COVID Vaccination Project

Adrian Cornejo and Victor Guillera

9/30/2021

Section 1

This data set will relate to our question if being a homeowner status and available housing impacts vaccination rates per CA county. While not a perfect representation of SES of cases, this data set will help us elucidate if there is a trend via linear regression between the homeowner-related to renter rates and the vaccination rate per county.

We have two data sets, one is of COVID-19 vaccination progress throughout CA provided by the CDPH following vaccine dosage by zip code from 1/5/21 to about present 9/14/21, and the other is demographic info in CA across all counties from the US census data for 2014 to 2019. Below, we read in our libraries, data sets, explore our data sets, and then proceed to clean the data sets.

```
library(readr)
library(knitr)
library(tibble)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
countydem <- read.csv("ca_county_demographics.csv")
covidvax <- read.csv("cov_vax_admin.csv")
```

To import the csv files, we used read_csv from base R and will inspect our imported files first before we clean our data set. Libraries we loaded in were readr, dplyr, and tibble. We are going to pull variables "persons_fully_vaccinated", "persons_partially_vaccinated", "zip_code_tabulation_area", "age12_plus_population", and "county" from the covidvax csv file. We will pull "name", "renter_occ","owner_occ", "hse_units" and "vacant" variables.

```
str(covidvax)
head(covidvax)
nrow(covidvax)
ncol(covidvax)
typeof(covidvax)
class(covidvax)
```

```
distinct(covidvax, as_of_date, .keep_all = F)
```

The structure of the "cov_vax_admin" file is a data frame with 11 columns and 65,268 rows. The data frame contains integer, number, and character data types and contains a redacted column that is inaccessible. There are also a total of 27 unique dates, each 7 weeks apart (weekly results).

```
str(countydem)
head(countydem)
nrow(countydem)
ncol(countydem)
typeof(countydem)
class(countydem)
```

The structure of "countydem" reveals a data frame containing 23 columns and 58 rows with integer, character and number data types.

```
#Cleaning the data by removing unnecessary columns, rearranging the data,
#and removing "N/A" data points.

#COVID Vaccination Rates
covidvax2 <- select(covidvax, "county","zip_code_tabulation_area", "as_of_date",
                    "age12_plus_population", "persons_fully_vaccinated",
                    "persons_partially_vaccinated")

names(covidvax2) <- c("county", "zip_code", "date", "total_pop_over_12",
                      "fully_vaccinated", "partially_vaccinated")


covidvax2[is.na(covidvax2)] <- 0

#County Data

countydem2 <- select(countydem, "name", "renter_occ","owner_occ",
                     "vacant", "hse_units")

names(countydem2) <-c("county_name","renters","owners",
                      "vacant_housing", "total_housing_units")

countydem2[is.na(countydem2)] <- 0
```

Using the function str() we see that the selected varibales in countydem2 are chr, int, int, int, int. All column types should be fine as is for our study question. Converting NA cells to 0 we allow ourselves to manipulate and describe entire numeric columns. We also changed column names to allow for easier/obvious verbiage.

```
#Descriptions

distinct(covidvax2, county, .keep_all = F)
distinct(covidvax2, date, .keep_all = F)
distinct(covidvax2, zip_code, .keep_all = F)
```

```
covidvax2 %>% group_by(zip_code) %>%
  summarise(fullyvax_range = range(fully_vaccinated), partiallyvax_range = range(partially_vaccinated))
```

```
## `summarise()` has grouped output by 'zip_code'. You can override using the `.groups` argument.
```

```
## # A tibble: 3,528 x 3
## # Groups:   zip_code [1,764]
##    zip_code fullyvax_range partiallyvax_range
##       <int>          <dbl>              <dbl>
```

```
##  1    90001             0                0
##  2    90001         36923            18224
##  3    90002             0                0
##  4    90002         24049             6320
##  5    90003             0                0
##  6    90003         33150             8899
##  7    90004            19              893
##  8    90004         37672            10232
##  9    90005            14              402
## 10    90005         22653             6785
## # ... with 3,518 more rows
```

Population Range and Median for COVID vaccinations for those over 12 years old.

```
covidvax2 %>% summarise(total_pop_range = range(total_pop_over_12),
                        total_pop_median = median(total_pop_over_12))
```

```
##   total_pop_range total_pop_median
## 1             0.0          13685.1
## 2         88556.7          13685.1
```

```
#Below, is the minimum, first quartile, median, third quartile, and max values.

fivenum(covidvax2$total_pop_over_12)
```

```
## [1]     0.00  1346.80 13685.10 31762.15 88556.70
```

```
fivenum(covidvax2$fully_vaccinated)
```

```
## [1]     0   189  1867 11637 67594
```

```
fivenum(covidvax2$partially_vaccinated)
```

```
## [1]     0    93  1033  3028 23195
```

```
#Descriptions

#We get the mean & range for all numeric and integer vectors in countydem2 below

distinct(countydem2, county_name, .keep_all = F)
```

```
countydem2 %>%
  summarise(renters_mean = mean(renters),
            owners_mean = mean(owners),
            vacant_mean = mean(vacant_housing),
            total_mean = mean(total_housing_units))
```

```
##   renters_mean owners_mean vacant_mean total_mean
## 1     95553.91    121299.5    19010.05   235863.5
```

```
countydem2 %>%
  summarise(renters_range = range(renters),
                        owners_range = range(owners),
                        vacant_range = range(vacant_housing),
                        total_range = range(total_housing_units))
```

```
##   renters_range owners_range vacant_range total_range
## 1           140          357          827        1760
## 2       1696455      1544749       203872     3445076
```

Below, we get the five number summary for data of interest in countydem2.

```
fivenum(countydem2$renters)
```

```
## [1]     140    5878   25140   87737 1696455
```

```
fivenum(countydem2$owners)
```

```
## [1]     357   12629   39306  123646 1544749
```

```
fivenum(countydem2$vacant_housing)
```

```
## [1]    827.0   3328.0   8580.5  18747.0 203872.0
```

```
fivenum(countydem2$total_housing_units)
```

```
## [1]    1760.0   23910.0   76183.5  233755.0 3445076.0
```