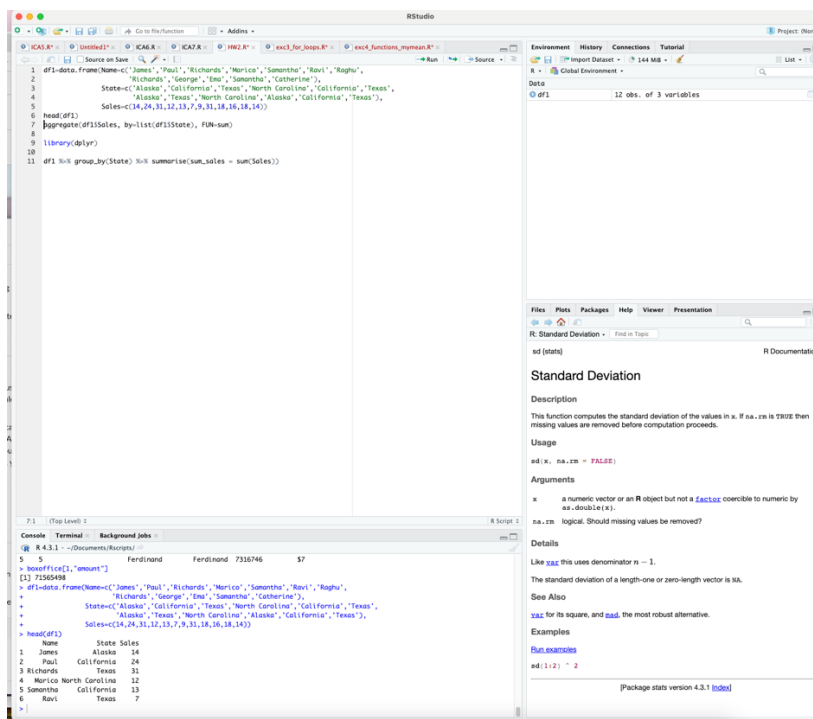


Creates a data frame df 1 with columns Name,State, Sales and 6 rows.



Computes the summary of sales by states

RStudio interface showing a script and its execution results.

```
1 df1=data.frame(Name=c('James','Paul','Richards','Marico','Samantha','Ravi','Raghu',
2                      'Richards','George','Ema','Samantha','Catherine'),
3                  State=c('Alaska','California','Texas','North Carolina','California','Texas',
4                          'Alaska','Texas','North Carolina','Alaska','California','Texas'),
5                      Sales=c(14,24,31,12,13,7,9,31,18,16,18,14))
6 aggregate(df1$Sales, by=list(df1$State), FUN=sum)
7
8 library(dplyr)
9
10 df1 %>% group_by(State) %>% summarise(sum_sales = sum(Sales))
11
12 setwd("~/Users/kennethsebastian/Documents/Rscripts")
13 boxoffice = read.csv("boxoffice.csv", header = TRUE)
```

Environment: R, Global Environment. Data: df1 (12 obs. of 3 variables).

Files | Plots | Packages | Help | Viewer | Presentation

R: dplyr: A Grammar of Data Manipulation

dplyr-package (dplyr) R Documentation

dplyr: A Grammar of Data Manipulation

Description

To learn more about dplyr, start with the vignettes: `browseVignettes(package = "dplyr")`

Author(s)

Maintainer: Hadley Wickham hadley@posit.co ([ORCID](#))

Authors:

- Romain François ([ORCID](#))
- Lionel Henry
- Kirill Müller ([ORCID](#))
- Davis Vaughan davis@posit.co ([ORCID](#))

Other contributors:

- Posit Software, PBC [copyright holder, funder]

See Also

Useful links:

- <https://dplyr.tidyverse.org>
- <https://github.com/tidyverse/dplyr>
- Report bugs at <https://github.com/tidyverse/dplyr/issues>

[Package dplyr version 1.1.3 [Index](#)]

Console | **Terminal** | **Background Jobs**

```
R 4.3.1 ~ /Documents/Rscripts/
> 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
> library(dplyr)
> df1 %>% group_by(State) %>% summarise(sum_sales = sum(Sales))
# A tibble: 4 x 2
  State      sum_sales
<chr>      <dbl>
1 Alaska         39
2 California      55
3 North Carolina  30
4 Texas          83
> ?dplyr
>
```

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

The screenshot displays the RStudio interface with the following components:

- Source Editor:** Contains R code for creating a data frame, aggregating sales by state, and summarizing the results using dplyr.
- Environment:** Shows the data frame 'df1' with 12 observations and 3 variables.
- Console:** Displays the execution of the code, showing the aggregated data and the summary output.
- Documentation Panel:** Shows the dplyr package documentation, including a description, authors, and useful links.

```
1 df1<-data.frame(Name=c('James','Paul','Richards','Marico','Samantha','Ravi','Raghu',  
2 'Richards','George','Ema','Samantha','Catherine'),  
3 State=c('Alaska','California','Texas','North Carolina','California','Texas',  
4 'Alaska','Texas','North Carolina','Alaska','California','Texas'),  
5 Sales=c(14,24,31,12,13,7,9,31,18,16,18,14))  
6 aggregate(df1$Sales, by=list(df1$State), FUN=sum)  
7  
8 library(dplyr)  
9  
10 df1 %>% group_by(State) %>% summarise(sum_sales = sum(Sales))  
11  
12 setwd("~/Users/kennethsebastian/Documents/Rscripts")  
13 boxoffice = read.csv("boxoffice.csv", header = TRUE)
```

Console Output:

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

dplyr: A Grammar of Data Manipulation

Description

To learn more about dplyr, start with the vignettes: `browseVignettes(package = "dplyr")`

Author(s)

Maintainer: Hadley Wickham hadley@posit.co ([ORCID](#))

Authors:

- Romain François ([ORCID](#))
- Lionel Henry
- Kirill Müller ([ORCID](#))
- Davis Vaughan davis@posit.co ([ORCID](#))

Other contributors:

- Posit Software, PBC [copyright holder, funder]

See Also

Useful links:

- <https://dplyr.tidyverse.org>
- <https://github.com/tidyverse/dplyr>
- Report bugs at <https://github.com/tidyverse/dplyr/issues>

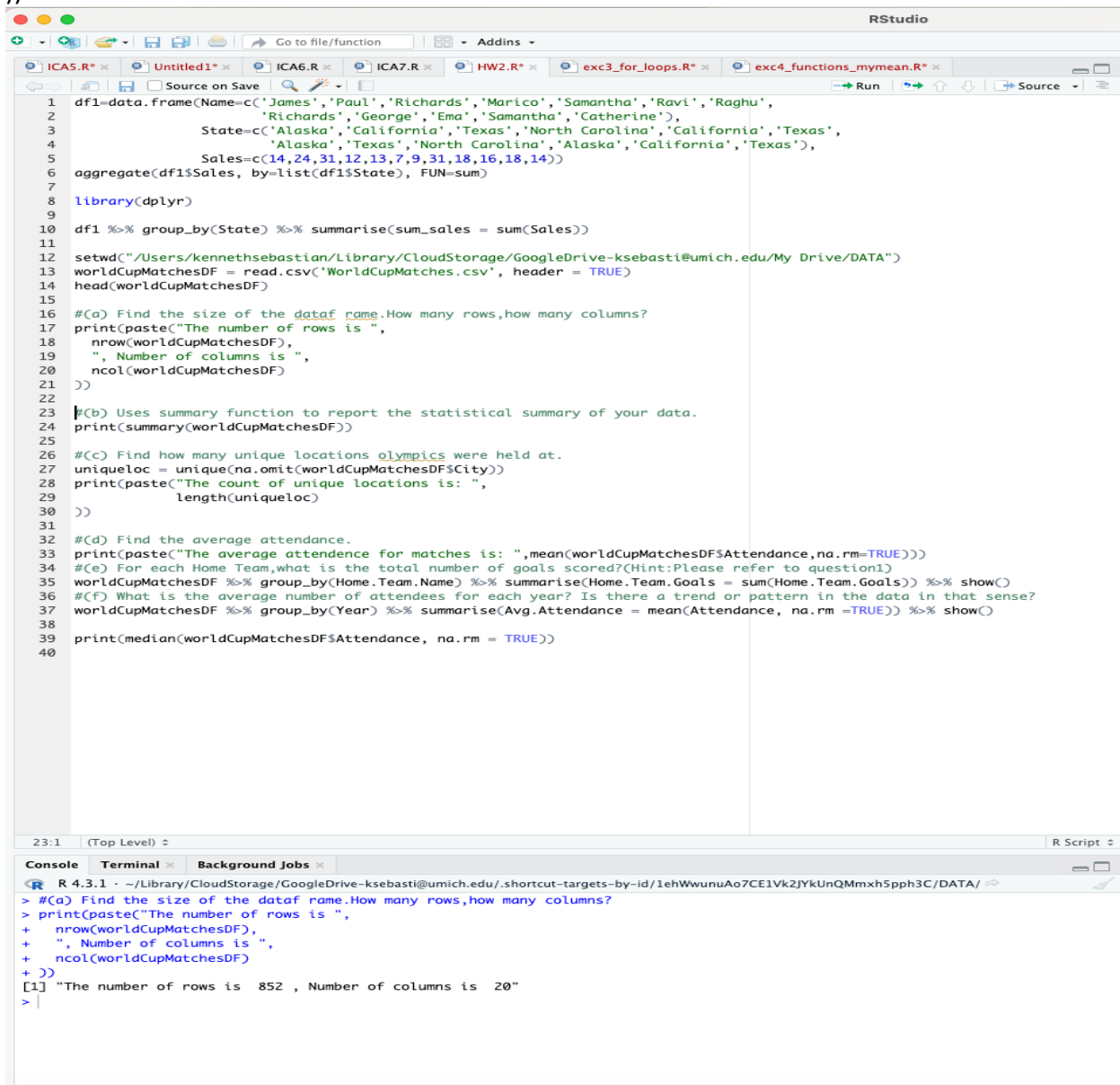
[Package dplyr version 1.1.3 [Index](#)]

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 data frame. How many rows, how many columns?

```
print(paste("The number of rows is ",
  nrow(worldCupMatchesDF),
  ", Number of columns is ",
  ncol(worldCupMatchesDF)
))
```



The screenshot shows the RStudio interface with a script editor and a console. The script editor contains R code for data manipulation and analysis. The console shows the output of the first part of the code, which prints the number of rows and columns of the 'worldCupMatchesDF' data frame.

```
1 df1=data.frame(Name=c('James','Paul','Richards','Marico','Samantha','Ravi','Raghu',
2   'Richards','George','Ema','Samantha','Catherine'),
3   State=c('Alaska','California','Texas','North Carolina','California','Texas',
4   'Alaska','Texas','North Carolina','Alaska','California','Texas'),
5   Sales=c(14,24,31,12,13,7,9,31,18,16,18,14))
6 aggregate(df1$Sales, by=list(df1$State), FUN=sum)
7
8 library(dplyr)
9
10 df1 %>% group_by(State) %>% summarise(sum_sales = sum(Sales))
11
12 setwd("~/Users/kennethsebastian/Library/CloudStorage/GoogleDrive-ksebasti@umich.edu/My Drive/DATA")
13 worldCupMatchesDF = read.csv("WorldCupMatches.csv", header = TRUE)
14 head(worldCupMatchesDF)
15
16 #(a) Find the size of the data frame. 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$City))
28 print(paste("The count of unique locations is: ",
29   length(uniqueLoc)
30 ))
31
32 #(d) Find the average attendance.
33 print(paste("The average attendance for matches is: ", mean(worldCupMatchesDF$Attendance, na.rm=TRUE)))
34 #(e) For each Home Team, what is the total number of goals scored? (Hint: Please refer to question 1)
35 worldCupMatchesDF %>% group_by(Home.Team.Name) %>% summarise(Home.Team.Goals = sum(Home.Team.Goals)) %>% show()
36 #(f) What is the average number of attendees for each year? Is there a trend or pattern in the data in that sense?
37 worldCupMatchesDF %>% group_by(Year) %>% summarise(Avg.Attendance = mean(Attendance, na.rm=TRUE)) %>% show()
38
39 print(median(worldCupMatchesDF$Attendance, na.rm = TRUE))
40
```

Console output:

```
> #(a) Find the size of the data frame. How many rows, how many columns?
> print(paste("The number of rows is ",
+   nrow(worldCupMatchesDF),
+   ", Number of columns is ",
+   ncol(worldCupMatchesDF)
+ ))
[1] "The number of rows is 852 , Number of columns is 20"
```

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

```
print(summary(worldCupMatchesDF))
```

RStudio

Project: (None)

```

1 df1=data.frame(Name=c('James','Paul','Richards','Marico','Samantha','Ravi','Raghu',
2                       'Richards','George','Ema','Samantha','Catherine'),
3                       State=c('Alaska','California','Texas','North Carolina','California','Texas',
4                               'Alaska','Texas','North Carolina','Alaska','California','Texas'),
5                               Sales=c(14,24,31,12,13,7,9,31,18,16,18,14))
6 aggregate(df1$Sales, by=list(df1$State), FUN=sum)
7
8 library(dplyr)
9
10 df1 %>% group_by(State) %>% summarise(sum_sales = sum(Sales))
11
12 setwd("/Users/kennethsebastian/Library/CloudStorage/GoogleDrive-ksebasti@umich.edu/My Drive/DATA")
13 worldCupMatchesDF = read.csv("WorldCupMatches.csv", header = TRUE)
14 head(worldCupMatchesDF)
15
16 # (a) Find the size of the data frame. 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$City))
28 print(paste("The count of unique locations is: ",
29             length(uniqueLoc)
30             ))
31
32 # (d) Find the average attendance.
33 print(paste("The average attendance for matches is: ", mean(worldCupMatchesDF$Attendance, na.rm=TRUE)))
34 # (e) For each Home Team, what is the total number of goals scored? (Hint: Please refer to question 1)
35 worldCupMatchesDF %>% group_by(Home.Team.Name) %>% summarise(Home.Team.Goals = sum(Home.Team.Goals)) %>% show()
36 # (f) What is the average number of attendees for each year? Is there a trend or pattern in the data in that sense?
37 worldCupMatchesDF %>% group_by(Year) %>% summarise(Avg.Attendance = mean(Attendance, na.rm=TRUE)) %>% show()
38
39 print(median(worldCupMatchesDF$Attendance, na.rm = TRUE))
40

```

26.1 (Top Level) R Script

Console

```

R 4.3.1 - ~/Library/CloudStorage/GoogleDrive-ksebasti@umich.edu/shortcut-targets-by-id/1ehWwunuAo7CE1V6z/YkUnQMmxh5p3C/DATA/
Median :0.0000 Mode :character Mode :character Mode :character Median : 337 Median : 2191
Mean :0.4284 Mean :10661773 Mean : 61346868
3rd Qu.:1.0000 3rd Qu.: 249722 3rd Qu.: 43950059
Max. :5.0000 Max. :97410600 Max. :300186515

Home.Team.Initials Away.Team.Initials
Length:852 Length:852
Class:character Class:character
Mode :character Mode :character

```

Environment

Global Environment

Data

Object	Class	Size
boxoffice	data.frame	852 obs. of 20 variables
df1	data.frame	12 obs. of 3 variables
worldCupMatchesDF	data.frame	852 obs. of 20 variables

Values

Object	Class	Values
df	chr	[1:151] "Montevideo" "Turin" "Naples" "Mila..."
locations	chr	[1:151] "Montevideo" "Turin" "Naples" "Mila..."
myDataSize	int	[1:2] 852 20
myLocations	chr	[1:151] "Montevideo" "Turin" "Naples" "Mila..."
uniqueLoc	chr	[1:151] "Montevideo" "Turin" "Naples" "Mila..."
v	NULL	

Files Plots Packages Help Viewer Presentation

R: Adjust for Missing Values Find in Topic

naprint (stats) R Documentation

Adjust for Missing Values

Description

Use missing value information to report the effects of an `na.action`.

Usage

```
naprint(x, ...)
```

Arguments

`x` An object produced by an `na.action` function.
`...` further arguments passed to or from other methods.

Details

This is a generic function, and the exact information differs by method. `naprint.omit` reports the number of rows omitted: `naprint.default` reports an empty string.

Value

A character string providing information on missing values, for example the number.

[Package stats version 4.3.1 [Index](#)]

```

#(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)
            ))

```


#(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()
```

Handle Missing Values in Objects

#(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.

The screenshot displays the RStudio interface with the following components:

- Source Editor:** Contains R code for data manipulation and analysis.
- Environment:** Lists loaded data frames: `boxoffice` (852 obs. of 20 variables), `df1` (12 obs. of 3 variables), and `worldCupMatchesDF` (852 obs. of 20 variables).
- Console:** Shows the execution of the code, including the median calculation result: `[1] 41579.5`.

```
1 df1=data.frame(Name=c('James','Paul','Richards','Marico','Samantha','Ravi','Raghu',  
2 'Richards','George','Ema','Samantha','Catherine'),  
3 State=c('Alaska','California','Texas','North Carolina','California','Texas',  
4 'Alaska','Texas','North Carolina','Alaska','California','Texas'),  
5 Sales=c(14,24,31,12,13,7,9,31,18,16,18,14))  
6 aggregate(df1$Sales, by=list(df1$State), FUN=sum)  
7  
8 library(dplyr)  
9  
10 df1 %>% group_by(State) %>% summarise(sum_sales = sum(Sales))  
11  
12 setwd("~/Users/kennethsebastian/Library/CloudStorage/GoogleDrive-ksebasti@umich.edu/My Drive/DATA")  
13 worldCupMatchesDF = read.csv("WorldCupMatches.csv", header = TRUE)  
14 head(worldCupMatchesDF)  
15  
16 #(a) Find the size of the data frame. 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$City))  
28 print(paste("The count of unique locations is: ",  
29 length(uniqueLoc))  
30 )  
31  
32 #(d) Find the average attendance.  
33 print(paste("The average attendance for matches is: ", mean(worldCupMatchesDF$Attendance, na.rm = TRUE)))  
34 #(e) For each Home Team, what is the total number of goals scored? (Hint: Please refer to question 3)  
35 worldCupMatchesDF %>% group_by(Home.Team.Name) %>% summarise(Home.Team.Goals = sum(Home.Team.Goals)) %>% show()  
36 #(f) What is the average number of attendees for each year? Is there a trend or pattern in the data in that sense?  
37 worldCupMatchesDF %>% group_by(Year) %>% summarise(Avg.Attendance = mean(Attendance, na.rm = TRUE)) %>% show()  
38  
39 print(median(worldCupMatchesDF$Attendance, na.rm = TRUE))  
40
```

Environment

Object	Class	Attributes
boxoffice	data.frame	852 obs. of 20 variables
df1	data.frame	12 obs. of 3 variables
worldCupMatchesDF	data.frame	852 obs. of 20 variables

Values

Variable	Class	Range
df	chr	[1:151] "Montevideo" "Turin" "Naples" "
locations	chr	[1:151] "Montevideo" "Turin" "Naples" "
myDataSize	int	[1:2] 852 20
myLocations	chr	[1:151] "Montevideo" "Turin" "Naples" "
uniqueLoc	chr	[1:151] "Montevideo" "Turin" "Naples" "
v	NULL	

Files | **Plots** | **Packages** | **Help** | **Viewer** | **Presentation**

R: Handle Missing Values in Objects - Find in Topic

na.fail (stats) R Documen

Handle Missing Values in Objects

Description

These generic functions are useful for dealing with `NA`s in e.g., data frames. `na.f` 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 vector 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 of the result, which is "exclude". This gives different behaviour in functions making of `par resid` and `na predict`: when `na.exclude` is used the residuals and `predi` are padded to the correct length by inserting `NA`s for cases omitted by `na.exclude`.

References

```
#(a) Find how many Alzheimers patients there are in the dataset.(Hint:Please refertoquestion1)
summary(metabolitesDF)
alzheimersCount = nrow(metabolitesDF[metabolitesDF$Label == "Alzheimer", ])
print(paste("Count of patients with Alzheimer is: ", alzheimersCount))
```

```
10 #1 group_by(State) summarise(Sum_Sales = sum(Sales))
11
12 setwd("~/Users/kennethsebastian/Library/CloudStorage/GoogleDrive-ksebasti@umich.edu/My Drive/DATA")
13 worldCupMatchesDF = read.csv('WorldCupMatches.csv', header = TRUE)
14 head(worldCupMatchesDF)
15
16 #(a) Find the size of the data frame. 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$City))
28 print(paste("The count of unique locations is: ",
29   length(uniqueLoc)
30 ))
31
32 #(d) Find the average attendance.
33 print(paste("The average attendance for matches is: ", mean(worldCupMatchesDF$Attendance, na.rm=TRUE)))
34 #(e) For each Home Team, what is the total number of goals scored? (Hint: Please refer to question 1)
35 worldCupMatchesDF %>% group_by(Home.Team.Name) %>% summarise(Home.Team.Goals = sum(Home.Team.Goals)) %>% show()
36 #(f) What is the average number of attendees for each year? Is there a trend or pattern in the data in that sense?
37 worldCupMatchesDF %>% group_by(Year) %>% summarise(Avg.Attendance = mean(Attendance, na.rm=TRUE)) %>% show()
38
39 print(median(worldCupMatchesDF$Attendance, na.rm = TRUE))
40
41 #####
42 setwd("~/Users/kennethsebastian/Library/CloudStorage/GoogleDrive-ksebasti@umich.edu/My Drive/DATA")
43 metabolitesDF = read.csv('metabolite.csv', header = TRUE)
44 #(a) Find how many Alzheimers patients there are in the dataset. (Hint: Please refer to question 1)
45 summary(metabolitesDF)
46 alzheimersCount = nrow(metabolitesDF[metabolitesDF$Label == "Alzheimer", ])
47 print(paste("Count of patients with Alzheimer is: ", alzheimersCount))
48
49 #(b) Determine the number of missing values for each column. (Hint: is.na())
50 print(colSums(is.na(metabolitesDF)))
51
52 #(c) Remove the rows which has missing value for the Dopamine column and assign the result to a new data frame. (Hint: is.na())
53 metabolitesMissingDopamineDF = metabolitesDF[!is.na(metabolitesDF$Dopamine), ]
54 print(metabolitesMissingDopamineDF)
55
56 #(d) In the new dataframe, replace the missing values in the c4-OH-Pro column with the median value of the same column. (Hint: there is median())
57 metabolitesMissingDopamineDF %>% mutate(c4.OH.Pro = ifelse(is.na(c4.OH.Pro), median(c4.OH.Pro, na.rm = T), c4.OH.Pro)) %>% show()
58
59 #(e) (Optional) Drop columns which have more than 25% missing values.
60 # (Hint: when you slice your data frame, you can use -c(.., ..., ...) where ... represent one column name)
61 metabolitesDF = metabolitesDF[, which((colMeans(is.na(metabolitesDF))) * 100 >= 25)]
62 print(metabolitesDF)
63
```

49:1 (Untitled) R Script

Console Terminal Background Jobs

```
R 4.3.1 ~ /Library/CloudStorage/GoogleDrive-ksebasti@umich.edu/shortcut-targets-by-id/1ehWwunuAo7CE1V6k2JYkUnQMmxh5pph3C/DATA/
> alzheimersCount = nrow(metabolitesDF[metabolitesDF$Label == "Alzheimer", ])
> print(paste("Count of patients with Alzheimer is: ", alzheimersCount))
[1] "Count of patients with Alzheimer is: 35"
>
```

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

```

11
12
13 setwd("~/Users/kennethsebastian/Library/CloudStorage/GoogleDrive-ksebasti@umich.edu/My Drive/DATA")
14 worldCupMatchesDF = read.csv('WorldCupMatches.csv', header = TRUE)
15 head(worldCupMatchesDF)
16
17 # (a) Find the size of the data frame. How many rows, how many columns?
18 print(paste("The number of rows is ",
19   nrow(worldCupMatchesDF),
20   ", Number of columns is ",
21   ncol(worldCupMatchesDF)
22 ))
23
24 # (b) Use summary function to report the statistical summary of your data.
25 print(summary(worldCupMatchesDF))
26
27 # (c) Find how many unique locations olympics were held at.
28 uniqueLoc = unique(na.omit(worldCupMatchesDF$City))
29 print(paste("The count of unique locations is: ",
30   length(uniqueLoc)
31 ))
32
33 # (d) Find the average attendance.
34 print(paste("The average attendance for matches is: ", mean(worldCupMatchesDF$Attendance, na.rm=TRUE)))
35
36 # (e) For each Home Team, what is the total number of goals scored? (Hint: Please refer to question 1)
37 worldCupMatchesDF %>% group_by(Home.Team.Name) %>% summarise(Home.Team.Goals = sum(Home.Team.Goals)) %>% show()
38
39 # (f) What is the average number of attendees for each year? Is there a trend or pattern in the data in that sense?
40 worldCupMatchesDF %>% group_by(Year) %>% summarise(Avg.Attendance = mean(Attendance, na.rm=TRUE)) %>% show()
41
42 print(median(worldCupMatchesDF$Attendance, na.rm = TRUE))
43
44 #####
45 setwd("~/Users/kennethsebastian/Library/CloudStorage/GoogleDrive-ksebasti@umich.edu/My Drive/DATA")
46 metabolitesDF = read.csv('metabolite.csv', header = TRUE)
47
48 # (a) Find how many Alzheimer's patients there are in the dataset. (Hint: Please refer to question 1)
49 summary(metabolitesDF)
50
51 alzheimersCount = nrow(metabolitesDF[metabolitesDF$Label == "Alzheimer", ])
52 print(paste("Count of patients with Alzheimer is: ", alzheimersCount))
53
54 # (b) Determine the number of missing values for each column. (Hint: is.na())
55 print(colSums(is.na(metabolitesDF)))
56
57 # (c) Remove the rows which has missing value for the Dopamine column and assign the result to a new data frame. (Hint: is.na())
58 metabolitesMissingDopamineDF = metabolitesDF[!is.na(metabolitesDF$Dopamine), ]
59 print(metabolitesMissingDopamineDF)
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```

Console Terminal Background Jobs

R 4.3.1 · ~/Library/CloudStorage/GoogleDrive-ksebasti@umich.edu/shortcut-targets-by-id/1ehWwunuAo7CE1V62JYkUnQMmxh5p3C/DATA/

```

> # (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
0	0	0	0	0	0	0	0

#(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)

The screenshot shows the RStudio interface with a script editor on the left, a console at the bottom, and a sidebar on the right. The script editor contains R code for data manipulation. The console shows the output of the code, which is a data frame with 139 rows and 139 columns. The sidebar on the right shows the Environment pane with a table of metabolites.

R Script:

```
13 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$City))
28 print(paste("The count of unique locations is: ",
29           length(uniqueLoc)
30 ))
31
32 #(d) Find the average attendance.
33 print(paste("The average attendance for matches is: ",mean(worldCupMatchesDF$Attendance,na.rm=TRUE)))
34 #(e) For each Home Team,what is the total number of goals scored?(Hint:Please refer to question1)
35 worldCupMatchesDF %>% group_by(Home.Team.Name) %>% summarise(Home.Team.Goals = sum(Home.Team.Goals)) %>% show()
36 #(f) What is the average number of attendees for each year? Is there a trend or pattern in the data in that sense?
37 worldCupMatchesDF %>% group_by(Year) %>% summarise(Avg.Attendance = mean(Attendance, na.rm =TRUE)) %>% show()
38
39 print(median(worldCupMatchesDF$Attendance, na.rm = TRUE))
40
41 #####
42 setwd("/Users/KennethSebastian/Library/CloudStorage/GoogleDrive-ksebasti@umich.edu/My Drive/DATA")
43 metabolitesDF = read.csv("metabolite.csv", header = TRUE)
44 #(a) Find how many Alzheimers patients there are in the dataset. (Hint:Please refer to question1)
45 summary(metabolitesDF)
46 alzhimersCount = nrow(metabolitesDF[metabolitesDF$Label == "Alzheimer", ])
47 print(paste("Count of patients with Alzheimer is: ", alzhimersCount))
48
49 #(b) Determine the number of missing values for each column. (Hint:is.na())
50 print(colSums(is.na(metabolitesDF)))
51
52 #(c) Remove the rows which has missing value for the Dopamine column and assign the result to a new data frame. (Hint: is.na( ))
53 metabolitesMissingDopamineDF = metabolitesDF[!is.na(metabolitesDF$Dopamine), ]
54 print(metabolitesMissingDopamineDF)
55
56 #(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.)
57 metabolitesMissingDopamineDF %>% mutate(c4.OH.Pro = ifelse(is.na(c4.OH.Pro), median(c4.OH.Pro, na.rm = T),c4.OH.Pro)) %>% show()
58
59 #(e) (Optional)Drop columns which have more than 25% missing values.
60 #Hint:when you slice your data frame,you can use <[ , ]> where ... represent one column name)
61 metabolitesDF = metabolitesDF[ , which(!colMeans(is.na(metabolitesDF))) * 100 >= 25]
62 print(metabolitesDF)
63
```

Console:

```
56:1 (Untitled) :
R 4.3.1 - ~/Library/CloudStorage/GoogleDrive-ksebasti@umich.edu/shortcut-targets-by-id/1ehWwunaAo7CE1VK2/YkUnQmXh5pph3C/DATA/
> print(metabolitesMissingDopamineDF)
[1] Label Phe Pro Ser Thr ADMA alpha.AAA Creatinine
[9] DOPA Histamine Kynurenine Putrescine Sarcosine Serotonin Spermidine t4.OH.Pro
[17] SDMA C0 C16 C16.1 C18 C18.1 C18.2 C18.2 C14
[25] C14.1 C14.2 C16 C16.1 C18 C18.1 C18.2 C18.2 C2
[33] C3 C4 C3.DC..C4.OH. C4.1 C5 C5.OH..C3.DC.M. C6..C4.1.DC. C8
[41] 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
[49] lysoPC.a.C20.4 lysoPC.a.C24.0 lysoPC.a.C26.0 lysoPC.a.C26.1 lysoPC.a.C28.0 lysoPC.a.C28.1 lysoPC.a.C28.2 lysoPC.a.C28.3
[57] PC.aa.C28.1 PC.aa.C30.0 PC.aa.C32.0 PC.aa.C32.1 PC.aa.C32.2 PC.aa.C32.3 PC.aa.C34.1 PC.aa.C34.2 PC.aa.C34.3
[65] 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 PC.aa.C36.7
[73] PC.aa.C38.0 PC.aa.C38.3 PC.aa.C38.4 PC.aa.C38.5 PC.aa.C38.6 PC.aa.C38.7 PC.aa.C40.1 PC.aa.C40.2 PC.aa.C40.3
[81] PC.aa.C40.4 PC.aa.C40.5 PC.aa.C40.6 PC.aa.C40.7 PC.aa.C40.8 PC.aa.C40.9 PC.aa.C40.10 PC.aa.C40.11 PC.aa.C40.12
[89] PC.aa.C42.0 PC.aa.C42.3 PC.aa.C42.4 PC.aa.C42.5 PC.aa.C42.6 PC.aa.C42.7 PC.aa.C42.8 PC.aa.C42.9 PC.aa.C42.10
[97] PC.aa.C44.0 PC.aa.C44.3 PC.aa.C44.4 PC.aa.C44.5 PC.aa.C44.6 PC.aa.C44.7 PC.aa.C44.8 PC.aa.C44.9 PC.aa.C44.10
[105] PC.aa.C46.0 PC.aa.C46.3 PC.aa.C46.4 PC.aa.C46.5 PC.aa.C46.6 PC.aa.C46.7 PC.aa.C46.8 PC.aa.C46.9 PC.aa.C46.10
[113] PC.aa.C48.0 PC.aa.C48.3 PC.aa.C48.4 PC.aa.C48.5 PC.aa.C48.6 PC.aa.C48.7 PC.aa.C48.8 PC.aa.C48.9 PC.aa.C48.10
[121] PC.aa.C44.4 PC.aa.C44.5 PC.aa.C44.6 PC.aa.C44.7 PC.aa.C44.8 PC.aa.C44.9 PC.aa.C44.10 PC.aa.C44.11 PC.aa.C44.12
[129] SM.C16.0 SM.C16.1 SM.C16.2 SM.C16.3 SM.C16.4 SM.C16.5 SM.C16.6 SM.C16.7 SM.C16.8
[137] SM.C26.1 H1.1 H1.2 H1.3 H1.4 H1.5 H1.6 H1.7 H1.8
<0 rows> (or 0-length row.names)
>
```

Environment:

Object	Class	Attributes
metabolitesDF	data.frame	69 obs. of 139 variables
metabolitesML...	data.frame	0 obs. of 139 variables
alzhimersCou...	integer	35L

#(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()

R Script

```
13 # Number of columns is
14 ncol(worldCupMatchesDF)
15
16
17
18
19
20
21
22
23 # (b) Uses summary function to report the statistical summary of your data.
24 print(summary(worldCupMatchesDF))
25
26
27
28 # (c) Find how many unique locations olympics were held at.
29 uniqueLoc = unique(na.omit(worldCupMatchesDF$City))
30 print(paste("The count of unique locations is: ",
31           length(uniqueLoc)))
32
33
34 # (d) Find the average attendance.
35 print(paste("The average attendance for matches is: ", mean(worldCupMatchesDF$Attendance, na.rm=TRUE)))
36
```

Console

```
R 4.3.1 - ~/Library/CloudStorage/GoogleDrive-ksebast@umich.edu/shortcut-targets-by-id/1ehWwuuA07CE1Vx2jYkUnQMmx5p33C/DATA/
> # (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()
  Label  Phe Pro Ser Thr ADMA alpha.AAA c4.OH.Pro Carnosine Creatinine DOPA Dopamine Histamine Kynurenine Met.50 Nitro.Tyr PEA
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 0.027 NA
2 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 NA
3 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 NA
4 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 0.002 NA
5 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 NA
6 Putrescine Carnosine Serotonin Spermidine Spermine t4.OH.Pro Taurine SDMA C0 C10 C10.1 C10.2 C12 C12.DC C12.1 C14 C14.1 C14.1.OH
1 0.068 17.8 0.147 0.188 NA 24.0 125 1.13 18.2 0.059 0.312 0.038 0.030 0.042 0.290 0.023 0.019 0.008
4 0.110 18.7 0.255 0.353 NA 23.1 159 1.34 23.5 0.071 0.317 0.040 0.045 0.048 0.275 0.026 0.028 0.010
5 0.118 22.5 0.390 0.473 NA 26.9 149 1.24 13.6 0.139 0.472 0.074 0.056 0.079 0.394 0.034 0.043 0.016
8 0.161 23.3 0.215 0.276 NA 10.7 133 1.04 13.3 0.051 0.217 0.030 0.041 0.035 0.174 0.024 0.017 0.007
9 0.121 22.1 0.166 0.327 NA 16.0 215 1.24 15.8 0.061 0.258 0.036 0.037 0.038 0.228 0.022 0.018 0.007
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 C2 C3 C3.OH C3.1 C4 C3.DC C4.OH C4.1 C5
1 0.008 0.006 0.046 0.008 0.009 0.007 0.005 0.013 0.013 0.024 0.003 0.016 1.97 0.354 0.008 0.015 0.082 0.045 0.025 0.094
4 0.013 0.011 0.074 0.011 0.015 0.008 0.006 0.009 0.020 0.035 0.004 0.033 2.10 0.278 0.010 0.017 0.110 0.077 0.031 0.145
5 0.025 0.017 0.062 NA 0.024 0.014 0.012 0.025 0.031 0.034 0.012 0.017 5.62 0.436 0.029 0.035 0.106 0.099 0.069 0.141
8 0.006 0.007 0.060 0.006 0.010 0.005 0.004 0.008 0.020 0.025 0.004 0.019 1.66 0.258 0.008 0.012 0.082 0.047 0.021 0.107
9 0.007 0.007 0.054 0.005 0.012 0.005 0.005 0.009 0.014 0.026 0.003 0.016 2.21 0.233 0.008 0.014 0.088 0.029 0.024 0.127
C5.M.DC C5.OH C3.DC.M C5.1 C5.1.DC C6 C6.1 C6.1.DC C5.DC C6.OH C6.1 C7.DC C8 C9 lysoPC.a C14.0 lysoPC.a C16.0 lysoPC.a C16.1
1 0.023 0.026 0.030 0.020 0.022 0.014 0.018 0.011 0.062 0.016 2.23 37.9 2.66
4 0.034 0.041 0.035 0.016 0.029 0.016 0.027 0.017 0.091 0.018 2.19 32.8 2.39
5 0.094 0.058 0.073 0.049 0.052 0.040 0.040 0.036 0.192 0.041 1.88 24.5 1.27
8 0.023 0.023 0.021 0.017 0.036 0.011 NA 0.009 0.062 0.011 2.13 33.7 3.09
9 0.024 0.024 0.025 0.016 0.026 0.018 0.015 0.013 0.064 0.014 2.10 36.0 3.46
lysoPC.a C17.0 lysoPC.a C18.0 lysoPC.a C18.1 lysoPC.a C18.2 lysoPC.a C20.3 lysoPC.a C20.4 lysoPC.a C24.0 lysoPC.a C26.0 lysoPC.a C26.1
1 0.446 9.00 8.58 7.27 1.83 8.25 0.079 0.113 0.053
4 0.323 7.21 7.22 7.62 1.64 6.75 0.066 0.086 0.045
5 0.382 6.66 5.39 3.60 0.97 6.26 0.084 0.118 0.053
8 0.455 6.96 7.31 7.53 2.35 8.73 0.061 0.083 0.047
9 0.435 7.27 8.11 6.75 2.08 7.82 0.068 0.083 0.037
lysoPC.a C28.0 lysoPC.a C28.1 PC.aa C24.0 PC.aa C26.0 PC.aa C28.1 PC.aa C30.0 PC.aa C32.0 PC.aa C32.1 PC.aa C32.2 PC.aa C32.3 PC.aa C34.1
1 0.108 0.072 0.082 0.082 0.438 0.571 2.35 11.4 9.22 NA 0.092 109.0
4 0.076 0.076 0.076 0.486 0.685 3.33 18.6 13.30 0.053 0.079 106.0
5 0.092 0.072 0.069 0.401 0.513 1.78 13.8 5.03 NA 0.102 83.4
8 0.124 0.078 0.082 0.424 0.605 2.32 12.1 11.50 NA 0.093 83.6
9 0.096 0.061 0.064 0.430 0.486 2.05 10.4 11.40 NA 0.082 89.6
PC.aa C34.2 PC.aa C34.3 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 PC.aa C38.0
1 71.0 1.430 0.200 2.38 21.7 42.4 120.0 1.86 0.084 1.230
4 93.6 1.590 0.190 2.57 20.9 48.8 41.2 122.0 1.76 0.070 1.160
5 35.9 0.709 0.135 1.83 20.5 28.5 21.9 98.1 1.70 0.048 1.100
8 60.6 1.580 0.251 2.16 18.4 34.4 41.5 110.0 2.03 0.106 1.150
9 55.1 1.390 0.221 1.62 18.3 32.0 41.8 102.0 1.77 0.081 0.902
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 PC.aa C40.4 PC.aa C40.5 PC.aa C40.6 PC.aa C42.0
1 32.1 95.1 16.8 41.6 0.195 0.074 0.491 3.48 5.66 21.8 0.364
4 28.7 92.7 14.3 29.9 0.220 0.097 0.433 3.59 5.06 14.0 0.427
5 23.3 101.0 13.8 36.2 0.165 0.044 0.525 3.37 5.29 22.5 0.125
8 31.3 81.7 14.5 42.8 0.225 0.099 0.384 4.02 5.49 20.6 0.365
9 31.1 75.9 13.6 37.6 0.181 0.100 0.298 2.88 4.88 17.5 0.396
```

Environment

Global Environment

Data

- metabolitesDF 69 obs. of 192 variables
- metabolitesMi... 49 obs. of 192 variables

Values

- alzheimersCou... 35L

Files Plots Packages Help Viewer Presentation

R: Handle Missing Values in Objects - Find in Topic

na.fail (stats) R Documentation

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 and 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 of the result, which is "exclude". This gives different behaviour in functions making use of [naresiduals](#) and [napredict](#): when `na.exclude` is used the residuals and predictions are padded to the correct length by inserting `NA`s for cases omitted by `na.exclude`.

#(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

Go to file/function Addins

ICA5.R x Untitled1 x ICA6.R x ICA7.R x HW2.R x exc3_for_loops.R x exc4_functions_mymean.R x

Source on Save Run Source

```
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/1ehWwunuAo7CE1Vvk2/YkUnQMmxh5pph3C/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.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.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	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.C38.1	PC.aa.C38.2	PC.aa.C38.3	PC.aa.C38.4	PC.aa.C38.5	PC.aa.C38.6	PC.aa.C40.1
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		