DATA SCIENCE PROGRAMMING

1. Basic Statistics
2. Creating and Exploring a Data Frame
3. Creating and Exploring a List
4. Creating and Exploring Factors
5. Read a CSV file
6. Filter and select using the library dplyr
7. Grouped summary using dplyr
8. Visualisation with Histogram, Boxplot, Scatterplot, Frequency table, Bar plot, Pie chart.
9. Export the data frame to a CSV file
10. Create a Matrix and perform a matrix operation
11. Create a User defined functions

CODE:

```
#basics of R


# Arithmetic operators:

a<- c(1,0.1)
b<- c(2.33,4)
print(a+b)
print(a-b)
print(a*b)
print(a/b)
print(a^b)
print(a%%b)

# Logical operators:

a<-c(TRUE,0.1)
b<-c(0,4+3i)
print(a&b)
print(a|b)
c<-c(0,FALSE)
print(!c)
print(a[1]&&b[1])
print(a[1]||b[1])

#relational operators:

d<-c(TRUE,0.1,"apple")
e<-c(0,0.1,"bat")
print(d<e)    #less than

f<-c(TRUE,0.1,"bat")
g<-as.character(d)
h<-as.character(f)
print(g<=h)      #less than or equal

print(d>f)     #greater than
print(d>=f)    #less than or equal

print(d!=e)  #not equal

#assignment operator:
vec1 = c("ab", TRUE)
print (vec1)

c("ab", TRUE) ->> vec1
print (vec1)

#datatypes
```

```
x=5.6
y=5
z = x > y
r = 4 + 3i
print(class(x))
print(typeof(x))
print(typeof(y))
print(is.integer(y))
print(z)
print(typeof(z))
print(typeof(r))


#creating variable


print("HelloWorld")
var1="Simple assignment"
var2<-"Left assignment"
"Right assignment" -> var3

rm(var3)     #to remove a variable

#mean and standard deviation
data<- c(5,10,15,20, 25, 30, 35, 40, 45, 50)
print(mean(data))
print(sd(data))


# Dataframe in R

friend.data <- data.frame(
  friend_id = c(1:5),
  friend_name = c("AAA", "BBB",
                  "LLL", "HHH",
                  "SSSS"),
  stringsAsFactors = FALSE
)
print(friend.data)        #DF
print(str(friend.data))   #Structure of DF

print(summary(friend.data))    #Summary of DF

result <- data.frame(friend.data$friend_name)
print(result)         #Extract specific data

friend.data$location <- c("Kolkata", "Delhi",
                          "Bangalore", "Hyderabad",
                          "Chennai")
resultant <- friend.data
print(resultant)      #Expand DF


dim(friend.data)    #no. of rows & cols in DF


#add a new row to DF
New_Friend <- c(6, "TTTT", "Goa")
friend.data <- rbind(friend.data, New_Friend)

cat("\nUpdated dataframe after adding a new friend:\n")
print(friend.data)
#cat-> concatenate and print


#add a new col to DF
Age <- c(5, 10, 8, 4, 12, 9)
friend.data <- cbind(friend.data, Age)
colnames(friend.data)[ncol(friend.data)] <- "Age"
print(friend.data)

#remove a row
friend.data <- subset(friend.data, friend_id != 3)
print(friend.data)


#remove a col
friend.data <- select(friend.data, -Age)
print(friend.data)
```

```
#vectors:

#creating vector
X<- c(61, 4, 21, 67, 89, 2)
cat('using c function', X, '\n')

Y<- seq(1, 10, length.out = 5)
cat('using seq() function', Y, '\n')

Z<- 2:7
cat('using colon', Z)


#numeric vectors:
v1 <- c(4, 5, 6, 7)
typeof(v1)

v2 <- c(1L, 4L, 2L, 5L)
typeof(v2)


#logical vector:

v1<-c(TRUE,FALSE,TRUE,NA)
typeof(v1)


#length of vector:
x <- c(1, 2, 3, 4, 5)
length(x)


#Accessing R vector:
X <- c(2, 5, 18, 1, 12)
cat('Using Subscript operator', X[2], '\n')

Y <- c(4, 8, 2, 1, 17)
cat('Using combine() function', Y[c(4, 1)], '\n')

# Sorting elements of a R Vector
X <- c(8, 2, 7, 1, 11, 2)

A <- sort(X)
cat('ascending order', A, '\n')

B <- sort(X, decreasing = TRUE)
cat('descending order', B)
```

output:

Console | Terminal | Background Jobs

R 4.3.3 · ~/

```
> source("~/practise.r")
[1] 3.33 4.10
[1] -1.33 -3.90
[1] 2.33 0.40
[1] 0.4291845 0.0250000
[1] 1e+00 1e-04
[1] 1.0 0.1
[1] FALSE  TRUE
[1] TRUE TRUE
[1] TRUE TRUE
[1] FALSE
[1] TRUE
[1] FALSE FALSE  TRUE
[1] TRUE TRUE TRUE
[1] FALSE FALSE FALSE
[1]  TRUE  TRUE FALSE
[1]  TRUE FALSE  TRUE
[1] "ab"   "TRUE"
[1] "ab"   "TRUE"
[1] "numeric"
[1] "double"
[1] "double"
[1] FALSE
[1] TRUE
[1] "logical"
[1] "complex"
[1] "HelloWorld"
[1] 27.5
[1] 15.13825
  friend_id friend_name
1         1         AAA
2         2         BBB
3         3         LLL
4         4         HHH
5         5        SSSS
'data.frame':   5 obs. of  2 variables:
 $ friend_id  : int  1 2 3 4 5
 $ friend_name: chr  "AAA" "BBB" "LLL" "HHH" ...
NULL
```

Console | Terminal | Background Jobs

R 4.3.3 · ~/

```
 $ friend_id  : int  1 2 3 4 5
 $ friend_name: chr  "AAA" "BBB" "LLL" "HHH" ...
NULL
   friend_id friend_name
 Min.   :1   Length:5
 1st Qu.:2   Class :character
 Median :3   Mode  :character
 Mean   :3
 3rd Qu.:4
 Max.   :5
  friend.data.friend_name
1                     AAA
2                     BBB
3                     LLL
4                     HHH
5                    SSSS
  friend_id friend_name  location
1         1         AAA   Kolkata
2         2         BBB     Delhi
3         3         LLL Bangalore
4         4         HHH Hyderabad
5         5        SSSS   Chennai

Updated dataframe after adding a new friend:
  friend_id friend_name  location
1         1         AAA   Kolkata
2         2         BBB     Delhi
3         3         LLL Bangalore
4         4         HHH Hyderabad
5         5        SSSS   Chennai
6         6        TTTT       Goa
  friend_id friend_name  location Age
1         1         AAA   Kolkata   5
2         2         BBB     Delhi  10
3         3         LLL Bangalore   8
4         4         HHH Hyderabad   4
5         5        SSSS   Chennai  12
6         6        TTTT       Goa   9
  friend_id friend_name  location Age
1         1         AAA   Kolkata   5
```

22MID0035
SWATHI D
DATA SCIENCE PROGRAMMING LAB

```
#Numeric Functions:
# 1. is.numeric()
x<-10
print(is.numeric(x))
z <- c(10, "Hello", 2+3i)
print(is.numeric(z))

# 2. abs()
print(abs(-10))

# 3. sq-rt()
p <- sqrt(12.28)
print(p)

# 4. ceiling()
ex_1 <- ceiling(23.48)
print(ex_1)

# 5. floor()
ex_2 <- floor(-18.25)
print(ex_2)

# 6. round()
ex_3 <- round(c(86.36349,  88.60154, 103.26688,  87.14673, 102.21277), digits = 2)
print(ex_3)

#Character Functions:
# 1. as.character
Y = as.character(25)
print(class(Y))

# 2. is.character
x = "I love R Programming"
print(is.character(x))


# 3. Concatenate Strings
x = "Swathi"
y ="Deena dhayalan"
print(paste(x, y))

# 4. extract substring

x = "abcdef"
print(substr(x, 1, 3))

# 5. string lenght:
```

```
x = "I love R Programming"
print(nchar(x))

# 6. to upper and to lower:
print(tolower(x))
print(toupper(x))

## access lubridate package
library(lubridate)
print(now())

dates <- c("2022-07-11", "2012-04-19", "2017-03-08")
print(year(dates))

#vectors:
fruits <- c("banana", "apple", "orange")
print(fruits)

numbers <- c(1, 2, 3)
print(numbers)

fruits[1] <- "pear"
print(fruits)


#matrices:

thismatrix <- matrix(c(1,2,3,4,5,6), nrow = 3, ncol = 2)
print(thismatrix)

#Access Matrix Items
thismatrix <- matrix(c("apple", "banana", "cherry", "orange"), nrow = 2, ncol = 2)
print(thismatrix[1, 2])
```

OUTPUT

```
> source("~/22mid0035.R")
[1] TRUE
[1] FALSE
[1] 10
[1] 3.504283
[1] 24
[1] -19
[1]  86.36  88.60 103.27  87.15 102.21
[1] "character"
[1] TRUE
[1] "Swathi Deena dhayalan"
[1] "abc"
[1] 20
[1] "i love r programming"
[1] "I LOVE R PROGRAMMING"
Error in library(lubridate) : there is no package called ‘lubridate’
> #vectors:
> fruits <- c("banana", "apple", "orange")
> print(fruits)
[1] "banana" "apple"  "orange"
> numbers <- c(1, 2, 3)
> print(numbers)
[1] 1 2 3
> fruits[1] <- "pear"
> print(fruits)
[1] "pear"   "apple"  "orange"
> thismatrix <- matrix(c(1,2,3,4,5,6), nrow = 3, ncol = 2)
> print(thismatrix)
     [,1] [,2]
[1,]    1    4
[2,]    2    5
[3,]    3    6
> #Access Matrix Items
> thismatrix <- matrix(c("apple", "banana", "cherry", "orange"), nrow = 2, nc
ol = 2)
> print(thismatrix[1, 2])
```

DATA SCIENCE PROGRAMMING (LAB)
DAY 3

SWATHI D
22MID0035

Ex 1:

x=c(4,6,7,10,9,4,15)
y=c(0,10,1,8,2,3,4)
print(x*y) #vector multiplication

```
> x=c(4,6,7,10,9,4,15)
> y=c(0,10,1,8,2,3,4)
> print(x*y)        #vector multiplication
[1]  0 60  7 80 18 12 60
```

Ex 2:

a=c(1,2,4,5,6)
b=c(3,2,4,1,9)
cbind(a,b) #column binding

```
> a=c(1,2,4,5,6)
> b=c(3,2,4,1,9)
> print(cbind(a,b))     #column binding
     a b
[1,] 1 3
[2,] 2 2
[3,] 4 4
[4,] 5 1
[5,] 6 9
>
```

Ex 3:

a=c(1,2,4,5,6)
b=c(3,2,4,1,9)
rbind(a,b) #row binding

```
> a=c(1,2,4,5,6)
> b=c(3,2,4,1,9)
> print(rbind(a,b))     #row binding
  [,1] [,2] [,3] [,4] [,5]
a    1    2    4    5    6
b    3    2    4    1    9
```

Ex 4:

a=c(1,2,4,5,6)
b=c(3,2,4,1,9)
a<=b #checks equality and returns boolean outputs

```
> a=c(1,2,4,5,6)
> b=c(3,2,4,1,9)
> print(a<=b) #checks equality and returns boolean outputs
[1]  TRUE  TRUE  TRUE FALSE  TRUE
```

Ex 5:

```
x<-c(1:12)
dim(x)                  #dimension of x
length(x)               #length of x
```

```
> x<-c(1:12)
> print(dim(x))  #dimension of x
NULL
> print(length(x)) #length of x
[1] 12
```

Ex 6:

```
a=c(12:5)
is.numeric(a)           #if the input is numeric or not
```

```
> a=c(12:5)
> print(is.numeric(a))  #if the input is numeric or not
[1] TRUE
```

Ex 7:

```
x=c(12:4)
y=c(0,1,2,0,1,2,0,1,2)
which(!is.finite(x/y))  #checks finiteness
```

```
> x=c(12:4)
> y=c(0,1,2,0,1,2,0,1,2)
> print(which(!is.finite(x/y)))    #checks finiteness
[1] 1 4 7
```

Ex 8:

```
x=letters[1:10]
y=letters[15:24]
x<y                     #checks equality
```

```
> x=letters[1:10]
> y=letters[15:24]
> print(x<y)
 [1] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
```

Ex 9:

```
x=c('blue','red"green','yellow')
is.character(x)         #checks character or not
```

```
> x=c('blue','red','green','yellow')
> print(is.character(x))
[1] TRUE
```

Ex 10:

x=c('blue',10,'green',20)
is.character(x) #checks character or not

```
> t=c('blue',10,'green',20)
> print(is.character(t))
[1] TRUE
```

TASK 2
1. load dataset

df=read.csv("/home/matlab/Downloads/abc.csv")
df

```
> df=read.csv("/home/matlab/Downloads/abc.csv")
> df
    OBJECTID AREASYMBOL SPATIALVER MUSYM   MUKEY                    MUNAME                    FARMCLASS
1          1      CT602          1   102 3396010 Pootatuck fine sandy loam All areas are prime farmland
2          2      CT602          1   102 3396010 Pootatuck fine sandy loam All areas are prime farmland
3          3      CT602          1   102 3396010 Pootatuck fine sandy loam All areas are prime farmland
4          4      CT602          1   102 3396010 Pootatuck fine sandy loam All areas are prime farmland
5          5      CT602          1   102 3396010 Pootatuck fine sandy loam All areas are prime farmland
6          6      CT602          1   102 3396010 Pootatuck fine sandy loam All areas are prime farmland
7          7      CT602          1   102 3396010 Pootatuck fine sandy loam All areas are prime farmland
8          8      CT602          1   102 3396010 Pootatuck fine sandy loam All areas are prime farmland
9          9      CT602          1   102 3396010 Pootatuck fine sandy loam All areas are prime farmland
10        10      CT602          1   102 3396010 Pootatuck fine sandy loam All areas are prime farmland
11        11      CT602          1   102 3396010 Pootatuck fine sandy loam All areas are prime farmland
12        12      CT602          1   102 3396010 Pootatuck fine sandy loam All areas are prime farmland
13        13      CT602          1   102 3396010 Pootatuck fine sandy loam All areas are prime farmland
14        14      CT602          1   102 3396010 Pootatuck fine sandy loam All areas are prime farmland
15        15      CT602          1   102 3396010 Pootatuck fine sandy loam All areas are prime farmland
16        16      CT602          1   102 3396010 Pootatuck fine sandy loam All areas are prime farmland
17        17      CT602          1   102 3396010 Pootatuck fine sandy loam All areas are prime farmland
18        18      CT602          1   102 3396010 Pootatuck fine sandy loam All areas are prime farmland
19        19      CT602          1   102 3396010 Pootatuck fine sandy loam All areas are prime farmland
20        20      CT602          1   102 3396010 Pootatuck fine sandy loam All areas are prime farmland
```

2. summary(df)

```
> summary(df)
    OBJECTID       AREASYMBOL          SPATIALVER    MUSYM              MUKEY          MUNAME          FARMCLASS          Shape__Area
 Min.   :     1   Length:120589      Min.   :1    Length:120589      Min.   :3395634   Length:120589      Length:120589      Min.   :       0
 1st Qu.: 30148   Class :character   1st Qu.:1    Class :character   1st Qu.:3395764   Class :character   Class :character   1st Qu.:  138064
 Median : 60295   Mode  :character   Median :1    Mode  :character   Median :3395840   Mode  :character   Mode  :character   Median :  261407
 Mean   : 60295                      Mean   :1                       Mean   :3397439                                         Mean   :  513016
 3rd Qu.: 90442                      3rd Qu.:1                       3rd Qu.:3400134                                         3rd Qu.:  535637
 Max.   :120589                      Max.   :1                       Max.   :3400352                                         Max.   :68916295
 Shape__Length
 Min.   :    4.55
 1st Qu.:  1825.39
 Median :  2780.77
 Mean   :  4001.15
 3rd Qu.:  4536.75
 Max.   :264036.94
```

3. head(df)

```
> head(df)
  OBJECTID AREASYMBOL SPATIALVER MUSYM   MUKEY                    MUNAME                    FARMCLASS Shape__Area Shape__Length
1        1      CT602          1   102 3396010 Pootatuck fine sandy loam All areas are prime farmland   265900.51      3883.277
2        2      CT602          1   102 3396010 Pootatuck fine sandy loam All areas are prime farmland    56496.38      1215.889
3        3      CT602          1   102 3396010 Pootatuck fine sandy loam All areas are prime farmland   355328.94      4275.181
4        4      CT602          1   102 3396010 Pootatuck fine sandy loam All areas are prime farmland   338741.36      4643.759
5        5      CT602          1   102 3396010 Pootatuck fine sandy loam All areas are prime farmland   671465.16      5002.829
6        6      CT602          1   102 3396010 Pootatuck fine sandy loam All areas are prime farmland   227106.23      3063.730
```

4. tail(df)

```
> tail(df)
       OBJECTID AREASYMBOL SPATIALVER MUSYM   MUKEY                                        MUNAME                       FARMCLASS Shape__Area
120584   120584      CT601          1     5 3395768 Wilbraham silt loam, 0 to 3 percent slopes Farmland of statewide importance  1400893.54
120585   120585      CT601          1     5 3395768 Wilbraham silt loam, 0 to 3 percent slopes Farmland of statewide importance    45539.96
120586   120586      CT601          1     5 3395768 Wilbraham silt loam, 0 to 3 percent slopes Farmland of statewide importance    76157.09
120587   120587      CT601          1     5 3395768 Wilbraham silt loam, 0 to 3 percent slopes Farmland of statewide importance   229423.15
120588   120588      CT601          1     5 3395768 Wilbraham silt loam, 0 to 3 percent slopes Farmland of statewide importance    91556.48
120589   120589      CT601          1     5 3395768 Wilbraham silt loam, 0 to 3 percent slopes Farmland of statewide importance   282571.67
       Shape__Length
120584    15599.4098
120585      944.0782
120586     1241.7753
120587     3951.3231
120588     1579.3019
120589     3039.4779
```

5. dim(df)

```
> dim(df)
[1] 120589      9
>
```

6. dimnames(df)

```
> dimnames(df)
[[1]]
   [1] "1"   "2"   "3"   "4"   "5"   "6"   "7"   "8"   "9"   "10"  "11"  "12"  "13"  "14"  "15"
  [21] "21"  "22"  "23"  "24"  "25"  "26"  "27"  "28"  "29"  "30"  "31"  "32"  "33"  "34"  "35"
  [41] "41"  "42"  "43"  "44"  "45"  "46"  "47"  "48"  "49"  "50"  "51"  "52"  "53"  "54"  "55"
  [61] "61"  "62"  "63"  "64"  "65"  "66"  "67"  "68"  "69"  "70"  "71"  "72"  "73"  "74"  "75"
  [81] "81"  "82"  "83"  "84"  "85"  "86"  "87"  "88"  "89"  "90"  "91"  "92"  "93"  "94"  "95"
 [101] "101" "102" "103" "104" "105" "106" "107" "108" "109" "110" "111" "112" "113" "114" "115"
 [121] "121" "122" "123" "124" "125" "126" "127" "128" "129" "130" "131" "132" "133" "134" "135"
 [141] "141" "142" "143" "144" "145" "146" "147" "148" "149" "150" "151" "152" "153" "154" "155"
 [161] "161" "162" "163" "164" "165" "166" "167" "168" "169" "170" "171" "172" "173" "174" "175"
 [181] "181" "182" "183" "184" "185" "186" "187" "188" "189" "190" "191" "192" "193" "194" "195"
 [201] "201" "202" "203" "204" "205" "206" "207" "208" "209" "210" "211" "212" "213" "214" "215"
 [221] "221" "222" "223" "224" "225" "226" "227" "228" "229" "230" "231" "232" "233" "234" "235"
```

7. df$
df[3,4]

```
> df$
+ df[3,4]
NULL
```

8. df[,4]

```
> df[,4]
   [1] "102"  "102"  "102"  "102"  "102"  "102"  "102"  "102"  "102"  "102"  "102"  "102"  "102"  "102"  "102"  "102"
  [21] "102"  "102"  "102"  "102"  "102"  "73C"  "73C"  "73C"  "73C"  "73C"  "73C"  "73C"  "73C"  "73C"  "73C"  "73C"
  [41] "73C"  "73C"  "73C"  "102"  "102"  "102"  "102"  "102"  "102"  "102"  "102"  "102"  "102"  "102"  "102"  "102"
  [61] "103"  "103"  "73C"  "73C"  "73C"  "72C"  "72C"  "72C"  "72C"  "72C"  "72C"  "72C"  "72C"  "72C"  "72C"  "72C"
  [81] "72C"  "72C"  "72C"  "72C"  "72C"  "72C"  "72C"  "72C"  "72C"  "72C"  "703B" "703B" "703B" "703B" "703B" "703B"
 [101] "703B" "703B" "703B" "703B" "703B" "703B" "703B" "703B" "703B" "703B" "703B" "703B" "703B" "703B" "703B" "703B"
 [121] "703B" "703B" "703B" "703B" "703B" "703B" "703B" "703B" "703B" "703B" "703B" "703B" "703B" "703B" "703B" "703B"
 [141] "703B" "703B" "703B" "703B" "701A" "701A" "701A" "701A" "701A" "701A" "701A" "701A" "701A" "701A" "701A" "701A"
 [161] "701A" "701A" "701A" "701A" "701A" "701A" "701A" "73C"  "73C"  "73C"  "73C"  "73C"  "73C"  "73C"  "73C"  "73C"
 [181] "73C"  "73C"  "73C"  "73C"  "73C"  "73C"  "73C"  "73C"  "73C"  "73C"  "73C"  "73C"  "73C"  "73C"  "73C"  "73C"
 [201] "73C"  "73C"  "73C"  "73C"  "73C"  "73C"  "73C"  "73C"  "73C"  "73C"  "73C"  "73C"  "73C"  "73C"  "73C"  "73C"
 [221] "73C"  "73C"  "73C"  "73C"  "73C"  "73C"  "73C"  "73C"  "73C"  "73C"  "73C"  "103"  "103"  "103"  "103"  "73C"
 [241] "73C"  "73C"  "73C"  "73C"  "73C"  "73C"  "73C"  "73C"  "73C"  "73C"  "73C"  "73C"  "73C"  "73C"  "73C"  "73C"
 [261] "73C"  "73C"  "73C"  "73C"  "73C"  "73C"  "103"  "103"  "103"  "103"  "103"  "103"  "103"  "103"  "103"  "103"
```

9.df[1:5,c(“OBJECTID”,”AREASYMBOL”]

```
> df[1:5,c("OBJECTID","AREASYMBOL")]
  OBJECTID AREASYMBOL
1        1      CT602
2        2      CT602
3        3      CT602
4        4      CT602
5        5      CT602
```

10. df[1:10,c("MUKEY","MUNAME")]

```
> df[1:10,c("MUKEY","MUNAME")]
    MUKEY                   MUNAME
1  3396010 Pootatuck fine sandy loam
2  3396010 Pootatuck fine sandy loam
3  3396010 Pootatuck fine sandy loam
4  3396010 Pootatuck fine sandy loam
5  3396010 Pootatuck fine sandy loam
6  3396010 Pootatuck fine sandy loam
7  3396010 Pootatuck fine sandy loam
8  3396010 Pootatuck fine sandy loam
9  3396010 Pootatuck fine sandy loam
10 3396010 Pootatuck fine sandy loam
```

11. Creating dataframe:

name<-c("abdul","rithesh","gokul","jagan","druvitha","neha")
height<-c(175,166,170,170,162,168)
weight<-c(63,64,53,65,53,60)
shoe_size<-c(8,7,8,9,7,7)
age<-c(22,20,21,20,20,20)
year_of_birth<-c(2002,2004,2004,2005,2005,2005)
friends<- data.frame(name,height,weight,shoe_size,age,year_of_birth)
print(friends)

```
>   name<-c("abdul","rithesh","gokul","jagan","druvitha","neha")
>   height<-c(175,166,170,170,162,168)
>   weight<-c(63,64,53,65,53,60)
>   shoe_size<-c(8,7,8,9,7,7)
>   age<-c(22,20,21,20,20,20)
>   year_of_birth<-c(2002,2004,2004,2005,2005,2005)
> friends<- data.frame(name,height,weight,shoe_size,age,year_of_birth)
> print(friends)
     name height weight shoe_size age year_of_birth
1   abdul    175     63         8  22          2002
2 rithesh    166     64         7  20          2004
3   gokul    170     53         8  21          2004
4   jagan    170     65         9  20          2005
5 druvitha    162     53         7  20          2005
6    neha    168     60         7  20          2005
```

INSTALL “dplyr”

```
> install.packages("dplyr")
Installing package into '/home/matlab/R/x86_64-pc-linux-gnu-library/4.3'
(as 'lib' is unspecified)
trying URL 'https://cloud.r-project.org/src/contrib/dplyr_1.1.4.tar.gz'
Content type 'application/x-gzip' length 1207521 bytes (1.2 MB)
==================================================
downloaded 1.2 MB

* installing *source* package 'dplyr' ...
** package 'dplyr' successfully unpacked and MD5 sums checked
** using staged installation
** libs
using C++ compiler: 'g++ (Ubuntu 13.3.0-6ubuntu2~24.04) 13.3.0'
g++ -std=gnu++17 -I"/usr/share/R/include" -DNDEBUG       -fpic  -g -O2 -fno-omit-frame-pointer
se-FPSnzf/r-base-4.3.3=. -fstack-protector-strong -fstack-clash-protection -Wformat -Werror=fo
ase-FPSnzf/r-base-4.3.3=/usr/src/r-base-4.3.3-2build2 -Wdate-time -D_FORTIFY_SOURCE=3  -c chop
```

aggregate(friends)

```
The downloaded source packages are in
        '/tmp/Rtmpg0B6Y6/downloaded_packages'
> aggregate(friends)
Error in aggregate.data.frame(friends) :
  argument "FUN" is missing, with no default
```