

## Class 14 Mini Project

Mirte Ciz Marieke Kuijpers

04/03/2022

### Introduction

In this class we will look at vaccine coverate across California. To start 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> should be downloaded. Once this is in the R project directory it can then be read into R. At this stage we should also do any necessary set-up, such as loading packages.

```
# Import vaccination data
vax <- read.csv("covid19vaccinesbyzipcode_test.csv", header = TRUE)
head(vax)
```

as_of_date	zip_code	tabulation_area	local_health_jurisdiction	county
## 1 2021-01-05	92549		Riverside	Riverside
## 2 2021-01-05	92130		San Diego	San Diego
## 3 2021-01-05	92397		San Bernardino	San Bernardino
## 4 2021-01-05	94563		Contra Costa	Contra Costa
## 5 2021-01-05	94519		Contra Costa	Contra Costa
## 6 2021-01-05	91042		Los Angeles	Los Angeles

vaccine_equity_metric	quartile	vem_source
## 1	3	Healthy Places Index Score
## 2	4	Healthy Places Index Score
## 3	3	Healthy Places Index Score
## 4	4	Healthy Places Index Score
## 5	3	Healthy Places Index Score
## 6	2	Healthy Places Index Score

age12_plus_population	age5_plus_population	persons_fully_vaccinated
## 1 2348.4	2461	NA
## 2 46300.3	53102	61
## 3 3695.6	4225	NA
## 4 17216.1	18896	NA
## 5 16861.2	18678	NA
## 6 23962.2	25741	NA

```

## persons_partially_vaccinated percent_of_population_fully_vaccinated
## 1 NA NA
## 2 27 0.001149
## 3 NA NA
## 4 NA NA
## 5 NA NA
## 6 NA NA
## percent_of_population_partially_vaccinated
## 1 NA
## 2 0.000508
## 3 NA
## 4 NA
## 5 NA
## 6 NA
## percent_of_population_with_1_plus_dose booster_recip_count
## 1 NA NA
## 2 0.001657 NA
## 3 NA NA
## 4 NA NA
## 5 NA NA
## 6 NA NA
## redacted
## 1 Information redacted in accordance with CA state privacy requirements
## 2 Information redacted in accordance with CA state privacy requirements
## 3 Information redacted in accordance with CA state privacy requirements
## 4 Information redacted in accordance with CA state privacy requirements
## 5 Information redacted in accordance with CA state privacy requirements
## 6 Information redacted in accordance with CA state privacy requirements

# Load packages
library("skimr")
library("lubridate")
library("zipcodeR")
library("leaflet")
library("dplyr")
library("ggplot2")
library("ggmap")
library("maps")
library("mapdata")
library("stringr")

```

### Question 1

The column that details the total number of people fully vaccinated is persons\_fully\_vaccinated, the 9th column.

### Question 2

The zip code tabulation area is detailed in the second column: zip\_code\_tabulation\_area.

### Q3

The earliest date in this dataset can be found using the `min()` function with `vax$as_of_date`, namely it is 2021-01-05 (format Year-Month-Day).

```
# Earliest data
min(vax$as_of_date)
## [1] "2021-01-05"
```

### Q4

The latest date in this dataset can be found in a similar way using the `max()` function with `vax$as_of_date`, namely it is 2022-03-01.

```
# Latest data
max(vax$as_of_date)
## [1] "2022-03-01"
```

## Exploratory analysis of the Data

```
# View data more comprehensively
skimr::skim(vax)
```

### Data summary

Name	vax
Number of rows	107604
Number of columns	15

---

### Column type frequency:

character	5
numeric	10

---











Group variables	None
-----------------	------

### Variable type: character

skim_variable	n_missin g	complete_rat e	mi n	ma x	empt y	n_uniqu e	whitespac e
as_of_date	0	1	10	10	0	61	0
local_health_jurisdicti on	0	1	0	15	305	62	0
county	0	1	0	15	305	59	0
vem_source	0	1	15	26	0	3	0

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
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	936 65.1 1	181 7.39	90 00 1	922 57.7 5	936 58.5 0	953 80.5 0	976 35.0	
vaccine_equity_metric_quartile	5307	0.95	2.44	1.11	1	1.00	2.00	3.00	4.0	
age12_plus_population	0	1.00	188 95.0 4	189 93.9 1	0	134 6.95	136 85.1 0	317 56.1 2	885 56.7	
age5_plus_population	0	1.00	208 75.2 4	211 06.0 2	0	146 0.50	153 64.0 0	348 77.0 0	101 902. 0	
persons_fully_vaccinated	18338	0.83	121 55.6 1	130 63.8 8	11	106 6.25	737 4.50	200 05.0 0	777 44.0	
persons_partially_vaccinated	18338	0.83	831. 74	134 8.68	11	76.0 0	372. 00	107 6.00	342 19.0	
percent_of_population_fully_vaccinated	18338	0.83	0.51	0.26	0	0.33	0.54	0.70	1.0	
percent_of_population_partially_vaccinated	18338	0.83	0.05	0.09	0	0.01	0.03	0.05	1.0	
percent_of_population_with_1_plus_dose	18338	0.83	0.54	0.28	0	0.36	0.58	0.75	1.0	
booster_recip_count	64317	0.40	410 0.55	590 0.21	11	176. 00	113 6.00	615 4.50	506 02.0	

```
# Store summary in a variable
sum.dat <- skimr::skim(vax)
```

### Question 5

According to the skimr package there are 15 columns in the dataset, of these 10 are numeric columns. However, it should be noted that the ZIP code is not a meaningful numeric, so in truth there are 9 numeric columns.

```
# Number of columns
dim(sum.dat)[1]

## [1] 15

# Number of columns with numeric data
sum(sum.dat$skim_type == "numeric")

## [1] 10
```

### Question 6

```
# Number NAs for persons_fully_vaccinated
pfv.nas <- sum.dat[(sum.dat$skim_variable == "persons_fully_vaccinated"),
  "n_missing"]
pfv.nas

## # A tibble: 1 x 1
##   n_missing
##       <int>
## 1      18338
```

Note that there are “missing values” in the dataset. The number of NA values in the persons\_fully\_vaccinated column can be found with the skimr package output to be 18338.

### Question 7

```
# Get the percentage using complete_rate
pfv.per.missing <- signif(sum.dat[(sum.dat$skim_variable ==
  "persons_fully_vaccinated"), "complete_rate"], digits = 2)*100

# Print this value
pfv.per.missing

##   complete_rate
## 1             83
```

The percent of persons\_fully\_vaccinated values that are missing (to 2 significant figures) could be found by dividing the answer for question 6 by the full number of fully vaccinated people. However, skimr gives a complete rate column, so we can simply turn this fraction into a percentage, giving 83%.

### Question 8

ZIP codes are not the best way to categorise geography, e.g. ZIP codes for a single post office or a military base are likely to come up as missing data. Furthermore, people can get a vaccination in a particular ZIP code, but live in a different one.

## Working with dates

The package lubridate can be used to work with dates.

```
# The date today
today()

## [1] "2022-03-05"

# We can use this package to add, subtract etc dates, make sure to specify your format

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

# Find the number of days between today and the earliest date the vaccinations were recorded
today() - vax$as_of_date[1]

## Time difference of 424 days

# Find teh number of days between the earliest and latest days that vaccinations were recoreded
vax$as_of_date[nrow(vax)] - vax$as_of_date[1]

## Time difference of 420 days
```

### Question 9

```
# Number of days between today and the Latest update
updated <- today() - vax$as_of_date[nrow(vax)]
updated

## Time difference of 4 days
```

The number of days since the last update fo the dataset is 4.

### Question 10

There are multiple inputs for the same dates in the dataset.

```
# Unique dates
uni.dates <- length(unique(vax$as_of_date))
uni.dates

## [1] 61
```

In fact, in the entire database there are actually only 61 unique dates.

## Working with ZIP codes

For this section the package zipcodeR is useful.

```

# Look at the geographic location of a specific ZIP code
geocode_zip('92037')

## # A tibble: 1 x 3
##   zipcode lat lng
##   <chr>   <dbl> <dbl>
## 1 92037   32.8 -117.

# Calculate the distance between two ZIP codes
zip_distance('92037', '92109')

##   zipcode_a zipcode_b distance
## 1      92037      92109      2.33

# Pull census data from these two ZIP codes
reverse_zipcode(c('92037', "92109") )

## # A tibble: 2 x 24
##   zipcode zipcode_type major_city post_office_city common_city_list county
##   <chr>   <chr>         <chr>         <chr>         <blob> <chr>
##   <chr>
## 1 92037   Standard      La Jolla     La Jolla, CA     <raw 20 B> San D~
##   CA
## 2 92109   Standard      San Diego    San Diego, CA     <raw 21 B> San D~
##   CA
## # ... with 17 more variables: lat <dbl>, lng <dbl>, timezone <chr>,
## #   radius_in_miles <dbl>, area_code_list <blob>, population <int>,
## #   population_density <dbl>, land_area_in_sqmi <dbl>,
## #   water_area_in_sqmi <dbl>, housing_units <int>,
## #   occupied_housing_units <int>, median_home_value <int>,
## #   median_household_income <int>, bounds_west <dbl>, bounds_east <dbl>,
## #   bounds_north <dbl>, bounds_south <dbl>

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

```

At this point it might be nice to overlay the data onto a map. There are packages in R (see set-up for their loading in this document) that contain geographical data, and these can be interpreted by ggplot. Much of the code that follows is heavily based on a tutorial that can be found at this link: <https://eriqande.github.io/rep-res-web/lectures/making-maps-with-R.html>.

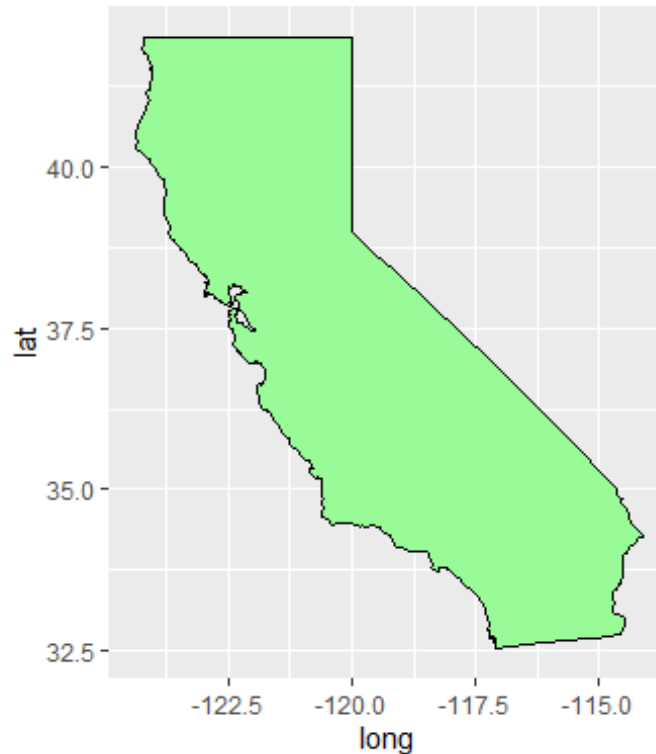
```

# Get the data to map California
states <- map_data("state")
cal <- subset(states, region %in% "california")

# Map california
ggplot(data = cal) +
  geom_polygon(aes(x = long, y = lat, group = group), fill = "palegreen",

```

```
color = "black") +  
  coord_fixed(1.3)
```



This illustrates how

we can plot a map. However, we now need county data.

```
# Get california data specifically  
ca_df <- subset(states, region == "california")  
  
# Get county data  
counties <- map_data("county")  
ca_county <- subset(counties, region == "california")  
  
# Plot the state with no background  
ca_base <- ggplot(data = ca_df, mapping = aes(x = long, y = lat, group =  
group)) +  
  coord_fixed(1.3) +  
  geom_polygon(color = "black", fill = "gray")  
  
ca_base + theme_nothing()
```





```
# and with county borders  
ca_base + theme_nothing() +  
  geom_polygon(data = ca_county, fill = NA, color = "white") +  
  geom_polygon(color = "black", fill = NA) # get the state border back on  
top
```



With counties in place we can now overlay our data. However, the map uses longitude and latitude, while the overlay data uses ZIP codes. These cannot easily be mapped over each other. So it is necessary to download a set of ZIP code to longitude and latitude data and subset this to our data.

```
# Use only the latest data
plotting.vax <- vax %>%
  filter(as_of_date == "2022-03-01")

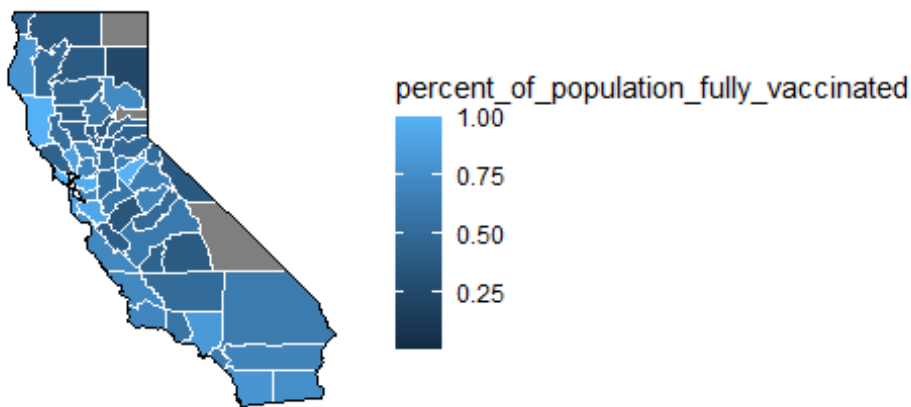
# Make the columns required to merge the data the same (r is case sensitive)
ca_county$subregion <- toupper(ca_county$subregion)
plotting.vax$local_health_jurisdiction <-
  toupper(plotting.vax$local_health_jurisdiction)

# Add Longitude and Latitude to this data
plotting.dat <- merge(ca_county, plotting.vax, by.x = "subregion", by.y =
  "local_health_jurisdiction", all.x = TRUE)

# Make own theme
ditch_the_axes <- theme(
  axis.text = element_blank(),
  axis.line = element_blank(),
  axis.ticks = element_blank(),
  panel.border = element_blank(),
  panel.grid = element_blank(),
  axis.title = element_blank()
)
```

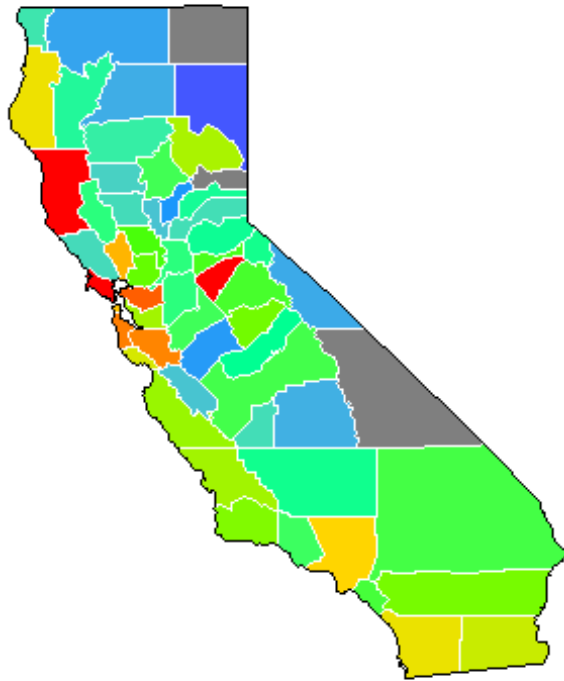
```
# Plot with data over-layed
v.geoplot <- ggplot(data = ca_df, mapping = aes(x = long, y = lat, group =
group)) +
  coord_fixed(1.3) +
  geom_polygon(data = ca_county, fill = NA, color = "white") +
  geom_polygon(data = plotting.dat, aes(fill =
percent_of_population_fully_vaccinated), color = "white") +
  geom_polygon(color = "black", fill = NA) +
  theme_bw() +
  ditch_the_axes

v.geoplot
```



The above map shows the percentage of the population fully vaccinated per subregion in California. The grey regions have insufficient data. For a more colourful representation we can change the colour gradient.

```
# Add a more colourful scale
v.geoplot +
  scale_fill_gradientn(colours = rev(rainbow(7)), breaks = c(2, 4, 10, 100,
1000, 10000))
```



## Focus on the San Diego County

We can use the dplyr package to look into the San Diego county alone.

```
# Subset to San Diego county only areas
sd <- filter(vax, county == "San Diego")

nrow(sd)

## [1] 6527

# Look for entries with populations who have greater than 10 000 people over
the age of 5
sd.10 <- filter(vax, county == "San Diego" & age5_plus_population > 10000)

nrow(sd.10)

## [1] 4636
```

This shows us that there are 6527 entries for the San Diego county, and 4636 have populations with more than 10 000 people over the age of 5.

### Question 11

```
# Find unique ZIP codes
sd.unique <- vax %>%
  filter(county == "San Diego") %>%
```

```

      select(zip_code_tabulation_area) %>%
      unique() %>%
      nrow()

sd.unique
## [1] 107

```

Within the San Diego county there are 107 distinct zip codes.

## Question 12

*# Find the San Diego Zip Code Area with the Largest 12 + population in this dataset*

```

l12 <- arrange(sd, -age12_plus_population)[1, "zip_code_tabulation_area"]

l12
## [1] 92154

```

The San Diego County Zip code area with the largest 12 + Population in this dataset is 92154. Another viable option would have been to use `ind <- which.max(sd$age12_plus_population)` and then `sd$zip_code_tabulation_area[ind]`.

## Question 13

*# Find the Percent of Population Fully Vaccinate for the San Diego County as of 2022-03-01*

```

q13 <- vax %>%
  filter(county == "San Diego", as_of_date == "2022-03-01") %>%
  select(percent_of_population_fully_vaccinated) %>%
  summary()

q13

## percent_of_population_fully_vaccinated
## Min.      :0.01017
## 1st Qu.:0.65132
## Median :0.72452
## Mean    :0.70529
## 3rd Qu.:0.82567
## Max.    :1.00000
## NA's    :1

```

The overall average “Percent of Population Fully Vaccinated” value for all San Diego “County” as of the most recent data (2022-03-01) is Mean :0.70529 or 70.5 %.

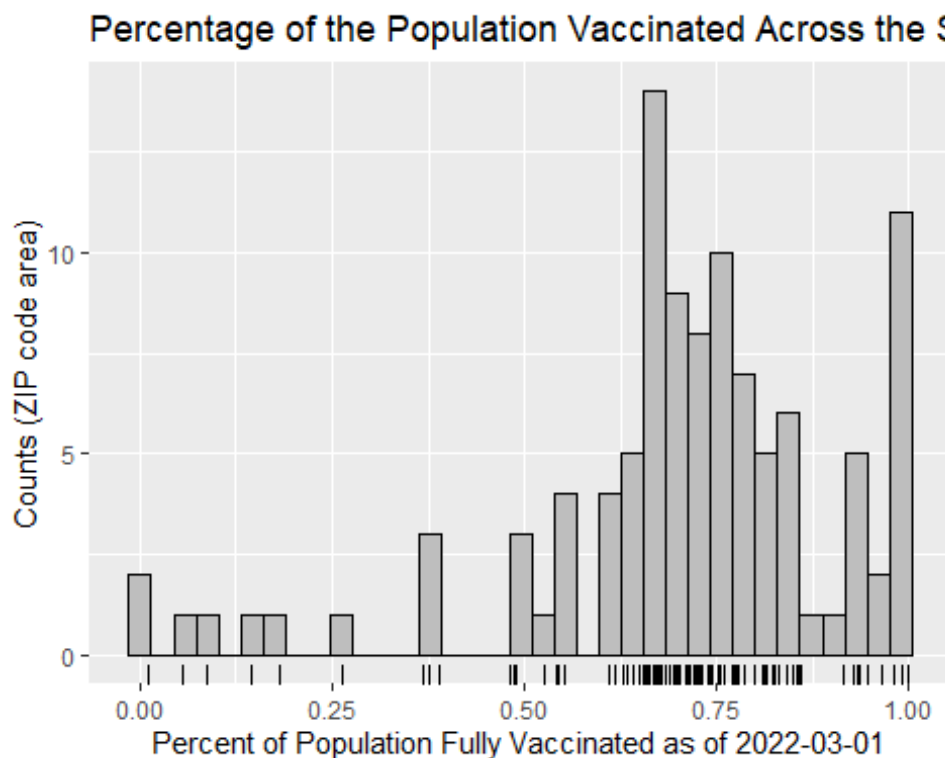
## Question 14

Using ggplot one can make a summary figure that shows the distribution of Percent of Population Fully Vaccinated values as of 2022-03-01.

```
# Get raw data
q13.dat <- vax %>%
  filter(county == "San Diego", as_of_date == "2022-03-01") %>%
  select(percent_of_population_fully_vaccinated)

# Plot raw data
ggplot(q13.dat, aes(x = percent_of_population_fully_vaccinated)) +
  geom_histogram(col = "black", fill = "grey", bins = 35) +
  labs(title = "Percentage of the Population Vaccinated Across the San Diego County", x = "Percent of Population Fully Vaccinated as of 2022-03-01", y = "Counts (ZIP code area)") +
  geom_rug()

## Warning: Removed 1 rows containing non-finite values (stat_bin).
```



## Focus on UCSD/La Jolla

UC San Diego resides in the 92037 ZIP code area.

```
# Assign a variable for UC San Diego
ucsd <- filter(sd, zip_code_tabulation_area=="92037")
```

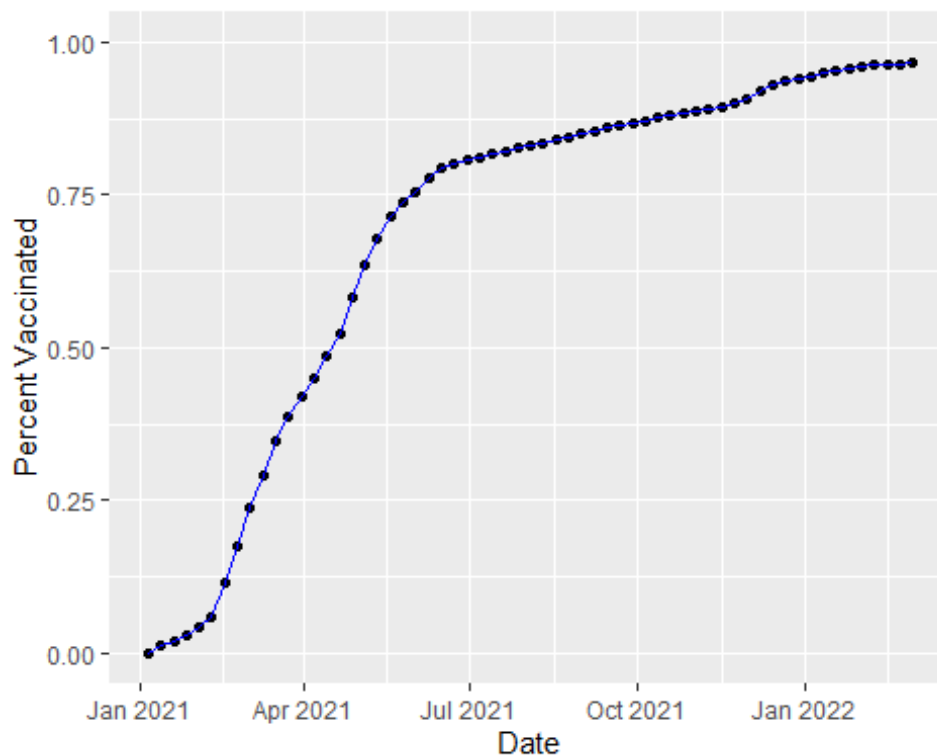
```
# Check pop over the age of 5
ucsd[1,]$age5_plus_population

## [1] 36144
```

### Question 15

Using ggplot a graph of the vaccination rate time course for the 92037 ZIP code area (UCSD), can be made.

```
ggplot(ucsd, aes(x = as_of_date, y = percent_of_population_fully_vaccinated)) +
  geom_point() +
  geom_line(col = "blue") +
  ylim(c(0,1)) +
  labs(x = "Date", y="Percent Vaccinated")
```



## Comparing to

similarly sized ZIP codes

```
# Subset to all CA areas with a population as large as 92037
vax.36 <- filter(vax, age5_plus_population > 36144 &
  as_of_date == "2022-03-01")

head(vax.36)

##   as_of_date zip_code_tabulation_area local_health_jurisdiction
##   county
```

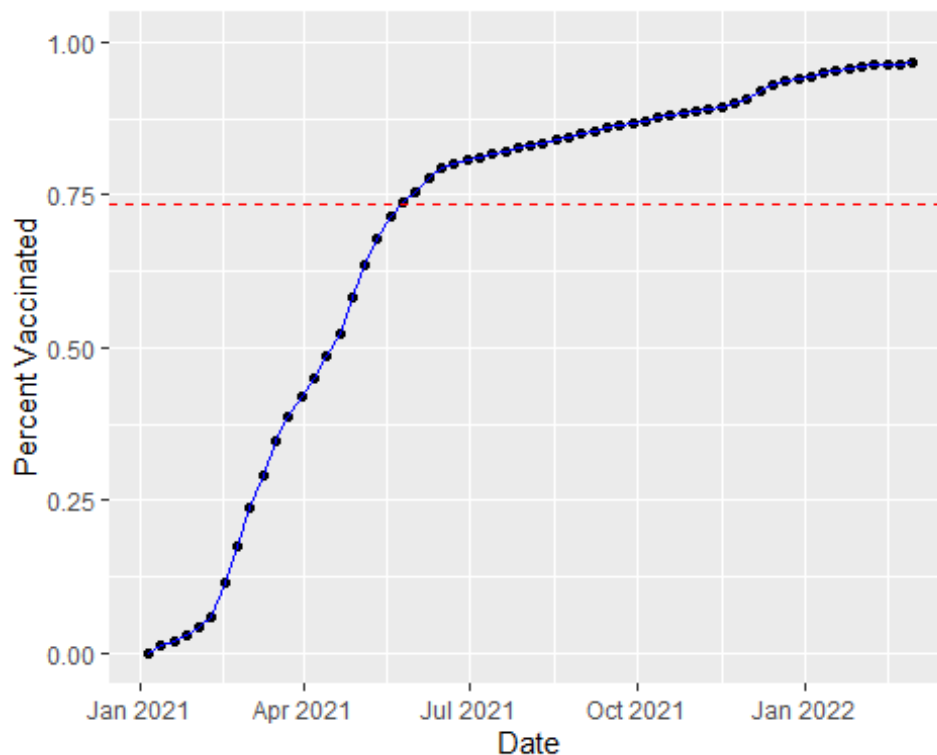
## 1 2022-03-01	95628	Sacramento
Sacramento		
## 2 2022-03-01	90808	Long Beach Los Angeles
## 3 2022-03-01	92507	Riverside
Riverside		
## 4 2022-03-01	92626	Orange
Orange		
## 5 2022-03-01	93257	Tulare
Tulare		
## 6 2022-03-01	90011	Los Angeles Los Angeles
Los Angeles		
## vaccine_equity_metric_quartile		vem_source
## 1	3 Healthy Places Index Score	
## 2	4 Healthy Places Index Score	
## 3	1 Healthy Places Index Score	
## 4	3 Healthy Places Index Score	
## 5	1 Healthy Places Index Score	
## 6	1 Healthy Places Index Score	
## age12_plus_population	age5_plus_population	persons_fully_vaccinated
## 1	35579.0	38694 28842
## 2	33952.3	37179 29383
## 3	51432.5	55253 34455
## 4	44238.8	47883 33767
## 5	61519.8	70784 42919
## 6	87902.8	101902 65342
## persons_partially_vaccinated	percent_of_population_fully_vaccinated	
## 1	1990	0.745387
## 2	2112	0.790312
## 3	3947	0.623586
## 4	2937	0.705198
## 5	5868	0.606338
## 6	15255	0.641224
## percent_of_population_partially_vaccinated		
## 1	0.051429	
## 2	0.056806	
## 3	0.071435	
## 4	0.061337	
## 5	0.082900	
## 6	0.149703	
## percent_of_population_with_1_plus_dose	booster_recip_count	redacted
## 1	0.796816	16913 No
## 2	0.847118	17253 No
## 3	0.695021	15073 No
## 4	0.766535	17595 No
## 5	0.689238	17740 No
## 6	0.790927	19928 No



## Question 16

We can calculate the mean “Percent of Population Fully Vaccinated” for ZIP code areas with a population as large as 92037 (La Jolla) as\_of\_date “2022-03-01” and add this as a straight horizontal line to the previous plot.

```
# Make improved plot with line for the average
ggplot(ucsd, aes(x = as_of_date, y = percent_of_population_fully_vaccinated)) +
  geom_point() +
  geom_line(col = "blue") +
  ylim(c(0,1)) +
  labs(x = "Date", y = "Percent Vaccinated") +
  geom_hline(yintercept = mean(vax.36$percent_of_population_fully_vaccinated,
na.omit = TRUE), col = "red", lty = 2)
```



So the UCSD percentage vaccination has been higher than the average percentage vaccination of the population for regions of similar population size for quite some time. Note, however, that this average is the average as of the last update of the data, in fact this average will vary across updates.

```
vax.36.all.dates <- filter(vax, age5_plus_population > 36144) %>%
  select(as_of_date, percent_of_population_fully_vaccinated)

vax.36.all.dates <- na.omit(vax.36.all.dates)

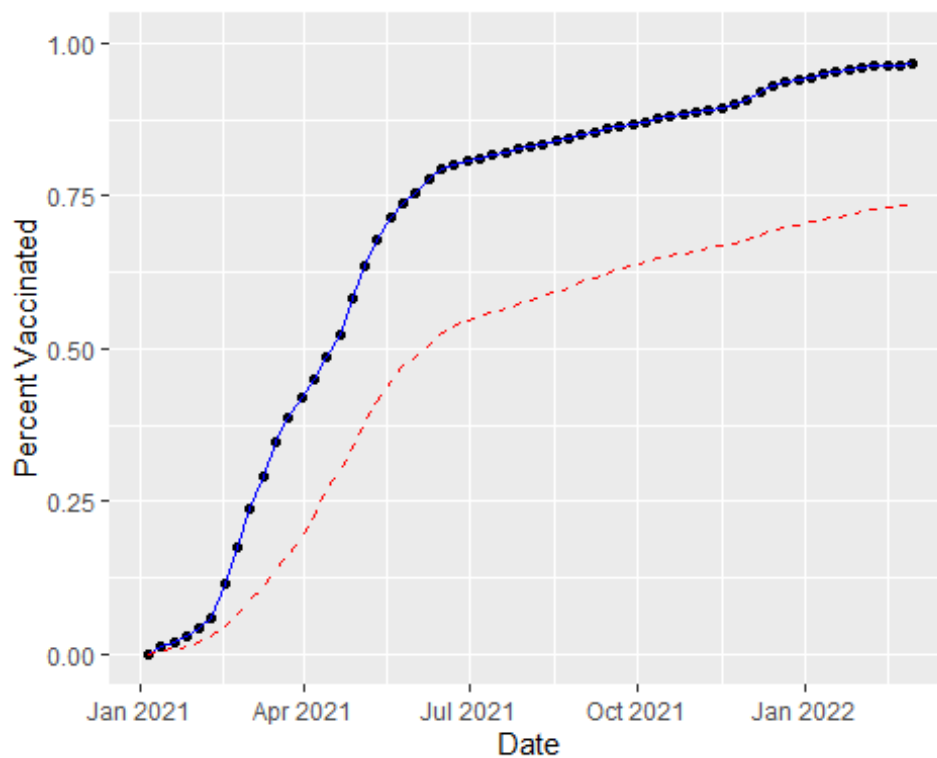
vax.36.agg <- aggregate(x =
```

```

vax.36.all.dates$percent_of_population_fully_vaccinated, by =
list(vax.36.all.dates$as_of_date), FUN = mean)
colnames(vax.36.agg) <- c("as_of_date",
"percent_of_population_fully_vaccinated")

ggplot(ucsd, aes(x = as_of_date, y = percent_of_population_fully_vaccinated))
+
  geom_point() +
  geom_line(col = "blue") +
  ylim(c(0,1)) +
  labs(x = "Date", y="Percent Vaccinated") +
  geom_line(dat = vax.36.agg, aes(x = as_of_date, y =
percent_of_population_fully_vaccinated), col = "red", lty = 2)

```



This shows that UCSD has consistently been above the average percentage of the population who is vaccinated.

### Question 17

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 the latest update can be found as follows.

```

# Summarise populations with a above 5 years old population as large as La
Jolla
filter(vax, age5_plus_population > ucsd[1,]$age5_plus_population & as_of_date
== "2022-03-01") %>%

```

```
select(percent_of_population_fully_vaccinated) %>%
summary()
```

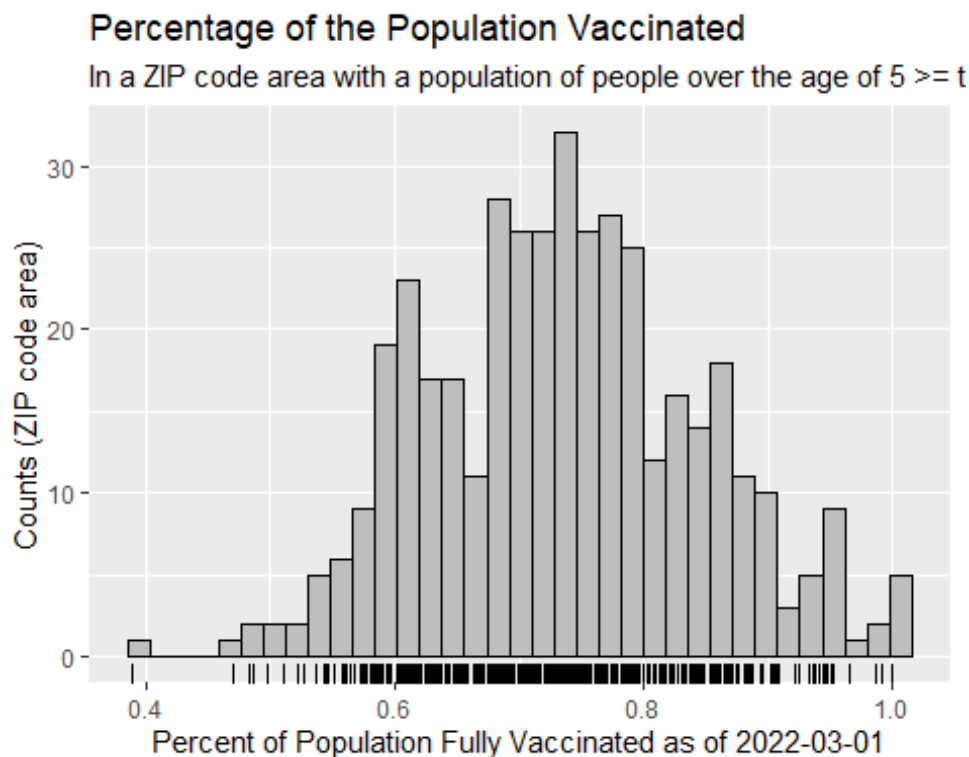
```
## percent_of_population_fully_vaccinated
## Min.      :0.3890
## 1st Qu.:0.6554
## Median :0.7350
## Mean      :0.7354
## 3rd Qu.:0.8044
## Max.      :1.0000
```

### Question 18

We can also plot this data.

```
# Assign raw data to a variable
q18 <- filter(vax, age5_plus_population > ucsd[1,]$age5_plus_population &
as_of_date == "2022-03-01") %>%
  select(percent_of_population_fully_vaccinated)

ggplot(q18, aes(x = percent_of_population_fully_vaccinated)) +
  geom_histogram(col = "black", fill = "grey", bins = 35) +
  labs(title = "Percentage of the Population Vaccinated", subtitle = "In a
ZIP code area with a population of people over the age of 5 >= that of La
Jolla", x = "Percent of Population Fully Vaccinated as of 2022-03-01", y =
"Counts (ZIP code area)") +
  geom_rug()
```



### Question 19

To determine if the 92109 and 92040 ZIP code areas above or below the average value you calculated for all these above the following code can be used

```
# Percentage of the population vaccinated for 92109
pfv.92040 <- vax %>% filter(as_of_date == "2022-02-22") %>%
  filter(zip_code_tabulation_area=="92040") %>%
  select(percent_of_population_fully_vaccinated)
as.numeric(pfv.92040)

## [1] 0.551304

# Percentage of the population vaccinated for 92040
pfv.92109 <- vax %>% filter(as_of_date == "2022-02-22") %>%
  filter(zip_code_tabulation_area=="92109") %>%
  select(percent_of_population_fully_vaccinated)
as.numeric(pfv.92109)

## [1] 0.723044

# Average percentage of the population vaccinated for the ZIP code areas with
# >5 year old pops greater than or equal to that of La Jolla
avg <- mean(q18$percent_of_population_fully_vaccinated)
avg

## [1] 0.7353974

# Code for output
if(pfv.92040 > avg){
  print(paste("The percent of the population fully vaccinated in ZIP code
area 92040 is greater than the average for those ZIP code areas with a
population of people older than 5 greater than or the same as La Jolla." ))
} else if(pfv.92040 == avg){
  print(paste("The percent of the population fully vaccinated in ZIP code
area 92040 is equal to the average for those ZIP code areas with a population
of people older than 5 greater than or the same as La Jolla." ))
} else{
  print(paste("The percent of the population fully vaccinated in ZIP code
area 92040 is less than the average for those ZIP code areas with a
population of people older than 5 greater than or the same as La Jolla." ))
}

## [1] "The percent of the population fully vaccinated in ZIP code area 92040
is less than the average for those ZIP code areas with a population of people
older than 5 greater than or the same as La Jolla."

if(pfv.92109 > avg){
  print(paste("The percent of the population fully vaccinated in ZIP code
area 92109 is greater than the average for those ZIP code areas with a
population of people older than 5 greater than or the same as La Jolla." ))
} else if(pfv.92109 == avg){
```

```

    print(paste("The percent of the population fully vaccinated in ZIP code
area 92109 is equal to the average for those ZIP code areas with a population
of people older than 5 greater than or the same as La Jolla." ))
}else{
    print(paste("The percent of the population fully vaccinated in ZIP code
area 92109 is less than the average for those ZIP code areas with a
population of people older than 5 greater than or the same as La Jolla." ))
}

## [1] "The percent of the population fully vaccinated in ZIP code area 92109
is less than the average for those ZIP code areas with a population of people
older than 5 greater than or the same as La Jolla."

```

## Question 20

A time course plot of vaccination progress for all areas in the full dataset with a `age5_plus_population > 36144` is below.

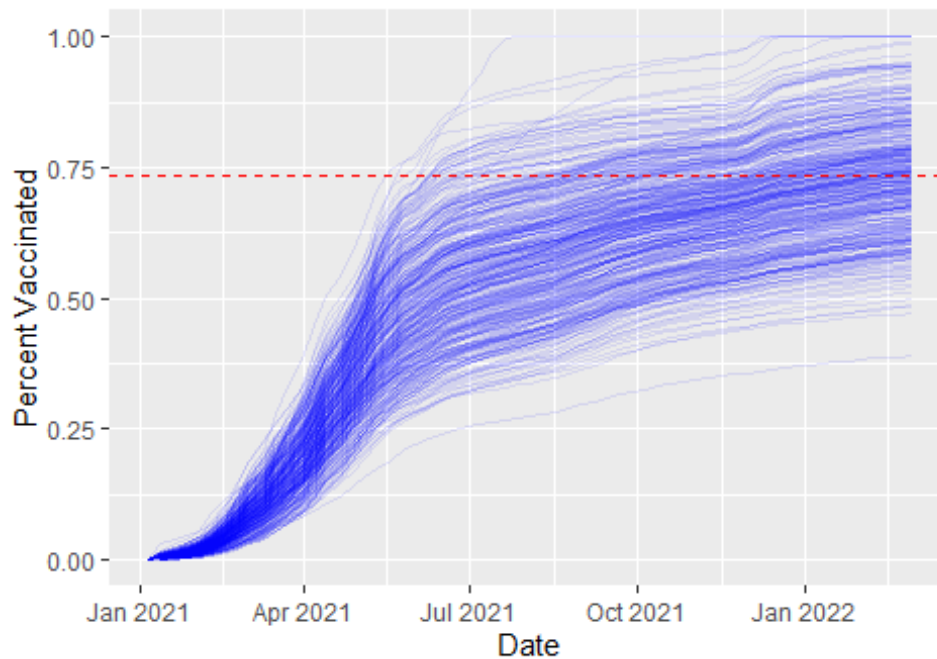
```

# Get raw data
q20.dat <- filter(vax, age5_plus_population > ucsd[1,]$age5_plus_population)

# Plot
ggplot(q20.dat, aes(x = as_of_date, y =
percent_of_population_fully_vaccinated, group = zip_code_tabulation_area)) +
  geom_line(col = "blue", alpha = 0.1) +
  ylim(c(0,1)) +
  labs(x = "Date", y="Percent Vaccinated", title = "", subtitle = "") +
  geom_hline(yintercept =
mean(vax.36$percent_of_population_fully_vaccinated, na.omit = TRUE), col =
"red", lty = 2)

## Warning: Removed 311 row(s) containing missing values (geom_path).

```



### Question 21

If I wanted to look at a particular place I could check that place's ZIP code to decide how cautious I should be (e.g. see below). However, as I am not going anywhere it doesn't matter.

```
head(filter(vax, zip_code_tabulation_area %in% "94555"))
```

	as_of_date	zip_code_tabulation_area	local_health_jurisdiction	county
## 1	2021-01-05	94555	Alameda	Alameda
## 2	2021-01-12	94555	Alameda	Alameda
## 3	2021-01-19	94555	Alameda	Alameda
## 4	2021-01-26	94555	Alameda	Alameda
## 5	2021-02-02	94555	Alameda	Alameda
## 6	2021-02-09	94555	Alameda	Alameda

	vaccine_equity_metric_quartile	vem_source
## 1	4	Healthy Places Index Score
## 2	4	Healthy Places Index Score
## 3	4	Healthy Places Index Score
## 4	4	Healthy Places Index Score
## 5	4	Healthy Places Index Score
## 6	4	Healthy Places Index Score

	age12_plus_population	age5_plus_population	persons_fully_vaccinated
## 1	31259.7	35970	NA
## 2	31259.7	35970	NA
## 3	31259.7	35970	332
## 4	31259.7	35970	463

```
## 5          31259.7          35970          669
## 6          31259.7          35970          992
##  persons_partially_vaccinated percent_of_population_fully_vaccinated
## 1                      NA                      NA
## 2                      NA                      NA
## 3                      11                      0.009230
## 4                      20                      0.012872
## 5                      24                      0.018599
## 6                      33                      0.027579
##  percent_of_population_partially_vaccinated
## 1                      NA
## 2                      NA
## 3                      0.000306
## 4                      0.000556
## 5                      0.000667
## 6                      0.000917
##  percent_of_population_with_1_plus_dose booster_recip_count
## 1                      NA                      NA
## 2                      NA                      NA
## 3                      0.009536                      NA
## 4                      0.013428                      NA
## 5                      0.019266                      NA
## 6                      0.028496                      NA
##
## 1 Information redacted in accordance with CA state privacy requirements
## 2 Information redacted in accordance with CA state privacy requirements
## 3 Information redacted in accordance with CA state privacy requirements
## 4 Information redacted in accordance with CA state privacy requirements
## 5 Information redacted in accordance with CA state privacy requirements
## 6 Information redacted in accordance with CA state privacy requirements
```

## About this document

```
sessionInfo()
```

```
## R version 4.1.2 (2021-11-01)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 22000)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=English_United Kingdom.1252
## [2] LC_CTYPE=English_United Kingdom.1252
## [3] LC_MONETARY=English_United Kingdom.1252
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United Kingdom.1252
##
## attached base packages:
## [1] stats      graphics  grDevices  utils      datasets  methods   base
```

```
##
## other attached packages:
## [1] stringr_1.4.0  mapdata_2.3.0  maps_3.4.0      ggmap_3.0.0.903
## [5] ggplot2_3.3.5  dplyr_1.0.8    leaflet_2.1.0   zipcodeR_0.3.3
## [9] lubridate_1.8.0 skimr_2.1.3
##
## loaded via a namespace (and not attached):
## [1] httr_1.4.2      tidyr_1.2.0      bit64_4.0.5
## [4] jsonlite_1.8.0  highr_0.9        sp_1.4-6
## [7] blob_1.2.2      yaml_2.2.2       tidycensus_1.1
## [10] pillar_1.7.0    RSQLite_2.2.10   lattice_0.20-45
## [13] glue_1.6.1      uuid_1.0-3       digest_0.6.29
## [16] rvest_1.0.2     colorspace_2.0-2 plyr_1.8.6
## [19] htmltools_0.5.2 pkgconfig_2.0.3  raster_3.5-15
## [22] purrr_0.3.4     scales_1.1.1     jpeg_0.1-9
## [25] terra_1.5-21    tzdb_0.2.0       tigris_1.6
## [28] tibble_3.1.6    proxy_0.4-26     farver_2.1.0
## [31] generics_0.1.2  ellipsis_0.3.2   withr_2.5.0
## [34] cachem_1.0.6    repr_1.1.4       cli_3.2.0
## [37] magrittr_2.0.2  crayon_1.5.0     memoise_2.0.1
## [40] maptools_1.1-2  evaluate_0.15    fansi_1.0.2
## [43] xml2_1.3.3      foreign_0.8-81   class_7.3-19
## [46] tools_4.1.2     RgoogleMaps_1.4.5.3 hms_1.1.1
## [49] lifecycle_1.0.1 munsell_0.5.0    compiler_4.1.2
## [52] e1071_1.7-9     rlang_1.0.1      classInt_0.4-3
## [55] units_0.8-0     grid_4.1.2       rstudioapi_0.13
## [58] rappdirs_0.3.3  htmlwidgets_1.5.4 crosstalk_1.2.0
## [61] labeling_0.4.2  bitops_1.0-7     base64enc_0.1-3
## [64] rmarkdown_2.11  gtable_0.3.0     codetools_0.2-18
## [67] DBI_1.1.2       curl_4.3.2       R6_2.5.1
## [70] knitr_1.37      rgdal_1.5-28     fastmap_1.1.0
## [73] bit_4.0.4       utf8_1.2.2       KernSmooth_2.23-20
## [76] readr_2.1.2     stringi_1.7.6    Rcpp_1.0.8
## [79] png_0.1-7       vctrs_0.3.8      sf_1.0-6
## [82] tidyselect_1.1.2 xfun_0.29
```