

## Выполнила: Бородулина Алена

```
In [ ]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import random
```

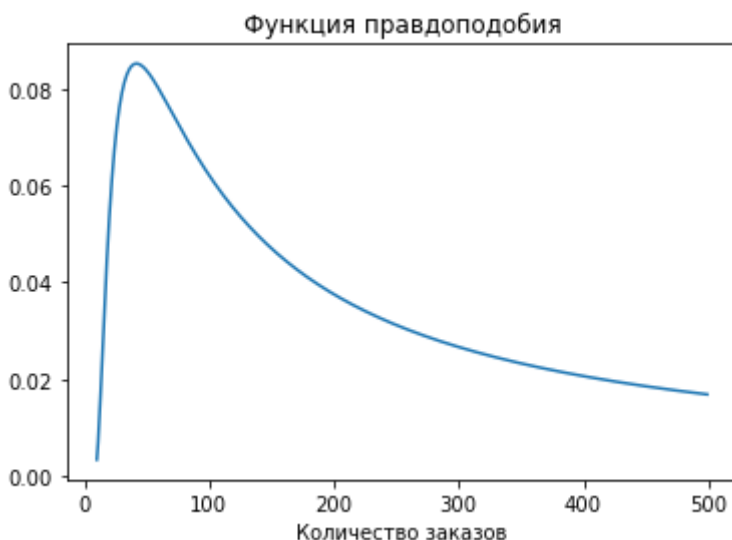
### №1

a)

$$L = \frac{n/n * (n-1)/n * (n-2)/n * \dots * (n-8)/n * 9/n}{n^{10}} =$$

```
In [92]: p_l = []
for n in range(10, 500):
    p = (n*(n-1)*(n-2)*(n-3)*(n-4)*(n-5)*(n-6)*(n-7)*(n-8)*9)/(n**10)
    p_l.append(p)
```

```
In [97]: plt.plot(range(10, 500), p_l)
plt.xlabel('Количество заказов')
plt.title('Функция правдоподобия')
plt.show()
```



$$l = \ln(n) + \ln(n-1) + \dots + \ln(n-8) + \ln(9) - 10\ln(n)$$

$$l' = \frac{1}{n} + \dots + \frac{1}{n-8} - \frac{10}{n}$$

$$n = 10n(n-1)(n-2)\dots(n-8)$$

$$n = 42$$

$$l'' < 0$$

Учитывая, что водителей хотя бы 9, получим, что максимум достигается при 42.

б) Начальный выборочный момент 1 порядка:  $\sum_{i=1}^{\infty} i \frac{(i-1)n(n-1)(n-2)\dots(n-i+2)}{n^i} = \overline{X}$ , где справа стоит выборочный начальный момент 1 порядка (средняя),  $i$  - номер заказа, на котором происходит повтор. Выразив  $n$  через  $X$  средний получим оценку  $n$  методом моментов.

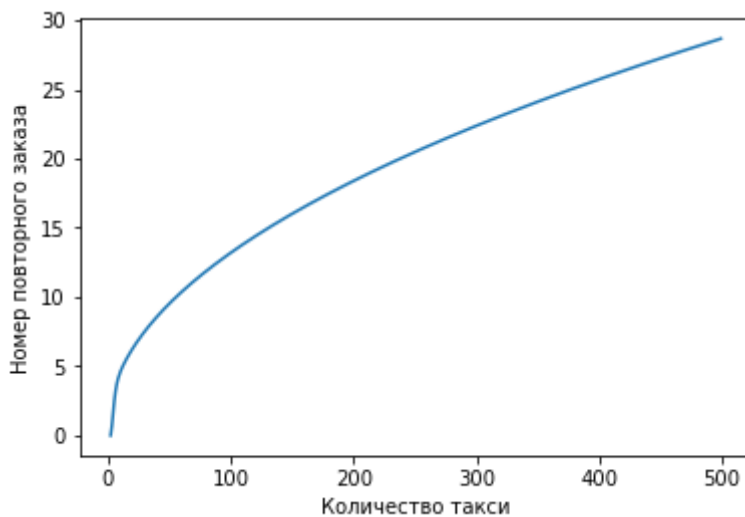
In [5]:

```
import math
x_exp = []
for n in range(2, 500):
    x_pr = []
    for i in range(2, n):
        x = math.prod(range(n - i + 2, n + 1))/n**i
        x_mn = x*i*(i-1)
        x_pr.append(x_mn)
    x = sum(x_pr)
    x_exp.append(x)
```

График математического ожидания номера заказа, на котором происходит первый повторный проезд.

In [116...]

```
plt.plot(range(2,500), x_exp)
plt.xlabel('Количество такси')
plt.ylabel('Номер повторного заказа')
plt.show()
```



в)

In [54]:

```
# Сделаем выборку номеров повторных заказов
n = 100
been = []
zak = []

for i in range(10**4):
    num = 0
    for i in range(100):
        choice = random.choice(range(100))
        if choice in been:
            num += 1
            zak.append(num)
            break
        else:
            num += 1
            been.append(choice)
    been = []
```

Метод максимального правдоподобия:  $\prod_{i=1}^{\infty} \frac{(i-1)n(n-1)(n-2)\dots(n-i+2)}{n^i} \rightarrow \max_n$

Ищу значения  $n$  "в лоб": по каждому заказу прогоняю функцию правдоподобия и ищу значение  $n$ , при котором функция достигает максимума.

In [190...

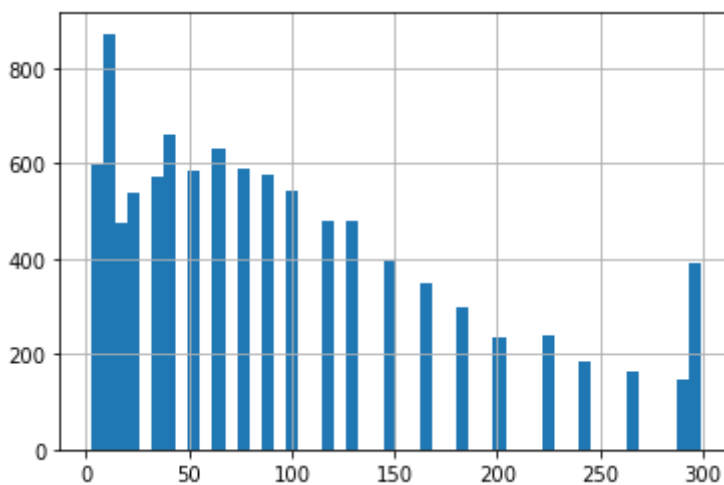
```
mmax = []
# Не повторять в домашних условиях, очень долго
for i in zak:
    df = pd.DataFrame({'mm':[0], 'n': [0]})
    for n in range(2, 300):
        x = math.prod(range(n - i + 2, n + 1))/n**i
        x = x*(i-1)
        df = df.append({'mm':x, 'n': n}, ignore_index=True)
    mmax.append(df[df['mm'] == max(df['mm'])]['n'].values[0])
```

In [213...

```
df.mmax.hist(bins = 50) # Гистограмма распределения
```

Out[213...

&lt;AxesSubplot:&gt;



In [210...

```
var = sum((df.mmax - 100)**2)/len(df.zak)
std = np.sqrt(var)
sm = sum((df.mmax - df.zak)**2)/len(df.zak) # Находим дисперсию реальных значений за

print(f'Стандартное отклонение метода максимального правдоподобия: {std}', "\n",
      f'Дисперсия: {var}', "\n",
      f'Смещение: {sm}')
```

Стандартное отклонение метода максимального правдоподобия: 81.72358778223091

Дисперсия: 6678.7448

Смещение: 12097.3062

## №2

a)

$$L = 210 \frac{n}{n} * \frac{n-1}{n} * \frac{n-2}{n} * \frac{n-3}{n} * \frac{n-4}{n} * \frac{n-5}{n} * \frac{4}{n}$$

$$l'n = \frac{1}{n} + \dots + \frac{1}{n-5} - \frac{10}{n}$$

$$n = 10n(n-1)(n-2) \dots (n-5)$$

$$n = 8$$

$$l'' < 0$$

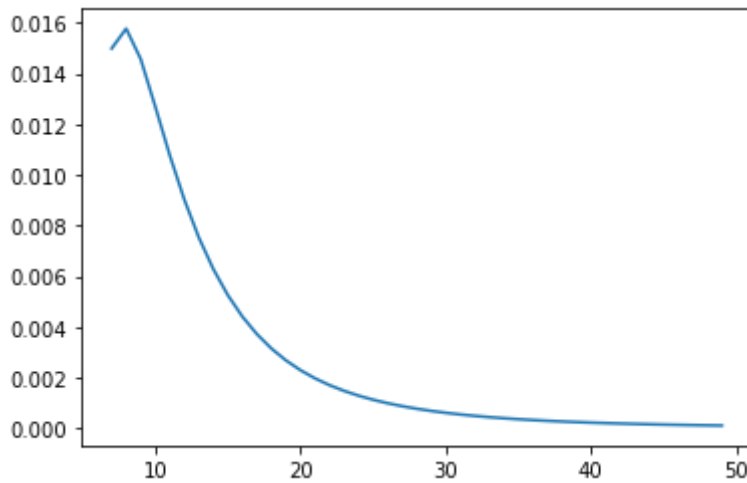
Учитывая, что водителей хотя бы 6, получим, что максимум достигается при 8

In [66]:

```

p_l = []
for n in range(7,50):
    p = (210*n*(n-1)*(n-2)*(n-3)*(n-4)*(n-5)*4)/(n**10)
    p_l.append(p)
plt.plot(range(7,50), p_l)
plt.show()

```



## №4

### а) Тест Уэлча

In [125...]

```

df1 = pd.read_csv('Downloads/22-23_hse_probability - Exam.csv', sep=',')
df2 = pd.DataFrame()
df2['Fam'] = df1['Last name']
df2['Exam'] = df1['Unnamed: 72']

df2 = df2.dropna()

letter = []
gl = ['a', 'e', 'ё', 'и', 'o', 'y', 'ы', 'э', 'ю', 'я']
for fam in df2['Fam']:
    if list(fam)[0].lower() in gl:
        letter.append(0)
    else:
        letter.append(1)
df2['Glasn'] = letter
df2['Median'] = np.where(df2['Exam'] > df2['Exam'].median(), 1, 0)
df2.head()

```

Out[125...]

	Fam	Exam	Glasn	Median
5	Репенкова	16.0	1	0
6	Ролдугина	0.0	1	0
7	Сафина	19.0	1	1
8	Сидоров	26.0	1	1
9	Солоухин	21.0	1	1

In [126...]

```

ex_g1 = df2[df2['Glasn'] == 1]['Exam']
ex_sog1 = df2[df2['Glasn'] == 0]['Exam']

```

```
from scipy import stats

# Рассчитаем t-статистику и p-value
st = stats.ttest_ind(ex_g1, ex_sog1, equal_var = False)

print("\n",
      f"Welch's t-test = {st[0]}", "\n",
      f"p-value = {st[1]}", "\n")
```

Welch's t-test = 0.8519661870595602  
p-value = 0.3974027153843839

p-value > 0.05, значит, гипотеза не отвергается.

## б) Наивный бутстрэп

In [127...

```
def get_bootstrap_sample(x, B_sample=1):
    np.random.seed(122)
    N = x.size
    sample = np.random.choice(x, size=(N, B_sample), replace=True)
    return sample

# Генерируем выборки и находим их медианы
x_boot = get_bootstrap_sample(ex_g1, B_sample=10**4)
x_med = np.median(x_boot, axis=0)

y_boot = get_bootstrap_sample(ex_sog1, B_sample=10**4)
y_med = np.median(y_boot, axis=0)

# Рассчитаем статистику и p-value
st = stats.ttest_ind(x_med, y_med, equal_var = False)

print("\n",
      f"Welch's t-test = {st[0]}", "\n",
      f"p-value = {st[1]}", "\n")
```

Welch's t-test = 78.57130869265276  
p-value = 0.0

p-value < 0.05, значит, гипотеза отвергается

## в) Бутстрэп t-статистики

In [128...

```
t_st = stats.ttest_ind(x_boot, y_boot, equal_var = False) [0]

left_t = pd.Series(t_st).quantile(0.025)
right_t = pd.Series(t_st).quantile(0.975)
std = np.sqrt(np.var(ex_g1)/ex_g1.size + np.var(ex_sog1)/ex_sog1.size)

left = ex_g1.mean() - ex_sog1.mean() - left_t * std
right = ex_g1.mean() - ex_sog1.mean() - right_t * std
print( f'[{left}, {right}]')
```

[2.5164903595667236, -2.4094876799881653]

0 входит в промежуток, значит, гипотеза не отвергается.

## №5

a)

```
In [129... df2['Median'] = np.where(df2['Exam'] > df2['Exam'].median(), 1, 0)
```

```
In [130... pd.crosstab(df2['Median'], df2['Glasn'])
# 1 - согласные, 0 - гласные
```

```
Out[130... Glasn  0    1
Median
0      28   138
1      21   145
```

```
In [131... OR = (145*28)/(138*21)
left = np.exp(np.log(OR) - 1.96 * (1/138 + 1/28 + 1/145 + 1/21))
right = np.exp( np.log(OR) + 1.96 * (1/138 + 1/28 + 1/145 + 1/21))
print( f'Доверительный интервал: [{left}, {right}]')
```

Доверительный интервал: [1.1573219317139123, 1.6959034420212988]

Предполагаем, что шансы относятся как 1/1.

```
In [132... z = np.log(OR)/(1/138 + 1/28 + 1/145 + 1/21)
p_v = stats.norm.sf(abs(z))/2

print(f'p-value = {p_v}')
```

p-value = 0.0001355888774524068

p-value < 0.05, значит, гипотеза отвергается.

б)

```
In [133... p_1 = 21/(21+28)
p_2 = 145/(145+138)
left = (p_2 - p_1) - 1.96* np.sqrt(p_2*(1-p_2)/(145+138)+p_1*(1-p_1)/(21+28))
right = (p_2 - p_1) + 1.96* np.sqrt(p_2*(1-p_2)/(145+138)+p_1*(1-p_1)/(21+28))
print( f'Доверительный интервал: [{left}, {right}]')
```

Доверительный интервал: [-0.06650883846186637, 0.2341009636511647]

Предполагаем, что вероятность относится как 1/1.

```
In [134... p = (21+145)/(21+145+138+28)
z = (p_2 - p_1)/np.sqrt((p*(1-p)*(1/21+1/145)))
p_v = stats.norm.sf(abs(z))/2

print(f'p-value = {p_v}')
```

p-value = 0.11822276293514869

p-value > 0.05, значит, гипотеза не отвергается

№ 6

а) Для оценки  $\beta$  методом моментов необходимо приравнять выборочный момент первого порядка к теоретическому моменту первого порядка:

$$E(Y) = \beta E(F)$$

Выборочный момент первого порядка равен среднему значению результатов экзаменов:

$$(1/n) * \sum_{i=1}^{\infty} Y_i = \beta * (1/n) * \sum_{i=1}^{\infty} F_i$$

Отсюда:

$$\beta = (1/n) * (\sum_{i=1}^{\infty} Y_i / \sum_{i=1}^{\infty} F_i)$$

Выборочная корреляция между результатами экзаменов и длиной фамилий:

$$r = \text{cov}(Y, F) / (\sigma_Y * \sigma_F)$$

```
In [38]: leng = []
for i in df2.Fam:
    i = len(i)
    leng.append(i)
df2['len'] = leng

b = (1/len(df2.len)) * (sum(df2.Exam)/sum(df2.len))
print(f'Оценка  $\beta$  = {b}')
```

Оценка  $\beta$  = 0.006208743018049209

```
In [48]: r = df2['Exam'].cov(df2['len'])/(np.sqrt(df2['Exam'].var()*np.sqrt(df2['len'].var()))
print(f'Выборочная корреляция между результатами экзаменов и длиной фамилий: {r}')
```

Выборочная корреляция между результатами экзаменов и длиной фамилий: 0.02532805266914765

```
In [89]: import copy
[r,pv] = stats.pearsonr(df2['Exam'],df2['len'])

pS = copy.copy(df2.len)
pR = []
p=10000

for i in range(0,p):
    random.shuffle(pS)
    pR.append(stats.pearsonr(pS,df2['Exam'])[0])

p_val = len(np.where(np.abs(pR)>=np.abs(r))[0])/p
print (p_val)
```


0.6524

Значит, гипотеза о корреляции, равной 0, не отвергается

## № 7

```
In [114... from IPython.display import Image
Image(url = 'https://sun9-4.userapi.com/impf/D_D92Ebrjx6cZwnUkyEzCLNIw9D43EU7Wiz_cQ/
```

Out [114...



Для построения 95% асимптотического интервала для отношения вероятностей "хорошо написать экзамен" необходимо знать количество студентов, попавших в каждую из категорий: тех, кто набрал больше медианы и тех, кто набрал меньше или равно медиане, а также тех, чьи фамилии начинаются на согласную и на гласную букву.

Пусть  $n_1$  будет количеством студентов, попавших в первую категорию (больше медианы),  $n_2$  - количество студентов во второй категории (меньше или равно медиане),  $n_3$  - количество студентов, у которых фамилии начинаются на согласную, и  $n_4$  - количество студентов с фамилиями, начинающимися на гласную.

Для построения интервала, можно использовать аппроксимацию нормальным распределением для больших выборок. Формула для расчета интервала имеет следующий вид:


$$\text{Интервал} = p_1 - p_2 \pm z \cdot \sqrt{(p_1 \cdot (1 - p_1) / n_1) + (p_2 \cdot (1 - p_2) / n_2)}$$

где  $p_1 = n_1 / (n_1 + n_3)$  и  $p_2 = n_2 / (n_2 + n_4)$  - оценки вероятностей успеха в каждой из категорий.

Для проверки гипотезы о равенстве отношения вероятностей 1, можно воспользоваться статистикой z-теста, где нулевая гипотеза предполагает, что отношение вероятностей равно 1.

Статистика z-теста рассчитывается по следующей формуле:

[Regenerate response](#)

Send a message. 

Free Research Preview. ChatGPT may produce inaccurate information about people, places, or facts. [ChatGPT May 24 Version](#)

In [115...

```
Image(url = 'https://sun9-77.userapi.com/impf/3LYTiDAwWbjuRy24H-FCsdTuvk_Bmk_MI3tUkg
```



Out[115...

Интервал =  $p_1 - p_2 \pm z \cdot \sqrt{(p_1 \cdot (1 - p_1) / n_1) + (p_2 \cdot (1 - p_2) / n_2)}$

где  $p_1 = n_1 / (n_1 + n_3)$  и  $p_2 = n_2 / (n_2 + n_4)$  - оценки вероятностей успеха в каждой из категорий.

Для проверки гипотезы о равенстве отношения вероятностей 1, можно воспользоваться статистикой z-теста, где нулевая гипотеза предполагает, что отношение вероятностей равно 1.

Статистика z-теста рассчитывается по следующей формуле:

$z = (p_1 - p_2) / \sqrt{(\hat{p} \cdot (1 - \hat{p}) \cdot (1/n_1 + 1/n_2))}$

где  $\hat{p} = (n_1 + n_2) / (n_1 + n_2 + n_3 + n_4)$  - оценка общей вероятности успеха.

P-значение можно определить, используя стандартное нормальное распределение и сравнивая статистику z со стандартной нормальной таблицей.

Предоставленные данные о таблице сопряженности студентов недостаточны для расчета интервала и проведения теста. Необходимо знать конкретные значения  $n_1$ ,  $n_2$ ,  $n_3$  и  $n_4$  для выполнения расчетов.

🔄 Regenerate response

Send a message.



Free Research Preview. ChatGPT may produce inaccurate information about people, places, or facts. [ChatGPT May 24 Version](#)

После этого я подставила известные мне  $n_1$ ,  $n_2$ ,  $n_3$ ,  $n_4$  и получила результат.

## № 8

С помощью этого видео я хотя бы немного понял, что такое бутстреп:

[https://www.youtube.com/watch?v=wIPq\\_OoYcjc](https://www.youtube.com/watch?v=wIPq_OoYcjc)