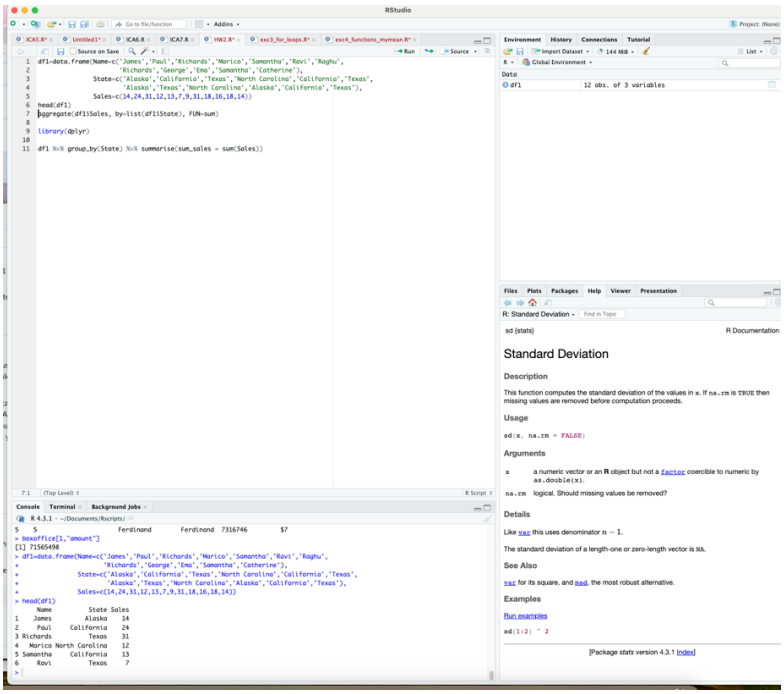
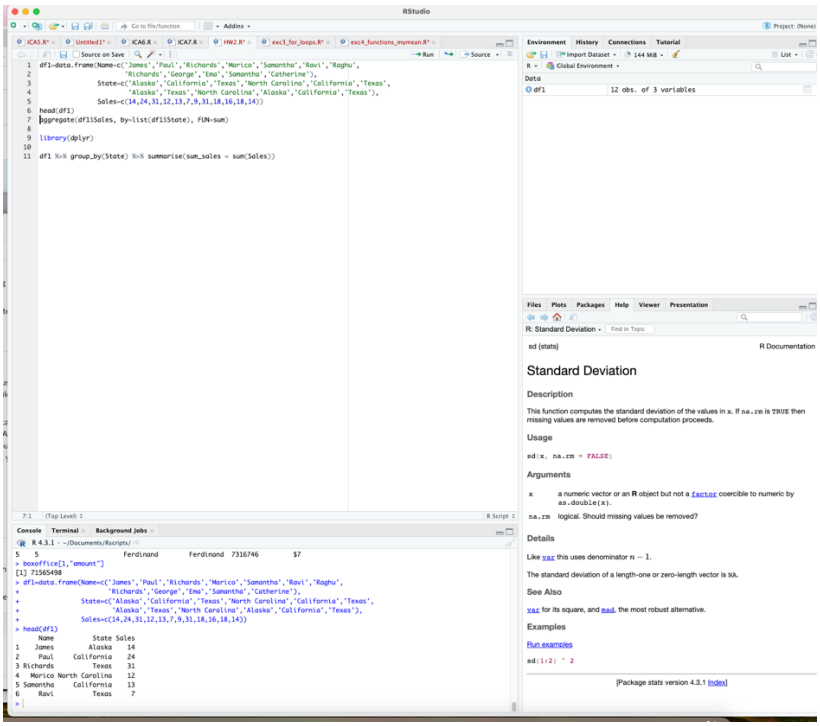
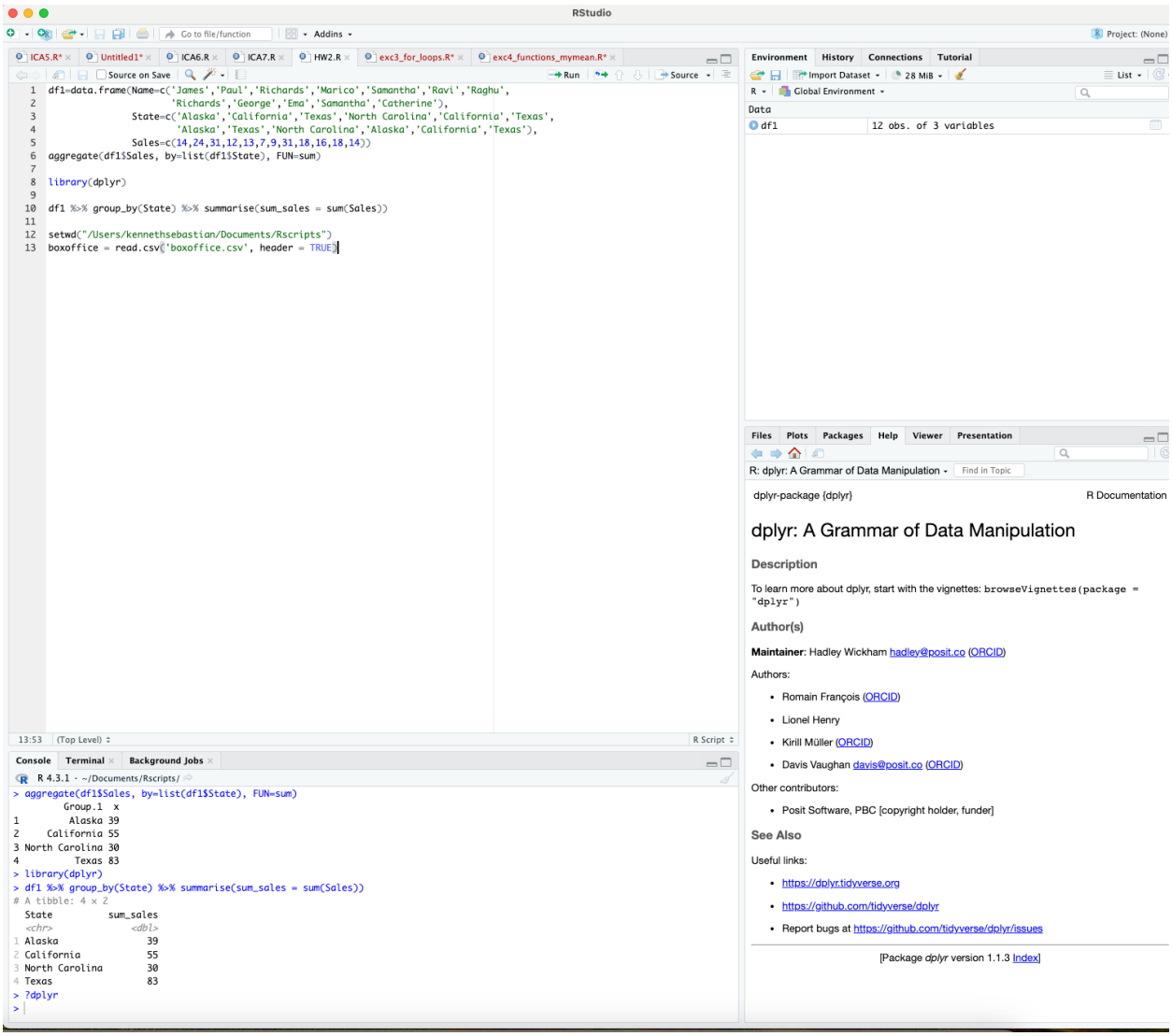


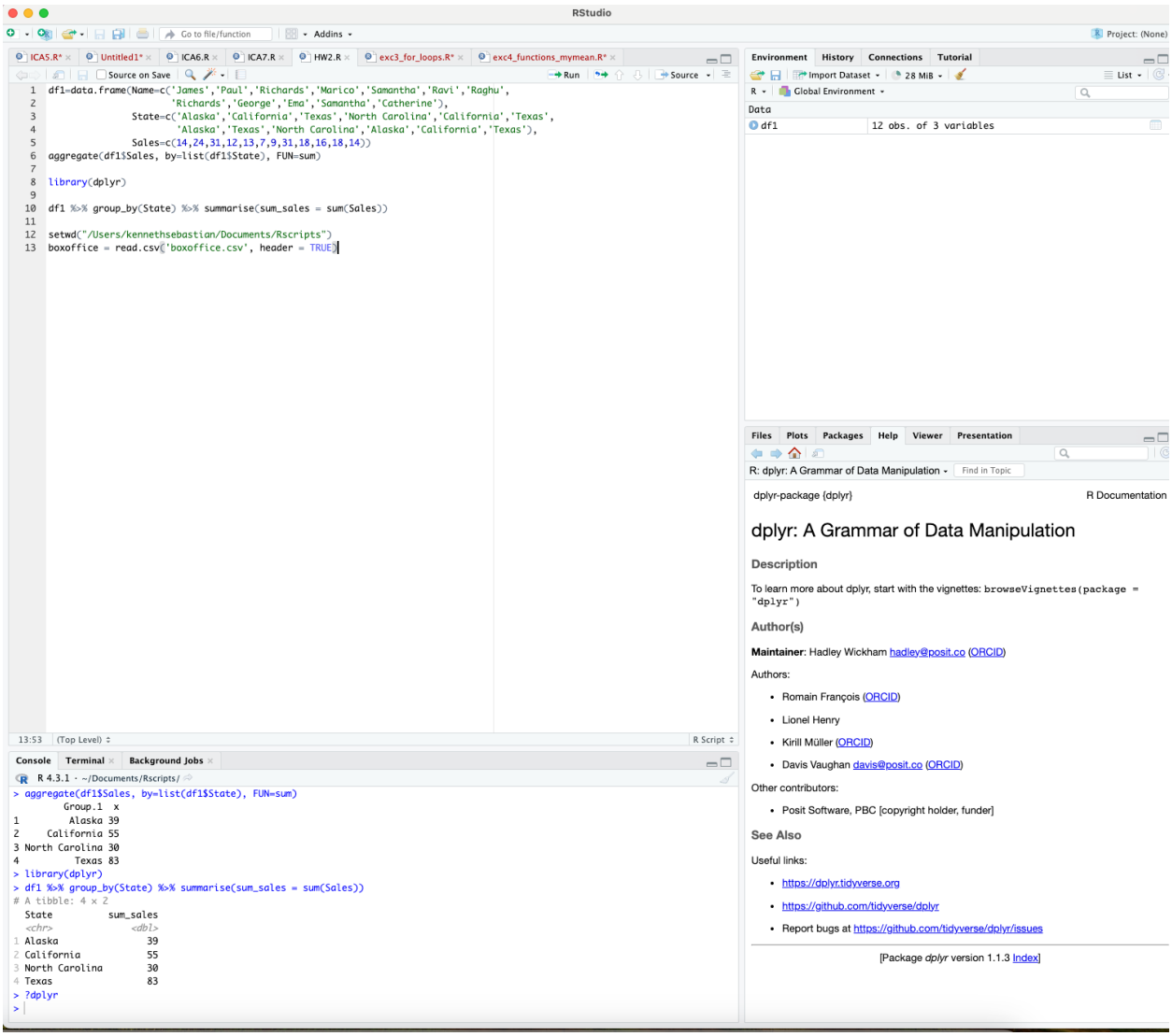
Creates a data frame df 1 with columns Name,State, Sales and 6 rows.



Computes the summary of sales by states



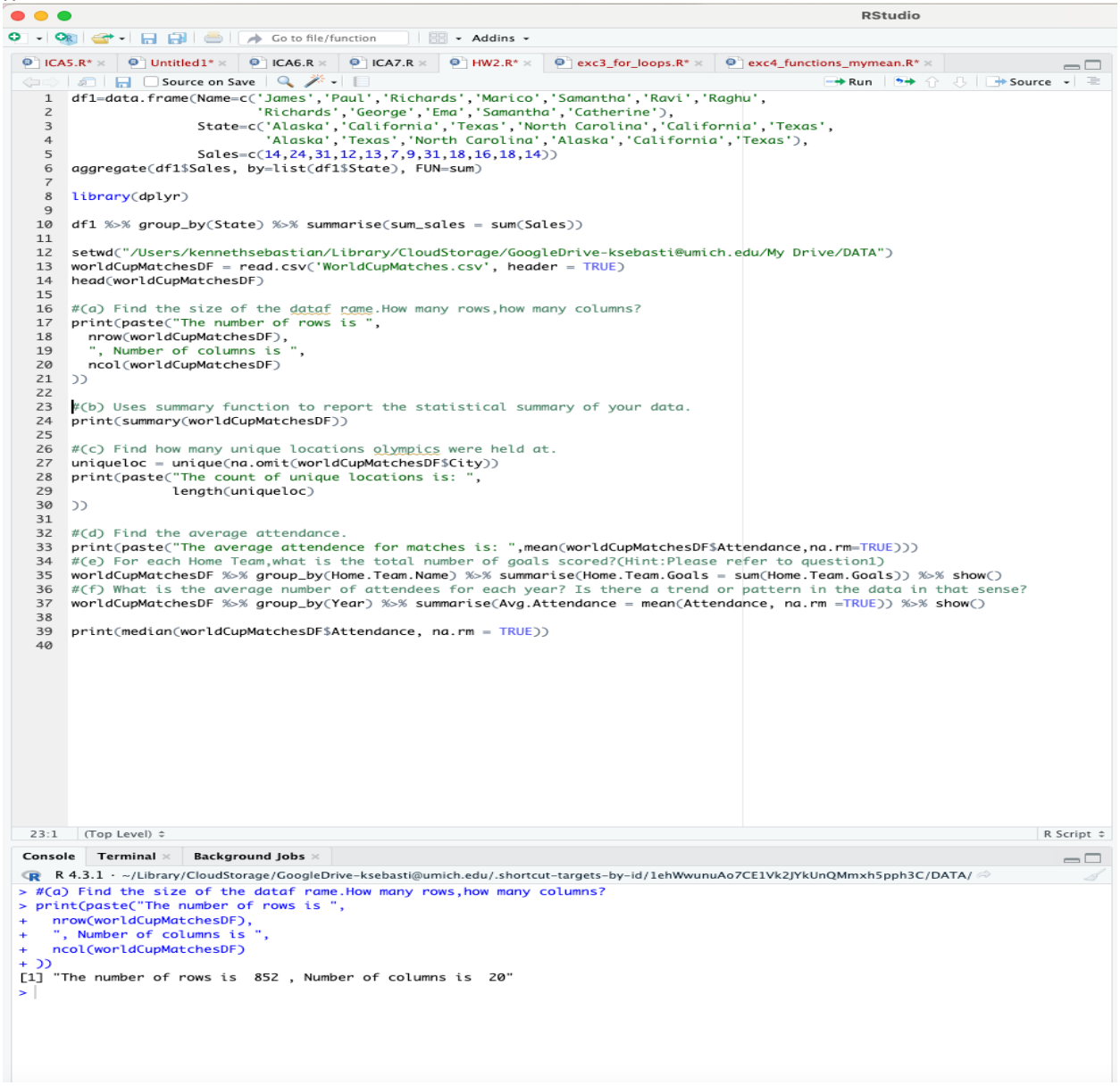
Library(dplyr) loads library dplyr which provides a set of verbs for data maniipulation



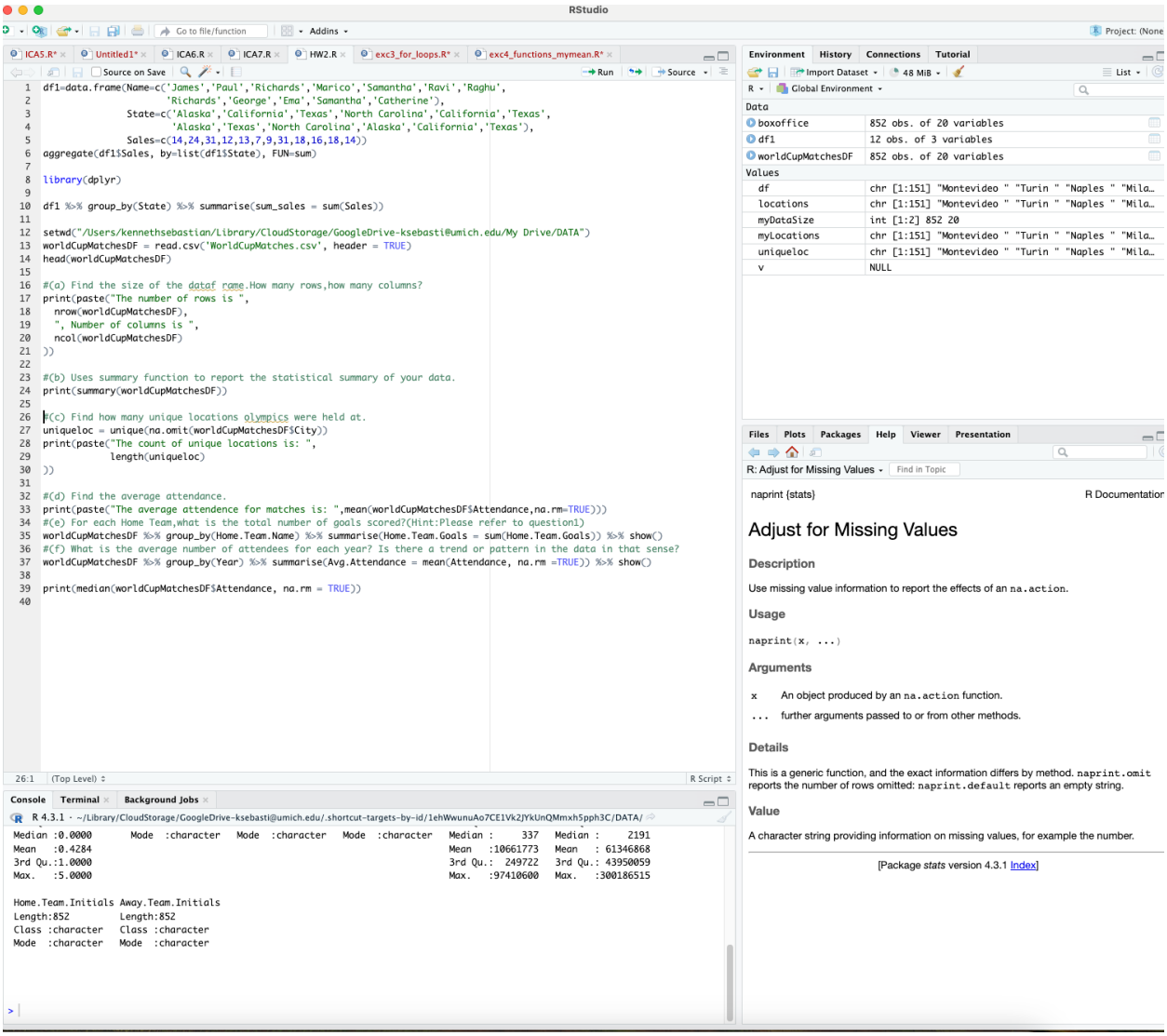
Line df1 %>% group_by(State) %>% summarise(sum_sales = sum(Sales))

Uses library dplyr to group summary of sales by states.

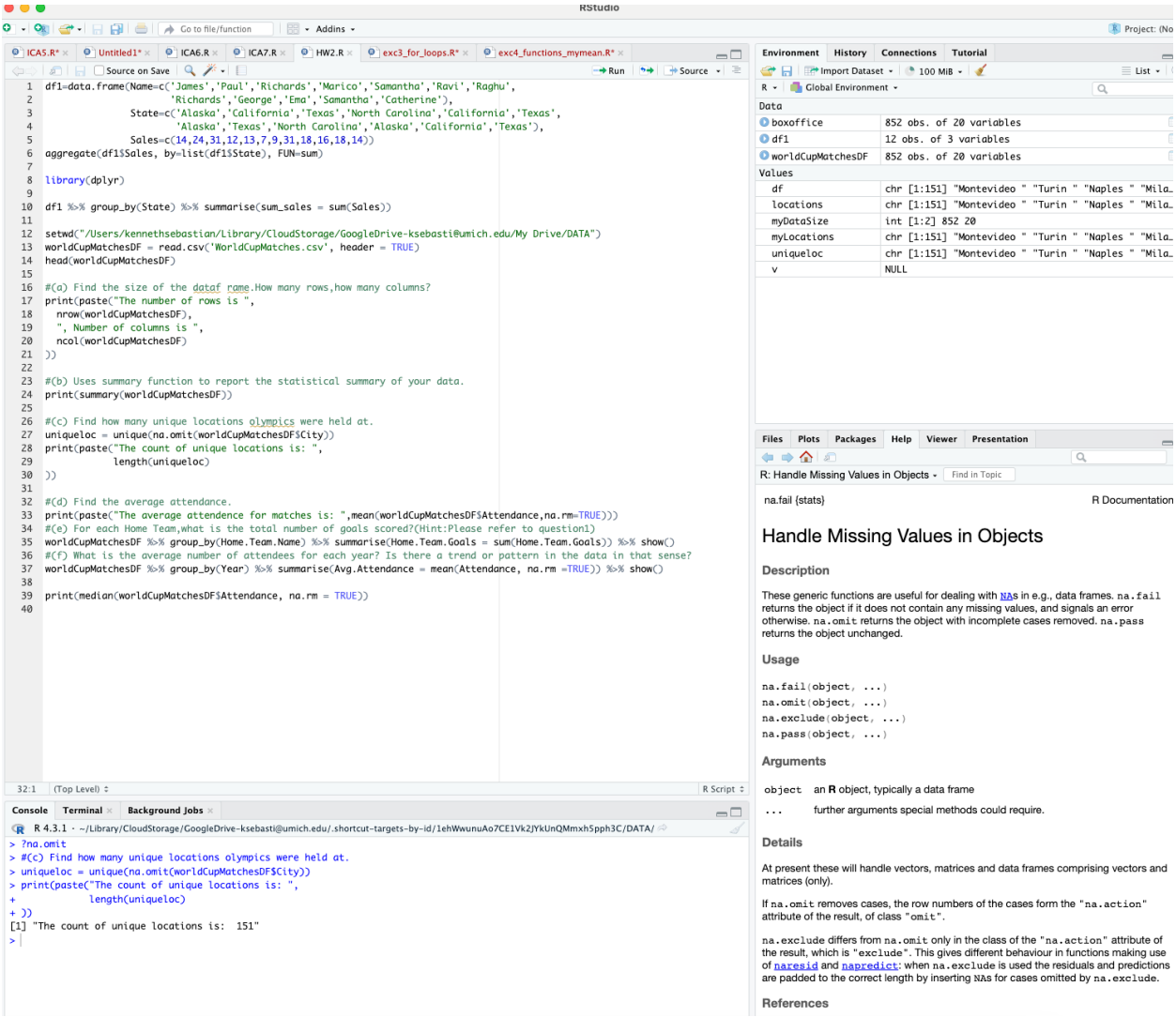
#(a) Find the size of the dataf rame.How many rows,how many columns?
print(paste("The number of rows is ",
 nrow(worldCupMatchesDF),
 ", Number of columns is ",
 ncol(worldCupMatchesDF)
))



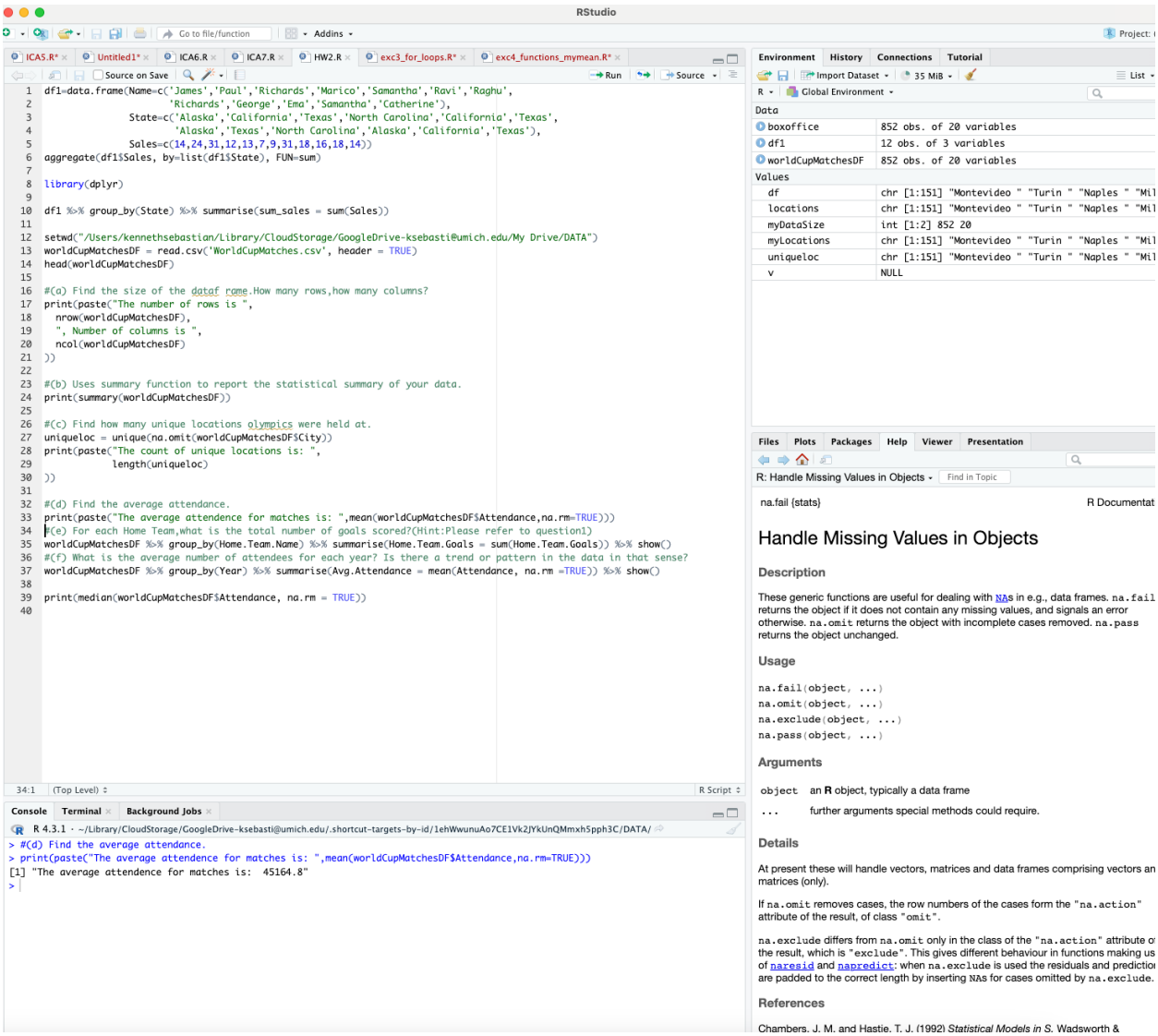
#(b) Uses summary function to report the statistical summary of your data.
print(summary(worldCupMatchesDF))




```
#(c) Find how many unique locations olympics were held at.
uniqueloc = unique(na.omit(worldCupMatchesDF$City))
print(paste("The count of unique locations is: ",
        length(uniqueloc)
))
```

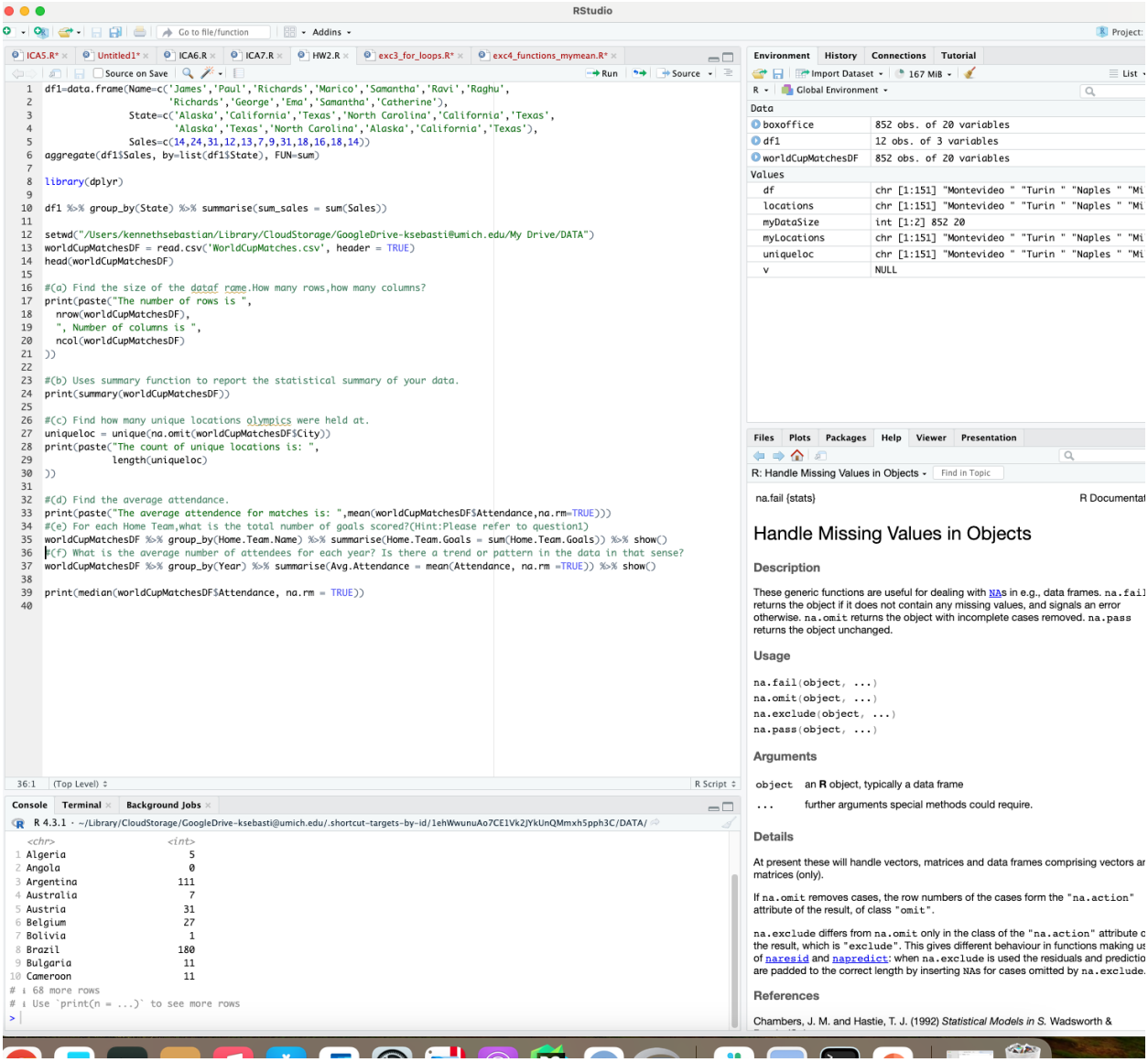


#(d) Find the average attendance.
print(paste("The average attendence for matches is: ",mean(worldCupMatchesDF$Attendance,na.rm=TRUE)))



#(e) For each Home Team,what is the total number of goals scored?(Hint:Please refer to question1)

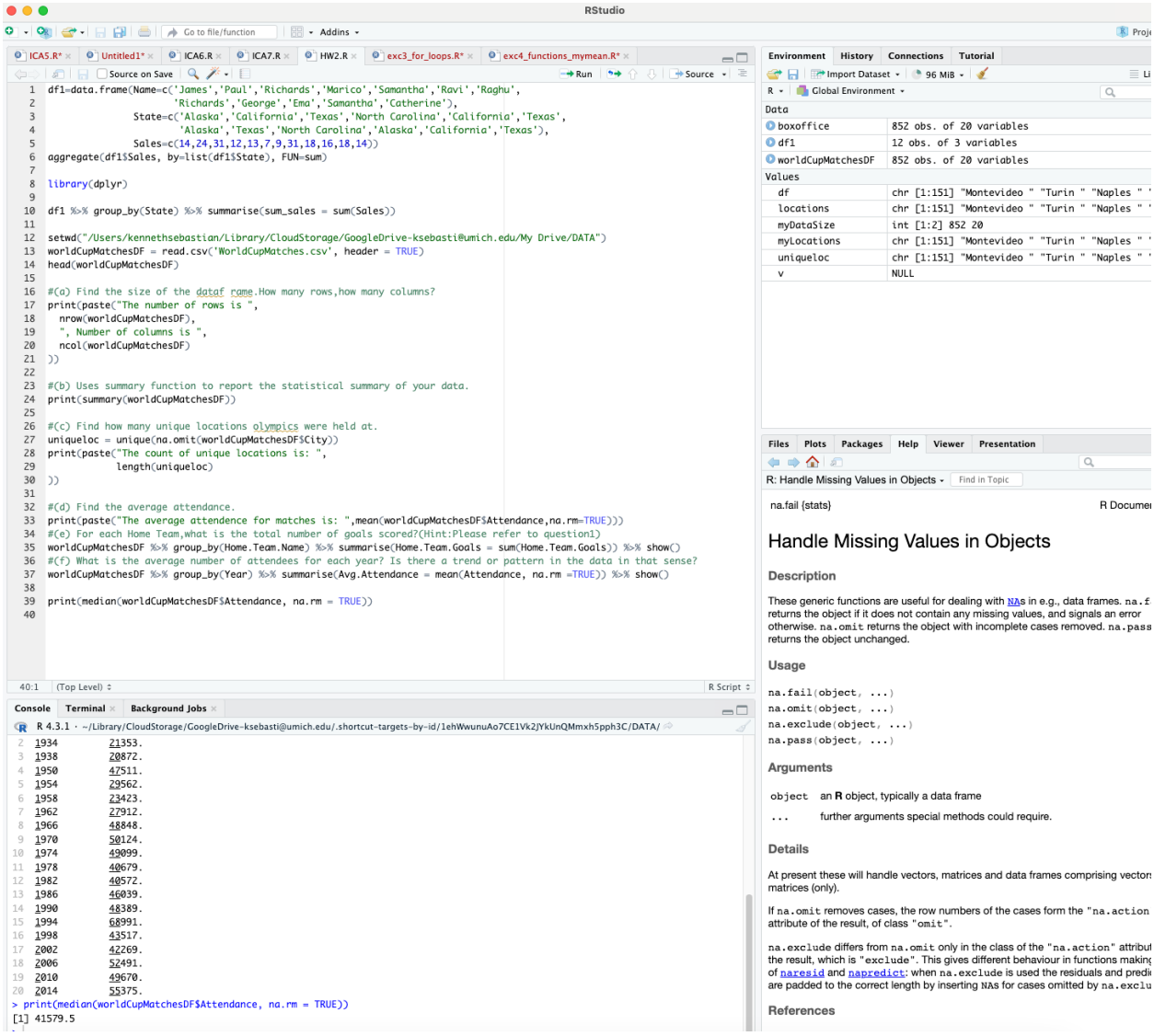
worldCupMatchesDF %>% group_by(Home.Team.Name) %>% summarise(Home.Team.Goals = sum(Home.Team.Goals)) %>% show()


RStudio
ICA5.R*
Untitled1*
ICA6.R
ICA7.R
HW2.R
exc3_for_loops.R*
exc4_functions_mymean.R*
Source on Save
Run
Source
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))
aggregate(df1$Sales, by=list(df1$State), FUN=sum)
library(dplyr)
df1 %>% group_by(State) %>% summarise(sum_sales = sum(Sales))
setwd("/Users/kennethsebastian/Library/CloudStorage/GoogleDrive-ksebasti@umich.edu/My Drive/DATA")
worldCupMatchesDF = read.csv('WorldCupMatches.csv', header = TRUE)
head(worldCupMatchesDF)
#(a) Find the size of the dataf rame.How many rows,how many columns?
print(paste("The number of rows is ",
nrow(worldCupMatchesDF),
", Number of columns is ",
ncol(worldCupMatchesDF)
))
#(b) Uses summary function to report the statistical summary of your data.
print(summary(worldCupMatchesDF))
#(c) Find how many unique locations olympics were held at.
uniqueloc = unique(na.omit(worldCupMatchesDF$City))
print(paste("The count of unique locations is: ",
length(uniqueloc)
))
#(d) Find the average attendance.
print(paste("The average attendence for matches is: ",mean(worldCupMatchesDF$Attendance,na.rm=TRUE)))
#(e) For each Home Team,what is the total number of goals scored?(Hint:Please refer to question1)
worldCupMatchesDF %>% group_by(Home.Team.Name) %>% summarise(Home.Team.Goals = sum(Home.Team.Goals)) %>% show()
#(f) What is the average number of attendees for each year? Is there a trend or pattern in the data in that sense?
worldCupMatchesDF %>% group_by(Year) %>% summarise(Avg.Attendance = mean(Attendance, na.rm =TRUE)) %>% show()
print(median(worldCupMatchesDF$Attendance, na.rm = TRUE))
36:1
(Top Level)
R Script
Console
Terminal
Background Jobs
R 4.3.1 · ~/Library/CloudStorage/GoogleDrive-ksebasti@umich.edu/.shortcut-targets-by-id/1ehWwunuAo7CE1Vk2JYkUnQMmxh5pph3C/DATA/
<chr> <int>
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
ℹ 68 more rows
ℹ Use `print(n = ...)` to see more rows
Environment
History
Connections
Tutorial
Import Dataset
167 MiB
List
R
Global Environment
Data
boxoffice 852 obs. of 20 variables
df1 12 obs. of 3 variables
worldCupMatchesDF 852 obs. of 20 variables
Values
df chr [1:151] "Montevideo " "Turin " "Naples " "Mi
locations chr [1:151] "Montevideo " "Turin " "Naples " "Mi
myDataSize int [1:2] 852 20
myLocations chr [1:151] "Montevideo " "Turin " "Naples " "Mi
uniqueloc chr [1:151] "Montevideo " "Turin " "Naples " "Mi
v NULL
Files
Plots
Packages
Help
Viewer
Presentation
R: Handle Missing Values in Objects
Find in Topic
na.fail {stats}
R Documentat
Handle Missing Values in Objects
Description
These generic functions are useful for dealing with NAs in e.g., data frames. na.fail
returns the object if it does not contain any missing values, and signals an error
otherwise. na.omit returns the object with incomplete cases removed. na.pass
returns the object unchanged.
Usage
na.fail(object, ...)
na.omit(object, ...)
na.exclude(object, ...)
na.pass(object, ...)
Arguments
object an R object, typically a data frame
... further arguments special methods could require.
Details
At present these will handle vectors, matrices and data frames comprising vectors ar
matrices (only).
If na.omit removes cases, the row numbers of the cases form the "na.action"
attribute of the result, of class "omit".
na.exclude differs from na.omit only in the class of the "na.action" attribute o
the result, which is "exclude". This gives different behaviour in functions making us
of naresid and napredict: when na.exclude is used the residuals and predictio
are padded to the correct length by inserting NAs for cases omitted by na.exclude.
References
Chambers, J. M. and Hastie, T. J. (1992) Statistical Models in S. Wadsworth &

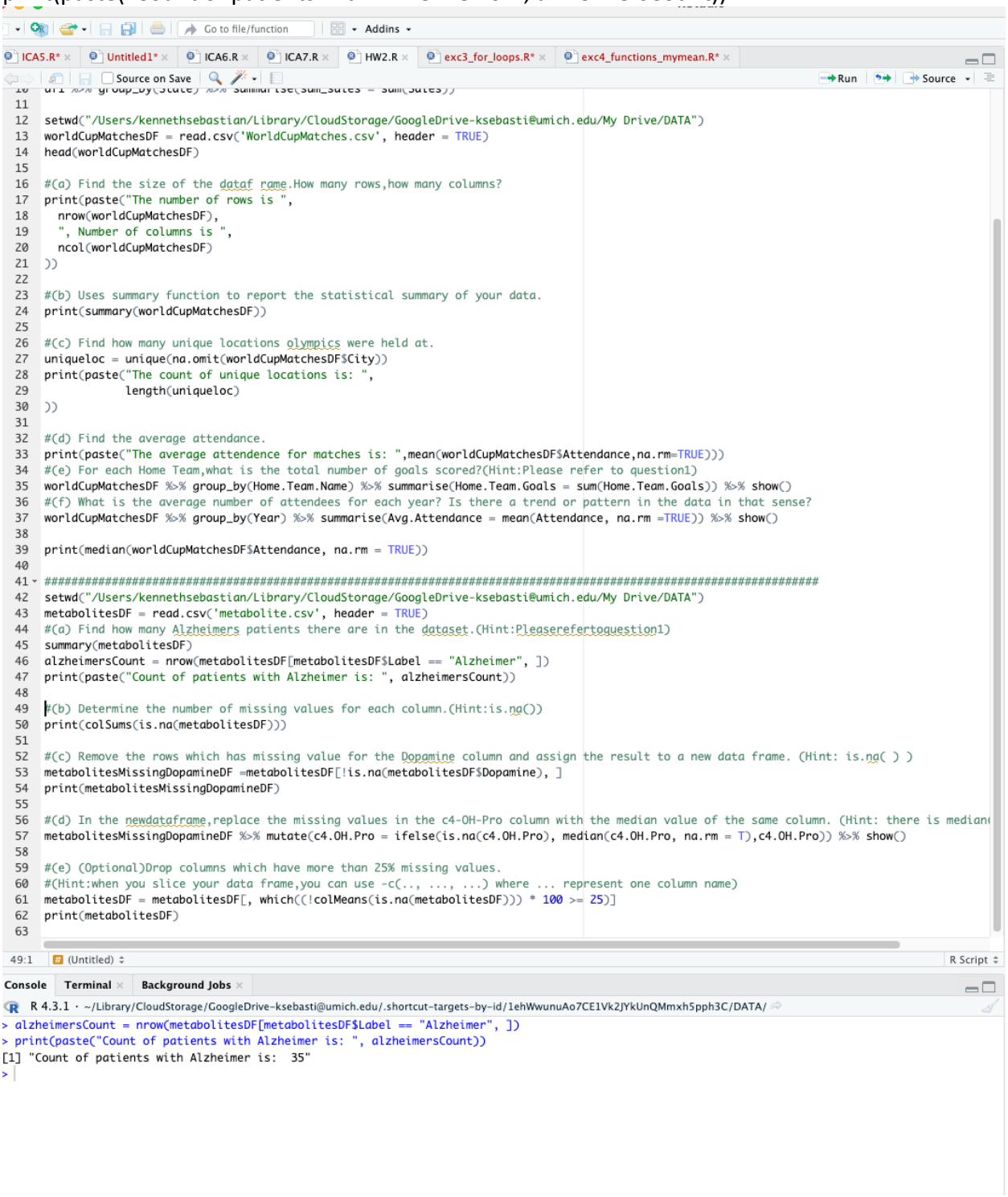
#(f) What is the average number of attendees for each year? Is there a trend or pattern in the data in that sense?
worldCupMatchesDF %>% group_by(Year) %>% summarise(Avg.Attendance = mean(Attendance, na.rm =TRUE)) %>% show()

print(median(worldCupMatchesDF$Attendance, na.rm = TRUE))

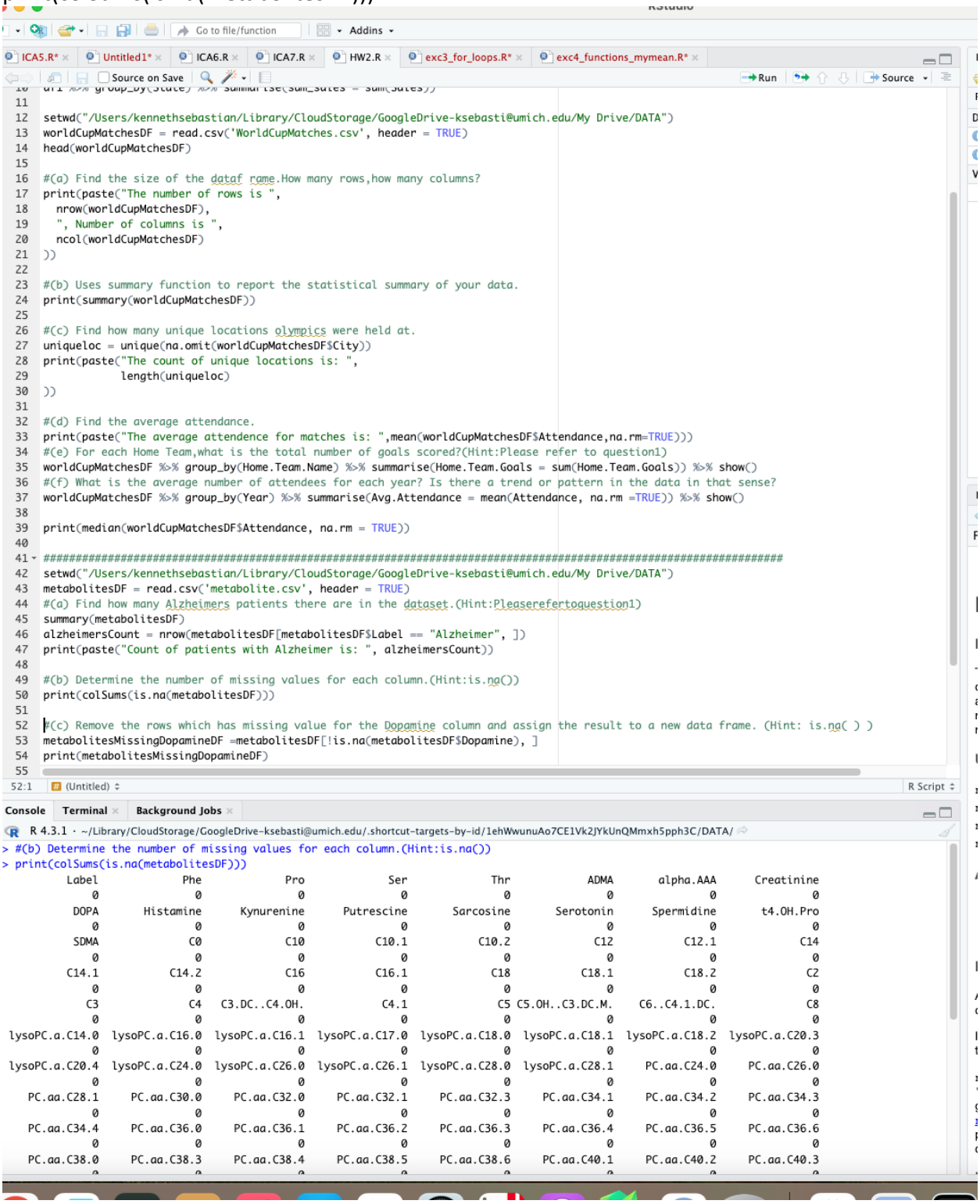
The average attendees seem to have rose from 1934 to 1966 then held close to the median of 41579 and in 2006,2010,2014 has been above the median.



#(a) Find how many Alzheimers patients there are in the dataset.(Hint:Pleaserefertoquestion1)
summary(metabolitesDF)
alzheimersCount = nrow(metabolitesDF[metabolitesDF$Label == "Alzheimer",])
print(paste("Count of patients with Alzheimer is: ", alzheimersCount))



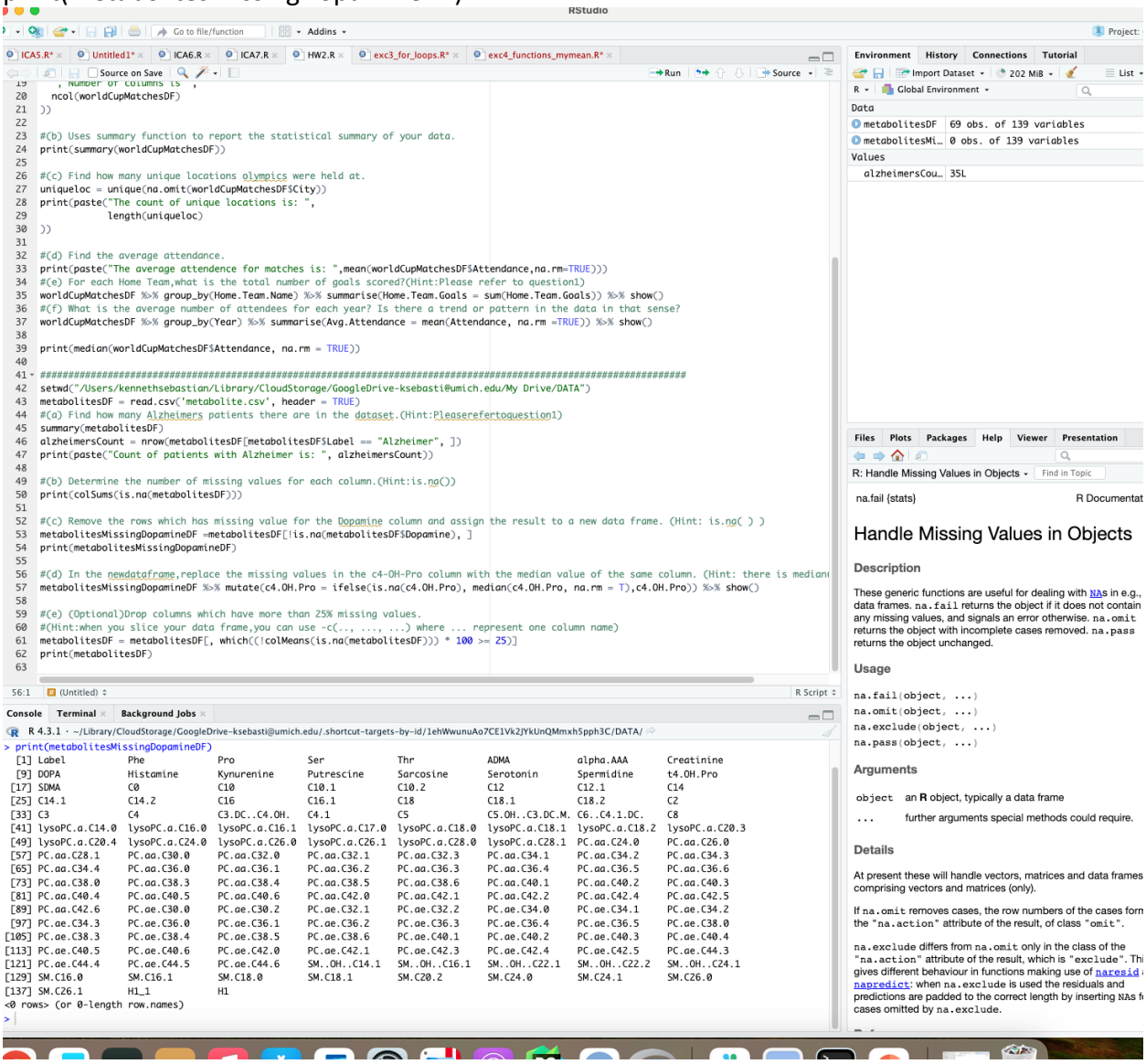
#(b) Determine the number of missing values for each column.(Hint:is.na())
print(colSums(is.na(metabolitesDF)))



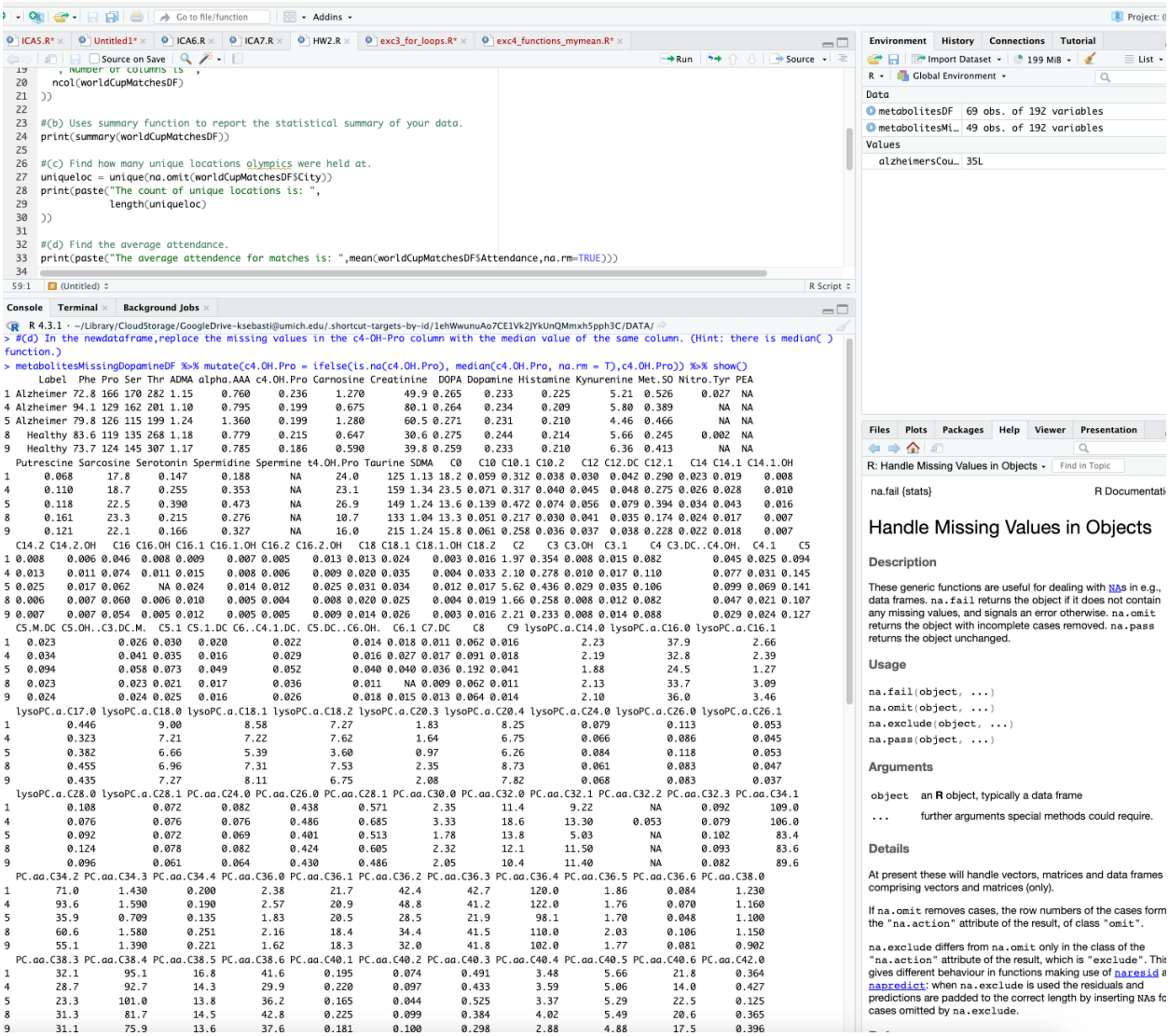
```
setwd("/Users/kennethsebastian/Library/CloudStorage/GoogleDrive-ksebasti@umich.edu/My Drive/DATA")
worldCupMatchesDF = read.csv('WorldCupMatches.csv', header = TRUE)
head(worldCupMatchesDF)

#(a) Find the size of the dataf rame.How many rows,how many columns?
print(paste("The number of rows is ",
  nrow(worldCupMatchesDF),
  ", Number of columns is ",
  ncol(worldCupMatchesDF)
))

#(b) Uses summary function to report the statistical summary of your data.
print(summary(worldCupMatchesDF))

#(c) Find how many unique locations olympics were held at.
uniqueloc = unique(na.omit(worldCupMatchesDF$City))
print(paste("The count of unique locations is: ",
            length(uniqueloc)
))

#(d) Find the average attendance.
print(paste("The average attendence for matches is: ",mean(worldCupMatchesDF$Attendance,na.rm=TRUE)))
#(e) For each Home Team,what is the total number of goals scored?(Hint:Please refer to question1)
worldCupMatchesDF %>% group_by(Home.Team.Name) %>% summarise(Home.Team.Goals = sum(Home.Team.Goals)) %>% show()
#(f) What is the average number of attendees for each year? Is there a trend or pattern in the data in that sense?
worldCupMatchesDF %>% group_by(Year) %>% summarise(Avg.Attendance = mean(Attendance, na.rm =TRUE)) %>% show()

print(median(worldCupMatchesDF$Attendance, na.rm = TRUE))

##############################################################################################################
setwd("/Users/kennethsebastian/Library/CloudStorage/GoogleDrive-ksebasti@umich.edu/My Drive/DATA")
metabolitesDF = read.csv('metabolite.csv', header = TRUE)
#(a) Find how many Alzheimers patients there are in the dataset.(Hint:Pleaserefertoquestion1)
summary(metabolitesDF)
alzheimersCount = nrow(metabolitesDF[metabolitesDF$Label == "Alzheimer", ])
print(paste("Count of patients with Alzheimer is: ", alzheimersCount))

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

#(c) Remove the rows which has missing value for the Dopamine column and assign the result to a new data frame. (Hint: is.na( ) )
metabolitesMissingDopamineDF =metabolitesDF[!is.na(metabolitesDF$Dopamine), ]
print(metabolitesMissingDopamineDF)
```

```
> #(b) Determine the number of missing values for each column.(Hint:is.na())
> print(colSums(is.na(metabolitesDF)))
         Label            Phe            Pro            Ser            Thr           ADMA      alpha.AAA     Creatinine
             0              0              0              0              0              0              0              0
          DOPA      Histamine     Kynurenine     Putrescine      Sarcosine      Serotonin     Spermidine      t4.OH.Pro
             0              0              0              0              0              0              0              0
          SDMA            C0            C10           C10.1          C10.2            C12          C12.1            C14
             0              0              0              0              0              0              0              0
        C14.1          C14.2            C16          C16.1            C18          C18.1          C18.2             C2
             0              0              0              0              0              0              0              0
            C3             C4   C3.DC..C4.OH.           C4.1             C5 C5.OH..C3.DC.M.    C6..C4.1.DC.             C8
             0              0              0              0              0              0              0              0
 lysoPC.a.C14.0 lysoPC.a.C16.0 lysoPC.a.C16.1 lysoPC.a.C17.0 lysoPC.a.C18.0 lysoPC.a.C18.1 lysoPC.a.C18.2 lysoPC.a.C20.3
             0              0              0              0              0              0              0              0
 lysoPC.a.C20.4 lysoPC.a.C24.0 lysoPC.a.C26.0 lysoPC.a.C26.1 lysoPC.a.C28.0 lysoPC.a.C28.1     PC.aa.C24.0     PC.aa.C26.0
             0              0              0              0              0              0              0              0
    PC.aa.C28.1     PC.aa.C30.0     PC.aa.C32.0     PC.aa.C32.1     PC.aa.C32.3     PC.aa.C34.1     PC.aa.C34.2     PC.aa.C34.3
             0              0              0              0              0              0              0              0
    PC.aa.C34.4     PC.aa.C36.0     PC.aa.C36.1     PC.aa.C36.2     PC.aa.C36.3     PC.aa.C36.4     PC.aa.C36.5     PC.aa.C36.6
             0              0              0              0              0              0              0              0
    PC.aa.C38.0     PC.aa.C38.3     PC.aa.C38.4     PC.aa.C38.5     PC.aa.C38.6     PC.aa.C40.1     PC.aa.C40.2     PC.aa.C40.3
```

#(c) Remove the rows which has missing value for the Dopamine column and assign the result to a new data frame. (Hint: is.na( ))
metabolitesMissingDopamineDF =metabolitesDF[!is.na(metabolitesDF$Dopamine),]
print(metabolitesMissingDopamineDF)



#(d) In the newdataframe,replace the missing values in the c4-OH-Pro column with the median value of the same column. (Hint: there is median( ) function.)
metabolitesMissingDopamineDF %>% mutate(c4.OH.Pro = ifelse(is.na(c4.OH.Pro), median(c4.OH.Pro, na.rm = T),c4.OH.Pro)) %>% show()



#(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)

```
metabolitesDF = metabolitesDF[, which((!colMeans(is.na(metabolitesDF))) * 100 >= 25)]
print(metabolitesDF)
```

RStudio

ICA5.R* | Untitled1* | ICA6.R | ICA7.R | HW2.R | exc3_for_loops.R* | exc4_functions_mymean.R*

```
14 head(worldCupMatchesDF)
15
16 #(a) Find the size of the dataf rame.How many rows,how many columns?
17 print(paste("The number of rows is ",
18   nrow(worldCupMatchesDF),
19   ", Number of columns is ",
20   ncol(worldCupMatchesDF)
21 ))
22
23 #(b) Uses summary function to report the statistical summary of your data.
24 print(summary(worldCupMatchesDF))
25
26 #(c) Find how many unique locations olympics were held at.
27 uniqueloc = unique(na.omit(worldCupMatchesDF$City))
28 print(paste("The count of unique locations is: ",
29             length(uniqueloc)
30
```

63:1 (Untitled) R Script

Console | Terminal | Background Jobs

R 4.3.1 · ~/Library/CloudStorage/GoogleDrive-ksebasti@umich.edu/.shortcut-targets-by-id/1ehWwunuAo7CE1Vk2JYkUnQMmxh5pph3C/DATA/

```
> #(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)
> metabolitesDF = metabolitesDF[, which((!colMeans(is.na(metabolitesDF))) * 100 >= 25)]
> print(metabolitesDF)
     Label  Phe   Pro Ser Thr  ADMA alpha.AAA Creatinine  DOPA Histamine Kynurenine Putrescine Sarcosine Serotonin Spermidine t4.OH.Pro SDMA
1 Alzheimer 72.8 166.0 170 282 1.150     0.760       49.9 0.265     0.225       5.21      0.068      17.8     0.147      0.188       24.0 1.13
2 Alzheimer 93.4 138.0 142 217 1.050     0.929       48.8 0.252     0.211       5.44      0.087      20.2     0.231      0.233       29.3 1.65
3 Alzheimer 68.6 161.0 158 208 1.000     0.620       30.4 0.268     0.217       5.20      0.260      14.4     0.196      0.384       20.9 1.57
4 Alzheimer 94.1 129.0 162 201 1.100     0.795       80.1 0.264     0.209       5.80      0.110      18.7     0.255      0.353       23.1 1.34
5 Alzheimer 79.8 126.0 115 199 1.240     1.360       60.5 0.271     0.210       4.46      0.118      22.5     0.390      0.473       26.9 1.24
6 Alzheimer 82.5 167.0 173 333 1.350     1.150       24.0 0.275     0.212       7.01      0.262      30.8     0.140      0.856       26.0 1.44
7   Healthy 69.7  95.6 143 244 0.991     0.927       41.6 0.260     0.211       6.18      0.176      16.3     0.162      0.060       15.7 1.32
    C0   C10 C10.1 C10.2   C12 C12.1   C14 C14.1 C14.2   C16 C16.1   C18 C18.1 C18.2   C2    C3    C4 C3.DC..C4.OH.  C4.1    C5
1 18.2 0.059 0.312 0.038 0.030 0.290 0.023 0.019 0.008 0.046 0.009 0.013 0.024 0.016 1.97 0.354 0.082         0.045 0.025 0.094
2 17.0 0.051 0.288 0.039 0.038 0.265 0.026 0.017 0.009 0.070 0.013 0.014 0.025 0.028 1.95 0.184 0.108         0.080 0.025 0.077
3 12.6 0.083 0.357 0.054 0.032 0.302 0.021 0.031 0.010 0.076 0.019 0.016 0.025 0.018 1.70 0.371 0.057         0.035 0.039 0.096
4 23.5 0.071 0.317 0.040 0.045 0.275 0.026 0.028 0.013 0.074 0.015 0.020 0.035 0.033 2.10 0.278 0.110         0.077 0.031 0.145
5 13.6 0.139 0.472 0.074 0.056 0.394 0.034 0.043 0.025 0.062 0.024 0.031 0.034 0.017 5.62 0.436 0.106         0.099 0.069 0.141
6 26.7 0.058 0.238 0.042 0.039 0.196 0.029 0.023 0.010 0.081 0.012 0.017 0.035 0.029 3.49 0.461 0.123         0.068 0.026 0.090
7 12.9 0.063 0.247 0.041 0.037 0.218 0.025 0.029 0.011 0.057 0.013 0.019 0.037 0.018 2.17 0.253 0.068         0.066 0.014 0.077
  C5.OH..C3.DC.M. C6..C4.1.DC.    C8 lysoPC.a.C14.0 lysoPC.a.C16.0 lysoPC.a.C16.1 lysoPC.a.C17.0 lysoPC.a.C18.0 lysoPC.a.C18.1 lysoPC.a.C18.2
1           0.026        0.022 0.062          2.23          37.9          2.66         0.446          9.00          8.58          7.27
2           0.026        0.030 0.058          1.97          22.1          1.31         0.270          5.35          3.94          4.42
3           0.024        0.022 0.090          2.12          33.7          2.53         0.399          7.51          7.73          8.02
4           0.041        0.029 0.091          2.19          32.8          2.39         0.323          7.21          7.22          7.62
5           0.058        0.052 0.192          1.88          24.5          1.27         0.382          6.66          5.39          3.60
6           0.037        0.063 0.073          2.11          29.1          2.09         0.348          5.84          6.30          8.10
7           0.022        0.029 0.056          2.32          42.4          3.16         0.437          9.63          9.44         10.90
  lysoPC.a.C20.3 lysoPC.a.C20.4 lysoPC.a.C24.0 lysoPC.a.C26.0 lysoPC.a.C26.1 lysoPC.a.C28.0 lysoPC.a.C28.1 PC.aa.C24.0 PC.aa.C26.0
1          1.830          8.25         0.079         0.113         0.053         0.108         0.072       0.082       0.438
2          0.958          4.60         0.059         0.066         0.042         0.076         0.058       0.065       0.409
3          2.050          9.84         0.075         0.126         0.049         0.078         0.092       0.099       0.458
4          1.640          6.75         0.066         0.086         0.045         0.076         0.076       0.076       0.486
5          0.970          6.26         0.084         0.118         0.053         0.092         0.072       0.069       0.401
6          1.970          7.04         0.083         0.112         0.050         0.099         0.083       0.073       0.450
7          2.540         10.80         0.069         0.095         0.049         0.107         0.088       0.074       0.424
  PC.aa.C28.1 PC.aa.C30.0 PC.aa.C32.0 PC.aa.C32.1 PC.aa.C32.3 PC.aa.C34.1 PC.aa.C34.2 PC.aa.C34.3 PC.aa.C34.4 PC.aa.C36.0 PC.aa.C36.1
1       0.571        2.35        11.4        9.22       0.092       109.0        71.0       1.430       0.200        2.38        21.7
2       0.521        1.99        12.7        5.40       0.067        64.2        60.5       0.879       0.127        2.05        14.3
3       0.605        2.69        16.6       11.60       0.105       108.0        83.1       1.930       0.210        2.30        19.9
4       0.685        3.33        18.6       13.30       0.079       106.0        93.6       1.590       0.190        2.57        20.9
5       0.513        1.78        13.8        5.03       0.102        83.4        35.9       0.709       0.135        1.83        20.5
6       0.620        2.61        14.7        8.98       0.107        90.2        85.6       1.790       0.213        2.48        15.5
7       0.788        2.42        12.4       10.40       0.121       111.0        92.7       2.040       0.315        2.22        21.3
  PC.aa.C36.2 PC.aa.C36.3 PC.aa.C36.4 PC.aa.C36.5 PC.aa.C36.6 PC.aa.C38.0 PC.aa.C38.3 PC.aa.C38.4 PC.aa.C38.5 PC.aa.C38.6 PC.aa.C40.1
1        42.4        42.7       120.0        1.86       0.084       1.230        32.1        95.1       16.80        41.6       0.195
2        35.6        24.3        83.7        1.05       0.046       0.946        21.9        78.9        9.91        25.1       0.211
3        44.9        43.9       146.0        2.09       0.057       1.210        34.5       107.0       17.50        36.6       0.212
4        48.8        41.2       122.0        1.76       0.070       1.160        28.7        92.7       14.30        29.9       0.220
5        28.5        21.9        98.1        1.70       0.048       1.100        23.3       101.0       13.80        36.2       0.165
```