Team project

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

Dataset Description

Scenario 2 description: Scenario 2: COVID Vaccination Updates in California You are part of a team who has been tasked with monitoring state level COVID-19 vaccination rates in the state of California. You have access to a *California Vaccine Progress Dashboard Data* that reports percentages of vaccinated persons by zip code. You have been tasked with organizing this data into county level observations and reporting them with a print quality data visualization. It's understood that certain values have been redacted, or there may be other missing counts of vaccinated persons on the zipcode level. In those cases you've been asked to compute a county level average to replace the 'NA' values in the zipcode observations. There's a concern going around that counties with younger median age values are less likely to have higher counts of vaccinated persons. You understand that it's difficult to investigate the question authoritatively at this time, but you decide to explore some CA census demographics to compare median age values and proportions of vaccinated persons together on a county level. You are looking for preliminary evidence of a *strong correlative relationship* between the two variables together, on a county level of observation, and the direction of that relationship, if any. Your task is to explore the Vaccine Progress Dashboard and locate the most recent vaccination data, including a percentage of fully vaccinated persons. If you find any missing values, you'll need to find a way to do *mean imputation* with the missing values on a zip code level, (substitute an average value to fill in the NAs values from the county level averages). Next you'll need to summarise those imputed zipcode observations into a county level summary, averaging again, and create a print quality visualization that estimates a total proportion of vaccinated persons, by county, rounded to two places. Next you'll need to explore the California Demographics dataset and locate a median age statistic `med_age` and join that statistic with your vaccinated persons data. Once that is complete, plot a visualization that's able to explore potential correlations between median age and vaccinated person prevalence on a county level, if any. Data Sources: • PHW251 Box folder (Links to an external site.) o cov_vax_admin.csv (original source: https://data.ca.gov/dataset/covid-19-vaccine-progress-dashboard-data-by-zip-code (Links to an external site.)) o ca_county_demographics.csv (original source: https://raw.githubusercontent.com/Averysaurus/reproducable_examples-/main/ca_county_demographics.csv (Links to an external site.))

======= ## Import statement

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

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   X1 = col_double(),
##   as_of_date = col_date(format = ""),
##   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()
## )
```

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

```
names(ca_county_demographics) [names(ca_county_demographics) == "X1"] <- "No."
names(cov_vax_admin) [names(cov_vax_admin) == "X1"] <- "No."
view(cov_vax_admin)
view(ca_county_demographics)
str(cov_vax_admin)
```

```
## spec_tbl_df [65,268 x 11] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ No.                          : num [1:65268] 1 2 3 4 5 6 7 8 9 10 ...
##  $ as_of_date                   : Date[1:65268], format: "2021-01-05" "2021-01-05" ...
##  $ 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 S
##  $ 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 pi
##  - attr(*, "spec")=
##   .. cols(
##   ..   X1 = col_double(),
##   ..   as_of_date = col_date(format = ""),
##   ..   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()
##   .. )
```

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

One dataset is California county demographic information. Another dataset is Covid-19 vaccine administration in 2021.

```r
str(cov_vax_admin)
```

```
## spec_tbl_df [65,268 x 11] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ No.                           : num [1:65268] 1 2 3 4 5 6 7 8 9 10 ...
##  $ as_of_date                    : Date[1:65268], format: "2021-01-05" "2021-01-05" ...
##  $ 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 S
##  $ 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 pr
##  - attr(*, "spec")=
##   .. cols(
##   ..   X1 = col_double(),
##   ..   as_of_date = col_date(format = ""),
##   ..   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()
##   .. )
```

Gives summary of values within columns. Allows us to quickly identify date range for Cov vac as Jan 05 to Sep 14, 2021.

```r
summary(cov_vax_admin)
```

```
##       No.          as_of_date         zip_code_tabulation_area
##  Min.   :    1   Min.   :2021-01-05   Min.   :90001
##  1st Qu.:16318   1st Qu.:2021-03-09   1st Qu.:92258
##  Median :32634   Median :2021-05-11   Median :93658
##  Mean   :32634   Mean   :2021-05-11   Mean   :93665
##  3rd Qu.:48951   3rd Qu.:2021-07-13   3rd Qu.:95380
##  Max.   :65268   Max.   :2021-09-14   Max.   :97635
##
##  local_health_jurisdiction    county          vaccine_equity_metric_quartile
##  Length:65268              Length:65268       Min.   :1.000
##  Class :character          Class :character   1st Qu.:1.000
##  Mode  :character          Mode  :character   Median :2.000
##                                               Mean   :2.436
##                                               3rd Qu.:3.000
##                                               Max.   :4.000
##                                               NA's   :3219
##   vem_source        age12_plus_population persons_fully_vaccinated
##  Length:65268       Min.   :    0         Min.   :   11
##  Class :character   1st Qu.: 1347         1st Qu.:  402
##  Mode  :character   Median :13685         Median : 3081
##                     Mean   :18895         Mean   : 8029
##                     3rd Qu.:31756         3rd Qu.:13154
##                     Max.   :88557         Max.   :67594
##                                           NA's   :7037
##  persons_partially_vaccinated   redacted
##  Min.   :   11.0              Length:65268
##  1st Qu.:  221.5              Class :character
##  Median : 1419.0              Mode  :character
##  Mean   : 2199.2
##  3rd Qu.: 3306.0
##  Max.   :23195.0
##  NA's   :7037
```

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

Ans: The datasets provide information on total potential population (demographics) and Covid vaccination (exposure to treatment).

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

as_of_date, zip_code_tabulation_area, local_health_jurisdiction, age12_plus_population, persons_fully_vaccinated, persons_partially_vaccinated

```r
class(cov_vax_admin$as_of_date)
```

```
## [1] "Date"
```

```r
typeof(cov_vax_admin$as_of_date)
```

```
## [1] "double"
```

```r
class(cov_vax_admin$zip_code_tabulation_area)
```

```
## [1] "numeric"
```

```r
typeof(cov_vax_admin$zip_code_tabulation_area)
```

```
## [1] "double"
```

```r
class(cov_vax_admin$local_health_jurisdiction)
```

```
## [1] "character"
```

```r
typeof(cov_vax_admin$local_health_jurisdiction)
```

```
## [1] "character"
```

```r
class(cov_vax_admin$age12_plus_population)
```

```
## [1] "numeric"
```

```r
typeof(cov_vax_admin$age12_plus_population)
```

```
## [1] "double"
```

```r
# data type is "double"/numeric therefore no need to change data type
class(cov_vax_admin$persons_fully_vaccinated)
```

```
## [1] "numeric"
```

```r
typeof(cov_vax_admin$persons_fully_vaccinated)
```

```
## [1] "double"
```

```r
# data type is "double"/numeric therefore no need to change data type
class(cov_vax_admin$persons_partially_vaccinated)
```

```
## [1] "numeric"
```

```r
typeof(cov_vax_admin$persons_partially_vaccinated)
```

```
## [1] "double"
```

```r
# data type is "double"/numeric yet given that there are only whole integer numbers of persons we may
```

Provide a basic description of the 5+ data elements

1. Numeric: mean, median, range
2. Character: unique values/categories
3. Or any other descriptives that will be useful to the analysis

```
summary(cov_vax_admin$as_of_date)
```

```
##         Min.      1st Qu.       Median         Mean      3rd Qu.         Max.
## "2021-01-05" "2021-03-09" "2021-05-11" "2021-05-11" "2021-07-13" "2021-09-14"
```

```
range(cov_vax_admin$as_of_date)
```

```
## [1] "2021-01-05" "2021-09-14"
```

```
summary(cov_vax_admin$zip_code_tabulation_area)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   90001   92258   93658   93665   95380   97635
```

```
range(cov_vax_admin$zip_code_tabulation_area)
```

```
## [1] 90001 97635
```

```
summary(cov_vax_admin$local_health_jurisdiction)
```

```
##    Length     Class      Mode
##     65268 character character
```

```
range(cov_vax_admin$local_health_jurisdiction)
```

```
## [1] NA NA
```

```
summary(cov_vax_admin$age12_plus_population)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       0    1347   13685   18895   31756   88557
```

```
range(cov_vax_admin$age12_plus_population)
```

```
## [1]     0.0 88556.7
```

```
summary(cov_vax_admin$persons_fully_vaccinated)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##      11     402    3081    8029   13154   67594    7037
```

```
range(cov_vax_admin$persons_fully_vaccinated)
```

```
## [1] NA NA
```

```
summary(cov_vax_admin$persons_partially_vaccinated)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    11.0   221.5  1419.0  2199.2  3306.0 23195.0    7037
```

```
range(cov_vax_admin$persons_partially_vaccinated)
```

```
## [1] NA NA
```

Unique values/categories for date, zip code, and county

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

```
##  [1] "2021-01-05" "2021-01-12" "2021-01-19" "2021-01-26" "2021-02-02"
##  [6] "2021-02-09" "2021-02-16" "2021-02-23" "2021-03-02" "2021-03-09"
## [11] "2021-03-16" "2021-03-23" "2021-03-30" "2021-04-06" "2021-04-13"
## [16] "2021-04-20" "2021-04-27" "2021-05-04" "2021-05-11" "2021-05-18"
## [21] "2021-05-25" "2021-06-01" "2021-06-08" "2021-06-15" "2021-06-22"
## [26] "2021-06-29" "2021-07-06" "2021-07-13" "2021-07-20" "2021-07-27"
## [31] "2021-08-03" "2021-08-10" "2021-08-17" "2021-08-24" "2021-08-31"
## [36] "2021-09-07" "2021-09-14"
```

```
unique(cov_vax_admin$zip_code_tabulation_area, nmax=NA)
```

```
##    [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
```

```r
unique(cov_vax_admin$local_health_jurisdiction)
```

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

Milestone 3

```
# Seems like cleaning out the 'NA' values is needed more than subsetting
# Subsetting by county may be necessary
# We may only need the most recent data - we don't have to show the trend right? Using the most recent
# Once we get the county average vaccination counts, put that number for the missing "NA"s, averaging r
df <- cov_vax_admin %>%
  select(No., as_of_date,zip_code_tabulation_area,local_health_jurisdiction,age12_plus_population,perso
  filter(as_of_date == as.Date("2021-09-14"))
view(df)
```

1. Subset rows or columns as needed

2. Create minimum of 2 new variables Calculating a rate Combining character strings

```
# From Scenario 2 summary:
# Explore some CA census demographics to compare median age values and proportions of vaccinated person
# Compute a county level average to replace the 'NA' values in the zipcode observations
df2 <- df%>%
  mutate(vaccination_rate = persons_fully_vaccinated/age12_plus_population)%>%
  mutate(vaccination_rate_percentage = vaccination_rate*100)
view(df2)
df3 <- df2 %>%
  group_by(local_health_jurisdiction)%>%
  mutate(total_vaccinated_persons_by_county=sum(persons_fully_vaccinated,na.rm=TRUE)) %>%
  mutate(total_population_by_county=sum(age12_plus_population,na.rm=TRUE))%>%
  ungroup()%>%
  mutate(persons_fully_vaccinated=if_else(is.na(persons_fully_vaccinated),age12_plus_population*total_va
view(df3)

# Select columns from Covid vaccination to prepare for joining with demographics table.
# Compute a county level average to replace the 'NA' values in the zipcode observations. Sort by county
df4 <- df3 %>%
  mutate(total_vx_rate = total_vaccinated_persons_by_county / total_population_by_county)%>%
  mutate(total_vx_rate_prcnt = total_vx_rate*100) %>%
  arrange(df3$local_health_jurisdiction)
 # cumsum(local_health_jurisdiction)
view(df4)

# Check that local health jurisdictions (i.e. county) would be same as county for population.
# unique(df4$local_health_jurisdiction)
# unique(df_ca_co_demog$name)
# They are not the sample batch of names, plus one data set is all caps and the other is not.

#df4 <- select(df3, c(local_health_jurisdiction, age12_plus_population, persons_fully_vaccinated, ))

# From CA demographics, extract simpler table. Convert to tibble. Show file.
df_ca_co_demog <- select(ca_county_demographics, No., name, pop2012, med_age)
df_ca_co_demog <- as_tibble(df_ca_co_demog)
# arrange(df_ca_co_demog$name)
df_ca_co_demog
```

```
## # A tibble: 58 x 4
##      No. name        pop2012 med_age
##    <dbl> <chr>         <dbl>   <dbl>
##  1     1 Kern         851089    30.7
##  2     2 Kings        155039    31.1
##  3     3 Lake          65253    45
##  4     4 Lassen        35039    37
##  5     5 Los Angeles 9904341    34.8
##  6     6 Madera       153025    33.1
##  7     7 Marin        255509    44.5
##  8     8 Mariposa      18455    49.2
##  9     9 Mendocino     88094    41.6
## 10    10 Merced       256841    29.6
## # ... with 48 more rows
```

```r
# Clean rows where county is NA.
# is.na() returns missing; !is.na() returns non-missing
# filter(df4$local_health_jurisdiction, !is.na())
# df5 <- filter(df4, local_health_jurisdiction !%in% c("NA")
# df5 <- filter(df4, local_health_jurisdiction == "KERN" | local_health_jurisdiction == "KINGS" |)

# Code above not working to clean "NA" rows, therefore brute force below to standardize counties across

df5 <- filter(df4, local_health_jurisdiction %in% c("KERN", "KINGS", "LAKE", "LASSEN", "LOS ANGELES",
                                                    "MADERA", "MARIN", "MARIPOSA", "MENDOCINO",
                                                    "MERCED", "MODOC", "MONO", "MONTEREY", "NAPA",
                                                    "NEVADA", "ORANGE", "PLACER", "PLUMAS",
                                                    "RIVERSIDE", "SACRAMENTO", "SAN BENITO",
                                                    "SAN BERNARDINO", "SAN DIEGO",
                                                    "SAN FRANCISCO", "SAN JOAQUIN",
                                                    "SAN LUIS OBISPO", "SAN MATEO",
                                                    "SANTA BARBARA", "SANTA CLARA", "SANTA CRUZ",
                                                    "SHASTA", "SIERRA", "SISKIYOU", "SOLANO",
                                                    "ALAMEDA", "ALPINE", "SONOMA", "AMADOR",
                                                    "STANISLAUS", "SUTTER", "BUTTE", "CALAVERAS",
                                                    "TEHAMA", "COLUSA", "TRINITY", "TULARE",
                                                    "CONTRA COSTA", "DEL NORTE", "TUOLUMNE",
                                                    "VENTURA", "EL DORADO", "YOLO", "FRESNO",
                                                    "GLENN", "YUBA", "HUMBOLDT","IMPERIAL",
                                                    "INYO"))

view(df5)
```

3. Clean minimum of 2 variables for analysis

4. Data dictionary based on clean dataset (minimum 4 data elements), including: Variable name Data type Description

```
# https://cran.r-project.org/web/packages/dataMeta/vignettes/dataMeta_Vignette.html
library(dataMeta)

data(df_ca_co_demog)
```

```
## Warning in data(df_ca_co_demog): data set 'df_ca_co_demog' not found
```

```
variable.names <- c("No.", "name", "pop2012", "med_age")
var_desc <- c("County number", "County name",
              "Population (2012)", "Median age")
var_type <- c(0, 1, 0, 0)

linker <- build_linker(my.data = df_ca_co_demog, variable_description = var_desc, variable_type = var_ty
```

```r
#install.packages("arsenal")
library(arsenal)
```

5. One or more tables with descriptive statistics for 4 data elements

```
##
## Attaching package: 'arsenal'
```

```
## The following object is masked from 'package:lubridate':
##
##     is.Date
```

```r
table_one <- tableby(name ~ med_age + pop2012, data = df_ca_co_demog)
```

```
## Warning in anova.lm(aov.out): ANOVA F-tests on an essentially perfect fit are
## unreliable

## Warning in anova.lm(aov.out): ANOVA F-tests on an essentially perfect fit are
## unreliable
```

```r
summary(table_one, title = "Median Age by County and Population in 2012")
```

```
##
##
## Table: Median Age by County and Population in 2012
##
## |                           |       Alameda (N=1)       |    Alpine (N=1)     |     Amador (N=1)
## |:--------------------------|:--------------------------:|:-------------------:|:--------------------
## |**med_age**                |                           |                     |
## |   Mean (SD) |        36.600 (NA)        |     46.400 (NA)     |      48.200 (NA)
## |   Range     |      36.600 - 36.600      |   46.400 - 46.400   |    48.200 - 48.200
## |**pop2012**                |                           |                     |
## |   Mean (SD) |     1534551.000 (NA)      |    1148.000 (NA)    |    38354.000 (NA)
## |   Range     | 1534551.000 - 1534551.000 | 1148.000 - 1148.000 | 38354.000 - 38354.0
```