

Agenda

- 1 Biblioteca Plotly
- 2 Exemplos
- 3 - Criando dashboard
- 4 - Exercícios

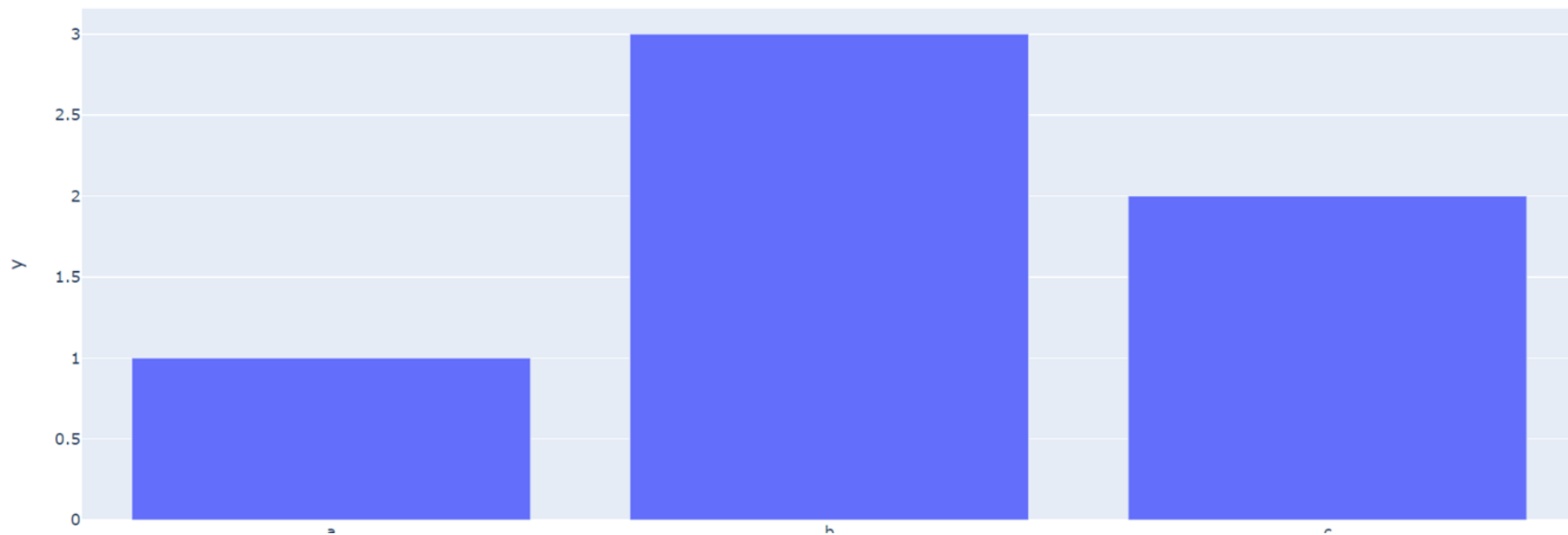
Biblioteca Plotly

Biblioteca utilizada para criar gráficos interativos
`pip install plotly`



Biblioteca Plotly - Gráfico de barras

```
1 import plotly.express as px
2 fig = px.bar(x=["a", "b", "c"], y=[1, 3, 2])
3 fig.show()
4
```

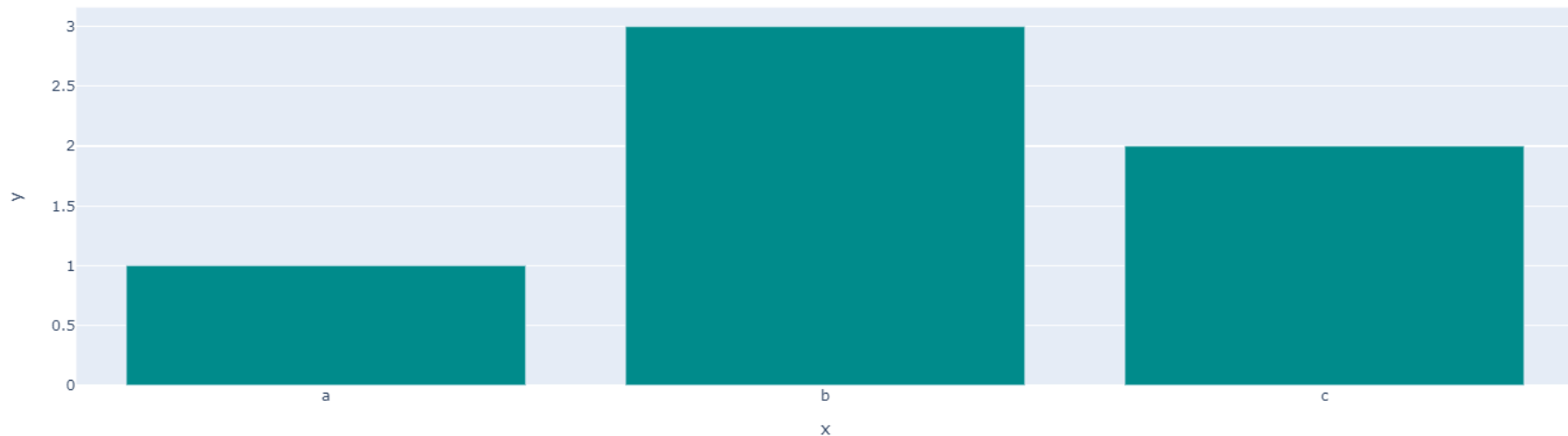


Biblioteca Plotly - Gráfico de barras

```
import plotly.express as px
fig = px.bar(x=["a", "b", "c"], y=[1, 3, 2])
fig.update_traces(marker_color='darkcyan')
fig.show()
```

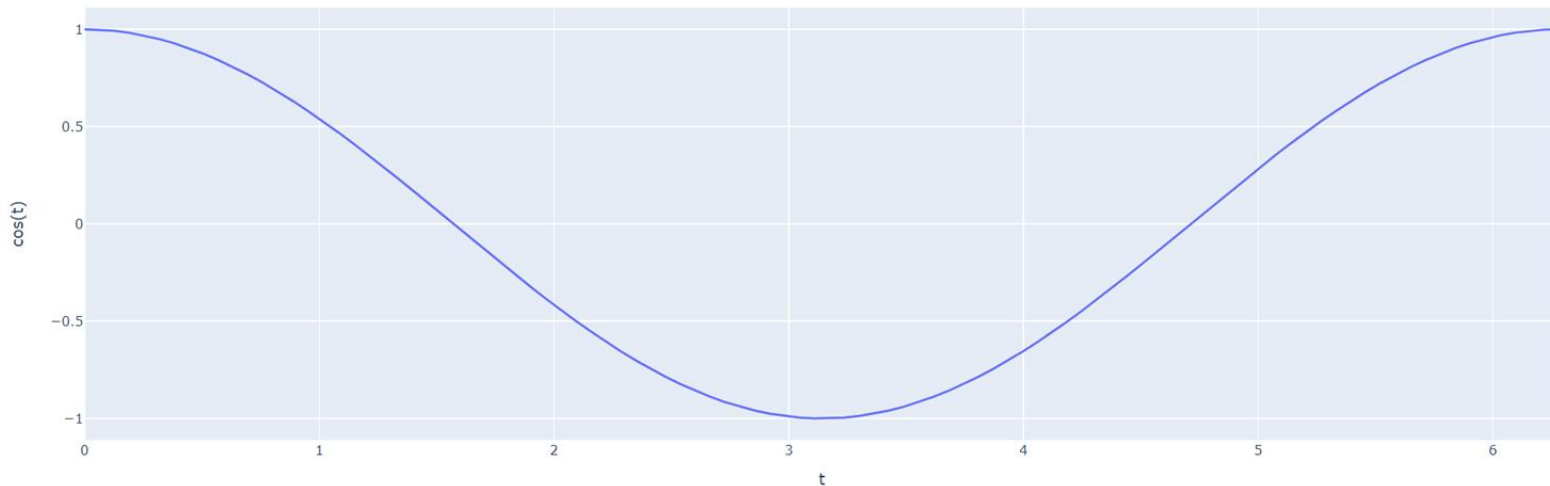
✓ 0.7s

Python



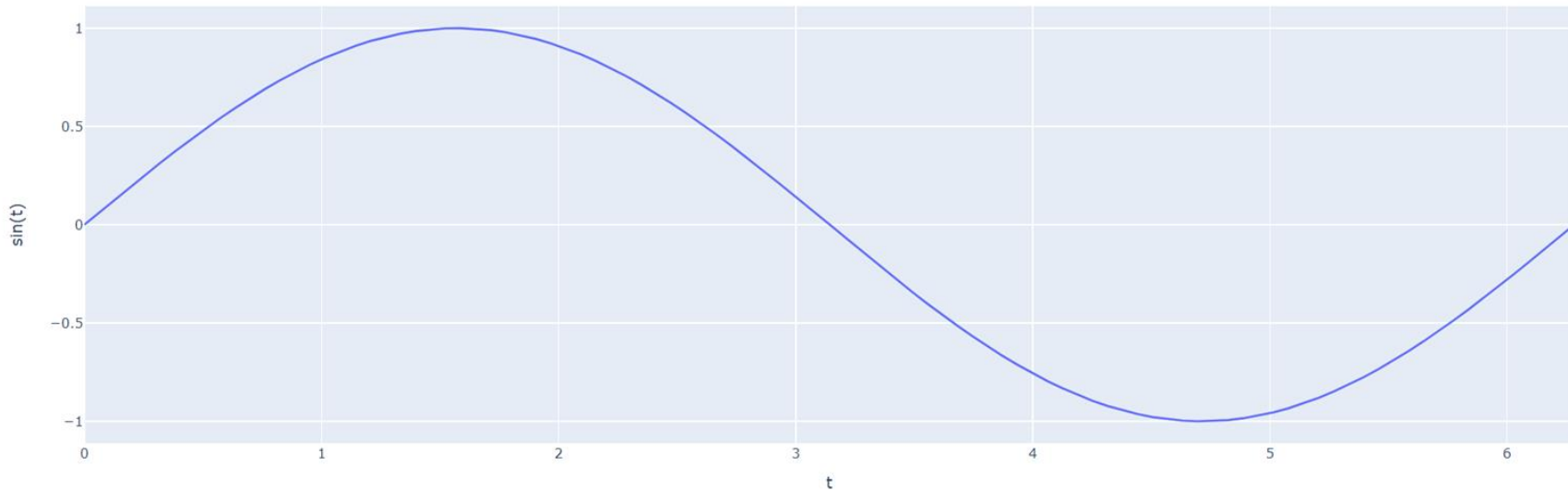
Biblioteca Plotly - Gráfico de linhas

```
[ ] 1 import plotly.express as px
    2 import numpy as np
    3
    4 t = np.linspace(0, 2*np.pi, 100)
    5
    6 fig = px.line(x=t, y=np.cos(t), labels={'x':'t', 'y':'cos(t)})
    7 fig.show()
    8
```



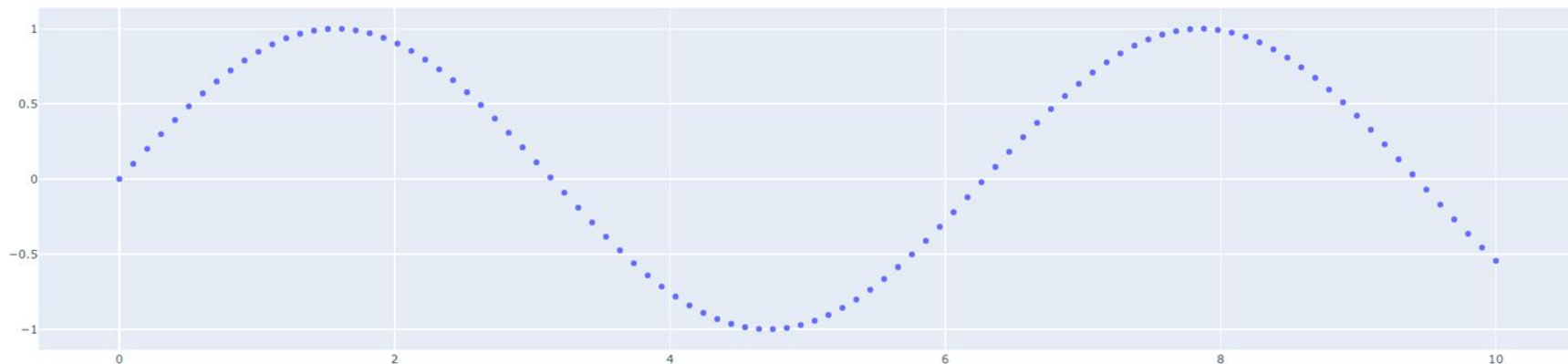
Biblioteca Plotly - Gráfico de linhas

```
1 import plotly.express as px
2 import numpy as np
3
4 t = np.linspace(0, 2*np.pi, 100)
5
6 fig = px.line(x=t, y=np.sin(t), labels={'x':'t', 'y':'sin(t)'})
7 fig.show()
```



Biblioteca Plotly - Gráfico de pontos

```
[2] 1 import plotly.graph_objects as go
    2 import numpy as np
    3
    4 N = 1000
    5 t = np.linspace(0, 10, 100)
    6 y = np.sin(t)
    7
    8 fig = go.Figure(data=go.Scatter(x=t, y=y, mode='markers'))
    9
   10 fig.show()
   11
```



Biblioteca Plotly - Gráfico múltiplos

```
import plotly.graph_objects as go

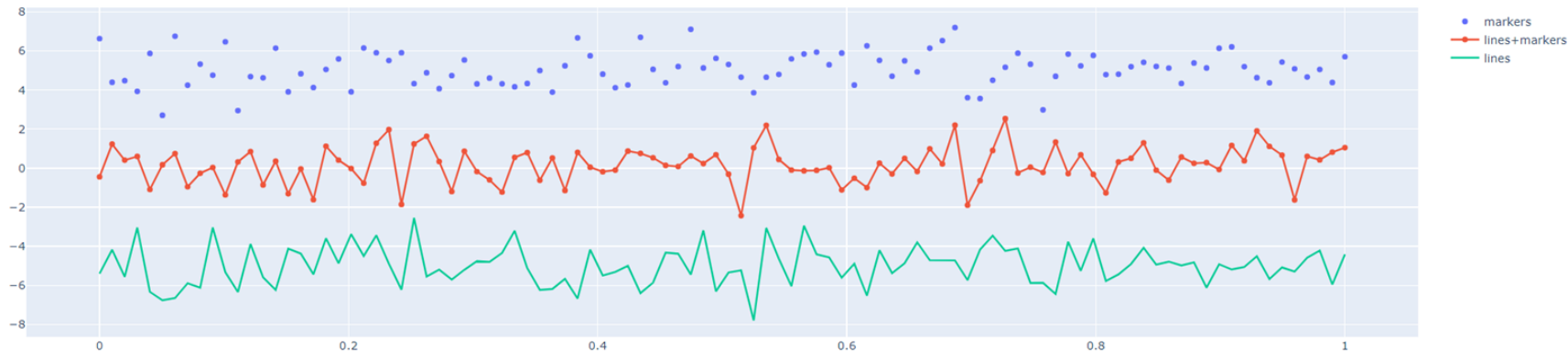
# Create random data with numpy
import numpy as np
np.random.seed(1)

N = 100
random_x = np.linspace(0, 1, N)
random_y0 = np.random.randn(N) + 5
random_y1 = np.random.randn(N)
random_y2 = np.random.randn(N) - 5

fig = go.Figure()
# Add traces
fig.add_trace(go.Scatter(x=random_x, y=random_y0,
                        mode='markers',
                        name='markers'))
```

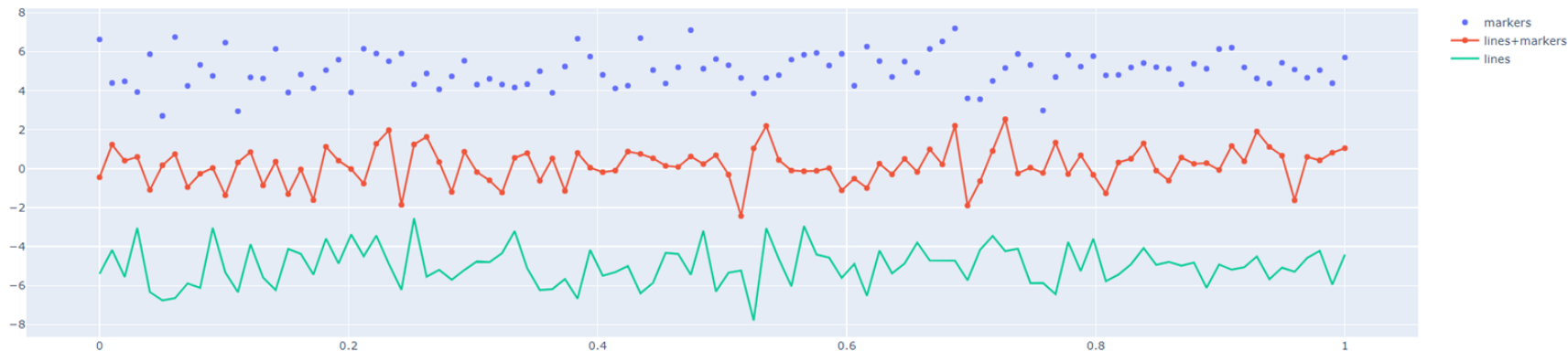

Biblioteca Plotly - Gráfico múltiplos

```
fig.add_trace(go.Scatter(x=random_x, y=random_y1,  
                        mode='lines+markers',  
                        name='lines+markers'))  
fig.add_trace(go.Scatter(x=random_x, y=random_y2,  
                        mode='lines',  
                        name='lines'))  
fig.write_html('Figura1.html')  
fig.show()
```



Biblioteca Plotly - Gráfico múltiplos

```
fig.add_trace(go.Scatter(x=random_x, y=random_y1,  
                        mode='lines+markers',  
                        name='lines+markers'))  
fig.add_trace(go.Scatter(x=random_x, y=random_y2,  
                        mode='lines',  
                        name='lines'))  
fig.write_html('Figura1.html')  
fig.show()
```



Biblioteca Plotly - Gráfico 3d

```
import numpy as np
import plotly.express as px
import plotly.graph_objects as go
from sklearn.svm import SVR

mesh_size = .02
margin = 0
df = px.data.iris()
X = df[['sepal_width', 'sepal_length']]
y = df['petal_width']
model = SVR(C=1.)
model.fit(X, y)
```

Biblioteca Plotly - Gráfico 3d

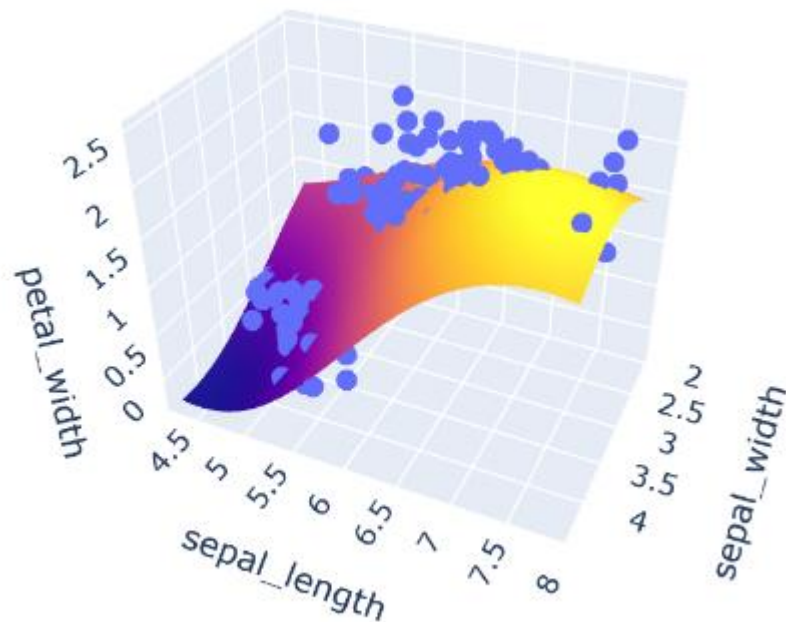
```
x_min, x_max = X.sepal_width.min() - margin, X.sepal_width.max() + margin  
y_min, y_max = X.sepal_length.min() - margin, X.sepal_length.max() + margin  
xrange = np.arange(x_min, x_max, mesh_size)  
yrange = np.arange(y_min, y_max, mesh_size)  
xx, yy = np.meshgrid(xrange, yrange)
```

Biblioteca Plotly - Gráfico 3d

```
pred = model.predict(np.c_[xx.ravel(), yy.ravel()])  
pred = pred.reshape(xx.shape)
```

```
fig = px.scatter_3d(df, x='sepal_width', y='sepal_length', z='petal_width')  
fig.update_traces(marker=dict(size=5))  
fig.add_traces(go.Surface(x=xrange, y=yrange, z=pred, name='pred_surface'))  
fig.show()
```

Biblioteca Plotly - Gráfico 3d



Biblioteca Plotly - Boxplot

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

```
x_data = ['Carmelo Anthony', 'Dwyane Wade',
          'Deron Williams', 'Brook Lopez',
          'Damian Lillard', 'David West',]
```

```
N = 50
```

```
y0 = (10 * np.random.randn(N) + 30).astype(np.int)
y1 = (13 * np.random.randn(N) + 38).astype(np.int)
y2 = (11 * np.random.randn(N) + 33).astype(np.int)
y3 = (9 * np.random.randn(N) + 36).astype(np.int)
y4 = (15 * np.random.randn(N) + 31).astype(np.int)
y5 = (12 * np.random.randn(N) + 40).astype(np.int)
```

```
y_data = [y0, y1, y2, y3, y4, y5]
colors = ['rgba(93, 164, 214, 0.5)', 'rgba(255, 144, 14, 0.5)', 'rgba(44, 160, 101, 0.5)',
          'rgba(255, 65, 54, 0.5)', 'rgba(207, 114, 255, 0.5)', 'rgba(127, 96, 0, 0.5)']
```

Biblioteca Plotly - Boxplot

```
fig = go.Figure()
for xd, yd, cls in zip(x_data, y_data, colors):
    fig.add_trace(go.Box(
        y=yd,
        name=xd,
        boxpoints='all',
        jitter=0.5,
        whiskerwidth=0.2,
        fillcolor=cls,
        marker_size=2,
        line_width=1)
    )
```

Biblioteca Plotly - Boxplot

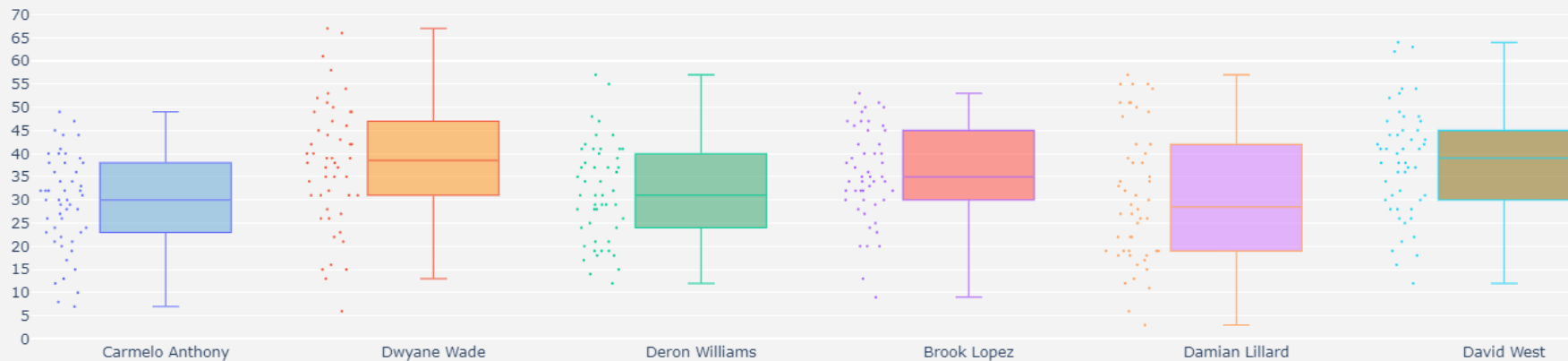
```
fig.update_layout(  
    title='Points Scored by the Top 9 Scoring NBA Players in 2012',  
    yaxis=dict(  
        autorange=True,  
        showgrid=True,  
        zeroline=True,  
        dtick=5,  
        gridcolor='rgb(255, 255, 255)',  
        gridwidth=1,  
        zerolinecolor='rgb(255, 255, 255)',  
        zerolinewidth=2,  
    ),
```

Biblioteca Plotly - Boxplot

```
margin=dict(  
    l=40,  
    r=30,  
    b=80,  
    t=100,  
),  
paper_bgcolor='rgb(243, 243, 243)',  
plot_bgcolor='rgb(243, 243, 243)',  
showlegend=False  
)  
  
fig.show()
```

Biblioteca Plotly - Boxplot

Points Scored by the Top 9 Scoring NBA Players in 2012



Biblioteca Plotly - Boxplot

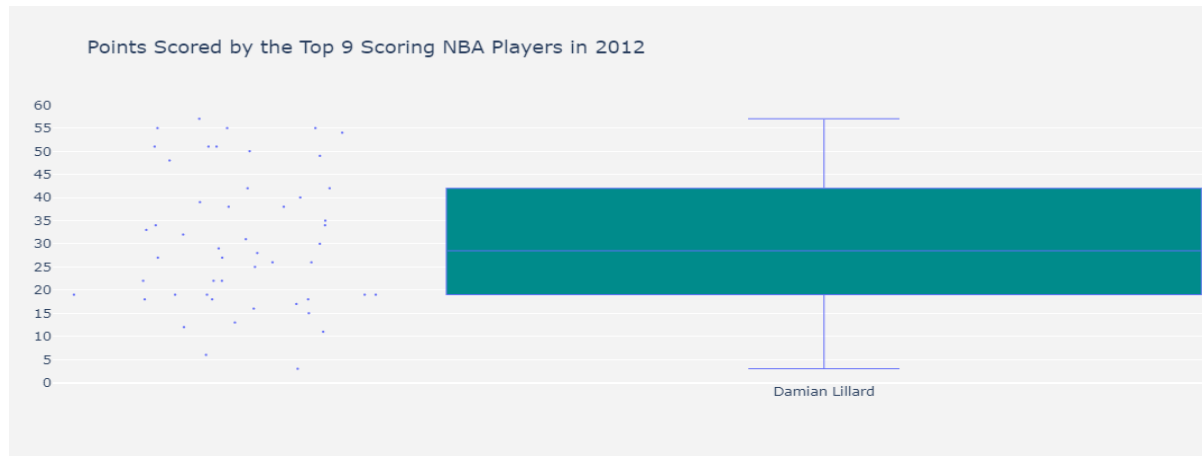
```
fig = go.Figure()  
fig.add_trace(go.Box(  
    y=y4,  
    name=x_data[4],  
    boxpoints='all',  
    jitter=0.5,  
    whiskerwidth=0.2,  
    fillcolor=cls,  
    marker_size=2,  
    line_width=1)  
)
```

Biblioteca Plotly - Boxplot

```
fig.update_layout(  
    title='Points Scored by the Top 9 Scoring NBA Players in  
2012',  
    yaxis=dict(  
        autorange=True,  
        showgrid=True,  
        zeroline=True,  
        dtick=5,  
        gridcolor='rgb(255, 255, 255)',  
        gridwidth=1,  
        zerolinecolor='rgb(255, 255, 255)',  
        zerolinewidth=2,  
    ),
```

Biblioteca Plotly - Boxplot

```
margin=dict(  
    l=40,  
    r=30,  
    b=80,  
    t=100,  
),  
paper_bgcolor='rgb(243, 243, 243)',  
plot_bgcolor='rgb(243, 243, 243)',  
showlegend=False  
)  
  
fig.show()
```



Biblioteca Plotly

Utilizando o Dataset monitoramento tempo plotar o gráfico da temperatura utilizando a biblioteca Plotly

```
import pandas as pd
```

```
import plotly.express as px
```

```
import datetime
```

```
df = pd.read_csv('Curso_Data_Science_Alura\monitoramento_tempo.csv', sep = ',')
```

```
df.head()
```

Biblioteca Plotly

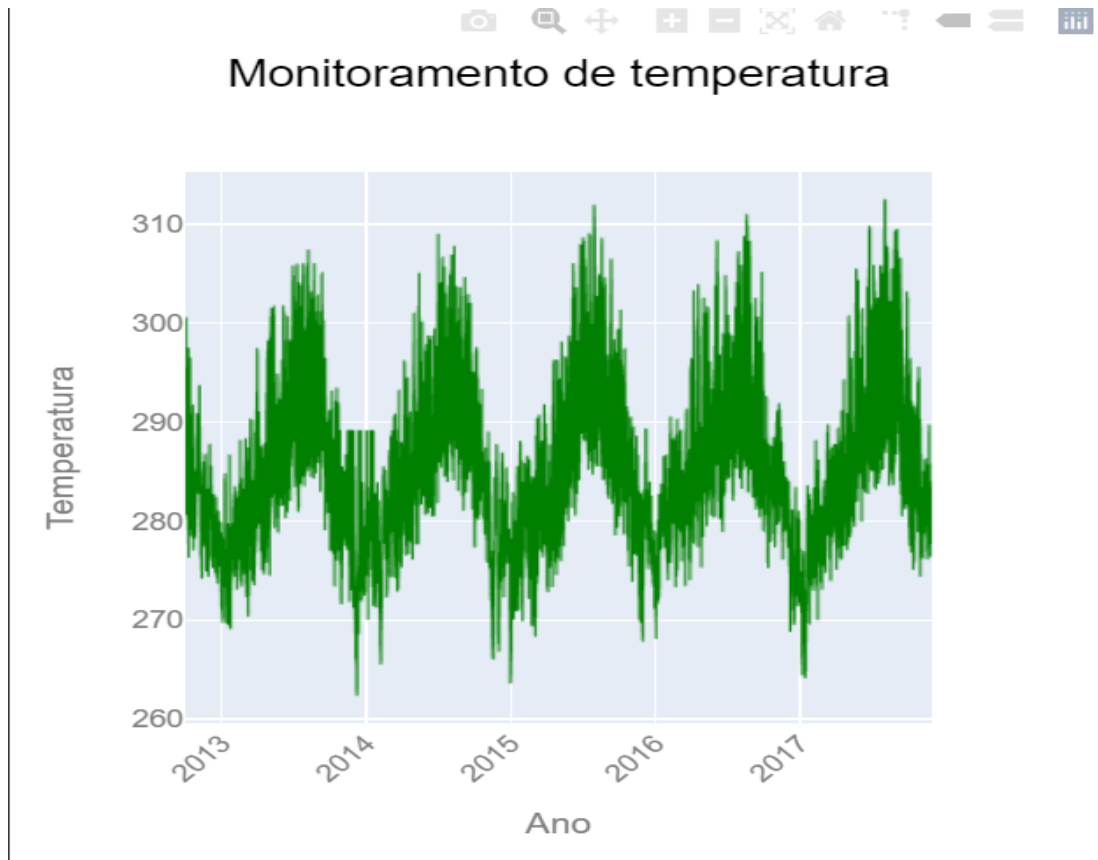
df.info()

```
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 45253 entries, 0 to 45252  
Data columns (total 7 columns):  
#   Column                Non-Null Count  Dtype  
---  ---  
0   temperatura           45253 non-null  float64  
1   pressão               45253 non-null  float64  
2   humidade              45253 non-null  float64  
3   direção do vento      45253 non-null  float64  
4   velocidade do vento   45253 non-null  float64  
5   dia_da_semana         45253 non-null  object  
6   data                  45253 non-null  object  
dtypes: float64(5), object(2)  
memory usage: 2.4+ MB
```

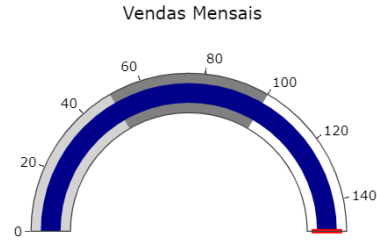
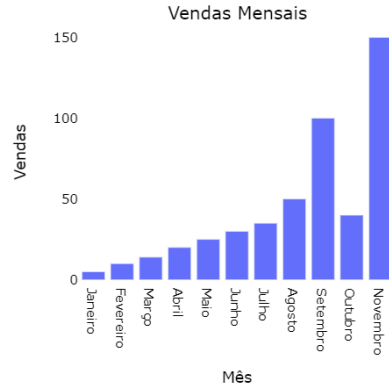
Biblioteca Plotly

```
df['data']=pd.to_datetime(df['data'])  
fig = px.line(df,x=df['data'],y=df['temperatura'],  
title= 'Monitoramento de temperatura')  
fig.update_layout(width = 500, height= 500,xaxis={'tickangle':-45},title_x=0.5,  
font_family= 'Arial', font_size= 14, font_color = 'grey',title_font_color = 'black',  
title_font_size = 22,  
xaxis_title='Ano',yaxis_title='Temperatura')  
  
fig.update_traces(line_color='green', line_width=1)  
fig.show()  
fig.write_html('Temperatura.html')
```

Biblioteca Plotly



Dashboard com Plotly



Mês	Vendas
1	5
2	10
3	14
4	20
5	25
6	30
7	35
8	50
9	100
10	40

Criar Dashboard com Plotly

```
import plotly.subplots as sp
import plotly.graph_objects as go
import pandas as pd
```

```
# Dados de vendas
vendas = [5, 10, 14, 20, 25, 30, 35, 50, 100, 40, 150]
meses = ['Janeiro', 'Fevereiro', 'Março', 'Abril', 'Maio', 'Junho', 'Julho', 'Agosto',
'Setembro', 'Outubro', 'Novembro']
```

```
# Mapear os nomes dos meses para números de mês
meses_numeros = {'Janeiro': 1, 'Fevereiro': 2, 'Março': 3, 'Abril': 4, 'Maio': 5,
'Junho': 6, 'Julho': 7, 'Agosto': 8, 'Setembro': 9, 'Outubro': 10, 'Novembro': 11}
```

```
# Criar dataframe para os dados
df = pd.DataFrame({'Mês': meses, 'Vendas': vendas})
```

Criar Dashboard com Plotly

Criar subplots

```
fig = sp.make_subplots(rows=2, cols=2,  
                        subplot_titles=('Vendas Mensais', ' Vendas Mensais'),  
                        specs=[[{'type': 'bar'}, {'type': 'indicator'}],  
                              [{'type': 'table', 'colspan': 2}, None]])
```

Criar Dashboard com Plotly

Gráfico de barras

```
fig.add_trace(go.Bar(x=df['Mês'], y=df['Vendas']), row=1, col=1)
fig.update_xaxes(title_text='Mês', row=1, col=1)
fig.update_yaxes(title_text='Vendas', row=1, col=1)
```

Gráfico de gauge

```
fig.add_trace(go.Indicator(
    mode='gauge',
    value=df['Vendas'][10],
    domain={'x': [0, 0.4], 'y': [0.4, 0.5]},
    title={'text': ""},
    gauge={'axis': {'range': [None, max(vendas)]},
          'bar': {'color': 'darkblue'},
          'steps': [
              {'range': [0, 50], 'color': 'lightgray'},
              {'range': [50, 100], 'color': 'gray'}],
          'threshold': {'line': {'color': 'red', 'width': 4}, 'thickness': 0.75, 'value':
max(vendas)}}), row=1, col=2)
```


Criar Dashboard com Plotly

```
# Tabela de vendas mensais
df_table = df.copy()
df_table['Mês'] = df_table['Mês'].map(meses_numeros)
df_table = df_table.sort_values('Mês')
df_table = df_table.reset_index(drop=True)

fig.add_trace(go.Table(
    header=dict(values=['Mês', 'Vendas'],
                  fill_color='paleturquoise',
                  align='left'),
    cells=dict(values=[df_table['Mês'], df_table['Vendas']],
               fill_color='lavender',
               align='left')), row=2, col=1)

fig.update_layout(height=800, width=800, showlegend=False,
                  plot_bgcolor='white',
                  paper_bgcolor='white',
                  font=dict(color='black'))

fig.write_html('Dashteste1')
fig.show()
```

Exercício 1

Criar uma dashboard com os seguintes dados:

Nº de produtos fabricados = 10,50,60,30,100,35,70

Setor = 1,2,3,4,5,6,7



Exercício 1

```
from plotly.validators import SankeyValidator
import plotly.subplots as sp
import plotly.graph_objects as go
import pandas as pd
# Dados de vendas
prod = [10,50,60,30,100,35,70]
setor = ['Setor 1','Setor 2','Setor 3','Setor 4','Setor 5','Setor 6','Setor 7']
# Mapear os nomes dos meses para números de mês
setor_numeros = {'Setor 1': 1, 'Setor 2': 2, 'Setor 3': 3, 'Setor 4': 4, 'Setor 5': 5, 'Setor 6': 6,
'Setor 7': 7, }
# Criar dataframe para os dados
df = pd.DataFrame({'Setor': setor, 'Produtos': prod})
```

Exercício 1

Criar subplots

```
fig = sp.make_subplots(rows=2, cols=2,  
subplot_titles=('Produtos', 'Produtos'),  
specs=[[{'type': 'bar'}, {'type': 'indicator'}],  
[{'type': 'table', 'colspan': 2}, None]])
```

Gráfico de barras

```
fig.add_trace(go.Bar(x=df['Setor'], y=df['Produtos']), row=1, col=1)
```

```
fig.update_xaxes(title_text='Setor', row=1, col=1)
```

```
fig.update_yaxes(title_text='Produtos', row=1, col=1)
```

Gráfico de gauge

```
fig.add_trace(go.Indicator(  
mode='gauge',  
value=df['Produtos'][6],
```

Exercício 1

```
domain={'x': [0, 0.4], 'y': [0.4, 0.5]},  
title={'text': ""},  
gauge={'axis': {'range': [None, max(prod)]},  
      'bar': {'color': 'darkblue'},  
      'steps': [  
        {'range': [0, 50], 'color': 'lightgray'},  
        {'range': [50, 100], 'color': 'gray'}],  
      'threshold': {'line': {'color': 'red', 'width': 4}, 'thickness': 0.75, 'value': max(prod)}}), row=1, col=2)
```

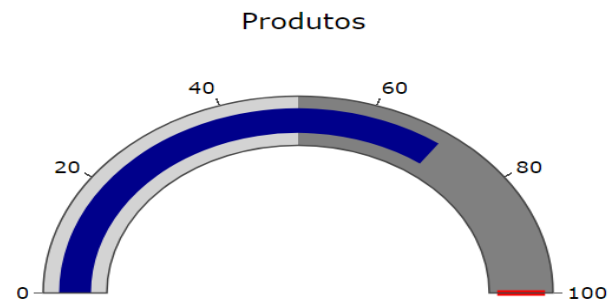
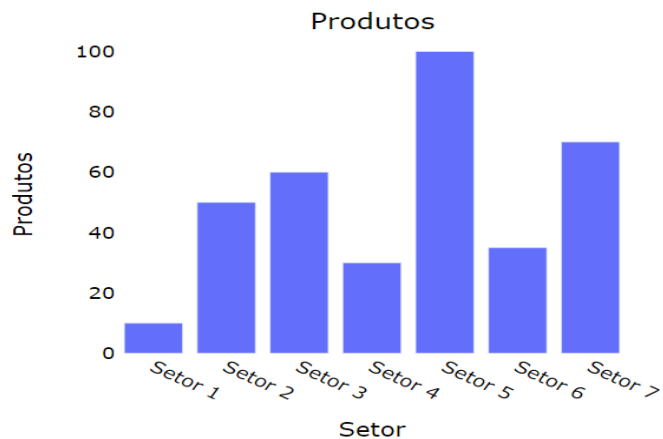
Exercício 1

```
# Tabela de vendas mensais
df_table = df.copy()
df_table['Setor'] = df_table['Setor'].map(setor_numeros)
df_table = df_table.sort_values('Setor')
df_table = df_table.reset_index(drop=True)
fig.add_trace(go.Table(
    header=dict(values=['Setor', 'Produtos'],
    fill_color='paleturquoise',
    align='left'),
    cells=dict(values=[df_table['Setor'], df_table['Produtos']],
    fill_color='lavender',
```

Exercício 1

```
align='left')), row=2, col=1)
fig.update_layout(height=800, width=800, showlegend=False,
plot_bgcolor='white',
paper_bgcolor='white',
font=dict(color='black'))
fig.write_html('Dashteste1')
fig.show()
```

Exercício 1



Setor	Produtos
1	10
2	50
3	60
4	30
5	100
6	35
7	70

Exercícios

1) Criar uma dashboard com os seguintes dados:

Entregadores= 12,4,6,8,9,28

Número de cargas entregues = 50,25,100,125,200,150

2) Utilizando a biblioteca Plotly plotar gráficos de linha da umidade, direção do vento.

Obrigado!

Prof. Me Daniel Vieira

Email: danielvieira2006@gmail.com

Linkedin: Daniel Vieira

Instagram: Prof daniel.vieira95

