Problem Set #2

INSERT YOUR NAME HERE

insert date here

Overview

In this problem set, you will practice investigating and subsetting objects using the Base R approach. We are asking you to practice working with different object types such as atomic vectors, lists, and data frames. Learning how to work with different object types will be very useful in the coming weeks and will help you understand the underlying structure of any data you are working with. While this problem set is fairly short, we want you to become familiar with these concepts – which are fundamental to data management and working with R.

Question 1: Practice making changes to the YAML header

We recommend reading R Markdown: The Definitive Guide section 3.3, then make the following changes to the YAML header of this Rmd:

1. Add your name to the YAML header.
2. Add a table of contents to YAML header.
3. The table of contents should have "depth" of 2.
4. Add section numbering to headers.

Question 2: Investigating objects, Base R

Run the following to load data frame objects:

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

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

dim(df_school_all)
#> [1] 21301    55
```

1. Answer the following questions about the object `df_school_all` by running the appropriate R command in the code chunks below and write your response below the question following the **ANSWER:** prompt. The first question will be answered for you to show how it works.
    - What "type" of object is `df_school_all`?
        - **ANSWER [GIVEN]: The object `df_school_all` has type equals list.**
    - What is the "length" of the object `df_school_all`? What does this specific value of length refer to?

  - **ANSWER:**

- How many "rows" are in the object `df_school_all`? What does each row represent?
  - **ANSWER:**

2. In the below code chunk, use the `str()` function to describe the contents of `df_school_all` and then answer the following questions:

   - What does each element of the object `df_school_all` represent? (Hint: Lecture Investigating data patterns in Base R, Slide 17)
     - **ANSWER:**

   - Are the individual elements within `df_school_all` lists or vectors?
     - **ANSWER:**

   - Are the individual elements within `df_school_all` named or un-named? If named, what do these element names refer to?
     - **ANSWER:**

3. These questions refer to the variable `school_type` within the object `df_school_all`. For the first two questions, run the appropriate R command in the code chunk below and write your response below the question.

   - What is the data "type" of `school_type`?
     - **ANSWER:**

   - What is the "length" of `school_type`? What does this specific value of length refer to?
     - **ANSWER:**

4. In these questions, you will apply the `table()` function to the variable `school_type` within the object `df_school_all`.

   - In your own words, what does the `table()` function do?
     - **ANSWER:**

   - What does the `useNA` argument of the `table()` function control?
     - **ANSWER:**

   - What is the default value of the `useNA` argument and what does this default value mean?
     - **ANSWER:**

   - What happens when you assign the value `"ifany"` to the `useNA` argument?
     - **ANSWER:**

   - What happens when you assign the value `"always"` to the `useNA` argument?
     - **ANSWER:**

   - In the below R code chunk, use the `table()` function to count the number of observations for each value `school_type` three different ways:
     - First, without specifying any value for `useNA`

     - Second, by assigning the value `"ifany"` to the `useNA` argument

     - Third, by assigning the value `"always"` to the `useNA` argument

Question 3: Subsetting, Base R

In the code chunk below, you will find 3 objects: a vector, a list, and the data frame `df_school_all`. Run the code chunk. You will use these objects in the following questions.

```
# Create a named numeric atomic vector
vec <- c(a = 2.4, b = 1.1, c = 3.4, d = 4, e = 6, f = 32, g = 21, h = 17, i = 10)
str(vec)
#>  Named num [1:9] 2.4 1.1 3.4 4 6 32 21 17 10
#>  - attr(*, "names")= chr [1:9] "a" "b" "c" "d" ...

# Create a list
list <- list(c(1:3), list("red", "orange"), list("LA", "NY", "DC"))
str(list)
#> List of 3
#>  $ : int [1:3] 1 2 3
#>  $ :List of 2
#>   ..$ : chr "red"
#>   ..$ : chr "orange"
#>  $ :List of 3
#>   ..$ : chr "LA"
#>   ..$ : chr "NY"
#>   ..$ : chr "DC"

# View the `df_school_all` data frame you loaded earlier
head(df_school_all, n = 5)
#>   state_code school_type      ncessch                          name
#> 1         AK      public 020000100208 Bethel Regional High School
#> 2         AK      public 020000100211      Ayagina'ar Elitnaurvik
#> 3         AK      public 020000100212         Kwigillingok School
#> 4         AK      public 020000100213   Nelson Island Area School
#> 5         AK      public 020000300216              Alakanuk School
#>                         address         city zip_code pct_white pct_black
#> 1 1006 Ron Edwards Memorial Dr       Bethel    99559   11.7764    0.5988
#> 2             106 Village Road   Kongiganak    99559    0.0000    0.0000
#> 3             108 Village Road Kwigillingok    99622    0.0000    0.0000
#> 4             118 Village Road  Toksook Bay    99637    0.0000    0.0000
#> 5                9 School Road     Alakanuk    99554    2.5210    0.0000
#>   pct_hispanic pct_asian pct_amerindian pct_other num_fr_lunch total_students
#> 1       1.5968     0.998        84.6307    0.3992          362            501
#> 2       0.0000     0.000        99.4505    0.5495          182            182
#> 3       0.0000     0.000       100.0000    0.0000          116            120
#> 4       0.0000     0.000       100.0000    0.0000          187            201
#> 5       0.0000     0.000        97.4790    0.0000          238            238
#>   num_took_math num_prof_math num_took_rla num_prof_rla avgmedian_inc_2564
#> 1           146         24.82          147        24.99            76160.0
#> 2            17          1.70           17         1.70            76160.0
#> 3            14          3.50           14         3.50                 NA
#> 4            30          3.00           30         3.00            57656.5
#> 5            28          2.80           28         2.80            37552.5
#>   latitude longitude visits_by_196097 visits_by_186380 visits_by_215293
#> 1 60.80258 -161.7704                0                0               0
#> 2 59.95389 -162.8953                0                0               0
#> 3 59.87676 -163.1616                0                0               0
#> 4 60.53270 -165.1091                0                0               0
```

```
#> 5 62.68317 -164.6523                0                0                0
#>   visits_by_201885 visits_by_181464 visits_by_139959 visits_by_218663
#> 1                0                0                0                0
#> 2                0                0                0                0
#> 3                0                0                0                0
#> 4                0                0                0                0
#> 5                0                0                0                0
#>   visits_by_100751 visits_by_199193 visits_by_110635 visits_by_110653
#> 1                0                0                0                0
#> 2                0                0                0                0
#> 3                0                0                0                0
#> 4                0                0                0                0
#> 5                0                0                0                0
#>   visits_by_126614 visits_by_155317 visits_by_106397 visits_by_149222
#> 1                0                0                0                0
#> 2                0                0                0                0
#> 3                0                0                0                0
#> 4                0                0                0                0
#> 5                0                0                0                0
#>   visits_by_166629 total_visits inst_196097 inst_186380 inst_215293 inst_201885
#> 1                0            0          NY          NJ          PA          OH
#> 2                0            0          NY          NJ          PA          OH
#> 3                0            0          NY          NJ          PA          OH
#> 4                0            0          NY          NJ          PA          OH
#> 5                0            0          NY          NJ          PA          OH
#>   inst_181464 inst_139959 inst_218663 inst_100751 inst_199193 inst_110635
#> 1          NE          GA          SC          AL          NC          CA
#> 2          NE          GA          SC          AL          NC          CA
#> 3          NE          GA          SC          AL          NC          CA
#> 4          NE          GA          SC          AL          NC          CA
#> 5          NE          GA          SC          AL          NC          CA
#>   inst_110653 inst_126614 inst_155317 inst_106397 inst_149222 inst_166629
#> 1          CA          CO          KS          AR          IL          MA
#> 2          CA          CO          KS          AR          IL          MA
#> 3          CA          CO          KS          AR          IL          MA
#> 4          CA          CO          KS          AR          IL          MA
#> 5          CA          CO          KS          AR          IL          MA
#str(df_school_all)
```

1. In this question we will use the `[]` to subset the atomic vector `vec`:
   - Return the 4th and 7th element of the vector `vec` (Hint: We are subsetting elements by position)
   - Return everything but the last element of the vector `vec`
   - Return elements named "a", "d", and "g"
   - Return elements that are less than `12`
2. In this question we will use the `[]` to subset columns in `df_school_all`:
   - Return the first 100 elements of the `state_code` column
   - Return all addresses in the `address` column that are more than 30 characters long (Hint: Use `nchar()` to get number of characters, you can type ?nchar in the console or code chunk to get

more information on the function)

- Return all cities in the `city` column that are either 21 or 25 characters in length

3. In this question we will use the `[]` to subset a list/data frame:
   - Return the 1st element of the list `list` using `[]`
     - What is the data type? **ANSWER:**
   - Return the 2nd, 4th, and 6th elements of the data frame `df_school_all`
   - Return the first 3 rows (observations) and the variable names `state_code` and `name` of the data frame `df_school_all`
4. In this question we will use `[[]]` and `$` to subset a list/data frame:
   - Return the 1st element of the list `list` using `[[]]`
     - What is the data type? **ANSWER:**
   - Return the variable `total_students` using `$`. The output here will be too large when you try to knit to PDF at the end of the problem set so just makem this line of code a comment by putting a hashtag # in front of it.
     - What is the data type? **ANSWER:**

Question 4: Create a GitHub issue

- Go to the class repository and create a new issue.
- Refer to rclass1 student issues readme for instructions on how to post questions or reflections.
- You are also required to respond to at least one issue posted by another student.
- Paste the url to your issue here:
- Paste the url to the issue you responded to here:

Knit to pdf and submit problem set

Knit to pdf by clicking the "Knit" button near the top of your RStudio window (icon with blue yarn ball) or drop down and select "Knit to PDF"

- Go to the class website and under the "Readings & Assignments" » "Week 2" tab, click on the "Problem set 2 submission link"
- Submit both .Rmd and pdf files
- Use this naming convention "lastname_firstname_ps#" for your .Rmd and pdf files (e.g. jaquette_ozan_ps2.Rmd & jaquette_ozan_ps2.pdf)