

1 Seaborn

Seaborn é uma biblioteca de visualização de dados para Python baseada no matplotlib. Ele fornece uma interface de alto nível para desenhar gráficos estatísticos.

2 Importante Base de Dados

Importando Bibliotecas

```
[1]: import pandas as pd
import seaborn as sns
```

Carregando base de dados

```
[2]: dados = pd.read_csv('tips.csv')
dados.head()
```

```
[2]:   total_bill  tip dessert  day  time  size
0      16.99  1.01      No  Sun  Dinner     2
1      10.34  1.66      No  Sun  Dinner     3
2      21.01  3.50      No  Sun  Dinner     3
3      23.68  3.31      No  Sun  Dinner     2
4      24.59  3.61      No  Sun  Dinner     4
```

Nos basearemos em gorjetas e sabemos aplicamos 10% do valor total da conta no Brasil, enquanto em demais países os clientes têm costume de dar caixinha livremente, sem uma porcentagem específica. Com isso, analisaremos diversas influências sobre este valor conforme citamos anteriormente.

Veremos que há 244 linhas com índice numérico e 6 colunas: *"total_bill"*, *"tip"*, *"dessert"*, *"day"*, *"time"* e *"size"*. Em um primeiro momento, entenderemos do que se tratam cada uma.

total_bill: valor total da conta paga pelo cliente

tip: valor da gorjeta em si

dessert: se a pessoa pediu sobremesa ou não

day: dia da semana

time: horário de almoço ou horário de jantar

size: quantidade de clientes na mesa

3 Análise 1 - Valor da conta e gorjeta

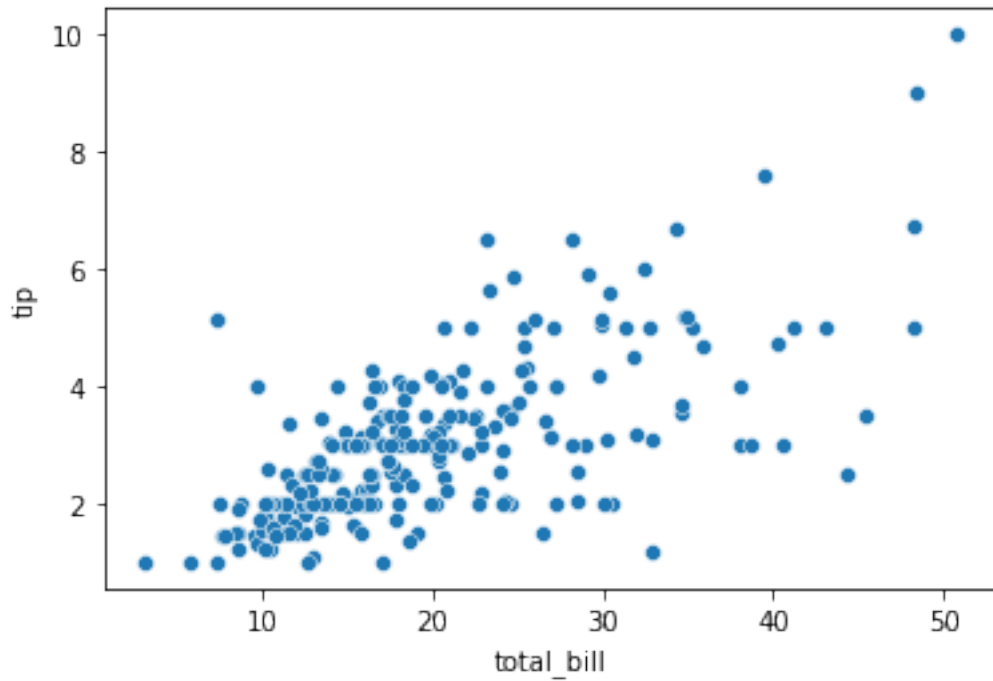
Verificando quais colunas existem no DataFrame

```
[3]: dados.columns
```

```
[3]: Index(['total_bill', 'tip', 'dessert', 'day', 'time', 'size'], dtype='object')
```

Os gráficos de dispersão ou Scatter plot são representações gráficas do relacionamento entre duas variáveis numéricas. O Scatter plot utiliza pontos para representar essa relação, cada ponto representa o valor de uma variável no eixo horizontal e o valor de outra variável no eixo vertical.

```
[4]: tip_value = sns.scatterplot(x='total_bill', y='tip', data=dados)
```

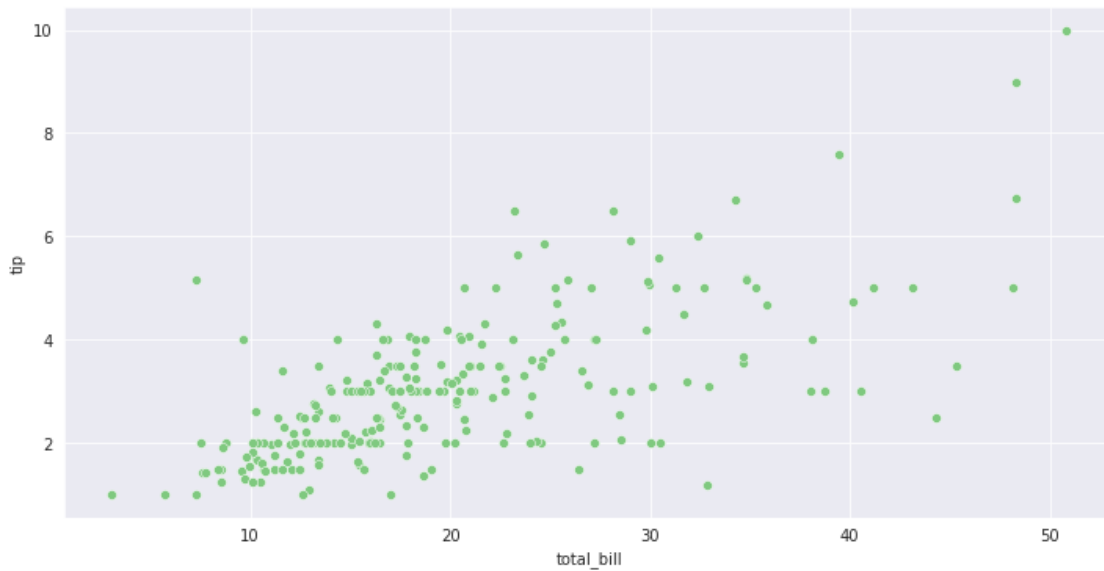


As funções do seaborn `set_palette()` e `set_style()` são responsáveis por mudar a aparência dos gráficos o `set_palette()` por definir a paleta de cores dos gráficos e a `set_style()`, o estilo.

```
[5]: sns.set_palette('Accent')  
sns.set_style('darkgrid')
```

O gráfico tem uma função para definir o tamanho da figura. `figure.set_size_inches(x, y)`, sendo *x* e *y* valores numéricos e inteiros

```
[6]: graphic = sns.scatterplot(x='total_bill', y='tip', data=dados)  
graphic.figure.set_size_inches(12, 6)
```



3.0.1 Porcentagem

Criação de uma nova coluna no DataFrame. Qual a porcentagem da gorjeta em relação ao valor total da conta.

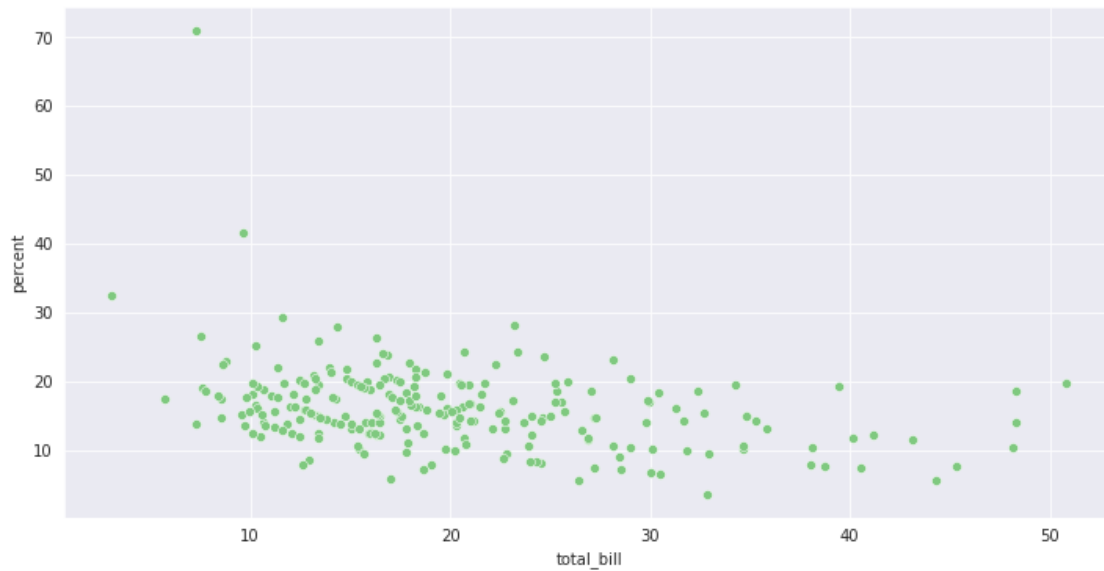
```
[7]: dados['percent'] = ((dados['tip']*100) / dados['total_bill']).round(2)
```

```
[8]: dados.head()
```

```
[8]:
```

	total_bill	tip	dessert	day	time	size	percent
0	16.99	1.01	No	Sun	Dinner	2	5.94
1	10.34	1.66	No	Sun	Dinner	3	16.05
2	21.01	3.50	No	Sun	Dinner	3	16.66
3	23.68	3.31	No	Sun	Dinner	2	13.98
4	24.59	3.61	No	Sun	Dinner	4	14.68

```
[9]: graphic = sns.scatterplot(x='total_bill', y='percent', data=dados)
      graphic.figure.set_size_inches(12, 6)
```

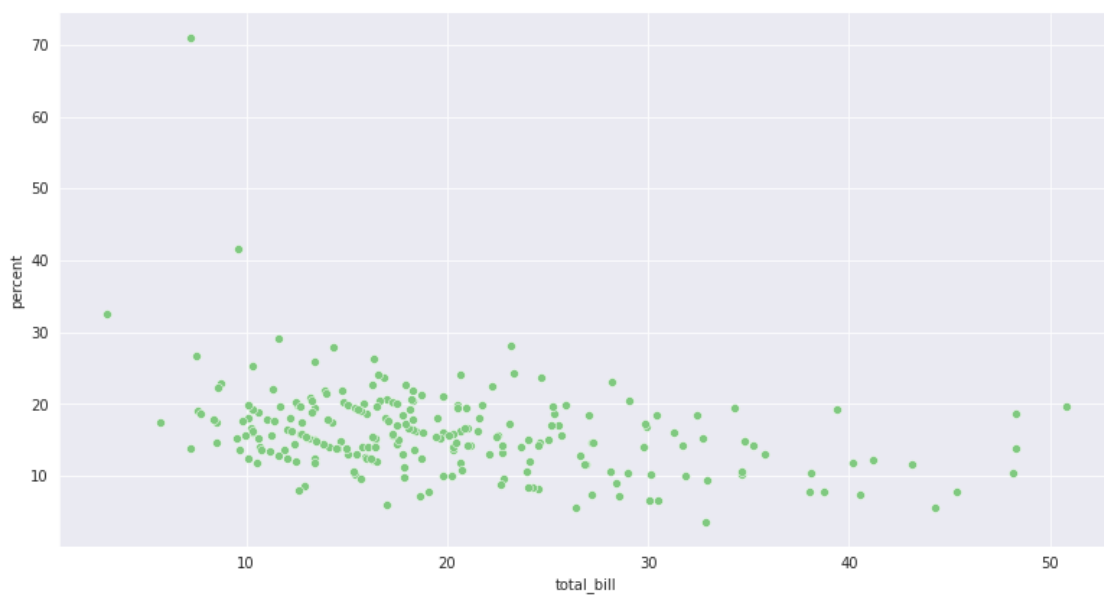


A função `relplot()` fornece acesso a duas funções de nível de eixos diferentes que mostram a relação entre duas variáveis com mapeamentos.

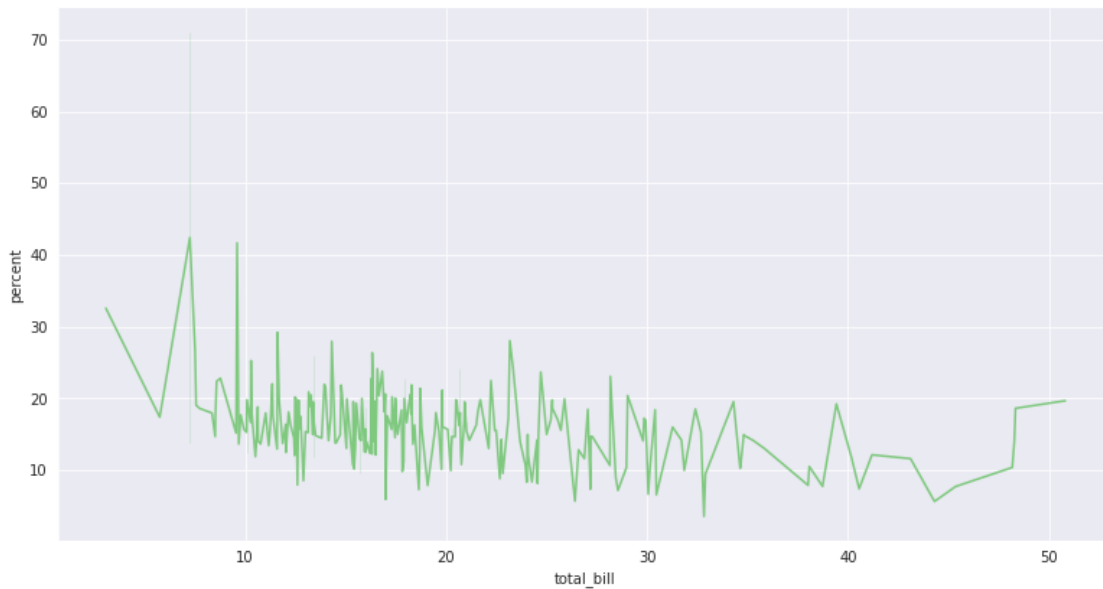
`scatterplot()` (`kind="scatter"`; padrão)

`lineplot()` (`kind="line"`)

```
[10]: graphic = sns.relplot(x='total_bill', y='percent', data=dados)
      graphic.figure.set_size_inches(12, 6)
```



```
[11]: graphic = sns.relplot(x='total_bill', y='percent', data=dados, kind='line')
graphic.figure.set_size_inches(12, 6)
```



Para deixar mais informativo o grafico, existem funções para isso.

```
set_title('titulo', fontsize=18)
set_xlabel('descrição eixo x', fontsize=14)
set_ylabel('descrição eixo y', fontsize=14)
```

```
[12]: graphic = sns.scatterplot(x='total_bill', y='tip', data=dados)
graphic.figure.set_size_inches(12, 6)
graphic.set_title('Análise do valor da gorjeta em função do valor da conta X_1
→gorjeta')
graphic.set_xlabel('Valor da Conta')
graphic.set_ylabel('Valor da Gorjeta')
graphic = graphic
```



4 Análise 2 - Sobre a mesa

A função `catplot()` fornece acesso a várias funções em nível de eixo que mostram a relação entre uma variável numérica e uma ou mais variáveis categóricas usando uma das várias representações visuais.

`stripplot()` (`kind="strip"`)

`swarmplot()` (`kind="swarm"`)

`boxplot()` (`kind="box"`)

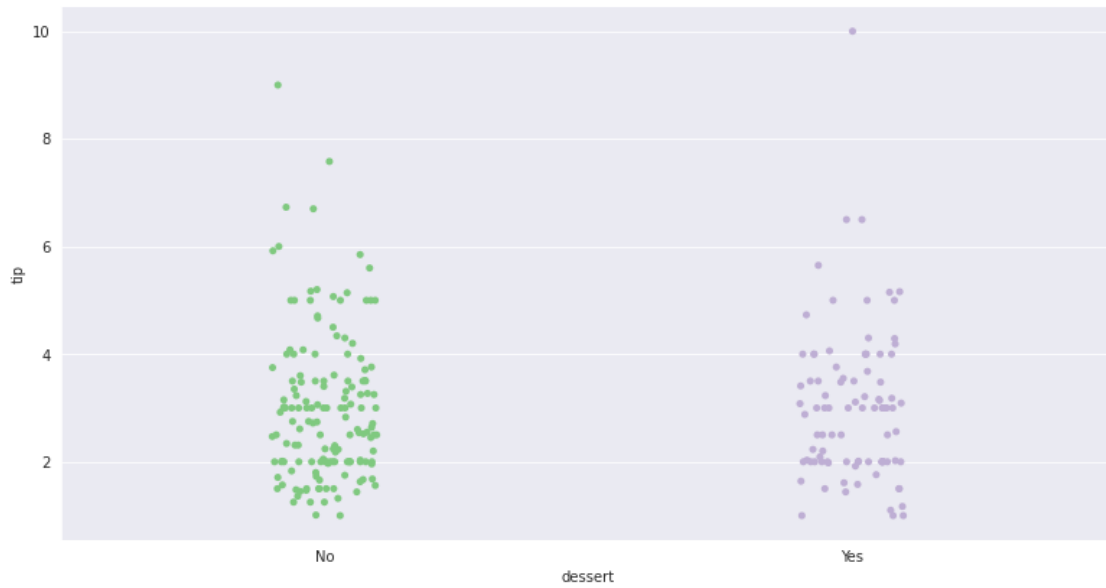
`violinplot()` (`kind="violin"`)

`boxenplot()` (`kind="boxen"`)

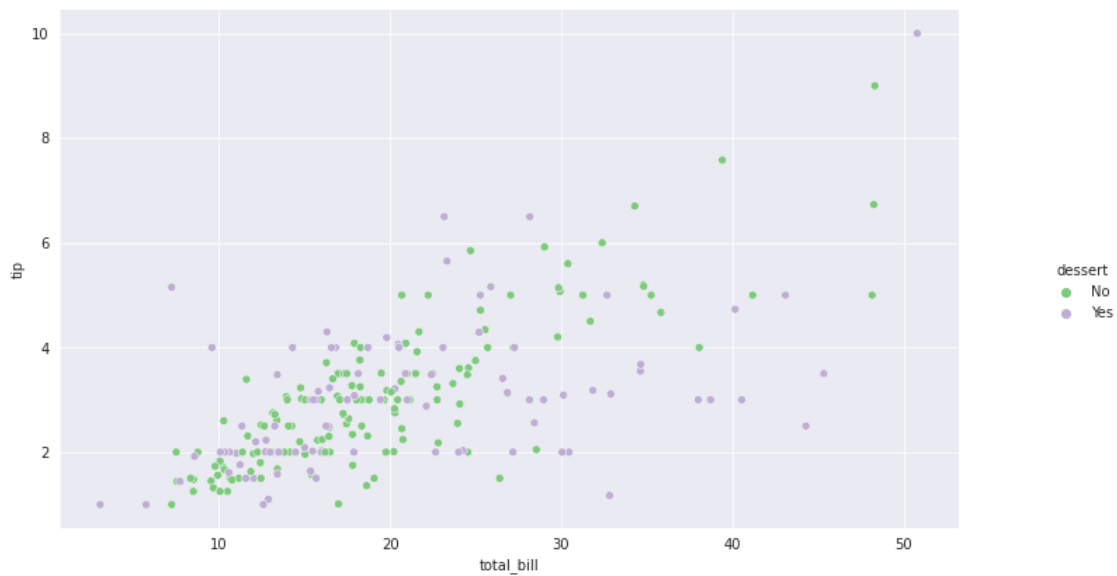
`pointplot()` (`kind="point"`)

`barplot()` (`kind="bar"`)

```
[13]: graphic = sns.catplot(x='dessert', y='tip', data=dados)
      graphic.figure.set_size_inches(12, 6)
```

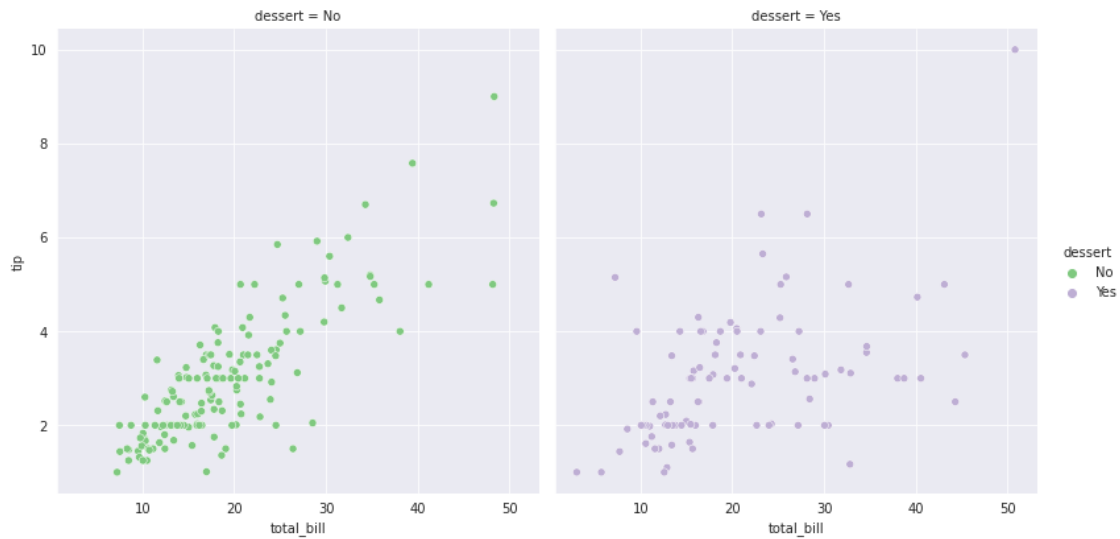


```
[14]: graphic = sns.relplot(x='total_bill', y='tip', hue='dessert', data=dados)
      graphic.figure.set_size_inches(12, 6)
```

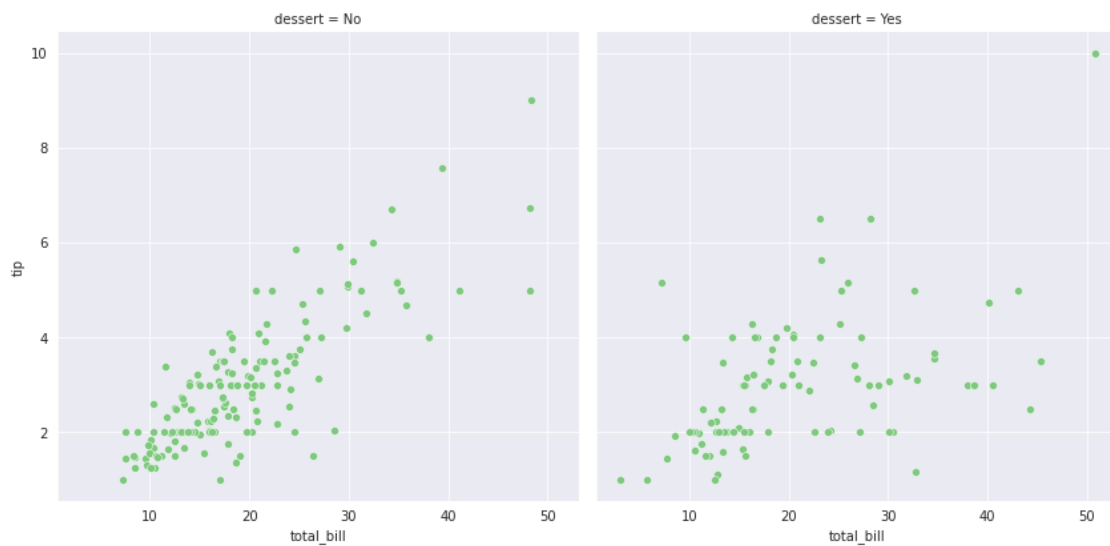


A variavel *col* define subconjuntos para plotar em diferentes facetas.

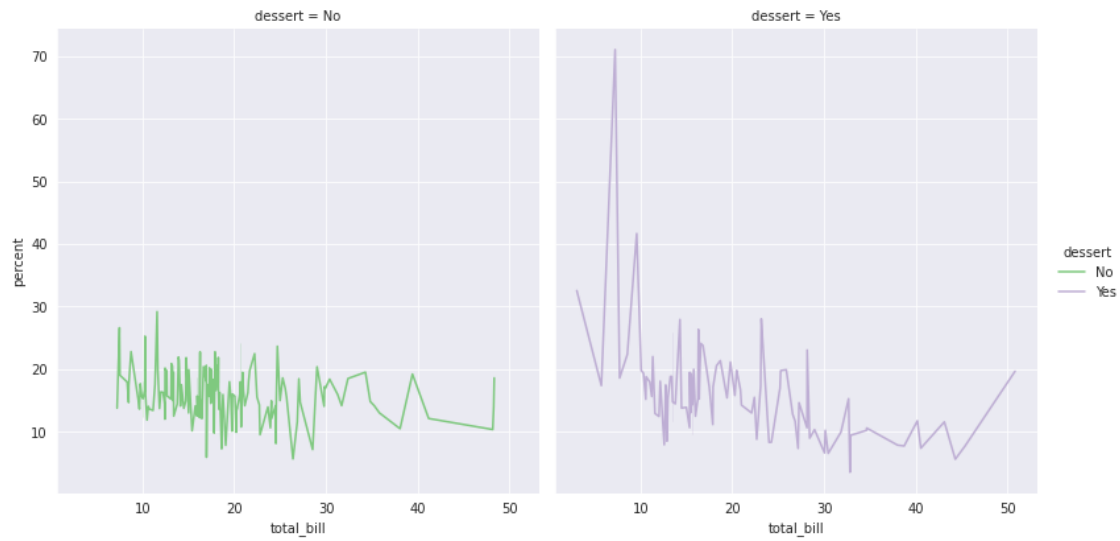
```
[15]: graphic = sns.relplot(x='total_bill', y='tip', hue='dessert', col='dessert',
      →data=dados)
      graphic.figure.set_size_inches(12, 6)
```



```
[16]: graphic = sns.relplot(x='total_bill', y='tip', col='dessert', data=dados)
      graphic.figure.set_size_inches(12, 6)
```



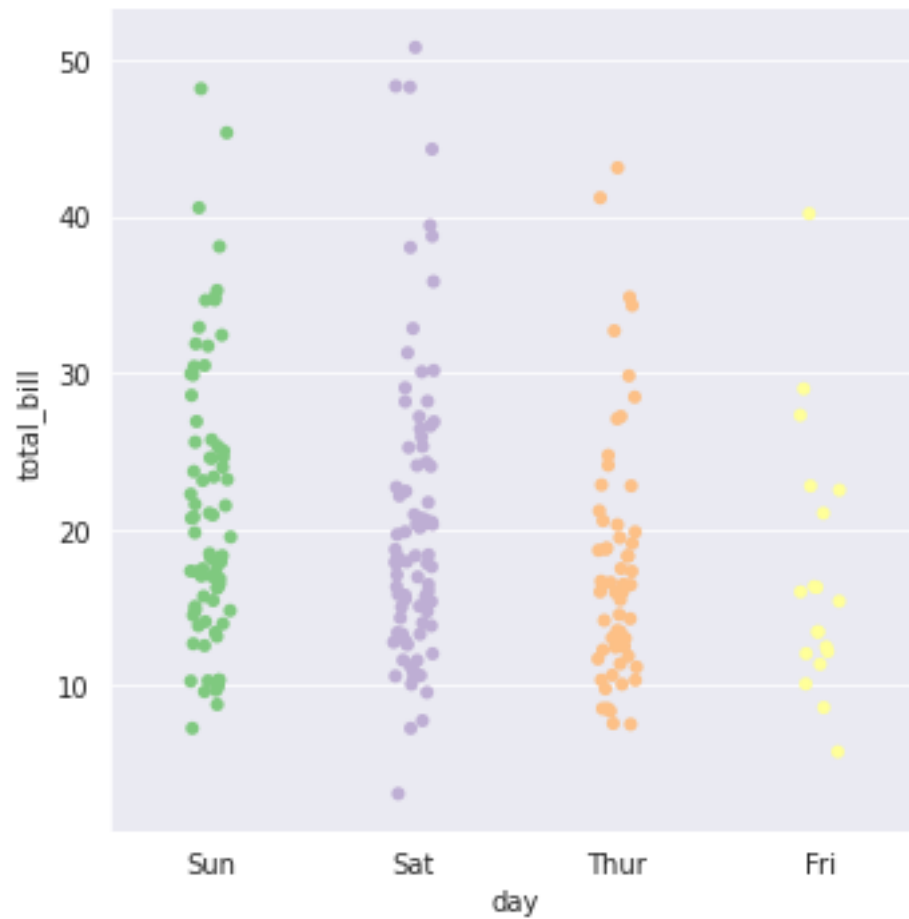
```
[17]: graphic = sns.relplot(x='total_bill', y='percent', hue='dessert', col='dessert',
      ↪ kind='line', data=dados)
      graphic.figure.set_size_inches(12, 6)
```

5 Análise 3 - Dia da Semana

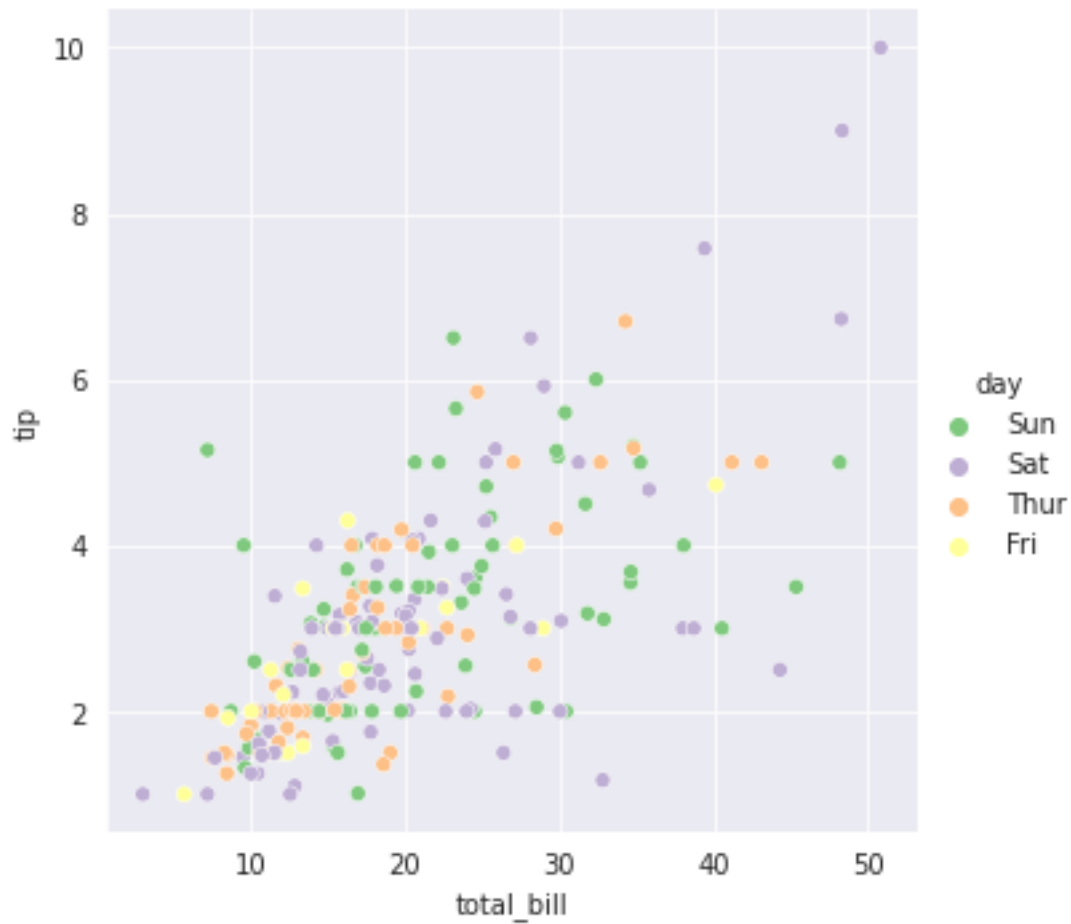
```
[18]: sns.catplot(x='day', y='total_bill', data=dados)
```

```
[18]: <seaborn.axisgrid.FacetGrid at 0x7fc799c68490>
```



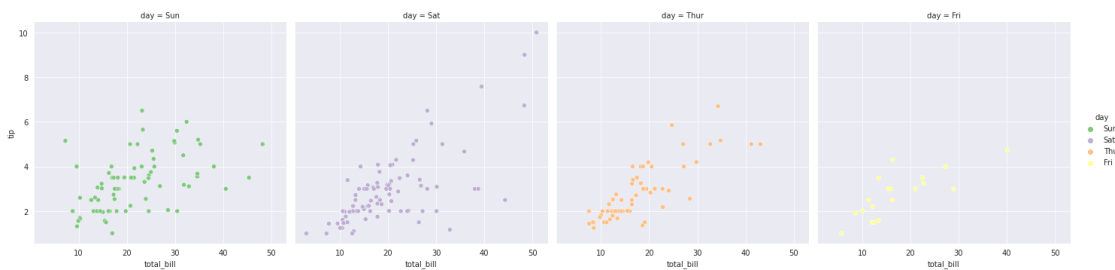
```
[19]: sns.relplot(x='total_bill', y='tip', hue='day', data=dados)
```

```
[19]: <seaborn.axisgrid.FacetGrid at 0x7fc799bcf220>
```



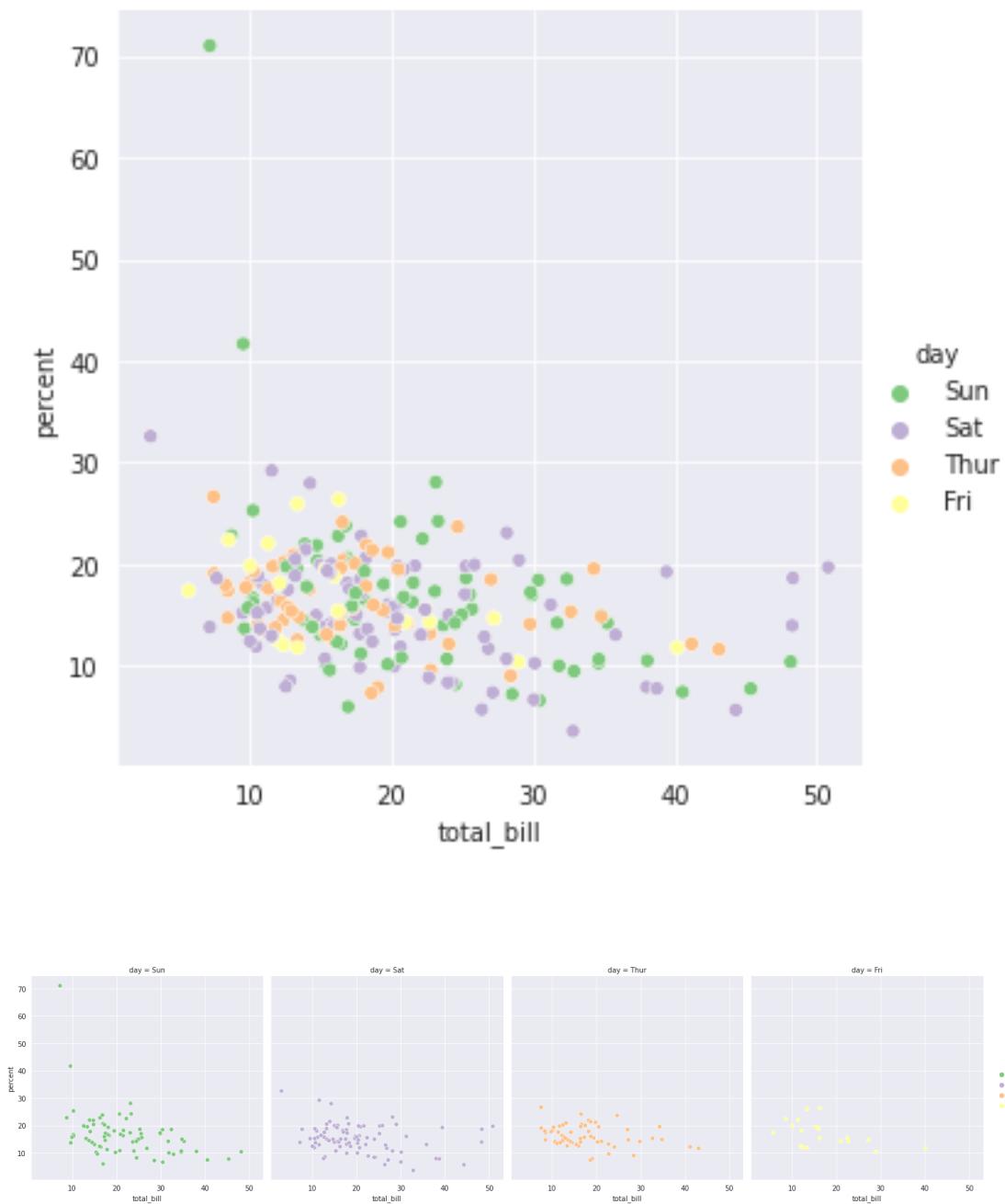
```
[20]: sns.relplot(x='total_bill', y='tip', hue='day', col='day', data=dados)
```

```
[20]: <seaborn.axisgrid.FacetGrid at 0x7fc799e0fa60>
```



```
[21]: sns.relplot(x='total_bill', y='percent', hue='day', data=dados)
sns.relplot(x='total_bill', y='percent', hue='day', col='day', data=dados)
```

```
[21]: <seaborn.axisgrid.FacetGrid at 0x7fc799b6e8e0>
```



6 Análise 4 - Hora do Dia

```
[22]: dados.head()
```

```
[22]:
```

	total_bill	tip	dessert	day	time	size	percent
0	16.99	1.01	No	Sun	Dinner	2	5.94
1	10.34	1.66	No	Sun	Dinner	3	16.05
2	21.01	3.50	No	Sun	Dinner	3	16.66
3	23.68	3.31	No	Sun	Dinner	2	13.98
4	24.59	3.61	No	Sun	Dinner	4	14.68

```
[23]: dados.time.unique()
```

```
[23]: array(['Dinner', 'Lunch'], dtype=object)
```

```
[24]: sns.catplot(x='time', y='total_bill', data=dados)
```

```
[24]: <seaborn.axisgrid.FacetGrid at 0x7fc799d24ca0>
```



```
[25]: sns.catplot(x='time', y='total_bill', kind='swarm', data=dados)
```

```
[25]: <seaborn.axisgrid.FacetGrid at 0x7fc799cc16a0>
```



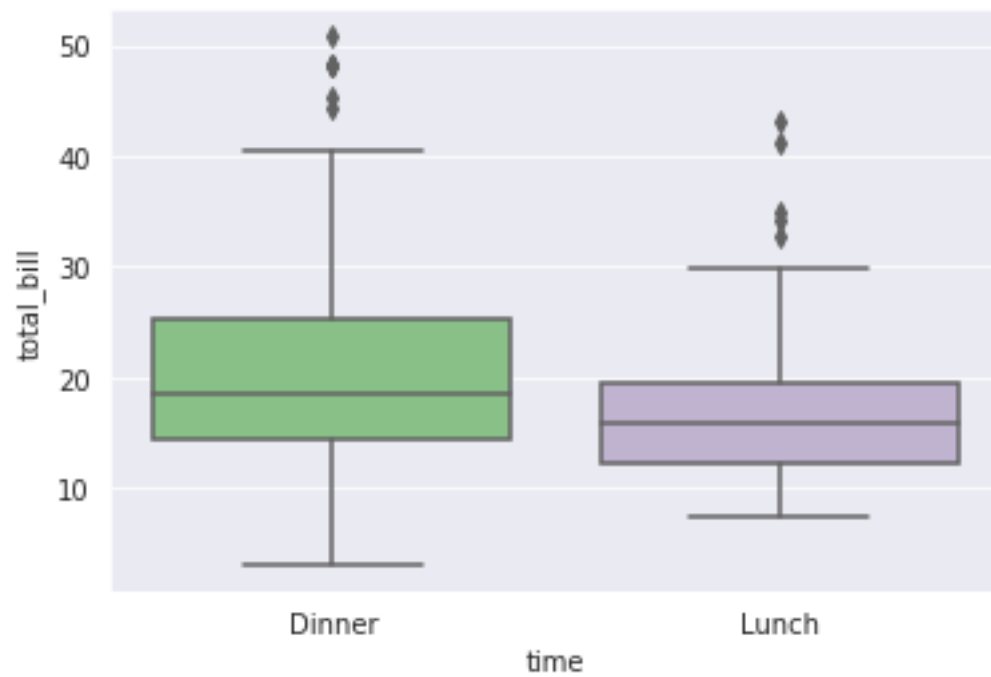
```
[26]: sns.violinplot(x='time', y='total_bill', data=dados)
```

```
[26]: <AxesSubplot:xlabel='time', ylabel='total_bill'>
```



```
[27]: sns.boxplot(x='time', y='total_bill', data=dados)
```

```
[27]: <AxesSubplot:xlabel='time', ylabel='total_bill'>
```

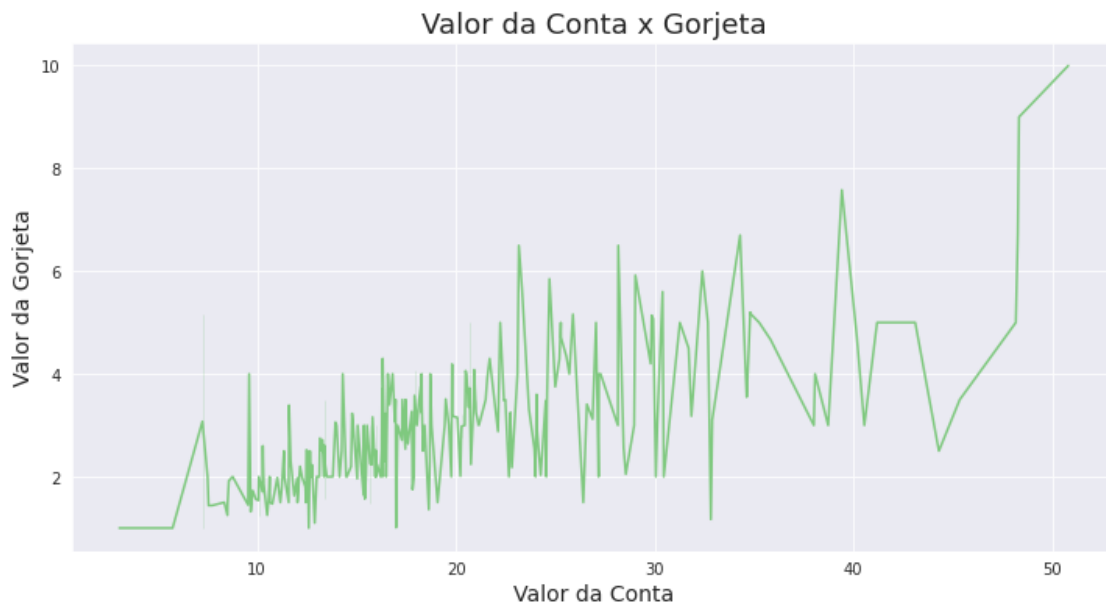


```
[ ]:
```

7 LinePlot

```
[28]: graphic = sns.lineplot(x='total_bill', y='tip', data=dados)
graphic.figure.set_size_inches(12, 6)
graphic.set_title('Valor da Conta x Gorjeta', fontsize=18)
graphic.set_xlabel('Valor da Conta', fontsize=14)
graphic.set_ylabel('Valor da Gorjeta', fontsize=14)

graphic=graphic
```



```
[ ]:
```