E-Content on Data Manipulation and Visualization through Statistical Software R (Open Source Software)

Dr Gurpreet Singh Tuteja and Dr Dhiraj Kumar Singh
Department of Mathematics
Zakir Husain Delhi College
(University of Delhi)
Jawaharlal Nehru Marg, Delhi - 110002

Day 5: July 31, 2020

1 Session 5

Today, we are going to have a hands-on practice on real data of COVID19 of all those concepts which we learnt in last four sessions.

1.1 Covid19 Data

There are various websites which provode daily updated data on Covid. We have selected a simple database with 6 variables from the website: https: datahub.io .
The dataframe Covid is populated with the latest data: Covid ¡- read.csv("https://datahub.io/core/covid-19/r/time-series-19-covid-combined.csv")
The fields can be observed using command:
names(Covid)
1] "Date" "Country.Region" "Province.State" "Confirmed" "Recovered" "Deaths"
unique(Country.Region) displays the list of countries which has 189 entries.
str(Covid)

```
> str(Covid)
'data.frame':   76112 obs. of  6 variables:
 $ Date          : chr  "2020-01-22" "2020-01-23" "2020-01-24" "2020-01-25" ...
 $ Country.Region: chr  "Afghanistan" "Afghanistan" "Afghanistan" "Afghanistan" ...
 $ Province.State: chr  "" "" "" "" ...
 $ Confirmed     : int  0 0 0 0 0 0 0 0 0 0 ...
 $ Recovered     : int  0 0 0 0 0 0 0 0 0 0 ...
 $ Deaths        : int  0 0 0 0 0 0 0 0 0 0 ...
```

1.2 Lag and Lead

Find the "previous" (lag()) or "next" (lead()) values in a vector. It is useful for comparing values behind of or ahead of the current values. Consider a vector a with 6 elements:
a ← c(1,2,3,4,5,6)
b ← lag(a)
The output is:
[1] NA 1 2 3 4 5

c ← lead(a)
[1] 2 3 4 5 6 NA

Value used for non-existent element defaults to NA.
Now, we use this to find the ratio today's deaths and yesterday's deaths. Similarly, we can find the ratio of today's and yesterday's Confirmed.Begore doing this, we need to remove all NA's from the columns:
$Death1 \leftarrow c(lag(Deaths))$
$Confirmed1 \leftarrow c(lag(Confirmed))$
$Death1[is.na(Death1)] = 0$
$Confirmed1[is.na(Confirmed1)] = 0$
These two new columns Death1 and Confirmed1 are binded with the dataframe df using bind command:
df ← as.data.frame(cbind(Covid,Confirmed1,Death1))
View(df)

	Date	Country	Confirmed	Recovered	Deaths	Confirmed1	Death1
1	2020-01-22	Afghanistan	0	0	0	0	0
2	2020-01-23	Afghanistan	0	0	0	0	0
3	2020-01-24	Afghanistan	0	0	0	0	0
4	2020-01-25	Afghanistan	0	0	0	0	0
5	2020-01-26	Afghanistan	0	0	0	0	0
6	2020-01-27	Afghanistan	0	0	0	0	0
7	2020-01-28	Afghanistan	0	0	0	0	0
8	2020-01-29	Afghanistan	0	0	0	0	0
9	2020-01-30	Afghanistan	0	0	0	0	0
10	2020-01-31	Afghanistan	0	0	0	0	0

Showing 1 to 11 of 76,380 entries, 7 total columns

We, will study the data of India, so we use select command which we learnt in the last session:
dfc ← filter(df, Country==”India”)
dfc ← filter(dfc, Deaths!=0 & Death1!=0 & $Confirmed1! = 0$ & Confirmed!=0 & Recovered!=0)
View(dfc)

	Date	Country	Confirmed	Recovered	Deaths	Confirmed1	Death1
1	2020-01-22	Afghanistan	0	0	0	0	0
2	2020-01-23	Afghanistan	0	0	0	0	0
3	2020-01-24	Afghanistan	0	0	0	0	0
4	2020-01-25	Afghanistan	0	0	0	0	0
5	2020-01-26	Afghanistan	0	0	0	0	0
6	2020-01-27	Afghanistan	0	0	0	0	0
7	2020-01-28	Afghanistan	0	0	0	0	0
8	2020-01-29	Afghanistan	0	0	0	0	0
9	2020-01-30	Afghanistan	0	0	0	0	0
10	2020-01-31	Afghanistan	0	0	0	0	0

Showing 1 to 11 of 76,380 entries, 7 total columns

We add another column Days which is difference of current day and first day.
dfc ← mutate(dfc,Date,
Days=as.Date(Date)-as.Date(Date[1]),
RDeath=as.numeric(Deaths)/as.numeric(Death1),
RConfirmed=as.numeric(Confirmed)/as.numeric(Confirmed1))
View(dfc)

Filter

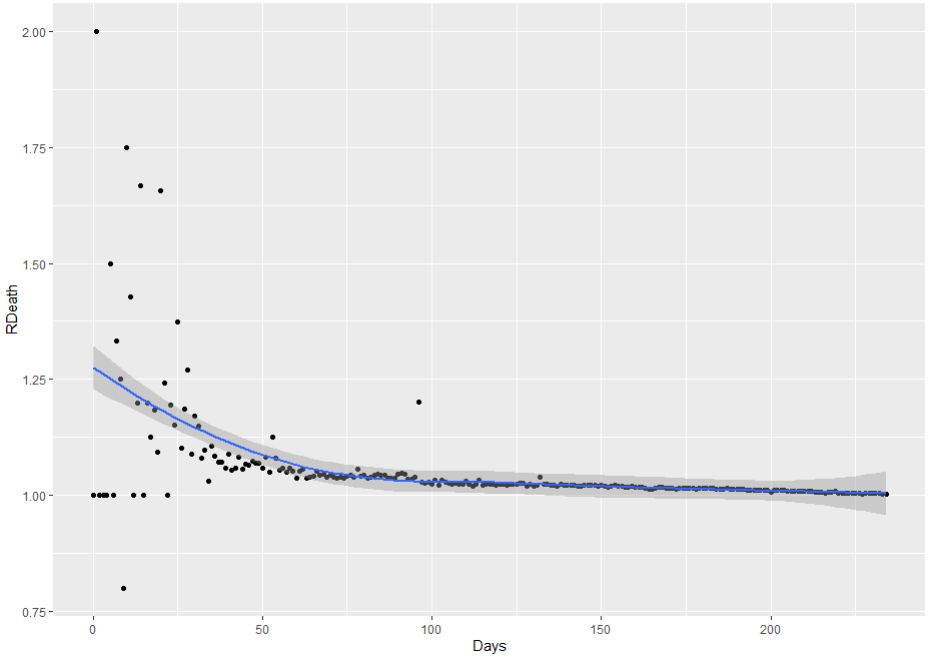
	Date	Country	Confirmed	Recovered	Deaths	Confirmed1	Death1	Days	RDeath	RConfirmed
1	2020-03-12	India	73	4	1	62	1	0	1.000000	1.177419
2	2020-03-13	India	82	4	2	73	1	1	2.000000	1.123288
3	2020-03-14	India	102	4	2	82	2	2	1.000000	1.243902
4	2020-03-15	India	113	13	2	102	2	3	1.000000	1.107843
5	2020-03-16	India	119	13	2	113	2	4	1.000000	1.053097
6	2020-03-17	India	142	14	3	119	2	5	1.500000	1.193277
7	2020-03-18	India	156	14	3	142	3	6	1.000000	1.098592
8	2020-03-19	India	194	15	4	156	3	7	1.333333	1.243590
9	2020-03-20	India	244	20	5	194	4	8	1.250000	1.257732
10	2020-03-21	India	330	23	4	244	5	9	0.800000	1.352

1.25773

Showing 1 to 11 of 235 entries, 10 total columns

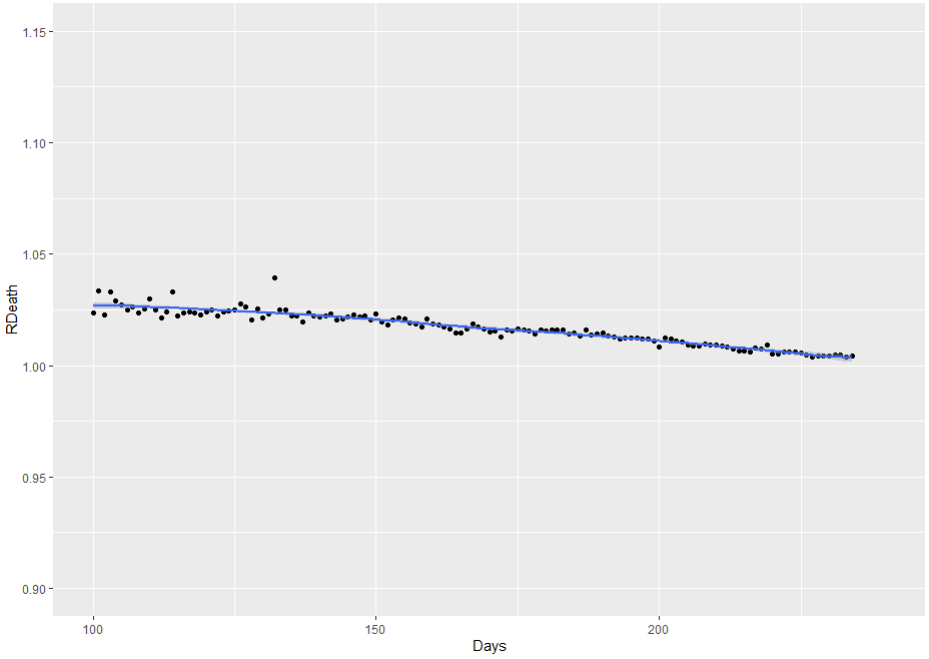
1.3 Plotting Covid Death and Confirmed Ratios

ggplot(dfc,aes(Days,RDeath))+
geom_point()+
geom_smooth(method="auto")+
View(dfc)

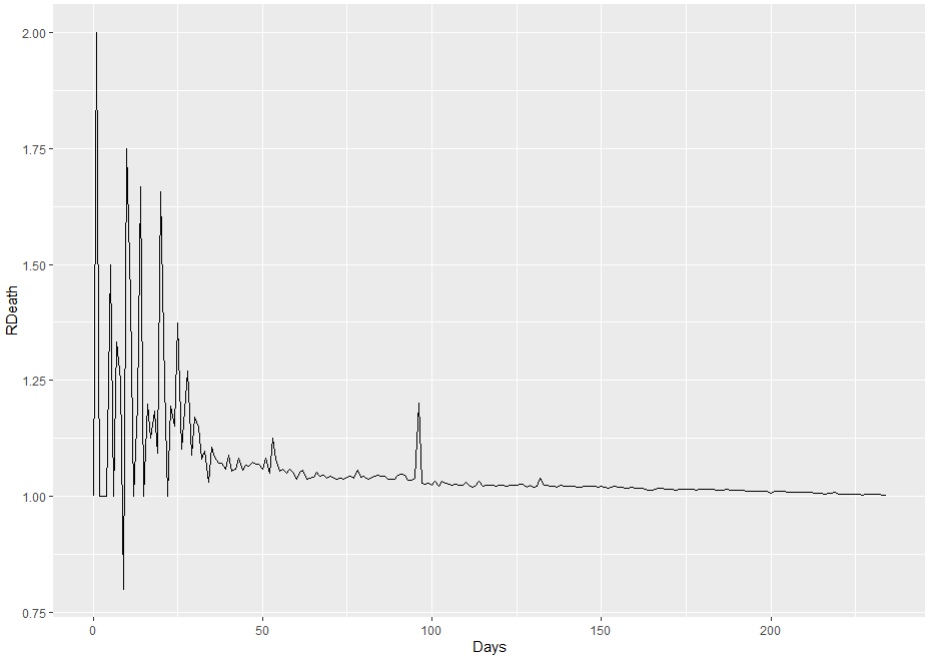


We can limit our graph for certail values of x-axia and y-axis, say we wish to observe after 100 days of first death, the ratio is decreasing or not, we put

limits in aobe graph:
ggplot(dfc,aes(Days,RDeath))+
geom_point()+
geom_smooth(method="auto")+
xlim(100,240)+
ylim(0.9,1.15)

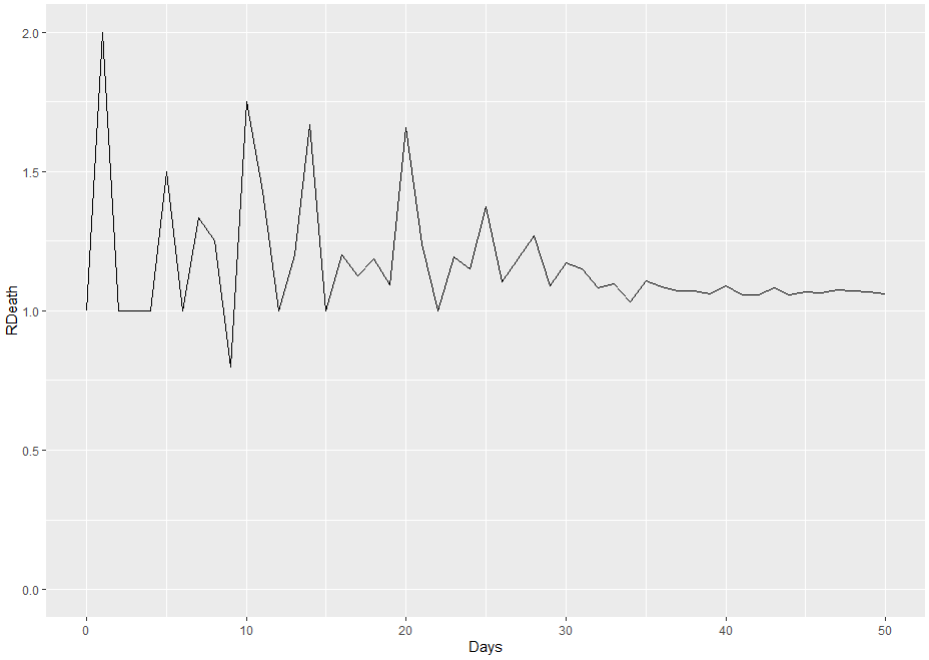


If we wish to see the actual movement of the death ration day-wise then we plot:

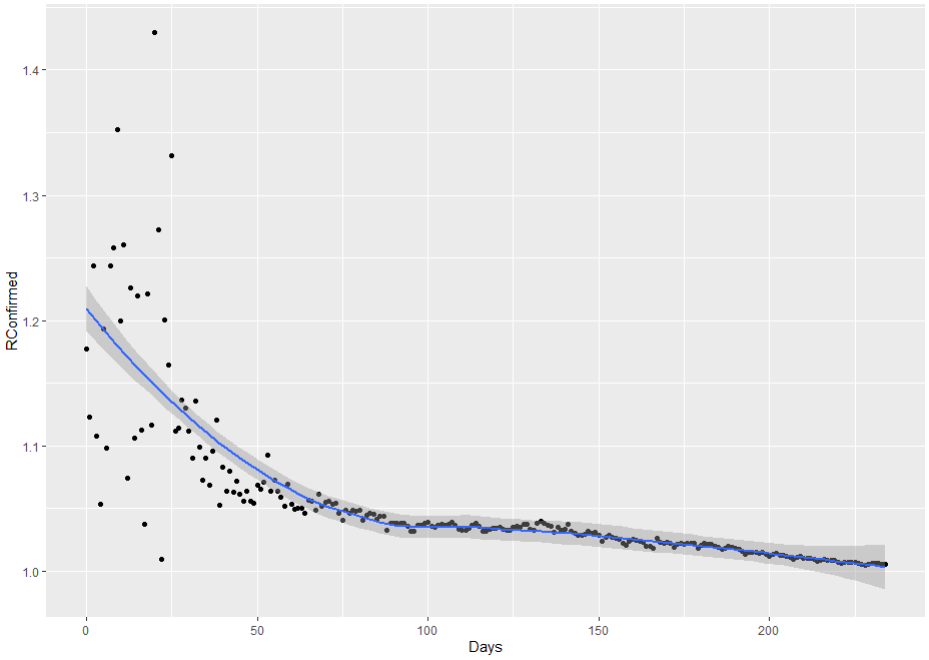


Let, blowup the graph for first 50 days, where the ratio was oscillating too much

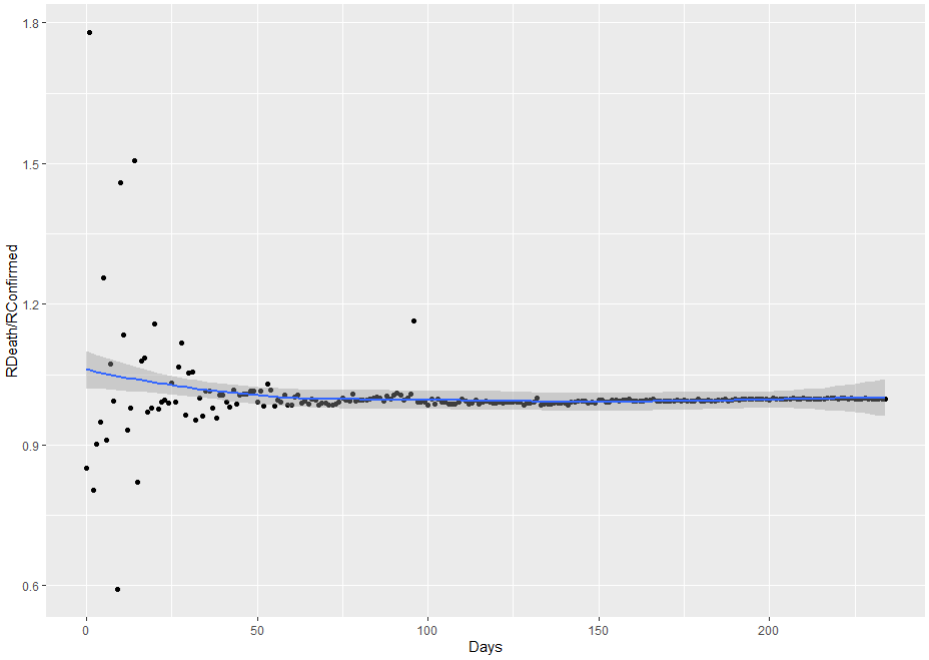
```
ggplot(dfc,aes(Days,RDeath))+
geom_path()+
xlim(0,50)+
ylim(0,2)
```



Now, we will be plotting graphs of Confirmed ratio
ggplot(dfc,aes(Days,RConfirmed))+
geom_point()+
geom_smooth(method="auto")



The death vs Confirmed ratio is given as: ggplot(dfc,aes(Days,RDeath/RConfirmed))+
geom_point()
geom_smooth(method="auto")+



This ends the final session of the Workshop. We thank DBT-Star College Scheme, Department of Biotechnology for making this programme to happen. The Quiz is available, kindly complete it in 15 minutes.