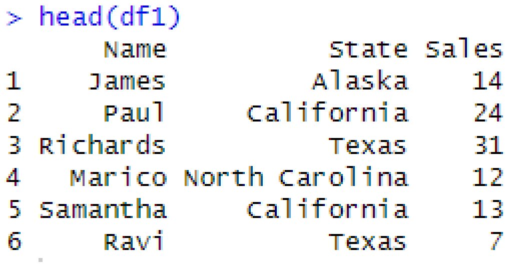
Posted on GitHub at https://github.com/1fastgranada/CSC302_HW2/

1. Run the following lines and study how they work. Then state what they do and output for us. (20 Points)

The following code creates a dataframe *df1* which consists of 3 columns and 12 rows:

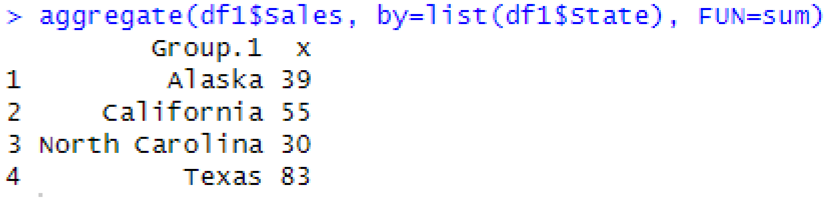
```
df1=data.frame(Name=c('James','Paul','Richards','Marico','Samantha','Ravi','Raghu',
                      'Richards','George','Ema','Samantha','Catherine'),
               State=c('Alaska','California','Texas','North Carolina','California','Texas',
                       'Alaska','Texas','North Carolina','Alaska','California','Texas'),
               Sales=c(14,24,31,12,13,7,9,31,18,16,18,14))
```



```
> head(df1)
      Name          State Sales
1    James         Alaska    14
2     Paul     California    24
3 Richards          Texas    31
4   Marico North Carolina    12
5 Samantha     California    13
6     Ravi          Texas     7
```

The following code utilizes the aggregate() function to sum the Sales by State:

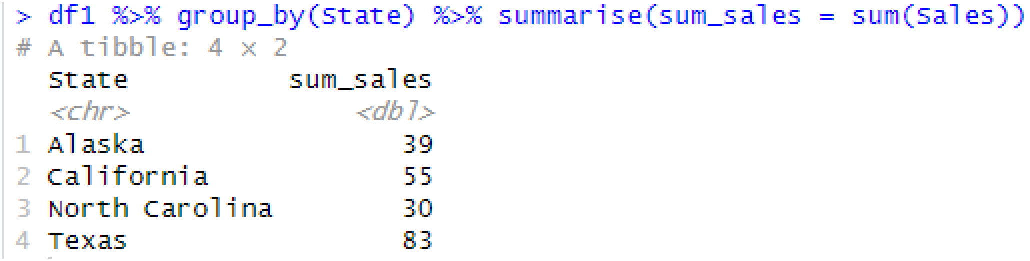
```
aggregate(df1$Sales, by=list(df1$State), FUN=sum)
```



```
> aggregate(df1$Sales, by=list(df1$State), FUN=sum)
         Group.1  x
1         Alaska 39
2     California 55
3 North Carolina 30
4          Texas 83
```

The following code utilizes dplyr function to do the same sum of the Sales by State:

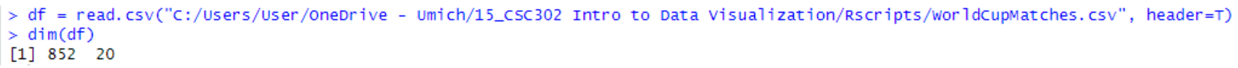
```
library(dplyr)
df1 %>% group_by(State) %>% summarise(sum_sales = sum(Sales))
```



```
> df1 %>% group_by(State) %>% summarise(sum_sales = sum(Sales))
# A tibble: 4 × 2
  State          sum_sales
  <chr>              <dbl>
1 Alaska                39
2 California            55
3 North Carolina        30
4 Texas                 83
```

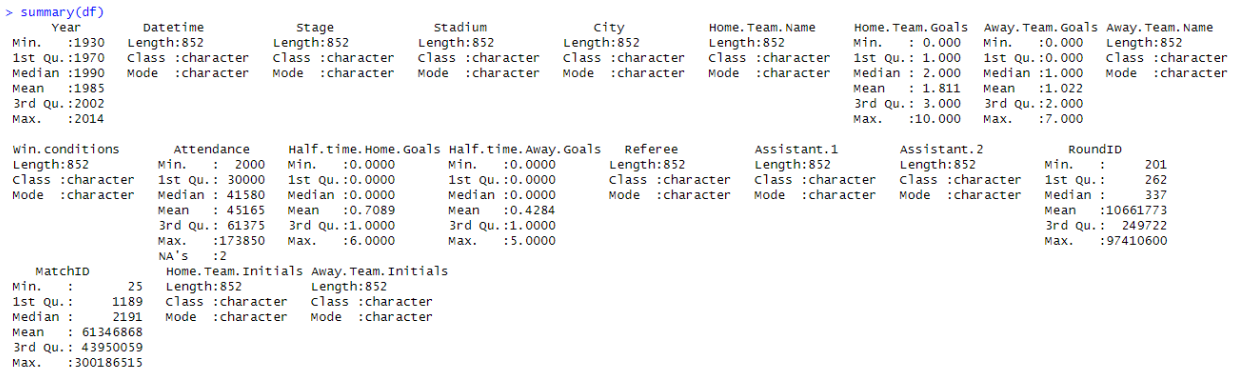
Posted on GitHub at https://github.com/1fastgranada/CSC302_HW2/

2. Use R to read the WorldCupMatches.csv from the DATA folder on Google Drive. Then perform the followings (48 points):
    a. Find the size of the data frame. How many rows, how many columns?



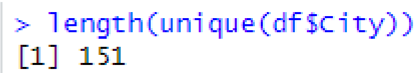
```
> df = read.csv("C:/Users/User/OneDrive - Umich/15_CSC302 Intro to Data Visualization/Rscripts/WorldCupMatches.csv", header=T)
> dim(df)
[1] 852  20
```

    b. Use summary function to report the statistical summary of your data.



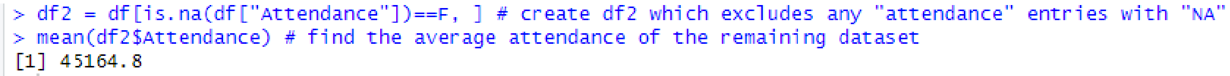
```
> summary(df)
      Year         Datetime              Stage              Stadium               City           Home.Team.Name     Home.Team.Goals  Away.Team.Goals Away.Team.Name    
 Min.   :1930   Length:852         Length:852         Length:852         Length:852         Length:852         Min.   : 0.000   Min.   :0.000   Length:852        
 1st Qu.:1970   Class :character   Class :character   Class :character   Class :character   Class :character   1st Qu.: 1.000   1st Qu.:0.000   Class :character  
 Median :1990   Mode  :character   Mode  :character   Mode  :character   Mode  :character   Mode  :character   Median : 2.000   Median :1.000   Mode  :character  
 Mean   :1985                                                                                                  Mean   : 1.811   Mean   :1.022                     
 3rd Qu.:2002                                                                                                  3rd Qu.: 3.000   3rd Qu.:2.000                     
 Max.   :2014                                                                                                  Max.   :10.000   Max.   :7.000                     
                                                                                                                                                                   
 Win.conditions       Attendance     Half.time.Home.Goals Half.time.Away.Goals   Referee           Assistant.1        Assistant.2           RoundID         
 Length:852         Min.   :  2000   Min.   :0.0000       Min.   :0.0000       Length:852         Length:852         Length:852         Min.   :      201  
 Class :character   1st Qu.: 30000   1st Qu.:0.0000       1st Qu.:0.0000       Class :character   Class :character   Class :character   1st Qu.:      262  
 Mode  :character   Median : 41580   Median :0.0000       Median :0.0000       Mode  :character   Mode  :character   Mode  :character   Median :      337  
                    Mean   : 45165   Mean   :0.7089       Mean   :0.4284                                                                Mean   :10661773  
                    3rd Qu.: 61375   3rd Qu.:1.0000       3rd Qu.:1.0000                                                                3rd Qu.:   249722  
                    Max.   :173850   Max.   :6.0000       Max.   :5.0000                                                                Max.   :97410600  
                    NA's   :2                                                                                                                            
    MatchID          Home.Team.Initials Away.Team.Initials
 Min.   :       25   Length:852         Length:852        
 1st Qu.:     1189   Class :character   Class :character  
 Median :     2191   Mode  :character   Mode  :character  
 Mean   : 61346868                                        
 3rd Qu.: 43950059                                        
 Max.   :300186515                                        
```

    c. Find how many unique locations olympics were held at.



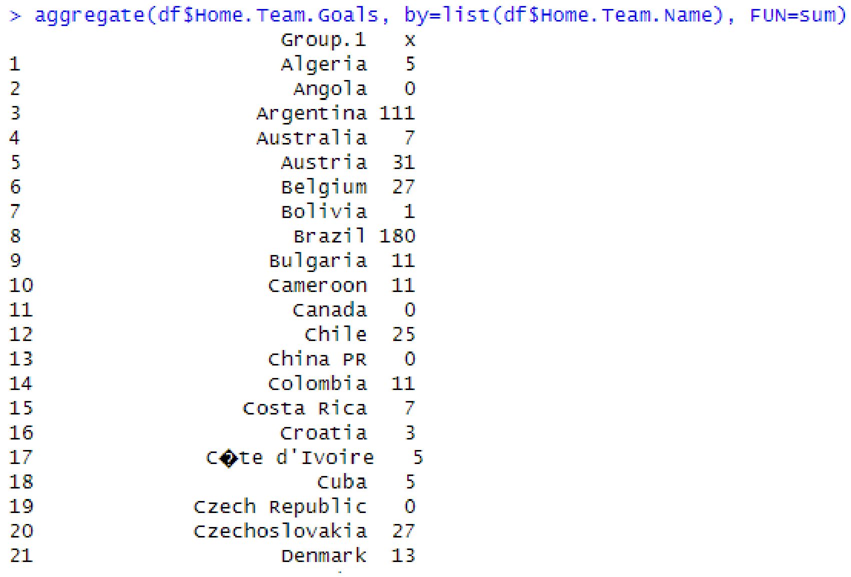
```
> length(unique(df$City))
[1] 151
```

    d. Find the average attendance.



```
> df2 = df[is.na(df["Attendance"])==F, ] # create df2 which excludes any "attendance" entries with "NA"
> mean(df2$Attendance) # find the average attendance of the remaining dataset
[1] 45164.8
```

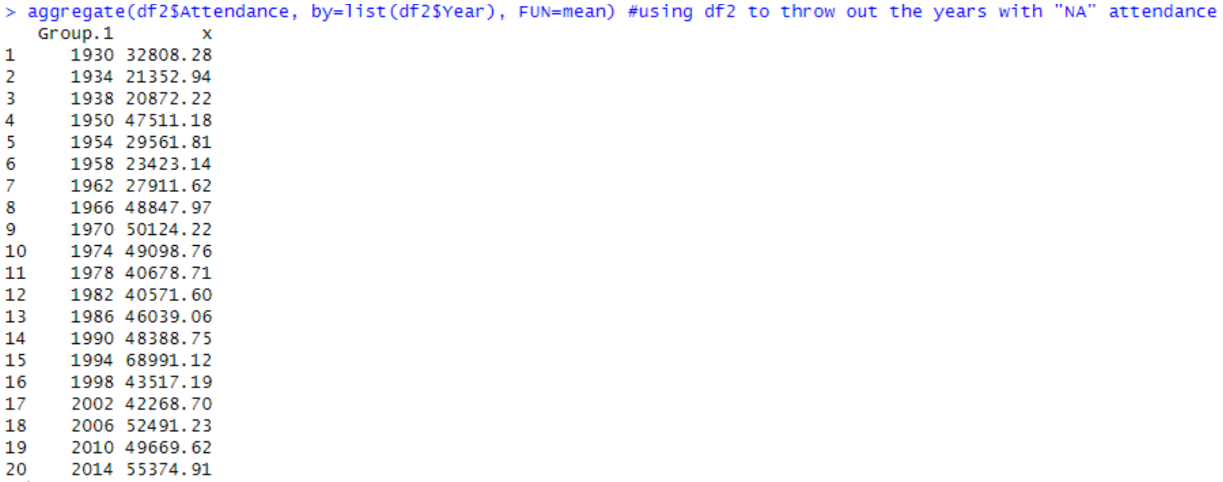
    e. For each Home Team, what is the total number of goals scored? (Hint: Please refer to question 1)



```
> aggregate(df$Home.Team.Goals, by=list(df$Home.Team.Name), FUN=sum)
                      Group.1   x
1                     Algeria   5
2                      Angola   0
3                   Argentina 111
4                   Australia   7
5                     Austria  31
6                     Belgium  27
7                     Bolivia   1
8                      Brazil 180
9                    Bulgaria  11
10                   Cameroon  11
11                     Canada   0
12                      Chile  25
13                   China PR   0
14                   Colombia  11
15                 Costa Rica   7
16                    Croatia   3
17              C�te d'Ivoire   5
18                       Cuba   5
19             Czech Republic   0
20             Czechoslovakia  27
21                    Denmark  13
```

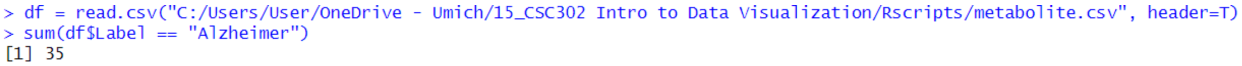
Posted on GitHub at https://github.com/1fastgranada/CSC302_HW2/

f. What is the average number of attendees for each year? Is there a trend or pattern in the data in that sense?



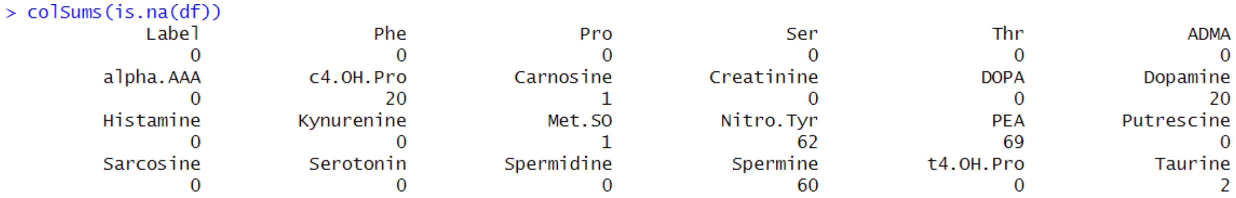
```
> aggregate(df2$Attendance, by=list(df2$Year), FUN=mean) #using df2 to throw out the years with "NA" attendance
   Group.1        x
1     1930 32808.28
2     1934 21352.94
3     1938 20872.22
4     1950 47511.18
5     1954 29561.81
6     1958 23423.14
7     1962 27911.62
8     1966 48847.97
9     1970 50124.22
10    1974 49098.76
11    1978 40678.71
12    1982 40571.60
13    1986 46039.06
14    1990 48388.75
15    1994 68991.12
16    1998 43517.19
17    2002 42268.70
18    2006 52491.23
19    2010 49669.62
20    2014 55374.91
```

3. Use R to read the metabolites.csv from the DATA folder on Google Drive. Then perform the followings (32 points):
    a. Find how many Alzheimers patients there are in the data set. (Hint: Please refer to question 1)



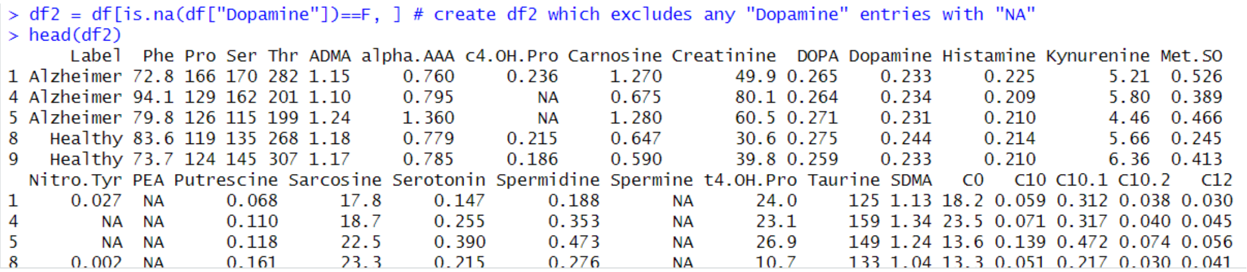
```
> df = read.csv("C:/Users/User/OneDrive - Umich/15_CSC302 Intro to Data Visualization/Rscripts/metabolite.csv", header=T)
> sum(df$Label == "Alzheimer")
[1] 35
```

    b. Determine the number of missing values for each column. (Hint: is.na( ))



```
> colSums(is.na(df))
      Label         Phe         Pro         Ser         Thr        ADMA 
          0           0           0           0           0           0 
  alpha.AAA    c4.OH.Pro   Carnosine  Creatinine        DOPA    Dopamine 
          0          20           1           0           0          20 
  Histamine  Kynurenine      Met.SO   Nitro.Tyr         PEA  Putrescine 
          0           0           1          62          69           0 
  Sarcosine   Serotonin  Spermidine    Spermine   t4.OH.Pro     Taurine 
          0           0           0          60           0           2 
```

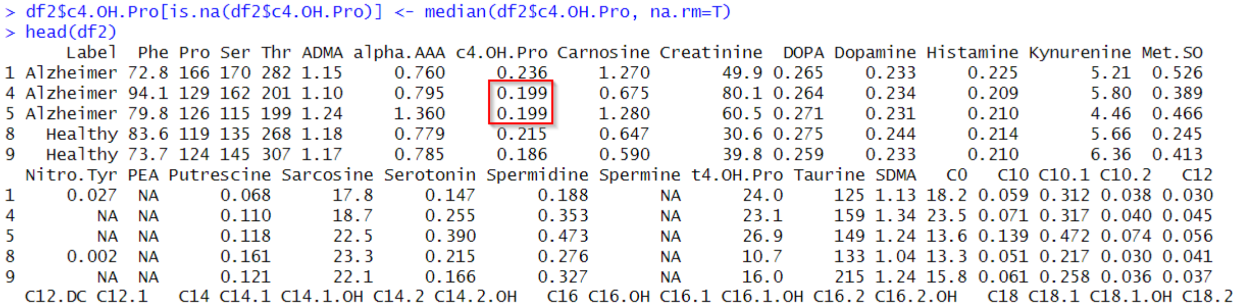
    c. Remove the rows which has missing value for the Dopamine column and assign the result to a new data frame. (Hint: is.na( ))



```
> df2 = df[is.na(df["Dopamine"])==F, ] # create df2 which excludes any "Dopamine" entries with "NA"
> head(df2)
      Label  Phe Pro Ser Thr ADMA alpha.AAA c4.OH.Pro Carnosine Creatinine  DOPA Dopamine Histamine Kynurenine Met.SO
1 Alzheimer 72.8 166 170 282 1.15     0.760     0.236     1.270       49.9 0.265    0.233     0.225       5.21  0.526
4 Alzheimer 94.1 129 162 201 1.10     0.795        NA     0.675       80.1 0.264    0.234     0.209       5.80  0.389
5 Alzheimer 79.8 126 115 199 1.24     1.360        NA     1.280       60.5 0.271    0.231     0.210       4.46  0.466
8   Healthy 83.6 119 135 268 1.18     0.779     0.215     0.647       30.6 0.275    0.244     0.214       5.66  0.245
9   Healthy 73.7 124 145 307 1.17     0.785     0.186     0.590       39.8 0.259    0.233     0.210       6.36  0.413
  Nitro.Tyr PEA Putrescine Sarcosine Serotonin Spermidine Spermine t4.OH.Pro Taurine SDMA   C0   C10 C10.1 C10.2   C12
1     0.027  NA      0.068      17.8     0.147      0.188       NA      24.0     125 1.13 18.2 0.059 0.312 0.038 0.030
4        NA  NA      0.110      18.7     0.255      0.353       NA      23.1     159 1.34 23.5 0.071 0.317 0.040 0.045
5        NA  NA      0.118      22.5     0.390      0.473       NA      26.9     149 1.24 13.6 0.139 0.472 0.074 0.056
8     0.002  NA      0.161      23.3     0.215      0.276       NA      10.7     133 1.04 13.3 0.051 0.217 0.030 0.041
```

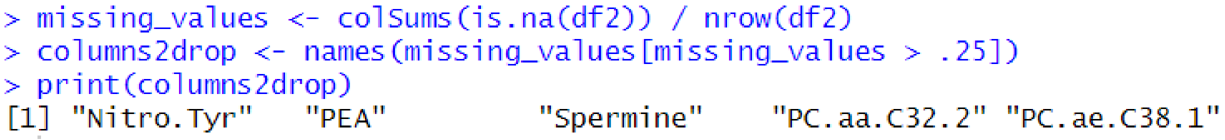
    d. In the new data frame, replace the missing values in the c4-OH-Pro column with the median value of the same column. (Hint: there is median( ) function.)

Posted on GitHub at https://github.com/1fastgranada/CSC302_HW2/



```
> df2$c4.OH.Pro[is.na(df2$c4.OH.Pro)] <- median(df2$c4.OH.Pro, na.rm=T)
> head(df2)
      Label  Phe Pro Ser Thr ADMA alpha.AAA c4.OH.Pro Carnosine Creatinine  DOPA Dopamine Histamine Kynurenine Met.SO
1 Alzheimer 72.8 166 170 282 1.15     0.760     0.236     1.270       49.9 0.265    0.233     0.225       5.21  0.526
4 Alzheimer 94.1 129 162 201 1.10     0.795     0.199     0.675       80.1 0.264    0.234     0.209       5.80  0.389
5 Alzheimer 79.8 126 115 199 1.24     1.360     0.199     1.280       60.5 0.271    0.231     0.210       4.46  0.466
8   Healthy 83.6 119 135 268 1.18     0.779     0.215     0.647       30.6 0.275    0.244     0.214       5.66  0.245
9   Healthy 73.7 124 145 307 1.17     0.785     0.186     0.590       39.8 0.259    0.233     0.210       6.36  0.413
  Nitro.Tyr PEA Putrescine Sarcosine Serotonin Spermidine Spermine t4.OH.Pro Taurine SDMA   C0   C10 C10.1 C10.2   C12
1     0.027  NA      0.068      17.8     0.147      0.188       NA      24.0     125 1.13 18.2 0.059 0.312 0.038 0.030
4        NA  NA      0.110      18.7     0.255      0.353       NA      23.1     159 1.34 23.5 0.071 0.317 0.040 0.045
5        NA  NA      0.118      22.5     0.390      0.473       NA      26.9     149 1.24 13.6 0.139 0.472 0.074 0.056
8     0.002  NA      0.161      23.3     0.215      0.276       NA      10.7     133 1.04 13.3 0.051 0.217 0.030 0.041
9        NA  NA      0.121      22.1     0.166      0.327       NA      16.0     215 1.24 15.8 0.061 0.258 0.036 0.037
  C12.DC C12.1   C14 C14.1 C14.1.OH C14.2 C14.2.OH   C16 C16.OH C16.1 C16.1.OH C16.2 C16.2.OH   C18 C18.1 C18.1.OH C18.2
```

e. (Optional) Drop columns which have more than 25% missing values. (Hint: when you slice your data frame, you can use -c(.., ..., ...) where ... represent one column name)



```
> missing_values <- colSums(is.na(df2)) / nrow(df2)
> columns2drop <- names(missing_values[missing_values > .25])
> print(columns2drop)
[1] "Nitro.Tyr"   "PEA"         "Spermine"    "PC.aa.C32.2" "PC.ae.C38.1"
```

I couldn't get the -c(...,...,...) to work with column names. I got the list of column names, but didn't actually drop them.