Lecture Assignment 18

Viraj Vijaywargiya

2022-06-01

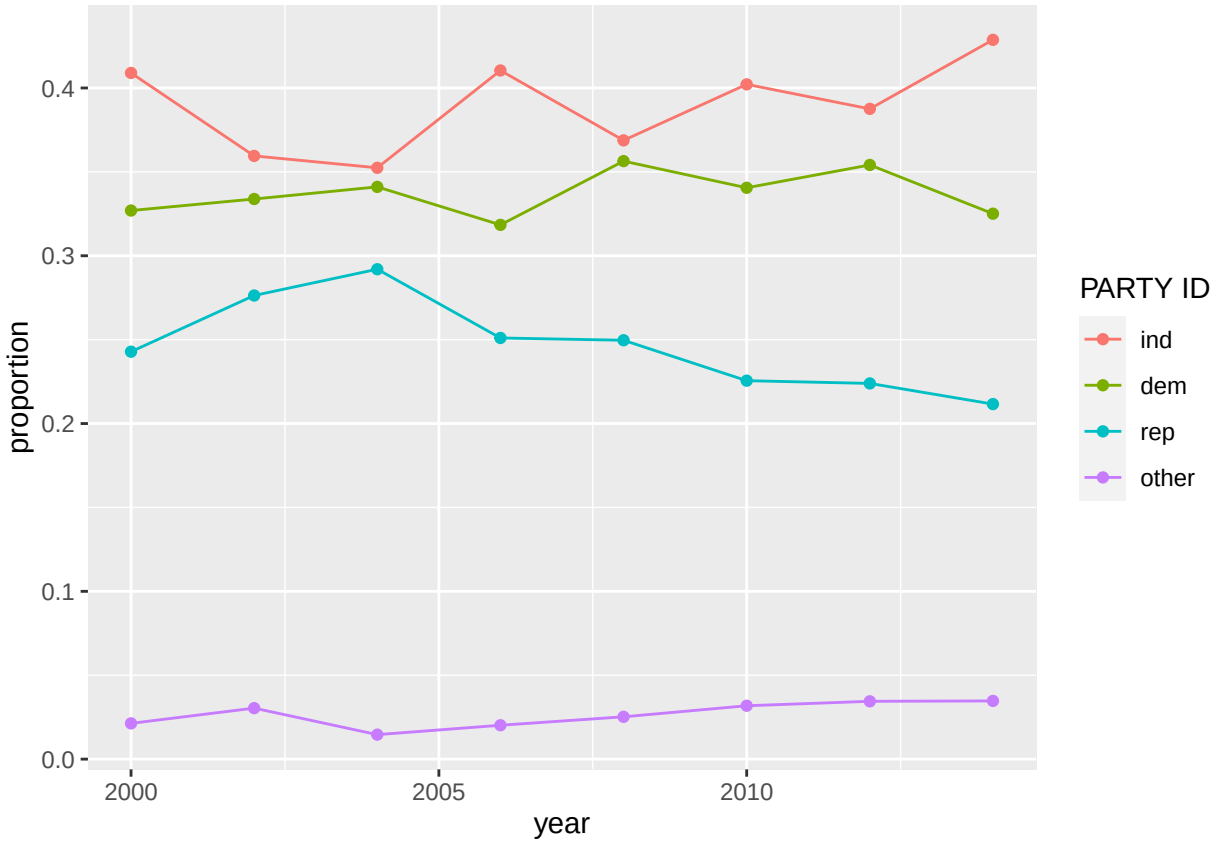
```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.8
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

Part 15.5.1

Question 1

```
gss_cat %>%
  mutate(partyid = fct_collapse(partyid,
    other = c("No answer", "Don't know", "Other party"),
    rep = c("Strong republican", "Not str republican"),
    ind = c("Ind,near rep", "Independent", "Ind,near dem"),
    dem = c("Not str democrat", "Strong democrat")
  )) %>%
  count(year, partyid) %>%
  group_by(year) %>%
  mutate(proportion = n / sum(n)) %>%
  ggplot(aes(x = year, y = proportion, color = fct_reorder2(partyid, year, proportion))) +
  geom_point() +
  geom_line() +
  labs(colour = "PARTY ID")
```



Part 19.2.1

Question 1

```
rescale01F <- function(x){
  rng <- range(x, na.rm = FALSE, finite = TRUE)
  (x - rng[1]) / (rng[2] - rng[1])}

rescale01T <- function(x){
  rng <- range(x, na.rm = TRUE, finite = TRUE)
  (x - rng[1]) / (rng[2] - rng[1])}

x <- c(NA, Inf, 1:10)
rescale01F(x)
```

```
##  [1]        NA       Inf 0.0000000 0.1111111 0.2222222 0.3333333 0.4444444
##  [8] 0.5555556 0.6666667 0.7777778 0.8888889 1.0000000
```

```
rescale01T(x)
```

```
##  [1]        NA       Inf 0.0000000 0.1111111 0.2222222 0.3333333 0.4444444
##  [8] 0.5555556 0.6666667 0.7777778 0.8888889 1.0000000
```

```
#rescale01(c(NA, 1:10), na.rm = TRUE)
```

If x contained a single missing value and na.rm was FALSE, then the function still returns non-missing value. In both cases, na.rm being TRUE or FALSE, the function returns the same output. Therefore, TRUE does not need to be a parameter to rescale01() as it does not make a difference in the output.

Question 2

```
rescale01 <- function(x){
  rng <- range(x, na.rm = TRUE, finite = TRUE)
  y <- (x - rng[1]) / (rng[2] - rng[1])
  y[y == -Inf] <- 0
  y[y == Inf] <- 1
  y
}

rescale01(c(Inf, -Inf, 0:10, NA))
```

```
##  [1] 1.0 0.0 0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0  NA
```

Question 3

Function 1,

```
mn <- function(x){
  mean(is.na(x))
}
```

Function 2,

```
sum1 <- function(x, na.rm = FALSE) {
  x / sum(x, na.rm = na.rm)
}
```

Function 3,

```
var <- function(x, na.rm = TRUE) {
  sd(x, na.rm = na.rm) / mean(x, na.rm = na.rm)
}
```

Part 19.3.1

Question 1

```
f1 <- function(string, prefix) {
  substr(string, 1, nchar(prefix)) == prefix
}
f2 <- function(x) {
  if (length(x) <= 1) return(NULL)
  x[-length(x)]
}
f3 <- function(x, y) {
  rep(y, length.out = length(x))
}
```

The function f1 tests whether the input string starts with the input prefix. A better name for this function could be isPrefix().

The function f2 removes the last element in x. A better name for this function could be removeLast().

The function f3 returns y, length of x times. A better name for this function could be repeat().