ASSIGNMENT-3

NAME: M Gyanada Chowdary

REG.NO: 21BCE7727

1. Construction of Interactive graphics – linked views

Gyanada

21BCE7727

SLOT - L39,L40

In [1]:
```
install.packages("ggplot2")
library(ggplot2)
```

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

package 'ggplot2' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\gyanada\AppData\Local\Temp\RtmpOu5cp1\downloaded_packages

Warning message:
"package 'ggplot2' was built under R version 4.2.3"
```

In [2]:
```
install.packages("ggvis")
library(ggvis)
```

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

package 'ggvis' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\gyanada\AppData\Local\Temp\RtmpOu5cp1\downloaded_packages

Warning message:
"package 'ggvis' was built under R version 4.2.3"

Attaching package: 'ggvis'


The following object is masked from 'package:ggplot2':

    resolution
```

In [3]:
```
data(airquality)
```

In [4]:
```
head(airquality)
```

In [4]:
```
head(airquality)
```

A data.frame: 6 × 6

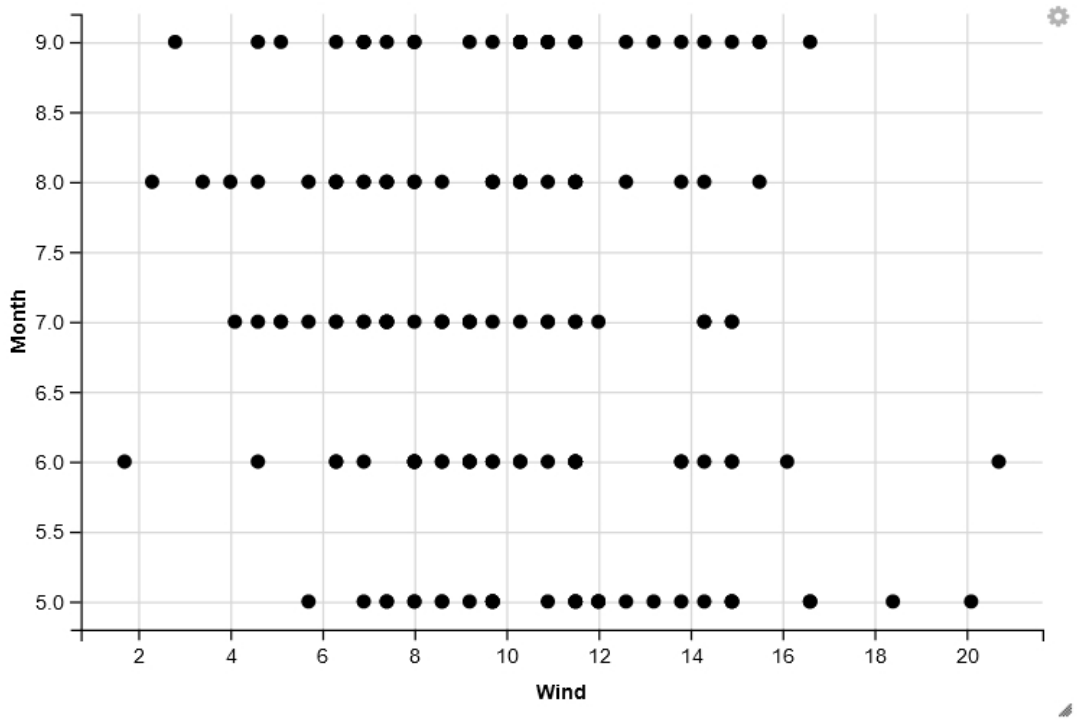
	Ozone <int>	Solar.R <int>	Wind <dbl>	Temp <int>	Month <int>	Day <int>
1	41	190	7.4	67	5	1
2	36	118	8.0	72	5	2
3	12	149	12.6	74	5	3
4	18	313	11.5	62	5	4
5	NA	NA	14.3	56	5	5
6	28	NA	14.9	66	5	6

In [5]:
```
summary(airquality)
```

```
     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  
```

In [6]:
```
v <- ggvis(airquality, x= ~Wind, y= ~Month)
```

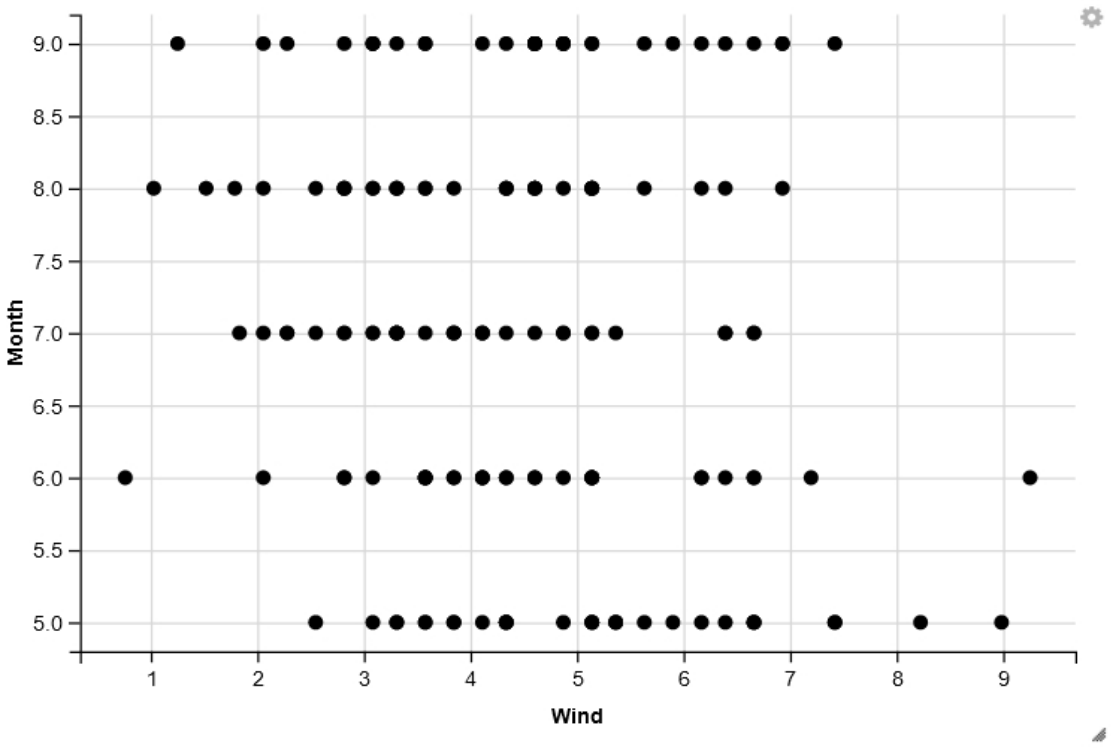
In [7]:
```
layer_points(v)
```



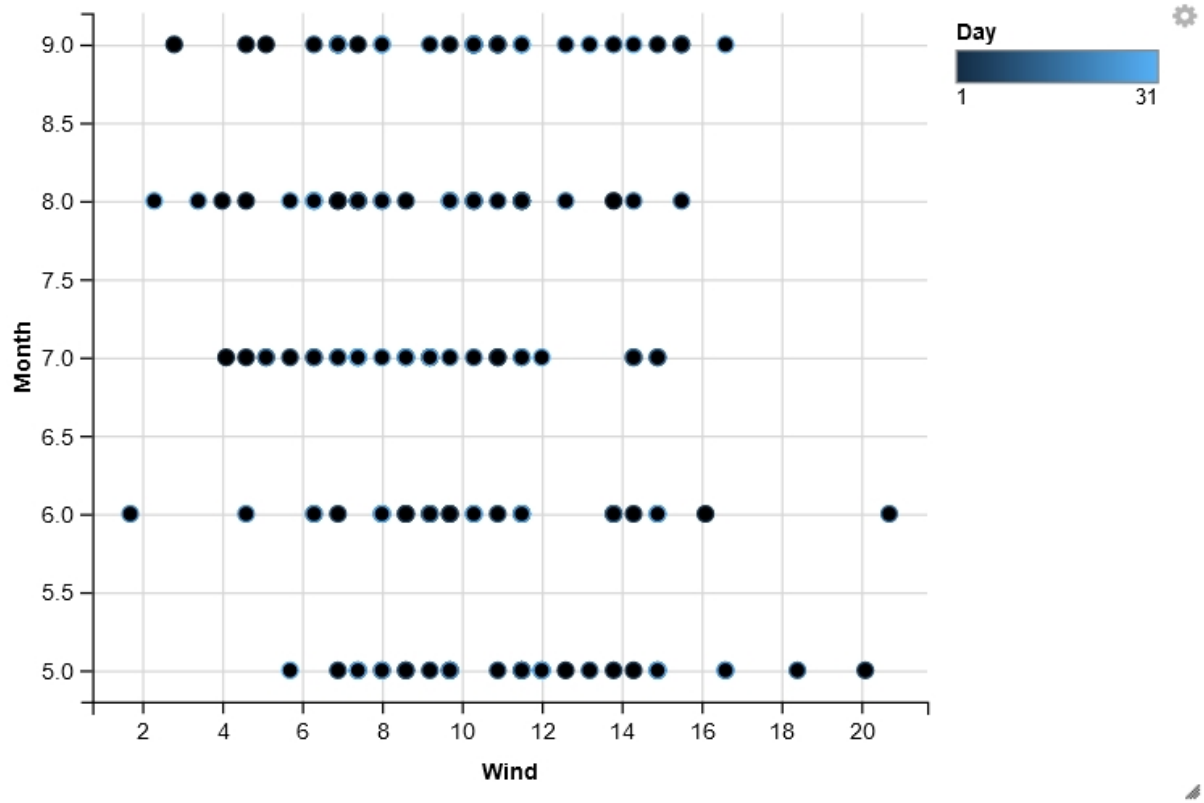
In [10]:
```
library(dplyr)
airquality %>%
   ggvis(x= ~Wind, y= ~Month) %>%
   mutate(Wind = Wind*0.447 )
   layer_points()
```

```
Warning message:
"package 'dplyr' was built under R version 4.2.3"

Attaching package: 'dplyr'


The following objects are masked from 'package:stats':

    filter, lag


The following objects are masked from 'package:base':

    intersect, setdiff, setequal, union


Guessing layer_points()
```

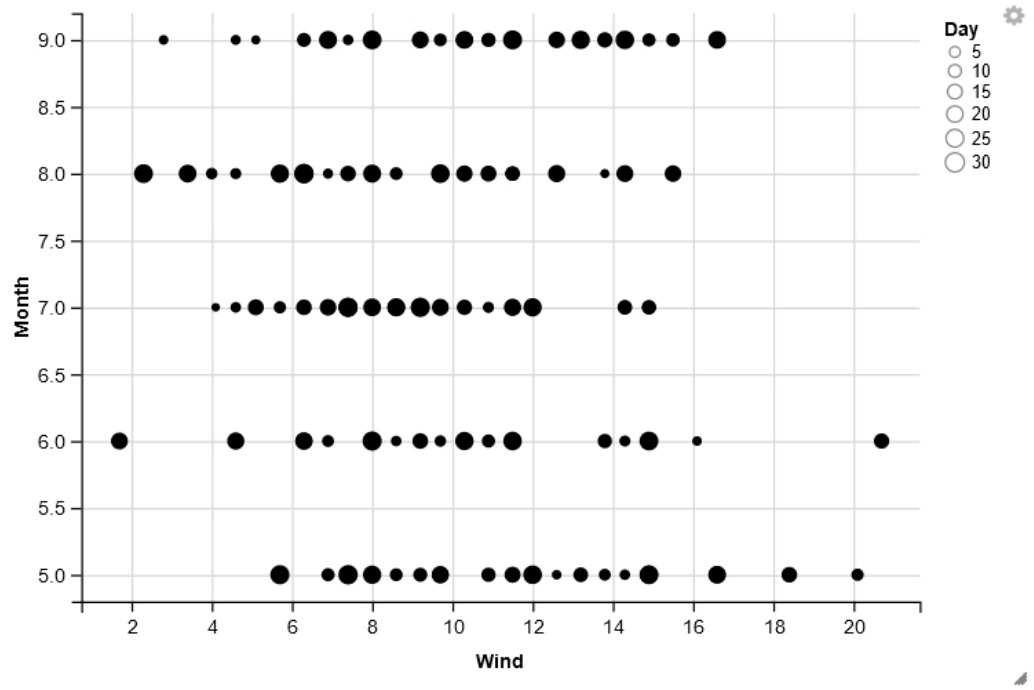
```
Error in add_mark(vis, "symbol", props(..., env = parent.frame()), data, : argu
ment "vis" is missing, with no default
Traceback:

1. layer_points()
2. add_mark(vis, "symbol", props(..., env = parent.frame()), data,
 .     deparse2(substitute(data)))
```



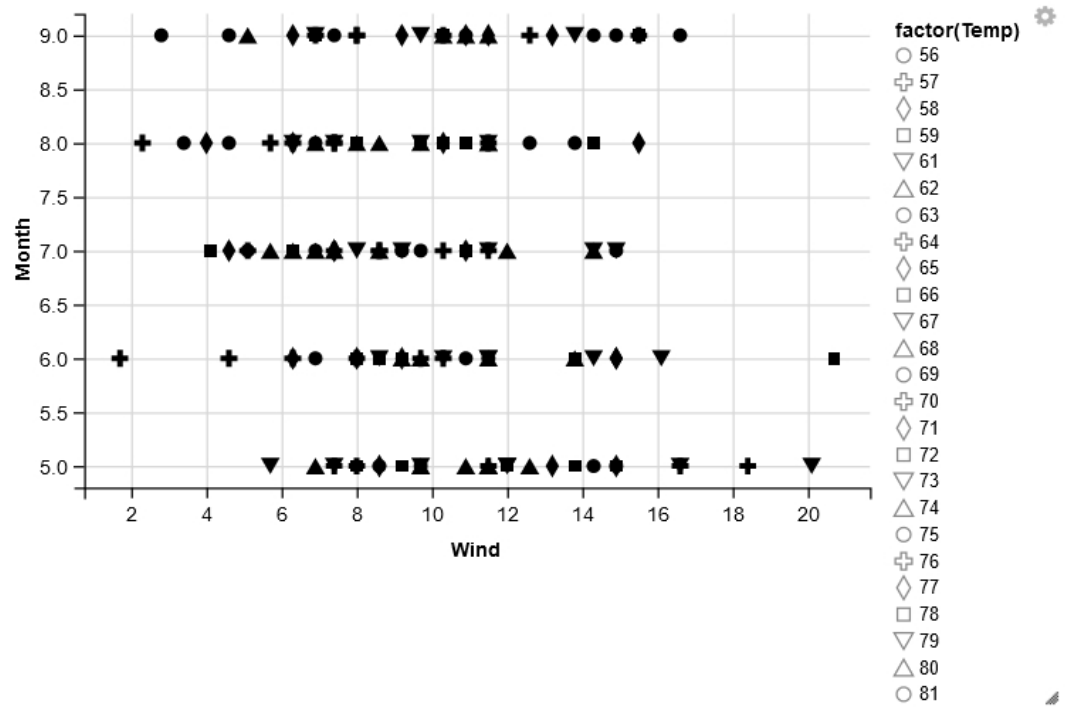
In [10]:
```
airquality %>% ggvis(~Wind, ~Month, stroke= ~Day) %>% layer_points()
```



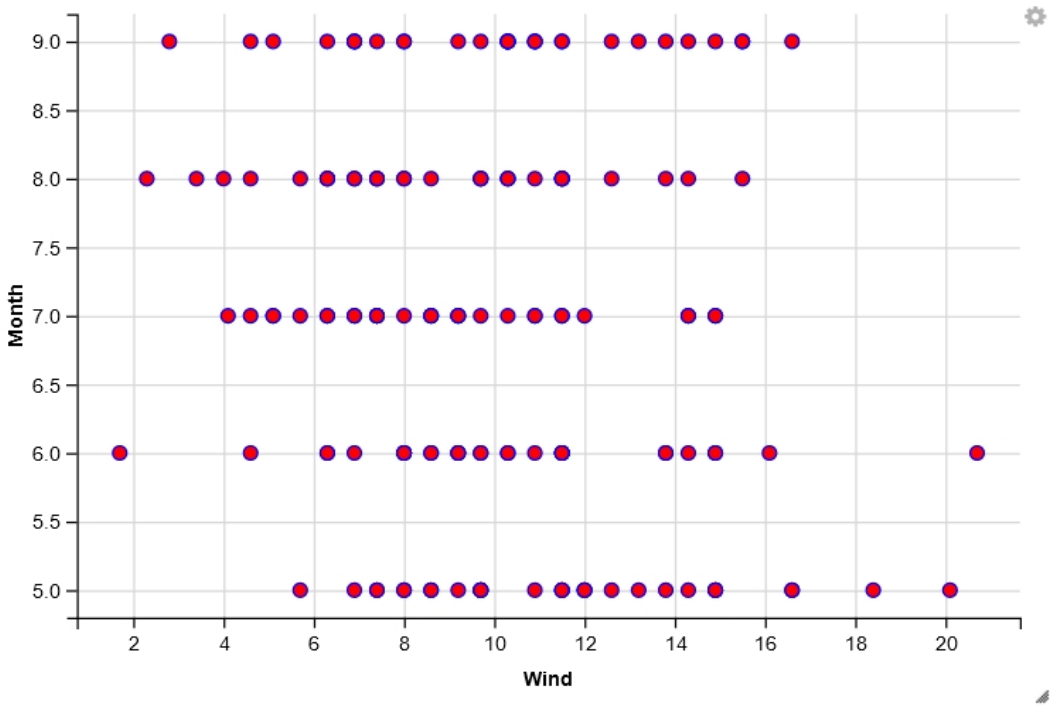
In [12]:
```
airquality %>% ggvis(~Wind, ~Month, size = ~Day) %>% layer_points()
```



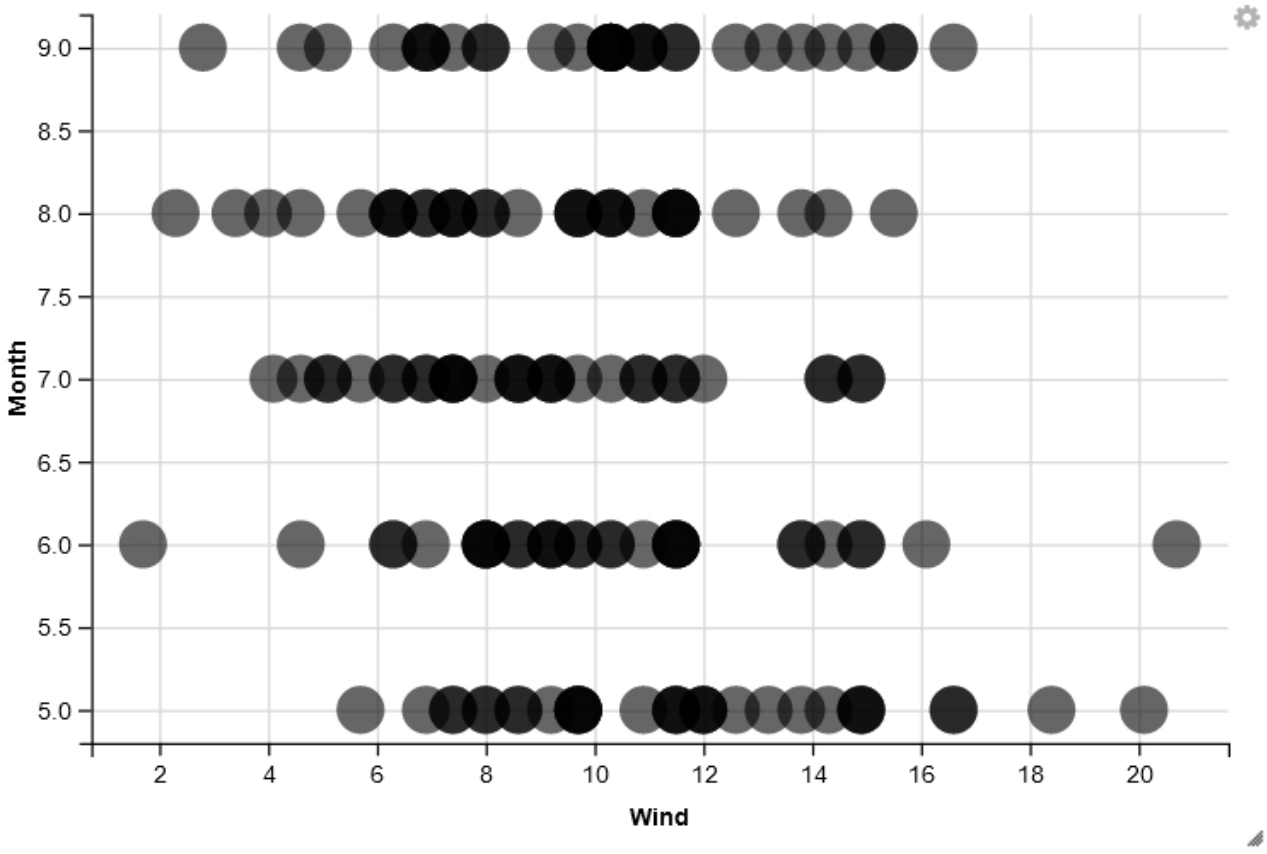
In [13]:
```
airquality %>% ggvis(~Wind, ~Month, shape = ~factor(Temp)) %>% layer_points()
```



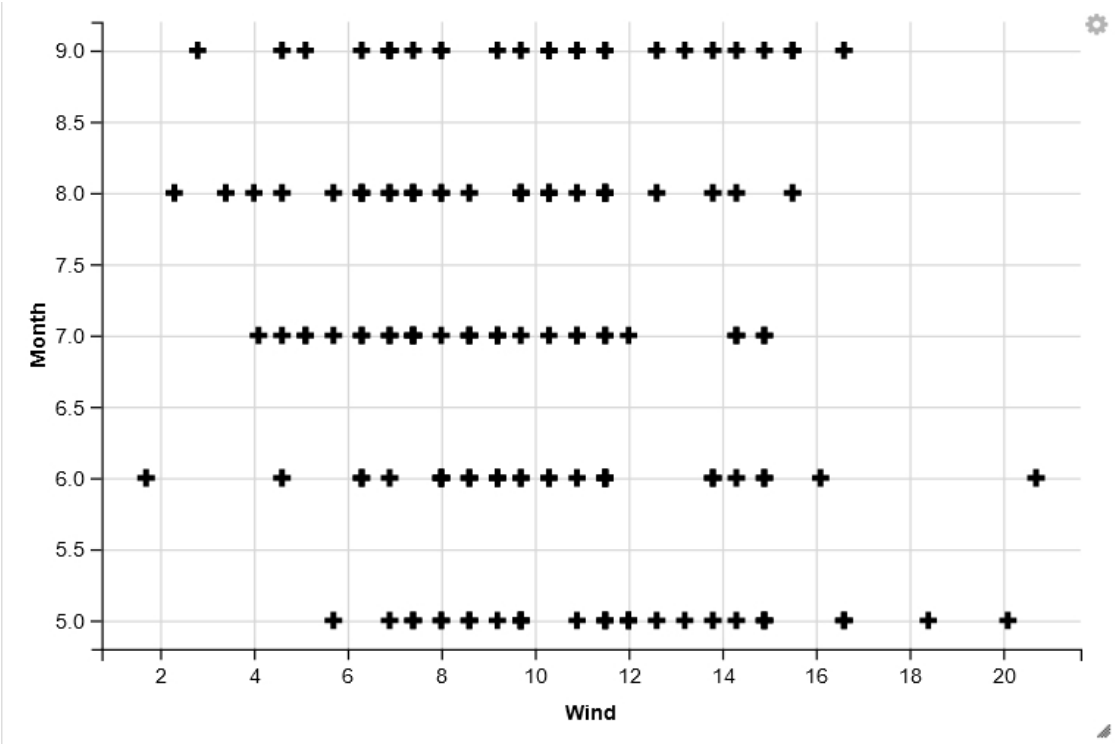
In [14]:
```
airquality %>% ggvis(~Wind, ~Month, fill := "red", stroke := "blue") %>% layer_points()
```



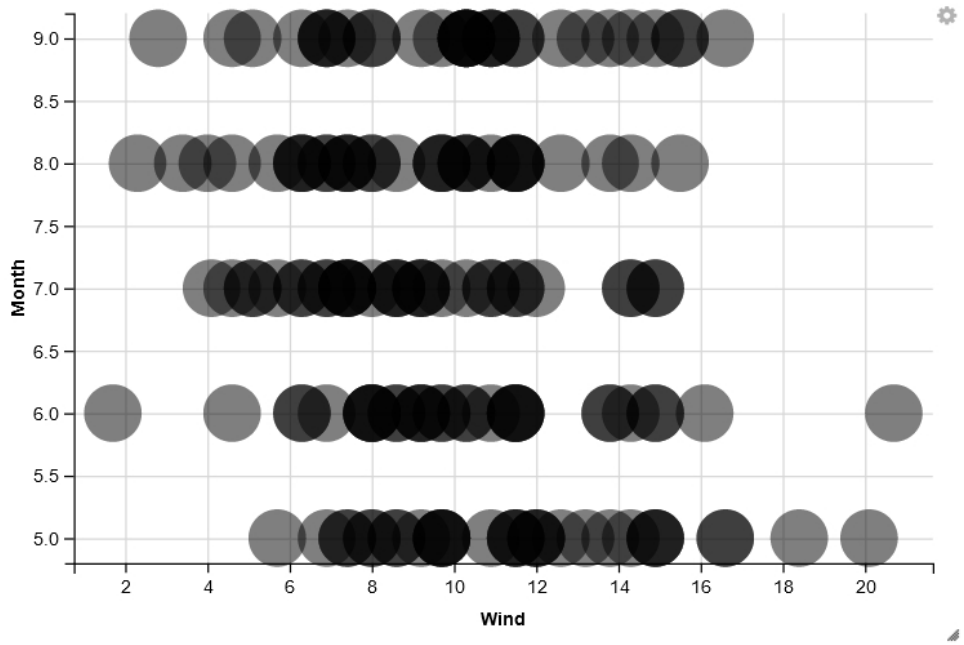
In [15]:
```
airquality %>% ggvis(~Wind, ~Month, size := 400, opacity := 0.6) %>% layer_points()
```



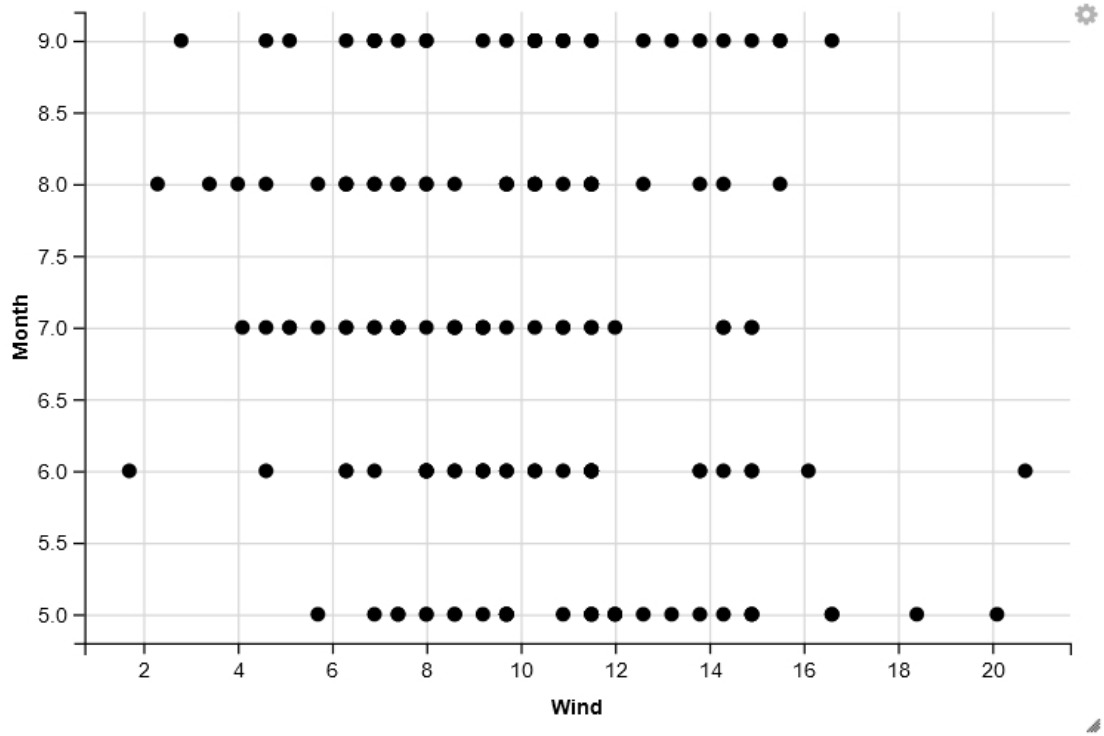
In [16]:
```
airquality %>% ggvis(~Wind, ~Month, shape := "cross") %>% layer_points()
```



In [30]:
```
keys_s <- left_right(10, 2000, step = 50)
airquality %>% ggvis(~Wind, ~Month, size := keys_s, opacity := 0.5) %>% layer_points()
```



In [31]:
```
airquality %>% ggvis(~Wind, ~Month) %>%
  layer_points() %>%
  add_tooltip(function(df) df$Wind)
```



LAYERS ¶

SIMPLE LAYERS

In [32]:
```
airquality %>% ggvis(~Wind, ~Month) %>% layer_points()
```

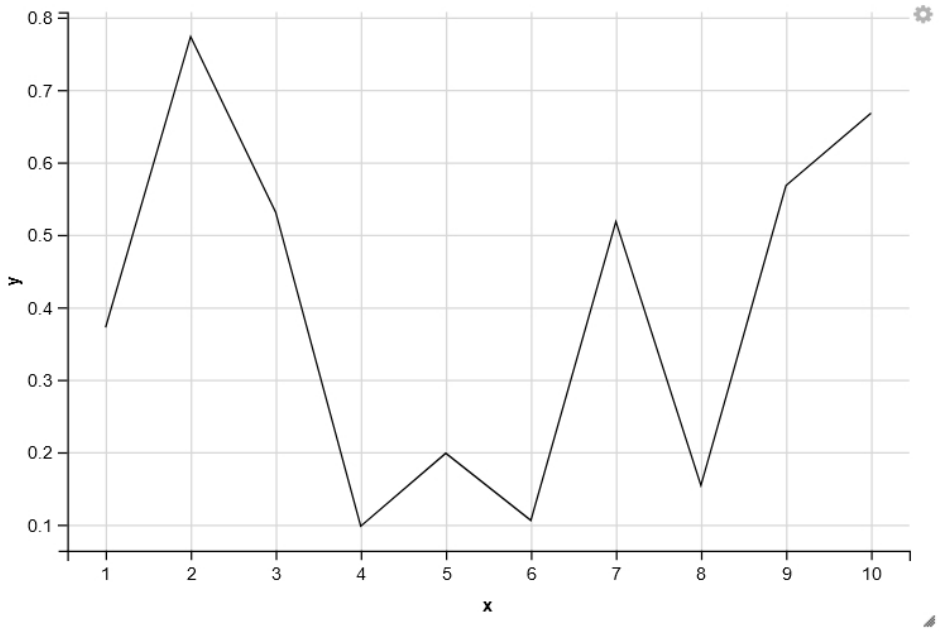
In [33]:
```
df <- data.frame(x = 1:10, y = runif(10))
df %>% ggvis(~x, ~y) %>% layer_paths()
```

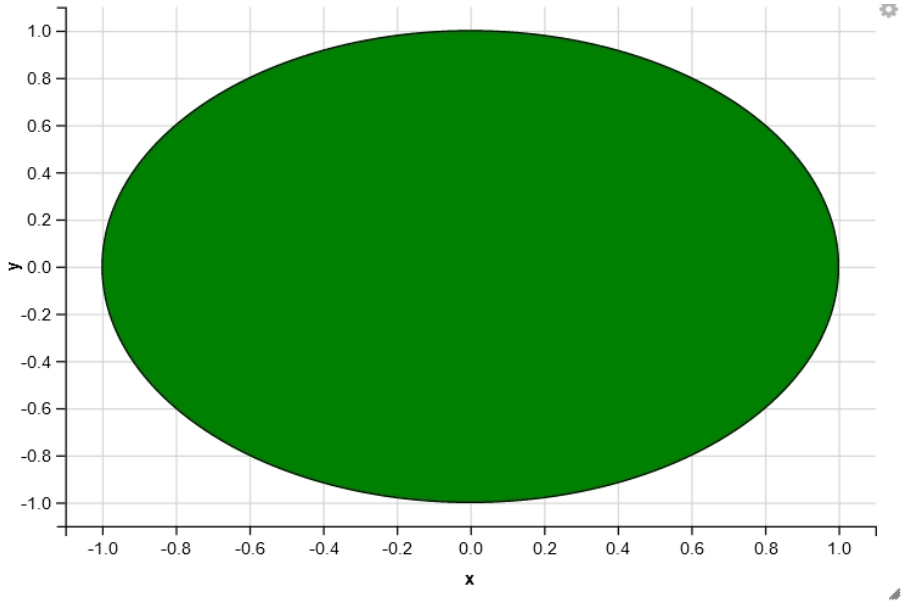
In [34]:
```
t <- seq(0, 2 * pi, length = 200)
df <- data.frame(x = sin(t), y = cos(t))
df %>% ggvis(~x, ~y) %>% layer_paths(fill := "green")
```

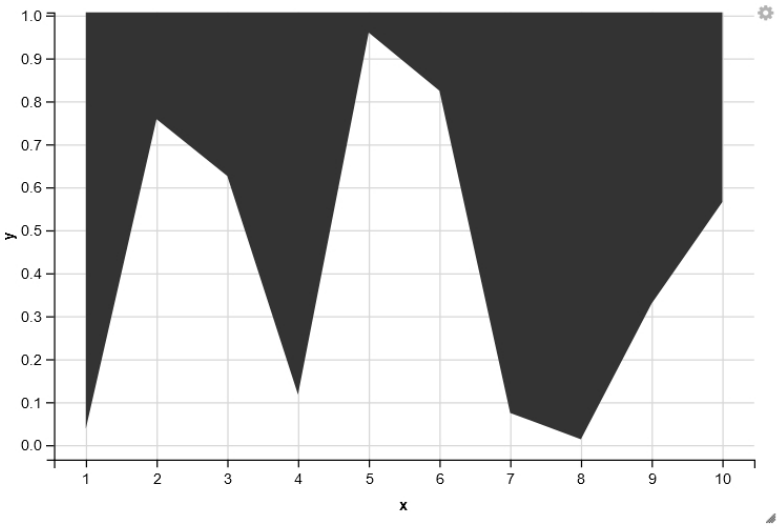
In [35]:
```
df <- data.frame(x = 1:10, y = runif(10))
df %>% ggvis(~x, ~y) %>% layer_ribbons()
```

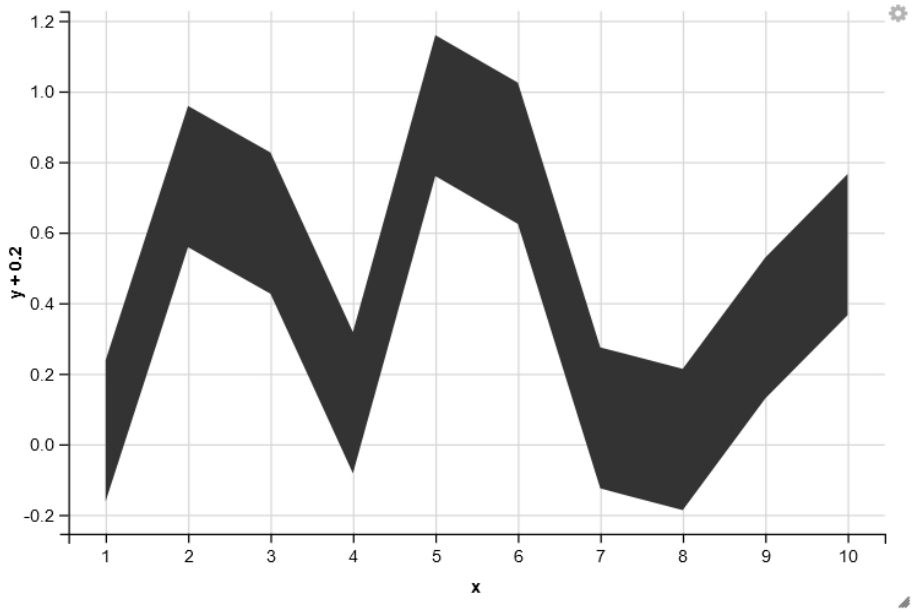
In [36]:
```
df %>% ggvis(~x, ~y +0.2, y2= ~y-0.2) %>% layer_ribbons()
```

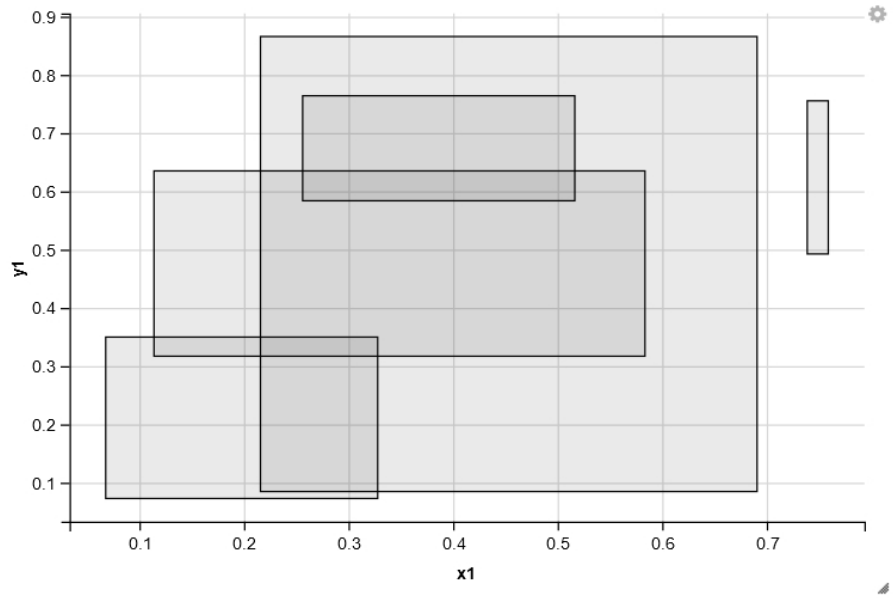
In [37]:
```
set.seed(1000)
df <- data.frame(x1 = runif(5), x2 = runif(5), y1 = runif(5), y2 = runif(5))
df %>% ggvis(~x1, ~y1, x2 = ~x2, y2 = ~y2, fillOpacity := 0.1) %>% layer_rects()
```


0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
y
1
2
3
4
5
6
7
8
9
10
x


1.0
0.8
0.6
0.4
0.2
0.0
-0.2
-0.4
-0.6
-0.8
-1.0
y
-1.0
-0.8
-0.6
-0.4
-0.2
0.0
0.2
0.4
0.6
0.8
1.0
x


1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
y
1
2
3
4
5
6
7
8
9
10
x

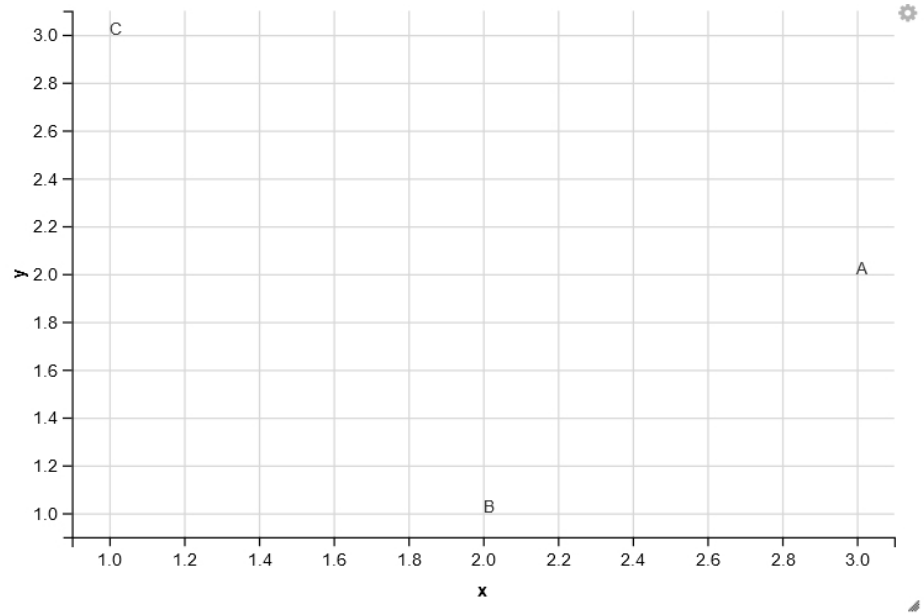


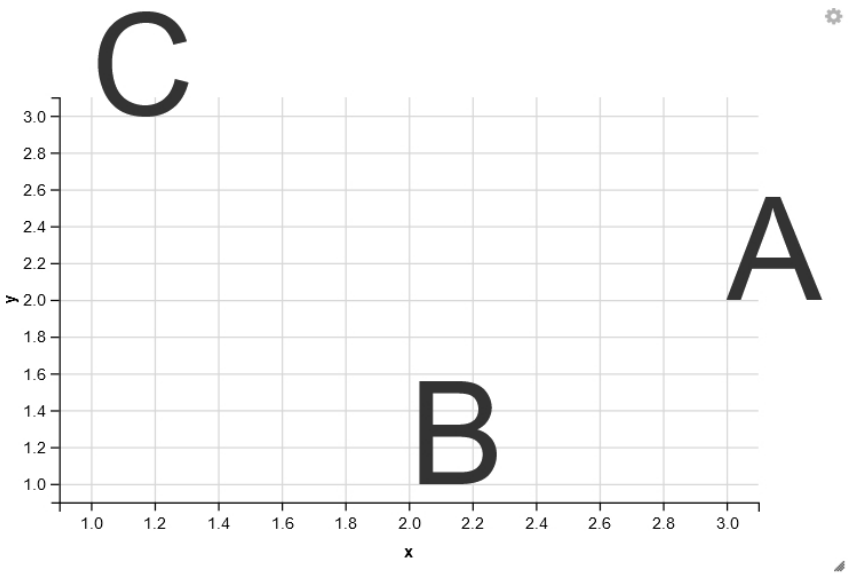


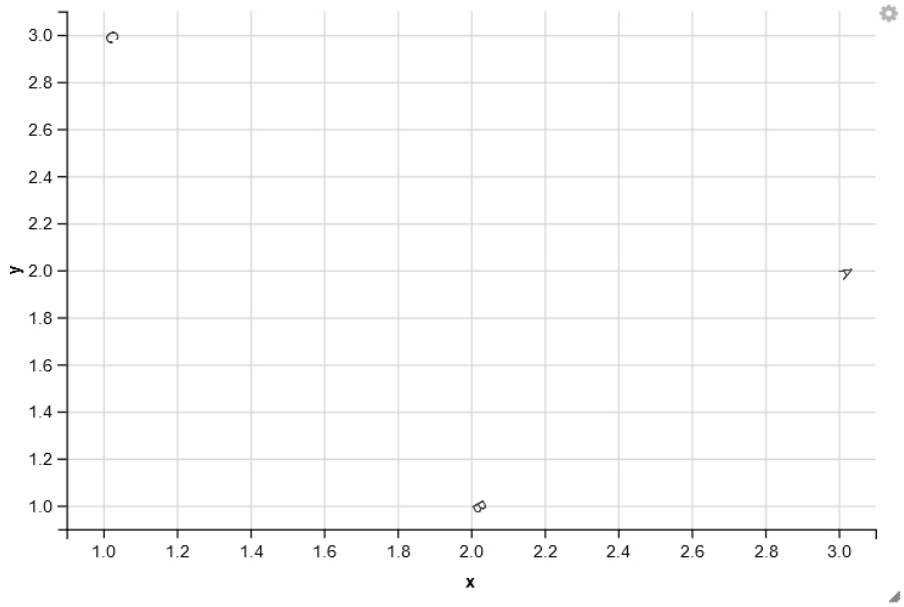

```
In [38]: df <- data.frame(x = 3:1, y = c(2,1,3), label = c("A", "B", "C"))
         df %>% ggvis(~x, ~y, text := ~label) %>% layer_text()
```

```
In [39]: df %>% ggvis(~x, ~y, text := ~label) %>% layer_text(fontSize := 100)
```

```
In [40]: df %>% ggvis(~x, ~y, text := ~label) %>% layer_text(angle := 60)
```


C
A
B
3.0
2.8
2.6
2.4
2.2
2.0
1.8
1.6
1.4
1.2
1.0
y
1.0
1.2
1.4
1.6
1.8
2.0
2.2
2.4
2.6
2.8
3.0
x


C
A
B
3.0
2.8
2.6
2.4
2.2
2.0
1.8
1.6
1.4
1.2
1.0
y
1.0
1.2
1.4
1.6
1.8
2.0
2.2
2.4
2.6
2.8
3.0
x


C
A
B
3.0
2.8
2.6
2.4
2.2
2.0
1.8
1.6
1.4
1.2
1.0
y
1.0
1.2
1.4
1.6
1.8
2.0
2.2
2.4
2.6
2.8
3.0
x

Compound Layers

In [41]:
```
t <- seq(0, 2 * pi, length = 50)
df <- data.frame(x = sin(t), y = cos(t))
df %>% ggvis(~x, ~y) %>% layer_paths()
```

In [42]:
```
df %>% ggvis(~x, ~y) %>% layer_lines()
```

In [43]:
```
df %>% ggvis(~x, ~y) %>% arrange(x) %>% layer_paths()
```

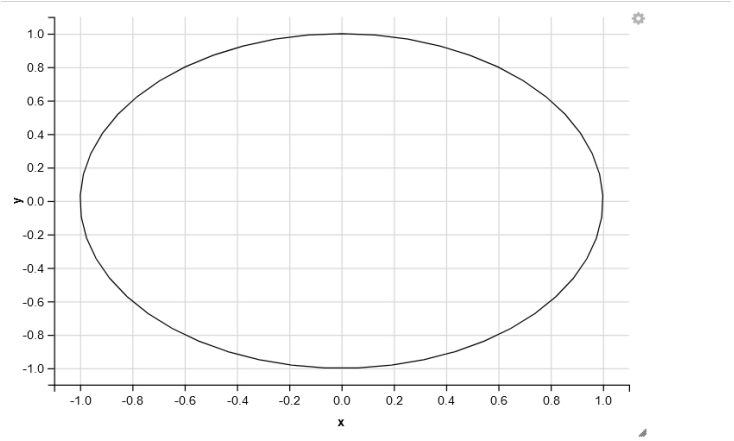
In [44]:
```
airquality %>% ggvis(~Month) %>% layer_histograms()
```

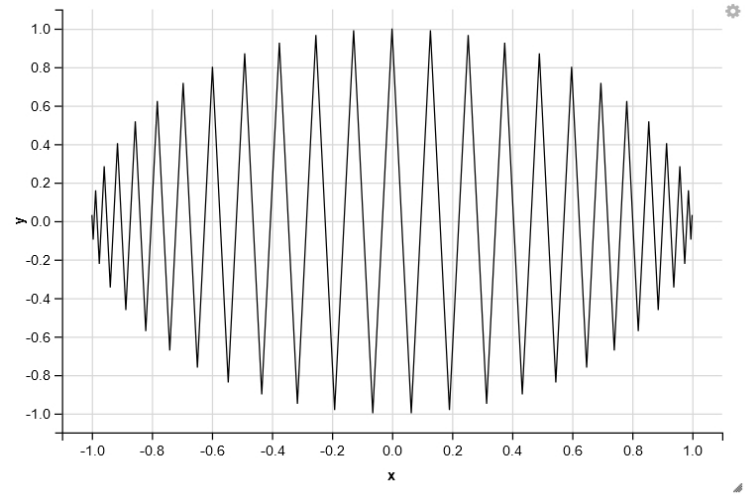
```
Guessing width = 0.1 # range / 40
```

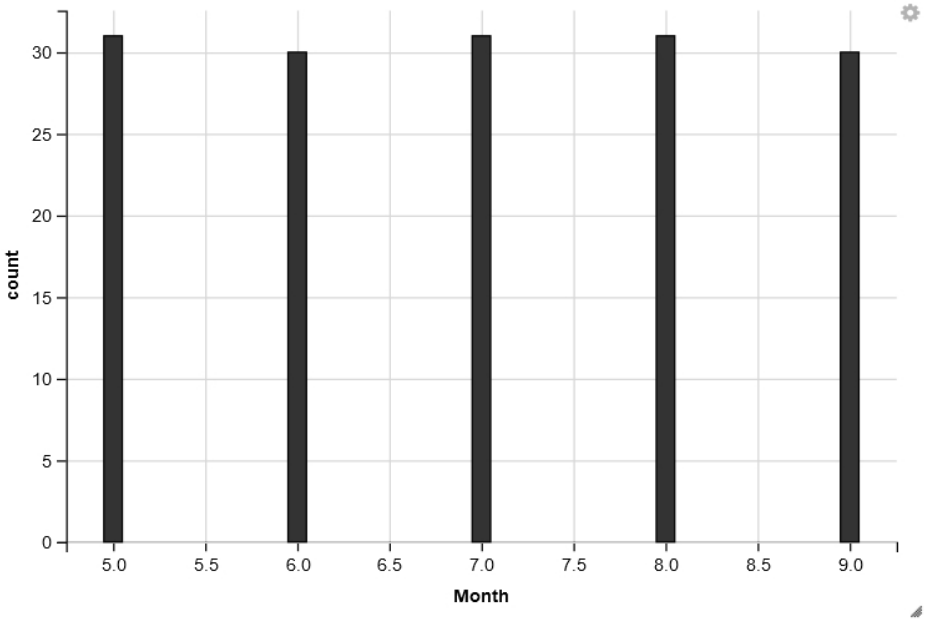
In [45]:
```
binned <- airquality %>% compute_bin(~Month)
```

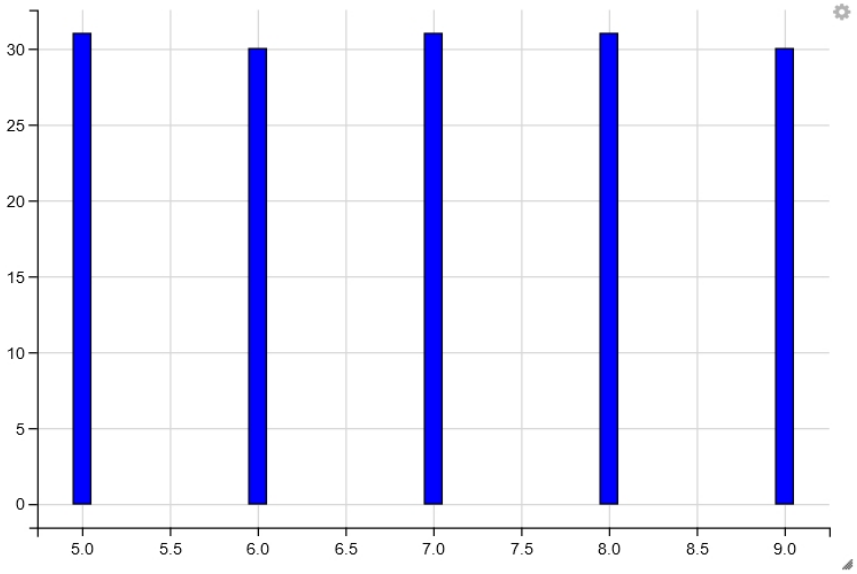
```
Guessing width = 0.1 # range / 40
```

In [46]:
```
binned %>%
  ggvis(x = ~xmin_, x2 = ~xmax_, y2 = 0, y = ~count_, fill := "blue") %>%
  layer_rects()
```





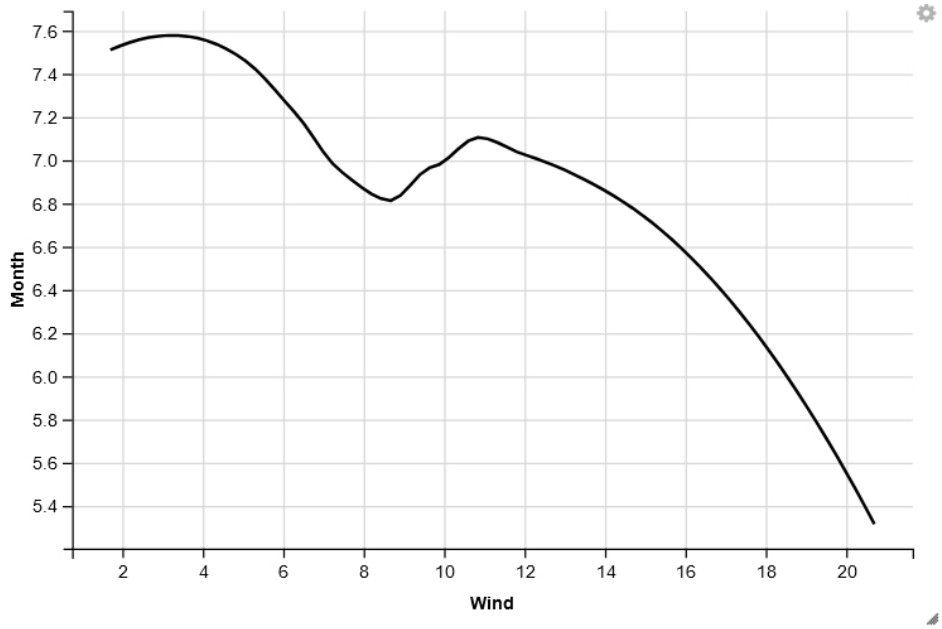


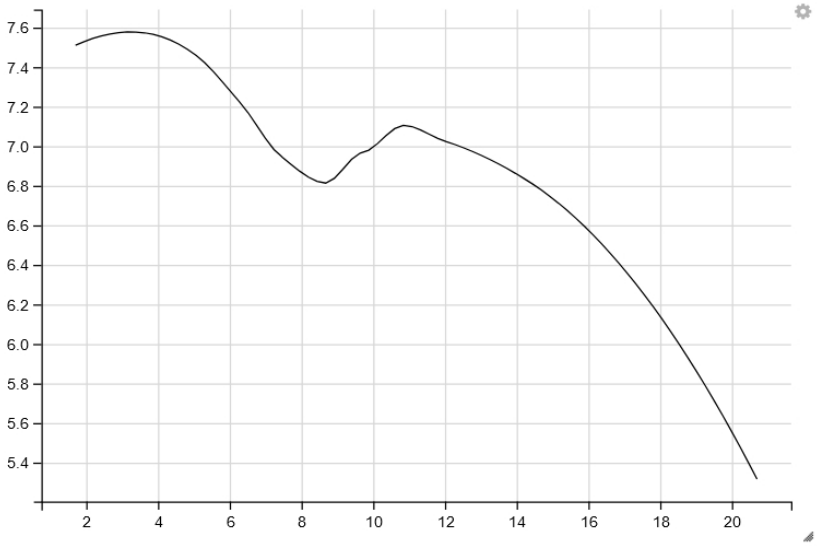


In [46]:
```
binned %>%
  ggvis(x = ~xmin_, x2 = ~xmax_, y2 = 0, y = ~count_, fill := "blue") %>%
  layer_rects()
```

In [47]:
```
airquality %>% ggvis(~Wind, ~Month) %>% layer_smooths()
```

In [48]:
```
smoothed <- airquality %>% compute_smooth(Month ~ Wind)
smoothed %>% ggvis(~pred_, ~resp_) %>% layer_paths()
```

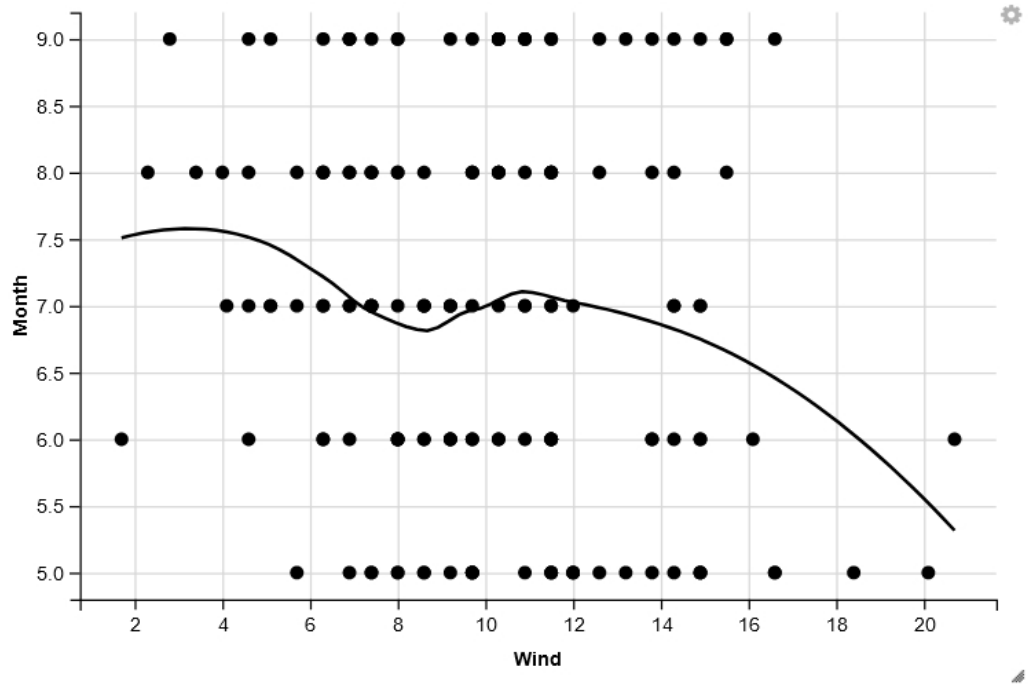


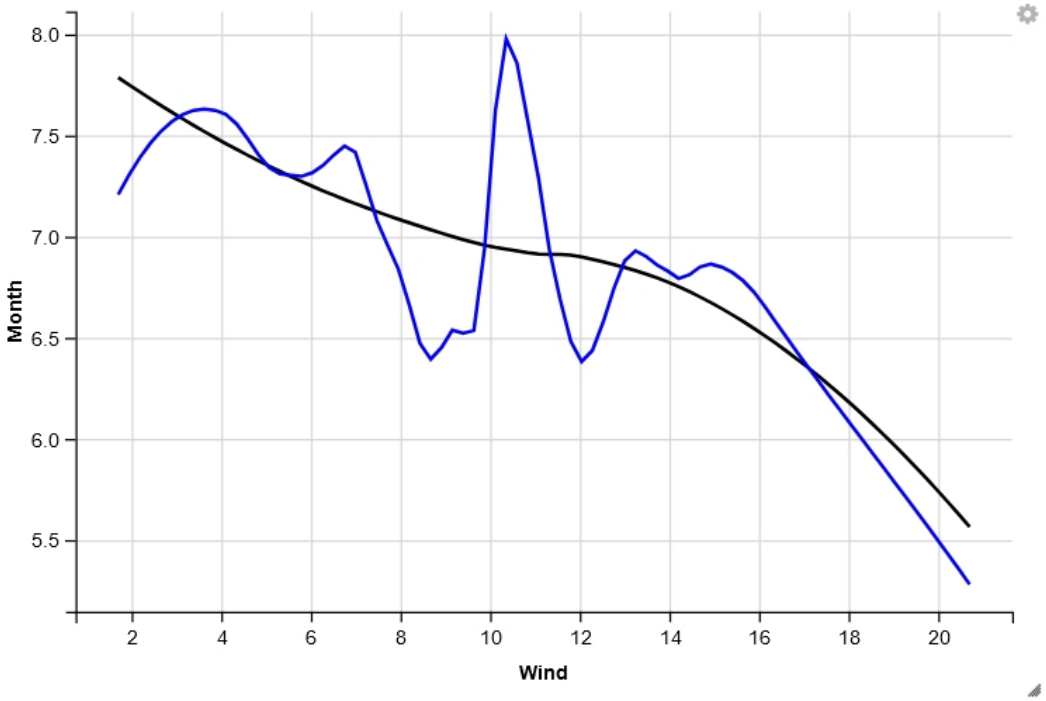


Multiple Layers

In [50]:
```
airquality %>%
   ggvis(~Wind,~Month) %>%
   layer_smooths() %>%
   layer_points()
```

In [51]:
```
airquality %>% ggvis(~Wind,~Month) %>%
  layer_smooths(span = 1) %>%
  layer_smooths(span = 0.3,stroke := "blue")
```


9.0
8.5
8.0
7.5
7.0
6.5
6.0
5.5
5.0
Month
2
4
6
8
10
12
14
16
18
20
Wind


8.0
7.5
7.0
6.5
6.0
5.5
Month
2
4
6
8
10
12
14
16
18
20
Wind

2.Draw the linked view for Cloud multivariate dataset (available in UCI repository)

In [9]:
```
import pandas as pd
data = pd.read_csv("C:/Users/gyanada/OneDrive/Desktop/VIT/WIN2024-2025/DV/cars_data.csv")
print(data.head())
print(data.describe())
```

```
     customer name                                     customer e-mail  \
0    Martina Avila  cubilia.Curae.Phasellus@quisaccumsanconvallis.edu
1    Harlan Barnes                                eu.dolor@diam.co.uk
2  Naomi Rodriquez  vulputate.mauris.sagittis@ametconsectetueradip...
3  Jade Cunningham                           malesuada@dignissim.com
4     Cedric Leach     felis.ullamcorper.viverra@egetmollislectus.net

        country  gender        age  annual Salary  credit card debt  \
0      Bulgaria       0  41.851720    62812.09301      11609.380910
1        Belize       0  40.870623    66646.89292       9572.957136
2       Algeria       1  43.152897    53798.55112      11160.355060
3  Cook Islands       1  58.271369    79370.03798      14426.164850
4        Brazil       1  57.313749    59729.15130       5358.712177

     net worth  car purchase amount
0  238961.2505          35321.45877
1  530973.9078          45115.52566
2  638467.1773          42925.70921
3  548599.0524          67422.36313
4  560304.0671          55915.46248
           gender         age  annual Salary  credit card debt      net worth  \
count  14.000000  14.000000      14.000000         14.000000      14.000000
mean    0.642857  48.656943   59173.633634       9824.792464  483286.367821
std     0.497245   5.892816   11871.141392       3213.653397  131888.540115
min     0.000000  40.870623   37336.338300       3440.823799  238961.250500
25%     0.000000  43.591960   53542.463770       8569.779301  422362.892750
50%     1.000000  47.551915   58934.205200      10116.645105  503808.631100
75%     1.000000  52.433373   67890.077965      11497.124448  599629.979600
max     1.000000  58.271369   79370.037980      14426.164850  638467.177300

       car purchase amount
count            14.000000
mean          46083.539857
std           11254.169328
min           28700.033400
25%           39214.332122
50%           46275.254155
75%           54251.315788
max           67422.363130
```

In [11]:
```
print(data.columns)
```

```
Index(['customer name', 'customer e-mail', 'country', 'gender', 'age',
       'annual Salary', 'credit card debt', 'net worth',
       'car purchase amount'],
      dtype='object')
```

INTERACTIVE SCATTER PLOTS:

CODE:

```
import plotly.graph_objs as go

from plotly.subplots import make_subplots

import pandas as pd

import plotly.io as pio
```

```
data = pd.read_csv("C:/Users/gyanada/OneDrive/Desktop/VIT/WIN2024-2025/DV/cars_data.csv")


callback = """
function(event, plotly_data) {
  var selected_data = {
      'net worth': [],
      'car purchase amount': [],
      'annual Salary': [],
      'age': [],
      'gender': [],
      'country': []
  };

  plotly_data.points.forEach(point => {
    selected_data['net worth'].push(point.x);
    selected_data['car purchase amount'].push(point.y);
    selected_data['annual Salary'].push(point.customdata[0]);
    selected_data['age'].push(point.customdata[1]);
    selected_data['gender'].push(point.customdata[2]);
    selected_data['country'].push(point.customdata[3]);
  });

  Plotly.restyle('selected-plot', selected_data);
}
"""


data = data.rename(columns={'annual Salary': 'annual_Salary', 'net worth': 'net_worth', 'car purchase
amount': 'car_purchase_amount'})


plots = []
features = ['net_worth', 'car_purchase_amount', 'annual_Salary', 'age', 'gender', 'country']
for i, feature1 in enumerate(features):
    for feature2 in features[i+1:]:
        fig = go.FigureWidget(
```

```
            data=[
                go.Scatter(
                    x=data[feature1],
                    y=data[feature2],
                    mode='markers',
                    customdata=data[['annual_Salary', 'age', 'gender', 'country']],
                    marker=dict(size=10, color='blue', opacity=0.5)
                )
            ],
            layout=dict(
                title=f"{feature1} vs {feature2}",
                xaxis=dict(title=feature1),
                yaxis=dict(title=feature2)
            )
        )
        fig.data[0].on_selection(lambda x, y: exec(callback))
        plots.append(fig)


rows = (len(plots) // 2) + (1 if len(plots) % 2 != 0 else 0)
fig = make_subplots(rows=rows, cols=2, subplot_titles=[plot.layout.title.text for plot in plots])
for i, plot in enumerate(plots, start=1):
    row = (i - 1) // 2 + 1
    col = (i - 1) % 2 + 1
    fig.add_trace(plot.data[0], row=row, col=col)

fig.update_layout(title_text="Interactive Scatter Plots")
pio.write_html(fig, "scatter_plots.html", auto_open=True)
```

“On moving cursor we can see values”

Output link:

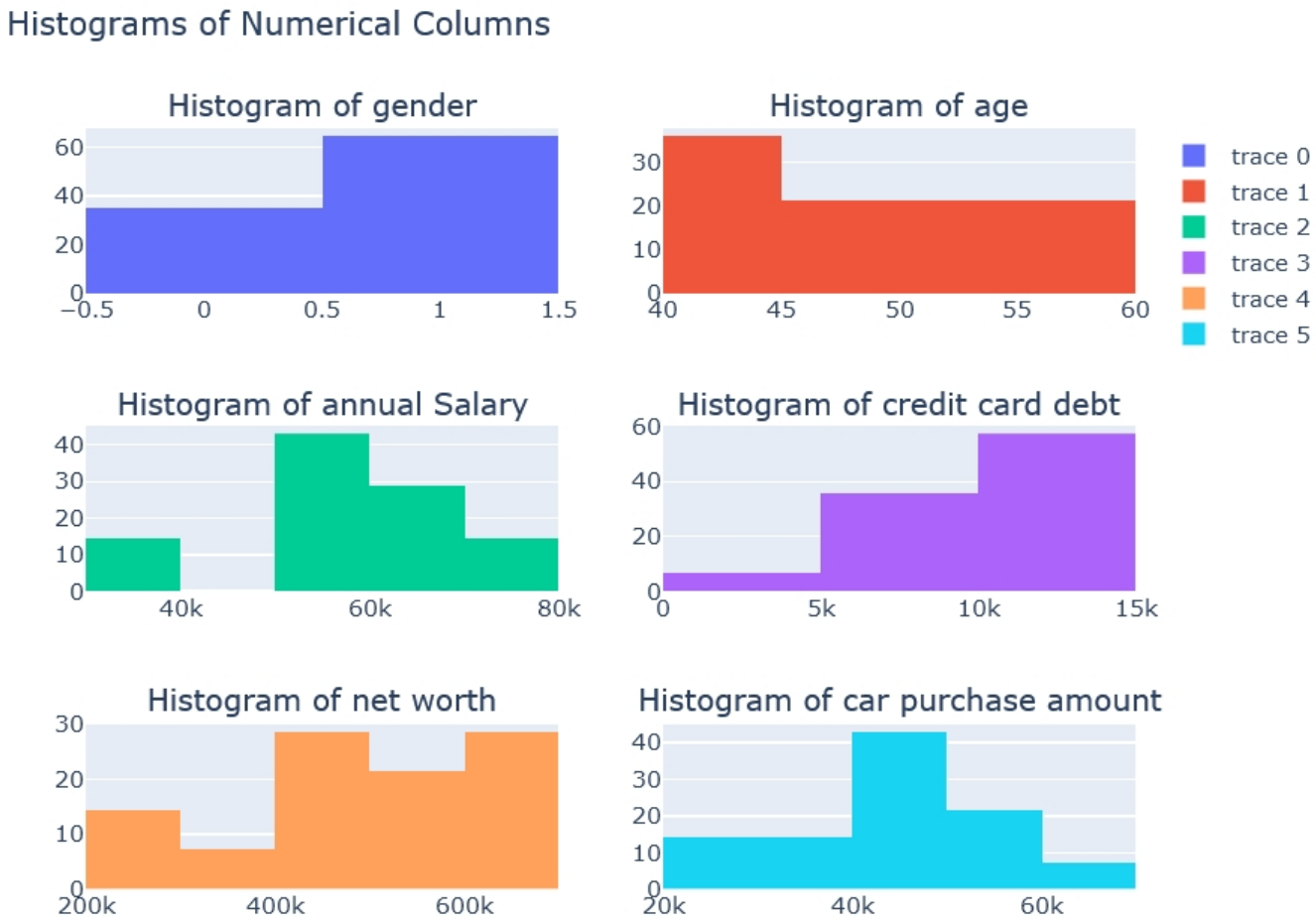
file:///C:/Users/gyanada/scatter_plots.html

INTERACTIVE HISTOGRAM:

CODE:

```
import plotly.graph_objs as go
from plotly.subplots import make_subplots
import pandas as pd
import plotly.io as pio
data = pd.read_csv("C:/Users/gyanada/OneDrive/Desktop/VIT/WIN2024-2025/DV/cars_data.csv")

fig = make_subplots(rows=len(data.select_dtypes(include=['float64', 'int64']).columns) // 2 + 1, cols=2,
                    subplot_titles=[f"Histogram of {col}" for col in data.select_dtypes(include=['float64', 'int64']).columns])

for i, col in enumerate(data.select_dtypes(include=['float64', 'int64']).columns, start=1):
    fig.add_trace(go.Histogram(x=data[col], histnorm='percent'), row=(i - 1) // 2 + 1, col=(i - 1) % 2 + 1)

fig.update_layout(title_text="Histograms of Numerical Columns")
```

pio.write_html(fig, "histograms.html", auto_open=True)



Output link:

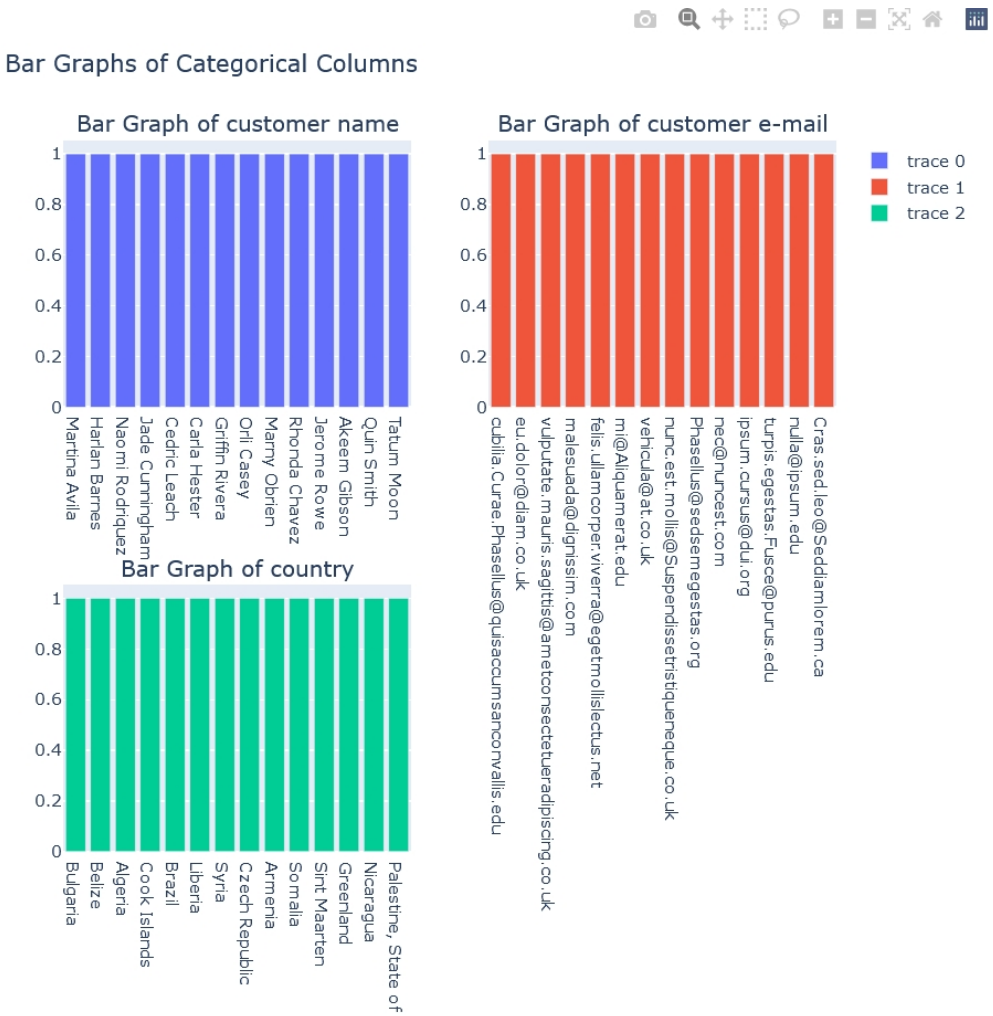
file:///C:/Users/gyanada/histograms.html

INTERACTIVE BARGRAPH:

CODE:

```
import plotly.graph_objs as go
from plotly.subplots import make_subplots
import pandas as pd
import plotly.io as pio

# Load the CSV file
data = pd.read_csv("C:/Users/gyanada/OneDrive/Desktop/VIT/WIN2024-2025/DV/cars_data.csv")

# Create bar graphs for each categorical column
```

```
fig = make_subplots(rows=len(data.select_dtypes(include=['object']).columns) // 2 + 1, cols=2,
                    subplot_titles=[f"Bar Graph of {col}" for col in data.select_dtypes(include=['object']).columns])

for i, col in enumerate(data.select_dtypes(include=['object']).columns, start=1):
    value_counts = data[col].value_counts()
    fig.add_trace(go.Bar(x=value_counts.index, y=value_counts.values), row=(i - 1) // 2 + 1, col=(i - 1) % 2 + 1)

fig.update_layout(title_text="Bar Graphs of Categorical Columns")

# Save the subplot as an HTML file and open it
pio.write_html(fig, "bar_graphs.html", auto_open=True)
```



Output link:

file:///C:/Users/gyanada/bar_graphs.html