POLI 428 - R Workshop

Sam Lee

Introduction

This workshop covers key R programming concepts used in POLI 428, including loops, indices, vectorization, and functions.

1. Read in the Data

The files below contain employment data (2016-2024) and election results by county in Minnesota:

- **Election Data**: Includes the proportions of votes for Republican (R) and Democrat (D) nominees in both 2020 and 2024 elections.
- **Employment Data**: Contains labor force (LF), employment (E), unemployment (U), employment rate (ER), and unemployment rate (UR).

I've done this step for you.

```
    County                R                D              State          
 Length:174         Min.   :0.2710   Min.   :0.2120   Length:174        
 Class :character   1st Qu.:0.5770   1st Qu.:0.3000   Class :character  
 Mode  :character   Median :0.6420   Median :0.3360   Mode  :character  
                    Mean   :0.6212   Mean   :0.3577                     
                    3rd Qu.:0.6787   3rd Qu.:0.4020                     
                    Max.   :0.7710   Max.   :0.7020                     
      Year     
 Min.   :2020  
 1st Qu.:2020  
 Median :2022  
 Mean   :2022  
 3rd Qu.:2024  
 Max.   :2024  
```

```
     Year         Month                  LF               E         
Min.   :2016   Length:9222        Min.   :  1565   Min.   :  1501  
1st Qu.:2018   Class :character   1st Qu.:  5830   1st Qu.:  5566  
Median :2020   Mode  :character   Median : 12284   Median : 11836  
Mean   :2020                      Mean   : 35402   Mean   : 34131  
3rd Qu.:2022                      3rd Qu.: 23451   3rd Qu.: 22502  
Max.   :2024                      Max.   :737566   Max.   :698660  
      U                 UR             County                ER       
Min.   :   20.0   Min.   : 1.000   Length:9222        Min.   :74.13  
1st Qu.:  216.0   1st Qu.: 2.800   Class :character   1st Qu.:95.26  
Median :  465.5   Median : 3.500   Mode  :character   Median :96.46  
Mean   : 1270.7   Mean   : 4.014                      Mean   :95.99  
3rd Qu.:  975.0   3rd Qu.: 4.700                      3rd Qu.:97.24  
Max.   :86464.0   Max.   :25.900                      Max.   :98.97  
   State          
Length:9222       
Class :character  
Mode  :character  
```

Note: Employment data was retrieved from the BLS, and election data was scraped from CNN.

2. Indices, Matrices, and Data Frames

Definitions

- A *matrix* in R is a two-dimensional array where all elements are of the same type (e.g., numeric, character).
- A *data frame* is a two-dimensional table-like structure where columns can have different types of data.

Run `dim()` and `str()` on both election and employment to see the dimensions (# of rows, # of columns) and data type of each column

```
dim(employment)
dim(election)

str(employment)
str(election)
```

Working with Indices

Indices in R are used to access specific elements of a matrix or data frame using their row and column positions.

This extracts the first element in the first column—since both data frames and matrices are two-dimensional indices function the same for both data types

```
employment[1, 1]
```

```
[1] 2016
```

This is also equivalent to... (since year represents the first column)

```
employment$Year[1]
```

```
[1] 2016
```

Implicit indices work by omitting either a column index or a row index to indicate that you want the **entire** row or **entire** column.

Omitting the first index (the row index) indicates that we want all rows for column 1.

```
employment[,1]
```

This is equivalent to

```
employment$Year
```

We can use vectors to indicate which index we want as well.

This statement gets the first five rows and columns 1, 6, and 7 in a single statement

```
employment[1:5, c(1, 6, 7)]
```

```
  Year  UR County
1 2016 9.3 Aitkin
2 2016 9.1 Aitkin
3 2016 9.8 Aitkin
4 2016 7.5 Aitkin
5 2016 5.5 Aitkin
```

3. Loops and Vectorization

Loops in R tend to be really slow (although they’re okay for a small number of iterations). We use vectorization to attain a similar speed demonstrated in lower level languages such as C++.

First, let’s use loops to find the sum of squares of the `R` column in 2024 with the election data.

I’ll help you out here by filtering to the year 2024. *Note that this gets all of the columns through implicit indexing and all of the years that satisfy ‘Year == 2024‘.*

```
election_2024 <- election[election$Year == 2024, ]
```

Now, subtract each value in the `R` column from the mean of `R`, square each value, and then place it in the `squared_deviations` vector.

```
squared_deviations = numeric(nrow(election_2024))
for(i in 1:nrow(election_2024)){
  squared_deviations[i] = (election_2024$R[i] - mean(election_2024$R))^2
}
```

Finally, add each element to find the sum of squares

```
squared_deviation_sum = 0
for(i in 1:length(squared_deviations)){
  squared_deviation_sum = squared_deviation_sum + squared_deviations[i]
}
squared_deviation_sum
```

Vectorization Approach

We can do the same thing as above with much less and faster code. Without going too much into the mathematical details of what qualifies as a “vector’’, vectors are simply one dimensional sets of elements.

When you add two vectors of the same length, each element is added to the corresponding element:

```
1:3 + 1:3
```

```
[1] 2 4 6
```

This is equivalent to $\begin{bmatrix}1+1 & 2+2 & 3+3\end{bmatrix}'$. This also works for any other one-to-one mathematical operator (i.e. the square function). Thus, we can apply this concept to the problem above.

Take some time to make sure you understand what this code does.

```
sum((election_2024$R - mean(election_2024$R))^2)
```

```
[1] 0.9126491
```

Does this match your answer from above?

Your turn: Find the sums of squares for the 2020 election column `R`

```
election_2020 <- election[election$Year == 2020, ]
```

4. Functions

A function in R is a reusable block of code that performs a specific task. Functions take inputs (called arguments), process them, and return an output.

Here's the basic syntax of defining a function

```
my_function <- function(...){
  ...
}
```

Ex: Create a function for finding the mean of the employment rate for a given county and year

Take time to make sure you understand what this function is doing here.

```
get_mean <- function(v="ER", county, year){
  #Note the 'filter' function which some of you may have used before works the exact same
  subset <- employment[employment$Year == year & employment$County == county, ]
  return(mean(subset[[v]], na.rm=T))
}
get_mean(county="Aitkin", year=2024)
```

```
[1] 94.85257
```

Your turn: Write a function that calculates the difference in `R` (from the elections data) for a given county between years 2024 and 2020 (*hint: you only need one argument*)

```
get_diff <- function(county){
  election$R[election$Year == 2024 & election$County == county] -
    election$D[election$Year == 2024 & election$County == county]
}
```

5. Putting it All Together

In the 2024 presidential election one of the most frequently cited top issues for voters was the economy and inflation. Suppose we are interested in seeing how the employment (as a general measure of the state of the economy) affects the way individuals voted on an aggregate level in Minnesota.

One way to examine this effect is to see how the *change* in employment from 2020 to 2024 affected the *change* in voting outcomes in Minnesota. This is a (for reasons I'll discuss later, an albeit simplified and perhaps naive) version of a difference-in-differences model which we will discuss later in the class.

To begin, use your `get_diff` function to calculate the voting differences in `D` (in the elections data for each county). You may use a for loop or vectorization here (using the `get_diff` function, as it turns out, is not required, but still may be used).

```
#You made need this vector...
counties <- unique(election$County) #87 counties

election_differences <- election_2024$D - election_2020$D
```

The differences vector defined above should only have 87 elements (the number of counties in MN)

```
length(election_differences)
```

```
[1] 87
```

Now we want to get the difference in employment from 2020 to 2024 for each county. To simplify things a bit, for now let's just compare the **mean** employment for each county from 2020 to 2024.

First get the mean employment rate for each county in year 2020 (you may use a similar process that you did above, only this time you may use the `get_mean` function that I defined earlier).

```r
e_2020 <- sapply(counties, function(c) get_mean("E", c, 2020))
```

Do the same thing for 2024

```r
e_2024 <- sapply(counties, function(c) get_mean("E", c, 2024))
```

Now find the differences for each county from 2020 to 2024

```r
e_differences <- e_2024 - e_2020
```

This again should be a vector of 87 elements

```r
length(e_differences)
```

```
[1] 87
```

Now we can measure the effect of the change of employment on election outcomes in MN.

Use the simple linear regression model, $y_i = \beta_0 + \beta_1 x_i + \varepsilon_i$, where y is the change in voting outcomes and x is the change in employment outcomes. We can do this with the `lm(.)` function:

To make this analysis `more appropriate` I will weight each observation by the labor force in that county (you don't need to do anything here)

```r
weights = sapply(counties, function(c) get_mean("LF", c, year=2024))
```

The weights represent the proportion of the state's total labor force in 2024. The following code standardizes the weights so that they sum to 1.

```r
weights = weights/sum(weights)
```

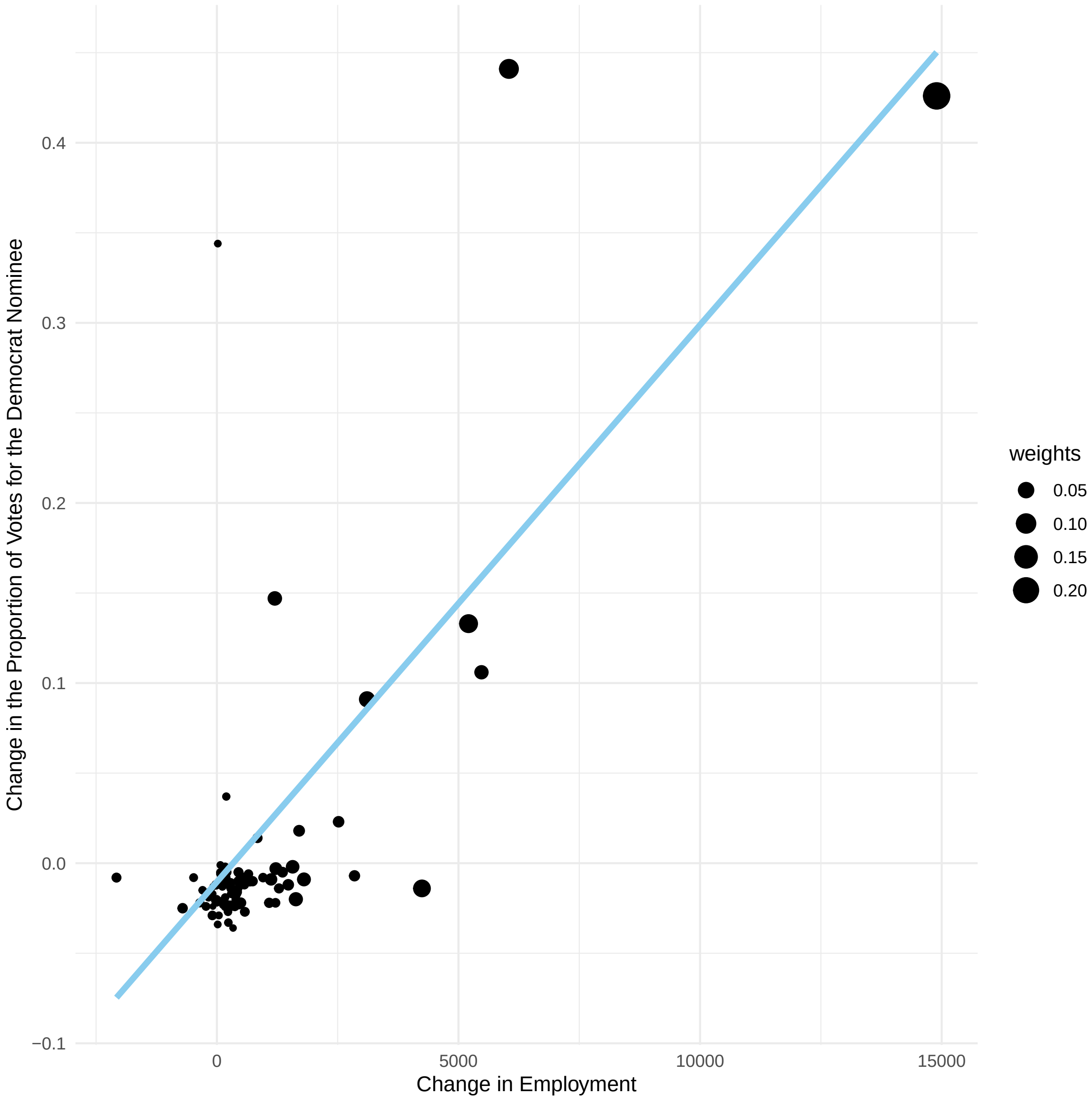
Your turn: Specify the linear model

```r
model <- lm(election_differences ~ e_differences, weights=weights)
```

Let's graph our results with the following `ggplot` code.

```r
library(ggplot2)

ggplot(mapping=aes(x = e_differences, y = election_differences))+
  geom_point(aes(size = weights))+
  geom_line(aes(y = model$fitted.values), color="#88CCEE", linewidth=1.5)+
  theme_minimal()+
  labs(
    x = "Change in Employment",
    y = "Change in the Proportion of Votes for the Democrat Nominee"
  )
```



What do we see?

The simple linear regression model may not be the best at estimating this relationship (misspecification). There is also plenty of room for *endogeneity*—meaning, that there are other confounding variables that are driving this relationship other than the change in employment alone; Small counties tended to vote Republican and Larger counties tended to vote Democrat.

From a practical standpoint, most counties voted for Trump regardless of the change in employment. And most counties experienced some kind of employment growth. The counties where the employment rate mattered the most were in \large counties—and in those counties, the better the economy did over the past four years the more people, on average, voted more blue.