

## WEEK - 3

- How to work with data in R?
  - stored in R [data frame]
  - Read in data into R
- Simulate samples from a given distribution.

## Recall - Week 1 and 2

- How to work with R and Rstudio?
- Data Visualisation :-
  - ggplot
  - plot ..

# ggplot2 - Recall layered grammar of graphics

Each template takes 7 statements / Parameters

```
ggplot(data = <DATA>)  
+ <GEOM_FUNCTION>(mapping = aes(<MAPPINGS>),  
stat = <STAT>,  
position = <POSITION>  
) +  
<COORDINATE_FUNCTION> +  
<FACET_FUNCTION>
```

Coordinate chart

aesthetics

.. geom\_point

stat - ...

dodge

coord\_flip  
... polar coordinates

faceting

# Data Types in R

R has many data types  
- Focus on three

```
> Course = "B.Sc."  
> Number = 40  
> Smart = TRUE  
> mode(Course)  
[1] "character"  
> mode(Number)  
[1] "numeric"  
> mode(Smart)  
[1] "logical"
```

• Character data = Surrounded by double quotes

• logical data = TRUE or FALSE

• Another important characteristic is class. For this we can use the class function.

# Creating Vectors in R

— stored as one variable / collection of objects in the same mode

```
> x = 3:7
```

```
> x
```

```
[1] 3 4 5 6 7
```

```
> s = seq(1,10, by=1)
```

```
> s
```

```
[1] 1 2 3 4 5 6 7 8 9 10
```

```
> s10 = seq(1,10,by=0.5)
```

```
> s10
```

```
[1] 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0 5.5 6.0 6.5 7.0 7.5 8.0
```

```
[16] 8.5 9.0 9.5 10.0
```

```
> rep(6,7)
```

```
[1] 6 6 6 6 6 6 6
```

```
> rep(x,3)
```

```
[1] 3 4 5 6 7 3 4 5 6 7 3 4 5 6 7
```

— Useful to have variables in R

— to recall the values in variable

— create a vector using `seq(..)`

check help for options

Position of 8.5 is 16 in the vector

repeats the specified element in the first argument **6**, as many times as the second argument **7**.

first argument can be a vector.

Vectors in R: c - combine / concatenate individual values to get a vector.

```
> KA_D = c(215,620,558, 1109,8813,350, 780, 420,  
+          144,478,816,242,1051,249,1238, 315,  
+          807,185,1993,515,1997,2886,371,156,  
+          589,1746,838,964,296,128)
```

Distinctwise

Discharge data of  
COVID-19 patients in  
Karnataka  
on 31-01-2022

```
> KA_D[c(1,3,5)]
```

```
[1] 215 558 8813
```

Select elements 1,3,5

```
> KA_D[-c(1:20)]
```

```
[1] 1997 2886 371 156 589 1746 838 964 296 128
```

Removed elements 1:20

```
> KA_Dp = (KA_D)/sum(KA_D)*100
```

```
> KA_Dp
```

Vector of percentage  
of discharge across districts.

```
[1] 0.6964916 2.0084875 1.8076387 3.5926010 28.5496777 1.1338236
```

```
[7] 2.5268068 1.3605883 0.4664874 1.5484791 2.6434287 0.7839580
```

```
[13] 3.4047102 0.8066345 4.0104960 1.0204412 2.6142732 0.5993067
```

```
[19] 6.4563154 1.6683404 6.4692734 9.3491853 1.2018530 0.5053614
```

```
[25] 1.9080631 5.6561599 2.7146976 3.1228741 0.9588908 0.4146555
```

# Vectors in R

```
> KA_D[KA_D < 1000]
```

```
[1] 215 620 558 350 780 420 144 478 816 242 249 315 807 185 515 371 156 589 838  
[20] 964 296 128
```

logical operator to select  
elements of the vector.

- choose all elements that  
return TRUE.

```
> sum(KA_D > 2000)
```

```
[1] 2
```

- Converts logical operator to  
numeric and then adds.

```
> max(KA_D)
```

```
[1] 8813
```

maximum of the vector

```
> which(KA_D == max(KA_D))
```

```
[1] 5
```

position of the maximum

# Missing Values in R

- when we collect data, there may be missing values.

```
> x = c(1,45,6,7,NA,99,0)
```

```
> x
```

```
[1] 1 45 6 7 NA 99 0
```

```
> x==NA
```

```
[1] NA NA NA NA NA NA NA
```

```
> mean(x)
```

```
[1] NA
```

```
> mean(x,na.rm=TRUE)
```

```
[1] 26.33333
```

```
> is.na(x)
```

```
[1] FALSE FALSE FALSE FALSE TRUE FALSE FALSE
```

```
> mean(x[!is.na(x)])
```

```
[1] 26.33333
```

- Missing values are represented by NA

- any computation with a missing value will return a missing value.

- { Cannot test equality to a missing value with ==

remove missing values before computing anything.

- { test if a value is missing or not

logical operator ! and is.na is used to select elements that are not missing values.

# Modes in R

```
> x = c("Siva", "looser", "3", "5")  
> mode(x)  
[1] "character"  
> #x[3] + x[4]  
> as.numeric(x[3]) + as.numeric(x[4])  
[1] 8
```

numbers have been changed  
to characters

Cannot perform  
Computation —  
non-numeric argument to  
binary operator

Treat it as  
numeric



Matrices in R  $\equiv$  all entries have to be in the same mode

`> A = matrix(seq(3,5, by=1/10), 7,3)`  
*vector of entries*

`> A`

	[,1]	[,2]	[,3]
[1,]	3.0	3.7	4.4
[2,]	3.1	3.8	4.5
[3,]	3.2	3.9	4.6
[4,]	3.3	4.0	4.7
[5,]	3.4	4.1	4.8
[6,]	3.5	4.2	4.9
[7,]	3.6	4.3	5.0

*number of columns*  
*number of rows*

*fill entries column wise*

`> B = matrix(seq(3,5, by=1/10), ncol=3)`  
`> C = matrix(seq(3,5, by=1/10), ncol=3, byrow=TRUE)`  
*fill entries row wise*

`> A[4,1]`  
`[1] 3.3`  
*select the element at Row = 4  
Column = 1*

# Data Frames in R

- Data frame is like a matrix / rectangular Array
- Each column in a data frame can be in a different mode

```
> xd = c("Siva", "looser", 3, 5)
```

```
> xd
```

```
[1] "Siva"    "looser"  "3"       "5"
```

} all elements of a vector / matrix have to be in the same mode

Dataframe - best way to store data in R

# Data Frames in R — Our first creation

Created a vector with (corresponding) names of districts.

```
> KA_District=c("Bagalakote","Ballari","Belagavi","Bengaluru Rural","Bengaluru Urban"  
+ "Bidar","Chamarajanagara","Chikkaballapura","Chikkamagaluru","Chitradurga",  
+ "Dakshina Kannada","Davanagere","Dharwada","Gadag","Hassana","Haveri",  
+ "Kalaburagi","Kodagu","Kolar","Koppala","Mandya","Mysuru","Raichuru",  
+ "Ramanagara","Shivamogga","Tumakuru","Udupi","Uttara Kannada","Vijayapura"  
+ , "Yadagiri"  
+ )
```

Creates a data frame

```
> KA_Discharge = data.frame(KA_District, KA_D)
```

Discharge data.

```
> class(KA_Discharge)
```

District names

```
[1] "data.frame"
```

Specifies as data frame

```
> mode(KA_Discharge)
```

```
[1] "list"
```

```
> sapply(KA_Discharge, mode) ←
```

```
KA_District      KA_D  
"character"      "numeric"
```

Simplifies  
"loop function"

... applies the  
mode function

to each  
variable / column in  
data frame.

# Data Frames as Matrix in R - Working with them.

Change the names of the variables in the data frame

```
> names(KA_Discharge)=c("District", "Recovered")
```

```
> KA_Discharge$Recovered
```

```
[1] 215 620 558 1109 8813 350 780 420 144 478 816 242 1051 249 1238  
[16] 315 807 185 1993 515 1997 2886 371 156 589 1746 838 964 296 128
```

```
> KA_Discharge[3,2]
```

```
[1] 558
```

```
> KA_Discharge[3,]
```

```
District Recovered  
3 Belagavi 558
```

```
> KA_Discharge[, "Recovered"]
```

```
[1] 215 620 558 1109 8813 350 780 420 144 478 816 242 1051 249 1238  
[16] 315 807 185 1993 515 1997 2886 371 156 589 1746 838 964 296 128
```

Select :- objects, Rows and columns from Data frame

# Data Frames as Matrix in R

- add another variable to data frame

```
> Deaths= c(346, 1712, 975, 903, 16593, 407, 515, 446, 400, 221, 1750, 611,  
+ 1333, 328, 1291, 652, 856, 343, 647, 530, 673, 2494, 346,  
+ 338, 1105, 1172, 509, 793, 500, 206  
+ )  
> KA_Discharge$Deaths = Deaths  
> head(KA_Discharge)
```

	District	Recovered	Deaths
1	Bagalakote	215	346
2	Ballari	620	1712
3	Belagavi	558	975
4	Bengaluru Rural	1109	903
5	Bengaluru Urban	8813	16593
6	Bidar	350	407

- Creates a variable "Deaths" in the data frame "KA\_Discharge"

- fill the entries from the vector Deaths.