Mini-Project

Code

COVID-19 Vaccination Rates

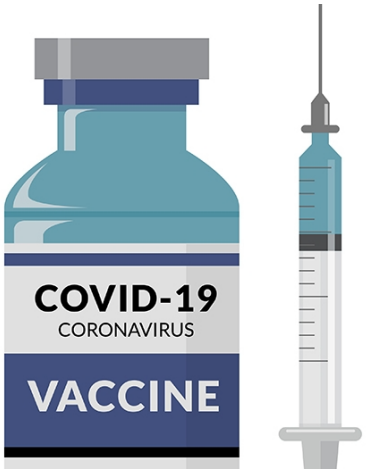
Barry Grant < http://thegrantlab.org/teaching/ (http://thegrantlab.org/teaching/) >
2022-11-21 (19:00:13 on Mon, Nov 21)

Background

The goal of this hands-on mini-project is to examine and compare the Covid-19 vaccination rates around San Diego.

We will start by downloading the most recently dated *"Statewide COVID-19 Vaccines Administered by ZIP Code"* CSV file from: https://data.ca.gov/dataset/covid-19-vaccine-progress-dashboard-data-by-zip-code (https://data.ca.gov/dataset/covid-19-vaccine-progress-dashboard-data-by-zip-code)

> **Side-Note:** With some web browsers this file may display as a new web page rather than simply download to your computer. In this case select "Save As" from your browser to obtain the CSV file.



Whilst you are on this website have a look at the **Data Dictionary** file that explains the various columns within the CSV file that you just downloaded.

There are also important notes about the limitations of the data. For example: "These data do NOT include doses administered by the following federal agencies who received vaccine allocated directly from CDC: Indian Health Service, Veterans Health Administration, Department of Defense, and the Federal Bureau of Prisons." One obvious implication here would be that Zip code areas that include military bases will likely show artificially low vaccination rates. We will bare this in mind for later.

Getting Started

Be sure to move your downloaded CSV file to your project directory and then read/import into an R object called `vax`. We will use this data to answer all the questions below.

Hide

```
# Import vaccination data
vax <- read.csv( ___ )
head(vax)
```

- **Q1.** What column details the total number of people fully vaccinated?
- **Q2.** What column details the Zip code tabulation area?
- **Q3.** What is the earliest date in this dataset?
- **Q4.** What is the latest date in this dataset?

Hint

As we have done previously, let's call the `skim()` function from the **skimr** package to get a quick overview of this dataset:

Hide

```
skimr::skim(vax)
```

Data summary

Name	vax
Number of rows	172872
Number of columns	18
_____	
Column type frequency:	
character	5
numeric	13
_____	
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
as_of_date	0	1	10	10	0	98	0
local_health_jurisdiction	0	1	0	15	490	62	0
county	0	1	0	15	490	59	0
vem_source	0	1	15	26	0	3	0
redacted	0	1	2	69	0	2	0

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
zip_code_tabulation_area	0	1.00	93665.11	1817.39	90001	92257.75	93658.50	95380.50	97635.0	▂▅▇▂
vaccine_equity_metric_quartile	8526	0.95	2.44	1.11	1	1.00	2.00	3.00	4.0	▇▁▃▇
age12_plus_population	0	1.00	18895.04	18993.88	0	1346.95	13685.10	31756.12	88556.7	▇▃▂▁
age5_plus_population	0	1.00	20875.24	21105.98	0	1460.50	15364.00	34877.00	101902.0	▇▃▂▁
tot_population	8428	0.95	23372.77	22628.51	12	2126.00	18714.00	38168.00	111165.0	▇▃▂▁
persons_fully_vaccinated	15440	0.91	13309.15	14740.07	11	859.00	7687.00	22253.00	87305.0	▇▃▁▁
persons_partially_vaccinated	15440	0.91	1679.13	1993.86	11	157.00	1158.00	2483.00	39201.0	▇▁▁▁
percent_of_population_fully_vaccinated	18986	0.89	0.54	0.26	0	0.36	0.58	0.73	1.0	▂▅▇▂
percent_of_population_partially_vaccinated	18986	0.89	0.08	0.09	0	0.05	0.06	0.08	1.0	▇▁▁▁
percent_of_population_with_1_plus_dose	19822	0.89	0.60	0.26	0	0.42	0.64	0.79	1.0	▂▅▇▇
booster_recip_count	70642	0.59	5701.06	6972.68	11	276.00	2546.00	9513.00	58301.0	▇▂▁▁
bivalent_dose_recip_count	156937	0.09	1512.94	1994.71	11	101.00	662.00	2236.00	16790.0	▇▁▁▁
eligible_recipient_count	0	1.00	12114.80	14551.97	0	438.00	5520.00	20714.00	86817.0	▇▂▁▁

- **Q5.** How many numeric columns are in this dataset?
- **Q6.** Note that there are "missing values" in the dataset. How many `NA` values there in the `persons_fully_vaccinated` column?
- **Q7.** What percent of `persons_fully_vaccinated` values are missing (to 2 significant figures)?
- **Q8.** [Optional]: Why might this data be missing?

Hint

Working with dates

One of the "character" columns of the data is `as_of_date` , which contains dates in the Year-Month-Day format.

Dates and times can be annoying to work with at the best of times. However, in R we have the excellent **lubridate** package, which can make life allot easier. Here is a quick example to get you started:

Hide

```
library(lubridate)
```

What is today's date (at the time I am writing this obviously)

Hide

```
today()
```

```
## [1] "2022-11-21"
```

The `as_of_date` column of our data is currently not that usable. For example we can't easily do math with it like answering the simple question how many days have passed since data was first recorded:

Hide

```
# This will give an Error!
today() - vax$as_of_date[1]
```

However if we convert our date data into a lubridate format things like this will be much easier as well as plotting time series data later on.

Hide

```
# Specify that we are using the year-month-day format
vax$as_of_date <- ymd(vax$as_of_date)
```

Now we can do math with dates. For example: How many days have passed since the first vaccination reported in this dataset?

Hide

```
today() - vax$as_of_date[1]
```

```
## Time difference of 685 days
```

Using the last and the first date value we can now determine how many days the dataset span?

Hide

```
vax$as_of_date[nrow(vax)] - vax$as_of_date[1]
```

```
## Time difference of 679 days
```

- **Q9.** How many days have passed since the last update of the dataset?
- **Q10.** How many unique dates are in the dataset (i.e. how many different dates are detailed)?

Working with ZIP codes

One of the numeric columns in the dataset (namely `vax$zip_code_tabulation_area`) are actually ZIP codes - a postal code used by the United States Postal Service (USPS). In R we can use the **zipcodeR** package to make working with these codes easier. For example, let's install and then load up this package and to find the centroid of the La Jolla 92037 (i.e. UC San Diego) ZIP code area.

Hide

```
library(zipcodeR)
```

Hide

```
geocode_zip('92037')
```

Calculate the distance between the centroids of any two ZIP codes in miles, e.g.

Hide

```
zip_distance('92037','92109')
```

More usefully, we can pull census data about ZIP code areas (including median household income etc.). For example:

Hide

```
reverse_zipcode(c('92037', "92109") )
```

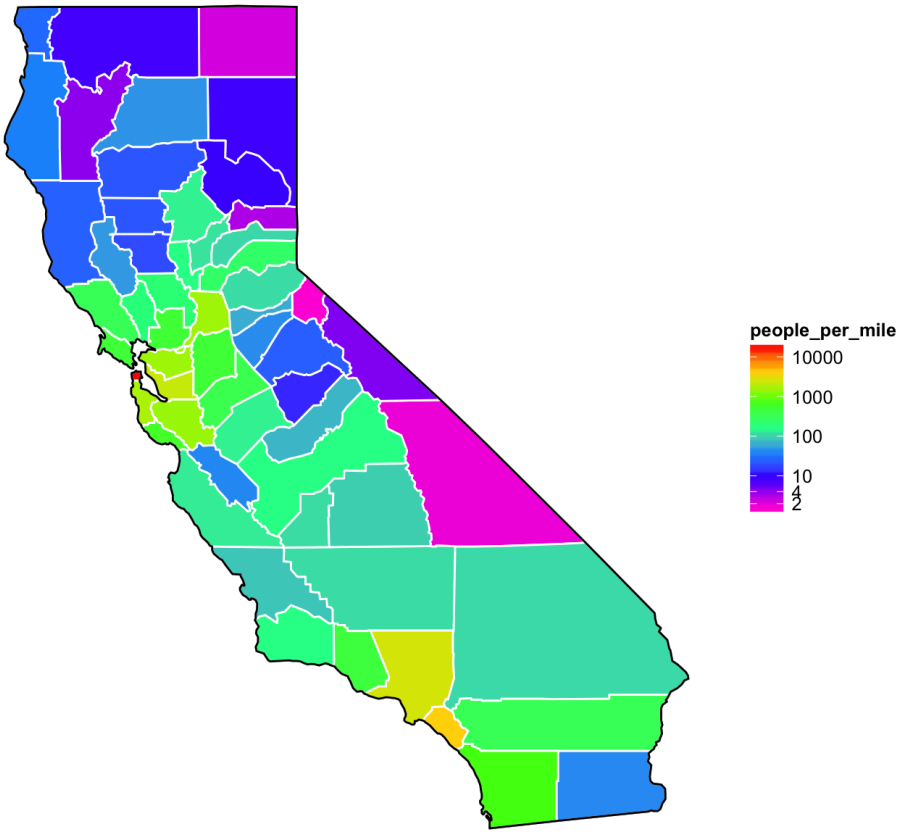
Optional: We can use this `reverse_zipcode()` to pull census data later on for any or all ZIP code areas we might be interested in.

Hide

```
# Pull data for all ZIP codes in the dataset
#zipdata <- reverse_zipcode( vax$zip_code_tabulation_area )
```

We could also access socioeconomic data for different ZIP code areas in a similar way if we wanted to investigate factors that might be correlated with different vaccine uptake rates.

Another informative data exploration might be to plot the various values along with the ZIP codes latitude and longitude values on a map using a package like **leafelet** (https://rstudio.github.io/leaflet/) or using **ggplot2** itself similar to this post (https://eriqande.github.io/rep-res-web/lectures/making-maps-with-R.html). For now we will leave this as an optional extension exercise.



Focus on the San Diego area

Let's now focus in on the San Diego County area by restricting ourselves first to `vax$county == "San Diego"` entries. We have two main choices on how to do this. The first using base R the second using the **dplyr** package:

Hide

```
# Subset to San Diego county only areas
sd <- vax[ ___ , ]
```

Using **dplyr** the code would look like this:

Hide

```
library(dplyr)

sd <- filter(vax, county == "San Diego")

nrow(sd)
```

```
## [1] 10486
```

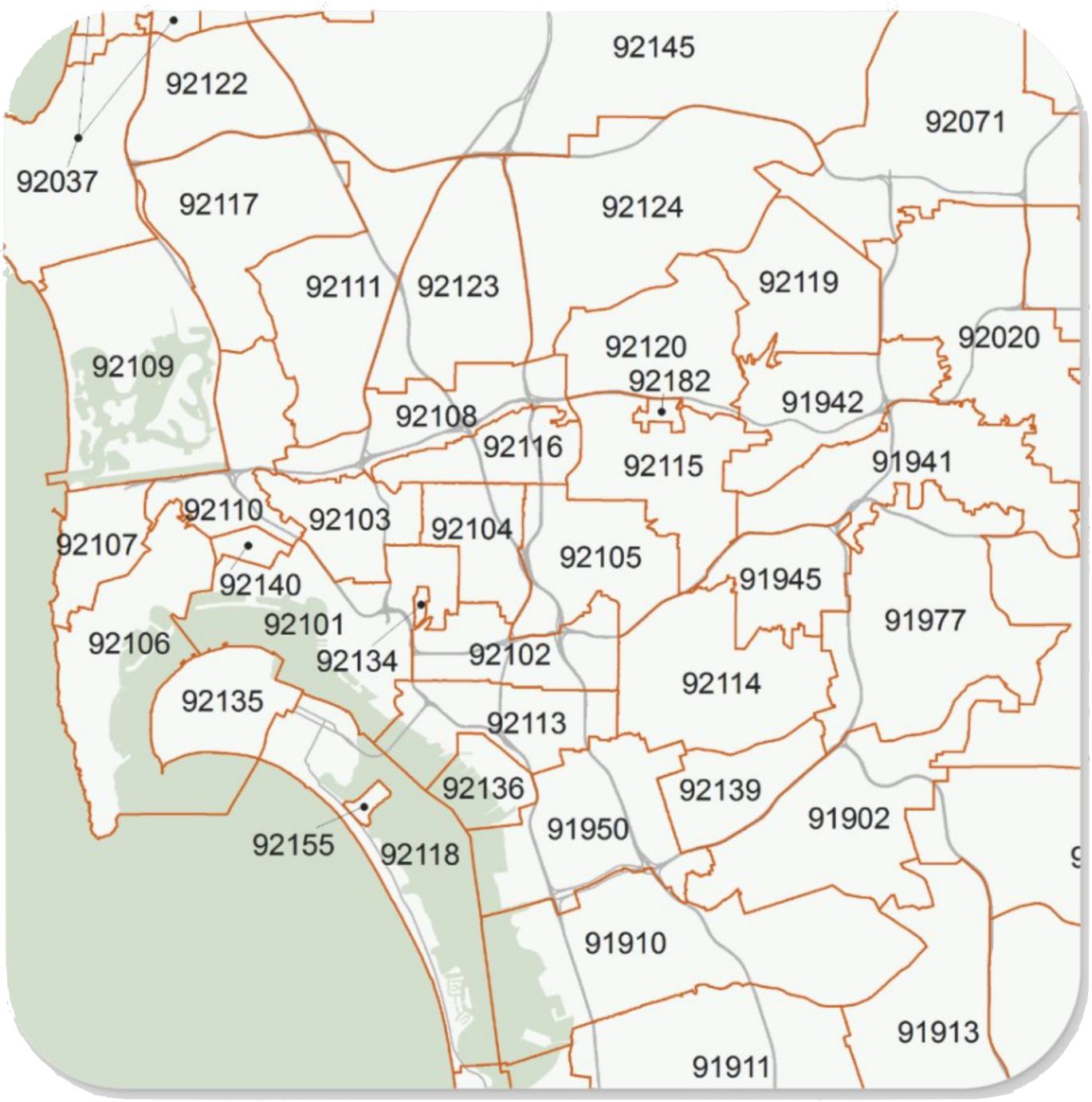
Using **dplyr** is often more convenient when we are subsetting across multiple criteria - for example all San Diego county areas with a population of over 10,000.

Hide

```
sd.10 <- filter(vax, county == "San Diego" &
                age5_plus_population > 10000)
```

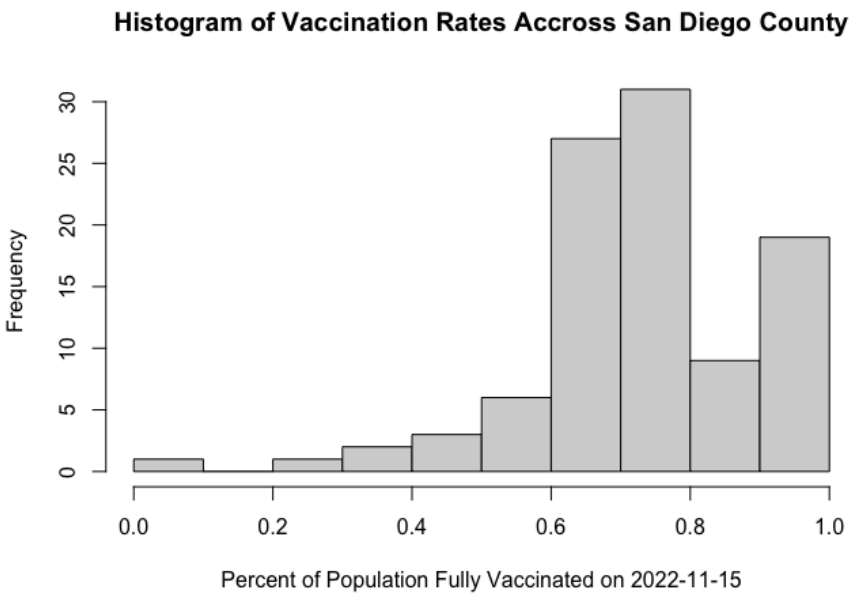
- **Q11.** How many distinct zip codes are listed for San Diego County? ☐
- **Q12.** What San Diego County Zip code area has the largest 12 + Population in this dataset? ☐

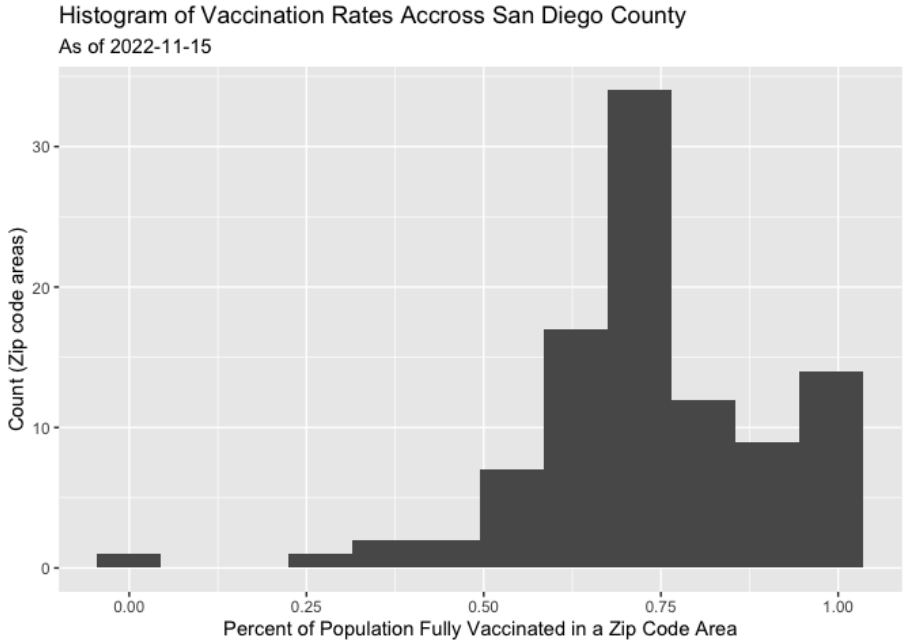
Hint



Using **dplyr** select all San Diego *"county"* entries on *"as_of_date"* "2022-11-15" and use this for the following questions.

- **Q13.** What is the overall average "Percent of Population Fully Vaccinated" value for all San Diego "County" as of "2022-11-15"?
- **Q14.** Using either ggplot or base R graphics make a summary figure that shows the distribution of Percent of Population Fully Vaccinated values as of "2022-11-15"?





Focus on UCSD/La Jolla

UC San Diego resides in the 92037 ZIP code area and is listed with an age 5+ population size of 36,144.

Hide

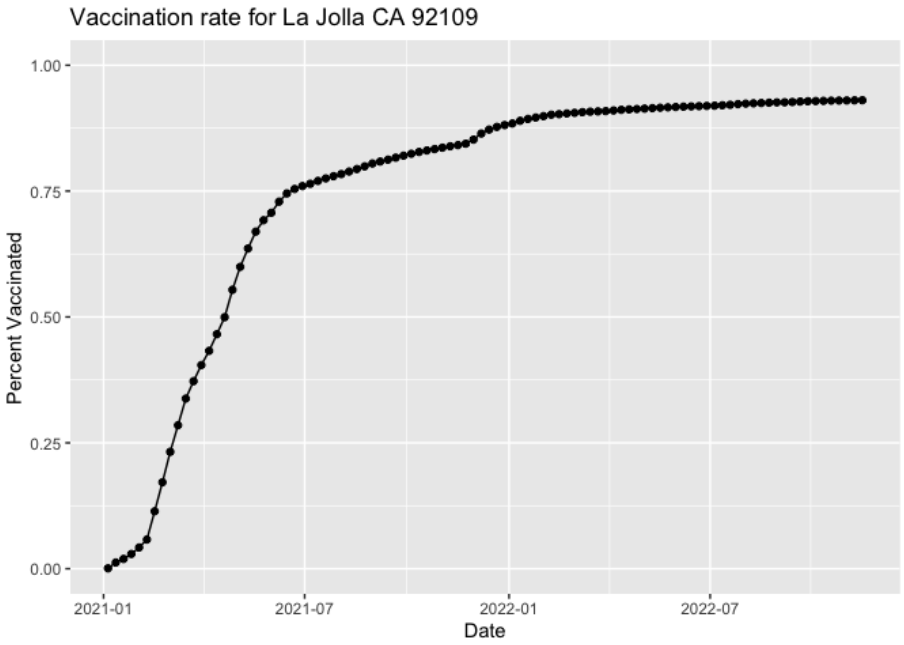
```
ucsd <- filter(sd, zip_code_tabulation_area=="92037")
ucsd[1,]$age5_plus_population
```

```
## [1] 36144
```

- **Q15**. Using **ggplot** make a graph of the vaccination rate time course for the 92037 ZIP code area:

Hide

```
ggplot(ucsd) +
  aes(___,
      ___) +
  geom____() +
  geom_line(group=1) +
  ylim(c(0,1)) +
  labs(___, y="Percent Vaccinated")
```



This plot shows an initial slow roll out in January into Febuary (likely due to limited vaccine availability). This is followed with rapid ramp up until a clear slowing trend from June time onward. Interpertation beyond this requies context from other zip code areas to answer questions such as: is this trend representative of other areas? Are more people fully vaccinated in this area compared to others? Etc.

Comparing to similar sized areas

Let's return to the full dataset and look across every zip code area with a population at least as large as that of 92037 on *as_of_date* "2022-02-22".

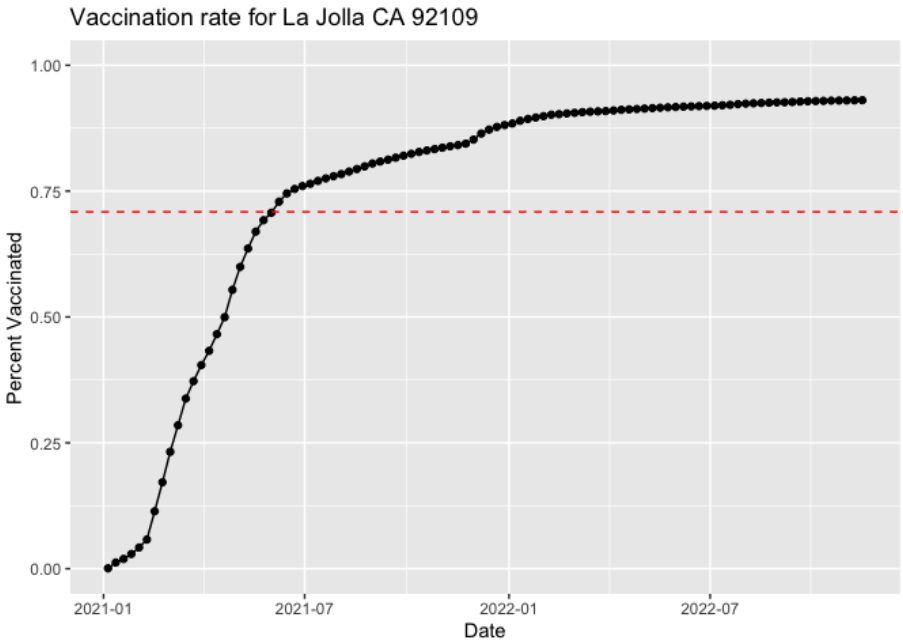
Hide

```
# Subset to all CA areas with a population as large as 92037
vax.36 <- filter(vax, age5_plus_population > 36144 &
                 as_of_date == "2022-11-15")

#head(vax.36)
```

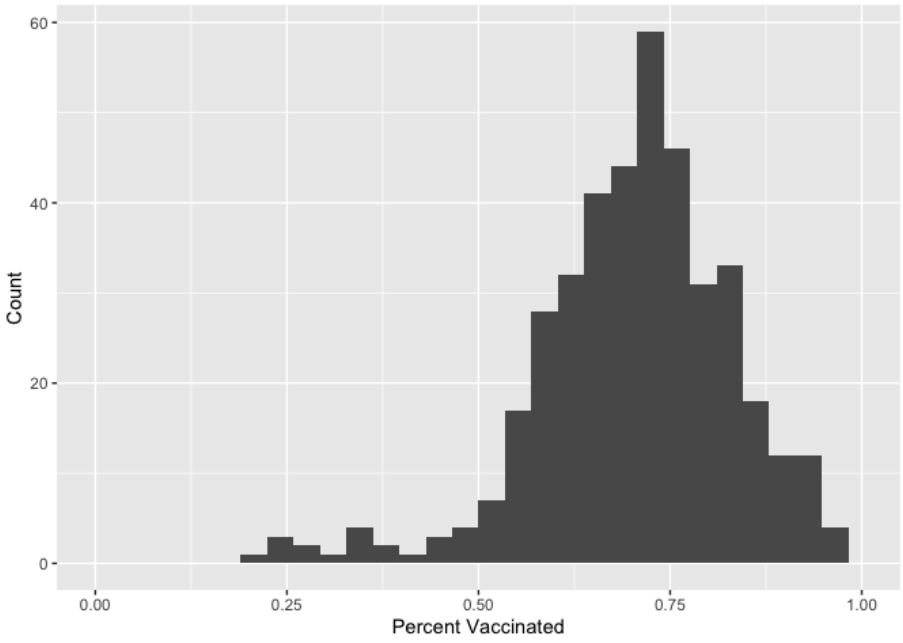
- **Q16**. Calculate the mean *"Percent of Population Fully Vaccinated"* for ZIP code areas with a population as large as 92037 (La Jolla) *as_of_date* "2022-11-15". Add this as a straight horizontal line to your plot from above with the `geom_hline()` function?

```
## [1] 0.7088141
```



- **Q17.** What is the 6 number summary (Min, 1st Qu., Median, Mean, 3rd Qu., and Max) of the *"Percent of Population Fully Vaccinated"* values for ZIP code areas with a population as large as 92037 (La Jolla) *as_of_date* "2022-11-15"?
- **Q18.** Using ggplot generate a histogram of this data.

```
## Warning: Removed 2 rows containing missing values (geom_bar).
```



- **Q19**. Is the 92109 and 92040 ZIP code areas above or below the average value you calculated for all these above?

Hide

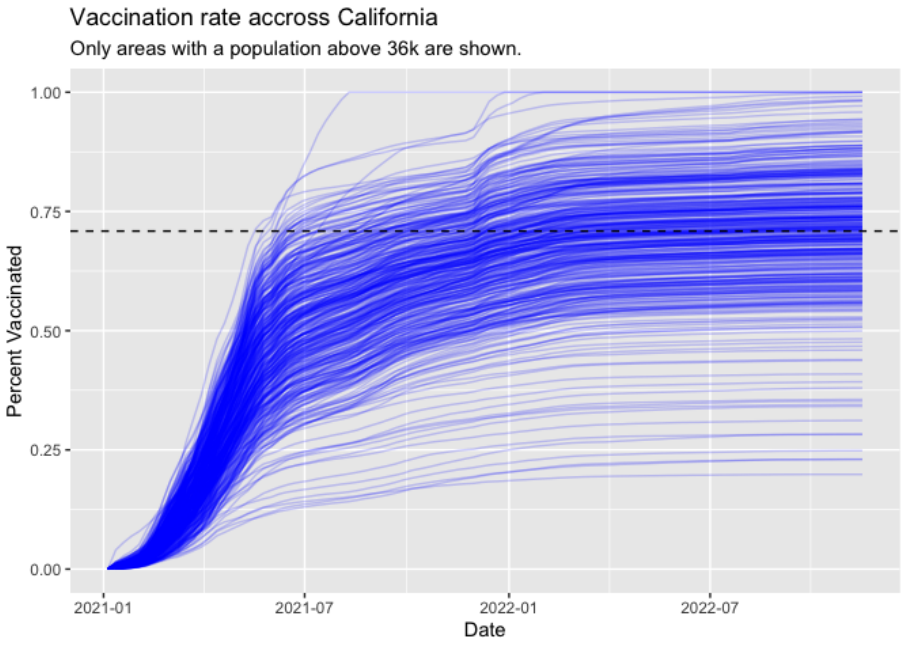
```
vax %>% filter(as_of_date == "2022-11-15") %>%
  filter(zip_code_tabulation_area=="92040") %>%
  select(percent_of_population_fully_vaccinated)
```

- **Q20.** Finally make a time course plot of vaccination progress for all areas in the full dataset with a `age5_plus_population > 36144`.

Hide

```
vax.36.all <- filter(vax, ___)


ggplot(vax.36.all) +
  aes(___,
      percent_of_population_fully_vaccinated,
      group=zip_code_tabulation_area) +
  geom_line(alpha=0.2, color=___) +
  ylim(___) +
  labs(x=___, y=___,
       title=___,
       subtitle=___) +
  geom_hline(yintercept = ___, linetype=___)
```



Q21. How do you feel about traveling for Thanksgiving Break and meeting for in-person class afterwards?

About this document

Here we use the `sessionInfo()` function to report on our R systems setup at the time of document execution.

Hide

```
sessionInfo()
```

```
## R version 4.1.2 (2021-11-01)
## Platform: x86_64-apple-darwin17.0 (64-bit)
## Running under: macOS Big Sur 10.16
##
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/4.1/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/4.1/Resources/lib/libRlapack.dylib
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] ggplot2_3.3.6   dplyr_1.0.10    zipcodeR_0.3.5  lubridate_1.8.0
## [5] labsheet_0.1.2
##
## loaded via a namespace (and not attached):
##  [1] httr_1.4.4         sass_0.4.2         tidyr_1.2.1        bit64_4.0.5
##  [5] jsonlite_1.8.3     bslib_0.4.0        assertthat_0.2.1   sp_1.5-0
##  [9] highr_0.9          blob_1.2.3         yaml_2.3.6         tidycensus_1.2.3
## [13] pillar_1.8.1       RSQLite_2.2.18     lattice_0.20-45    glue_1.6.2
## [17] uuid_1.1-0         digest_0.6.30      rvest_1.0.3        colorspace_2.0-3
## [21] htmltools_0.5.3    pkgconfig_2.0.3    raster_3.6-3       purrr_0.3.5
## [25] scales_1.2.1       terra_1.6-17       tzdb_0.3.0         tigris_1.6.1
## [29] tibble_3.1.8       proxy_0.4-27       farver_2.1.1       generics_0.1.3
## [33] ellipsis_0.3.2     cachem_1.0.6       withr_2.5.0        repr_1.1.4
## [37] skimr_2.1.4        cli_3.4.1          magrittr_2.0.3     crayon_1.5.2
## [41] memoise_2.0.1      maptools_1.1-5     evaluate_0.17      fansi_1.0.3
## [45] xml2_1.3.3         foreign_0.8-83     class_7.3-20       tools_4.1.2
## [49] hms_1.1.2          lifecycle_1.0.3    stringr_1.4.1      munsell_0.5.0
## [53] compiler_4.1.2     jquerylib_0.1.4    e1071_1.7-11       rlang_1.0.6
## [57] classInt_0.4-8     units_0.8-0        grid_4.1.2         rstudioapi_0.14
## [61] rappdirs_0.3.3     labeling_0.4.2     base64enc_0.1-3    rmarkdown_2.17
## [65] gtable_0.3.1       codetools_0.2-18   DBI_1.1.3          curl_4.3.3
## [69] R6_2.5.1           knitr_1.40         rgdal_1.5-32       fastmap_1.1.0
## [73] bit_4.0.4          utf8_1.2.2         KernSmooth_2.23-20 readr_2.1.3
## [77] stringi_1.7.8      Rcpp_1.0.9         vctrs_0.5.0        sf_1.0-8
## [81] tidyselect_1.2.0   xfun_0.34
```

Powered by labsheet (https://github.com/bioboot/labsheet/)