**Key points**

* Poll aggregators combine the results of many polls to simulate polls with a large sample size and therefore generate more precise estimates than individual polls.
* Polls can be simulated with a Monte Carlo simulation and used to construct an estimate of the spread and confidence intervals.
* The actual data science exercise of forecasting elections involves more complex statistical modeling, but these underlying ideas still apply.

**Code: Simulating polls**

Note that to compute the exact 95% confidence interval, we would use qnorm(.975)\*SE\_hat instead of 2\*SE\_hat.

d <- 0.039

Ns <- c(1298, 533, 1342, 897, 774, 254, 812, 324, 1291, 1056, 2172, 516)

p <- (d+1)/2

# calculate confidence intervals of the spread

confidence\_intervals <- sapply(Ns, function(N){

X <- sample(c(0,1), size=N, replace=TRUE, prob = c(1-p, p))

X\_hat <- mean(X)

SE\_hat <- sqrt(X\_hat\*(1-X\_hat)/N)

2\*c(X\_hat, X\_hat - 2\*SE\_hat, X\_hat + 2\*SE\_hat) - 1

})

# generate a data frame storing results

polls <- data.frame(poll = 1:ncol(confidence\_intervals),

t(confidence\_intervals), sample\_size = Ns)

names(polls) <- c("poll", "estimate", "low", "high", "sample\_size")

polls

**Code: Calculating the spread of combined polls**

Note that to compute the exact 95% confidence interval, we would use qnorm(.975) instead of 1.96.

d\_hat <- polls %>%

summarize(avg = sum(estimate\*sample\_size) / sum(sample\_size)) %>%

.$avg

p\_hat <- (1+d\_hat)/2

moe <- 2\*1.96\*sqrt(p\_hat\*(1-p\_hat)/sum(polls$sample\_size))

round(d\_hat\*100,1)

round(moe\*100, 1)