Reflection and Discussion for Problem Set #4

Ramon, Lupe, Zhaopeng

10/26/2018

Agenda

- Group presentation on our particular experiences, such as challenges/problems/issues (10-15 minutes)

- Open it to class for discussion of their experiences (15-20 minutes)
    - Groups of 3-4 (5 minutes)
    - Bring it back to entire classroom (10-15 minutes)

`select` versus `filter`

- Reminder…
  - `select` retains or drops particular vectors
  - `filter` retains particular observations that meet specified conditions of a vector
    - important to think about the impact these functions (including all others) have when assigning to a specified dataframe

Part I-Question 1

- *Count the number of observations that have `NA` for the variable `state`*

Multiple ways to achieve the purpose.

(1) **count(dataframe, vector)**

(2) **count(dataframe, is.na(vector))**

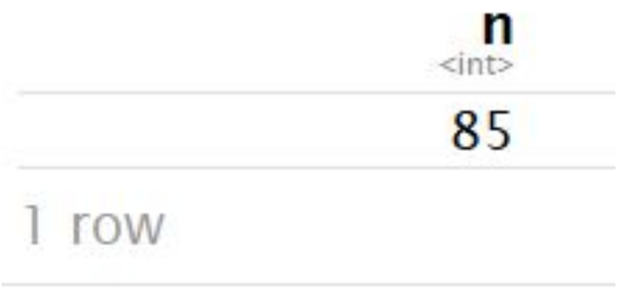
(3) **table(dataframe$vector, useNA = “always”)**

(4) **dataframe %>% count(vector)**

(5) **dataframe %>% count(is.na(vector))**

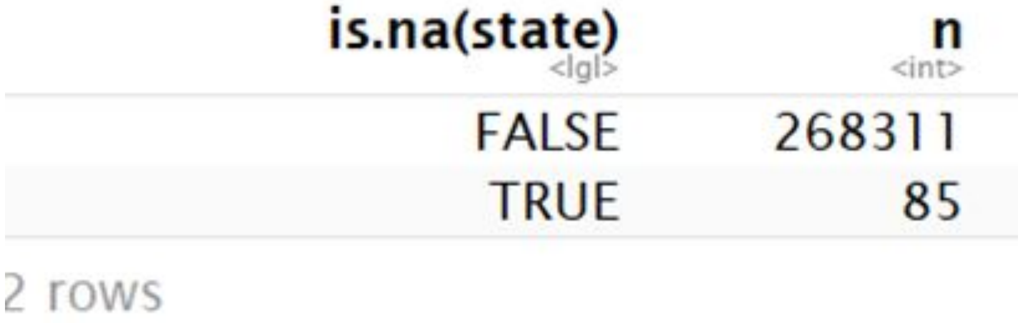
(6) **dataframe %>% filter(is.na(vector)) %>% count(vector)**

(1)**wwlist %>% filter(is.na(state)) %>% count()**



```
     n
 <int>
    85
1 row
```

(2)**wwlist %>% count (is.na(state))**

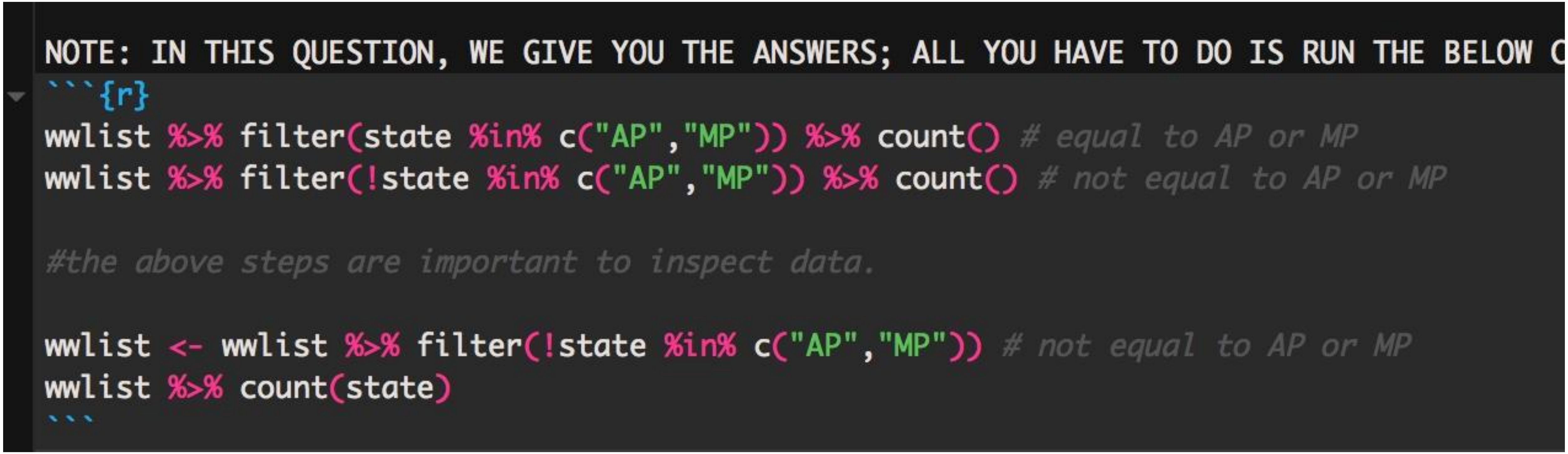


```
is.na(state)        n
       <lgl>    <int>
       FALSE   268311
        TRUE       85
2 rows
```

Part I, Question 2

- Why did we inspect and filter out observations for `pop_total_zip` equals to 0?
  - Cannot yield percentages of races/ethnicities in zip code?
  - May indicate a data entry error?

Part I, Question 3: Importance of Inspecting Data



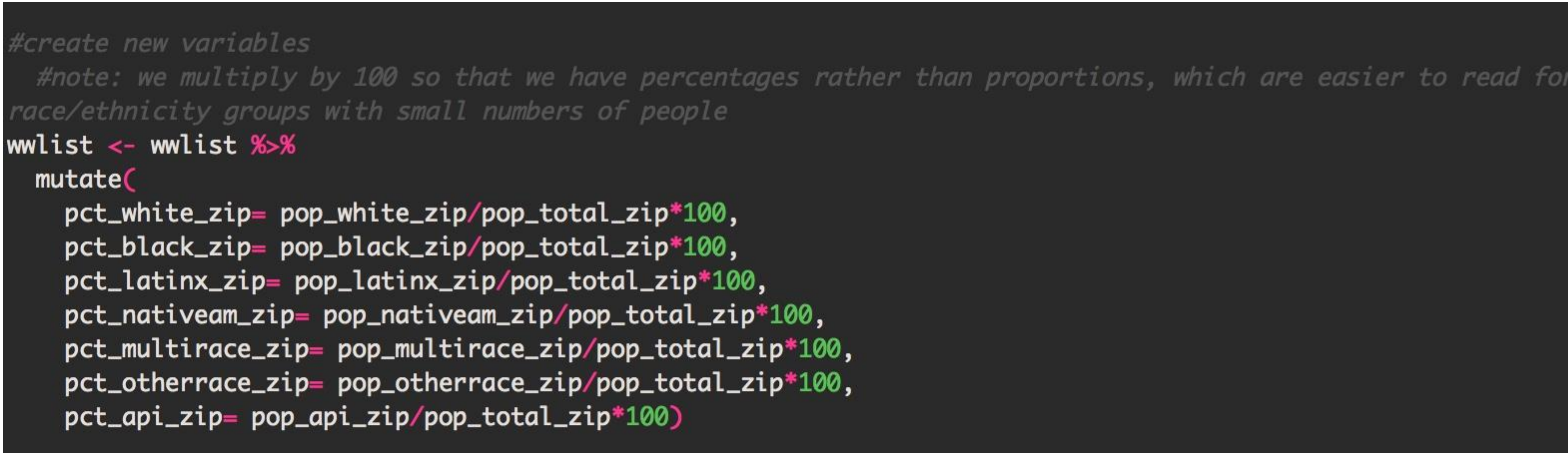
~~~
NOTE: IN THIS QUESTION, WE GIVE YOU THE ANSWERS; ALL YOU HAVE TO DO IS RUN THE BELOW C
```{r}
wwlist %>% filter(state %in% c("AP","MP")) %>% count() # equal to AP or MP
wwlist %>% filter(!state %in% c("AP","MP")) %>% count() # not equal to AP or MP

#the above steps are important to inspect data.

wwlist <- wwlist %>% filter(!state %in% c("AP","MP")) # not equal to AP or MP
wwlist %>% count(state)
```
~~~

Part II, question 4 Importance of Inspecting Data

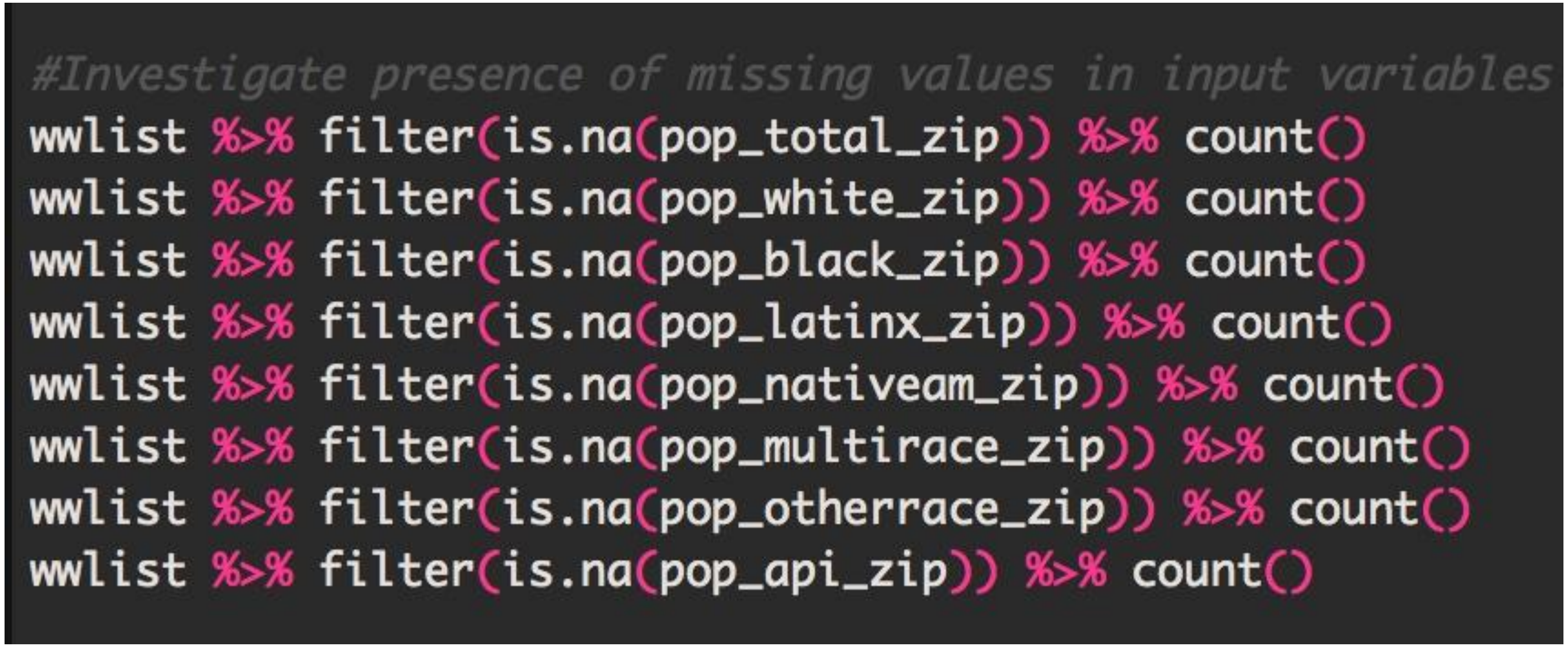
Create Variable:



```
#create new variables
  #note: we multiply by 100 so that we have percentages rather than proportions, which are easier to read fo
race/ethnicity groups with small numbers of people
wwlist <- wwlist %>%
  mutate(
    pct_white_zip= pop_white_zip/pop_total_zip*100,
    pct_black_zip= pop_black_zip/pop_total_zip*100,
    pct_latinx_zip= pop_latinx_zip/pop_total_zip*100,
    pct_nativeam_zip= pop_nativeam_zip/pop_total_zip*100,
    pct_multirace_zip= pop_multirace_zip/pop_total_zip*100,
    pct_otherrace_zip= pop_otherrace_zip/pop_total_zip*100,
    pct_api_zip= pop_api_zip/pop_total_zip*100)
```

Part II, Question 4: Importance of Inspecting Data

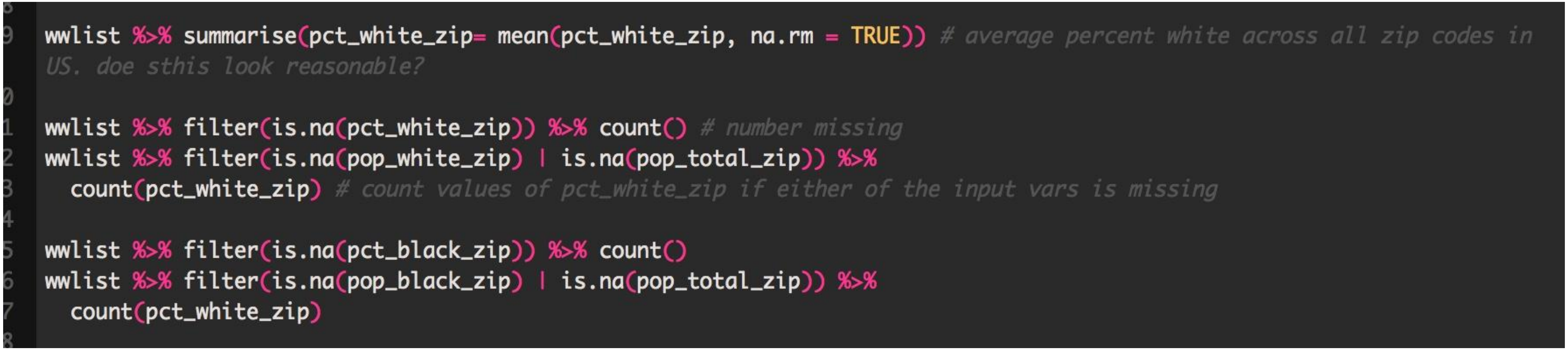
check data before for
missing values



```
#Investigate presence of missing values in input variables
wwlist %>% filter(is.na(pop_total_zip)) %>% count()
wwlist %>% filter(is.na(pop_white_zip)) %>% count()
wwlist %>% filter(is.na(pop_black_zip)) %>% count()
wwlist %>% filter(is.na(pop_latinx_zip)) %>% count()
wwlist %>% filter(is.na(pop_nativeam_zip)) %>% count()
wwlist %>% filter(is.na(pop_multirace_zip)) %>% count()
wwlist %>% filter(is.na(pop_otherrace_zip)) %>% count()
wwlist %>% filter(is.na(pop_api_zip)) %>% count()
```

Part II, Question 4: Importance of Inspecting Data

comparing new variables to those old variables used to compute new variable



```
wwlist %>% summarise(pct_white_zip= mean(pct_white_zip, na.rm = TRUE)) # average percent white across all zip codes in
US. doe sthis look reasonable?

wwlist %>% filter(is.na(pct_white_zip)) %>% count() # number missing
wwlist %>% filter(is.na(pop_white_zip) | is.na(pop_total_zip)) %>%
  count(pct_white_zip) # count values of pct_white_zip if either of the input vars is missing

wwlist %>% filter(is.na(pct_black_zip)) %>% count()
wwlist %>% filter(is.na(pop_black_zip) | is.na(pop_total_zip)) %>%
  count(pct_white_zip)
```

Part II, Question 7

- Why was `ethn_race` used to generate 1,0 vectors for each individual race/ethnicity?
  - Does it have to do with how each can be used with the `summarise` function later in the problem set? Are 1,0 vectors better to use for calculations instead of character vectors with categories?
    - Part III, question 5

mutate() and summarise()

- `mutate()` is creating new variables which are added into the data set

```
wwlist <- wwlist %>% mutate(pop_api_zip = pop_asian_zip +
pop_nativehawaii_zip)
```

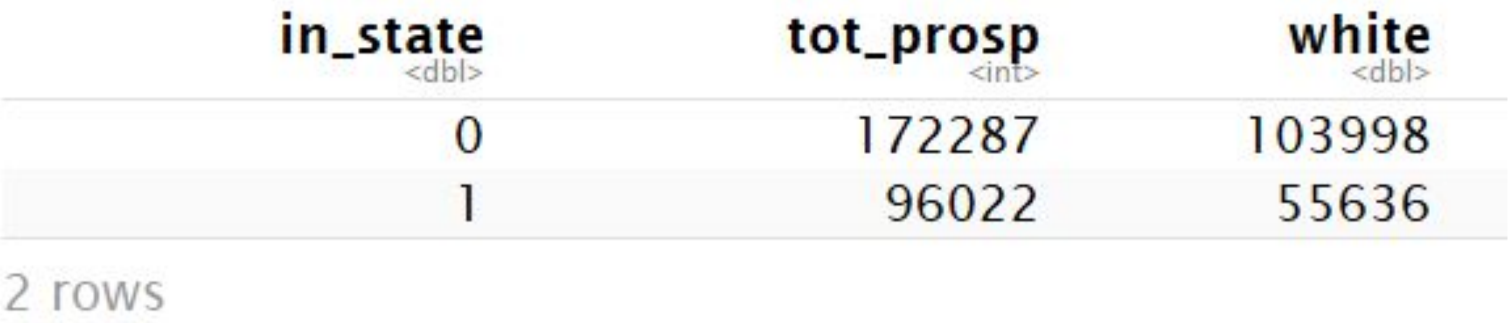
- `summarise()` is creating new variables that have summary statistics (e.g. mean, numbers, min, max, standard deviation) as their values (collapsing across rows/observations for particular variable in the data set);

```
wwlist %>% summarise(pct_white_zip= mean(pct_white_zip, na.rm =
TRUE))
```

mutate() and summarise()

- `group_by` is not required for ‘summarize ()’; when `summarise` is used without `group_by`, the summary statistics are computed based on all observations, when used along with ‘group_by’, the summary statistics are computed based on the observations within each group.

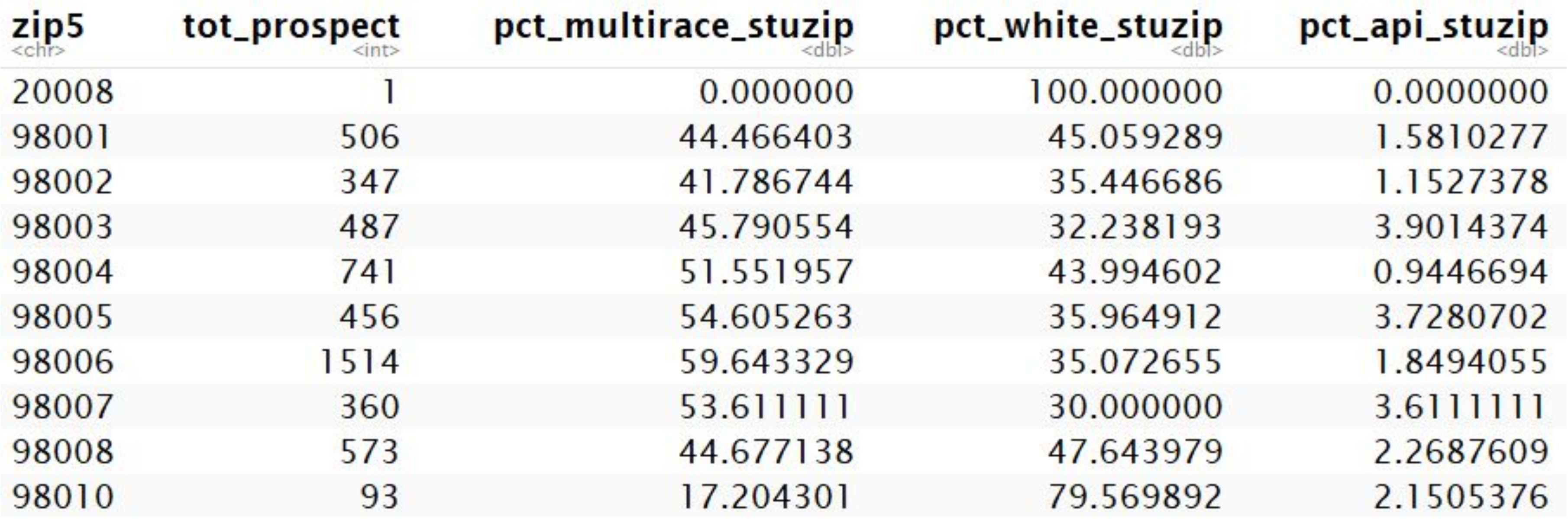
```
wwlist%>%group_by(in_state)%>%summarise(tot_prosp=n(),white=sum(white_stu,na.rm=TRUE))
```



```
 in_state   tot_prosp    white
    <dbl>       <int>    <dbl>
        0      172287   103998
        1       96022    55636
2 rows
```

FYI-Part IV-Question 4

When I check the data, I use the function: count(wwlist, state)



zip5	tot_prospect	pct_multirace_stuzip	pct_white_stuzip	pct_api_stuzip
<chr>	<int>	<dbl>	<dbl>	<dbl>
20008	1	0.000000	100.000000	0.0000000
98001	506	44.466403	45.059289	1.5810277
98002	347	41.786744	35.446686	1.1527378
98003	487	45.790554	32.238193	3.9014374
98004	741	51.551957	43.994602	0.9446694
98005	456	54.605263	35.964912	3.7280702
98006	1514	59.643329	35.072655	1.8494055
98007	360	53.611111	30.000000	3.6111111
98008	573	44.677138	47.643979	2.2687609
98010	93	17.204301	79.569892	2.1505376

Knit to pdf versus html

- When knit to pdf, the data view is not completely shown (e.g. Question 1-4 in Part IV)
- When knit to html, the data view is complete