Problem set 3

Siwei Dai

October 15, 2021

NOTE1: Start with the file `ps3_2021.Rmd` *(available from the github repository at https://github.com/UChicago-pol-methods/IntroQSS-F21/tree/main/assignments). Modify that file to include your answers. Make sure you can "knit" the file (e.g. in RStudio by clicking on the* `Knit` *button). Submit both the Rmd file and the knitted PDF via Canvas*

NOTE2: You will need to have a working LaTeX installation to compile your code.

Question 1:

Consider the random process of flipping a fair coin three times.

(1a) Describe the sample space, Ω.

$$\Omega = \{(HHH), (HHT), (HTT), (HTH), (THH), (THT), (TTH), (TTT)\}$$

(1b) The random variable X that we're interested is the number of heads that we get from our random process. Write out the probability mass function for this random variable. *Hint: the coin is fair, so each of the events in the sample space above occurs with equal probability. Note how many heads we get in each event. Then look at the proportion of times we get no heads, one head, etc. These proportions are equal to the probability. List the number of heads under the x column. List the associated probabilities under the $P(X = x)$ column.*

x	$P(X = x)$
0	1/8
1	3/8
2	3/8
3	1/8

(1c) Calculate the mean of this random variable. Please show your work.

$$\begin{aligned} E[X] &= \sum_x x f(x) \\ &= 0 \times \frac{1}{8} + 1 \times \frac{3}{8} + 2 \times \frac{3}{8} + 3 \times \frac{1}{8} \\ &= \frac{3}{2} \end{aligned}$$

(1c) Write out code to simulate this random process, where the output is a single realization of the random variable (i.e., a number that represents the number of heads in your coin flips).

NOTE3: I set a random seed here, so that every time you recompile your assignment, you'll get the same number. For analyses that involve sampling or random processes, it is really important to set a random seed so that you can get reproducible results. Feel free to change the seed number to anything you want. In general you only need to set your random seed ONCE per script.

```
set.seed(60637)
# your code here
coin_flip <- sum(sample(
    x = c(0, 1),
    size = 3,
    replace = TRUE))
coin_flip
```

```
## [1] 2
```

(1d) Now run your random process so you sample from it 10,000 times [PLEASE DON'T OUTPUT ALL 10,000 OBSERVATIONS IN YOUR HOMEWORK, just save it to an R object]. What is the average number of heads across these 10k observations? This is the sample mean for a given sample.

```
# your code here
omega <- c(0, 1, 2, 3)
probs <- c(1/8, 3/8, 3/8, 1/8)
coin_10k <- sample(
    x = omega,
    size = 10000,
    replace = TRUE,
    prob = probs)
coin_heads <- mean(coin_10k)
coin_heads
```

```
## [1] 1.506
```

(1e): Write your own function called `mymean()` to calculate the sample mean from a vector. Apply your function to your size 10k sample that you saved in the last problem. (Don't use `mean()` inside your function.)

```
# your code here
mymean <- function(x) {
    sum(x) / length(x)
}
mymean(coin_10k)
```

```
## [1] 1.506
```

Question 2:

Using the same random process of flipping three fair coins, code the random variable Y as 1 if we get three heads, and 0 otherwise.

(2a) Write out the probability mass function for this random variable Y.

y	$P(X = x)$
0	7/8
1	1/8

(2a) Write out the joint probability mass function for the joint distribution of X and Y.

x	y	$P(X = x, Y = y)$
0	0	1/8
1	0	3/8
2	0	3/8
3	1	1/8

(2b) Write out the probability mass function for this random variable X *conditional* on Y.

x	y	$P(X = x \mid Y = y)$
0	0	1/7
1	0	3/7
2	0	3/7
3	1	1

Question 3:

(3a) Load the data set that you selected for your independent project. If your data set is not already in tibble format, transform it into a tibble. Print the data set so that we can see the top few observations and the column names and types.

The variables are in Chinese; essentially the variables included the name of the bureaucrat, when a particular expericence started and ended, where he assumed the office and the level of the office.

```
# your code here
datapath <- './ps_data/FullData.csv'
df <- as.tibble(read.csv(datapath))
```

```
## Warning: `as.tibble()` was deprecated in tibble 2.0.0.
## Please use `as_tibble()` instead.
## The signature and semantics have changed, see `?as_tibble`.
```

```
head(df)
```

```
## # A tibble: 6 x 25
##                 .YYYY.MM.~    ..YYYY.MM.~
##      <int> <chr>     <int> <chr>              <chr>                <chr>
## 1        1           1 1974/6/1           1976/12/1
## 2        1           2 1976/12/1          1978/3/1
## 3        1           3 1978/3/1           1978/10/1
## 4        1           4 1978/10/1          1980/6/1
## 5        1           5 1980/6/1           1984/8/1
```

```
## 6        1           6 1984/8/1           1987/9/1
## # ... with 19 more variables:     <chr>,      <int>,
## #        <chr>,      <int>,      <chr>,
## #      <chr>,     <lgl>,      <chr>,     <chr>,
## #        <int>,      <chr>,     .1 <lgl>,
## #        <chr>,      <chr>,    <chr>,   <chr>,
## #      <chr>,   <chr>,   <int>
```

(3b) What do you think is the appropriate unit of observation in your data? Is your data set already formatted so that each row describes a unique unit of observation? If not, what does each row describe? I want to investigate why certain bureaucrats, after being removed from office or being demoted, can be re-promoted while others didn't. An appropriate unit of observation should be the bureaucrat, with variables denoting if he has been demoted, if after demotion he has been promoted again, and personal variables (connections with upper-level elites, alma mater, etc.)

Currently, each row denotes one experience within a political elite's career, i.e. what one's job was at a given year. After adding new variables (if he has been demoted, if after demotion he has been promoted again) by mutating, and filtering the observations, the dataset would be ready.

[**Extra credit:** if your data set will need to be reshaped using `pivot_longer()` or `pivot_wider()`, try reshaping it now. If it doesn't need to be reshaped, you can try reshaping it anyhow. Give your reshaped columns informative names. Explain what the unit of observation is in your reshaped data set.]

```
# The following reshaping is to integrate each observation from one career experience
# to an observation on one bureacrat, with extra variables denoting the position he had
# at each career stage
df_tidy <- df %>%
    # translate the variables into English
    # exper_num refers to the career stage
    rename('name' =  ,
           'position_ori' =  ,
           'exper_num' =   ) %>%
    select(name, position_ori, exper_num) %>%
    # recode the position from strings into numeric
    mutate('position_numeric' = case_when(position_ori == '  ' ~ 0,
                                          position_ori == '   ' ~ 1,
                                          position_ori == ' ' ~ 2,
                                          position_ori == ' ' ~ 3,
                                          position_ori == ' ' ~ 4,
                                          position_ori == ' ' ~ 5,
                                          position_ori == ' ' ~ 6,
                                          position_ori == ' ' ~ 7,
                                          position_ori == ' ' ~ 8,
                                          position_ori == ' ' ~ 9)) %>%
    select(name, position_numeric, exper_num) %>%
    pivot_wider(
        names_from = exper_num,
        values_from = position_numeric)
df_tidy
```

```
## # A tibble: 3,923 x 38
##    name    `1`   `2`   `3`   `4`   `5`   `6`   `7`   `8`   `9`  `10`  `11`  `12`
##    <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
##  1  ~     1     1     1     1     1     2     2     3     3     3     4     4
```

```
##  2  ~     0     0     0     1     1     1     2     2     2     2     2     2
##  3        0     0     1     1     0     2     2     3     3     3     4     5
##  4  ~     1     1     1     1     2     2     2     2     3     3     4     4
##  5  ~     0     0     0     0     1     1     2     2     3     3     3     3
##  6  ~     0     0     1     1     1     1     1     2     2     3     4     4
##  7  ~     0     0     0     0     0     0     0     1     0     0     1     3
##  8  ~     1     1     1     1     2     3     3     3     3     3     3     3
##  9  ~     0     1     1     1     2     2     3     3     4     4     4     5
## 10        0     0     1     1     1     2     2     2     3     4     4     4
## # ... with 3,913 more rows, and 25 more variables: 13 <dbl>, 14 <dbl>,
## #   15 <dbl>, 16 <dbl>, 17 <dbl>, 18 <dbl>, 19 <dbl>, 20 <dbl>, 21 <dbl>,
## #   22 <dbl>, 23 <dbl>, 24 <dbl>, 25 <dbl>, 26 <dbl>, 27 <dbl>, 28 <dbl>,
## #   29 <dbl>, 30 <dbl>, 31 <dbl>, 32 <dbl>, 33 <dbl>, 34 <dbl>, 35 <dbl>,
## #   36 <dbl>, 37 <dbl>
```