[Q.N.1] Import this JSON API data "https://data.askbhunte.com/api/v1/covid" using "jsonlite" or similar package.

Solution: I use the packages,

- `jsonlite` :It implements a bidirectional mapping between JSON data and the most important R data types.
- `RCurl` : Which help to provides the necessary tools for accessing URIs, data and services via HTTP.

```
install.packages('RCurl')
```

```
library(jsonlite)
library(RCurl)
url <- getURL("https://data.askbhunte.com/api/v1/covid",
.opts=list(followlocation=TRUE, ssl.verifyhost=FALSE,
ssl.verifypeer=FALSE))
```

[Q.N.2] Save the imported data as "covidtbl" data frame in R (Hint: use "flatten = TRUE" after 'fromJSON' function availble in jsonlite package).

Solution:

```
covidtbl <- fromJSON(txt=url, flatten=TRUE)
covidtbl
```

[Q.N.3]Check whether the saved "covidtbl" data passes the three conditions of the "tidy" data or not! If not, make it a tidy data with explanations of rows, columns and cells in the data.

Solution: The three conditions of tidy data are

- Each variable must have its own column
- Each observation must have its own row
- Each value must have its own cell.

Apply these condition to our covidtbl data set satisfy all the conditions mention above so, without a doubt a our dataset is tidy data.

```
covidtbl
```

[Q.N.4]Check if there are duplicate cases in the data using "id" variable, remove duplicate cases, if found, using R base functions: duplicated or unique (read about it and use it carefully).

Solution: Function `duplicated()` ensure that wether there is present of duplicated value or not. If duplicate value is found it return TRUE other wise FALSE. So, in our care there no present of duplicated values.

similarly function `unique()` is used to erase the duplicated value or to find the unique value in dataframe.

```
duplicated(covidtbl)
```

```
unique(covidtbl)
```

[Q.N.5]Clean the "gender" variable and show the number and percentage of males and females in the data (Hint: you can use table function and prop.table functions)

Solution: We must take care of following points while cleaning our data,

- Free of duplicate rows/values
- Error-free (e.g. free of misspellings)
- Relevant (e.g. free of special characters)
- The appropriate data type for analysis
- Free of outliers (or only contain outliers have been identified/understood), and
- Follows a “tidy data” structure

In case of our data there are some missing value as well as starting letter of some data are in small letter or in capital letter. To slove this problem I install the package

`stringer` : It is used to convert the first letter of every word of a string to Uppercase and the rest of the letters are converted to lower case.

```
install.packages("stringr")
```

Following code load the stringr library and remove the missing values present in our gender column.

```
df<- covidtbl[complete.cases(covidtbl$gender),]
```

In following case, first load the string library and then change all the gender data in upper case and by using `str_to_title()` function change all letter except first to lower case.

```
library(stringr)
df <-toupper(df$gender)
df <- str_to_title(df)
```

Now, it is time to calculate total number of male and female from gender column.

```
df1<- table(df)
df1
```

```
df
Female   Male
 21403  56355
```

To find the percentage of male and female we can do,

```
prop.table(df1)
```

```
df
   Female      Male
0.2752514 0.7247486
```

[Q.N.6] Clean the "age" variable and show the summary statistics of the age variable and interpret the results carefully (Search the online newspapers and clean the age data, if required)

Solution: In our age column there are so many missing value. First we should remove the missing values.

```
df <- table(covidtbl$age)
df_age <- covidtbl[complete.cases(covidtbl$age),]
age <- df_age$age
age
```

```
   [1]  28  34  26  29  20  72  60  24  58  52  41  41  34  41  28  28
28  22
  [19]  25  40  33  18  65  32  20  37  55  36  55  19  65  21  34  65
81  19
  [37]  27  32  57  26  44  35   9  32  34  50  25  62  21  34  40  55
60  30
  [55]  55  29  65  80  17  60  26  27  58  60  52  30  65  37   4  35
11  40
  [73]  22  24  65  32  18  18  38  45  63  30  25  27  28  28  45  32
60  16
  [91]  61  22  24  25  28  28  32  59  55  61  22  28  32  32   4  10
28  20
 [109]  35  22  40  18  36  32  27  25  25  22  49  36  29  20  28  23
30  36
 [127]  75  36   9  23  47  10  28  29  65  17  18  18  30  36  37  26
25  25
 [145]  32  25  48  39  48  28  30  33  21  25  25  29  42  21  30  22
18  19
 [163]  19  22  23  20  28  30  40  27  42  41  51  27  37  26  26  28
30  46
 [181]  17  18  53  55  22  22  45  74  19  35  34  35  24  30  27  34
19  38
```

 [199] 45 27 32 34 35 38 38 39 18 23 20 22 23 26 35 16
33 22
 [217] 20 21 52 29 69 32 45 48 26 32 39 36 40 37 19 22
27 30
 [235] 31 31 45 34 33 35 35 35 36 36 38 40 42 43 48 48
48 49
 [253] 50 51 53 16 23 23 24 25 19 20 23 31 70   7 62 36
32 36
 [271] 40 50 30 26 22 18 24 35 18 28 36 19 18 22 17 35
22 24
 [289] 22 41 29 20 40 46 38 29 35 25 30 19 18 23 35 41
35 35
 [307] 27 25 60   4 47 13 45 50 25   8 28   2   5 26 25 56
34 35
 [325] 41 15 60 28 27 40 35   2 32 23 34   4   9   1 40 19
20 34
 [343] 45 17 28 18 20 50 30 18 13 22 36 14 17 30 23 36
20 26
 [361] 40 31 38 25 60   1   2 11 28 25 31 22 25 18 23 54
17 19
 [379]   6 76 30 15 51 33 28 35 17 16 36 22 26   1 29 17
24   2
 [397] 37 36 38 22 34 52 58 58 55 18   7 39 40 17 14 21
23 26
 [415] 14 19 22 20 76 34 20 38 16 28 24 18 23 35 25 50
23 16
 [433] 27   2 60 16 27 40 70 50 41 58 60 26 23 32 17 30
20 39
 [451] 18 32 23 25 74 14 40 30 22   8 30 46 36 85 65   5
46 45
 [469] 18 36 49 21 17 23 16 30 45 30 22 30 52 26 15 36
24 20
 [487] 28 18 29 25 18 32 37 16 18 20 20 30 18 26 30 18
23 12
 [505] 36 23 36 25 45 32 46 17 15 27 40 40 25 40 38 17
20 29
 [523] 21 21 27 27 39 22 26 30 25 59 15   4 31 18 85 34
20 31
 [541] 45 26 18 20 40 32 47 38 45 30 28 19 33 30 51 83
72 61
 [559] 48 39 62 82 43 50 50 28 18   4 35 77 21 24 39 41
21 21
 [577] 16 34 61 19 27 45 55 68 74 68 58 85 69 20 32 75
27 36
 [595] 35 20 12   8 40 27 16 40 19 40 21 19 54 11 16 38
25   7
 [613] 29 21 30 25 15 18 27 29 25 25 25 20   8 36 24 17
62 60
 [631] 19 25 17 64 41 46 51 46 37 53 45 50 60 72 75   3
73 33

 [649] 33 19 10 40 19 20 21 18 18 15 50 25 45 40 15 36
38 33
 [667] 19 23 16 18 17 36   6 19 21 26 26 27 35 40   3 54
67 22
 [685] 78 58 37 65 31 55 35 20 64 62 38 60 58 84 62 70
35 72
 [703]   4 40 57 78 68 50 40 18 18 43 33 41 22 29 32 40
40 40
 [721] 42 25 34 44 28 20 37 37 24 22 35 48   4   6 16 25
32 12
 [739] 20 24 34 24 25 27 29 34 36 45 53   8 26 22 41 42
35 40
 [757] 523 27 37 35 20 20 24 24 27 30 30 55 20 20 22 27
50 32
 [775] 23 30 40 35 37 52 27 35 22 50 19 43 22   2 32 18
18 18
 [793] 18 19 19 20 20 20 24 24 25 25 25 42 26 30 49 17
24 36
 [811] 40 36   9 27 17 22 23 16 25 30 19 20 55 34 19 21
7 27
 [829] 28 41 45 23 30 28 23 26 38 22 40 47 23 35 24 29
18 38
 [847] 35 22 60 23 30   2 27 34 10 31 17 59 50 63 12 48
56 25
 [865] 17 30 19 21 54 17   2 15 49 52 70 45 48 12 16 22
30 18
 [883] 33 22 27 52 36 27   5 32 32 32   5 77 50 25 25 19
38 31
 [901]   6 55   4 53 31 35 36 46 22 52 20 36 21 32 45 48
18 50
 [919] 25 26 27 35   4 30 33 17 35 42 48 27 46   9 27 40
24 19
 [937] 24 20 20 51 25 11 37   1 80 28 23 40 13 54 30 20
4 42
 [955] 20 23 29 34 49 21 27 17 34 22 50 22 11 23 55 17
25 51
 [973] 19 32 26 21 41 31   8 45 33 19 52 21 21 14 57 59
19 22
 [991] 16 32 29 26 57 49 39 10 28   9 22 18 27   4 38 15
84 75
[1009] 44 60 39 40 40 40 25 70 38 18 38 52 41 55 18 60
5 49
[1027] 18 41 18 31 31 14 34 42 21 56 28 35 58 25 23 22
20 28
[1045]   8 54 45 60 60 70 60 17 60 36 53 45 68 65 70 65
56 35
[1063] 65 42 82 55 38 32 18 43 75 40 63 37 78 41 70 62
70 76
[1081] 64 45 45 72 65 45 42 45 68 46 50 59 65 68 16 60
70 64

```
[1099]  39  80  85  71  75  78  42  55  67  72  56  61  58  48  22  16
31  18
[1117]  19  20  20  24  30  34  34  19  23  30  32  38  15  34  29  36
32  54
[1135]  60  62  40  60  63  82  79  80  82  57  18  85  73  45  50  68
53  76
[1153]  71  74  87  60  76  51  54  60  40  81  50  57  42  77  65  65
59  70
[1171]  64  59  59  69  65  76  68  60  37  45  56  42  29  70  35  46
70  71
[1189]  74  58   6  40  45  65  70  68  55  45  23  77  62  73  53  67
54  56
[1207]  33  58  62  74  47  30  67  29  65  51  30  64  66  66  76  74
22  88
[1225]  48  66  41  82  31  75  77  65  70  70  62  50  32  23  37  90
83  76
[1243]  69  39  63  59  87  62  35  34  86  54  82  76  67  39  56  65
83  52
[1261]  55  45  51  52  71  72  72  92  77  40  77  65  40  70  50  20
74  88
[1279]  65  72  86  65  74  55  89  70  75  55  18  65  20  70  47  40
97  60
[1297]  60  72  80  67  40  75  35  70  64  45  77  73  80  67  83  38
61  76
[1315]  43  78  85  51  45  70  59  83  55  62  85  70  61  73  47  40
73  55
[1333]  63  72  60  55  36  61  55  55  58  35  64  86  34  96  66  69
20  61
[1351]  62  50  51  50  34  38  47  86  60  66  85  50  59  72  49  58
61  17
[1369]  68  88  58  73  84  62  55  52  63  50  34  75  78  55  50  39
68  70
[1387]  32  84  39  73  75  51  58  72  60  70  57  53  39  76
```

Now, I removed the NA values. To show summary statistics of the age

```
summary(age)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   1.00   23.00   35.00   38.93   53.00  523.00
```

Looking at summary statistics of age minimum age of people in covidbl table is 1, first quartile is 23 similarly median age of group is 35 on the same way average age of people is 38.93 like wise third quartile age group is 53 and maximum age group of people in this table is 523.

[Q.N.7] Transform cleaned age variable into broad age groups i.e. <15, 15-59, 60+ years, define it as factor variable and get number and percentage of this variable and interpret it carefully

```
less_then_15 <- df_age$age < 15
below_15<- df_age[less_then_15,]$age
below_15<- factor(below_15)
table(below_15)
```

```
below_15
 1  2  3  4  5  6  7  8  9 10 11 12 13 14
 4  8  2 12  5  5  4  7  6  5  5  5  3  6
```

```
prop.table(table(below_15))*100
```

```
below_15
        1         2         3         4         5         6         7
8
 5.194805 10.389610  2.597403 15.584416  6.493506  6.493506  5.194805
9.090909
        9        10        11        12        13        14
 7.792208  6.493506  6.493506  6.493506  3.896104  7.792208
```

```
between_15_59 <- (df_age$age <= 59)
between_15_59<- df_age[between_15_59,]$age
between_15_59 <- between_15_59[between_15_59>=15]
between_15_59<- factor(between_15_59)
table(between_15_59)
```

```
between_15_59
15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37
38 39 40
11 19 27 50 35 47 24 45 34 24 47 26 36 32 21 43 16 36 14 31 41 34 17
24 16 48
41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59
18 14  6  3 34 11  8 14  9 28 13 14  9 10 27  8  7 16 11
```

```
prop.table(table(between_15_59))*100
```

```
between_15_59
       15        16        17        18        19        20        21
22
1.0396975 1.7958412 2.5519849 4.7258979 3.3081285 4.4423440 2.2684310
4.2533081
       23        24        25        26        27        28        29
30
3.2136106 2.2684310 4.4423440 2.4574669 3.4026465 3.0245747 1.9848771
4.0642722
       31        32        33        34        35        36        37
38
1.5122873 3.4026465 1.3232514 2.9300567 3.8752363 3.2136106 1.6068053
2.2684310
```

```
       39        40        41        42        43        44        45
46
1.5122873 4.5368620 1.7013233 1.3232514 0.5671078 0.2835539 3.2136106
1.0396975
       47        48        49        50        51        52        53
54
0.7561437 1.3232514 0.8506616 2.6465028 1.2287335 1.3232514 0.8506616
0.9451796
       55        56        57        58        59
2.5519849 0.7561437 0.6616257 1.5122873 1.0396975

greater_than_60 <- df_age$age >= 60
above_60 <- df_age[greater_than_60,]$age
above_60 <- factor(above_60)
print(above_60)

  [1] 72  60  65  65  65  81  62  60  65  80  60  60  65  65  63  60
61  61
 [19] 75  65  74  69  70  62  60  60  60  76  76  60  70  60  74  85
65  85
 [37] 83  72  61  62  82  77  61  68  74  68  85  69  75  62  60  64
60  72
 [55] 75  73  67  78  65  64  62  60  84  62  70  72  78  68  523 60
63  70
 [73] 77  80  84  75  60  70  60  60  60  70  60  60  68  65  70  65
65  82
 [91] 75  63  78  70  62  70  76  64  72  65  68  65  68  60  70  64
80  85
[109] 71  75  78  67  72  61  60  62  60  63  82  79  80  82  85  73
68  76
[127] 71  74  87  60  76  60  81  77  65  65  70  64  69  65  76  68
60  70
[145] 70  71  74  65  70  68  77  62  73  67  62  74  67  65  64  66
66  76
[163] 74  88  66  82  75  77  65  70  70  62  90  83  76  69  63  87
62  86
[181] 82  76  67  65  83  71  72  72  92  77  77  65  70  74  88  65
72  86
[199] 65  74  89  70  75  65  70  97  60  60  72  80  67  75  70  64
77  73
[217] 80  67  83  61  76  78  85  70  83  62  85  70  61  73  73  63
72  60
[235] 61  64  86  96  66  69  61  62  86  60  66  85  72  61  68  88
73  84
[253] 62  63  75  78  68  70  84  73  75  72  60  70  76
35 Levels: 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79
... 523
```

[Q.N.8]Find the number of days between recovered and reported dates (define it as: diff1 variable), clean it if required, and get the summary statistics of this variable and interpret it carefully.

Solution: In column reportedOn and reportedOn I see many values are missing so my first attempt was removing these values and then our columns are in string format another task to do is change them in date formate finally difference is calculate.

```
colnames(covidtbl)
```

```
 [1] "id"                "province"          "district"
 [4] "municipality"      "createdOn"         "modifiedOn"
 [7] "label"             "gender"            "age"
[10] "occupation"        "reportedOn"        "recoveredOn"
[13] "deathOn"           "currentState"      "isReinfected"
[16] "source"            "comment"           "type"
[19] "nationality"       "ward"              "relatedTo"
[22] "point.type"        "point.coordinates"
```

```
covidtbl_2 <-covidtbl[complete.cases(covidtbl$recoveredOn),]
date_of_recoveredOn <-as.Date(covidtbl_2$recoveredOn)
covidtbl_1 <-covidtbl[complete.cases(covidtbl$reportedOn),]
date_of_reportedOn<- as.Date(covidtbl_2$reportedOn)

diff1<- date_of_recoveredOn - date_of_reportedOn
#print(diff1)

fivenum(diff1)
```

```
Time differences in days
[1]   0   1  15  27 179
```

From the above fivenum summary average days different between recovered and reported dates is 15 and median days difference is 27 similarly maximum days difference is 179. Data are heighly skiewed.

[Q.N.9] Find the number of days between deaths and reported dates (define it as: diff2 variable), clean it if required, and get the summary statistics of this variable and interpret it carefully

Solution:

```
covidtbl_3 <-covidtbl[complete.cases(covidtbl$deathOn),]
death_date <- covidtbl_3$deathOn

death_date<- as.Date(death_date)
diff2 <- death_date - date_of_reportedOn
#list(diff2)
```

```
Warning message in unclass(time1) - unclass(time2):
“longer object length is not a multiple of shorter object length”

fivenum(diff2)

Time differences in days
[1] -135   -8   16   53  156
```

From above data we can see that average days difference beween deaths and reported dates is 16 and median days different beween corresponding variable is 53 maximum days difference is 156.

[Q.N.10] Which measures of central tendency and dispersion is most appropriate for the age, diff1 and diff2 variables? Why? Justify the use the appropriate measure of central tendency and dispersion by creating and interpreting the histogram and boxplot for the age, diff1 and diff2 variables and their careful interpretations

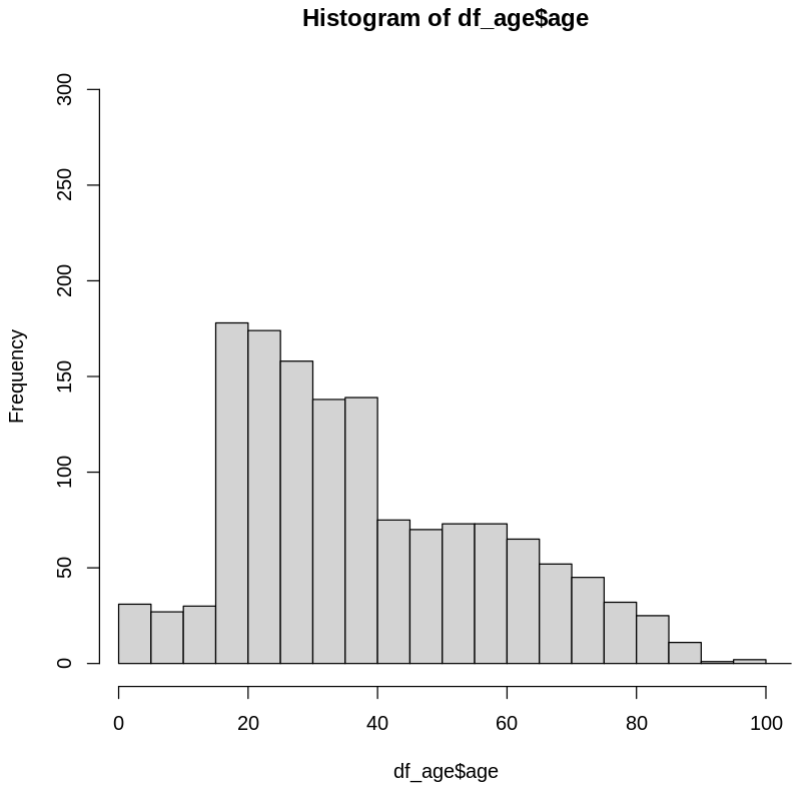
Solution: In case of age median is appropriate measure of central tendency. Because of following reasons.

- There are a few extreme scores in the distribution of the data. (NOTE: R
- single outlier can have a great effect on the mean.
- There are some missing or undetermined values in your data. c.
- There is an open ended distribution
  Corresponding measure of dispersion for age is IQR.

In case of diff1 median is approprate measure of central tendency. Because while watching data there are negative value, extermely differnt values. And corresponding measure of disperson is IQR.

Similarly, for diff2 data are skewed more so appropriate measure of central tendency is median and corresponding measure of dispersion is IRQ.

```
hist(df_age$age, xlim=c(1,100), ylim=c(0,300), breaks = 100)
```

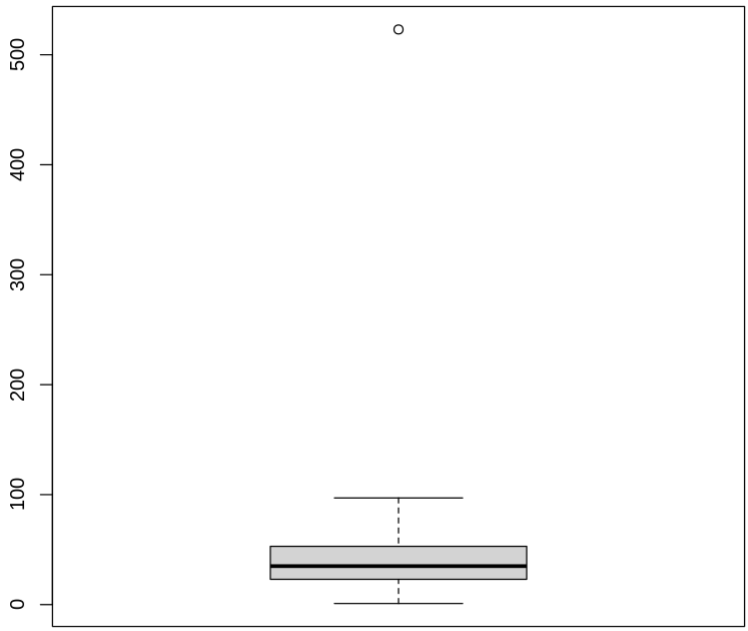


From histogram above we can notice that age groups with heigest frequency belong to the interval 20 - 40.

```
summary(age)
```

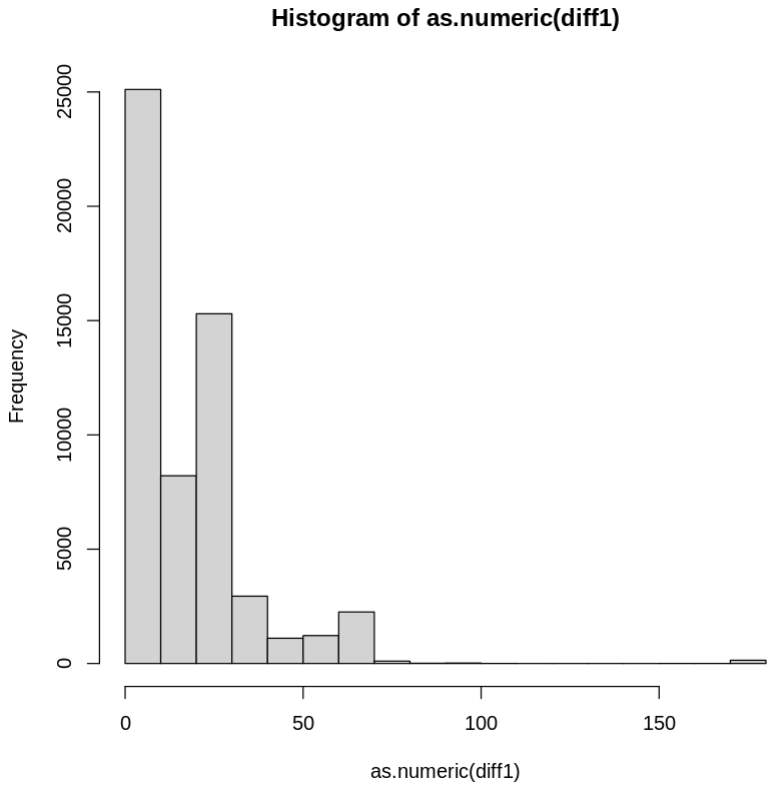
```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
   1.00   23.00   35.00   38.93   53.00  523.00 
```

```
boxplot(age)
```



From boxt plot we can see minimum value of age are 1 average value is appromately 39. We can see that there is presence of outliers.

```
hist(as.numeric(diff1))
```




```
colnames(covidtbl)
```

```
 [1] "id"                "province"          "district"
 [4] "municipality"      "createdOn"         "modifiedOn"
 [7] "label"             "gender"            "age"
[10] "occupation"        "reportedOn"        "recoveredOn"
[13] "deathOn"           "currentState"      "isReinfected"
[16] "source"            "comment"           "type"
[19] "nationality"       "ward"              "relatedTo"
[22] "point.type"        "point.coordinates"
```

[Q.N.11]Show the number and percentage of the "current state" variable and interpret it carefully

```
current_state <- covidtbl$currentState
num_current_sate<-table(current_state)

prop.table(num_current_sate)*100
```

```
current_state
   active     death recovered
26.846319  0.639963 72.513718
```

Current state variable have three types of values active, death, recovered and about 72% people are recovered similarly approximately 27% people have covid posite and as

compared to recoved ratio only 0.63% were deaths. From the data above maximum people can recovered from covid.

[Q.N.12]Show the number and percentage of the "isReinfected" variable, what percentage of cases were re-infected in Nepal at the given time period in the database? Was it realistic?

```
isreinfected <- covidtbl$isReinfected
num_isreinfected <- table(isreinfected)
prop.table(num_isreinfected)*100
```

```
isreinfected
       FALSE         TRUE
99.996144801  0.003855199
```

About 0.0038% people were reinfected from covid In Nepal. That ratio is too small few of the recovered people might be reinfected.

[Q.N.13] Show the number and percentage of "type" variable and discuss the result carefully with special reference to the missing values

```
type <- covidtbl$type
type_count <- table(type)
prop.table(type_count)*100
```

```
type
          imported local_transmission
          57.89474           42.10526
```

[Q.N.14]Show the number and percentage of "nationality" variable and discuss the result carefully with special reference to the missing values

```
nationality <- covidtbl$nationality
covidtbl_4<- covidtbl[complete.cases(covidtbl$nationality),]
currted_nationality <- covidtbl_4$nationality
nationality_count <- table(currted_nationality)
print(nationality_count)
prop.table(nationality_count)*100
```

```
currted_nationality
 2  3  4
42 14  1

currted_nationality
        2         3         4
73.684211 24.561404  1.754386
```

Nationality columns is also suffered from missing values due to the missing values our analysis might not be accurate.

[Q.N.15] Show cross-tabulation of province (row variable) and current status (column variable) with row percentage and interpret the result carefully (table and prop.table can be used!)

```
cross_tabul<- table(covidtbl$province, covidtbl$currentState)
cross_tabul
```

```
  active death recovered
1    643    56      6276
2   1825   133     13061
3  13086   187     16462
4   1108    28      3293
5   2496    76      8563
6    593     5      2950
7   1140    13      5823
```

```
prop.table(cross_tabul)*100
```

```
         active        death    recovered
1  0.826297596  0.071963710  8.065075755
2  2.345245897  0.170913811 16.784250228
3 16.816376884  0.240307388 21.154760528
4  1.423853400  0.035981855  4.231723145
5  3.207525348  0.097665035 11.004022257
6  0.762044283  0.006425331  3.790945423
7  1.464975519  0.016705861  7.482940746
```

Maximum corona active were in provience 3, minimum corona active were in provience 6 similarly maximum people were died from corona in provience 3 also maximum people recovered from coron in same provience.

[Q.N.16] Show the cross-tabulation of sex (row variable) and current status (column variable) with row percentage and interpret the result carefully

```
gender<- covidtbl$gender
gender <- str_to_title(gender)
cross_tabul<- table(gender, covidtbl$currentState)
cross_tabul
```

```
gender   active death recovered
  Female   6960   149     14294
  Male    13931   348     42076
```

```
prop.table(cross_tabul)*100
```

```
gender       active      death  recovered
  Female  8.9508475  0.1916202 18.3826745
  Male   17.9158415  0.4475424 54.1114741
```

Male were affected from corona more than female similarly death, recovered ratio of male is maximum than female.

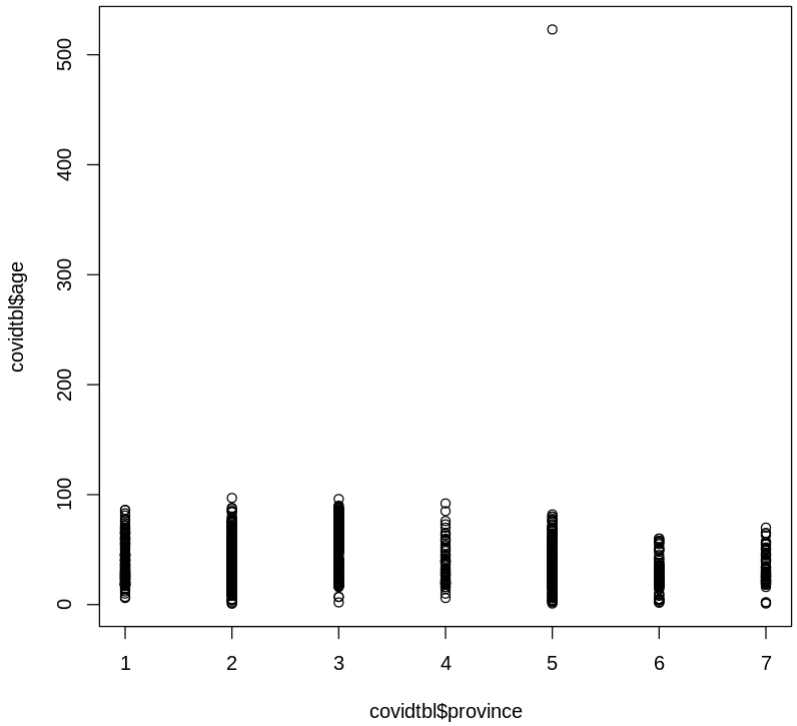
[Q.N.17]Show the cross-tabulation of broad age groups (row variable) and current status (column variable) with row percentage and interpret the result carefully

```
combined <- unlist(list(below_15,above_60, between_15_59))
borad_age <- as.numeric(combined)

cross_tabul<- table(borad_age, df_age$currentState)
cross_tabul
```

Above data show that death rate of broad age above 60 is maximum compared to the others. Recovered rate of broad age group between 15 to 59 is maximum compared others.

[Q.N.18]Show the scatterplot of province (x-axis) and cleaned age (y-axis) and get the appropriate correlation coefficient for this bi-variate data and interpret the result carefully

```
plot(covidtbl$province,covidtbl$age)
```



Scatter plot does not show any specific pattern, it is not linear so in above case spearman rank correlation is appropriate.

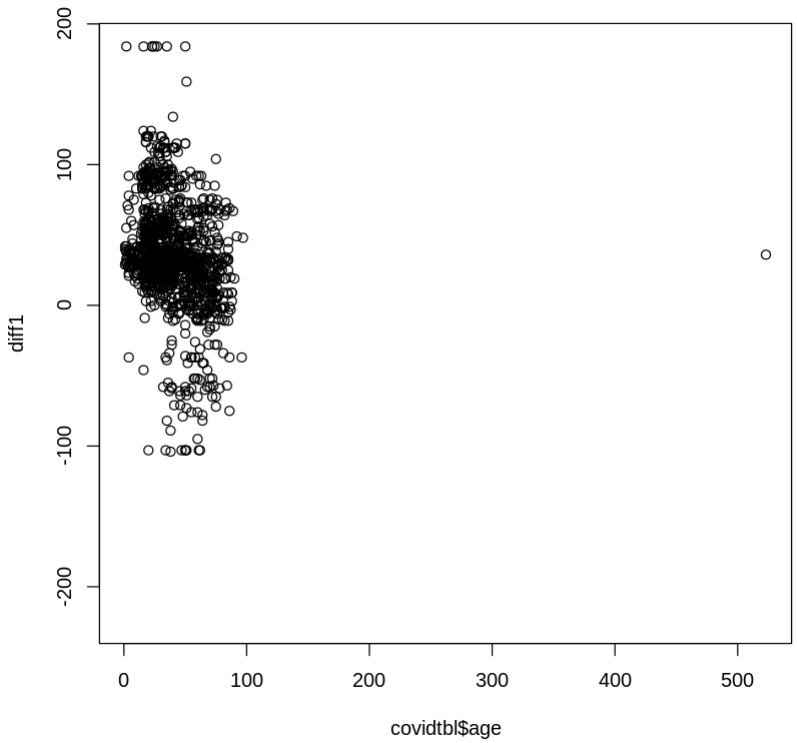
```
cor.test(covidtbl$province,covidtbl$age, method = "spearman")
```

```
Warning message in cor.test.default(covidtbl$province, covidtbl$age,
method = "spearman"):
“Cannot compute exact p-value with ties”


     Spearman's rank correlation rho

data:  covidtbl$province and covidtbl$age
S = 509628932, p-value = 1.796e-05
alternative hypothesis: true rho is not equal to 0
sample estimates:
       rho
-0.1143495
```

There is low degree of negative correlation

[Q.N.19]Show the scatterplot of age (x-axis) and diff1 (y-axis) and get the appropriate correlation coefficient for this bi-variate data and interpret the result carefully

```
plot(covidtbl$age,diff1)
```



Above scatterplot do not show any specific pattern so spearman rank correlation coefficent is appropriate.

```
cor.test(covidtbl$age,as.numeric(diff1), method = "spearman")
```

```
Warning message in cor.test.default(covidtbl$age, as.numeric(diff1),
method = "spearman"):
“Cannot compute exact p-value with ties”
```

```
     Spearman's rank correlation rho

data:  covidtbl$age and as.numeric(diff1)
S = 592527576, p-value < 2.2e-16
alternative hypothesis: true rho is not equal to 0
sample estimates:
       rho
-0.2956149
```

Between age and difference there is low degree of negative correlation.

[Q.N.20]Summarize your learning using SWOT analysis to complete this project in a 2x2 table; be honest and specific so that I can learn from you too

Strength	Weakness
Our data is already tidy, no duplicate presence. One can do easily an analysis between any columns.	Many values are missing in data, also spelling mistach in some column.

Oppertinuty	Threats
Have looked at broad age groups we can say which age group affected much from covid. We can find out affect of covid on gender basis, provience basis.	Due to presence of many missing values our EDA is not accurate.