

**AFARI JESSE**

**P191301**

**LAB THREE**

## **#VARIABLES**

#Variable can be assigned in more ways

```
var.1=c(1:10)
```

```
var.2<-c("Jesse","Aleena")
```

```
c("P191301","p191302")->var.3
```

```
var.4<-data.frame(var2,var3)
```

```
var.5<-c(TRUE, 1)
```

```
var_6<-list(name="Kapil",Age="40")
```

```
var_7<-matrix(1:8,nrow = 3)
```

```
var_8<-c('yes','no')
```

## **OUTPUT**

```
> var.1
[1] 1 2 3 4 5 6 7 8 9 10
> var.2
[1] "Jesse" "Aleena"
> var.3
[1] "P191301" "p191302"
> var.4
  var2    var3
1 Jesse P191301
2 Aleena p191302
> var.5
[1] 1 1
> var_6
$name
[1] "Kapil"

$Age
[1] "40"

> var_7
  [,1] [,2] [,3]
[1,]  1   4   7
[2,]  2   5   8
[3,]  3   6   1
> var_8
[1] "yes" "no"
> |
```

## #DATATYPES OF VARIABLE

```
class(var.1)
```

```
class(var.2)
```

```
class(var.3)
```

```
class(var.4)
```

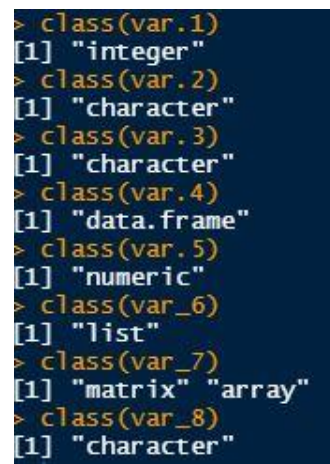
```
class(var.5)
```

```
class(var_6)
```

```
class(var_7)
```

```
class(var_8)
```

## OUTPUT



```
> class(var.1)
[1] "integer"
> class(var.2)
[1] "character"
> class(var.3)
[1] "character"
> class(var.4)
[1] "data.frame"
> class(var.5)
[1] "numeric"
> class(var_6)
[1] "list"
> class(var_7)
[1] "matrix" "array"
> class(var_8)
[1] "character"
```

## #TO FIND ALL VARIABLES ASSIGNED IN THE IDE

```
ls()
```

```
ls(pattern = "var")
```

#To find the variables that have var

```
ls(pattern = "var")
```

#to find variables with "."

```
ls(all.name = TRUE)
```

## OUTPUT

```

> ls()
[1] "a"      "a1"     "a2"     "a3"     "a4"     "b"      "var.1"  "var.2"  "var.3"  "var.4"
[11] "var.5"  "var_6"  "var_7"  "var_8"  "var1"   "var2"   "var3"   "var6"   "var7"
> ls(pattern = "var")
[1] "var.1"  "var.2"  "var.3"  "var.4"  "var.5"  "var_6"  "var_7"  "var_8"  "var1"   "var2"
[11] "var3"   "var6"   "var7"
> #to find the variables that have var
> ls(pattern = "var")
[1] "var.1"  "var.2"  "var.3"  "var.4"  "var.5"  "var_6"  "var_7"  "var_8"  "var1"   "var2"
[11] "var3"   "var6"   "var7"
> #to find variables with "."
> ls(all.name = TRUE)
[1] ".Random.seed" "a"      "a1"     "a2"     "a3"
[6] "a4"           "b"      "var.1"  "var.2"  "var.3"
[11] "var.4"        "var.5"  "var_6"  "var_7"  "var_8"
[16] "var1"         "var2"   "var3"   "var6"   "var7"

```

## #TO DELETE A VARIABLE

?rm

rm(var8)

var8

## OUTPUT

```

> var_8<-c('yes','no')
> rm(var_8)
> var_8
Error: object 'var_8' not found

```

## #RELATIONAL OPERATIONS

a<-10

b<-90

## #CHECK IS GREATER OR LESS

b>a

b<a

a1<-c(1:4)

a2<-c(5:2)

a1>a2

## #CHECK IS THEY ARE EQUAL

```
a3<-c(1,2,4,5)
```

```
a3==a1
```

## #CHECK IF THEY ARE GREATER OR EQUAL

```
a2>=a3
```

## #CHECK IF THEY ARE LESS OR EQUAL

```
a2<=a3
```

## #NOT EQUAL

```
a1!=a2
```

## OUTPUT

```
> a<-10
> b<-90
> #check is greater or lesse
> b>a
[1] TRUE
> b<a
[1] FALSE
>
> a1<-c(1:4)
> a2<-c(5:2)
> a1>a2
[1] FALSE FALSE FALSE TRUE
>
> #check is they are equal
> a3<-c(1,2,4,5)
> a3==a1
[1] TRUE TRUE FALSE FALSE
>
> #check if they are greater or equal
> a2>=a3
[1] TRUE TRUE FALSE FALSE
> #check if they are less or equal
> a2<=a3
[1] FALSE FALSE TRUE TRUE
>
> #not equal
> a1!=a2
[1] TRUE TRUE FALSE TRUE
```

## #LOGICAL OPERATOR

#logical operators are used only on vectors of type logical or numeric

#Number > 1 are deemed true and <1 false

**#USING & COMPARES THE ELEMENTS IN BOTH VECTORS AND RETURNS THE NUMBERS THAT ARE TRUE OR FALSE WHEN COMPARED**

## #AND OPERATOR

```
a1&a3
```

```
a4<-c(0,1,2,9)
```

```
a1&a4
```

## #OR OPERATOR

```
a4|a1
```

**#NOT OPERATOR GIVE THE OPPOSITE RESULT**

```
!a4
```

**#LOGICAL AND OPERATOR AND LOGICAL OR OPERATOR.**

a4&&a1 #it takes the first element of both the vectors and gives the true only if both are true

a4||a1 #it takes first element of both the vectors and gives the True if one of them is true.

## OUTPUT

```
> #AND Operator
> a1&a3
[1] TRUE TRUE TRUE TRUE
> a4<-c(0,1,2,9)
> a1&a4
[1] FALSE TRUE TRUE TRUE
> #OR Operator
> a4|a1
[1] TRUE TRUE TRUE TRUE
>
> #not operator give the opposite result
> !a4
[1] TRUE FALSE FALSE FALSE
>
> a4&&a1
[1] FALSE
> a4||a1
[1] TRUE
```

## #ASSIGNMENT OPERATORS ALREADY USED AT THE START

## #MISCELLANEOUS OPERATORS

#colon

c(1:4)#creates series of number in sequence

9%in% used to identify element in vectors

9%in%a4

##\*% this operator is used to multiply a matrix with its transpose

## OUTPUT

```
> #colon
> c(1:4)#creates series of number in sequence
[1] 1 2 3 4
>
> 9%in% used to identify element in vectors
> 9%in%a4
[1] TRUE
>
> #used to multiply a matrix with its transpose
> a1%% t(a1)
      [,1] [,2] [,3] [,4]
[1,]    1    2    3    4
[2,]    2    4    6    8
[3,]    3    6    9   12
[4,]    4    8   12   16
```