Assignment 3 – Data Manipulation

Total Points: **10**

Instructions: Open the `Assign3_Data_Manip.Rproj` R project file located within the `Assign3_Data_Manip` subfolder of the `Assign` folder within `05_Data_Manip` 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 and data frame as specified in the problem. Failure to do so will result in an appropriate point deduction.

Problems 1 and 2 are located in `Assign41_Fortune500_FDIC.r` file.

Problem 1 (2 points): **Fortune500 Companies**

The **Fortune500.csv** file contains historical data on revenues and profits for the Fortune 500 companies since the 50's and as such is a sizable dataset. Make sure to open this file and briefly review its content. Execute the top portion of the code to read the file into R and display the standard information.

a) Use the clean data frame to find the average profit for companies in **1988** using filter and summarize functions from **dplyr** package.

```
  AvgProfit88
1    188.6937
```

b) List the year, rank, company name and revenue of top 5 ranked companies for each of the first 10 years of this millennium (2000 – 2009) using **filter** and **select** functions from **dplyr** package. Save the result into **fortune500_top5_df**.

```
  Year Rank          Company Revenue
1 2000    1   General Motors  189058
2 2000    2  Wal-Mart Stores  166809
3 2000    3      Exxon Mobil  163881
4 2000    4       Ford Motor  162558
5 2000    5 General Electric  111630
6  2001    1      Exxon Mobil  210392
```

Then find the average by year during the same time period using **group_by** and summarize functions from **dplyr** package.

```
# A tibble: 10 x 2
    Year AvgRevenue
   <int>      <dbl>
 1  2000    158787.
 2  2001    179754
 3  2002    177957.
 4  2003    182216.
 5  2004    193242.
 6  2005    215415.
 7  2006    242977.
 8  2007    255752
 9  2008    264662.
10  2009    305118.
```

Problem 2 (3 points): **FDIC Banks**

The US Federal Deposit Insurance Corporation (FDIC) is responsible for insuring US banks against failure with sample data provided in **FDIC.csv** file. Make sure to open this file and briefly review its content. Execute the top portion of the code to read the file into R and display the standard information

a) Using **dplyr** functions **count** and **arrange**, count the number of banks by state into **state_freq_df** data frame, name the data frame column **BankCount** and sort the data frame on **BankCount** descending. The header is shown below.

```
  ST BankCount
1 GA        93
2 FL        75
3 IL        69
4 CA        41
5 MN        23
6 WA        19
```

b) Extract the year out of **Closing.Date** column using a combination of **nchar** and **substr** functions. Append **Acquisition.Year** column to the **fdic_df** data frame and display the header. Summarize the number of acquisitions by year, using the same **count** function from the previous step, into **year_freq_df** data frame, rename the columns **Year** and **AcqCount** and sort the data frame on **AcqCount** descending. The (partial) headers of both are shown below and above.

```
                        Bank.Name          City ST  CERT      Acquiring.Institution
1            The First State Bank Barboursville WV 14361             MVB Bank, Inc.
2              Ericson State Bank       Ericson NE 18265 Farmers and Merchants Bank
3 City National Bank of New Jersey        Newark NJ 21111            Industrial Bank
       Closing.Date Acquisition.Year
1     April 3, 2020             2020
2 February 14, 2020             2020
3  November 1, 2019             2019
```

c) Summarize the number of acquisitions by year, using the same **count** function from the previous step, into **year_freq_df** data frame, rename the columns **Year** and **AcqCount** and sort the data frame on **AcqCount** descending. The (partial) headers of both are shown below and above.

```
   Year AcqCount
1  2010      157
2  2009      140
3  2011       92
4  2012       51
5  2008       25
6  2013       24
```

Problem 3 is located in `Assign42_Dillards_Stores.r` file.

Problem 3 (5 points): **Dillard's Department Stores**

The holiday season in between US Thanksgiving and Christmas is an important time for US retailers. The file **Dillards2004.txt** contains a sample of transactions at various Dillard's department stores during the peak of the holiday shopping season in 2004. The MSA, STORE and DEPT columns are all unique ID numbers of various metropolitan statistical areas, stores, and departments. The STYPE column indicates whether the transaction is a Purchase or Return. The ORIG column represents the original price of all items in the transaction, ACTUAL represents the actual price paid by the customer. MARKDOWN should be ORIG – ACTUAL, but this is real data with some anomalies.

a) Examine the issues that arise when you attempt to convert the **ORIG** prices to numbers. Notice the single missing value and a number of zeros? Remove these anomalies in a redefined **dillars_df** data frame. Make sure to convert the **ORIG** prices to numbers and verify the reduced size of the frame.

```
'data.frame':    1827 obs. of  8 variables:
 $ MSA     : int  2680 2680 2680 2680 2680 2680 2680 2680 2680 2680 ...
 $ STORE   : int  5002 5002 5002 5002 5002 5002 5002 5002 5002 5002 ...
 $ DEPT    : int  800 800 801 801 1100 1100 1202 1202 1301 1301 ...
 $ DEPTDESC: chr  "CLINIQUE" "CLINIQUE" "LESLIE  " "LESLIE  " ...
 $ STYPE   : chr  "R" "P" "R" "P" ...
 $ ORIG    : num  2385 51576 5132 29448 6039 ...
 $ ACTUAL  : num  2385 51561 4330 18104 6039 ...
 $ MARKDOWN: num  0 15 802 11344 0 ...
```

b) Add a column to the data frame named **MARKDOWN_PCT** that indicates the percentage markdown of each transaction. **MARKDOWN_PCT** should be a decimal number, rounded to 3 places.

```
   MSA STORE DEPT DEPTDESC STYPE     ORIG   ACTUAL MARKDOWN MARKDOWN_PCT
1 2680  5002  800 CLINIQUE     R  2385.00  2385.00     0.00        0.000
2 2680  5002  800 CLINIQUE     P 51575.50 51560.50    15.00        0.000
3 2680  5002  801 LESLIE      R  5131.99  4329.62   802.37        0.156
4 2680  5002  801 LESLIE      P 29447.91 18104.05 11343.86        0.385
```

d) Given the variable store and using **dplyr** functions, calculate the average of the transaction amounts for the store named **AvgOrigStore**.

```
  AvgOrigStore
1     11738.08
```

e) Using **dplyr** functions find the number of transactions per store, named **NumTrans**, and average markdown per store, named **AvgMkdDwn**. Arrange the result descending on the number of transactions per store and show only those stores with over 110 transactions.

```
# A tibble: 5 x 3
  STORE NumTrans AvgMkdDwn
  <int>    <int>     <dbl>
1  7007      114     4387.
2  7507      112     6545.
3  5102      111     4666.
4  7707      111     4778.
5  7907      111     6108.
```

f) Using **dplyr** functions find the average markdown percentage, named **AvgMkdDwnPct** by store and department. Arrange the result descending on the average markdown percentage. Shown only those stores and departments where average markdown percentage is over 75%.

```
# A tibble: 9 x 4
# Groups:   STORE, DEPT [9]
  STORE  DEPT DEPTDESC   AvgMkdDwnPct
  <int> <int> <chr>             <dbl>
1  5302  9801 "CATALIN "        0.933
2  5102  9801 "CATALIN "        0.905
3  5402  4801 "GOTTEX  "        0.883
4  5002  9801 "CATALIN "        0.867
5  5402  9801 "CATALIN "        0.863
6  7007  8104 "COP KEY "        0.812
7  5202  9801 "CATALIN "        0.798
8  7707  9801 "CATALIN "        0.787
9  5602  9801 "CATALIN "        0.752
```

Submission: You must submit the completed `Assign4_Data_Manip` project folder, zipped up into a compressed folder by the same name on Canvas by the designated due date.