CURSO DE ESTATÍSTICA - PARTE 2

1 CONHECENDO OS DADOS

1.1 Dataset do projeto

Pesquisa Nacional por Amostra de Domicílios - 2015

A **Pesquisa Nacional por Amostra de Domicílios - PNAD** investiga anualmente, de forma permanente, características gerais da população, de educação, trabalho, rendimento e habitação e outras, com periodicidade variável, de acordo com as necessidades de informação para o país, como as características sobre migração, fecundidade, nupcialidade, saúde, segurança alimentar, entre outros temas. O levantamento dessas estatísticas constitui, ao longo dos 49 anos de realização da pesquisa, um importante instrumento para formulação, validação e avaliação de políticas orientadas para o desenvolvimento socioeconômico e a melhoria das condições de vida no Brasil.

Fonte dos Dados

https://ww2.ibge.gov.br/home/estatistica/populacao/trabalhoerendimento/pnad2015/microdados.shtm

Variáveis utilizadas

> ##### Renda

Rendimento mensal do trabalho principal para pessoas de 10 anos ou mais de idade.

> ##### Idade

Idade do morador na data de referência em anos.

> ##### Altura (elaboração própria)

Altura do morador em metros.

> ##### UF

Código	Descrição
11	Rondônia
12	Acre
13	Amazonas

Código	Descrição
14	Roraima
15	Pará
16	Amapá
17	Tocantins
21	Maranhão
22	Piauí
23	Ceará
24	Rio Grande do Norte
25	Paraíba
26	Pernambuco
27	Alagoas
28	Sergipe
29	Bahia
31	Minas Gerais
32	Espírito Santo
33	Rio de Janeiro
35	São Paulo
41	Paraná
42	Santa Catarina
43	Rio Grande do Sul
50	Mato Grosso do Sul
51	Mato Grosso
52	Goiás
53	Distrito Federal

Sexo

Código	Descrição
0	Masculino
1	Feminino

Anos de Estudo

Código	Descrição
1	Sem instrução e menos de 1 ano
2	1 ano
3	2 anos
4	3 anos
5	4 anos
6	5 anos

Código	Descrição
7	6 anos
8	7 anos
9	8 anos
10	9 anos
11	10 anos
12	11 anos
13	12 anos
14	13 anos
15	14 anos
16	15 anos ou mais
17	Não determinados
	Não aplicável

> ## Cor

Código	Descrição
0	Indígena
2	Branca
4	Preta
6	Amarela
8	Parda
9	Sem declaração

Observação

> Os seguintes tratamentos foram realizados nos dados originais:
>
> 1. Foram eliminados os registros onde a **Renda** era inválida (999 999 999 999);
> 2. Foram eliminados os registros onde a **Renda** era missing;
> 3. Foram considerados somente os registros das **Pessoas de Referência** de cada domicílio (responsável pelo domicílio).

Importando pandas e lendo o dataset do projeto

https://pandas.pydata.org/

In [1]:
```
import pandas as pd
```

In [2]:
```
dados = pd.read_csv("dados.csv")
```

In [3]:
```
dados.head()
```

Out[3]:

	UF	Sexo	Idade	Cor	Anos de Estudo	Renda	Altura

	UF	Sexo	Idade	Cor	Anos de Estudo	Renda	Altura
0	11	0	23	8	12	800	1.603808
1	11	1	23	2	12	1150	1.739790
2	11	1	35	8	15	880	1.760444
3	11	0	46	2	6	3500	1.783158
4	11	1	47	8	9	150	1.690631

---

2 DISTRIBUIÇÕES DE PROBABILIDADE

Problema

Em um concurso para preencher uma vaga de cientista de dados temos um total de **10 questões** de múltipla escolha com **3 alternativas possíveis** em cada questão. **Cada questão tem o mesmo valor.** Suponha que um candidato resolva se aventurar sem ter estudado absolutamente nada. Ele resolve fazer a prova de olhos vendados e chutar todas as resposta. Assumindo que a prova **vale 10 pontos e a nota de corte seja 5**, obtenha a probabilidade deste candidato **acertar 5 questões** e também a probabilidade deste candidato **passar para a próxima etapa do processo seletivo**.

2.1 Distribuição Binomial

Um evento **binomial** é caracterizado pela possibilidade de ocorrência de apenas duas categorias. Estas categorias somadas representam todo o espaço amostral, sendo também mutuamente excludentes, ou seja, a ocorrência de uma implica na não ocorrência da outra.

Em análises estatísticas o uso mais comum da distribuição binomial é na solução de problemas que envolvem situações de **sucesso** e **fracasso**.

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

Onde:

p = probabilidade de sucesso

$q = (1 - p)$ = probabilidade de fracasso

n = número de eventos estudados

k = número de eventos desejados que tenham sucesso

Experimento Binomial

1. Realização de n ensaios idênticos.

2. Os ensaios são independentes.
3. Somente dois resultados são possíveis, exemplo: Verdadeiro ou falso; Cara ou coroa; Sucesso ou fracasso.
4. A probabilidade de sucesso é representada por p e a de fracasso por $1 - p = q$. Estas probabilidades não se modificam de ensaio para ensaio.

Média da distribuição binomial

O valor esperado ou a média da distribuição binomial é igual ao número de experimentos realizados multiplicado pela chance de ocorrência do evento.

$$\mu = n \times p$$

Desvio padrão da distribuição binomial

O desvio padrão é o produto entre o número de experimentos, a probabilidade de sucesso e a probabilidade de fracasso.

$$\sigma = \sqrt{n \times p \times q}$$

Importando bibliotecas

https://docs.scipy.org/doc/scipy/reference/generated/scipy.special.comb.html

In [4]:
```
from scipy.special import comb
```

Combinações

Número de combinações de n objetos, tomados k a cada vez, é:

$$C_k^n = \binom{n}{k} = \frac{n!}{k!(n-k)!}$$

Onde

$$n! = n \times (n-1) \times (n-2) \times \ldots \times (2) \times (1)$$

$$k! = k \times (k-1) \times (k-2) \times \ldots \times (2) \times (1)$$

Por definição

$$0! = 1$$

Exemplo: Mega Sena

Em um volante de loteria da Mega Sena temos um total de **60 números** para escolher onde a aposta mínima é de **seis números**. Você que é curiosa(o) resolve calcular a probabilidade de se acertar na Mega Sena com apenas **um jogo**. Para isso precisamos saber quantas **combinações de seis números podem ser formadas com os 60 números disponíveis**.

$$C_6^{60} = \binom{60}{6} = \frac{60!}{6!(60-6)!}$$

In [5]:
```
combinacoes = comb(60, 6)
combinacoes
```

Out[5]: 50063860.0

In [6]:
```
probabilidade = 1 / combinacoes
print(f"{probabilidade:.15f}")
```

```
0.000000019974489
```

Exemplo: Concurso para cientista de dados

Em um concurso para preencher uma vaga de cientista de dados temos um total de **10 questões** de múltipla escolha com **3 alternativas possíveis** em cada questão. **Cada questão tem o mesmo valor.** Suponha que um candidato resolva se aventurar sem ter estudado absolutamente nada. Ele resolve fazer a prova de olhos vendados e chutar todas as resposta. Assumindo que a prova **vale 10 pontos e a nota de corte seja 5**, obtenha a probabilidade deste candidato **acertar 5 questões** e também a probabilidade deste candidato **passar para a próxima etapa do processo seletivo**.

Qual o número de ensaios (n)?

In [7]:
```
n = 10
n
```

Out[7]: 10

Os ensaios são independentes?

Sim. A opção escolhida em uma questão não influencia em nada a opção escolhida em outra questão.

Somente dois resultados são possíveis em cada ensaio?

Sim. O candidato tem duas possibilidades, ACERTA ou ERRAR uma questão.

Qual a probabilidade de sucesso (p)?

In [8]:
```
numero_de_alternativas_por_questao = 3
p = 1 / numero_de_alternativas_por_questao
p
```

Out[8]: 0.3333333333333333

Qual a probabilidade de fracasso (q)?

In [9]:
```
q = 1 - p
q
```

Out[9]: 0.6666666666666667

Qual o total de eventos que se deseja obter sucesso (k)?

In [10]:
```
k = 5
k
```

Out[10]: 5

Solução 1

In [11]:
```
probabilidade = (comb(n, k)) * (p ** k) * (q ** (n - k))
print(f"{probabilidade:.8f}")
```

```
0.13656455
```

Importando bibliotecas

https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.binom.html

In [12]:
```
from scipy.stats import binom
```

Solução 2

In [13]:
```
probabilidade = binom.pmf(k, n, p)
print('%0.8f' % probabilidade)
```

```
0.13656455
```

Obter a probabilidade do candidato passar

$$P(acertar \geq 5) = P(5) + P(6) + P(7) + P(8) + P(9) + P10)$$

In [14]:
```
binom.pmf(5, n, p) + binom.pmf(6, n, p) + binom.pmf(7, n, p) + binom.pmf(8, n, p) + binom.pmf(9
```

Out[14]: 0.21312808006909525

In [15]:
```
binom.pmf([5, 6, 7, 8, 9, 10], n, p).sum()
```

Out[15]: 0.21312808006909525

In [16]:
```
1 - binom.cdf(4, n, p)
```

Out[16]: 0.21312808006909512

In [17]:
```
binom.sf(4, n, p)
```

Out[17]: 0.21312808006909517

Exemplo: Gincana

Uma cidade do interior realiza todos os anos uma gincana para arrecadar fundos para o hospital da cidade. Na última gincana se sabe que a **proporção de participantes do sexo feminino foi de 60%**. **O total de equipes, com 12 integrantes, inscritas na gincana deste ano é de 30**. Com as informações acima responda: Quantas equipes deverão ser formadas por **8 mulheres**?

Solução

In [18]:
```
n = 12
n
```

Out[18]: 12

In [19]:
```
p = 0.6
p
```

Out[19]: 0.6

In [20]:
```
k = 8
k
```

Out[20]: 8

In [21]:
```
probabilidade = binom.pmf(k, n, p)
print(f"{probabilidade:.8f}")
```

```
0.21284094
```

In [22]:
```
equipes = 30 * probabilidade
equipes
```

Out[22]: 6.385228185599988

Problema

Um restaurante recebe em média **20 pedidos por hora**. Qual a chance de que, em determinada hora escolhida ao acaso, o restaurante receba **15 pedidos**?

2.2 Distribuição Poisson

É empregada para descrever o número de ocorrências em um intervalo de tempo ou espaço específico. Os eventos são caracterizados pela possibilidade de contagem dos sucessos, mas a não possibilidade de contagem dos fracassos.

Como exemplos de processos onde podemos aplicar a distribuição de Poisson temos a determinação do número de clientes que entram em uma loja em determinada hora, o número de carros que chegam em um drive-thru de uma lanchonete na hora do almoço, a determinação do número de acidentes registrados em um trecho de estrada etc.

$$P(k) = \frac{e^{-\mu}(\mu)^k}{k!}$$

Onde:

e = constante cujo valor aproximado é 2,718281828459045

μ = representa o número médio de ocorrências em um determinado intervalo de tempo ou espaço

k = número de sucessos no intervalo desejado

Experimento Poisson

1. A probabilidade de uma ocorrência é a mesma em todo o intervalo observado.
2. O número de ocorrências em determinado intervalo é independente do número de ocorrências em outros intervalos.
3. A probabilidade de uma ocorrência é a mesma em intervalos de igual comprimento.

Média da distribuição Poisson

$$\mu$$

Desvio padrão da distribuição Poisson

$$\sigma = \sqrt{\mu}$$

Importando bibliotecas

http://www.numpy.org/

In [23]:
```
import numpy as np
```

In [24]:
```
np.e
```

Out[24]:
```
2.718281828459045
```

Exemplo: Delivery

Um restaurante recebe em média **20 pedidos por hora**. Qual a chance de que, em determinada hora escolhida ao acaso, o restaurante receba **15 pedidos**?

Qual o número médio de ocorrências por hora (μ)?

In [25]:
```
media = 20
media
```

Out[25]: 20

Qual o número de ocorrências que queremos obter no período (k)?

In [26]:
```
k = 15
k
```

Out[26]: 15

Solução 1

In [27]:
```
probabilidade = ((np.e ** (-media)) * (media ** k)) / (np.math.factorial(k))
print(f"{probabilidade:.8f}")
```

```
0.05164885
```

Importando bibliotecas

https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.poisson.html

Solução 2

In [28]:
```
from scipy.stats import poisson

probabilidade = poisson.pmf(k, media)
print(f"{probabilidade:.8f}")
```

```
0.05164885
```

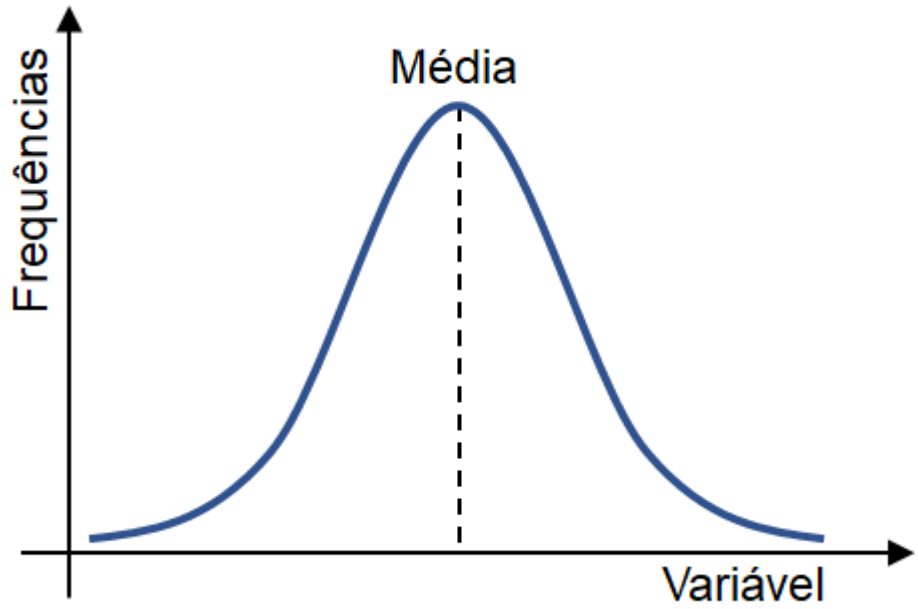
Problema

Em um estudo sobre as alturas dos moradores de uma cidade verificou-se que o conjunto de dados segue uma **distribuição aproximadamente normal**, com **média 1,70** e **desvio padrão de 0,1**. Com estas informações obtenha o seguinte conjunto de probabilidades:

> **A.** probabilidade de uma pessoa, selecionada ao acaso, ter menos de 1,80 metros.
>
> **B.** probabilidade de uma pessoa, selecionada ao acaso, ter entre 1,60 metros e 1,80 metros.
>
> **C.** probabilidade de uma pessoa, selecionada ao acaso, ter mais de 1,90 metros.

2.3 Distribuição Normal

A distribuição normal é uma das mais utilizadas em estatística. É uma distribuição contínua, onde a distribuição de frequências de uma variável quantitativa apresenta a forma de sino e é simétrica em relação a sua média.



Características importantes

1. É simétrica em torno da média;
2. A área sob a curva corresponde à proporção 1 ou 100%;
3. As medidas de tendência central (média, mediana e moda) apresentam o mesmo valor;
4. Os extremos da curva tendem ao infinito em ambas as direções e, teoricamente, jamais tocam o eixo x;
5. O desvio padrão define o achatamento e largura da distribuição. Curvas mais largas e mais achatadas apresentam valores maiores de desvio padrão;
6. A distribuição é definida por sua média e desvio padrão;
7. A probabilidade sempre será igual à área sob a curva, delimitada pelos limites inferior e superior.

$$f(x) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

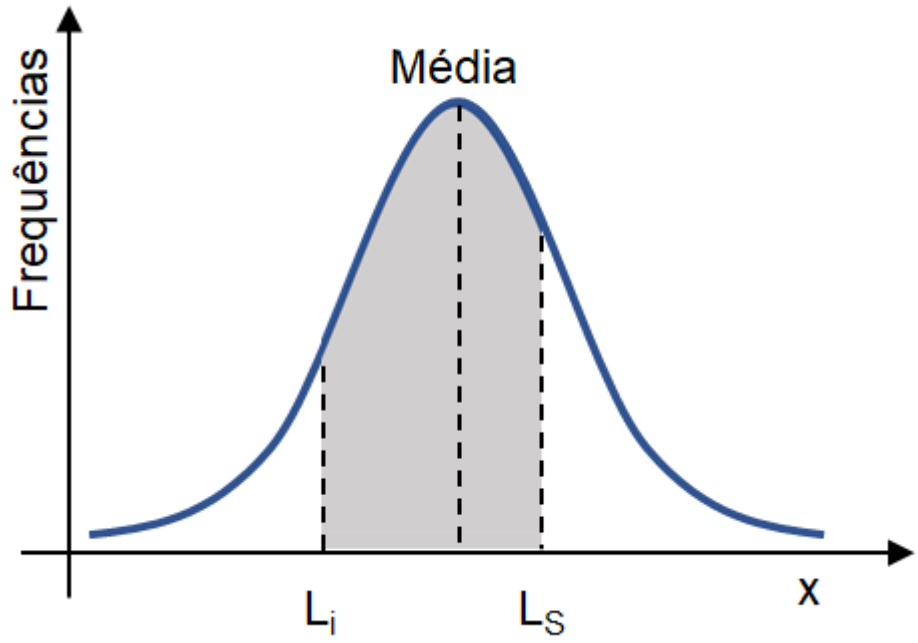
Onde:

x = variável normal

σ = desvio padrão

μ = média

A probabilidade é obtida a partir da área sob a curva, delimitada pelos limites inferior e superior especificados. Um exemplo pode ser visto na figura abaixo.



Para obter a área acima basta calcular a integral da função para os intervalos determinados. Conforme equação abaixo:

$$P(L_i < x < L_s) = \int_{L_i}^{L_s} \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

Onde:

x = variável normal

σ = desvio padrão

μ = média

L_i = limite inferior

L_s = limite superior

Tabelas padronizadas

As tabelas padronizadas foram criadas para facilitar a obtenção dos valores das áreas sob a curva normal e eliminar a necessidade de solucionar integrais definidas.

Para consultarmos os valores em uma tabela padronizada basta transformarmos nossa variável em uma variável padronizada Z.

Esta variável Z representa o afastamento em desvios padrões de um valor da variável original em relação à média.

$$Z = \frac{x - \mu}{\sigma}$$

Onde:

x = variável normal com média μ e desvio padrão σ

σ = desvio padrão

μ = média

Construindo tabela normal padronizada

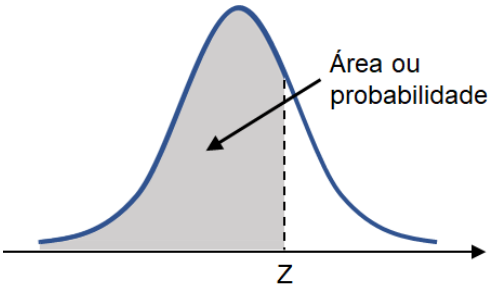
https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.norm.html

In [29]:
```
import pandas as pd
import numpy as np
from scipy.stats import norm

tabela_normal_padronizada = pd.DataFrame(
    [],
    index=["{0:0.2f}".format(i / 100) for i in range(0, 400, 10)],
    columns = ["{0:0.2f}".format(i / 100) for i in range(0, 10)])

for index in tabela_normal_padronizada.index:
    for column in tabela_normal_padronizada.columns:
        Z = np.round(float(index) + float(column), 2)
        tabela_normal_padronizada.loc[index, column] = "{0:0.4f}".format(norm.cdf(Z))

tabela_normal_padronizada.rename_axis('Z', axis = 'columns', inplace = True)

tabela_normal_padronizada
```

Out[29]:

Z	0.00	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09
0.00	0.5000	0.5040	0.5080	0.5120	0.5160	0.5199	0.5239	0.5279	0.5319	0.5359
0.10	0.5398	0.5438	0.5478	0.5517	0.5557	0.5596	0.5636	0.5675	0.5714	0.5753
0.20	0.5793	0.5832	0.5871	0.5910	0.5948	0.5987	0.6026	0.6064	0.6103	0.6141
0.30	0.6179	0.6217	0.6255	0.6293	0.6331	0.6368	0.6406	0.6443	0.6480	0.6517
0.40	0.6554	0.6591	0.6628	0.6664	0.6700	0.6736	0.6772	0.6808	0.6844	0.6879
0.50	0.6915	0.6950	0.6985	0.7019	0.7054	0.7088	0.7123	0.7157	0.7190	0.7224
0.60	0.7257	0.7291	0.7324	0.7357	0.7389	0.7422	0.7454	0.7486	0.7517	0.7549
0.70	0.7580	0.7611	0.7642	0.7673	0.7704	0.7734	0.7764	0.7794	0.7823	0.7852
0.80	0.7881	0.7910	0.7939	0.7967	0.7995	0.8023	0.8051	0.8078	0.8106	0.8133
0.90	0.8159	0.8186	0.8212	0.8238	0.8264	0.8289	0.8315	0.8340	0.8365	0.8389
1.00	0.8413	0.8438	0.8461	0.8485	0.8508	0.8531	0.8554	0.8577	0.8599	0.8621
1.10	0.8643	0.8665	0.8686	0.8708	0.8729	0.8749	0.8770	0.8790	0.8810	0.8830
1.20	0.8849	0.8869	0.8888	0.8907	0.8925	0.8944	0.8962	0.8980	0.8997	0.9015
1.30	0.9032	0.9049	0.9066	0.9082	0.9099	0.9115	0.9131	0.9147	0.9162	0.9177
1.40	0.9192	0.9207	0.9222	0.9236	0.9251	0.9265	0.9279	0.9292	0.9306	0.9319
1.50	0.9332	0.9345	0.9357	0.9370	0.9382	0.9394	0.9406	0.9418	0.9429	0.9441
1.60	0.9452	0.9463	0.9474	0.9484	0.9495	0.9505	0.9515	0.9525	0.9535	0.9545
1.70	0.9554	0.9564	0.9573	0.9582	0.9591	0.9599	0.9608	0.9616	0.9625	0.9633
1.80	0.9641	0.9649	0.9656	0.9664	0.9671	0.9678	0.9686	0.9693	0.9699	0.9706
1.90	0.9713	0.9719	0.9726	0.9732	0.9738	0.9744	0.9750	0.9756	0.9761	0.9767
2.00	0.9772	0.9778	0.9783	0.9788	0.9793	0.9798	0.9803	0.9808	0.9812	0.9817

Z	0.00	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09
2.10	0.9821	0.9826	0.9830	0.9834	0.9838	0.9842	0.9846	0.9850	0.9854	0.9857
2.20	0.9861	0.9864	0.9868	0.9871	0.9875	0.9878	0.9881	0.9884	0.9887	0.9890
2.30	0.9893	0.9896	0.9898	0.9901	0.9904	0.9906	0.9909	0.9911	0.9913	0.9916
2.40	0.9918	0.9920	0.9922	0.9925	0.9927	0.9929	0.9931	0.9932	0.9934	0.9936
2.50	0.9938	0.9940	0.9941	0.9943	0.9945	0.9946	0.9948	0.9949	0.9951	0.9952
2.60	0.9953	0.9955	0.9956	0.9957	0.9959	0.9960	0.9961	0.9962	0.9963	0.9964
2.70	0.9965	0.9966	0.9967	0.9968	0.9969	0.9970	0.9971	0.9972	0.9973	0.9974
2.80	0.9974	0.9975	0.9976	0.9977	0.9977	0.9978	0.9979	0.9979	0.9980	0.9981
2.90	0.9981	0.9982	0.9982	0.9983	0.9984	0.9984	0.9985	0.9985	0.9986	0.9986
3.00	0.9987	0.9987	0.9987	0.9988	0.9988	0.9989	0.9989	0.9989	0.9990	0.9990
3.10	0.9990	0.9991	0.9991	0.9991	0.9992	0.9992	0.9992	0.9992	0.9993	0.9993
3.20	0.9993	0.9993	0.9994	0.9994	0.9994	0.9994	0.9994	0.9995	0.9995	0.9995
3.30	0.9995	0.9995	0.9995	0.9996	0.9996	0.9996	0.9996	0.9996	0.9996	0.9997
3.40	0.9997	0.9997	0.9997	0.9997	0.9997	0.9997	0.9997	0.9997	0.9997	0.9998
3.50	0.9998	0.9998	0.9998	0.9998	0.9998	0.9998	0.9998	0.9998	0.9998	0.9998
3.60	0.9998	0.9998	0.9999	0.9999	0.9999	0.9999	0.9999	0.9999	0.9999	0.9999
3.70	0.9999	0.9999	0.9999	0.9999	0.9999	0.9999	0.9999	0.9999	0.9999	0.9999
3.80	0.9999	0.9999	0.9999	0.9999	0.9999	0.9999	0.9999	0.9999	0.9999	0.9999
3.90	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000



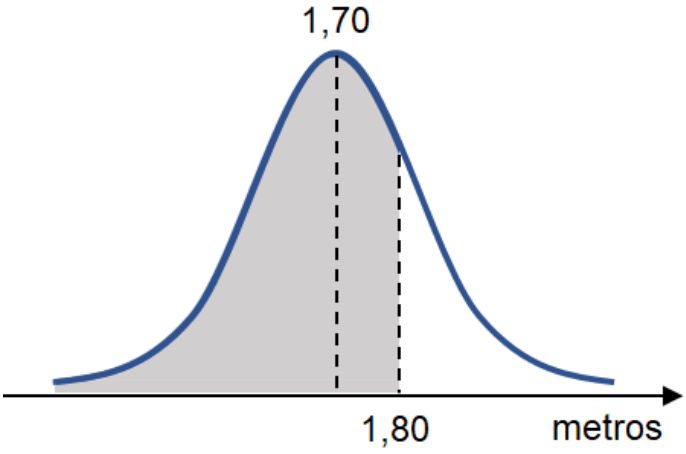
A tabela acima fornece a área sob a curva entre $-\infty$ e Z desvios padrão acima da média. Lembrando que por se tratar de valores padronizados temos $\mu = 0$.

Exemplo: Qual sua altura?

Em um estudo sobre as alturas dos moradores de uma cidade verificou-se que o conjunto de dados segue uma **distribuição aproximadamente normal**, com **média 1,70** e **desvio padrão de 0,1**. Com estas informações obtenha o seguinte conjunto de probabilidades:

> **A.** probabilidade de uma pessoa, selecionada ao acaso, ter menos de 1,80 metros.
>
> **B.** probabilidade de uma pessoa, selecionada ao acaso, ter entre 1,60 metros e 1,80 metros.
>
> **C.** probabilidade de uma pessoa, selecionada ao acaso, ter mais de 1,90 metros.

Problema A - Identificação da área sob a curva



Obter a variável padronizada Z

In [30]:
```
media = 1.7
media
```

Out[30]: 1.7

In [31]:
```
desvio_padrao = 0.1
desvio_padrao
```

Out[31]: 0.1

In [32]:
```
z = (1.8 - media) / desvio_padrao
z
```

Out[32]: 1.0000000000000009

Solução 1 - Utilizando tabela

In [33]:
```
probabilidade = 0.8413
probabilidade
```

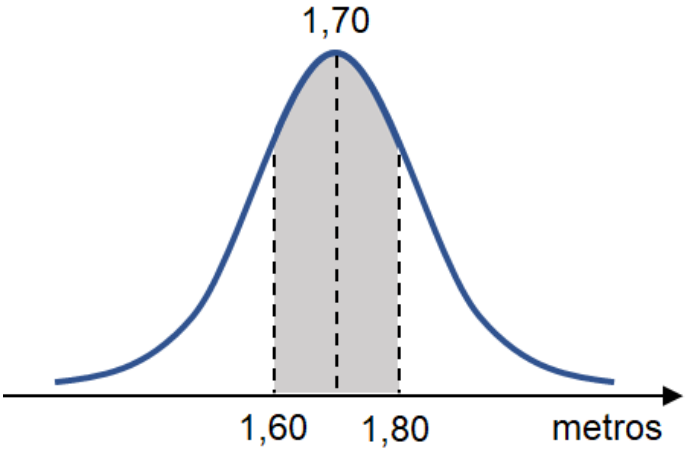
Out[33]: 0.8413

Solução 2 - Utilizando Scipy

In [34]:
```
norm.cdf(z)
```

Out[34]: 0.8413447460685431

Problema B - Identificação da área sob a curva



Obter a variável padronizada Z

In [35]:
```
z_inferior = (1.6 - media) / desvio_padrao
round(z_inferior, 2)
```

Out[35]: -1.0

In [36]:
```
z_superior = (1.8 - media) / desvio_padrao
round(z_superior, 2)
```

Out[36]: 1.0

Solução 1 - Utilizando tabela

In [37]:
```
probabilidade = (0.8413 - 0.5) * 2
probabilidade
```

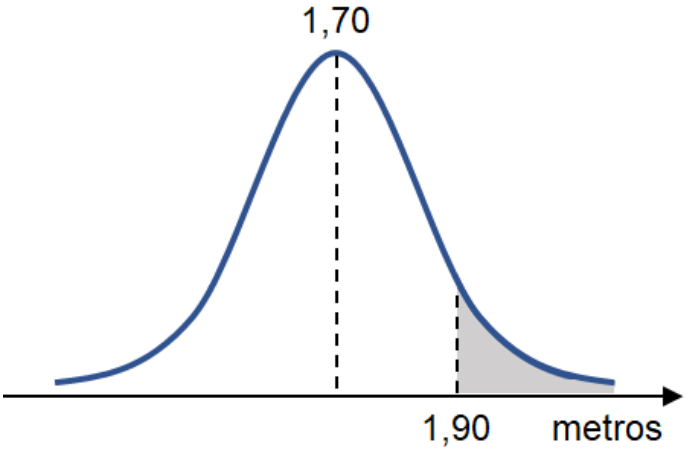
Out[37]: 0.6826000000000001

Solução 2 - Utilizando Scipy

In [38]:
```
probabilidade = norm.cdf(z_superior) - norm.cdf(z_inferior)
probabilidade
```

Out[38]: 0.6826894921370857

Problema C - Identificação da área sob a curva



Obter a variável padronizada Z

In [39]:
```
z = (1.9 - media) / desvio_padrao
z
```

Out[39]:
```
1.9999999999999996
```

Solução 1 - Utilizando tabela

In [40]:
```
probabilidade = 1 - 0.9767
probabilidade
```

Out[40]:
```
0.023299999999999987
```

Solução 2 - Utilizando Scipy

In [41]:
```
probabilidade =  1 - norm.cdf(z)
probabilidade
```

Out[41]:
```
0.02275013194817921
```

In [42]:
```
probabilidade = norm.cdf(z * -1)
probabilidade
```

Out[42]:
```
0.022750131948179216
```

3 AMOSTRAGEM

3.1 População e Amostra

População

Conjunto de todos os elementos de interesse em um estudo. Diversos elementos podem compor uma população, por exemplo: pessoas, idades, alturas, carros etc.

Com relação ao tamanho, as populações podem ser limitadas (populações finitas) ou ilimitadas (populações infinitas).

Populações finitas

Permitem a contagem de seus elementos. Como exemplos temos o número de funcionário de uma empresa, a quantidade de alunos em uma escola etc.

Populações infinitas

Não é possível contar seus elementos. Como exemplos temos a quantidade de porções que se pode extrair da água do mar para uma análise, temperatura medida em cada ponto de um território etc.

Quando os elementos de uma população puderem ser contados, porém apresentando uma quantidade muito grande, assume-se a população como infinita..

Amostra

Subconjunto representativo da população.

Os atributos numéricos de uma população como sua média, variância e desvio padrão, são conhecidos como **parâmetros**. O principal foco da inferência estatística é justamente gerar estimativas e testar hipóteses sobre os parâmetros populacionais utilizando as informações de amostras.

3.2 Quando utilizar uma amostra?

Populações infinitas

O estudo não chegaria nunca ao fim. Não é possível investigar todos os elementos da população.

Testes destrutivos

Estudos onde os elementos avaliados são totalmente consumidos ou destruídos. Exemplo: testes de vida útil, testes de segurança contra colisões em automóveis.

Resultados rápidos

Pesquisas que precisam de mais agilidade na divulgação. Exemplo: pesquisas de opinião, pesquisas que envolvam problemas de saúde pública.

Custos elevados

Quando a população é finita mas muito numerosa, o custo de um censo pode tornar o processo inviável.

3.3 Amostragem Aleatória Simples

É uma das principais maneiras de se extrair uma amostra de uma população. A exigência fundamental deste tipo de abordagem é que cada elemeto da população tenha as mesmas chances de ser selecionado para fazer parte da amostra.

In [43]:
```
dados.shape[0]
```

Out[43]:
```
76840
```

In [44]:
```
dados.Renda.mean()
```

Out[44]:
```
2000.3831988547631
```

In [45]:
```
amostra = dados.sample(n = 1000, random_state = 101)
amostra
```

Out[45]:

	UF	Sexo	Idade	Cor	Anos de Estudo	Renda	Altura
29042	29	0	39	8	5	480	1.719128
62672	43	0	55	2	6	250	1.639205
29973	29	1	36	2	12	788	1.654122
22428	26	0	46	8	8	1680	1.622450
55145	41	0	37	2	9	2500	1.625268
...	...	...	...	...	...	...	...
28141	29	0	22	4	11	788	1.720672

	UF	Sexo	Idade	Cor	Anos de Estudo	Renda	Altura
8473	15	0	33	8	5	800	1.782539
72127	52	0	33	2	12	2000	1.795621
56491	41	0	56	2	12	1000	1.730259
14800	23	0	46	8	3	788	1.706331

1000 rows × 7 columns

In [46]:
```
amostra.shape[0]
```

Out[46]:
```
1000
```

In [47]:
```
amostra.Renda.mean()
```

Out[47]:
```
1998.783
```

In [48]:
```
dados.Sexo.value_counts(normalize = True)
```

Out[48]:
```
0    0.692998
1    0.307002
Name: Sexo, dtype: float64
```

In [49]:
```
amostra.Sexo.value_counts(normalize = True)
```

Out[49]:
```
0    0.706
1    0.294
Name: Sexo, dtype: float64
```

3.4 Amostragem Estratificada

É uma melhoria do processo de amostragem aleatória simples. Neste método é proposta a divisão da população em subgrupos de elementos com características similares, ou seja, grupos mais homogêneos. Com estes subgrupos separados, aplica-se a técnica de amostragem aleatória simples dentro de cada subgrupo individualmente.

3.5 Amostragem por Conglomerados

Também visa melhorar o critério de amostragem aleatória simples. Na amostragem por conglomerados são também criados subgrupos, porém não serão homogêneas como na amostragem estratificada. Na amostragem por conglomerados os subgrupos serão heterogêneos, onde, em seguida, serão aplicadas a amostragem aleatória simples ou estratificada.

Um exemplo bastante comum de aplicação deste tipo de técnica é na divisão da população em grupos territoriais, onde os elementos investigados terão características bastante variadas.

4 ESTIMAÇÃO

Problema

Suponha que os pesos dos sacos de arroz de uma indústria alimentícia se distribuem aproximadamente

como uma normal de **desvio padrão populacional igual a 150 g**. Selecionada uma **amostra aleatório de 20 sacos** de um lote específico, obteve-se um **peso médio de 5.050 g**. Construa um **intervalo de confiança para a média populacional** assumindo um **nível de significância de 5%**.

É a forma de se fazer suposições generalizadas sobre os parâmetros de uma população tendo como base as informações de uma amostra.

- **Parâmetros** são os atributos numéricos de uma população, tal como a média, desvio padrão etc.
- **Estimativa** é o valor obtido para determinado parâmetro a partir dos dados de uma amostra da população.

4.1 Teorema do limite central

> O **Teorema do Limite Central** afirma que, com o aumento do tamanho da amostra, a distribuição das médias amostrais se aproxima de uma distribuição normal com média igual à média da população e desvio padrão igual ao desvio padrão da variável original dividido pela raiz quadrada do tamanho da amostra. Este fato é assegurado para n maior ou igual a 30.

$$\sigma_{\bar{x}} = \frac{\sigma}{\sqrt{n}}$$

O desvio padrão das médias amostrais é conhecido como **erro padrão da média**

Entendendo o Teorema do Limite Central

In [50]:
```
n = 2000
total_de_amostras = 1500
```

In [51]:
```
amostras = pd.DataFrame()
amostras
```

Out[51]: —

In [52]:
```
for i in range(total_de_amostras):
    _ = dados.Idade.sample(n)
    _.index = range(0, len(_))
    amostras['Amostra_' + str(i)] = _
amostras
```

Out[52]:

	Amostra_0	Amostra_1	Amostra_2	Amostra_3	Amostra_4	Amostra_5	Amostra_6	Amostra_7	Amostra_8
0	54	35	40	30	58	24	39	40	54
1	25	25	48	59	33	79	50	29	66
2	30	49	19	23	61	54	38	45	38
3	37	53	58	43	57	44	47	30	55
4	50	41	91	34	45	30	23	57	41
...	...	...	...	...	...	...	...	...	...

	Amostra_0	Amostra_1	Amostra_2	Amostra_3	Amostra_4	Amostra_5	Amostra_6	Amostra_7	Amostra_8
1995	41	37	62	42	37	50	48	39	48
1996	42	38	32	44	33	37	35	58	42
1997	35	50	65	30	58	47	60	60	25
1998	60	46	24	53	34	62	43	35	39
1999	48	22	33	49	76	42	45	59	34

2000 rows × 1500 columns

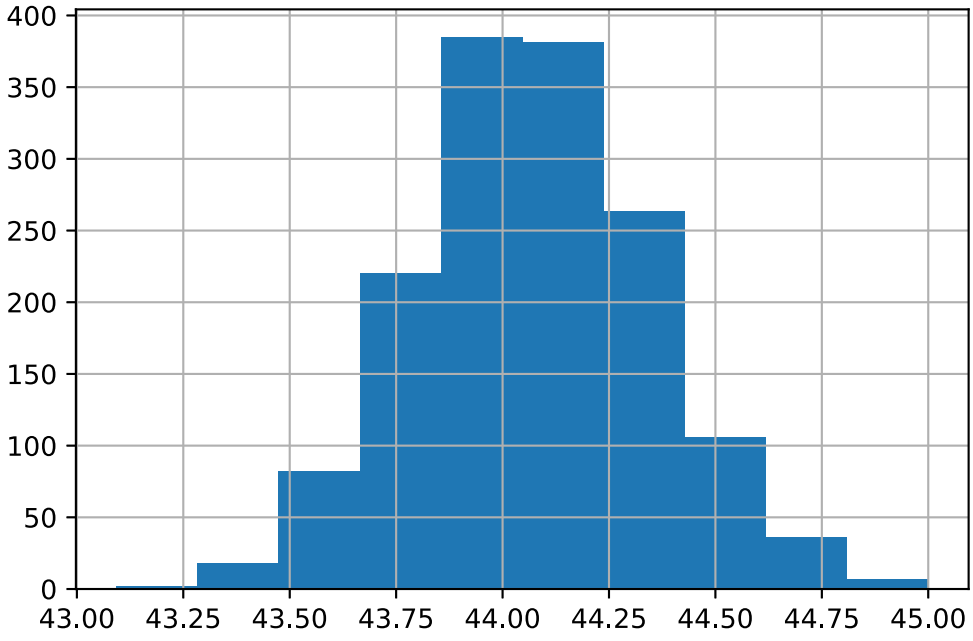
In [53]:
```
amostras.mean()
```

Out[53]:
```
Amostra_0       43.9525
Amostra_1       43.7870
Amostra_2       44.2170
Amostra_3       43.7850
Amostra_4       44.2655
                 ...
Amostra_1495    43.5775
Amostra_1496    44.2140
Amostra_1497    44.1275
Amostra_1498    44.2040
Amostra_1499    44.0070
Length: 1500, dtype: float64
```

> O Teorema do Limite Central afirma que, **com o aumento do tamanho da amostra, a distribuição das médias amostrais se aproxima de uma distribuição normal** com média igual à média da população e desvio padrão igual ao desvio padrão da variável original dividido pela raiz quadrada do tamanho da amostra. Este fato é assegurado para n maior ou igual a 30.

In [54]:
```
amostras.mean().hist()
```

Out[54]:
```
<AxesSubplot:>
```



> O Teorema do Limite Central afirma que, com o aumento do tamanho da amostra, a distribuição das médias amostrais se aproxima de uma distribuição normal **com média igual**

> **à média da população** e desvio padrão igual ao desvio padrão da variável original dividido pela raiz quadrada do tamanho da amostra. Este fato é assegurado para n maior ou igual a 30.

In [55]:
```
dados.Idade.mean()
```

Out[55]:
```
44.07142113482561
```

In [56]:
```
amostras.mean().mean()
```

Out[56]:
```
44.07504066666671
```

> O Teorema do Limite Central afirma que, com o aumento do tamanho da amostra, a distribuição das médias amostrais se aproxima de uma distribuição normal com média igual à média da população e **desvio padrão igual ao desvio padrão da variável original dividido pela raiz quadrada do tamanho da amostra**. Este fato é assegurado para n maior ou igual a 30.

$$\sigma_{\bar{x}} = \frac{\sigma}{\sqrt{n}}$$

In [57]:
```
amostras.mean().std()
```

Out[57]:
```
0.2783089822588544
```

In [58]:
```
dados.Idade.std()
```

Out[58]:
```
12.480583465360187
```

In [59]:
```
dados.Idade.std() / np.sqrt(n)
```

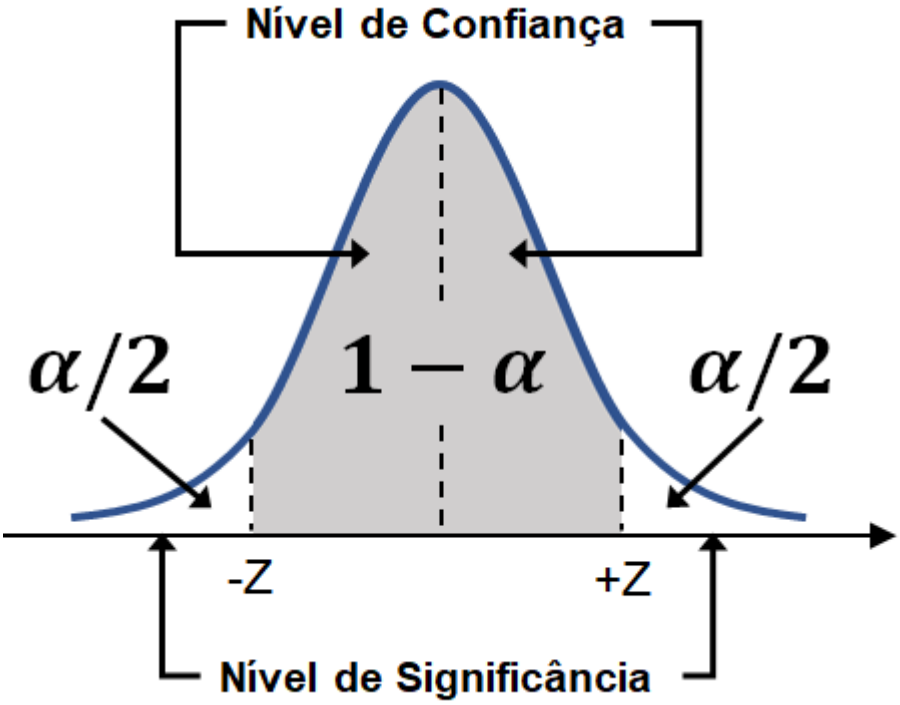
Out[59]:
```
0.2790743302740527
```

4.2 Níveis de confiança e significância

O **nível de confiança** $(1 - \alpha)$ representa a probabilidade de acerto da estimativa. De forma complementar o **nível de significância** (α) expressa a probabilidade de erro da estimativa.

O **nível de confiança** representa o grau de confiabilidade do resultado da estimativa estar dentro de determinado intervalo. Quando fixamos em uma pesquisa um **nível de confiança** de 95%, por exemplo, estamos assumindo que existe uma probabilidade de 95% dos resultados da pesquisa representarem bem a realidade, ou seja, estarem corretos.

O **nível de confiança** de uma estimativa pode ser obtido a partir da área sob a curva normal como ilustrado na figura abaixo.



4.3 Erro inferencial

O **erro inferencial** é definido pelo **desvio padrão das médias amostrais** $\sigma_{\bar{x}}$ e pelo **nível de confiança** determinado para o processo.

$$e = z\frac{\sigma}{\sqrt{n}}$$

4.4 Intervalos de confiança

Intevalo de confiança para a média da população

Com desvio padrão populacional conhecido

$$\mu = \bar{x} \pm z\frac{\sigma}{\sqrt{n}}$$

Com desvio padrão populacional desconhecido

$$\mu = \bar{x} \pm z\frac{s}{\sqrt{n}}$$

Exemplo:

Suponha que os pesos dos sacos de arroz de uma indústria alimentícia se distribuem aproximadamente como uma normal de **desvio padrão populacional igual a 150 g**. Selecionada uma **amostra aleatório de**

20 sacos de um lote específico, obteve-se um **peso médio de 5.050 g**. Construa um intervalo de confiança para a **média populacional** assumindo um **nível de significância de 5%**.

Média amostral

In [60]:
```
media_amostral = 5050
media_amostral
```

Out[60]: 5050

Nível de significância (α)

In [61]:
```
significancia = 0.05
significancia
```

Out[61]: 0.05

Nível de confiança ($1 - \alpha$)

In [62]:
```
confianca = 1 - significancia
confianca
```

Out[62]: 0.95

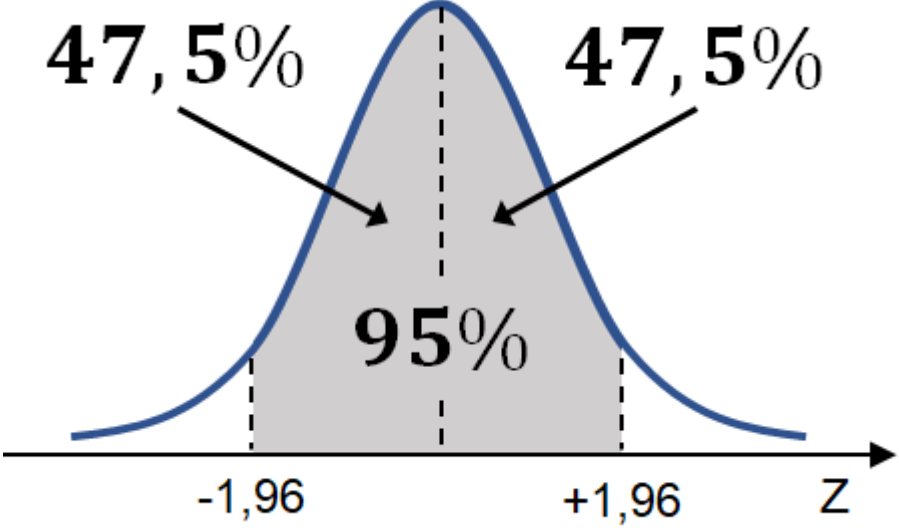
Obtendo z

In [63]:
```
tabela_normal_padronizada[16:26]
```

Out[63]:

Z	0.00	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09
1.60	0.9452	0.9463	0.9474	0.9484	0.9495	0.9505	0.9515	0.9525	0.9535	0.9545
1.70	0.9554	0.9564	0.9573	0.9582	0.9591	0.9599	0.9608	0.9616	0.9625	0.9633
1.80	0.9641	0.9649	0.9656	0.9664	0.9671	0.9678	0.9686	0.9693	0.9699	0.9706
1.90	0.9713	0.9719	0.9726	0.9732	0.9738	0.9744	0.9750	0.9756	0.9761	0.9767
2.00	0.9772	0.9778	0.9783	0.9788	0.9793	0.9798	0.9803	0.9808	0.9812	0.9817
2.10	0.9821	0.9826	0.9830	0.9834	0.9838	0.9842	0.9846	0.9850	0.9854	0.9857
2.20	0.9861	0.9864	0.9868	0.9871	0.9875	0.9878	0.9881	0.9884	0.9887	0.9890
2.30	0.9893	0.9896	0.9898	0.9901	0.9904	0.9906	0.9909	0.9911	0.9913	0.9916
2.40	0.9918	0.9920	0.9922	0.9925	0.9927	0.9929	0.9931	0.9932	0.9934	0.9936
2.50	0.9938	0.9940	0.9941	0.9943	0.9945	0.9946	0.9948	0.9949	0.9951	0.9952

Obtendo z



In [64]:
```
0.95 / 2
```

Out[64]: 0.475

In [65]:
```
0.5 + (0.95 / 2)
```

Out[65]: 0.975

In [66]:
```
z = 1.96
z
```

Out[66]: 1.96

In [67]:
```
z = norm.ppf(0.5 + (0.95 / 2))
z
```

Out[67]: 1.959963984540054

Valores de z para os níveis de confiança mais utilizados

Nível de confiança	Valor da área sob a curva normal	z
90%	0,95	1,645
95%	0,975	1,96
99%	0,995	2,575

Obtendo $\sigma_{\bar{x}}$

In [68]:
```
desvio_padrao = 150
desvio_padrao
```

Out[68]: 150

In [69]:
```
n = 20
n
```

Out[69]: 20

In [70]:
```
raiz_de_n = np.sqrt(n)
raiz_de_n
```

Out[70]: 4.47213595499958

In [71]:
```
sigma = desvio_padrao / raiz_de_n
sigma
```

Out[71]: 33.54101966249684

Obtendo e

In [72]:
```
e = z * sigma
e
```

Out[72]: 65.73919054324361

Solução 1 - Calculando o intervalo de confiança para a média

In [73]:
```
intervalo = (
    media_amostral - e,
    media_amostral + e
    )
intervalo
```

Out[73]: (4984.260809456757, 5115.739190543243)

Solução 2 - Calculando o intervalo de confiança para a média

In [74]:
```
norm.interval(alpha = 0.95, loc = media_amostral, scale = sigma)
```

Out[74]: (4984.260809456757, 5115.739190543243)

5 CÁLCULO DO TAMANHO DA AMOSTRA

Problema

Estamos estudando o rendimento mensal dos chefes de domicílios com renda até R$ 5.000,00 no Brasil. Nosso supervisor determinou que o **erro máximo em relação a média seja de R$ 10,00**. Sabemos que o **desvio padrão populacional** deste grupo de trabalhadores é de **R$ 1.082,79**. **Para um** nível de confiança de 95%**, qual deve ser o tamanho da amostra de nosso estudo?

5.1 Variáveis quantitativas e população infinita

$$e = z\frac{\sigma}{\sqrt{n}}$$

Com desvio padrão conhecido

$$n = \left(z\frac{\sigma}{e}\right)^2$$

Com desvio padrão desconhecido

$$n = \left(z\frac{s}{e}\right)^2$$

Onde:

z = variável normal padronizada

σ = desvio padrão populacional

s = desvio padrão amostral

e = erro inferencial

Observações

1. O desvio padrão (σ ou s) e o erro (e) devem estar na mesma unidade de medida.
2. Quando o erro (e) for representado em termos percentuais, deve ser interpretado como um percentual relacionado à média.

Exemplo: Rendimento médio

Estamos estudando o rendimento mensal dos chefes de domicílios no Brasil. Nosso supervisor determinou que o **erro máximo em relação a média seja de R$ 100,00**. Sabemos que o **desvio padrão populacional** deste grupo de trabalhadores é de **R$ 3.323,39**. Para um **nível de confiança de 95%**, qual deve ser o tamanho da amostra de nosso estudo?

In [75]:
```
z = norm.ppf(0.975)
z
```

Out[75]: 1.959963984540054

Obtendo σ

In [76]:
```
sigma = 3323.39
sigma
```

Out[76]: 3323.39

Obtendo e

In [77]:
```
e = 100
e
```

Out[77]: 100

Obtendo n

In [78]:
```
n = (z * (sigma / e)) ** 2
int(n.round())
```

Out[78]: 4243

Problema

Em um lote de **10.000 latas** de refrigerante foi realizada uma amostra aleatória simples de **100 latas** e foi obtido o **desvio padrão amostral do conteúdo das latas igual a 12 ml**. O fabricante estipula um **erro máximo sobre a média populacional de apenas 5 ml**. Para garantir um **nível de confiança de 95%** qual o tamanho de amostra deve ser selecionado para este estudo?

5.2 Variáveis quantitativas e população finita

Com desvio padrão conhecido

$$n = \frac{z^2 \sigma^2 N}{z^2 \sigma^2 + e^2(N-1)}$$

Com desvio padrão desconhecido

$$n = \frac{z^2 s^2 N}{z^2 s^2 + e^2(N-1)}$$

Onde:

N = tamanho da população

z = variável normal padronizada

σ = desvio padrão populacional

s = desvio padrão amostral

e = erro inferencial

Exemplo: Indústria de refrigerantes

Em um lote de **10.000 latas** de refrigerante foi realizada uma amostra aleatória simples de **100 latas** e foi obtido o **desvio padrão amostral do conteúdo das latas igual a 12 ml**. O fabricante estipula um **erro máximo sobre a média populacional de apenas 5 ml**. Para garantir um **nível de confiança de 95%** qual o tamanho de amostra deve ser selecionado para este estudo?

Obtendo N

In [79]:
```
N = 10000
N
```

Out[79]: 10000

Obtendo z

In [80]:
```
z = norm.ppf(0.975)
z
```

Out[80]: 1.959963984540054

Obtendo s

In [81]:
```
s = 12
s
```

Out[81]: 12

Obtendo e

In [82]:
```
e = 5
e
```

Out[82]: 5

Obtendo n

$$n = \frac{z^2 s^2 N}{z^2 s^2 + e^2(N-1)}$$

In [83]:
```
n = ((z ** 2) * (s ** 2) * (N)) / (((z ** 2) * (s ** 2)) + ((e ** 2) * (N - 1)))
int(n.round())
```

Out[83]: 22

6 FIXANDO O CONTEÚDO

Exemplo: Rendimento médio

Estamos estudando o **rendimento mensal dos chefes de domicílios com renda até R$ 5.000,00 no Brasil**. Nosso supervisor determinou que o **erro máximo em relação a média seja de R$ 10,00**. Sabemos que o **desvio padrão populacional** deste grupo de trabalhadores é de **R$ 1.082,79** e que a **média populacional** é de **R$ 1.426,54**. Para um **nível de confiança de 95%**, qual deve ser o tamanho da amostra de nosso estudo? Qual o intervalo de confiança para a média considerando o tamanho de amostra obtido?

Construindo o dataset conforme especificado pelo problema

In [84]:
```
renda_5k = dados.query("Renda <= 5000").Renda
```

In [85]:
```
sigma = renda_5k.std()
sigma
```

Out[85]: 1082.794549030635

In [86]:
```
media = renda_5k.mean()
media
```

Out[86]: 1426.5372144947232

Calculando o tamanho da amostra

In [87]:
```
z = norm.ppf(0.975)
e = 10
n = (z * (sigma / e)) ** 2
n = int(n.round())
n
```

Out[87]: 45039

Calculando o intervalo de confiança para a média

In [88]:
```
intervalo = norm.interval(alpha = 0.95, loc = media, scale = sigma / np.sqrt(n))
intervalo
```

Out[88]: (1416.5372195108241, 1436.5372094786223)

Realizando uma prova gráfica

In [89]:
```
import matplotlib.pyplot as plt

tamanho_simulacao = 1000

medias = [renda_5k.sample(n = n).mean() for i in range(1, tamanho_simulacao)]
medias = pd.DataFrame(medias)

ax = medias.plot(style = '.')
ax.figure.set_size_inches(12, 6)
ax.hlines(y = media, xmin = 0, xmax = tamanho_simulacao, colors='black', linestyles='dashed')
ax.hlines(y = intervalo[0], xmin = 0, xmax = tamanho_simulacao, colors='red', linestyles='dashe
ax.hlines(y = intervalo[1], xmin = 0, xmax = tamanho_simulacao, colors='red', linestyles='dashe
ax
```

Out[89]: <AxesSubplot:>