Lecture 3 problem set

INSERT YOUR NAME HERE

October 11, 2018

Required reading and instructions

Required reading

- Grolemund and Wickham 5.5 (Add new variables with `mutate()`)
- Xie, Allaire, and Grolemund (XAG) section 3.3 (R Markdown, PDF document) LINK HERE

General instructions

In this homework, you will specify `pdf_document` as the output format. You must have LaTeX installed in order to create pdf documents.

If you have not yet installed MiKTeX/MacTeX, I recommend installing TinyTeX, which is much simpler to install!

- Instructions for installation of TinTeX can be found HERE
- General Instructions for Problem Sets Here

Make changes to YAML header

Read XAG section 3.3 before answering these questions

1. Add a table of contents to YAML header
2. table of contents should have "depth" of 2
3. Add section numbering to headers
4. Change "data frame printing" option to "tibble"

Load packages, load data, and rename variables

1. Load the tidyverse package

```
#install.packages("tidyverse") #install if you do not have tidyverse installed
library(tidyverse)
#> -- Attaching packages ---------------------------------------------------------------------------
#> v ggplot2 3.0.0     v purrr   0.2.5
#> v tibble  1.4.2     v dplyr   0.7.6
#> v tidyr   0.8.1     v stringr 1.3.1
#> v readr   1.1.1     v forcats 0.3.0
#> -- Conflicts ------------------------------------------------------------------------------------ tid
#> x dplyr::filter() masks stats::filter()
#> x dplyr::lag()    masks stats::lag()
```

2. Load the data frame data frame `df_school_all`

- The URL for this data frame is: (https://github.com/ozanj/rclass/raw/master/data/recruiting/recruit_school_allvars.RData)
- The data frame `df_school_all` has one observation for each high school (public and private).
- The variables that begin with `visits_by_...` identify how many off-campus recruiting visits the high school received from a particular public university. For example, UC Berkeley has the ID `110635` so the variable `visits_by_110635` identifies how many visits the high school received from UC Berkeley.
- The variable `total_visits` identifies the number of visits the high school received from all (16) public research universities in this data collection sample.

```
load(url("https://github.com/ozanj/rclass/raw/master/data/recruiting/recruit_school_allvars.RData"))
```

3. Run the following code which drops some variables, renames other variables, and assigns these changes to the existing object `df_school_all` and then print the names of all the variables using the `names()` function.

```
df_school_all <- df_school_all %>%
  select(-contains("inst_")) %>% # remove vars that start with "inst_"
  rename(
    visits_by_berkeley = visits_by_110635,
    visits_by_boulder = visits_by_126614,
    visits_by_bama = visits_by_100751,
    visits_by_stonybrook = visits_by_196097,
    visits_by_rutgers = visits_by_186380,
    visits_by_pitt = visits_by_215293,
    visits_by_cinci = visits_by_201885,
    visits_by_nebraska = visits_by_181464,
    visits_by_georgia = visits_by_139959,
    visits_by_scarolina = visits_by_218663,
    visits_by_ncstate = visits_by_199193,
    visits_by_irvine = visits_by_110653,
    visits_by_kansas = visits_by_155317,
    visits_by_arkansas = visits_by_106397,
    visits_by_sillinois = visits_by_149222,
    visits_by_umass = visits_by_166629,
    num_took_read = num_took_rla,
    num_prof_read = num_prof_rla,
    med_inc = avgmedian_inc_2564
  )

names(df_school_all)
#>  [1] "state_code"          "school_type"         "ncessch"
#>  [4] "name"                "address"             "city"
#>  [7] "zip_code"            "pct_white"           "pct_black"
#> [10] "pct_hispanic"        "pct_asian"           "pct_amerindian"
#> [13] "pct_other"           "num_fr_lunch"        "total_students"
#> [16] "num_took_math"       "num_prof_math"       "num_took_read"
#> [19] "num_prof_read"       "med_inc"             "latitude"
#> [22] "longitude"           "visits_by_stonybrook" "visits_by_rutgers"
#> [25] "visits_by_pitt"      "visits_by_cinci"     "visits_by_nebraska"
#> [28] "visits_by_georgia"   "visits_by_scarolina" "visits_by_bama"
#> [31] "visits_by_ncstate"   "visits_by_berkeley"  "visits_by_irvine"
#> [34] "visits_by_boulder"   "visits_by_kansas"    "visits_by_arkansas"
#> [37] "visits_by_sillinois" "visits_by_umass"     "total_visits"
```

Filter and arrange questions

For the questions below, imagine that you have been asked by a major news outlet to identify which high schools receive the most off-campus recruiting visits from the 16 public universities in the sample. Therefore, you will focus on the variable `total_visits`, which counts the total number of visits to the high school across all public 16 public research universities in the sample

- For questions that ask you to print the "top 10" observations, you can either:
  - just print the object and rely on the fact that the default option for printing tibbles is to print the first 10 observations
  - OR you can wrap the command in the `head()` function and explicitly tell R to print 10 observations.

1. Without using pipes (`%>%`), sort (i.e., `arrange()` function) descending by `total_visits` and print the the following variables for the top 10 schools in terms of total number of visits:
   - variables to print: `name`, `state_code`, `city`, `school_type`,`total_visits`, `med_inc`, `pct_white`, `pct_black`, `pct_hispanic`, `pct_asian`, `pct_amerindian`
   - Note: You can do this in one step by wrapping the `select()` function around the `arrange()` (i.e., sort) function; or you can do this in two steps by creating a new data frame first.
2. Answer the question above, but this time use pipes (`%>%`) to answer the question in one line of code
3. Without using pipes, print the following (same variables as above):
   - (A) the top 10 public high schools in terms of total number of visits and then
   - (B) the top 10 private high schoools in terms of total number of visits
4. Answer the question above, but this time using pipes (`%>%`) to answer the question in one line of code for part (A) and one line of code for part (B)
5. Using pipe operator (`%>%`), print the following (same variables as above; one line of code for each part (A), (B), (C), (D)):
   - (A) the top 10 public high schools in Massachusetts in terms of total number of visits and then
   - (B) the top 10 private high schools in Massachusetts in terms of total number of visits
   - (C) the top 10 public high schools in California in terms of total number of visits and then
   - (D) the top 10 private high schools in California in terms of total number of visits

Creating variables using mutate()

The focus of this set of questions will be practicing creating some variables from the data frame `df_school_all`. You will be using the `mutate()` function, often combined with the `if_else()` function. Additionally, questions will ask you to investigate the values of "input" variables before creating new "analysis" variables using `mutate()`

Before presenting questions, here are some examples of code that may be useful in checking variable values. The below lines of code count:

- the number of observations in the data frame `df_school_all`
- the number of observations that have missing values for the variable `state_code`
- the number of observations that have missing values for the variable `school_type`
- a frequency count of the variable `school_type`

```
df_school_all %>% count()
#> # A tibble: 1 x 1
#>       n
```

```
#>   <int>
#> 1 21301
count(df_school_all) # same as above
#> # A tibble: 1 x 1
#>       n
#>   <int>
#> 1 21301
df_school_all %>% filter(is.na(state_code)) %>% count() # number with NA for state_code
#> # A tibble: 1 x 1
#>       n
#>   <int>
#> 1     0
df_school_all %>% filter(is.na(school_type)) %>% count() # number with NA for school_type
#> # A tibble: 1 x 1
#>       n
#>   <int>
#> 1     0
df_school_all %>% count(school_type) # frequency count of school_type
#> # A tibble: 2 x 2
#>   school_type     n
#>   <chr>       <int>
#> 1 private      3822
#> 2 public      17479
```

1. Using `mutate()` with `ifelse()` create a 0/1 indicator called `ca_school` that indicates whether the high school is in California and then use `count()` to create a frequency table for the values of `ca_school` (you don't need to assign/retain the new variable)
2. Using `mutate()` with `ifelse()` create a 0/1 indicator called `ca_pub_school` that indicates whether the school is a public high school in California and then use `count()` to create a frequency table for the values of `ca_pub_school` (you don't need to assign/retain the new variable)
3. By combining the `is.na()` function with the `filter()` function, identify the number of observations that have missing values for the following variables:
    - `pct_black`, `pct_hispanic`, `pct_amerindian`
4. Create a new variable pct_bl_hisp_nat that represents the percent of students at the school that identify as black, hispanic, or american indian. Retain this variable by assigning it to the object `df_school_all`
5. Create a new 0/1 indicator variable gt50pct_bl_hisp_nat that identifies whether more than 50% of students identify as black, hispanic, or american indian and create a frequency count of this variable (no need to retain this variable)
6. Create the following 0/1 indicator variables, retain them (assign to object `df_school_all`), and then create frequency counts of these variables:
    - Variable `miss_took_math` for whether the school has missing values for the variable `num_took_math`
    - Variable `miss_prof_math` for whether the school has missing values for the variable `num_prof_math`
    - Variable `miss_took_or_prof_math` for whether the school has missing values for the variable `num_took_math` OR `num_prof_math`
7. create a variable of `pct_prof_math` that measures the percent of students who score proficient in the state math assessment(assign to object `df_school_all`).
8. create a frequency count of value of the variable `pct_prof_math` separately for the three following filters:

- Observations where `miss_took_math==1`
- Observations where `miss_prof_math==1`
- Observations where `miss_took_or_prof_math==1`

case_when() question

For this set of questions, you will work with the data frame `wwlist` which has one observation for each prospective student purchased by Western Washington University from the College Board.

The objective of this set of questions is to create a three-category variable that identifies whether the prospect lives: - (1) in-state (i.e., in Washington), (2) out-of-state but in a US state/territory; (3) not in the US

1. Load the data frame `wwlist` which has information on prospects purchased by Western Washington University

```
load(url("https://github.com/ozanj/rclass/raw/master/data/prospect_list/wwlist_merged.RData"))
```

2. Apply the `str()` function to the variables `state` and `for_country`; and using the `count()` function to create frequency tables for the variables `state`
    - `state`
    - `for_country`
3. Using the `filter()` function and `is.na()` function do the following:
    - count how many missing observations (`NAs`) the variable `state` has
    - count how many missing observations the variable `for_country` has
4. Create a frequency count for the variable `for_country` for the observations where `state` equals `NA` (hint: use the `is.na()`) function
5. Create a frequency count for the variable `for_country` for the observations where `state` does not equal `NA` (hint: use `!is.na()`) function
6. Count the number of observations that have the value "No Response" for the variable `for_country`
7. Using the `case_when` function within `mutate()` create a character variable called `residency` that has the following values: "in_state"; "out_state_us"; "not_in_us"

- This variable should have the value `NA` for observations where `for_country=="No Response"`
- Retain this variable (assign to object `wwlist`) and create a frequency count of this variable

Once finished, knit to (pdf) and upload both .Rmd and PDF files to class website under the week 3 tab
Remeber to use this naming convention "lastname_firstname_ps3"