DS ACTIVE LEARNING GROUP - 12

In [84]:

```
pip install plotly
```

```
Requirement already satisfied: plotly in c:\programdata\anaconda3\lib\site-packages (4.12.
0)
Requirement already satisfied: six in c:\programdata\anaconda3\lib\site-packages (from plo
tly) (1.15.0)
Requirement already satisfied: retrying>=1.3.3 in c:\programdata\anaconda3\lib\site-packag
es (from plotly) (1.3.3)
Note: you may need to restart the kernel to use updated packages.
```

In [1]:

```
import pandas as pd
import numpy as np
import plotly.graph_objects as go
```

In [2]:

```
df=pd.read_csv('iris.csv')
df.rename(columns = {'sepal width' : 'sepalwidth'}, inplace = True)
```

In [3]:

```
df.head()
df['Species'].unique()
```

Out[3]:

```
array(['Iris-setosa', 'Iris-versicolor', 'Iris-virginica'], dtype=object)
```

In [4]:

```
df.isnull().sum()
```

Out[4]:

```
sepallentgth    0
sepalwidth      0
petallength     0
petalwidth      0
Species         0
dtype: int64
```

In [5]:

```
df.shape
```

Out[5]:

```
(150, 5)
```

In [6]:

```
df.describe()
```

Out[6]:

	sepallentgth	sepalwidth	petallength	petalwidth
count	150.000000	150.000000	150.000000	150.000000
mean	5.843333	3.054000	3.758667	1.198667
std	0.828066	0.433594	1.764420	0.763161
min	4.300000	2.000000	1.000000	0.100000
25%	5.100000	2.800000	1.600000	0.300000
50%	5.800000	3.000000	4.350000	1.300000
75%	6.400000	3.300000	5.100000	1.800000
max	7.900000	4.400000	6.900000	2.500000

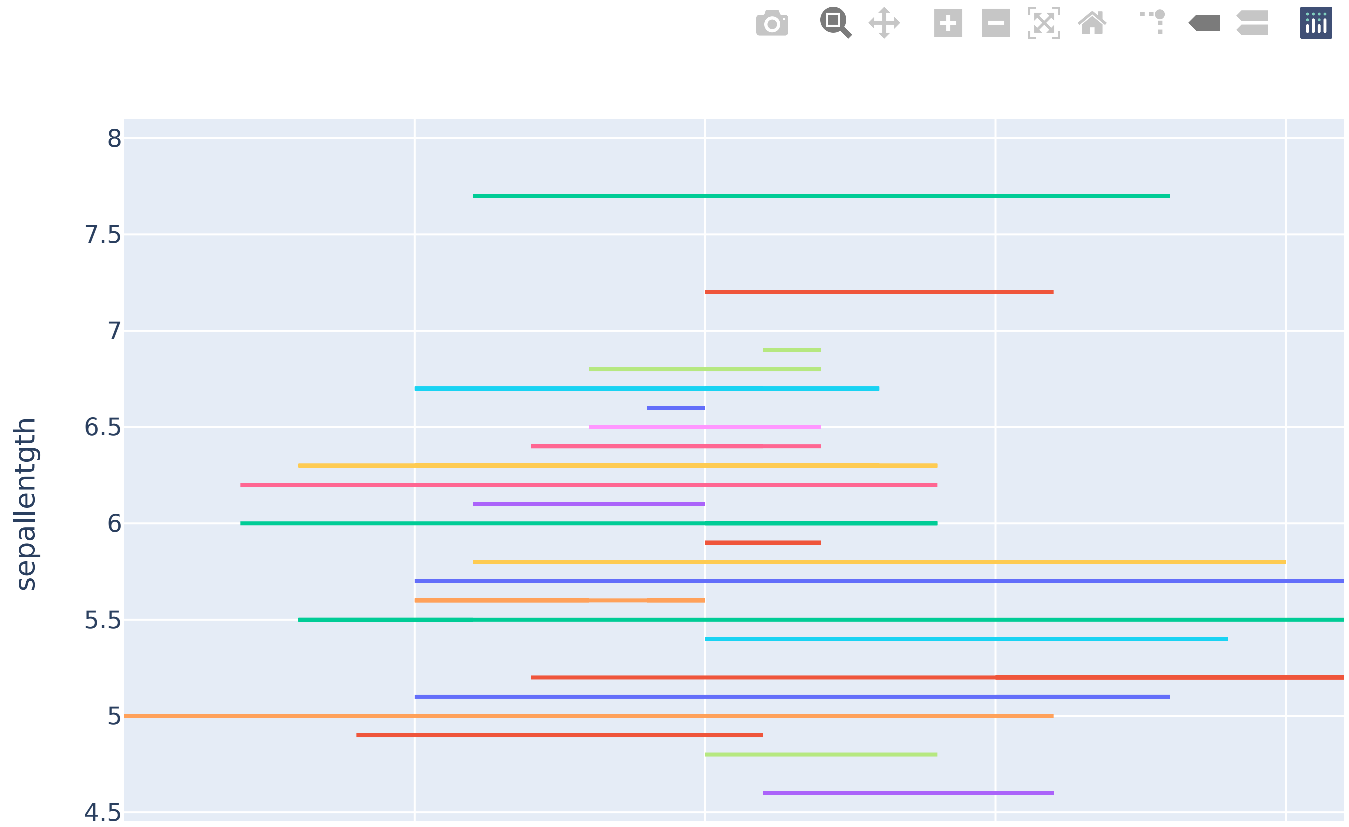
Line plot

In [7]:

```
import plotly.express as px
fig = px.line(df, x = "sepalwidth",
              y = "sepallentgth"
```

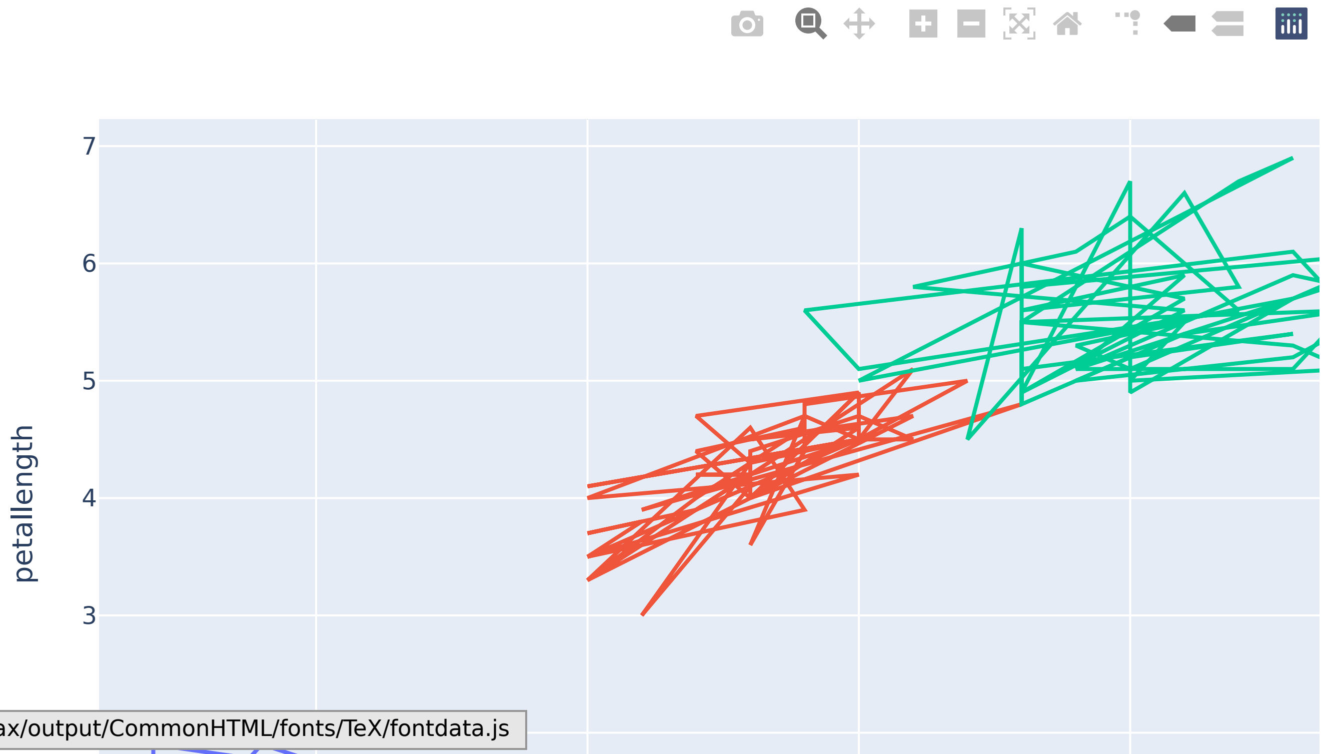
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

```
              color = "sepallentgth")
fig.show()
```

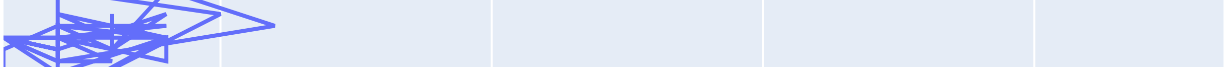


In [8]:

```
fig = px.line(df, x = "petalwidth",
              y = "petallength" ,
              color = "Species")
fig.show()
```



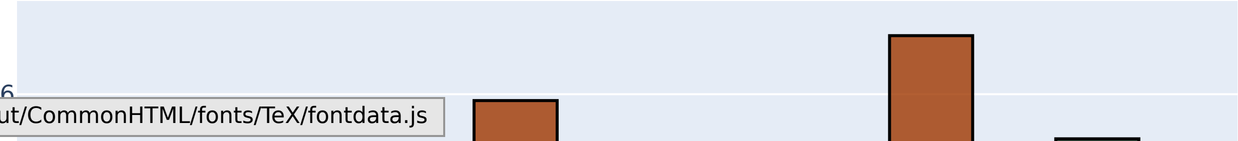
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js



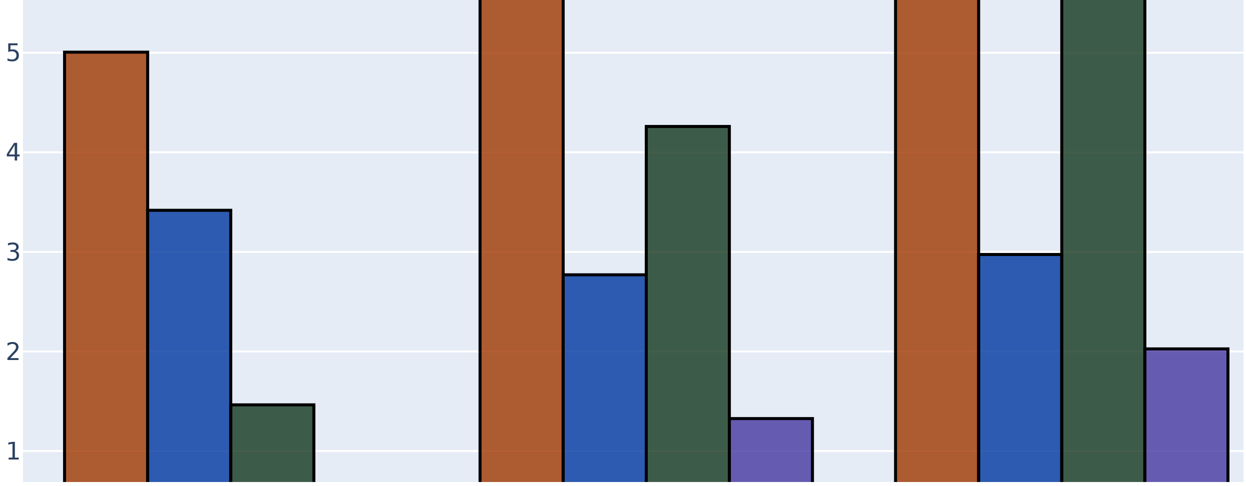
Bar graph

In [9]:

```
from plotly.offline import init_notebook_mode, iplot

d1 = df.groupby(df.Species).mean()
d1['Species'] = d1.index

t1 = go.Bar(
            x = d1.Species,
            y = d1.sepallentgth,
            name = "Sepal Length (cm)",
            marker = dict(color = 'rgba(160, 55, 0, 0.8)', line = dict(color = 'rgb(0,0,0
            text = d1.Species
)

t2 = go.Bar(
            x = d1.Species,
            y = d1.sepalwidth,
            name = "Sepal Width (cm)",
            marker = dict(color = 'rgba(0, 55, 160, 0.8)', line = dict(color = 'rgb(0,0,0
            text = d1.Species
)

t3 = go.Bar(
            x = d1.Species,
            y = d1.petallength,
            name = "Petal Length (cm)",
            marker = dict(color = 'rgba(20, 55, 30, 0.8)', line = dict(color = 'rgb(0,0,0
            text = d1.Species
)

t4 = go.Bar(
            x = d1.Species,
            y = d1.petalwidth,
            name = "Petal Width (cm)",
            marker = dict(color = 'rgba(70, 55, 160, 0.8)', line = dict(color = 'rgb(0,0,
            text = d1.Species
)

b = [t1,t2,t3,t4]
layout_bar = go.Layout(barmode = "group")
fig_bar = go.Figure(data = b, layout = layout_bar)
iplot(fig_bar)
```



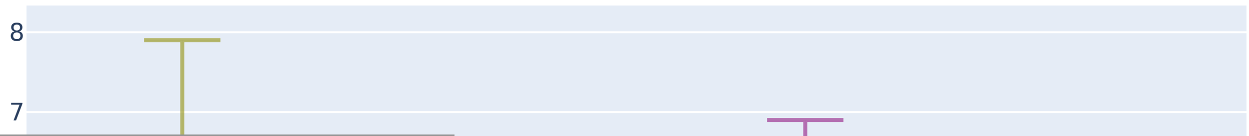
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js



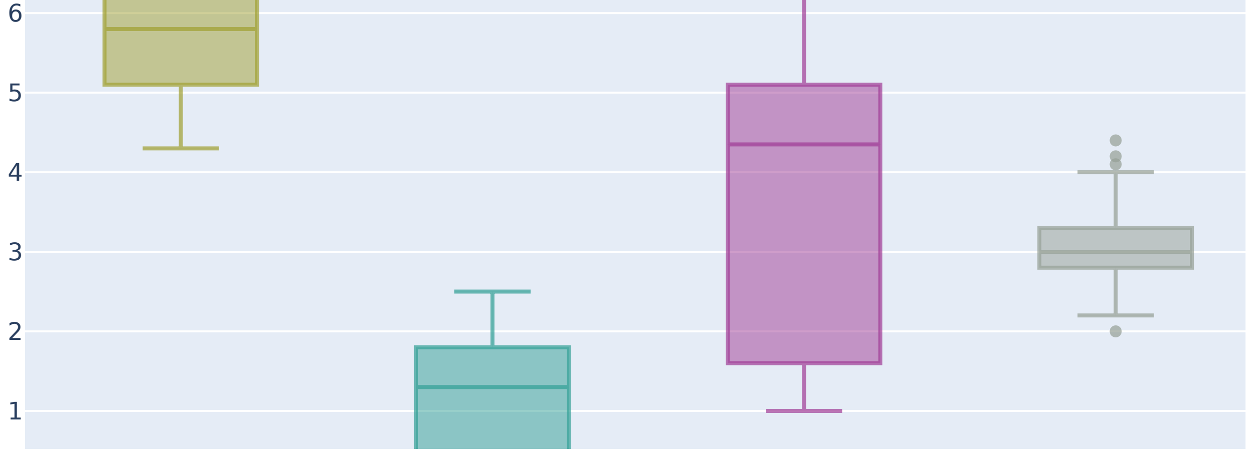
Box plot

In [10]:

```
t1_box = go.Box(
                name = 'Sepal Length (cm)',
                y = df.sepallentgth,
                marker = dict(color = 'rgba(160,160,50,0.7)')
)

t2_box = go.Box(
                name = 'Sepal Width (cm)',
                y = df.petalwidth,
                marker = dict(color = 'rgba(50,160,150,0.7)')
)

t3_box = go.Box(
                name = 'Petal Length (cm)',
                y = df.petallength,
                marker = dict(color = 'rgba(160,60,150,0.7)')
)

t4_box = go.Box(
                name = 'Petal Width (cm)',
                y = df.sepalwidth,
                marker = dict(color = 'rgba(150,160,150,0.7)')
)

fig_box = [t1_box, t2_box, t3_box, t4_box]

iplot(fig_box)
```



Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

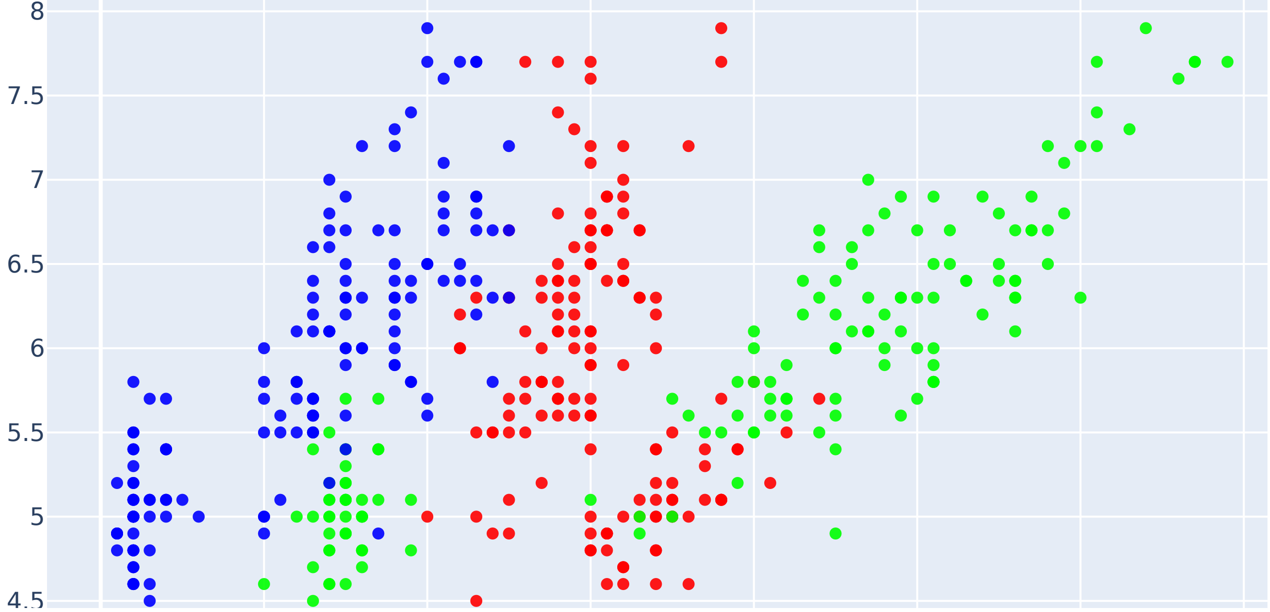


Scatter plot

In [11]:

```
S_SW = go.Scatter(
                    x = df.sepalwidth,
                    y = df.sepallentgth,
                    mode = "markers",
                    name = "Sepal Width (cm)",
                    marker = dict(color = 'rgba(255, 0, 0, 0.9)'),
                    text = df.Species
)

S_PL = go.Scatter(
                    x = df.petallength,
                    y = df.sepallentgth,
                    mode = "markers",
                    name = "Petal Length (cm)",
                    marker = dict(color = 'rgba(0, 255, 0, 0.9)'),
                    text = df.Species
)

S_PW = go.Scatter(
                    x = df.petalwidth,
                    y = df.sepallentgth,
                    mode = "markers",
                    name = "Petal Width (cm)",
                    marker = dict(color = 'rgba(0, 0, 255, 0.9)'),
                    text = df.Species
)

layout = dict(title = 'Change of Sepal Length by Other Properties',
              xaxis= dict(title= 'centimeters',ticklen= 5,zeroline= False)
             )
u = [S_SW, S_PL,S_PW]
fig = dict(data = u)
iplot(fig)
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

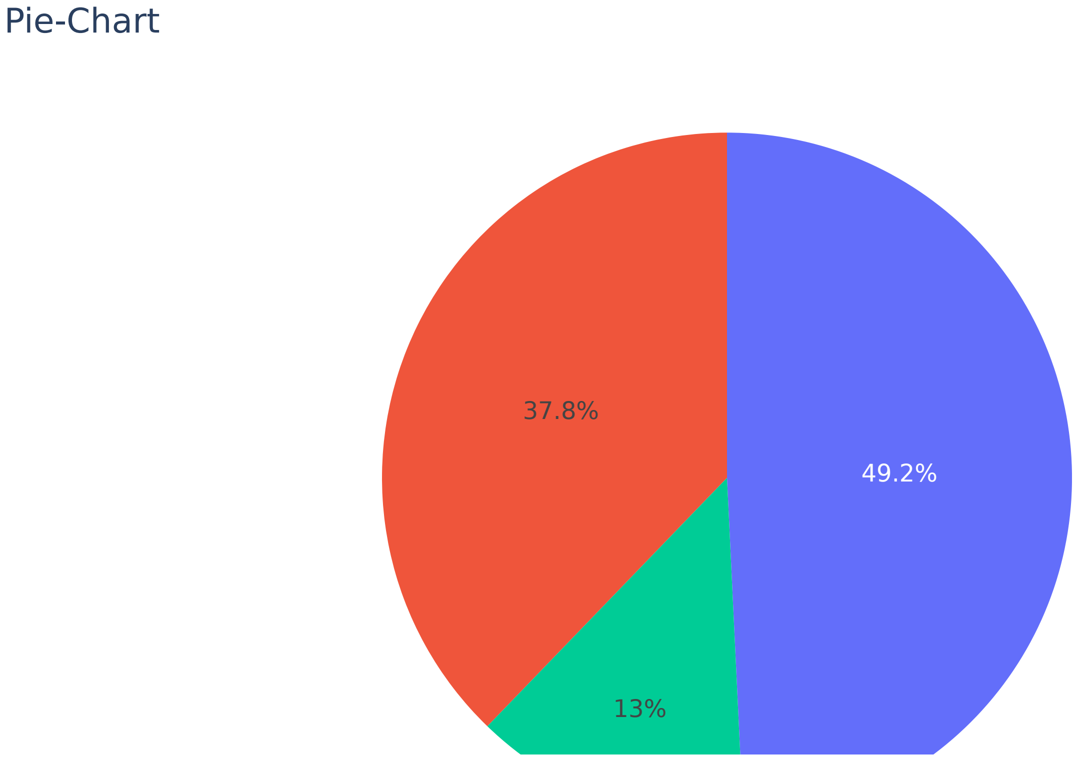


ML graph

Pie chart

In [12]:

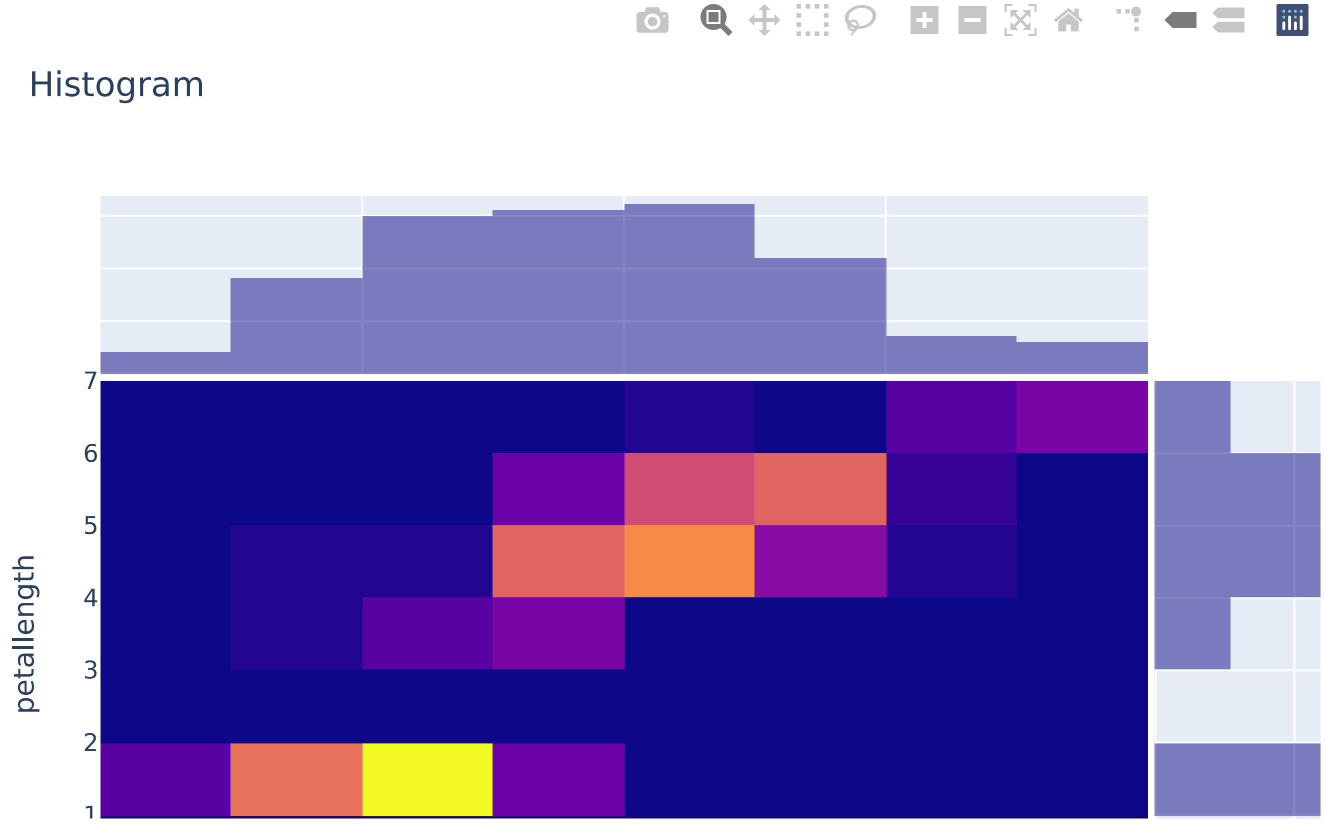
```
fig = px.pie(df, values='petallength', names='Species', title='Pie-Chart')
fig.show()
```



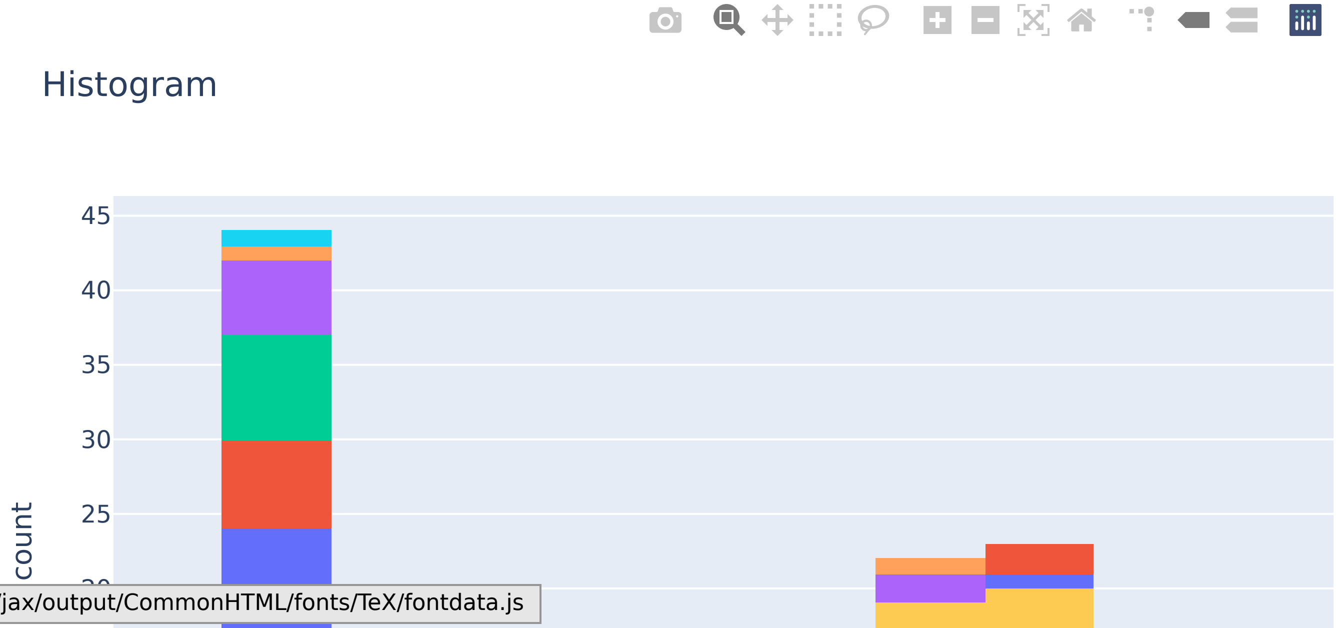
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

Histogram

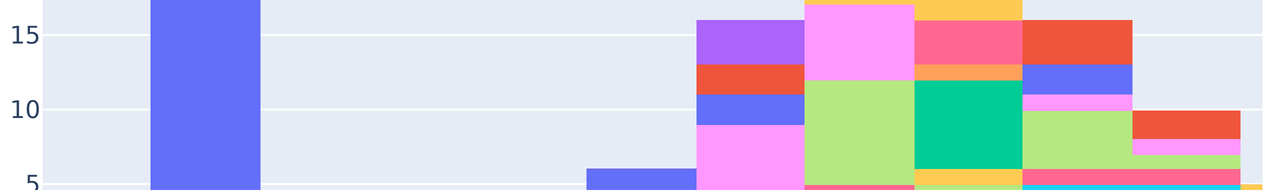
In [13]:
```
fig = px.density_heatmap(df, x="sepallentgth", y="petallength", marginal_x="histogram", ma
fig.show()
```



In [14]:
```
fig = px.histogram(df, x="petallength", color="petalwidth", title='Histogram')
fig.show()
```



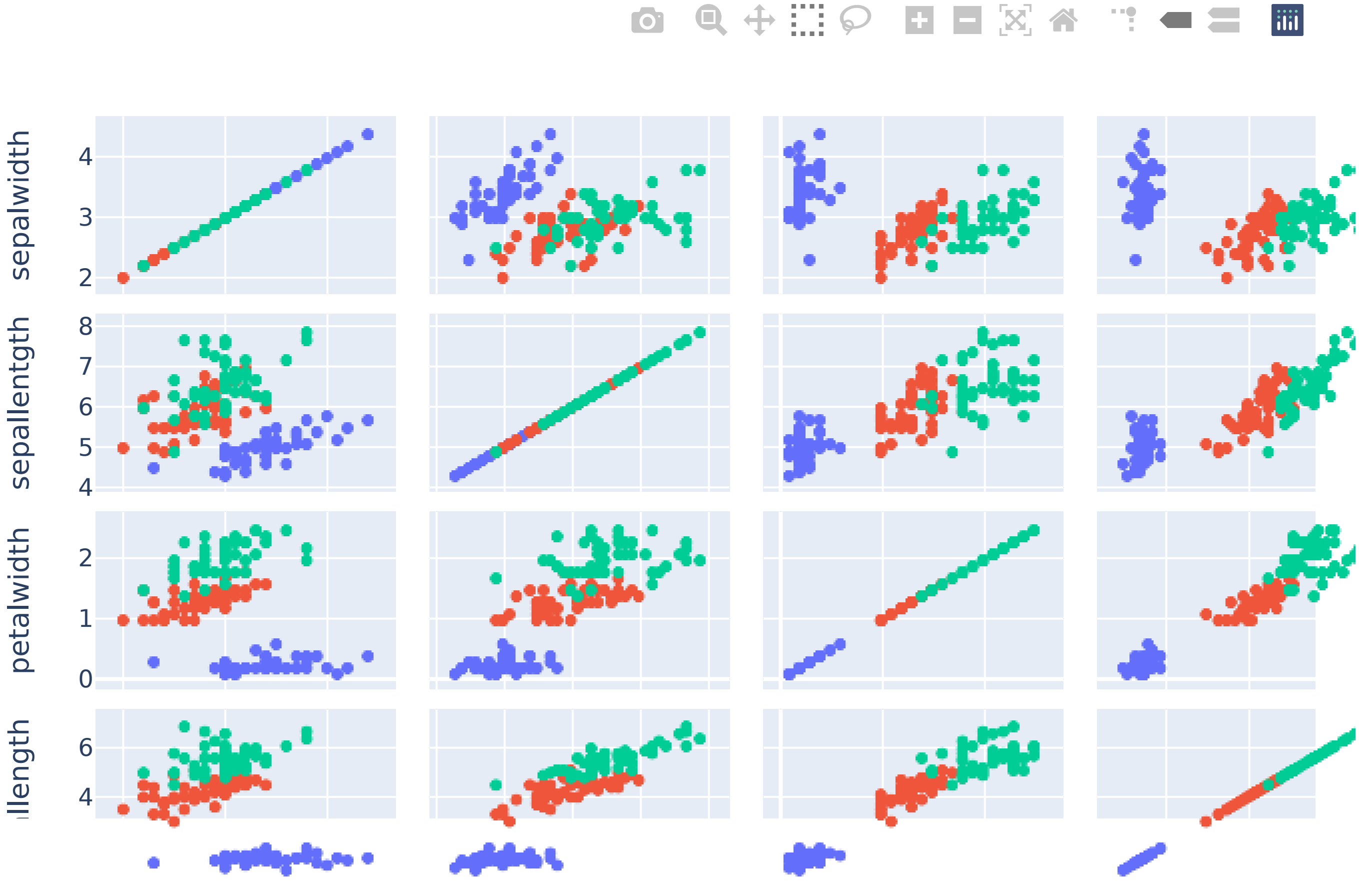
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js



Pair plot

In [15]:

```
fig = px.scatter_matrix(df,dimensions=["sepalwidth", "sepallentgth", "petalwidth", "petal
    color="Species")
fig.show()
```



In [ ]:

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js