Module 3 problem set

INSERT YOUR NAME HERE

INSERT DATE HERE

Extracting and Sorting Data via Tidyverse and base R

The aim of this problem set is to demonstrate there are many different ways to complete the same data management tasks.

Last week you learned to extract variables and observations as well as sort observations the `tidyverse` way via the `select`, `filter`, and `arrange` functions. Module 3 demonstrated how some of the tasks done with `tidyverse` functions have a corresponding solution using `base R` syntax.

For the following questions, you'll be asked to complete the same task multiple ways based on the `tidyverse` and `base R` approaches.

Step 1: Remove objects in current R session, load `tidyverse`, and open the data

1. Begin by removing any objects in your current R session by using `rm(list = ls())`. Then load the `tidyverse` library. Lastly, use the `load` function to open the `df_event` dataset via url link

- The url for the `df_event` dataset is https://github.com/ksalazar3/HED696C_RClass/raw/master/data/recruiting/recruit_event_somevars.RData
- The data frame `df_event` has one observation for each recruiting event.

```
rm(list = ls()) # remove all objects

library(tidyverse)
#> -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
#> v ggplot2 3.3.6     v purrr   0.3.4
#> v tibble  3.1.8     v dplyr   1.0.9
#> v tidyr   1.2.0     v stringr 1.4.0
#> v readr   2.1.2     v forcats 0.5.1
#> -- Conflicts ------------------------------------------ tidyverse_conflicts() --
#> x dplyr::filter() masks stats::filter()
#> x dplyr::lag()    masks stats::lag()
load(url("https://github.com/ksalazar3/HED696C_RClass/raw/master/data/recruiting/recruit_event_somevars
```

Step 2: Extract columns, extract observations, sort observations

Complete all the following questions in three different ways: (1) by using the tidyverse `select`, `filter`, or `arrange` functions, (2) by using base R's subsetting operators, and/or (3) by using base R's `subset` or `order` functions.

I have included rchunks below to indicate how many different ways you should be attempting the tasks.

2. Create a new dataframe by extracting the columns `univ_id`, `event_date`, `event_type`, `zip`, and `med_inc` from `df_event`. Use the names() function to show what columns (variables) are in the newly created dataframe. Print the first 10 observations of the newly created dataframe.

tidyverse

```
df2_tv <- select(df_event, univ_id, event_date, event_type, zip, med_inc)
names(df2_tv)
#> [1] "univ_id"    "event_date" "event_type" "zip"        "med_inc"
```

base R using subsetting operators

```
df2_b1 <- df_event[, c("univ_id", "event_date", "event_type", "zip", "med_inc"), drop = FALSE] #good ha
names(df2_b1)
#> [1] "univ_id"    "event_date" "event_type" "zip"        "med_inc"
```

base R using subset()

```
df2_b2 <- subset(df_event, select=c(univ_id, event_date, event_type, zip, med_inc), drop = FALSE) #good
names(df2_b2)
#> [1] "univ_id"    "event_date" "event_type" "zip"        "med_inc"
```

3. Create a new dataframe from `df_event` that includes recruiting events by the University of Massachusetts Amherst (`univ_id`==166629), that were located at in-state public high schools (`event_type` and `event_state`) where the average median household income (`med_inc`) is equal to or greater than $100,000. Use `nrow` to make sure you are extracting the same number of observations across each approach below.

tidyverse

```
df3_tv <- filter(df_event, univ_id == 166629 & event_state == "MA" & event_type == "public hs" & med_in
nrow(df3_tv)
#> [1] 85
```

base R using subsetting operators

```
df3_b1 <- df_event[df_event$univ_id == 166629 & df_event$event_state == "MA" & df_event$event_type == "
nrow(df3_b1) #has 2 extra obs
#> [1] 87
head(df3_b1, n=10) #includes NA obs!
#> # A tibble: 10 x 33
#>    instnm  univ_id instst   pid event_date event~1 zip   schoo~2 ipeds~3 event~4
#>    <chr>     <int> <chr>  <int> <date>     <chr>   <chr> <chr>     <int> <chr>
#>  1 UM Amh~  166629 MA     57091 2017-10-23 public~ 01095 250573~      NA MA
#>  2 UM Amh~  166629 MA     56902 2017-09-19 public~ 01106 250699~      NA MA
#>  3 UM Amh~  166629 MA     57088 2017-10-23 public~ 01106 250699~      NA MA
#>  4 <NA>         NA <NA>      NA NA         <NA>    <NA>  <NA>         NA <NA>
#>  5 UM Amh~  166629 MA     56993 2017-10-05 public~ 01430 250204~      NA MA
#>  6 <NA>         NA <NA>      NA NA         <NA>    <NA>  <NA>         NA <NA>
#>  7 UM Amh~  166629 MA     56929 2017-09-25 public~ 01450 250550~      NA MA
#>  8 UM Amh~  166629 MA     57042 2017-10-13 public~ 01451 250588~      NA MA
#>  9 UM Amh~  166629 MA     57125 2017-10-27 public~ 01460 250696~      NA MA
#> 10 UM Amh~  166629 MA     57069 2017-10-18 public~ 01462 250708~      NA MA
#> # ... with 23 more variables: event_inst <chr>, med_inc <dbl>, pop_total <dbl>,
#> #   pct_white_zip <dbl>, pct_black_zip <dbl>, pct_asian_zip <dbl>,
#> #   pct_hispanic_zip <dbl>, pct_amerindian_zip <dbl>,
#> #   pct_nativehawaii_zip <dbl>, pct_tworaces_zip <dbl>,
```

```
#> #   pct_otherrace_zip <dbl>, fr_lunch <dbl>, titlei_status_pub <fct>,
#> #   total_12 <dbl>, school_type_pri <int>, school_type_pub <int>,
#> #   g12offered <dbl>, g12 <dbl>, total_students_pub <dbl>, ...
#> # i Use `colnames()` to see all variable names

#use the which() function to remove those NA obs
df3_b1.2 <- df_event[which(df_event$univ_id == 166629 & df_event$event_state == "MA" & df_event$event_ty
nrow(df3_b1.2) #now has the same number of obs
#> [1] 85
head(df3_b1.2, n=10) #no NA obs!
#> # A tibble: 10 x 33
#>    instnm  univ_id instst   pid event_date event~1 zip   schoo~2 ipeds~3 event~4
#>    <chr>     <int> <chr>  <int> <date>     <chr>   <chr> <chr>     <int> <chr>
#>  1 UM Amh~  166629 MA     57091 2017-10-23 public~ 01095 250573~      NA MA
#>  2 UM Amh~  166629 MA     56902 2017-09-19 public~ 01106 250699~      NA MA
#>  3 UM Amh~  166629 MA     57088 2017-10-23 public~ 01106 250699~      NA MA
#>  4 UM Amh~  166629 MA     56993 2017-10-05 public~ 01430 250204~      NA MA
#>  5 UM Amh~  166629 MA     56929 2017-09-25 public~ 01450 250550~      NA MA
#>  6 UM Amh~  166629 MA     57042 2017-10-13 public~ 01451 250588~      NA MA
#>  7 UM Amh~  166629 MA     57125 2017-10-27 public~ 01460 250696~      NA MA
#>  8 UM Amh~  166629 MA     57069 2017-10-18 public~ 01462 250708~      NA MA
#>  9 UM Amh~  166629 MA     56978 2017-10-04 public~ 01505 250258~      NA MA
#> 10 UM Amh~  166629 MA     57104 2017-10-25 public~ 01519 250537~      NA MA
#> # ... with 23 more variables: event_inst <chr>, med_inc <dbl>, pop_total <dbl>,
#> #   pct_white_zip <dbl>, pct_black_zip <dbl>, pct_asian_zip <dbl>,
#> #   pct_hispanic_zip <dbl>, pct_amerindian_zip <dbl>,
#> #   pct_nativehawaii_zip <dbl>, pct_tworaces_zip <dbl>,
#> #   pct_otherrace_zip <dbl>, fr_lunch <dbl>, titlei_status_pub <fct>,
#> #   total_12 <dbl>, school_type_pri <int>, school_type_pub <int>,
#> #   g12offered <dbl>, g12 <dbl>, total_students_pub <dbl>, ...
#> # i Use `colnames()` to see all variable names
```

base R using subset()

```
df3_b2 <- subset(df_event, univ_id == 166629 & event_state == "MA" & event_type == "public hs" & med_in
nrow(df3_b2)
#> [1] 85
```

4. Create a new dataframe from `df_event` that includes recruiting events by the University of South Carolina Columbia (`univ_id`==218663), that were located at out-of-state public high schools (`event_type` and `event_state`) where the average median household income (`med_inc`) is equal to or greater than $100,000 and the White population in the surrounding area is equal to or greater than 50% of the total population (`pct_white_zip`). Use `nrow` to make sure you are extracting the same number of observations across each approach below.

tidyverse

```
df4_tv <- filter(df_event, univ_id == 218663 & event_state != "SC" & event_type == "public hs" & med_in
nrow(df4_tv)
#> [1] 336
```

base R using subsetting operators

```
df4_b1 <- df_event[df_event$univ_id == 218663 & df_event$event_state != "SC" & df_event$event_type == "
                   & df_event$med_inc >= 100000 & df_event$pct_white_zip>=50 ,  , drop=FALSE]
nrow(df4_b1) #has 1 extra obs
```

```
#> [1] 337
```

```
df4_b1.2 <- df_event[which(df_event$univ_id == 218663 & df_event$event_state != "SC" & df_event$event_ty
                  & df_event$med_inc >= 100000 & df_event$pct_white_zip>=50) ,  , drop=FALSE]
nrow(df4_b1.2) #now has the same number of obs
#> [1] 336
```

base R using `subset()`

```
df4_b2 <- subset(df_event, univ_id == 218663 & event_state != "SC" & event_type == "public hs" & med_inc
nrow(df4_b2)
#> [1] 336
```

5. Create a new dataframe from df_events that sorts by ascending `univ_id`, ascending by `event_date` , ascending `event_state`, descending `pct_white_zip`, descending `med_inc`.

tidyverse

```
df5_tv <- arrange(df_event, univ_id, event_date, event_state, desc(pct_white_zip), desc(med_inc))
head(df5_tv, n=10)
#> # A tibble: 10 x 33
#>    instnm univ_id instst   pid event_date event_~1 zip   schoo~2 ipeds~3 event~4
#>    <chr>    <int> <chr>  <int> <date>     <chr>    <chr> <chr>     <int> <chr>
#>  1 Bama    100751 AL      2667 2017-01-10 private~ 75001 X13284~      NA TX
#>  2 Bama    100751 AL      2674 2017-01-11 2yr col~ 35010 <NA>     100760 AL
#>  3 Bama    100751 AL      2675 2017-01-11 other    35044 <NA>         NA AL
#>  4 Bama    100751 AL      2691 2017-01-12 private~ 75244 A03031~      NA TX
#>  5 Bama    100751 AL      2676 2017-01-17 2yr col~ 36350 <NA>     101286 AL
#>  6 Bama    100751 AL      2851 2017-01-17 public ~ 21769 240033~      NA MD
#>  7 Bama    100751 AL      2733 2017-01-17 public ~ 75002 480789~      NA TX
#>  8 Bama    100751 AL      2677 2017-01-18 2yr col~ 36330 <NA>     101143 AL
#>  9 Bama    100751 AL      2645 2017-01-18 public ~ 30277 130150~      NA GA
#> 10 Bama    100751 AL      2736 2017-01-18 public ~ 30281 130282~      NA GA
#> # ... with 23 more variables: event_inst <chr>, med_inc <dbl>, pop_total <dbl>,
#> #   pct_white_zip <dbl>, pct_black_zip <dbl>, pct_asian_zip <dbl>,
#> #   pct_hispanic_zip <dbl>, pct_amerindian_zip <dbl>,
#> #   pct_nativehawaii_zip <dbl>, pct_tworaces_zip <dbl>,
#> #   pct_otherrace_zip <dbl>, fr_lunch <dbl>, titlei_status_pub <fct>,
#> #   total_12 <dbl>, school_type_pri <int>, school_type_pub <int>,
#> #   g12offered <dbl>, g12 <dbl>, total_students_pub <dbl>, ...
#> # i Use `colnames()` to see all variable names
```

base R using `order()`

```
df5_b1 <- df_event[order(df_event$univ_id, df_event$event_date, df_event$event_state, -df_event$pct_whi
head(df5_b1, n=10)
#> # A tibble: 10 x 33
#>    instnm univ_id instst   pid event_date event_~1 zip   schoo~2 ipeds~3 event~4
#>    <chr>    <int> <chr>  <int> <date>     <chr>    <chr> <chr>     <int> <chr>
#>  1 Bama    100751 AL      2667 2017-01-10 private~ 75001 X13284~      NA TX
#>  2 Bama    100751 AL      2674 2017-01-11 2yr col~ 35010 <NA>     100760 AL
#>  3 Bama    100751 AL      2675 2017-01-11 other    35044 <NA>         NA AL
#>  4 Bama    100751 AL      2691 2017-01-12 private~ 75244 A03031~      NA TX
#>  5 Bama    100751 AL      2676 2017-01-17 2yr col~ 36350 <NA>     101286 AL
```

```
#>  6 Bama    100751 AL      2851 2017-01-17 public ~ 21769 240033~      NA MD
#>  7 Bama    100751 AL      2733 2017-01-17 public ~ 75002 480789~      NA TX
#>  8 Bama    100751 AL      2677 2017-01-18 2yr col~ 36330 <NA>     101143 AL
#>  9 Bama    100751 AL      2645 2017-01-18 public ~ 30277 130150~      NA GA
#> 10 Bama    100751 AL      2736 2017-01-18 public ~ 30281 130282~      NA GA
#> # ... with 23 more variables: event_inst <chr>, med_inc <dbl>, pop_total <dbl>,
#> #   pct_white_zip <dbl>, pct_black_zip <dbl>, pct_asian_zip <dbl>,
#> #   pct_hispanic_zip <dbl>, pct_amerindian_zip <dbl>,
#> #   pct_nativehawaii_zip <dbl>, pct_tworaces_zip <dbl>,
#> #   pct_otherrace_zip <dbl>, fr_lunch <dbl>, titlei_status_pub <fct>,
#> #   total_12 <dbl>, school_type_pri <int>, school_type_pub <int>,
#> #   g12offered <dbl>, g12 <dbl>, total_students_pub <dbl>, ...
#> # i Use `colnames()` to see all variable names
```