In [3]:

```
import pandas as pd
import numpy as np
import plotly.express as px
import seaborn as sns
import matplotlib.pyplot as plt
```

In [ ]:

```
#1. Write a python program to read a dataset from company sales data.
#a) Read Total profit of all months and show it using a line plot.
#b) Read all product sales data and show it using a multiline plot.
#c) Read toothpaste sales data of each month and show it using a scatter plot.
#d) Read sales data of bathing soap of all months and show it using a bar chart.
#e) Calculate total sale data for last year for each product and show it using a Pie chart.
```

In [5]:

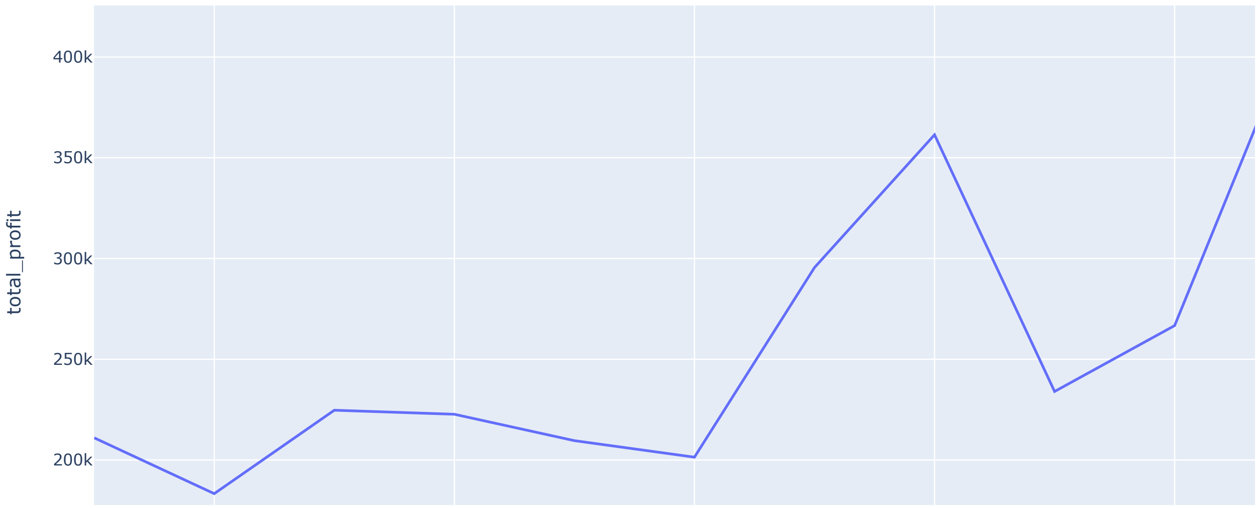
```
cs=pd.read_csv('company-sales.csv')
cs.head()
```

Out[5]:

	month_number	facecream	facewash	toothpaste	bathingsoap	shampoo	moisturizer	total_units	total_profit
0	1	2500	1500	5200	9200	1200	1500	21100	211000
1	2	2630	1200	5100	6100	2100	1200	18330	183300
2	3	2140	1340	4550	9550	3550	1340	22470	224700
3	4	3400	1130	5870	8870	1870	1130	22270	222700
4	5	3600	1740	4560	7760	1560	1740	20960	209600

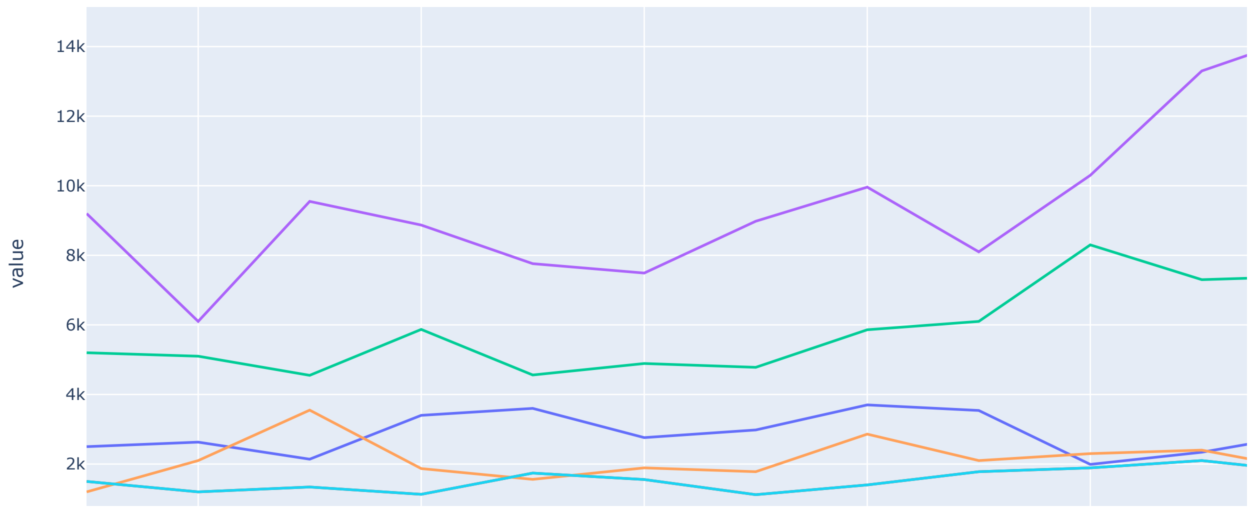
In [6]:

```
px.line(cs,x='month_number',y='total_profit')
```

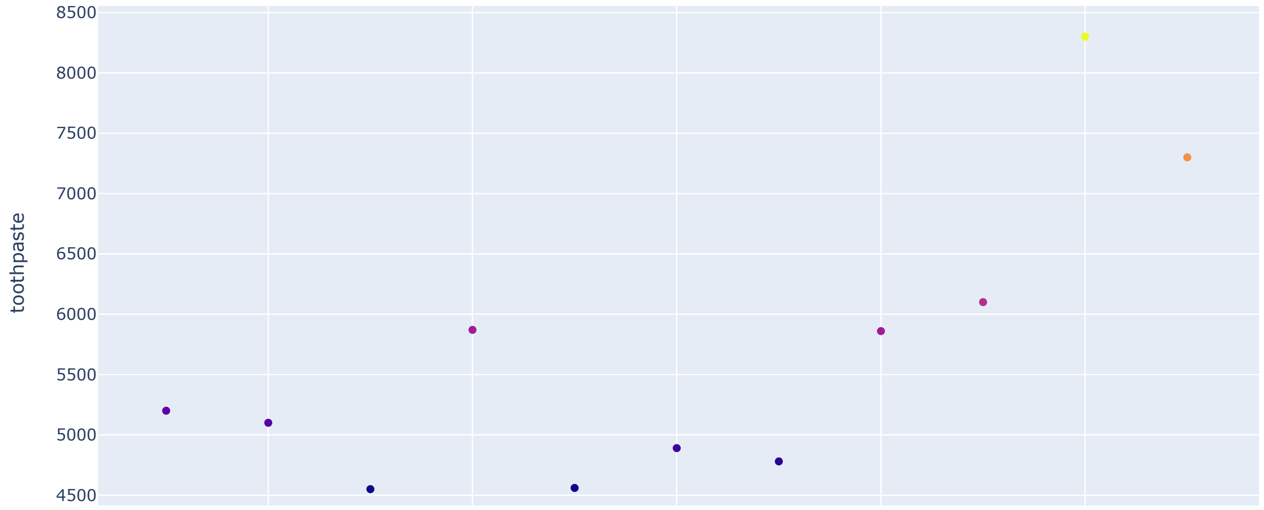


In [9]:

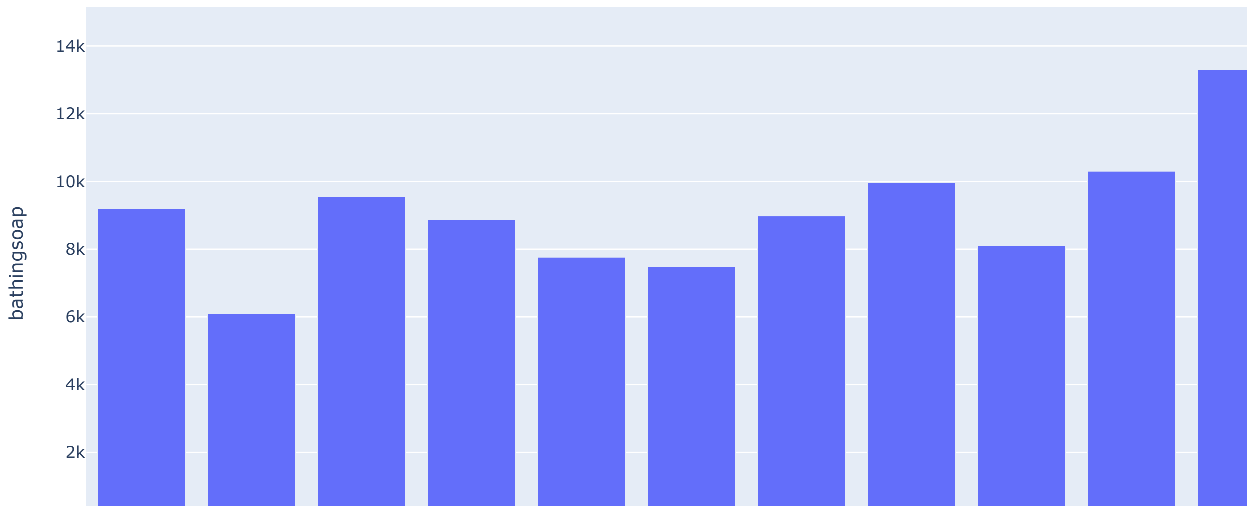
```
px.line(cs,x='month_number',y=['facecream','facewash','toothpaste','bathingsoap','shampoo','moisturizer'])
```



```
In [11]: px.scatter(cs,x='month_number',y='toothpaste',color='toothpaste')
```

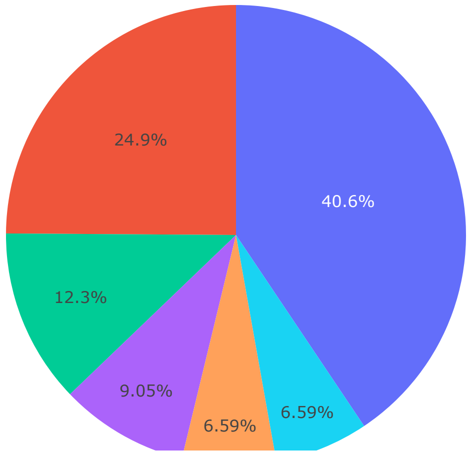


```
In [12]: px.bar(cs,x='month_number',y='bathingsoap')
```



```
In [31]: fc=cs['facecream'].sum()
         fw=cs['facewash'].sum()
         tp=cs['toothpaste'].sum()
         bs=cs['bathingsoap'].sum()
         sh=cs['shampoo'].sum()
         ms=cs['moisturizer'].sum()
         val=[fc,fw,tp,bs,sh,ms]
         px.pie(cs,values=val,names=['facecream','facewash','toothpaste','bathingsoap','shampoo','moisturizer'],title='Total Sale Data for Las
```

Total Sale Data for Last Year



```
In [ ]: #2. Read the weather csv file and plot the heatmap of correlated columns.
```

```
In [48]: wt=pd.read_csv('weather.csv')
         wt.head()
```

Out[48]:

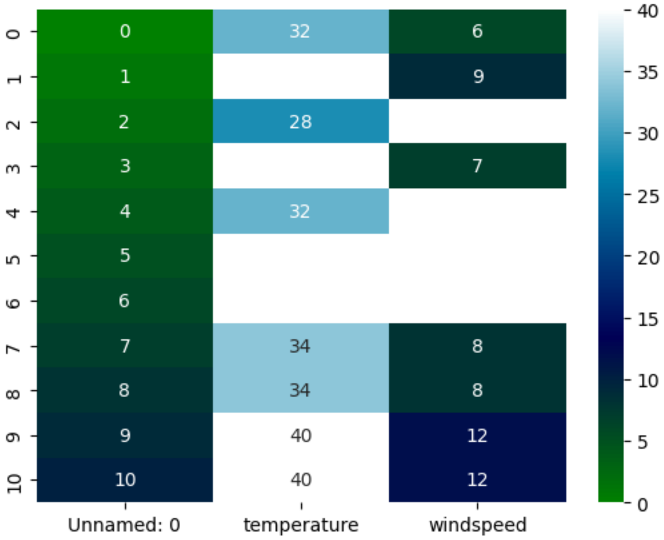
	Unnamed: 0	day	temperature	windspeed	event
0	0	01-01-2022	32.0	6.0	Rain
1	1	01-04-2022	NaN	9.0	Sunny
2	2	01-05-2022	28.0	NaN	Snow
3	3	01-06-2022	NaN	7.0	NaN
4	4	01-07-2022	32.0	NaN	Rain

In [49]:

```
a=wt.select_dtypes(include=['number'])
a.corr()
sns.heatmap(a,cmap='ocean',annot=True)
```

Out[49]:

```
<Axes: >
```



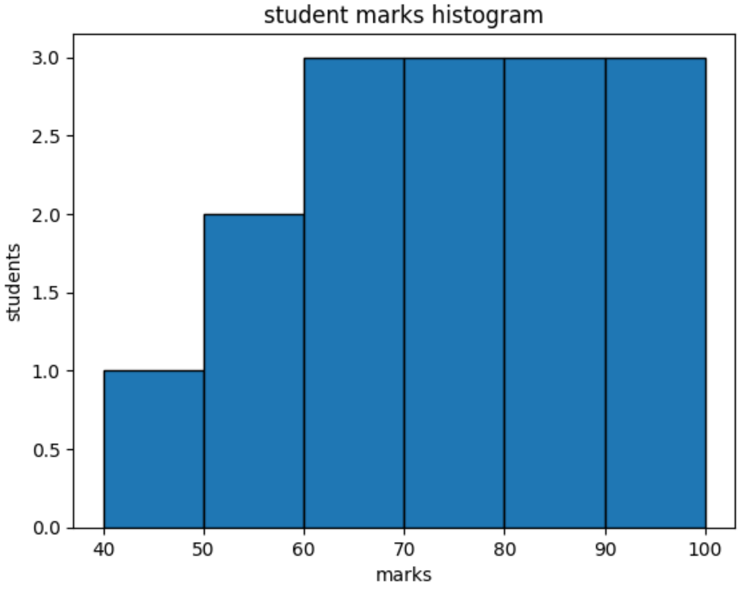
In [ ]:

```
#3. Create a list with student marks and make the bins for range of marks and plot that using histogram.
```

In [54]:

```
smarks = [70, 65, 80, 90, 60, 75, 85, 95, 55, 50, 45, 65, 75, 80, 90]
b = [40, 50, 60, 70, 80, 90, 100]

plt.hist(smarks, bins=b, edgecolor='k')
plt.title('student marks histogram')
plt.xlabel('marks')
plt.ylabel('students')
plt.show()
```



In [ ]:

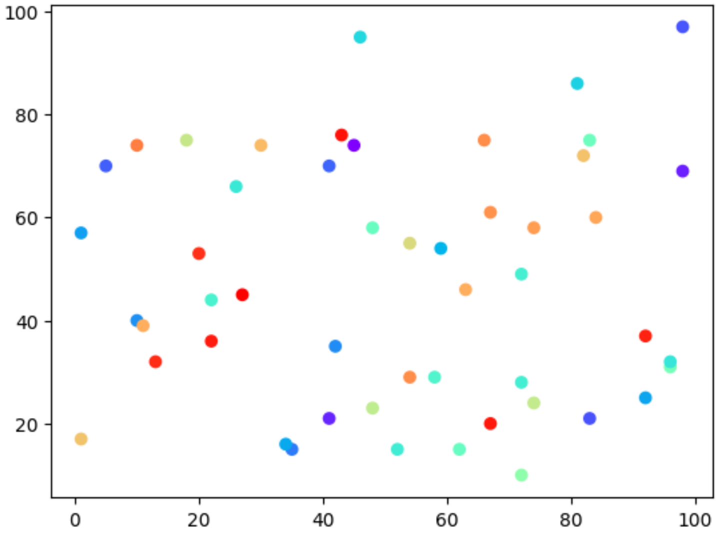
```
#4. Create a scatter plot with random integers and decorate them with color maps.
```

In [58]:

```
x = np.random.randint(1, 100, 50)
y = np.random.randint(1, 100, 50)
colors = np.random.rand(50)
plt.scatter(x, y, c=colors, cmap='rainbow')
```

Out[58]:

```
<matplotlib.collections.PathCollection at 0x21ff19453a0>
```



In [ ]:

```
#5. Read the iris dataset in seaborn and make the pairplot based on their species.
```

In [68]:

```
iris=pd.read_csv('iris.csv')
iris.head()
```

Out[68]:

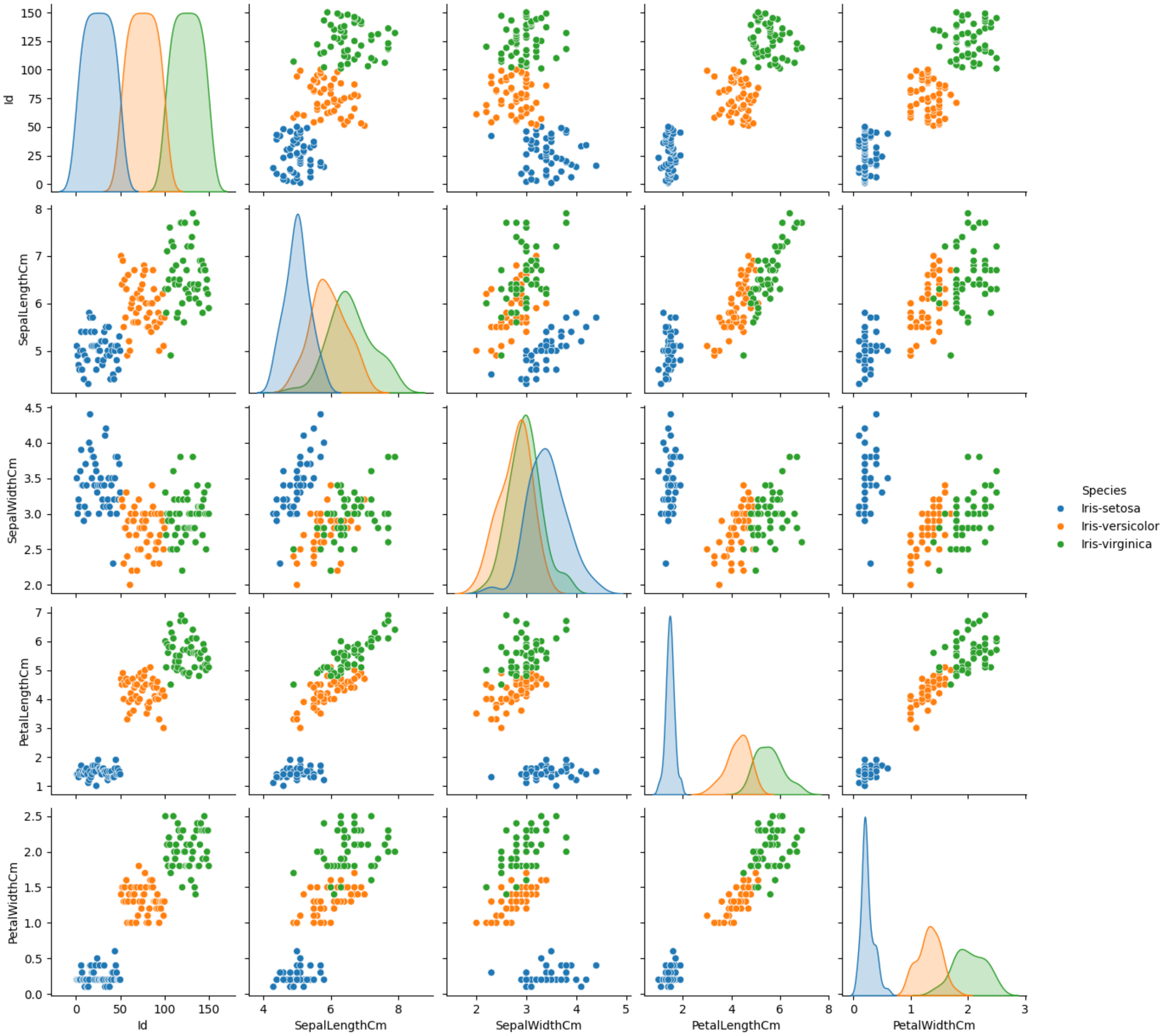
	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa

In [69]:

```
sns.pairplot(iris,hue="Species")
```

Out[69]:

```
<seaborn.axisgrid.PairGrid at 0x21ff4f5e6f0>
```



In [ ]:

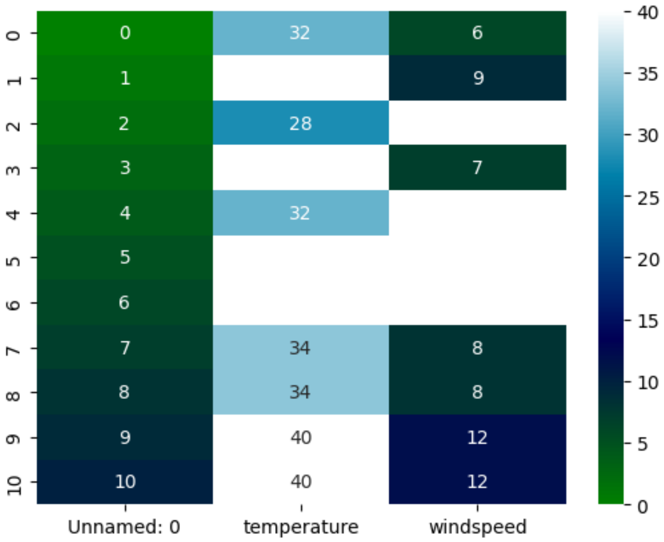
```
#6. Import csv file using pandas, find correlation and plot heatmap of correlation using seaborn, plot the scatter plot for any two hi
#correlated columns using matplotlib.
```

In [4]:

```
wt1=pd.read_csv('weather.csv')
a1=wt1.select_dtypes(include=['number'])
a1.corr()
sns.heatmap(a1,cmap='ocean',annot=True)
```

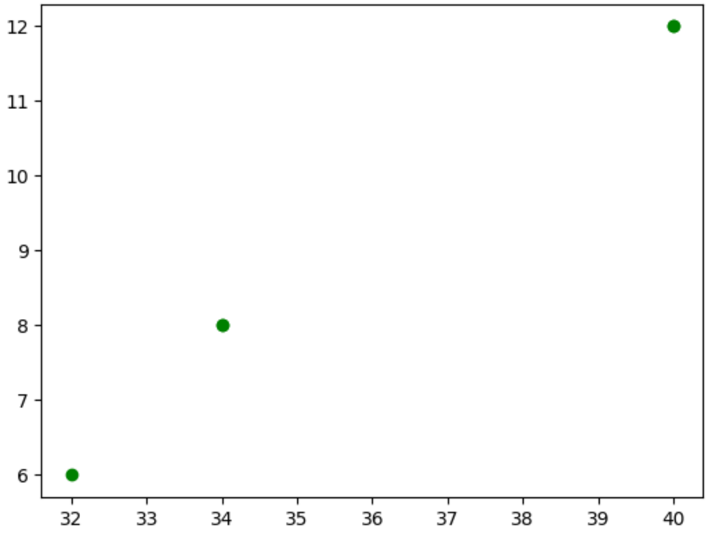
Out[4]:

```
<Axes: >
```



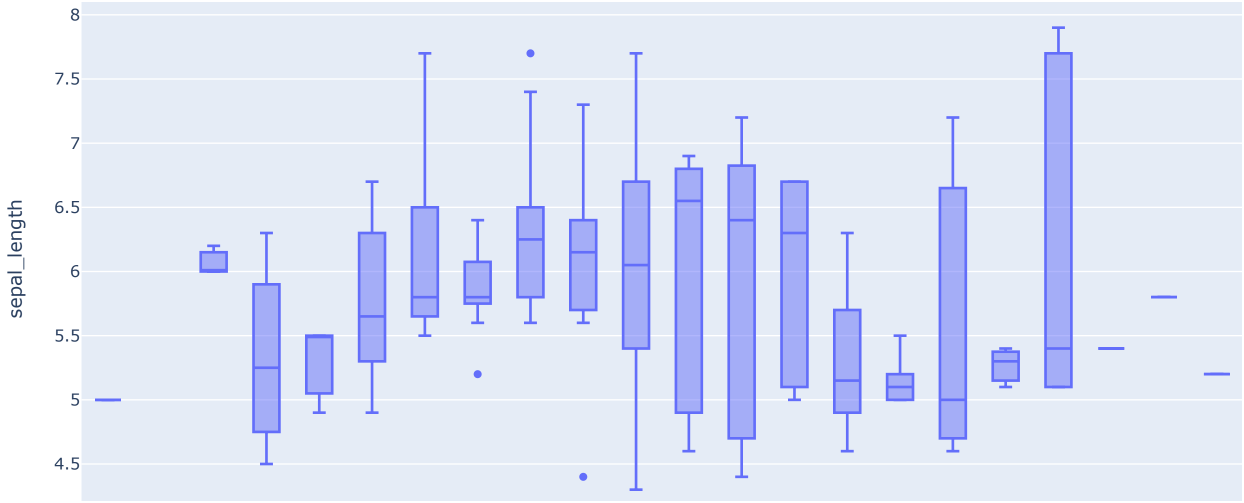
In [73]:
```
plt.scatter(a1['temperature'],a1['windspeed'],color='g')
```

Out[73]:
```
<matplotlib.collections.PathCollection at 0x21ff502ed80>
```



In [ ]:
```
#7. Write a python program to and make the following charts using plotly:
#• Boxplot of iris data x axis values is sepal-width and y value as sepal-length.
#• Gantt chart for A(2023-01-01, 2023-02-28), B(2023-03-05, 2023-04-15), C(2023-02-20, 2023-05-30).
#• Sunburst plot, Treemap and Icicle maps for the tips data. Take path as day,sex and value as total-bill.
#• Heatmap with metals-wide data.
```

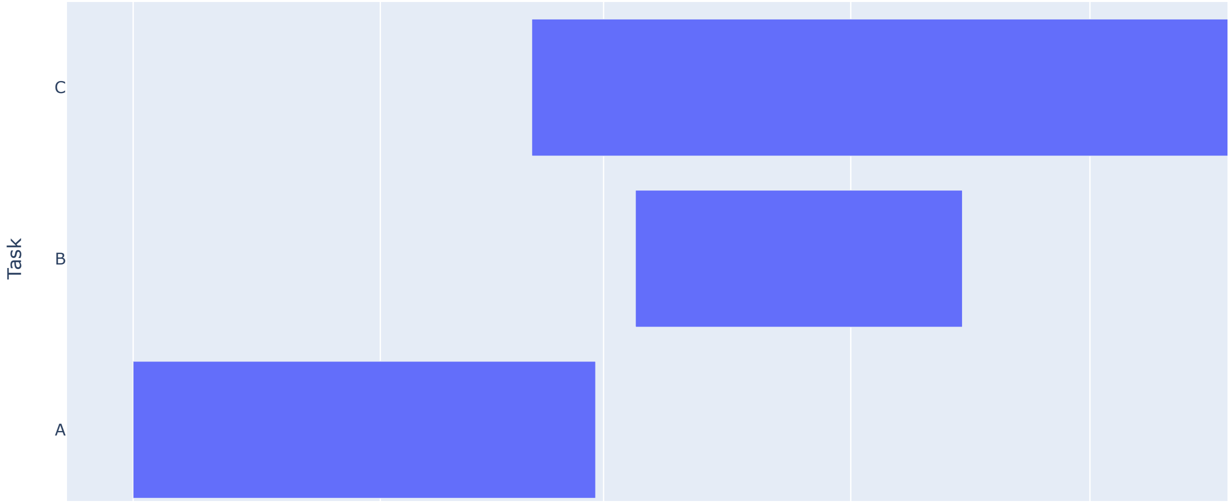
In [6]:
```
iris = px.data.iris()
px.box(iris, x='sepal_width', y='sepal_length')
```



In [7]:

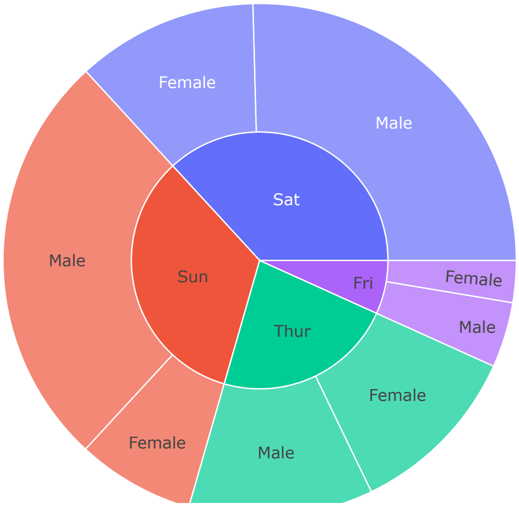
```
data = [
    dict(Task='A', Start='2023-01-01', Finish='2023-02-28'),
    dict(Task='B', Start='2023-03-05', Finish='2023-04-15'),
    dict(Task='C', Start='2023-02-20', Finish='2023-05-30')
]

px.timeline(data, x_start='Start', x_end='Finish', y='Task')
```

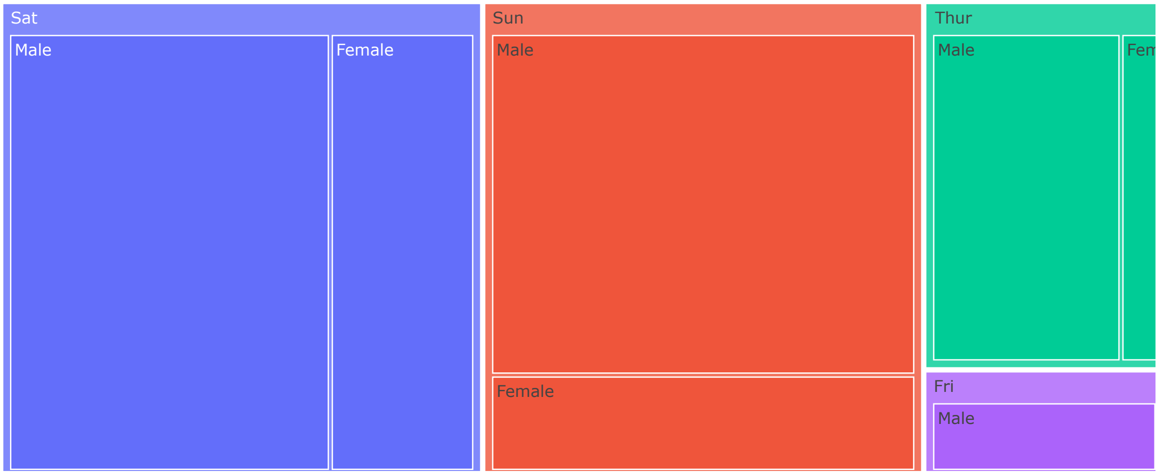


In [8]:

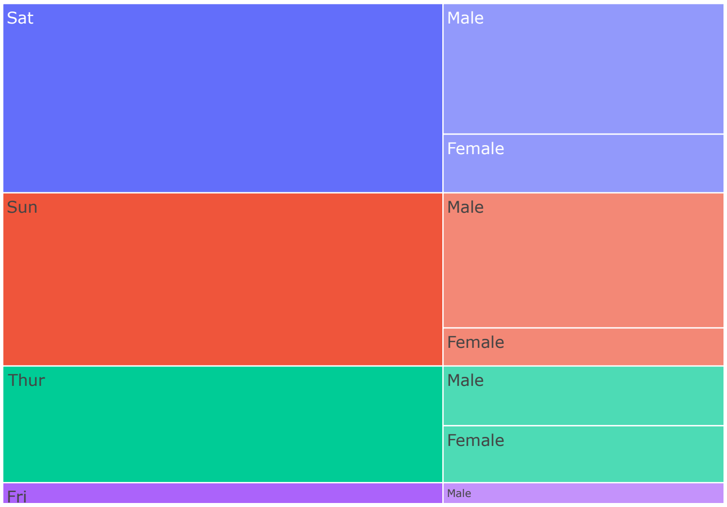
```
tips = px.data.tips()
px.sunburst(tips, path=['day', 'sex'], values='total_bill')
```



In [9]:
```
px.treemap(tips, path=['day', 'sex'], values='total_bill')
```

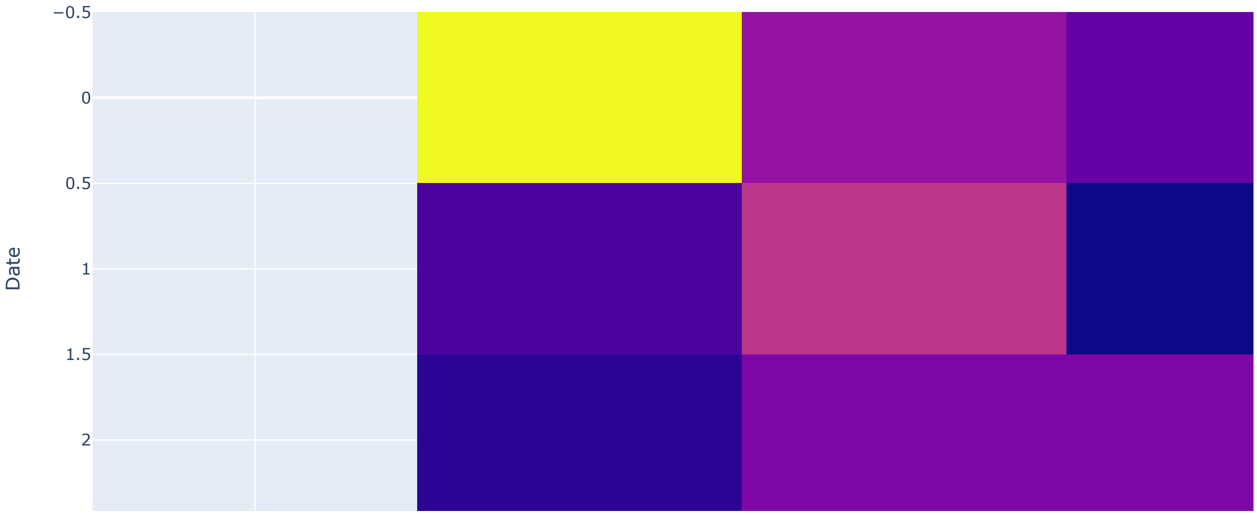


In [10]:
```
px.icicle(tips, path=['day', 'sex'], values='total_bill')
```



In [14]:

```
medals_wide = px.data.medals_wide()
px.imshow(medals_wide, labels=dict(x="Metal", y="Date", color="Price"))
```



In [ ]: