Estimação de Parâmetros Estatísticos

Resolução dos exemplos e exercícios utilizando o Python

Intervalo de confiança para a média quando σ é conhecido

Exemplo 1

(Barbetta, 2004) Em uma indústria de cerveja, a quantidade de cerveja inserida em latas tem-se comportado como uma variável aleatória distribuída normalmente com média 350 ml e desvio padrão 3 ml. Após alguns problemas na linha de produção, suspeita-se que houve alteração na média. Uma amostra de 120 latas acusou média 346 ml. Encontre a estimativa pontual e construa um intervalo de confiança para o novo valor da quantidade média de cerveja inserida em latas, com nível de confiança de 95%, supondo que não tenha ocorrido alteração no desvio padrão do processo.

Utilizando o Pyhton:

```
from scipy.stats import norm
import numpy as np

# Parâmetros do problema
media_amostral = 346  # média da amostra
desvio_padrao_populacional = 3  # desvio padrão populacional
tamanho_amostra = 120  # tamanho da amostra
nivel_confianca = 0.95  # nível de confiança

# Calculando a estimativa pontual (média da população)
estimativa_pontual = media_amostral

# Calculando o erro padrão
erro_padrao = desvio_padrao_populacional / np.sqrt(tamanho_amostra)

# Calculando o intervalo de confiança
z_score = norm.ppf((1 + nivel_confianca) / 2)  # valor z para o nível de confiança
margem_de_erro = z_score * erro_padrao
intervalo_confianca = (
    estimativa_pontual - margem_de_erro,
    estimativa_pontual + margem_de_erro,
)

print("Estimativa Pontual:", estimativa_pontual)
print("Valor Crítico (z-score):", z_score)
print("Margem de Erro:", margem_de_erro)
print("Intervalo de Confiança (95%):", intervalo_confianca)
```

```
Estimativa Pontual: 346
Valor Crítico (z-score): 1.959963984540054
```

Margem de Erro: 0.5367582431151471
Intervalo de Confiança (95%): (345.4632417568848, 346.5367582431152)

Neste código, estamos calculando a estimativa pontual, o erro padrão e o intervalo de confiança para a média populacional da quantidade de cerveja em latas. O valor crítico de z é calculado usando a função norm.ppf() para obter o valor z correspondente ao nível de confiança de 95%. Por fim, imprimimos a estimativa pontual e o intervalo de confiança.

Exercício 1

(Adaptado de Larson & Farber, 2004) O diretor do comitê de admissão de uma universidade deseja estimar a média de idade de todos os estudantes aprovados no momento. Sabe-se que, de levantamentos anteriores, o desvio padrão da população é de 1,5 ano. Em uma amostra aleatória de 20 estudantes, a idade média encontrada foi de 22,9 anos.

a) Com base nessa amostra, qual é a estimativa pontual da idade média dos estudantes aprovados?
b) Construa um intervalo de 90% de confiança para a idade média da população.

Respostas: a) 22,9 anos b) $z_{\frac{\alpha}{2}}$ =1,64, E=0,5501, IC=[22,3499 ; 23,4501]

Utilizando o Python:

```
from scipy.stats import norm
import numpy as np

# Parâmetros do problema
media_amostral = 22.9  # média da amostra
desvio_padrao_populacional = 1.5  # desvio padrão populacional
tamanho_amostra = 20  # tamanho da amostra
nivel_confianca = 0.90  # nível de confiança

# a) Calculando a estimativa pontual (média da população)
estimativa_pontual = media_amostral
print("a) Estimativa Pontual da Idade Média:", estimativa_pontual)

# b) Calculando o intervalo de confiança
erro_padrao = desvio_padrao_populacional / np.sqrt(tamanho_amostra)
z_score = norm.ppf((1 + nivel_confianca) / 2)  # valor z para o nível de confiança
margem_de_erro = z_score * erro_padrao
intervalo_confianca = (
    estimativa_pontual - margem_de_erro,
    estimativa_pontual + margem_de_erro,
)

print("b) Valor Crítico (z-score):", z_score)
print("b) Margem de Erro:", margem_de_erro)
print("b) Intervalo de Confiança (90%) para a Idade Média da População:", intervalo_confianca)
```

```
a) Estimativa Pontual da Idade Média: 22.9
b) Valor Crítico (z-score): 1.6448536269514722
b) Margem de Erro: 0.5517006784350859
b) Intervalo de Confiança (90%) para a Idade Média da População: (22.348299321564912, 23.451700678435085)
```

Exercício 2

Um investidor planeja abrir uma agência de viagens e deseja estimar o faturamento médio mensal em dólares. Suponha que os faturamentos mensais de agências de viagens, do porte que o investidor pretende abrir, se distribuam normalmente com um desvio padrão de US$130,00. Durante nove meses, o investidor anotou o faturamento líquido mensal de uma agência de viagem do mesmo porte. Os dados encontram-se abaixo:

3810	3690	3350	3400	3320
3250	3430	3600	3670	

Construa e interprete um intervalo de 92% de confiança para o faturamento médio mensal de uma agência de viagem desse porte.

Respostas: $\overline{X}$ = 3502,22 ; $z_{\frac{\alpha}{2}}$ =1,75 ; I.C.=[3426,38 ; 3578,05]

Utilizando o Python:

```
from scipy.stats import norm
import numpy as np

# Faturamentos mensais observados
faturamentos = [3810, 3690, 3350, 3400, 3320, 3250, 3430, 3600, 3670]

# Parâmetros do problema
desvio_padrao_populacional = 130  # desvio padrão populacional
tamanho_amostra = len(faturamentos)  # tamanho da amostra
nivel_confianca = 0.92  # nível de confiança

# Calculando a média amostral dos faturamentos mensais
media_amostral = np.mean(faturamentos)

# Calculando o erro padrão
erro_padrao = desvio_padrao_populacional / np.sqrt(tamanho_amostra)

# Calculando o intervalo de confiança
z_score = norm.ppf((1 + nivel_confianca) / 2)  # valor z para o nível de confiança
margem_de_erro = z_score * erro_padrao
intervalo_confianca = (
    media_amostral - margem_de_erro,
    media_amostral + margem_de_erro,
)

# Imprimindo os resultados
print("Intervalo de Confiança (92%) para o Faturamento Médio Mensal (em dólares):", intervalo_confianca)
print("Média Amostral dos Faturamentos Mensais:", media_amostral)
print("Valor Crítico (z-score):", z_score)
print("Margem de Erro:", margem_de_erro)
```

```
Intervalo de Confiança (92%) para o Faturamento Médio Mensal (em dólares): (3426.359159134628, 3578.0852853098163)
Média Amostral dos Faturamentos Mensais: 3502.222222222222
Valor Crítico (z-score): 1.7506860712521692
Margem de Erro: 75.863063087594
```

Interpretação: Com 92% de confiança, podemos afirmar que o verdadeiro faturamento médio mensal da agência de viagens está entre US$ 3426,36 e US$ 3578,09. A média amostral dos faturamentos mensais observados foi de US$ 3502,22. O valor crítico (z-score) para um intervalo de confiança de 92% é aproximadamente 1,75 e a margem de erro é de aproximadamente US$ 75,86.

Determinação do tamanho da amostra

Exemplo 2

(Triola, 1998) Um economista deseja estimar a renda média de bacharéis em Economia que tiveram a feliz ideia de fazer um curso de Estatística. Quantos valores de renda devem ser tomados, se o economista deseja ter 95% de confiança em que a média amostral esteja a menos de R$500,00 da verdadeira média populacional? Suponha que saibamos, por um estudo prévio, que, para tais rendas, σ = R$6250,00.

Utilizando o Python:

```
import scipy.stats as stats
import math

# Parâmetros do problema
confianca = 0.95  # 95% de confiança
sigma = 6250  # desvio padrão populacional
margem_erro = 500  # margem de erro desejada

# Calculando o valor crítico (Z)
z = stats.norm.ppf((1 + confianca) / 2)

# Calculando o tamanho da amostra necessário e arredondando para cima
n = math.ceil(((z * sigma) / margem_erro) ** 2)

print("Valor crítico (Z):", z)
print("Tamanho da amostra necessário (arredondado para cima):", n)

Valor crítico (Z): 1.959963984540054
Tamanho da amostra necessário (arredondado para cima): 601
```

Observação:

Nesse código, utilizamos a função stats.norm.ppf que é usada para calcular o percentil de uma distribuição normal padrão (ou seja, média 0 e desvio padrão 1) para um determinado valor de probabilidade. Também foi usada a função math.ceil() que é uma função da biblioteca padrão do Python que arredonda para cima um número decimal para o próximo número inteiro.

Agora é a sua vez!

Refaça esse exercício considerando agora:
a) Margem de erro igual a R$1000,00.
b) Margem de erro igual a R$250,00 e nível de confiança de 85%.

Respostas:
a) n=150,0625, ou seja, 151
b) $z_{\frac{\alpha}{2}}$ =1,44, n=1296

Utilizando o Python:

a) Margem de erro igual a R$1000,00.

```
import scipy.stats as stats
import math

# Parâmetros do problema
confianca = 0.95  # 95% de confiança
sigma = 6250  # desvio padrão populacional
margem_erro_a = 1000  # margem de erro desejada

# Calculando o valor crítico (Z)
z = stats.norm.ppf((1 + confianca) / 2)

# Calculando o tamanho da amostra necessário e arredondando para cima
n_a = math.ceil(((z * sigma) / margem_erro_a) ** 2)

print("a) Valor crítico (Z):", z)
print("a) Tamanho da amostra necessário (arredondado para cima):", n_a)

a) Valor crítico (Z): 1.959963984540054
a) Tamanho da amostra necessário (arredondado para cima): 151
```

b) Margem de erro igual a R$250,00 e nível de confiança de 85%.

```
# Parâmetros do problema (cenário b)
confianca_b = 0.85  # 85% de confiança
margem_erro_b = 250  # margem de erro desejada

# Calculando o valor crítico (Z) para o novo nível de confiança
z_b = stats.norm.ppf((1 + confianca_b) / 2)

# Calculando o tamanho da amostra necessário e arredondando para cima
n_b = math.ceil(((z_b * sigma) / margem_erro_b) ** 2)

print("b) Valor crítico (Z) para 85% de confiança:", z_b)
print("b) Tamanho da amostra necessário (arredondado para cima):", n_b)

b) Valor crítico (Z) para 85% de confiança: 1.4395314709384563
b) Tamanho da amostra necessário (arredondado para cima): 1296
```

Exemplo 3

Um pesquisador deseja estimar a renda média para o primeiro ano de trabalho de bacharéis em Direito formados por uma determinada instituição. Sabe-se que no último ano 420 bacharéis formaram-se nessa faculdade. Quantos valores de renda devem ser tomados, se o pesquisador deseja ter 90% de confiança em que a média amostral esteja a menos de R$600 da verdadeira média populacional? Suponha que saibamos, por um estudo prévio, que, para tais rendas, σ = R$7000,00.

Utilizando o Python:

```
import math
from scipy.stats import norm

def tamanho_amostral_para_media(N, sigma, E, confianca):
    # Calculando o valor crítico com base no nível de confiança
    z = norm.ppf((1 + confianca) / 2)

    # Calculando o tamanho da amostra necessário
    n = math.ceil((N * sigma**2 * z**2) / ((N - 1) * E**2 + sigma**2 * z**2))

    return n

# Valores dados
N = 420  # Tamanho da população
sigma = 7000  # Desvio padrão populacional
E = 600  # Margem de erro desejada
confianca = 0.90  # Nível de confiança desejado

# Calculando o tamanho amostral necessário
tamanho_amostral = tamanho_amostral_para_media(N, sigma, E, confianca)

print("O pesquisador precisa tomar pelo menos", tamanho_amostral, "valores de renda.")
```

```
O pesquisador precisa tomar pelo menos 197 valores de renda.
```

Agora é a sua vez!
Refaça esse exercício considerando:
a) Tamanho populacional desconhecido (Mantendo-se as outras características do enunciado).
b) Margem de erro igual a R$1200,00 (Mantendo-se as outras características do enunciado).
c) Nível de confiança de 80% (Mantendo-se as outras características do enunciado).

Respostas:
a) n=367 rendas
b) n=76 rendas
c) $z_{\frac{\alpha}{2}}$ =1,28 , n=145,8897, ou seja, 146 rendas

a) Tamanho populacional desconhecido (Mantendo-se as outras características do enunciado).

Resposta: n=367 rendas

Utilizando o Python:

```
import math
from scipy.stats import norm

def tamanho_amostral_para_media(sigma, E, confianca):
    # Calculando o valor crítico com base no nível de confiança
    Z = norm.ppf((1 + confianca) / 2)

    # Calculando o tamanho da amostra necessário
    n = math.ceil((Z * sigma / E) ** 2)

    return n

# Valores dados
sigma = 7000  # Desvio padrão populacional
E = 600  # Margem de erro desejada
confianca = 0.90  # Nível de confiança desejado

# Calculando o tamanho amostral necessário
tamanho_amostral = tamanho_amostral_para_media(sigma, E, confianca)

print("O pesquisador precisa tomar pelo menos", tamanho_amostral, "valores de renda.")
```

O pesquisador precisa tomar pelo menos 369 valores de renda.

b) Margem de erro igual a R$1200,00 (Mantendo-se as outras características do enunciado).

Resposta: n=76 rendas

Utilizando o Python:

```
import math
from scipy.stats import norm

def tamanho_amostral_para_media(N, sigma, E, confianca):
  # Calculando o valor crítico com base no nível de confiança
  z = norm.ppf((1 + confianca) / 2)

  # Calculando o tamanho da amostra necessário
  n = math.ceil((N * sigma**2 * z**2) / ((N - 1) * E**2 + sigma**2 * z**2))

  return n

# Valores dados
N = 420  # Tamanho da população
sigma = 7000  # Desvio padrão populacional
E = 1200  # Margem de erro desejada
confianca = 0.90  # Nível de confiança desejado

# Calculando o tamanho amostral necessário
tamanho_amostral = tamanho_amostral_para_media(N, sigma, E, confianca)

print("O pesquisador precisa tomar pelo menos", tamanho_amostral, "valores de renda.")
```

O pesquisador precisa tomar pelo menos 76 valores de renda.

c) Nível de confiança de 80% (Mantendo-se as outras características do enunciado).

Resposta: n=146 rendas

Utilizando o Python:

```
import math
from scipy.stats import norm

def tamanho_amostral_para_media(N, sigma, E, confianca):
    # Calculando o valor crítico com base no nível de confiança
    z = norm.ppf((1 + confianca) / 2)

    # Calculando o tamanho da amostra necessário
    n = math.ceil((N * sigma**2 * z**2) / ((N - 1) * E**2 + sigma**2 * z**2))

    return n

# Valores dados
N = 420  # Tamanho da população
sigma = 7000  # Desvio padrão populacional
E = 600  # Margem de erro desejada
confianca = 0.80  # Nível de confiança desejado

# Calculando o tamanho amostral necessário
tamanho_amostral = tamanho_amostral_para_media(N, sigma, E, confianca)

print("O pesquisador precisa tomar pelo menos", tamanho_amostral, "valores de renda.")
```

O pesquisador precisa tomar pelo menos 147 valores de renda.

Exercício 3

Uma pesquisa é planejada para determinar as despesas médicas anuais das famílias dos empregados de uma empresa. A gerência da empresa deseja ter 95% de confiança de que a média da amostra está no máximo com uma margem de erro de $50 da média real das despesas médicas familiares. Um estudo-piloto indica que o desvio padrão pode ser considerado como sendo igual a $400.

a) Qual o tamanho de amostra necessário?

b) Se a gerência deseja estar certa em uma margem de erro de $25, que tamanho de amostra será necessário?

c) Sabe-se que a empresa tem, atualmente, 386 empregados. Qual deve ser o tamanho amostral necessário para termos um nível de 92% de confiança? (Mantendo-se inalteradas as outras informações do enunciado)

Respostas:
a) n=245,8624, ou seja, 246 empregados.
b) n=983,4496, ou seja, 984 empregados.
c) $z_{\frac{\alpha}{2}}$ =1,75 , n=130,2169, ou seja, 131 empregados.

Utilizando o Pyhton:

Letra a)

```
import math
from scipy.stats import norm

def tamanho_amostral(sigma, E, confianca):
    # Calculando o valor crítico com base no nível de confiança
    Z = norm.ppf((1 + confianca) / 2)

    # Calculando o tamanho da amostra necessário
    n = math.ceil((Z * sigma / E) ** 2)

    return n

# Valores dados
sigma = 400  # Desvio padrão populacional
E = 50  # Margem de erro
confianca = 0.95  # Nível de confiança desejado

# Calculando o tamanho amostral necessário
tamanho_amostral = tamanho_amostral(sigma, E, confianca)

print("O tamanho de amostra necessário é:", tamanho_amostral)
```

O tamanho de amostra necessário é: 246

Letra b)

```
import math
from scipy.stats import norm

def tamanho_amostral(sigma, E, confianca):
    # Calculando o valor crítico com base no nível de confiança
    Z = norm.ppf((1 + confianca) / 2)

    # Calculando o tamanho da amostra necessário
    n = math.ceil((Z * sigma / E) ** 2)

    return n

# Valores dados
sigma = 400  # Desvio padrão populacional
E = 25  # Nova margem de erro
confianca = 0.95  # Nível de confiança desejado

# Calculando o tamanho amostral necessário
tamanho_amostral = tamanho_amostral(sigma, E, confianca)

print("O tamanho de amostra necessário para uma margem de erro de $25 é:", tamanho_amostral)
```

O tamanho de amostra necessário para uma margem de erro de $25 é: 984

Letra c)

```
import math
from scipy.stats import norm

def tamanho_amostral_para_media(N, sigma, E, confianca):
    # Calculando o valor crítico com base no nível de confiança
    z = norm.ppf((1 + confianca) / 2)

    # Calculando o tamanho da amostra necessário
    n = math.ceil((N * sigma**2 * z**2) / ((N - 1) * E**2 + sigma**2 * z**2))

    return n

# Valores dados
N = 386  # Tamanho da população
sigma = 400  # Desvio padrão populacional
E = 50  # Margem de erro
confianca = 0.92  # Nível de confiança desejado

# Calculando o tamanho amostral necessário
tamanho_amostral = tamanho_amostral_para_media(N, sigma, E, confianca)

print("O tamanho amostral necessário é:", tamanho_amostral)
```

O tamanho amostral necessário é: 131

Intervalo de confiança para a média quando σ é desconhecido

Exemplo 4

(Triola, 1998) Com um teste destrutivo, as amostras são destruídas no processo de teste. O teste de colisão de carros é um exemplo muito dispendioso de teste destrutivo. Suponha que tenhamos feito teste de colisão em 12 carros esporte sob uma diversidade de condições que simulam colisões típicas. A análise dos 12 carros danificados resulta em custos de conserto com média igual a US$26.227,00 e um desvio padrão de US$15.873,00. Sabe-se que os dados têm distribuição aproximadamente normal. Determine:

a) A melhor estimativa pontual de μ, o custo médio de conserto de todos os carros esporte envolvidos em colisões.
b) A estimativa intervalar de 95% de confiança para μ.

Utilizando o Python:

```
import scipy.stats as stats

# Dados
media_amostral = 26227.00  # Média amostral
desvio_padrao = 15873.00  # Desvio padrão
n = 12  # Tamanho da amostra
nivel_confianca = 0.95  # Nível de confiança desejado

# a) Melhor estimativa pontual da média populacional
melhor_estimativa_pontual = media_amostral
print("a) A melhor estimativa pontual da média populacional é:", melhor_estimativa_pontual)

# b) Estimativa intervalar de 95% de confiança para a média populacional
graus_liberdade = n - 1
valor_critico_t = stats.t.ppf((1 + nivel_confianca) / 2, graus_liberdade)
erro_padrao = desvio_padrao / (n ** 0.5)
intervalo_confianca = valor_critico_t * erro_padrao

limite_inferior = media_amostral - intervalo_confianca
limite_superior = media_amostral + intervalo_confianca

print("b) A estimativa intervalar de 95% de confiança para a média populacional é: (${:.2f}, ${:.2f})".format(limite_inferior, limite_superior))
print("   Valor crítico da distribuição t-Student para o nível de confiança de 95%:", valor_critico_t)
print("   Margem de erro para o intervalo de confiança:", intervalo_confianca)
```

```
a) A melhor estimativa pontual da média populacional é: 26227.0
b) A estimativa intervalar de 95% de confiança para a média populacional é: ($16141.78, $36312.22)
   Valor crítico da distribuição t-Student para o nível de confiança de 95%: 2.200985160082949
   Margem de erro para o intervalo de confiança: 10085.223046959425
```

Exercício 4

Em uma pesquisa de orçamento familiar desenvolvida pelo instituto ZX, solicitou-se a 16 domicílios de certa região que anotassem suas despesas com alimentação durante uma semana. O resultado foi uma despesa média de R$330,00 com um desvio padrão de R$40,00. Construa e interprete um intervalo de 98% de confiança para a verdadeira despesa média semanal com alimentação por domicílio de toda a região. Suponha que a população tenha uma distribuição aproximadamente normal.

Repostas: $t_{\frac{\alpha}{2}}$ =2,602 ; I.C.=[303,98 ; 356,02]

Utilizando o Python:

```
import scipy.stats as stats

# Dados da amostra
media_amostra = 330.00  # média da amostra
desvio_padrao_amostra = 40.00  # desvio padrão da amostra
tamanho_amostra = 16  # tamanho da amostra

# Nível de confiança
nivel_confianca = 0.98

# Graus de liberdade
graus_liberdade = tamanho_amostra - 1

# Calculando o valor crítico da distribuição t-Student
valor_critico_t = stats.t.ppf((1 + nivel_confianca) / 2, graus_liberdade)

# Calculando a margem de erro
margem_erro = valor_critico_t * (desvio_padrao_amostra / (tamanho_amostra ** 0.5))

# Calculando o intervalo de confiança
limite_inferior = media_amostra - margem_erro
limite_superior = media_amostra + margem_erro

# Imprimindo o valor crítico e a margem de erro
print("Valor crítico da distribuição t-Student:", valor_critico_t)
print("Margem de erro:", margem_erro)

# Imprimindo o intervalo de confiança
print("Intervalo de confiança de {}%:".format(nivel_confianca * 100))
print("({:.2f}, {:.2f})".format(limite_inferior, limite_superior))

Valor crítico da distribuição t-Student: 2.602480294995493
Margem de erro: 26.02480294995493
Intervalo de confiança de 98.0%:
(303.98, 356.02)
```

Intervalo de confiança para a proporção

Exemplo 5

(Adaptado de Morettin, 2004) Em uma linha de produção de certa peça cerâmica, colheu-se uma amostra de 125 itens, constatando-se 7 peças eram defeituosas. Com esses resultados amostrais, determine a estimativa intervalar de 87% de confiança da proporção de peças defeituosas produzidas pela empresa.

Utilizando o Python:

```
import scipy.stats as stats

# Dados da amostra
n = 125  # tamanho da amostra
defeituosas = 7  # número de peças defeituosas na amostra

# Proporção amostral
p_hat = defeituosas / n

# Nível de confiança
nivel_confianca = 0.87

# Calculando o valor crítico da distribuição normal
valor_critico_z = stats.norm.ppf((1 + nivel_confianca) / 2)

# Calculando a margem de erro
margem_erro = valor_critico_z * ((p_hat * (1 - p_hat)) / n) ** 0.5

# Calculando o intervalo de confiança
limite_inferior = p_hat - margem_erro
limite_superior = p_hat + margem_erro

# Imprimindo o valor crítico e a margem de erro
print("Valor crítico da distribuição normal:", valor_critico_z)
print("Margem de erro:", margem_erro)

# Imprimindo o intervalo de confiança
print("Intervalo de confiança de {}%:".format(nivel_confianca * 100))
print("({:.4f}, {:.4f})".format(limite_inferior, limite_superior))
```

Valor crítico da distribuição normal: 1.514101887619284
Margem de erro: 0.031137239346781177
Intervalo de confiança de 87.0%: (0.0249, 0.0871)

Agora é com você!

a) Encontre os intervalos com 80% e 99% de confiança.

b) Compare os intervalos obtidos.

Resposta:

a) 80% de confiança: $z_{\frac{\alpha}{2}}$ =1,28 ; E=0,0263 ; I.C.=[0,0297 ; 0,0823]

99% de confiança: $z_{\frac{\alpha}{2}}$ =2,57 ; E=0,0529 ; I.C.=[0,0031 ; 0,1089]

b) Ao aumentarmos o nível de confiança, houve um aumento da margem de erro e, consequentemente, um aumento da amplitude do intervalo.

Utilizando o Python:

80% de confiança:

```
import scipy.stats as stats

# Dados da amostra
n = 125  # tamanho da amostra
defeituosas = 7  # número de peças defeituosas na amostra

# Proporção amostral
p_hat = defeituosas / n

# Nível de confiança
nivel_confianca = 0.8

# Calculando o valor crítico da distribuição normal
valor_critico_z = stats.norm.ppf((1 + nivel_confianca) / 2)

# Calculando a margem de erro
margem_erro = valor_critico_z * ((p_hat * (1 - p_hat)) / n) ** 0.5

# Calculando o intervalo de confiança
limite_inferior = p_hat - margem_erro
limite_superior = p_hat + margem_erro

# Imprimindo o valor crítico e a margem de erro
print("Valor crítico da distribuição normal:", valor_critico_z)
print("Margem de erro:", margem_erro)

# Imprimindo o intervalo de confiança
print("Intervalo de confiança de {}%:".format(nivel_confianca * 100))
print("({:.4f}, {:.4f})".format(limite_inferior, limite_superior))
```

Valor crítico da distribuição normal: 1.2815515655446004
Margem de erro: 0.026354882823868504
Intervalo de confiança de 80.0%: (0.0296, 0.0824)

99% de confiança:

```
import scipy.stats as stats

# Dados da amostra
n = 125  # tamanho da amostra
defeituosas = 7  # número de peças defeituosas na amostra

# Proporção amostral
p_hat = defeituosas / n

# Nível de confiança
nivel_confianca = 0.99

# Calculando o valor crítico da distribuição normal
valor_critico_z = stats.norm.ppf((1 + nivel_confianca) / 2)

# Calculando a margem de erro
margem_erro = valor_critico_z * ((p_hat * (1 - p_hat)) / n) ** 0.5

# Calculando o intervalo de confiança
limite_inferior = p_hat - margem_erro
limite_superior = p_hat + margem_erro

# Imprimindo o valor crítico e a margem de erro
print("Valor crítico da distribuição normal:", valor_critico_z)
print("Margem de erro:", margem_erro)

# Imprimindo o intervalo de confiança
print("Intervalo de confiança de {}%:".format(nivel_confianca * 100))
print("({:.4f}, {:.4f})".format(limite_inferior, limite_superior))
```

Valor crítico da distribuição normal: 2.5758293035489004
Margem de erro: 0.052971477148849484
Intervalo de confiança de 99.0%: (0.0030, 0.1090)

Exercício 5

A figura abaixo mostra os resultados de uma pesquisa com 400 homens, 500 mulheres, 650 pessoas que usam frequentemente o forno microondas e 50 pessoas que raramente o usam. Foi perguntado se eles eram favoráveis à irradiação da carne vermelha para matar micróbios transmissores de doenças. (Fonte: Peter D. Hart Research Associetes for Grocery Manufactures of America)

Tratar a Carne?	
Pessoas favoráveis à irradiação de alimentos para eliminar micróbios infecciosos:	
	A favor
Homens	61%
Mulheres	44%
Pessoas que usam frequentemente o microondas	55%
Pessoas que raramente usam o microondas	40%

Construa um intervalo de 98% de confiança para:

a.) A proporção dos usuários frequentes do microondas favoráveis à irradiação da carne vermelha.
b.) A proporção dos usuários esporádicos do microondas favoráveis à irradiação da carne vermelha.
c.) A proporção de mulheres que são contra a irradiação da carne vermelha.

Respostas: $z_{\frac{\alpha}{2}}$ =2,33, a) [50,45% ; 59,55%], b) [23,86% ; 56,14%], c) [50,83% ; 61,17%]

Utilizando o Python:

Letra a)

```
import scipy.stats as stats

# Dados da pesquisa
n_freq_microondas = 650
favoraveis_freq_microondas = 0.55

# Nível de confiança
nivel_confianca = 0.98

# Função para calcular o intervalo de confiança para uma proporção
def calcular_intervalo_proporcao(n, p_hat):
   # Calculando o valor crítico da distribuição normal
   valor_critico_z = stats.norm.ppf((1 + nivel_confianca) / 2)

   # Calculando a margem de erro
   margem_erro = valor_critico_z * ((p_hat * (1 - p_hat)) / n) ** 0.5

   # Calculando o intervalo de confiança
   limite_inferior = p_hat - margem_erro
   limite_superior = p_hat + margem_erro
```

    return limite_inferior, limite_superior, margem_erro

# Calculando o intervalo de confiança para a proporção de usuários frequentes de microondas favoráveis à irradiação da carne vermelha
intervalo_freq_microondas = calcular_intervalo_proporcao(n_freq_microondas, favoraveis_freq_microondas)

# Imprimindo o resultado
print("Intervalo de confiança para a proporção de usuários frequentes de microondas favoráveis à irradiação da carne vermelha: {:.2f}% - {:.2f}% (margem de erro: {:.2f}%)".format(intervalo_freq_microondas[0]*100, intervalo_freq_microondas[1]*100, intervalo_freq_microondas[2]*100))

Intervalo de confiança para a proporção de usuários frequentes de microondas favoráveis à irradiação da carne vermelha: 50.46% - 59.54% (margem de erro: 4.54%)

Letra b)
# Dados da pesquisa
n_raros_microondas = 50
favoraveis_raros_microondas = 0.40

# Calculando o intervalo de confiança para a proporção de usuários esporádicos de microondas favoráveis à irradiação da carne vermelha
intervalo_raros_microondas = calcular_intervalo_proporcao(n_raros_microondas, favoraveis_raros_microondas)

# Imprimindo o resultado
print("Intervalo de confiança para a proporção de usuários esporádicos de microondas favoráveis à irradiação da carne vermelha: {:.2f}% - {:.2f}% (margem de erro: {:.2f}%)".format(intervalo_raros_microondas[0]*100, intervalo_raros_microondas[1]*100, intervalo_raros_microondas[2]*100))

Intervalo de confiança para a proporção de usuários esporádicos de microondas favoráveis à irradiação da carne vermelha: 23.88% - 56.12% (margem de erro: 16.12%)

Letra c)
# Dados da pesquisa
n_mulheres = 500
contra_irradiacao_mulheres = 0.56

# Calculando o intervalo de confiança para a proporção de mulheres contra a irradiação da carne vermelha
intervalo_contra_irradiacao_mulheres = calcular_intervalo_proporcao(n_mulheres, contra_irradiacao_mulheres)

# Imprimindo o resultado
print("Intervalo de confiança para a proporção de mulheres contra a irradiação da carne vermelha: {:.2f}% - {:.2f}% (margem de erro: {:.2f}%)".format(intervalo_contra_irradiacao_mulheres[0]*100, intervalo_contra_irradiacao_mulheres[1]*100, intervalo_contra_irradiacao_mulheres[2]*100))

Intervalo de confiança para a proporção de mulheres contra a irradiação da carne vermelha: 50.84% - 61.16% (margem de erro: 5.16%)

Determinação do tamanho da amostra

Exemplo 6

(Triola, 1998) As companhias de seguro estão ficando preocupadas com o fato de que o número crescente de telefones celulares resulte em maior número de colisões de carros; estão, por isso, pensando em cobrar prêmios mais elevados para os motoristas que utilizam celulares. Desejamos estimar, com uma margem de erro de três pontos percentuais, a percentagem de motoristas que falam ao celular enquanto estão dirigindo. Supondo que se pretende um nível de 95% de confiança nos resultados, quantos motoristas devem ser pesquisados?

a) Suponha que tenhamos uma estimativa, com base em estudos anteriores, de 18% de motoristas que falam ao celular.
b) Suponha que não tenhamos qualquer informação que possa sugerir um valor de $\hat{p}$.

Utilizando o Python:

a) Estimativa com base em estudos anteriores:

```
import scipy.stats as stats
import math

# Estimativa da proporção de motoristas que falam ao celular com base em estudos anteriores
p_estimado = 0.18

# Margem de erro desejada
margem_erro = 0.03

# Nível de confiança
nivel_confianca = 0.95

# Calculando o valor crítico da distribuição normal
valor_critico_z = stats.norm.ppf((1 + nivel_confianca) / 2)

# Calculando o tamanho da amostra
n = ((valor_critico_z ** 2) * p_estimado * (1 - p_estimado)) / (margem_erro ** 2)
n = math.ceil(n)  # Arredondando para cima

# Imprimindo o resultado
print("Número de motoristas que devem ser pesquisados (estimativa baseada em estudos anteriores):", n)
```

Número de motoristas que devem ser pesquisados (estimativa baseada em estudos anteriores): 630

b) Sem informação prévia sobre a proporção amostral:

```
import scipy.stats as stats
import math

# Estimativa conservadora da proporção amostral
p_estimado = 0.5

# Margem de erro desejada
margem_erro = 0.03

# Nível de confiança
nivel_confianca = 0.95

# Calculando o valor crítico da distribuição normal
valor_critico_z = stats.norm.ppf((1 + nivel_confianca) / 2)

# Calculando o tamanho da amostra com base na estimativa conservadora
n = ((valor_critico_z ** 2) * p_estimado * (1 - p_estimado)) / (margem_erro ** 2)
n = math.ceil(n)  # Arredondando para cima

# Imprimindo o resultado
print("Número de motoristas que devem ser pesquisados (sem informação prévia sobre a proporção
amostral):", n)
```

Número de motoristas que devem ser pesquisados (sem informação prévia sobre a proporção amostral): 1068

Exemplo 7

O reitor de uma universidade deseja estimar com uma margem de erro de 5% e um nível de confiança de 97%, a proporção de estudantes matriculados nos programas de MBA que fizeram graduação em Sistemas de Informação. Sabe-se que, atualmente, há 550 alunos fazendo MBA. Qual tamanho mínimo de amostra deve ser coletado?

a) Estima-se que a proporção populacional seja de 23%.

b) Não há base para estimar o valor aproximado da proporção populacional.

Utilizando o Python:

a) Estimativa com base na proporção populacional:
Neste caso, temos uma estimativa prévia da proporção populacional de estudantes matriculados nos programas de MBA que fizeram graduação em Sistemas de Informação, que é de 23%. Vamos calcular o tamanho da amostra necessário para uma margem de erro de 5% e um nível de confiança de 97%.

```
import scipy.stats as stats
import math

# Dados do problema
N = 550  # Tamanho da população
p_estimado = 0.23  # Proporção populacional estimada

# Margem de erro desejada
E = 0.05

# Nível de confiança
nivel_confianca = 0.97

# Calculando o valor crítico da distribuição normal
valor_critico_z = stats.norm.ppf((1 + nivel_confianca) / 2)

# Calculando o tamanho da amostra
n = (N * p_estimado * (1 - p_estimado) * (valor_critico_z ** 2)) / ((p_estimado * (1 - p_estimado) *
(valor_critico_z ** 2)) + ((N - 1) * (E ** 2)))
n = math.ceil(n)  # Arredondando para cima

# Imprimindo o resultado
print("Tamanho mínimo de amostra necessário (estimativa baseada na proporção populacional):", n)
```

Tamanho mínimo de amostra necessário (estimativa baseada na proporção populacional): 208

b) Sem base para estimar a proporção populacional:
Neste caso, não temos uma estimativa prévia da proporção populacional. Vamos calcular o tamanho da amostra necessário considerando uma proporção amostral de 0.5 (máxima variabilidade) para garantir o tamanho da amostra mais conservador.

```
# Proporção conservadora amostral
p_estimado = 0.5

# Calculando o tamanho da amostra com base na estimativa conservadora
n = (N * p_estimado * (1 - p_estimado) * (valor_critico_z ** 2)) / ((p_estimado * (1 - p_estimado) *
(valor_critico_z ** 2)) + ((N - 1) * (E ** 2)))
n = math.ceil(n)  # Arredondando para cima

# Imprimindo o resultado
print("Tamanho mínimo de amostra necessário (sem base para estimar a proporção populacional):", n)
```

Tamanho mínimo de amostra necessário (sem base para estimar a proporção populacional): 254

Exercício 6

Suponha que você vai realizar um levantamento para estimar a proporção de crianças matriculadas em um programa de saúde infantil na cidade de Nova Lima-MG. Sabe-se, segundo o IBGE, que o município apresentou no ano de 2014 o número de 1286 nascidos
vivos e registrados na cidade. Considerando 90% de confiança e um erro amostral de 4%, determine:
a) O tamanho da amostra.
b) Considerando agora uma proporção de 70% de crianças matriculadas no programa
de saúde, qual seria o tamanho da amostra?

Respostas: a) n=316,9280, ou seja, 317 crianças. b) n=277,1478, ou seja, 278 crianças.

Utilizando o Python:

a) Tamanho da amostra com base apenas no tamanho da população:
Neste caso, vamos calcular o tamanho da amostra necessário considerando apenas o tamanho da população de crianças nascidas vivas e registradas em Nova Lima-MG. Vamos considerar um nível de confiança de 90% e uma margem de erro de 4%.

```
import scipy.stats as stats
import math

# Dados do problema
N = 1286  # Tamanho da população
E = 0.04  # Margem de erro
nivel_confianca = 0.90  # Nível de confiança

# Calculando o valor crítico da distribuição normal
valor_critico_z = stats.norm.ppf((1 + nivel_confianca) / 2)

# Calculando o tamanho da amostra
n = (N * 0.5 * 0.5 * (valor_critico_z ** 2)) / ((0.5 * 0.5 * (valor_critico_z ** 2)) + ((N - 1) * (E ** 2)))
n = math.ceil(n)  # Arredondando para cima

# Imprimindo o resultado
print("Valor crítico:", valor_critico_z)
print("Tamanho mínimo de amostra necessário (considerando apenas o tamanho da população):", n)
```

Valor crítico: 1.6448536269514722

Tamanho mínimo de amostra necessário (considerando apenas o tamanho da população): 319

b) Tamanho da amostra com base na proporção e no tamanho da população:
Neste caso, além do tamanho da população, temos uma estimativa da proporção de crianças matriculadas no programa de saúde infantil (70%). Vamos calcular o tamanho da amostra considerando um nível de confiança de 90% e uma margem de erro de 4%.

```
# Proporção estimada de crianças matriculadas no programa de saúde infantil
p_estimado = 0.70

# Calculando o tamanho da amostra
n = (N * p_estimado * (1 - p_estimado) * (valor_critico_z ** 2)) / ((p_estimado * (1 - p_estimado) *
(valor_critico_z ** 2)) + ((N - 1) * (E ** 2)))
n = math.ceil(n)  # Arredondando para cima

# Imprimindo o resultado
print("Tamanho mínimo de amostra necessário (considerando proporção e tamanho da população):", n)
```

Tamanho mínimo de amostra necessário (considerando proporção e tamanho da população): 279