Data Visualization Assigment 1

Amruth Karun M V

13-Mar-2022

Contents

Introduction 1
Dataset 1
Clean data 2
Plots 3
1. Bar Chart of Total Disease Cases 3
2. Density Plot for Respiratory Infection Cases 4
3. Mirrored Histogram for Japanese Encephalitis 5
4. Pie Chart for Total Cases 6
5. Scatter plot of Malaria Cases vs Deaths 7
6. Box plot of Deaths in each State/UT 8
7. Violin plot of Deaths due to Diseases 9
8. Lollipop Chart of Missing Values in the Deaths Data 10
9. Stacked Barchart of Deaths in Kerala and Uttar Pradesh 12
10. Circular Bar Plot for Mean Death Cases 12
11. Heatmap fo Diseases in Different States 14
12. Parallel Coordinate Plot for the Dataset 16
Conclusion 17

Introduction

The objective of this assignment is to visualize the given dataset using different charts, describe the relevance of these visualizations and what insights we get by using the particular graph.

Dataset

The data refers to State-wise (2010 and 2011) number of cases and deaths due to specified diseases (Acute Diarrhoeal Diseases, Malaria, Acute Respiratory Infection, Japanese Encephalitis, Viral Hepatitis). The dataset consist of the year, total cases and the total deaths in a state due to each of these disease. It contains 38 rows and 12 columns which includes null and missing values.

```
data <- read.csv(file = 'Number_of_Cases_And_Deaths_Due_To_Diseases.csv')
print("Dataset Dimension:")
```

```
## [1] "Dataset Dimension:"
```

```
dim(data)
```

```
## [1] 38 12
```

Clean data

In this step, we remove extra items in column headers and add total cases and deaths for each state and UTs for the year. The data contains only the details for the year 2011, so it can be removed and null values are kept because there are only 38 rows for visualization.

```
# Remove extra items in column names
names(data) <- sub("...Cases", ".Cases", names(data))
names(data) <- sub("...Deaths", ".Deaths", names(data))
print("Columns:")
```

```
## [1] "Columns:"
```

```
names(data)
```

```
##  [1] "Year"                                  "State.UTs"
##  [3] "Acute.Diarrhoeal.Diseases.Cases"       "Acute.Diarrhoeal.Diseases.Deaths"
##  [5] "Malaria.Cases"                         "Malaria.Deaths"
##  [7] "Acute.Respiaratory.Infection.Cases"    "Acute.Respiaratory.Infection.Deaths"
##  [9] "Japanese.Encephalitis.Cases"           "Japanese.Encephalitis.Deaths"
## [11] "Viral.Hepatitis.Cases"                 "Viral.Hepatitis.Deaths"
```

```
# Remove null values and convert factors to numbers
data[3:12] =  apply(data[3:12], 2, function(x) as.numeric(as.character(x)))

# Calculate total cases and deaths for each state
data <- data %>% mutate(total_cases = rowSums(select(data, ends_with('Cases'))))
data <- data %>% mutate(total_deaths = rowSums(select(data, ends_with('Deaths'))))
data <- data[,-1] # remove first column
head(data)
```

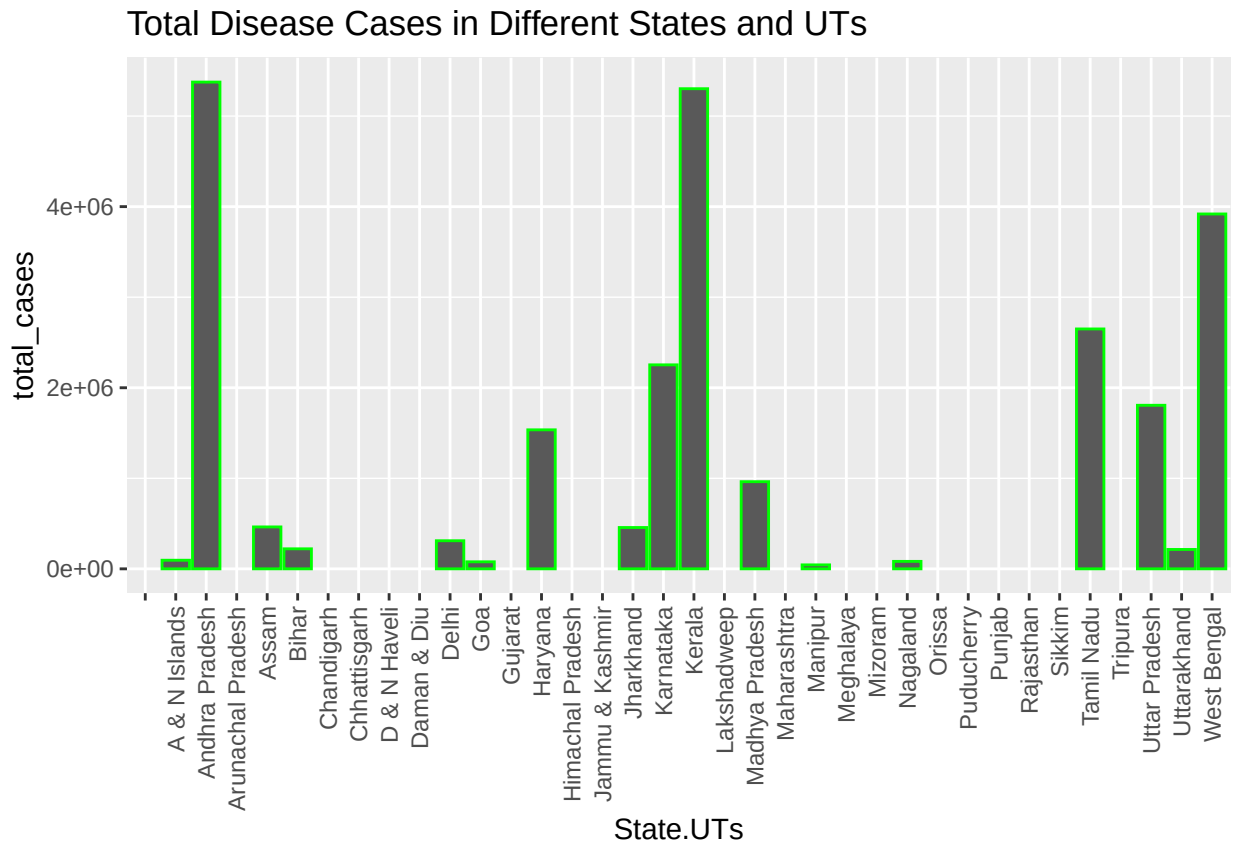
```
##           State.UTs Acute.Diarrhoeal.Diseases.Cases
## 1       GRAND TOTAL                         10231049
## 2    Andhra Pradesh                          2235614
## 3 Arunachal Pradesh                            32228
## 4             Assam                            96816
## 5             Bihar                           130276
## 6      Chhattisgarh                            64575
##   Acute.Diarrhoeal.Diseases.Deaths Malaria.Cases Malaria.Deaths
## 1                             1269       1278760            463
## 2                              107         39559              5
```

```
## 3                              11         10961             NA
## 4                              16         47397             42
## 5                              NA          2390              0
## 6                               5        131179             18
##   Acute.Respiaratory.Infection.Cases Acute.Respiaratory.Infection.Deaths
## 1                           26300208                                2492
## 2                            3089290                                 236
## 3                              48602                                   9
## 4                             314824                                  NA
## 5                              87486                                  NA
## 6                             155743                                  18
##   Japanese.Encephalitis.Cases Japanese.Encephalitis.Deaths
## 1                        8249                         1169
## 2                          73                            1
## 3                          NA                           NA
## 4                        1319                          250
## 5                         821                          197
## 6                          NA                           NA
##   Viral.Hepatitis.Cases Viral.Hepatitis.Deaths total_cases total_deaths
## 1                 94402                   520    37912668         5913
## 2                 11050                    61     5375586          410
## 3                   636                     4          NA           NA
## 4                  2557                    25      462913           NA
## 5                   202                    NA      221175           NA
## 6                   139                     1          NA           NA
```

Plots

1. Bar Chart of Total Disease Cases

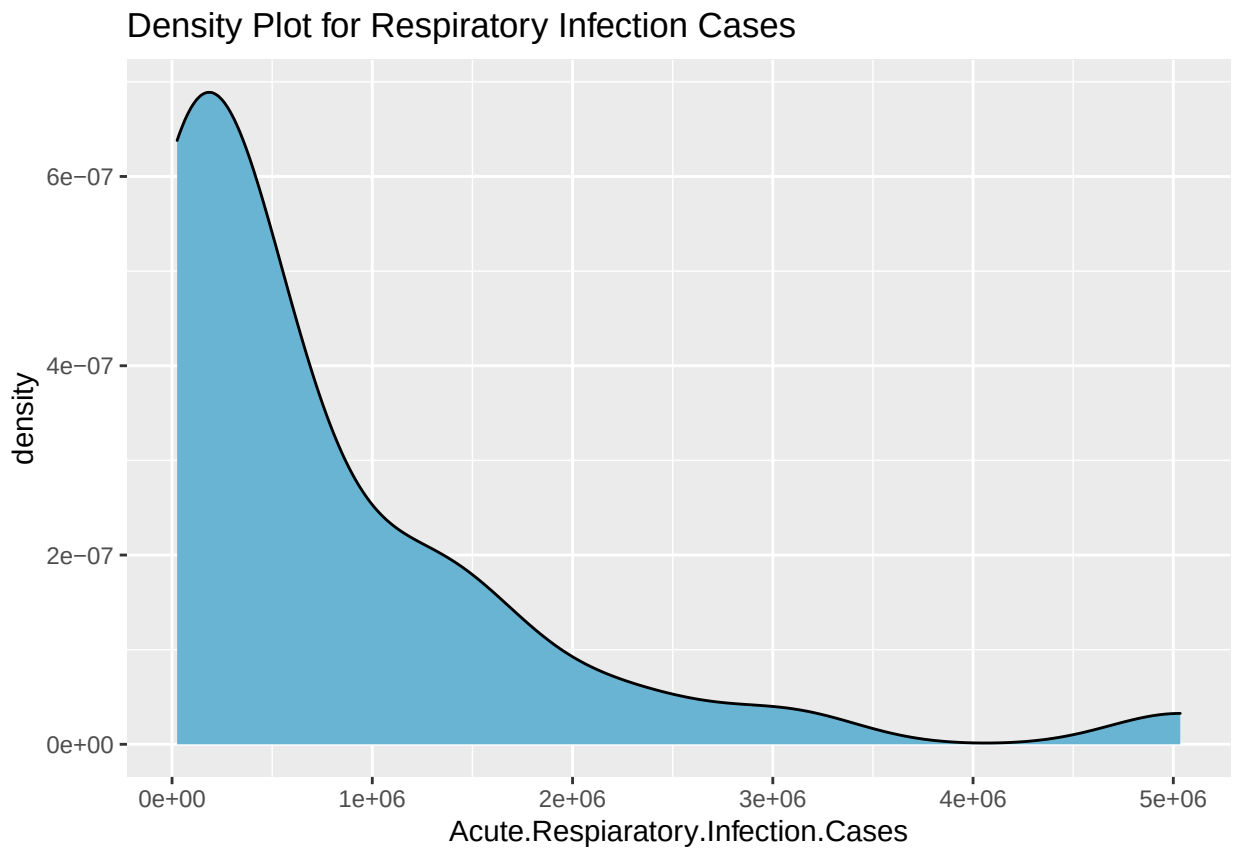
```
# data[-1,] --> Remove the GRAND TOTAL row
state_ut_data <- data[-1,]
plot1 <- ggplot(state_ut_data, aes(x=State.UTs, y=total_cases)) +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1)) +
  labs(title="Total Disease Cases in Different States and UTs") +
  geom_col(color="green")

plot1
```



Bar chart represents the categorical data using rectangular bars with different heights. Here for some data, the value is missing, so the height is zero and it shows the empty space in between bars and the height of the bar represents the total combined cases of diseases for the particular state of UT given in the x-axis. This gives an idea about how the cases are in different places and helps us make inferences like "the number of cases in Manipur is very low and for Kerala it is very high".

2. Density Plot for Respiratory Infection Cases

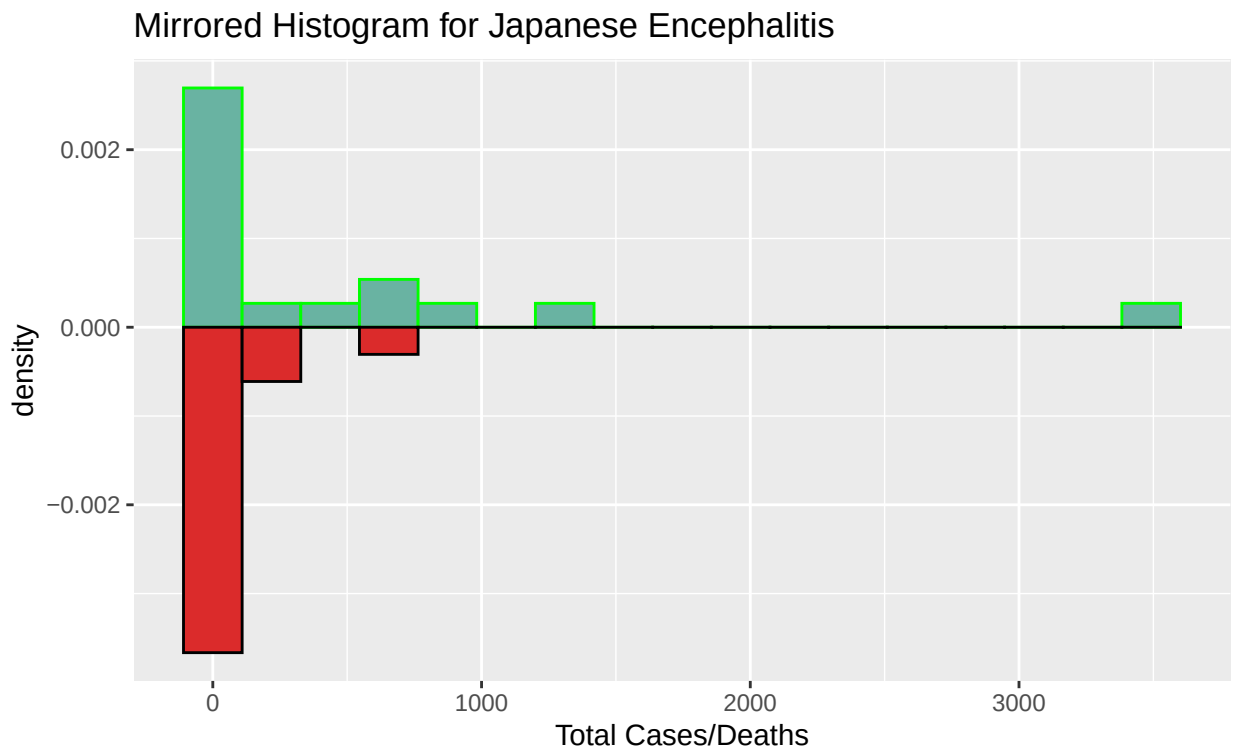
```
# data[-1,] --> Remove the GRAND TOTAL row
state_ut_data <- data[-1,]
plot2 <- ggplot(state_ut_data) +
  geom_density(aes(x=Acute.Respiaratory.Infection.Cases), fill="#69b4d2") +
  labs(title="Density Plot for Respiratory Infection Cases")

plot2
```



The above density plot shows the distribution of acute respiratory infection cases across different states and UTs. This is a smoothed version of the histogram using kernel density estimation and gives the general distribution of disease and makes it easier to compare regions with high and low density.

3. Mirrored Histogram for Japanese Encephalitis

```
plot3 <- ggplot(state_ut_data) +
  geom_histogram(bins=17, aes(x=Japanese.Encephalitis.Cases, y=..density..),
                 fill="#69b3a2", col="green" ) +
  geom_histogram(bins=17, aes(x=Japanese.Encephalitis.Deaths, y=-..density..),
                 fill= "#db2b2b", col="black") +
  xlab("Total Cases/Deaths") +
  labs(title="Mirrored Histogram for Japanese Encephalitis")

plot3
```



The mirrored histogram helps to compare the distribution of two different variables. Here the variables are Japanese Encephalitis cases and deaths. The graph is split into two, positive and negative. The negative part of the graph gives the distribution of deaths and positive part gives the distribution of cases. Also, histogram representation does not smooth out the data, so we can clearly see the gaps in the visualization and we are able to see the actual distribution of data than the smoothed version.

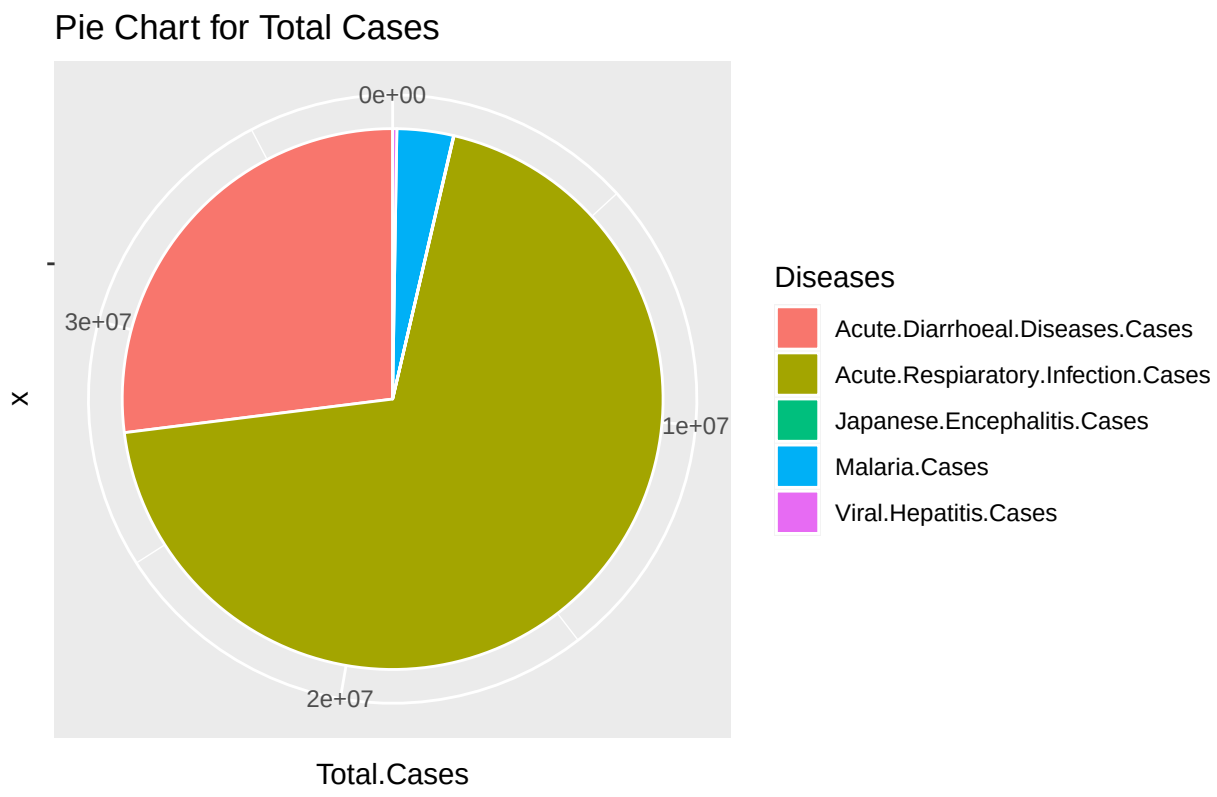
4. Pie Chart for Total Cases

```
# Create total case data for each disease with grand total value
cases <- c("Acute.Diarrhoeal.Diseases.Cases", "Malaria.Cases",
           "Acute.Respiaratory.Infection.Cases", "Japanese.Encephalitis.Cases",
           "Viral.Hepatitis.Cases")
case_data <- data.frame(t(data[cases][1,]))
colnames(case_data)[1] <- "Total.Cases"
head(case_data)
```

```
##                                 Total.Cases
## Acute.Diarrhoeal.Diseases.Cases    10231049
## Malaria.Cases                       1278760
## Acute.Respiaratory.Infection.Cases 26300208
## Japanese.Encephalitis.Cases            8249
## Viral.Hepatitis.Cases                 94402
```

```
Diseases <- row.names(case_data)

# Plot the pie chart of cases
plot4 <- ggplot(case_data, aes(x="", y=Total.Cases, fill=Diseases)) +
```

```
  geom_bar(stat="identity", width=1, color="white") +
  coord_polar("y", start=0) +
  labs(title="Pie Chart for Total Cases")


plot4
```

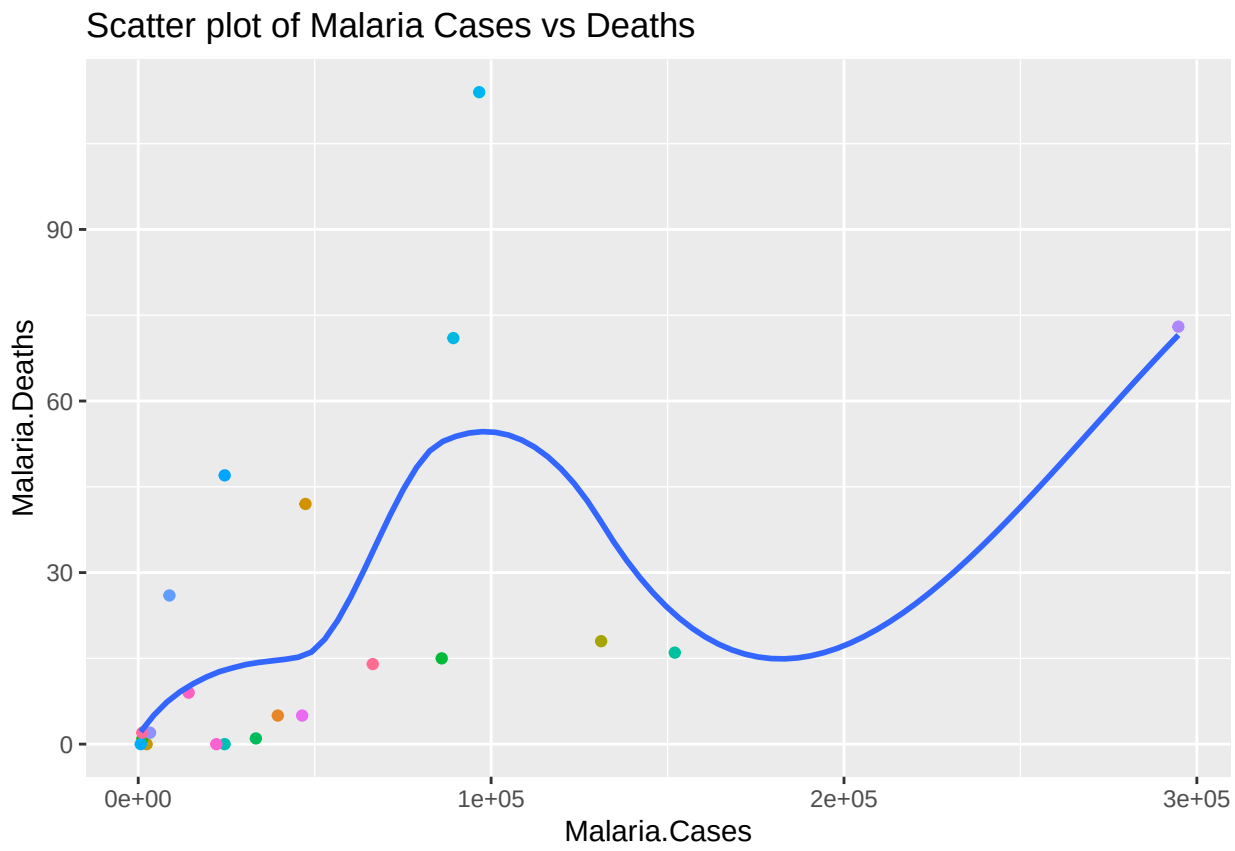


Pie chart is used for showing composition in the data. There is no direct way to implement pie chart in ggplot2, we can create bar chart using *geom_bar()* and then set the coordinates to polar. Here, I created a subset of the original dataset showing the total cases of each disease and the grand total. This data is visualized in the form of a pie chart. From this we can see that Respiratory Infection cases are around 75% followed by Diarrhoeal cases and the total cases of other diseases are less in comparison. Using this visualization we can reach general conclusion by saying that most of the population in India is infected by Acute Respiratory Diseases and require proper treatment and need to take other precautionary measures.

5. Scatter plot of Malaria Cases vs Deaths

```
# Create the scatter plot
plot5 <-  ggplot(state_ut_data, aes(x=Malaria.Cases, y=Malaria.Deaths)) +
  geom_point(aes(col=State.UTs)) + theme(legend.position = "none") +
  geom_smooth(method="loess", se=F) +
  labs(title="Scatter plot of Malaria Cases vs Deaths")

plot5
```

```
## `geom_smooth()` using formula 'y ~ x'
```



Scatterplot is mainly used for understanding the nature of relationship between two variables. Here, the two variables are the cases of Malaria and the number of deaths due to Malaria in different states/UTs. We can observe a linear correlation between total cases in general, but in some states the death rates are high and in some the rates are low. High death rates may be due to the lack of attention and care the patients get and lower rates indicate good medical practices.

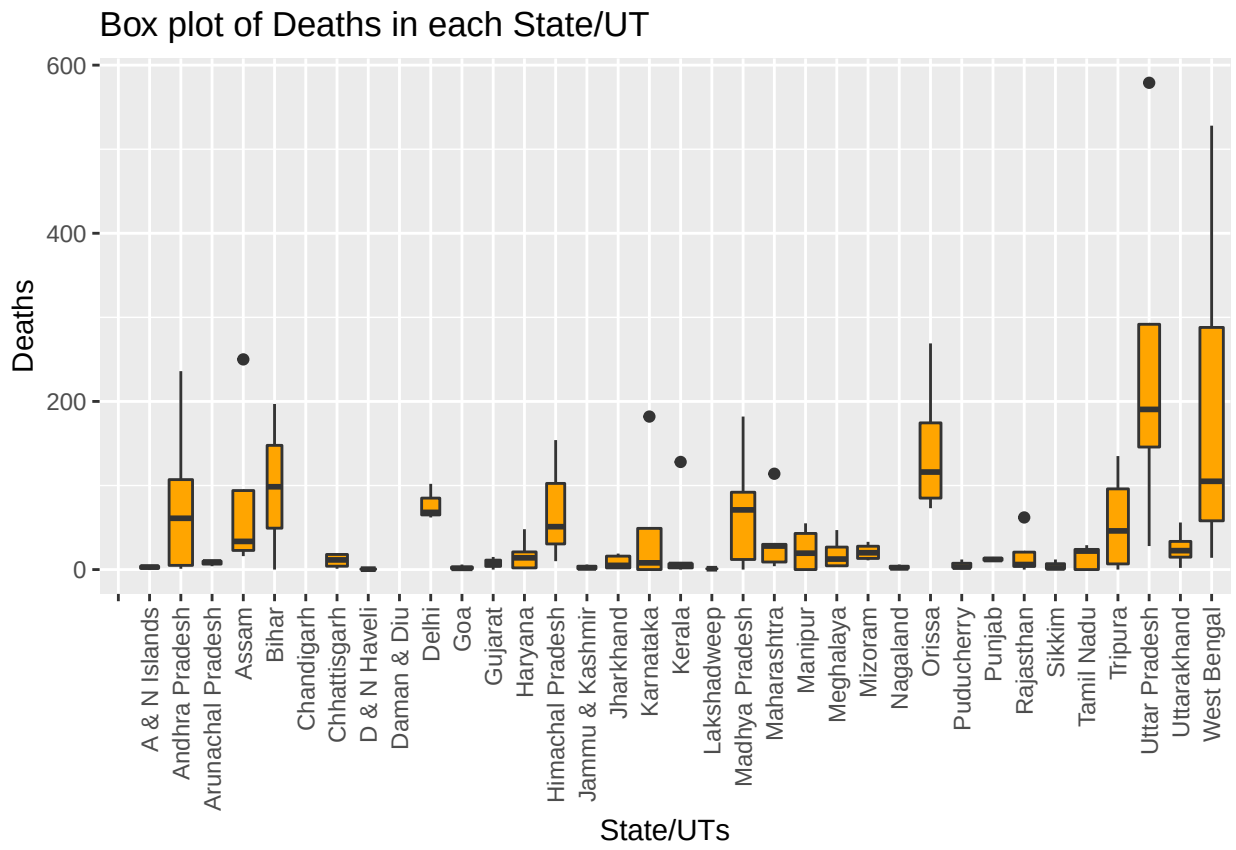
6. Box plot of Deaths in each State/UT

```
# Create total deaths data for each disease
columns <- c("State.UTs", "Acute.Diarrhoeal.Diseases.Deaths", "Malaria.Deaths",
             "Acute.Respiaratory.Infection.Deaths", "Japanese.Encephalitis.Deaths",
             "Viral.Hepatitis.Deaths")

# Gather the data
deaths_data <- data.frame(state_ut_data[columns]) %>%
  gather(key="Diseases", value="Deaths", -State.UTs)

head(deaths_data)
```

```
##           State.UTs                          Diseases Deaths
## 1    Andhra Pradesh Acute.Diarrhoeal.Diseases.Deaths    107
## 2 Arunachal Pradesh Acute.Diarrhoeal.Diseases.Deaths     11
```

```
## 3             Assam Acute.Diarrhoeal.Diseases.Deaths     16
## 4             Bihar Acute.Diarrhoeal.Diseases.Deaths     NA
## 5      Chhattisgarh Acute.Diarrhoeal.Diseases.Deaths      5
## 6             Delhi Acute.Diarrhoeal.Diseases.Deaths     62
```

```
plot6 <- ggplot(deaths_data, aes(State.UTs, Deaths)) +
  geom_boxplot(varwidth=T, fill="orange") +
  labs(title="Box plot of Deaths in each State/UT", x="State/UTs", y="Deaths") +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))

plot6
```

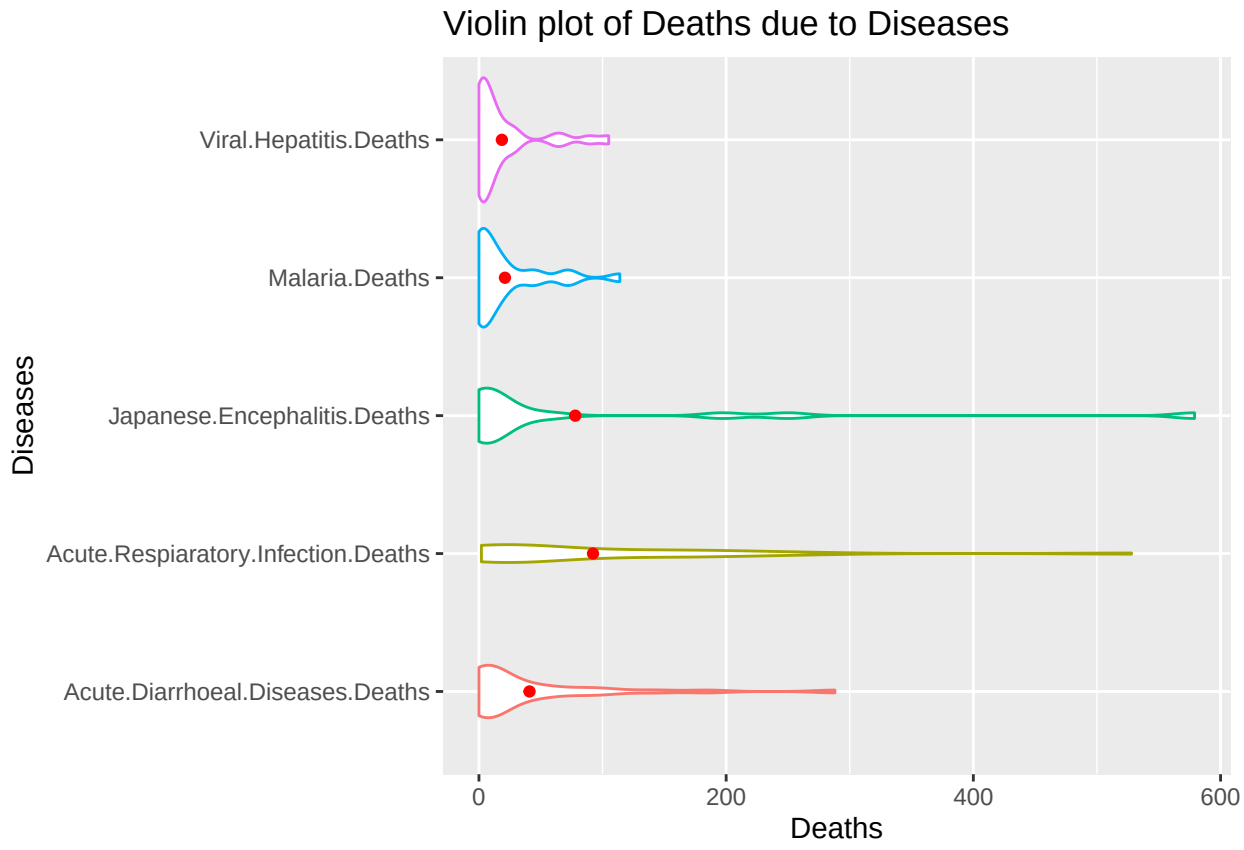


Box plot is an ideal way to represent the distribution of data across multiple groups. It also gives us additional information about the median, range and outliers in the data. Setting *varwidth=T* adjusts the width of the boxes to be proportional to the number of observation it contains. Here, we can see the box plot for each State/UT. For the state of West Bengal we can see the huge box which shows the high variation of deaths in the state and for states like Uttar Pradesh we can observe the large outliers in black which is far from the actual distribution of data. This shows the deaths due to one disease is much higher than the deaths due to other diseases.

7. Violin plot of Deaths due to Diseases

```
# Using the previously created deaths_data for creating the violin plot
plot7 <- ggplot(deaths_data, aes(Deaths, Diseases, color=Diseases)) +
```

```
  geom_violin() + theme(legend.position = "none") +
  labs(title="Violin plot of Deaths due to Diseases", x="Deaths", y="Diseases") +
  stat_summary(fun.y=mean, geom="point", color="red")

plot7
```



A violin plot of the deaths due to each disease used to visualize the density of death due to each disease using density curve. This is similar to the previous box plot, but used to represent the diseases rather than the state. The width of the curve corresponds to the approximate frequency of data points in each region. From this plot we can see that the density is high the region between 0 and 100. The violin plots gives a good comparison of deaths due to each disease and other plots like box plots can be overlayed to provide additional information.

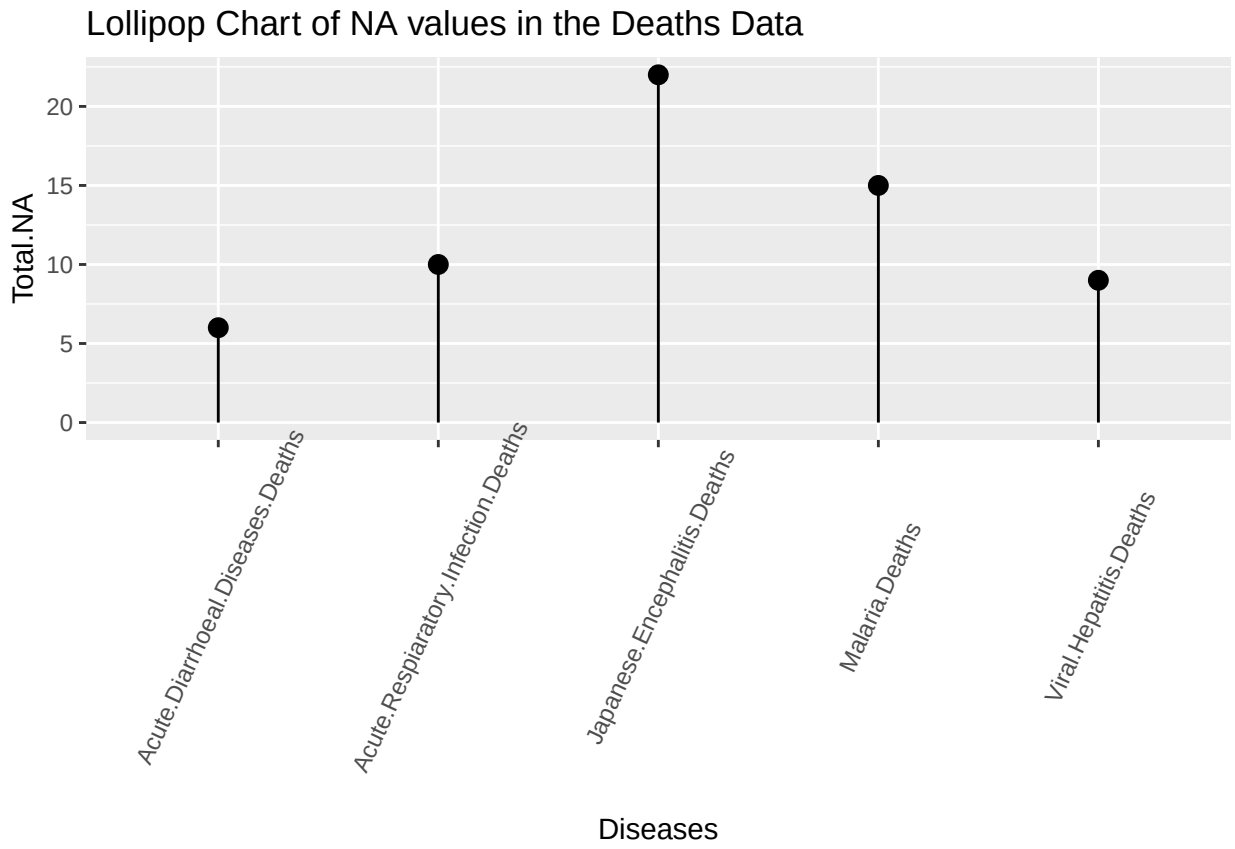
8. Lollipop Chart of Missing Values in the Deaths Data

```
# Count the number of na/missing values in the data
deaths_data$na_count <- apply(deaths_data, 1, function(x) sum(is.na(x)))
deaths_data_na <- deaths_data %>%
  group_by(Diseases) %>%
  summarise(Total.NA = sum(na_count), Deaths=sum(Deaths, na.rm=TRUE))

deaths_data_na
```

```
## # A tibble: 5 x 3
```

```
##   Diseases                              Total.NA Deaths
##   <chr>                                    <int>  <dbl>
## 1 Acute.Diarrhoeal.Diseases.Deaths             6   1269
## 2 Acute.Respiaratory.Infection.Deaths         10   2492
## 3 Japanese.Encephalitis.Deaths                22   1169
## 4 Malaria.Deaths                              15    463
## 5 Viral.Hepatitis.Deaths                       9    520
```

```
plot8 <- ggplot(deaths_data_na, aes(x=Diseases, y=Total.NA)) +
  geom_point(size=3) +
  geom_segment(aes(x=Diseases, xend=Diseases,
                   y=0, yend=Total.NA)) +
  labs(title="Lollipop Chart of NA values in the Deaths Data") +
  theme(axis.text.x = element_text(angle=65, vjust=0.6))

plot8
```

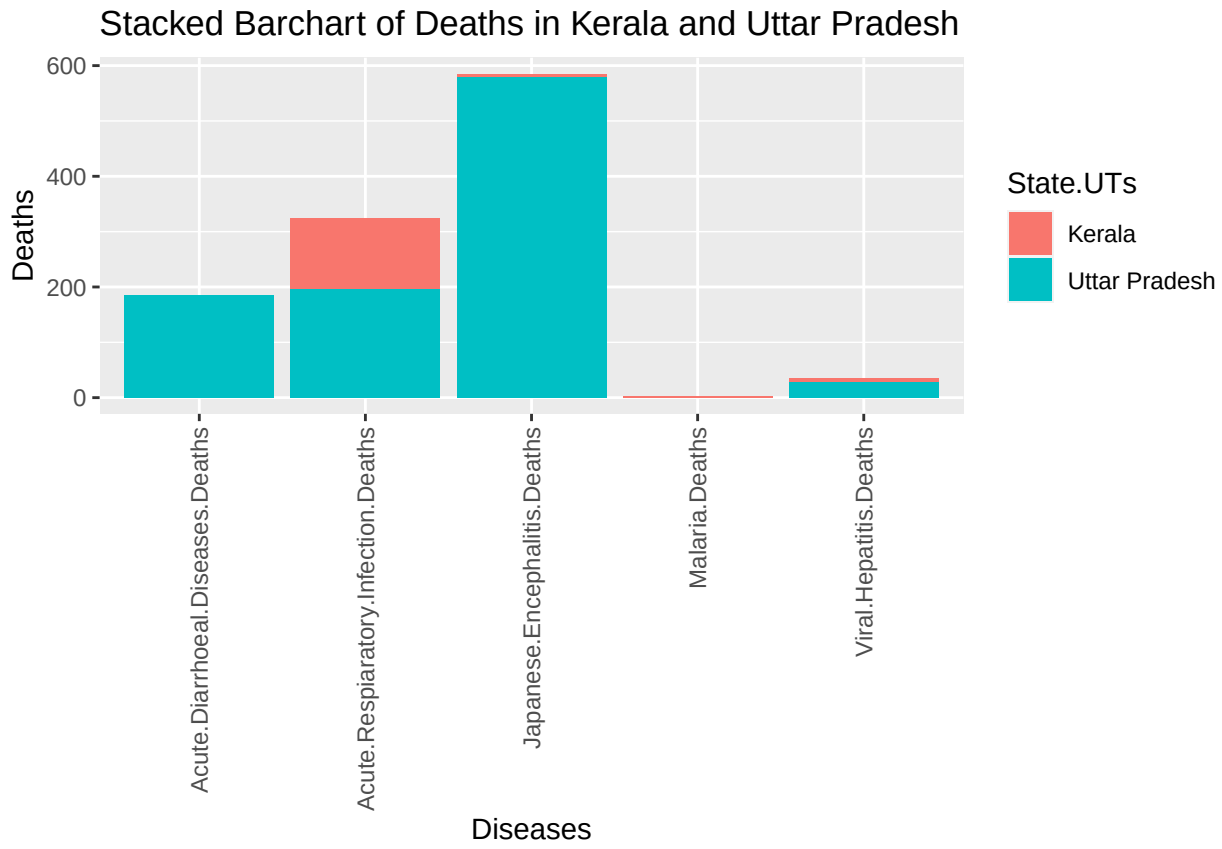


Lollipop chart is similar and amodern form of bar chart and conveys the same information. It reduces the thick bars into thin lines and reduces clutter in the visualization. Here, in this data, we have more emphasis on the the actual count of missing values in the data and now we are able to understand that the variations we observed while plotting the distributions are because of these missing values.

9. Stacked Barchart of Deaths in Kerala and Uttar Pradesh

```
# Create a stacked barchart of deaths in Kerala and UP
kerala_up_data <- deaths_data %>%
  filter(State.UTs =="Kerala" | State.UTs == "Uttar Pradesh")

plot9 <- ggplot(kerala_up_data, aes(fill=State.UTs, y=Deaths, x=Diseases)) +
  geom_bar(position="stack", stat="identity") +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1)) +
  labs(title="Stacked Barchart of Deaths in Kerala and Uttar Pradesh")

plot9
```

```
## Warning: Removed 1 rows containing missing values (position_stack).
```



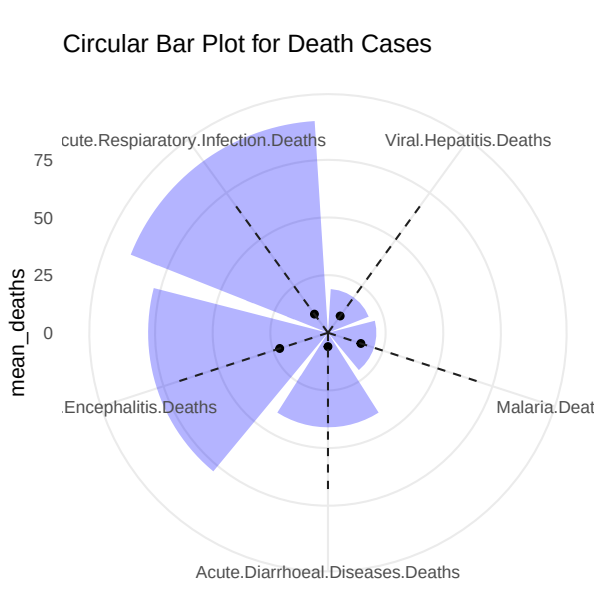
The above stacked barchart gives the comparison of deaths due to disease in Kerala and Uttar Pradesh. This is done by changing the position attribute to "stack" when using *geom_bar()*. The subgroups are stacked on top of the other and the width of the bar gives a good visual comparision and we can clearly see that the death rates in Kerala is much less than that of UP.

10. Circular Bar Plot for Mean Death Cases

```r
# Create data with mean cases and total missing values
deaths_data_mean <- deaths_data %>% group_by(Diseases) %>%
  summarise(
    total_deaths = sum(Deaths, na.rm=TRUE),
    mean_deaths = mean(as.numeric(Deaths), na.rm=TRUE),
    total_na = sum(na_count)) %>%
  mutate(mean_deaths = round(mean_deaths, digits = 0))

head(deaths_data_mean)
```

```
## # A tibble: 5 x 4
##   Diseases                              total_deaths mean_deaths total_na
##   <chr>                                        <dbl>       <dbl>    <int>
## 1 Acute.Diarrhoeal.Diseases.Deaths              1269          41        6
## 2 Acute.Respiaratory.Infection.Deaths           2492          92       10
## 3 Japanese.Encephalitis.Deaths                  1169          78       22
## 4 Malaria.Deaths                                 463          21       15
## 5 Viral.Hepatitis.Deaths                         520          19        9
```

```r
plot10 <- ggplot(deaths_data_mean,
                 aes(x=reorder(str_wrap(Diseases, 5), mean_deaths),
                     y=mean_deaths)) +
  geom_bar(stat="identity", fill=alpha("blue", 0.3)) +
  geom_point(aes(x=reorder(str_wrap(Diseases, 5), mean_deaths), y=total_na)) +
  geom_segment(aes(x = reorder(str_wrap(Diseases, 5), mean_deaths),
      y = 0, xend = reorder(str_wrap(Diseases, 5), mean_deaths),
      yend = 70), linetype = "dashed", color = "gray12") +
  theme_minimal() + coord_polar(start = 0) +
  labs(title="Circular Bar Plot for Death Cases") + xlab("")


plot10
```



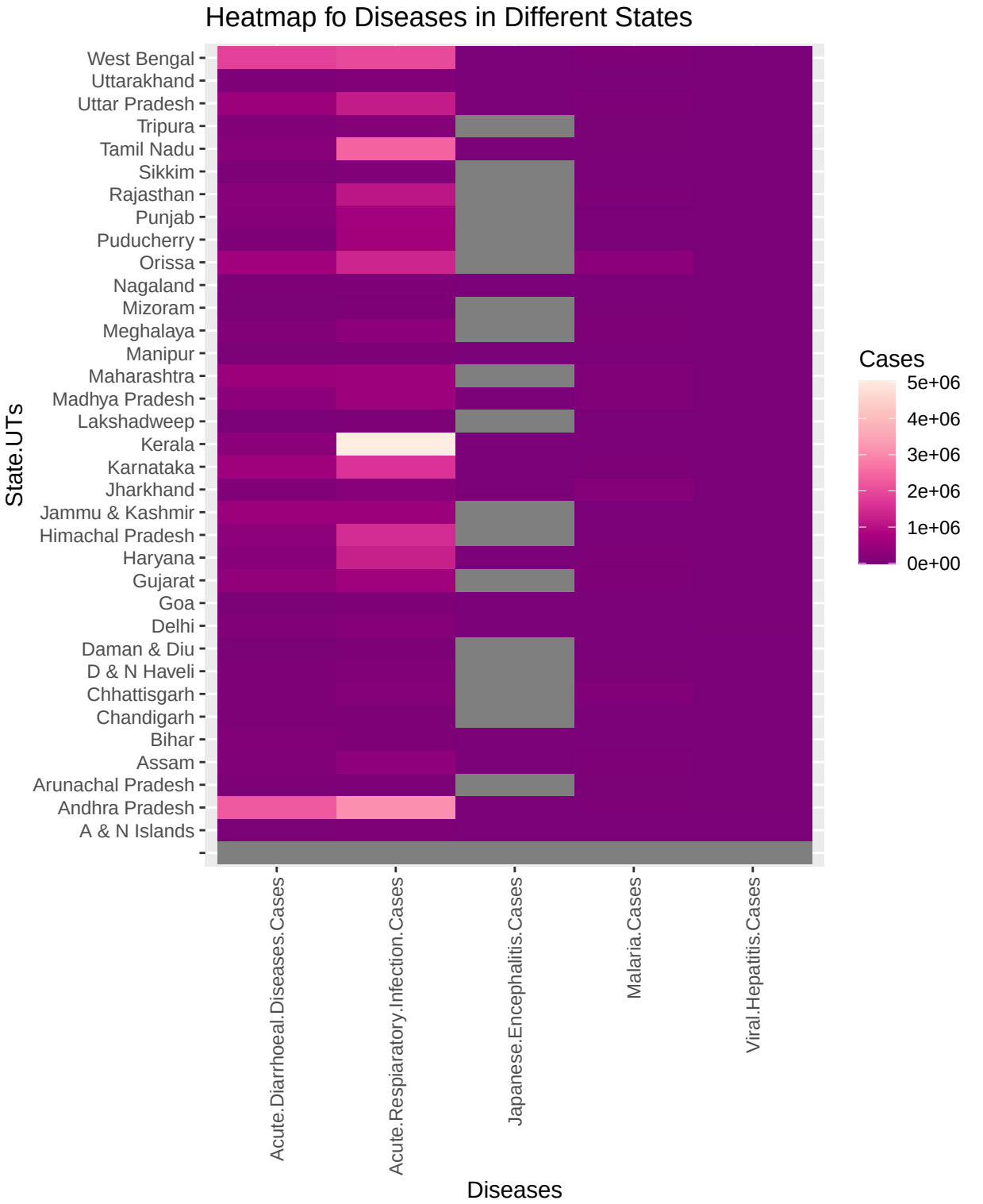
A circular barplot is a barplot that is displayed on a circle instead of a line. Here, we use it to represent the mean cases of deaths due to each disease. Also, we include total missing values/NA to the plot using a lollipop plot style within the main circular barplot. The data is sorted based on the mean deaths to provide a good visual representation of decreasing data along the circle.

11. Heatmap fo Diseases in Different States

```
cases <- c("State.UTs","Acute.Diarrhoeal.Diseases.Cases", "Malaria.Cases",
           "Acute.Respiaratory.Infection.Cases", "Japanese.Encephalitis.Cases",
           "Viral.Hepatitis.Cases")

# Gather the data
cases_data <- data.frame(state_ut_data[cases]) %>%
  gather(key="Diseases", value="Cases", -State.UTs)
head(cases_data)
```

```
##           State.UTs                         Diseases   Cases
## 1    Andhra Pradesh Acute.Diarrhoeal.Diseases.Cases 2235614
## 2 Arunachal Pradesh Acute.Diarrhoeal.Diseases.Cases   32228
## 3             Assam Acute.Diarrhoeal.Diseases.Cases   96816
## 4             Bihar Acute.Diarrhoeal.Diseases.Cases  130276
## 5      Chhattisgarh Acute.Diarrhoeal.Diseases.Cases   64575
## 6             Delhi Acute.Diarrhoeal.Diseases.Cases  102983
```

```
# Heatmap
plot11 <- ggplot(cases_data, aes(Diseases, State.UTs, fill= Cases)) +
  scale_fill_distiller(palette = "RdPu")  + geom_tile() +
  labs(title="Heatmap fo Diseases in Different States") +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))

plot11
```



Heatmap is a graphical representation of individual values represented in the form of colors. Here we created a heatmap of disease in each state. So, the total count of each disease in a state is give as a color and from the intensity of this color we can identify which all states have more cases of that particular disease in a single glance.

12. Parallel Coordinate Plot for the Dataset

```
# Load library for parallel coordinate plot
library(GGally)
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

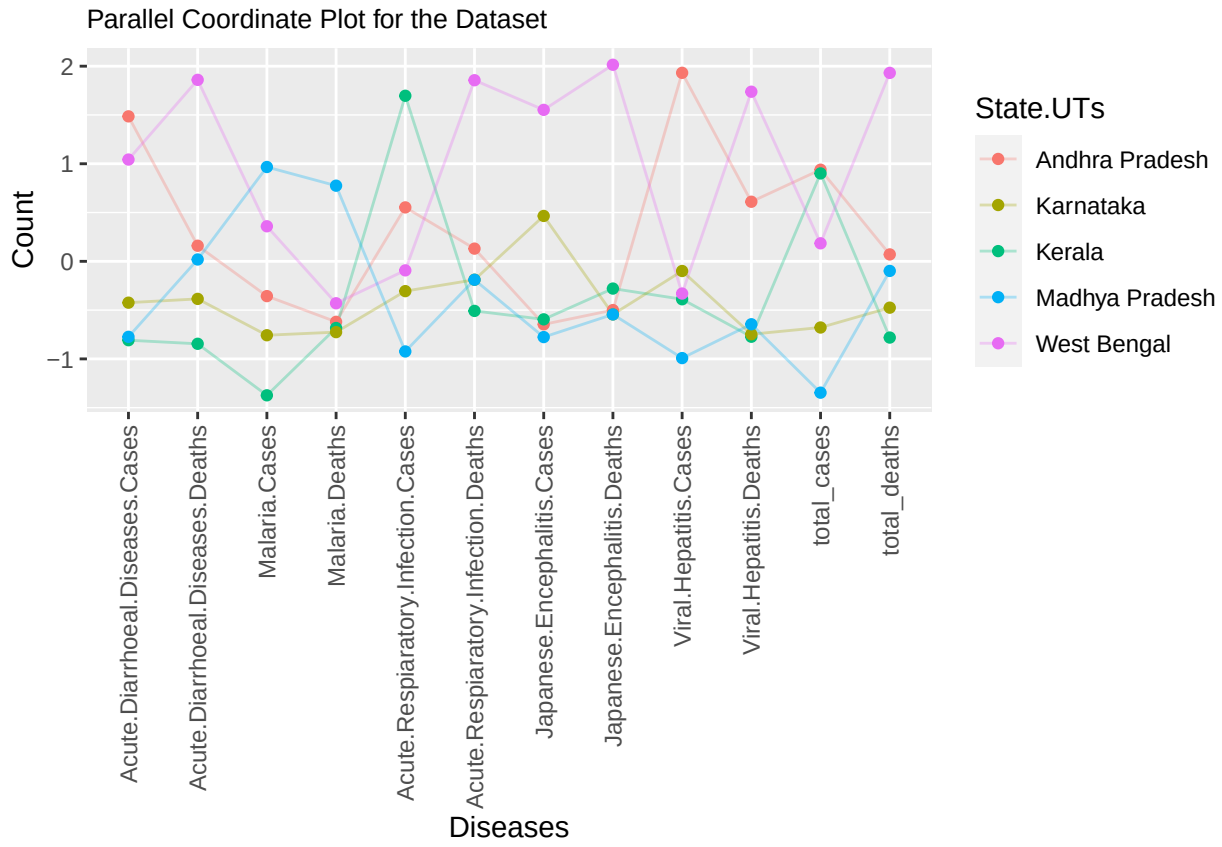
```
# Filter top states with most number of cases
top_states <- state_ut_data %>% arrange(desc(total_cases)) %>% top_n(6)
```

```
## Selecting by total_deaths
```

```
top_states[,1]
```

```
## [1] Andhra Pradesh Kerala         West Bengal    Karnataka      Madhya Pradesh
## [6] Maharashtra
## 37 Levels:  A & N Islands Andhra Pradesh Arunachal Pradesh Assam ... West Bengal
```

```
plot12 <- ggparcoord(top_states, showPoints = TRUE,
                     title = "Parallel Coordinate Plot for the Dataset",
                     alphaLines = 0.3, columns = 2:13, groupColumn = 1,
                     scale="std") +
  theme(plot.title = element_text(size=10)) +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1)) +
  xlab("Diseases") + ylab("Count")

plot12
```



With the *ggally* library and *ggparcoord()* function we can generate the parallel coordinate visualization of the whole dataset. This parallel coordinate representation is a good way to get a complete overview of high dimensional data. We can observe the disease cases and deaths across the top states from the parallel lines and each State/UT is given different colors which makes it easier to identify them.

Conclusion

Different types of visualizations using the state wise disease dataset is performed by aggregating, transforming and cleaning the original dataset. This dataset also contains some missing values which is also discussed here. The data is analysed in detail and different aspects of the data points like their relationships, composition, distribution of disease cases and deaths is visualized using different charts and the inference obtained from these visualizations are given below each chart.