

Milestone #2

Virginia Chan, Patrick Traynor, Antoinette Stetzenmeyer

9/29/2021

```
#load libraries
library(tidyverse)

## -- Attaching packages ----- tidyverse 1.3.1 --
## v ggplot2 3.3.3      v purrr  0.3.4
## v tibble  3.1.2      v dplyr  1.0.6
## v tidyr   1.1.3      v stringr 1.4.0
## v readr   1.4.0      v forcats 0.5.1

## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()

library(lubridate)
```

```
##
## Attaching package: 'lubridate'

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

Description of dataset What is the data source? (1-2 sentences on where the data is coming from, dates included, etc.)

The data source for the COVID-19 Vaccine Administered by Zip Code dataset is the California Immunization Registry and the American Community Survey's 2015-2019 5-Year data.

The data source for the CA County Demographic dataset is based on US census data.

How does the dataset relate to the group problem statement and question?

There is growing concern that counties with younger median age are less likely to have higher counts of vaccinated persons. To investigate this, we need to explore the CA census demographics to compare median age values and proportions of vaccinate persons together on a county level. This will help us determine if there is any relationship between those two variables.

```

#Import statement
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()
## )
cov_vax_admin

## # A tibble: 65,268 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 B~          1
## 3     3 1/5/2021          92284 SAN BERNARDINO SAN B~          1
## 4     4 1/5/2021          92275 IMPERIAL          IMPER~          1
## 5     5 1/5/2021          92532 RIVERSIDE          RIVER~          3
## 6     6 1/5/2021          92376 SAN BERNARDINO SAN B~          1
## 7     7 1/5/2021          92345 SAN BERNARDINO SAN B~          1
## 8     8 1/5/2021          91343 LOS ANGELES          LOS A~          2
## 9     9 1/5/2021          91910 SAN DIEGO          SAN D~          2
## 10    10 1/5/2021          91773 LOS ANGELES          LOS A~          3
## # ... with 65,258 more rows, and 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.
ca_county_demographics

## # A tibble: 58 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 Lass~ 35039       7.42  2.55e4 2834      1234 3.56e2      165   6117
## 5      5 Los ~ 9904341    2423.  4.94e6 856874    72828 1.35e6    26094 4687889
## 6      6 Made~ 153025      71.1  9.45e4 5629      4136 2.80e3      162   80992
## 7      7 Marin 255509      486.  2.02e5 6987      1523 1.38e4      509   39069
## 8      8 Mari~ 18455      12.6  1.61e4 138        527 2.04e2       26   1676
## 9      9 Mend~ 88094      25.1  6.72e4 622        4277 1.45e3      119   19505
## 10     10 Merc~ 256841     130.  1.48e5 9926      3473 1.88e4      583  140485
## # ... with 48 more rows, and 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>
```

Identify data types for 5+ data elements/columns/variables for both data sets

```
#2 Identify data types for 5+ data elements/columns/variables
```

```
#viewing structure of both datasets
```

```
str(ca_county_demographics)
```

```
## spec_tbl_df [58 x 23] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
## $ X1      : num [1:58] 1 2 3 4 5 6 7 8 9 10 ...
## $ name     : chr [1:58] "Kern" "Kings" "Lake" "Lassen" ...
## $ pop2012  : num [1:58] 851089 155039 65253 35039 9904341 ...
## $ pop12_sqmi : num [1:58] 104.28 111.43 49.08 7.42 2423.26 ...
## $ white    : num [1:58] 499766 83027 52033 25532 4936599 ...
## $ black    : num [1:58] 48921 11014 1232 2834 856874 ...
## $ ameri_es : num [1:58] 12676 2562 2049 1234 72828 ...
## $ asian    : num [1:58] 34846 5620 724 356 1346865 ...
## $ hawn_pi   : num [1:58] 1252 271 108 165 26094 ...
## $ hispanic  : num [1:58] 413033 77866 11088 6117 4687889 ...
## $ other     : num [1:58] 204314 42996 5455 3562 2140632 ...
## $ mult_race : num [1:58] 37856 7492 3064 1212 438713 ...
## $ males     : num [1:58] 433108 86344 32469 22416 4839654 ...
## $ females   : num [1:58] 406523 66638 32196 12479 4978951 ...
## $ med_age   : num [1:58] 30.7 31.1 45 37 34.8 33.1 44.5 49.2 41.6 29.6 ...
## $ households : num [1:58] 254610 41233 26548 10058 3241204 ...
## $ families  : num [1:58] 191739 31939 16255 6800 2194080 ...
## $ hse_units  : num [1:58] 284367 43867 35492 12710 3445076 ...
## $ ave_fam_sz : num [1:58] 3.61 3.59 2.94 2.98 3.58 3.63 2.94 2.77 3.02 3.74 ...
## $ vacant    : num [1:58] 29757 2634 8944 2652 203872 ...
## $ owner_occ : num [1:58] 152828 22329 17472 6590 1544749 ...
## $ renter_occ : num [1:58] 101782 18904 9076 3468 1696455 ...
## $ county_fips: num [1:58] 6103 6089 6106 6086 6073 ...
## - attr(*, "spec")=
## .. cols(
## ..   X1 = col_double(),
## ..   name = col_character(),
## ..   pop2012 = col_double(),
## ..   pop12_sqmi = col_double(),
## ..   white = col_double(),
## ..   black = col_double(),
## ..   ameri_es = col_double(),
## ..   asian = col_double(),
## ..   hawn_pi = col_double(),
## ..   hispanic = col_double(),
## ..   other = col_double(),
## ..   mult_race = col_double(),
## ..   males = col_double(),
## ..   females = col_double(),
## ..   med_age = col_double(),
## ..   households = col_double(),
## ..   families = col_double(),
## ..   hse_units = col_double(),
## ..   ave_fam_sz = col_double(),
## ..   vacant = col_double(),
## ..   owner_occ = col_double(),
```

```

## .. renter_occ = col_double(),
## .. county_fips = col_double()
## .. )

str(cov_vax_admin)

## spec_tbl_df [65,268 x 11] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
## $ X1 : num [1:65268] 1 2 3 4 5 6 7 8 9 10 ...
## $ as_of_date : chr [1:65268] "1/5/2021" "1/5/2021" "1/5/2021" "1/5/2021" ...
## $ zip_code_tabulation_area : num [1:65268] 92703 92285 92284 92275 92532 ...
## $ local_health_jurisdiction : chr [1:65268] "ORANGE" "SAN BERNARDINO" "SAN BERNARDINO" "IMPERIAL" ...
## $ county : chr [1:65268] "ORANGE" "SAN BERNARDINO" "SAN BERNARDINO" "IMPERIAL" ...
## $ vaccine_equity_metric_quartile: num [1:65268] 1 1 1 1 3 1 1 2 2 3 ...
## $ vem_source : chr [1:65268] "Healthy Places Index Score" "Healthy Places Index Score" ...
## $ age12_plus_population : num [1:65268] 57183 2317 22255 2269 19882 ...
## $ persons_fully_vaccinated : num [1:65268] NA NA NA NA NA NA NA 17 28 27 ...
## $ persons_partially_vaccinated : num [1:65268] NA NA NA NA NA ...
## $ redacted : chr [1:65268] "Information redacted in accordance with CA state p
## - attr(*, "spec")=
## .. 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()
## .. )

```

The 5+ data elements/columns/variables we are interested in:

From COVID vax data set: county (county: chr), persons fully vaxxed (persons_fully_vaccinated: num), age 12+ population (age12_plus_population: num), zip code (zip_code_tabulation_area: num).

County is the only character value of interest while the rest of the variables are numeric.

From the Census data set: median age (med_age: num), county name (name: chr), and overall population (pop2012: num).

County name is the only character value of interest while the rest of the variables are numeric.

All the variables we are interested in are in the desired type/format and we will not need to convert any columns to numeric or another type.

```
\newpage
#3 Provide a basic description of the 5+ data elements
Numeric: mean, median, range
Character: unique values/categories
Or any other descriptives that will be useful to the analysis
```

```
'''r
#This is the unweighted mean of the medium ages across each county in
#California
```

```
ca_county_demographic_total_mean_age <- ca_county_demographics %>%
  summarize(total_mean_age = mean(med_age, na.rm = TRUE))
```

```
(view(ca_county_demographic_total_mean_age))
```

```
## # A tibble: 1 x 1
##   total_mean_age
##           <dbl>
## 1           38.5
```

```
# A tibble: 1 x 1
# total_mean_age
#           <dbl>
#1           38.5
```

```
#Exploring the mean age for each county in California
```

```
mean_age_county <- ca_county_demographics %>%
  group_by(name)%>%
  summarize(med_age = mean(sum(med_age,na.rm = TRUE)))
```

```
#Exploring distinct county and zipcodes in California
```

```
unique(cov_vax_admin$county)
```

```
## [1] "ORANGE" "SAN BERNARDINO" "IMPERIAL" "RIVERSIDE"
## [5] "LOS ANGELES" "SAN DIEGO" "TRINITY" "SAN FRANCISCO"
## [9] "TULARE" "MARIN" "CONTRA COSTA" "KERN"
## [13] "VENTURA" "SANTA BARBARA" "SAN MATEO" "SOLANO"
## [17] "FRESNO" "MONTEREY" "SONOMA" "NAPA"
## [21] "ALAMEDA" "MADERA" "KINGS" "INYO"
## [25] "SACRAMENTO" "SAN LUIS OBISPO" "SANTA CLARA" "EL DORADO"
## [29] "GLENN" "YUBA" "BUTTE" "PLACER"
## [33] "AMADOR" "SUTTER" "MONO" "LAKE"
## [37] "YOLO" "HUMBOLDT" "SAN JOAQUIN" "TUOLUMNE"
## [41] "CALAVERAS" "SHASTA" NA "SISKIYOU"
## [45] "LASSEN" "MERCED" "SANTA CRUZ" "SAN BENITO"
## [49] "MODOC" "STANISLAUS" "MENDOCINO" "SIERRA"
## [53] "TEHAMA" "PLUMAS" "MARIPOSA" "DEL NORTE"
## [57] "NEVADA" "ALPINE" "COLUSA"
```

```
unique(cov_vax_admin$zip_code_tabulation_area)
```

```
## [1] 92703 92285 92284 92275 92532 92376 92345 91343 91910 91773 92239 92057
```

##	[13]	92868	92865	92612	92026	92341	92339	95595	92234	91016	91105	91761	91405
##	[25]	91950	91914	92009	92707	92845	92807	92861	92368	92648	92647	92801	92311
##	[37]	92262	92253	92231	92869	92503	92410	91342	91722	92281	92407	92841	92614
##	[49]	92346	91303	91784	91384	92268	92203	92082	92029	92627	92832	92309	94127
##	[61]	92145	92081	94129	94103	92879	92821	92548	92230	93247	92147	92596	92886
##	[73]	92880	94925	92626	94523	94115	92604	93206	92544	92394	92395	92222	92140
##	[85]	92058	92055	92805	92562	92694	92332	92273	92555	92399	91748	91916	92258
##	[97]	92061	92014	92844	92663	92782	92656	92561	92358	92373	92259	92225	92347
##	[109]	92342	92866	92553	92551	92254	91902	93249	93004	93101	93041	94521	94025
##	[121]	94123	94535	92115	93730	93220	93064	93953	93932	94954	91602	94582	94573
##	[133]	94158	94528	94607	94588	94063	92114	93235	93723	93636	93524	93036	93103
##	[145]	91978	91962	91502	94503	94806	94534	94505	94963	94021	94611	94567	92220
##	[157]	92122	93258	93652	93628	93245	93928	93023	94805	91001	91948	91906	91401
##	[169]	94850	94720	94060	94585	94561	93545	93592	93043	91604	91204	91701	91607
##	[181]	91505	91935	91934	92374	92676	92833	92831	92692	92651	92124	92405	92359
##	[193]	91304	91905	92352	92131	92102	92655	92653	92404	95621	92337	91108	91324
##	[205]	91381	91932	92233	92308	93611	93426	93033	93108	94951	91608	94613	94548
##	[217]	94516	94085	94587	94022	92274	94804	92154	93633	93042	93603	93601	91942
##	[229]	91710	91423	91355	91107	91331	91306	91335	95639	95720	95920	94109	93643
##	[241]	93637	93630	93604	93434	93446	93311	93265	93251	92620	91963	95901	95973
##	[253]	95919	95717	95669	95674	94107	94002	92117	90713	90630	93614	93510	93201
##	[265]	93109	94924	95928	94972	90242	93535	93204	93207	93451	93287	93226	93430
##	[277]	91770	91767	95943	95842	93224	92173	92123	92078	90623	93926	93625	93517
##	[289]	93440	93238	93727	93230	93272	91766	91104	92518	92336	92325	92084	94564
##	[301]	94558	95662	91101	91352	91360	91361	91351	91320	91206	92584	91030	91344
##	[313]	91208	92083	94015	95823	95485	91755	91732	91301	91020	91775	91214	95138
##	[325]	92536	93205	93063	93701	91941	95830	95659	95606	95524	91040	91803	91321
##	[337]	91307	95237	92557	91367	92883	92508	92389	95528	95335	95257	95492	95476
##	[349]	91786	91730	91311	95233	95130	92677	92570	92507	92240	91901	91302	92364
##	[361]	92067	92344	92322	92011	92010	92780	94525	94128	92804	91201	93242	93234
##	[373]	93022	93021	92637	92378	92155	92135	92134	94619	94104	94024	94133	92704
##	[385]	94568	94549	94030	94005	94514	94507	92354	92324	92313	93222	92545	92603
##	[397]	92385	92019	92646	94501	94124	92860	92662	92567	92377	91006	93001	92678
##	[409]	92617	92391	94037	94502	94949	93312	93429	91911	91776	95668	95826	94956
##	[421]	90620	93631	93555	93544	93221	93427	93962	93111	94920	94933	93513	93428
##	[433]	93266	93401	93285	93254	93309	93240	92660	93402	93308	92139	92316	92280
##	[445]	91011	93013	92250	92132	92867	94560	94019	94544	94539	94130	94971	92223
##	[457]	94575	90715	93543	93286	92657	92630	92587	91980	95630	95957	95828	95925
##	[469]	94941	93641	95680	95653	96065	95930	95554	95035	97635	95626	96094	96009
##	[481]	95691	95547	95961	95821	95699	95610	95604	95458	95635	95442	95322	95148
##	[493]	95448	95982	95618	95615	96086	96146	95757	95377	96037	95073	95831	95665
##	[505]	95565	95062	95023	96134	96132	95652	95467	95319	95211	95545	95053	95341
##	[517]	95372	95139	96107	96038	95978	95968	95838	95742	95692	95585	95679	95314
##	[529]	95457	95822	95818	95713	95234	95616	95361	95351	95421	96118	96080	96071
##	[541]	95947	95910	95677	95121	95542	95815	95232	95240	95327	95318	95133	95008
##	[553]	95974	95420	95466	96058	96142	96141	95569	95758	95304	95553	95645	95655
##	[565]	95651	96085	96067	96044	96035	95984	95605	95568	95461	95765	95375	95746
##	[577]	95391	95248	95075	95013	95632	95627	96126	96027	95625	95558	95549	95543
##	[589]	96109	96063	96091	96011	96051	96129	96049	96112	95728	95640	95988	95220
##	[601]	96008	96119	96105	95527	96047	96084	96046	96029	96039	96002	96135	96074
##	[613]	95914	95695	95666	96014	96128	96148	95658	96117	96116	96108	96062	96054
##	[625]	96052	96013	96075	96061	96041	96010	95936	95726	95690	96133	96017	96123
##	[637]	96101	96097	96096	96137	96125	96103	96022	96155	96111	95963	95631	96007
##	[649]	96057	96020	96016	96115	96104	95355	96121	95693	95681	95116	95070	96110

##	[661]	96093	96088	95638	96113	96076	96048	95589	95835	95833	96056	96122	95724
##	[673]	95698	95697	95486	95435	95811	96106	96032	96001	96023	96019	95923	95829
##	[685]	95686	95207	95206	95113	95219	95054	95915	95468	96031	96120	96090	96006
##	[697]	96050	95450	96021	96145	95949	95934	95864	95816	95709	90024	95122	95046
##	[709]	95310	95446	96073	96150	95386	94526	94074	94547	94541	94401	90094	90059
##	[721]	90732	90010	95358	95236	94306	94043	90095	90250	90605	90025	95118	95379
##	[733]	95338	95252	95429	90706	90031	90029	95364	95345	95333	95369	94599	94590
##	[745]	94577	94552	90831	90079	90073	90254	95410	90280	90021	95010	95354	95311
##	[757]	95453	96064	96059	95360	94546	94536	94518	90742	91913	90302	90201	95076
##	[769]	94551	94531	94709	94592	92120	93623	93943	93619	93558	93519	93314	90601
##	[781]	91711	91706	91106	91024	91702	93283	93410	93449	93003	94930	94903	91406
##	[793]	94597	90290	96136	96130	96069	95376	95368	95254	95251	94565	94118	90090
##	[805]	90018	90740	90703	90606	90063	90245	90035	95117	94957	95471	95469	96087
##	[817]	95370	95329	95226	94131	94515	90755	95445	95223	95215	95125	95064	96034
##	[829]	96028	96024	95437	95417	95388	95672	95367	95571	95348	95439	95312	95562
##	[841]	96033	96025	96140	95222	95209	95033	95563	95245	95374	95131	95051	95404
##	[853]	95112	95460	95323	95444	95366	95224	95110	95020	95449	95313	95135	96068
##	[865]	96003	96092	95134	95305	95663	95660	95556	95463	95494	95382	95415	95601
##	[877]	96143	95202	95432	95401	95834	90405	95124	95018	95005	95317	95551	96055
##	[889]	95039	95007	95443	95357	94530	94116	90089	90710	95303	90743	90212	95320
##	[901]	94566	94508	94122	94108	94089	94087	90747	90506	90503	90277	95227	95212
##	[913]	95136	95490	96161	95014	95427	95426	95407	95249	95465	95321	95111	95065
##	[925]	95041	92590	95132	95620	95587	96040	96015	96124	96114	95126	95546	95511
##	[937]	95488	95428	95340	92356	92111	92110	92059	92624	92264	92128	93305	93223
##	[949]	93648	93606	93441	90631	93653	93647	92127	92210	92201	92071	92028	92887
##	[961]	92808	93292	93452	90504	93640	92338	92270	92251	92243	93424	93405	93244
##	[973]	93654	93528	94014	93646	92583	92252	92070	92024	92021	92701	92105	93433
##	[985]	92119	93010	93725	94904	94704	90815	90814	93635	92585	92320	92278	92267
##	[997]	92266	92242	92036	94707	94702	90071	90602	92129	92327	92706	92691	92257
##	[1009]	92106	92283	92625	92870	92408	93624	92505	92328	92688	92592	92065	92027
##	[1021]	92020	92679	92586	92543	92108	92363	92256	92025	93522	92277	93518	93657
##	[1033]	92530	92384	92305	92304	92241	92882	92683	92649	92107	93117	93656	93432
##	[1045]	93703	94020	92382	92282	92276	92260	92211	92101	92629	92315	93313	93202
##	[1057]	93015	94111	94305	95383	95363	95334	95326	95970	95946	95926	95951	93665
##	[1069]	93923	93550	94603	94403	93465	95247	95634	95140	95623	92802	93906	93610
##	[1081]	93450	94964	94945	94040	95942	95315	95959	94612	94044	94102	95356	95337
##	[1093]	95948	95912	95328	95246	95966	95938	95960	93706	93644	93460	95935	94010
##	[1105]	95983	95953	95950	95231	95965	95922	95824	95817	95684	95650	95646	93561
##	[1117]	93668	93726	94579	94303	95987	95944	95841	95993	95918	95917	95819	95694
##	[1129]	95673	93905	93702	93705	93551	93277	93436	92595	92411	92003	92307	92675
##	[1141]	92301	92249	92227	92121	92109	95714	95703	95937	93662	93605	93602	93532
##	[1153]	94706	94572	94105	95325	95204	95955	95932	95843	92037	93940	93908	93704
##	[1165]	93563	94121	94070	94028	95956	95230	95762	95689	95664	95939	90248	90001
##	[1177]	93455	93274	93212	93040	93907	93925	93505	90744	93304	90712	90806	90670
##	[1189]	90275	90807	90065	95423	93612	93260	93711	93728	93626	93461	94929	90013
##	[1201]	93627	93552	93530	93526	90016	91792	91768	91733	91506	92130	90232	90210
##	[1213]	90028	90813	90745	90401	90305	90270	90036	93501	93621	93276	93267	93218
##	[1225]	93608	93560	93618	93536	93454	93420	90717	93270	90720	93458	90716	90810
##	[1237]	90804	90064	90017	95425	95403	93422	93261	93720	93667	93651	93638	93924
##	[1249]	94065	90822	93291	90704	90660	90012	90008	95422	95409	93634	93012	93673
##	[1261]	93664	93660	93933	90731	93616	93591	93546	94304	90746	93307	93306	90638
##	[1273]	90034	90404	95387	93921	93669	94086	91605	93252	90501	90066	90061	90402
##	[1285]	90241	93263	93110	93030	93930	93609	94110	94066	90041	94513	90301	90293
##	[1297]	90022	91330	91724	91504	95573	95548	95459	92386	91759	91752	91945	91411


```
## [1309] 91210 95464 94580 94538 91744 91436 91745 94061 94533 92069 90221 90069
## [1321] 90067 90803 92371 92618 92506 92401 92392 92843 92806 92509 92008 92007
## [1333] 92004 92708 92539 92504 92372 90723 91387 91316 95555 95456 92591 94402
## [1345] 91723 91326 91931 91780 91741 91731 91709 91402 91917 94112 92064 90068
## [1357] 92335 95570 92563 92549 92397 94559 94520 94080 91750 91746 91325 91356
## [1369] 91737 91763 91754 94511 90048 91915 91708 91371 95454 92610 94114 91740
## [1381] 92086 91008 91345 91403 91377 92236 92126 90240 90057 94578 94563 94519
## [1393] 91501 91765 94132 93710 90274 90247 90037 90019 91390 91046 91103 95526
## [1405] 95525 95493 94062 91762 91739 90044 93208 94922 95602 90621 90222 90211
## [1417] 90027 90272 95006 95002 95962 95612 93060 90004 95127 90040 90011 90005
## [1429] 90292 90062 90263 90262 90260 95128 95747 95735 95837 95721 93534 94705
## [1441] 94606 94973 91362 91010 93463 93255 93553 93523 94596 94591 93444 93950
## [1453] 94609 93620 93219 93105 93035 95628 95567 95550 90802 90640 90023 90015
## [1465] 90403 90230 93065 90304 90003 95975 95916 95903 95827 95722 95687 95675
## [1477] 93066 94931 94608 94602 90650 93927 95603 90680 90808 90002 90220 90077
## [1489] 90046 90045 90039 90603 90303 90291 90265 90701 90026 90255 90278 90604
## [1501] 90502 95119 95969 95941 95820 93203 93067 90038 93225 90056 90020 90249
## [1513] 95945 95636 95832 95814 95736 93215 95514 90805 90033 90014 95316 91977
## [1525] 91764 95210 92501 92606 92365 92060 94038 95667 95629 95521 95452 95336
## [1537] 95430 95306 91791 91789 91601 91042 91205 91207 91202 92571 92705 92333
## [1549] 92104 92103 91350 92398 92075 92118 94041 94940 95682 95637 95560 95552
## [1561] 95540 95307 91801 91606 94027 95255 92113 92091 95678 95350 95501 91007
## [1573] 91790 92582 92116 92066 95825 95633 93280 93675 91364 91340 93954 93529
## [1585] 93445 93301 93256 93516 93243 94610 93960 93453 93514 93920 93721 93527
## [1597] 93622 93210 94571 94555 91203 93241 93901 93666 93554 93512 93442 94965
## [1609] 94937 94703 94595 94946 93271 93239 93650 93262 93257 93250 94618 93268
## [1621] 95954 93437 93722 93531 93645 93615 94598 94576 94950 91354 93955 93562
## [1633] 93549 93541 92823 94586 94556 95661 95641 92661 92054 90006 95060 95004
## [1645] 95436 94550 94601 95979 95003 94583 95012 95017 95648 92314 90043 95531
## [1657] 94512 94117 90032 90042 90049 95203 94621 95608 95451 95642 95688 95385
## [1669] 94517 95380 95258 95242 92672 94952 94947 94509 94803 95497 94542 95624
## [1681] 95559 92602 95228 95482 95250 95431 95346 95330 94545 95324 95301 95519
## [1693] 95991 94710 94574 95205 95032 92040 92835 95462 95405 95412 95441 95607
## [1705] 90047 95225 92840 95045 95037 94510 92673 92310 92321 94960 94506 92056
## [1717] 95701 95715 94569 92881 90007 95986 95977 95981 95683 95030 95670 95129
## [1729] 90058 95043 95619 95050 95564 95365 95537 95536 95685 94301 94589 94605
## [1741] 95019 94923 94928 94938 95971 94134 94939 95614 94708 95776 94553 90505
## [1753] 90266 94801 95470 95472 95066 94901 94404 95389 95503 95123 94970 95120
```

#Exploring the total number of persons totally vaccinated per County

```
total_vaccinated_california <- cov_vax_admin %>%
group_by(county)%>%
summarize(total_vax = sum(persons_fully_vaccinated,na.rm = TRUE))
```

#Exploring the total population for each County

```
total_pop_california <- ca_county_demographics %>%
group_by(name)%>%
summarize(total_pop = sum(pop2012, na.rm = TRUE))
```

#Exploring the total number of persons age 12 and above per County

```
total_age12andabove_california <- cov_vax_admin %>%  
group_by(county)%>%  
summarize(total_age12andabove = sum(age12_plus_population,na.rm = TRUE))
```