Milestone #3

Virginia Chan, Patrick Traynor, Antoinette Stetzenmeyer

11/4/2021

This is the code used to load the two data sets of interest: cox_vax_admin.cvs and ca_county_demographics.csv. The first 10 rows are provided to provide a sample of the variables and values

```
library(readr)
cov_vax_admin <- read_csv("cov_vax_admin.csv")
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   X1 = col_double(),
##   as_of_date = col_character(),
##   zip_code_tabulation_area = col_double(),
##   local_health_jurisdiction = col_character(),
##   county = col_character(),
##   vaccine_equity_metric_quartile = col_double(),
##   vem_source = col_character(),
##   age12_plus_population = col_double(),
##   persons_fully_vaccinated = col_double(),
##   persons_partially_vaccinated = col_double(),
##   redacted = col_character()
## )
```

```
head(cov_vax_admin)
```

```
## # A tibble: 6 x 11
##      X1 as_of_date zip_code_tabulat~ local_health_jur~ county  vaccine_equity_m~
##   <dbl> <chr>                  <dbl> <chr>             <chr>                <dbl>
## 1     1 1/5/2021               92703 ORANGE            ORANGE                   1
## 2     2 1/5/2021               92285 SAN BERNARDINO    SAN BE~                  1
## 3     3 1/5/2021               92284 SAN BERNARDINO    SAN BE~                  1
## 4     4 1/5/2021               92275 IMPERIAL          IMPERI~                  1
## 5     5 1/5/2021               92532 RIVERSIDE         RIVERS~                  3
## 6     6 1/5/2021               92376 SAN BERNARDINO    SAN BE~                  1
## # ... with 5 more variables: vem_source <chr>, age12_plus_population <dbl>,
## #   persons_fully_vaccinated <dbl>, persons_partially_vaccinated <dbl>,
## #   redacted <chr>
```

```
ca_county_demographics <- read_csv("ca_county_demographics.csv")
```

```
## Warning: Missing column names filled in: 'X1' [1]
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   .default = col_double(),
```

```
##   name = col_character()
## )
## i Use `spec()` for the full column specifications.
```

```
head(ca_county_demographics)
```

```
## # A tibble: 6 x 23
##      X1 name   pop2012 pop12_sqmi  white  black ameri_es  asian hawn_pi hispanic
##   <dbl> <chr>    <dbl>      <dbl>  <dbl>  <dbl>    <dbl>  <dbl>   <dbl>    <dbl>
## 1     1 Kern    851089     104.   5.00e5  48921    12676 3.48e4    1252   413033
## 2     2 Kings   155039     111.   8.30e4  11014     2562 5.62e3     271    77866
## 3     3 Lake     65253      49.1  5.20e4   1232     2049 7.24e2     108    11088
## 4     4 Lassen   35039       7.42 2.55e4   2834     1234 3.56e2     165     6117
## 5     5 Los A~ 9904341    2423.   4.94e6 856874    72828 1.35e6   26094  4687889
## 6     6 Madera  153025      71.1  9.45e4   5629     4136 2.80e3     162    80992
## # ... with 13 more variables: other <dbl>, mult_race <dbl>, males <dbl>,
## #   females <dbl>, med_age <dbl>, households <dbl>, families <dbl>,
## #   hse_units <dbl>, ave_fam_sz <dbl>, vacant <dbl>, owner_occ <dbl>,
## #   renter_occ <dbl>, county_fips <dbl>
```

This is to identify all of the unique dates in the cov_vax_admin.csv data frame. It shows cumulative totals, that is, prior to, the date provided.

```
unique(cov_vax_admin$as_of_date)
```

```
##  [1] "1/5/2021"  "1/12/2021" "1/19/2021" "1/26/2021" "2/2/2021"  "2/9/2021"
##  [7] "2/16/2021" "2/23/2021" "3/2/2021"  "3/9/2021"  "3/16/2021" "3/23/2021"
## [13] "3/30/2021" "4/6/2021"  "4/13/2021" "4/20/2021" "4/27/2021" "5/4/2021"
## [19] "5/11/2021" "5/18/2021" "5/25/2021" "6/1/2021"  "6/8/2021"  "6/15/2021"
## [25] "6/22/2021" "6/29/2021" "7/6/2021"  "7/13/2021" "7/20/2021" "7/27/2021"
## [31] "8/3/2021"  "8/10/2021" "8/17/2021" "8/24/2021" "8/31/2021" "9/7/2021"
## [37] "9/14/2021"
```

We can see that the first date is January 5, 2021 and the final date is September 14, 2021.

First Task: Subset rows or columns, as needed

```
#Antoinette: Subset cov_vax_admin dataset by county and recent dates (September 2021)
total_age12andabove_california_county<- cov_vax_admin %>%
  select(c(as_of_date, zip_code_tabulation_area, county, vaccine_equity_metric_quartile, age12_plus_popu
  filter(as_of_date ==  "9/7/2021"| as_of_date == "9/14/2021") %>%
  group_by(county) %>%
  arrange(county)%>%
summarize(total_age12andabove = sum(age12_plus_population,na.rm = TRUE), median_of_age_12_and_above = me
```

We would like to answer the question, "By how much has the monthly vaccination rate been increasing in LA County as of January 1, 2021?" First we need to select only pertinent variables (columns). We need the as_of_date, county, age12_plus_population, persons_fully_vaccinated, persons_partially_vaccinated variables. We also need to filter out all other counties but keep Los Angeles.

```
#Patrick: Research Question: By how much has the monthly vaccination rate been increasing in LA County

library("lubridate")
age12plus_la_monthly <- cov_vax_admin %>%
  select(as_of_date, county, age12_plus_population, persons_fully_vaccinated, persons_partially_vaccinat
  drop_na(persons_fully_vaccinated) %>%
  filter(county == "LOS ANGELES")
```

Second Task: Create new variables needed for analysis (minimum 2)

New variables should be created based on existing columns; for example Calculating a rate, Combining character strings Etc If no new values are needed for final tables/graphs, please create 2 new variables anyway We needed to create another variable for the rate fully vaccinated. Also, for interest we will create a variable for the rate partially vaccinated. Additionally, we created a column month.

```
#Patrick - we created month, rate,
age12plus_la_monthly <- cov_vax_admin %>%
  select(as_of_date, county, age12_plus_population, persons_fully_vaccinated, persons_partially_vaccinat
  #this is part of cleaning - we had some zip codes with NA values in January.
  #this may be due to the vaccine not being accessible in many zip codes at that time
  drop_na(persons_fully_vaccinated) %>%
  filter(county == "LOS ANGELES") %>%
  group_by(as_of_date) %>%
  summarize(persons_fully_vaccinated = sum(persons_fully_vaccinated),
            persons_partially_vaccinated = sum(persons_partially_vaccinated),
            age12_plus_population = sum(age12_plus_population)) %>%
  mutate(as_of_date = mdy(as_of_date)) %>%
  #this is part of cleaning as not all dates in the as_of_date field were arranged in chronological ord
  arrange(as_of_date) %>%
  #we will create months from the as_of_date field.
  mutate(month_name = month(as_of_date, label = T)) %>%
  mutate(month = month(as_of_date, label = F))%>%
  group_by(month_name) %>%
  #after grouping by month above, we then get the cumulative total (max) for each month.
  summarize(persons_fully_vaccinated = max(persons_fully_vaccinated),
            persons_partially_vaccinated = max(persons_partially_vaccinated),
            age12_plus_population = max(age12_plus_population)) %>%
  #finally we add the two rate fields.
  mutate(rate_fully_vax = (persons_fully_vaccinated/age12_plus_population)*100,
         rate_part_vax =
           (persons_partially_vaccinated/age12_plus_population)*100)
```

Third Task Clean variables needed for analysis (minimum 2)

Examples Recode invalid values Handle missing fields Recode categories Etc. If not needed for final analysis, please create at least 2 new variables anyway Although the data were mostly from above. To demonstrate that our team can change variable names to upper or lower cases remove characters, we have included the code below. For the part, we simply changed variable names to upper case and replaced underscores with spaces. Then, for the second code, we changed them back.

Then, this can considered part of cleaning. We changed the month numbers, e.g. 1, 2, 3, to January, February, March.

```
#Patrick: We are creating two variables to clean
#first create variable names in upper case to spaces
capitalized_data_set  <- age12plus_la_monthly %>%
  rename_with(., ~toupper(gsub("_"," ", .x, fixed = T)))
#Patrick: now we are going to clean by changing it back to snake case
capitalized_data_set <- capitalized_data_set %>%
  rename_with(., ~tolower(gsub(" ","_", .x, fixed = T)))

#Patrick: Change abbreviated dates to full names
age12plus_la_monthly_w_fl_mths <- age12plus_la_monthly %>%
  mutate(month_name =
           if_else(month_name == "Jan","January",
                   if_else(month_name == "Feb","February",
                           if_else(month_name == "Mar","March",
                                   if_else(month_name == "Apr","April",
                                           if_else(month_name == "May","May",
                                                   if_else(
    month_name == "Jun","June",
    if_else(month_name == "Jul","July",if_else(month_name == "Aug","August",if_else(month_name == "Sep"
```

Forth Task: Data dictionary based on clean dataset (minimum 4 data elements), including:

Variable name Data type Description

Below is a data dictionary descibing all the variables used for this analysis.

```
#Patrick this is format. We will professionalize it later perhaps using kable
data_dict_age12_pl_w_mon <- data.frame(
Variable_Name = c("month_name","persons_fully_vaccinated",
                  "persons_partially_vaccinated",
                  "age12_plus_population",
                  "rate_fully_vax",
                  "rate_part_vax"),
                           Data_Type = c("Character", rep("Double", 5)),
                           Description = c("Month Name",
  "Cumulative number of vaccinated people",
  "Cumulative number of partially vaccinated people",
  "Number of eligible people to be vaccinated as of the given date",
  "Percent of eligible population that is fully vaccinated",
  "Percent of partially vaccinated people"), stringsAsFactors = F)
head(age12plus_la_monthly_w_fl_mths)
```

```
## # A tibble: 6 x 6
##   month_name persons_fully_va~ persons_partiall~ age12_plus_popu~ rate_fully_vax
##   <chr>                  <dbl>             <dbl>            <dbl>          <dbl>
## 1 January               130878            529752          8610605           1.52
## 2 February              680317            796415          8613542           7.90
## 3 March                1804871           1375805          8619027.         20.9
## 4 April                3273428           1528052          8619980.         38.0
## 5 May                  4525359           1342363          8619980.         52.5
## 6 June                 5194521            910138          8619980.         60.3
## # ... with 1 more variable: rate_part_vax <dbl>
```

```r
colnames(age12plus_la_monthly_w_fl_mths)
```

```
## [1] "month_name"                   "persons_fully_vaccinated"
## [3] "persons_partially_vaccinated" "age12_plus_population"
## [5] "rate_fully_vax"               "rate_part_vax"
```

```r
  #Patrick: Make this a professionally looking table
library(kableExtra)
```

```
##
## Attaching package: 'kableExtra'
```

```
## The following object is masked from 'package:dplyr':
##
##     group_rows
```

```r
kable(data_dict_age12_pl_w_mon, format = "pipe", booktabs = T, caption = "Data
      Dictionary for COVID-19 Vaccination Rates for LA County from January to
      Mid September")
```

Table 1: Data Dictionary for COVID-19 Vaccination Rates for LA County from January to Mid September

Variable_Name	Data_Type	Description
month_name	Character	Month Name
persons_fully_vaccinated	Double	Cumulative number of vaccinated people
persons_partially_vaccinated	Double	Cumulative number of partially vaccinated people
age12_plus_population	Double	Number of eligible people to be vaccinated as of the given date
rate_fully_vax	Double	Percent of eligible population that is fully vaccinated
rate_part_vax	Double	Percent of partially vaccinated people

Fifth Task: One or more tables with descriptive statistics for 4 data elements

Below is the R code used for showing the number of individuals in LA county who were fully vaccinated in January, 2021 as well as the number fully vaccinated as of mid September (September 14, 2021). These are the minimum and maximum values derived using the summary function in R. We also used the mean function to get the average number of people vaccinated during this time period. We also got the standard deviation using R's sd function.

```
#this shows the minimum number of vaccinated and max # vaccinated
summary(age12plus_la_monthly_w_fl_mths$persons_fully_vaccinated)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  130878 1804871 4525359 3639921 5449573 5913908
```

```
#this shows the mean and standard deviation
mean(age12plus_la_monthly_w_fl_mths$persons_fully_vaccinated)
```

```
## [1] 3639921
```

```
sd(age12plus_la_monthly_w_fl_mths$persons_fully_vaccinated)
```

```
## [1] 2259107
```

```
#create a table of these descriptive statistics using these commands directly in a data.frame function.

data_table_of_descriptives <- data.frame(
  Variable_Names = c("min","max","mean","standard deviation"),
  Persons_fully_vaccinated = c(min(age12plus_la_monthly_w_fl_mths$persons_fully_vaccinated),
  max(age12plus_la_monthly_w_fl_mths$persons_fully_vaccinated),
  mean(age12plus_la_monthly_w_fl_mths$persons_fully_vaccinated),
  sd(age12plus_la_monthly_w_fl_mths$persons_fully_vaccinated)),
  Persons_partially_vaccinated = c(min(age12plus_la_monthly_w_fl_mths$persons_partially_vaccinated),
  max(age12plus_la_monthly_w_fl_mths$persons_partially_vaccinated),
  mean(age12plus_la_monthly_w_fl_mths$persons_partially_vaccinated),
  sd(age12plus_la_monthly_w_fl_mths$persons_partially_vaccinated)),
  Population_12_or_more_of_age = c(min(age12plus_la_monthly_w_fl_mths$age12_plus_population),
  max(age12plus_la_monthly_w_fl_mths$age12_plus_population),
  mean(age12plus_la_monthly_w_fl_mths$age12_plus_population),
  sd(age12plus_la_monthly_w_fl_mths$age12_plus_population)))

kable(data_table_of_descriptives, booktabs = T, format = "pipe", caption = "This shows the minimum, maxi
```

Table 2: This shows the minimum, maximum, mean, and standard deviation for the variables included

Variable_Names	Persons_fully_vaccinated	Persons_partially_vaccinated	Population_12_or_more_of_age
min	130878	529752.0	8610605.000
max	5913908	1528052.0	8620001.200
mean	3639921	976255.6	8618089.156
standard deviation	2259107	347916.6	3501.902

From this we can see that the minimum number is 130,0878. This coincides with our start date in January. The maximum number of 5,913,908 coincides with September.

#6 PDF that is professionally prepared for presentation Each part of the milestone is clearly on one page (use

to push to a new page) Only the necessary information is outputted (you should suppress, for example, entire data frame outputs) Use of headers and sub headers to create an organized document