Election

Chien-Che Hung 1004330164

10/4/2020

Import the census data

```
# insert the 2016 census report
census <- read_excel("10952320201004121154.xlsx")
```

```
## New names:
## * `` -> ...5
```

Simulating the data from the census data

```
# Find the ratio of sex and age group
# male age range 18-24
male_18_24 <- census[26, ]$Male + census[27, ]$Male + census[28, ]$Male
# female age range 18-24
female_18_24 <- census[26, ]$Female + census[27, ]$Female + census[28, ]$Female
# male age range 25-34
male_25_34 <- census[34, ]$Male + census[40, ]$Male
# female age range 25-34
female_25_34 <- census[34, ]$Female + census[40, ]$Female
# male age range 35-44
male_35_44 <- census[46, ]$Male + census[52, ]$Male
# female age range 35-44
female_35_44 <- census[46, ]$Female + census[52, ]$Female
# male age range 45-54
male_45_54 <- census[58, ]$Male + census[64, ]$Male
# female age range 45-54
female_45_54 <- census[58, ]$Female + census[64, ]$Female
# male age range 55-64
male_55_64 <- census[70, ]$Male + census[76, ]$Male
# female age range 55-64
female_55_64 <- census[70, ]$Female + census[76, ]$Female
# male age range 65+
male_65p <- census[82, ]$Male
# female age range 65+
female_65p <- census[82, ]$Female

#combine different age group and sex into one
male <- c(male_18_24, male_25_34, male_35_44, male_45_54, male_55_64, male_65p)
female <- c(female_18_24, female_25_34, female_35_44,female_45_54, female_55_64, female_65p)
```

```r
total <- male + female

# calculate the ratio for male and female
ratio_total <- total/sum(total)
sex_ratio_male <- male/total
sex_ratio_female <- female/total
sample_size_total <- round(2000*ratio_total)
sample_size_total[1] <- sample_size_total[1]+1
sample_size_male <- round(sample_size_total*sex_ratio_male)
sample_size_female <- round(sample_size_total*sex_ratio_female)
ratio_male_age <- male/sum(male)
ratio_female_age <- female/sum(female)
demo_table <- tibble(male, ratio_male_age, sex_ratio_male, sample_size_male, female, ratio_female_age, s
rownames(demo_table) <- c("18-24", "25-34", "35-44", "45-54", "55-64", "65+")
```

```
## Warning: Setting row names on a tibble is deprecated.
```

```r
demo_table <- as.data.frame(demo_table)

# after finding the ratio and the number of male, we sample the amount of age and sex according to the
age <- c(replicate(demo_table$sample_size_male[1], "18-24"))
sex <- c(replicate(demo_table$sample_size_male[1], "Male"))
age_table <- as.data.frame(cbind(age, sex))
age <- c(replicate(demo_table$sample_size_male[2], "25-34"))
sex <- c(replicate(demo_table$sample_size_male[2], "Male"))
age_table <- rbind(age_table, cbind(age, sex))
age <- c(replicate(demo_table$sample_size_male[3], "35-44"))
sex <- c(replicate(demo_table$sample_size_male[3], "Male"))
age_table <- rbind(age_table, cbind(age, sex))
age <- c(replicate(demo_table$sample_size_male[4], "45-54"))
sex <- c(replicate(demo_table$sample_size_male[4], "Male"))
age_table <- rbind(age_table, cbind(age, sex))
age <- c(replicate(demo_table$sample_size_male[5], "55-64"))
sex <- c(replicate(demo_table$sample_size_male[5], "Male"))
age_table <- rbind(age_table, cbind(age, sex))
age <- c(replicate(demo_table$sample_size_male[6], "65+"))
sex <- c(replicate(demo_table$sample_size_male[6], "Male"))
age_table <- rbind(age_table, cbind(age, sex))

# after finding the ratio and the number of male, we sample the amount of age and sex according to the
age <- c(replicate(demo_table$sample_size_female[1], "18-24"))
sex <- c(replicate(demo_table$sample_size_female[1], "Female"))
age_table_f <- as.data.frame(cbind(age, sex))
age <- c(replicate(demo_table$sample_size_female[2], "25-34"))
sex <- c(replicate(demo_table$sample_size_female[2], "Female"))
age_table_f <- rbind(age_table_f, cbind(age, sex))
age <- c(replicate(demo_table$sample_size_female[3], "35-44"))
sex <- c(replicate(demo_table$sample_size_female[3], "Female"))
age_table_f <- rbind(age_table_f, cbind(age, sex))
age <- c(replicate(demo_table$sample_size_female[4], "45-54"))
sex <- c(replicate(demo_table$sample_size_female[4], "Female"))
age_table_f <- rbind(age_table_f, cbind(age, sex))
age <- c(replicate(demo_table$sample_size_female[5], "55-64"))
sex <- c(replicate(demo_table$sample_size_female[5], "Female"))
```

```
age_table_f <- rbind(age_table_f, cbind(age, sex))
age <- c(replicate(demo_table$sample_size_female[6], "65+"))
sex <- c(replicate(demo_table$sample_size_female[6], "Female"))
age_table_f <- rbind(age_table_f, cbind(age, sex))
age <- rbind(age_table, age_table_f)

# randomized the order of the data
rows <- sample(nrow(age))
age <- age[rows, ]
colnames(age) <- c("How old are you?", "Please indicate your sex")

# simulate the provinces
provinces <- c("British Columbia","Alberta", "Saskatchewan", "Manitoba", "Ontario",
               "Quebec", "New Brunswick", "Nova Scotia", "Prince-Edward-Island",
               "Newfoundland and Labrador", "Northwest Territories", "Nunavut", "Yukon")
# we are simulating the province according to the 2016 census result
province_ratio <- c(0.1322, 0.1157, 0.0312, 0.0363, 0.3826, 0.2323, 0.0213, 0.0263, 0.0041,
                    0.0148, 0.0012, 0.001, 0.001)
province <- as.data.frame(sample(x = provinces, prob = province_ratio, size = 2000, replace = TRUE))
colnames(province) <- "Which province or territory do you live in?"
data <- as.data.frame(cbind(age, province))
rownames(data) <- c(1:nrow(data))


#import the degree distribution from the 2016 census data
education<- read_csv("98-402-X2016010-11.csv")
```

```
## Parsed with column specification:
## cols(
##   `Geographic code` = col_double(),
##   `Geographic name` = col_character(),
##   `Global non-response rate` = col_double(),
##   `Data quality flag` = col_double(),
##   `Total - Highest certificate, diploma or degree[1]` = col_double(),
##   `No certificate, diploma or degree` = col_double(),
##   `Secondary (high) school diploma or equivalency certificate[2]` = col_double(),
##   `Apprenticeship or trades certificate or diploma[3],[4]` = col_double(),
##   `College, CEGEP or other non-university certificate or diploma` = col_double(),
##   `University certificate or diploma below bachelor level` = col_double(),
##   `University certificate, diploma or degree at bachelor level or above[5]` = col_double()
## )
```

```
degree_list <- c("No certificate, diploma or degree",
                 "Secondary (high) school diploma or equivalency certificate",
                 "Apprenticeship or trades certificate or diploma",
                 "College, CEGEP or other non-university certificate or diploma",
                 "University certificate or diploma below bachelor level",
                 "University certificate, diploma or degree at bachelor level or above")
# from the 2016 census data, we get these probability for each categories listed above
degree_ratio <- c(0.115, 0.237, 0.108, 0.224, 0.031, 0.285)
degree <- as.data.frame(sample(x = degree_list, prob = degree_ratio, size = 2000, replace = TRUE))
colnames(degree) <- "What is your highest education level?"
data <- as.data.frame(cbind(data, degree))
```

```r
# sample of whether people are following the upcoming election (Based on the turnout rate in)
answers <- c("Yes", "No")
follow_ratio <- c(0.6595, 1-0.6595)
follow_election <- as.data.frame(sample(x = answers, prob = follow_ratio, size = 2000, replace = TRUE))
colnames(follow_election) <- "Have you been following the election?"
data <- as.data.frame(cbind(data, follow_election))

# sample how much people value their alignment with the candidates
n <- 2000
align_prob <- c(0, 0.0, 0.01, 0.015, 0.020, 0.025, 0.05, 0.15, 0.38, 0.35)
align <- as.data.frame(sample(c(1:10), size = 2000, prob = align_prob, replace = TRUE))
colnames(align) <- "How much do you value the alignment between you and the candidate's policies?"
data <- as.data.frame(cbind(data, align))

# sample Satisfaction
sat_list <- c("Very Satisfied", "Somewhat Satisfied", "Somewhat Dissatisfied",
              "Very Dissatisfied", "No Opinion")
sat_prob <- c(0.05, 0.45, 0.3, 0.15, 0.05)
sat <- as.data.frame(sample(sat_list, size = 2000, prob = sat_prob, replace = TRUE))
colnames(sat) <- "How's your satisfaction with the current Canadian government?"
data <- as.data.frame(cbind(data, sat))

# sample COVID response
response <- as.data.frame(c(10-c(rpois(2000, 3))))
colnames(response) <- "How would you rate the current government's response on combating COVID-19?"
data <- as.data.frame(cbind(data, response))

# most important issue
issues <- c("The Economy/Jobs", "Environment", "Health Care",
            "Housing/Homelessness/Proverty",
            "Accountability/Leadership", "Energy/Piplines", "Crime/Public Safety", "Foreign Affairs")
issue_ratio <- c(0.35, 0.10, 0.25, 0.10, 0.10, 0.03, 0.02, 0.05)
issue <- as.data.frame(sample(issues, size = 2000, prob = issue_ratio, replace = TRUE))
colnames(issue) <- "In your opinion, what is the most important issue facing in Canada today?"
data <- as.data.frame(cbind(data, issue))

# sample canadidates
cand_list <- c("Justin Trudeau's Liberal Party of Canada",
               "Jagmeet Singh's New Democratic Party of Canada",
               "Erin O'Toole's Conservative Party of Canada",
               "Annamie Paul's Green Party of Canada",
               "Yves-Francois Blanchet's Bloc Québécois",
               "Others")

# we obtained these through CBC polling updates
cand_prob <- c(0.364, 0.177, 0.310, 0.059, 0.068, 0.022)
cand <- as.data.frame(sample(cand_list, size = 2000, prob = cand_prob, replace = TRUE))
colnames(cand) <- "If the Federal Election were being held today, who would you vote for?"
data <- as.data.frame(cbind(data, cand))

# sampel timestamp
u <- runif(2000, 0, 20) # "noise" to add or subtract from some timepoint
timestamp <- as.data.frame(as.POSIXlt(u, origin = "2020-10-07 08:00:00"))
```

```r
colnames(timestamp) <- "Timestamp"
data <- as.data.frame(cbind(data, timestamp))

data <- data %>%
  select("Timestamp", everything())
```

Export the dataframe

```r
# write the dataframe into csv and xlsx format to make it accessible in other fiels
write.csv(data, "/Users/frankhung/Documents/FHDocument/2020_2021/STA304/Assignments/Assignment\ 2/electi
write_xlsx(data, "/Users/frankhung/Documents/FHDocument/2020_2021/STA304/Assignments/Assignment\ 2/elec
```

Reference

- Hadley Wickham and Jennifer Bryan (2019). readxl: Read Excel Files. R package version 1.3.1. https://CRAN.R-project.org/package=readxl
- Hadley Wickham, Romain François, Lionel Henry and Kirill Müller (2020). dplyr: A Grammar of Data Manipulation. R package version 1.0.2. https://CRAN.R-project.org/package=dplyr
- H. Wickham. ggplot2: Elegant Graphics for Data Analysis. Springer-Verlag New York, 2016.
- Hadley Wickham, Jim Hester and Romain Francois (2018). readr: Read Rectangular Text Data. R package version 1.3.1. https://CRAN.R-project.org/package=readr
- Jeroen Ooms (2020). writexl: Export Data Frames to Excel 'xlsx' Format. R package version 1.3.1. https://CRAN.R-project.org/package=writexl
- R Core Team (2020). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.
- JJ Allaire and Yihui Xie and Jonathan McPherson and Javier Luraschi and Kevin Ushey and Aron Atkins and Hadley Wickham and Joe Cheng and Winston Chang and Richard Iannone (2020). rmarkdown: Dynamic Documents for R. R package version 2.3. URL https://rmarkdown.rstudio.com.
- Yihui Xie and J.J. Allaire and Garrett Grolemund (2018). R Markdown: The Definitive Guide. Chapman and Hall/CRC. ISBN 9781138359338. URL https://bookdown.org/yihui/rmarkdown.