21BDS0340

Abhinav Dinesh Srivatsa

Exploratory Data Analysis Lab

Practice Problem Set – I

Experiment 1

Code:

```
# branching with if
x = 5
if (x == 5) {
  print("x is greater than 5")
}
```

Output:

```
[1] "x is greater than 5"
```

Code:

```
# branching with if... else
x = 2
if (x > 3) {
  print("x is greater than 3")
} else {
  print("x is less than or equal to 3")
}
```

Output:

```
[1] "x is less than or equal to 3"
```

Code:

```
# branching with if... else if... else
x = 3
if (x > 3) {
  print("x is greater than 3")
} else if (x < 3) {
  print("x is less than 3")
} else {
  print("x is equal to 3")
}
```

Output:

```
[1] "x is equal to 3"
```

Code:

```
# looping with range
for (i in 1:5) {
  print(i)
```

```
}
```
Output:
```
[1] 1
[1] 2
[1] 3
[1] 4
[1] 5
```

Code:
```
# looping through elements
names = c("abhinav", "tanush", "dinesh", "sumathi")
for (name in names) {
  print(name)
}
```

Output:
```
[1] "abhinav"
[1] "tanush"
[1] "dinesh"
[1] "sumathi"
```

Code:
```
# while loop
x = 0
while (x < 5) {
  print(x)
  x = x + 1
}
```

Output:
```
[1] 0
[1] 1
[1] 2
[1] 3
[1] 4
```

Code:
```
# repeat loop
x = 1
repeat {
  print(x)
  if (x >= 5) {
    break
  }
  x = x + 1
}
```

Output:
```
[1] 1
[1] 2
```

```
[1] 3
[1] 4
[1] 5
```

Code:

```
# branching and looping combo
for (x in 0:10) {
  if (x %% 2 == 0) {
    print(paste(x, "is even"))
  } else {
    print(paste(x, "is odd"))
  }
}
```

Output:

```
[1] "0 is even"
[1] "1 is odd"
[1] "2 is even"
[1] "3 is odd"
[1] "4 is even"
[1] "5 is odd"
[1] "6 is even"
[1] "7 is odd"
[1] "8 is even"
[1] "9 is odd"
[1] "10 is even"
```

Experiment 2

Code:

```
# vectors
num_vec = c(1:5)
char_vec = c("a", "b", "c")
logi_vec = c(TRUE, FALSE, TRUE)

print(num_vec)
print(num_vec[2])
print(char_vec)
print(char_vec[3])
print(logi_vec)
```

Output:

```
> # vectors
> num_vec = c(1:5)
> char_vec = c("a", "b", "c")
> logi_vec = c(TRUE, FALSE, TRUE)
>
> print(num_vec)
[1] 1 2 3 4 5
> print(num_vec[2])
[1] 2
```

```
> print(char_vec)
[1] "a" "b" "c"
> print(char_vec[3])
[1] "c"
> print(logi_vec)
[1]  TRUE FALSE  TRUE
```

Code:

```
# lists
l = list(
  numbers = c(1:3),
  names = c("abhinav", "papaya"),
  is_list = TRUE
)
print(l)
print(l$names)
```

Output:

```
> # lists
> l = list(
+   numbers = c(1:3),
+   names = c("abhinav", "papaya"),
+   is_list = TRUE
+ )
> print(l)
$numbers
[1] 1 2 3

$names
[1] "abhinav" "papaya"

$is_list
[1] TRUE

> print(l$names)
[1] "abhinav" "papaya"
```

Code:

```
# matrices
m = matrix(
  c(1:6),
  nrow = 3,
  ncol = 2
)
print(m)
print(m[3,2])
```

Output:

```
> # matrices
> m = matrix(
+   c(1:6),
```

```
+   nrow = 3,
+   ncol = 2
+ )
> print(m)
     [,1] [,2]
[1,]    1    4
[2,]    2    5
[3,]    3    6
> print(m[3,2])
[1] 6
```

Code:

```
# data frame
df = data.frame(
  id = 1:3,
  fruit = c("pineapple", "papaya", "mango"),
  age = c(1, 2, 1)
)
print(df)
print(df$age)
print(df[df$age > 1, ])
```

Output:

```
> # data frame
> df = data.frame(
+   id = 1:3,
+   fruit = c("pineapple", "papaya", "mango"),
+   age = c(1, 2, 1)
+ )
> print(df)
  id     fruit age
1  1 pineapple   1
2  2    papaya   2
3  3     mango   1
> print(df$age)
[1] 1 2 1
> print(df[df$age > 1, ])
  id  fruit age
2  2 papaya   2
```

Code:

```
# factors
colors = factor(c("red", "blue", "green", "blue", "red"))
print(levels(colors))
print(colors)
```

Output:

```
> # factors
> colors = factor(c("red", "blue", "green", "blue", "red"))
> print(levels(colors))
[1] "blue"  "green" "red"
```

```
> print(colors)
[1] red   blue  green blue  red
Levels: blue green red
```

Code:

```
# arrays
arr = array(
  c(1:12),
  dim = c(3, 2, 2)
)
print(arr)
print(arr[1,1,1])
```

Output:

```
> # arrays
> arr = array(
+   c(1:12),
+   dim = c(3, 2, 2)
+ )
> print(arr)
, , 1

     [,1] [,2]
[1,]    1    4
[2,]    2    5
[3,]    3    6

, , 2

     [,1] [,2]
[1,]    7   10
[2,]    8   11
[3,]    9   12

> print(arr[1,1,1])
[1] 1
```