

# Labwork - 09

## Naming Columns and Rows

```
> vector1 <- c(1,3,5)
> vector2 <- c(11,22,33,44,55,66,77)
> vArray <- array (c (vector1, vector2), dim = c (5,5), dimnames = list (c ("ROW1",
"ROW2", "ROW3", "ROW4", "ROW5"), c("COL1", "COL2", "COL3", "COL4",
"COL5")) )
> vArray
```

	COL1	COL2	COL3	COL4	COL5
ROW1	1	33	1	33	1
ROW2	3	44	3	44	3
ROW3	5	55	5	55	5
ROW4	11	66	11	66	11
ROW5	22	77	22	77	22

## Accessing Array Elements

```
> print(vArray[3,3])
[1] 5
> print(vArray[1,5])
[1] 1
> print(vArray[,5])
ROW1 ROW2 ROW3 ROW4 ROW5
 1   3   5  11  22
> print(vArray[2,])
COL1 COL2 COL3 COL4 COL5
 3  44   3  44   3
```

## Manipulating Array Elements

```
> vArray[1,] + vArray[2,]
COL1 COL2 COL3 COL4 COL5
 4  77   4  77   4
> vArray[1,] + vArray[,1]
COL1 COL2 COL3 COL4 COL5
 2  36   6  44  23
> vArray[1,] * vArray[,1]
COL1 COL2 COL3 COL4 COL5
 1  99   5 363  22
```

## Calculations across array elements

```
> apply(vArray, c(1),sum)
ROW1 ROW2 ROW3 ROW4 ROW5
 69  97 125 165 220
> apply(vArray, c(2),sum)
COL1 COL2 COL3 COL4 COL5
 42 275  42 275  42
```

## Generating factor levels using gl (a, b) function

# where 'a' is the integers for numbers of levels and 'b' is an integer for the number of times each level has repeated.

```
> gl (3, 2)
[1] 1 1 2 2 3 3
Levels: 1 2 3
> gl (4, 3)
[1] 1 1 1 2 2 2 3 3 3 4 4 4
Levels: 1 2 3 4
> gl(3, 2, labels = c("mango", "berry", "apple"))
[1] mango mango berry berry apple apple
Levels: mango berry apple
```

## Changing the order of levels

# By specifying the levels arguments

```
> myData = c (1, 2, 2, 4, 6, 3, 1, 2, 3, 4, 6, 6, 1, 3, 1, 2, 3, 3, 1)
> myFactor = factor (myData) # Creating a factor using factor function by
> myFactor passing a vector into it
[1] 1 2 2 4 6 3 1 2 3 4 6 6 1 3 1 2 3 3 1
Levels: 1 2 3 4 6
> myNewFactor = factor (myData, labels= c("One", "Two", "Three", "Four", "Six" ))
> myNewFactor
[1] One Two Two Four Six Three One Two Three Four Six Six
[13] One Three One Two Three Three One
Levels: One Two Three Four Six
```

## Extract data from Data Frame

# Using the name of the column to extract a specific column

```
> myDataFrame <- data.frame (items = c ("apple", "carrot", "tomato"), cost_per_item =
c(110,60,25))
> myDataFrame
  items cost_per_item
1  apple         110
2 carrot          60
3 tomato          25
```

```

> print (myDataFrame$items)
[1] apple carrot tomato
Levels: apple carrot tomato
> print (data.frame (myDataFrame$items))
myDataFrame.items
1      apple
2      carrot
3      tomato
> print (data.frame (myDataFrame$items, myDataFrame$cost_per_item))
myDataFrame.items myDataFrame.cost_per_item
1      apple      110
2      carrot      60
3      tomato      25
# extracting 1st two rows
> print (data.frame (myDataFrame$items[1:2]))
myDataFrame.items.1.2.
1      apple
2      carrot

```

## Expand R Data Frame

# A data frame can be expanded by adding columns and rows.

# Adding column

```

> myDataFrame$total_items <- c(5,2,6)

```

```

> myDataFrame

```

```

  items cost_per_item total_items
1 apple      110         5
2 carrot      60         2
3 tomato      25         6

```

# Adding row

```

> myNewDataFrame <- data.frame (items = c ("Beetroot", "Mosambi"),
cost_per_item=c(39,95), total_items = c(2,3))

```

```

> rowAddedDataFrame <- rbind(myDataFrame, myNewDataFrame)

```

```

> rowAddedDataFrame

```

```

  items cost_per_item total_items
1 apple      110         5
2 carrot      60         2
3 tomato      25         6
4 Beetroot      39         2
5 Mosambi      95         3

```

## Merging Data Frames

```
> myDataFrame <- data.frame (id = c ("i001", "i002", "i003"), items = c ("apple", "carrot", "tomato"), cost_per_item = c (110,60,25))
```

```
> myNewDataFrame <- data.frame (id = c ("i004", "i005"), items = c ("beetroot", "mosambi"), cost_per_item = c (39,95))
```

```
> finale <- merge (myDataFrame, myNewDataFrame, by = "id")
```

```
> finale
```

```
[1] id      items.x    cost_per_item.x items.y
```

```
[5] cost_per_item.y
```

```
<0 rows> (or 0-length row.names)
```

```
//Natural join
```

```
> finale <- merge (myDataFrame, myNewDataFrame, by = "id", all = FALSE)
```

```
> finale
```

```
[1] id      items.x    cost_per_item.x items.y
```

```
[5] cost_per_item.y
```

```
<0 rows> (or 0-length row.names)
```

```
//fullouter join
```

```
> finale <- merge(myDataFrame, myNewDataFrame,by="id",all=TRUE)
```

```
> finale
```

```
  id items.x cost_per_item.x items.y cost_per_item.y
```

```
1 i001  apple          110  <NA>         NA
```

```
2 i002  carrot           60  <NA>         NA
```

```
3 i003  tomato           25  <NA>         NA
```

```
4 i004  <NA>           NA beetroot         39
```

```
5 i005  <NA>           NA  mosambi          95
```

```
//cross-join
```

```
> finale <- merge(x = myDataFrame,y = myNewDataFrame,by=NULL)
```

```
> finale
```

```
  id.x items.x cost_per_item.x id.y items.y cost_per_item.y
```

```
1 i001  apple          110 i004 beetroot         39
```

```
2 i002  carrot           60 i004 beetroot         39
```

```
3 i003  tomato           25 i004 beetroot         39
```

```
4 i001  apple          110 i005  mosambi          95
```

```
5 i002  carrot           60 i005  mosambi          95
```

```
6 i003  tomato           25 i005  mosambi          95
```

## Melting the Data

```
> melt (myDataFrame, id = c("id", "items"))
```

	id	items	variable	value
1	i001	apple	cost_per_item	110
2	i002	carrot	cost_per_item	60
3	i003	tomato	cost_per_item	25