

Lecture 3 problem set

INSERT YOUR NAME HERE

10/11/2019

/20

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. Lecture 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/ozanj/rclass/raw/master/data/recruiting/recruit_event_somevars.RData
 - The data frame `df_event` has one observation for each recruiting event.

/2

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

library(tidyverse)
#> -- Attaching packages -----
#> v ggplot2 3.2.1      v purrr  0.3.2
#> v tibble  2.1.3      v dplyr  0.8.3
#> v tidyr   1.0.0      v stringr 1.4.0
#> v readr   1.3.1      v forcats 0.4.0
#> -- Conflicts ----- tidyver.
#> x dplyr::filter() masks stats::filter()
#> x dplyr::lag()    masks stats::lag()
load(url("https://github.com/ozanj/rclass/raw/master/data/recruiting/recruit_event_somevars.RData"))
```

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.

/5

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.

/5

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_type zip school_id
#>   <chr>   <int> <chr> <int> <date>   <chr>   <chr> <chr>
#> 1 UM Am~ 166629 MA 57091 2017-10-23 public hs 01095 25057300~
#> 2 UM Am~ 166629 MA 56902 2017-09-19 public hs 01106 25069900~
#> 3 UM Am~ 166629 MA 57088 2017-10-23 public hs 01106 25069900~
#> 4 <NA>    NA <NA>    NA NA      <NA>   <NA> <NA>
#> 5 UM Am~ 166629 MA 56993 2017-10-05 public hs 01430 25020400~
#> 6 <NA>    NA <NA>    NA NA      <NA>   <NA> <NA>
#> 7 UM Am~ 166629 MA 56929 2017-09-25 public hs 01450 25055000~
#> 8 UM Am~ 166629 MA 57042 2017-10-13 public hs 01451 25058800~
#> 9 UM Am~ 166629 MA 57125 2017-10-27 public hs 01460 25069600~
#> 10 UM Am~ 166629 MA 57069 2017-10-18 public hs 01462 25070800~
#> # ... with 25 more variables: ipeds_id <int>, event_state <chr>,
#> # 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>, total_students_pri <dbl>, event_name <chr>,
#> # event_location_name <chr>, event_datetime_start <dtm>

#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_t,
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_type zip school_id
#>   <chr> <int> <chr> <int> <date> <chr> <chr> <chr>
#> 1 UM Am~ 166629 MA 57091 2017-10-23 public hs 01095 25057300~
#> 2 UM Am~ 166629 MA 56902 2017-09-19 public hs 01106 25069900~
#> 3 UM Am~ 166629 MA 57088 2017-10-23 public hs 01106 25069900~
#> 4 UM Am~ 166629 MA 56993 2017-10-05 public hs 01430 25020400~
#> 5 UM Am~ 166629 MA 56929 2017-09-25 public hs 01450 25055000~
#> 6 UM Am~ 166629 MA 57042 2017-10-13 public hs 01451 25058800~
#> 7 UM Am~ 166629 MA 57125 2017-10-27 public hs 01460 25069600~
#> 8 UM Am~ 166629 MA 57069 2017-10-18 public hs 01462 25070800~
#> 9 UM Am~ 166629 MA 56978 2017-10-04 public hs 01505 25025800~
#> 10 UM Am~ 166629 MA 57104 2017-10-25 public hs 01519 25053700~
#> # ... with 25 more variables: ipeds_id <int>, event_state <chr>,
#> # 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>, total_students_pri <dbl>, event_name <chr>,
#> # event_location_name <chr>, event_datetime_start <dtm>

```

base R using subset()

```

df3_b2 <- subset(df_event, univ_id == 166629 & event_state == "MA" & event_type == "public hs" & med_inc
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.

/5

tidyverse

```

df4_tv <- filter(df_event, univ_id == 218663 & event_state != "SC" & event_type == "public hs" & med_inc
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$event_type == "public hs" & df_event$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$event_type == "public hs" & df_event$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 >= 100000 & pct_white_zip >= 50)
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.

/3

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_type zip school_id
#>   <chr>   <int> <chr> <int> <date>      <chr>    <chr> <chr>
#> 1 Bama    100751 AL      2667 2017-01-10 private hs 75001 X1328481
#> 2 Bama    100751 AL      2674 2017-01-11 2yr colle~ 35010 <NA>
#> 3 Bama    100751 AL      2675 2017-01-11 other      35044 <NA>
#> 4 Bama    100751 AL      2691 2017-01-12 private hs 75244 A0303150
#> 5 Bama    100751 AL      2676 2017-01-17 2yr colle~ 36350 <NA>
#> 6 Bama    100751 AL      2851 2017-01-17 public hs 21769 24003300~
#> 7 Bama    100751 AL      2733 2017-01-17 public hs 75002 48078900~
#> 8 Bama    100751 AL      2677 2017-01-18 2yr colle~ 36330 <NA>
#> 9 Bama    100751 AL      2645 2017-01-18 public hs 30277 13015000~
#> 10 Bama   100751 AL      2736 2017-01-18 public hs 30281 13028200~
#> # ... with 25 more variables: ipeds_id <int>, event_state <chr>,
#> #   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>, total_students_pri <dbl>, event_name <chr>,
#> #   event_location_name <chr>, event_datetime_start <dtm>
```

base R using order()

```
df5_b1 <- df_event[order(df_event$univ_id, df_event$event_date, df_event$event_state, -df_event$pct_white_zip), ]
head(df5_b1, n=10)
#> # A tibble: 10 x 33
#>   instnm univ_id instst   pid event_date event_type zip school_id
#>   <chr>   <int> <chr> <int> <date>      <chr>    <chr> <chr>
```

```

#> 1 Bama 100751 AL 2667 2017-01-10 private hs 75001 X1328481
#> 2 Bama 100751 AL 2674 2017-01-11 2yr colle~ 35010 <NA>
#> 3 Bama 100751 AL 2675 2017-01-11 other 35044 <NA>
#> 4 Bama 100751 AL 2691 2017-01-12 private hs 75244 A0303150
#> 5 Bama 100751 AL 2676 2017-01-17 2yr colle~ 36350 <NA>
#> 6 Bama 100751 AL 2851 2017-01-17 public hs 21769 24003300~
#> 7 Bama 100751 AL 2733 2017-01-17 public hs 75002 48078900~
#> 8 Bama 100751 AL 2677 2017-01-18 2yr colle~ 36330 <NA>
#> 9 Bama 100751 AL 2645 2017-01-18 public hs 30277 13015000~
#> 10 Bama 100751 AL 2736 2017-01-18 public hs 30281 13028200~
#> # ... with 25 more variables: ipeds_id <int>, event_state <chr>,
#> # 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>, total_students_pri <dbl>, event_name <chr>,
#> # event_location_name <chr>, event_datetime_start <dtm>

```