

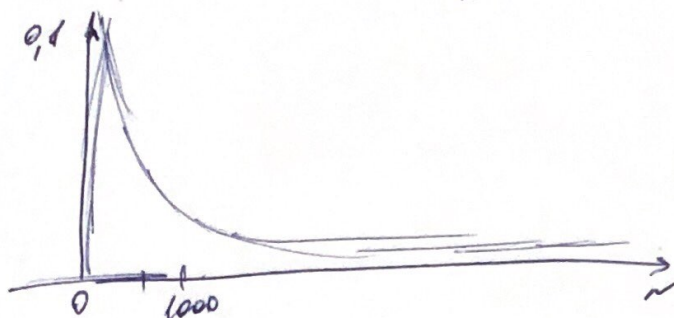
Задача 1

$$\begin{aligned} a) \quad P(X=2) &= \frac{1}{n} \\ P(X=3) &= \frac{n-1}{n} \cdot \frac{2}{n} \\ P(X=8) &= \frac{8-1}{n} \cdot \prod_{i=0}^{8-1} \frac{n-i}{n} \end{aligned}$$

Вариант 1. φ предположим

$$L = \frac{n-1}{n} \cdot \dots \cdot \frac{n-8}{n} \cdot \frac{2}{n} = \frac{2(n-1)(n-2) \dots (n-8)}{n^9}$$

на решение
задача является
такой, которая
была бы



$$\ln L = \ln 2 + \ln(n-1) + \dots + 9 \ln n \rightarrow$$

$$\frac{\partial \ln L}{\partial n} = \frac{1}{n-1} + \dots + \frac{9}{n} = 0$$

$$\Rightarrow n_m = 92$$

```

import numpy as np
import pandas as pd
import scipy.stats as sts
from scipy.stats import ttest_ind

#№4

#a)

df = pd.read_csv('22-23_hse_probability - Exam.csv')
df = df[['Last name', 'Unnamed: 72']]
df.dropna(inplace = True)

vowel = df[df['Last name'].str[0].isin(['А', 'Е', 'Ё', 'И', 'О', 'У',
    'Ы', 'Э', 'Ю', 'Я'])]['Unnamed: 72'].tolist()
consonant = df[~df['Last name'].str[0].isin(['А', 'Е', 'Ё', 'И', 'О',
    'У', 'Ы', 'Э', 'Ю', 'Я'])]['Unnamed: 72'].tolist()

t, p_value = ttest_ind (vowel, consonant, equal_var = False)

if p_value < 0.05:
    print("Отвергаем нулевую гипотезу")
else:
    print("Нет оснований отвергать нулевую гипотезу")

print ('p-value:', p_value)

Нет оснований отвергать нулевую гипотезу
p-value: 0.3974027153843839

<ipython-input-76-e3aded8e107d>:5: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation:
https://pandas.pydata.org/pandas-docs/stable/user\_guide/indexing.html#returning-a-view-versus-a-copy
    df.dropna(inplace = True)

#б)

obs_diff = np.mean(vowel) - np.mean(consonant)

def boots_samples(x):
    return np.random.choice(x, size = len(x))

num_bootstrap_samples = 10000
bootstrap_differences = np.zeros(num_bootstrap_samples)
for i in range(num_bootstrap_samples):
    vowel_sample = boots_samples(vowel)

```

```
consonant_sample = boots_samples(consonant)
bootstrap_differences[i] = vowel_sample.mean() -
consonant_sample.mean()
```

```
p_value = (np.abs(bootstrap_differences) > np.abs(obs_diff)).mean()
```

```
if p_value < 0.05:
    print("Отвергаем нулевую гипотезу")
else:
    print("Нет оснований отвергать нулевую гипотезу")
```

```
print ('p-value:', p_value)
```

```
Нет оснований отвергать нулевую гипотезу
p-value: 0.5492
```

```
#B)
```

```
obs_diff = np.mean(vowel) - np.mean(consonant)
```

```
def bootstrap(sample, n):
    return np.random.choice(sample, size=(n, len(sample)),
replace=True)
```

```
n_iterations = 10000
diffs = []
for i in range(n_iterations):
    vowel_sample = bootstrap(vowel, 100)
    consonant_sample = bootstrap(consonant, 100)
    diff = vowel_sample.mean() - consonant_sample.mean()
    diffs.append(diff)
```

```
p_value = (np.abs(diffs) >= np.abs(obs_diff)).mean()
```

```
if p_value < 0.05:
    print("Отвергаем нулевую гипотезу")
else:
    print("Нет оснований отвергать нулевую гипотезу")
```

```
print('p-value:', p_value)
```

```
Нет оснований отвергать нулевую гипотезу
p-value: 0.5071
```

```
#r)
```

```
obs_diff = np.mean(vowel) - np.mean(consonant)
```

```
# Merge scores and shuffle them randomly to create bootstrap samples
combined_scores = np.concatenate((vowel, consonant))
```

```

n_samples = 10000
boot_diffs = np.zeros(n_samples)
for i in range(n_samples):
    np.random.shuffle(combined_scores)
    boot_vowel_scores = combined_scores[:len(vowel)]
    boot_consonant_scores = combined_scores[len(vowel):]
    boot_diffs[i] = np.mean(boot_vowel_scores) -
np.mean(boot_consonant_scores)

p_value = np.sum (boot_diffs >= obs_diff) / n_samples

if p_value < 0.05:
    print('Отвергаем нулевую гипотезу')
else:
    print('Нет оснований отвергать нулевую гипотезу')

print('p-value bootstrap:', p_value)

Нет оснований отвергать нулевую гипотезу
p-value bootstrap: 0.8161

#№5

#a)
vowel_more = len([i for i in vowel if i > df['Unnamed: 72'].median()])
vowel_less = len([i for i in vowel if i <= df['Unnamed:
72'].median()])
consonant_more = len([i for i in consonant if i > df['Unnamed:
72'].median()])
consonant_less = len([i for i in consonant if i <= df['Unnamed:
72'].median()])

matrix = np.array([[vowel_more, vowel_less], [consonant_more,
consonant_less]])

print (matrix)

odds_vowel = vowel_more / vowel_less
odds_consonant = consonant_more / consonant_less

ln_shans = np.log(odds_vowel / odds_consonant)
se_ln_shans = np.sqrt(1/vowel_more + 1/vowel_less + 1/consonant_more +
1/consonant_less)
alpha = 0.05
z_crit = sts.norm.ppf(1 - alpha/2)

z_obs_5_a = ln_shans / se_ln_shans

q_l = np.exp(ln_shans - z_crit*se_ln_shans)

```

```

q_r = np.exp(ln_shans + z_crit*se_ln_shans)
p_value = 2*(sts.norm.cdf(z_obs_5_a))

if p_value < 0.05:
    print('Отвергаем нулевую гипотезу')
else:
    print('Нет оснований отвергать нулевую гипотезу')

print('p-value:', p_value)
print('q_l:', q_l)
print('q_r:', q_r)

[[ 21  28]
 [145 138]]
Нет оснований отвергать нулевую гипотезу
p-value: 0.280180274566451
q_l: 0.3870945958254781
q_r: 1.3162172761513564

#6)
vowel1 = matrix[0].sum()
consonant1 = matrix[1].sum()

p_glas = vowel_more / vowel1
p_consonant = consonant_more / consonant1

ln_p = np.log(p_glas / p_consonant)
se_ln_p = np.sqrt(1/(vowel_more + vowel_less) + 1/(consonant_more +
consonant_less))

z_crit = sts.norm.ppf(1 - alpha/2)

z_obs_5_b = ln_p / se_ln_p

q_l = np.exp(ln_p - z_crit*se_ln_p)
q_r = np.exp(ln_p + z_crit*se_ln_p)
p_value = 2*(sts.norm.cdf(z_obs_5_b))

print('p-value:', p_value)
print('q_l:', q_l)
print('q_r:', q_r)

p-value: 0.24843407914401505
q_l: 0.6176383370749431
q_r: 1.1327890726474006

#№6

#a)

```

```
df['длина'] = df['Last name'].apply(len)
beta = df['Unnamed: 72'].mean()/df['длина'].mean()
corr = df['Unnamed: 72'].corr(df['длина'])
print('Оценка  $\beta$ :', beta)
print('Выборочная корреляция:', corr)
```

Оценка β : 2.0613026819923372
 Выборочная корреляция: 0.025328052669147665

#6)

```
obs_corr = df['Unnamed: 72'].corr(df['длина'])
n_samples = 10000
p_corrs = []

for i in range(n_samples):
    df['длина'] = np.random.permutation(df['длина'])
    p_corr = df['Unnamed: 72'].corr(df['длина'])
    p_corrs.append(p_corr)
p_value = (np.abs(p_corrs) >= np.abs(obs_corr)).mean()
print('p-value:', p_value)
if p_value < 0.05:
    print('Отвергаем нулевую гипотезу')
else:
    print('Нет оснований отвергать нулевую гипотезу')
#поэтому корреляция равна нулю
```

p-value: 0.6479
 Нет оснований отвергать нулевую гипотезу

#№7

Решаем задачу номер 2 из контрольной 4 прошлых лет 2017-2018 с помощью чата GPT.

[]()

Чат GPT прав, однако вывод можно было сделать чуть проще: зная, что при уровне значимости 0,05 z критическое будет равно 1,96, а полученное 1,2 по модулю меньше критического, следовательно, нет оснований отвергать нулевую гипотезу.

#№8

https://www.youtube.com/watch?v=rGM_arqlFQA

Это видео я посмотрела в начале курса и мне оно очень понравилось! Понятным языком объясняют некоторые моменты и что такое в целом теория вероятностей. Если бы у меня спросили, что такое теория вероятностей и зачем её изучать, я бы отправила этому человеку данное видео)