Министерство науки и высшего образования Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«Национальный исследовательский университет ИТМО»
(Университет ИТМО)

Факультет инфокоммуникационных технологий

Отчет по дисциплине: **«Современные инструменты анализа данных»**

Лабораторная работа №2

Выполнила: Шурубова П.М.

Проверила: Максимова Татьяна Геннадьевна

Санкт-Петербург
2024

Задание 2.1
Проверить гипотезу о статистической значимости различия между доходами двух групп работающих и получающих доход граждан Петербурга:
1 группа - имеющие образование среднее и ниже,
2 группа - имеющие среднее специальное или высшее образование.

Для проверки гипотезы использовать однофакторный дисперсионный анализ.
Проверять по критерию Фишера. Вывести значения описательной статистики.

Листинг кода

```
import pandas as pd
from scipy import stats
import matplotlib.pyplot as plt
import seaborn as sns

file_path = '2.1.1.xlsx'
df = pd.read_excel(file_path, sheet_name='Лист1')

df = df[~df['w_diplom'].isin([99999997]) & df['w_diplom'].notna()]

df['wj10'] = pd.to_numeric(df['wj10'], errors='coerce')
df['wj10'] = df['wj10'].fillna(0).astype(int)

df['w_diplom'] = pd.to_numeric(df['w_diplom'], errors='coerce')
df['w_diplom'] = df['w_diplom'].fillna(0).astype(int)

count_non_empty = df['w_diplom'].count()
print(f"Количество непустых строк в столбце: {count_non_empty}")

education = df['w_diplom']
salary = df['wj10']

group1 = salary[education.isin([1, 2, 3, 4])]
group2 = salary[education.isin([5, 6])]

# Описательная статистика
desc_stats_group1 = group1.describe()
desc_stats_group2 = group2.describe()
print("Описательная статистика группы 1:\n", desc stats group1)
print("Описательная статистика группы 2:\n", desc_stats_group2)

# Однофакторный дисперсионный анализ
f_stat, p_value = stats.f_oneway(group1, group2)
print(f"Статистика Фишера: {f_stat}, p-значение: {p_value}")


data = {
    'Группа': ['Группа 1'] * len(group1) + ['Группа 2'] * len(group2),
    'Зарплата': list(group1) + list(group2)
}
viz_df = pd.DataFrame(data)

plt.figure(figsize=(10, 6))
sns.boxplot(x='Группа', y='Зарплата', data=viz_df, hue='Группа',
palette='Set2', legend=False)
plt.title('Box Plot для зарплат по группам')
plt.xlabel('Группа')
plt.ylabel('Зарплата')
plt.show()

# Рассчет коэффициента корреляции
correlation_coefficient = df['w_diplom'].corr(df['wj10'])
```

```
print(f'Коэффициент корреляции между w_diplom и wj10:
{correlation_coefficient}')

# Построение графика
plt.figure(figsize=(10, 6))
sns.scatterplot(x='w_diplom', y='wj10', data=df)
plt.title('Корреляция между w_diplom и wj10')
plt.xlabel('Уровень образования (w_diplom)')
plt.ylabel('Зарплата (wj10)')
plt.axhline(y=df['wj10'].mean(), color='r', linestyle='--', label='Средняя
зарплата')
plt.axvline(x=df['w_diplom'].mean(), color='g', linestyle='--',
label='Средний уровень образования')
plt.legend()
plt.show()
```

Вывод программы:

```
Количество непустых строк в столбце: 149
Описательная статистика группы 1:
 count       61.000000
mean     29785.245902
std      13170.457295
min       4300.000000
25%      20000.000000
50%      28000.000000
75%      36000.000000
max      70000.000000
Name: wj10, dtype: float64
Описательная статистика группы 2:
 count        88.000000
mean      35835.227273
std       14646.399579
min        8000.000000
25%       25000.000000
50%       34000.000000
75%       44250.000000
max      100000.000000
Name: wj10, dtype: float64
Статистика Фишера: 6.668023048918451, p-значение: 0.010792595536295713
```

Описательная статистика для группы 1:

- **Количество**: 61.
- **Среднее значение зарплаты (wj10)**: 29,785.25.
- **Стандартное отклонение**: 13,170.46.
- **Минимальная зарплата**: 4,300.
- **Максимальная зарплата**: 70,000.
- **Медиана (50%)**: 28,000, что указывает на то, что половина наблюдений имеет зарплату ниже этого значения.

- **Квартильные значения**: 25% (20,000), 75% (36,000) показывают, что 25% людей зарабатывают меньше 20,000, а 25% зарабатывают больше 36,000.

Описательная статистика для группы 2:

- **Количество**: 88.
- **Среднее значение зарплаты**: 35,835.23, что значительно выше, чем в первой группе.
- **Стандартное отклонение**: 14,646.40.
- **Минимальная и максимальная зарплата**: 8,000 и 100,000 соответственно.
- **Медиана**: 34,000, что говорит о том, что половина наблюдений имеет зарплату ниже этого значения.
- **Квартильные значения**: 25% (25,000), 75% (44,250).

Результаты статистического анализа

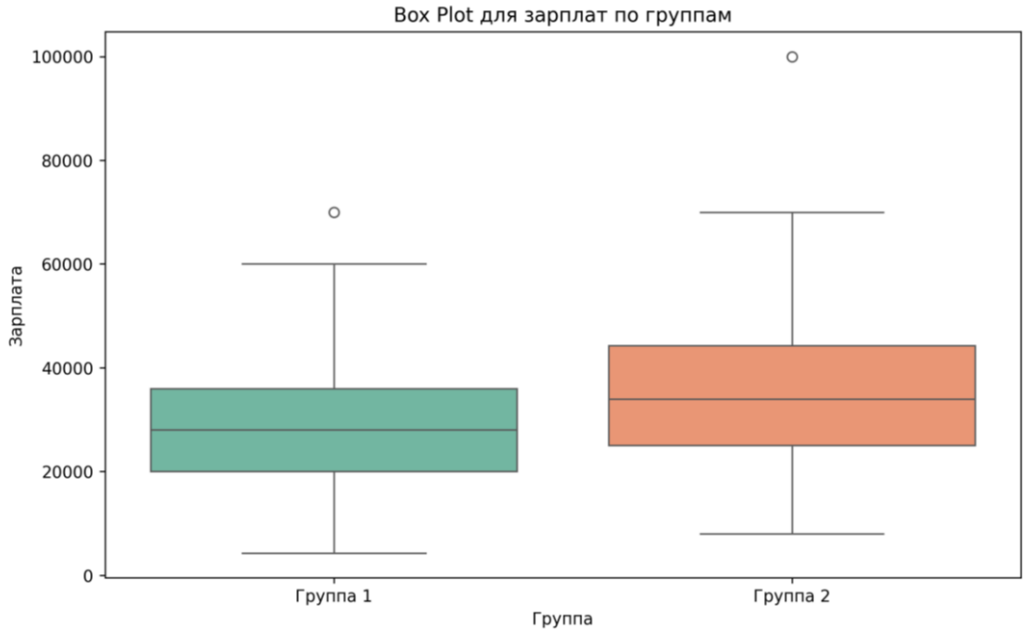
Статистика Фишера и p-значение:

Критерий Фишера: 6.67, что указывает на наличие различий между группами.

p-значение: 0.0108, что меньше 0.05, позволяет отвергнуть нулевую гипотезу. Это означает, что есть статистически значимые различия в средних значениях зарплаты между двумя группами.

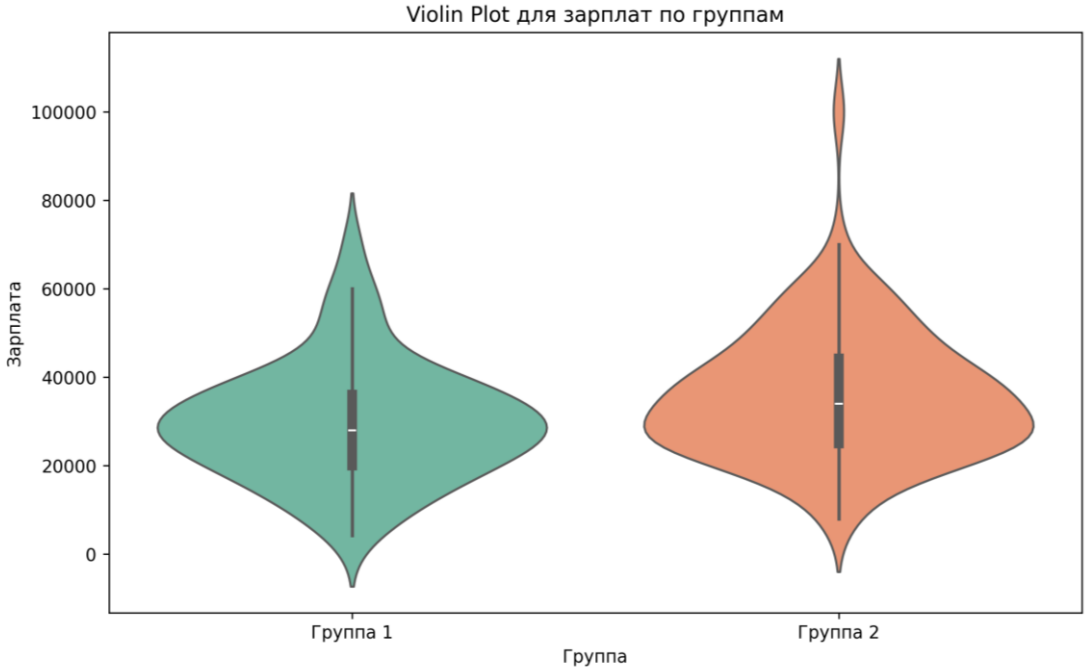
Общий вывод: вторая группа имеет значительно более высокие средние зарплаты по сравнению с первой группой, что может указывать на влияние фактора уровня образования.

Графическое представление результатов.



Виолончельный график показывает, как распределены данные вдоль оси значений. Это позволяет увидеть, где сосредоточены данные, и насколько они разрежены. Он

отображает не только медиану и квартильные значения, но и плотность распределения значений в различных диапазонах.



Для проверки гипотезы использовать Т-тест для независимых выборок. Проверять по критерию Стьюдента. Вывести значения описательной статистики и описательные графики.

Листинг кода.

```
import pandas as pd
from scipy import stats
import matplotlib.pyplot as plt
import seaborn as sns

file_path = '2.1.1.xlsx'
df = pd.read_excel(file_path, sheet_name='Лист1')

df = df[~df['w_diplom'].isin([99999997]) & df['w_diplom'].notna()]

df['wj10'] = pd.to_numeric(df['wj10'], errors='coerce').fillna(0).astype(int)
df['w_diplom'] = pd.to_numeric(df['w_diplom'],
errors='coerce').fillna(0).astype(int)

education = df['w_diplom']
salary = df['wj10']

group1 = salary[education.isin([1, 2, 3, 4])]
group2 = salary[education.isin([5, 6])]

# Описательная статистика
desc_stats_group1 = group1.describe()
desc_stats_group2 = group2.describe()
print("Описательная статистика группы 1:\n", desc_stats_group1)
print("Описательная статистика группы 2:\n", desc_stats_group2)

# Т-тест для независимых выборок
t_stat, p_value = stats.ttest_ind(group1, group2, equal_var=False)
print(f"Статистика t: {t_stat}, p-значение: {p_value}")

data = {
    'Группа': ['Группа 1'] * len(group1) + ['Группа 2'] * len(group2),
```

```
    'Зарплата': list(group1) + list(group2)
}
viz_df = pd.DataFrame(data)
data = {
    'Группа': ['Группа 1'] * len(group1) + ['Группа 2'] * len(group2),
    'Зарплата': list(group1) + list(group2),
    'Уровень образования': list(education[education.isin([1, 2, 3, 4])]) +
list(education[education.isin([5, 6])])
}
viz_df = pd.DataFrame(data)

# Построение точечной диаграммы
plt.figure(figsize=(10, 6))
sns.scatterplot(x='Уровень образования', y='Зарплата', hue='Группа',
data=viz_df, palette='Set2')
plt.title('Точечная диаграмма зарплат по уровням образования')
plt.xlabel('Уровень образования (w_diplom)')
plt.ylabel('Зарплата (wj10)')
plt.legend()
plt.show()
```

Вывод программы:

```
Описательная статистика группы 1:
 count       61.000000
mean     29785.245902
std      13170.457295
min       4300.000000
25%      20000.000000
50%      28000.000000
75%      36000.000000
max      70000.000000
Name: wj10, dtype: float64
Описательная статистика группы 2:
 count        88.000000
mean      35835.227273
std       14646.399579
min        8000.000000
25%       25000.000000
50%       34000.000000
75%       44250.000000
max      100000.000000
Name: wj10, dtype: float64
Статистика t: -2.632588514983112, p-значение: 0.009444010854348452
```

Описательная статистика

Группа 1:

- **Количество наблюдений**: 61.
- **Средняя зарплата**: 29,785.25.
- **Стандартное отклонение**: 13,170.46.

- **Минимальная и максимальная зарплата**: 4,300 и 70,000 соответственно.
- **Медиана**: 28,000.
- **Квартильные значения**: 25% (20,000) и 75% (36,000) показывают распределение доходов в группе.

Группа 2:

- **Количество наблюдений**: 88.
- **Средняя зарплата**: 35,835.23.
- **Стандартное отклонение**: 14,646.40.
- **Минимальная и максимальная зарплата**: 8,000 и 100,000.
- **Медиана**: 28,000.
- **Квартильные значения**: 25% (25,000) и 75% (44,250) показывают распределение доходов в группе.

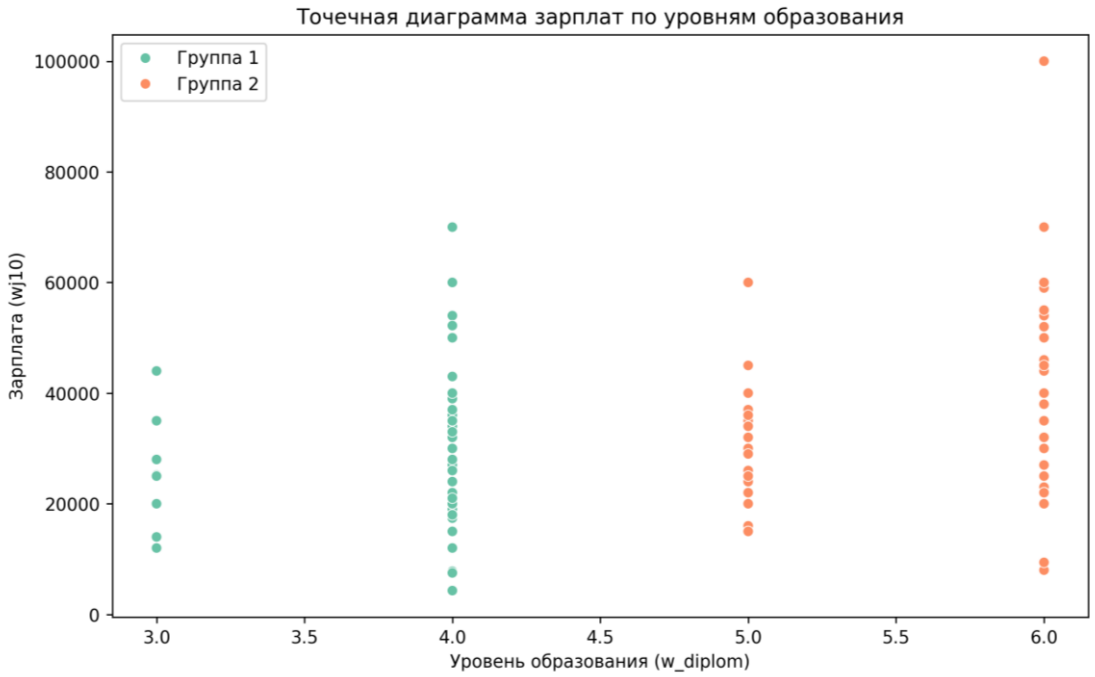
Результаты статистического анализа

t-тест: -2.63. Это значение указывает на то, что средние значения зарплат в двух группах различаются.

p-значение: 0.0094. Это значение значительно меньше 0.05, что позволяет отвергнуть нулевую гипотезу о равенстве средних значений зарплат в двух группах. Это означает, что существует статистически значимая разница в зарплатах между группами.

Общий вывод: вторая группа имеет значительно более высокие средние зарплаты по сравнению с первой, что может указывать на влияние различных факторов, таких как образование, опыт или сфера деятельности. Уровни значимости практически совпадают, различие составляет в 0.001349.

Графическое представление результатов.



Задание 2.2
Для выделенных ранее групп проверить гипотезу о равенстве средней продолжительности работы в неделю.
Использовала критерий Стьюдента.

Листинг кода.

```
import pandas as pd
from scipy import stats
import matplotlib.pyplot as plt
import seaborn as sns

file_path = '2.1.1.xlsx'
df = pd.read_excel(file_path, sheet_name='Лист2')

df = df[~df['w_diplom'].isin([99999997]) & df['w_diplom'].notna()]

df['wj10'] = pd.to_numeric(df['wj10'], errors='coerce').fillna(0).astype(int)
df['w_diplom'] = pd.to_numeric(df['w_diplom'],
errors='coerce').fillna(0).astype(int)
df['wj6.2'] = pd.to_numeric(df['wj6.2'],
errors='coerce').fillna(0).astype(int)

count_non_empty = df['w_diplom'].count()
print(f"Количество непустых строк в столбце: {count_non_empty}")

education = df['w_diplom']
work_hours = df['wj6.2']

group1 = work_hours[education.isin([1, 2, 3, 4])]
group2 = work_hours[education.isin([5, 6])]

# Описательная статистика
desc_stats_group1 = group1.describe()
desc_stats_group2 = group2.describe()
print("Описательная статистика группы 1:\n", desc_stats_group1)
print("Описательная статистика группы 2:\n", desc_stats_group2)

# T-тест для независимых выборок
t_stat, p_value = stats.ttest_ind(group1, group2, equal_var=False)
print(f"Статистика t: {t_stat}, p-значение: {p_value}")

# Подготовка данных для визуализации
data = {
    'Группа': ['Группа 1'] * len(group1) + ['Группа 2'] * len(group2),
    'Часы работы': list(group1) + list(group2)
}
viz_df = pd.DataFrame(data)

# Построение графика
plt.figure(figsize=(10, 6))
sns.boxplot(x='Группа', y='Часы работы', hue='Группа', data=viz_df,
palette='Set2', legend=False)
plt.title('Box Plot для часов работы по группам')
plt.xlabel('Группа')
plt.ylabel('Часы работы (wj6.2)')
plt.show()
```

Вывод программы:

```
Количество непустых строк в столбце: 140
Описательная статистика группы 1:
 count    58.000000
mean     42.568966
std      11.839737
min       8.000000
25%      40.000000
50%      41.000000
75%      48.000000
max      72.000000
Name: wj6.2, dtype: float64
Описательная статистика группы 2:
 count    82.000000
mean     43.378049
std      11.102931
min      24.000000
25%      40.000000
50%      40.000000
75%      45.000000
max      96.000000
Name: wj6.2, dtype: float64
Статистика t: -0.4086362396338476, p-значение: 0.6835480786984957
```

Описательная статистика

Группа 1:

- **Количество**: 58
- **Среднее значение**: 42.57 часов
- **Стандартное отклонение**: 11.84, что указывает на то, что данные довольно разнообразны.
- **Минимум**: 8 часов
- **Максимум**: 72 часа
- **Квартиль 25%** : 40 часов
- **Медиана**: 41 час
- **Квартиль 75%**: 48 часов

Группа 2:

- **Количество**: 82
- **Среднее значение**: 43.38 часов
- **Стандартное отклонение**: 11.10, чуть меньше, чем в группе 1.
- **Минимум**: 24 часа

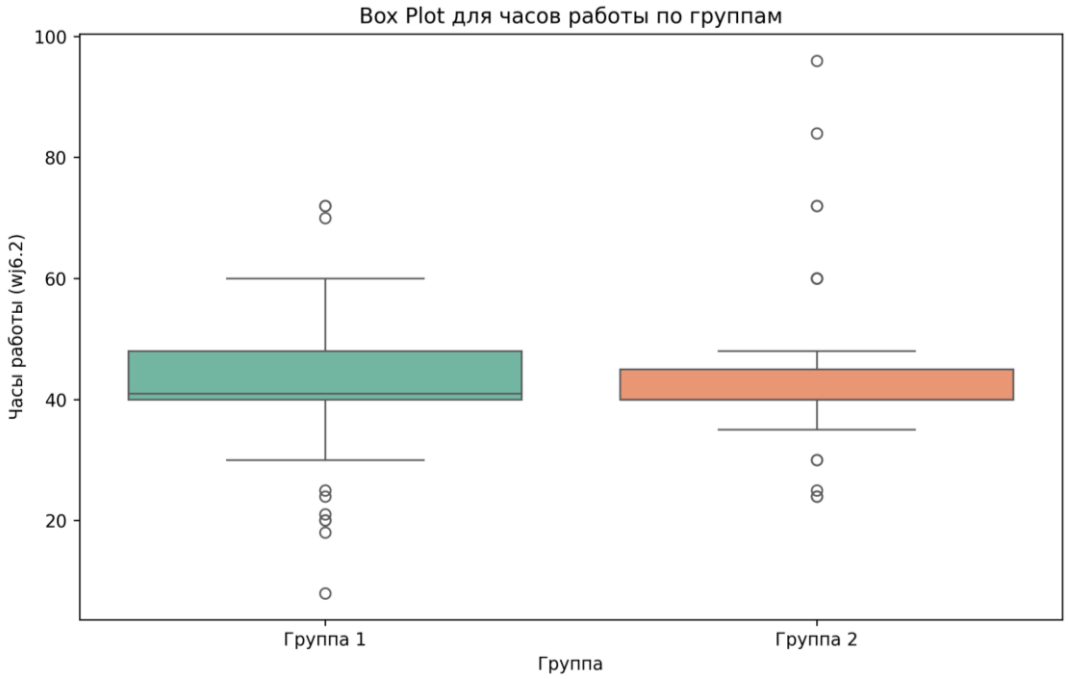
- **Максимум**: 96 часов
- **Квартиль 25%**: 40 часов
- **Медиана**: 40 часов
- **Квартиль 75%**: 45 часов

Результаты теста

- **Статистика t**: -0.41
- **p-значение**: 0.68

Общий вывод: в целом, продолжительность работы между группами не отличается значимо, хотя в группе 2 наблюдается чуть выше среднее значение. Поскольку p-значение (0.68) значительно выше уровня значимости (обычно 0.05), это указывает на то, что нет статистически значимых различий в средней продолжительности работы между двумя группами. Следовательно, мы не можем отвергнуть гипотезу о равенстве средних значений для этих двух групп.

Графическое представление результатов.



Задание 2.3
Для работающих, указавших продолжительность работы и получающих доход граждан (проживающих в любом населенном пункте) исследовать взаимосвязь двух признаков: курение и употребление алкоголя за последние 30 дней. Использовать Частотный анализ, Таблицы сопряженности парных выборок.

Рассматривала респондентов, которые проживают в Краснодаре.

Листинг кода.

```
import pandas as pd
from scipy import stats
import matplotlib.pyplot as plt
import seaborn as sns

file_path = '2.1.1.xlsx'
df = pd.read_excel(file_path, sheet_name='Лист3')
```

```
df = df[~df['w_diplom'].isin([99999997]) & df['w_diplom'].notna()]

df['wm71'] = pd.to_numeric(df['wm71'], errors='coerce').fillna(0).astype(int)
df['wm80'] = pd.to_numeric(df['wm80'], errors='coerce').fillna(0).astype(int)

df_working = df[df['wj6.2'] > 0]

count_non_empty = df_working['wm71'].count()
print(f"Количество непустых строк в столбце: {count_non_empty}")

# Создание таблицы сопряженности
contingency_table = pd.crosstab(df_working['wm71'], df_working['wm80'],
                                rownames=['Курение'],
colnames=['Употребление алкоголя'])
print("Таблица сопряженности:\n", contingency_table)

# Выполнение критерия хи-квадрат для независимости
chi2_stat, p_value, dof, expected = stats.chi2_contingency(contingency_table)
print(f"Статистика хи-квадрат: {chi2_stat}, p-значение: {p_value}")

# Проверка гипотезы
alpha = 0.05
if p_value < alpha:
    print("Отвергаем нулевую гипотезу: признаки связаны.")
else:
    print("Не отвергаем нулевую гипотезу: признаки независимы.")

# Визуализация таблицы сопряженности
plt.figure(figsize=(8, 6))
sns.heatmap(contingency_table, annot=True, fmt='d', cmap='Blues')
plt.title('Таблица сопряженности: Курение и Употребление алкоголя')
plt.xlabel('Употребление алкоголя')
plt.ylabel('Курение')
plt.show()
```

Вывод программы:

```
Количество непустых строк в столбце: 73
Таблица сопряженности:
 Употребление алкоголя   1   2
Курение
1                       23   8
2                       27  15
Статистика хи-квадрат: 0.4171550457490146, p-значение: 0.5183600455269031
Не отвергаем нулевую гипотезу: признаки независимы.
```

Из таблицы видно, что:

- Из 31 курящего (1) 23 употребляют алкоголь (1) и 8 — нет (2).
- Из 42 некурящих (2) 27 употребляют алкоголь (1) и 15 — нет (2).

Статистика хи-квадрат: 0.4171550457490146 — низкое значение, указывающее на небольшую разницу между ожидаемыми и фактическими значениями.

p-значение: 0.5183600455269031 — значительно выше 0.05, что означает, что нет статистически значимой связи между курением и употреблением алкоголя.

Общий вывод: так как p-значение больше 0.05, вы не отвергаете нулевую гипотезу о независимости признаков. Это означает, что между курением и употреблением алкоголя нет статистически значимой связи в вашей выборке.

Графическое представление результатов.