Big Data

R programming basics

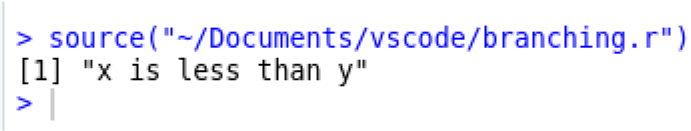
DK Venkata Hem

2020503007

Que 1 – Branching statements in R programming.

```
x <- 5

y <- 6


if (x>y) {

  print("x is greater than y")

} else if(x==y){

  print("x and y are equal")

}else{

   print("x is less than y")

}
```

Output:

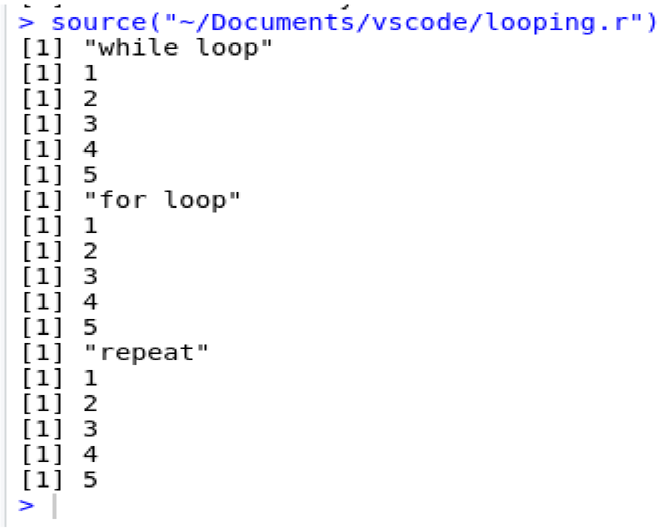


```
> source("~/Documents/vscode/branching.r")
[1] "x is less than y"
>
```

Que 2 – Looping statements in R

```
#while

i<-1

print("while loop")

while(i<6){

   print(i)
   i<-i+1
}

#for
print("for loop")
for(i in 1:5){
   print(i)
}

#repeat
i<-1
print("repeat")
repeat{
   print(i)
   i<-i+1
   if(i>5){
      break
   }
}
```

Output:



```
> source("~/Documents/vscode/looping.r")
[1] "while loop"
[1] 1
[1] 2
[1] 3
[1] 4
[1] 5
[1] "for loop"
[1] 1
[1] 2
[1] 3
[1] 4
[1] 5
[1] "repeat"
[1] 1
[1] 2
[1] 3
[1] 4
[1] 5
>
```

Que 3 – Create two dataframes for Batch 1 and Batch 2 with reg.no, grade, name and cgpa and combine them

- Append one more field class as categorial value

```
batch1 <- data.frame(
   reg_no = c(3001,3002,3003,3004,3005),
   stu_name = c("a","b","c","d","e"),
   cgpa = c(9.1,9.3,9.2,9.2,9.5)
)
# batch1
print("batch1")
print(batch1)
grades <- c("a","a+","a","a","o")
batch1 <- cbind(batch1,grades)
print("Adding grades column")
print(batch1)
print("batch2")
batch2 <- data.frame(
   reg_no = c(3501,3502,3503,3504,3505),
   stu_name = c("a","b","c","d","e"),
   cgpa = c(9.1,9.3,5.2,9.2,9.5)
)

#batch2
print(batch2)
grades <- c("a","a+","a","a","o")
batch2 <- cbind(batch2,grades)
print(batch2)
batch_info <- rbind(batch1,batch2)
```

batch_info

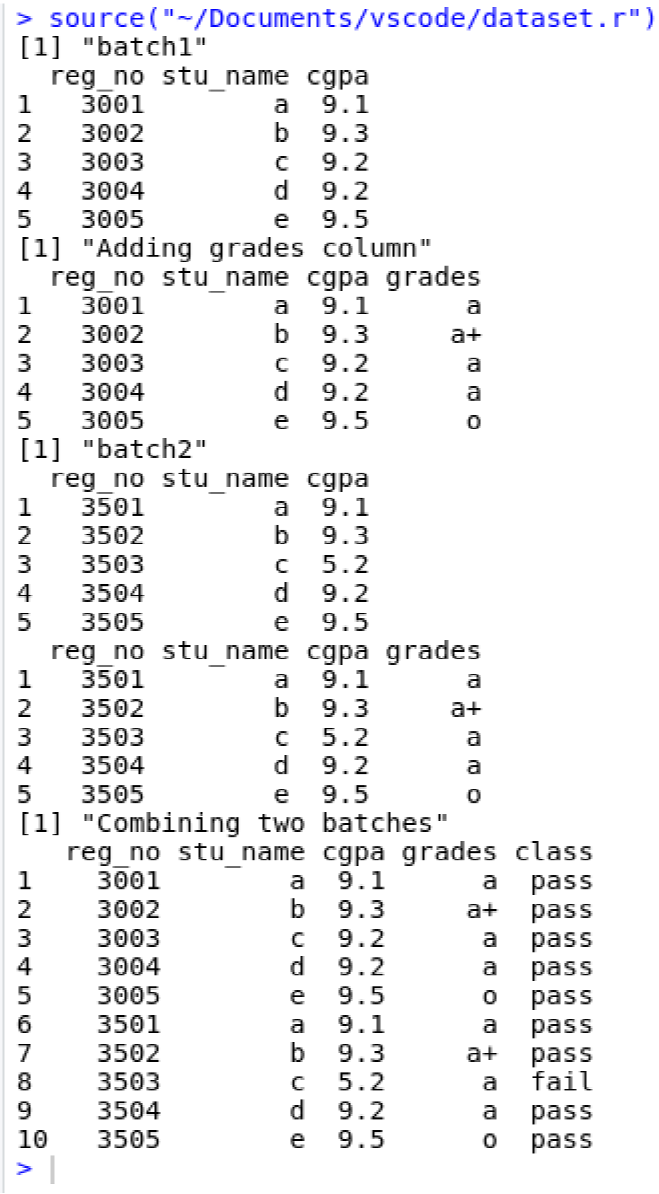
batch_info$class <- as.factor(ifelse(batch_info$cgpa>=6.0,"pass"

,"fail"))

print("Combining two batches")

print(batch_info)

Output:



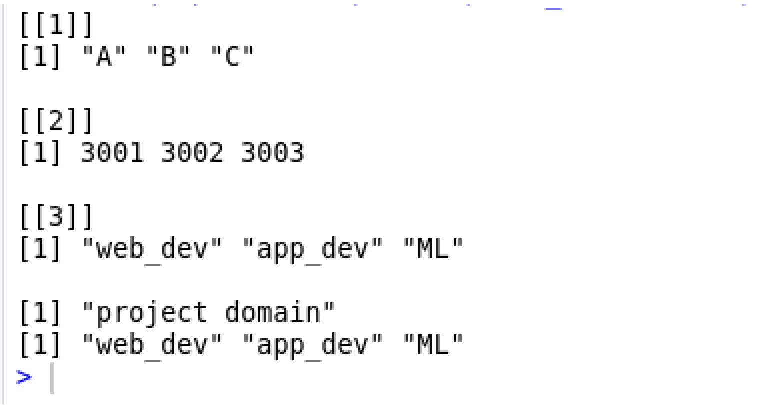
```
> source("~/Documents/vscode/dataset.r")
[1] "batch1"
  reg_no stu_name cgpa
1   3001        a  9.1
2   3002        b  9.3
3   3003        c  9.2
4   3004        d  9.2
5   3005        e  9.5
[1] "Adding grades column"
  reg_no stu_name cgpa grades
1   3001        a  9.1      a
2   3002        b  9.3     a+
3   3003        c  9.2      a
4   3004        d  9.2      a
5   3005        e  9.5      o
[1] "batch2"
  reg_no stu_name cgpa
1   3501        a  9.1
2   3502        b  9.3
3   3503        c  5.2
4   3504        d  9.2
5   3505        e  9.5
  reg_no stu_name cgpa grades
1   3501        a  9.1      a
2   3502        b  9.3     a+
3   3503        c  5.2      a
4   3504        d  9.2      a
5   3505        e  9.5      o
[1] "Combining two batches"
   reg_no stu_name cgpa grades class
1    3001        a  9.1      a  pass
2    3002        b  9.3     a+  pass
3    3003        c  9.2      a  pass
4    3004        d  9.2      a  pass
5    3005        e  9.5      o  pass
6    3501        a  9.1      a  pass
7    3502        b  9.3     a+  pass
8    3503        c  5.2      a  fail
9    3504        d  9.2      a  pass
10   3505        e  9.5      o  pass
> 
```

Que 4 – Create a list of string, numeric with sid, sname, project domain

- Extract only project domain

```
stu_list <- list(c("A","B","C"),c(3001,3002,3003),c("web_dev","app_dev","ML"))

print(stu_list)

names(stu_list) <- c("stu_names","stu_id","proj_domain")

print("project domain")

print(stu_list$proj_domain)
```

Output:



```
[[1]]
[1] "A" "B" "C"

[[2]]
[1] 3001 3002 3003

[[3]]
[1] "web_dev" "app_dev" "ML"

[1] "project domain"
[1] "web_dev" "app_dev" "ML"
>
```

Que 5 – Create a list for eid, ename, designation, salary and company location

- Change designation from manager to ceo
- Append project details between designation and salary
- Remove location information

```
emp_data <-
list(c(1,2,3),c("a","b","c"),c("Manager","Team_Lead","Worker"),c(1000,2000,1500),c("Chennai","Banglore","Chenn
ai"))

names(emp_data) <- c("emp_id","emp_name","emp_desig","emp_sal","emp_loc")

print(emp_data)

# emp_data

emp_data <- lapply(emp_data,function(x) replace(x,x=="Manager","CEO"))

print("changing manger to ceo")

print(emp_data)
```

append(emp_data,c("web","app","ml"),after=3)

print("appending projects")

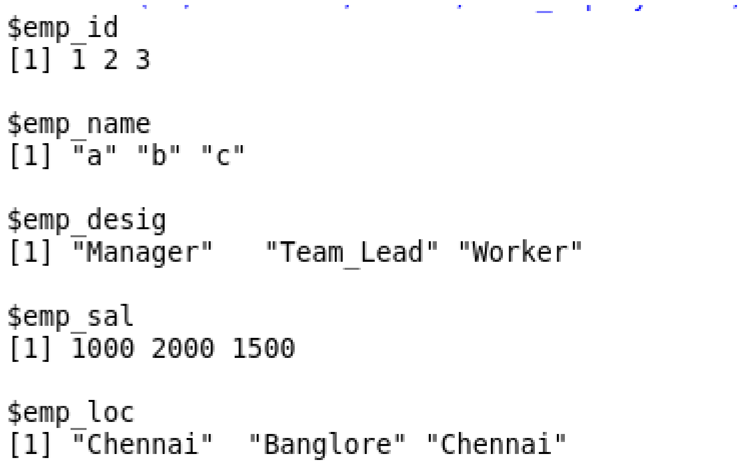
print(emp_data)

emp_data[["emp_loc"]] = NULL

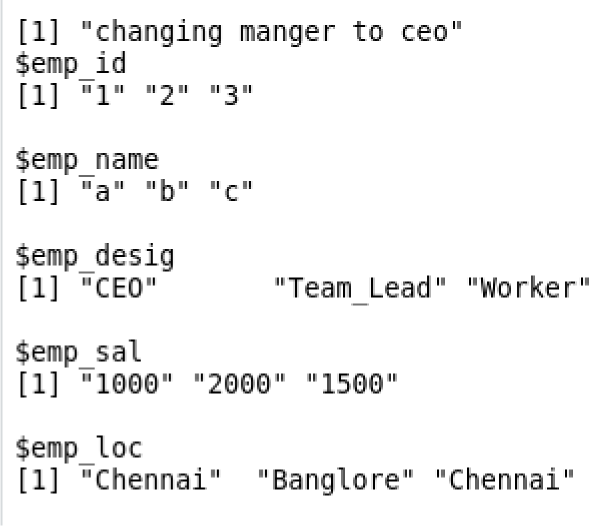
print("removing employee work location")

print(emp_data)

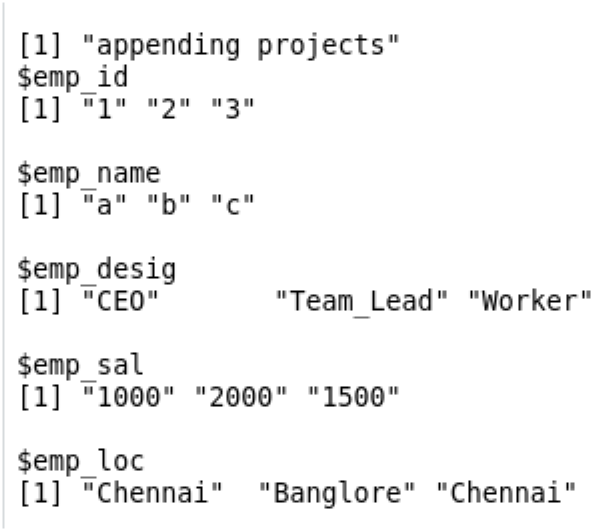
Output:



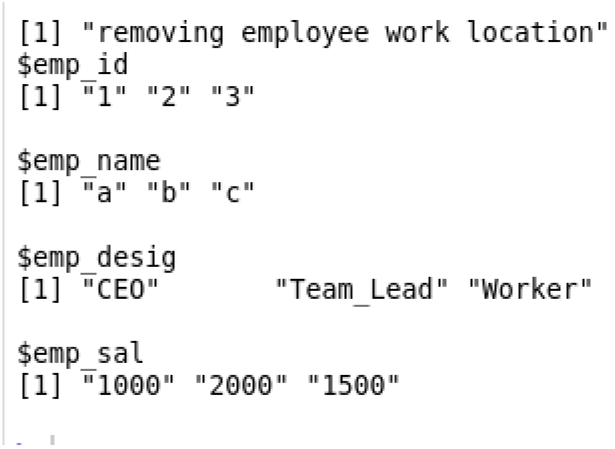
```
$emp_id
[1] 1 2 3

$emp_name
[1] "a" "b" "c"

$emp_desig
[1] "Manager"   "Team_Lead" "Worker"

$emp_sal
[1] 1000 2000 1500

$emp_loc
[1] "Chennai"  "Banglore" "Chennai"
```



```
[1] "changing manger to ceo"
$emp_id
[1] "1" "2" "3"

$emp_name
[1] "a" "b" "c"

$emp_desig
[1] "CEO"       "Team_Lead" "Worker"

$emp_sal
[1] "1000" "2000" "1500"

$emp_loc
[1] "Chennai"  "Banglore" "Chennai"
```



```
[1] "appending projects"
$emp_id
[1] "1" "2" "3"

$emp_name
[1] "a" "b" "c"

$emp_desig
[1] "CEO"       "Team_Lead" "Worker"

$emp_sal
[1] "1000" "2000" "1500"

$emp_loc
[1] "Chennai"  "Banglore" "Chennai"
```



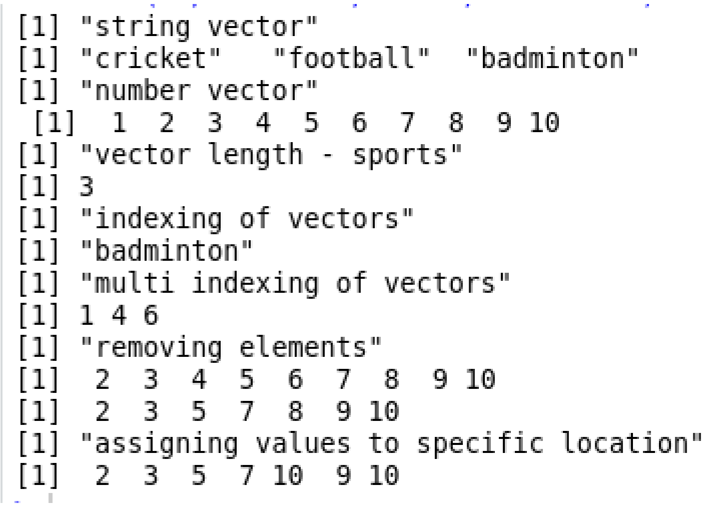
```
[1] "removing employee work location"
$emp_id
[1] "1" "2" "3"

$emp_name
[1] "a" "b" "c"

$emp_desig
[1] "CEO"       "Team_Lead" "Worker"

$emp_sal
[1] "1000" "2000" "1500"
```

Que 6 – create a vector with string values, numeric values and find vector length, extract 3rd element, extract 1,4,6 elements, remove 1st element, remove 1,3,6 elements and change item at 5th index

```
#string values

sports <- c("cricket","football","badminton")

print("string vector")

print(sports)

#numbers

a <- 1:10

print("number vector")

print(a)

#length
```

```
print("vector length - sports")
print(length(sports))
#indexing
print("indexing of vectors")
print(sports[3])
#multi indexing
print("multi indexing of vectors")
print(a[c(1,4,6)])
#remove first ele
print("removing elements")
a <- a[! a%in% c(1)]
print(a)
a <- a[! a%in% c(1,4,6)]
print(a)
print("assigning values to specific location")
a[5]<-10
print(a)
```

output:



```
[1] "string vector"
[1] "cricket"   "football"  "badminton"
[1] "number vector"
 [1]  1  2  3  4  5  6  7  8  9 10
[1] "vector length - sports"
[1] 3
[1] "indexing of vectors"
[1] "badminton"
[1] "multi indexing of vectors"
[1] 1 4 6
[1] "removing elements"
[1]  2  3  4  5  6  7  8  9 10
[1]  2  3  5  7  8  9 10
[1] "assigning values to specific location"
[1]  2  3  5  7 10  9 10
```