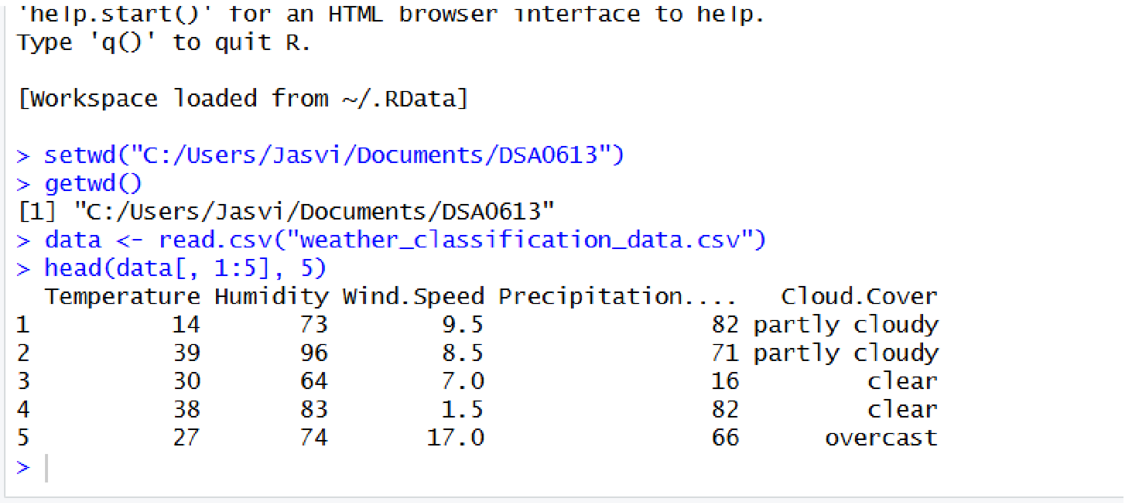
Name: Cheredla Jasvina **Reg. No.:** 192324026

Course Code: DSA0613

Course Name: Data Handling and Visualization for Data Analytics

1. Visualizing Amounts and Distributions



```
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

[Workspace loaded from ~/.RData]

> setwd("C:/Users/Jasvi/Documents/DSA0613")
> getwd()
[1] "C:/Users/Jasvi/Documents/DSA0613"
> data <- read.csv("weather_classification_data.csv")
> head(data[, 1:5], 5)
  Temperature Humidity Wind.Speed Precipitation....   Cloud.Cover
1          14       73        9.5                82 partly cloudy
2          39       96        8.5                71 partly cloudy
3          30       64        7.0                16         clear
4          38       83        1.5                82         clear
5          27       74       17.0                66      overcast
> 
```

setwd("C:/Users/Jasvi/Documents/DSA0613")

getwd()

# Read dataset

data <- read.csv("weather_classification_data.csv")

head(data[, 1:5], 5)

# Install and load packages

install.packages("ggplot2")

install.packages("ggridges")

install.packages("reshape2")

library(ggplot2)

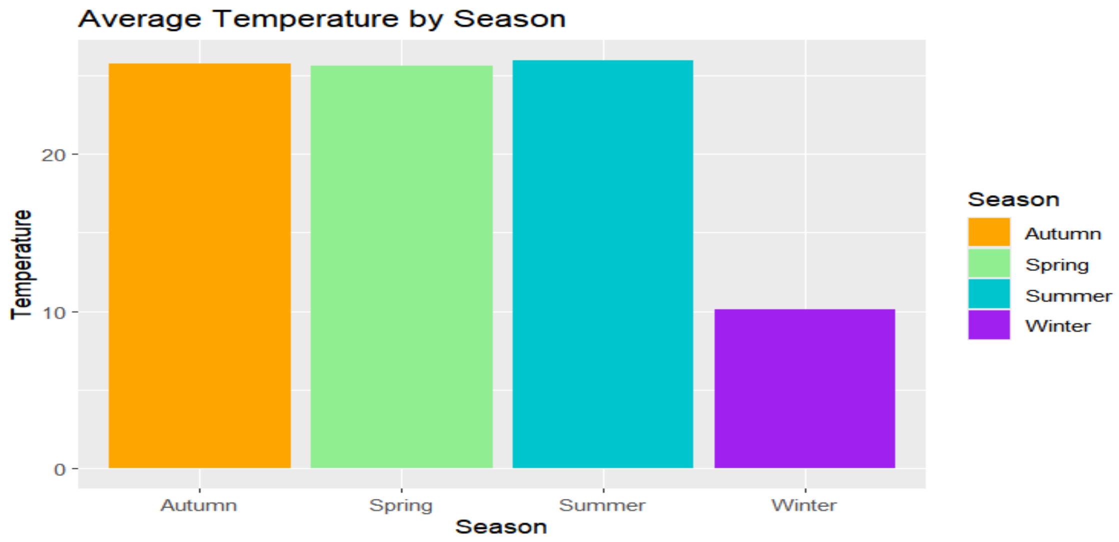
library(ggridges)

library(reshape2)

1. Bar Plot – Average Temperature by Season

```
ggplot(data, aes(x=Season, y=Temperature, fill=Season)) +

 stat_summary(fun=mean, geom="bar") +

 scale_fill_manual(values = c(

  "Winter" = "purple",

  "Spring" = "lightgreen",

  "Summer" = "turquoise3",

  "Autumn" = "orange"

 )) +

 labs(title = "Average Temperature by Season",

    x = "Season",

    y = "Temperature")
```

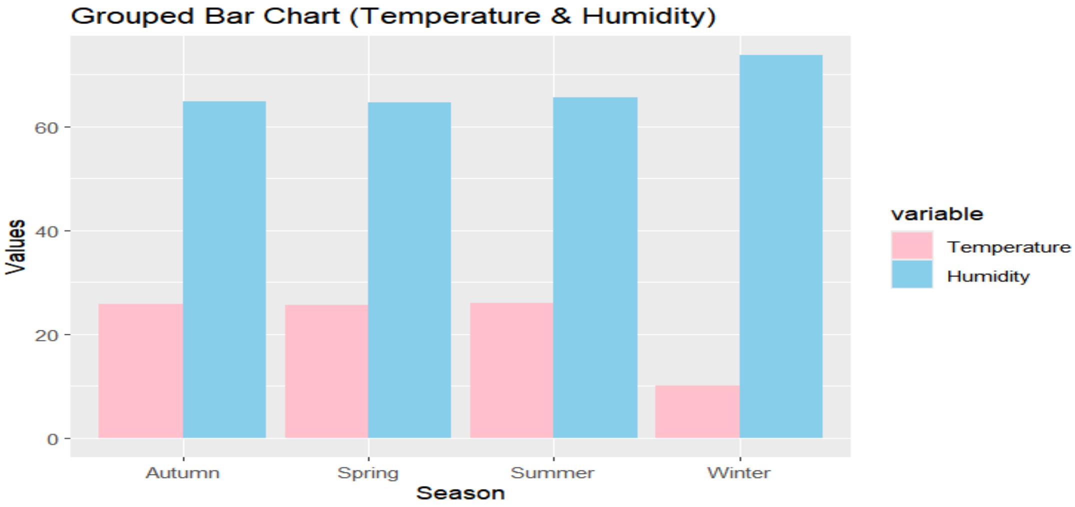
OUTPUT :



2. Grouped Bar Plot – Temperature & Humidity

```
data2 <- aggregate(cbind(Temperature, Humidity) ~ Season, data, mean)

data_melt <- melt(data2, id.vars="Season")

ggplot(data_melt, aes(x=Season, y=value, fill=variable)) +
 geom_bar(stat="identity", position="dodge") +
 scale_fill_manual(values=c("Temperature"="pink","Humidity"="skyblue")) +
 labs(title="Grouped Bar Chart (Temperature & Humidity)",
      x="Season", y="Values")
```

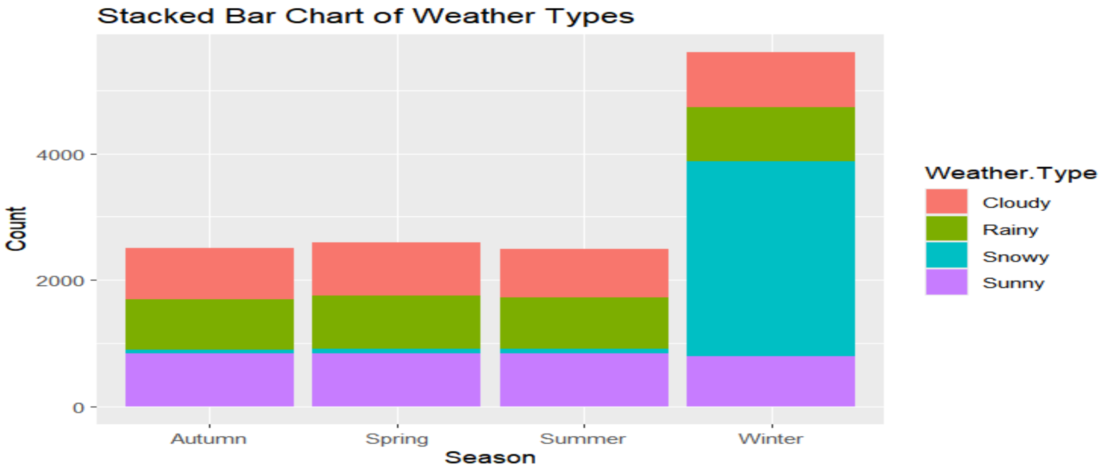
OUTPUT :



3. Stacked Bar Plot – Weather Type by Season

```
ggplot(data, aes(x=Season, fill=Weather.Type)) +
 geom_bar() +
 labs(title="Stacked Bar Chart of Weather Types",
      x="Season", y="Count")
```

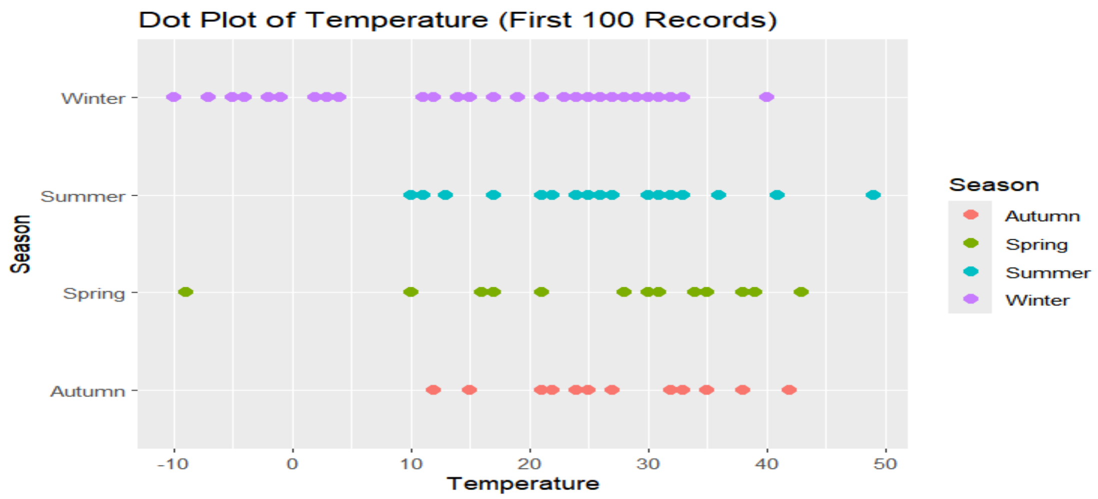
OUTPUT :



4. Dot Plot – Temperature vs Season

```
ggplot(data[1:100, ], aes(x = Temperature, y = Season, color = Season)) +
  geom_point(size = 2.5) +
  labs(title = "Dot Plot of Temperature (First 100 Records)",
       x = "Temperature",
       y = "Season")
```

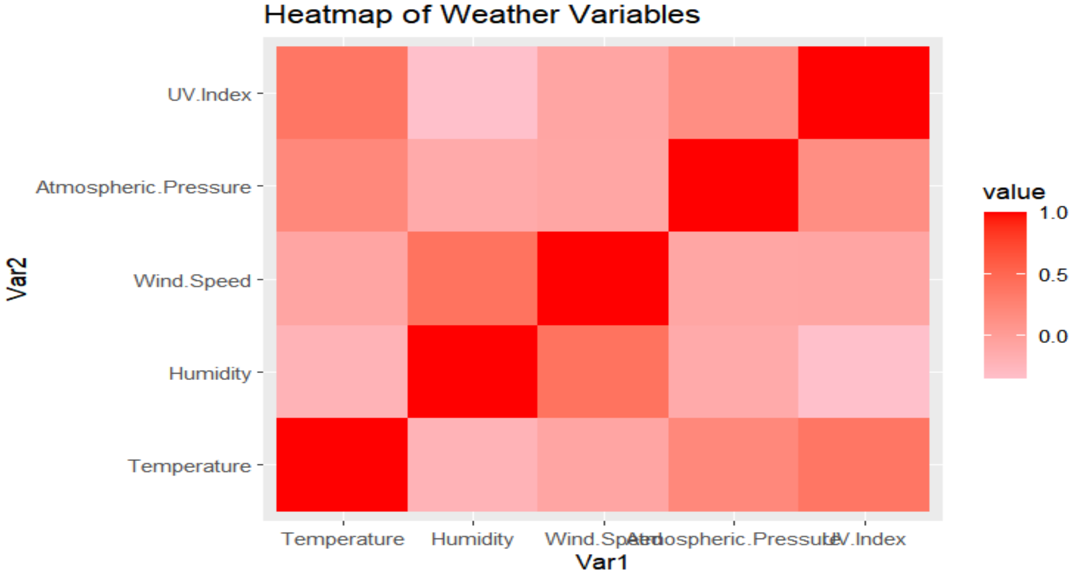
OUTPUT :



5. Heatmap – Correlation of Variables

```
num_data <- data[, c("Temperature","Humidity","Wind.Speed",
                     "Atmospheric.Pressure","UV.Index")]
corr <- cor(num_data)
corr_melt <- melt(corr)
ggplot(corr_melt, aes(Var1, Var2, fill=value)) +
  geom_tile() +
  scale_fill_gradient(low="pink", high="red") +
  labs(title="Heatmap of Weather Variables")
```

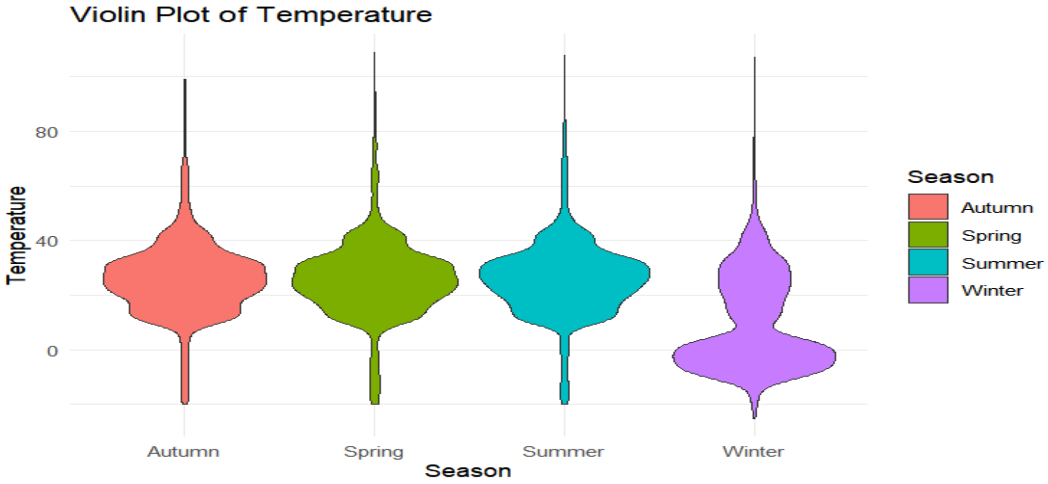
OUTPUT :



6. Violin Plot – Temperature by Season

```
ggplot(data, aes(x = Season, y = Temperature, fill = Season)) +
  geom_violin() +
  labs(title = "Violin Plot of Temperature",
       x = "Season",
       y = "Temperature") +
  theme_minimal()
```

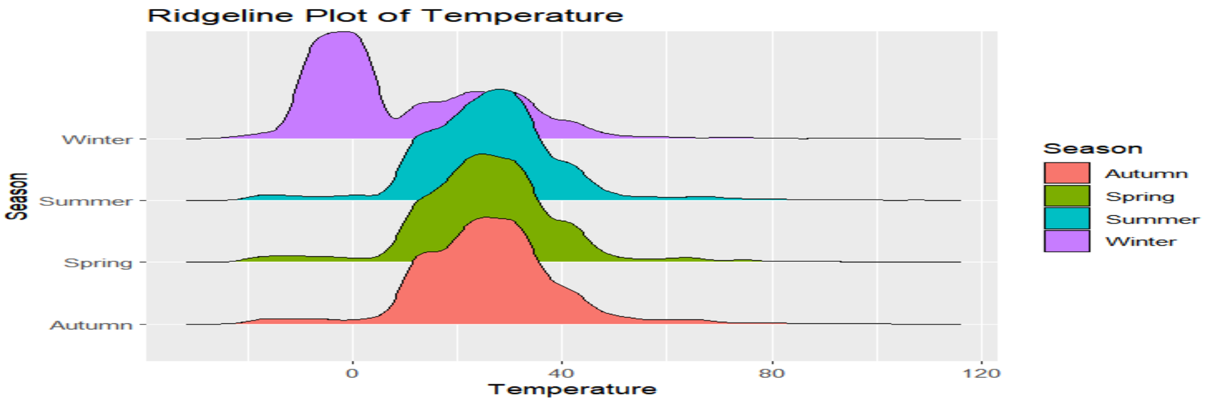
OUTPUT :



7. Ridgeline Plot – Temperature Distribution

```
ggplot(data, aes(x=Temperature, y=Season, fill=Season)) +
  geom_density_ridges() +
  labs(title="Ridgeline Plot of Temperature")
```

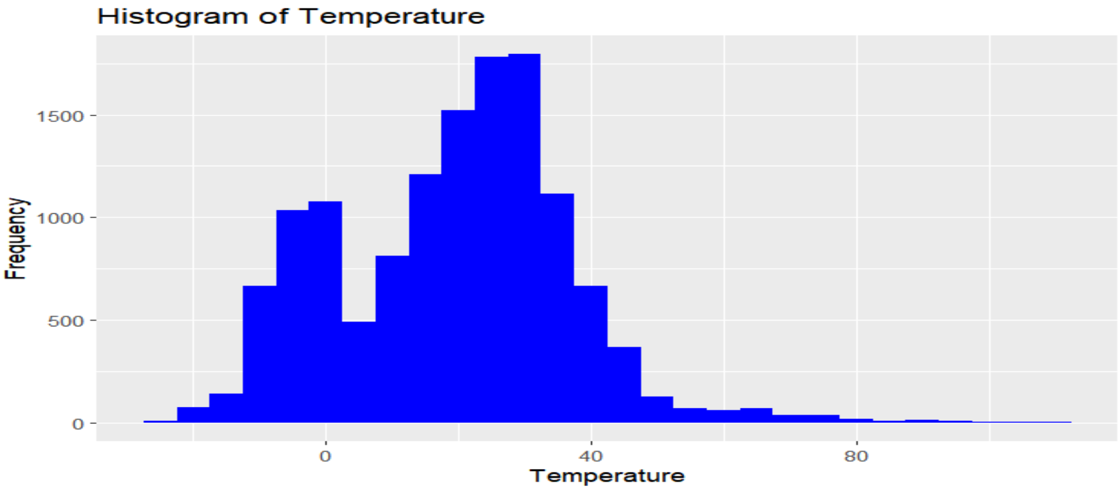
OUTPUT :



8. Histogram – Temperature

```
ggplot(data, aes(x=Temperature)) +
geom_histogram(binwidth=5, fill="blue") +
 labs(title="Histogram of Temperature",
  x="Temperature", y="Frequency")
```

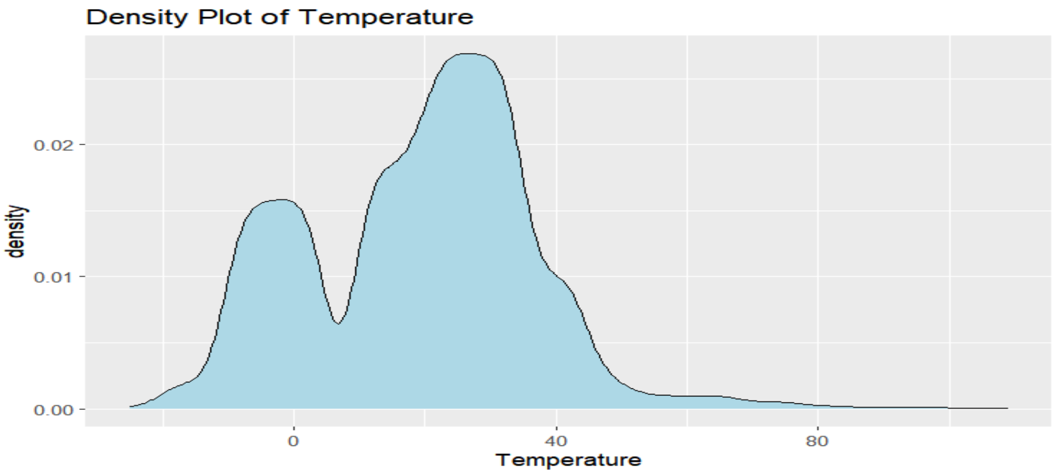
OUTPUT :



9. Density Plot – Temperature

```
ggplot(data, aes(x=Temperature)) +
  geom_density(fill="lightblue") +
  labs(title="Density Plot of Temperature")
```

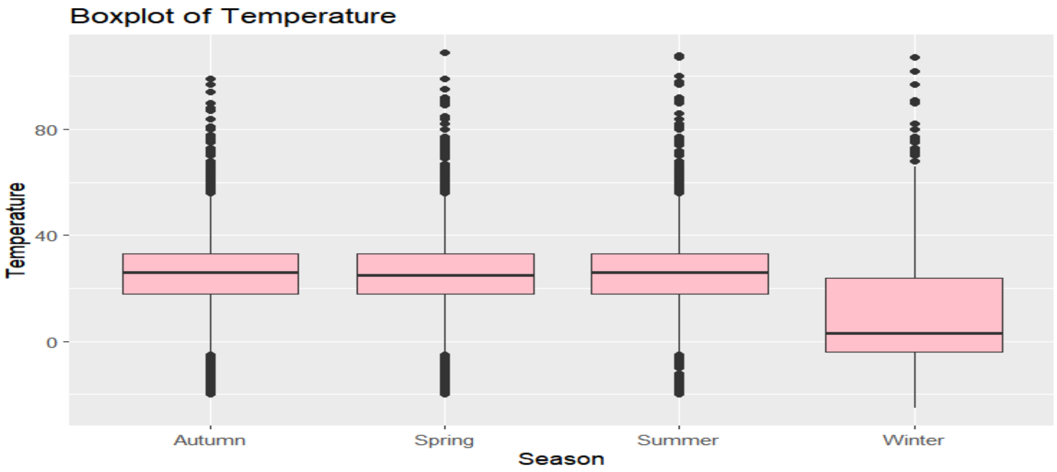
OUTPUT :



10. Boxplot – Temperature by Season

```
ggplot(data, aes(x=Season, y=Temperature)) +
  geom_boxplot(fill="pink") +
  labs(title="Boxplot of Temperature",
       x="Season", y="Temperature")
```

OUTPUT :

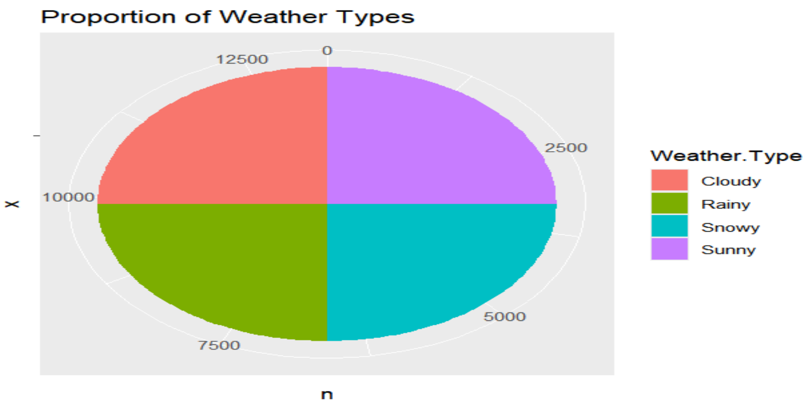


2.Visualizing Proportions

1. PIE CHART (Weather Type)

```
pie_data <- data %>% count(Weather.Type)
ggplot(pie_data, aes(x="", y=n, fill=Weather.Type)) +
  geom_bar(stat="identity") +
  coord_polar("y") +
  labs(title="Proportion of Weather Types")
```

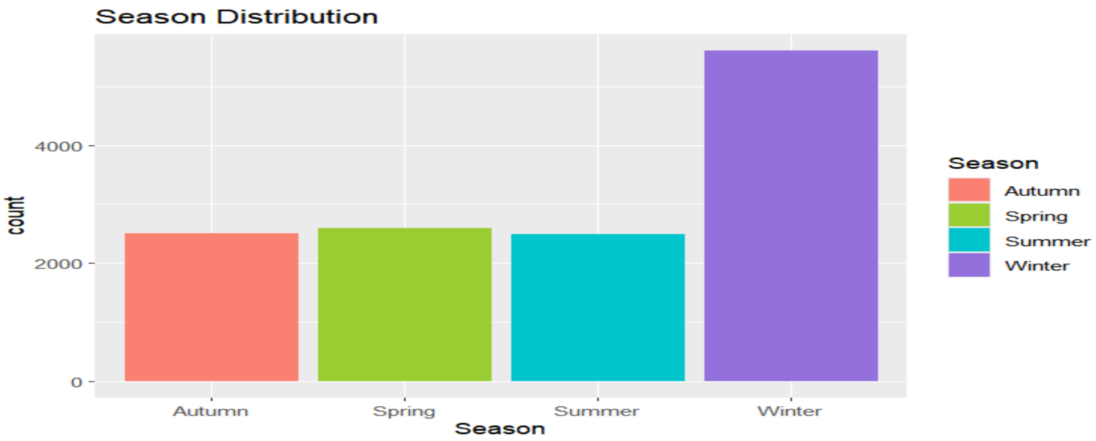
OUTPUT :



2. STACKED BAR (Season vs Weather Type)

```
ggplot(data, aes(x=Season, fill=Season)) +
  geom_bar() +
  scale_fill_manual(values=c(
    "Autumn"="salmon",
    "Spring"="olivedrab3",
    "Summer"="turquoise3",
    "Winter"="mediumpurple"
  )) +
  labs(title="Season Distribution")
```

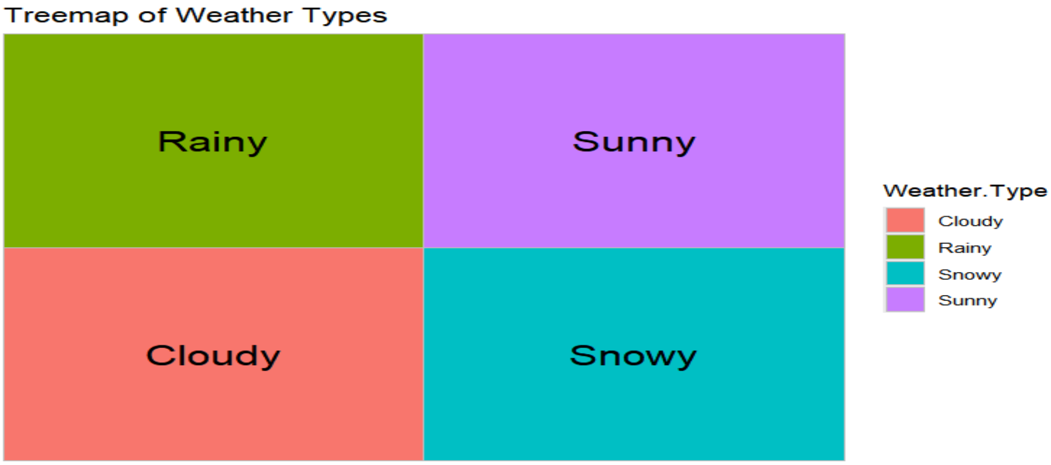
OUTPUT :



3. TREEMAP (Weather Type)

```
ggplot(pie_data, aes(area=n, fill=Weather.Type, label=Weather.Type)) +
  geom_treemap() +
  geom_treemap_text(colour="black", place="centre") +
  labs(title="Treemap of Weather Types")
```

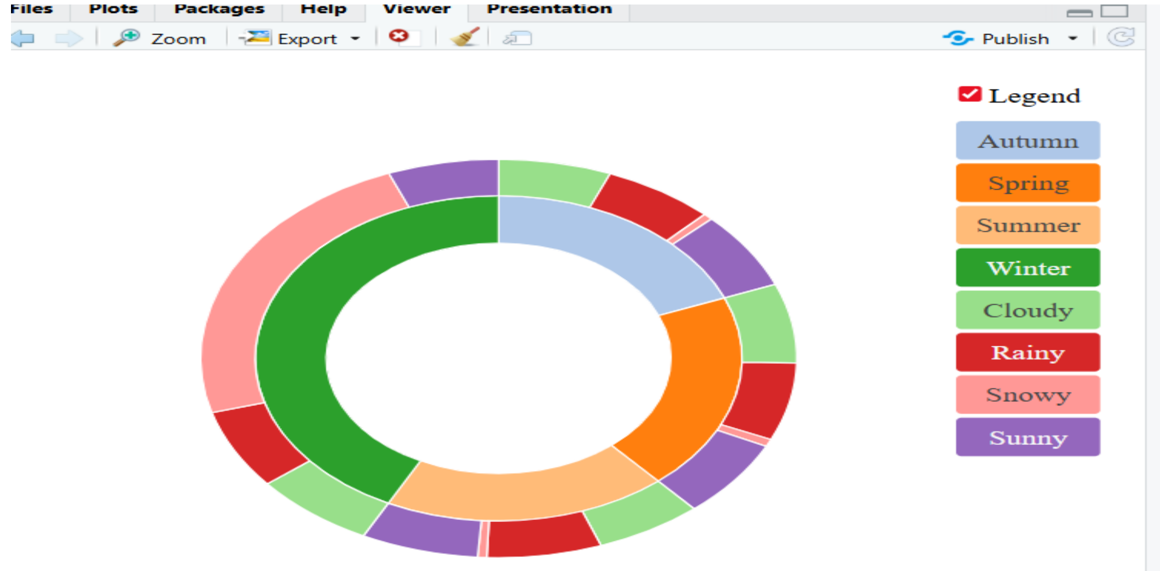
OUTPUT :

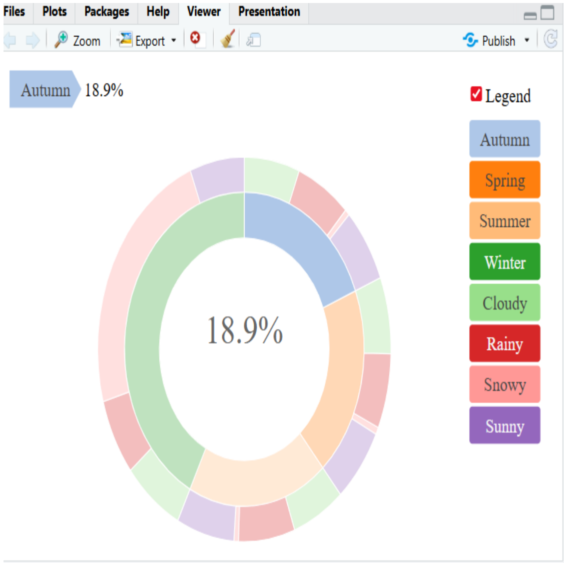


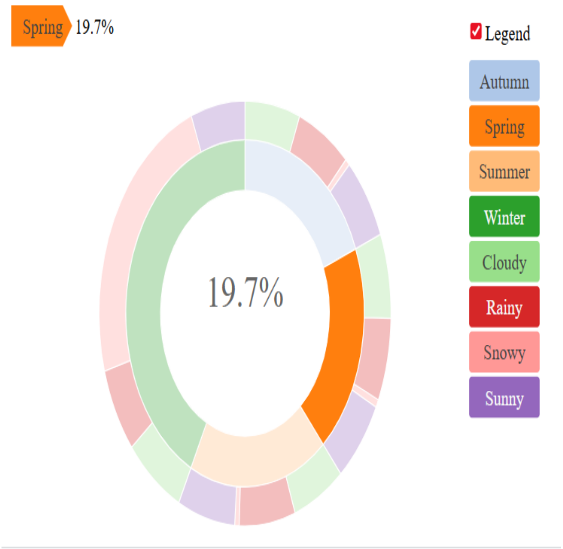
4. SUNBURST (Season → Weather Type)

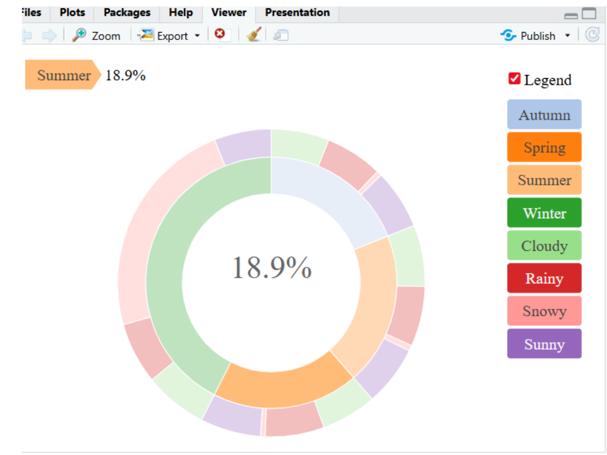
```
sun_data <- data %>% count(Season, Weather.Type)
sun_data2 <- data.frame(
  seq = paste(sun_data$Season, sun_data$Weather.Type, sep="-"),
  freq = sun_data$n
)
sunburst(sun_data2)
```

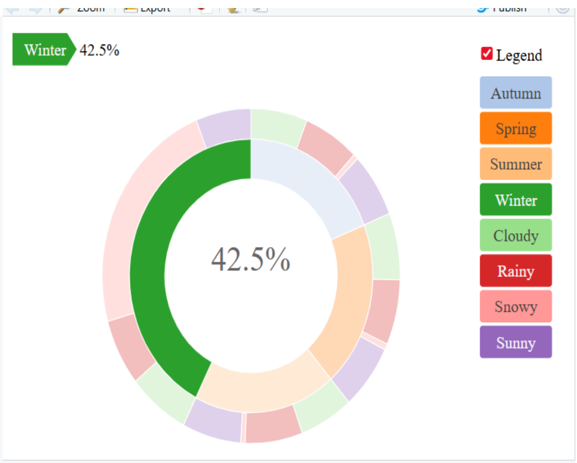
OUTPUT :

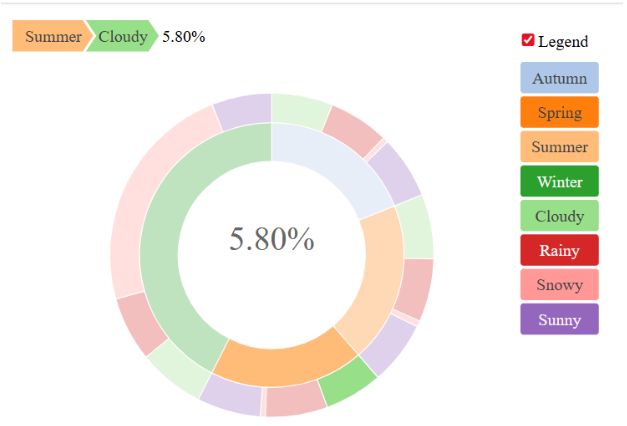


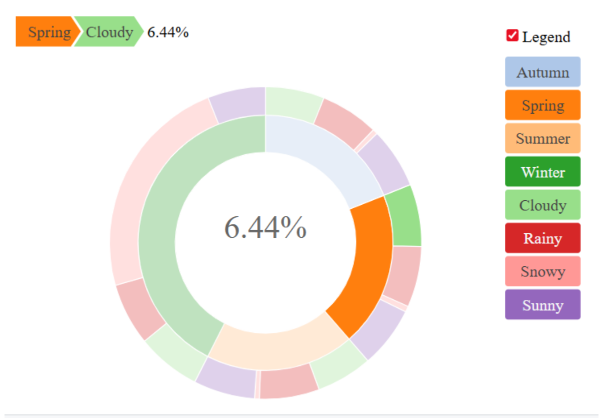


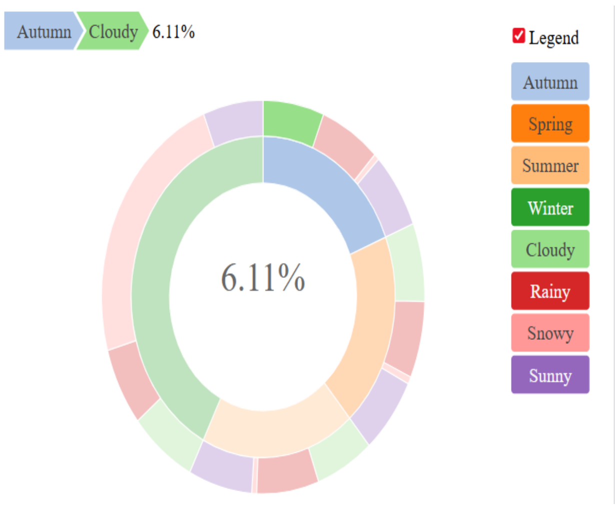


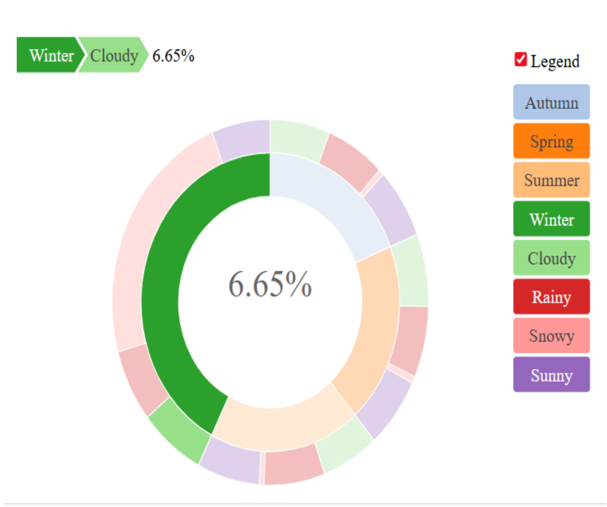

Files
Plots
Packages
Help
Viewer
Presentation
Zoom
Export
Publish
Summer
18.9%
Legend
Autumn
Spring
Summer
Winter
Cloudy
Rainy
Snowy
Sunny
18.9%

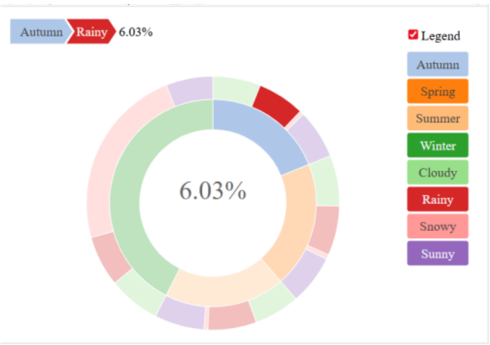

Zoom
Export
Publish
Winter
42.5%
Legend
Autumn
Spring
Summer
Winter
Cloudy
Rainy
Snowy
Sunny
42.5%

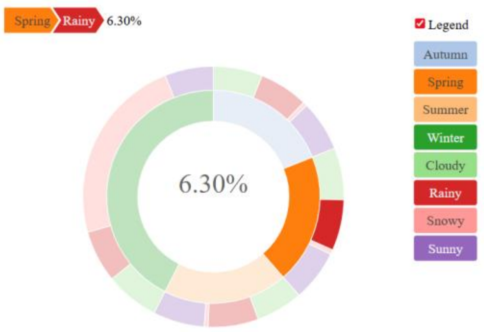

Summer
Cloudy
5.80%
Legend
Autumn
Spring
Summer
Winter
Cloudy
Rainy
Snowy
Sunny
5.80%

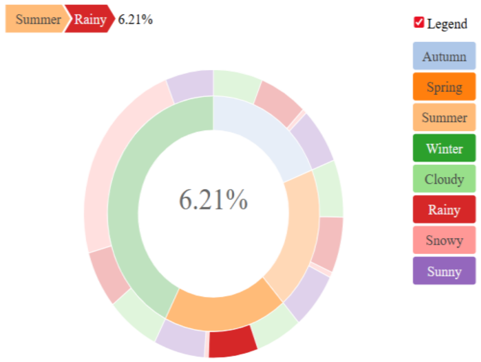

Spring
Cloudy
6.44%
Legend
Autumn
Spring
Summer
Winter
Cloudy
Rainy
Snowy
Sunny
6.44%

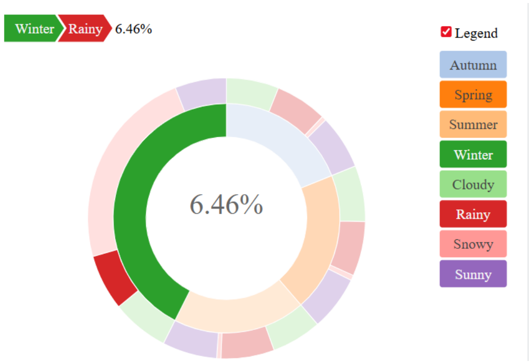

Autumn
Cloudy
6.11%
Legend
Autumn
Spring
Summer
Winter
Cloudy
Rainy
Snowy
Sunny
6.11%

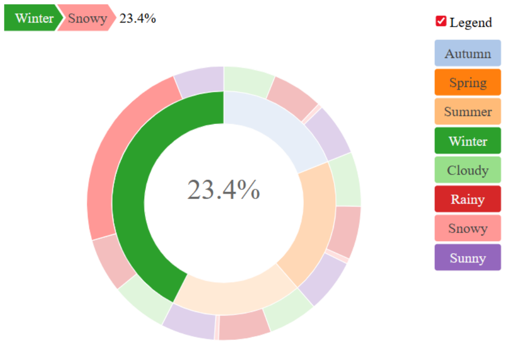

Winter
Cloudy
6.65%
Legend
Autumn
Spring
Summer
Winter
Cloudy
Rainy
Snowy
Sunny
6.65%

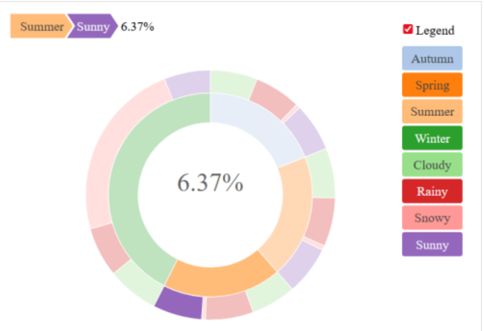

Autumn
Rainy
6.03%
6.03%
☑ Legend
Autumn
Spring
Summer
Winter
Cloudy
Rainy
Snowy
Sunny

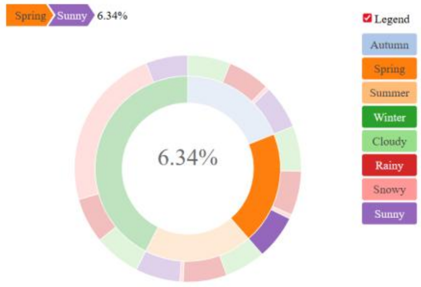

Spring
Rainy
6.30%
6.30%
☑ Legend
Autumn
Spring
Summer
Winter
Cloudy
Rainy
Snowy
Sunny

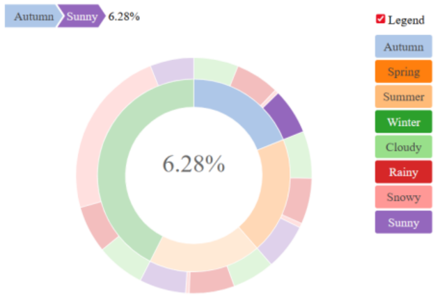

Summer
Rainy
6.21%
6.21%
☑ Legend
Autumn
Spring
Summer
Winter
Cloudy
Rainy
Snowy
Sunny

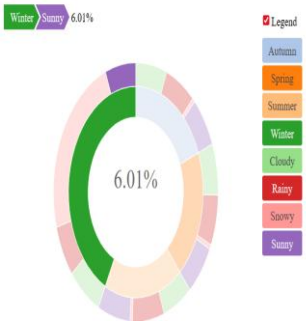

Winter
Rainy
6.46%
6.46%
☑ Legend
Autumn
Spring
Summer
Winter
Cloudy
Rainy
Snowy
Sunny


Winter
Snowy
23.4%
23.4%
☑ Legend
Autumn
Spring
Summer
Winter
Cloudy
Rainy
Snowy
Sunny


Summer
Sunny
6.37%
6.37%
☑ Legend
Autumn
Spring
Summer
Winter
Cloudy
Rainy
Snowy
Sunny


Spring
Sunny
6.34%
6.34%
☑ Legend
Autumn
Spring
Summer
Winter
Cloudy
Rainy
Snowy
Sunny

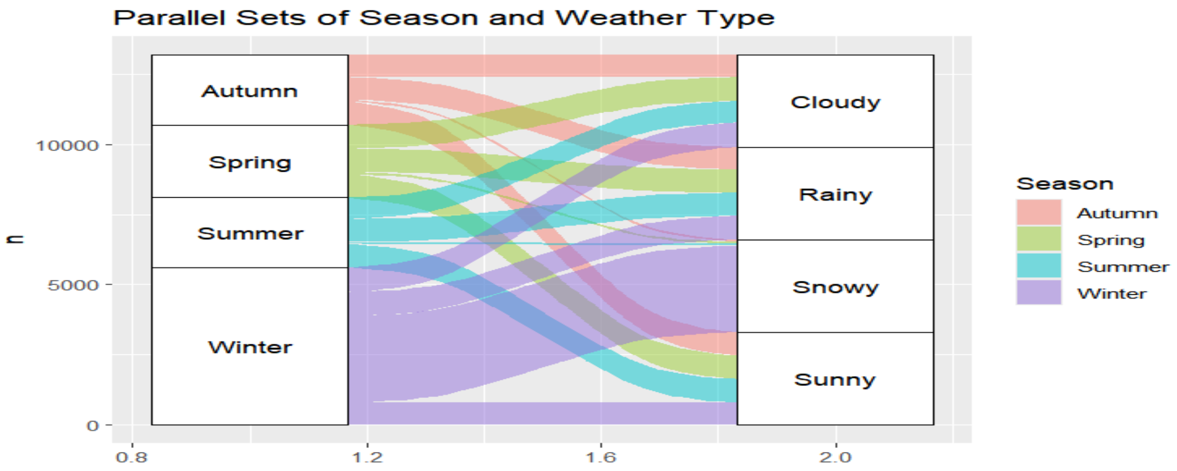

Autumn
Sunny
6.28%
6.28%
☑ Legend
Autumn
Spring
Summer
Winter
Cloudy
Rainy
Snowy
Sunny


Winter
Sunny
6.01%
6.01%
☑ Legend
Autumn
Spring
Summer
Winter
Cloudy
Rainy
Snowy
Sunny

5. PARALLEL SETS

```
alluvial_data <- data %>% count(Season, Weather.Type)

ggplot(alluvial_data,

       aes(axis1=Season, axis2=Weather.Type, y=n)) +

  geom_alluvium(aes(fill=Season)) +

  geom_stratum() +

  geom_text(stat="stratum", aes(label=after_stat(stratum))) +

  scale_fill_manual(values=c(

    "Autumn"="salmon",

    "Spring"="olivedrab3",

    "Summer"="turquoise3",

    "Winter"="mediumpurple"

  )) +

  labs(title="Parallel Sets of Season and Weather Type")
```

OUTPUT :

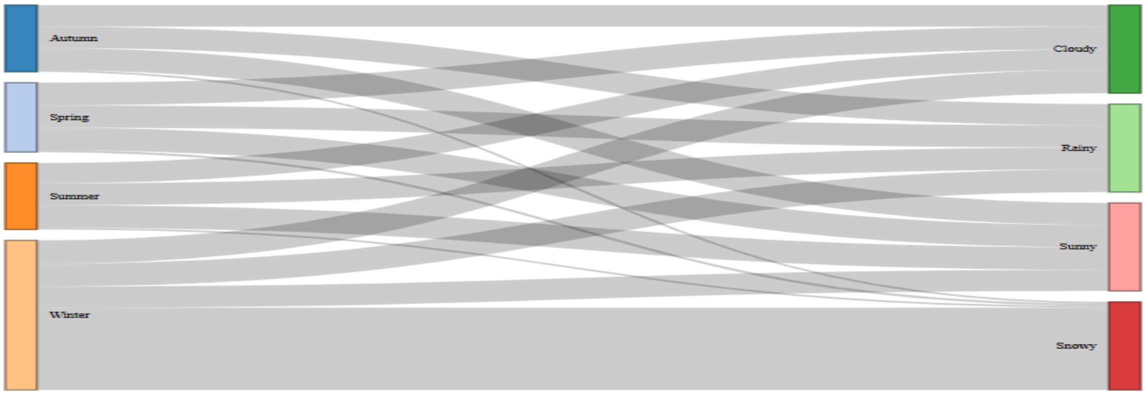


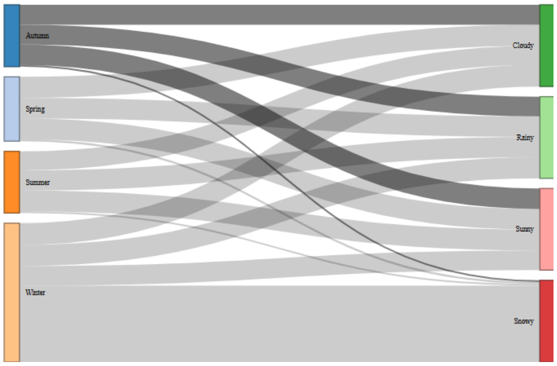
6. SANKEY DIAGRAM

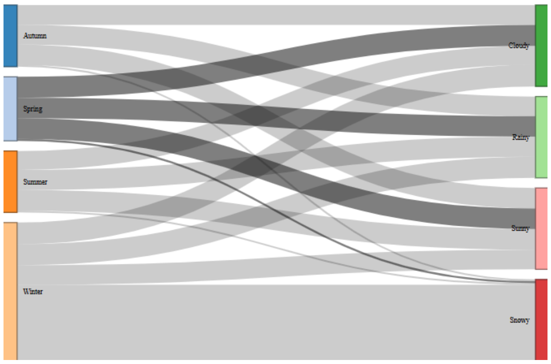
```
links <- data %>% count(Season, Weather.Type)

nodes <- data.frame(name = unique(c(links$Season, links$Weather.Type)))

links$source <- match(links$Season, nodes$name)-1
```

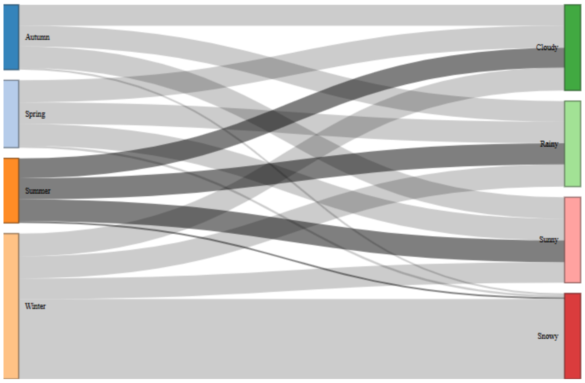
```
links$target <- match(links$Weather.Type, nodes$name)-1
sankeyNetwork(Links=links, Nodes=nodes,
          Source="source", Target="target",
          Value="n", NodeID="name")
```

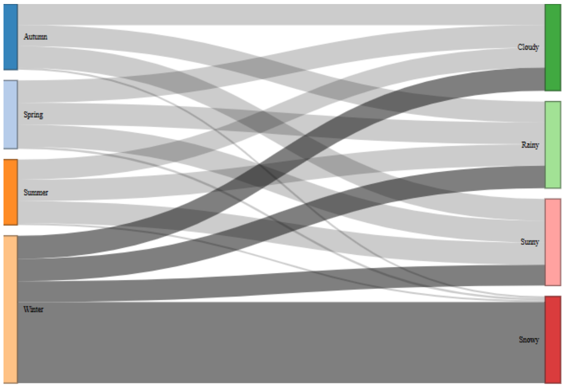
OUTPUT :









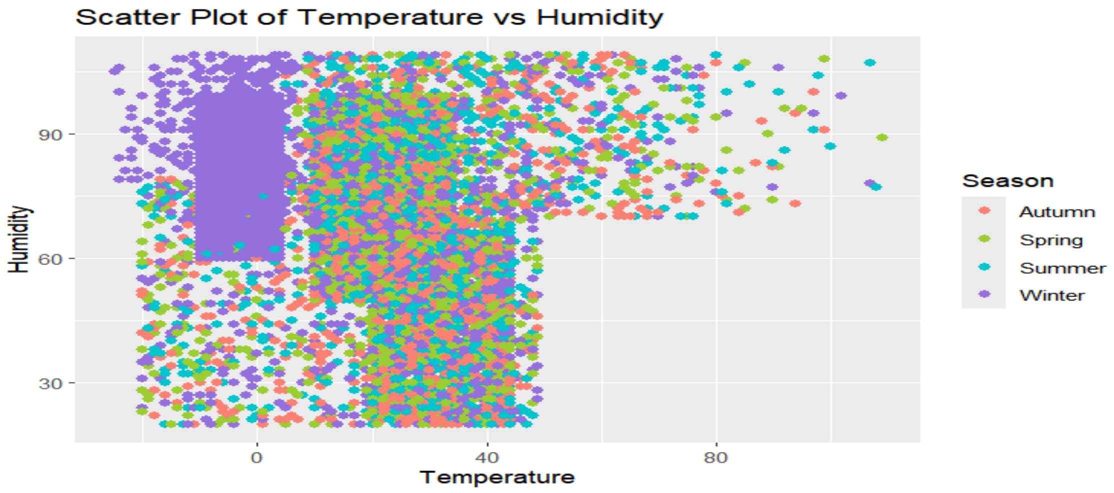


3.Visualizing Relationships and Associations in Weather Data Using R

1. Scatter Plot

```
ggplot(data, aes(x=Temperature, y=Humidity, color=Season)) +
  geom_point(size=2) +
  scale_color_manual(values = season_colors) +
  labs(title="Scatter Plot of Temperature vs Humidity")
```

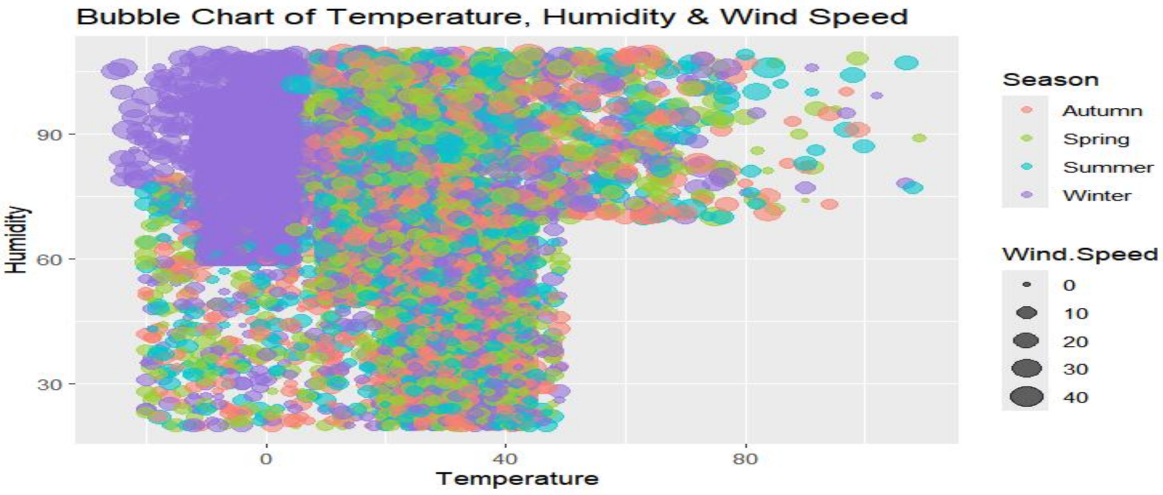
OUTPUT :



2. Bubble Chart

```
ggplot(data, aes(x=Temperature, y=Humidity, size=Wind.Speed, color=Season)) +
  geom_point(alpha=0.6) +
  scale_color_manual(values = season_colors) +
  labs(title="Bubble Chart of Temperature, Humidity & Wind Speed")
```

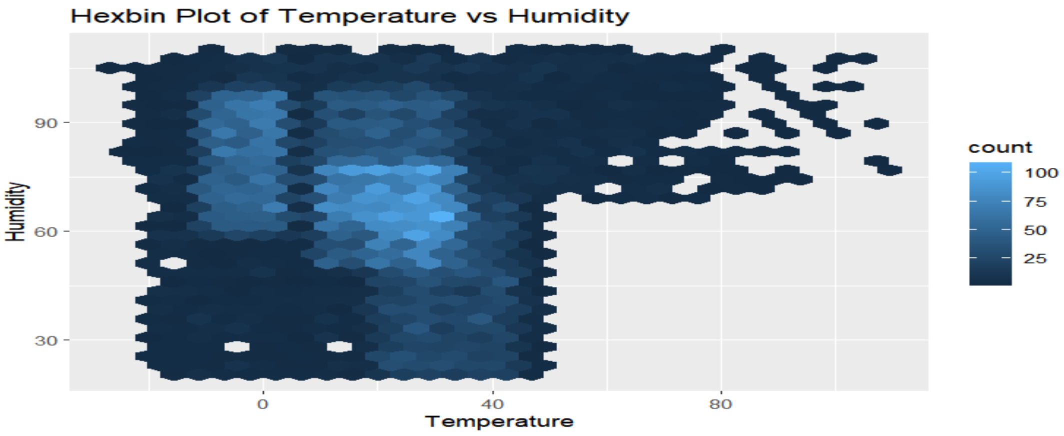
OUTPUT :



3. Hexbin Plot

```
ggplot(data, aes(x=Temperature, y=Humidity)) +
  geom_hex() +
  labs(title="Hexbin Plot of Temperature vs Humidity")
```

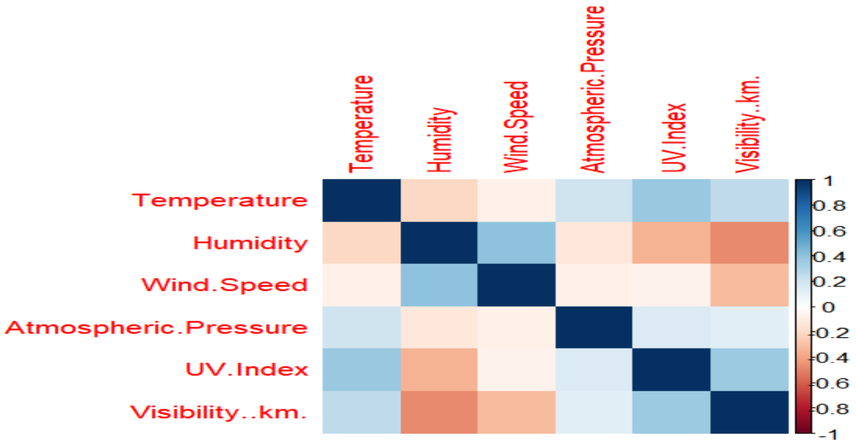
OUTPUT :



4. Correlogram

```
num_data <- data[, c("Temperature","Humidity","Wind.Speed",
                     "Atmospheric.Pressure","UV.Index","Visibility..km.")]
corr <- cor(num_data, use="complete.obs")
corrplot(corr, method="color")
```

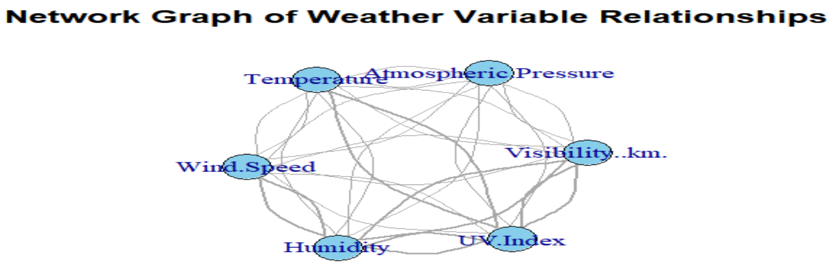
OUTPUT :



5. Network Graph

```
edges <- data.frame(
  from = rep(colnames(num_data), each = ncol(num_data)),
  to   = rep(colnames(num_data), times = ncol(num_data)),
  weight = as.vector(corr)
)
edges <- edges[edges$from != edges$to, ]   # remove only self-links
graph <- graph_from_data_frame(edges, directed = FALSE)
plot(graph,
     vertex.color = "skyblue",
     vertex.size = 30,
     edge.width = abs(E(graph)$weight) * 5,
     main = "Network Graph of Weather Variable Relationships")
```

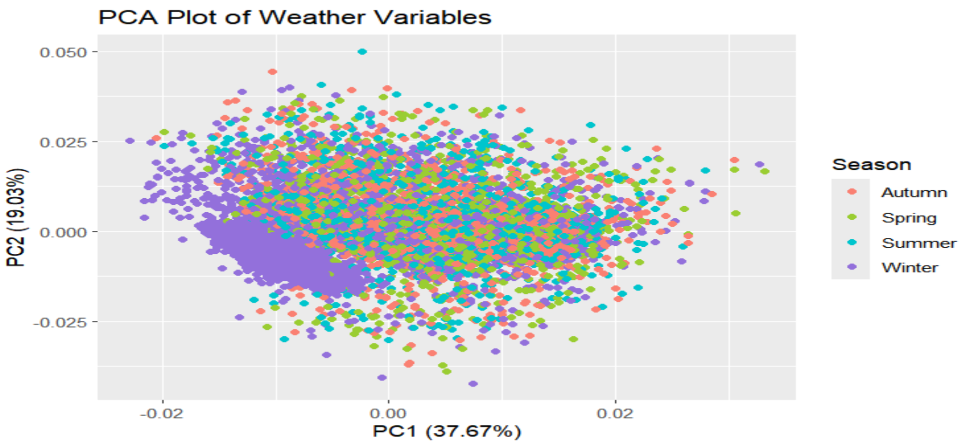
OUTPUT :



6. PCA

```
pca <- prcomp(num_data, scale.=TRUE)
autoplot(pca, data=data, colour="Season") +
  scale_color_manual(values = season_colors) +
  labs(title="PCA Plot of Weather Variables")
suppressWarnings(
  autoplot(pca, data = data, colour = "Season") +
    scale_color_manual(values = season_colors) +
    labs(title = "PCA Plot of Weather Variables")
)
```

OUTPUT :


PCA Plot of Weather Variables
PC2 (19.03%)
PC1 (37.67%)
Season
Autumn
Spring
Summer
Winter