



**ACADGILD**

## SESSION 3: FOUNDATIONAL R PROGRAMMING

### Assignment 4

Submitted by: Munmun Ghosal

Login Id: munmun55@gmail.com

(M):+91-8007178659

Table of Contents

1. Problem Statement ..... 3

2. Solution ..... 3

## 1. Problem Statement

- A. Implement user defined functions within apply function using the mtcars data set and produce column wise summary statistics using apply function and mtcars dataset.
- B. Write a program to extract the names of the list.

## 2. Solution

- A. Implement user defined functions within apply function using the mtcars data set and produce column wise summary statistics using apply function and mtcars dataset.**

Apply function returns a vector or array or list of values obtained by applying a function to margins of an array or matrix.

Here mtcars.csv dataset is imported using readr package

**The R-script for the given problem is as follows:**

```
library(readr)
mtcars <- read_csv("C:/Users/Munmun/Downloads/mtcars.csv")
View(mtcars)
# show first few(6) rows of mtcars
head(mtcars)
# get the mean of each column
apply(mtcars, 2, mean)
# get the sum of each row
apply(mtcars, 2, sum)
# get column quantiles
apply(mtcars, 2, quantile, probs = c(0.10, 0.25, 0.50, 0.75, 0.90))
# get the mean of the mpg column grouped by cylinders
tapply(mtcars$mpg, mtcars$cyl, mean)
# get the mean of all columns grouped by cylinders
apply(mtcars, 2, function(x) tapply(x, mtcars$cyl, mean))
#Sort the columns of a matrix
apply(mtcars, 2, sort)
```

The output of the R-Script (from Console window) is given as follows:

```
> library(readr)
> mtcars <- read_csv("C:/Users/Munmun/Downloads/mtcars.csv")
Parsed with column specification:
cols(
  mpg = col_double(),
  cyl = col_double(),
  disp = col_double(),
  hp = col_double(),
  drat = col_double(),
  wt = col_double(),
  qsec = col_double(),
  vs = col_double(),
  am = col_double(),
  gear = col_double(),
  carb = col_double()
)
> View(mtcars)
> head(mtcars)
# A tibble: 6 x 11
  mpg   cyl  disp    hp  drat    wt   qsec    vs  am  gear  carb
<dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
1  21     6   160   110   3.9   2.62  16.5     0     1     4     4
2  21     6   160   110   3.9   2.88  17.0     0     1     4     4
3 22.8     4   108    93   3.85   2.32  18.6     1     1     4     1
4 21.4     6   258   110   3.08   3.22  19.4     1     0     3     1
5 18.7     8   360   175   3.15   3.44  17.0     0     0     3     2
6 18.1     6   225   105   2.76   3.46  20.2     1     0     3     1
> apply(mtcars, 2, mean)
      mpg      cyl      disp      hp      drat      wt      qsec
vs
20.090625  6.187500 230.721875 146.687500  3.596563  3.217250 17.848750
0.437500
      am      gear      carb
0.406250  3.687500  2.812500
> apply(mtcars, 2, sum)
      mpg      cyl      disp      hp      drat      wt      qsec      vs
am      gear
642.900 198.000 7383.100 4694.000 115.090 102.952 571.160 14.000
13.000 118.000
      carb
90.000
> apply(mtcars, 2, quantile, probs = c(0.10, 0.25, 0.50, 0.75, 0.90))
      mpg cyl  disp  hp drat    wt  qsec vs am gear carb
10% 14.340  4  80.610 66.0 3.007 1.95550 15.5340 0 0  3  1
25% 15.425  4 120.825 96.5 3.080 2.58125 16.8925 0 0  3  2
50% 19.200  6 196.300 123.0 3.695 3.32500 17.7100 0 0  4  2
75% 22.800  8 326.000 180.0 3.920 3.61000 18.9000 1 1  4  4
90% 30.090  8 396.000 243.5 4.209 4.04750 19.9900 1 1  5  4
> tapply(mtcars$mpg, mtcars$cyl, mean)
      4      6      8
26.66364 19.74286 15.10000
> apply(mtcars, 2, function(x) tapply(x, mtcars$cyl, mean))
      mpg cyl  disp  hp drat    wt  qsec      vs
am      gear
4  26.66364 6  19.74286 8  15.10000
```

```

4 26.66364    4 105.1364  82.63636 4.070909 2.285727 19.13727 0.9090909
0.7272727 4.090909
6 19.74286    6 183.3143 122.28571 3.585714 3.117143 17.97714 0.5714286
0.4285714 3.857143
8 15.10000    8 353.1000 209.21429 3.229286 3.999214 16.77214 0.0000000
0.1428571 3.285714

```

carb

```
4 1.545455
```

```
6 3.428571
```

```
8 3.500000
```

```
> apply(mtcars, 2, sort)
```

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
[1,]	10.4	4	71.1	52	2.76	1.513	14.50	0	0	3	1
[2,]	10.4	4	75.7	62	2.76	1.615	14.60	0	0	3	1
[3,]	13.3	4	78.7	65	2.93	1.835	15.41	0	0	3	1
[4,]	14.3	4	79.0	66	3.00	1.935	15.50	0	0	3	1
[5,]	14.7	4	95.1	66	3.07	2.140	15.84	0	0	3	1
[6,]	15.0	4	108.0	91	3.07	2.200	16.46	0	0	3	1
[7,]	15.2	4	120.1	93	3.07	2.320	16.70	0	0	3	1
[8,]	15.2	4	120.3	95	3.08	2.465	16.87	0	0	3	2
[9,]	15.5	4	121.0	97	3.08	2.620	16.90	0	0	3	2
[10,]	15.8	4	140.8	105	3.15	2.770	17.02	0	0	3	2
[11,]	16.4	4	145.0	109	3.15	2.780	17.02	0	0	3	2
[12,]	17.3	6	146.7	110	3.21	2.875	17.05	0	0	3	2
[13,]	17.8	6	160.0	110	3.23	3.150	17.30	0	0	3	2
[14,]	18.1	6	160.0	110	3.54	3.170	17.40	0	0	3	2
[15,]	18.7	6	167.6	113	3.62	3.190	17.42	0	0	3	2
[16,]	19.2	6	167.6	123	3.69	3.215	17.60	0	0	4	2
[17,]	19.2	6	225.0	123	3.70	3.435	17.82	0	0	4	2
[18,]	19.7	6	258.0	150	3.73	3.440	17.98	0	0	4	3
[19,]	21.0	8	275.8	150	3.77	3.440	18.00	1	0	4	3
[20,]	21.0	8	275.8	175	3.85	3.440	18.30	1	1	4	3
[21,]	21.4	8	275.8	175	3.90	3.460	18.52	1	1	4	4
[22,]	21.4	8	301.0	175	3.90	3.520	18.60	1	1	4	4
[23,]	21.5	8	304.0	180	3.92	3.570	18.61	1	1	4	4
[24,]	22.8	8	318.0	180	3.92	3.570	18.90	1	1	4	4
[25,]	22.8	8	350.0	180	3.92	3.730	18.90	1	1	4	4
[26,]	24.4	8	351.0	205	4.08	3.780	19.44	1	1	4	4
[27,]	26.0	8	360.0	215	4.08	3.840	19.47	1	1	4	4
[28,]	27.3	8	360.0	230	4.11	3.845	19.90	1	1	5	4
[29,]	30.4	8	400.0	245	4.22	4.070	20.00	1	1	5	4
[30,]	30.4	8	440.0	245	4.22	5.250	20.01	1	1	5	4
[31,]	32.4	8	460.0	264	4.43	5.345	20.22	1	1	5	6
[32,]	33.9	8	472.0	335	4.93	5.424	22.90	1	1	5	8

**B. Write a program to extract the names of the list.**

**The R-script for the given problem is as follows:**

**#EXAMPLE 1:**

```
list_data <- list(c("jan","feb","mar"),matrix(c(1,9,7,4,5,7),nrow=2),list("green",1.2,3))
names(list_data) <- c("3months","my_matrix","inner_list")
list_data
print(list_data[1])
print(list_data[2])
print(list_data[3])
names(list_data)
```

**#EXAMPLE 2:**

```
x <- list("Los Angeles" = 1, Boston = 2, London = 3)
x
names(x)
```

**Explanation:**

- Lists are the R objects which contain elements of different types like – numbers, strings, vectors and another list inside it. A list can also contain a matrix or a function as its elements. List is created using list() function.
- Elements of the list can be accessed by the index of the element in the list. In case of named lists it can also be accessed using the names.
- The names of the list can be extracted using names() function

**The output of the R-Script (from Console window) is given as follows:**

```
> #B. Write a program to extract the names of the list
>
> #EXAMPLE 1:
> list_data <-
list(c("jan","feb","mar"),matrix(c(1,9,7,4,5,7),nrow=2),list("green",1.2
,3))
> names(list_data) <- c("3months","my_matrix","inner_list")
> list_data
$`3months`
[1] "jan" "feb" "mar"

$my_matrix
      [,1] [,2] [,3]
[1,]    1    7    5
[2,]    9    4    7

$inner_list
$inner_list[[1]]
[1] "green"
```

```
$inner_list[[2]]  
[1] 1.2
```

```
$inner_list[[3]]  
[1] 3
```

```
> print(list_data[1])  
$`3months`  
[1] "jan" "feb" "mar"
```

```
> print(list_data[2])  
$`my_matrix`  
      [,1] [,2] [,3]  
[1,]     1     7     5  
[2,]     9     4     7
```

```
> print(list_data[3])  
$`inner_list`  
$`inner_list`[[1]]  
[1] "green"
```

```
$`inner_list`[[2]]  
[1] 1.2
```

```
$`inner_list`[[3]]  
[1] 3
```

```
> names(list_data)  
[1] "3months" "my_matrix" "inner_list"
```

```
> #EXAMPLE 2:
```

```
> x <- list("Los Angeles" = 1, Boston = 2, London = 3)
```

```
> x  
$`Los Angeles`  
[1] 1
```

```
$Boston  
[1] 2
```

```
$London  
[1] 3
```

```
> names(x)  
[1] "Los Angeles" "Boston"      "London"
```