

Parte II: Prática (usando a linguagem Python)

Leitura da base de dados survey:

```
import pandas as pd

# URL do arquivo CSV
url = 'https://r-data.pmagonia.com/system/files/datasets/dataset-66172.csv' # Link do dataset

# Ler o arquivo CSV a partir do URL
survey = pd.read_csv(url)

# Exibir o DataFrame
print(survey)
```

```

0      Female    18.5    18.0    Right    R on L    92.0    Left    Some    Never
1      Male     19.5    20.5    Left     R on L   104.0    Left    None    Regul
2      Male     18.0    13.3    Right    L on R    87.0    Neither  None    Occas
3      Male     18.8    18.9    Right    R on L     NaN    Neither  None    Never
4      Male     20.0    20.0    Right    Neither   35.0     Right    Some    Never
..      ...      ...      ...      ...      ...      ...      ...      ...      ...
232 Female     18.0    18.0    Right    L on R    85.0     Right    Some    Never
233 Female     18.5    18.0    Right    L on R    88.0     Right    Some    Never
234 Female     17.5    16.5    Right    R on L     NaN     Right    Some    Never
235 Male      21.0    21.5    Right    R on L    90.0     Right    Some    Never
236 Female     17.6    17.3    Right    R on L    85.0     Right    Freq    Never

      Height      M.I      Age
0      173.0    Metric   18.250
1      177.8  Imperial   17.583
2       NaN      NaN    16.917
3      160.0    Metric   20.333
4      165.0    Metric   23.667
..      ...      ...      ...
232   165.1  Imperial   17.667
233   160.0    Metric   16.917
234   170.0    Metric   18.583
235   183.0    Metric   17.167
236   168.5    Metric   17.750

[237 rows x 12 columns]
```

1) Encontre uma estimativa pontual da idade média do estudante universitário com os dados de amostra da pesquisa survey.

Solução:

```
import pandas as pd

# Salvar a idade do aluno dos dados de pesquisa
age_survey = survey['Age']

# Encontrar a estimativa pontual da idade média do aluno
point_estimate = age_survey.mean()

# Imprimir a estimativa pontual
print(f"Estimativa pontual da idade média do aluno: {point_estimate} anos.")

Estimativa pontual da idade média do aluno: 20.37451476793249 anos.
```

```
import numpy as np
from scipy import stats

# Calcular o intervalo de confiança
confidence_level = 0.95
t_statistic = stats.t.ppf(1 - (1 - confidence_level) / 2, len(age_survey) - 1)
mean = np.mean(age_survey)
std_dev = np.std(age_survey, ddof=1)
margin_of_error = t_statistic * std_dev / np.sqrt(len(age_survey))
```

```
confidence_interval = (mean - margin_of_error, mean + margin_of_error)

# Imprimir o intervalo de confiança
print(f"Os intervalos de confiança para a idade média do estudante universitário são {confidence_interval[0]} - {confidence_interval[1]}")
print(f"\nPortanto, o nível de confiança de {confidence_level*100}% inclui a verdadeira média populacional que é igual a {mean} anos.")
```

Os intervalos de confiança para a idade média do estudante universitário são 19.54599675948782 - 21.203032776377157

Portanto, o nível de confiança de 95.0% inclui a verdadeira média populacional que é igual a 20.37451476793249 anos.

- ▼ **2) Assuma o desvio padrão da população σ da idade do aluno na pesquisa de dados é 7. Encontre a margem de erro e a estimativa de intervalo com nível de confiança de 95%.**

Solução:

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

# População de desvio padrão
sigma = 7

# Tamanho da amostra
n = len(age_survey)

# Erro padrão
SE = sigma / np.sqrt(n)

# Margem de erro
E = norm.ppf(0.975) * SE

# Média da amostra
xbar = np.mean(age_survey)

# Intervalo de confiança
confidence_interval = (xbar - E, xbar + E)

# Imprimir a média da amostra, a margem de erro e o intervalo de confiança
print(f"A margem de erro da idade do aluno assumindo que o desvio padrão da população é {sigma} no nível de confiança de 95% é {E} anos.")
print(f"\nA média da amostra é {xbar} anos.")
print(f"\nO intervalo de confiança para este caso está entre {confidence_interval[0]} e {confidence_interval[1]} anos.")
```

A margem de erro da idade do aluno assumindo que o desvio padrão da população é 7 no nível de confiança de 95% é 0.8911933927887157

A média da amostra é 20.37451476793249 anos.

O intervalo de confiança para este caso está entre 19.483321375143774 e 21.265708160721204 anos.

▼ **Solução alternativa:**

- O método `ztest` não está disponível na versão do SciPy instalada no Colab. Uma solução alternativa é calcular manualmente o valor do teste z usando os dados fornecidos.

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

# População de desvio padrão
sigma = 7

# Calcular a média da amostra
xbar = np.mean(age_survey)

# Calcular o desvio padrão da média
se = sigma / np.sqrt(len(age_survey))

# Calcular o valor do teste z
z = (xbar - sigma) / se

# Calcular o intervalo de confiança
confidence_interval = norm.interval(0.95, loc=xbar, scale=se)

# Imprimir os resultados
print(f"O valor do teste z é {z}.")
print(f"\nO intervalo de confiança para este caso está entre {confidence_interval[0]} e {confidence_interval[1]} anos.")
```

O valor do teste z é 29.4140053864397.

O intervalo de confiança para este caso está entre 19.483321375143774 e 21.265708160721204 anos.

- ▼ **3) Sem assumir o desvio padrão da população, σ , da idade do aluno na pesquisa, encontre a margem de erro e a estimativa do intervalo com nível de confiança de 95%.**

Solução:

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

# Filtrar valores ausentes
age_response = np.array(age_survey)[~np.isnan(age_survey)]

# Atribuir o comprimento
n = len(age_response)

# Desvio padrão amostral
s = 7

# Estimando o erro padrão
SE = s / np.sqrt(n)

# Margem de erro (cauda superior 95% do intervalo de confiança)
E = t.ppf(0.975, df=n-1) * SE

# Média da amostra
xbar = np.mean(age_response)

# Intervalo de confiança
confidence_interval = (xbar - E, xbar + E)

# Imprimir os resultados
print(f"A margem de erro para a pesquisa de idade do aluno é de {E} anos com nível de confiança de 95%.")
print(f"\nO intervalo de confiança está entre {confidence_interval[0]} e {confidence_interval[1]} anos.")
```

A margem de erro para a pesquisa de idade do aluno é de 0.8957871556235714 anos com nível de confiança de 95%.

O intervalo de confiança está entre 19.478727612308916 e 21.27030192355606 anos.

- ▼ **Solução alternativa:**

```
import numpy as np
from scipy.stats import ttest_1samp

# Filtrar valores ausentes
age_response = np.array(age_survey)[~np.isnan(age_survey)]

# Realizar o teste t de uma amostra
t_statistic, p_value = ttest_1samp(age_response, popmean=0)

# Imprimir os resultados
print(f"O valor do teste t é {t_statistic}.")
print(f"\nO valor de p é {p_value}.")
```

O valor do teste t é 48.44693269573472.

O valor de p é 1.2778887334192208e-124.

- ▼ **4) Melhore a qualidade de uma pesquisa amostral aumentando o tamanho da amostra com desvio padrão desconhecido, σ .**

Solução:

```
from scipy.stats import norm

# Variáveis de entrada
p = 0.5 # Probabilidade de sucesso (assumindo variabilidade máxima)
margin_error = 0.05 # Margem de erro desejada
confidence_level = 0.95 # Nível de confiança desejado

# Calcular o valor crítico Z
```

```

z_star = norm.ppf((1 + confidence_level) / 2)

# Calcular o tamanho da amostra necessário
sample_size = ((z_star ** 2) * p * (1 - p)) / (margin_error ** 2)

# Arredondar o tamanho da amostra para o próximo número inteiro
sample_size = int(round(sample_size))

# Imprimir o tamanho da amostra necessário
print(f"O tamanho da amostra necessário para melhorar a qualidade da pesquisa amostral é de aproximadamente {sample_size}.")

```

O tamanho da amostra necessário para melhorar a qualidade da pesquisa amostral é de aproximadamente 384.

5) Suponha que você não tenha uma estimativa de proporção planejada, encontre o tamanho da amostra necessário para atingir uma margem de erro de 5% para a pesquisa de estudantes do sexo masculino com um nível de confiança de 95%.

Solução:

```

import numpy as np
from scipy.stats import norm

# Dados de resposta de gênero (substitua pela coluna correta da base de dados)
sex = survey["Sex"]

# Número total de respostas
n = len(sex)

# Número de respostas do sexo masculino
k = np.sum(np.array(sex) == "Male")

# Proporção de respostas do sexo masculino
p_bar = k / n

# Valor crítico Z para o nível de confiança desejado
z_star = norm.ppf(0.975) # Para um nível de confiança de 95%

# Margem de erro desejada
E = 0.05

# Tamanho da amostra necessário
sample_size = (z_star ** 2 * p_bar * (1 - p_bar)) / (E ** 2)

# Arredondar o tamanho da amostra para o próximo número inteiro
sample_size = int(np.ceil(sample_size))

# Imprimir o tamanho da amostra necessário
print(f"O tamanho da amostra necessário para a pesquisa de estudantes do sexo masculino é de aproximadamente {sample_size}.")
print(f"\nO número de alunos do sexo masculino é {k}.")
print(f"\nA proporção de alunos do sexo masculino é {p_bar}.")

```

O tamanho da amostra necessário para a pesquisa de estudantes do sexo masculino é de aproximadamente 385.

O número de alunos do sexo masculino é 118.

A proporção de alunos do sexo masculino é 0.4978902953586498.