The Florida Primary 2012

J.A.Hartigan Yale University
September 2013

1 Presidential Primaries

In the United States, the Democrats and the Republicans nominate their candidates for President at national conventions. Some of the delegates at the conventions are party officials and former office holders; the majority are delegates from the different states pledged to vote according to the results of primaries in those states. A number of states have “non-binding “ caucuses , and their delegates may vote how they please at the convention. The primaries occur between January and June in preparation for the presidential election in November. In 2012, President Obama was chosen to be the Democratic candidate without serious opposing candidates, but the Republican primaries were hard fought.

The early Republican primaries:
Iowa caucus, 1 January, won narrowly by Rick Santorum over Mitt Romney ;
New Hampshire, 10 January, won convincingly by Mitt Romney over Newt Gingrich,
South Carolina, 21 January, won convincingly by Newt Gingrich over Mitt Romney

At this point, Newt Gingrich, who had previously been knocked out of contention in Iowa by attack advertisements from Mitt Romney, was regarded as the front runner. But Mitt Romney again used his advantage in money to saturate the Florida TV market with attacks on Newt, and his advantage increased in the polls as the election date 31 January approached. We will use the polls to predict the difference in percentage of voters favoring Romney and Gingrich on primary day in Florida, 31 January 2012.

2 Romney Gingrich poll differences over time

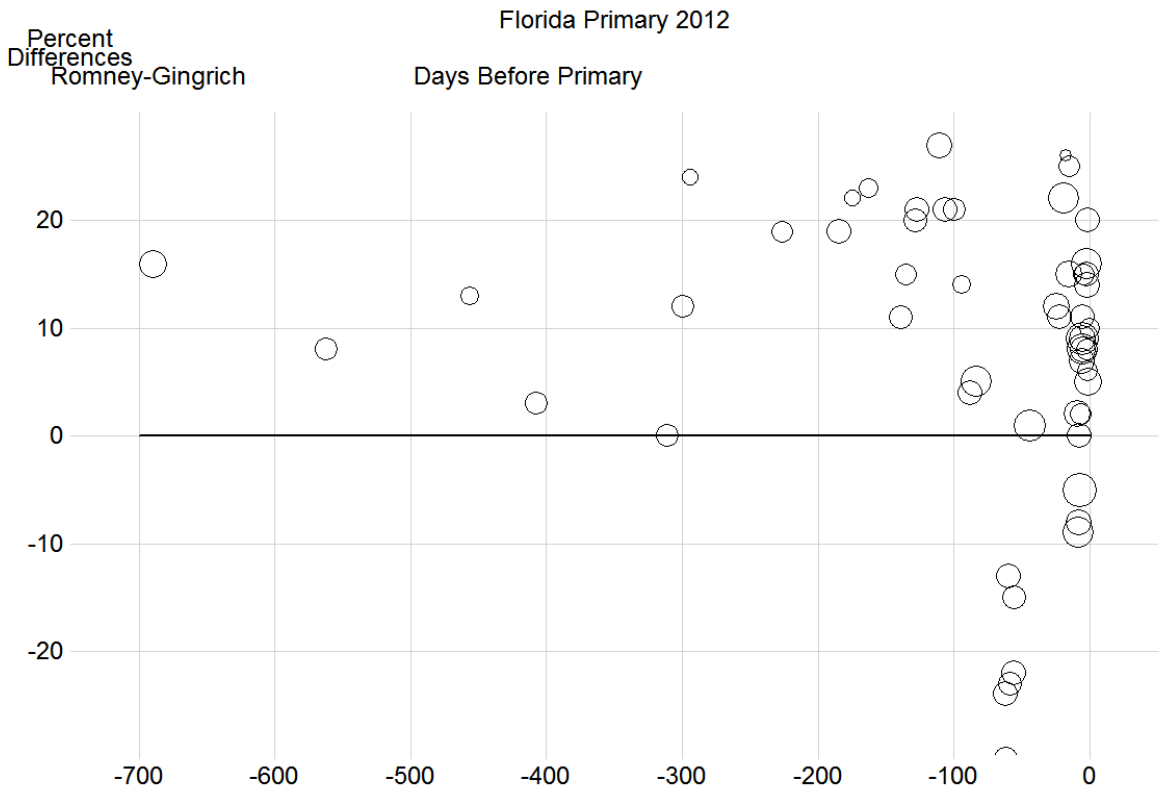
We use www.realclearpolitics.com for political polls. The data is in:
http://www.realclearpolitics.com/epolls/2012/president/fl/florida_republican_presidential_primary-1597.html
The data preparation section produces the following data frame:

```
polls <- read.csv("data/FloridaPolls2102.csv",
                  header=T, as.is=T)
headtail(polls)
```

```
         Date Sample Romney Gingrich  day pbe
1  3/11 - 3/11    639     29       13 -690  54
2  7/16 - 7/18    400     31       23 -562  53
..          ..     ..     ..       ..   ..  ..
54 1/29 - 1/29    646     36       31   -1   1
55 1/30 - 1/30    354     39       29    0   0
```

2.1 Romney Gingrich Differences by days to Primary

```
tiff("pictures/Romney Gingrich differences.tif",
w=1200, h=800)

Grid(xticks=c(-750, seq(-700,0,100), 50),
yticks=seq(-30, 30, 10),
ylab="Florida Primary 2012/ Percent /Differences/
Romney-Gingrich                          Days Before
Primary",at = c(-350, -750, -750, -550), cex=2)

points(polls$day,  polls$Romney - polls$Gingrich,
    cex = sqrt(polls$Sample)/5)
lines(c(-700,0), c(0,0), lwd=2)

dev.off()
```

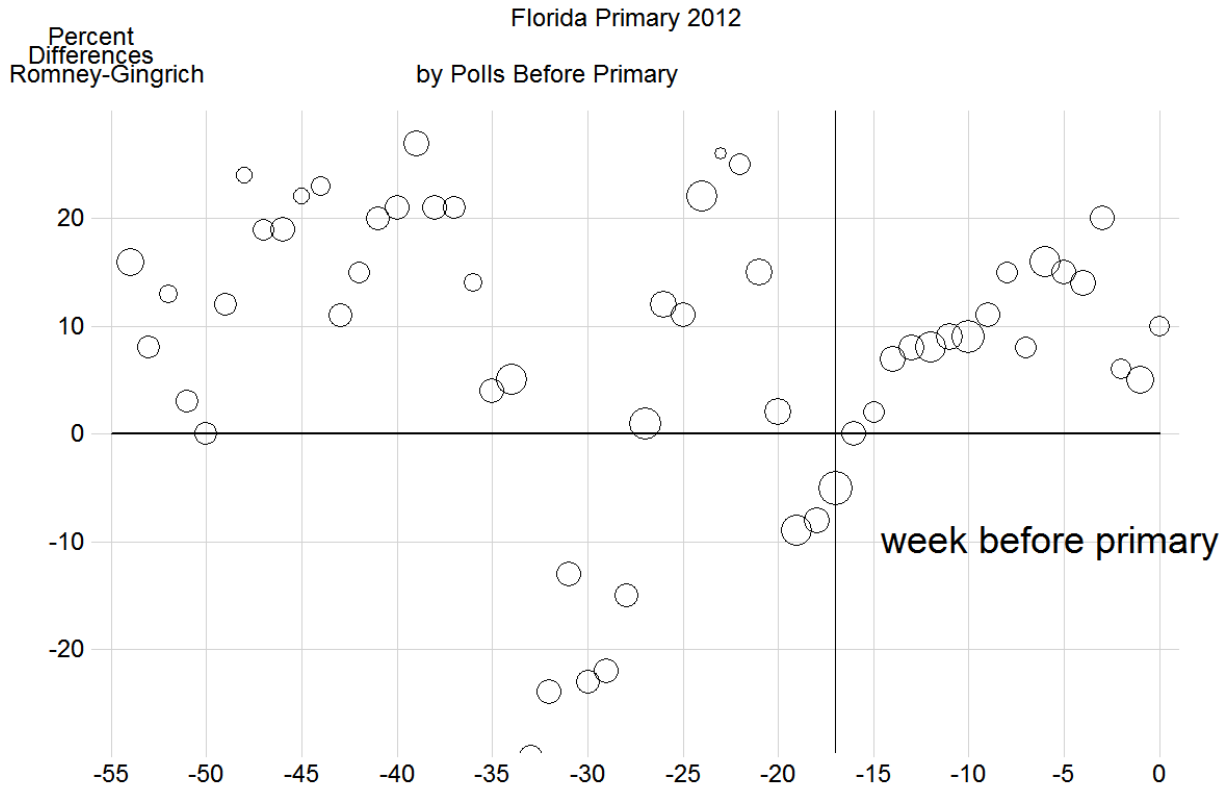


We plot poll differences for 2 years before the Florida primary. The area of the circle for each poll is proportional to sample size for the poll. Romney won the New Hampshire primary 20 days before the Florida primary, and this caused a boost in his poll figures. Gingrich won the South Carolina primary 10 days before the Florida primary, and this caused a boost in *his* poll figures.

As election day nears, the candidates pour in resources, the polls become more frequent, and the differences change rapidly in a short period. Therefore, we will index the time series by "polls before election" rather than days before election

2.2 Romney Gingrich Differences by polls to Primary

```
tiff("pictures/Romney Gingrich differences by
poll.tif", w=1200, h=800)

Grid(xticks=c(-56, seq(-55,0,5), 1),
yticks=seq(-30, 30, 10),
ylab="Florida Primary 2012/ Percent /Differences/
Romney-Gingrich                                    by Polls
Before Primary",at = c(-28, -56, -56, -43), cex=2)

points(-polls$pbe,  polls$Romney - polls$Gingrich,
    cex = sqrt(polls$Sample)/5)
lines(c(-55,0), c(0,0), lwd=2)
lines(c(-17, -17), c(-30, 30))
text(-15, -10, "week before primary",pos=4, cex=3)

dev.off()
```



We see important behaviour close to the election. It's up after -30 (New Hampshire) , it's down after -27(South Carolina) , it's a steady trend after -20 (Romney money), it wanders around, it does everything, and who knows where it might go.

3 Some regressions

The standard realclearpolitics practice for prediction is to average the last week of polls weighted by the sample sizes. We get the regression function to do this for us:

```
options(digits=2)
summary(lm(Romney-Gingrich ~ 1, data=polls[polls$day >
-7,],
        weights=Sample))$coef[1:2]
```

```
[1] 10.7  1.1
```

The estimate would be 11 ± 1 . The error estimate is far too optimistic, because we are assuming that the underlying population difference is not changing over the last week.

```
options(digits=2)
summary(lm(Romney-Gingrich ~ 1,
        data=polls[polls$day > -7,]))$coef[1:2]
```

```
[1] 10.7  1.1
```

Thus weighting by the sample sizes of the different polls has no material effect. The sample sizes are nearly equal.

4 Population difference as integrated Brownian motion

We assume that the actual population differences are an integrated Brownian motion, that is, a twice cumulated sequence of independent normals (with mean zero and constant variance), one for each poll. Each of the actual polled differences is assumed normal with mean equal to that day's population difference, and with a variance given by the sampling distribution for the sample taken that poll. The values are indexed by the number of polls remaining to election day, to allow for the more rapid daily variation in polls results as election day approaches, and as polls are taken more frequently.

Then, given all the observed polls, and given the population differences at neighbouring days, each population difference is normally distributed with mean a weighted sum of the observed poll at that index, and of the neighbouring population differences. By sampling the individual population differences according to this rule, the sample converges (following Markov Chain Monte Carlo protocol) to the conditional distribution of the population difference curve given the observed polls. This is what FitIBM does. It also does a little something to sample the unknown integrated Brownian motion variance of the original population curve. The smaller that variance, the more effect distant polls have on the final prediction. If the true data were in an exact straight line, then the fitted curve would be in the same straight line, with estimated population variance zero. This "line-preserving" property is the reason we are using integrated brownian motion, rather than an ordinary brownian motion that preserves constants.

5 Fit Integrated Brownian population model

We fit the Integrated Brownian model to the last two years, but indexed by the number of polls till the election:

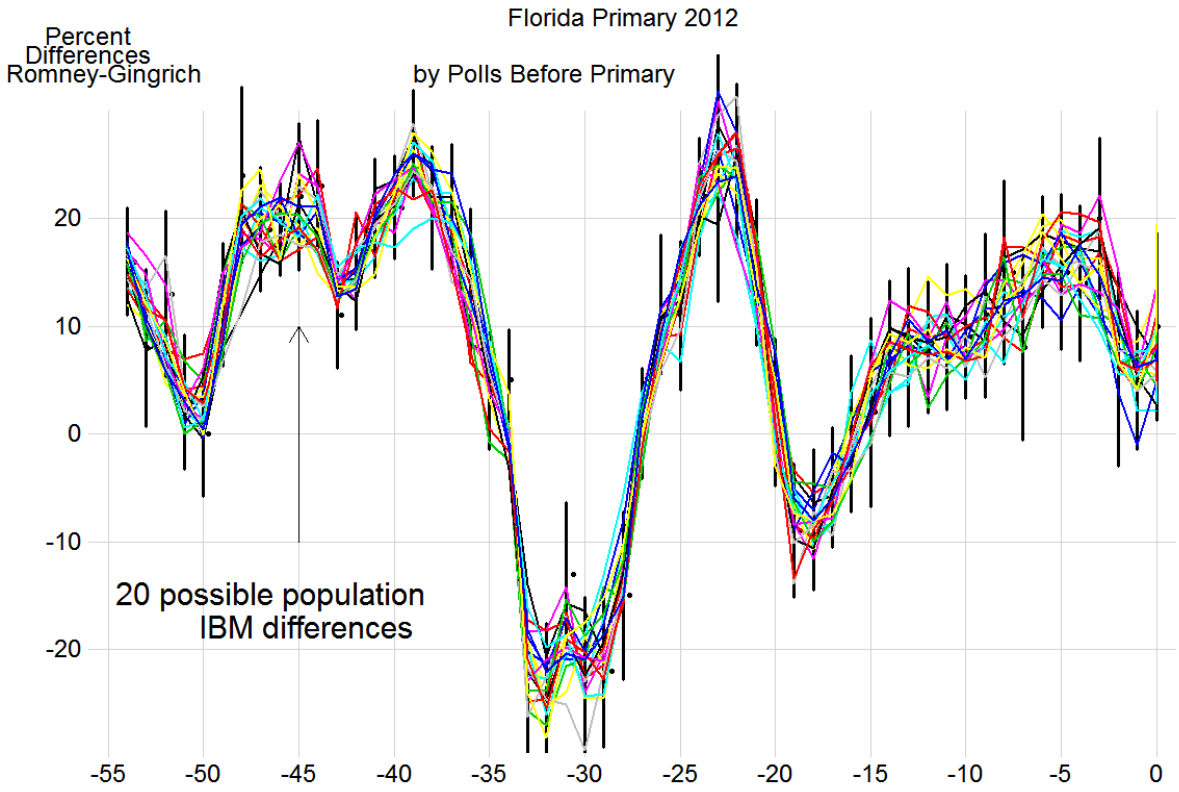
```
Romney <- polls$Romney
Gingrich <- polls$Gingrich
diff  <- Romney - Gingrich

# the variance of the difference between two multinomial
proportions
vx <- ((Romney+Gingrich)*100 - (Romney-Gingrich)^2) /
polls$Sample
sdiff <- sqrt(vx)
yy <- FitIBM(diff, v=1/vx, iter=1000, ahead=1)
```

```
sdBrown:  6.4
 mpredict:  9.3 +- 3.9
```

Now plot the observed points with sd bars, and the fitting brownians:

```
tiff("pictures/fitted IBM.tif", w=1200, h=800)

Grid(xticks=c(-56, seq(-55,0,5), 1),
  yticks=seq(-30, 30, 10),
ylab="Florida Primary 2012/ Percent /Differences/
Romney-Gingrich                                  by Polls
Before Primary",at = c(-28, -56, -56, -43), cex=2)

points(-polls$pbe + 0.5* runif(dim(polls)[1]), diff,
pch=16)
for (i in 1:dim(polls)[1] )
lines(c(-polls$pbe[i], -polls$pbe[i]), lwd=3,
      c(diff[i] - 2 * sdiff[i], diff[i] + 2 * sdiff[i]))

for(i in seq(101, 1000, 45) )
lines(c(-polls$pbe,0), yy[i, ], col=i, lwd=2)

arrows(-45,-10,-45,10)
```

```
text(-55,-15,"20 possible population",pos=4,cex=2.5)
text(-51, -18," IBM differences", pos=4, cex=2.5)
dev.off()
```



6 Conclusions

The integrated Brownian motion fits lie respectably within the 95% error bars from the observed data. It is very willing to bend onto a new straight line path.

Our prediction was Romney by 9±4 points.

The actual outcome was 14.5 points in favour of Romney, so our prediction is about 1.5 standard deviations off. We are doing better than the straight mean prediction, about 4 standard deviations off.

7 Data Preparation

We use www.realclearpolitics.com for political polls:
http://www.realclearpolitics.com/epolls/2012/president/fl/florida_republican_presidential_primary-1597.html
The web table is copied and pasted into "data/RawFloridaPolls.csv", irrelevant text headers and tails are deleted, and an R ready data frame is read from that file.

7.1 Read the data:

```
f <-read.csv("data/RawFloridaPolls2012.csv",
            header=T,as.is=T)
head(f)
```

```
                Poll        Date  Sample MoE Romney
1      Final Results          --      --  --     46
2        RCP Average 1/24 - 1/30      --  --     42
3            PPP (D) 1/28 - 1/30 1087 LV   3     39
4  Insider Advantage 1/29 - 1/29  646 LV 3.8     36
5            PPP (D) 1/28 - 1/29  733 LV 3.6     39
6 Suffolk University 1/28 - 1/29  500 LV 4.5     47
  Gingrich Santorum Paul       Spread
1       32     13.4    7 Romney +14.5
2       29       13 10.3 Romney +13.0
3       31       15   11    Romney +8
4       31       12   12    Romney +5
5       32       14   11    Romney +7
6       27       12    9   Romney +20
```

7.2 Disaggregate the PPP polls, cumulated over 3 days:

```
fd <- f
fd[3, 2] <- "1/30 - 1/30"
fd[3, 3] <- "354 LV"
fd[3, 5] <- round( (1087 * 39 - 733 * 39) / 354)
fd[3, 6] <- round( (1087 * 31 - 733 * 32) / 354)
fd[5, 2] <- "1/29 - 1/29"
fd[5, 3] <- "346 LV"
```

```
fd[5, 5] <- round( (733 * 39 - 387 * 40) / 346)
fd[5, 6] <- round( (733 * 32 - 387 * 32) / 346)
head(fd)
```

```
                Poll        Date Sample MoE Romney Gingrich
1      Final Results          --     --  --     46       32
2        RCP Average 1/24 - 1/30     --  --     42       29
3            PPP (D) 1/30 - 1/30 354 LV   3     39       29
4  Insider Advantage 1/29 - 1/29 646 LV 3.8     36       31
5            PPP (D) 1/29 - 1/29 346 LV 3.6     38       32
6 Suffolk University 1/28 - 1/29 500 LV 4.5     47       27
  Santorum Paul       Spread
1     13.4    7 Romney +14.5
2       13 10.3 Romney +13.0
3       15   11    Romney +8
4       12   12    Romney +5
5       14   11    Romney +7
6       12    9   Romney +20
```

7.3 Reduce data to needed rows and columns:

Eliminate the LV (likely voters) from the sample counts, and select variables:

```
polls <- fd[57:3, c(2,3,5,6)]
polls$Sample <- as.numeric(substr(polls$Sample, 1, 3))
headtail(polls)
```

```
         Date Sample Romney Gingrich
57 3/11 - 3/11    639     29       13
56 7/16 - 7/18    400     31       23
..           ..     ..     ..       ..
4  1/29 - 1/29    646     36       31
3  1/30 - 1/30    354     39       29
```

7.4 Fix dates:

Get the dates into a form that R recognizes:

```
Dateday <- function(date){
# produce day from polls date
# using r functions as.Date and julian
year <- rep(11,55)
year[1:4] <- 10
year[29:55] <- 12

rdate <- paste(date, year, sep="/")
rdate <- gsub(" ", "", rdate)
rdate <- as.Date(rdate, "%m/%d/%y")
return( julian(rdate) - max(julian(rdate)) )
}

newdate <- gsub("-", "", polls$Date)
first  <- substr(newdate, 1, 5)
last  <- substr(newdate,6, nchar(newdate))
polls$day  <-  (Dateday(first) + Dateday(last)) / 2
```

7.5 Count number of polls till election day:

```
polls$pbe <- (dim(polls)[1]:1) -1
```

Save the data for analysis:

```
write.csv(polls, "data/FloridaPolls2102.csv",
row.names=F)
```

8 Functions

```
headtail <- function(x, nrows=2){
# give both head and tail of data
xm <- x[1:1,]
rownames(xm) <- ".."
xm[1,] <- rep("..", dim(x)[2])
return( rbind(head(x, nrows), xm, tail(x, nrows)) )
}
```

```
Grid <- function(xticks, yticks, ylab="",
at=(min(xticks)+ mean(xticks))/2, cex=2.5){
# background for plot using grid of light grey lines

par(mar=c(3,3,6,2))
plot(1, 1,  xlim=range(xticks),  ylim = range(yticks),
        xlab="", ylab="", axes=F, pch="")

# use only interior values of tick ranges in plots
usey <- rep( T, length(yticks) )
usey[c( 1, length(yticks) )] <- F
usex <- rep( T, length(xticks) )
usex[c( 1, length(xticks) )] <- F

# grey lines in both directions
for ( row in yticks[usey] )
lines(range(xticks), c(row, row), col="light grey")
for ( col in xticks[usex] )
lines(c(col, col), range(yticks), col="light grey")

# put ylab on left top, using / to split long expressions
ylabs <- unlist(strsplit(ylab,"/"))

# identify tick marks on both axes
if (length(yticks) > 2)
text(pos=2, rep(min(xticks), length(yticks)-2 ),
    yticks[usey], yticks[usey], cex=2, xpd=T)
if (length(xticks)>2)
text(pos=1, xticks[usex],  rep(min(yticks),
   length(xticks)-2), xticks[usex], cex=2, xpd=T)
lylabs <- min(5, length(ylabs))

if(lylabs > 0)
mtext(ylabs, side=3,line = (5/lylabs)*(lylabs-1):0,
   at = at, cex=cex)

par(mar=c(5, 4, 4, 2))
```

```
invisible()
}



FitIBM <- function(x, v=rep(0.25, length(x)), iter=20,
ahead=1){
# fit integrated brownian motion y to a series x ,
# with inverse variances ( v=0 handles missing values)
# the integrated brownian motion
# is a sequence of iid normals twice cumulated
# predict ahead value in the series
# do something less fancy for small length series

if (length(x) < 3){
cat(" sdBrown: ", NA, "\n")
cat(" mpredict: ", round(mean(x),1), "+-",
                 round(2/sqrt(length(x)),1), "\n")
return("series length < 3")
}

# add to x for advance
x <- c(x, rep(x[length(x)], ahead-1) )
v <- c(v, rep(0, ahead-1) )

# handle two or three extra places at either end
lx <- length(x)
x <- c(x[1], x[1], x, x[lx], x[lx], x[lx])
v <- c(0, 0, v, 0, 0, 0)

# initialise y and vy
y <- x
vx <-(lx-ahead-1)/sum( diff(diff(x[3:(lx+2-ahead)])) ^
2)
iv <- vx
vy <- c(0, 0,  rep(iv, lx + 1), 0, 0)
yy <- matrix(0, iter, lx + 1)
yyv <- rep(0, iter)
```

```
# select integrated brownian motion one normal at a time
# depending on data and 4 neighbors
for(it in 1:iter){
yy[it,] <- y[3:(lx + 3)]
yyv[it] <- iv ^ ( - 1/2)
for( i in 3:(lx + 3) ) {

ivi <- v[i] + 4*vy[i] + vy[i - 1] + vy[i + 1]
 mi  <- (v[i] * x[i] + 2 * vy[i] * (y[i - 1] + y[i+1])
+
          vy[i - 1] * ( 2 * y[i - 1] - y[i - 2] ) +
          vy[i + 1] * ( 2 * y[i + 1] - y[i + 2] ) ) / ivi
 y[i] <- rnorm(1, mi, ivi ^ (-1/2) )
}

# a random posterior variance for the IBM terms
iv <- rgamma(1, (lx-2)/2, 1/2)/
   sum( diff(diff(y[3:(lx+3)])) ^ 2  )
vy[3:(lx+3)] <- iv

# put out sd for brownian and prediction
}
cat(" sdBrown: ", round(mean(yyv),2), "\n")
cat(" mpredict: ", round(mean( yy[, dim(yy)[2]]
),2),"+-",
                round(sd( yy[, dim(yy)[2]] ), 2) ,"\n")

return(yy)
}
```