

Análise de Serviços de Streaming

Projeto Semestral de Estatística



Integrantes

Luan Teixeira

Bruno Davidovitch Bertanha

Joao Paulo M Socio

Ubiratan da Motta Filho

R.A - 20.01681-6

R.A - 20.01521-6

R.A - 20.00704-3

R.A - 20.00928-3

Sumário

Objetivo do projeto	2
Introdução	2
Resultados	2
Estimativa da média populacional do preço que os usuários consideram justo para um serviço de streaming (variável quantitativa)	2
Estimativa da proporção populacional de usuários que assinam mais de um serviço (variável qualitativa)	4
Estimativas do tamanho da amostra com metade do erro	5

Objetivo do projeto

Estimação de parâmetros da população e dimensionamento amostral.

Introdução

Desde a consolidação dos serviços de streaming como um dos principais meios de entretenimento do mercado global, por se tratar de um setor recente (que começou a se popularizar com a consolidação da Netflix em meados de 2010) ainda possui várias possibilidades e caminhos ainda não explorados. Tendo isso em mente, procuramos realizar uma pesquisa para analisar quais são as preferências e padrões de consumo de serviços de streaming, especificamente no estado de São Paulo, Brasil.

Para entendermos melhor o comportamento dos usuários de streaming, é necessário estimar os parâmetros da população por meio dos dados coletados da nossa amostra. Então, vamos estimar a média populacional do preço que os usuários consideram justo para um serviço de streaming e a proporção populacional de usuários que assinam mais de um serviço. Além disso, vamos calcular o tamanho da amostra necessário para reduzir pela metade o erro – tanto a partir da média populacional quanto a partir da proporção populacional encontradas.

Resultados

Estimativa da média populacional do preço que os usuários consideram justo para um serviço de streaming (variável quantitativa)

Para estimar a média populacional do preço que os usuários consideram justo para um serviço de streaming, como em nossa pesquisa perguntamos o intervalo do preço, necessitamos transformar essa variável qualitativa para quantitativa. Para isso, calculamos a frequência para cada intervalo:

```
1 freq_preco_justo = pd.DataFrame(df.preco_justo.value_counts())
2 freq_preco_justo
```

Python

preco_justo	
R\$15,00 - R\$25,00	108
R\$5,00 - R\$15,00	49
R\$25,00 - R\$35,00	45
Mais de R\$35,00	11

Então, a partir da mediana de cada intervalo (a não ser a categoria “Mais de R\$35,00”, que consideramos como 40 para seguir a incremento de 10 em cada valor médio), encontramos a frequência para cada um desses valores:

```
1 chaves = [20, 10, 30, 40] # valores médios para cada faixa
2 freq_preco_justo.index = chaves # substitui os índices da tabela
3 freq_preco_justo.reset_index(inplace=True) # cria uma coluna com os índices
4 freq_preco_justo.rename(columns={
5     "preco_justo": "freq",
6     "index": "preco_justo"
7 }, inplace=True) # renomeia as colunas
8 freq_preco_justo.sort_values("preco_justo", inplace=True) # ordena os valores de acordo com a
9 freq_preco_justo # mostra a tabela
```

[63]

Python

	preco_justo	freq
1	10	49
0	20	108
2	30	45
3	40	11

A partir da fórmula para calcular a média e desvio padrão amostral a partir de uma tabela de frequências:

```
1 freq_preco_justo["xf"] = freq_preco_justo.preco_justo * freq_preco_justo.freq
2 freq_preco_justo["x2f"] = freq_preco_justo.preco_justo**2 * freq_preco_justo.freq
3 freq_preco_justo
```

[42] ✓ 0.1s

Python

	preco_justo	freq	xf	x2f
1	10	49	490	4900
0	20	108	2160	43200
2	30	45	1350	40500
3	40	11	440	17600

```
1 x_barra = sum(freq_preco_justo.xf) / sum(freq_preco_justo.freq)
2 print("E(X) = %.3f" % x_barra)
```

[51] ✓ 0.3s

Python

... E(X) = 20.845

```

1 s = np.sqrt((sum(freq_preco_justo.x2f) - sum(freq_preco_justo.xf)**2 / sum(freq_preco_justo.freq))
2           / (sum(freq_preco_justo.freq) - 1))
3 print("s = %.3f" % s)

```

[52] ✓ 0.5s Python

... s = 8.024

Para estimar a média populacional, vamos utilizar o Caso 2 por desconhecemos o desvio padrão populacional. Considerando um intervalo de confiança de 90% e sabendo que o tamanho da nossa amostra é 213:

```

1 from scipy import stats
2 alfa = 0.9
3 sigma = 212 # n - 1
4 t = stats.t.ppf(1 - (1 - alfa) / 2, df=sigma)
5 ic_mi = (
6     np.round(x_barra - t * s / np.sqrt(sigma+1), 3),
7     np.round(x_barra + t * s / np.sqrt(sigma+1), 3)
8 )
9 print("IC da média populacional: [%.3f, %.3f]" % ic_mi)

```

[54] ✓ 0.4s Python

... IC da média populacional: [19.937, 21.753]

Estimativa da proporção populacional de usuários que assinam mais de um serviço (variável qualitativa)

Durante a nossa análise realizada no projeto semestral anterior, encontramos um fato muito interessante em nossa amostra – a grande maioria dos entrevistados assinam mais de 1 serviço de streaming. Com isso em mente, vamos estimar a proporção populacional de usuários que assinam mais de 1 serviço. Para isso, definimos como “0” usuários que assinam até 1 serviço, e “1” para usuários que assinam mais de 1.

```

1 df["n_servico_mais_de_1"] = np.where(df.n_servicos_assinados > 1, 1, 0)
2 df[["n_servicos_assinados", "n_servico_mais_de_1"]]

```

[46] ✓ 0.4s Python

...

	n_servicos_assinados	n_servico_mais_de_1
0	3	1
1	3	1
2	6	1
3	4	1
4	5	1
...
208	2	1
209	7	1
210	2	1
211	3	1
212	5	1

213 rows x 2 columns

```

1 p_barra = df.n_servico_mais_de_1.value_counts(normalize=True)[1]
2 print("p/ = %.3f" % p_barra)

```

[55] ✓ 0.5s Python

... p/ = 0.836

Sendo nossa proporção amostral igual à 0,836, calculamos o a proporção populacional com intervalo de confiança de 90%:

```
1 n = 213
2 alfa = 0.9
3 z = stats.norm.ppf(1 - (1 - alfa) / 2)
4 ic_p = (
5     np.round(p_barra - z * np.sqrt(p_barra * (1 - p_barra) / n), 3),
6     np.round(p_barra + z * np.sqrt(p_barra * (1 - p_barra) / n), 3)
7 )
8 print("IC da proporção populacional: [%.3f, %.3f]" % ic_p)
```

[56] ✓ 0.3s Python

... IC da proporção populacional: [0.794, 0.877]

Estimativas do tamanho da amostra com metade do erro

```
1 erro_mi = (ic_mi[1] - x_barra)/2
2 n_mi = (t * s / erro_mi)**2
3 print("Tamanho da amostra necessário a partir da média populacional = %.3f" % n_mi)
```

[63] ✓ 0.5s Python

... Tamanho da amostra necessário a partir da média populacional = 852.597

```
1 erro_p = (ic_p[1] - p_barra)/2
2 n_p = (z / erro_p)**2 * p_barra * (1 - p_barra)
3 print("Tamanho da amostra necessário a partir da proporção populacional = %.3f" % n_p)
```

[64] ✓ 0.3s Python Python

... Tamanho da amostra necessário a partir da proporção populacional = 870.439

Como os resultados possuem casas decimais, então o tamanho da amostra necessário para reduzir o erro pela metade a partir da média e da proporção populacionais são 853 e 871, respectivamente. Ou seja, seria necessário praticamente quadruplicar o tamanho da amostra para reduzir a margem de erro pela metade.