

Aula 04 - Minicurso Python

Property Ítalo Lima

Introdução

Na última aula, nos aprofundamos nas descrições estatísticas e em algumas de suas medidas e plotamos diferentes tipos de gráficos, gráficos estes que são apresentados nos capítulos iniciais da disciplina de probabilidade e estatística, além disso, utilizando a biblioteca Matplotlib introduzimos algumas formas e métodos para uma configuração que possibilite uma melhor visualização dos dados. Nesta aula, nos aprofundaremos na biblioteca Matplotlib, visualizaremos novos tipos de gráficos, como o Scatter Plot e o de Pizza, verificaremos novos parâmetros e métodos para a configuração dos mesmos, e por fim, será introduzida uma biblioteca bastante utilizada, a **Seaborn**.



<https://seaborn.pydata.org/>

A Seaborn é uma biblioteca para visualização de dados e geração de gráficos, é baseada na Matplotlib e provê uma interface de alto nível para a configuração dos gráficos estatísticos gerados.



Para importar a biblioteca no nosso Notebook, adicionaremos essa linha:

```
import seaborn
```

Prática

Importando as Bibliotecas e os Dados

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

```
#Carregando dataset da biblioteca Seaborn
tips = sns.load_dataset('tips')
```

Estes dados representam uma amostra de dados coletados por um restaurante e é um dataset utilizado para estudos iniciais na biblioteca Seaborn.

Explorando os Dados

```
#Visualizar as 5 primeiras linhas do meu Dataset
tips.head()
```

Descrição dos Dados

Este é um dataset que a própria biblioteca fornece, o mesmo contém alguns registros de clientes de um restaurante.

Total_bill: Custo total da refeição em dólares.

Tip: Valor da gorjeta em dólares.

Sex: Sexo (Feminino / Masculino).

Smoker: Se a área escolhida era para fumantes ou não.

Day: Dia da semana.

Time: Almoço (Lunch) ou jantar (Dinner).

Size: Tamanho da refeição.

```
#Verificar a estrutura do DataFrame
tips.shape
```

```
#Verificando a descrição estatística dos dados, com 2 casas decimais
tips.describe().round(2)
```



Muitas vezes os dados são muito "quebrados", com várias casas decimais, por isso, a nível de visualização é interessante delimitar a forma de exibição.

```
#Verificar o número (absoluto) de Homens e Mulheres na Amostra
tips['sex'].value_counts()
```

```
#Verificar a porcentagem de Homens e Mulheres na Amostra
tips['sex'].value_counts(normalize=True)
```

Visualizando e Configurando Gráficos

Gráfico de Barras

```
#Aumentar o tamanho da figura
plt.figure(figsize=(10, 6))

#Setar uma cor para cada coluna exibida
tips['sex'].value_counts().plot(kind='bar', color='pink')
```

```
#Definir um título para o Gráfico
plt.title('Gráfico de Barras para os Clientes de um Restaurante')

#Nomear os eixos do Gráfico
plt.xlabel('Gêneros')
plt.ylabel('Frequência')

#Exibir o Gráfico
plt.show()
```



Podemos setar uma cor para cada barra do gráfico, de forma simples. E rotacionar os ticks para cada uma delas.

```
plt.figure(figsize=(10, 6))

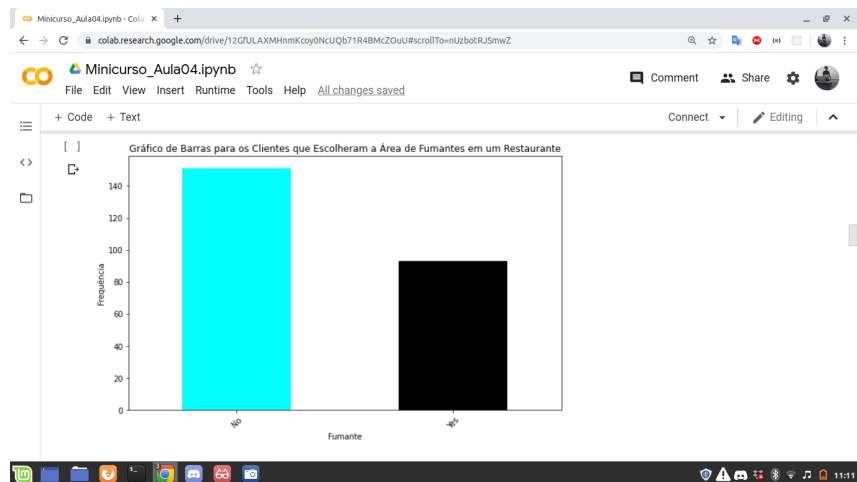
#Ao invés de passar apenas uma cor, passamos uma lista com as cores.
tips['smoker'].value_counts().plot(kind='bar', color=['cyan', 'black'])

plt.title('Gráfico de Barras para os Clientes que Escolheram a Área de Fumantes em um Restaurante')

plt.xlabel('Fumante')
plt.ylabel('Frequência')

#Rotacionar os ticks. É passado como valor para a propriedade rotation a quantidade de graus de rotação.
plt.xticks(rotation=45)

plt.show()
```



Distribuição de Frequências



Como os nossos dados são coletados de um dataset do exterior, a prática de gorjeta é diferente da do Brasil (Porcentagem fixa em cima do valor gasto), em outros países, geralmente esse valor é livre e variável, de acordo com os clientes.



No entanto, podemos verificar as distribuições dos valores gastos e das gorjetas, e então observar algum comportamento relativo.

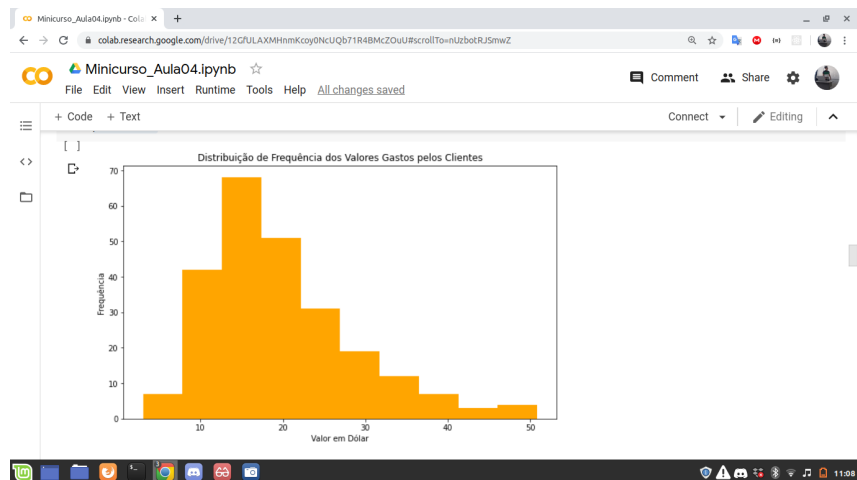
```
plt.figure(figsize=(10, 6))

tips['total_bill'].plot(kind='hist', color='orange')

plt.title('Distribuição de Frequência dos Valores Gastos pelos Clientes')

plt.xlabel('Valor em Dólar')
plt.ylabel('Frequência')

plt.show()
```



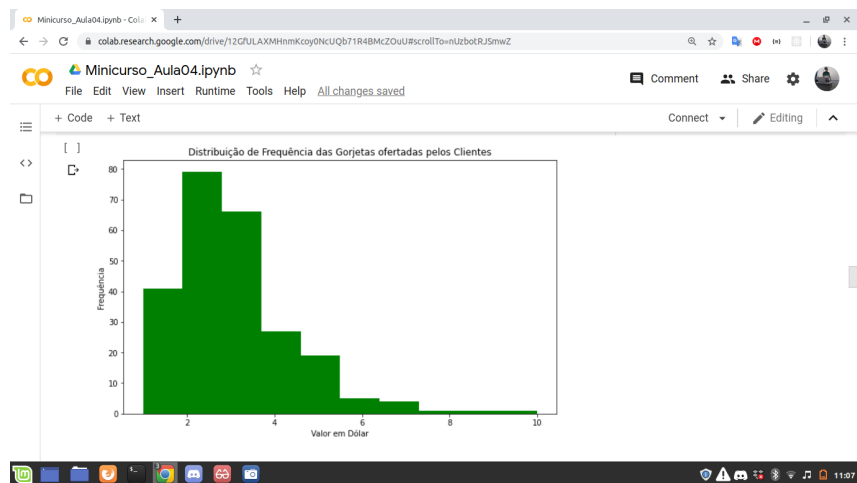
```
plt.figure(figsize=(10, 6))

tips['tip'].plot(kind='hist', color='green')

plt.title('Distribuição de Frequência das Gorjetas ofertadas pelos Clientes')

plt.xlabel('Valor em Dólar')
plt.ylabel('Frequência')

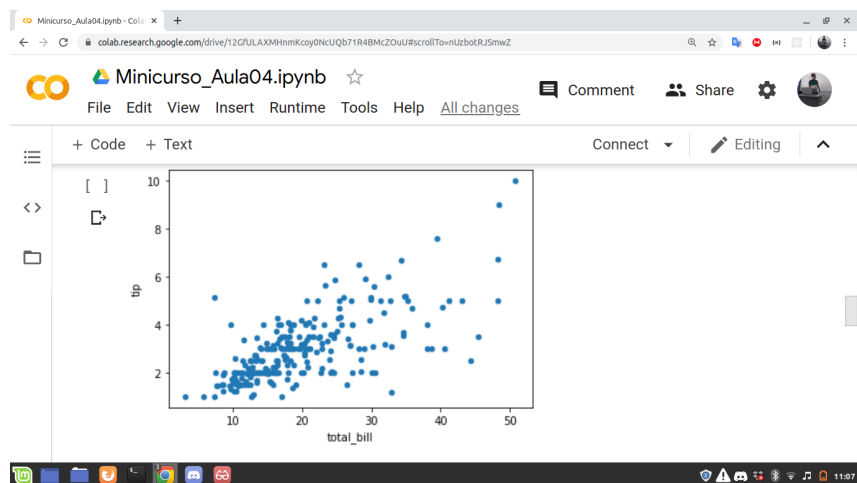
plt.show()
```



Scatter Plot

O Scatter é também conhecido como gráfico de dispersão, e apresenta o comportamento dos dados a partir de duas variáveis selecionadas. Dessa forma, podemos dá continuidade a observação da relação citada acima.

```
#Plotando um gráfico do tipo Scatter e definindo as variáveis do eixo X e Y.
tips.plot(x='total_bill', y='tip', kind='scatter')
```



Alterando as Propriedades do Scatter Plot

Por padrão, a cor vem azul e o marcador vem como um ponto com tamanho pré definido. Podemos alterar isto.

```
#Sobrescrevendo as propriedades color, marker e size.
tips.plot(x='total_bill', y='tip', kind='scatter', color='cyan', marker='h', s=130);
```



O marcador virou um hexágono, a cor foi alterada para Ciano e o tamanho dos marcadores foi aumentado.

Gráfico de Pizza

```
#Filtando os fumantes.
fumantes = tips.loc[tips.smoker == 'Yes']
fumantes = fumantes['smoker'].count()
fumantes
```

```
#Filtando os não fumantes.
nao_fumantes = tips.loc[tips.smoker == 'No']
nao_fumantes = nao_fumantes['smoker'].count()
nao_fumantes
```

```
#Labels e Valores correspondentes
labels = ['Fumante', 'Não Fumante']
sizes = [fumantes, nao_fumantes]
```

```
#Criar a Área da Figura
fig1, ax1 = plt.subplots()
```

```
#Definir Função de Explosão, no Gráfico
explode = (0.1, 0)
```

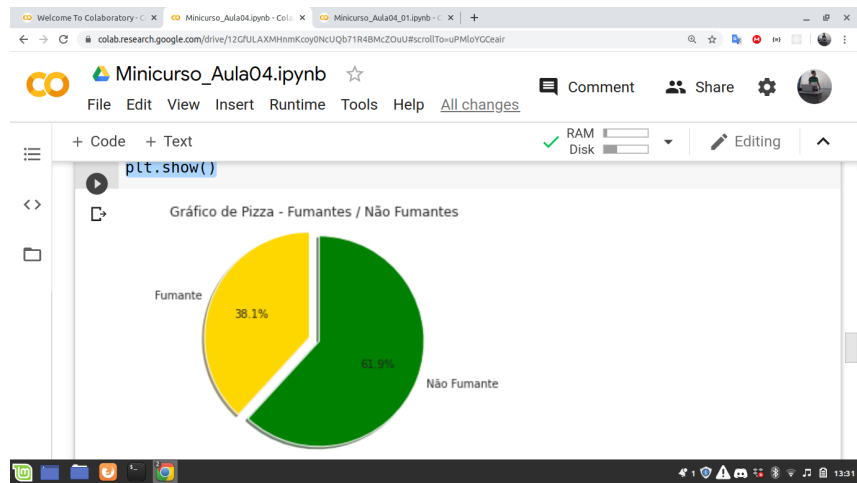
```
#Criar o Gráfico de Pizza
ax1.pie(sizes, labels=labels, explode=explode, autopct='%1.1f%%', shadow=True, startangle=90, colors=['gold', 'green'])
```

```
plt.title('Gráfico de Pizza - Fumantes / Não Fumantes')
```

```
#Deixando o Gráfico Redondo
```

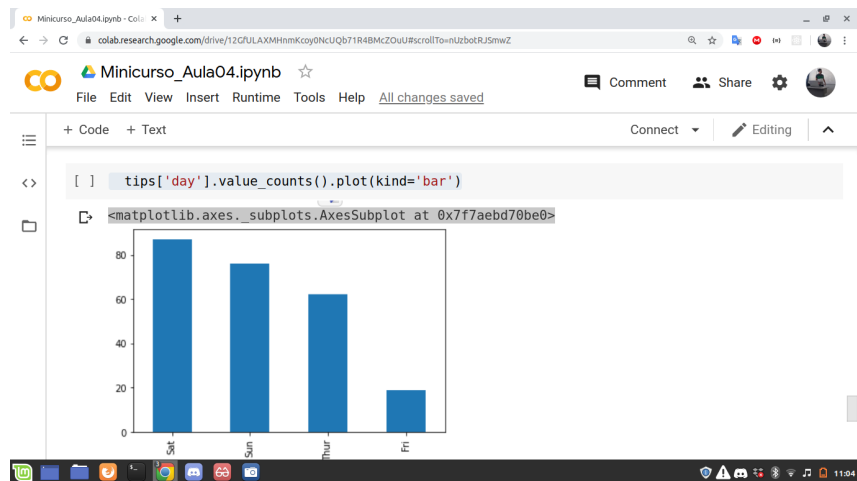
```
ax1.axis('equal')

plt.show()
```



Introdução a Seaborn

```
#Gerando um gráfico de Barras com a biblioteca matplotlib.
tips['day'].value_counts().plot(kind='bar')
```



```
#Definindo o Estilo dos Gráficos
sns.set_style('darkgrid')

#Aumentando o tamanho da figura
plt.figure(figsize=(8, 6))

#Gerando um gráfico de Barras com Seaborn
#Configurando exibição a partir de uma Paleta de Cores
sns.barplot(x=tips['day'].unique(), y=tips.day.value_counts(),
            palette=sns.color_palette('GnBu_d'))

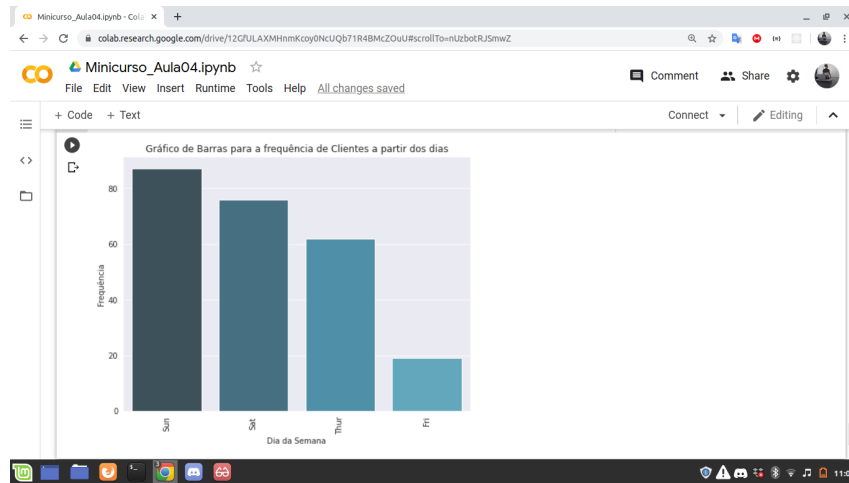
plt.title('Gráfico de Barras para a frequência de Clientes a partir dos dias')

plt.xlabel('Dia da Semana')
```

```
plt.ylabel('Frequência')

plt.xticks(rotation=90)

#Exibir Gráfico
plt.show()
```



Nova visualização, permite um degradê com intensidade associada aos valores correspondentes (frequência).



Diversificadas paletas de cores, que tem seu uso indicado de acordo com os tipos de dados, veremos esse assunto em uma próxima aula.