Assignment 2 – Data Extraction and Exploration

Total Points: 10

Instructions: Open the `Assign2_Data_Ext_Exp.Rproj` R project file located within the `Assign2_Data_Ext_Exp` subfolder of the `Assign` folder within `04_Data_Explore` folder and follow the instructions for each of the problems below. Unless otherwise indicated, each problem must be self-contained, i.e., if the R environment is cleared and only the code for a particular problem selected, the code must run without errors and produce the correct results in order to earn full credit for the problem. In addition, when indicated, you must use the vector, data frame and list names as specified in the problem. Failure to do so will result in an appropriate point deduction.

Problems 1 and 2 are located in `Assign21_Import_CSV_DB.r` file.

Problem 1 (3 points): United States Energy

The objective of this problem is to read and briefly analyze the content of **US_Energy.csv** file. The file contains information on total energy consumption, production (both in billions of BTUs) and expenditure (in millions of dollars) by state for year 2014. In addition, the GDP (in millions of dollars) and estimated population of each state is provided as well.

a) Set the current folder and path to the file using **cur_fld** and **us_energy_file** variables. You must use **read.csv** function to read the content of the file into **us_energy_df** data frame, display its structure and header.

```
'data.frame':	51 obs. of  7 variables:
 $ State   : chr  "Alabama" "Alaska" "Arizona" "Arkansas"
 $ Abbr    : chr  "AL" "AK" "AZ" "AR" ...
 $ GDP     : num  197535 58067 281559 121065 2324996 ...
 $ PopEst  : int  4849377 736732 2966369 6731484 38802500 5355866 3596677 935614 19893297 10097343
 $ TotCons : int  1958221 603119 1422590 1114409 7620082 1477177 750019 274013 4121680 2850990
 $ TotProd : int  1353725 1475129 635050 1454325 2413494 3041634 197271 4189 553738 597955
 $ TotExpnd: num  24147 6891 22610 13885 137720

       State Abbr        GDP   PopEst TotCons TotProd TotExpnd
1    Alabama   AL  197534.50  4849377 1958221 1353725  24146.5
2     Alaska   AK   58066.75   736732  603119 1475129   6890.6
3    Arizona   AZ  281558.75  2966369 1422590  635050  22609.6
4   Arkansas   AR  121064.75  6731484 1114409 1454325  13884.9
5 California   CA 2324995.50 38802500 7620082 2413494 137719.8
6   Colorado   CO  305366.75  5355866 1477177 3041634  19994.4
```

b) Calculate and add net exports (production minus consumption) to the data frame as **NetExport** variable. Store and display the names of the net exporting states in a vector **net_exporting_states**.

```
[1] "Alaska"    "Arkansas"    "Colorado"    "Kentucky"    "Montana"    "New Mexico"
[7] "North Dakota"   "Oklahoma"    "Pennsylvania"   "Texas"    "Utah"    "West Virginia"
[13] "Wyoming"
```

c) Calculate the fraction of GDP spent on energy (rounded to 2 decimals) and add to the data frame as **PctGDP** variable. Display the state names where the fraction of GDP spent on energy is above 10%.

```
[1] "Alabama"   "Alaska"    "Arkansas"   "Idaho"    "Indiana"
[6] "Iowa"   "Kentucky"   "Louisiana"   "Maine"    "Mississippi"
[11] "Montana"   "North Dakota"   "Oklahoma"   "South Carolina" "South Dakota"
[16] "Vermont"   "West Virginia"   "Wyoming"
```

Problem 2 (2 points): Order Entry Database

The Order Entry database contains data on customers and orders for computer technology products, including the employees taking those orders. The first step in the process is to recognize the already setup **OrderEntry.db** SQLite database in the project folder. The SQL code in **OrderEntry.sql** file is provided as a backup and there is nothing you need to do with it.

a) Load (after installing if necessary) the **RSQLite** package, and run the provided code to connect to the **OrderEntry.db** database and list all the tables, followed by all the columns in the **Sales** view and then create an SQL query to select all the rows and columns from the view. Execute the query into **cust_ords_df** data frame, show the structure and display the entire data frame (not shown below).

```
[1] "Customer"  "Employee"  "OrderLine" "OrderTbl"  "Product"  "Sales"

[1] "CustNo"        "CustFirstName" "CustLastName"  "TotQty"        "TotSales"

'data.frame':     13 obs. of  5 variables:
 $ CustNo       : chr  "C0954327" "C1010398" "C2388597" "C3340959" ...
 $ CustFirstName: chr  "Sheri" "Jim" "Beth" "Betty" ...
 $ CustLastName : chr  "Gordon" "Glussman" "Taylor" "Wise" ...
 $ TotQty       : int  4 3 4 5 7 1 3 2 2 2 ...
 $ TotSales     : num  295 149 592 190 1467 ...
```

b) In a single row data frame named **top_cust_df**, display the entire record of the customer with the highest sales, stored in a variable named **max_sales**. Create a data frame, named **bottom_qty_cust_df** of customers who had a total quantity ordered of less than 3.

```
     CustNo CustFirstName CustLastName TotQty TotSales
12 C9865874          Mary         Hill     13  2415.99

     CustNo CustFirstName CustLastName TotQty TotSales
6  C8574932         Wally        Jones      1   199.99
8  C9128574         Jerry        Wyatt      2   103.99
9  C9403348          Mike        Boren      2   288.99
10 C9432910         Larry       Styles      2   183.99
13 C9943201         Harry      Sanders      2   124.69
```

Problems 3 and 4 are located in `Assign22_Import_Web.r` file.

Problem 3 (2 points): US Media Markets

The main purpose of this problem is to get you to read Web data in HTML format into R for the purpose of further analysis. Start by learning about the major US media markets by first examining the Nielsen's Web page (https://mediatracks.com/resources/nielsen-dma-rankings-2020/) showing the Designated Market Area (DMA) rankings. There is only one table you need to read into R using **rvest** package, **read_html**, **html_nodes** and **html_table** functions as shown at the top of the problem.

Note: From now on getting a **tibble** instead of the **classic data frame** and vice versa is **NOT important**, both represent the same data structure.

a) Modify **dma_rank_df** data frame by removing the rank column and total row, and then rename the column headers as **DMA**, **TV_Homes**, and **PCT_US**. Display the structure and the header, and convert TV_Homes from character to numeric column (use the provided code with **as.numeric** and **gsub** functions). Redisplay the structure and header to verify it worked.

```
'data.frame':  210 obs. of  3 variables:
 $ DMA      : chr  "New York" "Los Angeles" "Chicago" "Philadelphia" ...
 $ TV_Homes: num  6824120 5145350 3256400 2758330 2563320 ...
 $ PCT_US   : num  6.38 4.81 3.04 2.58 2.4 ...
```

b) List the top part of the data frame where number of TV homes is higher than its average. Roughly the top quarter (or 57 to be precise) out of 210 DMA's have the number of TV homes higher than the average. This should cover the 41 US metro areas with all major league sports we explore in the next problem. Make sure to unload the **rvest** package.

```
# A tibble: 57 x 3
   DMA                         TV_Homes PCT_US
   <chr>                          <dbl>  <dbl>
 1 New York                     6824120   6.38
 2 Los Angeles                  5145350   4.81
 3 Chicago                      3256400   3.04
 4 Philadelphia                 2758330   2.58
 5 Dallas-Ft. Worth             2563320   2.40
 6 San Francisco-Oak-San Jose   2364740   2.21
 7 Washington, DC (Hagrstwn)    2351930   2.20
 8 Houston                      2330180   2.18
 9 Boston (Manchester)          2302680   2.15
10 Atlanta                      2269270   2.12
# ... with 47 more rows
```

Problem 4 (3 point): Minneapolis Weather

The US National Weather Service measures cloud cover in percentage terms (0%=clear, 100%=overcast). The file **MSP_Weather.json** contains 48 weather observations of Minneapolis provided by OpenWeather at openweathermap.com. The structure of the file is a bit more complex than some of our classroom examples, and some of the data may not initially make sense, so you should spend some time exploring the data and its structure before attempting the tasks below.

a) Read the data from the file and locate the portion of the relevant data frame naming it **minn_df**. Display the structure and header for examination (I am only showing the header below).

```
  main.temp main.humidity main.pressure main.temp_min main.temp_max wind.speed wind.gust wind.deg all
1    301.15            43          1011        299.82        302.59       1.54      2.06  45.0000   0
2    300.93            39          1012        298.71        302.59       2.60        NA 110.0000   1
3    299.83            54          1011        298.15        301.15       1.50        NA  90.0000   1
4    300.07            50          1010        298.71        301.48       0.51      1.54 130.0000   0
5    298.27            60          1012        294.82        301.15       3.81        NA  92.5024   1
6    297.89            60          1012        295.15        300.37       1.50        NA 100.0000   1
                     weather         dt rain.1h rain.3h
1 800, Clear, Sky is Clear, 01n 1403396630      NA      NA
2 800, Clear, sky is clear, 01n 1403396672      NA      NA
3 800, Clear, sky is clear, 01n 1403400223      NA      NA
4 800, Clear, Sky is Clear, 01n 1403400281      NA      NA
5 800, Clear, sky is clear, 01n 1403403862      NA      NA
6 800, Clear, sky is clear, 01n 1403403899      NA      NA
```

b) Using **minn_df** data frame, and given the **cloud_cover** variable, calculate the number of days with the specified cloud cover. Store your result in a variable named **num_days**.

```
[1] 11
```

c) Find all temperature observations (in degrees Fahrenheit) for records with the specified cloud cover. Store your result in a vector variable named **temp_F**.

```
[1] 68.630 68.702 72.158 72.212 72.032 72.086 72.086 72.662 72.662 73.490 73.526
```

Submission: You must submit the completed `Assign2_Data_Ext_Exp` project folder, zipped up into a compressed folder by the same name on Canvas by the designated due date.