Extra 3

Seung Min Song & Ted Kim

2022-10-18

GitHub: https://github.com/seung-m1nsong/607
rpubs: https://rpubs.com/seungm1nsong/958363

read data Instead of using the header, and load the csv file with making empty cells to NA.

```
csv <- read.csv(file = 'https://raw.githubusercontent.com/blacksmilez/DATA607/main/extra2/vaccine.csv',
```

transform for Population
Select 1st(Age), 2nd(Population % Not Vax), and 3rd(Population % Fully Vax) column using *select()* function. Use *fill()* function in tidyr package to fill empty cells in the first column with above value. Use *add_column()* function in tibble package to add column *Population* after the first column. Use *slice()* function to remove the first row and insert it into a data frame, pop.

```
pop <- csv %>%
        select(c(1, 2, 3)) %>%
        fill(V1, .direction = "down") %>%
        add_column(V4 = str_replace(.[1,2], '\\s%', ''), .after = 'V1') %>%
        slice(-c(1))
```

transform for Population %
In order to transform the population % use the *slice()* function to bring the first raw and odd rows. use *mutate_at( )*, *vars( )*, *str_replace( )*, and *str_remove( )* function to massage 2nd(def), 3rd(state), and 4th(number of percent).
Remove the first row after defining the column name using the value of the first row. Pivot after converting the percentage value in character data type to numeric data type.

```
pop_p <- slice(pop, c(1, 3, 5)) %>%
            mutate_at(vars(3, 4), list(~str_remove(., '\\\n%'))) %>%
            mutate_at(vars(2), list(~paste0(., ' %')))

colnames(pop_p) = pop_p[1,] %>%
            mutate_at(vars(2), list(~str_replace(., '.*', 'def'))) %>%
            str_replace_all('\\s', '_')
```

```r
pop_p <- pop_p %>%
         slice(-c(1))  %>%
         mutate_at(vars(3, 4), list(~as.numeric(str_remove(., '%'))/100)) %>%
         pivot_longer(Not_Vax:Fully_Vax, names_to = 'State', values_to = 'Num')
pop_p
```

```
## # A tibble: 4 x 4
##   Age   def          State       Num
##   <chr> <chr>        <chr>     <dbl>
## 1 <50   Population % Not_Vax   0.233
## 2 <50   Population % Fully_Vax 0.73
## 3 >50   Population % Not_Vax   0.079
## 4 >50   Population % Fully_Vax 0.904
```

transform for Population

In order to transform the population use the *slice()* function to bring the first raw and even rows. use *mutate_at( )*, *vars( )*, and *str_remove( )* function to massage 3rd(state), and 4th(number of percent). Remove the first row after defining the column name using the value of the first row. Pivot after remove , in the column three(Not Vax) and four(Fully Vax) and convert to numeric data type.

```r
pop <- slice(pop, c(1, 2, 4)) %>%
         mutate_at(vars(3, 4), list(~str_remove(., '\\\n%')))

colnames(pop) = pop[1,]  %>%
         mutate_at(vars(2), list(~str_replace(., '.*', 'def'))) %>%
           str_replace_all('\\s', '_')

pop <- pop %>%
       slice(-c(1)) %>%
       mutate_at(vars(3, 4), list(~as.numeric(gsub(',', '', .)))) %>%
       pivot_longer(Not_Vax:Fully_Vax, names_to = 'State', values_to = 'Num')

pop
```

```
## # A tibble: 4 x 4
##   Age   def        State         Num
##   <chr> <chr>      <chr>       <dbl>
## 1 <50   Population Not_Vax   1116834
## 2 <50   Population Fully_Vax 3501118
## 3 >50   Population Not_Vax    186078
## 4 >50   Population Fully_Vax 2133516
```

transform for Severe Cases

Select 1st(Age), 4th(Not Vax), and 5th(Fully Vax) column using *select()* function. Use *fill()* function in tidyr package to fill empty cells in the first column with value above. Use *add_column()* function in tibble package to add column *Severe Cases per 100K* after the first column. Use *slice()* function to remove the first, forth and sixth row and insert it into a data frame, sev. Remove the first row after defining the column name using the value of the first row. Pivot after converting the character data type value to numeric data type.

```
sev <- csv %>%
          select(c(1, 4, 5)) %>%
          fill(V1, .direction = "down") %>%
          add_column(V6 = paste0(.[1,2], ' per 100K'), .after = 'V1') %>%
          slice(-c(1, 4, 6))

colnames(sev) = sev[1,] %>%
                  mutate_at(vars(3, 4), list(~str_remove_all(., '\\\n.*'))) %>%
                  mutate_at(vars(2), list(~str_replace(., '.*', 'def'))) %>%
                  str_replace_all('\\s', '_')

sev <- sev %>%
        slice(-c(1)) %>%
        mutate_at(vars(3, 4), list(~as.numeric(.))) %>%
        pivot_longer(Not_Vax:Fully_Vax, names_to = 'State', values_to = 'Num')

sev
```

```
## # A tibble: 4 x 4
##   Age   def                   State       Num
##   <chr> <chr>                 <chr>     <dbl>
## 1 <50   Severe Cases per 100K Not_Vax      43
## 2 <50   Severe Cases per 100K Fully_Vax    11
## 3 >50   Severe Cases per 100K Not_Vax     171
## 4 >50   Severe Cases per 100K Fully_Vax   290
```

```
df <- as.data.frame(union(pop, pop_p) %>%
                    union(sev))

df
```

```
##    Age                   def     State         Num
## 1  <50            Population   Not_Vax 1116834.000
## 2  <50            Population Fully_Vax 3501118.000
## 3  >50            Population   Not_Vax  186078.000
## 4  >50            Population Fully_Vax 2133516.000
## 5  <50          Population %   Not_Vax       0.233
## 6  <50          Population % Fully_Vax       0.730
## 7  >50          Population %   Not_Vax       0.079
## 8  >50          Population % Fully_Vax       0.904
## 9  <50 Severe Cases per 100K   Not_Vax      43.000
## 10 <50 Severe Cases per 100K Fully_Vax      11.000
## 11 >50 Severe Cases per 100K   Not_Vax     171.000
## 12 >50 Severe Cases per 100K Fully_Vax     290.000
```

1. **Total Population**
   Do you have enough information to calculate the total population? What does this total population represent?

Populations in the table represent those who responded to being vaccinated. Therefore, it is necessary to consider the population who did not respond to both sides to know the results of the vaccination and non-vaccination status of the entire population. Use the sum of population and sum of population % to identify the total population for two age groups.

$$TotalPopulation = NumofApplicablePopulation/percentageofApplicablePopulation$$

```
df %>%
  group_by(Age) %>%
  summarize(
    Toral_Population =sum(Num[def == 'Population']) / sum(Num[def == 'Population %'])
  )
```

```
## # A tibble: 2 x 2
##   Age   Toral_Population
##   <chr>            <dbl>
## 1 <50           4795381.
## 2 >50           2359709.
```

The total population under the age of 50 is 4,795,381, and the total population over the age of 50 is 2,359,709. Thus, the total population covered by the vaccine in Israel is 4,795,381 + 2,359,709 = 7,155,090.

2. **Efficacy vs. Disease**
   Calculate the Efficacy vs. Disease; Explain your results.

According to the formula below, when the percentage of *Fully Vax* increases and the percent of *Not Vax* decreases, *Efficacy vs. Severe disease* has a negative value. If you look at the results, **0.744** for those under 50, and **-0.696** for those over 50. In other words, the vaccine was 74% effective in the younger population, but the vaccine was not as effective in the elderly.

$$Efficacyvs.severedisease = 1 - (\%fullyvaxedseverecasesper100K/\%notvaxedseverecasesper100K)$$

```
df %>%
  filter(def == 'Severe Cases per 100K') %>%
   group_by(Age) %>%
   summarize(
     Efficacy_vs_Severe_Disease = 1 - (Num[State == 'Fully_Vax'] / Num[State == 'Not_Vax'])
   )
```

```
## # A tibble: 2 x 2
##   Age   Efficacy_vs_Severe_Disease
##   <chr>                      <dbl>
## 1 <50                        0.744
## 2 >50                       -0.696
```

3. From your calculation of efficacy vs. disease, are you able to compare the rate of severe cases in unvaccinated individuals to that in vaccinated individuals?

A value toward -99 means that the vaccine is ineffective, and a value toward 1 means that the vaccine is effective. It is unlikely, but in severe cases, if the percent of *Fully Vax* increases as much as possible and approaches 100, and the percent of *Not Vax* decreases as much as possible and approaches 0, *Efficacy vs. The Severe Disease* value converges to about -99. Conversely, if the percent of *Fully Vax* is reduced as much as possible and the percent of *Not Vax* is increased as much as possible, *Efficacy vs. The value of severe disease* converges to 1.

Conclusion

For vaccinated people under the age of 50, the probability of developing the severe disease was reduced by 74% compared to those who aren't vaccinated. However, for those over 50, when they were vaccinated, they were 70% more likely to become seriously ill than when they aren't vaccinated. Errors are caused by people who do not respond.