Data Science Principles and Practice
CS x415.1, Fall 2017
Assignment #2

Due: **Tuesday, March 7, 2017 11:59 PM** (no late submissions accepted)
Submit on Canvas / onlinelearning.berkeley.edu (no email submissions accepted)

Problem 1

Mixpanel (www.mixpanel.com) provides a service that tracks and records user-initiated events (e.g., page clicks) and profile information submitted by users on Mixpanel-enabled web sites. JSON files with event and user information (people) can be exported from Mixpanel system for analysis

Use the exported Mixpanel web analytics data files: berkeley_event-export.json and berkeley_people-export.json (archived in **mixpanel_data.zip** posted in the Files section under Assignment 2) to accomplish the following

- Read in both json files and tidy them (i.e., convert the jason to two tidy data frames, events and people, respectively)
- Join them together (by distinct_id) into a a third tidy data frame, **events_people**
- Clean up events_people by transforming the column names of events_people to snake case (here is an R function that converts a string to snake case):

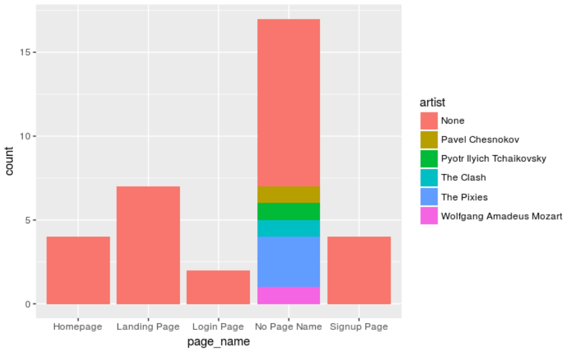
```
snake_case <- function( x ) {
  s <- gsub("\\.", "_", x)      # Replace dots with underscores.
  s <- gsub("(.)([A-Z][a-z]+)", "\\1_\\2", s)  # Separate w/ underscores on
capitalization
  s <- tolower(gsub("([a-z0-9])([A-Z])", "\\1_\\2", s)) # lowercase
  s <- gsub("__", "_", s) # double to single underscore
  s <- gsub("^[_, .]", "", s)  # del first char underscore "_" or period "."
  s <- gsub(' ', '', s) # remove spaces
}
```

- To do this write a function **fix_column_names** usable as a dplyr verb with a pipe to convert the column names of **events_people** to snake case:

```
# convert data frame result column names to snake case

events_people %>%
  fix_column_names
```

- Replace NA's for **artist** with "None"
- Replace NA's for **page_name** with "No Page Name"
- Use events_people and ggplot to construct a plot similar to this, showing the count of page view events by page_name and artist:



- Do all of the above in an RMarkdown file (submission instructions for Assignment 2 below)

Problem 2

Background: Data Scientists analyze visitor *sessions*, a sequence of user-initiated events on a web site that take place within a given time interval, to improve customer engagement and outcomes (usually the characteristics of sessions that lead to signups, purchases, or return visits).

In 2006, a 3.5 million row data set of AOL user's detailed search logs was released. The release was public, intended for research purposes, and AOL did not redact any personal information from the original relase (which lead to privacy concerns and several law suits). Here is a glimpse of the data in the data set:

```
Observations: 3,558,411
Variables: 5
$ AnonID    <int> 142, 142, 142, 142, 142, 142, 142, 142, 142, 142, 142, 14...
$ Query     <chr> "rentdirect.com", "www.prescriptionfortime.com", "staple....
$ QueryTime <dttm> 2006-03-01 07:17:12, 2006-03-12 12:31:06, 2006-03-17 21:...
$ ItemRank  <int> NA, NA, NA, NA, NA, NA, 1, NA, NA, NA, NA, NA, NA, NA, 1,...
$ ClickURL  <chr> NA, NA, NA, NA, NA, NA, "http://www.westchestergov.com", ...
```

Use dplyr, ggplot, tidy data principles, and the AOL search data (**aol_search_data.zip** posted in the Files/Assignments/Assignment 2 folder on Canvas) to accomplish the following:

- Convert all column names to snake case for all data frames using the dplyr **function fix_column_names** you developed for problem 1 (you do not need to repeat the code, just use the function)
- "Sessionize" the search records: aggregate the search records by *user session*, defined as all events for a given user where there is no more than a thirty minute gap between events. In other words, a session, which should be assigned a unique (per user) session sequence number, is series of visits (search events in this data set) by a given user, over any length of time, but with no more than a thirty minute interval from their last visit. Generate at least the following session statistics:

    - annon_id : annonomized AOL user id (from the AOL data set)
    - session_sequence_number : a unique (per user) session number (1, 2, 3, ... etc.)
    - session_id : a unique session id number generating by concatenating the user annon_id, underscore "_" session_sequence_number
    - number_searches : the number of searches per the session
    - session_started_at : session start time
    - session_ended_at : session end time
    - session_length : length of time between the start and end of the session
    - number_clicks : number of URL clicks per session
    - mean_item_rank : the mean item rank per session
    - mean_number_search_terms : mean number of search terms (whitespace separated words) per query per session

Compute and **plot** (using ggplot - geom_histogram and geom_density are useful for this, with appropriate titles, x and y axis labels):

- Statistics by session:

1. The distribution of session durations (histogram count)
2. The distribution of the number of clicks per session (histogram count)

- Statistics by user:

3. The distribution of the number of sessions by user (histogram count)
4. The distribution of mean session duration by user (histogram count)

Extra: in addition to the above further explore other aspects of the data set, such as item_rank, query_time, query and/or click_url fields. Be sure to clearly explain what you are doing and the results that you obtain. This is an opportunity for you to strike out on your own, define a question or two that interests you, and present some results.

Hand in three files (both problems 1 and 2 above in the same file):

1. Your RMarkdown file: asn2_firstname_lastname.Rmd
2. Knitted HTML with source: asn2_firstname_lastname_with_source.html (with source code, i.e. chunk option echo = TRUE)
3. Knitted HTML with no source: asn2_firstname_lastname.html (with no source code, i.e. chunk option echo = FALSE)

Additional Mandatory Specifications

1. Assume the json files are on the same directory as your .Rmd file (don't use setwd() in your code). Your HTML file results should be 100% reproducible: I should be able to Knit it to HTML without modification (of course once I set the working directory myself within RStudio)
2. Include in your results the as much problem description text as needed for the reader to understand what you are doing

3. In the YAML section of your .Rmd file, include title "Programming Assignment 2," author (your name and email on the same line), and date. Of course, output should be HTML

4. Use dplyr and ggplot2 in your solution, focus on using these and not (few) other packages to solve the problems

5. Follow the style guidelines described by Hadley Wickham in this R style guide

Grading:

300 points total. You will be graded on:

- Completeness

- Follows specifications (especially the mandatory specificcations above!)

- Implementation quality (code quality, adherence to style guidelines)

Remember: produce a knitted HTML product that you could hand to your Data Science Manager with the confidence that she would understand, and could reproduce, your report!