Introdução à estatística com Python

Autor: Ronisson Lucas Calmon da Conceição

GitHub: https://github.com/ronissonlucas (https://github.com/ronissonlucas)
Linkedin: https://www.linkedin.com/in/ronisson-lucas-calmon-da-concei%C3%A7%C3%A3o-7aa884202/ (https://www.linkedin.com/in/ronisson-lucas-calmon-da-concei%C3%A7%C3%A3o-7aa884202/)
Contato: +5573991456250

1. Tipos de variáveis

As variáveis podem ser mensuradas em quatro escalas distintas. As variáveis quantitativas podem ser classificadas em discretas ou contínuas, enquanto que as variáveis qualitativas podem ser agrupadas em nominais ou ordinais.

1. Variáveis qualitativas

- variável nominal: não há qualquer ordenação na distribuição dos dados (ou dentre as categorias).
  - Exemplos: gênero, cor dos olhos,região de procedência, etc.
- variável ordinal: há uma ordem dentre as categorias (escala ordinal).
  - Exemplos: escolaridade/grau de instrução, hierarquia militar, estágio de uma doença, mês de observação, etc.

1. Variáveis quantitativas

- variável discreta: seus valores podem ser oriundos de um conjunto finito ou enumerável (contagem).
  - Exemplos: número de filhos, número de refeições em um dia, etc.
- variável contínua: seus valores pertencem a um intervalo (mensuração).
  - Exemplos: temperatura, preço de uma ação, altura, peso, etc.

2. Medidas de tendência central

Vamos definir medidas para caracterizar o valor central de um conjunto de dados.

Média aritimética populacional

A média aritimética de uma população de N elementos $(X_1, X_2, X_3, \dots X_N)$, é por definição:

$$\mu = \frac{X_1 + X_2 + \cdots + X_N}{N}$$

Média aritimética amostral

A média amostral é a razão entre a soma dos valores amostrais e o número total de elementos desta amostra, de tamanho n.

$$\overline{X} = \frac{1}{n}\sum_{i=1}^{n} X_i$$

Considerando um conjunto de dados análogo e conhecidos os respectivos fatores de ponderação $\left(q_1, q_2, q_3 \dots q_n\right)$, definimos a média ponderada como:

$$W = \frac{\sum_{i=1}^{n} X_i q_i}{\sum_{i=1}^{n} q_i}$$

Média geométrica

A média geométrica de n valores não-negativos $\left(X_1, X_2, X_3 \dots X_n\right)$ é, por definição:

$$\overline{G} = \sqrt[n]{X_1 \times X_2 \times X_3 \dots X_n}$$

$$\overline{G} = \left(\prod_{i=1}^{n} X_i\right)^{\frac{1}{n}}$$

Média Harmônica

A média harmônica de n valores não-nulos $\left(X_1, X_2, X_3 \dots X_n\right)$ é, por definição:

$$\overline{H} = \frac{n}{\frac{1}{X_1} + \frac{1}{X_2} + \dots + \frac{1}{X_n}}$$

$$\overline{H} = \frac{n}{\sum_{i=1}^{n} \frac{1}{X_i}}$$

Temos que:

$$\overline{X} \geq \overline{G} \geq \overline{H}$$

Mediana

Consideremos que os nossos dados estejam em ordem crescente, sendo que $X_{(1)}$ denota o valor da menor representação, até um n-ésimo termo $X_{(n)}$, que será o maior termo de uma distribuição $X_1, X_2, \cdots, X_n$, então temos que:

$$X_{(1)} \leq X_{(2)} \leq \cdots \leq X_{(n-1)} \leq X_{(n)}$$

Denominamos tal ordenação de estatística de ordem.

Podemos então definir a mediana deste conjunto de dados:

$$\mathrm{md(X)} = \begin{cases} X_{(\frac{n+1}{2})}, \text{ se n ímpar;} \\ \dfrac{X_{(\frac{n}{2})} + X_{(\frac{n}{2}+1)}}{2}, \text{ se n par.} \end{cases}$$

Moda

A moda refere-se ao valor que ocorre com maior frequência em um conjunto de dados. Podemos ter um conjunto de dados com uma única moda (unimodal),duas modals (bimodal), várias modas (multimodal), ou nenhuma moda (amodal).

3. Medidas de dispersão ou variabilidade

As medidas de dispersão buscam mensurar o grau de dispersão de um conjunto de valores em torno da média.

Amplitude

A amplitude corresponde a diferença entre o maior e o menor valor de um conjunto de dados. Então para um conjunto genérico ordenado de dados:

$$X_{(1)} \leq X_{(2)} \leq \cdots \leq X_{(n-1)} \leq X_{(n)}$$

Calculamos então a amplitude:$A = X_{(n)} - X_{(1)}$

Uma forma inicial de compreender a dispersão dos dados (ou seja, distância em torno da média) é definir uma métrica de desvio de uma determinada observação em relação ao valor médio. Então o desvio d_i é tal que:

$$d_i = X_i - \overline{X}$$

Vamos construir um exemplo inicial, para verificarmos que se somarmos cada desvio obteremos um valor nulo, ou seja, para o exemplo a seguir: $d_1 + d_2 + d_3 + d_4 + d_5 = 0$

In [1]:

```
values = [2, 1, 7, 12, 13]
mean = sum(values)/len(values)  #média
di = sum([value-mean for value in values])  #soma dos desvios em relação à média
di
```

Out[1]:

```
0.0
```

In [2]:

```
#resultado do desvio de cada valor em relação à média
for value in values:
    print(value-mean)
```

```
-5.0
-6.0
0.0
5.0
6.0
```

Precisamos então definir uma outra medida de dispersão, qual seja: variância.

Variância populacional

A variância mostra o grau de dispersão de um conjunto de valores em torno do valor central.

A variância populacional de um conjunto de N elementos é o somatório do quadrado dos desvios em relação a média divido pelo tamanho da população.

$$\sigma^2 = \frac{(X_1 - \mu)^2 + (X_2 - \mu)^2 + \cdots + (X_N - \mu)^2}{N}$$

Logo, de forma compacta temos:

$$\sigma = \frac{\sum_{i=1}^{N} (X_i - \mu)^2}{N}$$

Variância amostral

Sejam $X_1, X_2, \ldots X_n$ valores amostrais, podemos definir a variância deste conjunto como segue:

$$s^2 = \frac{\sum_{i=1}^{n} \left(X_i - \overline{X}\right)^2}{n-1}$$

Podemos então definir a soma dos quadrados dos desvios: $\sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n} \left(X_i - \overline{X} \right)^2$.

Desvio Médio Absoluto

Outra forma de calcular o grau de dispersão é tomando o módulo dos desvios em relação à média:

$$\mathrm{dm(X)} = \frac{\left| X_1 - \overline{X} \right| + \left| X_2 - \overline{X} \right| + \cdots + \left| X_n - \overline{X} \right|}{n}$$

$$\therefore \mathrm{dm(X)} = \frac{\sum_{i=1}^{n} \left| X_i - \overline{X} \right|}{n}$$

Desvio padrão

O desvio padrão populacional é simplesmente a raiz quadrada da variância populacional:

$$\sigma = \sqrt{\sigma^2} = \sqrt{\frac{\sum_{i=1}^{N} \left(X_i - \mu \right)^2}{N}}$$

O desvio amostral é definido de forma análoga, como a raiz quadrada da variância amostral:

$$s = \sqrt{s^2} = \sqrt{\frac{\sum_{i=1}^{n} \left(X_i - \overline{X} \right)^2}{n-1}}$$

Vejamos outra maneira de computar a variância:

$$
\begin{aligned}
\mathrm{Var(X)} &= \frac{1}{n}\sum_{i=1}^{n}\left(X_i - \overline{X}\right)^2 \\
&= \frac{1}{n}\sum_{i=1}^{n}\left(X_i^2 - 2X_i\overline{X} + \overline{X}^2\right) \\
&= \frac{1}{n}\sum_{i=1}^{n}X_i^2 - \frac{1}{n}\sum_{i=1}^{n}2X_i\overline{X} + \frac{1}{n}\sum_{i=1}^{n}\overline{X}^2 \\
&= \frac{1}{n}\sum_{i=1}^{n}X_i^2 - 2\overline{X}\frac{1}{n}\sum_{i=1}^{n}X_i + \frac{1}{n}n\overline{X}^2 \\
&= \frac{1}{n}\sum_{i=1}^{n}X_i^2 - 2\overline{X}^2 + \overline{X}^2 \\
&= \frac{1}{n}\sum_{i=1}^{n}X_i^2 - \overline{X}^2
\end{aligned}
$$

$\therefore \mathrm{Var(X)}$ = média dos quadrados - quadrado da média

Covariância

A variância conjunta (ou covariância) mensura o grau de interdependência entre duas variáveis aleatórias, como segue.

$$\mathrm{cov(X,Y)} = \frac{1}{n}\sum_{i=1}^{n}\left(X_i - \overline{X}\right)\left(Y_i - \overline{Y}\right)$$

Outra forma de definir a covariância é expressando-a como a diferença entre a média dos produtos e o produto das médias:

$$\mathrm{Cov(X,Y)} = \frac{1}{n}\sum_{i=1}^{n}X_iY_i - \overline{X}\overline{Y}$$

Coeficiente de variação

Podemos definir uma outra medida de dispersão que não seja tão afetada pela magnitude dos dados, como a variância. Para tanto, podemos calcular o coeficiente de variação:

$$CV = \frac{S}{\overline{X}}$$

4. Quantis

Bussab e Morettin (2010) ressaltam que tanto a média, quanto o desvio padrão podem não ser medidas suficientes para se representar um conjunto de dados, pois:

- ambos são afetados de forma demasiada por valores extremos;
- não podemos ter uma noção clara da simetria ou assimetria da distribuição dos dados.

Precisamos então definir outras medidas para mitigar tais problemas.

Definição: um quantil de ordem p é definido como $q(p)$, sendo p uma proporção $0 < p < 1$, tal que $100p\%$ das observações sejam menores que $q(p)$.

- $q(0,25) =$ 1º Quartil (25% das observações abaixo e 75% das observações acima)
- $q(0,5) =$ 2º Quartil (50% das observações abaixo e 50% das observações acima)
- $q(0,75) =$ 3º Quartil (75% das observações abaixo e 25% das observações acima)

Então os quartis dividem o conjunto de dados em quatro partes iguais.

Um boxtpot (gráfico de caixa) fornece uma poderosa visualização da distribuição dos dados e valores discrepantes (outliers), sendo formados pelo primeiro quartil, segundo quartil (mediana), terceiro quartil e limites superior e inferior.

5. Correlação e regressão

5.1 Coeficiente de correlação

A partir das definições anteriores, podemos mensurar o grau de correlação entre duas variáveis X e Y.

$$\mathrm{corr(X,Y)} = \frac{\mathrm{cov(X,Y)}}{\mathrm{dp(X)dp(Y)}}$$

- alta correlação não se traduz necessariamente em uma relação de causalidade entre as variáveis;
- os valores podem variar em uma escala de -1 a 1, sendo que valores positivos (negativos) denotam correlação positiva (negativa).

5.2 Regressão

O objetivo de uma regressão é modelar a relação entre uma variável dependente (target) e uma variável independente, ou um conjunto de variáveis independentes.

5.2.1 Regressão linear simples

- Uma variável target (y) e uma variável explicativa x (single feature).
- Assume-se a hipótese que as variável x e y seja linearmente relacionadas.
- Exemplos:
  - Consumo e Renda
  - Salário e anos de estudo
  - Vendas e gastos em propaganda

$$Y = f(x_i) = \beta_0 + \beta_1 x_i$$

$$Y_i = \beta_0 + \beta_1 x_i + \epsilon_i$$
$$\epsilon_i = y_i - f(x_i)$$

$$E(\epsilon_i) = 0$$

Podemos simplificar o processo álgebrico definindo variáveis centradas na média.

$$y_i = Y_i - \overline{Y_i}$$
$$x_i = X_i - \overline{X}$$
$$\overline{Y_i} = \alpha + \beta\overline{X} + 0$$

$$Y_i - \overline{Y} = (\alpha - \alpha) + \beta(X_i - \overline{X}) + \epsilon_i$$

$$y_i = \beta x_i + \epsilon_i$$

$$\epsilon_i = y_i - \beta x_i$$

Assim, a soma dos quadrados dos erros é denotada como:

$$\sum_{i=1}^{n} (\epsilon_i)^2 = \sum_{i=1}^{n} (y_i - \beta x_i)^2$$

$$\sum_{i=1}^{n}(\epsilon_i)^2 = \sum_{i=1}^{n}(y_i^2 - 2\beta x_i y_i + \beta^2 x_i^2)$$
$$\sum_{i=1}^{n}(\epsilon_i)^2 = \sum_{i=1}^{n}(y_i^2 + \beta^2 x_i^2 - 2\beta x_i y_i)$$
$$\sum_{i=1}^{n}(\epsilon_i)^2 = \sum_{i=1}^{n} y_i^2 + \sum_{i=1}^{n} \beta^2 x_i^2 - 2\sum_{i=1}^{n} \beta x_i y_i$$

Sendo β uma constante dentro do somatório, temos que:

$$\sum_{i=1}^{n}(\epsilon_i)^2 = \sum_{i=1}^{n} y_i^2 + \beta^2 \sum_{i=1}^{n} x_i^2 - 2\beta \sum_{i=1}^{n} x_i y_i$$

O objetivo é minimizar a função acima, para tanto podemos derivar a expressão em relação β e igualar a zero (condição de primeira ordem):

$$\frac{\partial}{\partial \beta} \sum_{i=1}^{n}(\epsilon_i)^2 = 0$$

Usaremos a notação $\hat{\beta}$ (estimador) para β. Assim:

$$2\hat{\beta} \sum_{i=1}^{n} x_i^2 - 2\sum_{i=1}^{n} x_i y_i = 0$$
$$\hat{\beta} \sum_{i=1}^{n} x_i^2 - \sum_{i=1}^{n} x_i y_i = 0$$

Obtemos assim os estimadores:

$$\hat{\beta} = \frac{\sum_{i=1}^{n} x_i y_i}{\sum_{i=1}^{n} x_i^2}$$
$$\hat{\alpha} = \overline{Y} - \hat{\beta}\overline{X}$$

Podemos denotar:

$$\hat{\beta} = \frac{S_{xy}}{S_x^2}$$

Em que:

$$S_{xy} = \sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y}) = \sum_{i=1}^{n} y_i x_i - n\bar{x}\bar{y}$$

$$S_x^2 = \sum_{i=1}^{n}(x_i - \bar{x})^2 = \sum_{i=1}^{n} x_i^2 - n(\bar{x})^2$$

Código

In [3]:

```
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
import warnings
warnings.filterwarnings("ignore")
```

In [4]:

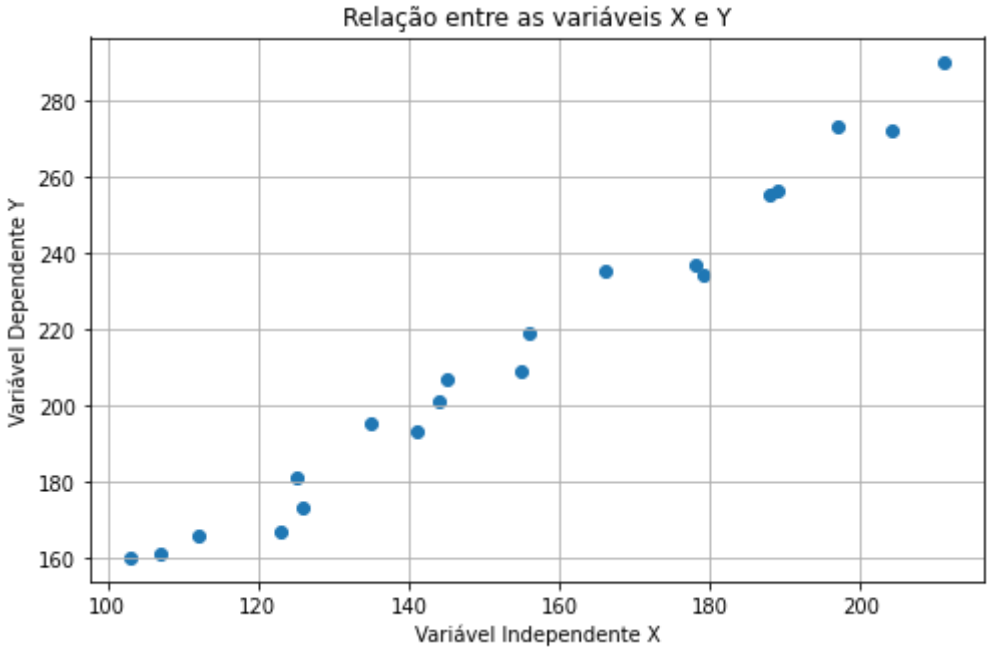
```
x = [103,123,145,126,189,211,178,155,141,156,166,179,197,204,125,112,107,135,144
,188]
y = [160,167,207,173,256,290,237,209,193,219,235,234,273,272,181,166,161,195,201
,255]
dados = {'X': x,
        'Y': y}
```

In [5]:

```
df = pd.DataFrame(dados)
```

In [6]:

```
plt.figure(figsize=(8, 5))
plt.title('Relação entre as variáveis X e Y')
plt.scatter(df['X'],df['Y'])
plt.xlabel('Variável Independente X')
plt.ylabel('Variável Dependente Y')
plt.grid(True)
plt.show()
```



In [7]:

```
df.sum()
```

Out[7]:

```
X    3084
Y    4284
dtype: int64
```

In [8]:

```
df.mean()
```

Out[8]:

```
X    154.2
Y    214.2
dtype: float64
```

Denotaremos:

$$x = X_i - \overline{X}$$
$$y = Y_i - \overline{Y}$$
$$x^2 = (X_i - \overline{X})^2$$
$$y^2 = (Y_i - \overline{Y})^2$$
$$xy = (\sum_{i=1}^{n}(X_i - \overline{X})(Y_i - \overline{Y})$$

In [9]:

```
df['x'] = df['X'] - df['X'].mean()
df['y'] = df['Y'] - df['Y'].mean()
df['x^2'] = (df['X'] - df['X'].mean())**2
df['y^2'] = (df['Y'] - df['Y'].mean())**2
df['xy'] = df['x']*df['y']
```

In [10]:

```
df
```

Out[10]:

	X	Y	x	y	x^2	y^2	xy
0	103	160	-51.2	-54.2	2621.44	2937.64	2775.04
1	123	167	-31.2	-47.2	973.44	2227.84	1472.64
2	145	207	-9.2	-7.2	84.64	51.84	66.24
3	126	173	-28.2	-41.2	795.24	1697.44	1161.84
4	189	256	34.8	41.8	1211.04	1747.24	1454.64
5	211	290	56.8	75.8	3226.24	5745.64	4305.44
6	178	237	23.8	22.8	566.44	519.84	542.64
7	155	209	0.8	-5.2	0.64	27.04	-4.16
8	141	193	-13.2	-21.2	174.24	449.44	279.84
9	156	219	1.8	4.8	3.24	23.04	8.64
10	166	235	11.8	20.8	139.24	432.64	245.44
11	179	234	24.8	19.8	615.04	392.04	491.04
12	197	273	42.8	58.8	1831.84	3457.44	2516.64
13	204	272	49.8	57.8	2480.04	3340.84	2878.44
14	125	181	-29.2	-33.2	852.64	1102.24	969.44
15	112	166	-42.2	-48.2	1780.84	2323.24	2034.04
16	107	161	-47.2	-53.2	2227.84	2830.24	2511.04
17	135	195	-19.2	-19.2	368.64	368.64	368.64
18	144	201	-10.2	-13.2	104.04	174.24	134.64
19	188	255	33.8	40.8	1142.44	1664.64	1379.04

In [11]:

```
beta = round(df['xy'].sum() / df['x^2'].sum(), 3)
```

In [12]:

```
beta
```

Out[12]:

```
1.207
```

In [13]:

```
alpha = round(df['Y'].mean() - beta*df['X'].mean(), 2)
```

In [14]:

```
alpha
```

Out[14]:

```
28.08
```

$$Y_i = 28,08 + 1,207X_i$$

Recuperação de parâmetros:

In [15]:

```
reta = [round(alpha+beta*x,2) for x in df['X']]
```

In [16]:

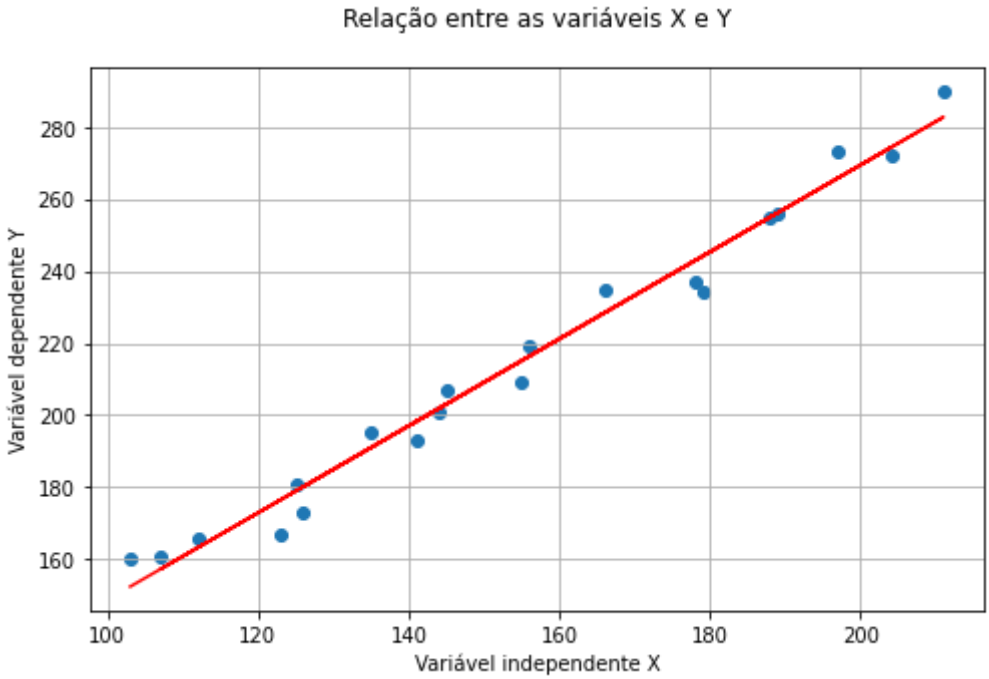
```
reta
```

Out[16]:

```
[152.4,
 176.54,
 203.1,
 180.16,
 256.2,
 282.76,
 242.93,
 215.17,
 198.27,
 216.37,
 228.44,
 244.13,
 265.86,
 274.31,
 178.96,
 163.26,
 157.23,
 191.03,
 201.89,
 255.0]
```

In [17]:

```
plt.figure(figsize = (8, 5))
plt.scatter(df['X'], df['Y'])
plt.plot(df['X'], reta, color='r')
plt.title('Relação entre as variáveis X e Y', pad=20)
plt.xlabel('Variável independente X')
plt.ylabel('Variável dependente Y')
plt.grid(True)
plt.show()
```



In [18]:

```
df['Y-Y^'] = round(df['Y'] -reta,1)
```

In [19]:

```
df['(Y-Y^)^2'] = round((df['Y'] -reta)**2,1)
```

In [20]:

```
df['Y^'] = reta
```

In [21]:

```
df
```

Out[21]:

	X	Y	x	y	x^2	y^2	xy	Y-Y^	(Y-Y^)^2	Y^
0	103	160	-51.2	-54.2	2621.44	2937.64	2775.04	7.6	57.8	152.40
1	123	167	-31.2	-47.2	973.44	2227.84	1472.64	-9.5	91.0	176.54
2	145	207	-9.2	-7.2	84.64	51.84	66.24	3.9	15.2	203.10
3	126	173	-28.2	-41.2	795.24	1697.44	1161.84	-7.2	51.3	180.16
4	189	256	34.8	41.8	1211.04	1747.24	1454.64	-0.2	0.0	256.20
5	211	290	56.8	75.8	3226.24	5745.64	4305.44	7.2	52.4	282.76
6	178	237	23.8	22.8	566.44	519.84	542.64	-5.9	35.2	242.93
7	155	209	0.8	-5.2	0.64	27.04	-4.16	-6.2	38.1	215.17
8	141	193	-13.2	-21.2	174.24	449.44	279.84	-5.3	27.8	198.27
9	156	219	1.8	4.8	3.24	23.04	8.64	2.6	6.9	216.37
10	166	235	11.8	20.8	139.24	432.64	245.44	6.6	43.0	228.44
11	179	234	24.8	19.8	615.04	392.04	491.04	-10.1	102.6	244.13
12	197	273	42.8	58.8	1831.84	3457.44	2516.64	7.1	51.0	265.86
13	204	272	49.8	57.8	2480.04	3340.84	2878.44	-2.3	5.3	274.31
14	125	181	-29.2	-33.2	852.64	1102.24	969.44	2.0	4.2	178.96
15	112	166	-42.2	-48.2	1780.84	2323.24	2034.04	2.7	7.5	163.26
16	107	161	-47.2	-53.2	2227.84	2830.24	2511.04	3.8	14.2	157.23
17	135	195	-19.2	-19.2	368.64	368.64	368.64	4.0	15.8	191.03
18	144	201	-10.2	-13.2	104.04	174.24	134.64	-0.9	0.8	201.89
19	188	255	33.8	40.8	1142.44	1664.64	1379.04	0.0	0.0	255.00

$$SQT = \sum y_i^2$$

In [22]:

```
SQT = df['y^2'].sum().round(2)
```

In [23]:

```
SQT
```

Out[23]:

```
31513.2
```

In [24]:

```
SQE = sum((df['Y^'] - df['Y^'].mean())**2)
```

In [25]:

```
SQE
```

Out[25]:

30884.645694999996

In [26]:

```
SQR = round(sum(df['(Y-Y^)^2']),2)
```

In [27]:

```
SQR
```

Out[27]:

620.1

In [28]:

```
R2 = SQE/SQT
```

In [29]:

```
round(R2*100,2)
```

Out[29]:

98.01

OLS usando statsmodels

In [30]:

```
import statsmodels.api as sm
```

In [31]:

```
X = df['X']
X = sm.add_constant(X)
Y = df['Y']
resultados = sm.OLS(Y,X).fit()
```

In [32]:

```
resultados.summary()
```

Out[32]:

OLS Regression Results

Dep. Variable:	Y	**R-squared:**	0.980
Model:	OLS	**Adj. R-squared:**	0.979
Method:	Least Squares	**F-statistic:**	896.8
Date:	Wed, 27 Jan 2021	**Prob (F-statistic):**	8.27e-17
Time:	11:34:27	**Log-Likelihood:**	-62.720
No. Observations:	20	**AIC:**	129.4
Df Residuals:	18	**BIC:**	131.4
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	28.0532	6.353	4.416	0.000	14.706	41.400
X	1.2072	0.040	29.946	0.000	1.122	1.292

Omnibus:	2.621	**Durbin-Watson:**	2.683
Prob(Omnibus):	0.270	**Jarque-Bera (JB):**	1.430
Skew:	-0.345	**Prob(JB):**	0.489
Kurtosis:	1.887	**Cond. No.**	763.

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

5.2.2 Regressão Linear Múltipla

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \beta_3 x_{i3} + \ldots + \beta_p x_{ip} + \epsilon_i$$

Generalizando, em notação matricial temos que:

$$\mathbf{y} = \mathbf{X}\beta + \epsilon$$

Onde:

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_p \end{bmatrix}$$

$$\beta = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_p \end{bmatrix}$$

$$\epsilon = \begin{bmatrix} \epsilon_0 \\ \epsilon_1 \\ \vdots \\ \epsilon_p \end{bmatrix}$$

$$\mathbf{X} = \begin{bmatrix} 1 & x_{11} & \dots & x_{1p} \\ 1 & x_{21} & \dots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & \dots & x_{np} \end{bmatrix}$$

Com um problema de otimização análogo ao anterior obtemos a partir do vetor de erros:

$$\mathbf{e}'\mathbf{e} = (\mathbf{Y} - \mathbf{X}\beta)'(\mathbf{Y} - \mathbf{X}\beta)$$
$$\mathbf{e}'\mathbf{e} = \mathbf{Y}'\mathbf{Y} - \mathbf{Y}'\mathbf{X}\beta - \beta'\mathbf{X}'\mathbf{Y} + \beta'\mathbf{X}'\mathbf{X}\beta$$
$$\mathbf{e}'\mathbf{e} = \mathbf{Y}'\mathbf{Y} - \mathbf{2}\beta'\mathbf{X}'\mathbf{Y} + \beta'\mathbf{X}'\mathbf{X}\beta$$

Derivando em relação a β e igualando a zero:

$$-2\mathbf{X}'\mathbf{Y} + 2\mathbf{X}'\mathbf{X}\hat{\beta} = 0$$
$$2\mathbf{X}'\mathbf{X}\hat{\beta} = 2\mathbf{X}'\mathbf{Y}$$

Pré-multiplicando ambos os lados da equação por $(\mathbf{X}'\mathbf{X})^{-1}$:

$$(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{X}\hat{\beta} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y}$$

$$\therefore \hat{\beta} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'y$$

Estimando uma regressão múltipla

Modelo a ser estimado:

$$Y = \beta_1 + \beta_2 X_2 + \beta_3 X_3 + \epsilon_i$$

In [33]:

```
Y = {'Y':[800, 1160, 1580, 2010, 1890, 2600, 2070,1890,1830,1740,1380,1060]}
X = {1:np.ones(12),
     'X2':[2, 4, 6, 8,7,12,11,10,9,8,6,4],
    'X3':[0.8,0.7,0.5,0.4,0.2,0.2,0.8,0.7,0.6,0.1,0.5,0.4]}
```

In [34]:

```
Y = pd.DataFrame(Y)
X = pd.DataFrame(X).values
```

In [35]:

```
X
```

Out[35]:

```
array([[ 1. ,  2. ,  0.8],
       [ 1. ,  4. ,  0.7],
       [ 1. ,  6. ,  0.5],
       [ 1. ,  8. ,  0.4],
       [ 1. ,  7. ,  0.2],
       [ 1. , 12. ,  0.2],
       [ 1. , 11. ,  0.8],
       [ 1. , 10. ,  0.7],
       [ 1. ,  9. ,  0.6],
       [ 1. ,  8. ,  0.1],
       [ 1. ,  6. ,  0.5],
       [ 1. ,  4. ,  0.4]])
```

In [36]:

```
beta = np.linalg.inv(X.T.dot(X)).dot(X.T.dot(Y))
```

In [37]:

```
beta.round(2)
```

Out[37]:

```
array([[ 789.33],
       [ 149.56],
       [-419.26]])
```

Valor dos parâmetros:

$$\hat{\beta_1} = 789,33$$
$$\hat{\beta_2} = 149,56$$
$$\hat{\beta_3} = -419.26$$

Modelo estimado:

$$\hat{Y} = 789,33 + 149,56X_2 - 419,26X_3$$

OLS usando statsmodels

In [38]:

```
X = sm.add_constant(X)
resultados = sm.OLS(Y,X).fit()
```

In [39]:

```
resultados.summary()
```

Out[39]:

OLS Regression Results

Dep. Variable:	Y	**R-squared:**	0.937
Model:	OLS	**Adj. R-squared:**	0.923
Method:	Least Squares	**F-statistic:**	66.82
Date:	Wed, 27 Jan 2021	**Prob (F-statistic):**	3.98e-06
Time:	11:34:28	**Log-Likelihood:**	-74.499
No. Observations:	12	**AIC:**	155.0
Df Residuals:	9	**BIC:**	156.5
Df Model:	2		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	789.3296	155.258	5.084	0.001	438.112	1140.547
x1	149.5593	14.225	10.514	0.000	117.381	181.738
x2	-419.2566	179.557	-2.335	0.044	-825.443	-13.070

Omnibus:	1.230	**Durbin-Watson:**	0.438
Prob(Omnibus):	0.541	**Jarque-Bera (JB):**	0.739
Skew:	-0.122	**Prob(JB):**	0.691
Kurtosis:	1.809	**Cond. No.**	43.3

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

6. Implementações em Python

Com a biblioteca Pandas podemos facilmente implementar estatísticas descritivas. O quadro a seguir demonstra as possibilidades.

Função	Descrição
sum()	Retorna a soma dos valores.
prod()	Retorna o produto dos valores.
count()	Retorna o número de observações não nulas.
mean()	Retorna a média.
median()	Retorna a mediana.
mode()	Retorna a moda.
std()	Retorna o desvio padrão.
var()	Retorna a variância.
mad()	Retorna o desvio médio absoluto
corr()	Retorna a correlação.
cov()	Retorna a covariância.
describe()	Retorna um sumário das estatísticas descritivas
max()	Retorna o valor máximo.
min()	Retorna o valor mínimo.
abs()	Retorna o valor absoluto.
cumsum()	Retorna a soma cumulativa.
cumprod()	Retorna o produto cumulativo.

In [40]:

```
import numpy as np
import pandas as pd
```

In [41]:

```
#DataFrame com dados sintéticos
np.random.seed(21)
n = 30
time = pd.date_range(start = '2021-1-25', periods = n)
data = {
    'Brand A':np.random.normal(loc = 100, scale = 0.5, size = n).round(1),
    'Brand B':np.random.uniform(low = 2, high = 10, size = n).round(1),
    'Brand C':np.random.randint(low = 1, high = 350, size = n)}
df = pd.DataFrame(data = data, index = time)
df.loc['2021-1-26':'2021-1-27','Brand A'] = 100
```

Medidas de tendência central

In [42]:

```
df.head()
```

Out[42]:

	Brand A	Brand B	Brand C
2021-01-25	100.0	6.9	346
2021-01-26	100.0	7.2	338
2021-01-27	100.0	5.1	68
2021-01-28	99.4	5.3	2
2021-01-29	100.4	8.5	66

In [43]:

```
df.mean()  #média do período
```

Out[43]:

```
Brand A     99.930000
Brand B      6.816667
Brand C    175.133333
dtype: float64
```

In [44]:

```
df.mean(axis = 1)  #média diária
```

Out[44]:

```
2021-01-25    150.966667
2021-01-26    148.400000
2021-01-27     57.700000
2021-01-28     35.566667
2021-01-29     58.300000
2021-01-30    104.633333
2021-01-31     90.366667
2021-02-01     51.933333
2021-02-02     47.733333
2021-02-03    125.633333
2021-02-04    125.333333
2021-02-05    105.266667
2021-02-06    118.466667
2021-02-07    117.100000
2021-02-08     96.233333
2021-02-09    112.800000
2021-02-10     76.400000
2021-02-11     62.433333
2021-02-12    129.733333
2021-02-13     71.566667
2021-02-14     37.533333
2021-02-15     66.366667
2021-02-16     49.066667
2021-02-17    152.133333
2021-02-18    100.200000
2021-02-19    134.033333
2021-02-20    122.300000
2021-02-21    109.033333
2021-02-22     62.900000
2021-02-23     98.666667
Freq: D, dtype: float64
```

In [45]:

```
df.mode()  #moda
```

Out[45]:

	Brand A	Brand B	Brand C
0	100.0	8.6	268

In [46]:

```
df.median()  #mediana
```

Out[46]:

```
Brand A    100.00
Brand B      7.05
Brand C    192.00
dtype: float64
```

Medidas de dispersão

In [47]:

```
A = df.max()-df.min()  #amplitude
```

In [48]:

```
A
```

Out[48]:

```
Brand A      1.8
Brand B      6.1
Brand C    347.0
dtype: float64
```

In [49]:

```
df.std()  #desvio padrão
```

Out[49]:

```
Brand A      0.447329
Brand B      1.824183
Brand C    105.577471
dtype: float64
```

In [50]:

```
df.var()  #variância
```

Out[50]:

```
Brand A        0.200103
Brand B        3.327644
Brand C    11146.602299
dtype: float64
```

In [51]:

```
df.mad()  #desvio médio absoluto
```

Out[51]:

```
Brand A     0.348000
Brand B     1.568889
Brand C    90.648889
dtype: float64
```

Quantis

In [52]:

```
df.quantile([0.25, 0.5, 0.75])
```

Out[52]:

	Brand A	Brand B	Brand C
0.25	99.70	5.375	81.25
0.50	100.00	7.050	192.00
0.75	100.25	8.475	255.25

Sumário dos dados

In [53]:

```
df.describe()
```

Out[53]:

	Brand A	Brand B	Brand C
count	30.000000	30.000000	30.000000
mean	99.930000	6.816667	175.133333
std	0.447329	1.824183	105.577471
min	99.000000	3.100000	2.000000
25%	99.700000	5.375000	81.250000
50%	100.000000	7.050000	192.000000
75%	100.250000	8.475000	255.250000
max	100.800000	9.200000	349.000000

Correlação e covariância

In [54]:

```
corr = df.corr()  #matrix de correlação
```

In [55]:

```
cov = df.cov()  #matrix de covariância
```

In [56]:

```
corr
```

Out[56]:

	Brand A	Brand B	Brand C
Brand A	1.000000	0.077966	-0.039953
Brand B	0.077966	1.000000	0.020238
Brand C	-0.039953	0.020238	1.000000

In [57]:

```
cov
```

Out[57]:

	Brand A	Brand B	Brand C
Brand A	0.200103	0.063621	-1.886897
Brand B	0.063621	3.327644	3.897701
Brand C	-1.886897	3.897701	11146.602299

Tips dataset

In [58]:

```
import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
#carrega o dataset tips
df = sns.load_dataset('tips')
```

In [59]:

```
df.head()
```

Out[59]:

	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4

In [60]:

```
df.tail()
```

Out[60]:

	total_bill	tip	sex	smoker	day	time	size
239	29.03	5.92	Male	No	Sat	Dinner	3
240	27.18	2.00	Female	Yes	Sat	Dinner	2
241	22.67	2.00	Male	Yes	Sat	Dinner	2
242	17.82	1.75	Male	No	Sat	Dinner	2
243	18.78	3.00	Female	No	Thur	Dinner	2

In [61]:

```
df.columns
```

Out[61]:

```
Index(['total_bill', 'tip', 'sex', 'smoker', 'day', 'time', 'siz
e'], dtype='object')
```

In [62]:

```
df.shape
```

Out[62]:

```
(244, 7)
```

In [63]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 244 entries, 0 to 243
Data columns (total 7 columns):
 #   Column      Non-Null Count  Dtype   
---  ------      --------------  -----   
 0   total_bill  244 non-null    float64 
 1   tip         244 non-null    float64 
 2   sex         244 non-null    category
 3   smoker      244 non-null    category
 4   day         244 non-null    category
 5   time        244 non-null    category
 6   size        244 non-null    int64   
dtypes: category(4), float64(2), int64(1)
memory usage: 7.3 KB
```

In [64]:

```
df.size
```

Out[64]:

```
1708
```

In [65]:

```
df.dtypes
```

Out[65]:

```
total_bill     float64
tip            float64
sex           category
smoker        category
day           category
time          category
size             int64
dtype: object
```

In [66]:

```
df.isna().sum()  #mostra missing data
```

Out[66]:

```
total_bill    0
tip           0
sex           0
smoker        0
day           0
time          0
size          0
dtype: int64
```

In [67]:

```
#mostra os valores únicos das colunas excluindo-se as colunas do tipo float
for column in df.select_dtypes(exclude = float):
    print(df[column].unique())
```

```
[Female, Male]
Categories (2, object): [Female, Male]
[No, Yes]
Categories (2, object): [No, Yes]
[Sun, Sat, Thur, Fri]
Categories (4, object): [Sun, Sat, Thur, Fri]
[Dinner, Lunch]
Categories (2, object): [Dinner, Lunch]
[2 3 4 1 6 5]
```

In [68]:

```
#proporção entre homens e mulheres
df['sex'].value_counts()
```

Out[68]:

```
Male      157
Female     87
Name: sex, dtype: int64
```

In [69]:

```
#proporção entre fumantes e não fumantes
df['smoker'].value_counts()
```

Out[69]:

```
No     151
Yes     93
Name: smoker, dtype: int64
```

In [70]:

```
#cria um agrupamento dos dados por sexo
metrics = [min, max, np.mean, np.std]
grouped_gender = df.groupby(by = 'sex')[['total_bill','tip']].agg(metrics)
```

In [71]:

```
grouped_gender.T
```

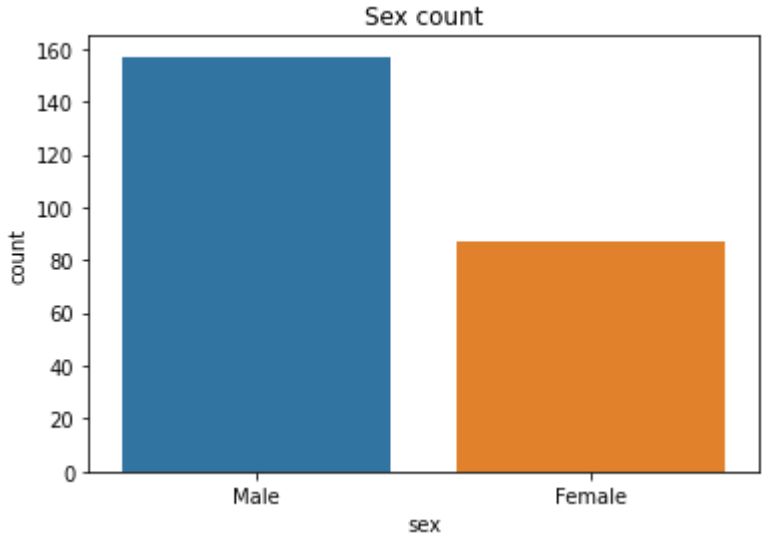
Out[71]:

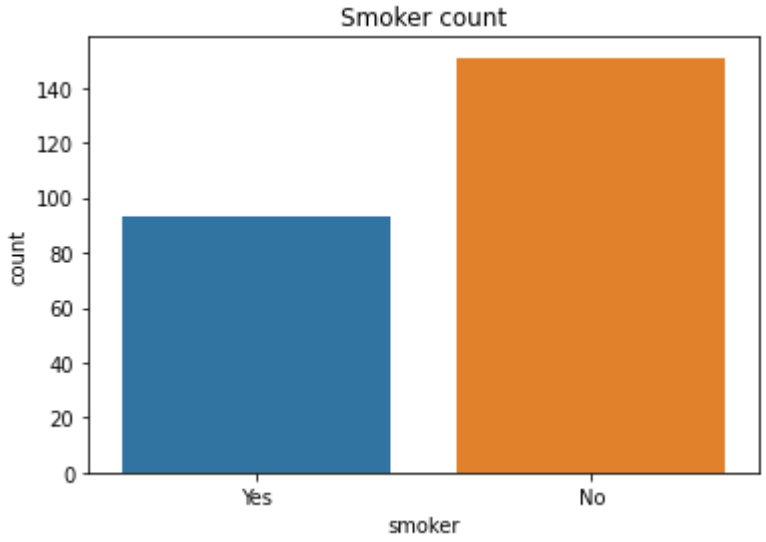
	sex	Male	Female
total_bill	**min**	7.250000	3.070000
	max	50.810000	44.300000
	mean	20.744076	18.056897
	std	9.246469	8.009209
tip	**min**	1.000000	1.000000
	max	10.000000	6.500000
	mean	3.089618	2.833448
	std	1.489102	1.159495

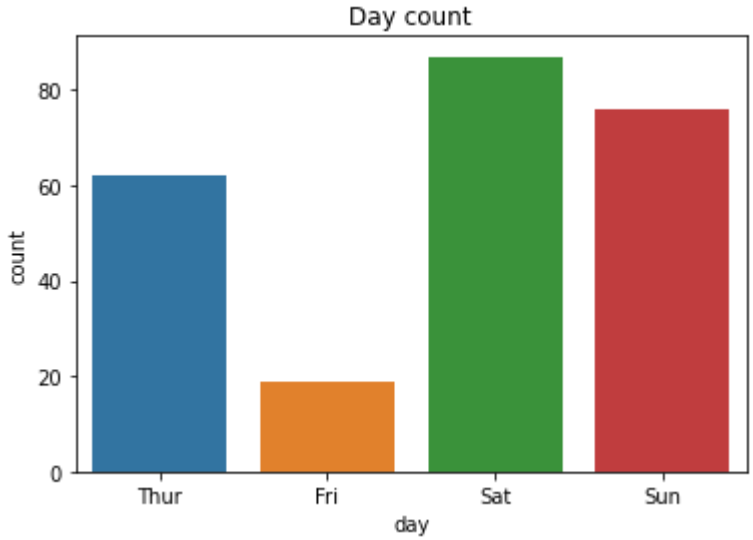
Visualizando os dados

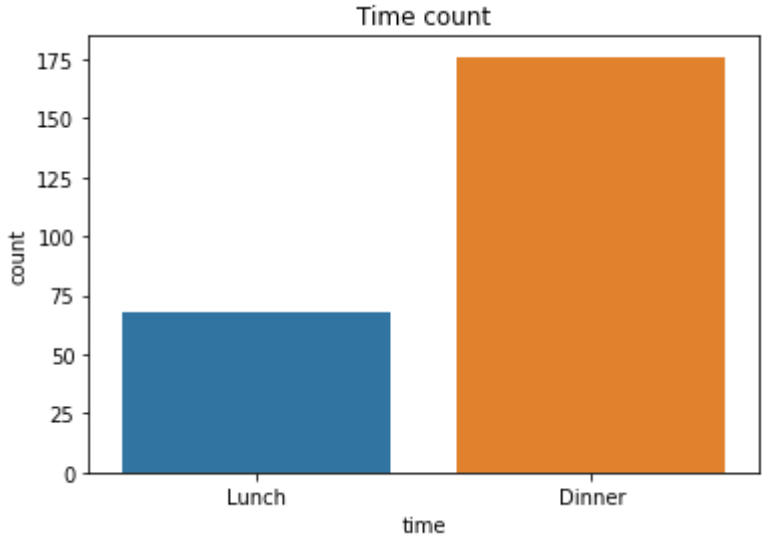
In [72]:

```
#gráfico countplot para as variáveis categóricas
for variable in df.select_dtypes(include = 'category'):
    sns.countplot(x = variable, data = df)
    plt.title(variable.title()+' count')
    plt.show()
```


Sex count
160
140
120
100
80
60
40
20
0
count
Male
Female
sex

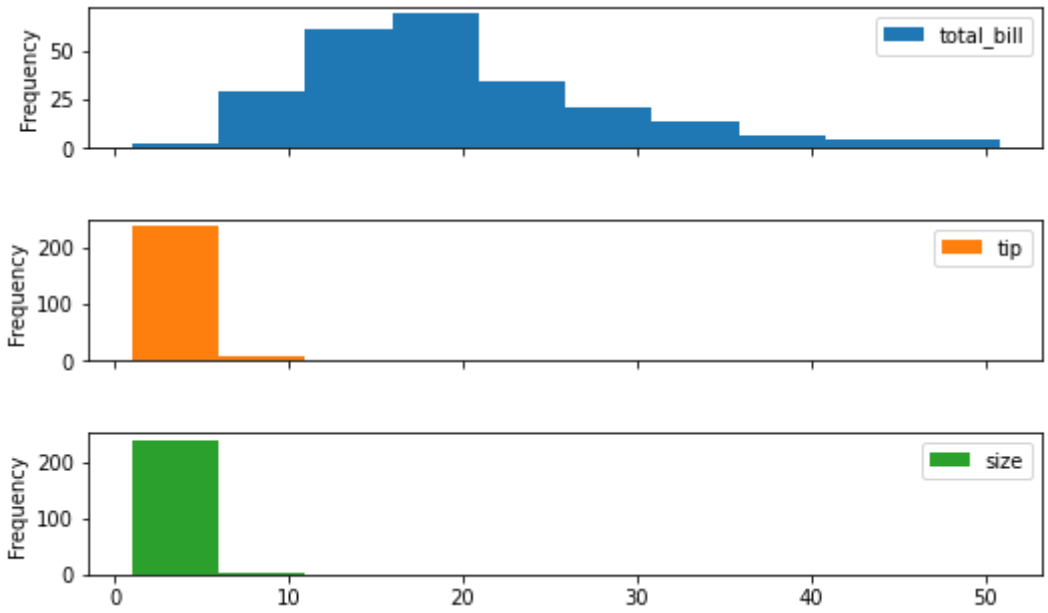

Smoker count
140
120
100
80
60
40
20
0
count
Yes
No
smoker


Day count
count
0
20
40
60
80
Thur
Fri
Sat
Sun
day


Time count
count
0
25
50
75
100
125
150
175
Lunch
Dinner
time

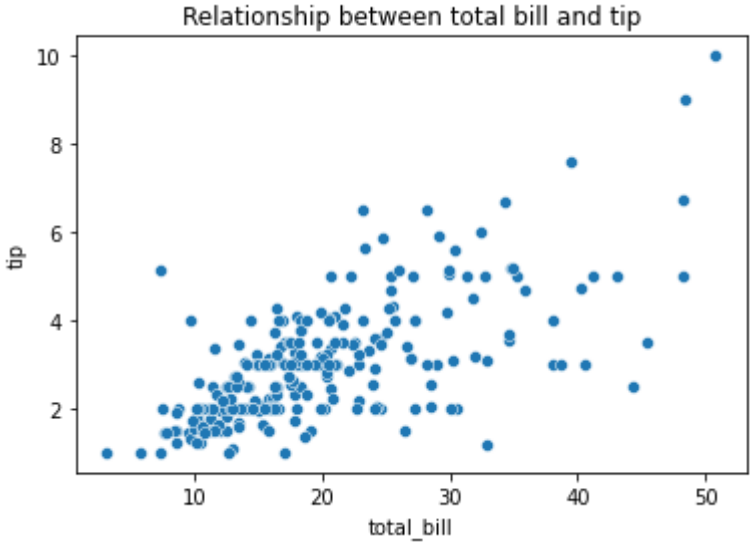
In [73]:

```
#histograma para as variáveis quantitativas
df.plot(kind = 'hist', subplots = True, figsize = (8,5))
plt.tight_layout(pad = 3)
```



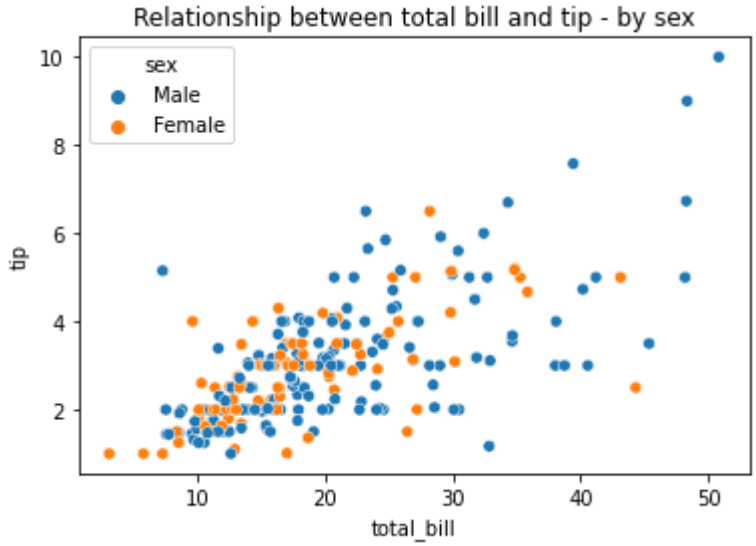
In [74]:

```
#relação entre a variável valor total da conta e valor da gorjeta
sns.scatterplot(x = 'total_bill', y = 'tip', data = df)
plt.title('Relationship between total bill and tip')
plt.show()
```



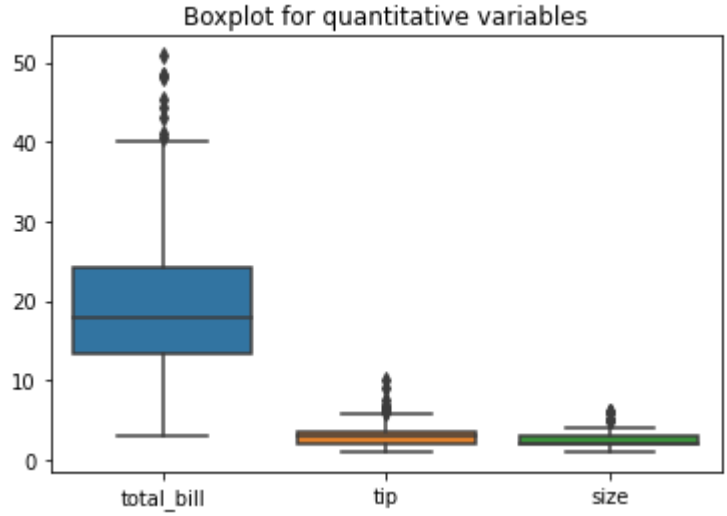
In [75]:

```
#relação entre o total da conta e a gorjeta por gênero
sns.scatterplot(x = 'total_bill', y = 'tip', hue = 'sex', data = df)
plt.title('Relationship between total bill and tip - by sex')
plt.show()
```



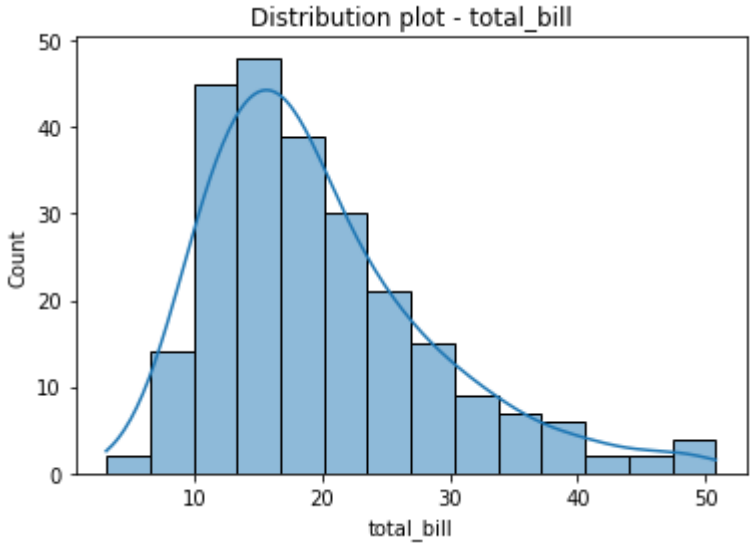
In [76]:

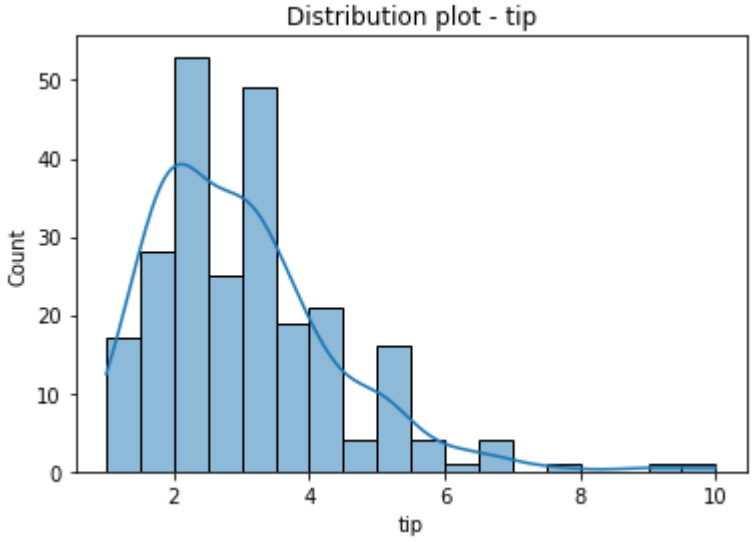
```
#boxplot para as variáveis quantitativas do dataset
sns.boxplot(data = df)
plt.title('Boxplot for quantitative variables')
plt.show()
```



In [77]:

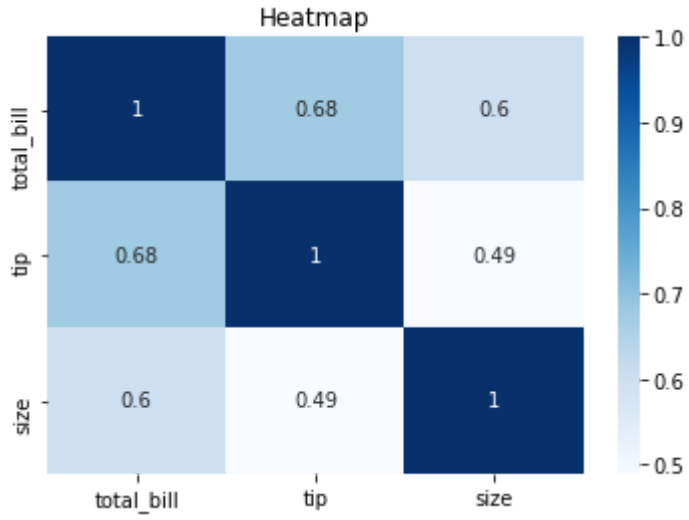
```
#mostra um histograma com a função kde para as variáveis total_bill e tip
for column in df.columns[:2]:
    sns.histplot(df[column], kde = True)
    plt.title('Distribution plot - '+column)
    plt.show()
```





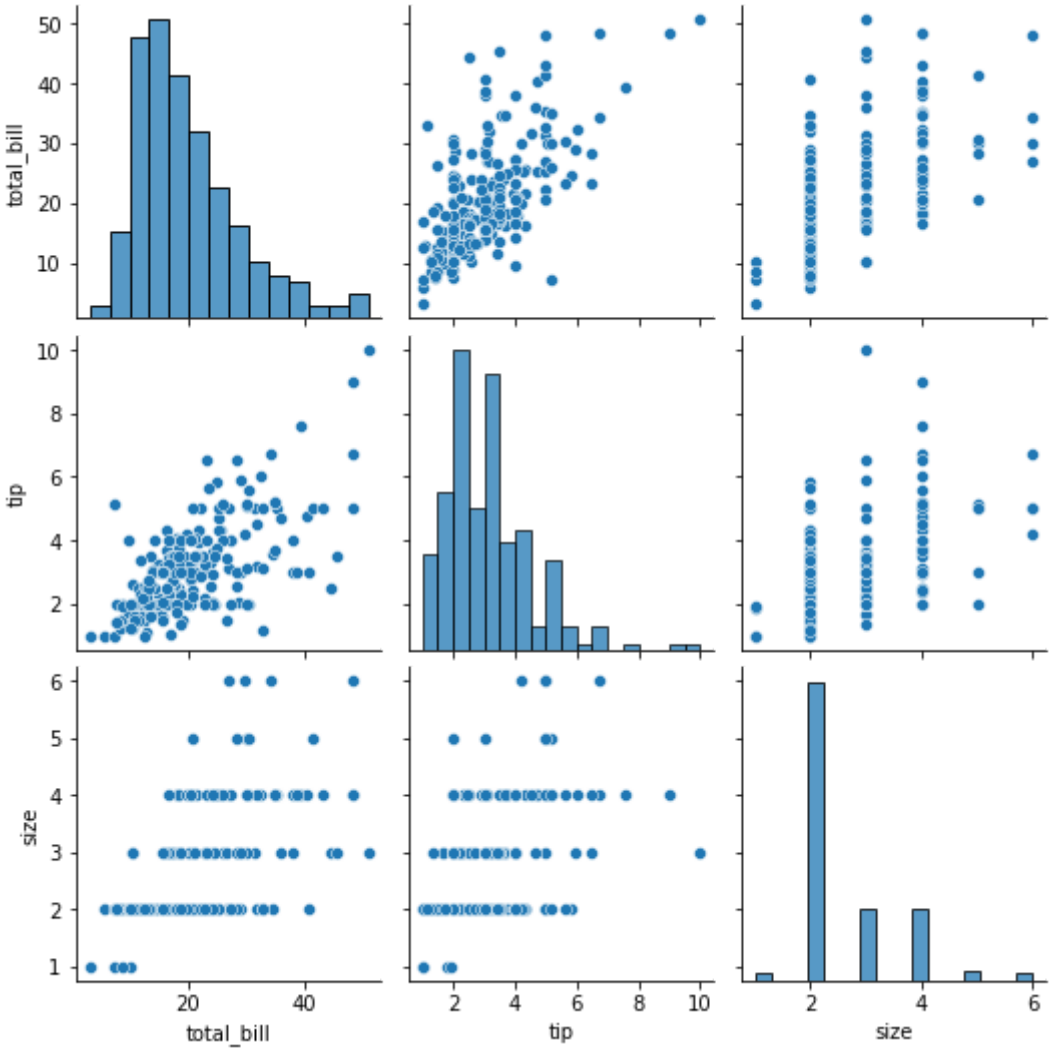
In [78]:

```
#cálcula a matrix de correlação para gerar um mapa de calor
corr = df.corr()
sns.heatmap(corr, cmap = 'Blues', annot = True)
plt.title('Heatmap')
plt.show()
```



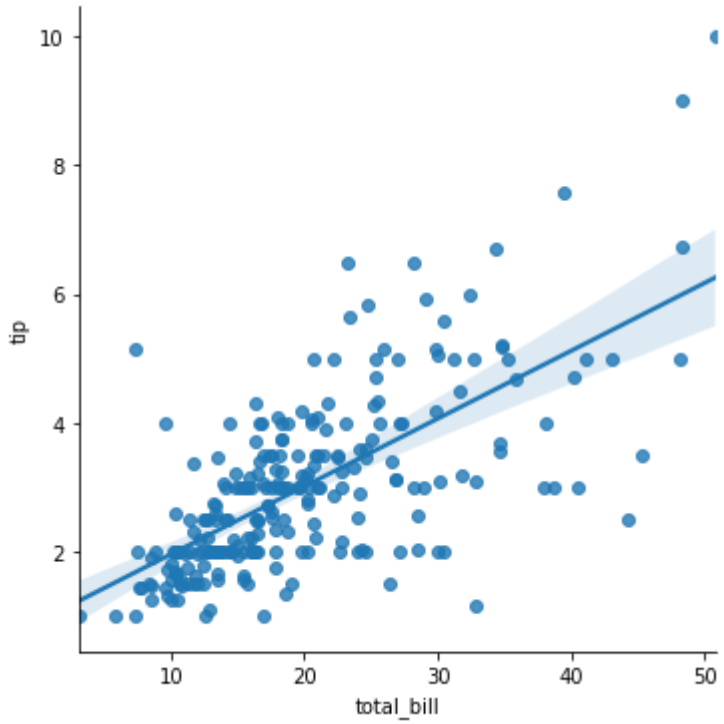
In [79]:

```
sns.pairplot(df);
```


total_bill
tip
size
total_bill
tip
size

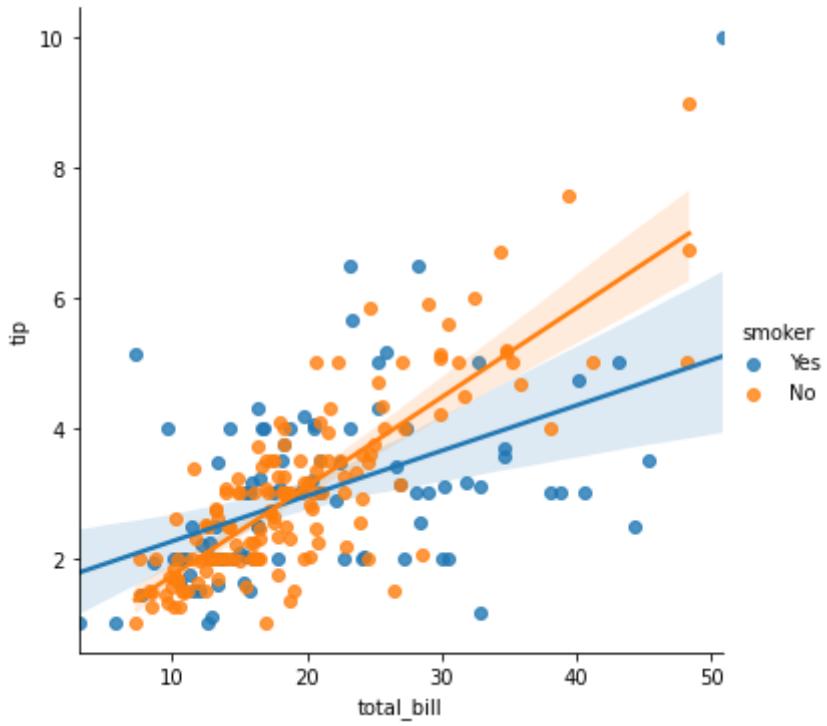
In [80]:

```
#gera uma visualização da relação entre total_bill e tip com uma reta de regress
ão
sns.lmplot(x = 'total_bill', y = 'tip', data = df);
```



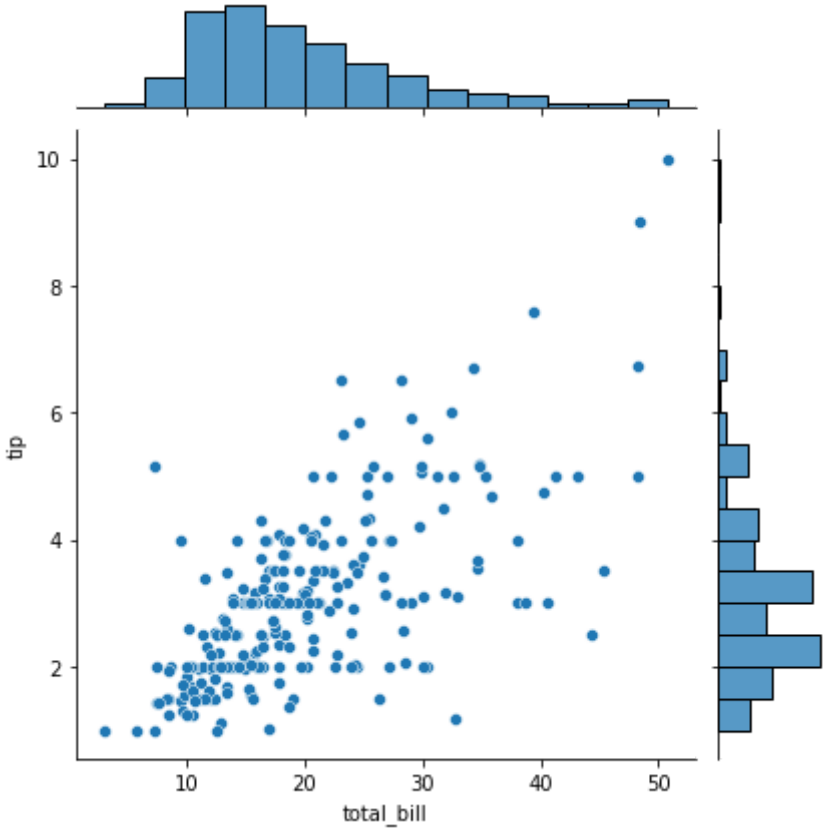
In [81]:

```
#gera uma visualização da relação entre total_bill e tip com uma reta de regress
ão com hue =smoker
sns.lmplot(x = 'total_bill', y = 'tip', data = df, hue = 'smoker');
```



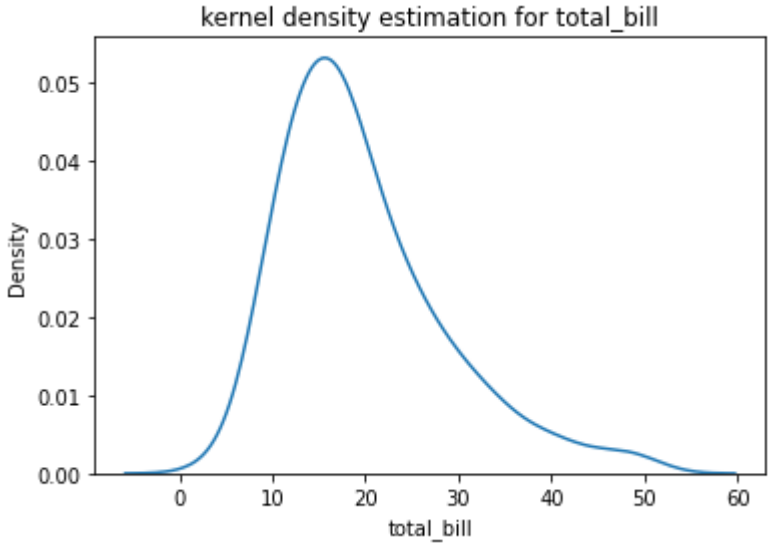
In [82]:

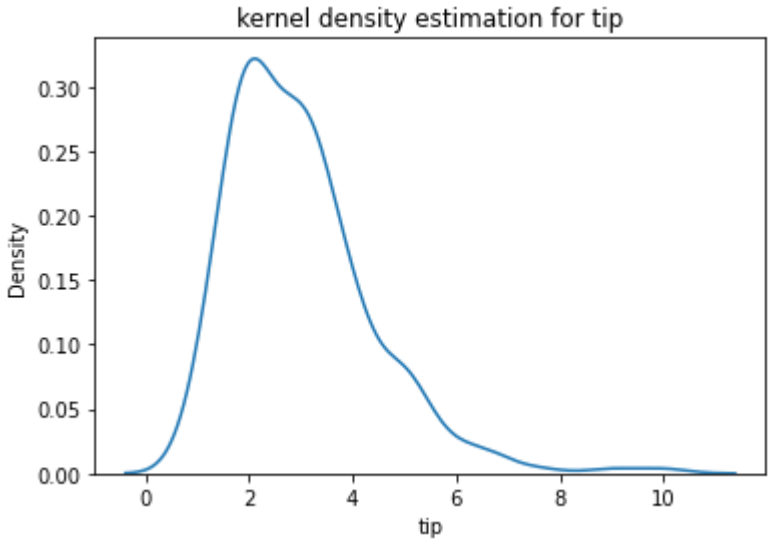
```
#relação entre tip e total_bill e suas respectivas distribuições
sns.jointplot(x = 'total_bill', y = 'tip', data = df);
```



In [83]:

```
#gera um kde para as variáveis quantitativas (distribuição dos dados)
for column in df.columns[:2]:
    sns.kdeplot(x = column , data = df)
    plt.title('kernel density estimation for '+column)
    plt.show()
```



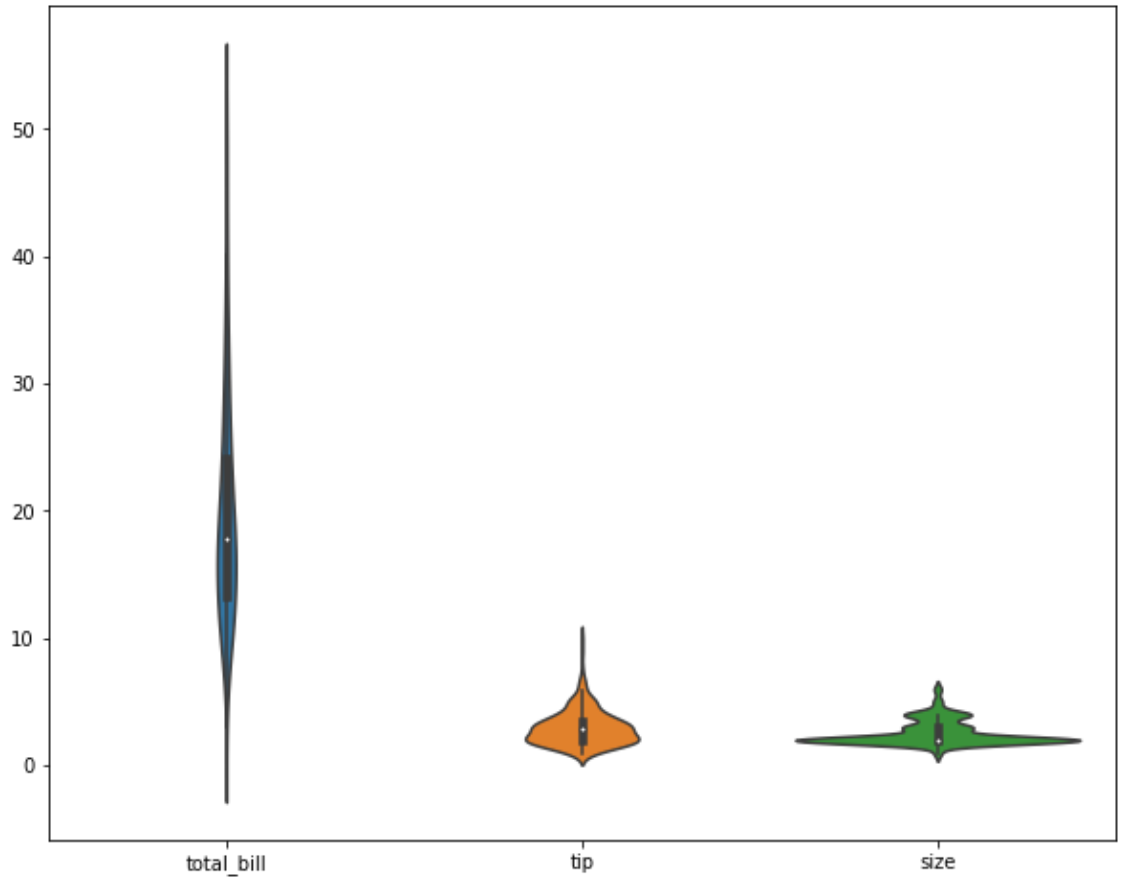


In [84]:

```
#violinplot para as variáveis quantitativas
plt.figure(figsize = (10, 8))
sns.violinplot(data = df)
```

Out[84]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fe8c3409f10>
```



7. Probabilidade Básica, Esperança matemática e Variância

7.1 Probabilidade Básica

Espaço amostral

Definição: para um dado experimento ε (estocástico), definimos o espaço amostral S como o conjunto de todos os resultados possíveis deste experimento.

Evento

Definição: Um evento A é um subconjunto do espaço amostral: $A \subset S$. Se o espaço amostral for finito ou infinito numerável, todo subconjunto poderá ser considerado um evento, tal que se S possui n elementos, então existirão 2^n subconjuntos associados (eventos).Contudo, se o espaço amostral for infinito não-numerável, verifica-se que nem todo subconjunto associado poderá ser considerado um evento deste.

Usando as técnicas da teoria dos conjuntos, se A e B forem eventos, temos que:

1. $A \cup B$ será o evento que ocorrerá se, e somente se, A ou B (ou ambos) ocorrerem;
2. $A \cap B$ será o evento que ocorrerá se, e somente se, A e B ocorrerem;
3. $\overline{A}$ ocorrerá se, e somentese se, o evento A não ocorrer;
4. Sendo $A_1, \ldots A_n$ uma coleção finita de eventos, então o evento $\bigcup_{i=1}^{n} A_i$ ocorrerá se, e somente se, ao menos um dos eventos A_i ocorrer;
5. Sendo $A_1, \ldots A_n$ uma coleção finita de eventos, então o evento $\bigcap_{i=1}^{n} A_i$ ocorrerá se, e somente se todos os eventos A_i ocorrerem;
6. Sendo $A_1, \ldots A_n$ uma coleção infinita (numerável) de eventos, então o evento $\bigcup_{i=1}^{\infty}$ ocorrerá se, e somente se, ao menos um dos eventos A_i ocorrer;
7. Sendo $A_1, \ldots A_n$ uma coleção infinita (numerável) de eventos, então o evento $\bigcap_{i=1}^{\infty} A_i$ ocorrerá se, e somente se, todos os eventos A_i ocorrerem;
8. Seja n a quantidade de repetições de um experimente ε de um espaço amostral S, então o conjunto de todos os possíveis resultados quando ε for executado n vezes será denotado como:
$S \times S \times S \times \cdots \times S = \{(s_1, s_2, \ldots, s_n), s_i \in S, \forall i = i, \ldots, n\}$

Definição: A e B são denominados eventos mutuamente excludentes (ou disjuntos) se $A \cap B = \emptyset$ (não podem ocorrer juntos tal que a interseção entre ambos seja um conjunto vazio).

Adicionalmente, se o evento é entendido como impossível de ocorrer temos então que $P(A) = 0$, mas se A ocorre com certeza então, $P(A) = 1$. Além do que, a probabilidade de um evento impossível ocorrer é tal que $P(\emptyset) = 0$.

De forma simplificada podemos definir a probabilidade de ocorrência de um evento A da seguinte forma:

$$P(A) = \frac{\text{número de vezes em que A ocorre}}{\text{número de vezes em que todos os eventos ocorrem}} = \frac{n_A}{n}$$

Definição frequentista de probabilidade, sendo n o número de vezes em que o experimento é feito:
$P(A) = \lim_{n \to \infty} \frac{n_A}{n}$

Definição:Seja ε um experimento. Seja S um espaço amostral associado a ε . A cada evento A associaremos um número real representado por $P(A)$ e denominado probabilidade de A, que satisfaça as seguintes propriedades:

1. $0 \leq P\left(A\right) \leq 1$
2. $P(S) = 1$
3. Se A e B forem eventos mutuamente excludentes então $P\left(A \cup B\right) = P\left(A\right) + P\left(B\right)$
4. Se $A_1, A_2, \ldots, A_n, \ldots$ forem eventos, dois a dois, eventos mutuamente excludentes então: $P\left(\bigcup_{i=1}^{\infty} A_i\right) = P\left(A_1\right) + P\left(A_2\right) + \ldots + P\left(A_n\right) + \ldots$

7.2 Esperança matemática

Definição: variável aleatória (v.a.) é uma variável que está associada a uma distribuição de probabilidade. Uma v.a. não possui um valor fixo, podendo assumir vários valores.

Esperança de uma variável aleatória discreta

$$\mathrm{E(X) = X_1P(X_1) + X_2P(X_2) + \ldots + X_nP(X_n) = \sum_{i=1}^{n} X_nP(X_n)}$$

Esperança de uma variável aleatória contínua

Uma variável aleatória é contínua se existir uma função $f(x)$ (denominada função densidade de probabilidade) tal que as condições seguintes são satisfeitas:

1. $f(x) \geq 0$ (condição de não negatividade)
2. $\int_{-\infty}^{\infty} f(x)dx = 1$

Definimos ainda a função de distribuição acumulada (ou função de distribuição) $F(X)$, sendo uma soma de todos os valores possíveis até um valor determinado:
$F(X) = \int_{-\infty}^{x} f(dt)dt$

No caso da função de distribuição os seguintes requisitos devem ser satisfeitos:

- $0 \leq F(x) \leq 1$ (soma de probabilidades)
- $\lim_{x \to \infty} F(x) = 1$

Então, se X é uma variável aleatória contínua: $P(a \leq X \leq b) = \int_{a}^{b} f(x)dx$. E sua função de distribuição acumulada é dada por: $\mathrm{P(X \leq x) = \int_{-\infty}^{x} f(t)dt}$. Note que: $\frac{dF(X)}{dx} = F'(x) = f(x)$.

$\mathrm{E(X) = \int_{-\infty}^{\infty} xf(x)dx}$

$\mathrm{E}(\mathrm{X}^2) = \int_{-\infty}^{+\infty} x^2 f(x) dx$

Podemos definir ainda a mediana de uma variável aleatória contínua (valor que divide a distribuição em duas) como, sendo m a mediana: $P(x > m) = \int_{m}^{+\infty} f(x) dx = 0.5$ ou
$P(x < m) = \int_{-\infty}^{m} f(x) dx = 0.5.$

Propriedades

Seja $\mathrm{E}(\mathrm{X}), \mathrm{E}(\mathrm{Y})$ e $\forall \alpha, \beta \in \mathbb{R}$:

- $\mathrm{E}(\alpha \mathrm{X}) = \alpha \mathrm{E}(\mathrm{X})$
- $\mathrm{E}(\mathrm{X} + \alpha) = \mathrm{E}(\mathrm{X}) + \alpha$
- $\mathrm{E}(\mathrm{X} + \mathrm{Y}) = \mathrm{E}(\mathrm{X}) + \mathrm{E}(\mathrm{Y})$
- $\mathrm{E}(\alpha \mathrm{X} \pm \beta \mathrm{Y}) = \alpha \mathrm{E}(\mathrm{X}) \pm \beta \mathrm{E}(\mathrm{Y})$
- $\mathrm{E}(\mathrm{XY}) = \mathrm{E}(\mathrm{X}) \cdot \mathrm{E}(\mathrm{Y})$, se X,Y são independentes
- $\mathrm{E}(\mathrm{XY}) = \mathrm{E}(\mathrm{X}) \cdot \mathrm{E}(\mathrm{Y}) + \operatorname{cov}(\mathrm{X}, \mathrm{Y})$

7.3 Variância

Lembremos que a variância é definida em termos da média dos quadrados dos desvios em relação a média. Podemos obter ainda uma outra relação para a variância: média dos quadrados menos o quadrado da média.

$$\begin{aligned}
\operatorname{Var}(\mathrm{X}) &= \mathrm{E}(\mathrm{X} - \mathrm{E}(\mathrm{X}))^2 \\
&= \mathrm{E}\left(X^2 - 2X\mathrm{E}(\mathrm{X}) + (\mathrm{E}(\mathrm{X}))^2\right) \\
&= \mathrm{E}(\mathrm{X}^2) - 2(\mathrm{E}(\mathrm{X}))^2 + (\mathrm{E}(\mathrm{X}))^2 \\
&= \mathrm{E}(\mathrm{X}^2) - (\mathrm{E}(\mathrm{X}))^2
\end{aligned}$$

$\therefore \sigma^2 = Var(X) = E(X - \mu)^2$
$Var(X) = E(X^2) - (E(X))^2$

Propriedades:

$\forall \alpha, \beta \in \mathbb{R}$:

- $\mathrm{Var}(\alpha) = 0$
- $\mathrm{Var}(\mathrm{X} + \alpha) = \mathrm{Var}(\mathrm{X})$
- $\mathrm{Var}(\alpha \mathrm{X}) = \alpha^2 \mathrm{Var}(\mathrm{X})$
- $\mathrm{Var}(\alpha \mathrm{X} + \beta) = \alpha^2 \mathrm{Var}(\mathrm{X})$
- $\mathrm{Var}(\mathrm{X} + \mathrm{Y}) = \mathrm{Var}(\mathrm{X}) + \mathrm{Var}(\mathrm{Y}) + 2\mathrm{Cov}(\mathrm{X}, \mathrm{Y})$
- $\mathrm{Var}(\mathrm{X} - \mathrm{Y}) = \mathrm{Var}(\mathrm{X}) + \mathrm{Var}(\mathrm{Y}) - 2\mathrm{Cov}(\mathrm{X}, \mathrm{Y})$
- $\mathrm{Var}(\alpha \mathrm{X} \pm \beta \mathrm{Y}) = \alpha^2 \mathrm{Var}(\mathrm{X}) + \beta^2 \mathrm{Var}(\mathrm{Y})$, se X e Y forem independetes
- $\mathrm{Var}(\alpha \mathrm{X} \pm \beta \mathrm{Y}) = \alpha^2 \mathrm{Var}(\mathrm{X}) + \beta^2 \mathrm{Var}(\mathrm{Y}) \pm 2\alpha\beta \mathrm{Cov}(\mathrm{X}, \mathrm{Y})$
- $\mathrm{Var}(\mathrm{XY}) = \mathrm{E}\left(X^2 Y^2\right) - (\mathrm{E}(\mathrm{XY}))^2$

8. Distribuições de probabilidade e suas características

Nesta seção aprenderemos sobre as principais distribuições de probabilidade e ainda algumas implementações em Python.

8.1 Distribuições Discretas

8.1.1 Distribuição de Bernouilli

Características/uso

- Caracteriza-se pela existência de dois eventos, sucesso e fracasso, realizados uma única vez, com probabalidade de sucessso p e probabilidade de fracasso $(1-p)$.
- Exemplo: lançamento de uma moeda uma única vez. A probabilidade de sucesso (cara) é $p = 1/2$ e a probabilidade de fracasso (coroa) $(1-p) = 1/2$ (Mostre que neste caso, para $p = 1$: $E(X) = 0.5, Var(X) = 0.25$).
  - $E(X) = 1 \times p + 0 \times (1-p) = p$
  - $E(X^2) = 1^2 \times p + 0^2 \times (1-p) = p$
  - $Varx(X) = 1/2 - (1/2)^2 = 1/4$
- O lançamento de um dado: a aposta em um determinado número implica em uma probabilidade de sucesso $p = 1/6$ e que a probabilidade de fracasso seja $(1-p) = 5/6$
- Uso: sucesso/fracasso.

Forma geral $P(X = k)$

$$P(X = k) = p^k (1-p)^{1-k}, k = 0, 1$$

Esperança

$$\mathrm{E}(\mathrm{X}) = 1 \times p + 0 \times (1-p) = p$$

$$\mathrm{E}(\mathrm{X}) = p$$

Variância

$$\mathrm{E}(\mathrm{X}^2) = X_1^2 P(X_1) + X_2^2 P(X_2)$$
$$\mathrm{E}(\mathrm{X}^2) = 1^2 \times p + 0^2 \times (1-p) = p$$
$$\mathrm{Var}(\mathrm{X}) = \mathrm{E}(\mathrm{X}^2) - [\mathrm{E}(\mathrm{X})]^2$$
$$\mathrm{Var}(\mathrm{X}) = p - p^2$$

$$\mathrm{Var}(\mathrm{X}) = p(1-p)$$

8.1.2 Distribuição Binomial

Características/uso

- Generalização da distribuição de Bernouilli: p é denotada como a probabilidade de sucesso, $(1-p)$ a probabilidade de fracasso, para n experimentos.

Então, para um determinado experimento, com p probabilidade de sucesso e $(1-p)$ probabilidade de fracasso, a probabilidade de que, em n repetições, obtenhamos um número k de sucesos é dada por :

$$P(X = k) = \binom{n}{k} p^k (1-p)^{n-k}$$

Esperança

$$E(X) = np$$

Variância

$$Var(X) = np(1-p)$$

Vamos utilizar o NumPy para gerar números pseudo-aleatórios de uma distribuição normal. Usaremos a função np.random.binomial() passando três argumentos: n = número de tentativas, p = probabilidade de ocorrência de cada tentativa e size = shape do array.

In [85]:

```
import numpy as np
array = np.random.binomial(n = 5, p = 0.5, size = 5)
array
```

Out[85]:

```
array([3, 3, 3, 2, 4])
```

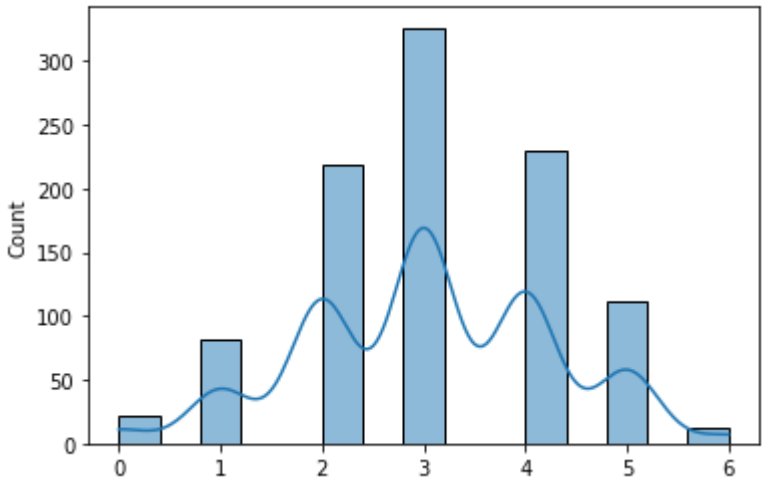
Vamos gerar uma visualização com o Seaborn:

In [86]:

```
import seaborn as sns
sns.histplot(np.random.binomial(n = 6, p = 0.5, size = 1000), kde = True)
```

Out[86]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fe8c57d6d60>
```



8.2 Distribuições contínuas

8.2.1 Distribuição Uniforme

Uma variável aleatória contínua é dita uniforme no intervalo $[a, b]$, se sua f.d.p. é definida como, tal que $X \sim U([a, b])$:

$$f(x) = \begin{cases} k, & \text{se } a \leq x \leq b \\ 0, & \text{caso contrário} \end{cases}$$

Para encontrar o valor de k aplicamos a condição $\int_{-\infty}^{\infty} f(x)dx = 1$ de variável aleatória:

$$\int_a^b kdx = 1 \Rightarrow k\int_a^b dx = 1 \Rightarrow kx|_a^b = 1$$

Então:

$$k(b-a) = 1 \Rightarrow k = \frac{1}{b-a}$$

Logo:

$$f(x) = \begin{cases} \frac{1}{b-a}, & \text{se } a \leq x \leq b \\ 0, & \text{caso contrário} \end{cases}$$

Forma geral $P(X = k)$

$$f(x) = \frac{1}{b-a}, \forall x \in [a,b]$$

Esperança

$$E(X) = \frac{a+b}{2}$$

Variância

$$Var(X) = \frac{(b-a)^2}{12}$$

Demonstrações:

$$\begin{aligned}
\mathrm{E(X)} &= \int_a^b \frac{1}{b-a} x dx \\
&= \frac{1}{b-a} \frac{x^2}{2}|_a^b \\
&= \frac{1}{b-a} \left(\frac{b^2}{2} - \frac{a^2}{2} \right) \\
&= \frac{1}{b-a} \left(\frac{b^2 - a^2}{2} \right) \\
&= \frac{(b-a)(b+a)}{(b-2)2} \\
&= \frac{a+b}{2}
\end{aligned}$$

$$\begin{aligned}
\mathrm{E(X^2)} &= \int_a^b \frac{1}{b-a} x^2 dx \\
&= \frac{1}{b-a} \frac{x^3}{3}|_a^b \\
&= \frac{1}{3(b-a)} (b^3 - a^3)
\end{aligned}$$

$$\begin{aligned}\mathrm{Var}(\mathrm{X}) &= \frac{b^3 - a^3}{3(b-a)} - \left(\frac{a+b}{2}\right)^2 \\ &= \frac{(b-a)(b^2+ab+a^2)}{3(b-a)} - \frac{a^2+2ab+b^2}{4} \\ &= \frac{b^2+ab+a^2}{3} - \frac{a^2+2ab+b^2}{4} \\ &= \frac{(b-a)^2}{12}\end{aligned}$$

Utilizaremos o SymPy para obter a esperança, a variância e a f.d.p. da distribuição uniforme.

In [87]:

```
from sympy.stats import Uniform, density, cdf, E, variance
from sympy import Symbol, simplify
a = Symbol("a", negative=True)
b = Symbol("b", positive=True)
z = Symbol("z")
X = Uniform('x',a ,b)
```

In [88]:

```
density(X)(z)
```

Out[88]:

$$\begin{cases} \frac{1}{-a+b} & \text{for } b \geq z \wedge a \leq z \\ 0 & \text{otherwise} \end{cases}$$

In [89]:

```
cdf(X)(z)
```

Out[89]:

$$\begin{cases} 0 & \text{for } a > z \\ \frac{-a+z}{-a+b} & \text{for } b \geq z \\ 1 & \text{otherwise} \end{cases}$$

In [90]:

```
E(X)
```

Out[90]:

$$\frac{a}{2} + \frac{b}{2}$$

In [91]:

```
simplify(variance(X))
```

Out[91]:

$$\frac{a^2}{12} - \frac{ab}{6} + \frac{b^2}{12}$$

Podemos ainda utilizar o NumPy para gerar números pseudo-aleatórios de uma distribuição uniforme.

In [92]:

```
array = np.random.uniform(low = 1, high = 10, size = 30)
array
```

Out[92]:

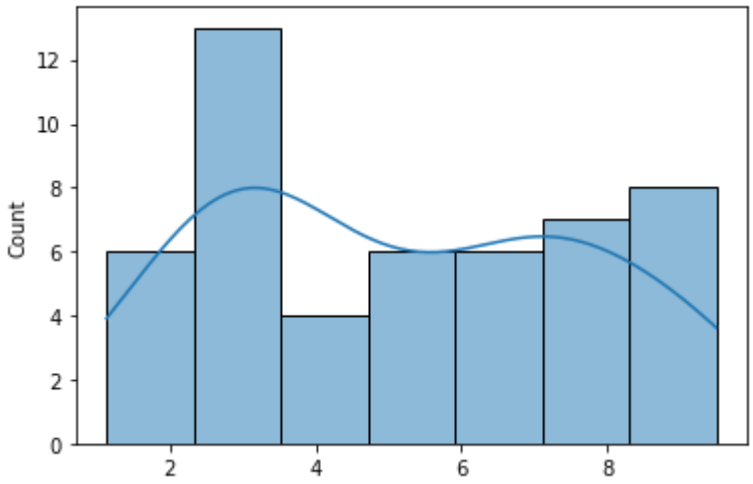
```
array([1.70051754, 9.14051453, 5.66191774, 8.12472904, 4.35112404,
       9.77965791, 8.99033222, 6.62542548, 9.1750991 , 6.73588222,
       9.7822474 , 5.94254803, 8.66033611, 4.49329235, 4.67967715,
       7.17923399, 6.06031922, 2.78090449, 5.73939916, 6.54940214,
       4.72728031, 5.15426881, 3.79478593, 5.55416298, 3.55309247,
       3.56827561, 2.62259299, 1.36108578, 9.6105116 , 2.34279447])
```

In [93]:

```
sns.histplot(np.random.uniform(low = 1, high = 10, size = 50), kde = True)
```

Out[93]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fe8c1fda5e0>
```



8.2.2 Distribuição Normal

Uma distribuição $X \sim N\left(\mu, \sigma\right)$ com as seguintes propriedades:

- A função densidade de probabilidade $f_x\left(x\right)$ tem ponto máximo em $x = \mu$.
- $\mu + \sigma$ e $\mu - \sigma$ são pontos de inflexão da curva.
- A área total da curva vale 1.
- A área é simétrica em relação a μ.
- $E(X) = \mu$ e $Var(X) = \sigma^2$
- Uma distribuição normal padrão é tal que : $Z = \frac{X - \mu}{\sigma}$, $Z \sim N(0, 1)$.

Gerando números de uma distribuição normal com NumPy (função np.random.normal()):

- loc = média;
- scale = desvio padrão;
- size = shape do array.

In [94]:

```
array = np.random.normal(loc = 10, scale = 0.5, size = 50)
array
```

Out[94]:

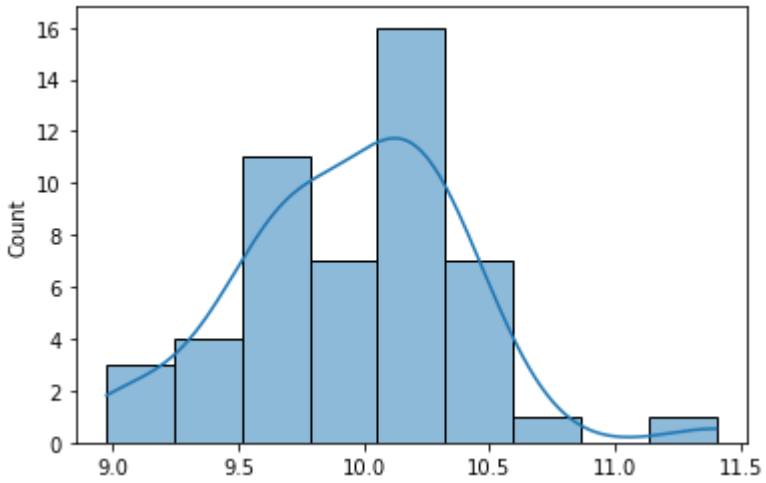
```
array([10.16488191, 10.07134778,  9.71211947,  9.13758363,  9.99774
089,
        9.9476879 , 10.57897572, 10.00062114,  9.87616404, 10.07567
768,
       10.29154722,  9.49328223,  9.9323229 , 10.78718199,  9.78798
749,
        9.18473931, 10.45454663, 10.65997257, 10.62030884,  8.95483
633,
       10.04292199,  9.01013591, 10.47979769,  9.67750639,  9.08122
648,
        9.76759592,  9.79215003,  9.92864263, 10.8919025 , 10.25730
919,
       10.1155218 ,  9.47057514, 10.05307782, 10.83462851,  9.26974
833,
       10.03240943, 10.29105909, 10.62083566, 10.18505043,  9.47474
36 ,
       10.20183784, 10.43109523,  9.67564783,  9.7616791 , 10.92236
703,
        9.7096689 , 10.35320351, 10.12330367, 10.17889922,  9.61432
017])
```

In [95]:

```
sns.histplot(np.random.normal(loc = 10, scale = 0.5, size = 50), kde = True)
```

Out[95]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fe8c1fa9190>
```



8.2.3 Distribuição Exponencial

Forma geral $P(X = k)$

Uma variável X tem distribuição de probablidade exponencial com $\beta > 0$ se, e somente se, sua f.d.p. é dada por, $X \sim \text{Exp}\,(\beta)$:

$$f(x) = \begin{cases} \frac{1}{\beta} e^{(-\frac{x}{\beta})}, \ \forall x \geq 0 \\ 0, \ \ \text{caso contrário} \end{cases}$$

Esperança: $\text{E(X)} = \beta$
Variância: $\text{Var(X)} = \beta^2$

Ou ainda se:

$$f(x) = \begin{cases} \beta e^{-\beta x}, \ \forall x \geq 0 \\ 0, \ \ \text{caso contrário} \end{cases}$$

Esperança: $\text{E(X)} = \frac{1}{\beta}$
Variância: $\text{Var(X)} = \frac{1}{\beta^2}$

Informando o parâmetro β (scale) com a função np.random.exponential() podemos gerar números de uma distribuição exponencial.

In [96]:

```
array = np.random.exponential(scale = 5, size = 50)
array
```

Out[96]:

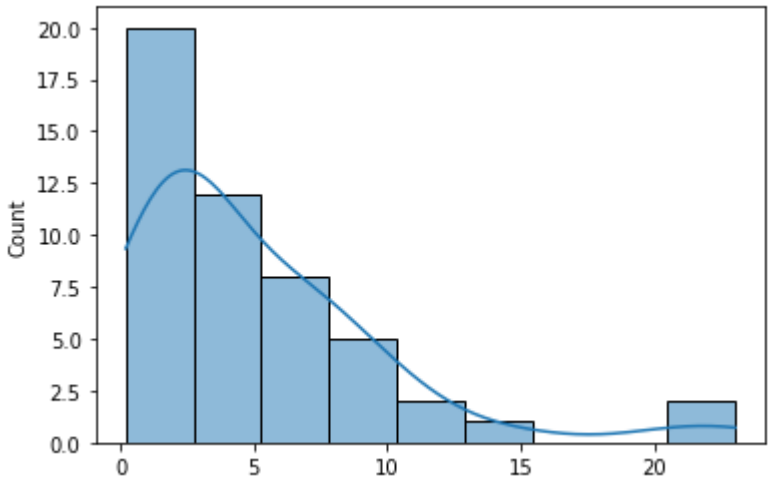
```
array([ 3.0512069 ,  1.81694768,  7.09290529,  6.61185449,  8.75196
856,
        7.4484647 , 20.64646311,  8.1174591 ,  4.98695122,  0.56407
904,
        0.19633046,  2.76120554,  7.84546593,  6.75612473,  2.74776
052,
        5.14526867,  1.28027092,  0.39675312,  0.95541726, 11.18944
762,
        2.19175293, 10.04662866,  9.82936648,  3.46452606, 23.02641
838,
        6.45491322,  7.4392286 ,  1.46959052,  1.59130211,  2.41400
363,
        6.13647238,  2.0488292 ,  1.8402891 ,  2.11677134,  1.32641
118,
        2.86064567,  6.86565991, 10.72566649,  3.23310994,  5.20019
58 ,
        3.51198655,  1.11885416,  1.98445759,  2.98506243,  2.05491
449,
        4.2923172 ,  2.40056833,  0.69650479, 14.58164523,  0.89249
172])
```

In [97]:

```
sns.histplot(array, kde = True)
```

Out[97]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fe8c1eb94c0>
```



8.2.4 Distribuição Lognormal

Uma variável aleatória contínua possui distribuição lognormal se sua f.d.p. é dada:

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{(\ln(x) - \mu)^2}{2\sigma^2}\right], \forall x > 0$$

Dizemos que uma variável aleatória X tem distribuição lognormal se, e somente se, o seu logaritmo tem distribuição normal (isto é, uma outra distribuição igual a $Y = \log X$, normalmente distribuída), com parâmetros μ (valor esperado) e σ^2 (variância).Usamos a notação:
$X \sim \text{Lognormal}\left(\mu, \sigma^2\right), -\infty < \mu < +\infty, \sigma > 0.$

$\mathrm{E(X)} = e^{\mu+\sigma^2/2}$

$\mathrm{Var(X)} = e^{2(\mu+\sigma^2)} - e^{2\mu+\sigma^2}$

$Md(X) = e^{\mu}$
$Mo(X) = e^{\mu-\sigma^2}$

$E(X) > Md(X) > Mo(X) \Rightarrow$ distribuição assimétrica positiva ou assimétrica à direita

Com a função np.random.lognormal() podemos gerar números de uma distribuição lognormal, passando os argumentos mean (média), sigma(desvio padrão) e size (shape do array).

In [98]:

```
array = np.random.lognormal(mean = 5, sigma = 0.5, size = 50)
array
```

Out[98]:

```
array([165.18758063, 103.9454573 , 132.53598479,  83.57854739,
       145.08825034, 443.4398576 , 133.61621699,  73.93377252,
       151.21657883, 154.81871481, 177.88525982, 110.76044328,
       271.53478022,  55.71199361, 120.88405576,  82.76489646,
       136.7485879 , 124.4277724 , 198.29844657, 181.12103206,
       167.1479835 , 163.03845343, 180.73831876, 103.42037527,
       135.15094439,  99.96173651,  67.56890388, 231.43844631,
       269.95982628,  94.61191334, 152.69841654, 147.31145202,
       181.07604473, 140.66124934,  85.09853318, 117.70158158,
       135.13853879, 144.61356572, 114.07566349, 273.71608811,
        96.36270861, 104.70090826, 184.80142844,  93.50708162,
       162.66138203,  92.07225871, 175.13307877, 228.59042995,
       108.87637157, 383.10919249])
```

9. Hands On!

Com base no DataFrame abaixo responda:

In [99]:

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

np.random.seed(20)
n = 2_000  #size
data = {
    'A': np.random.normal(loc = 100, scale = 1, size = n),
    'B': np.random.normal(loc = 100, scale = 2, size = n),
    'C': np.random.normal(loc = 100, scale = 5, size = n),
    'D': np.random.lognormal(mean = 2, sigma = 1.1, size = n),
    'E': np.random.lognormal(mean = 2, sigma = 1.2, size = n),
    'F': np.random.lognormal(mean = 2, sigma = 1.3, size = n),
    'G': np.random.randn(n),
    'H': np.random.binomial(n = 10, p = 0.8, size = n)
}

df = pd.DataFrame(data)
```

1. Use a função sns.kdeplot() para demonstrar em um único plot a distribuição das variáveis 'A', 'B' e 'C', que seguem uma distribuição normal.
2. Use a função sns.kdeplot() para demonstrar em um único plot a distribuição das variáveis 'D', 'E' e 'F', que seguem uma distribuição lognormal.
3. Use a função sns.joinplot() para demonstrar a relação entre as variáveis 'A' e 'B',evidenciando suas respectivas distribuições. Repita o mesmo procedimento para as variáveis 'D' e 'F'.
4. Elabore um histograma individual para as variáveis do dataset.
5. Elabore um violinplot para as variáveis 'G' e 'H', respectivamente. Este gráfico consiste em uma combinação do boxplot com o kernel density estimate (kde).

1. Gere uma distribuição normal aleatória de ordem 20 x 5, com média 10 e desvio padrão 3. Armazene o resultado em uma variável. Em seguida crie um DataFrame a partir dos dados criados.

1. Gere uma distribuição normal padrão e utilize o módulo Seaborn para construir uma visualização dos dados. A amostra deve conter 3000 números.

10. Demonstrações

$$\mathrm{Var}(\mathrm{aX}) = a^2\mathrm{Var}(\mathrm{X})$$

$$\begin{aligned}
\text{Var(aX)} &= \frac{1}{n}\sum_{i=1}^{n}\left(aX_i - a\overline{X}\right)^2 \\
&= \frac{1}{n}\sum_{i=1}^{n}\left[a\left(X_i - \overline{X}\right)\right]^2 \\
&= \frac{1}{n}\sum_{i=1}^{n}a^2\left(X_i - \overline{X}\right)^2 \\
&= a^2\frac{1}{n}\sum_{i=1}^{n}\left(X_i - \overline{X}\right)^2 \\
&= a^2\text{Var(X)}
\end{aligned}$$

$$\text{Var(X + a)} = \text{Var(X)}$$

$$\begin{aligned}
\text{Var(X + a)} &= \frac{1}{n}\sum_{i=1}^{n}\left[X_i + a - \left(\overline{X} + a\right)\right]^2 \\
&= \frac{1}{n}\sum_{i=1}^{n}\left[X_i + a - a + \overline{X}\right]^2 \\
&= \frac{1}{n}\sum_{i=1}^{n}\left(X_i - \overline{X}\right)^2
\end{aligned}$$

$$\text{Var(X + Y)} = \text{Var(X)} + \text{Var(Y)} + 2\text{Cov(X, Y)}$$

$$\begin{aligned}
\text{Var(X + Y)} &= \frac{1}{n}\sum_{i=1}^{n}\left(X_i + Y_i\right)^2 - \left(\overline{X} + \overline{Y}\right)^2 \\
&= \frac{1}{n}\sum_{i=1}^{n}\left(X_i^2 + 2X_iY_i + Y_i^2\right) + \left(\overline{X}^2 + 2\overline{X}\overline{Y} + \overline{X}^2\right) \\
&= \frac{1}{n}\sum_{i=1}^{n}\left(X_i^2 + Y_i^2 + 2X_iY_i\right) - \left(\overline{X}^2 + \overline{Y}^2 + 2\overline{X}\overline{Y}\right) \\
&= \left(\frac{1}{n}\sum_{i=1}^{n}X_i^2 - \overline{X}^2\right) + \left(\frac{1}{n}\sum_{i=1}^{n}Y_i^2 - \overline{Y}^2\right) + 2\left(\frac{1}{n}\sum_{i=1}^{n}X_iY_i - \overline{X}\overline{Y}\right) \\
&= \text{Var(X)} + \text{Var(Y)} + 2\text{Cov(X, Y)}
\end{aligned}$$

$$\mathrm{Var}(\mathrm{X}-\mathrm{Y}) = \mathrm{Var}(\mathrm{X}) + \mathrm{Var}(\mathrm{Y}) - 2\mathrm{Cov}(\mathrm{X},\mathrm{Y})$$

$$\begin{aligned}\mathrm{Var}(\mathrm{X}-\mathrm{Y}) &= \mathrm{Var}\left[\mathrm{X}+(-\mathrm{Y})\right] \\ &= \mathrm{Var}(\mathrm{X}) + \mathrm{Var}(-\mathrm{Y}) + 2\mathrm{Cov}(\mathrm{X},-\mathrm{Y}) \\ &= \mathrm{Var}(\mathrm{X}) + \mathrm{Var}(\mathrm{Y}) - 2\mathrm{Cov}(\mathrm{X},\mathrm{Y})\end{aligned}$$

$$\mathrm{Var}(\mathrm{aX}+\mathrm{bY}) = a^2\mathrm{Var}(\mathrm{X}) + b^2\mathrm{Var}(\mathrm{Y}) - 2ab\mathrm{Cov}(\mathrm{X},\mathrm{Y})$$

$$\begin{aligned}\mathrm{Var}(\mathrm{aX}+\mathrm{bY}) &= \mathrm{E}\left[(aX+bY)^2\right] - \left[\mathrm{E}(aX+bY)\right]^2 \\ &= \mathrm{E}\left[a^2X^2 + 2abXY + b^2Y^2\right] - \left[\mathrm{E}(aX)+\mathrm{E}(bY)\right]^2 \\ &= a^2\mathrm{E}(X^2) + 2ab\mathrm{E}(XY) + b^2\mathrm{E}(Y^2) - \left[a\mathrm{E}(X)+b\mathrm{E}(Y)\right]^2 \\ &= a^2\mathrm{E}(X^2) + 2ab\mathrm{E}(XY) + b^2\mathrm{E}(Y^2) - \left[a^2\left[\mathrm{E}(X)\right]^2 + 2ab\mathrm{E}(X)\mathrm{E}(Y) + b^2\right. \\ &= a^2\mathrm{E}(X^2) + 2ab\mathrm{E}(XY) + b^2\mathrm{E}(Y^2) - a^2\left[\mathrm{E}(X)\right]^2 - 2ab\mathrm{E}(X)\mathrm{E}(Y) - b^2\left[\right. \\ &= a^2\left\{\mathrm{E}(X^2) - \left[\mathrm{E}(X)\right]^2\right\} + b^2\left\{\mathrm{E}(Y^2) - \left[\mathrm{E}(Y)\right]^2\right\} + 2ab\left\{\mathrm{E}(XY) - \mathrm{E}(\right. \\ &= a^2\mathrm{Var}(\mathrm{X}) + b^2\mathrm{Var}(\mathrm{Y}) - 2ab\mathrm{Cov}(\mathrm{X},\mathrm{Y})\end{aligned}$$

$$\mathrm{Cov}(\mathrm{X},\mathrm{Y}) = \frac{1}{n}\sum_{i=1}^{n}(X_i-\overline{X})(Y_i-\overline{Y})$$

$$\begin{aligned}\mathrm{Cov}(\mathrm{X},\mathrm{Y}) &= \frac{1}{n}\sum_{i=1}^{n}\left(X_iY_i - X_i\overline{Y} - \overline{X}Y_i + \overline{X}\,\overline{Y}\right) \\ &= \frac{1}{n}\sum_{i=1}^{n}X_iY_i - \frac{1}{n}\sum_{i=1}^{n}X_i\overline{Y} - \frac{1}{n}\sum_{i=1}^{n}\overline{X}Y_i + \frac{1}{n}\sum_{i=1}^{n}\overline{X}\,\overline{Y} \\ &= \frac{1}{n}\sum_{i=1}^{n}X_iY_i - \overline{Y}\frac{1}{n}\sum_{i=1}^{n}X_i - \overline{X}\frac{1}{n}\sum_{i=1}^{n}Y_i + \frac{1}{n}n\overline{X}\,\overline{Y} \\ &= \frac{1}{n}\sum_{i=1}^{n}X_iY_i - \overline{X}\,\overline{Y} - \overline{X}\,\overline{Y} + \overline{X}\,\overline{Y} \\ &= \frac{1}{n}\sum_{i=1}^{n}X_iY_i - \overline{X}\,\overline{Y}\end{aligned}$$

$$\mathrm{Cov}(\mathrm{aX},\mathrm{bY}) = ab\mathrm{Cov}(\mathrm{X},\mathrm{Y})$$

$$\begin{aligned}
\operatorname{Cov}(\mathrm{aX}, \mathrm{bY}) &= \frac{1}{n}\sum_{i=1}^{n}\left(aX_i - a\overline{X}\right)\left(bY_i - b\overline{Y}\right) \\
&= \frac{1}{n}\sum_{i=1}^{n} a\left(X_i - \overline{X}\right) b\left(Y_i - \overline{Y}\right) \\
&= ab\frac{1}{n}\sum_{i=1}^{n}\left(X_i - \overline{X}\right)\left(Y_i - \overline{Y}\right) \\
&= \operatorname{Cov}(\mathrm{X}, \mathrm{Y})
\end{aligned}$$