

Political Science 209 - Fall 2018

Prediction

Florian Hollenbach

4th October 2018

In-class Exercise Measurement

Carvalho, Leandro S., Meier, Stephen, and Wang, Stephanie W. (2016). "Poverty and economic decision-making: Evidence from changes in financial resources at payday." American Economic Review, Vol. 106, No. 2, pp. 260-284.

In-class Exercise Measurement

Do changes in one's financial circumstances affect one's decision-making process and cognitive capacity? In an experimental study, researchers randomly selected a group of US respondents to be surveyed before their payday and another group to be surveyed after their payday. Under this design, the respondents of the Before Payday group are more likely to be financially strained than those of the After Payday group. The researchers were interested in investigating whether or not changes in people's financial circumstances affect their decision making and cognitive performance. Other researchers have found that scarcity induce an additional mental load that impedes cognitive capacity.

Poverty and economic decision-making

In this study, the researchers administered a number of decision-making and cognitive performance tasks to the Before Payday and After Payday groups. We focus on the numerical stroop task, which measures cognitive control. In general, taking more time to complete this task indicates less cognitive control and reduced cognitive ability. They also measured the amount of cash the respondents have, the amount in their checking and saving accounts, and the amount of money spent.

Poverty and economic decision-making

Load the poverty.csv data set.

Poverty and economic decision-making

Variables:

- *treatment*: Treatment conditions: Before Payday and After Payday
- *cash*: Amount of cash respondent has on hand
- *accts_amt*: Amount in checking and saving accounts
- *stroop_time*: Log-transformed average response time for cognitive stroop test
- *income_less20k*: Binary variable: 1 if respondent earns less than 20k a year and 0 otherwise

Look at a summary of the poverty data set to get a sense of what its variables looks like.

Poverty and economic decision-making

Question 1

1. Use histograms to examine the univariate distributions of the two financial resources measures: cash and accts_amt. What can we tell about these variables' distributions from looking at the histograms? Evaluate what the shape of these distributions could imply for the authors' experimental design.
2. Now, take the natural logarithm of these two variables and plot the histograms of these transformed variables. How does the distribution look now? What are the advantages and disadvantages of transforming the data in this way?

NOTE: Since the natural logarithm of 0 is undefined, researchers often add a small value (in this case, we will use \$1 so that $\log 1 = 0$) to the 0 values for the variables being transformed.

Poverty and economic decision-making

Question 2a

Now, let's examine the primary outcome of interest for this study—the effect of a change in financial situation (in this case, getting paid on payday) on economic decision-making and cognitive performance. Begin by calculating the treatment effect for the stroop_time variable (a log-transformed variable of the average response time for the stroop cognitive test), using first the mean and then the median. What does this tell you about differences in the outcome across the two experimental conditions?

Poverty and economic decision-making

Question 2b

Secondly, let's look at the relationship between financial circumstances and the cognitive test variable. Produce two scatter plots side by side (hint: use the `par(mfrow)`) before your plot commands to place graphs side-by-side), one for each of the two experimental conditions, showing the bivariate relationship between your log-transformed cash variable and the amount of time it took subjects to complete the stroop cognitive test administered in the survey (`stroop_time`). Place the `stroop_time` variable on the y-axis. Be sure to title your graphs to differentiate between the Before Payday and After Payday conditions. Now do the same, for the log-transformed `accts_amt` variable.

Poverty and economic decision-making

Question 3

Now, let's take a closer look at whether or not the Before Payday versus After Payday treatment created measurable differences in financial circumstances. What is the effect of payday on participants' financial resources? To help with interpretability, use the original variables cash and accts_amt to calculate this effect. Calculate both the mean and median effect. Does the measure of central tendency you use affect your perception of the effect?

Poverty and economic decision-making

Question 4

Compare the distributions of the Before Payday and After Payday groups for the log-transformed cash and accts_amt variables. Use quantile-quantile plots to do this comparison, and add a 45-degree line in a color of your choice (not black). Briefly interpret your results and their implications for the authors' argument that their study generated variation in financial resources before and after payday. When appropriate, state which ranges of the outcome variables you would focus on when comparing decision-making and cognitive capacity across these two treatment conditions.

Poverty and economic decision-making

Question 5

In class, we covered the difference-in-difference design for comparing average treatment effects across treatment and control groups. This design can also be used to compare average treatment effects across different ranges of a pre-treatment variable- a variable that asks about people's circumstances before the treatment and thus could not be affected by the treatment. This is known as heterogeneous treatment effects – the idea that the treatment may have differential effects for different subpopulations. Let's look at the pre-treatment variable income_less20k. Calculate the treatment effect of Payday on amount in checking and savings accounts separately for respondents earning more than 20,000 dollars a year and those earning less than 20,000 dollars. Use the original accts_amt variable for this calculation. Then take the difference between the effects you calculate. What does this comparison tell you about how payday affects the amount that people have in their accounts? Are you convinced by the authors' main finding from Question 2 in light of your investigation of their success in manipulating cash and account balances before and after payday?

The Amazing Tale of Paul the Psychic Octopus: Germany's World Cup Soothsayer

Sure, Germany is back in the World Cup final. But it'll have to beat Argentina without Paul, the cephalopod that correctly predicted the results of all eight (!) German matches last go-around.



Emily Shire · 07.12.14 12:00 AM ET



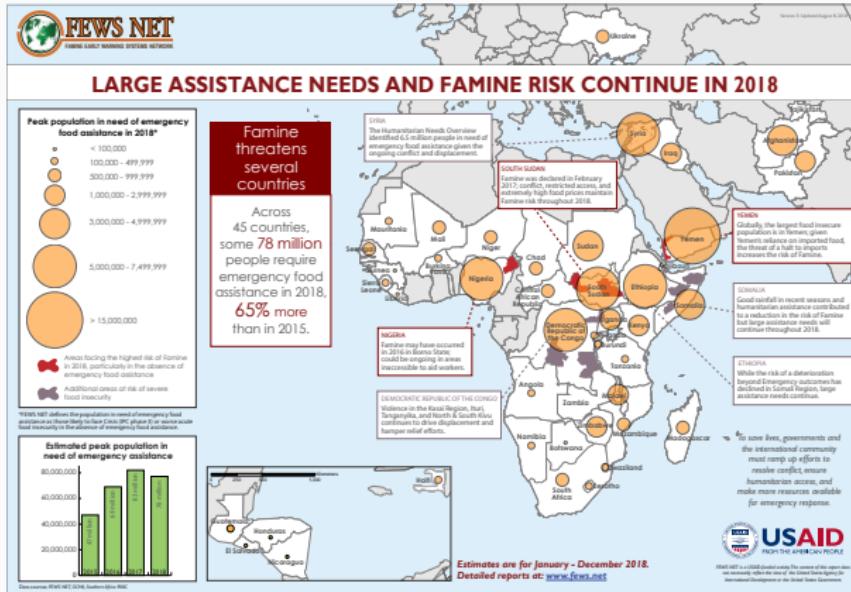
Prediction

- One important task of (social) scientists can be *prediction*
- Forecasting future events, e.g., conflict, unrest, elections
- Causal inference, also involves prediction, of what?

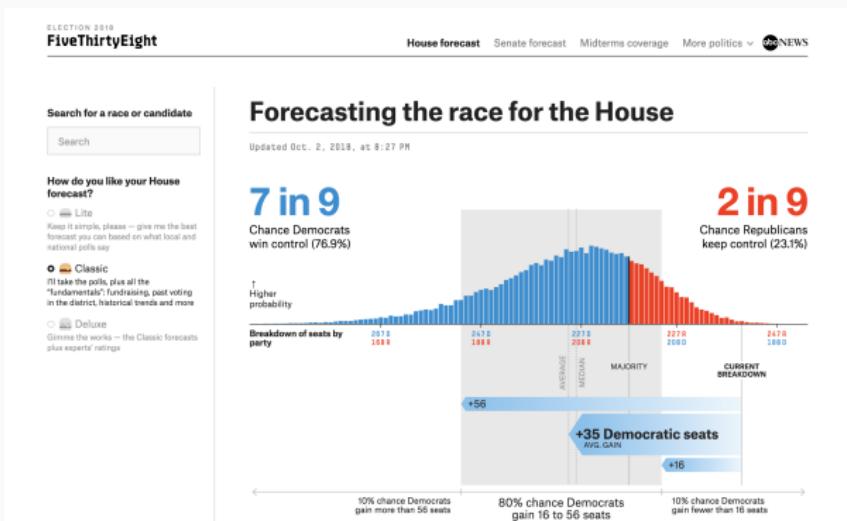
Prediction

- One important task of (social) scientists can be *prediction*
- Forecasting future events, e.g., conflict, unrest, elections
- Causal inference, also involves prediction, of what?
- To estimate the causal effect we are essentially predicting the *counterfactual*

Prediction



Prediction



Prediction

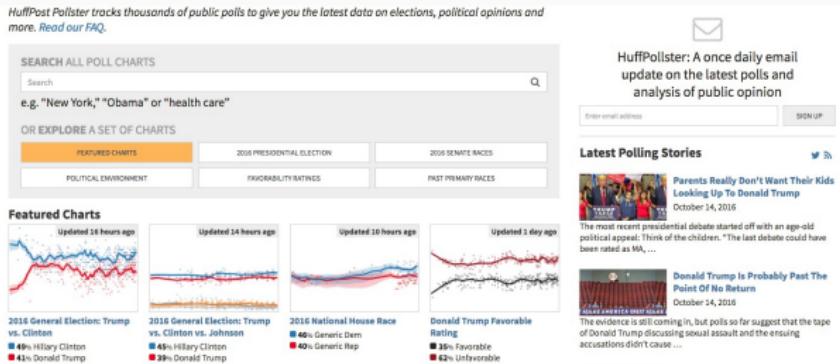
- Elections can be predicted using fundamentals

Prediction

- Elections can be predicted using fundamentals
- Or we can use polls to predict results

Prediction with polls

- We will use a nice *R* package called *pollstR*, which scrapes the data from Huffington Post:



Prediction with polls

```
library(pollsterR)
chart_name <- "2016-general-election-trump-vs-clinton"
polls2016 <- pollster_charts_polls(chart_name)[["content"]]
```

Prediction with polls

- Let's calculate a variable that is *days until the election*

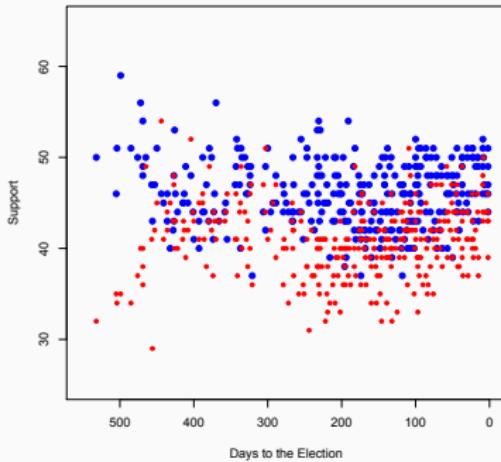
```
class(polls2016$end_date)
polls2016$DaysToElection <-
    as.Date("2016-11-8") - polls2016$end_date
```

Prediction with polls

We could make a very simple plot of all the polls over time

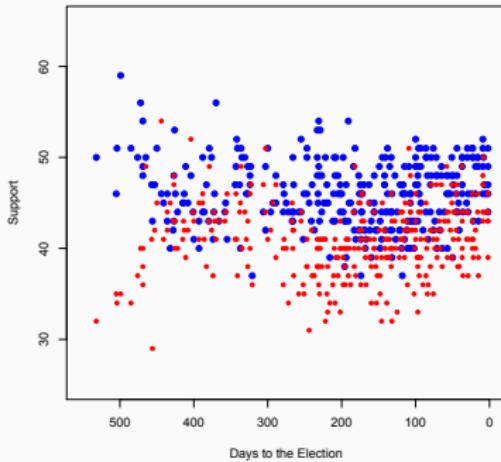
```
plot(polls2016$DaysToElection, polls2016$Clinton,
      xlab = "Days to the Election", ylab = "Support",
      xlim = c(550, 0), ylim = c(25, 65), pch = 19,
      col = "blue")
points(polls2016$DaysToElection, polls2016$Trump,
       pch = 20, col = "red")
```

Prediction with polls



But that looks kind of dumb

Prediction with polls



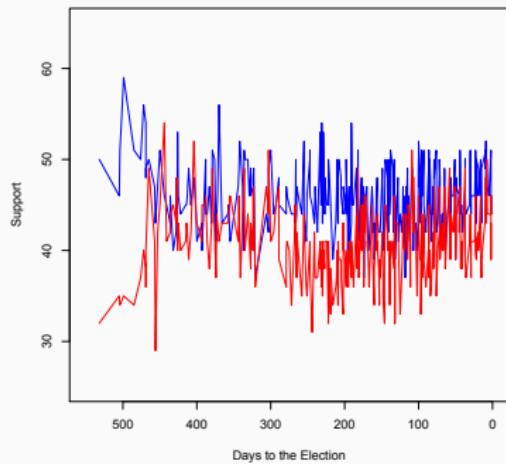
But that looks kind of dumb

Lines?

Plotting polls

```
plot(polls2016$DaysToElection, polls2016$Clinton, type = "l",
      xlab = "Days to the Election", ylab = "Support",
      xlim = c(550, 0), ylim = c(25, 65), pch = 19,
      col = "blue")
lines(polls2016$DaysToElection, polls2016$Trump,
      col = "red")
```

Prediction with polls



Prediction with polls

- Never trust a single poll
- Maybe we could smooth the polls over time?
- Average the polls that are close to each other

Prediction with polls

- This is called a *moving average*
- Average all the polls within a certain time window
- Window size determines amount of smoothing

Creating a Moving Average

- In *R*, for each day, we subset the relevant polls and compute the average
- That's a lot of subsetting and averaging (532 days)
- Any ideas of how to do this fast?

Creating a Moving Average

- In *R*, for each day, we subset the relevant polls and compute the average
- That's a lot of subsetting and averaging (532 days)
- Any ideas of how to do this fast?

Loops

Loops in R

```
for (i in X) {  
    expression1  
    expression2  
    ...  
    expressionN  
}
```

Loops in R

Elements of a loop:

- i : counter (can use any object name other than i)
- X : vector containing a set of ordered values the counter takes
- *expression*: a set of expressions that will be repeatedly evaluated

Loops in R

Elements of a loop:

- i : counter (can use any object name other than i)
- X : vector containing a set of ordered values the counter takes
- *expression*: a set of expressions that will be repeatedly evaluated

{ }: curly braces to define the beginning and the end

Loops in R

Simple Example:

```
for (i in c(1,2,3,4,5) {  
    print(i)  
}
```

What does this loop do?

Loops in R

- Indentation is important for the readability of code (Rstudio does this automagically)
- Test Code without loop first by setting the counter to a specific value

Loops in R

Printing out an iteration number can be helpful for debugging:

```
values <- c(1, -1, 2)
results <- rep(NA, 3)
for (i in 1:3) {
  cat("iteration", i, "\n")
  results[i] <- log(values[i])
}
```

Loops in R

Let's create a moving average:

- Begin by creating vector for counter & setting window size

```
days <- 500:26  
window <- 7
```

Loops in R

Create empty vectors

```
Clinton.pred <- Trump.pred <- rep(NA, length(days))
```

Loops in R

Create empty vectors

```
Clinton.pred <- Trump.pred <- rep(NA, length(days))
```

Now the loops:

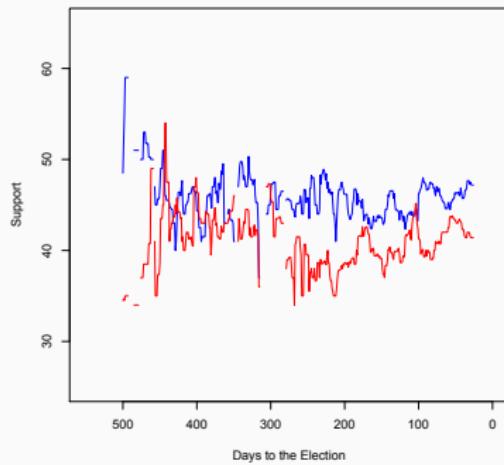
```
for (i in 1:length(days)) {  
  week.data <-  
    subset(polls2016,  
           subset = ((DaysToElection < (days[i] + window))  
                      & (DaysToElection >= days[i])))  
  Clinton.pred[i] <- mean(week.data$Clinton)  
  Trump.pred[i] <- mean(week.data$Trump)  
}
```

Loops in R

Smoothed Plot:

```
plot(days, Clinton.pred, type = "l", col = "blue",
      xlab = "Days to the Election", ylab = "Support",
      xlim = c(550, 0), ylim = c(25, 65))
lines(days, Trump.pred, col = "red")
```

Smoothed Plot:



2 week Smoothing

```
Clinton.pred <- Trump.pred <- rep(NA, length(days))
window <- 14
```

2 week Smoothing

```
Clinton.pred <- Trump.pred <- rep(NA, length(days))
window <- 14
```

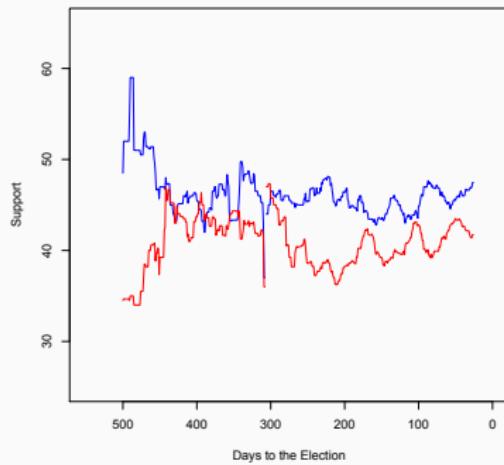
Now the loops:

```
for (i in 1:length(days)) {
  week.data <-
    subset(polls2016,
           subset = ((DaysToElection < (days[i] + window))
                      & (DaysToElection >= days[i])))
  Clinton.pred[i] <- mean(week.data$Clinton)
  Trump.pred[i] <- mean(week.data$Trump)
}
```

2 week Smoothing

```
plot(days, Clinton.pred, type = "l", col = "blue",
      xlab = "Days to the Election", ylab = "Support",
      xlim = c(550, 0), ylim = c(25, 65))
lines(days, Trump.pred, col = "red")
```

Smoothed Plot:



Smoothed Plot:

Let's add some explanations/legend to the plot

```
text(400, 50, "Clinton", col = "blue")
text(400, 40, "Trump", col = "red")
```

Smoothed Plot:

Let's add some explanations/legend to the plot

```
text(200, 60, "party\n conventions")
abline(v = as.Date("2016-11-8") - as.Date("2016-7-28"),
       lty = "dotted", col = "blue")
abline(v = as.Date("2016-11-8") - as.Date("2016-7-21"),
       lty = "dotted", col = "red")
text(50, 30, "debates")
abline(v = as.Date("2016-11-8") - as.Date("2016-9-26"),
       lty = "dashed")
abline(v = as.Date("2016-11-8") - as.Date("2016-10-9"),
       lty = "dashed")
```