Lecture 3 problem set

INSERT YOUR NAME HERE

October 11, 2018

Contents

1 Required reading and instructions **1**

1.1 Required reading 1

1.2 General instructions 1

2 Make changes to YAML header **1**

3 Load packages, load data, and rename variables **2**

4 Filter and arrange questions **3**

5 Creating variables using mutate() **9**

6 THIS IS A BONUS QUESTION **13**

7 case_when() question **13**

Grade: /20

1 Required reading and instructions

1.1 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

1.2 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

2 Make changes to YAML header

/0.25

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"

3 Load packages, load data, and rename variables

/0.25

1. Load the tidyverse package

```
#install.packages("tidyverse") #install if you do not have tidyverse installed
library(tidyverse)
#> -- Attaching packages --------------------------------------------------- tidyverse 1.2.1 --
#> v ggplot2 3.1.0       v purrr   0.2.5
#> v tibble  2.1.1       v dplyr   0.8.0.1
#> v tidyr   0.8.3       v stringr 1.4.0
#> v readr   1.3.1       v forcats 0.3.0
#> -- Conflicts ------------------------------------------------------ tidyverse_conflicts() --
#> x dplyr::filter() masks stats::filter()
#> x dplyr::lag()    masks stats::lag()
```

2. Load the data frame data frame `df_school_all`
   /0.25
   - 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
   /0.25

```
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"
```

4 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:
   /1
   - 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.

```
#In one step, use head to print first 10 obs
head(select(arrange(df_school_all,desc(total_visits)),name,state_code,city,school_type,
    total_visits,med_inc,pct_white,pct_black,pct_hispanic,pct_asian,pct_amerindian,
    pct_other),n=10)
#> # A tibble: 10 x 12
#>    name  state_code city  school_type total_visits med_inc pct_white
#>    <chr> <chr>      <chr> <chr>              <int>   <dbl>     <dbl>
#>  1 EPIS~ VA         ALEX~ private               26 109558.      77.8
#>  2 Lyon~ IL         La G~ public                23  94306.      74.1
#>  3 ALLE~ TX         ALLEN public                23 100809      57.2
#>  4 COPP~ TX         COPP~ public                23 123382.     49.9
#>  5 FLOW~ TX         FLOW~ public                22 157234.     74
#>  6 NOLA~ TX         FORT~ private               21  39490.     55.8
#>  7 FORT~ TX         FORT~ private               20  89470.      4.09
#>  8 LOVE~ TX         LUCAS public                19 100809      81.9
#>  9 STRA~ TX         HOUS~ private               18  29630.     56.7
#> 10 TRIN~ TX         ADDI~ private               18  77380      83.5
#> # ... with 5 more variables: pct_black <dbl>, pct_hispanic <dbl>,
#> #   pct_asian <dbl>, pct_amerindian <dbl>, pct_other <dbl>

#in one step, without using head()
select(arrange(df_school_all,desc(total_visits)),name,state_code,city,school_type,
       total_visits,med_inc,pct_white,pct_black,pct_hispanic,pct_asian,
       pct_amerindian,pct_other)
#> # A tibble: 21,301 x 12
#>    name  state_code city  school_type total_visits med_inc pct_white
#>    <chr> <chr>      <chr> <chr>              <int>   <dbl>     <dbl>
#>  1 EPIS~ VA         ALEX~ private               26 109558.     77.8
#>  2 Lyon~ IL         La G~ public                23  94306.     74.1
#>  3 ALLE~ TX         ALLEN public                23 100809      57.2
#>  4 COPP~ TX         COPP~ public                23 123382.     49.9
#>  5 FLOW~ TX         FLOW~ public                22 157234.     74
#>  6 NOLA~ TX         FORT~ private               21  39490.     55.8
#>  7 FORT~ TX         FORT~ private               20  89470.      4.09
#>  8 LOVE~ TX         LUCAS public                19 100809      81.9
#>  9 STRA~ TX         HOUS~ private               18  29630.     56.7
#> 10 TRIN~ TX         ADDI~ private               18  77380      83.5
#> # ... with 21,291 more rows, and 5 more variables: pct_black <dbl>,
#> #   pct_hispanic <dbl>, pct_asian <dbl>, pct_amerindian <dbl>,
#> #   pct_other <dbl>

#in two steps
df_temp <- select(df_school_all,name,state_code,city,school_type,total_visits,med_inc,pct_white,pct_blac
       pct_hispanic,pct_asian,pct_amerindian,pct_other)
head(arrange(df_temp,desc(total_visits)),n=10)
#> # A tibble: 10 x 12
#>    name  state_code city  school_type total_visits med_inc pct_white
#>    <chr> <chr>      <chr> <chr>              <int>   <dbl>     <dbl>
#>  1 EPIS~ VA         ALEX~ private               26 109558.     77.8
#>  2 Lyon~ IL         La G~ public                23  94306.     74.1
#>  3 ALLE~ TX         ALLEN public                23 100809      57.2
#>  4 COPP~ TX         COPP~ public                23 123382.     49.9
#>  5 FLOW~ TX         FLOW~ public                22 157234.     74
#>  6 NOLA~ TX         FORT~ private               21  39490.     55.8
#>  7 FORT~ TX         FORT~ private               20  89470.      4.09
#>  8 LOVE~ TX         LUCAS public                19 100809      81.9
#>  9 STRA~ TX         HOUS~ private               18  29630.     56.7
#> 10 TRIN~ TX         ADDI~ private               18  77380      83.5
#> # ... with 5 more variables: pct_black <dbl>, pct_hispanic <dbl>,
#> #   pct_asian <dbl>, pct_amerindian <dbl>, pct_other <dbl>
rm(df_temp)
```

2. Answer the question above, but this time use pipes (%>%) to answer the question in one line of code

/1

```
df_school_all %>%
  select(name,state_code,city,school_type,total_visits,med_inc,pct_white,pct_black,
       pct_hispanic,pct_asian,pct_amerindian,pct_other) %>%
  arrange(desc(total_visits)) %>%
  head(n=10)
#> # A tibble: 10 x 12
#>    name  state_code city  school_type total_visits med_inc pct_white
#>    <chr> <chr>      <chr> <chr>              <int>   <dbl>     <dbl>
#>  1 EPIS~ VA         ALEX~ private               26 109558.     77.8
#>  2 Lyon~ IL         La G~ public                23  94306.     74.1
#>  3 ALLE~ TX         ALLEN public                23 100809      57.2
#>  4 COPP~ TX         COPP~ public                23 123382.     49.9
#>  5 FLOW~ TX         FLOW~ public                22 157234.     74
#>  6 NOLA~ TX         FORT~ private               21  39490.     55.8
#>  7 FORT~ TX         FORT~ private               20  89470.      4.09
#>  8 LOVE~ TX         LUCAS public                19 100809      81.9
#>  9 STRA~ TX         HOUS~ private               18  29630.     56.7
#> 10 TRIN~ TX         ADDI~ private               18  77380      83.5
#> # ... with 5 more variables: pct_black <dbl>, pct_hispanic <dbl>,
#> #   pct_asian <dbl>, pct_amerindian <dbl>, pct_other <dbl>

# OR you can arrange descending first and then select variables
df_school_all %>%
  arrange(desc(total_visits)) %>%
  select(name,state_code,city,school_type,total_visits,med_inc,pct_white,pct_black,
       pct_hispanic,pct_asian,pct_amerindian,pct_other) %>%
  head(n=10)
#> # A tibble: 10 x 12
#>    name  state_code city  school_type total_visits med_inc pct_white
#>    <chr> <chr>      <chr> <chr>              <int>   <dbl>     <dbl>
#>  1 EPIS~ VA         ALEX~ private               26 109558.     77.8
#>  2 Lyon~ IL         La G~ public                23  94306.     74.1
#>  3 ALLE~ TX         ALLEN public                23 100809      57.2
#>  4 COPP~ TX         COPP~ public                23 123382.     49.9
#>  5 FLOW~ TX         FLOW~ public                22 157234.     74
#>  6 NOLA~ TX         FORT~ private               21  39490.     55.8
#>  7 FORT~ TX         FORT~ private               20  89470.      4.09
#>  8 LOVE~ TX         LUCAS public                19 100809      81.9
#>  9 STRA~ TX         HOUS~ private               18  29630.     56.7
#> 10 TRIN~ TX         ADDI~ private               18  77380      83.5
#> # ... with 5 more variables: pct_black <dbl>, pct_hispanic <dbl>,
#> #   pct_asian <dbl>, pct_amerindian <dbl>, pct_other <dbl>
```

3. Without using pipes, print the following (same variables as above):
   /1.5
   - (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

```
#Public, In one step
head(select(arrange(filter(df_school_all,school_type == "public"),desc(total_visits)),
       name,state_code,city,school_type,total_visits,med_inc,pct_white,pct_black,
       pct_hispanic,pct_asian,pct_amerindian,pct_other),n=10)
#> # A tibble: 10 x 12
#>    name  state_code city  school_type total_visits med_inc pct_white
#>    <chr> <chr>      <chr> <chr>              <int>   <dbl>     <dbl>
#>  1 Lyon~ IL         La G~ public                23  94306.      74.1
#>  2 ALLE~ TX         ALLEN public                23 100809       57.2
#>  3 COPP~ TX         COPP~ public                23 123382.      49.9
#>  4 FLOW~ TX         FLOW~ public                22 157234.      74
#>  5 LOVE~ TX         LUCAS public                19 100809       81.9
#>  6 HIGH~ TX         DALL~ public                17 164063       89.2
#>  7 Barr~ IL         Barr~ public                16 155305       69.1
#>  8 St C~ IL         St C~ public                16  95389       78.5
#>  9 Milt~ GA         Alph~ public                15 113362.      67.5
#> 10 Nape~ IL         Nape~ public                15  92668       65.2
#> # ... with 5 more variables: pct_black <dbl>, pct_hispanic <dbl>,
#> #   pct_asian <dbl>, pct_amerindian <dbl>, pct_other <dbl>
#Public, in multiple steps
df_temp <- filter(df_school_all,school_type == "public")
df_temp2 <- arrange(df_temp,desc(total_visits))
head(select(df_temp2,name,state_code,city,school_type,total_visits,med_inc,pct_white,pct_black,
       pct_hispanic,pct_asian,pct_amerindian,pct_other),n=10)
#> # A tibble: 10 x 12
#>    name  state_code city  school_type total_visits med_inc pct_white
#>    <chr> <chr>      <chr> <chr>              <int>   <dbl>     <dbl>
#>  1 Lyon~ IL         La G~ public                23  94306.      74.1
#>  2 ALLE~ TX         ALLEN public                23 100809       57.2
#>  3 COPP~ TX         COPP~ public                23 123382.      49.9
#>  4 FLOW~ TX         FLOW~ public                22 157234.      74
#>  5 LOVE~ TX         LUCAS public                19 100809       81.9
#>  6 HIGH~ TX         DALL~ public                17 164063       89.2
#>  7 Barr~ IL         Barr~ public                16 155305       69.1
#>  8 St C~ IL         St C~ public                16  95389       78.5
#>  9 Milt~ GA         Alph~ public                15 113362.      67.5
#> 10 Nape~ IL         Nape~ public                15  92668       65.2
#> # ... with 5 more variables: pct_black <dbl>, pct_hispanic <dbl>,
#> #   pct_asian <dbl>, pct_amerindian <dbl>, pct_other <dbl>

rm(df_temp,df_temp2)

#Privates In one step
head(select(arrange(filter(df_school_all,school_type == "private"),desc(total_visits)),
       name,state_code,city,school_type,total_visits,med_inc,pct_white,pct_black,
       pct_hispanic,pct_asian,pct_amerindian,pct_other),n=10)
#> # A tibble: 10 x 12
#>    name  state_code city  school_type total_visits med_inc pct_white
#>    <chr> <chr>      <chr> <chr>              <int>   <dbl>     <dbl>
#>  1 EPIS~ VA         ALEX~ private               26 109558.     77.8
#>  2 NOLA~ TX         FORT~ private               21  39490.     55.8
#>  3 FORT~ TX         FORT~ private               20  89470.      4.09
#>  4 STRA~ TX         HOUS~ private               18  29630.     56.7
#>  5 TRIN~ TX         ADDI~ private               18  77380      83.5
#>  6 JESU~ TX         DALL~ private               16  89203      71.7
#>  7 SANT~ CA         RANC~ private               15 105576.     66.6
#>  8 JSER~ CA         SAN ~ private               14  88324      60.1
#>  9 WOOD~ GA         COLL~ private               14  34561      16.7
#> 10 TRIN~ TX         FORT~ private                14  59778.     72.7
#> # ... with 5 more variables: pct_black <dbl>, pct_hispanic <dbl>,
#> #   pct_asian <dbl>, pct_amerindian <dbl>, pct_other <dbl>
```

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)
   /1.5

```
#part a
df_school_all %>%
  arrange(desc(total_visits)) %>%
  select(name,state_code,city,school_type,total_visits,med_inc,pct_white,pct_black,
       pct_hispanic,pct_asian,pct_amerindian,pct_other) %>%
  filter(school_type == "public") %>%
  head(n = 10)
#> # A tibble: 10 x 12
#>    name  state_code city  school_type total_visits med_inc pct_white
#>    <chr> <chr>      <chr> <chr>                <int>   <dbl>     <dbl>
#>  1 Lyon~ IL         La G~ public                  23  94306.      74.1
#>  2 ALLE~ TX         ALLEN public                  23 100809       57.2
#>  3 COPP~ TX         COPP~ public                  23 123382.      49.9
#>  4 FLOW~ TX         FLOW~ public                  22 157234.      74
#>  5 LOVE~ TX         LUCAS public                  19 100809       81.9
#>  6 HIGH~ TX         DALL~ public                  17 164063       89.2
#>  7 Barr~ IL         Barr~ public                  16 155305       69.1
#>  8 St C~ IL         St C~ public                  16  95389       78.5
#>  9 Milt~ GA         Alph~ public                  15 113362.      67.5
#> 10 Nape~ IL         Nape~ public                  15  92668       65.2
#> # ... with 5 more variables: pct_black <dbl>, pct_hispanic <dbl>,
#> #   pct_asian <dbl>, pct_amerindian <dbl>, pct_other <dbl>

#part b
df_school_all %>%
  arrange(desc(total_visits)) %>%
  select(name,state_code,city,school_type,total_visits,med_inc,pct_white,pct_black,
       pct_hispanic,pct_asian,pct_amerindian,pct_other) %>%
  filter(school_type == "private") %>%
  head(n = 10)
#> # A tibble: 10 x 12
#>    name  state_code city  school_type total_visits med_inc pct_white
#>    <chr> <chr>      <chr> <chr>                <int>   <dbl>     <dbl>
#>  1 EPIS~ VA         ALEX~ private                26 109558.     77.8
#>  2 NOLA~ TX         FORT~ private                21  39490.     55.8
#>  3 FORT~ TX         FORT~ private                20  89470.      4.09
#>  4 STRA~ TX         HOUS~ private                18  29630.     56.7
#>  5 TRIN~ TX         ADDI~ private                18  77380      83.5
#>  6 JESU~ TX         DALL~ private                16  89203      71.7
#>  7 SANT~ CA         RANC~ private                15 105576.     66.6
#>  8 JSER~ CA         SAN ~ private                14  88324      60.1
#>  9 WOOD~ GA         COLL~ private                14  34561      16.7
#> 10 TRIN~ TX         FORT~ private                14  59778.     72.7
#> # ... with 5 more variables: pct_black <dbl>, pct_hispanic <dbl>,
#> #   pct_asian <dbl>, pct_amerindian <dbl>, pct_other <dbl>
```

5. Using pipe operator (%>%), print the following (same variables as above; one line of code for each part

(A), (B), (C), (D)):
/2

- (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

```
#MA, public
df_school_all %>%
  arrange(desc(total_visits)) %>%
  select(name,state_code,city,school_type,total_visits,med_inc,pct_white,pct_black,
       pct_hispanic,pct_asian,pct_amerindian,pct_other) %>%
  filter(school_type == "public", state_code == "MA") %>%
  head(n=10)
#> # A tibble: 10 x 12
#>    name  state_code city  school_type total_visits med_inc pct_white
#>    <chr> <chr>      <chr> <chr>              <int>   <dbl>     <dbl>
#>  1 Broo~ MA         Broo~ public                 8 122258.      59.0
#>  2 Newt~ MA         Newt~ public                 7 176431       65.1
#>  3 Hing~ MA         Hing~ public                 6 168706.      92.6
#>  4 Nort~ MA         Nort~ public                 6 121032.      82.1
#>  5 Algo~ MA         Nort~ public                 6 125844.      84.8
#>  6 Nort~ MA         Quin~ public                 6  80276.      37.8
#>  7 West~ MA         West~ public                 6 121038.      72.1
#>  8 Ando~ MA         Ando~ public                 5 149114       77.4
#>  9 Bost~ MA         Bost~ public                 5  55690.      47.5
#> 10 Coha~ MA         Coha~ public                 5 159476.      92.7
#> # ... with 5 more variables: pct_black <dbl>, pct_hispanic <dbl>,
#> #   pct_asian <dbl>, pct_amerindian <dbl>, pct_other <dbl>

#MA, private
df_school_all %>%
  arrange(desc(total_visits)) %>%
  select(name,state_code,city,school_type,total_visits,med_inc,pct_white,pct_black,
       pct_hispanic,pct_asian,pct_amerindian,pct_other) %>%
  filter(school_type == "private", state_code == "MA") %>%
  head(n=10)
#> # A tibble: 10 x 12
#>    name  state_code city  school_type total_visits med_inc pct_white
#>    <chr> <chr>      <chr> <chr>              <int>   <dbl>     <dbl>
#>  1 NOTR~ MA         HING~ private                8 168706.      92.4
#>  2 BOST~ MA         DORC~ private                8  57334       81.8
#>  3 WORC~ MA         WORC~ private                7  56466.      75.3
#>  4 THAY~ MA         BRAI~ private                6 102247       90.4
#>  5 BISH~ MA         ATTL~ private                4  83076.      91.6
#>  6 PHIL~ MA         ANDO~ private                4 149114       54.1
#>  7 TABO~ MA         MARI~ private                4  98198.      79.7
#>  8 DEXT~ MA         BROO~ private                4 122258.      89.3
#>  9 MILT~ MA         MILT~ private                4 150738       62.0
#> 10 MARI~ MA         FRAM~ private                3  55090.      50.8
#> # ... with 5 more variables: pct_black <dbl>, pct_hispanic <dbl>,
#> #   pct_asian <dbl>, pct_amerindian <dbl>, pct_other <dbl>

#CA, public
```

```
df_school_all %>%
  arrange(desc(total_visits)) %>%
  select(name,state_code,city,school_type,total_visits,med_inc,pct_white,pct_black,
       pct_hispanic,pct_asian,pct_amerindian,pct_other) %>%
  filter(school_type == "public", state_code == "CA") %>%
  head(n=10)
#> # A tibble: 10 x 12
#>    name  state_code city  school_type total_visits med_inc pct_white
#>    <chr> <chr>      <chr> <chr>              <int>   <dbl>     <dbl>
#>  1 Coro~ CA         Newp~ public                12 133966       82.6
#>  2 Trab~ CA         Miss~ public                12 112446.      57.2
#>  3 Mont~ CA         Danv~ public                10 168605       67.9
#>  4 Sant~ CA         Sant~ public                10  93942       41.4
#>  5 Tust~ CA         Tust~ public                10  70780.      13.3
#>  6 Cala~ CA         Cala~ public                 9 123449       78.7
#>  7 Palo~ CA         Palo~ public                 9 211304.      69.5
#>  8 Mira~ CA         Manh~ public                 8 168271       58.8
#>  9 Burr~ CA         Burb~ public                 8  87288       37.2
#> 10 Alis~ CA         Alis~ public                 8 110660.      59.2
#> # ... with 5 more variables: pct_black <dbl>, pct_hispanic <dbl>,
#> #   pct_asian <dbl>, pct_amerindian <dbl>, pct_other <dbl>

#CA, private
df_school_all %>%
  arrange(desc(total_visits)) %>%
  select(name,state_code,city,school_type,total_visits,med_inc,pct_white,pct_black,
       pct_hispanic,pct_asian,pct_amerindian,pct_other) %>%
  filter(school_type == "private", state_code == "CA") %>%
  head(n=10)
#> # A tibble: 10 x 12
#>    name  state_code city  school_type total_visits med_inc pct_white
#>    <chr> <chr>      <chr> <chr>              <int>   <dbl>     <dbl>
#>  1 SANT~ CA         RANC~ private               15 105576.      66.6
#>  2 JSER~ CA         SAN ~ private               14  88324       60.1
#>  3 MATE~ CA         SANT~ private               12  64052.      38.3
#>  4 SERV~ CA         ANAH~ private               11  55142       41.0
#>  5 ST F~ CA         LA C~ private                9 177146.      48.0
#>  6 CHAM~ CA         WEST~ private                8  64568.      49.1
#>  7 NOTR~ CA         SHER~ private                8  91428.      62.6
#>  8 JUNI~ CA         SAN ~ private                8 123328       61.7
#>  9 CATH~ CA         SAN ~ private                8 143160       87.1
#> 10 ST I~ CA         SAN ~ private                6 121018.      60.1
#> # ... with 5 more variables: pct_black <dbl>, pct_hispanic <dbl>,
#> #   pct_asian <dbl>, pct_amerindian <dbl>, pct_other <dbl>
```

5 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`
  /0.25

```
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)

/1

```
str(df_school_all$state_code)
#>  chr [1:21301] "AK" "AK" "AK" "AK" "AK" "AK" "AK" "AK" "AK" "AK" "AK" ...
df_school_all %>% mutate(ca_school = ifelse(state_code=="CA",1,0)) %>%
  count(ca_school)
#> # A tibble: 2 x 2
#>   ca_school     n
#>       <dbl> <int>
#> 1         0 19531
#> 2         1  1770
```

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)

/1

```
str(df_school_all$state_code)
#>  chr [1:21301] "AK" "AK" "AK" "AK" "AK" "AK" "AK" "AK" "AK" "AK" "AK" ...
str(df_school_all$school_type)
#>  chr [1:21301] "public" "public" "public" "public" "public" "public" ...
df_school_all %>%
  mutate(ca_pub_school = ifelse(state_code=="CA" & school_type == "public",1,0)) %>%
  count(ca_pub_school)
#> # A tibble: 2 x 2
#>   ca_pub_school     n
#>           <dbl> <int>
#> 1             0 19897
#> 2             1  1404
```

3. By combining the is.na() function with the filter() function, identify the number of observations that have missing values for the following variables:
   /0.75
   - pct_black, pct_hispanic, pct_amerindian

```
df_school_all %>% filter(is.na(pct_black)) %>% count()
#> # A tibble: 1 x 1
#>       n
#>   <int>
#> 1     0
df_school_all %>% filter(is.na(pct_hispanic)) %>% count()
#> # A tibble: 1 x 1
#>       n
#>   <int>
#> 1     0
df_school_all %>% filter(is.na(pct_amerindian)) %>% count()
#> # A tibble: 1 x 1
#>       n
#>   <int>
#> 1     0
```

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
   /1.5

```
df_school_all <- df_school_all %>% mutate(pct_bl_hisp_nat = pct_black + pct_hispanic + pct_amerindian)
```

5. Create a new 0/1 indicator variable gt50pct_bl_hisp_nat 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 thie variable)
   /1.5

```
df_school_all %>% mutate(gt50pct_bl_hisp_nat = ifelse(pct_bl_hisp_nat>50,1,0)) %>%
  count(gt50pct_bl_hisp_nat)
#> # A tibble: 2 x 2
#>   gt50pct_bl_hisp_nat     n
#>                 <dbl> <int>
#> 1                   0 15701
#> 2                   1  5600
```

6. Create the following 0/1 indicator variables, retain them (assign to object df_school_all), and then create frequency counts of these variables:

/3

- 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`

```
df_school_all <- df_school_all %>%
  mutate(
    miss_took_math = ifelse(is.na(num_took_math),1,0),
    miss_prof_math = ifelse(is.na(num_prof_math),1,0),
    miss_took_or_prof_math = ifelse(is.na(num_took_math) | is.na(num_prof_math),1,0)
  )

df_school_all %>% count(miss_took_math)
#> # A tibble: 2 x 2
#>   miss_took_math     n
#>            <dbl> <int>
#> 1              0 17198
#> 2              1  4103
df_school_all %>% count(miss_prof_math)
#> # A tibble: 2 x 2
#>   miss_prof_math     n
#>            <dbl> <int>
#> 1              0 17050
#> 2              1  4251
df_school_all %>% count(miss_took_or_prof_math)
#> # A tibble: 2 x 2
#>   miss_took_or_prof_math     n
#>                    <dbl> <int>
#> 1                      0 17050
#> 2                      1  4251
```

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`).
   /2

```
df_school_all <- df_school_all %>%
  mutate(pct_prof_math=num_prof_math/num_took_math)
```

8. create a frequency count of value of the variable `pct_prof_math` separately for the three following filters:
   /1
   - Observations where `miss_took_math==1`
   - Observations where `miss_prof_math==1`
   - Observations where `miss_took_or_prof_math==1`

```
df_school_all %>% filter(miss_took_math==1) %>% count(pct_prof_math)
#> # A tibble: 1 x 2
#>   pct_prof_math     n
#>           <dbl> <int>
#> 1            NA  4103
df_school_all %>% filter(miss_prof_math==1) %>% count(pct_prof_math)
#> # A tibble: 1 x 2
#>   pct_prof_math     n
#>           <dbl> <int>
#> 1            NA  4251
```

```
df_school_all %>% filter(miss_took_or_prof_math==1) %>% count(pct_prof_math)
#> # A tibble: 1 x 2
#>   pct_prof_math     n
#>           <dbl> <int>
#> 1            NA  4251
```

6 THIS IS A BONUS QUESTION

7 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`
   /0.5
   - `state`
   - `for_country`

```
str(wwlist$state)
#>  chr [1:268396] "WA" "WA" "WA" "WA" "WA" "WA" "WA" "WA" "WA" "ID" "ID" ...
wwlist %>% count(state)
#> # A tibble: 54 x 2
#>    state     n
#>    <chr> <int>
#>  1 <NA>     85
#>  2 AK     3671
#>  3 AL      136
#>  4 AP        1
#>  5 AR       78
#>  6 AZ    10358
#>  7 CA    62382
#>  8 CO    24831
#>  9 CT      173
#> 10 DC       35
#> # ... with 44 more rows

str(wwlist$for_country)
#>  chr [1:268396] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA ...
wwlist %>% count(for_country)
#> # A tibble: 30 x 2
#>    for_country          n
#>    <chr>            <int>
#>  1 <NA>            268311
#>  2 Afghanistan          6
#>  3 Australia             2
#>  4 Bahamas               1
#>  5 Brazil                2
#>  6 Canada                1
#>  7 Chad                  1
#>  8 China                11
#>  9 Christmas Island      2
#> 10 Cote D'Ivoire         1
#> # ... with 20 more rows
```

3. Using the `filter()` function and `is.na()` function do the following:
   /0.5
   - count how many missing observations (`NA`s) the variable `state` has
   - count how many missing observations the variable `for_country` has

```
wwlist %>% filter(is.na(state)) %>% count()
#> # A tibble: 1 x 1
#>       n
#>   <int>
#> 1    85
wwlist %>% filter(is.na(for_country)) %>% count()
#> # A tibble: 1 x 1
#>        n
#>    <int>
#> 1 268311
```

4. Create a frequency count for the variable `for_country` for the observations where `state` equals `NA` (hint: use the `is.na()`) function
   /0.5

```
wwlist %>% filter(is.na(state)) %>% count(for_country)
#> # A tibble: 29 x 2
#>    for_country          n
#>    <chr>            <int>
#>  1 Afghanistan          6
#>  2 Australia            2
#>  3 Bahamas              1
#>  4 Brazil               2
#>  5 Canada               1
#>  6 Chad                 1
#>  7 China               11
#>  8 Christmas Island     2
#>  9 Cote D'Ivoire        1
#> 10 Czech Republic       1
#> # ... with 19 more rows
```

5. Create a frequency count for the variable `for_country` for the observations where `state` does not equal `NA` (hint: use `!is.na()`) function
   /0.5

```
wwlist %>% filter(!is.na(state)) %>% count(for_country)
#> # A tibble: 1 x 2
#>   for_country      n
#>   <chr>        <int>
#> 1 <NA>        268311
```

6. Count the number of observations that have the value "No Response" for the variable `for_country`

/0.5

```
wwlist %>% filter(for_country == "No Response") %>% count()
#> # A tibble: 1 x 1
#>       n
#>   <int>
#> 1    17
```

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”
   /1.5

- 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

```
wwlist <- wwlist %>%
  mutate(residency=
    case_when(
      state == "WA" ~ "in_state",
      state != "WA" & (!is.na(state)) ~ "out_state_us",
      (is.na(state)) & for_country != "No Response" ~ "not_in_us"
    )
  )

wwlist %>% count(residency)
#> # A tibble: 4 x 2
#>   residency         n
#>   <chr>         <int>
#> 1 <NA>             17
#> 2 in_state      96022
#> 3 not_in_us        68
#> 4 out_state_us 172289
```

Once finished, knit to (pdf) and upload both .Rmd and HTML files to class website under the week 3 tab
Remember to use this naming convention “lastname_firstname_ps3”