Домашнаяя работа

Выполнили студенты группы *БПИ201*:

- **Клоков Станислав** (номер *16* по списку)
- **Попов Матвей** (номер *29* по списку)
- **Прокудин Максим** (номер *31* по списку)

Вариант *31* (Томск: Томск)

Целевая выборка

Томск: Томск

```
import pandas as pd
import matplotlib as mpl
import seaborn as sns
import numpy as np
import matplotlib.pyplot as plt

sns.set(palette='Set2')


table = pd.read_excel("data.xlsx")

location = "Томск: Томск"
table = table[table['psu'] == location]

display(table)
```

	idind	psu	status	age	male	industry	wage	public	internet
180	4202	Томск: Томск	Областной це	46	1	ТРАНСПОРТ, С	25000	0	0
181	4229	Томск: Томск	Областной це	38	0	СТРОИТЕЛЬСТВ	9000	1	1
182	4246	Томск: Томск	Областной це	55	0	ЮРИСПРУДЕНЦИ	90000	1	1

Номера 1 - 11

Задание 1

Рассчитайте описательные статистики (минимум, максимум, среднее значение, стандартное отклонение, размах) для всех переменных в Вашей выборке кроме отрасли и номера региона.

```
desc = table.describe()

# removed extra
del desc['idind']
for id_ in range(1, 76):
    del desc['id' + str(id_)]


def map_var_to_description(raw):
    need = ['min', 'max', 'mean', 'std']
    for key in raw.keys():
        if key not in need:
            del raw[key]

    raw['range'] = raw['max'] - raw['min']
    return raw.to_string()


for [key, value] in desc.items():
    print(f'[--- {key} ---]', end='\n\n')
    print(map_var_to_description(value.copy()), end='\n\n\n')
```

```
[--- age ---]

mean     42.835294
std      11.871181
min      18.000000
max      60.000000
range    42.000000


[--- male ---]
```

```
mean     0.447059
std      0.500140
min      0.000000
max      1.000000
range    1.000000


[--- wage ---]

mean     27481.176471
std      16093.635936
min       3000.000000
max      90000.000000
range    87000.000000


[--- public ---]

mean     0.423529
std      0.497050
min      0.000000
max      1.000000
range    1.000000


[--- internet ---]

mean     0.847059
std      0.362067
min      0.000000
max      1.000000
range    1.000000


[--- children ---]

mean     1.294118
std      0.985895
min      0.000000
max      6.000000
range    6.000000


[--- urban ---]

mean     1.0
std      0.0
```

Задание 2

Оцените квартили (25%, 50%, 75%) распределения для непрерывных переменных в выборке. Определите межквартильный размах.

```
desc = table.describe()
```

```
def map_var_to_quartile(raw):
    need = ['25%', '50%', '75%']
    for key in dict(raw).keys():
        if key not in need:
            del raw[key]

    raw['qurtile_range'] = raw['75%'] - raw['25%']
    return raw.to_string()

continuous = ['ln_wage', 'wage', 'age']

for key in continuous:
    print(f'[--- {key} ---]', end='\n\n')
    print(map_var_to_quartile(desc[key].copy()), end='\n\n\n')
```

```
[--- ln_wage ---]

25%              9.11758
50%              9.12751
75%              9.13389
qurtile_range    0.01631


[--- wage ---]

25%              15000.0
50%              25000.0
75%              33000.0
qurtile_range    18000.0


[--- age ---]

25%              31.0
50%              46.0
75%              53.0
qurtile_range    22.0
```

Задание 3

Сравните среднее, медиану и моду для непрерывных переменных в выборке. Что можно сказать об их соотношении?

```
desc = table.describe()


def find_mod(arr):
    arr = list(arr)
    counter = list()
    was = set()
    for item in arr:
        if item in was:
```

```
            continue
        was.add(item)
        counter.append((arr.count(item), item))
    counter.sort(reverse=True)
    return counter[0][1]


for key in continuous:
    print(f'[--- {key} ---]', end='\n\n')
    print('mean =', desc[key]['mean'])
    print('median =', table[key].median())
    print('mod =', find_mod(table[key].values))
    print(end='\n\n\n')
```

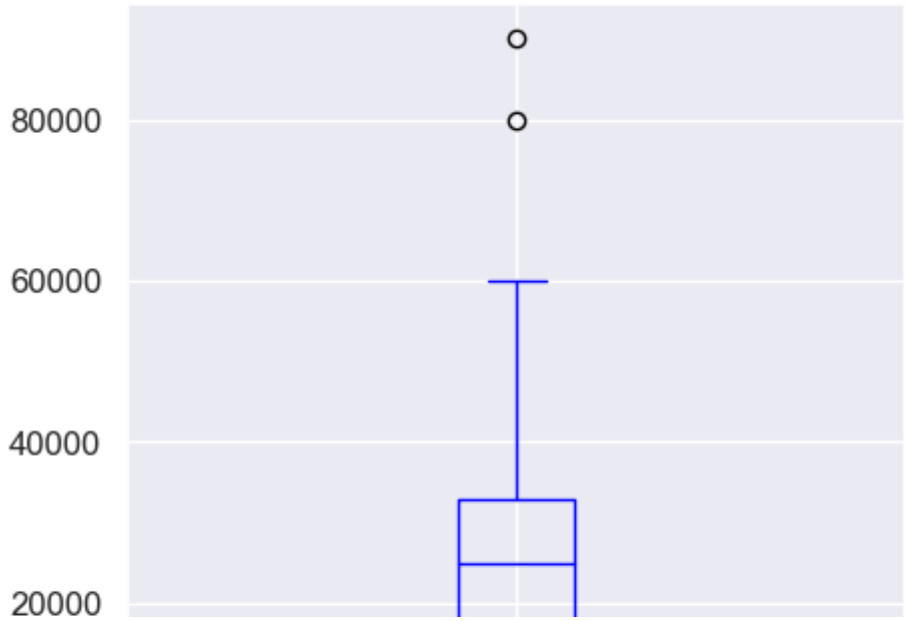
```
[--- ln_wage ---]

mean = 9.125011058823528
median = 9.12751
mod = 9.14241



[--- wage ---]

mean = 27481.176470588234
median = 25000.0
mod = 30000



[--- age ---]

mean = 42.83529411764706
median = 46.0
mod = 57
```

Задание 4

Постройте box-plot для всех непрерывных переменных. Есть ли выбросы?

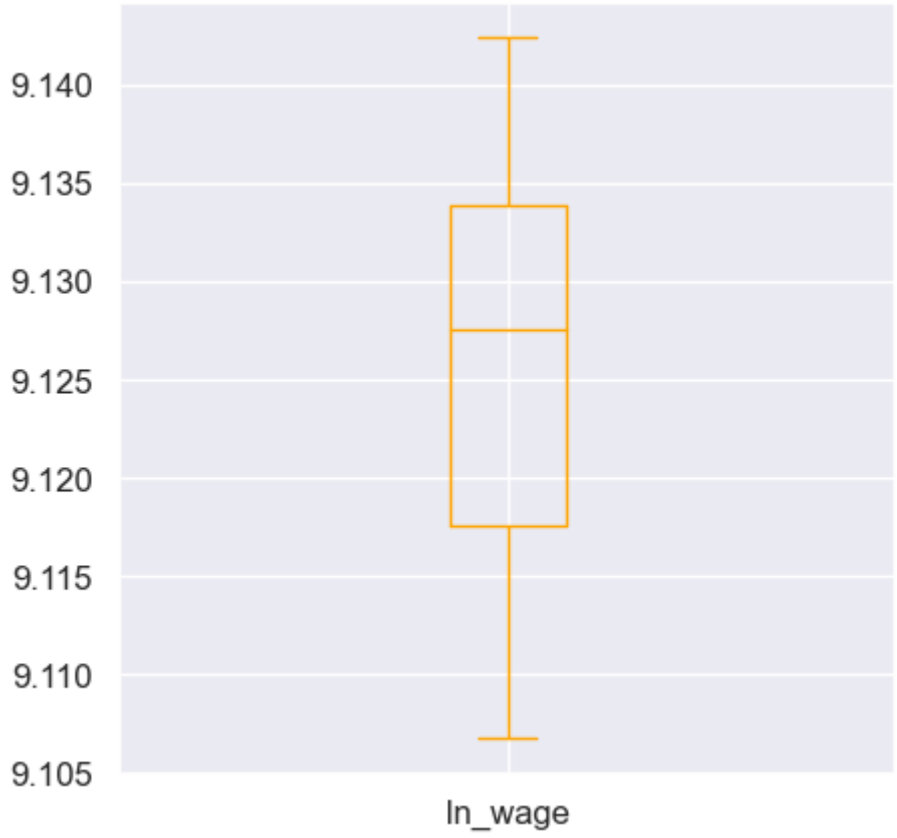
```
table.boxplot(column='wage', fontsize=12, figsize=(5, 5), color='blue')

# выбросы есть в районе 80000
```

```
<AxesSubplot:>
```



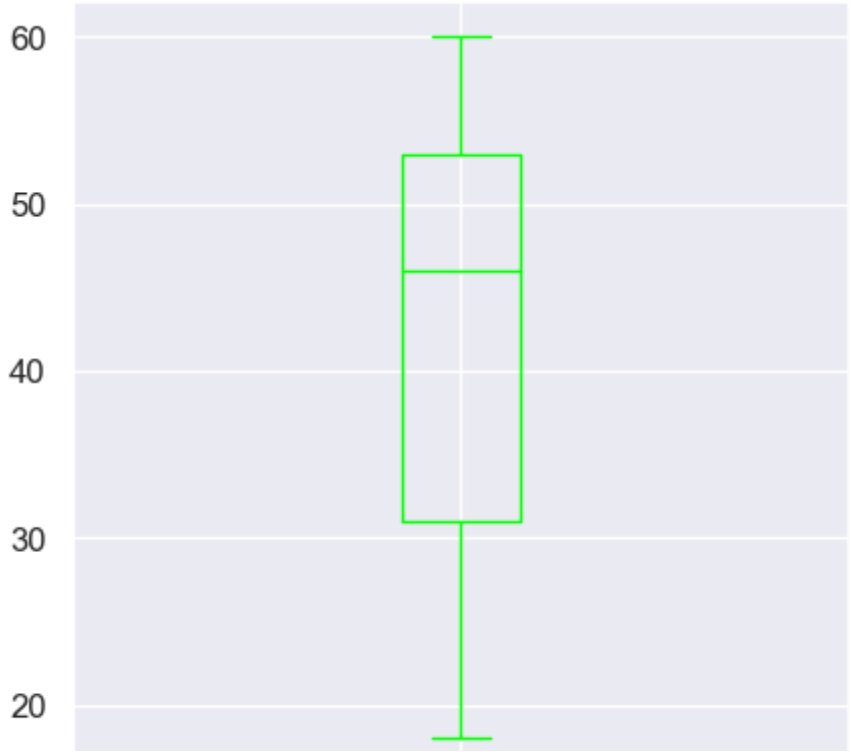

```
table.boxplot(column='ln_wage', fontsize=12, figsize=(5, 5), color='orange')

# выбросов нет
```

```
<AxesSubplot:>
```




```
table.boxplot(column='age', fontsize=12, figsize=(5, 5), color='lime')

# выбросов нет
```

```
<AxesSubplot:>
```



Задание 5

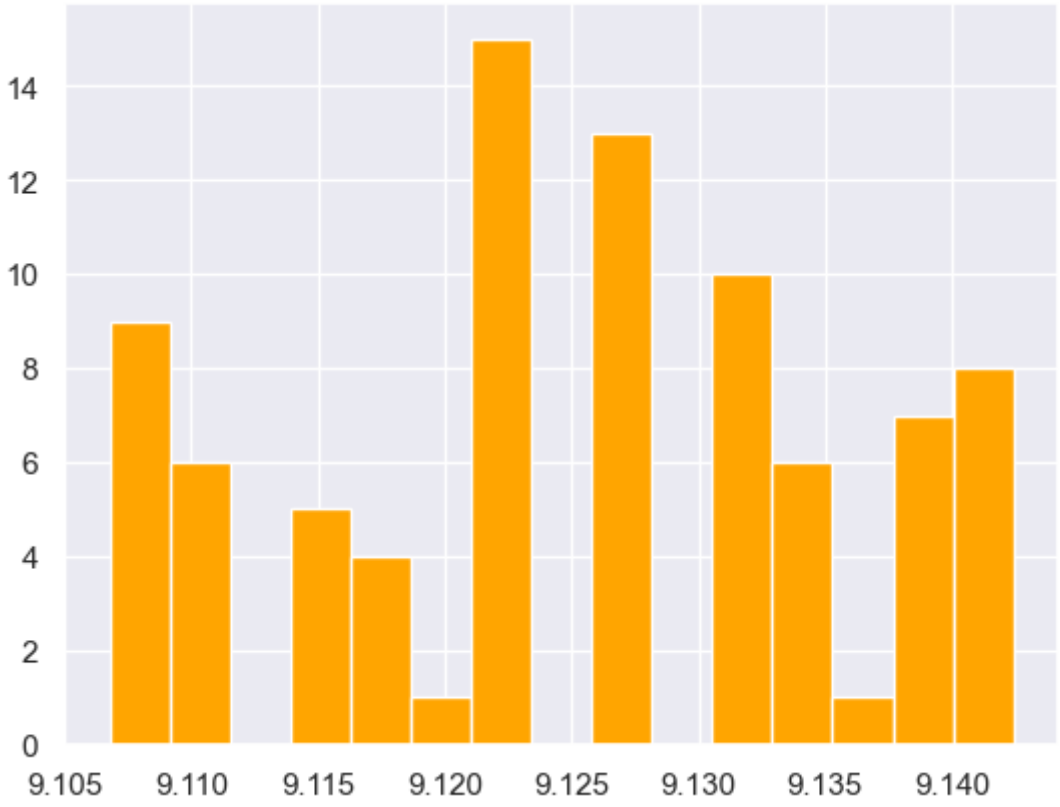
Постройте гистограммы распределения для непрерывных переменных в выборке. Что можно сказать о скошенности (асимметрии) и островершинности их распределений?

```
# распределение слабо скошенно без учета выбросов и островершинно

display(mpl.pyplot.hist(table['wage'], color='blue', bins=15))
```

```
(array([ 5., 16., 12., 11., 16.,  6.,  7.,  3.,  4.,  3.,  0.,  0.,  0.,
         1.,  1.]),
 array([ 3000.,  8800., 14600., 20400., 26200., 32000., 37800., 43600.,
        49400., 55200., 61000., 66800., 72600., 78400., 84200., 90000.]),
 <BarContainer object of 15 artists>)
```

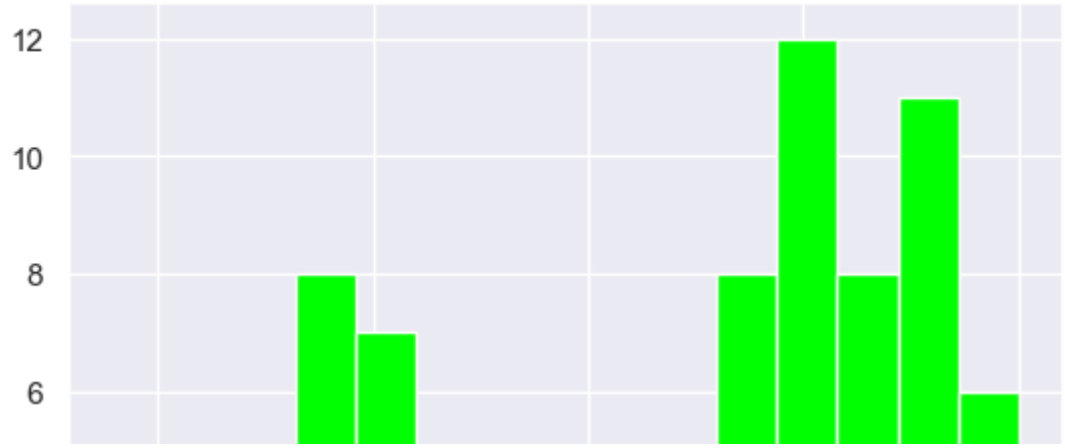
```
# распределение не скошено на отрезке от моды до максимума без учета выбросов

display(mpl.pyplot.hist(table['ln_wage'], color='orange', bins=15))
```

```
(array([ 9.,  6.,  0.,  5.,  4.,  1., 15.,  0., 13.,  0., 10.,  6.,  1.,
         7.,  8.]),
 array([9.10678   , 9.10915533, 9.11153067, 9.113906  , 9.11628133,
        9.11865667, 9.121032  , 9.12340733, 9.12578267, 9.128158  ,
        9.13053333, 9.13290867, 9.135284  , 9.13765933, 9.14003467,
        9.14241   ]),
 <BarContainer object of 15 artists>)
```




```
# распределение смещено в сторону максимлаьного значения

display(mpl.pyplot.hist(table['age'], color='lime', bins=15))
```

```
(array([ 1.,  2.,  5.,  8.,  7.,  5.,  2.,  5.,  3.,  2.,  8., 12.,  8.,
        11.,  6.]),
 array([18. , 20.8, 23.6, 26.4, 29.2, 32. , 34.8, 37.6, 40.4, 43.2, 46. ,
        48.8, 51.6, 54.4, 57.2, 60. ]),
 <BarContainer object of 15 artists>)
```

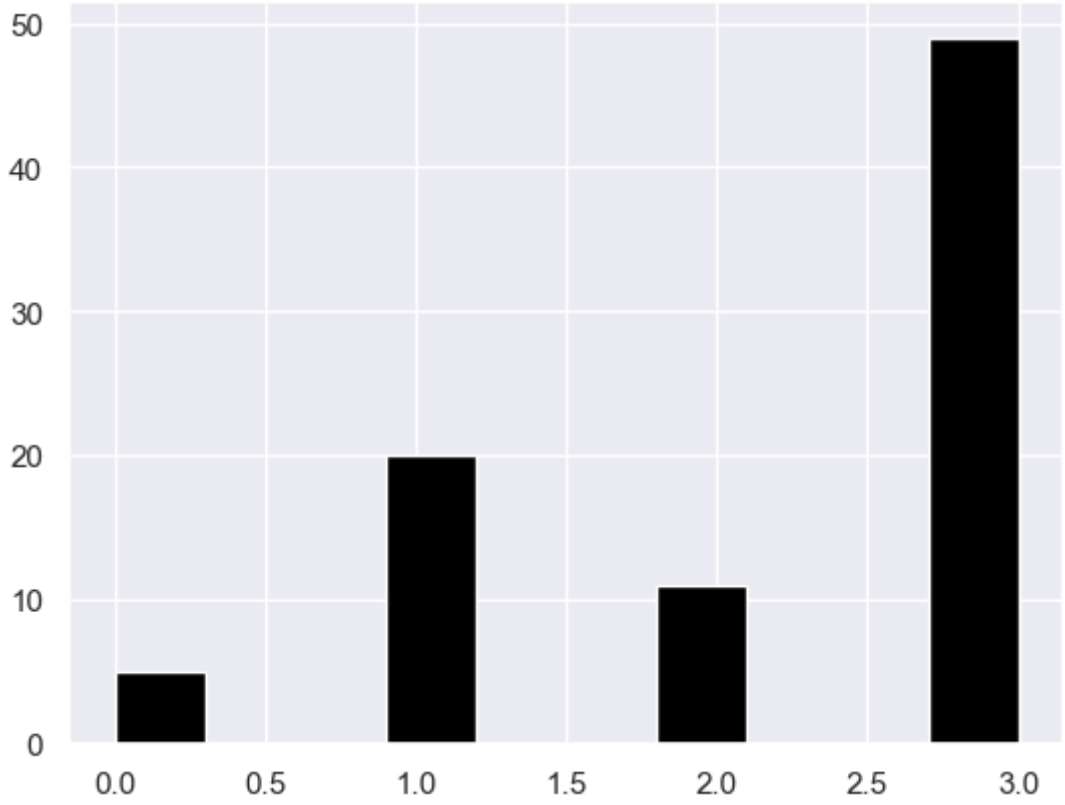


Задание 6

Как распределены респонденты в Вашей выборке по уровню образования? Постройте гистограмму.

```
# распределение смещено в сторону максимлаьного значения

display(mpl.pyplot.hist(table['educ'], histtype='stepfilled', color='black'))
```

```
(array([ 5.,  0.,  0., 20.,  0.,  0., 11.,  0.,  0., 49.]),
 array([0. , 0.3, 0.6, 0.9, 1.2, 1.5, 1.8, 2.1, 2.4, 2.7, 3. ]),
 [<matplotlib.patches.Polygon at 0x24df375a230>])
```

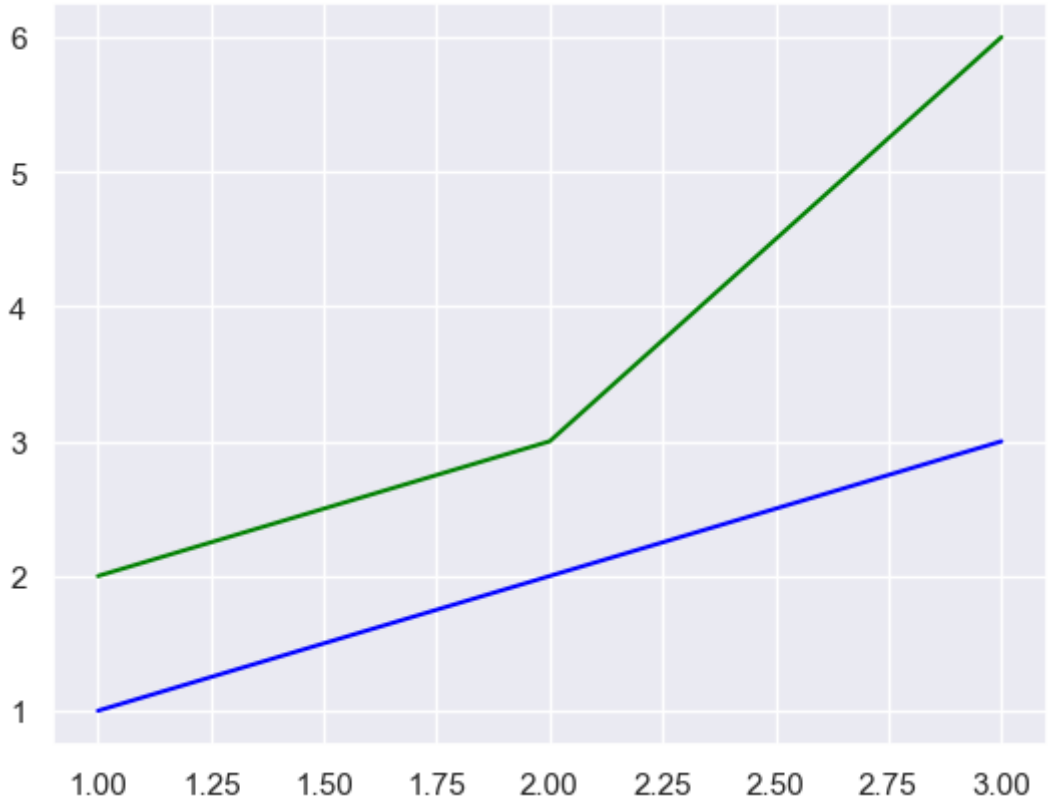


Задание 7

Постройте 95% и 99% доверительные интервалы для математического ожидания и стандартного отклонения генеральной совокупности для логарифма заработной платы.

```
# матожидание
data = table['ln_wage']

# test test
sns.lineplot(x=[1,2,3], y=[1,2,3], ci=95, color='blue')
sns.lineplot(x=[1,2,3], y=[2,3,6], ci=99, color='green')
```

```
<AxesSubplot:>
```




```
# стандартное отклонение
data = table['ln_wage']

# TODO

# test test
# sns.lineplot(x=[1,2,3], y=[1,2,3], ci=95, color='blue')
# sns.lineplot(x=[1,2,3], y=[2,3,6], ci=99, color='green')
```

Задание 8

Постройте 90% и 95% доверительный интервал для доли женщин в генеральной

```
female = table['male'] == 0

bar_X = len(female.values) / len(table.values)
S_X = female.values.std()
N = len(table.values) # 85

perc1 = 90
perc2 = 95

alpha1 = (100 - perc1) / 100 # 0.10
alpha2 = (100 - perc2) / 100 # 0.05

student1 = 1.6629785
student2 = 1.9882679

left1 = bar_X - (student1 * S_X / np.sqrt(N - 1))
right1 = bar_X + (student1 * S_X / np.sqrt(N - 1))

left2 = bar_X - (student2 * S_X / np.sqrt(N - 1))
right2 = bar_X + (student2 * S_X / np.sqrt(N - 1))

print("90%\t:\t", (left1, right1), sep='')
print("95%\t:\t", (left2, right2), sep='')

# sns.lineplot(x=[1,2,3], y=[1,2,3], ci=90, color='blue')
# sns.lineplot(x=[1,2,3], y=[2,3,6], ci=95, color='green')
```

```
90%     :       (0.9097870690471248, 1.0902129309528752)
95%     :       (0.8921408335835261, 1.1078591664164739)
```

Задание 9

Проверьте гипотезу, что матожидание логарифма заработной платы равно 10.17 против двусторонней и правосторонней альтернативной гипотезы.

```
# TODO
```

Задание 10

Проверьте гипотезу, что матожидание логарифма заработной платы женщин ниже матожидания логарифма заработной платы мужчин.

```
# TODO
```

Задание 11

Проверьте гипотезу, что дисперсия логарифма заработных плат работников, пользующихся Интернетом выше, чем дисперсия логарифма заработной платы работников, не пользующихся Интернетом. Примечание: выберите иной бинарный признак (тип населенного пункта, пол и т.д.), если в Вашей выборке нет различий по переменной «Интернет».

```
# TODO
```

Номера 12 - 14

```
all_data = pd.read_excel('data.xlsx')
```

```
data = all_data[all_data['id31'].astype(int) == 1].drop([col for col in all_data.columns i
data['ln_wage'] = np.log(data['wage'])
```

```
data = data.loc[:, ['public', 'ln_wage', 'educ', 'urban', 'male', 'age', 'children', 'indu
data = data.astype({'age': 'int', 'male': 'int', 'public': 'int', 'internet': 'int', 'chil
                    'urban': 'int', 'educ': 'int'})
data.head(3)
```

	public	ln_wage	educ	urban	male	age	children	industry	internet
180	0	10.126631	1	1	1	46	1	ТРАНСПОРТ, С	0
181	1	9.104980	3	1	0	38	1	СТРОИТЕЛЬСТВ	1
182	1	11.407565	3	1	0	55	2	ЮРИСПРУДЕНЦИ	1

Задание №12

Пусть `p` - доля работников, имеющих одно ребенка. Необходимо проверить гипотезу:
$H_0 : p = 0.5$
$H_1 : p < 0.5$

```
hat_p = data[data['children'] == 1].shape[0] / data.shape[0]
print(f"Доля работников с 1 ребенком в выборке: {hat_p:.4f}")
print(f"Размер выборки: {data.shape[0]}")
```

```
Доля работников с 1 ребенком в выборке: 0.3412
Размер выборки: 85
```

Таким образом, по нашей выборке: $\hat{p} = 0.3412$

$Z = \frac{\hat{p} - p_0}{\sqrt{\frac{p_0 q_0}{n}}}$, где $p_0 = 0.5, q_0 = 1 - p_0 = 0.5$ и n - размер выборки.

Тогда расчетное значение статистики: $Z(X^{(n)}) = \frac{0.3412 - 0.5}{\sqrt{\frac{1}{4n}}} = -2.928$

$Z \overset{D}{\longrightarrow} \mathcal{N}(0, 1)$ при $n \to \infty$ и справедливости гипотезы H_0

Тогда критическое значение статистики: $z_{cr} = \Phi^{-1}(0.05) = -1.65$

$p\_value = P(Z < Z(X^{(n)})|H_0) = 1 - P(Z \geq -2.928) = 0.002$

Так как $p\_value < 0.05$, то гипотеза H_0 отвергается в пользу альтернативной. То есть доля работников, имеющих одного ребенка не равна 0.5.

Задание №13

```
fig, ax = plt.subplots(1, 1, figsize=(10, 10))

ax.set_title("Корреляционная таблица", fontsize=20)
sns.heatmap(data.loc[:, ['public', 'ln_wage', 'educ', 'urban', 'male', 'age', 'children',
            ax=ax, annot=True, vmax=1, vmin=-1)
```

```
<AxesSubplot:title={'center':'Корреляционная таблица'}>
```


```
print(f"Количество различных значений в колонке urban: {data['urban'].nunique()}")
```

```
Количество различных значений в колонке urban: 1
```

Так как в колонке `urban` все значения одинаковы, то корреляционная таблица не содержит значений для этой колонки.

Вывод: В данных не так много сильно скоррелированных признаков. Некоторые из них:

1. Корреляция между `educ` и `internet` достигает `0.46`. Это, в целом, логично: образованные люди чаще пользуются интернетом, чем не образованные.
2. Корреляция между `age` и `children` также высока. Это отражает следующую логичную зависимость: чем старше человек, тем больше у него детей.

Задача №14

Предположительно `ln_wage` может зависеть, например, от образования человека. Построим график, демонстрирующий это.

Также логарифм зараплаты, может зависеть от пола, построим и такой график.

```
fig, ax = plt.subplots(1, 2, figsize=(20, 10))
```

```
ax[0].set_title("Среднее значение логарифма зарплаты от уровня образования", fontsize=20)
ax[0].set_ylabel("Логарифм зарплаты", fontsize=15)
ax[0].set_xlabel("Уровень образования", fontsize=15)
ax[0].bar(x=[0, 1, 2, 3], height=data.groupby('educ')['ln_wage'].mean() - 9, bottom=9)

ax[1].set_title("Среднее значение логарифма зарплаты от пола", fontsize=20)
ax[1].set_ylabel("Логарифм зарплаты", fontsize=15)
ax[1].set_xlabel("Пол", fontsize=15)
ax[1].bar(x=[0, 1], height=data.groupby('male')['ln_wage'].mean() - 9, bottom=9)
```

```
<BarContainer object of 2 artists>
```