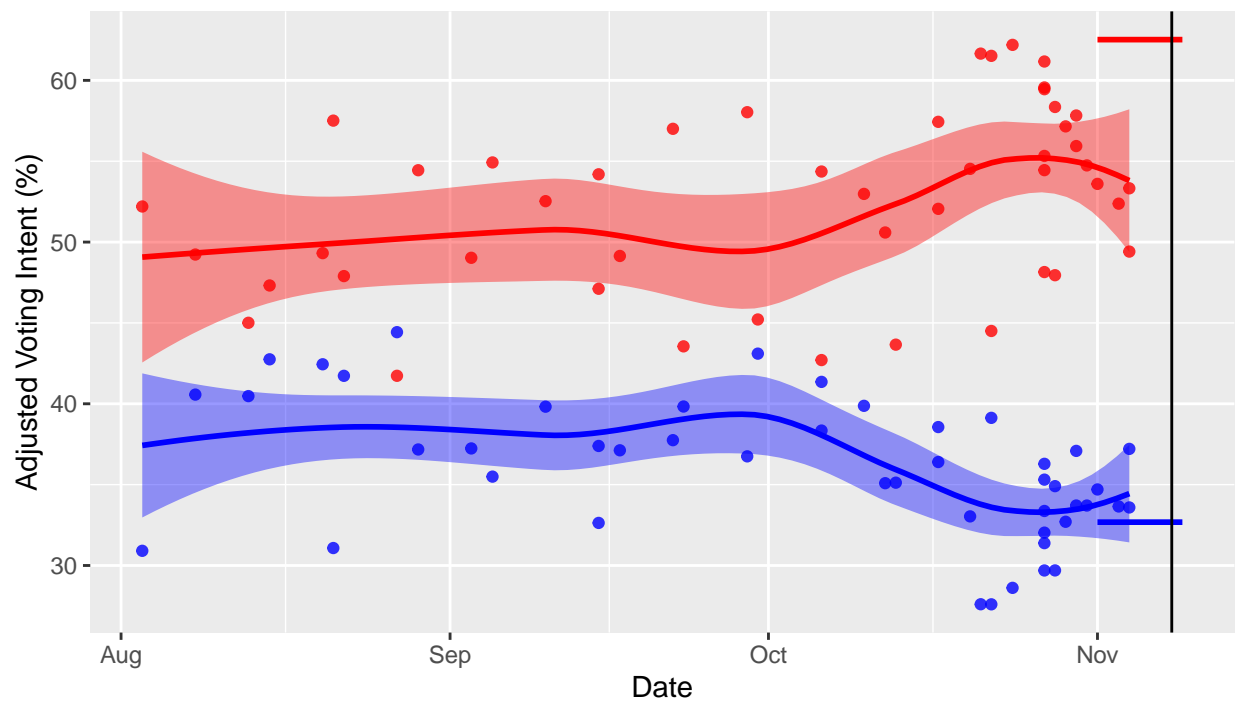
Kansas – Adjusted Polls



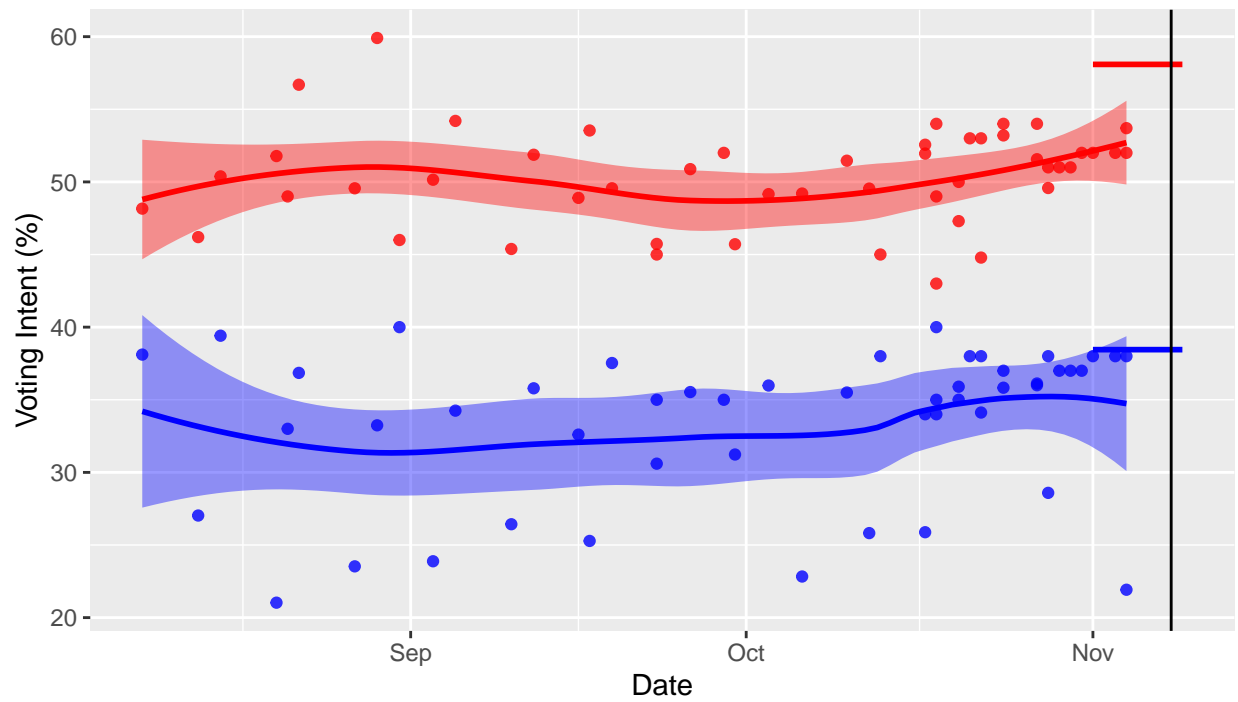
Kentucky – Raw Polls



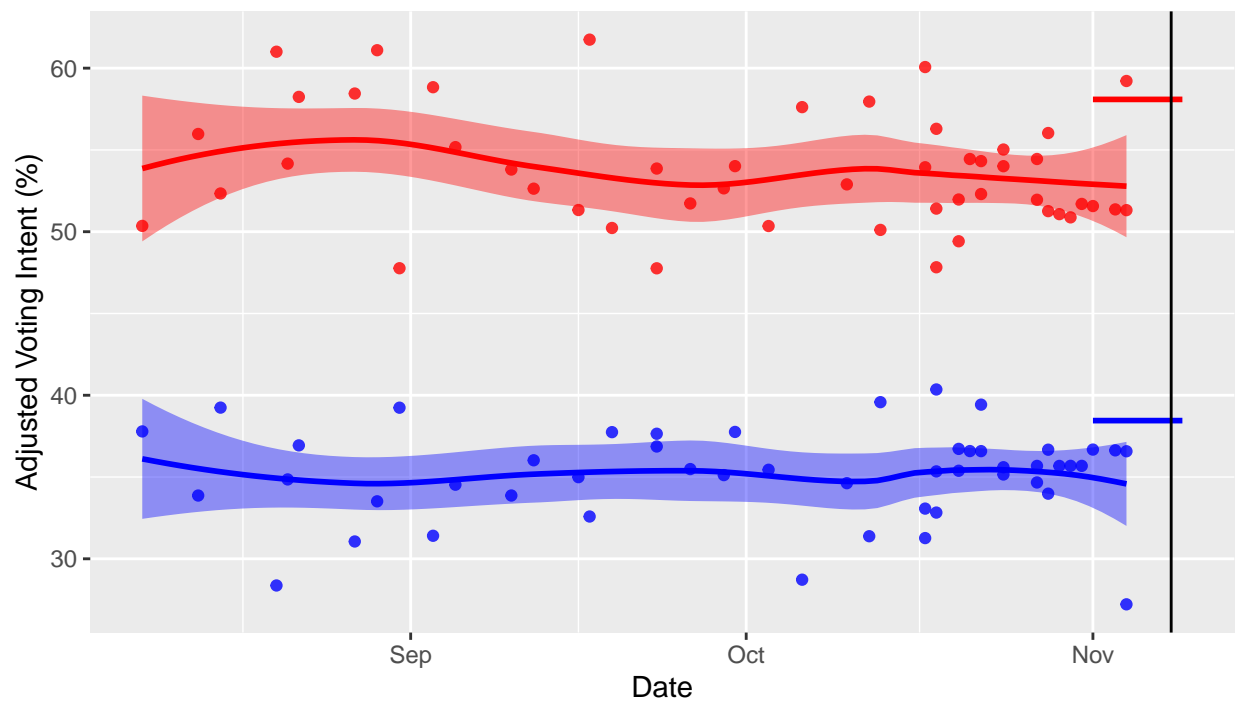
Kentucky – Adjusted Polls



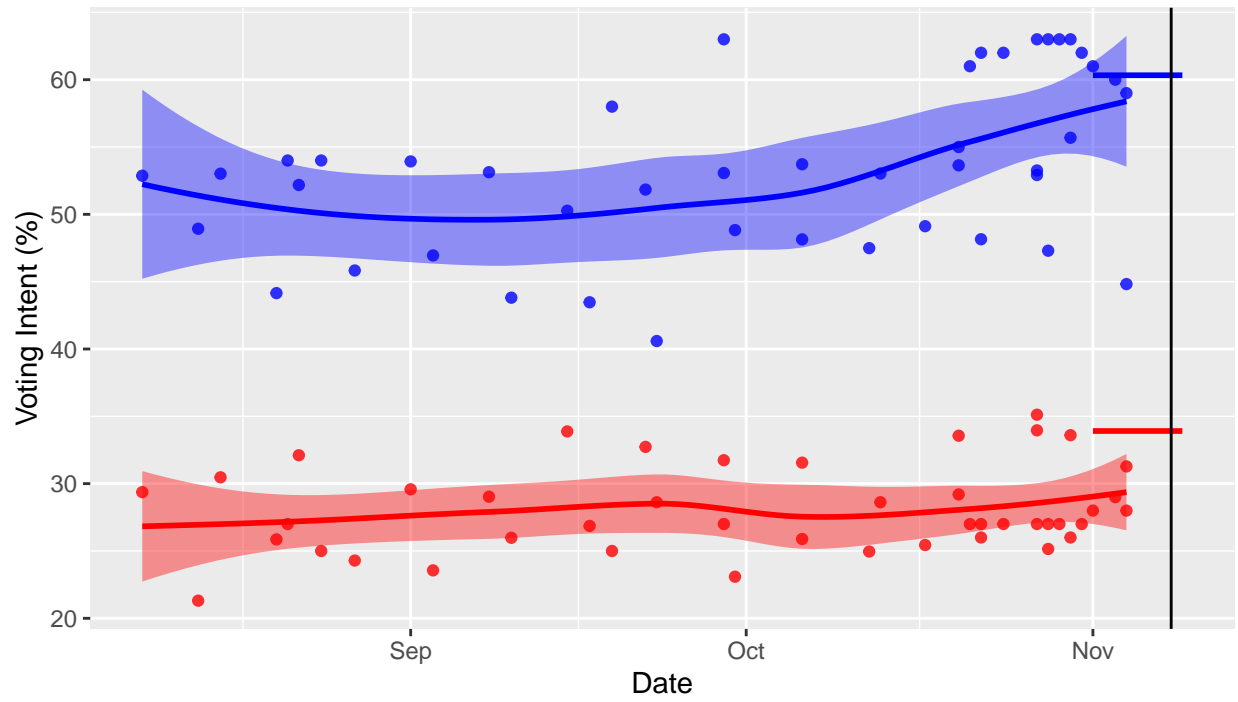
Louisiana – Raw Polls



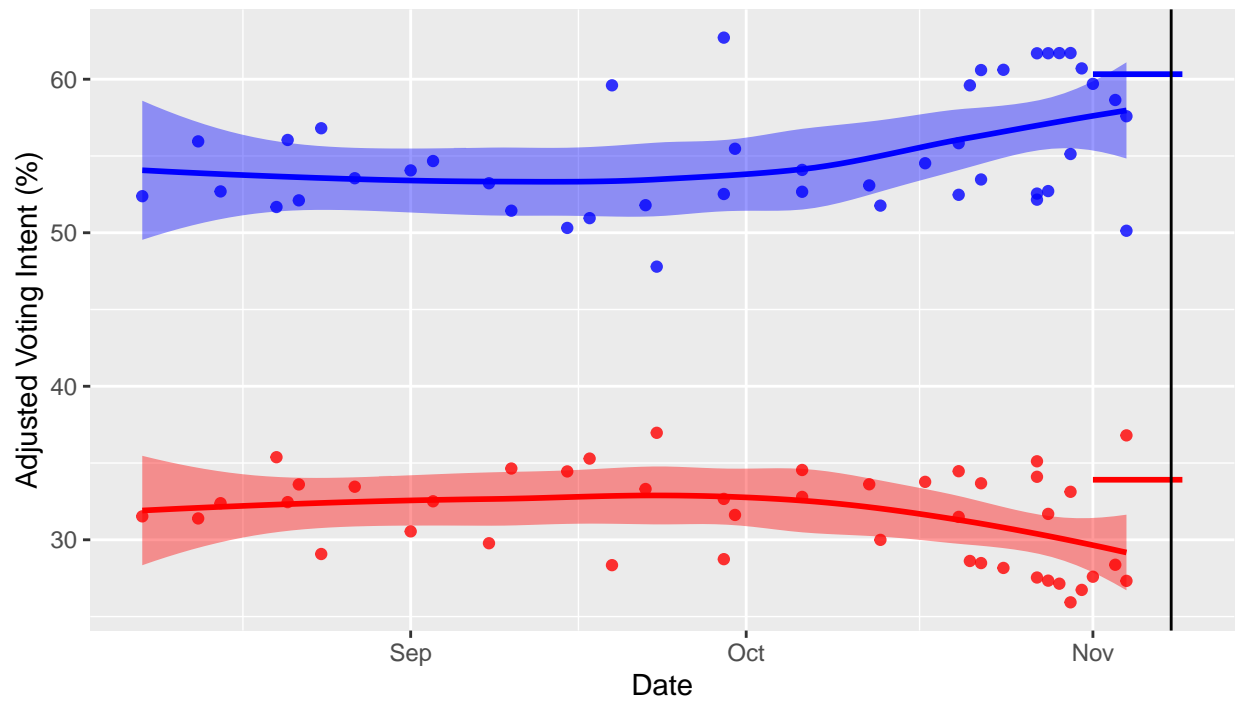
Louisiana – Adjusted Polls



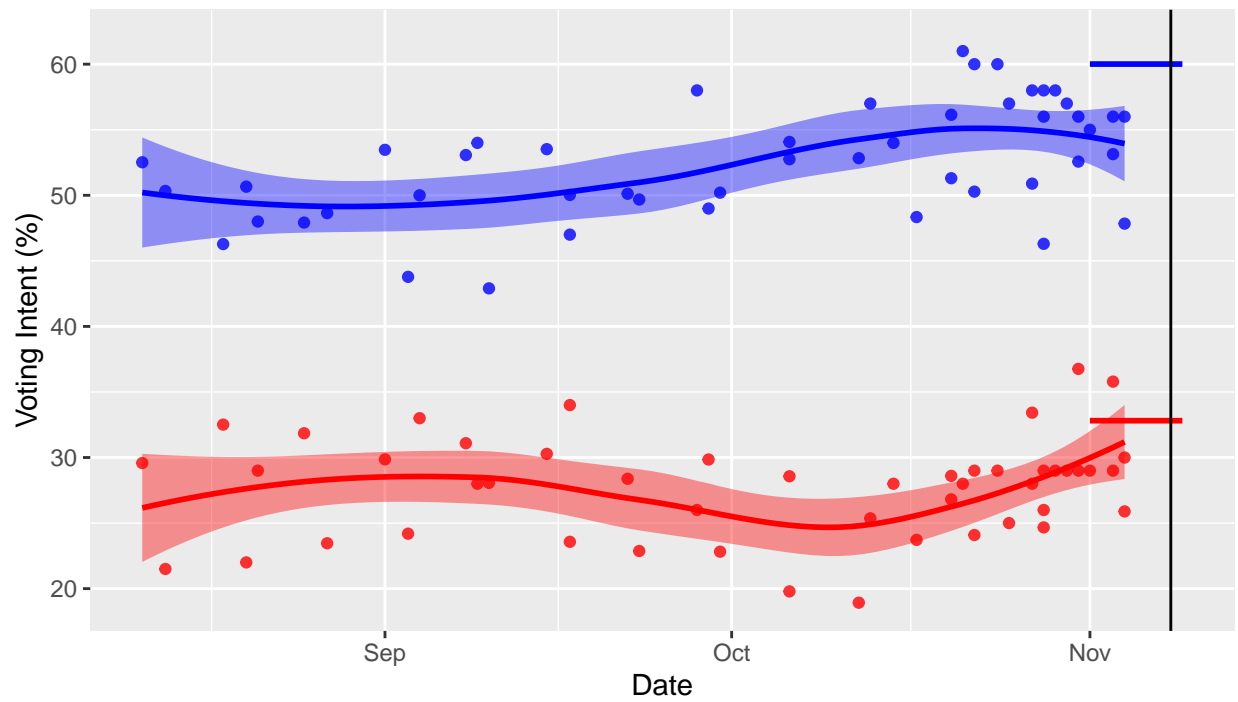
Maryland – Raw Polls



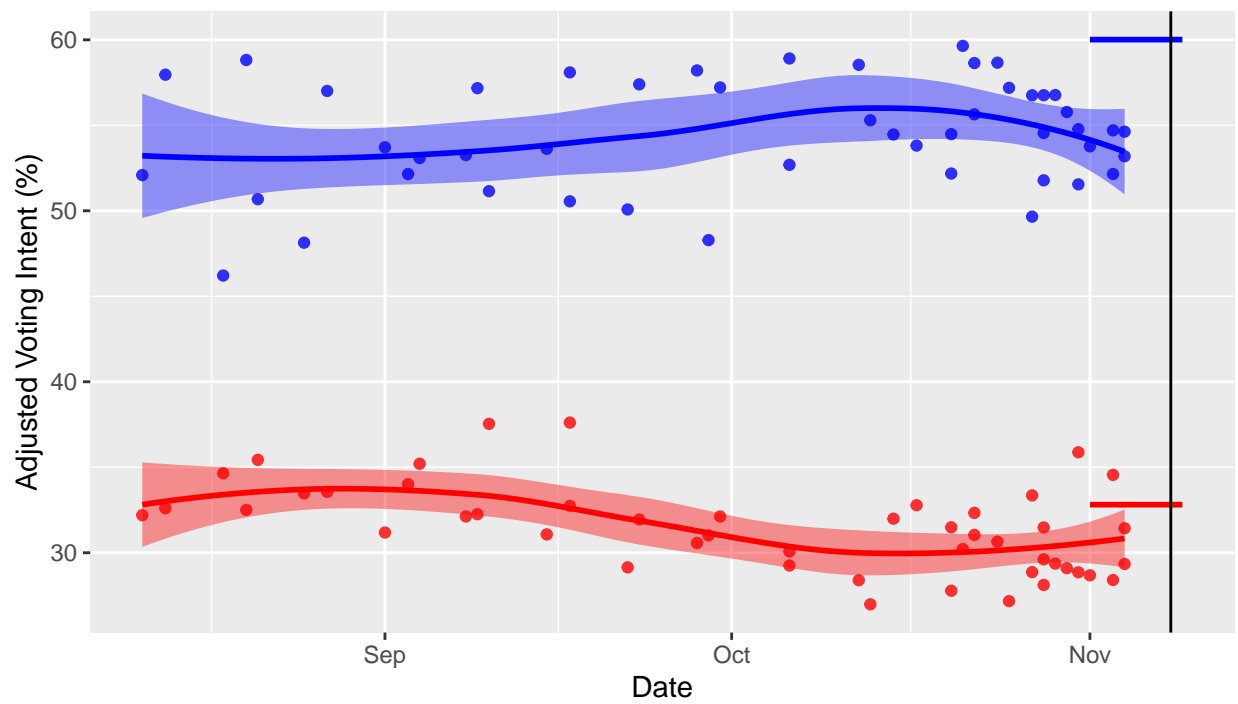
Maryland – Adjusted Polls



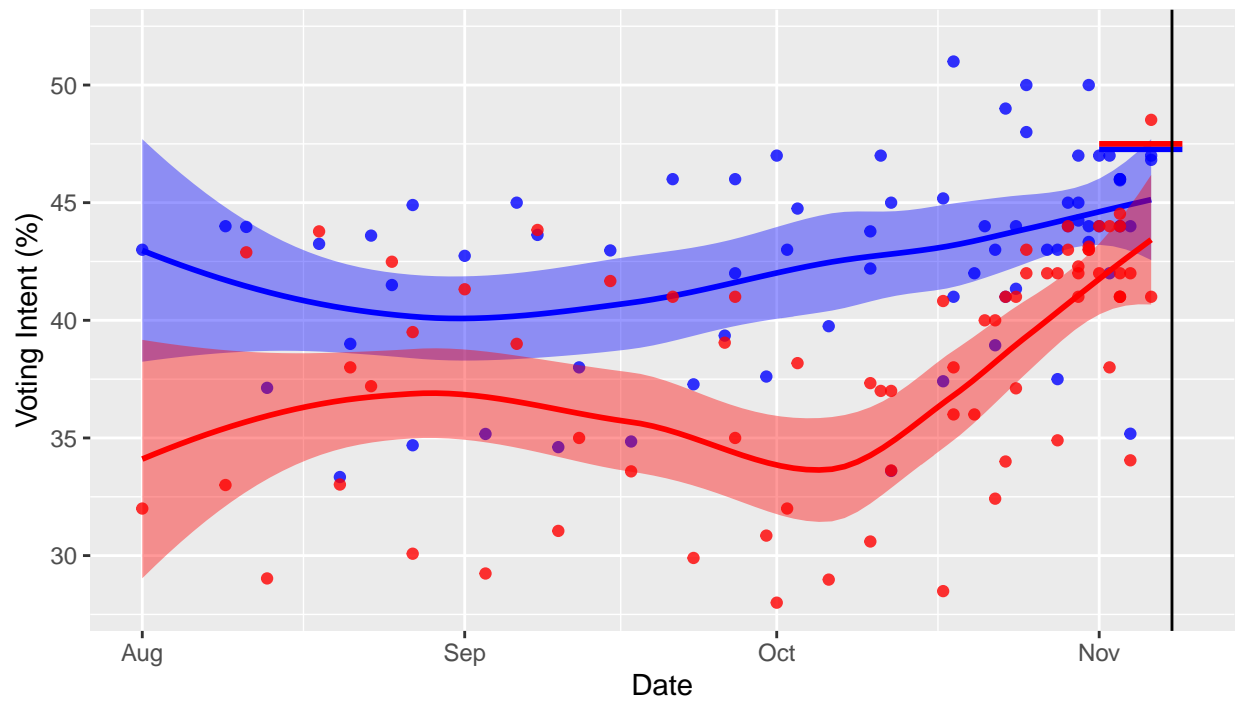
Massachusetts – Raw Polls



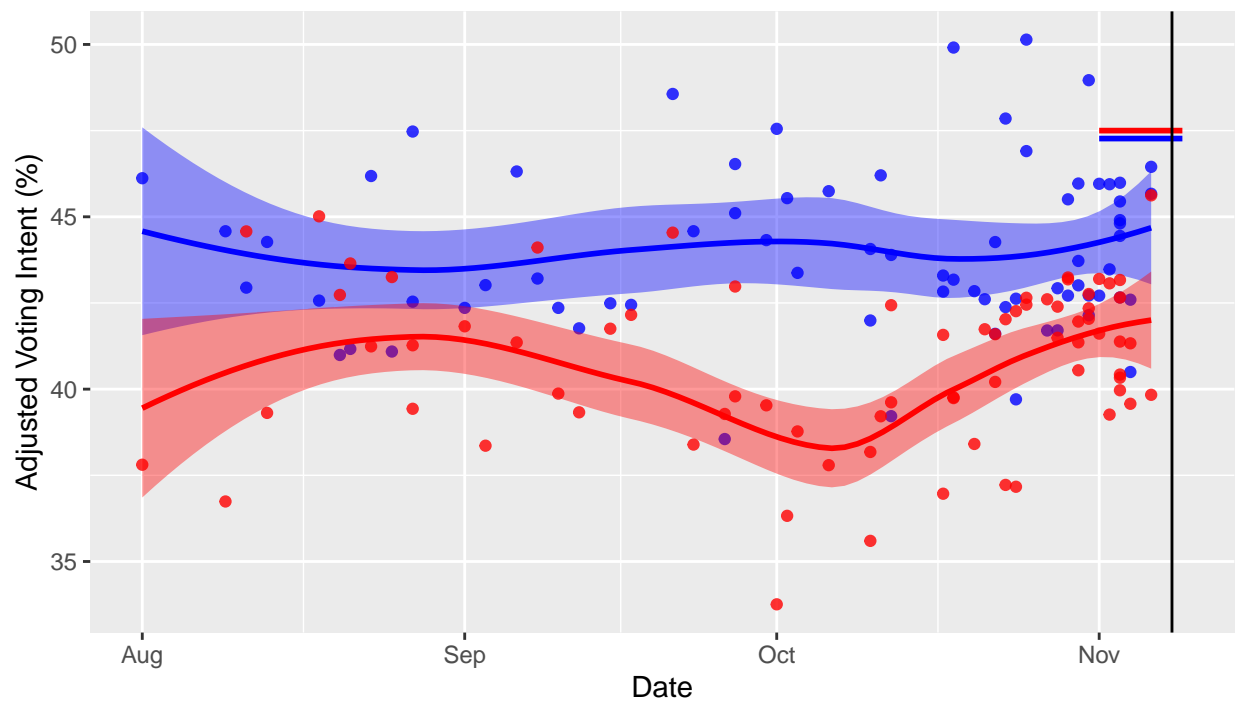
Massachusetts – Adjusted Polls



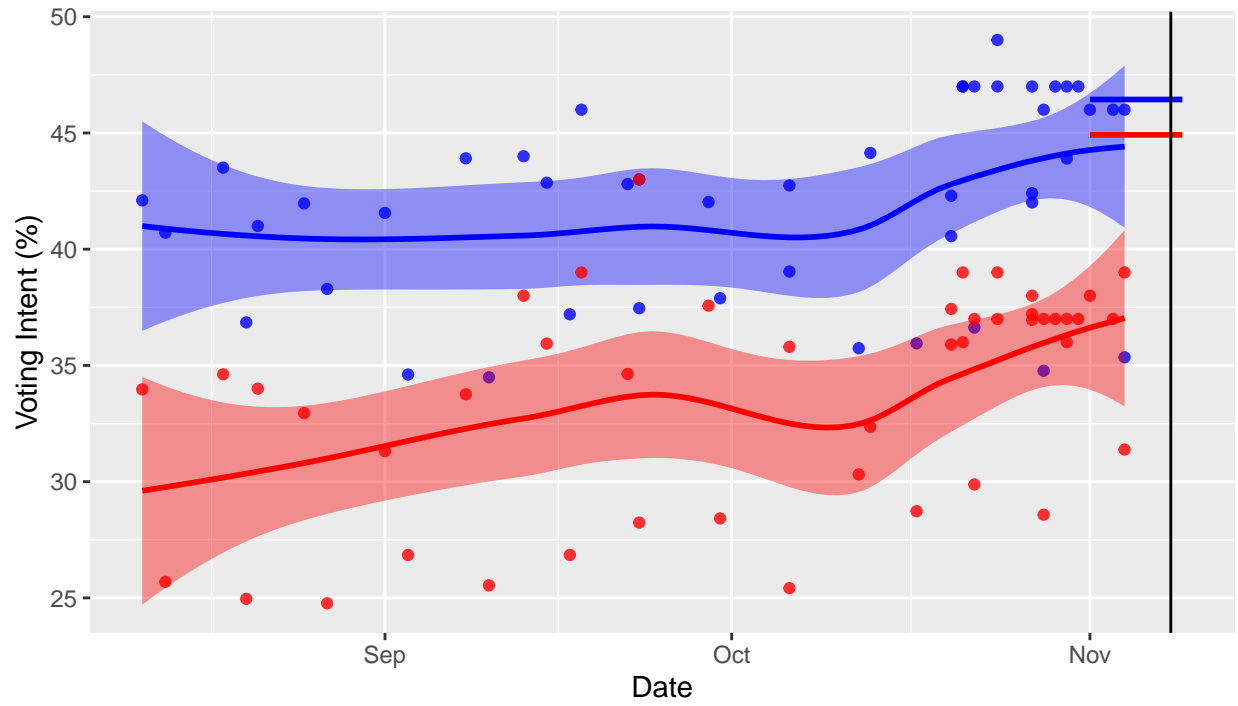
Michigan – Raw Polls



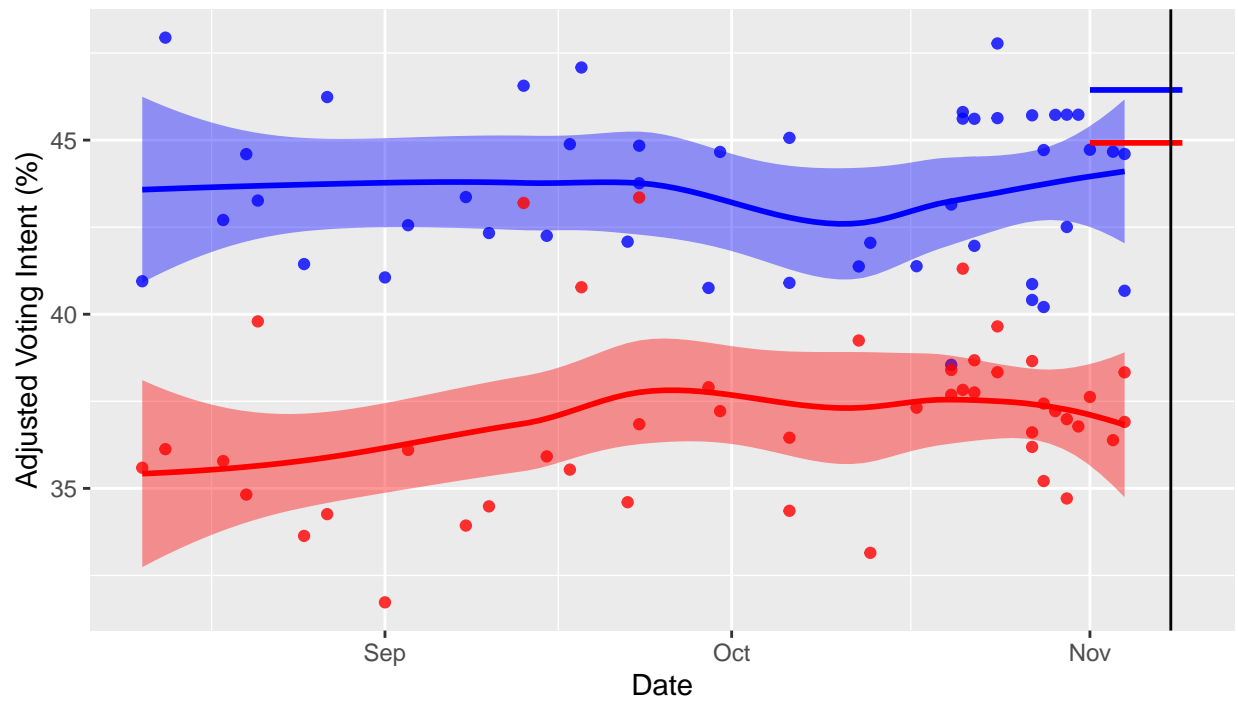
Michigan – Adjusted Polls



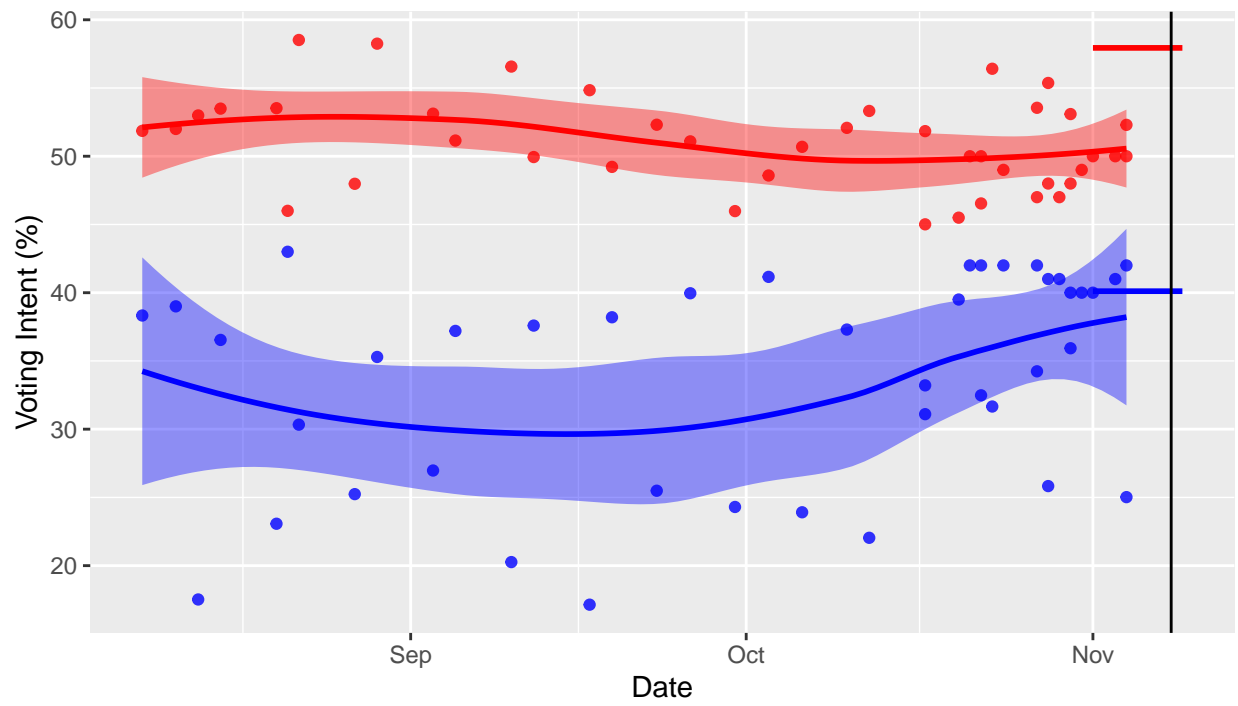
Minnesota – Raw Polls



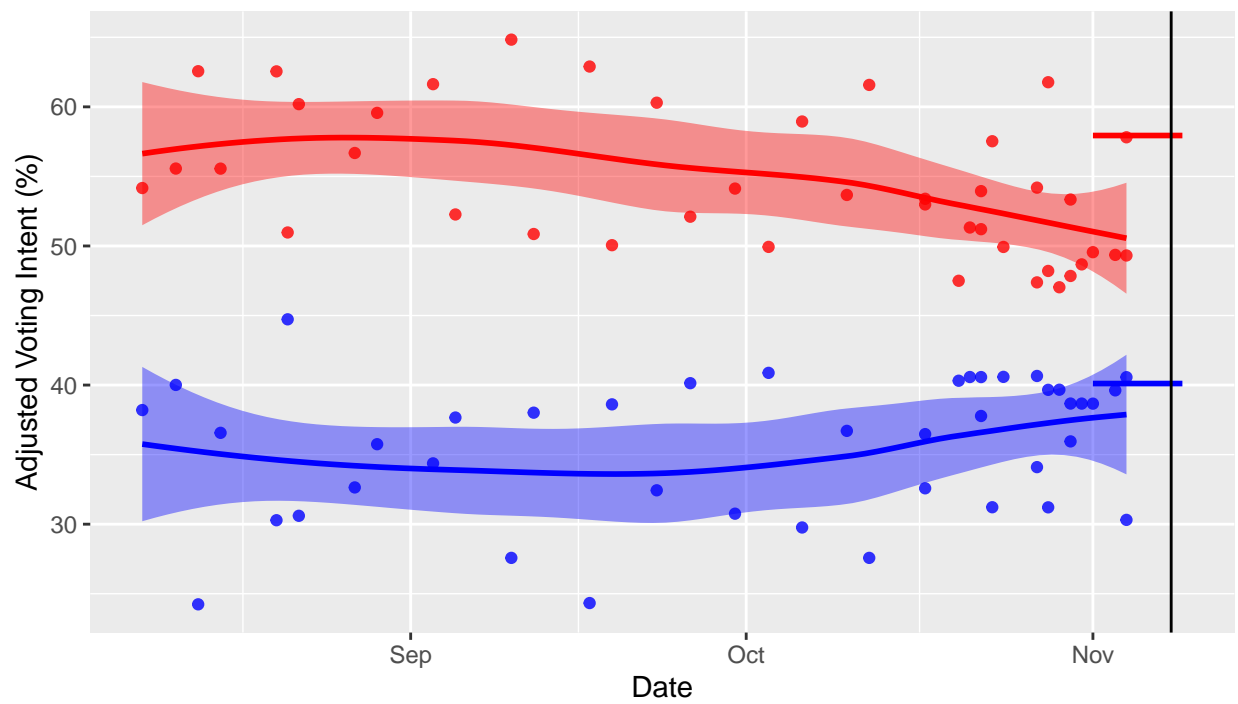
Minnesota – Adjusted Polls



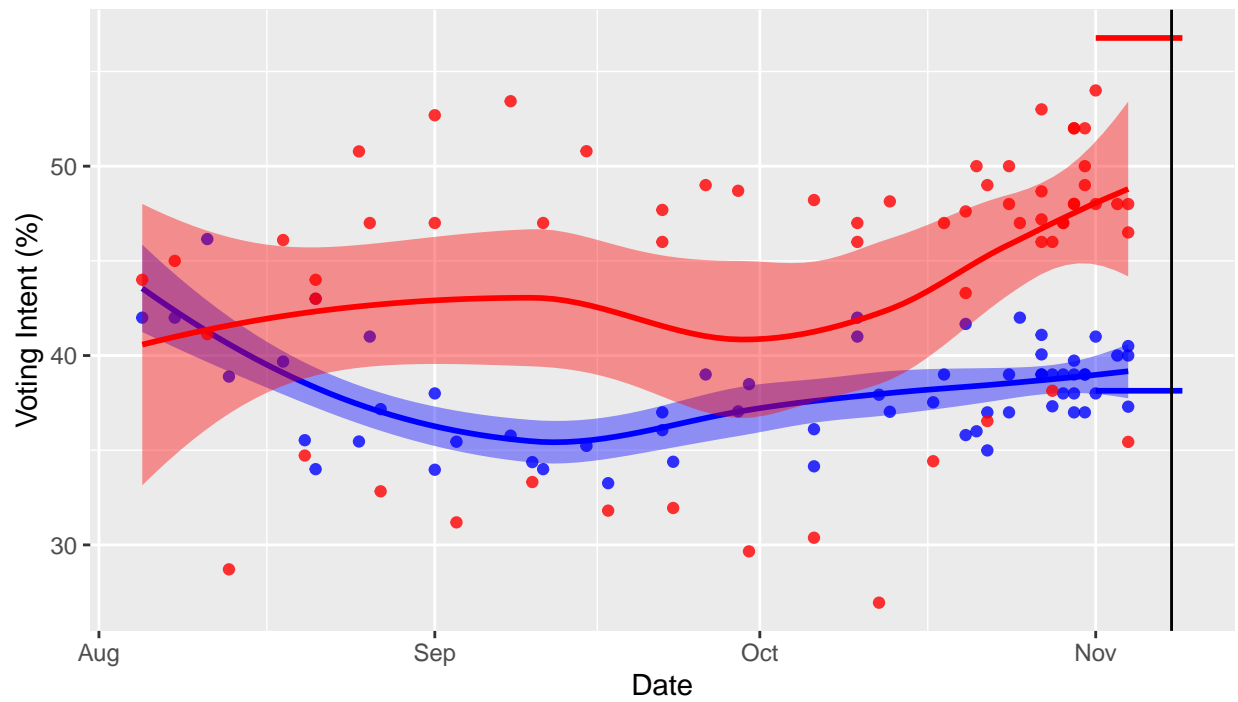
Mississippi – Raw Polls



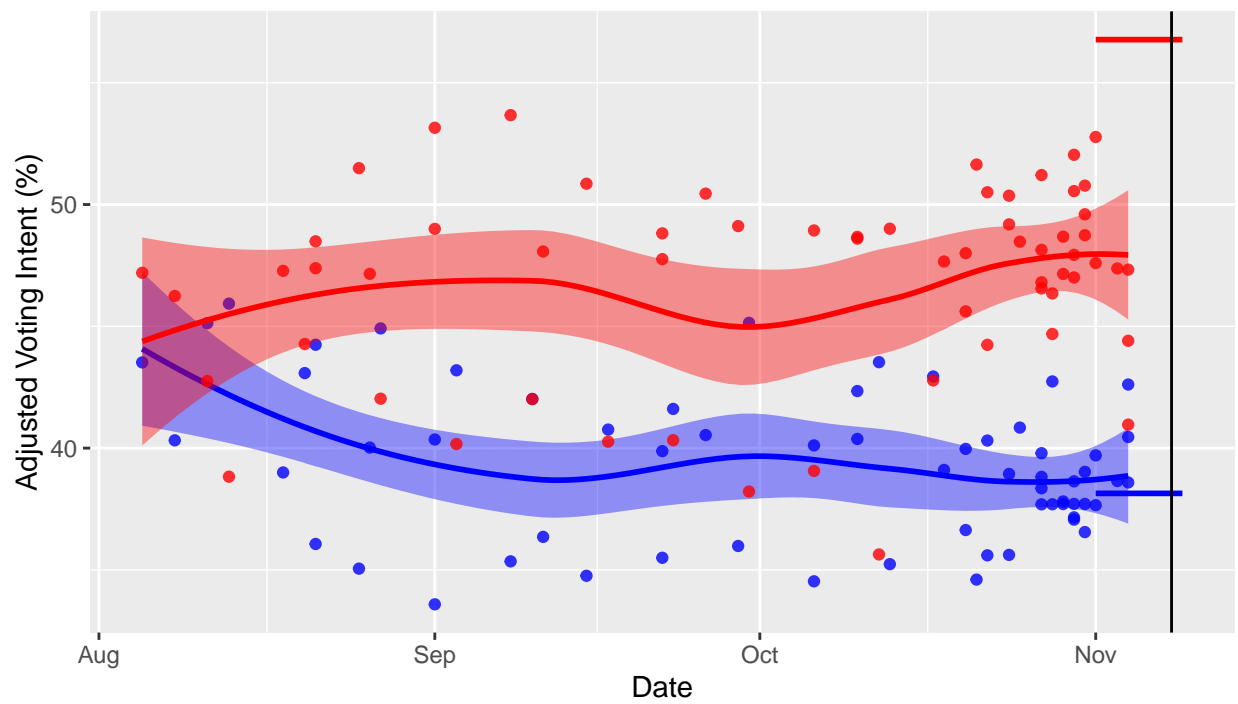
Mississippi – Adjusted Polls



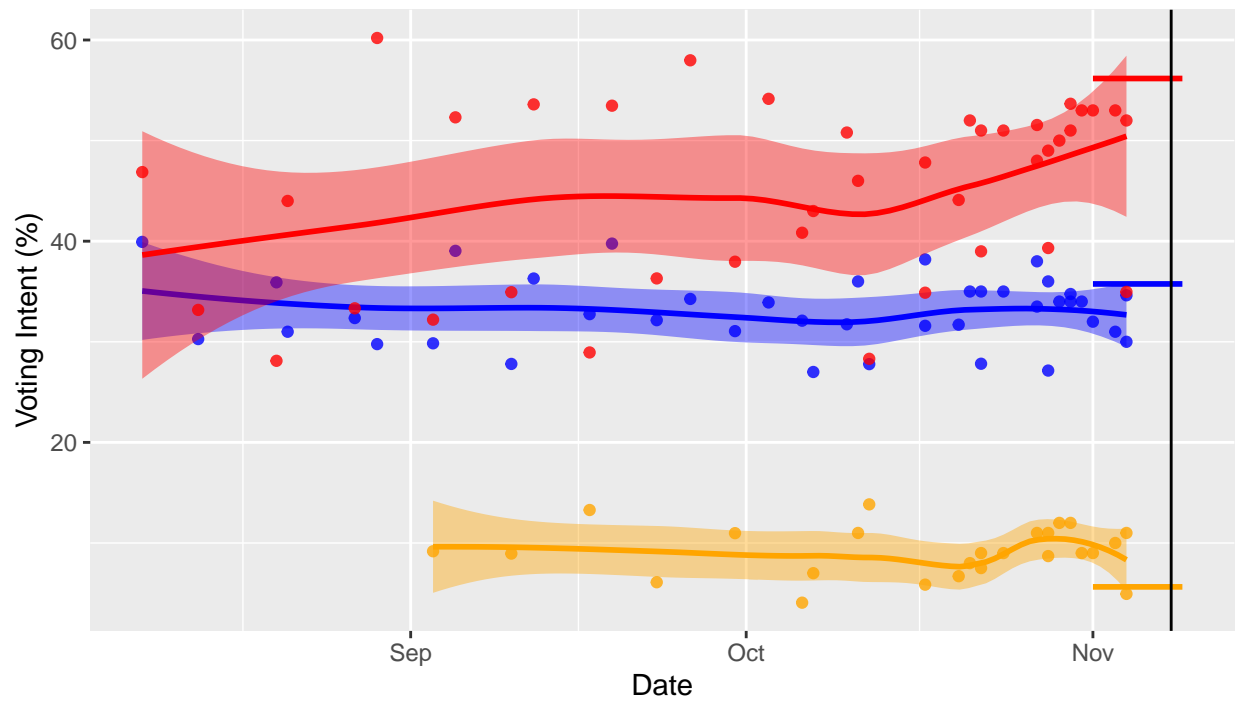
Missouri – Raw Polls



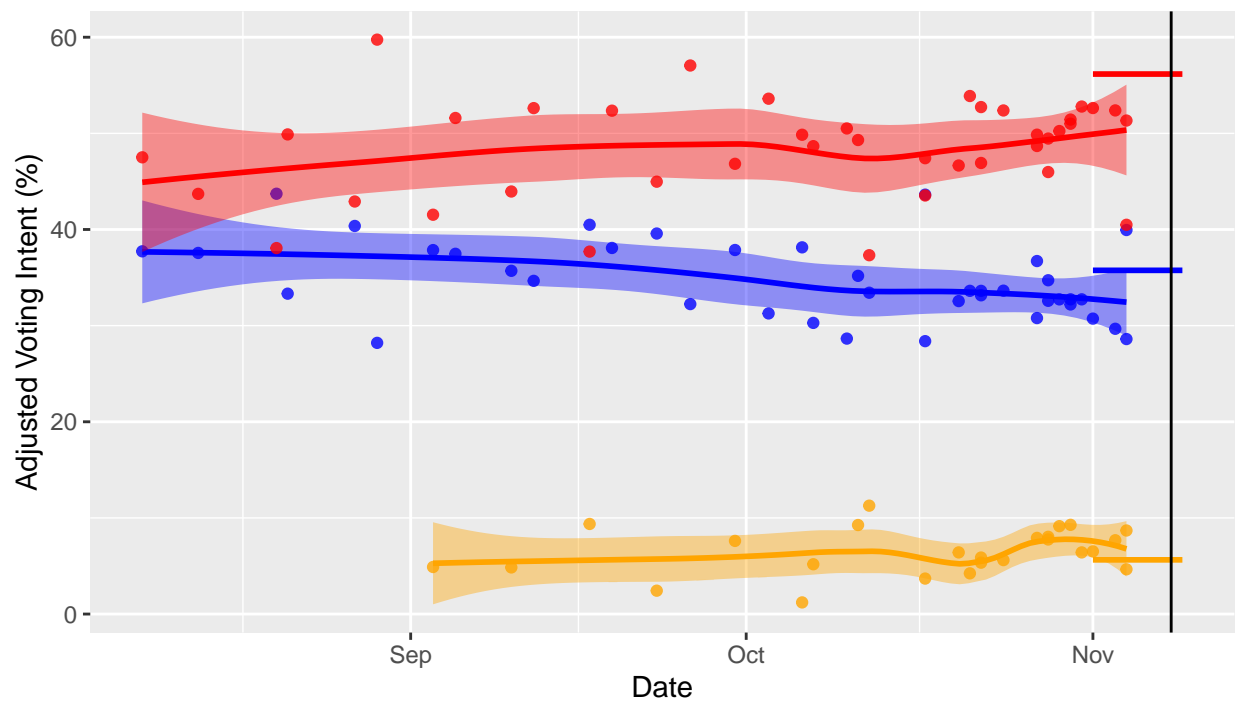
Missouri – Adjusted Polls



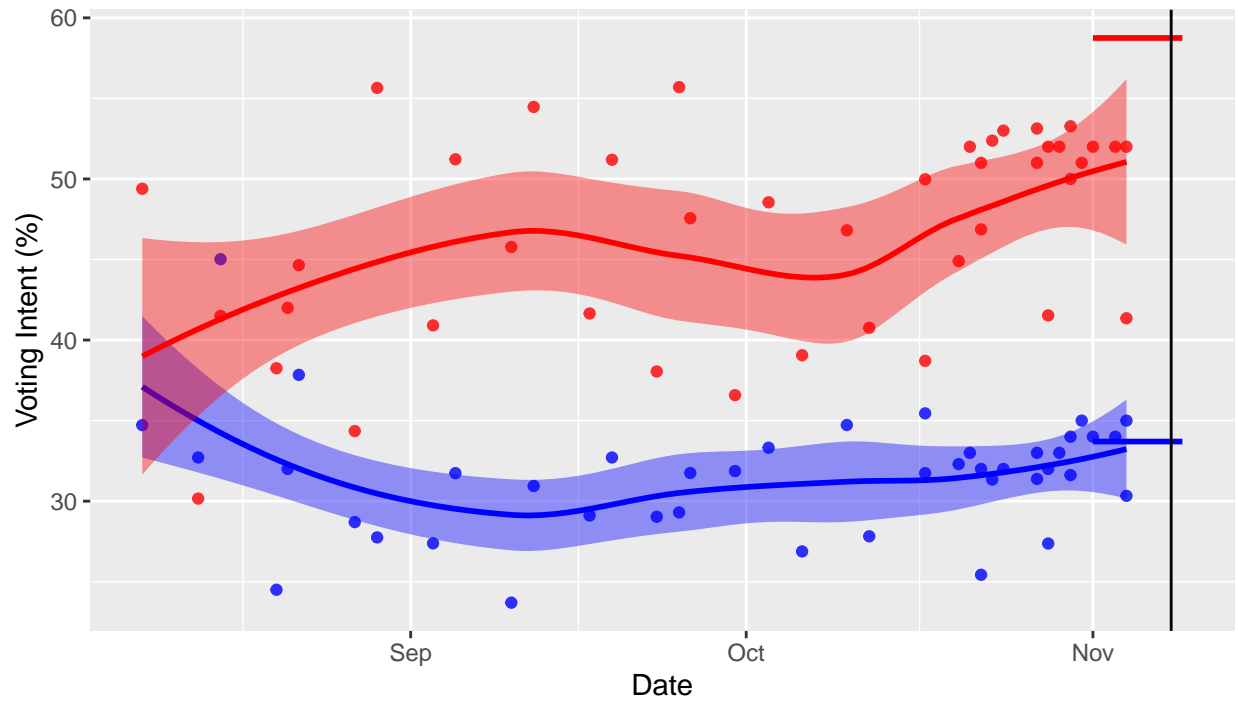
Montana – Raw Polls



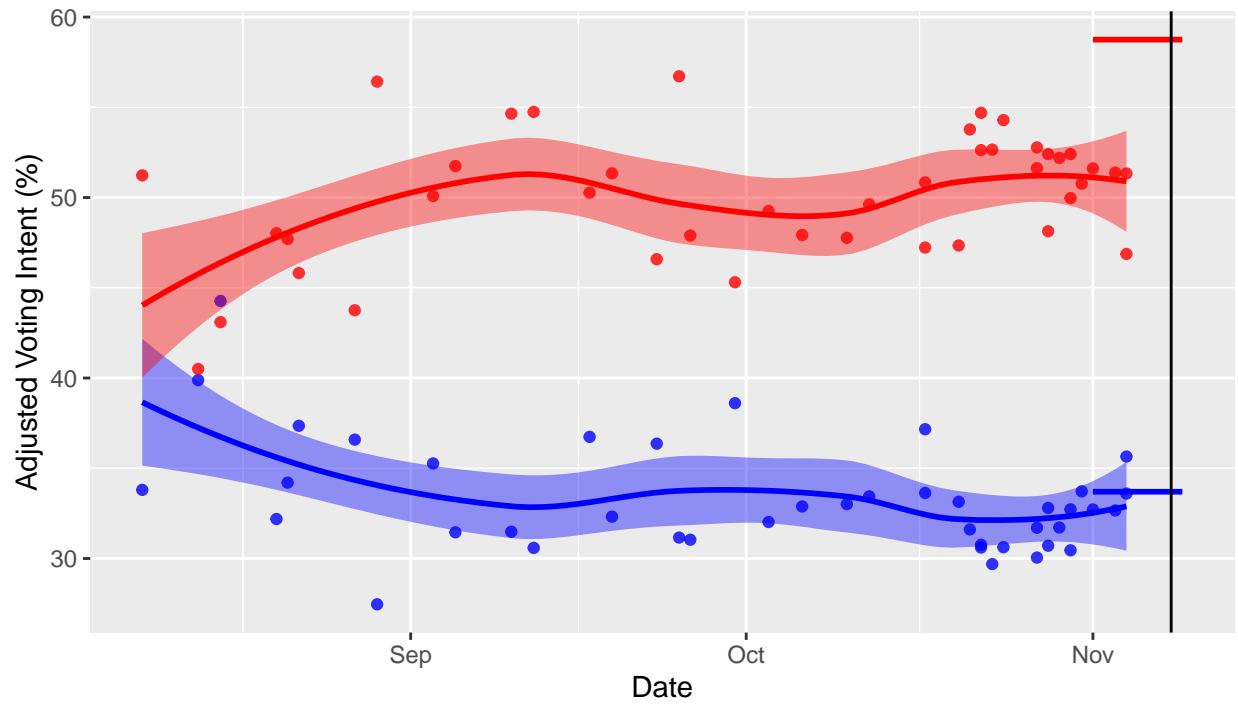
Montana – Adjusted Polls



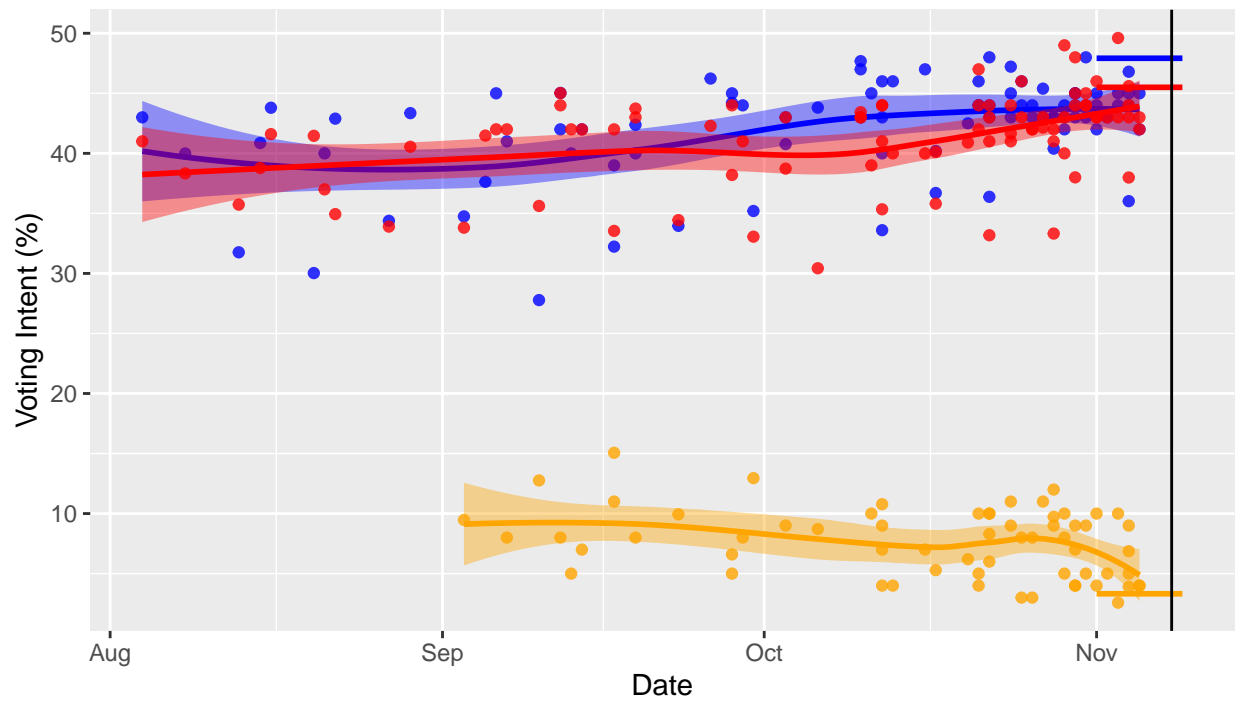
Nebraska – Raw Polls



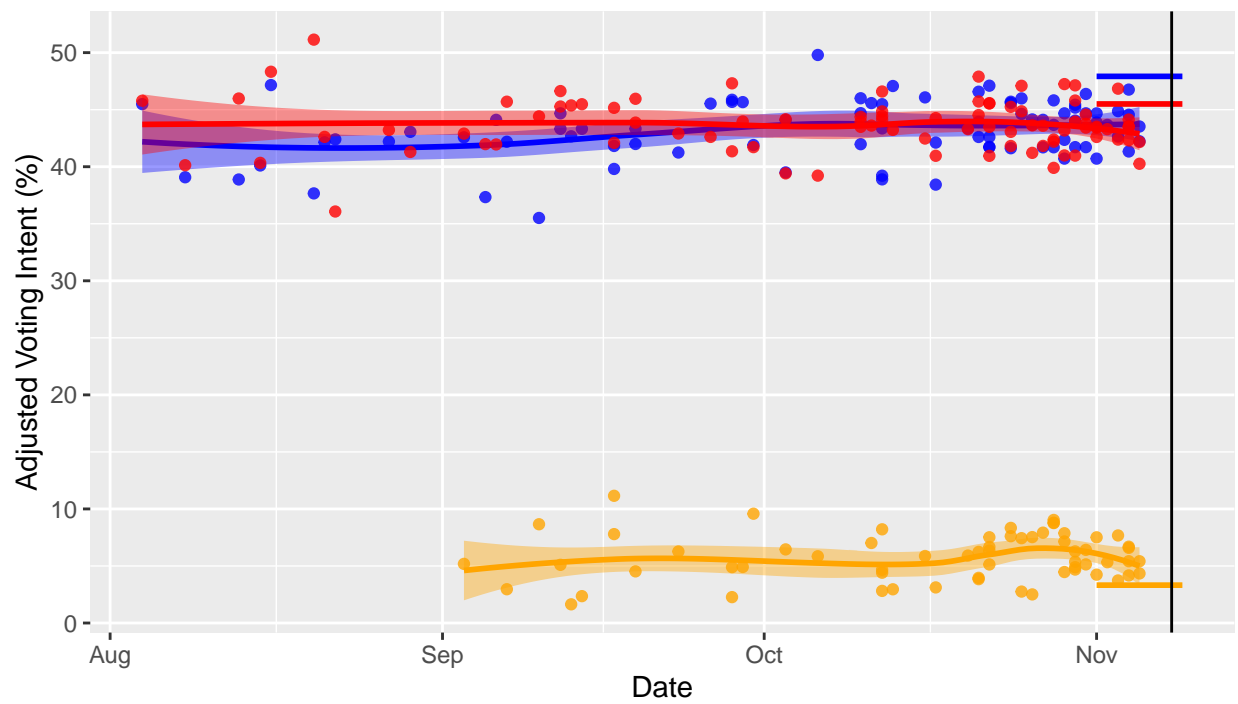
Nebraska – Adjusted Polls



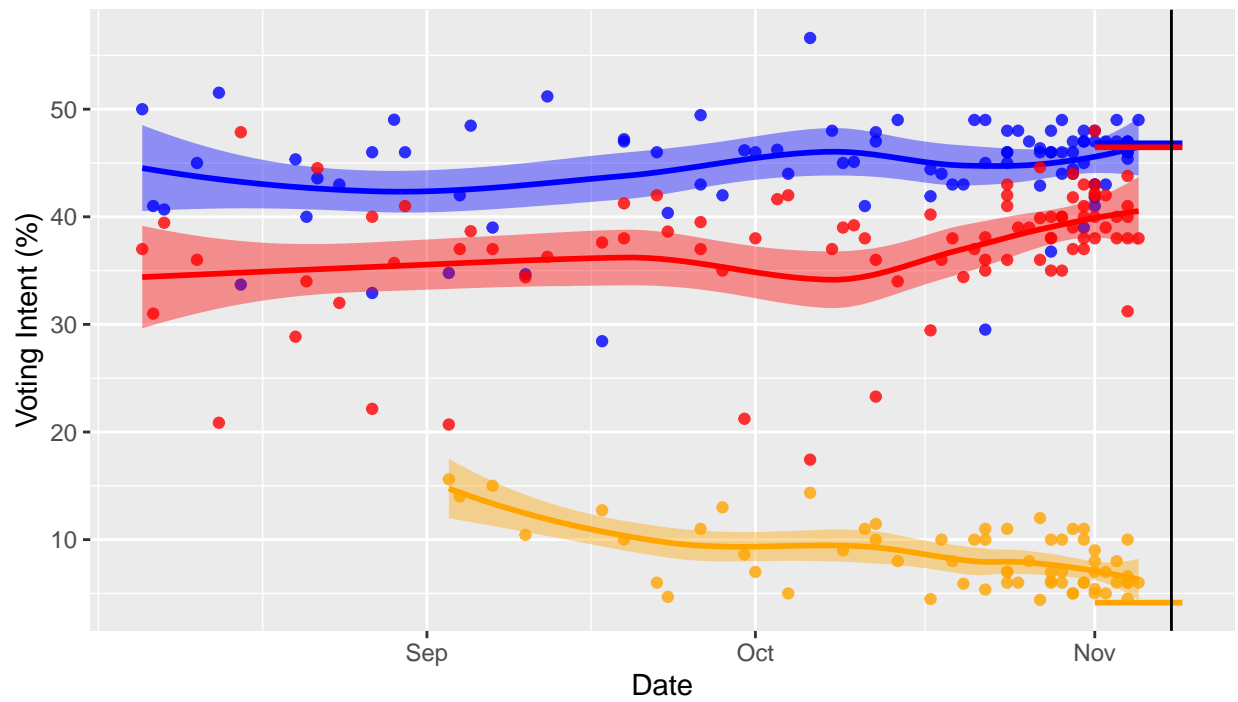
Nevada – Raw Polls



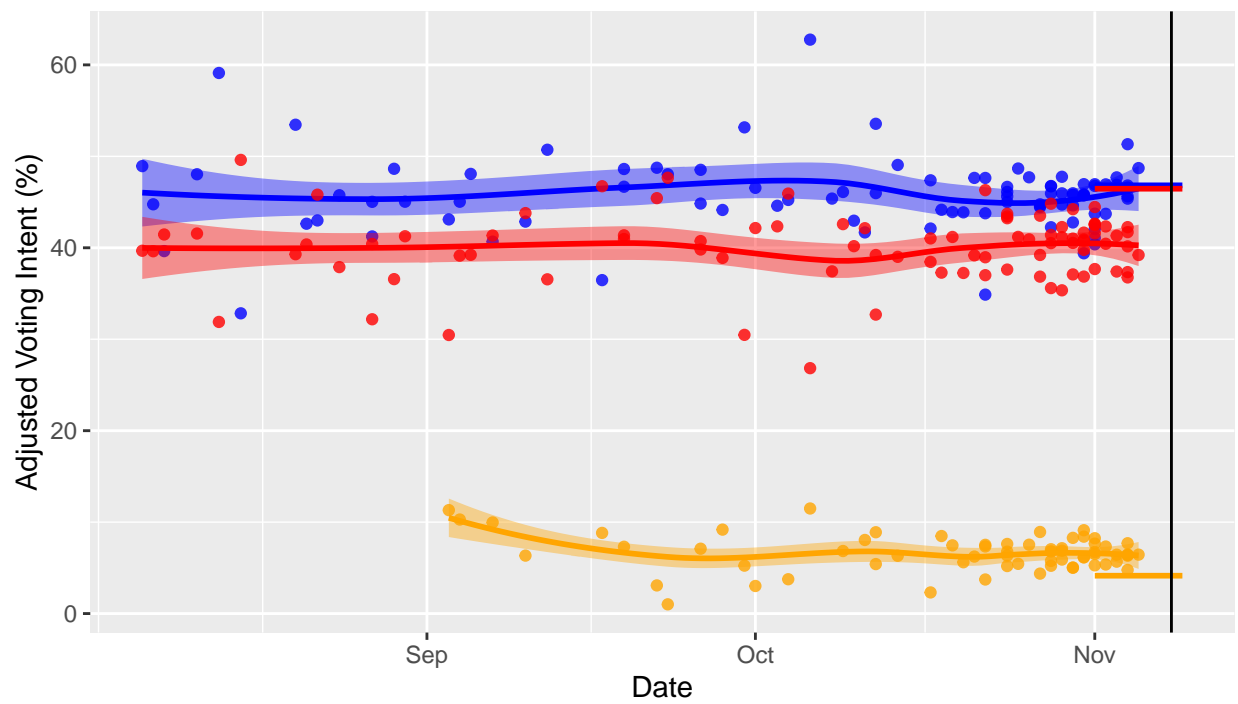
Nevada – Adjusted Polls



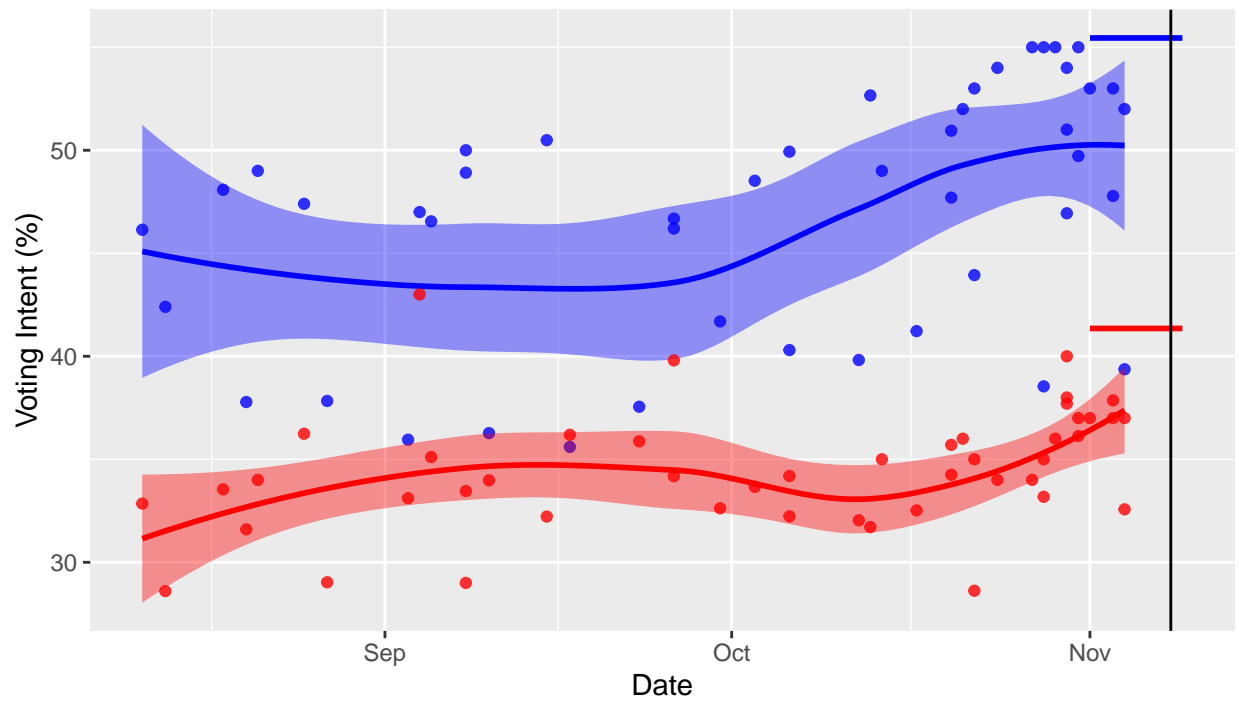
New Hampshire – Raw Polls



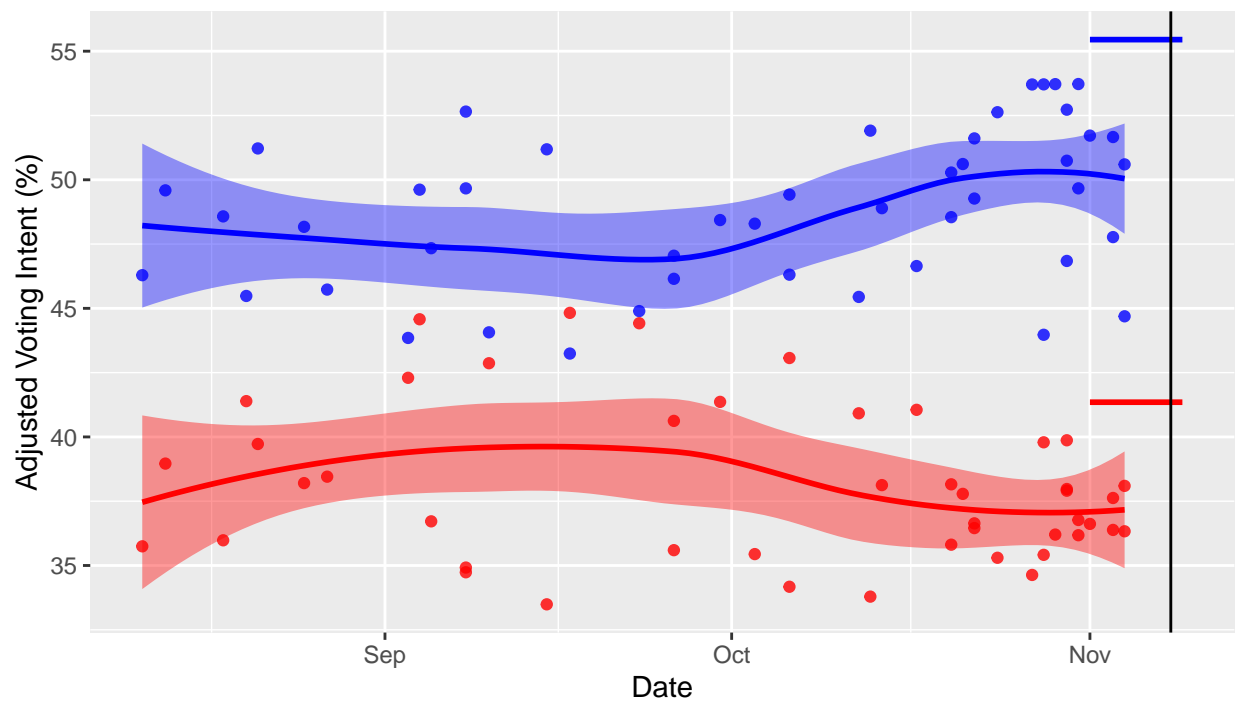
New Hampshire – Adjusted Polls



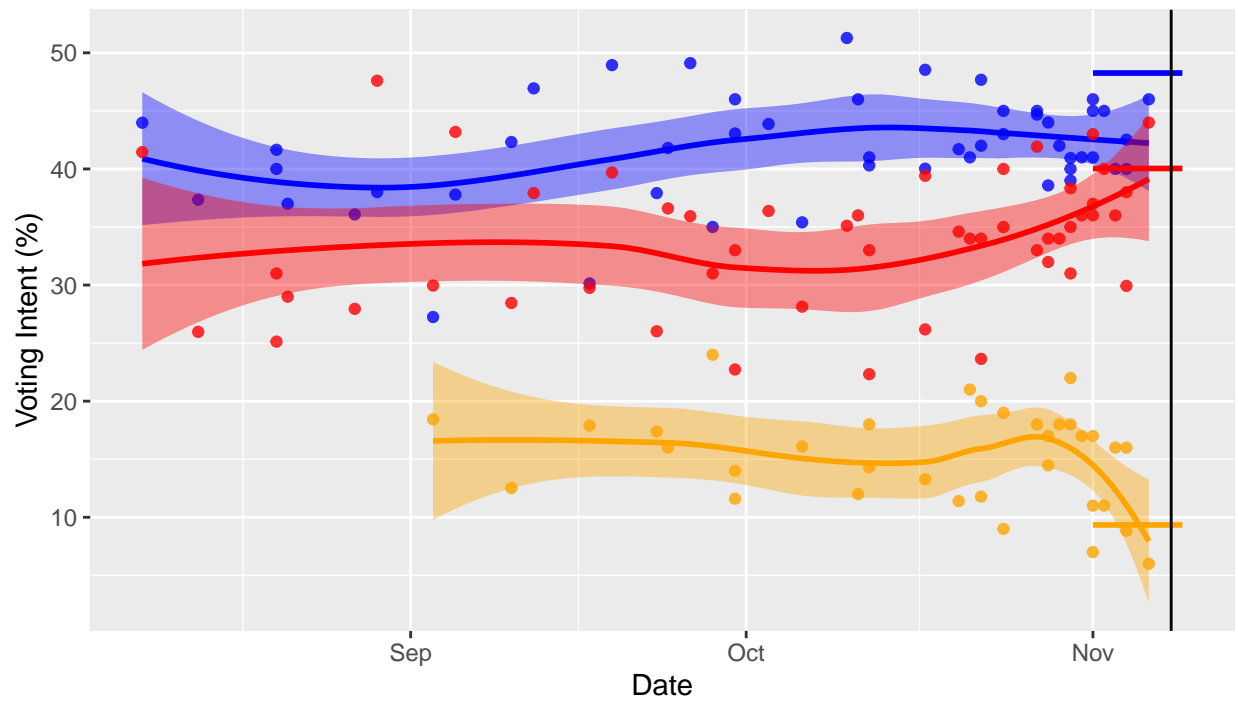
New Jersey – Raw Polls



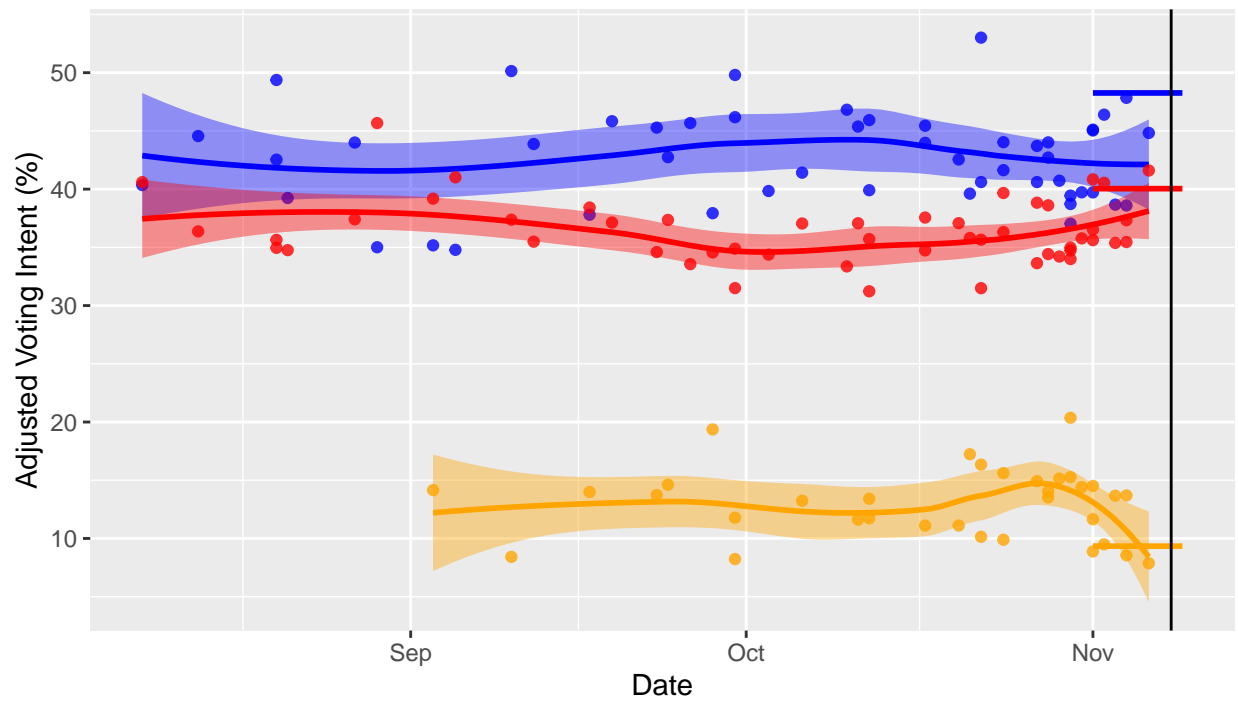
New Jersey – Adjusted Polls



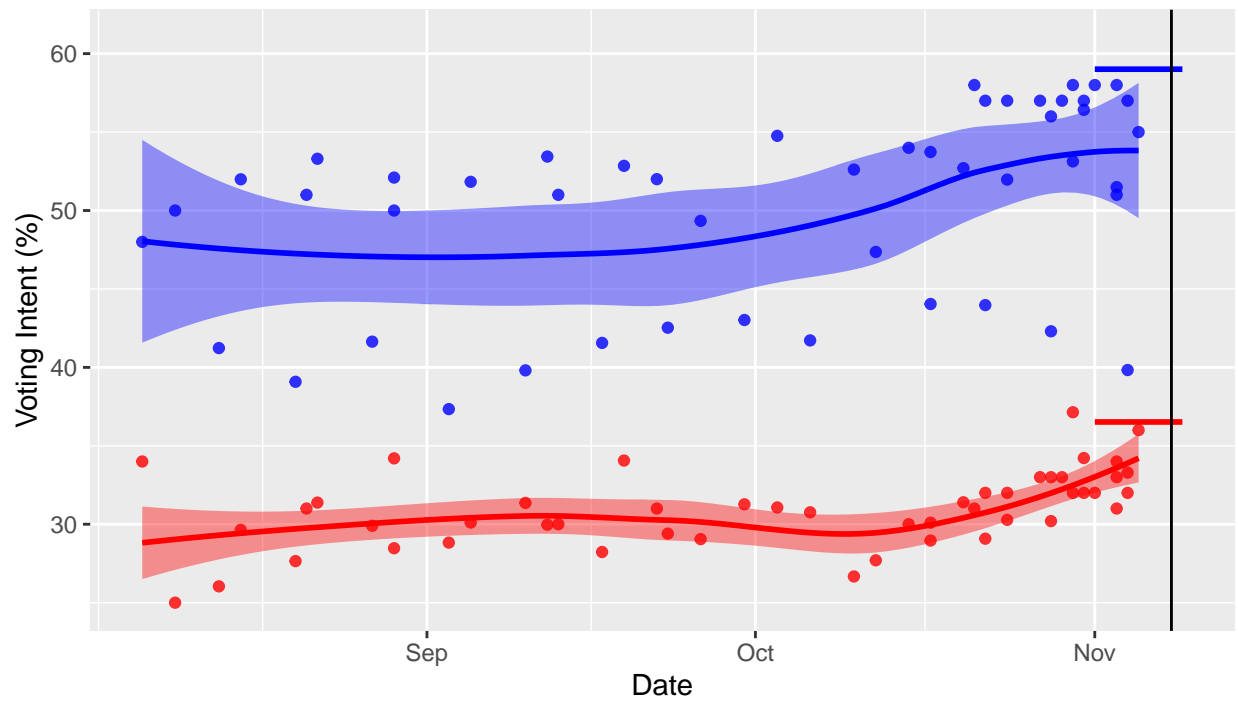
New Mexico – Raw Polls



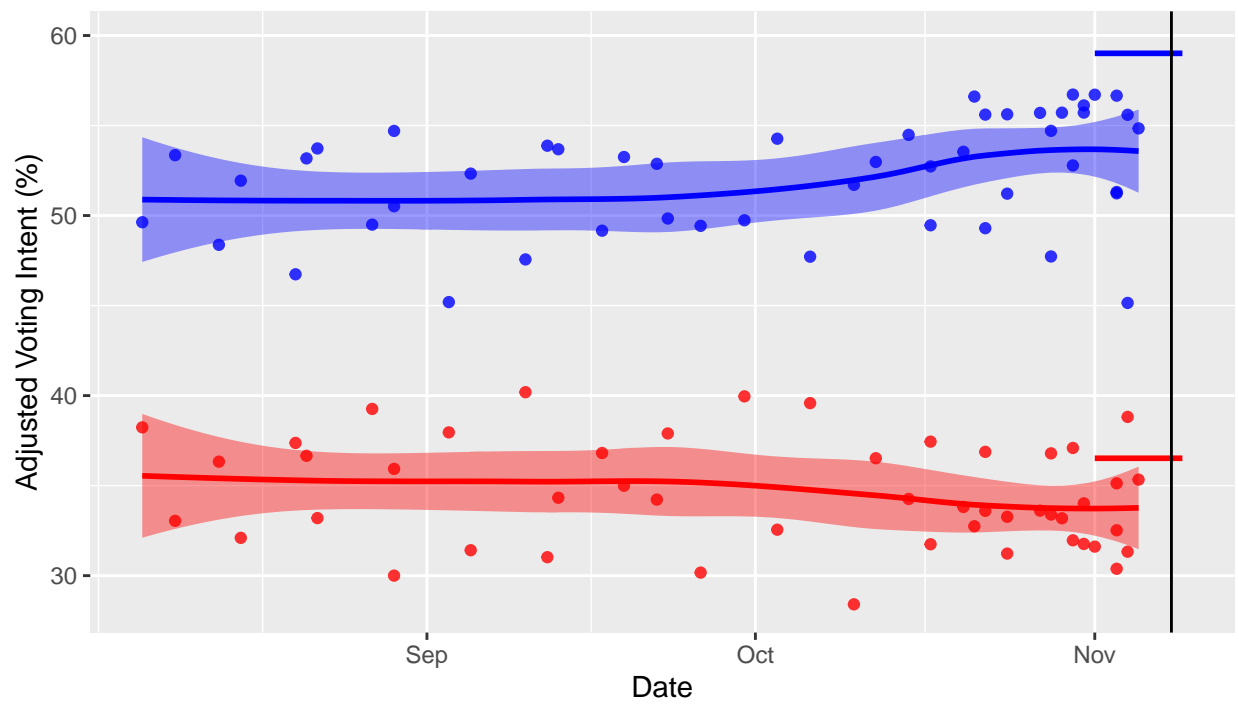
New Mexico – Adjusted Polls



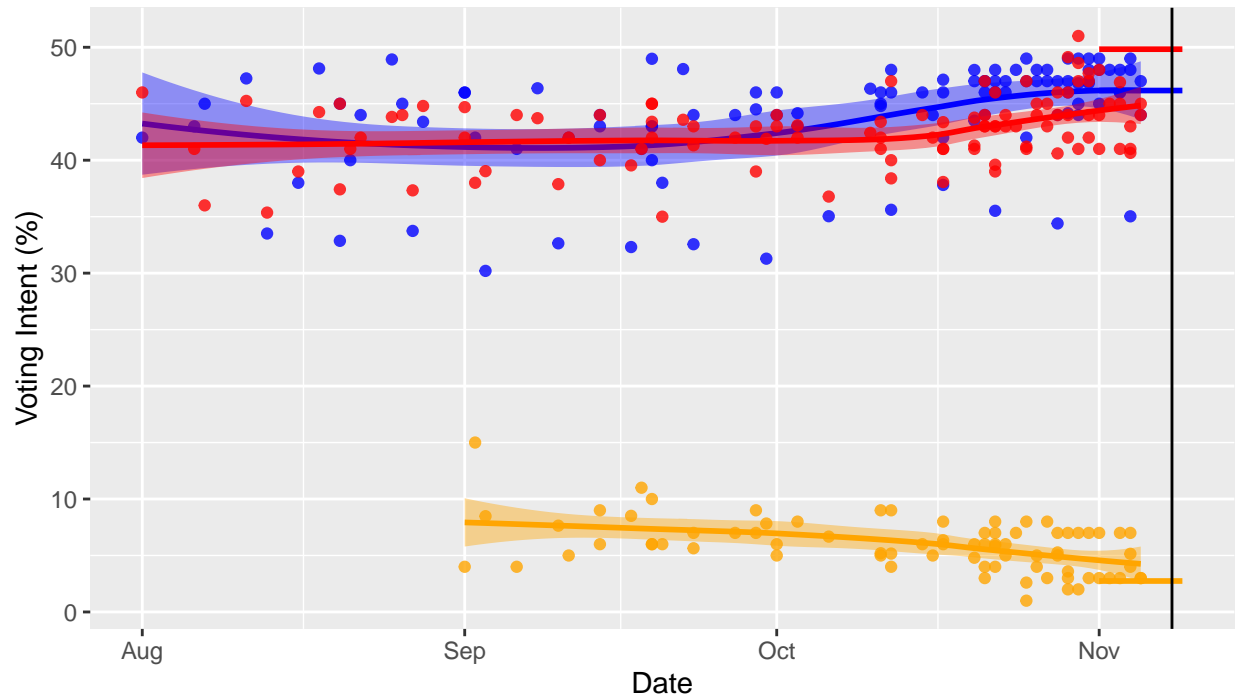
New York – Raw Polls



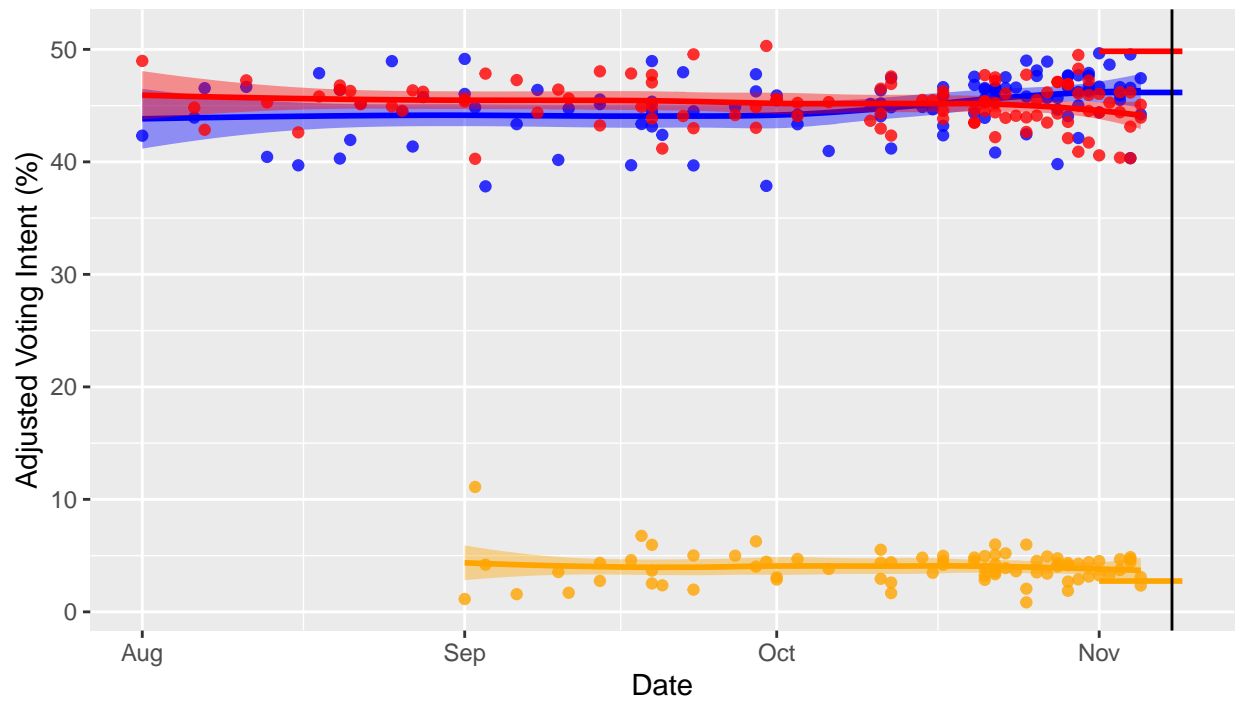
New York – Adjusted Polls



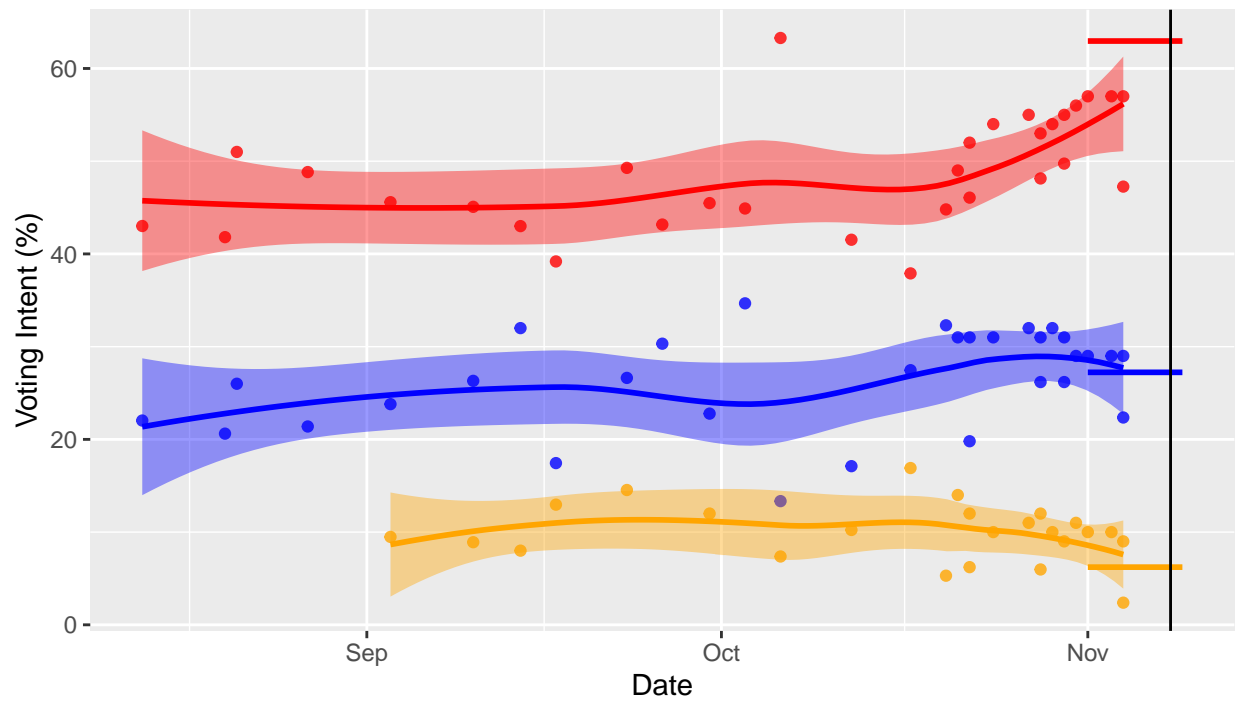
North Carolina – Raw Polls



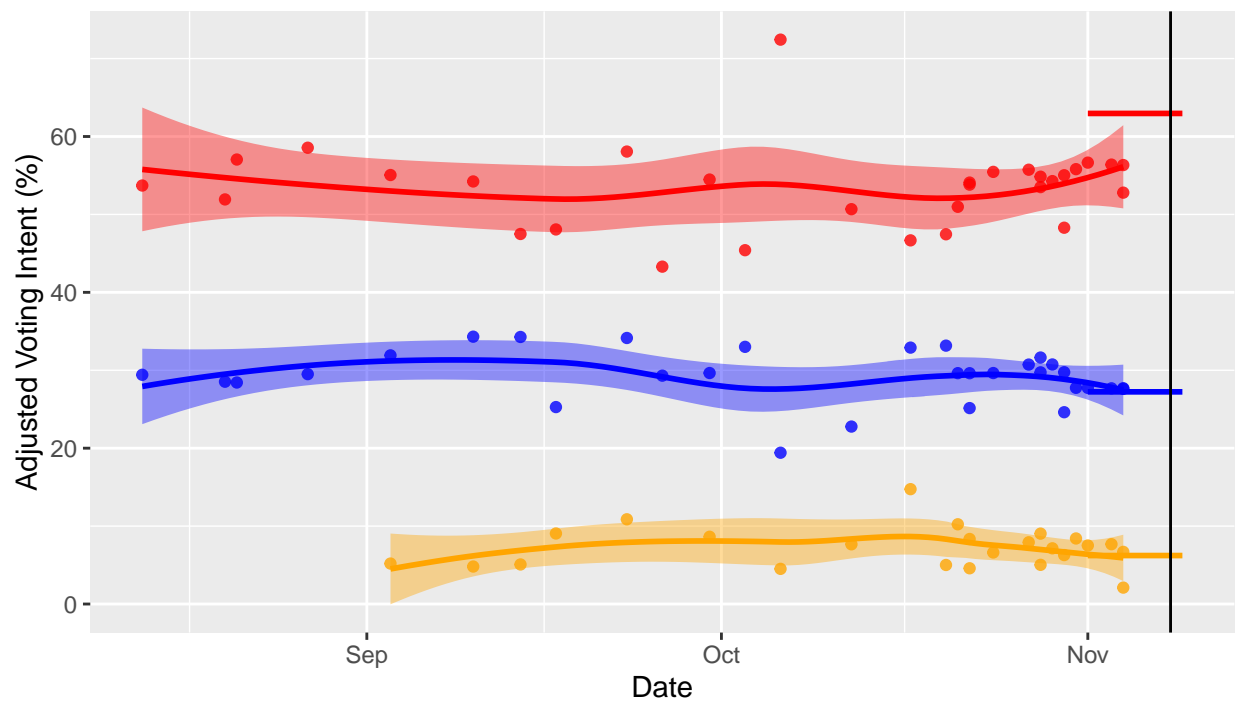
North Carolina – Adjusted Polls



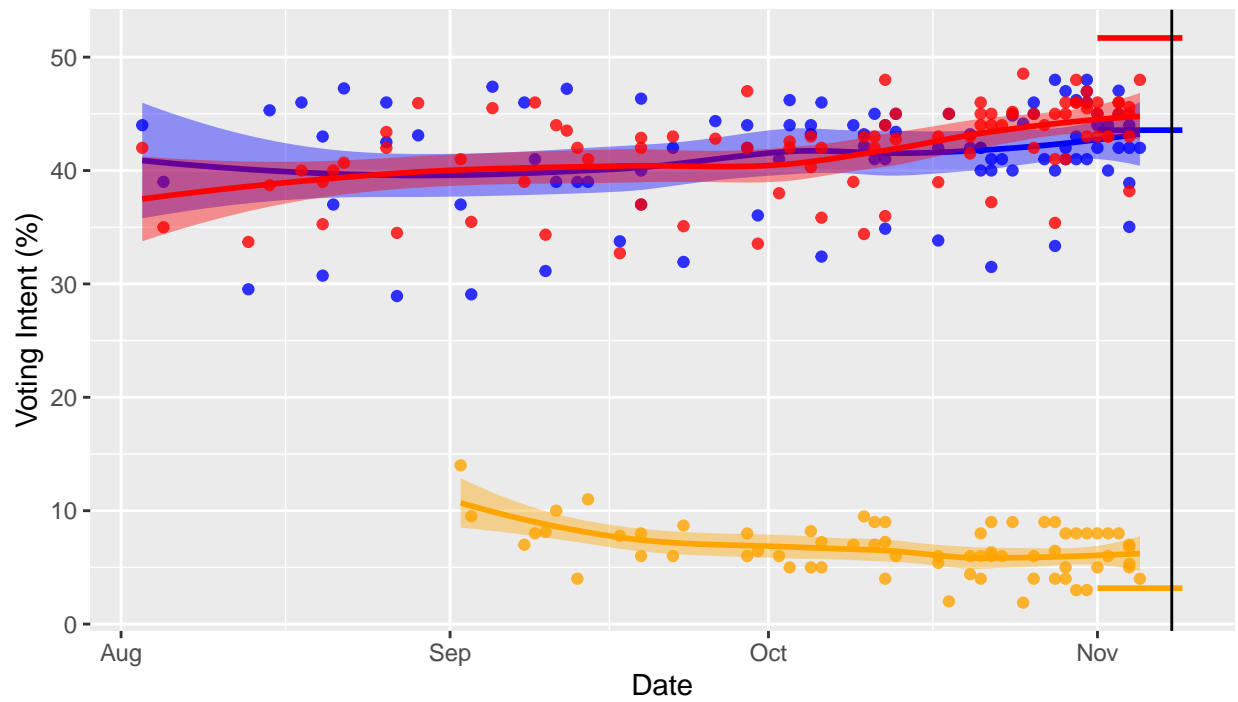
North Dakota – Raw Polls



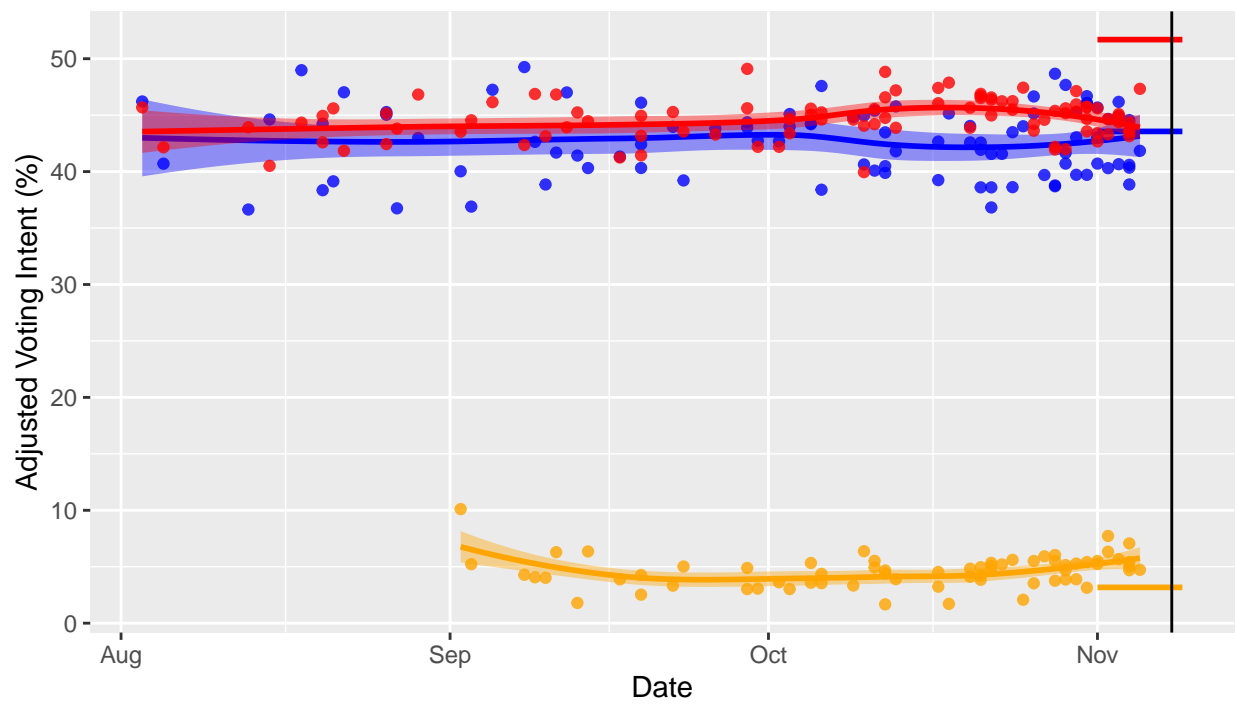
North Dakota – Adjusted Polls



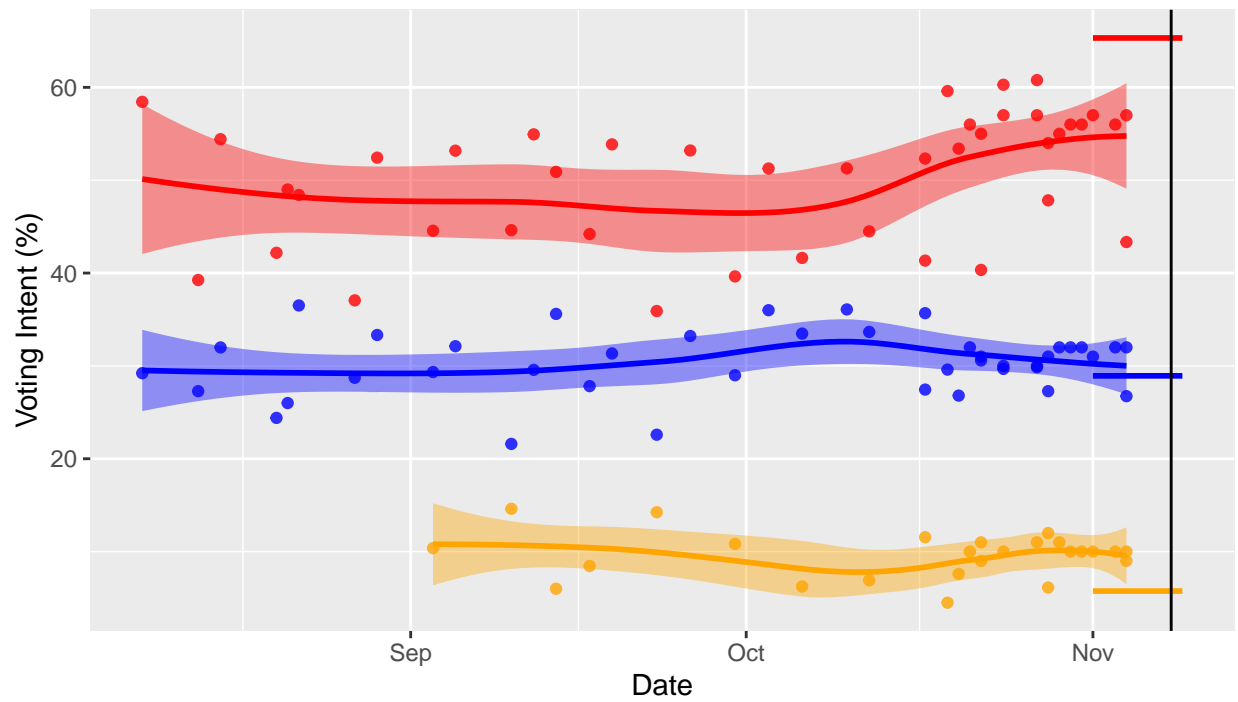
Ohio – Raw Polls



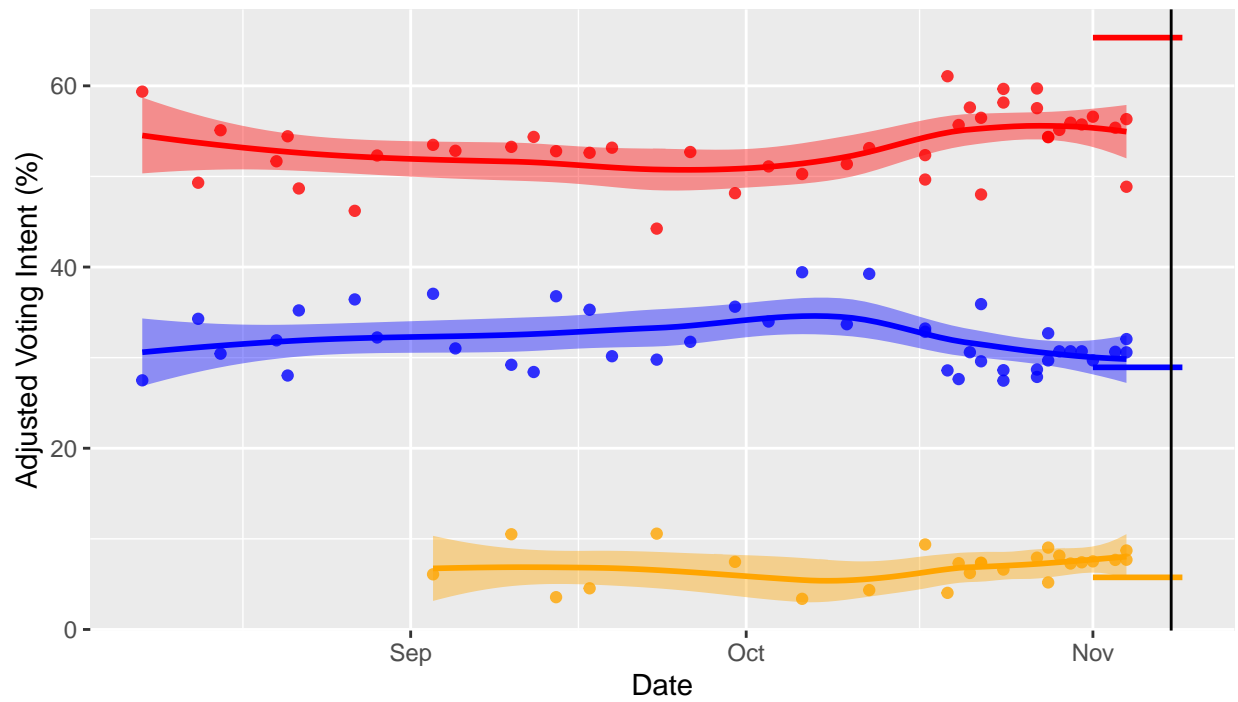
Ohio – Adjusted Polls



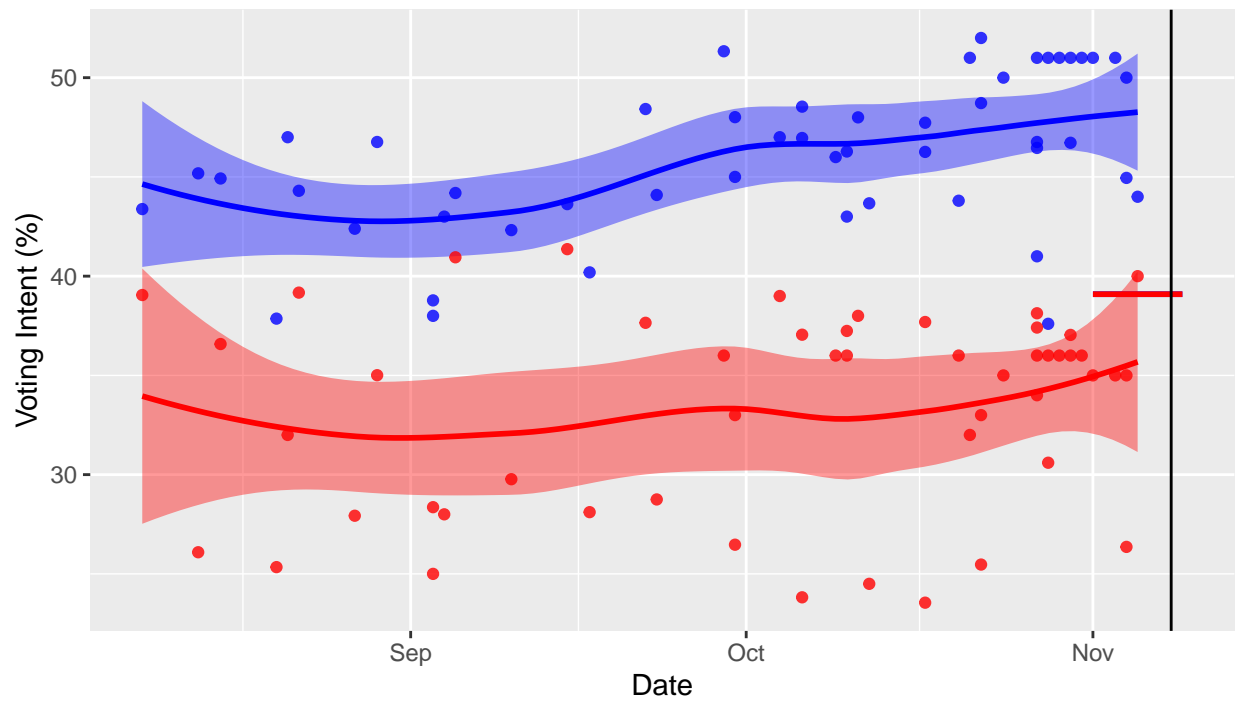
Oklahoma – Raw Polls



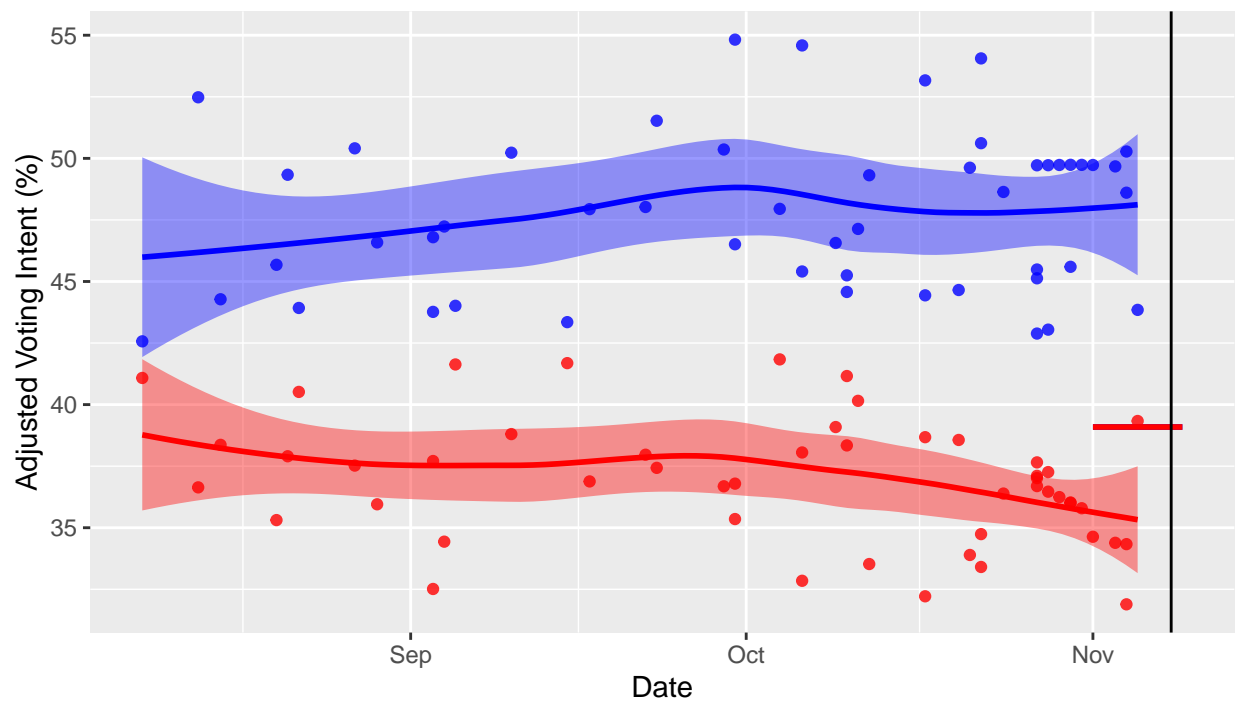
Oklahoma – Adjusted Polls



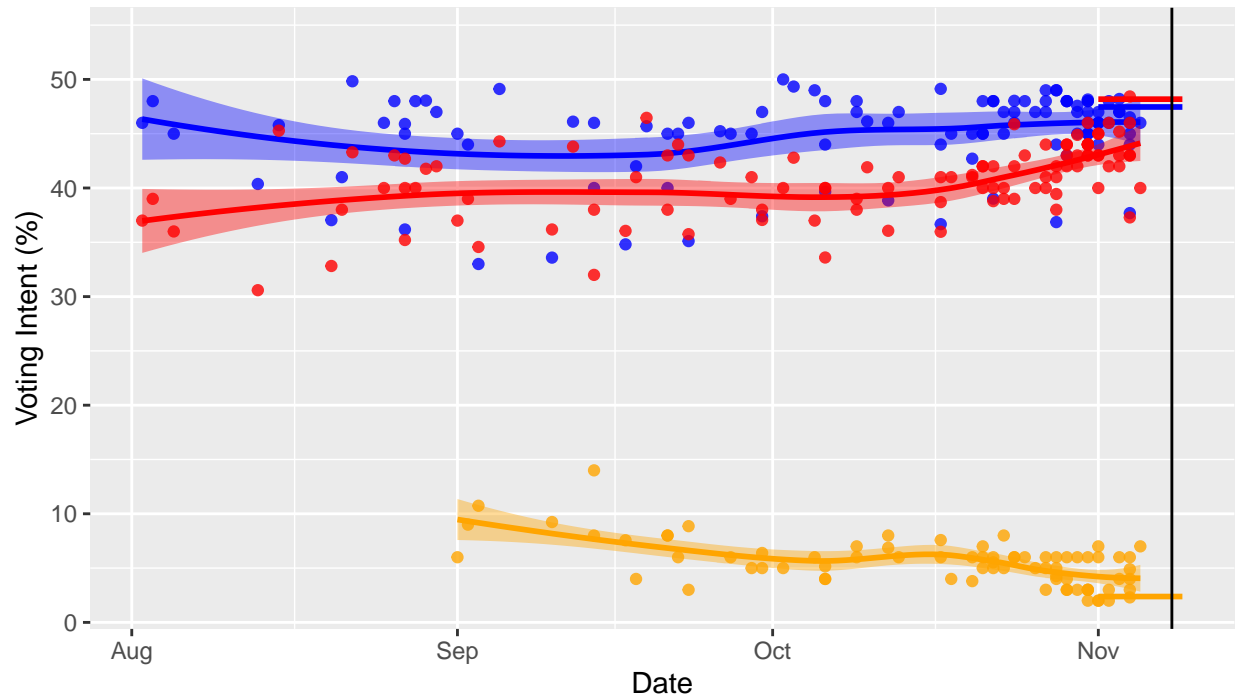
Oregon – Raw Polls



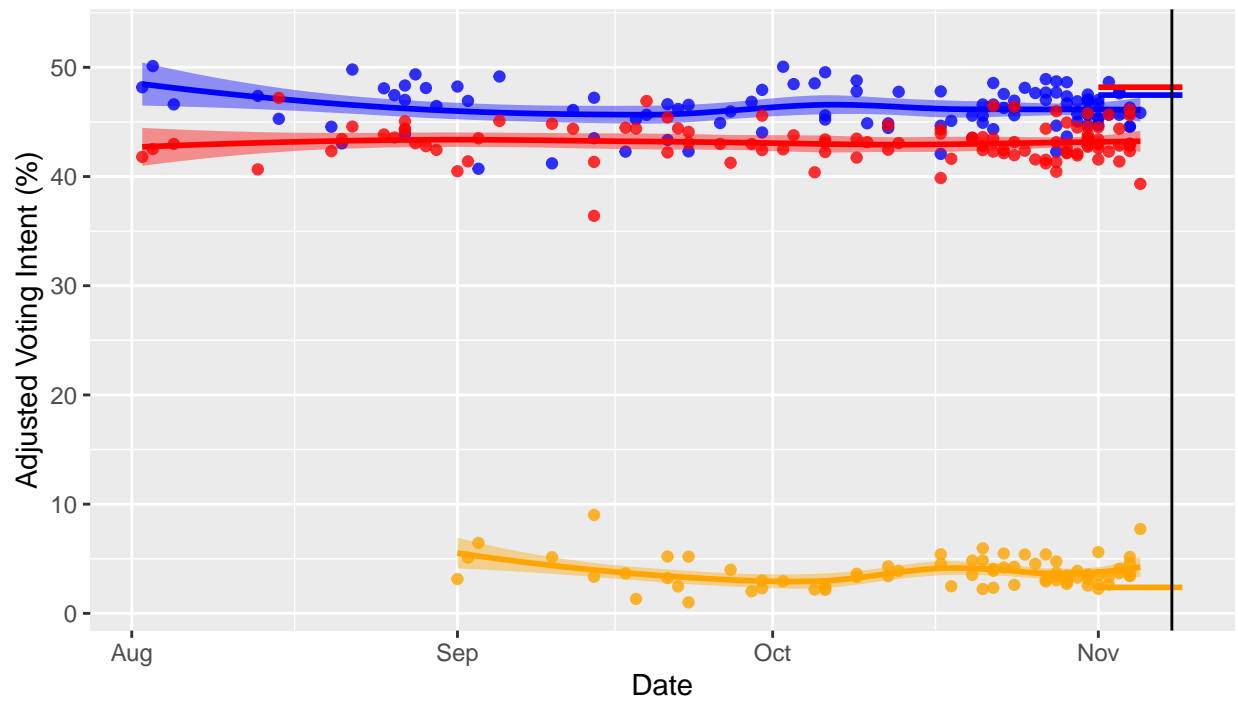
Oregon – Adjusted Polls



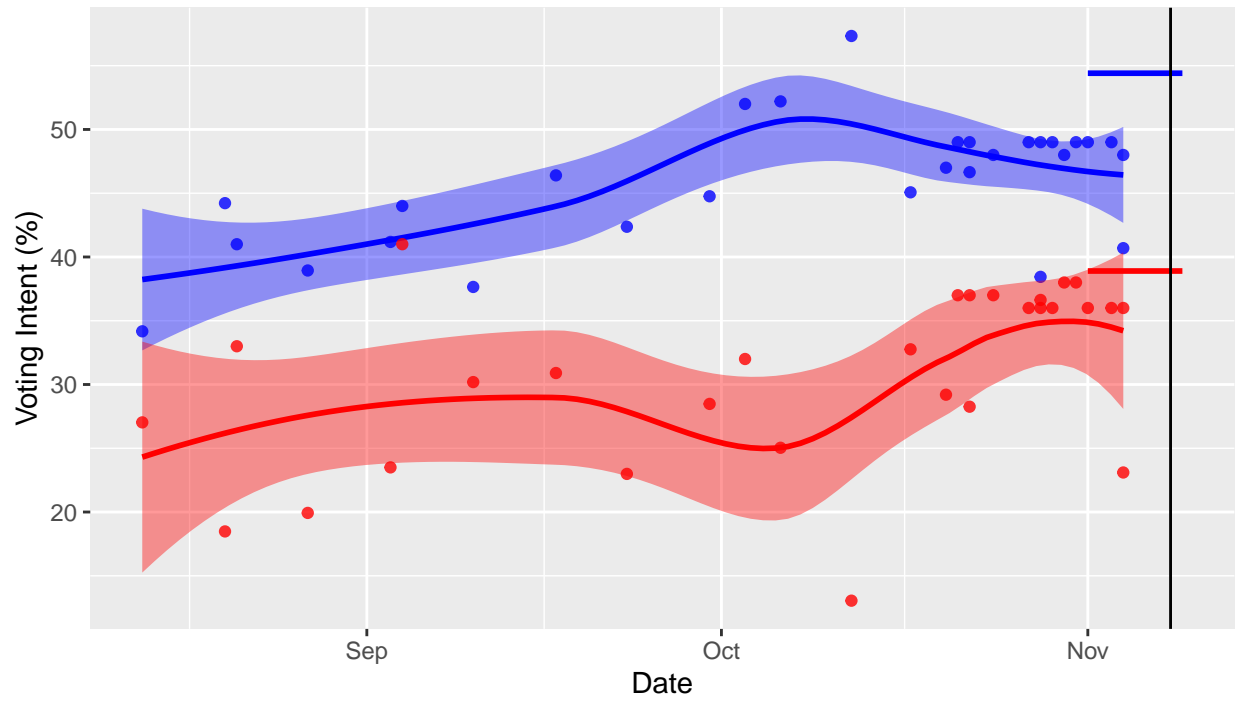
Pennsylvania – Raw Polls



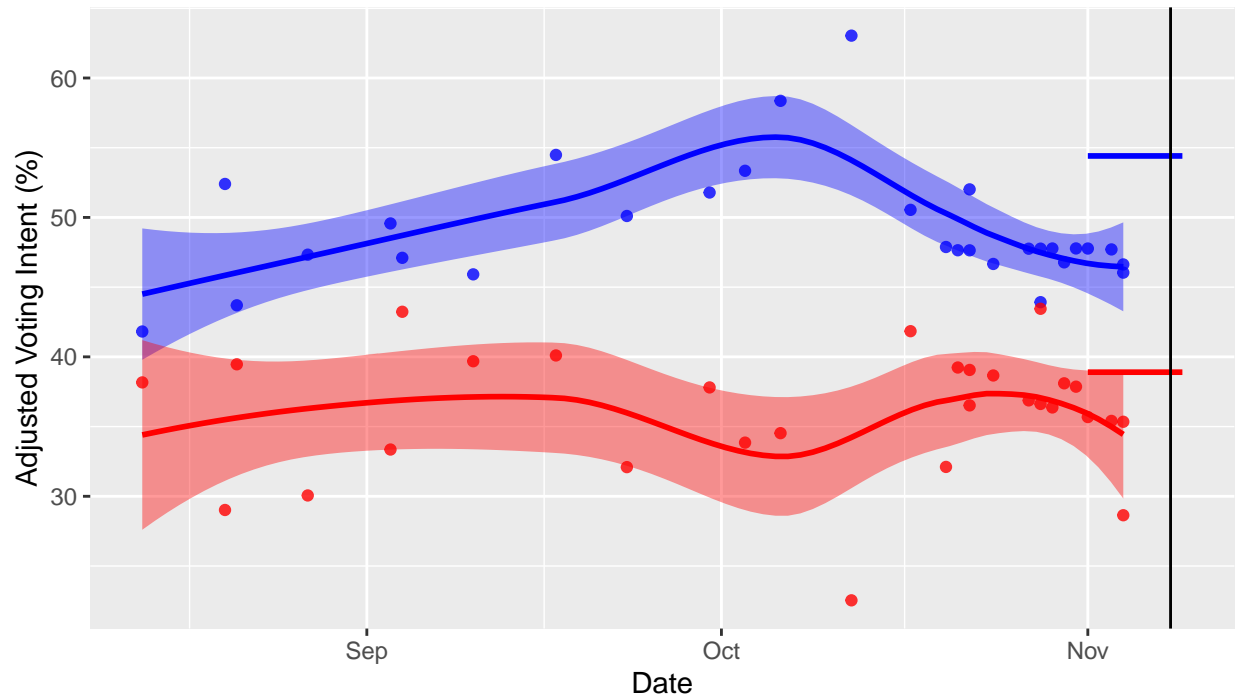
Pennsylvania – Adjusted Polls



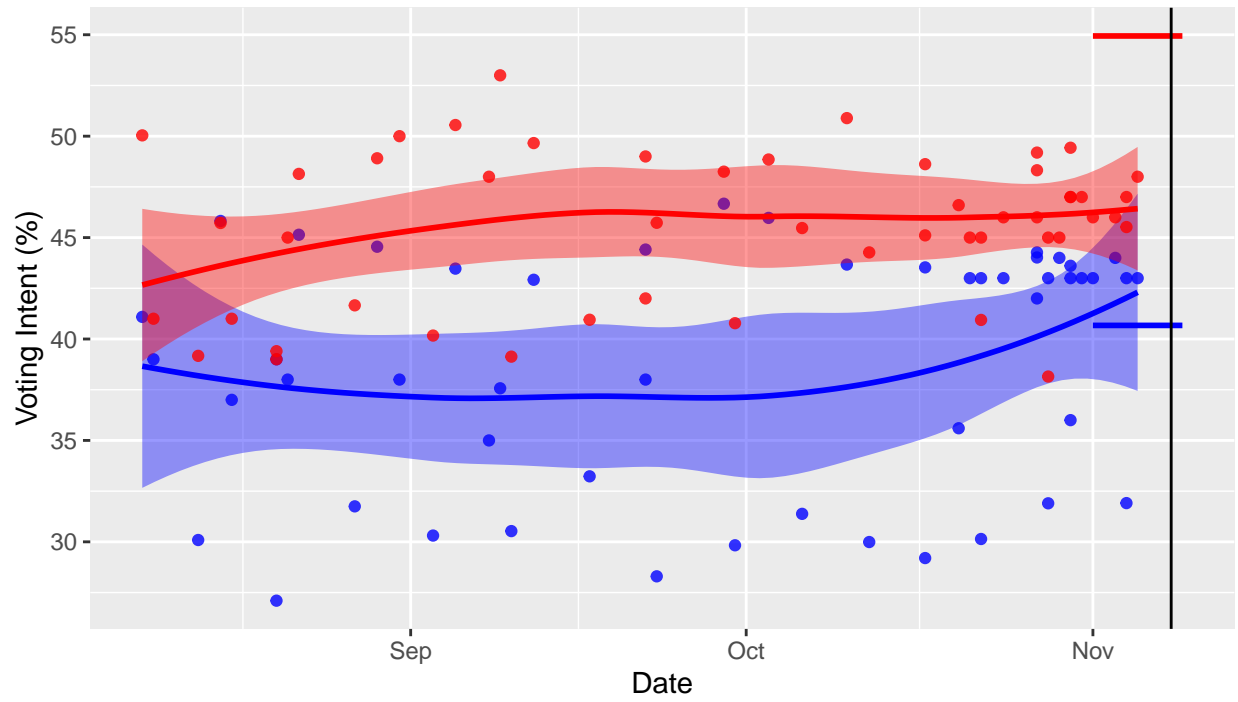
Rhode Island – Raw Polls



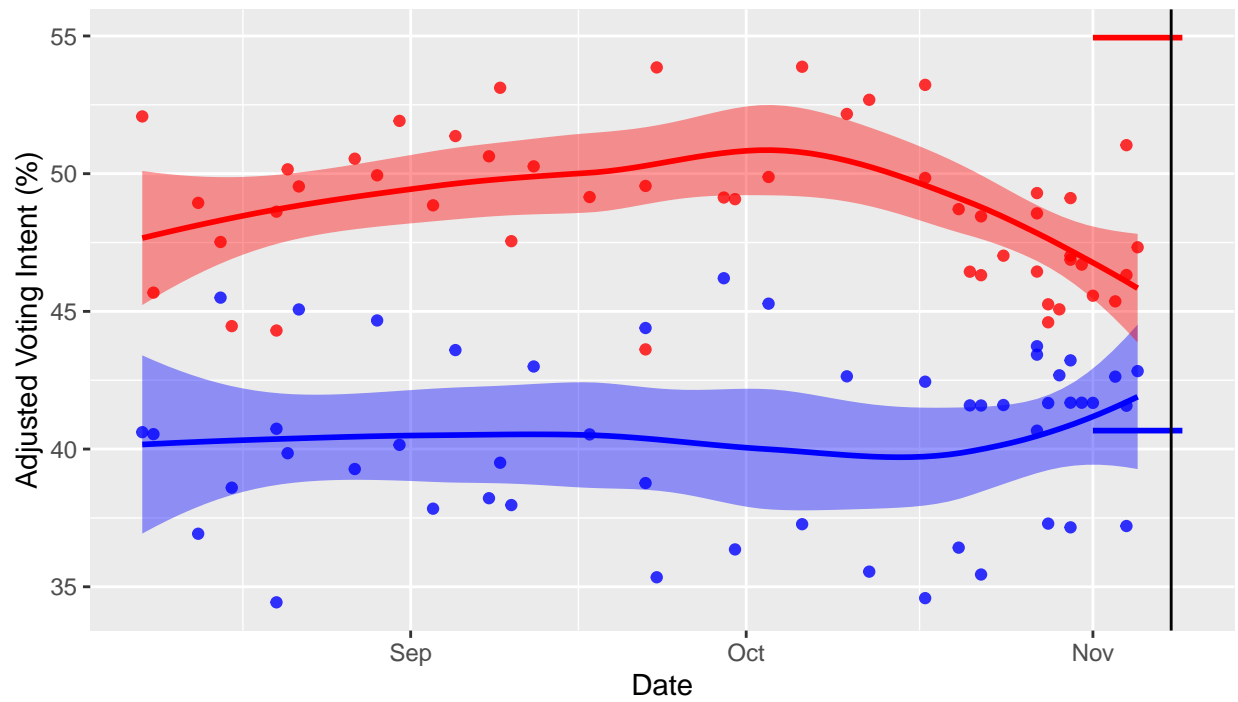
Rhode Island – Adjusted Polls



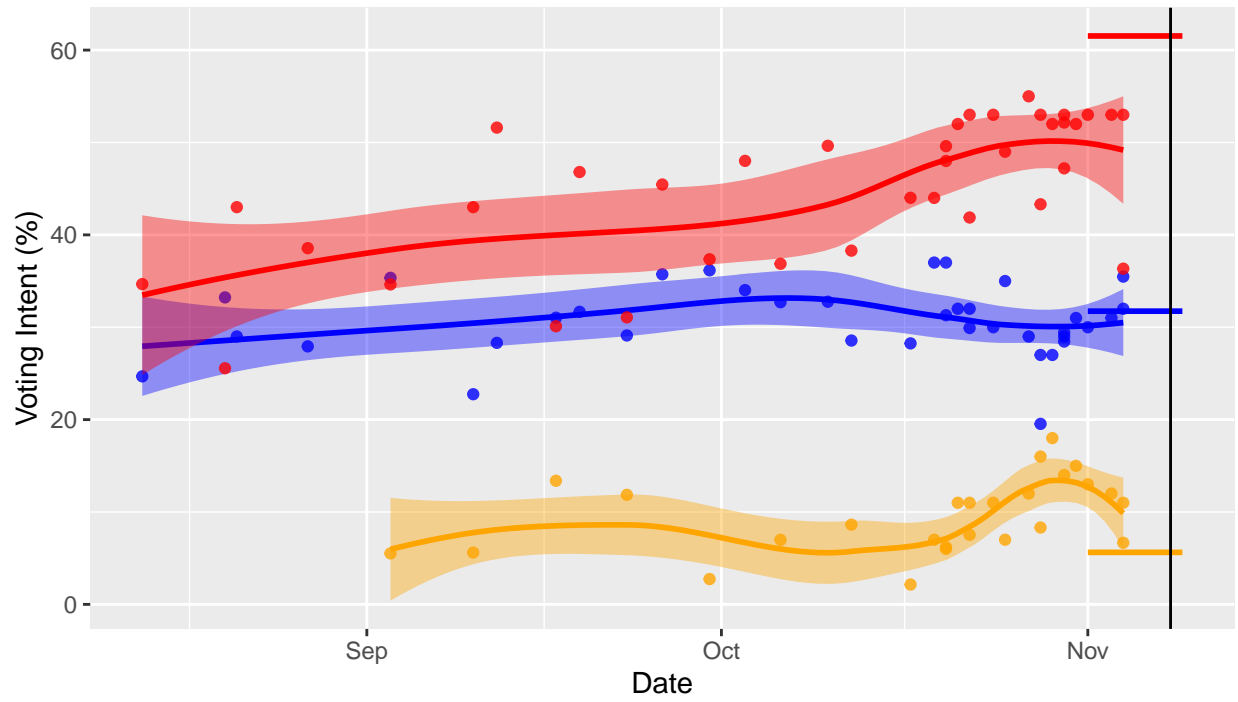
South Carolina – Raw Polls



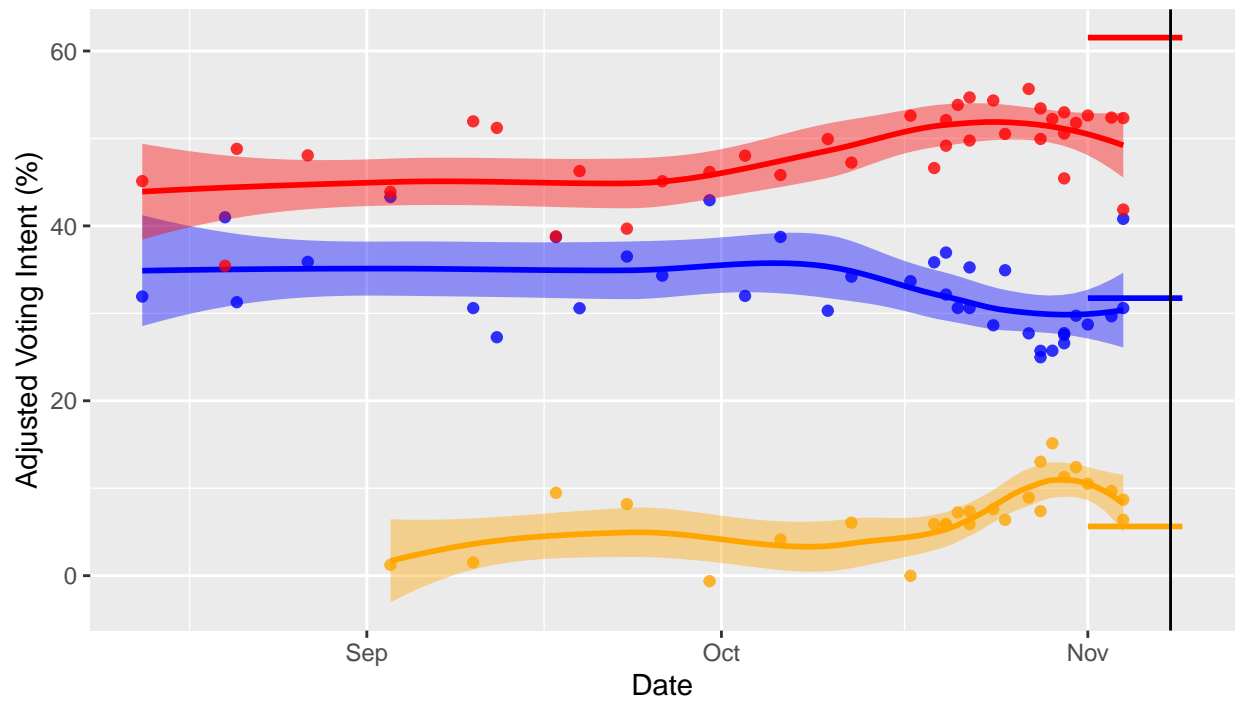
South Carolina – Adjusted Polls



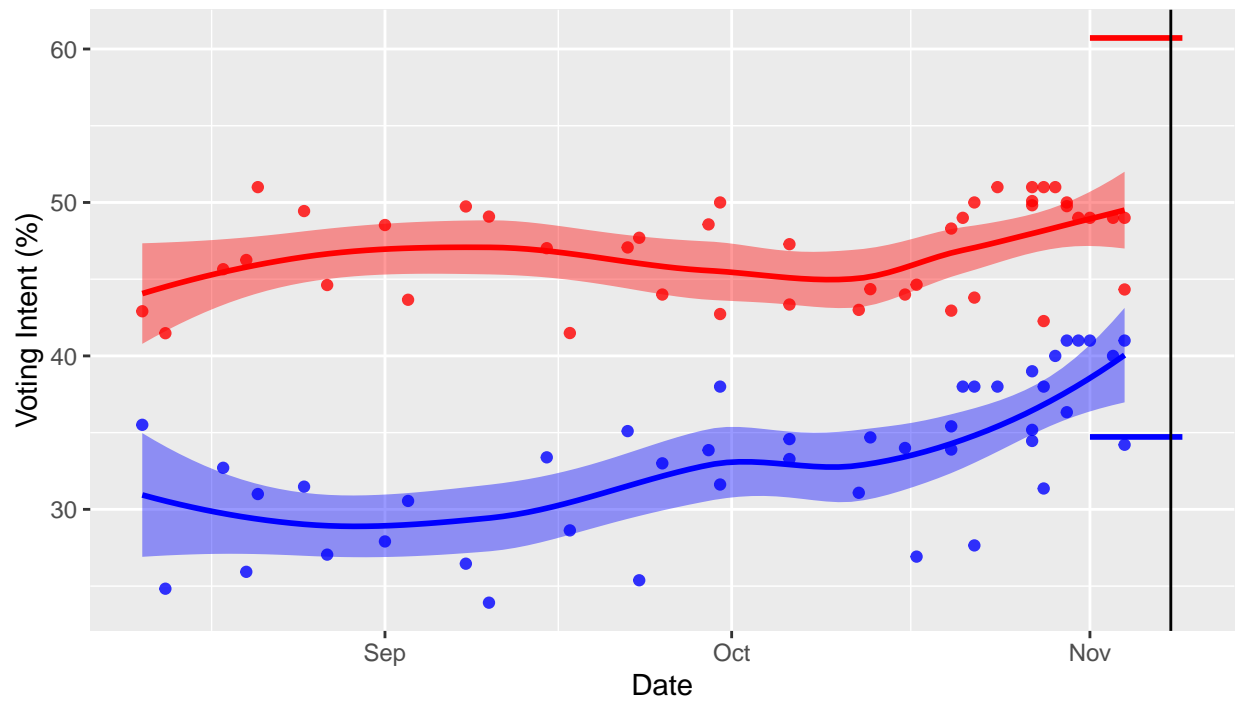
South Dakota – Raw Polls



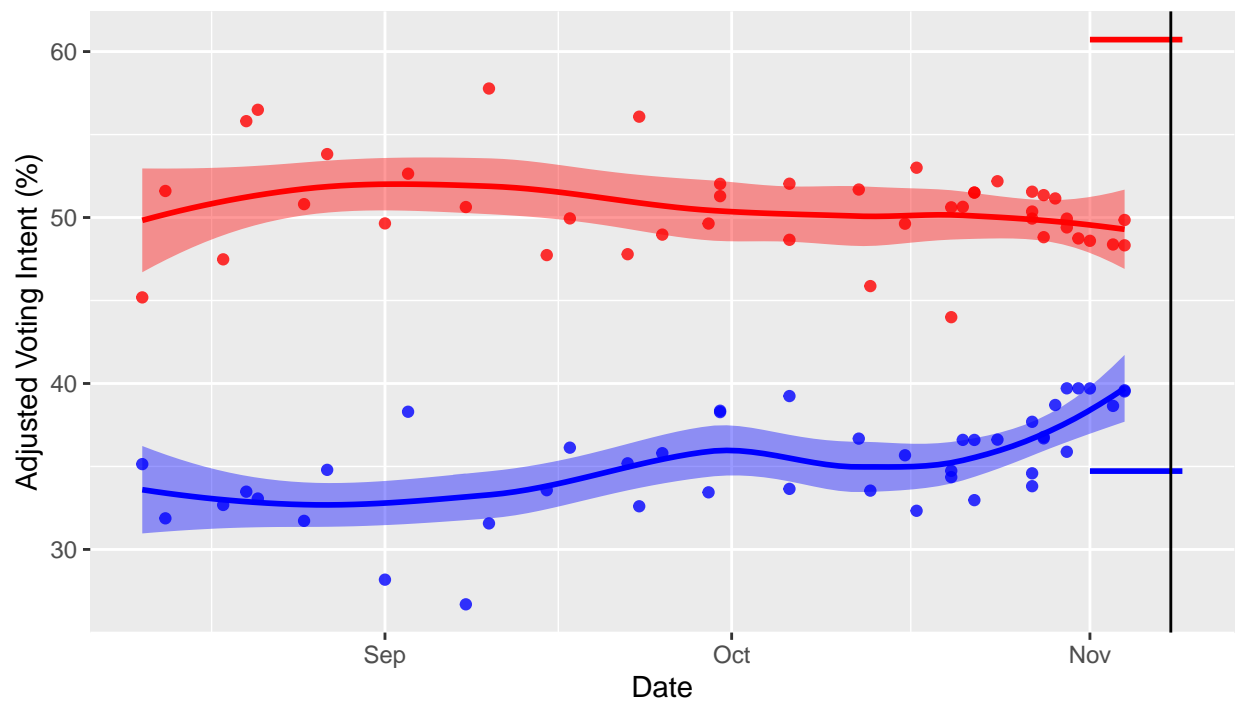
South Dakota – Adjusted Polls



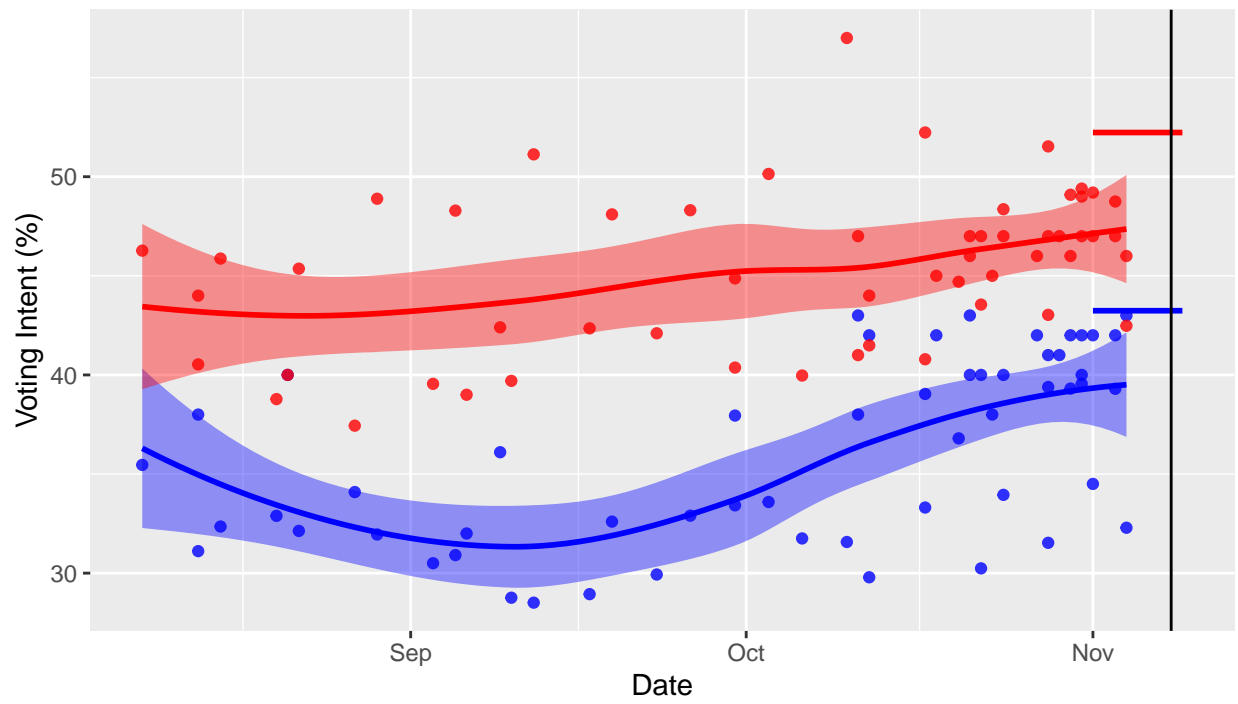
Tennessee – Raw Polls



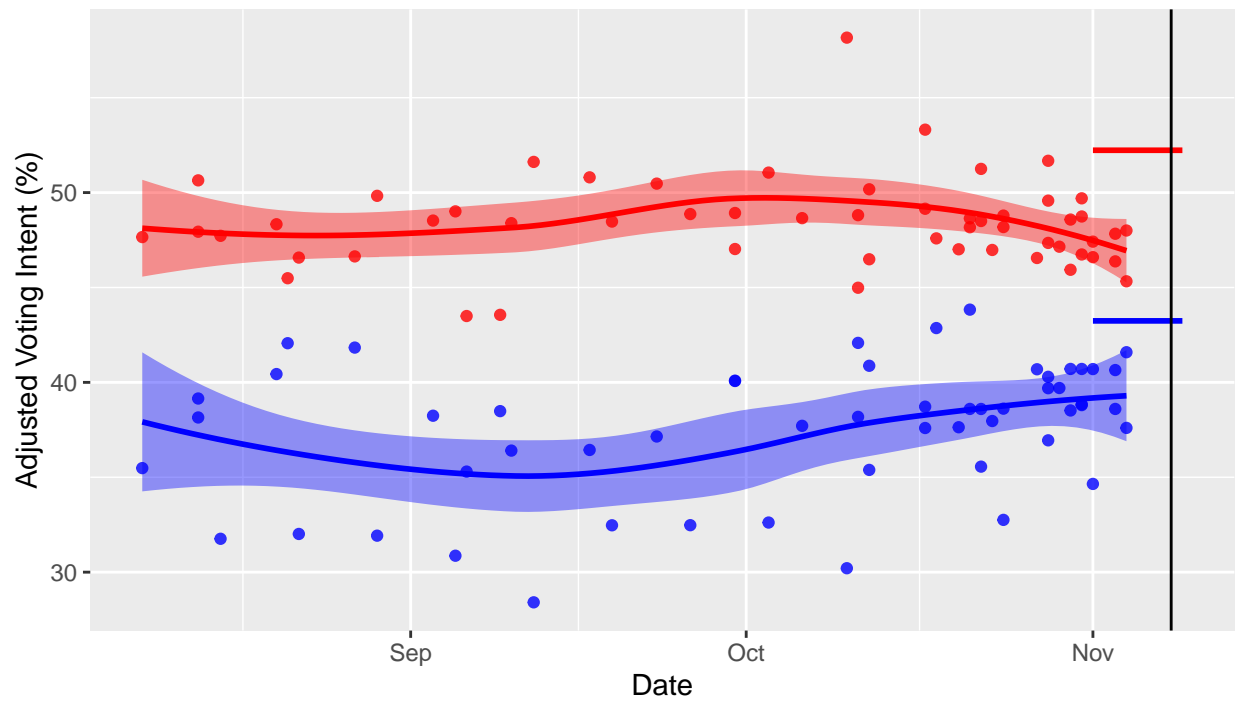
Tennessee – Adjusted Polls



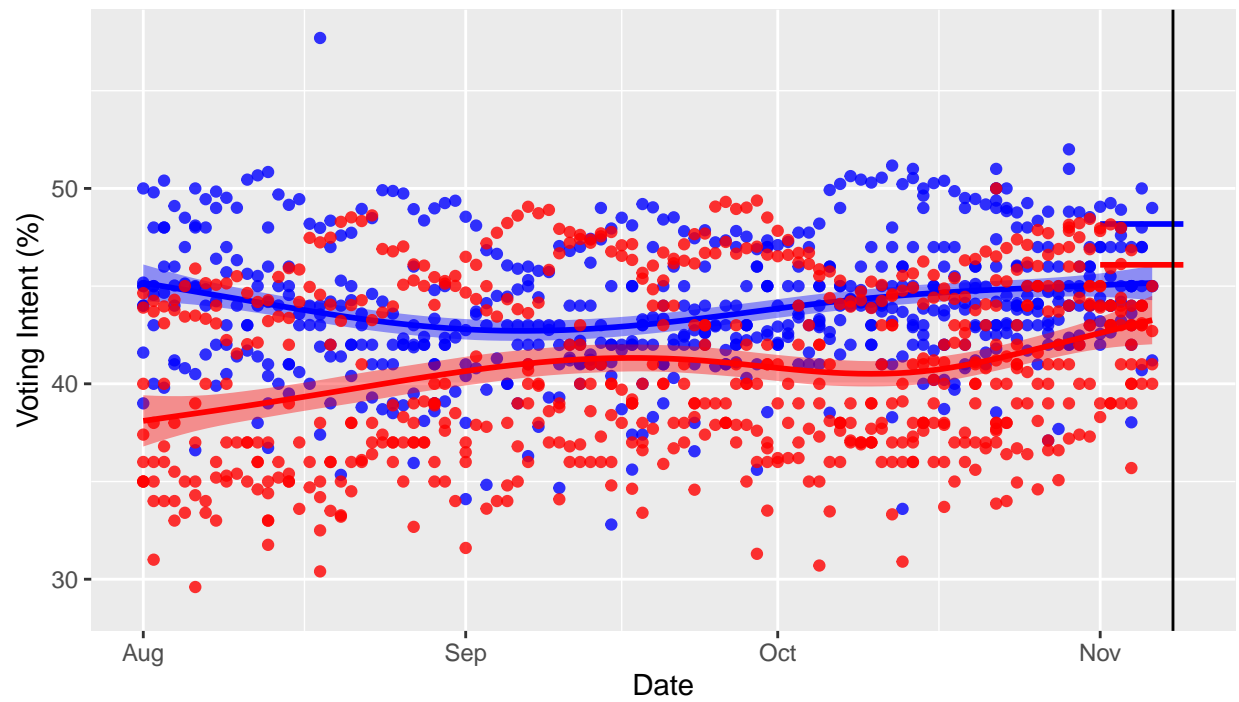
Texas – Raw Polls



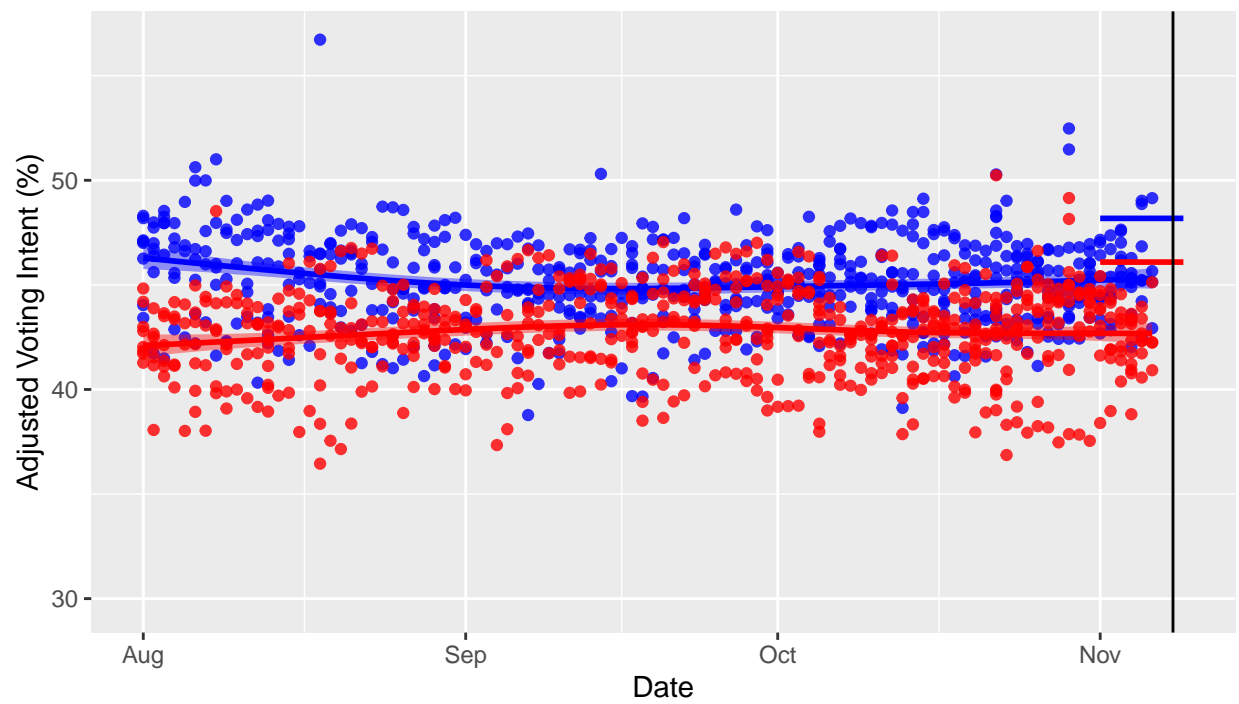
Texas – Adjusted Polls



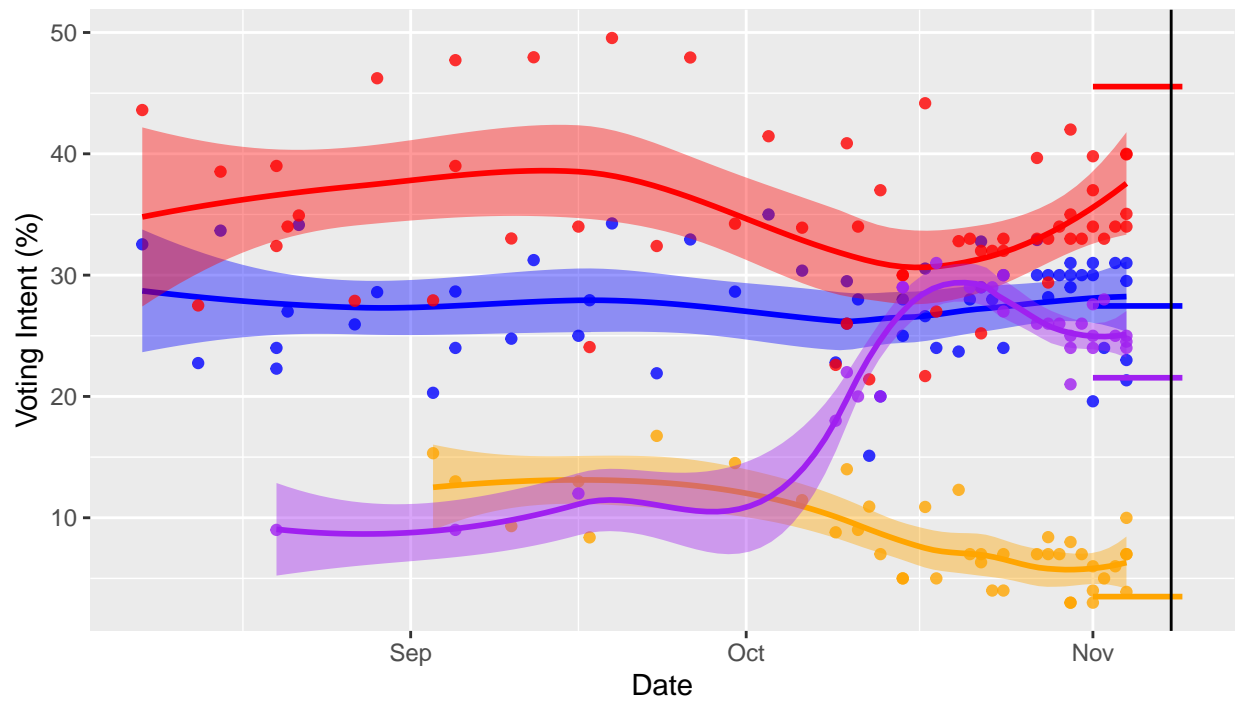
U.S. – Raw Polls



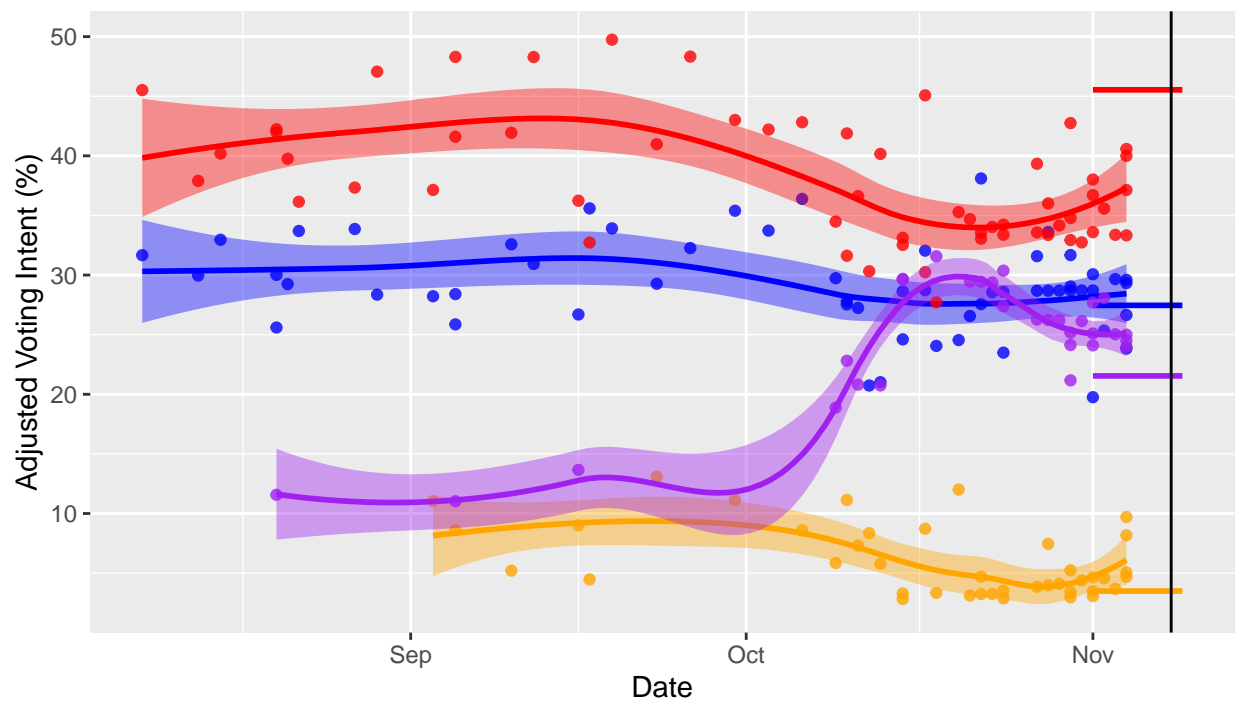
U.S. – Adjusted Polls



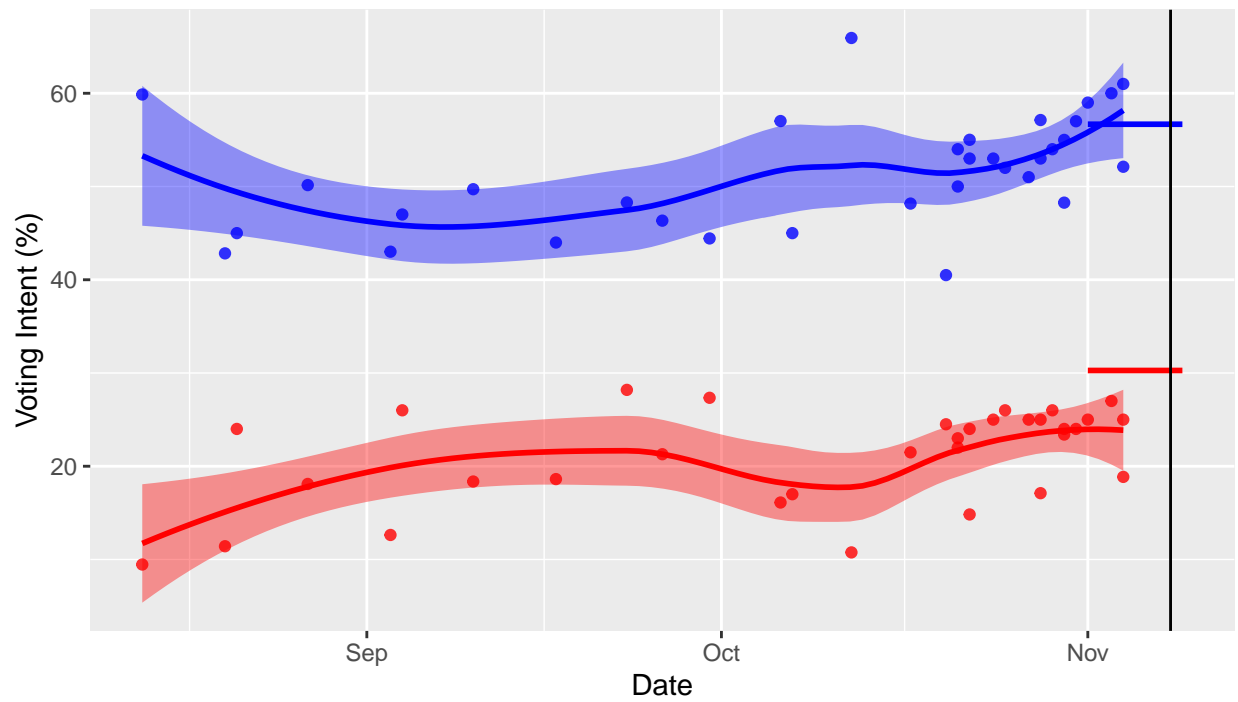
Utah – Raw Polls



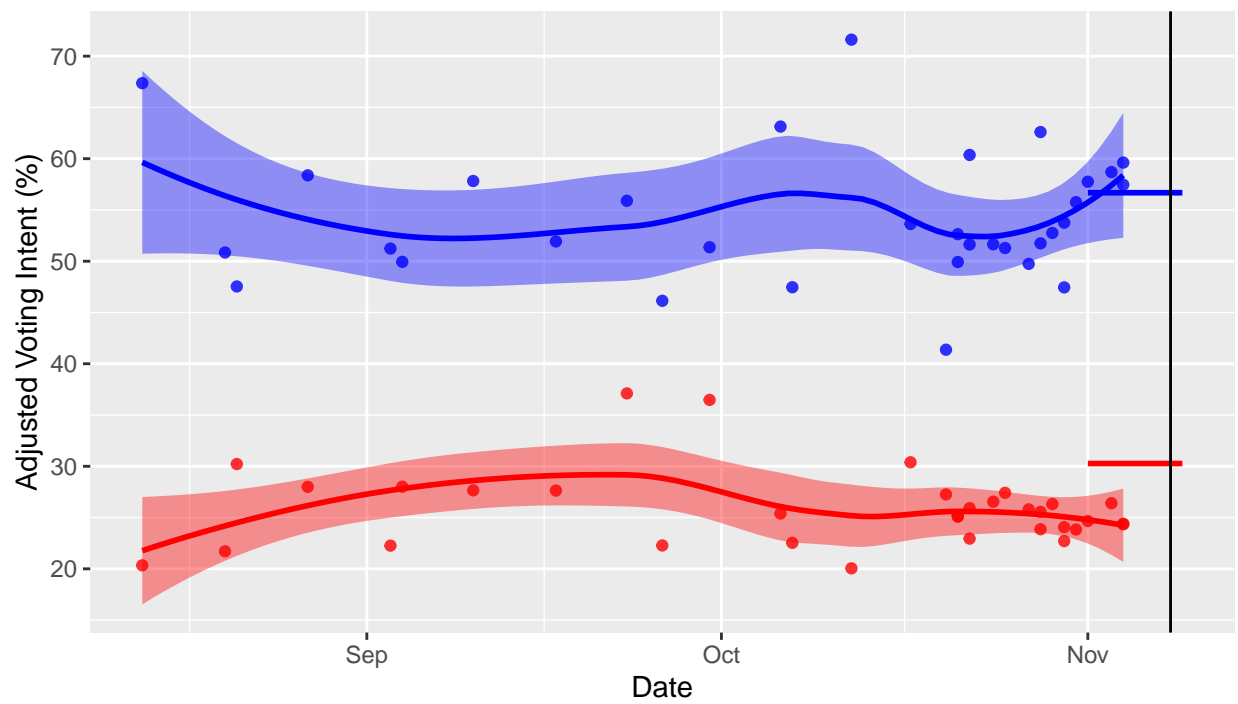
Utah – Adjusted Polls



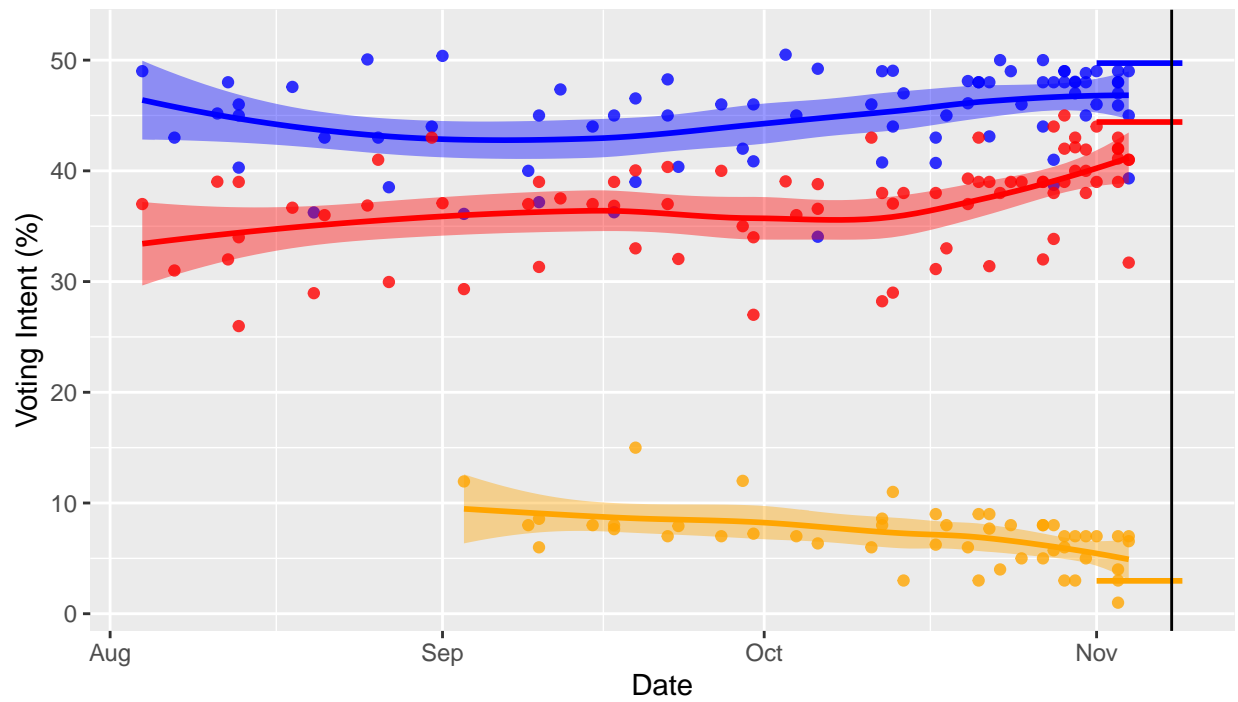
Vermont – Raw Polls



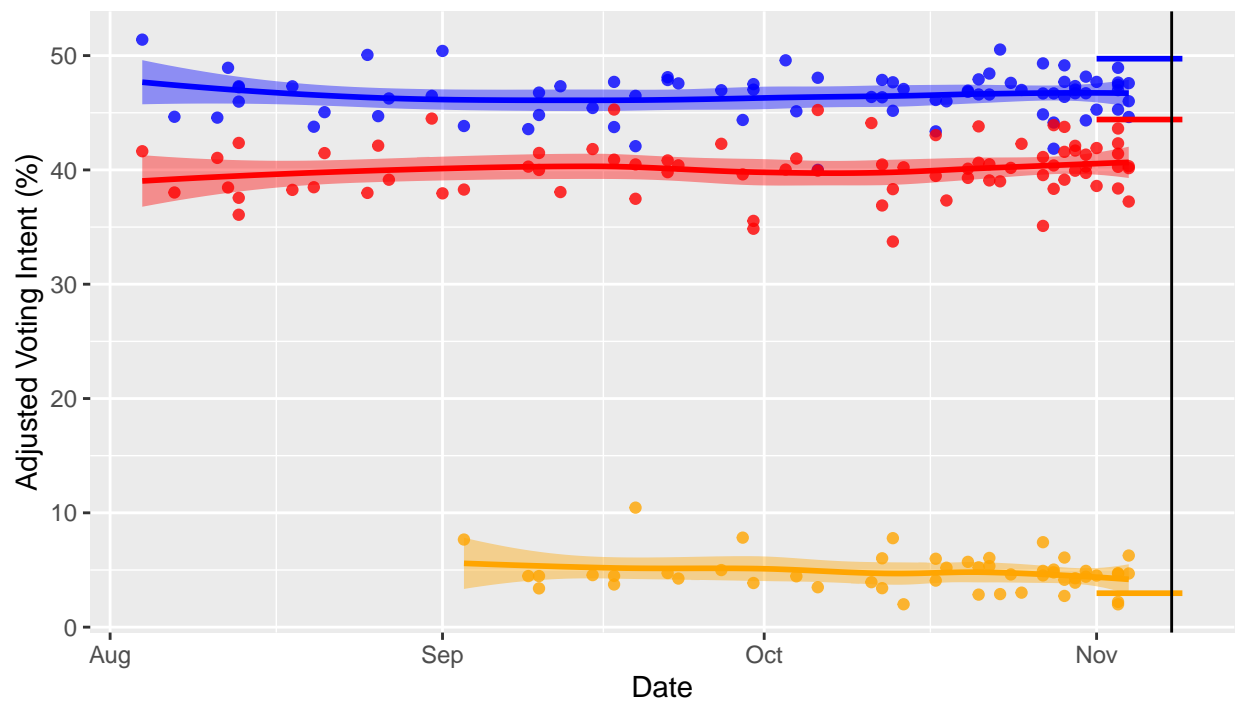
Vermont – Adjusted Polls



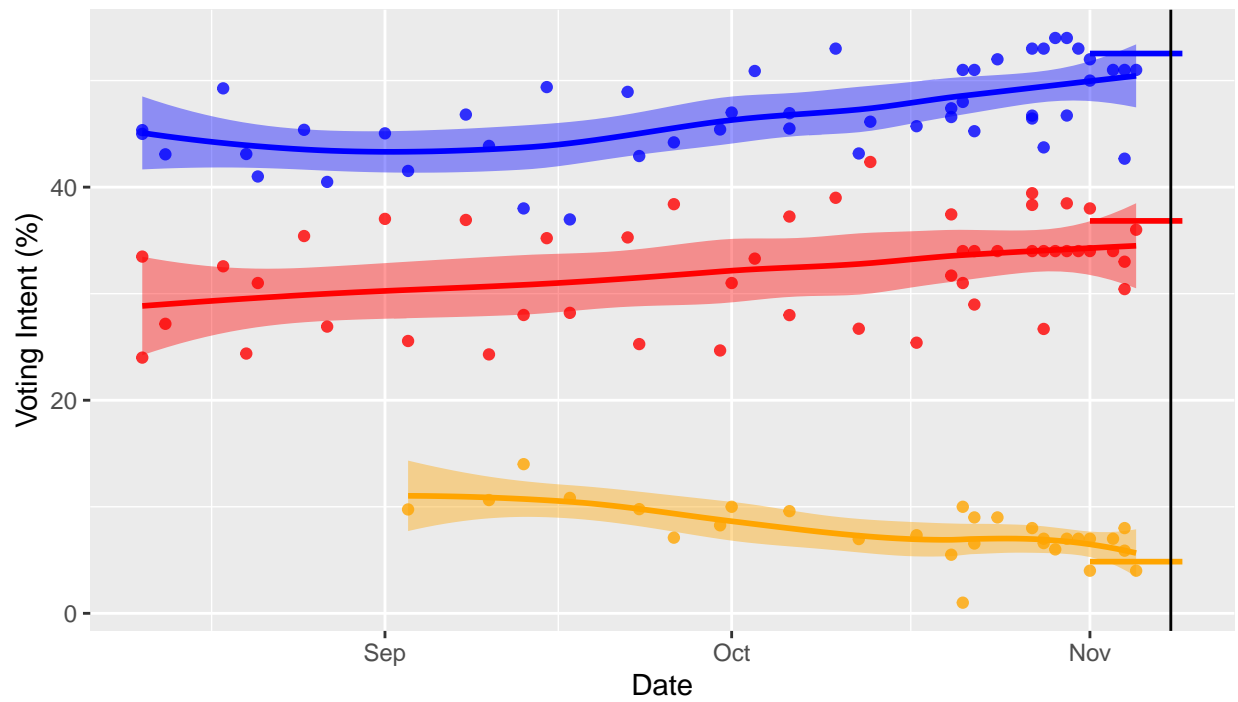
Virginia – Raw Polls



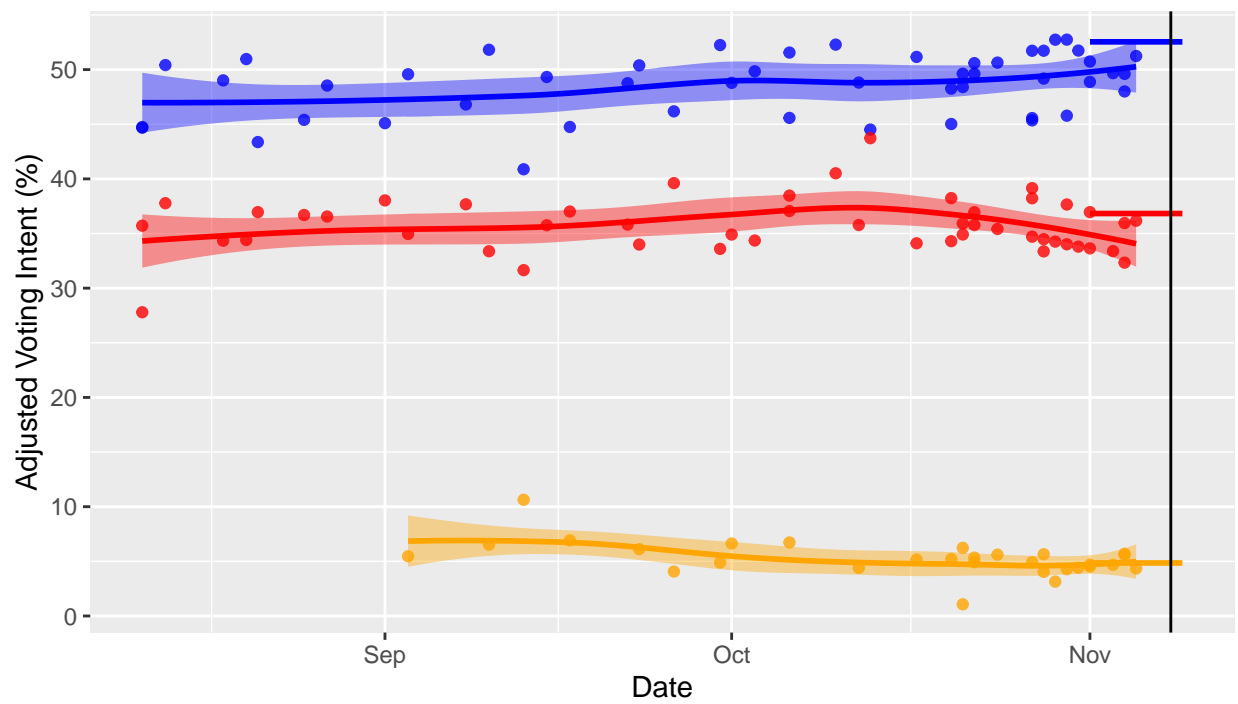
Virginia – Adjusted Polls



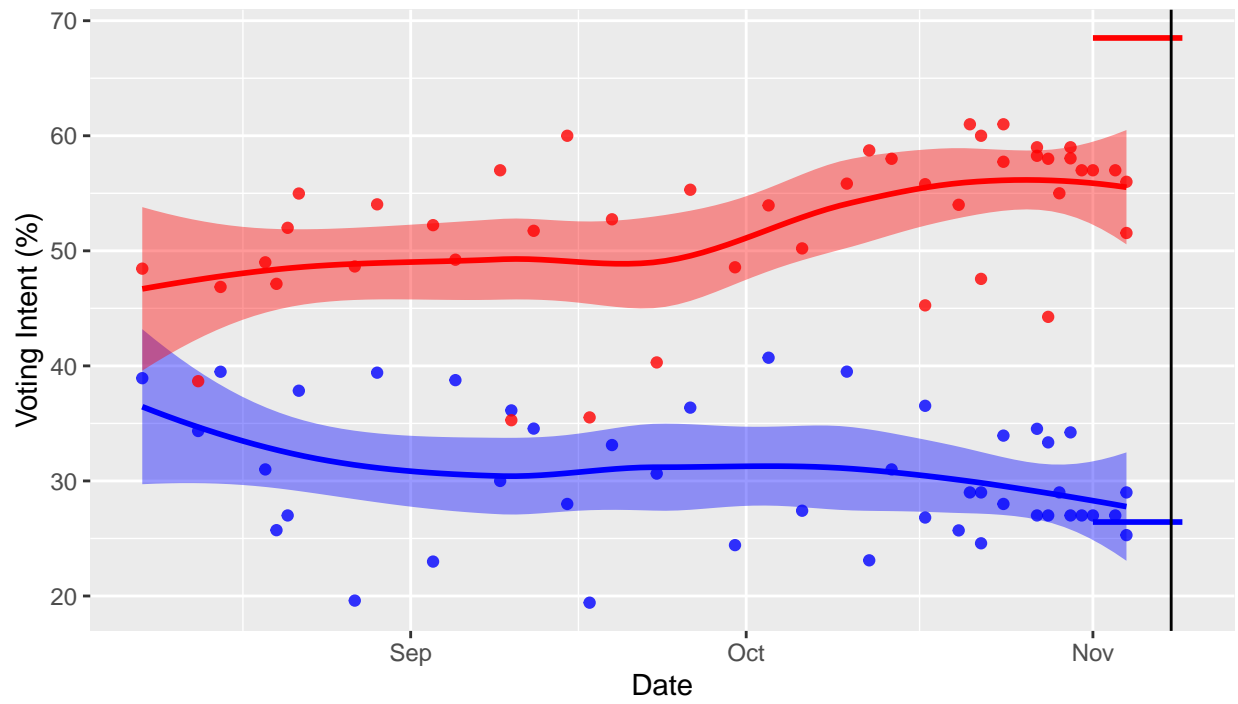
Washington – Raw Polls



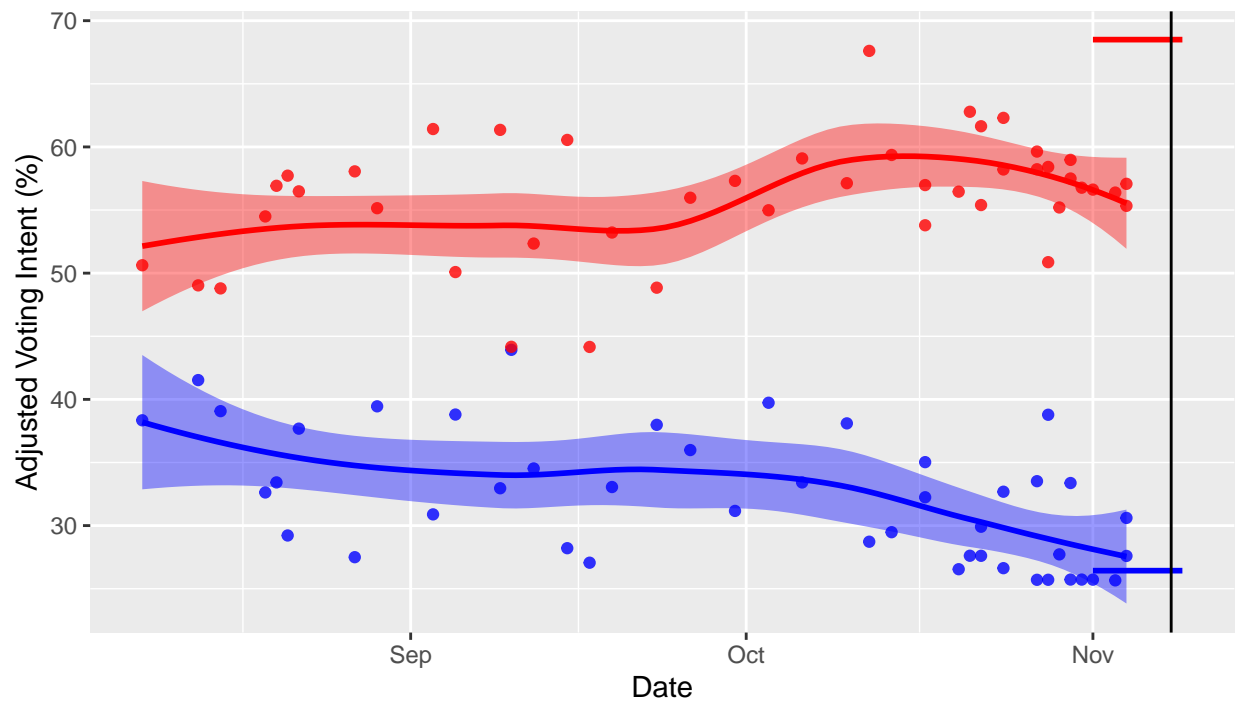
Washington – Adjusted Polls



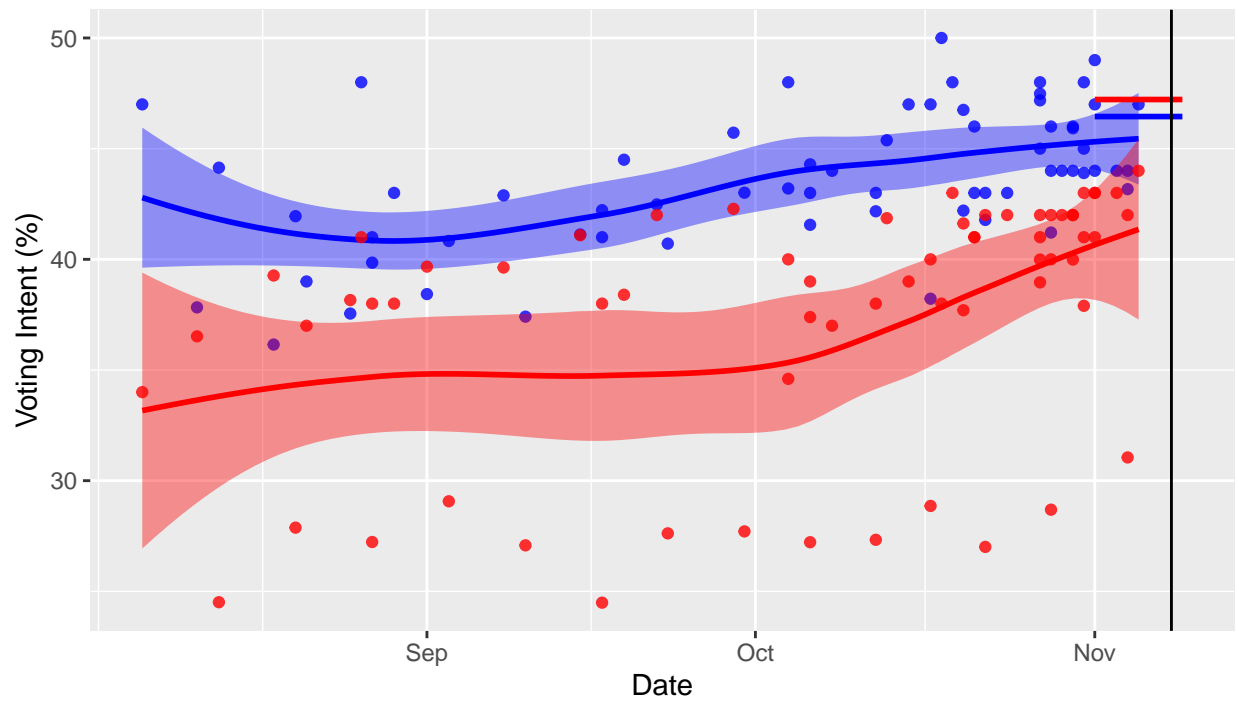
West Virginia – Raw Polls



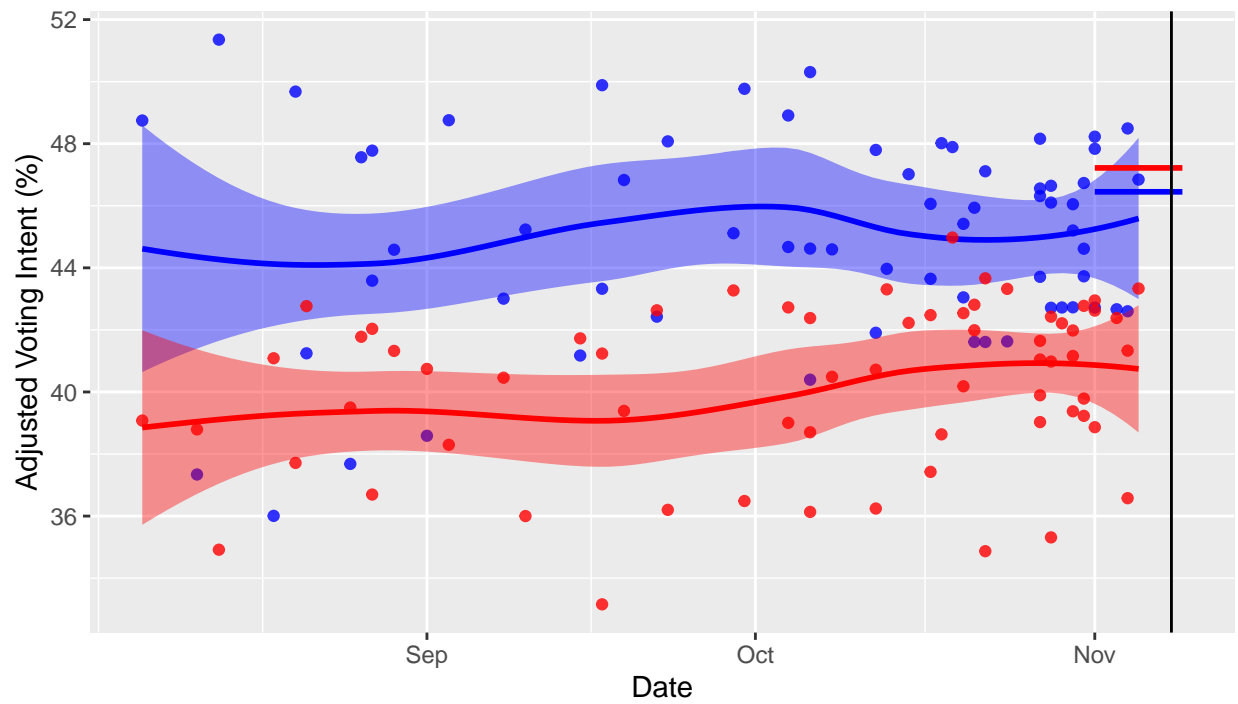
West Virginia – Adjusted Polls



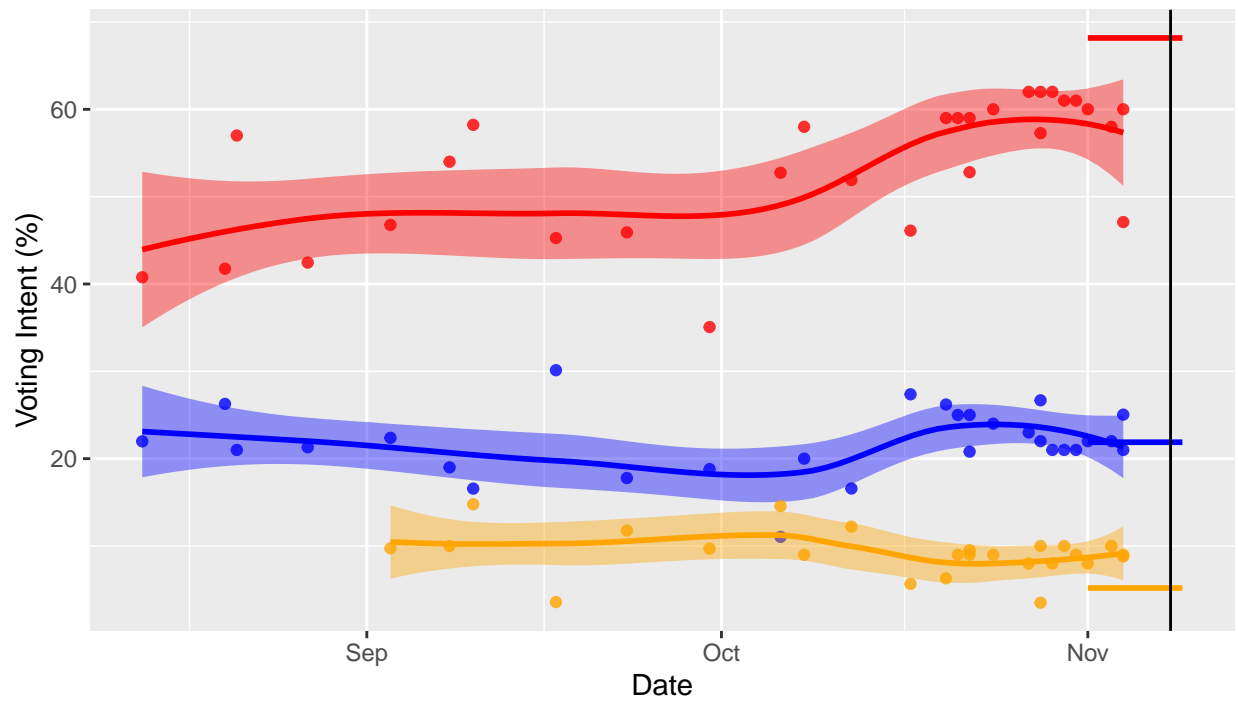
Wisconsin – Raw Polls



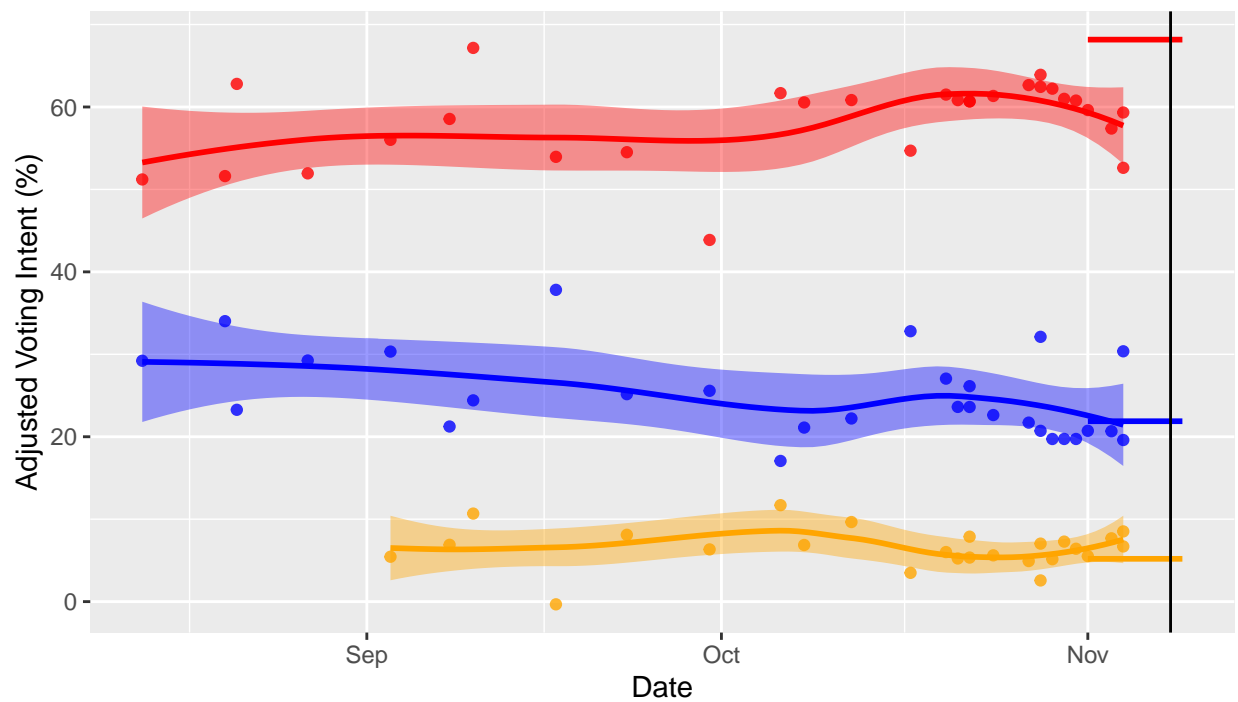
Wisconsin – Adjusted Polls



Wyoming – Raw Polls



Wyoming – Adjusted Polls



Maine has two congressional districts; as such, from the data of one congressional district and of the state as a whole, we can expect to reconstruct the data for the other congressional district (as long as the poll dates are relatively close together).

```

yaxes = c(0, 30)

for (cand in c(clinton, trump, johnson)) { # get common y-limits to use on axes
  for (dist in c("Maine", "Maine CD-1", "Maine CD-2")){
    voterresult = filter(cand$finalresults, state == dist)[1,2] |>
      as.numeric()

    candpolls = cand$polls |>
      filter(state == dist)

    datevsraw = aes(
      x = candpolls$middatetime,
      y = candpolls$rawpolls
    )
    plot = ggplot() + geom_point(
      mapping = datevsraw,
      colour = cand$colour,
      alpha = 0.8,
      na.rm = TRUE
    ) + geom_smooth(
      mapping = datevsraw,
      colour = cand$colour,
      fill = cand$colour,
      alpha = 0.4,
      na.rm = TRUE
    ) + geom_segment(
      mapping = aes(
        x = ymd("2016-11-01"),
        y = voterresult,
        xend = dayafter,
        yend = voterresult
      ),
      colour = cand$colour,
      linewidth = 1.0
    )

    yaxes = c(min(yaxes[1], layer_scales(plot)$y$range$range[1]), max(yaxes[2], layer_scales(plot)$y$range$range[2]))
  }
}

```

```

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

```

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

```
for (state in c("Maine", "Maine CD-1", "Maine CD-2")){
  plot = stateplotraw(state, begindate)

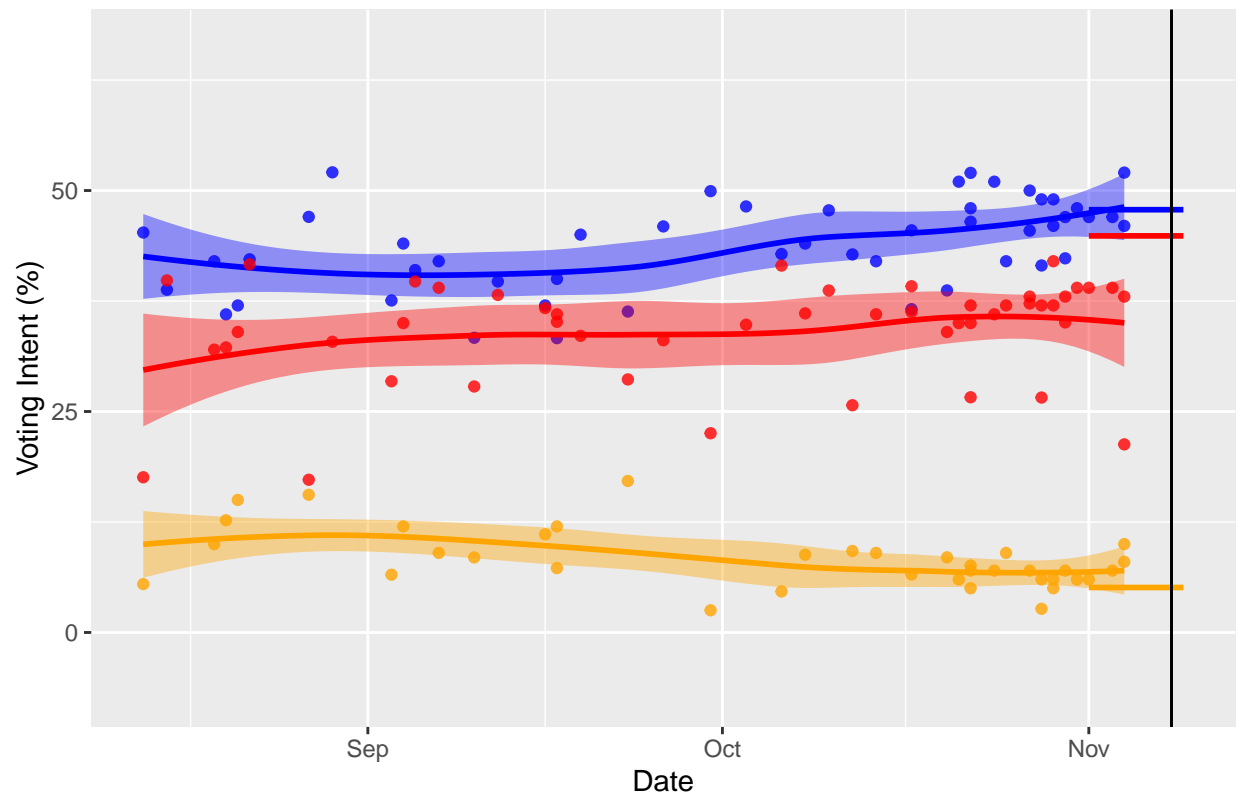
  stateraw = state |>
    paste("Raw Polls", sep = " - ")

  plot = plot + geom_vline(xintercept = finaldate) + labs(
    title = stateraw,
    x = "Date",
    y = "Voting Intent (%)"
  ) + xlim(begindate, dayafter) + ylim(yaxes+c(0,2))

  print(plot)
}
```

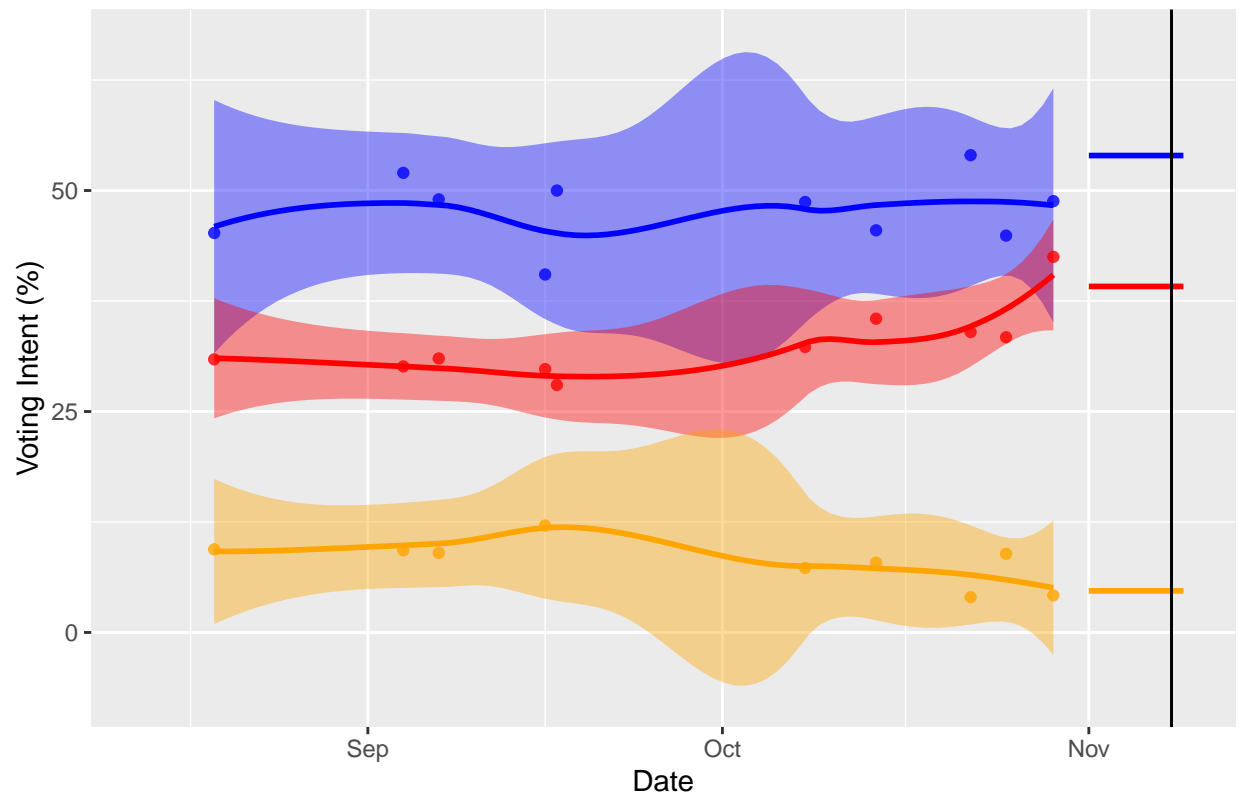
```
## Scale for x is already present.
## Adding another scale for x, which will replace the existing scale.
## 'geom_smooth()' using method = 'loess' and formula = 'y ~ x'
## 'geom_smooth()' using method = 'loess' and formula = 'y ~ x'
## 'geom_smooth()' using method = 'loess' and formula = 'y ~ x'
## Scale for x is already present.
## Adding another scale for x, which will replace the existing scale.
```

Maine – Raw Polls



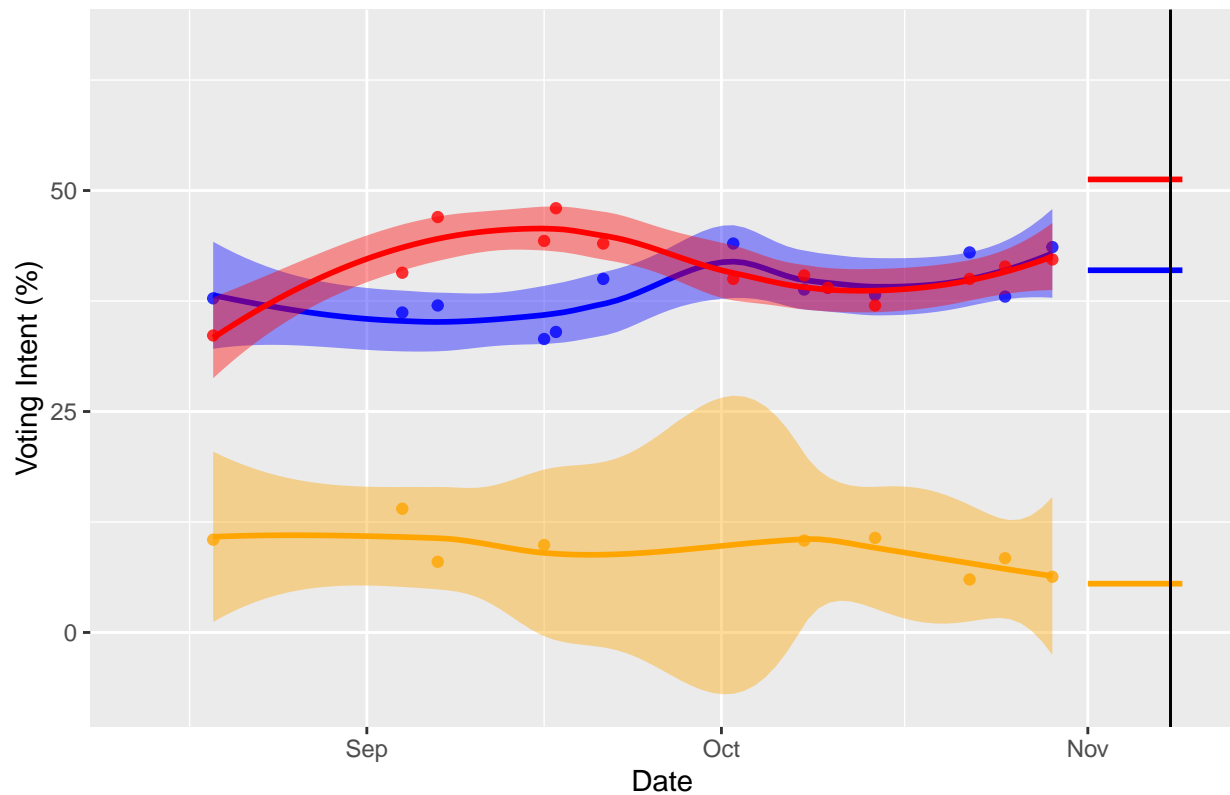
```
## 'geom_smooth()' using method = 'loess' and formula = 'y ~ x'  
## 'geom_smooth()' using method = 'loess' and formula = 'y ~ x'  
## 'geom_smooth()' using method = 'loess' and formula = 'y ~ x'  
## Scale for x is already present.  
## Adding another scale for x, which will replace the existing scale.
```

Maine CD-1 – Raw Polls



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

Maine CD-2 – Raw Polls



Seeing as the polls for the Nebraskan congressional districts are so sparse, it's difficult to supplement the data with data from pan-Nebraskan polls – the CD polls are so few in number that they can't serve to support or refute any supplemental data. Although Clinton and Trump polled comparably in the 2nd Nebraskan congressional district, it makes the most sense (for the sake of electoral simulation) to consider Nebraska's electoral votes as a whole, rather than allocating electoral votes by district.

If we were to eliminate some of the polls from lower-graded pollsters, in some states we would run the risk of only keeping polls from one or two sources. Given that each pollster has their own bias, there is reason to avoid overly relying on a single source for polls.

We supplement Maine congressional district data with Maine polls from higher-rated pollsters.

```
suppstates = c("District of Columbia", "Maine", "Maine CD-1", "Maine CD-2", "North Dakota", "Rhode Island")

gpolls = polls |>
  filter(state %in% suppstates) |>
  mutate(midddate = round_date(midddate, unit = "week"))
mainepolls = gpolls |>
  filter(state == "Maine", grade %in% c("B", "B+", "A-", "A", "A+")) |>
  select(midddate, rawpoll_clinton, rawpoll_trump)
d1polls = gpolls |>
  filter(state == "Maine CD-1")
d1polls2 = gpolls |>
  filter(state == "Maine CD-1") |>
  mutate(rawpoll_clinton_dist1 = rawpoll_clinton, rawpoll_trump_dist1 = rawpoll_trump) |>
  select(midddate, rawpoll_clinton_dist1, rawpoll_trump_dist1)
```

```
x = merge(mainepolls, d1polls2) |>
  as_tibble()
regc = lm(rawpoll_clinton_dist1 ~ rawpoll_clinton, data = x)
regc
```

```
##
## Call:
## lm(formula = rawpoll_clinton_dist1 ~ rawpoll_clinton, data = x)
##
## Coefficients:
##      (Intercept)  rawpoll_clinton
##           41.6753           0.1558
```

```
anova(regc) # pretty lousy. May be due to repeated X-values with different Y-values;
```

```
## Analysis of Variance Table
##
## Response: rawpoll_clinton_dist1
##              Df Sum Sq Mean Sq F value Pr(>F)
## rawpoll_clinton  1  12.17   12.171   0.9035 0.3494
## Residuals       30 404.13   13.471
```

```
# low number of datapoints
regt = lm(rawpoll_trump_dist1 ~ rawpoll_trump, data = x)
regt
```

```
##
## Call:
## lm(formula = rawpoll_trump_dist1 ~ rawpoll_trump, data = x)
##
## Coefficients:
##      (Intercept)  rawpoll_trump
##          33.177205          -0.008009
```

```
anova(regt)
```

```
## Analysis of Variance Table
##
## Response: rawpoll_trump_dist1
##              Df Sum Sq Mean Sq F value Pr(>F)
## rawpoll_trump  1   0.04   0.0422   0.0023 0.9618
## Residuals     30 542.25  18.0749
```

```
newdata = mainepolls |>
  mutate(rawpoll_clinton = regc$coefficients["(Intercept)"] + regc$coefficients["rawpoll_clinton"]*rawpoll_clinton,
  mutate(rawpoll_trump = regt$coefficients["(Intercept)"] + regt$coefficients["rawpoll_trump"]*rawpoll_trump,
  bind_rows(d1polls)
newclinton = aes(
  x = newdata$middate,
  y = newdata$rawpoll_clinton
```

```

)

newtrump = aes(
  x = newdata$middate,
  y = newdata$rawpoll_trump
)

clintonresult = filter(clinton$finalresults, state == "Maine CD-1")[1,2] |>
  as.numeric()
trumpresult = filter(trump$finalresults, state == "Maine CD-1")[1,2] |>
  as.numeric()

plot = ggplot() + geom_point(
  mapping = newclinton,
  colour = clinton$colour,
  alpha = 0.8,
  na.rm = TRUE
) + geom_point(
  mapping = newtrump,
  colour = trump$colour,
  alpha = 0.8,
  na.rm = TRUE
) + geom_smooth(
  mapping = newclinton,
  colour = clinton$colour,
  fill = clinton$colour,
  alpha = 0.4,
  na.rm = TRUE
) + geom_segment(
  mapping = aes(
    x = ymd("2016-11-01"),
    y = clintonresult,
    xend = dayafter,
    yend = clintonresult
  ),
  colour = clinton$colour,
  linewidth = 1.0
) + geom_smooth(
  mapping = newtrump,
  colour = trump$colour,
  fill = trump$colour,
  alpha = 0.4,
  na.rm = TRUE
) + geom_segment(
  mapping = aes(
    x = ymd("2016-11-01"),
    y = trumpresult,
    xend = dayafter,
    yend = trumpresult
  ),
  colour = trump$colour,
  linewidth = 1.0
) + geom_vline(xintercept = finaldate) + labs(

```



```

    title = "Maine CD-1 raw polls (supplemented)",
    x = "Date",
    y = "Voting Intent (%)"
  ) + xlim(begindate, dayafter) + ylim(yaxes+c(0,2))
print(plot)

```

```

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

```

