

# LAB 1

NAME : GYANADA

REG.NO: 21BCE7727

1.Explore different types of graphs with your own data using Python / R.

```
In [66]: install.packages("ggplot2")
```

```
Warning message:  
"package 'ggplot2' is in use and will not be installed"
```

```
In [67]: library(ggplot2)
```

```
In [68]: install.packages('tidyverse')  
library("tidyverse")
```

```
Warning message:  
"package 'tidyverse' is in use and will not be installed"
```

```
In [69]: data()
```

```
In [70]: data(airquality)  
dim(airquality)  
head(airquality)  
summary(airquality)
```

153 · 6

A data.frame: 6 × 6

	Ozone	Solar.R	Wind	Temp	Month	Day
	<int>	<int>	<dbl>	<int>	<int>	<int>
1	41	190	7.4	67	5	1
2	36	118	8.0	72	5	2
3	12	149	12.6	74	5	3
4	18	313	11.5	62	5	4
5	NA	NA	14.3	56	5	5
6	28	NA	14.9	66	5	6

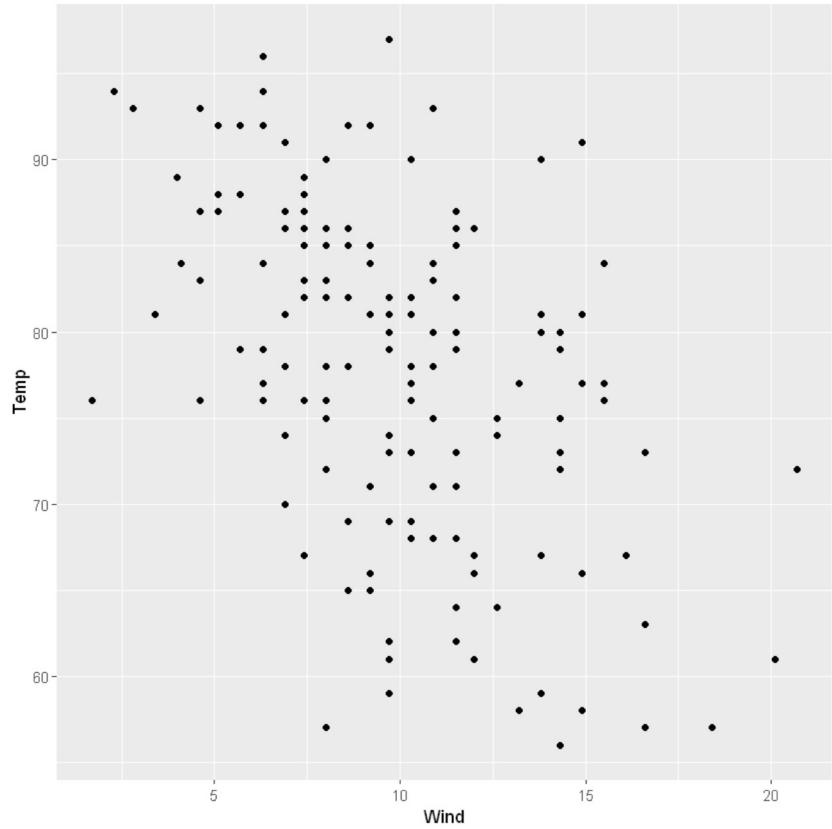
Ozone	Solar.R	Wind	Temp
Min. : 1.00	Min. : 7.0	Min. : 1.700	Min. :56.00
1st Qu.: 18.00	1st Qu.:115.8	1st Qu.: 7.400	1st Qu.:72.00
Median : 31.50	Median :205.0	Median : 9.700	Median :79.00
Mean : 42.13	Mean :185.9	Mean : 9.958	Mean :77.88
3rd Qu.: 63.25	3rd Qu.:258.8	3rd Qu.:11.500	3rd Qu.:85.00
Max. :168.00	Max. :334.0	Max. :20.700	Max. :97.00
NA's :37	NA's :7		

Month	Day
Min. :5.000	Min. : 1.0
1st Qu.:6.000	1st Qu.: 8.0
Median :7.000	Median :16.0
Mean :6.993	Mean :15.8
3rd Qu.:8.000	3rd Qu.:23.0
Max. :9.000	Max. :31.0

# 1) SCATTER PLOT

In [71]: `ggplot(airquality, aes(x=Wind, y=Temp))+geom_point()`

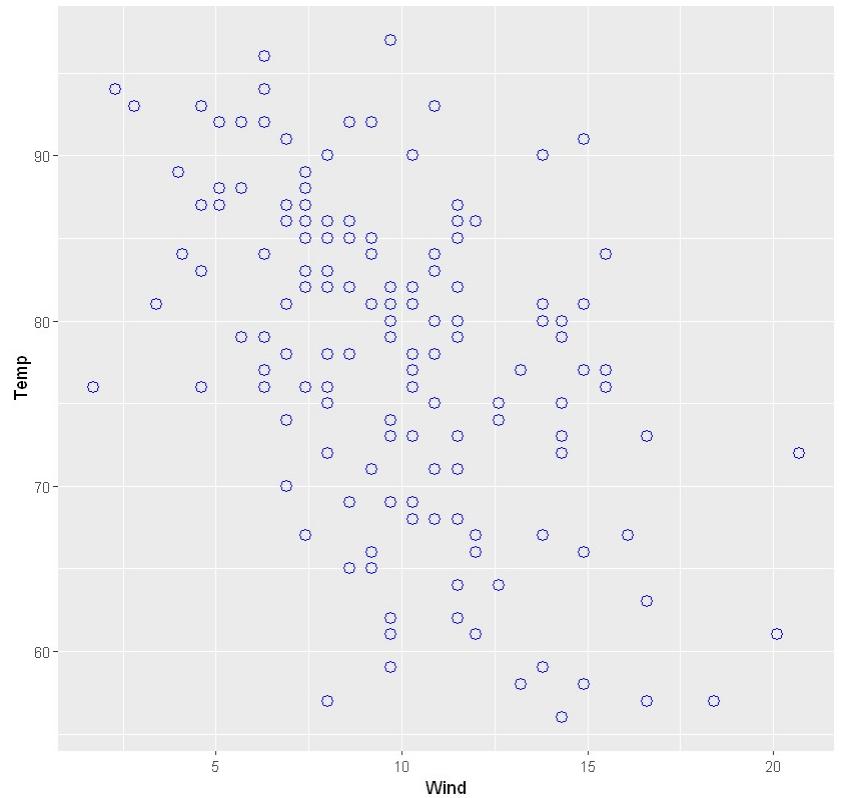


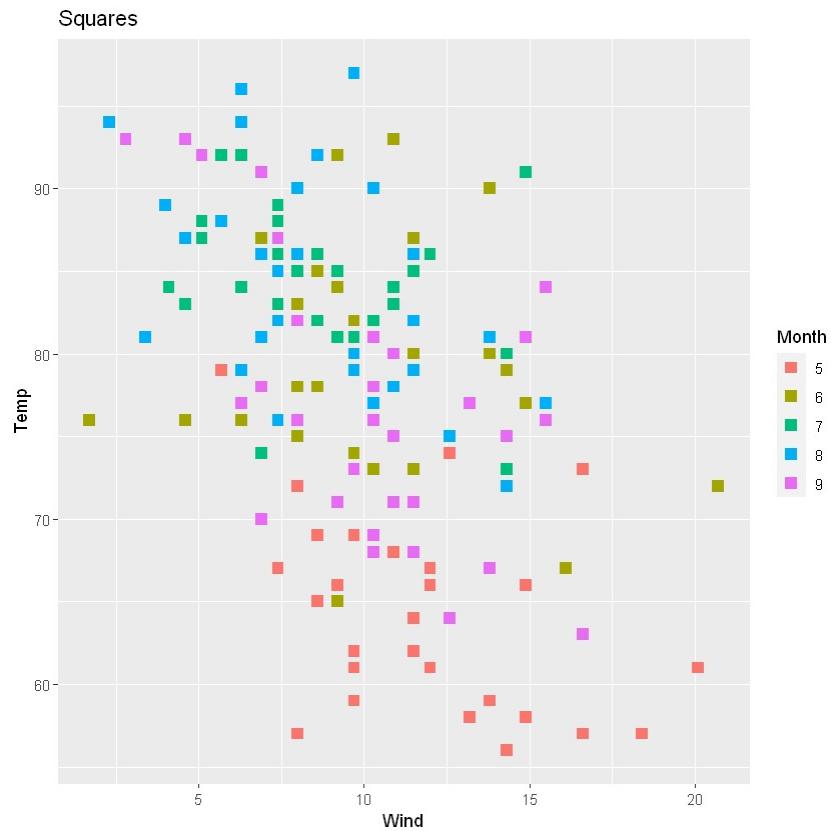
```
In [72]: # Scatter Plot with Circles
ggplot(airquality, aes(x = Wind, y = Temp)) +
  geom_point(shape = 1, size = 3, color = "blue") +
  labs(title = "Circles", x = "Wind", y = "Temp")

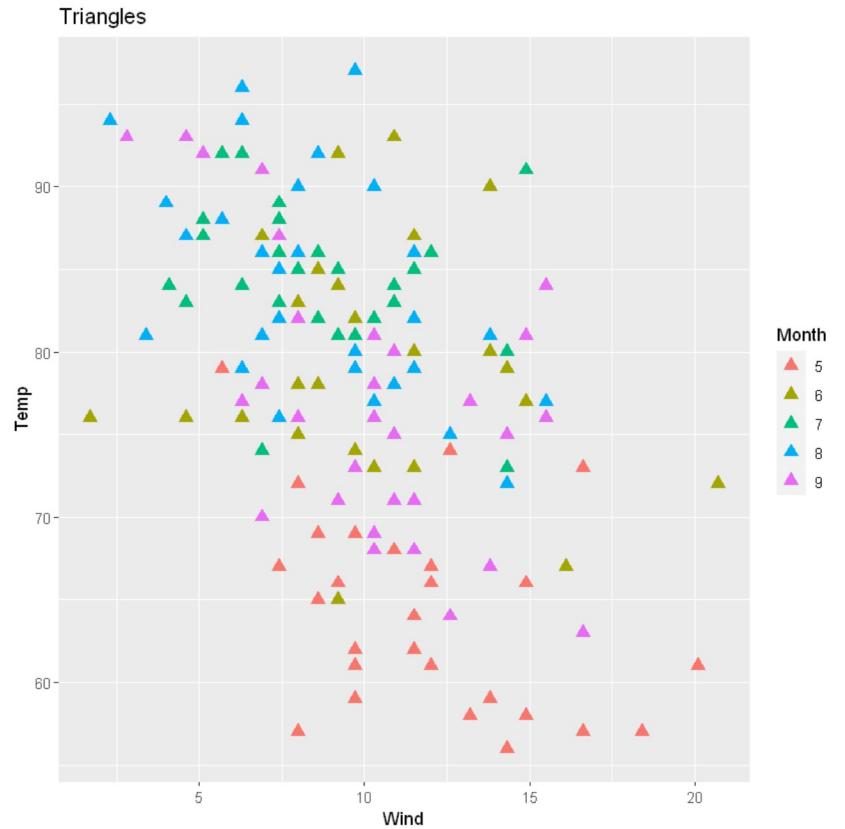
# Scatter Plot with Squares
ggplot(airquality, aes(x = Wind, y = Temp, col = factor(Month))) +
  geom_point(shape = 15, size = 3) +
  labs(title = "Squares", x = "Wind", y = "Temp", color = "Month")

# Scatter Plot with Triangles
ggplot(airquality, aes(x = Wind, y = Temp, col = factor(Month))) +
  geom_point(shape = 17, size = 3) +
  labs(title = "Triangles", x = "Wind", y = "Temp", color = "Month")
```

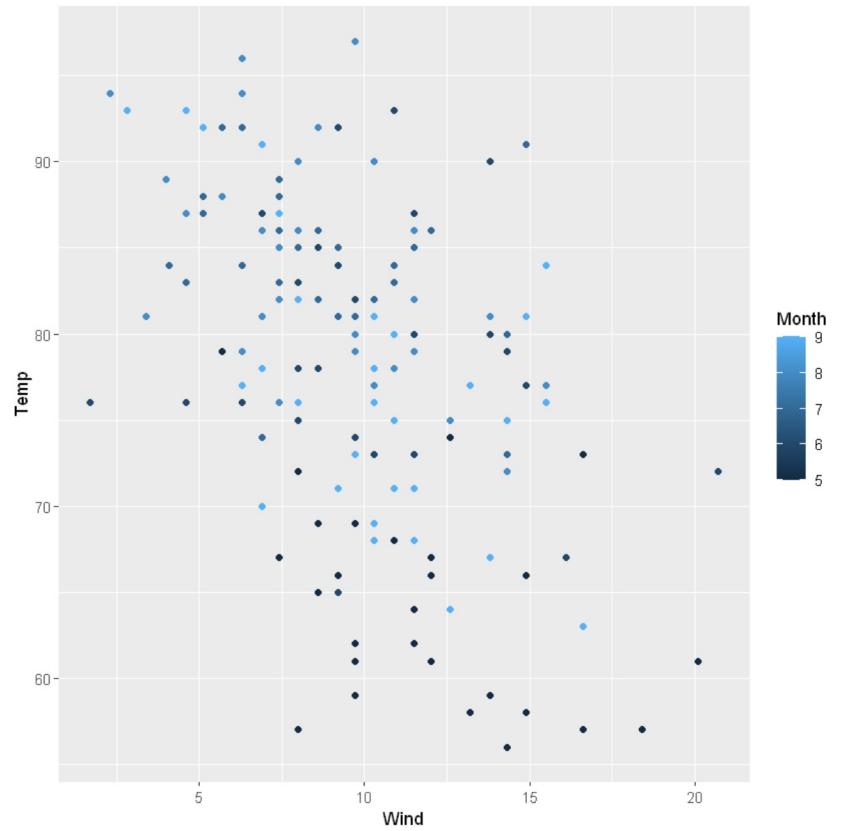
Circles



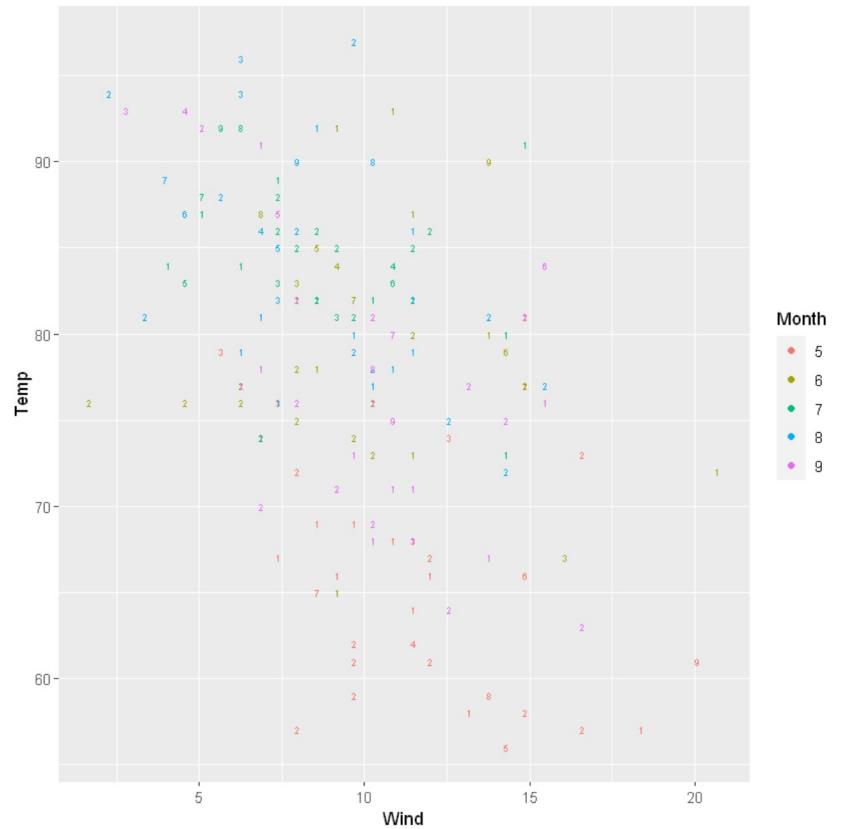




```
In [73]: ggplot(airquality, aes(x=Wind, y=Temp, col=Month))+geom_point()
```

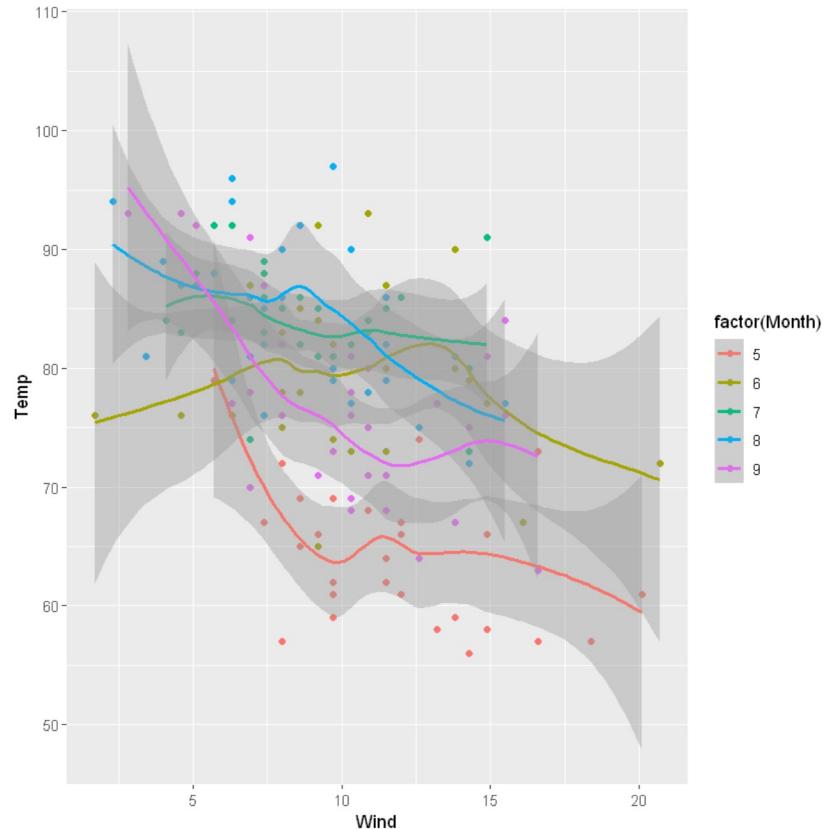


```
In [74]: ggplot(airquality, aes(x = Wind, y = Temp, col = factor(Month), shape = factor(Day))) +  
  geom_point() + scale_color_discrete(name = "Month") + scale_shape_identity()
```



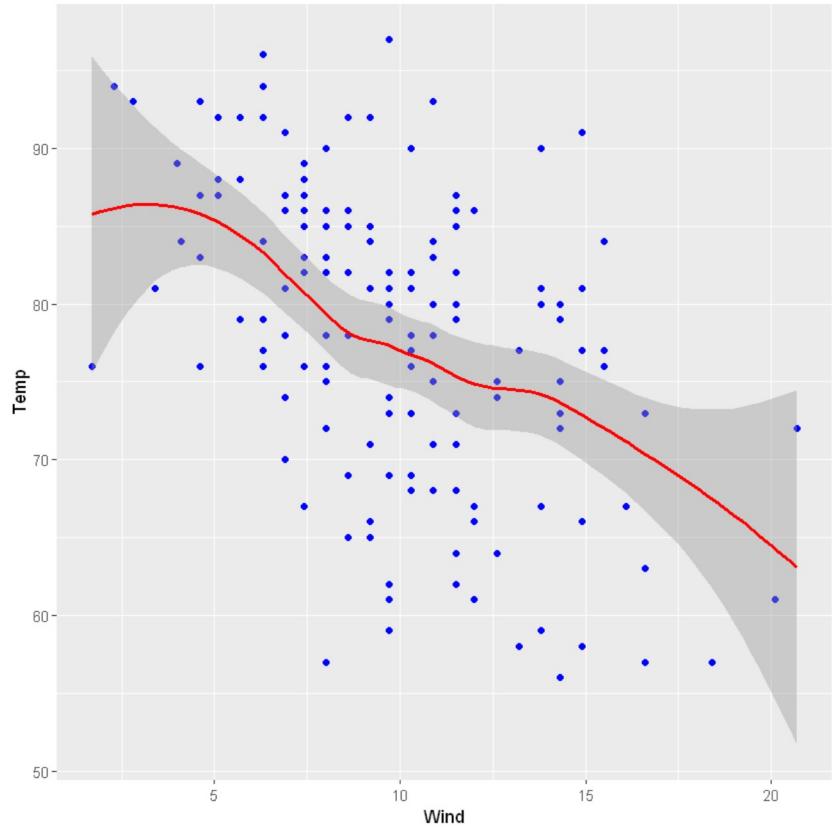
```
In [75]: ggplot(airquality, aes(x = Wind, y = Temp, col = factor(Month))) +  
geom_point() + geom_smooth()
```

```
`geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```



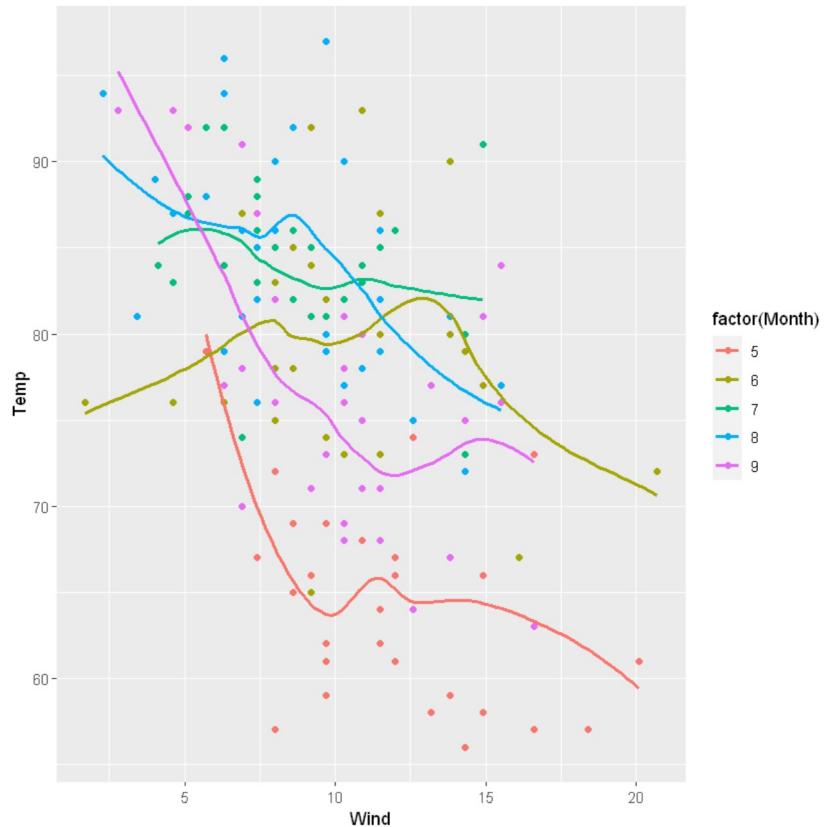
```
In [76]: ggplot(airquality, aes(x = Wind, y = Temp, col = factor(Month))) +  
geom_point(color = "blue") +geom_smooth(color = "red")
```

```
`geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```

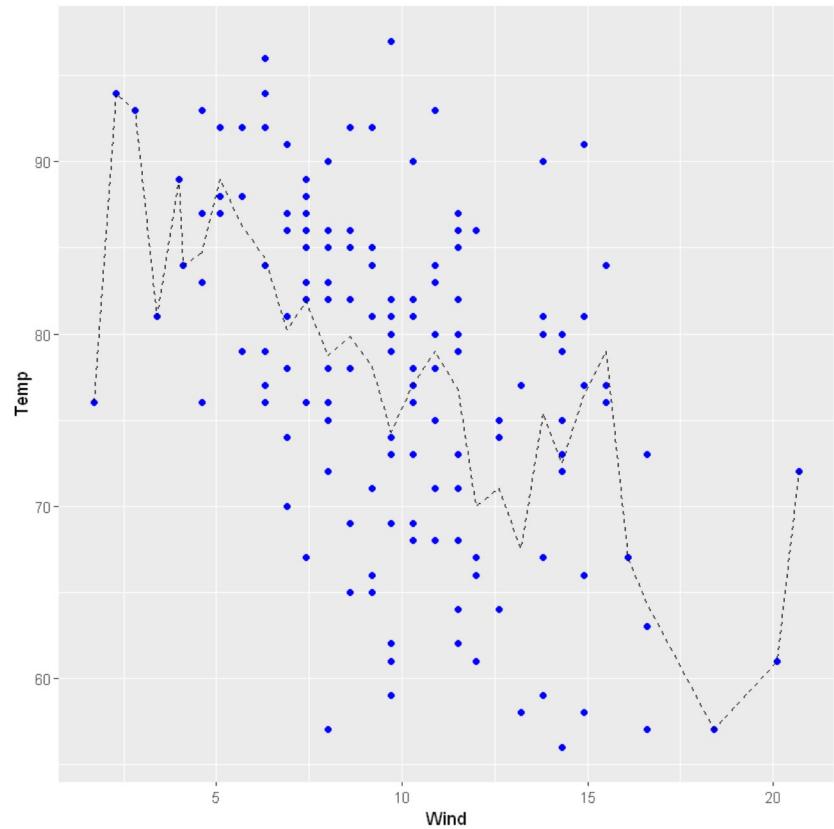


```
In [77]: ggplot(airquality, aes(x = Wind, y = Temp, col = factor(Month))) +  
geom_point() +geom_smooth(se = FALSE)
```

```
`geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```

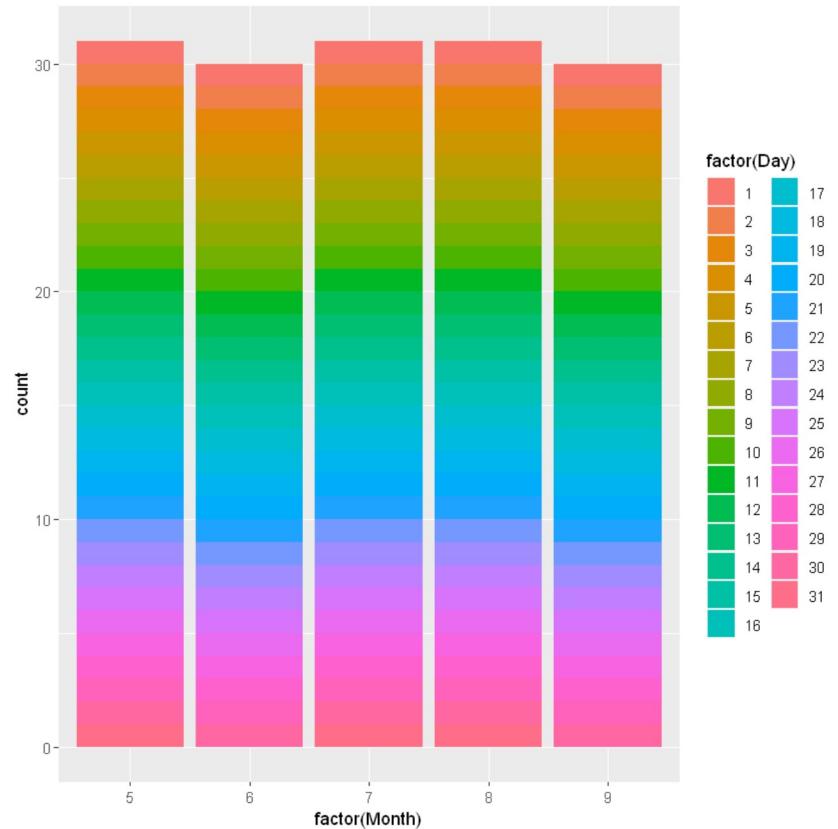


```
In [78]: ggplot(airquality, aes(Wind, Temp)) +geom_point(color = "blue") +  
stat_summary(fun.y = "mean", geom = "line", linetype = "dashed")
```

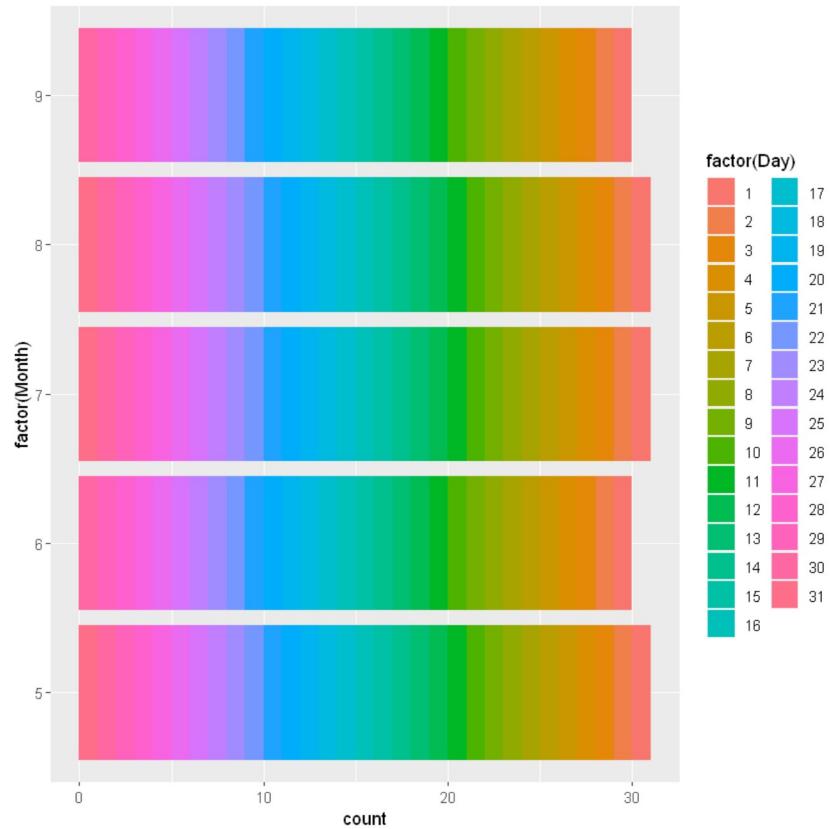


## 2)BAR PLOT

```
In [79]: ggplot(airquality, aes(x = factor(Month), fill = factor(Day))) +geom_bar()
```

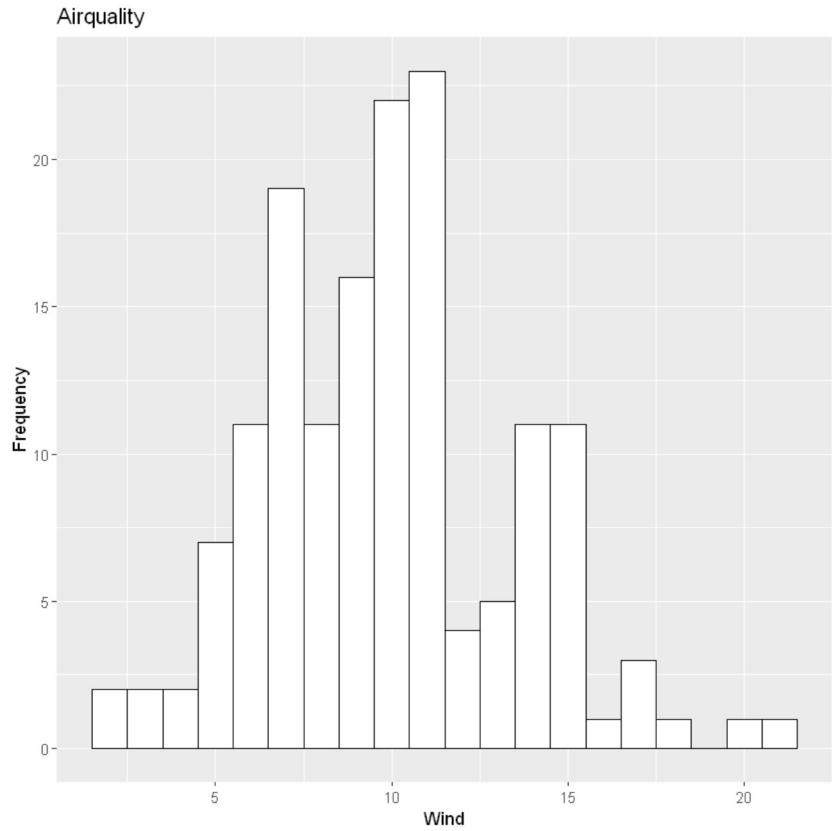


```
In [80]: ggplot(airquality, aes(x = factor(Month), fill = factor(Day))) + geom_bar() +coord_flip()
```

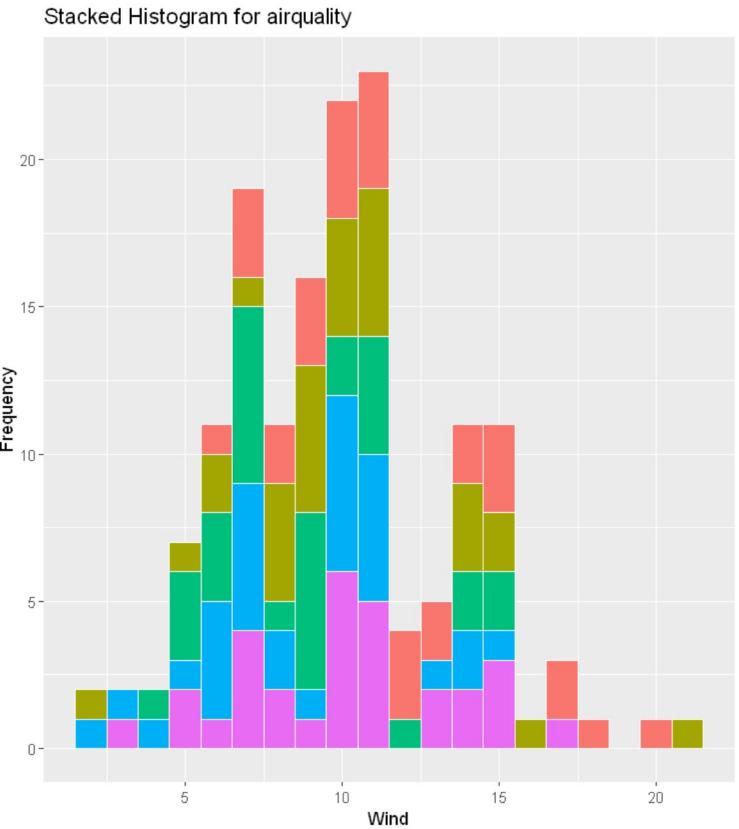


### 3)Histogram

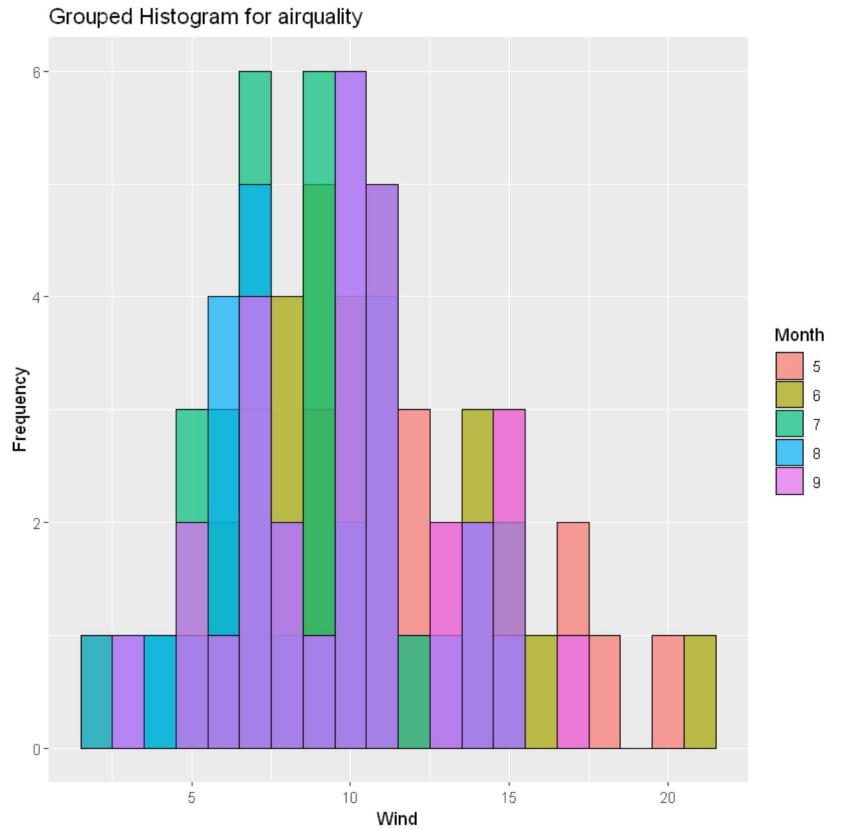
```
In [81]: ggplot(airquality, aes(x = Wind)) +  
  geom_histogram(color = "black", fill = "white", bins = 20) +  
  labs(title = "Airquality", x = "Wind", y = "Frequency")
```



```
In [82]: ggplot(airquality, aes(x = Wind, fill = factor(Month))) +  
  geom_histogram(color = "white", position = "stack", bins = 20) +  
  labs(title = "Stacked Histogram for airquality", x = "Wind", y = "Frequency", fill = "Month")
```

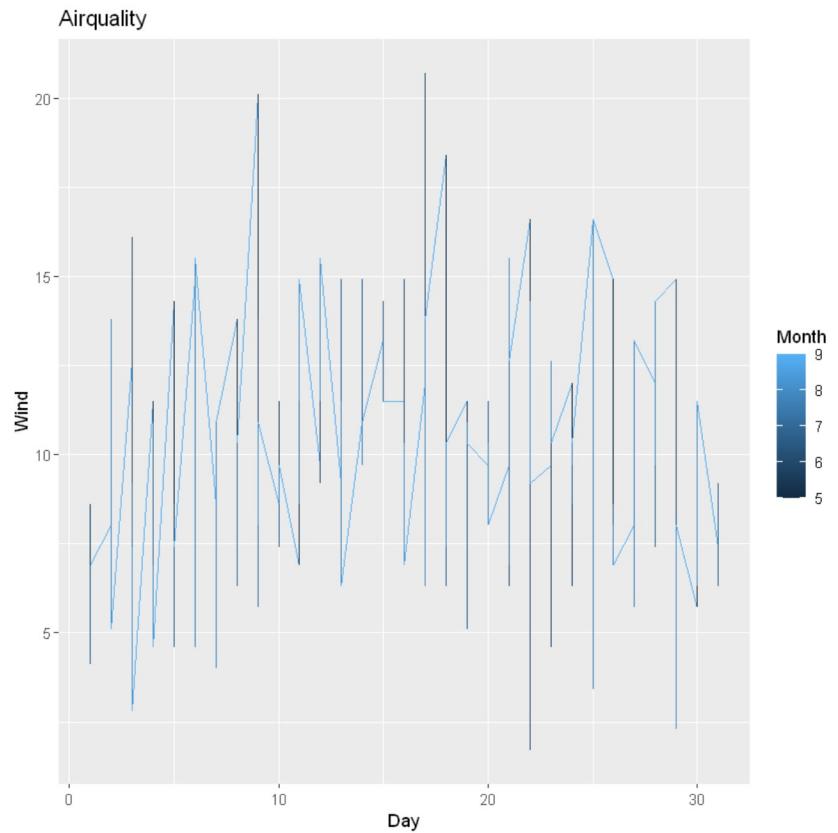


```
In [83]: ggplot(airquality, aes(x = Wind, fill = factor(Month))) +  
  geom_histogram(color = "black", alpha = 0.7, position = "identity", bins = 20) +  
  labs(title = "Grouped Histogram for airquality", x = "Wind", y = "Frequency", fill = "Month")
```



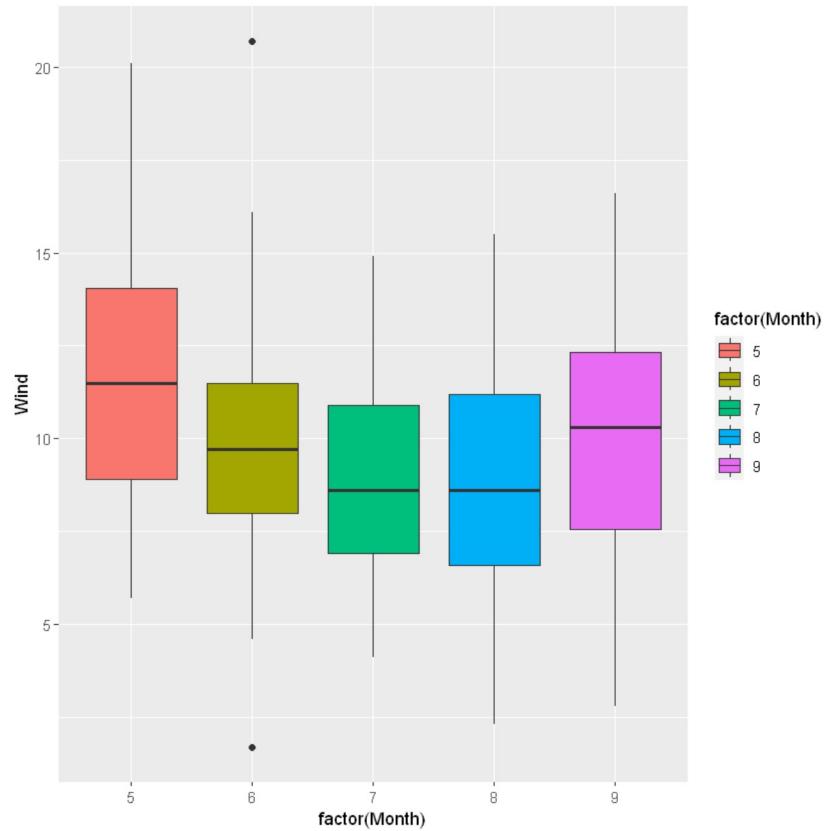
## 4) LINE PLOT

```
In [84]: ggplot(airquality, aes(x = Day, y = Wind, color = Month)) +geom_line() +
  labs(title = "Airquality", x = "Day", y = "Wind", color = "Month")
```



## 5)BOX PLOT

```
In [85]: ggplot(airquality, aes(x = factor(Month), y = Wind, fill = factor(Month))) +  
geom_boxplot()
```



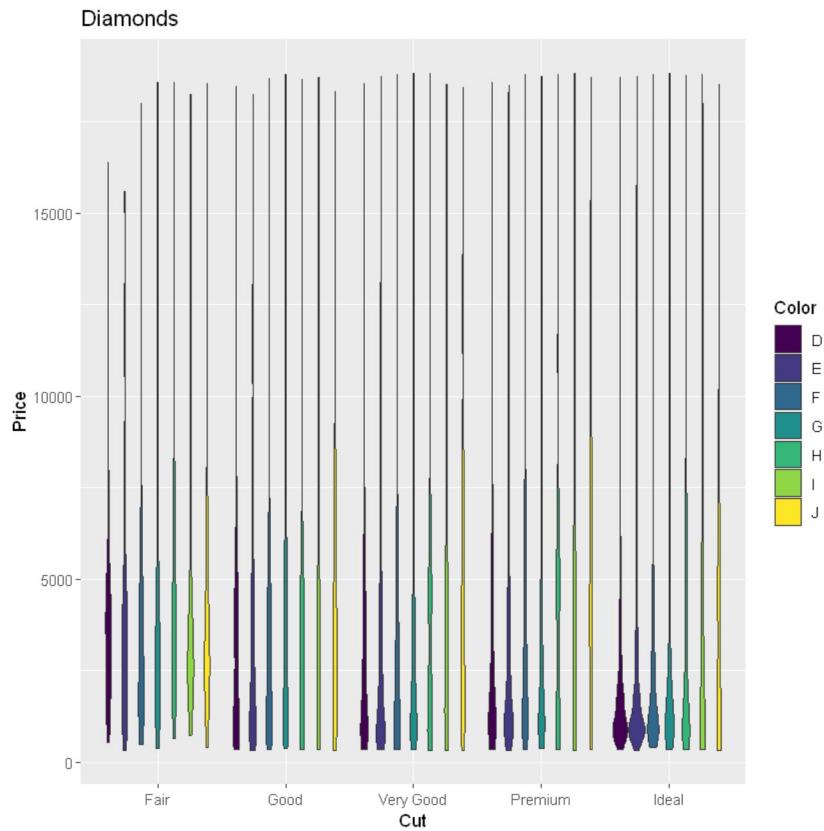
## 6)VIOLIN PLOT

```
In [86]: #Taking diamonds dataset for Violin plot
data(diamonds)
dim(diamonds)
head(diamonds)
summary(diamonds)
ggplot(diamonds, aes(cut, price)) +
  geom_violin(aes(fill = color)) +
  labs(title = "Diamonds", x = "Cut", y = "Price", fill = "Color")
```

53940 · 10

A tibble: 6 × 10

carat	cut	color	clarity	depth	table	price	x	y	z		
<dbl>	<ord>	<ord>	<ord>	<dbl>	<dbl>	<int>	<dbl>	<dbl>	<dbl>		
0.23	Ideal	E	SI2	61.5	55	326	3.95	3.98	2.43		
0.21	Premium	E	SI1	59.8	61	326	3.89	3.84	2.31		
0.23	Good	E	VS1	56.9	65	327	4.05	4.07	2.31		
0.29	Premium	I	VS2	62.4	58	334	4.20	4.23	2.63		
0.31	Good	J	SI2	63.3	58	335	4.34	4.35	2.75		
0.24	Very Good	J	VVS2	62.8	57	336	3.94	3.96	2.48		
<hr/>											
carat		cut		color		clarity		depth			
Min. :0.2000	Fair	: 1610		D: 6775	SI1	:13065		Min. :43.00			
1st Qu.:0.4000	Good	: 4906		E: 9797	VS2	:12258		1st Qu.:61.00			
Median :0.7000	Very Good	:12082		F: 9542	SI2	: 9194		Median :61.80			
Mean :0.7979	Premium	:13791		G:11292	VS1	: 8171		Mean :61.75			
3rd Qu.:1.0400	Ideal	:21551		H: 8304	VVS2	: 5066		3rd Qu.:62.50			
Max. :5.0100				I: 5422	VVS1	: 3655		Max. :79.00			
				J: 2808	(Other)	: 2531					
<hr/>											
table		price		x		y					
Min. :43.00		Min. : 326		Min. : 0.000		Min. : 0.000					
1st Qu.:56.00		1st Qu.: 950		1st Qu.: 4.710		1st Qu.: 4.720					
Median :57.00		Median : 2401		Median : 5.700		Median : 5.710					
Mean :57.46		Mean : 3933		Mean : 5.731		Mean : 5.735					
3rd Qu.:59.00		3rd Qu.: 5324		3rd Qu.: 6.540		3rd Qu.: 6.540					
Max. :95.00		Max. :18823		Max. :10.740		Max. :58.900					
<hr/>											
z											
Min. : 0.000											
1st Qu.: 2.910											
Median : 3.530											
Mean : 3.539											
3rd Qu.: 4.040											
Max. :31.800											



## 7)WATERFALLS CHART

```
In [87]: install.packages("ggalluvial")
library(ggalluvial)
```

Warning message:  
"package 'ggalluvial' is in use and will not be installed"

```
In [88]: group1 <- LETTERS[1:30]
value1 <- c(6000, 1000, 2000, 3000, 4000, 2000, -1500, 500, -1000, -2500,
        1200, -800, 1800, -2200, 900, 3000, -1200, 700, -2500, 1500,
        -800, 1200, 1800, -2200, 900, -1200, 700, -2500, 1500, -800)
df <- data.frame(x = group1, y = value1)
df
```

A data.frame: 30

  × 2

**x**      **y**

**<chr>**  **<dbl>**

  A    6000

  B    1000

  C    2000

  D    3000

  E    4000

  F    2000

  G   -1500

  H    500

  I   -1000

  J   -2500

  K   1200

  L   -800

  M   1800

  N   -2200

  O    900

  P   3000

  Q   -1200

  R    700

  S   -2500

  T   1500

  U   -800

  V   1200

x	y
<chr>	<dbl>
W	1800
X	-2200
Y	900
Z	-1200
NA	700
NA	-2500
NA	1500
NA	-800

```
In [89]: install.packages("waterfalls")
library(waterfalls)
waterfall(values = value1, labels = group1)
```

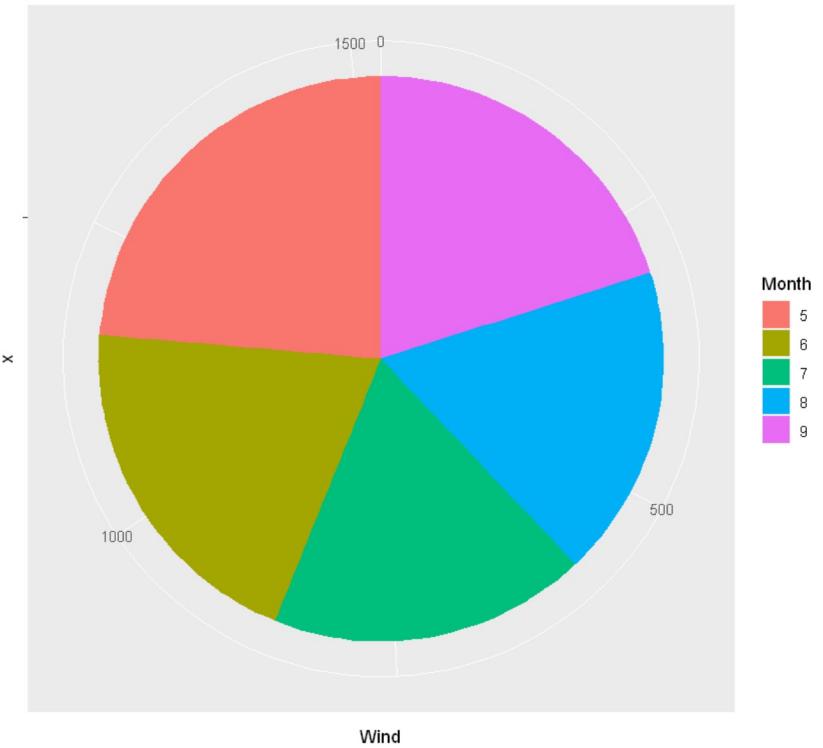
Warning message:  
"package 'waterfalls' is in use and will not be installed"



## 8)PIE CHART

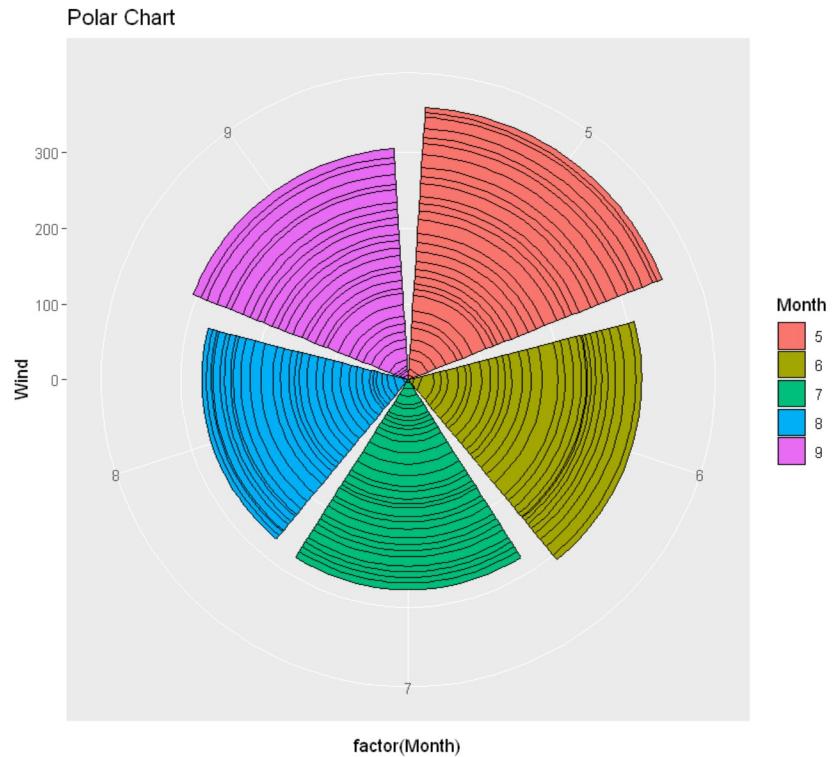
```
In [90]: ggplot(airquality, aes(x = "", y = Wind, fill = factor(Month))) +  
  geom_bar(stat = "identity", width = 1) +  
  coord_polar("y", start = 0) +  
  labs(title = "Pie Chart for Airquality", fill = "Month")
```

Pie Chart for Airquality



## 9)POLAR CHART

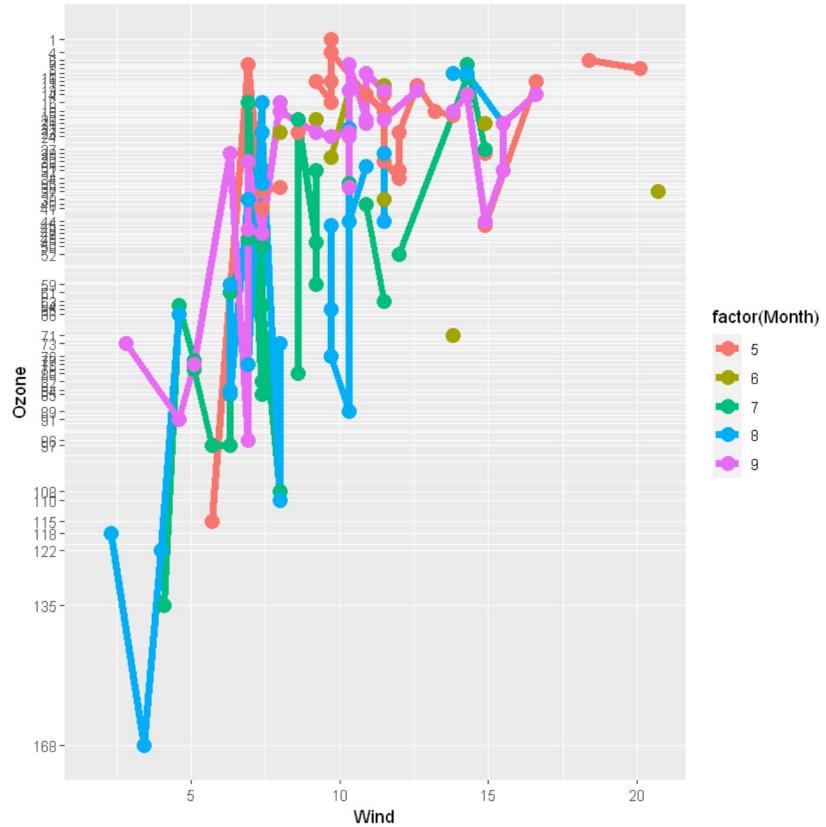
```
In [91]: ggplot(airquality, aes(x = factor(Month), y = Wind, fill = factor(Month))) +  
  geom_bar(stat = "identity", color = "black", linetype = 1, size = 0.5) +  
  labs(title = "Polar Chart", fill = "Month") +  
  coord_polar(theta = "x")
```



## 10)BUMP CHART

```
In [92]: ggplot(airquality, aes(x = Wind, y = Ozone, group = Month)) +  
  geom_line(aes(color = factor(Month)), size = 2) +  
  geom_point(aes(color = factor(Month)), size = 4) +  
  scale_y_reverse(breaks = unique(airquality$Ozone))
```

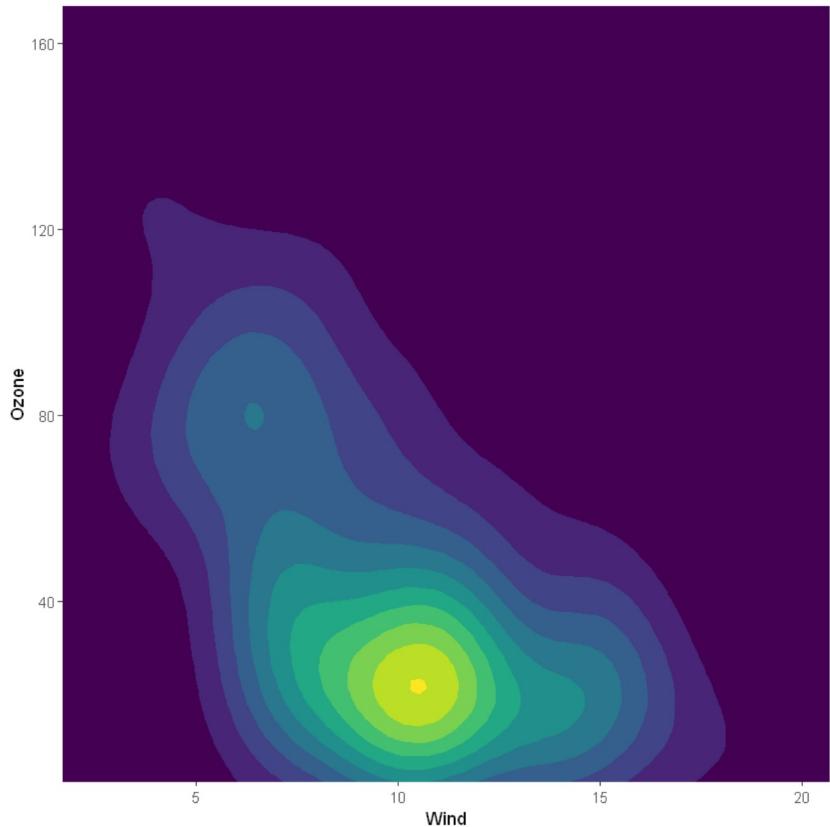
Warning message:  
"Removed 6 rows containing missing values (`geom\_line()`)."  
Warning message:  
"Removed 37 rows containing missing values (`geom\_point()`)."



## 11) CONTOUR PLOT

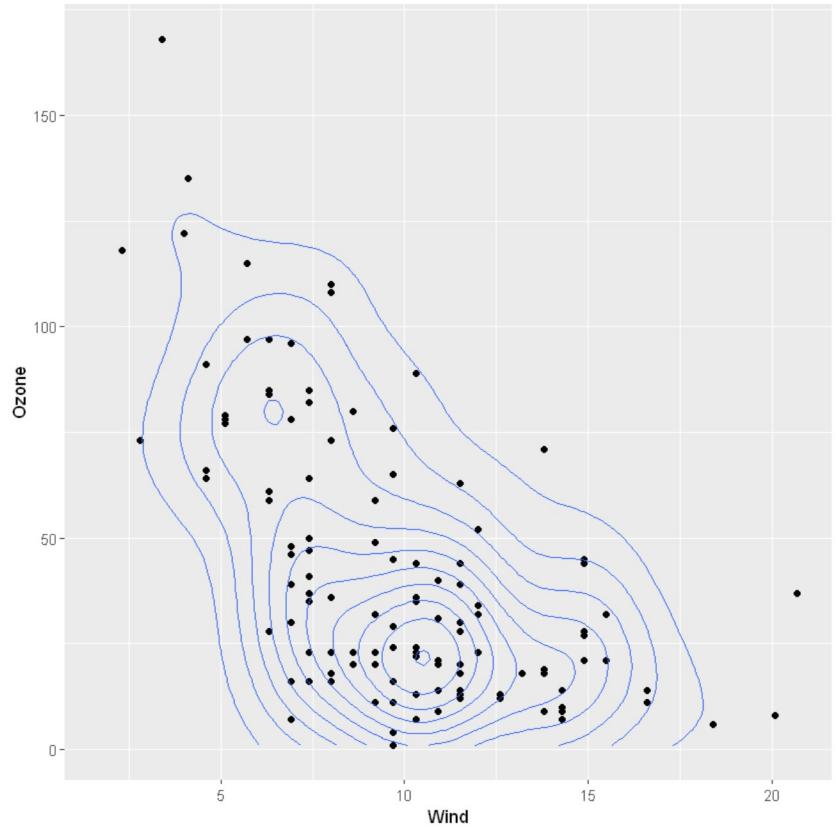
```
In [93]: ggplot(airquality, aes(Wind, Ozone)) +  
  geom_density_2d_filled(show.legend = FALSE) +  
  coord_cartesian(expand = FALSE) +  
  labs(x = "Wind", y = "Ozone")
```

Warning message:  
"Removed 37 rows containing non-finite values (`stat\_density2d\_filled()`)."



```
In [94]: ggplot(airquality, aes(x = Wind, y = Ozone)) +geom_point() +geom_density_2d()
```

Warning message:  
"Removed 37 rows containing non-finite values (`stat\_density2d()`)."  
Warning message:  
"Removed 37 rows containing missing values (`geom\_point()`)."

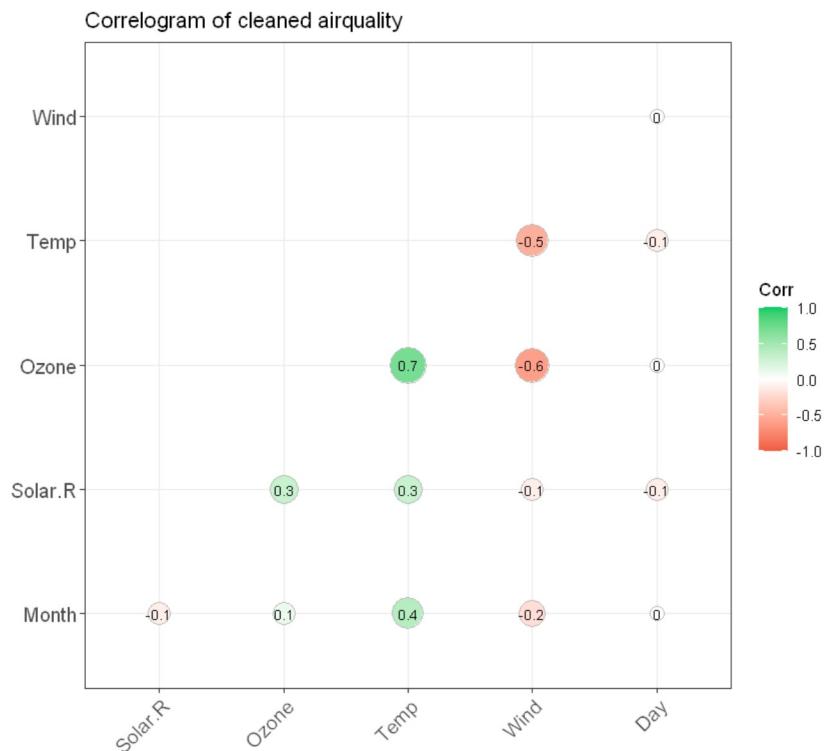


## 12)CORRELOGRAM

```
In [95]: install.packages("ggcorrplot")
library(ggcorrplot)
```

Warning message:  
"package 'ggcorrplot' is in use and will not be installed"

```
In [96]: cleaned_airquality <- na.omit(airquality)
corr_matrix <- round(cor(cleaned_airquality), 1)
ggcorrplot(corr_matrix, hc.order = TRUE,
            type = "lower",
            lab = TRUE,
            lab_size = 3,
            method = "circle",
            colors = c("tomato2", "white", "springgreen3"),
            title = "Correlogram of cleaned airquality",
            ggtheme = theme_bw,
            hc.method = "average")
```



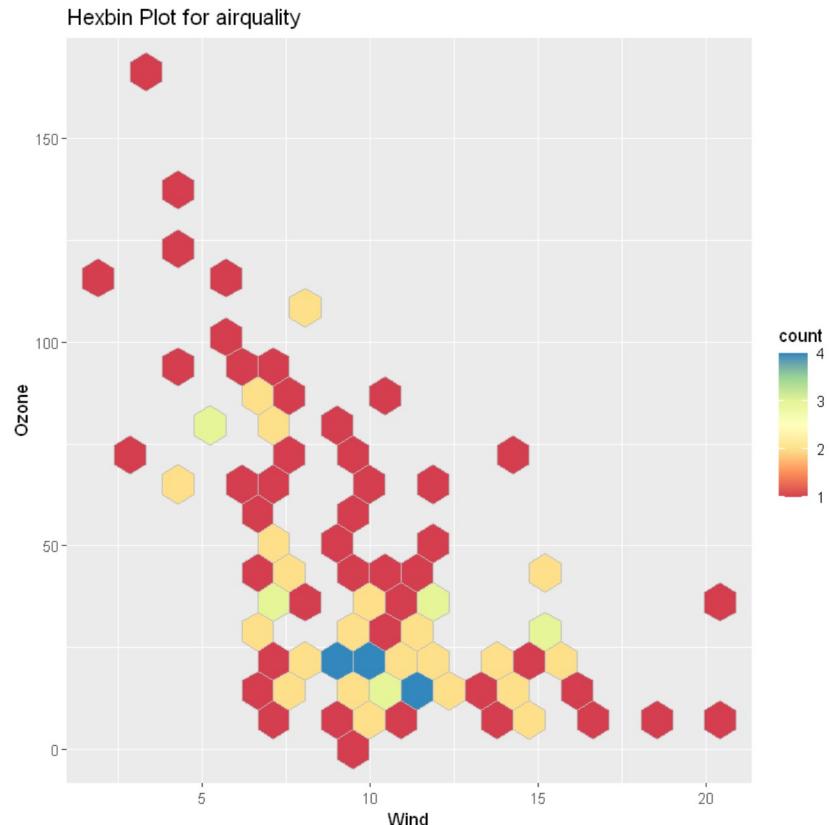
## 13)HEAT MAP

```
In [97]: install.packages("hexbin")
library(hexbin)
```

Warning message:  
"package 'hexbin' is in use and will not be installed"

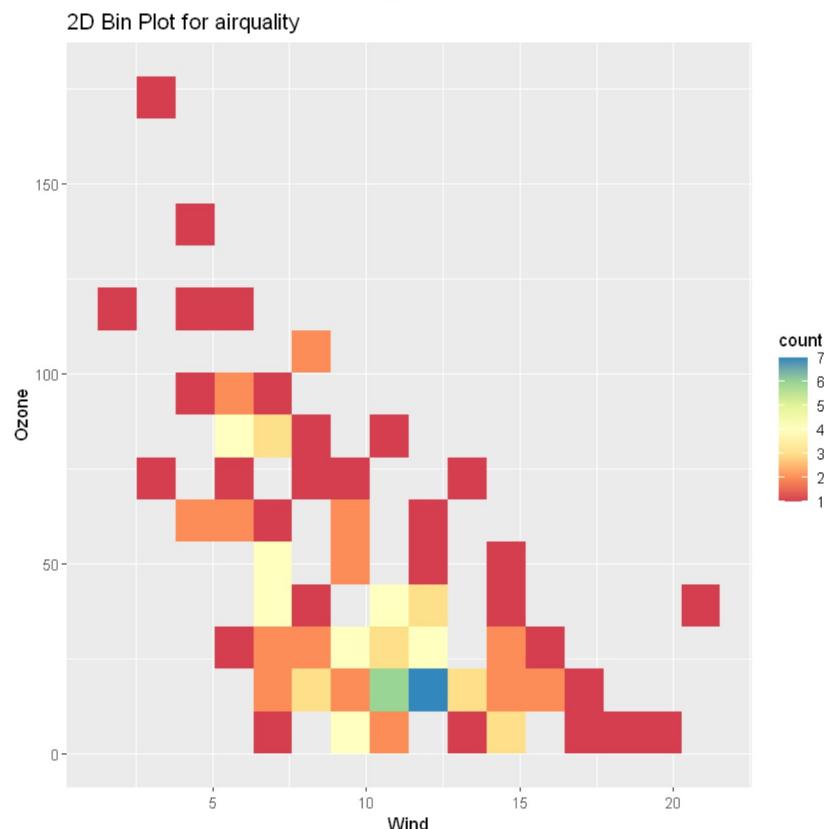
```
In [98]: ggplot(airquality, aes(Wind, Ozone)) +
  geom_hex(bins = 20, color = "grey") +
  scale_fill_distiller(palette = "Spectral", direction = 1) +
  labs(x = "Wind", y = "Ozone", title = "Hexbin Plot for airquality")
```

Warning message:  
"Removed 37 rows containing non-finite values (`stat\_binhex()`)."



```
In [99]: ggplot(airquality, aes(Wind, Ozone)) +  
  geom_bin2d(bins = 15) +  
  scale_fill_distiller(palette = "Spectral", direction = 1) +  
  labs(x = "Wind", y = "Ozone", title = "2D Bin Plot for airquality")
```

Warning message:  
"Removed 37 rows containing non-finite values (`stat\_bin2d()`)."



## 14) RIDGE PLOT

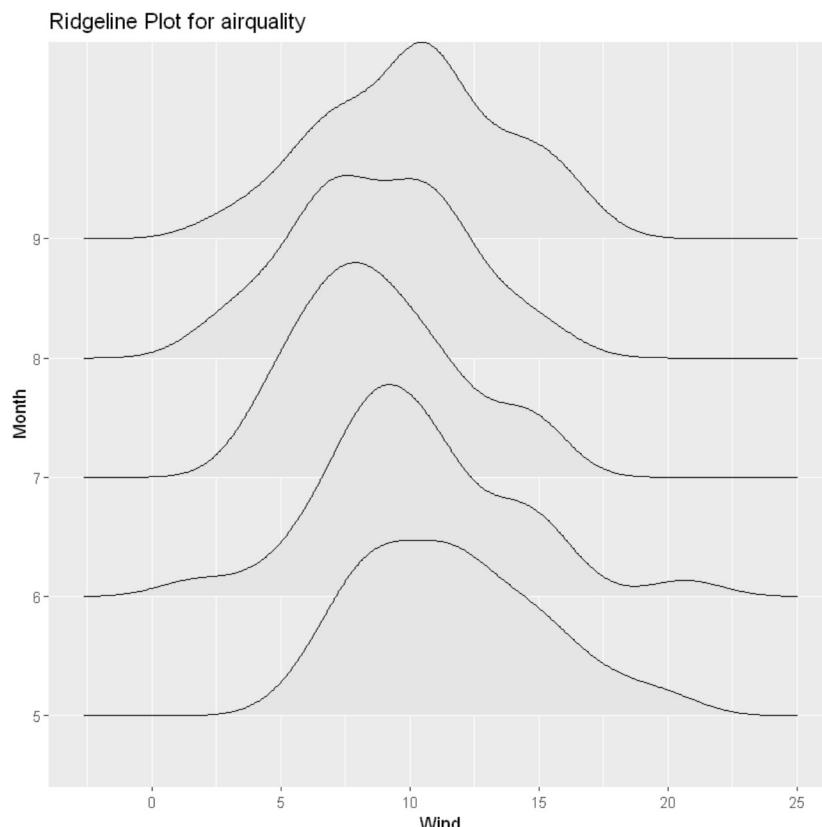
```
In [100...]: install.packages("ggridges")  
library(ggridges)
```

Warning message:  
"package 'ggridges' is in use and will not be installed"

In [101...]

```
ggplot(airquality, aes(x = Wind, y = factor(Month))) +  
  geom_density_ridges(fill = "gray90") +  
  labs(x = "Wind", y = "Month", title = "Ridgeline Plot for airquality")
```

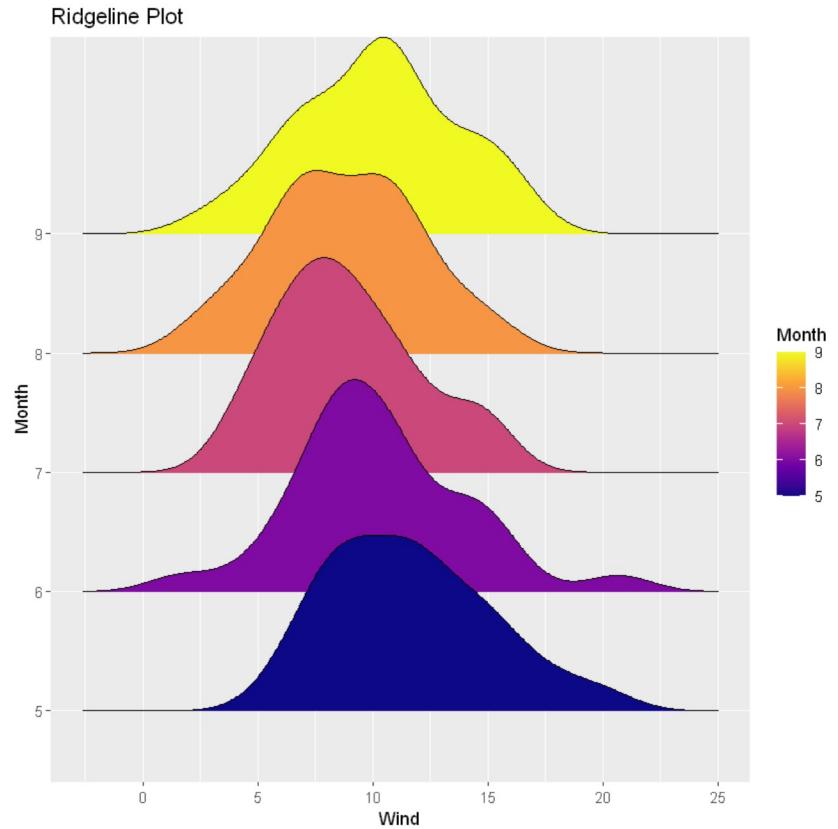
Picking joint bandwidth of 1.44



In [102...]

```
ggplot(airquality, aes(x = Wind, y = factor(Month), fill = Month)) +  
  geom_density_ridges_gradient() +  
  scale_fill_viridis_c(name = "Month", option = "C") +  
  labs(x = "Wind", y = "Month", title = "Ridgeline Plot")
```

Picking joint bandwidth of 1.44



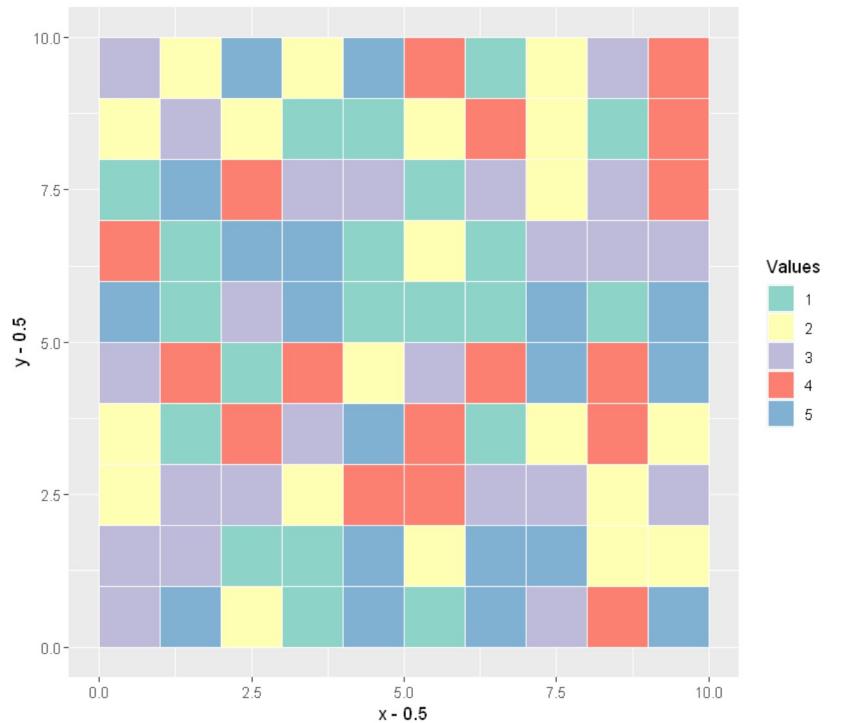
## 15) WAFFLE CHART

In [103...]

```
library(RColorBrewer)
set.seed(123)
random_values <- sample(1:5, 100, replace = TRUE)
```

In [104...]

```
waffle_data <- data.frame(x = rep(1:10, each = 10), y = rep(1:10, times = 10), value = random_values)
ggplot(waffle_data, aes(x = x - 0.5, y = y - 0.5, fill = factor(value))) +
  geom_tile(color = "white", size = 0.5) +
  scale_fill_manual(values = brewer.pal(5, "Set3")) +
  coord_fixed(ratio = 1) +
  guides(fill = guide_legend(title = "Values"))
```



## 16) LIME CHART

In [106...]

```
install.packages('remotes')
```

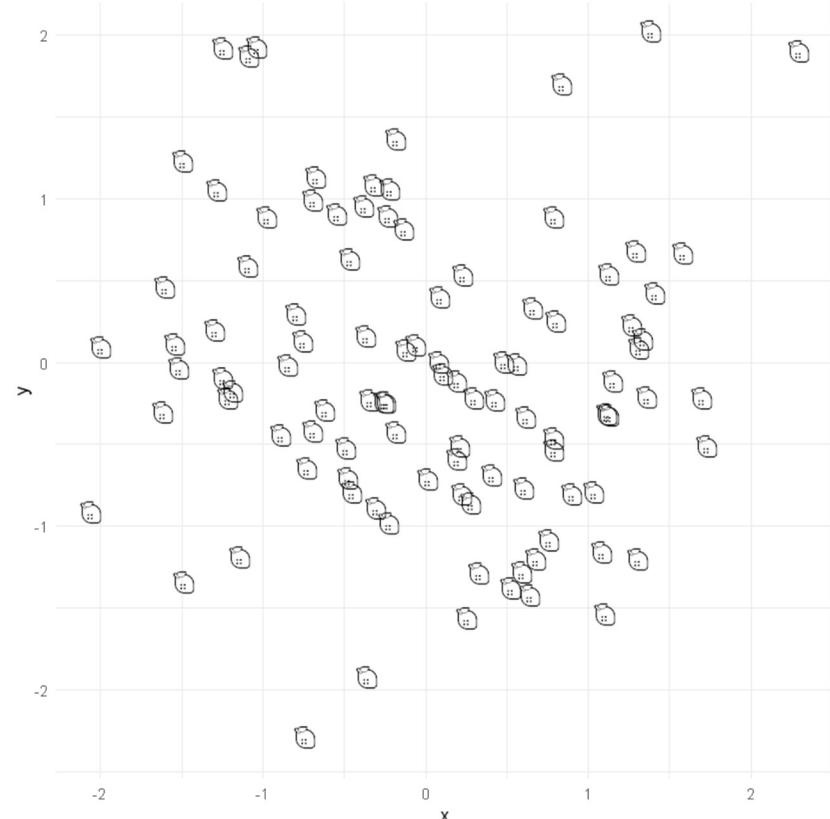
```
Installing package into 'C:/Users/gyanada/AppData/Local/R/win-library/4.2'  
(as 'lib' is unspecified)
```

```
package 'remotes' successfully unpacked and MD5 sums checked
```

```
The downloaded binary packages are in  
C:\Users\gyanada\AppData\Local\Temp\RtmpOSRe8B\downloaded_packages
```

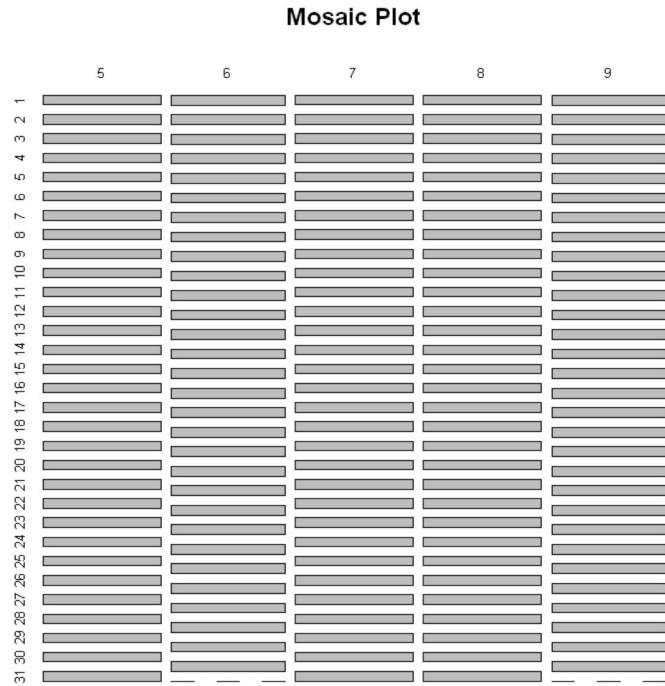
```
In [11]: install.packages("ggplot2")
library(ggplot2)
# Creating a sample dataset
set.seed(190)
data <- data.frame(x = rnorm(100), y = rnorm(100))
ggplot(data, aes(x, y)) +
  geom_point(shape = "\u1f34b", size = 5, fill = "yellow") + # Use the lime emoji and fill it with yellow color
  theme_minimal()
```

Warning message:  
"package 'ggplot2' is in use and will not be installed"



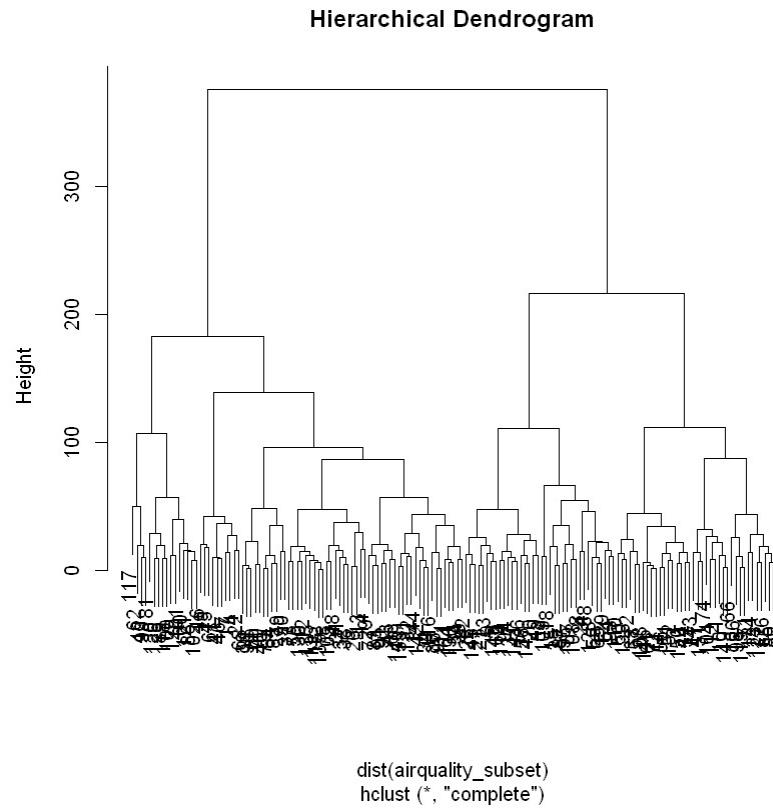
## 17) MOSIAC PLOT

```
In [112... mosaicplot(table(airquality$Month, airquality$Day), main = "Mosaic Plot")
```



## 18) HIERARCHICAL PLOT

```
In [113]:  
airquality_subset <- airquality[c("Ozone", "Solar.R", "Wind", "Temp")]  
hc <- hclust(dist(airquality_subset))  
plot(hc, main = "Hierarchical Dendrogram", labels = rownames(airquality_subset))
```

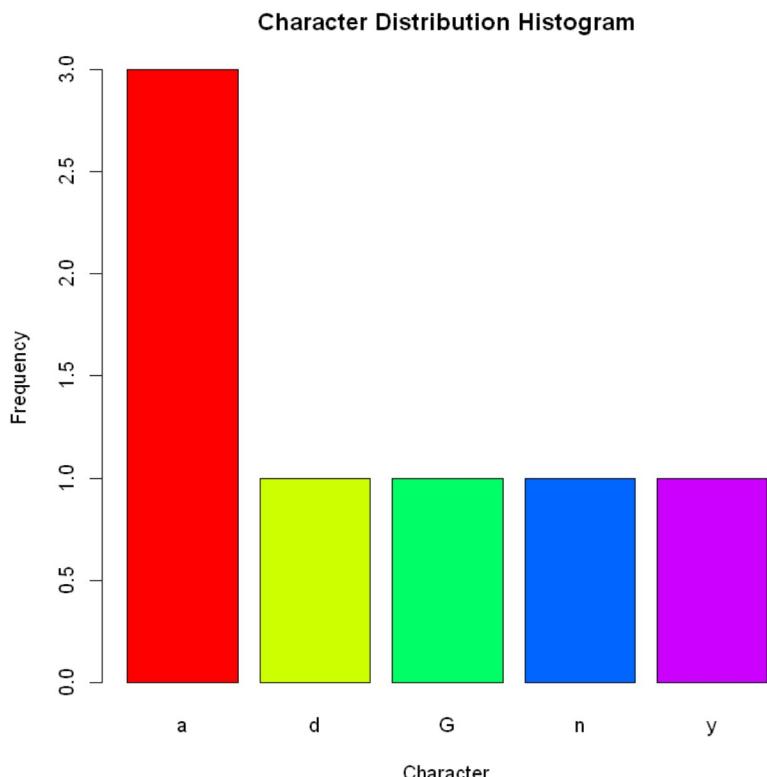


2.Create a histogram that shows the distribution of characters in a string.

Use the function to draw a bar chart for the letters in your first name.

```
In [25]: hist <- function(input_str) {  
  char_counts <- table(strsplit(input_str, NULL)[[1]])  
  
  barplot(char_counts, main = "Character Distribution Histogram",  
          xlab = "Character", ylab = "Frequency", col = rainbow(length(char_counts)), border = "black")  
}
```

```
In [26]: hist("Gyanada")
```



```
In [ ]:
```