МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования
«Дальневосточный федеральный университет»
(ДВФУ)

ИНСТИТУТ МАТЕМАТИКИ И КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ

Лабораторная работа №7
Тестирование гипотез о распределениях

Дисциплина «Теория вероятностей и математическая статистика»

Студент группы Б9123-01.03.02ии
Моттуева Уруйдана Михайловна

г. Владивосток
2025

1. Напишите функцию, реализующую тестирование гипотезы о распределении случайной величины с помощью критерия Колмогорова. На вход функции подаются: выборка, предполагаемая в рамках нулевой гипотезы функция распределения. Возвращает функция p-значение. Можно использовать библиотечную эмпирическую функцию распределения.

Теорема Колмогорова

Критерий Колмогорова используется для проверки гипотезы о том, что выборка происходит из заданного распределения

Случайная выборка $X_1, X_2, \dots, X_n$ из X.

Гипотезы
$H_0: F_X = F;$
$H_1: F_X \neq F.$

Статистика

$$D_n = sup_{x \in R} \left| \hat{F}_n(x) - F_X(x) \right|,$$

Где

- $\hat{F}_n(x)$ - эмпирическая функция распределения
- $F_X(x)$ - теоретическая функция распределения

Асимптотическое распределение статистики $\sqrt{n} D_n$ при $n \to \infty$ задается функцией Колмогорова:

$$F_K(x) = \sum_{k=-\infty}^{+\infty} (-1)^k e^{-2k^2x^2} \cdot I(x > 0)$$

p-значение

$$1 - F_K(x)$$

Моя реализация

```python
def kolmogorov_test(sample, cdf):
    n = len(sample)
    sample_sorted = np.sort(sample)

    y = np.arange(1, n + 1) / n #ЭФР

    # Вычисляем статистику D_n = sup |F_n(x) - F(x)|
    D_plus = np.max(y - cdf(sample_sorted))
    D_minus = np.max(cdf(sample_sorted) - (y - 1/n))
    D_n = max(D_plus, D_minus)

    # Вычисляем p-значение через асимптотическое распределение Колмогорова
    sqrt_n_D = np.sqrt(n) * D_n
    p_value = 1 - kolmogorov_cdf(sqrt_n_D)

    return p_value

def kolmogorov_cdf(x, terms=100):
    if x <= 0:
        return 0.0

    total = 0.0

    for k in range(-terms, terms + 1):
        term = (-1)**k * np.exp(-2 * k**2 * x**2)
        total += term

    return total
```

Реализация со scipy

```python
stats.kstest(sample1, 'uniform', args=(0, 1))[1])
```

2. Напишите функцию, реализующую тестирование гипотезы о распределении с помощью критерия хи квадрат. На вход функции подаются: выборка, разбиение числовой оси на интервалы, предполагаемая функция распределения. Возвращает функция p-значение.

Пусть случайные величины $X_1, X_2, \dots, X_n \sim N(0;1)$ не зависимы. Тогда случайная величина

$$\sum_{i=1}^{n} X_i^2 = X \sim \chi^2(n)$$

Имеет распределение χ^2 с n степенями свободы.
Плотность распределения $\chi^2(n)$

$$f_X(x) = \frac{x^{\frac{n}{2}-1} e^{-\frac{x}{2}}}{2^{\frac{n}{2}} \Gamma\left(\frac{n}{2}\right)}.$$

Критерий хи-квадрат проверяет гипотезу о том, что случайная выборка $X_1, X_2, \dots, X_n$ происходит из заданного распределения X.

Гипотезы
$H_0: F_X = F$;
$H_1: F_X \neq F$.

Статистика:

$$\chi^2 = \sum_{i=1}^{k} \frac{(O_i - E_i)^2}{E_i} \sim \chi^2(k-1)$$

Где:
O_i – наблюдаемая частота в i-м интервале
$E_i = np_i$ – ожидаемая частота в i-м интервале
p_i – вероятность попадания в i-й интервал согласно теоретическому распределению

p-значение

$$p = 1 - F_{\chi^2}(\chi^2)$$

Моя реализация

```python
def chi_square_test(sample, bins, cdf):
    # Разбиваем выборку на интервалы
    observed, _ = np.histogram(sample, bins=bins)
    n = len(sample)
    k = len(bins) - 1

    # Вычисляем ожидаемые частоты
    expected = []
    for i in range(k):
        p = cdf(bins[i+1]) - cdf(bins[i])
        expected.append(n * p)
    expected = np.array(expected)

    chi2 = np.sum((observed - expected)**2 / expected)
    p_value = 1 - stats.chi2.cdf(chi2, df=k-1)
    return p_value
```

Реализация со scipy

```python
chisquare(np.histogram(sample1, bins=bins)[0], f_exp=len(sample1)*np.diff(uniform.cdf(bins, loc=0, scale=1)))[1])
chisquare(observed, expected)
```

3. Напишите функцию, реализующую тестирование гипотезы об однородности выборок. На вход функции подаются 2 выборки. Возвращает функция p-значение.

Даны 2 случайные выборки $X_1, X_2, \dots, X_n$ из X и $Y_1, Y_2, \dots, Y_\mathrm{m}$ из Y.

Гипотезы
$H_0: F_X = F_Y$;
$H_1: F_X \neq F_Y$.

Для проверки однородности двух выборок можно использовать критерий Колмогорова-Смирнова.

Статистика:

$$K = \sqrt{\frac{n+m}{nm}} \cdot sup_{x \in R} \left| \hat{\mathrm{F}}_n(x) - \hat{\mathrm{F}}_m(x) \right|.$$

p-значение

$$p = 1 - F_K(k).$$

Моя реализация

```python
# 3. Критерий однородности двух выборок
def homogeneity_test(sample1, sample2):
    n = len(sample1)
    m = len(sample2)
    combined = np.concatenate([sample1, sample2])
    combined_sorted = np.sort(combined)

    ecdf1 = np.searchsorted(np.sort(sample1), combined_sorted, side='right') / n
    ecdf2 = np.searchsorted(np.sort(sample2), combined_sorted, side='right') / m

    K = np.sqrt((n * m) / (n + m)) * np.max(np.abs(ecdf1 - ecdf2))

    p_value = 0
    for k in range(1, 100):
        p_value += (-1)**(k-1) * np.exp(-2 * k**2 * K **2)
    p_value = 2 * p_value

    return p_value
```

Реализация со scipy

```python
ks_2samp(sample1, sample3)[1])
```

4. Вычислите остаток от деления своего номера в списке группы на 3. Это номер распределения: 0) нормальное, 1) равномерное, 2) биномиальное. Из распределения сгенерируйте 3 выборки. Две с одинаковыми параметрами распределения, а третью с другими.

```python
remainder = 13 % 3  # 1 → равномерное распределение
print(f"Остаток от деления 13 на 3: {remainder} → равномерное распределение")

np.random.seed(42)
sample1 = uniform.rvs(loc=0, scale=1, size=100)
sample2 = uniform.rvs(loc=0, scale=1, size=100)
sample3 = uniform.rvs(loc=2, scale=3, size=100)  # другие параметры
```

5. Проверьте гипотезы:
1) ваша выборка имеет то распределение, откуда она взялась (по Колмогорову и по хи квадрат);

```
print("Гипотеза 1: выборка имеет исходное распределение")
print("\nКолмогоров (наша реализация):", kolmogorov_test(sample1, lambda x: uniform.cdf(x, loc=0, scale=1)))
print("Колмогоров (scipy):", stats.kstest(sample1, 'uniform', args=(0, 1))[1])

bins = np.linspace(0, 1, 6)
print("\nХи-квадрат (наша реализация):", chi_square_test(sample1, bins, lambda x: uniform.cdf(x, loc=0, scale=1)))
print("Хи-квадрат (scipy):", chisquare(np.histogram(sample1, bins=bins)[0],
                                       f_exp=len(sample1)*np.diff(uniform.cdf(bins, loc=0, scale=1)))[1])
```

```
Гипотеза 1: выборка имеет исходное распределение

Колмогоров (наша реализация): 0.5388440301329993
Колмогоров (scipy): 0.5130102757799564

Хи-квадрат (наша реализация): 0.3926414559428024
Хи-квадрат (scipy): 0.3926414559428025
```

2) ваша выборка имеет какое-то другое конкретное распределение (теми же критериями);

```
print("\nГипотеза 2: выборка имеет нормальное распределение N(0.5, 0.1)")
print("\nКолмогоров (наша реализация):", kolmogorov_test(sample1, lambda x: norm.cdf(x, loc=0.5, scale=0.1)))
print("Колмогоров (scipy):", stats.kstest(sample1, 'norm', args=(0.5, 0.1))[1])

bins = np.linspace(-0.5, 1.5, 6)
print("\nХи-квадрат (наша реализация):", chi_square_test(sample1, bins, lambda x: norm.cdf(x, loc=0.5, scale=0.1)))
print("Хи-квадрат (scipy):", chisquare(np.histogram(sample1, bins=bins)[0],
                                       f_exp=len(sample1) * np.diff(norm.cdf(bins, loc=0.5, scale=0.1)))[1])
```

```
Гипотеза 2: выборка имеет нормальное распределение N(0.5, 0.1)

Колмогоров (наша реализация): 5.697029032832625e-12
Колмогоров (scipy): 2.0222170034673587e-12

Хи-квадрат (наша реализация): 0.0
Хи-квадрат (scipy): 1.367901688713591e-175
```

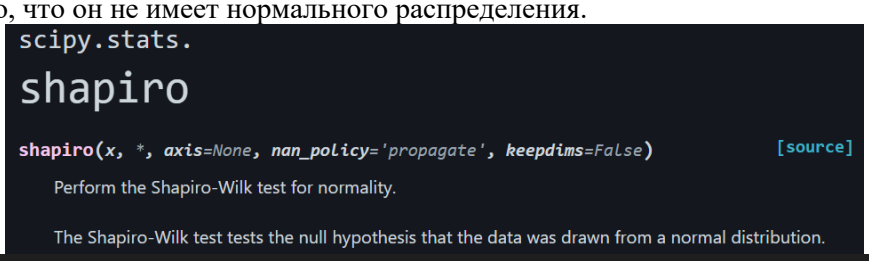
3) для каждой пары выборок гипотезу однородности;

```
print("\nГипотеза 3: проверка однородности выборок")
print("\nsample1 и sample2 (наша реализация):", homogeneity_test(sample1, sample2))
print("sample1 и sample2 (scipy):", ks_2samp(sample1, sample2)[1])

print("\nsample1 и sample3 (наша реализация):", homogeneity_test(sample1, sample3))
print("sample1 и sample3 (scipy):", ks_2samp(sample1, sample3)[1])

print("\nsample2 и sample3 (наша реализация):", homogeneity_test(sample2, sample3))
print("sample2 и sample3 (scipy):", ks_2samp(sample2, sample3)[1])
```

```
Гипотеза 3: проверка однородности выборок

sample1 и sample2 (наша реализация): 0.5806177304235198
sample1 и sample2 (scipy): 0.5830090612540064

sample1 и sample3 (наша реализация): 7.440151952041566e-44
sample1 и sample3 (scipy): 2.2087606931995054e-59

sample2 и sample3 (наша реализация): 7.440151952041566e-44
sample2 и sample3 (scipy): 2.2087606931995054e-59
```

4) нормальность распределения по критерию Шапиро-Вилка (библиотечный). Для гипотез 1-3) сверьте результат с библиотечными тестами.

Тест Шапиро-Уилка — это проверка гипотез, которая применяется к выборке данных с нулевой гипотезой о том, что выборка имеет нормальное распределение. В этом тесте высокое значение p указывает на то, что набор данных имеет нормальное распределение, тогда как низкое значение p указывает на то, что он не имеет нормального распределения.




```
print("\nГипотеза 4: проверка нормальности распределения")
print("\nsample1 (Шапиро-Вилк):", shapiro(sample1)[1])
print("sample2 (Шапиро-Вилк):", shapiro(sample2)[1])
print("sample3 (Шапиро-Вилк):", shapiro(sample3)[1])
```

```
Гипотеза 4: проверка нормальности распределения

sample1 (Шапиро-Вилк): 0.00021827262119561032
sample2 (Шапиро-Вилк): 0.00019088827928236428
sample3 (Шапиро-Вилк): 0.0005908620134494505
```

https://colab.research.google.com/drive/1thne1JNRAEuqBC9aHpyZ4PbRHkb6aaOU?usp=sharing