**БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**ФАКУЛЬТЕТ ПРИКЛАДНОЙ МАТЕМАТИКИ И ИНФОРМАТИКИ**

**Кафедра теории вероятностей и математической статистики**

**ОТЧЕТ**

по лабораторной работе №2

«Ковариационная функция, семивариограмма и спектральная плотность стационарного в широком смысле случайного процесса»

учебной дисциплины

«Математические методы анализа данных»

Вариант №3

**Выполнила:**

Лавринович Анна Павловна,

3 курс 7а группа, специальность «прикладная математика»

**Преподаватель:**

Цеховая Татьяна Вячеславовна,

кандидат физико-математических наук, доцент

Минск, 2025

**Постановка задачи.** Для стационарных в широком смысле случайных процессов с известными ковариационными функциями найти аналитический вид их семивариограмм и спектральных плотностей. Сделать вывод о свойствах процессов.

**Необходимо:**

Рассмотреть требуемые модели ковариационных функций стационарных случайных процессов с непрерывным временем . Указать, к какому классу относятся исследуемые модели: *колебательному, монотонно убывающему, …*

* Модель ковариационной функции представить в общем виде с указанием всех параметров.
* Найти аналитический вид семивариограммы
* Найти аналитический вид спектральной плотности
* Построить графики функций и при различных сочетаниях параметров (минимум 3 значения для каждого параметра). Сделать сравнительный анализ.
* Вычислить время корреляции по представленным ниже формулам:

а) б) в)

Сделать сравнительный анализ длин интервалов корреляции.

* Вычислить эффективную ширину спектра по формуле:

Если основная мощность процесса сосредоточена в точке т. е. . Проверить выполнение неравенства неопределенности:

Если основная мощность процесса сосредоточена вблизи экстремальной частоты спектральной плотности, т. е. , то ширину спектра вычислить по формуле:

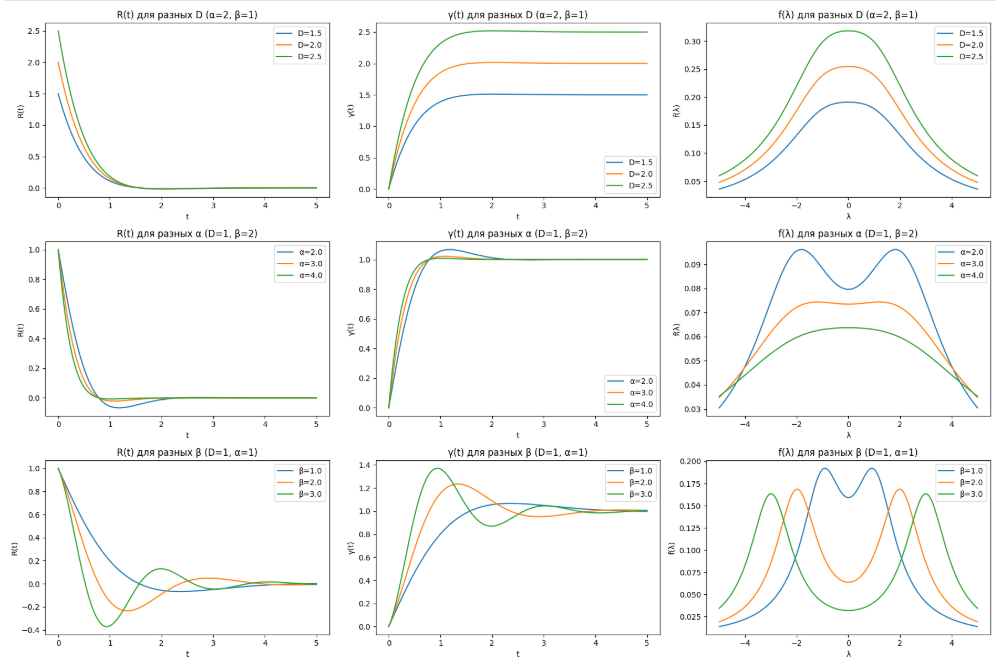
**Исходные данные (алгоритм выполнения).**

1. Дана ковариационная функция

Получим аналитический вид спектральной плотности:

Получим аналитический вид семивариограммы:

Построим графики функций и при различных сочетаниях параметров:



Вычислим время корреляции по представленным выше формулам:

Вычислим эффективную ширину спектра .

, но условие выполняется только при и тогда

При максимум достигается в точках и тогда , а ширина спектра

Проверим выполнение неравенства неопределенности:

Подставляем приближенное значение :

.

Так как , то , и , однако , поэтому неравенство выполняется не всегда.

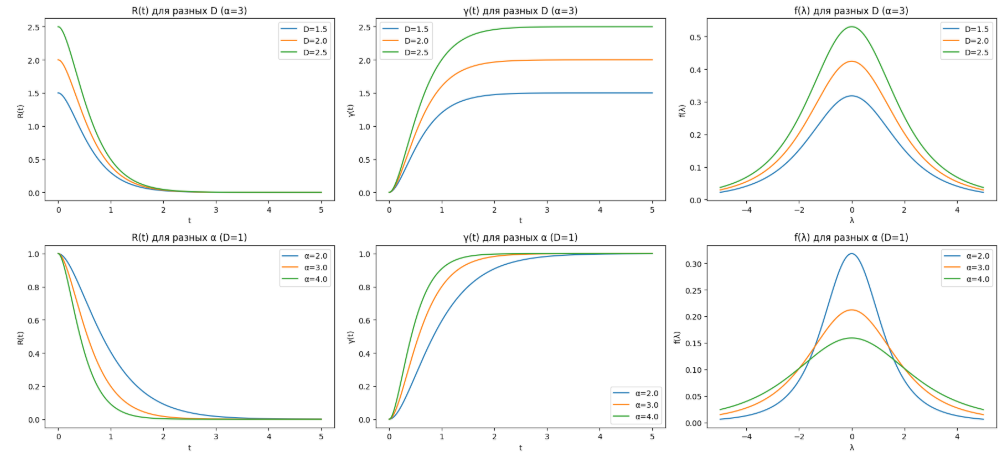
Таким образом

1. Дана ковариационная функция

Получим аналитический вид спектральной плотности:

Получим аналитический вид семивариограммы:

Построим графики функций и при различных сочетаниях параметров:



Вычислим время корреляции по представленным выше формулам:

Вычислим эффективную ширину спектра (эффективная мощность спектра сосредоточена в нуле для всех значений параметра):

Неравенство неопределенности выполняется:

**Вывод. Часть первая.**

**Анализ графиков.**

**Параметр D.**

* Ковариационная функция: Увеличение D приводит к увеличению амплитуды функции, что указывает на более выраженные корреляции.
* Спектральная плотность: Высота графиков спектральной плотности увеличивается с ростом D, что указывает на усиление мощности сигнала.
* Семивариограмма: Увеличение D приводит к увеличению значений семивариограммы, указывая на большую вариативность данных.

**Параметры альфа и бета.**

* Ковариационная функция: При фиксированном D увеличение относительно делает графики более узкими и смещает их влево, указывая на более быстрое затухание ковариации.
* Спектральная плотность: Увеличение параметра приводит к расширению и сглаживанию функции распределения, в то время как большие значения соответствуют более узким и острым пикам, указывающим на концентрацию значений вокруг центра распределения.
* Семивариограмма: Увеличение приводит к более быстрой потере корреляции, а увеличение может сделать затухание более плавным, указывая на долгосрочные корреляции.

**Сравнительный анализ длин интервалов корреляции.** Все три формулы показывают, что увеличение параметра приводит к увеличению длины интервала корреляции, тогда как увеличение может вызвать его уменьшение.

**Часть вторая.**

**Анализ графиков.**

**Параметр D.**

* Ковариационная функция: Увеличение D приводит к более выраженному затуханию функции. Графики показывают, что с увеличением D корреляция становится менее выраженной на коротких временных интервалах.
* Спектральная плотность: С ростом D наблюдается увеличение амплитуды, что указывает на более высокую мощность сигнала на определенных частотах.
* Семивариограмма: Увеличение D ведет к увеличению значений семивариограммы, что свидетельствует о более высокой вариативности данных.

**Параметр альфа.**

* Ковариационная функция: При фиксированном D изменение влияет на скорость затухания. При больших значениях графики становятся более пологими, указывая на замедленное затухание.
* Спектральная плотность: Увеличение приводит к смещению пиков к более низким значениям частот, делая спектры более пологими.
* Семивариограмма: Большие значения соответствуют более высоким и острым пикам функции.

**Сравнительный анализ длин интервалов корреляции.** Все три формулы показывают, что увеличение параметра вызывает уменьшение длины интервала корреляции.

**Листинг программы**

# Первая часть

import numpy as np

import matplotlib.pyplot as plt

from scipy.integrate import quad

# Параметры для построения графиков

D\_values = [1.5, 2.0, 2.5]

alpha\_values = [2.0, 3.0, 4.0]

beta\_values = [1.0, 2.0, 3.0]

# Временная ось

t = np.linspace(0, 5, 500)

# Частотная ось

lambda\_vals = np.linspace(-5, 5, 500)

# Функция ковариации R(t)

def covariance\_func(t, D, alpha, beta):

return D \* np.exp(-alpha \* np.abs(t)) \* np.cos(beta \* t)

# Семивариограмма gamma(t) = R(0) - R(t)

def semivariogram(t, D, alpha, beta):

return D - covariance\_func(t, D, alpha, beta)

# Спектральная плотность f(lambda)

def spectral\_density(lambda\_val, D, alpha, beta):

term1 = D \* alpha \* (alpha\*\*2 + lambda\_val\*\*2 + beta\*\*2)

term2 = (alpha\*\*2 + (lambda\_val - beta)\*\*2) \* (alpha\*\*2 + (lambda\_val + beta)\*\*2)

return term1 / (np.pi \* term2)

# Построение графиков

plt.figure(figsize=(18, 12))

# Графики для разных D (alpha=2, beta=1)

plt.subplot(3, 3, 1)

for D in D\_values:

plt.plot(t, covariance\_func(t, D, 2.0, 1.0), label=f'D={D}')

plt.title('R(t) для разных D (α=2, β=1)')

plt.xlabel('t')

plt.ylabel('R(t)')

plt.legend()

plt.subplot(3, 3, 2)

for D in D\_values:

plt.plot(t, semivariogram(t, D, 2.0, 1.0), label=f'D={D}')

plt.title('γ(t) для разных D (α=2, β=1)')

plt.xlabel('t')

plt.ylabel('γ(t)')

plt.legend()

plt.subplot(3, 3, 3)

for D in D\_values:

plt.plot(lambda\_vals, spectral\_density(lambda\_vals, D, 2.0, 1.0), label=f'D={D}')

plt.title('f(λ) для разных D (α=2, β=1)')

plt.xlabel('λ')

plt.ylabel('f(λ)')

plt.legend()

# Графики для разных alpha (D=1, beta=2)

plt.subplot(3, 3, 4)

for alpha in alpha\_values:

plt.plot(t, covariance\_func(t, 1.0, alpha, 2.0), label=f'α={alpha}')

plt.title('R(t) для разных α (D=1, β=2)')

plt.xlabel('t')

plt.ylabel('R(t)')

plt.legend()

plt.subplot(3, 3, 5)

for alpha in alpha\_values:

plt.plot(t, semivariogram(t, 1.0, alpha, 2.0), label=f'α={alpha}')

plt.title('γ(t) для разных α (D=1, β=2)')

plt.xlabel('t')

plt.ylabel('γ(t)')

plt.legend()

plt.subplot(3, 3, 6)

for alpha in alpha\_values:

plt.plot(lambda\_vals, spectral\_density(lambda\_vals, 1.0, alpha, 2.0), label=f'α={alpha}')

plt.title('f(λ) для разных α (D=1, β=2)')

plt.xlabel('λ')

plt.ylabel('f(λ)')

plt.legend()

# Графики для разных beta (D=1, alpha=1)

plt.subplot(3, 3, 7)

for beta in beta\_values:

plt.plot(t, covariance\_func(t, 1.0, 1.0, beta), label=f'β={beta}')

plt.title('R(t) для разных β (D=1, α=1)')

plt.xlabel('t')

plt.ylabel('R(t)')

plt.legend()

plt.subplot(3, 3, 8)

for beta in beta\_values:

plt.plot(t, semivariogram(t, 1.0, 1.0, beta), label=f'β={beta}')

plt.title('γ(t) для разных β (D=1, α=1)')

plt.xlabel('t')

plt.ylabel('γ(t)')

plt.legend()

plt.subplot(3, 3, 9)

for beta in beta\_values:

plt.plot(lambda\_vals, spectral\_density(lambda\_vals, 1.0, 1.0, beta), label=f'β={beta}')

plt.title('f(λ) для разных β (D=1, α=1)')

plt.xlabel('λ')

plt.ylabel('f(λ)')

plt.legend()

plt.tight\_layout()

plt.show()

# Вторая часть

import numpy as np

import matplotlib.pyplot as plt

from scipy.integrate import quad

# Параметры для построения графиков

D\_values = [1.5, 2.0, 2.5]

alpha\_values = [2.0, 3.0, 4.0]

# Временная ось

t = np.linspace(0, 5, 500)

# Частотная ось

lambda\_vals = np.linspace(-5, 5, 500)

# Функция ковариации R(t)

def covariance\_func(t, D, alpha):

return D \* np.exp(-alpha \* np.abs(t)) \* (1 + alpha \* np.abs(t))

# Семивариограмма gamma(t) = R(0) - R(t)

def semivariogram(t, D, alpha):

return D - covariance\_func(t, D, alpha)

# Спектральная плотность f(lambda)

def spectral\_density(lambda\_val, D, alpha):

return (D / np.pi) \* (2 \* alpha\*\*3) / (alpha\*\*2 + lambda\_val\*\*2)\*\*2

# Построение графиков

plt.figure(figsize=(18, 12))

# Графики для разных D (alpha=3)

plt.subplot(3, 3, 1)

for D in D\_values:

plt.plot(t, covariance\_func(t, D, 3.0), label=f'D={D}')

plt.title('R(t) для разных D (α=3)')

plt.xlabel('t')

plt.ylabel('R(t)')

plt.legend()

plt.subplot(3, 3, 2)

for D in D\_values:

plt.plot(t, semivariogram(t, D, 3.0), label=f'D={D}')

plt.title('γ(t) для разных D (α=3)')

plt.xlabel('t')

plt.ylabel('γ(t)')

plt.legend()

plt.subplot(3, 3, 3)

for D in D\_values:

plt.plot(lambda\_vals, spectral\_density(lambda\_vals, D, 3.0), label=f'D={D}')

plt.title('f(λ) для разных D (α=3)')

plt.xlabel('λ')

plt.ylabel('f(λ)')

plt.legend()

# Графики для разных alpha (фиксируем D=1)

plt.subplot(3, 3, 4)

for alpha in alpha\_values:

plt.plot(t, covariance\_func(t, 1.0, alpha), label=f'α={alpha}')

plt.title('R(t) для разных α (D=1)')

plt.xlabel('t')

plt.ylabel('R(t)')

plt.legend()

plt.subplot(3, 3, 5)

for alpha in alpha\_values:

plt.plot(t, semivariogram(t, 1.0, alpha), label=f'α={alpha}')

plt.title('γ(t) для разных α (D=1)')

plt.xlabel('t')

plt.ylabel('γ(t)')

plt.legend()

plt.subplot(3, 3, 6)

for alpha in alpha\_values:

plt.plot(lambda\_vals, spectral\_density(lambda\_vals, 1.0, alpha), label=f'α={alpha}')

plt.title('f(λ) для разных α (D=1)')

plt.xlabel('λ')

plt.ylabel('f(λ)')

plt.legend()

plt.tight\_layout()

plt.show()

**Постановка задачи:** Для реальных временных рядов с периодической компонентой и без нее выполнить вариограммный (ковариационный) анализ. В качестве реальных временных рядов рассмотреть температурные ряды (архив погоды в Минске) с сайта <http://rp5.by>

**Необходимо:**

* Выбрать ряды из таблицы согласно варианту. Объем ряда без периодической компоненты равен количеству дней в рассматриваемом месяце. Объем ряда с периодической компонентой равен 48.
* Временные ряды представить графически.
* Выполнить предварительный статистический анализ временных рядов (описательные статистики, гистограмма, тренд-анализ). Сделать вывод.
* Вычислить оценки семивариограммы и ковариационной функции.

Пусть n последовательных, полученных через равные промежутки времени наблюдений за случайным процессом. В качестве оценки семивариограммы рассмотрим статистику вида

Положим и для

В качестве оценки ковариационной функции рассмотрим статистику

Положим и для

* Построить графики оценок семивариограммы и ковариационной функции для лага
* Визуальным методом подобрать модель семивариограммы [6]. Модель представить в общем виде с указанием всех параметров. Объяснить выбор параметров.
* Найти аналитический вид ковариационной функции.
* Сделать вывод о свойствах процесса. Указать, к какому классу относится процесс: широкополосный или узкополосный.

**Исходные данные (алгоритм выполнения).**

**Часть первая**

1. Дан временной ряд без периодической компоненты:

1,5 1,4 3,7 6,5 9,3 15,3 12,6 5,7 6,0 8,9 10,3 8,4 9,1 6,2 0,8 -0,3 -1,2

-2,8 4,5 5,3 5,2 7,2 2,5 5,7 6,9 5,0 9,6 6,4 8,3 8,7 11,1

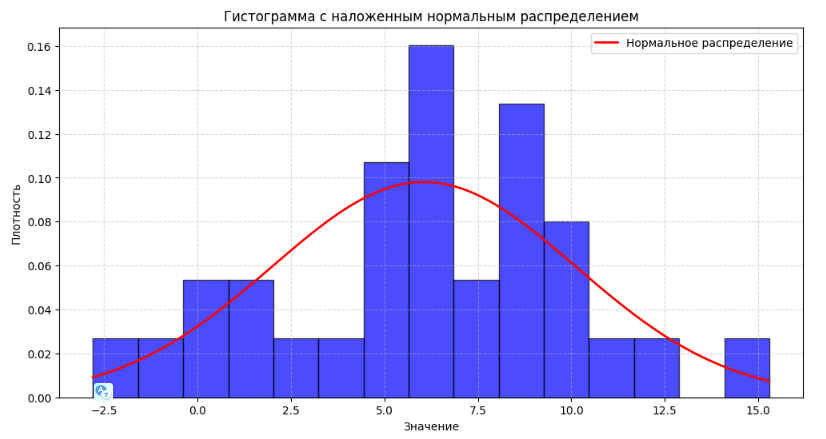
1. Представим этот ряд графически:



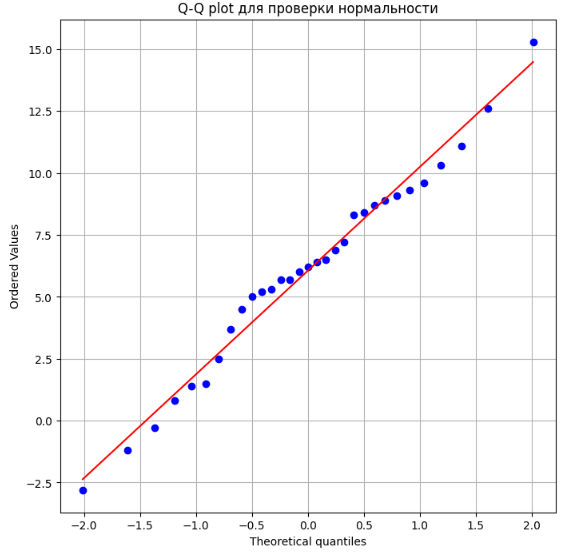
1. Описательные статистики:

* Среднее: 6.0581
* Дисперсия: 16.5625
* Стандартное отклонение: 4.0697
* Медиана: 6.2000
* Первый квартиль (Q1): 4.1000
* Третий квартиль (Q3): 8.8000
* Асимметрия: -0.1466
* Эксцесс: -0.1052

Гистограмма:



Проверка гипотезы на нормальность (тест Шапиро-Уилка):



Тест Шапиро-Уилка: p-value = 0.9257. Не отвергаем гипотезу о нормальности (p > 0.05)

Тренд-анализ (тест Манна-Кендалла):

Тест Манна-Кендалла на тренд: p-value = 0.6101. Значимый тренд не обнаружен (p > 0.05).

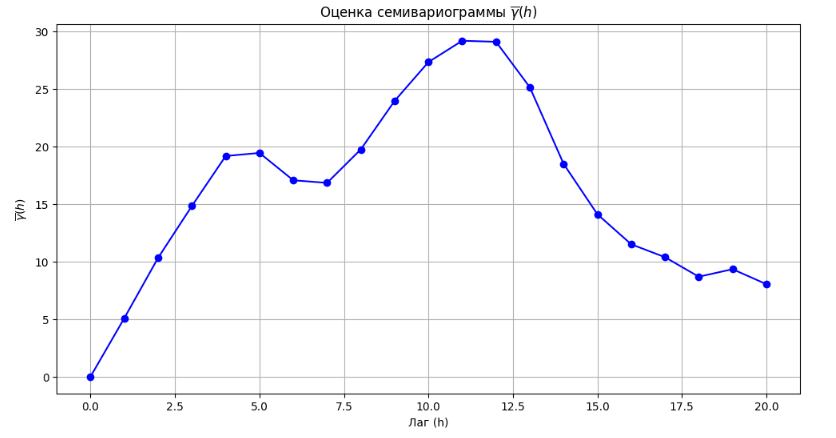
Вывод: Описательные статистики. Распределение имеет небольшую левостороннюю асимметрию, что означает легкий сдвиг в сторону меньших значений. Распределение близко к нормальному по форме, но с незначительной плосковершинностью, что указывает на немного более сглаженный пик по сравнению с эталонным нормальным распределением.

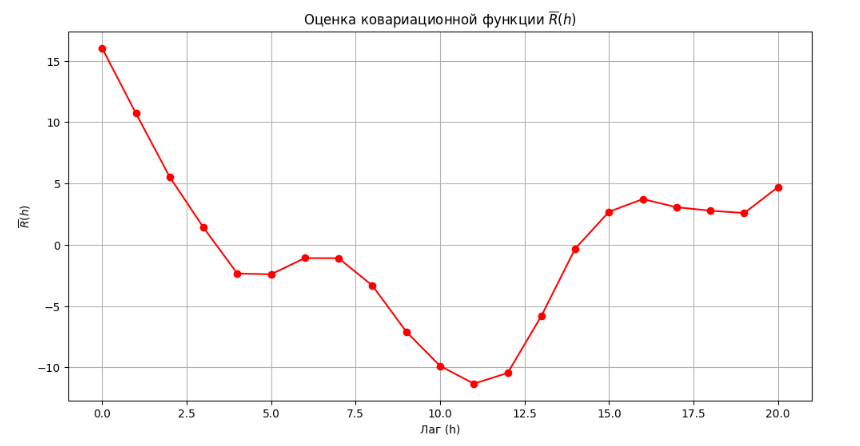
Проверка на нормальность. Гистограмма демонстрирует распределение значений с наложенной кривой нормального распределения. Визуально наблюдается соответствие эмпирических данных теоретической нормальной кривой. Тест Шапиро-Уилка не отвергает гипотезу о нормальности, что подтверждает возможность соответствия данных нормальному распределению.

Тренд-анализ. Тест Манна-Кендалла не выявил статистически значимого тренда. Поскольку p-value превышает 0.05, можно сделать вывод об отсутствии выраженного монотонного тренда в данных. Это позволяет рассматривать ряд как стационарный относительно среднего уровня.

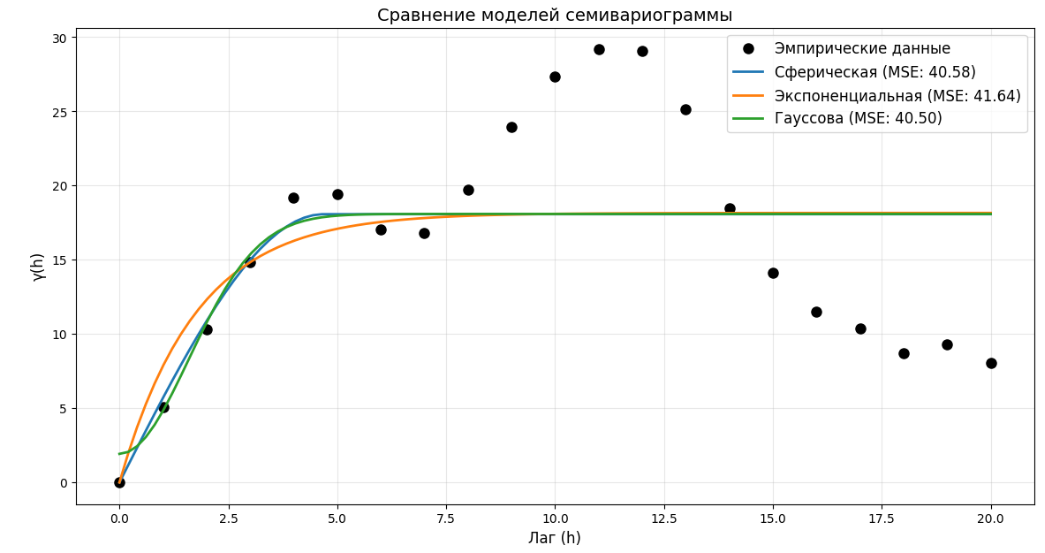
Общий вывод. Данные демонстрируют распределение, близкое к нормальному, с незначительной асимметрией и отсутствием тренда.

1. Вычислим оценки семивариограммы и ковариационной функции и построим их графики для лага





1. Теперь визуальным методом подберём модель семивариограммы. Для этого сравним три модели: сферическую, экспоненциальную и Гауссову и сделаем вывод на основе среднеквадратичной ошибки и визуального представления.



Здесь мы можем видеть, что у Гауссовой модели самая маленькая среднеквадратичная ошибка, однако ошибка сферической модели ненамного больше и она лучше повторяет эмпирическую кривую, поэтому выберем её.

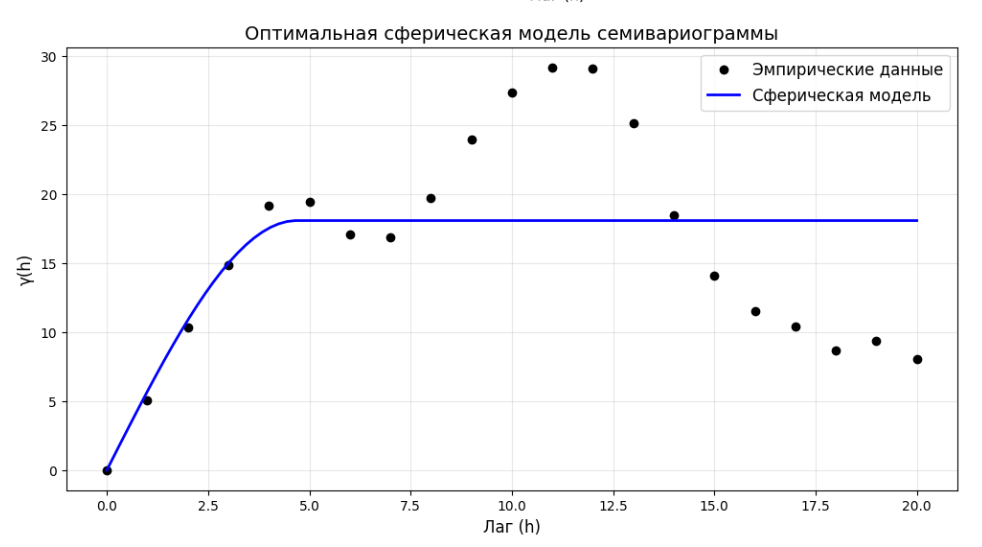
В качестве изначальных параметров возьмём параметры, полученные с помощью метода наименьших квадратов, и посмотрим, подходят ли они к нашей модели.

Сферическая модель:

c0 (эффект самородков): 0.00 – да, семивариограмма начинается с 0

c (предельное значение): 18.08 – да, значение близко к значению дисперсии

a (диапазон): 4.69 – да, на графике рост замедляется при h≈5



1. Найдём аналитический вид ковариационный функции.

Для нахождения аналитического вида ковариационной функции используем фундаментальную связь между семивариограммой и ковариационной функцией , где — семивариограмма, — ковариационная функция, — полная дисперсия процесса . И отсюда выразим ковариационную функцию

Из оптимальной сферической модели семивариограммы имеем:

Получаем окончательный вид ковариационной функции для сферической модели:

Подставляя наши параметры, найденные выше, получаем

1. Выводы о свойствах процесса:

Процесс является стационарным; ковариационная функция становится нулевой при , значит процесс узкополосной.

Листинг программы:

#1

import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

from scipy import stats

from scipy.stats import kendalltau

from scipy.optimize import curve\_fit

from sklearn.metrics import mean\_squared\_error

data = np.array([

1.5, 1.4, 3.7, 6.5, 9.3, 15.3, 12.6, 5.7, 6.0, 8.9, 10.3, 8.4, 9.1, 6.2, 0.8, -0.3, -1.2,

-2.8, 4.5, 5.3, 5.2, 7.2, 2.5, 5.7, 6.9, 5.0, 9.6, 6.4, 8.3, 8.7, 11.1

])

n = len(data)

t = np.arange(n)

#2. График временного ряда

plt.figure(figsize=(12, 6))

plt.plot(t, data, marker='o', linestyle='-', label='Среднесуточная температура в марте')

plt.title('Среднесуточная температура в марте')

plt.xlabel('День')

plt.ylabel('Температура')

plt.grid(True)

plt.legend()

plt.show()

#3.Предварительный статистический анализ

# Описательные статистики

mu\_hat = np.mean(data)

var\_hat = np.var(data, ddof=1)

skew\_hat = stats.skew(data)

kurt\_hat = stats.kurtosis(data)

median = np.median(data)

q1 = np.percentile(data, 25)

q3 = np.percentile(data, 75)

print("\nОписательные статистики:")

print(f"Среднее: {mu\_hat:.4f}")

print(f"Дисперсия: {var\_hat:.4f}")

print(f"Стандартное отклонение: {np.sqrt(var\_hat):.4f}")

print(f"Медиана: {median:.4f}")

print(f"Первый квартиль (Q1): {q1:.4f}")

print(f"Третий квартиль (Q3): {q3:.4f}")

print(f"Асимметрия: {skew\_hat:.4f}")

print(f"Эксцесс: {kurt\_hat:.4f}")

# Гистограмма с нормальным распределением

plt.figure(figsize=(12, 6))

plt.hist(data, bins=15, density=True, alpha=0.7, color='blue', edgecolor='black')

x\_norm = np.linspace(min(data), max(data), 100)

plt.plot(x\_norm, stats.norm.pdf(x\_norm, mu\_hat, np.sqrt(var\_hat)),

'r-', lw=2, label='Нормальное распределение')

plt.title('Гистограмма с наложенным нормальным распределением')

plt.xlabel('Значение')

plt.ylabel('Плотность')

plt.legend()

plt.grid(True, linestyle='--', alpha=0.5)

plt.show()

# Q-Q plot для проверки нормальности

plt.figure(figsize=(8, 8))

stats.probplot(data, dist="norm", plot=plt)

plt.title('Q-Q plot для проверки нормальности')

plt.grid(True)

plt.show()

# Проверка на нормальность (тест Шапиро-Уилка)

shapiro\_test = stats.shapiro(data)

print(f"\nТест Шапиро-Уилка: p-value = {shapiro\_test.pvalue:.4f}")

if shapiro\_test.pvalue > 0.05:

print("Не отвергаем гипотезу о нормальности (p > 0.05)")

else:

print("Отвергаем гипотезу о нормальности (p ≤ 0.05)")

# Проверка на тренд (тест Манна-Кендалла)

tau, p\_value = kendalltau(t, data)

print(f"\nТест Манна-Кендалла на тренд: p-value = {p\_value:.4f}")

if p\_value > 0.05:

print("Значимый тренд не обнаружен (p > 0.05)")

else:

print("Обнаружен значимый тренд (p ≤ 0.05)")

#4.Оценки семивариограммы и ковариационной функции

def semivariogram\_estimate(data, max\_lag=None):

n = len(data)

if max\_lag is None:

max\_lag = n - 1

gamma = np.zeros(max\_lag + 1)

for h in range(max\_lag + 1):

if h == 0:

gamma[h] = 0

else:

sum\_sq = 0

for s in range(n - h):

sum\_sq += (data[s + h] - data[s])\*\*2

gamma[h] = sum\_sq / (2 \* (n - h))

return gamma

def covariance\_estimate(data, max\_lag=None):

n = len(data)

if max\_lag is None:

max\_lag = n - 1

mu = np.mean(data)

R = np.zeros(max\_lag + 1)

for h in range(max\_lag + 1):

if h == 0:

R[h] = np.mean((data - mu)\*\*2)

else:

sum\_prod = 0

for s in range(n - h):

sum\_prod += (data[s + h] - mu) \* (data[s] - mu)

R[h] = sum\_prod / (n - h)

return R

max\_lag = int(2 \* n / 3) # Лаг до 2n/3

gamma = semivariogram\_estimate(data, max\_lag)

R = covariance\_estimate(data, max\_lag)

lags = np.arange(max\_lag + 1)

plt.figure(figsize=(12, 6))

plt.plot(lags, gamma, marker='o', linestyle='-', color='b')

plt.title(r'Оценка семивариограммы $\overline{\gamma}(h)$')

plt.xlabel('Лаг (h)')

plt.ylabel(r'$\overline{\gamma}(h)$')

plt.grid(True)

plt.show()

plt.figure(figsize=(12, 6))

plt.plot(lags, R, marker='o', linestyle='-', color='r')

plt.title(r'Оценка ковариационной функции $\overline{R}(h)$')

plt.xlabel('Лаг (h)')

plt.ylabel(r'$\overline{R}(h)$')

plt.grid(True)

plt.show()

# 5.Сравнение моделей

def spherical\_model(h, c0, c, a):

gamma = np.zeros\_like(h)

mask = (h <= a) & (h > 0)

gamma[mask] = c0 + c\*(1.5\*h[mask]/a - 0.5\*(h[mask]/a)\*\*3)

gamma[h > a] = c0 + c

return gamma

def exponential\_model(h, c0, c, a):

return c0 + c\*(1 - np.exp(-3\*h/a))

def gaussian\_model(h, c0, c, a):

return c0 + c\*(1 - np.exp(-3\*(h/a)\*\*2))

# Функция подгонки параметров

def fit\_and\_evaluate(model\_func, lags, gamma\_empirical):

try:

params, \_ = curve\_fit(model\_func, lags[1:], gamma\_empirical[1:],

bounds=([0, 0, 0], [np.inf, np.inf, np.inf]))

predicted = model\_func(lags[1:], \*params)

mse = mean\_squared\_error(gamma\_empirical[1:], predicted)

return {

'params': params,

'mse': mse,

'model\_func': model\_func

}

except:

return None

models = {

"Сферическая": spherical\_model,

"Экспоненциальная": exponential\_model,

"Гауссова": gaussian\_model

}

results = {}

for name, model in models.items():

results[name] = fit\_and\_evaluate(model, lags, gamma)

results = {k: v for k, v in results.items() if v is not None}

plt.figure(figsize=(14, 7))

plt.plot(lags, gamma, 'ko', markersize=8, label='Эмпирические данные')

for name, result in results.items():

h\_fit = np.linspace(0, max\_lag, 100)

plt.plot(h\_fit, result['model\_func'](h\_fit, \*result['params']),

linewidth=2, label=f'{name} (MSE: {result["mse"]:.2f})')

print(f'\n{name} модель:')

print(f'c0 (эффект самородков): {result["params"][0]:.2f}')

print(f'c (предельное значение): {result["params"][1]:.2f}')

print(f'a (диапазон): {result["params"][2]:.2f}')

print(f'Ошибка MSE: {result["mse"]:.2f}')

plt.title(f'Сравнение моделей семивариограммы', fontsize=14)

plt.xlabel('Лаг (h)', fontsize=12)

plt.ylabel('γ(h)', fontsize=12)

plt.legend(fontsize=12)

plt.grid(True, alpha=0.3)

plt.show()

if "Сферическая" in results:

sph\_result = results["Сферическая"]

plt.figure(figsize=(12, 6))

plt.plot(lags, gamma, 'ko', markersize=6, label='Эмпирические данные')

h\_fit = np.linspace(0, max\_lag, 100)

plt.plot(h\_fit, sph\_result['model\_func'](h\_fit, \*sph\_result['params']),

'b-', linewidth=2, label='Сферическая модель')

plt.title('Оптимальная сферическая модель семивариограммы', fontsize=14)

plt.xlabel('Лаг (h)', fontsize=12)

plt.ylabel('γ(h)', fontsize=12)

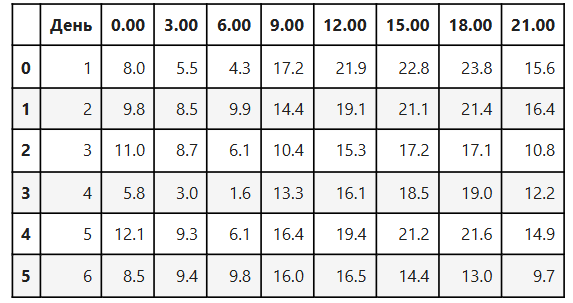
plt.legend(fontsize=12)

plt.grid(True, alpha=0.3)

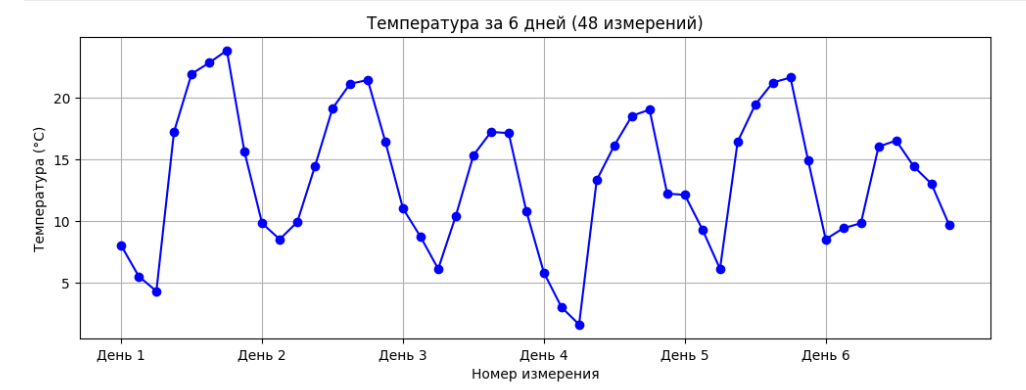
plt.show()

**Часть вторая**

1. Дан временной ряд с периодической компонентой:



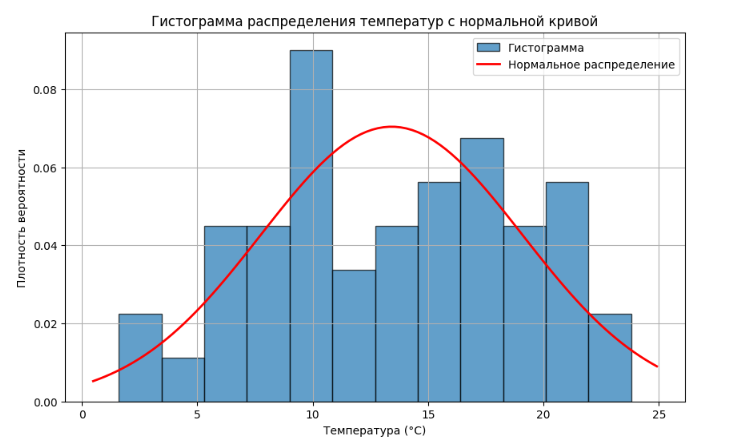
1. Представим этот ряд графически:

****

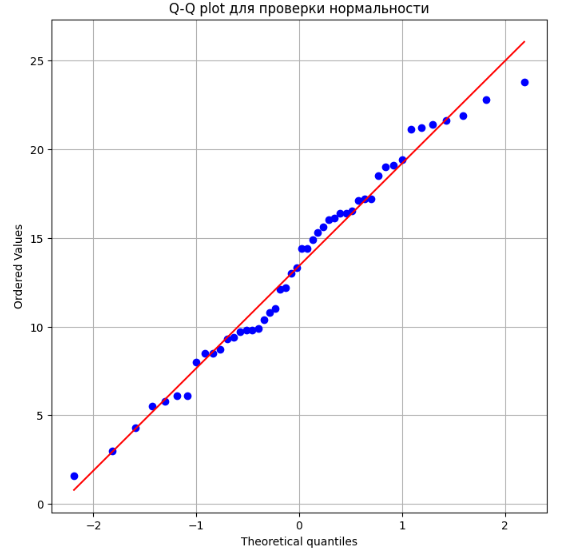
1. Описательные статистики:

* Среднее: 13.42 °C
* Дисперсия: 32.10 °C²
* Стандартное отклонение: 5.67 °C
* Медиана: 13.85 °C
* Первый квартиль (Q1): 9.38 °C
* Третий квартиль (Q3): 17.20 °C
* Асимметрия: -0.07
* Эксцесс: -0.88

Гистограмма с наложенным нормальным распределением



Проверка гипотезы на нормальность (тест Шапиро-Уилка):



Тест Шапиро-Уилка: p-value = 0.4027. Не отвергаем гипотезу о нормальности (p > 0.05)

Тренд-анализ (тест Манна-Кендалла):

Тест Манна-Кендалла на тренд: p-value = 0.9009. Значимый тренд не обнаружен (p > 0.05).

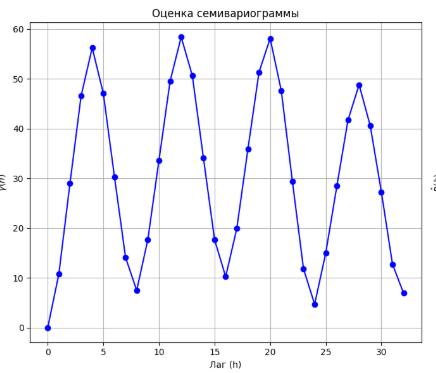
Вывод: Описательные статистики. Распределение имеет почти нулевую асимметрию, что говорит о симметричности распределения. Эксцесс (-0.88) указывает на более плоскую и легковесную форму распределения по сравнению с нормальным.

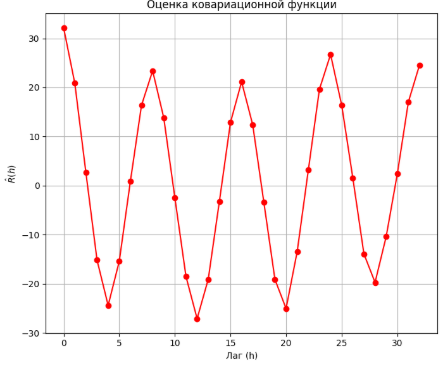
Проверка на нормальность. Гистограмма демонстрирует распределение значений с наложенной кривой нормального распределения. Визуально наблюдается соответствие эмпирических данных теоретической нормальной кривой. Тест Шапиро-Уилка не отвергает гипотезу о нормальности, что подтверждает возможность соответствия данных нормальному распределению.

Тренд-анализ. Тест Манна-Кендалла не выявил статистически значимого тренда. Поскольку p-value превышает 0.05, можно сделать вывод об отсутствии выраженного монотонного тренда в данных. Это позволяет рассматривать ряд как стационарный относительно среднего уровня.

Общий вывод: Данные о температуре имеют распределение, близкое к нормальному, без значимой асимметрии или выбросов.

1. Вычислим оценки семивариограммы и ковариационной функции и построим их графики для лага :





1. Теперь визуальным методом подберём модель семивариограммы.

Исходя из наших данных семивариограммы, можно наблюдать периодическое поведение с затуханием амплитуды, что характерно для волнообразных процессов. Тогда рассмотрим три подходящие модели семивариограммы: периодическая модель (Hole-Effect), смешанная модель (экспоненциальная + периодическая), модель Гаусса с периодичностью - и сделаем вывод на основе среднеквадратичной ошибки и визуального представления.

Периодическая модель (Hole-Effect): ,

где - эффект самородка, - структурная дисперсия, - период колебаний

Смешанная модель (Экспоненциальная + Периодическая):

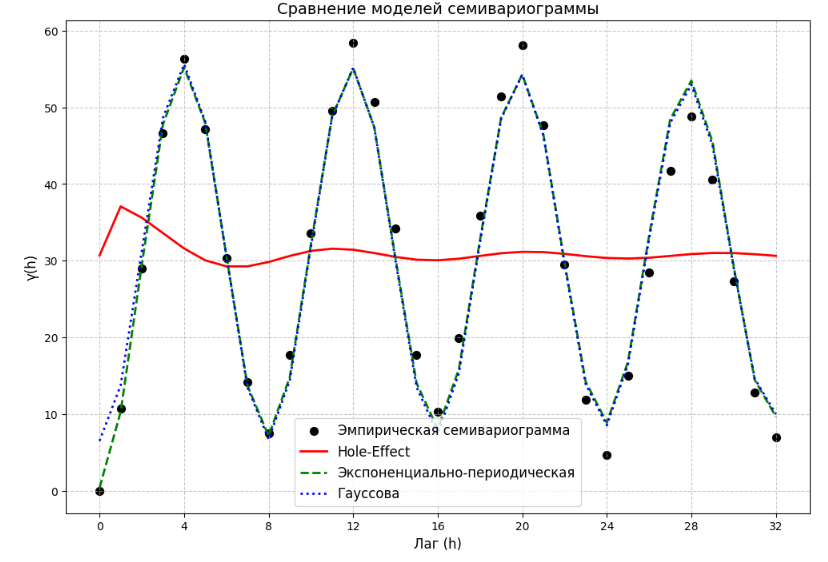
,

где - эффект самородка, , - параметры экспоненциальной компоненты, , , - параметры периодической компоненты

Смешанная модель: Гаусса + периодическая: ,

где - эффект самородка - структурная дисперсия, - параметр затухания, - период колебаний.

В качестве параметров для всех моделей возьмём параметры, полученные с помощью метода наименьших квадратов, и посмотрим, подходят ли они к нашей модели.



Сравнение MSE:

Hole-Effect: 279.63

Экспоненциально-периодическая: 8.57

Гауссова: 9.07

Здесь мы можем видеть, что у Hole-Effect модели самая высокая среднеквадратичная ошибка и она визуально далека от нашей эмпирической семивариограммы, в то время, как Гауссовой и экспоненциально-периодической моделей маленькие среднеквадратичные ошибки, они стремятся следовать эмпирической семивариограмме и схожи между собой. Мы остановимся на экспоненциально-периодической модели, потому что у ней самая маленькая MSE, график показывает почти идеальное совпадение модели с эмпирическими точками, особенно точное положение пиков, правильную амплитуду колебаний и сохранение формы на больших лагах

С помощью метода наименьших квадратов мы получили следующие параметры:

=0.38, =6.01, =1.89

=24.99, =7.97, =236.12

Они хорошо нам подходят, потому что:

* =0.38 - чрезвычайно малая величина (практически 0), указывает на высокую пространственную непрерывность данных, соседние измерения почти идентичны;
* =6.01 - отвечает за амплитуду краткосрочных изменений, показывает, что слабый непериодический тренд
* =1.89 - характерное расстояние корреляции ≈2 лага
* =24.99 - основной вклад в дисперсию (≈88% общей дисперсии), объясняет волнообразный характер графика
* =7.97 - период ≈8 лагов (совпадает с пиками в данных)
* =236.12 - очень медленное затухание колебаний, указывает на устойчивые циклические изменения

Таким образом экспоненциально-периодическая модель будет лучшей в нашем случае.

1. Найдём аналитический вид ковариационный функции.

Для нахождения аналитического вида ковариационной функции используем фундаментальную связь между семивариограммой и ковариационной функцией , где — семивариограмма, — ковариационная функция, — полная дисперсия процесса. И отсюда выразим ковариационную функцию .

При

= 0 =

Тогда .

Таким образом, окончательный вид ковариационной функции:

.

+

Для наших параметров:

+

1. Выводы о свойствах процесса:

Явно выраженная периодическая компонента, корреляция сохраняется на больших расстояниях благодаря периодичности. Процесс является узкополосным, что подтверждается отношением периода к времени затухания: 236.12/7.97 ≈ 30 (характерно для узкополосных процессов).

Листинг программы:

#1.Задание параметров

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

from scipy.stats import skew, kurtosis, norm

from scipy import stats

from scipy.stats import kendalltau

from scipy.optimize import curve\_fit

from sklearn.metrics import mean\_squared\_error

data = [

[1, 8.0, 5.5, 4.3, 17.2, 21.9, 22.8, 23.8, 15.6],

[2, 9.8, 8.5, 9.9, 14.4, 19.1, 21.1, 21.4, 16.4],

[3, 11.0, 8.7, 6.1, 10.4, 15.3, 17.2, 17.1, 10.8],

[4, 5.8, 3.0, 1.6, 13.3, 16.1, 18.5, 19.0, 12.2],

[5, 12.1, 9.3, 6.1, 16.4, 19.4, 21.2, 21.6, 14.9],

[6, 8.5, 9.4, 9.8, 16.0, 16.5, 14.4, 13.0, 9.7]

]

columns = ['День', '0.00', '3.00', '6.00', '9.00', '12.00', '15.00', '18.00', '21.00']

df = pd.DataFrame(data, columns=columns)

pd.set\_option('display.colheader\_justify', 'center')

pd.set\_option('display.unicode.east\_asian\_width', True)

styled\_df = df.style \

.set\_properties(\*\*{'border': '1px solid black'}) \

.set\_table\_styles([{

'selector': 'th',

'props': [('border', '1px solid black')]

}]) \

.format(precision=1, na\_rep='-')

display(styled\_df)

# 2. График временного ряда

all\_temps = []

for day\_data in data:

all\_temps.extend(day\_data[1:])

plt.figure(figsize=(12, 4))

plt.plot(all\_temps, marker='o', linestyle='-', color='blue')

plt.title('Температура за 6 дней (48 измерений)')

plt.xlabel('Номер измерения')

plt.ylabel('Температура (°C)')

plt.grid(True)

plt.xticks(np.arange(0, 48, 8), labels=[f'День {i}' for i in range(1, 7)])

plt.show()

# 3. Предварительный статистический анализ

# Описательные статистики

mean = np.mean(all\_temps)

variance = np.var(all\_temps, ddof=1)

std = np.std(all\_temps, ddof=1)

median = np.median(all\_temps)

q1 = np.percentile(all\_temps, 25)

q3 = np.percentile(all\_temps, 75)

skewness = skew(all\_temps)

kurt = kurtosis(all\_temps)

print("Описательные статистики:")

print(f"Среднее: {mean:.2f} °C")

print(f"Дисперсия: {variance:.2f} °C²")

print(f"Стандартное отклонение: {std:.2f} °C")

print(f"Медиана: {median:.2f} °C")

print(f"Первый квартиль (Q1): {q1:.2f} °C")

print(f"Третий квартиль (Q3): {q3:.2f} °C")

print(f"Асимметрия: {skewness:.2f}")

print(f"Эксцесс: {kurt:.2f}")

# Гистограмма с нормальным распределением

plt.figure(figsize=(10, 6))

n, bins, patches = plt.hist(all\_temps, bins=12, density=True, edgecolor='black', alpha=0.7, label='Гистограмма')

xmin, xmax = plt.xlim()

x = np.linspace(xmin, xmax, 100)

p = norm.pdf(x, mean, std)

plt.plot(x, p, 'r-', linewidth=2, label='Нормальное распределение')

plt.title('Гистограмма распределения температур с нормальной кривой')

plt.xlabel('Температура (°C)')

plt.ylabel('Плотность вероятности')

plt.legend()

plt.grid(True)

plt.show()

# Q-Q plot для проверки нормальности

plt.figure(figsize=(8, 8))

stats.probplot(all\_temps, dist="norm", plot=plt)

plt.title('Q-Q plot для проверки нормальности')

plt.grid(True)

plt.show()

# Проверка на нормальность (тест Шапиро-Уилка)

shapiro\_test = stats.shapiro(all\_temps)

print(f"\nТест Шапиро-Уилка: p-value = {shapiro\_test.pvalue:.4f}")

if shapiro\_test.pvalue > 0.05:

print("Не отвергаем гипотезу о нормальности (p > 0.05)")

else:

print("Отвергаем гипотезу о нормальности (p ≤ 0.05)")

# Проверка на тренд (тест Манна-Кендалла)

t = np.arange(len(all\_temps)) # Создаем временную ось

tau, p\_value = kendalltau(t, all\_temps)

print(f"\nТест Манна-Кендалла на тренд: p-value = {p\_value:.4f}")

if p\_value > 0.05:

print("Значимый тренд не обнаружен (p > 0.05)")

else:

print("Обнаружен значимый тренд (p ≤ 0.05)")

# 4. Оценки семивариограммы и ковариационной функции для пространственных данных

all\_temps = np.array(data)[:, 1:].flatten()

n = len(all\_temps)

max\_lag = int(2 \* n / 3) # Лаг до 2n/3

def semivariogram\_estimate(data, max\_lag=None):

n = len(data)

if max\_lag is None:

max\_lag = n - 1

gamma = np.zeros(max\_lag + 1)

for h in range(max\_lag + 1):

if h == 0:

gamma[h] = 0

else:

sum\_sq = 0

count = 0

for s in range(n - h):

sum\_sq += (data[s + h] - data[s])\*\*2

count += 1

gamma[h] = sum\_sq / (2 \* count) if count > 0 else 0

return gamma

def covariance\_estimate(data, max\_lag=None):

n = len(data)

if max\_lag is None:

max\_lag = n - 1

mu = np.mean(data)

R = np.zeros(max\_lag + 1)

for h in range(max\_lag + 1):

if h == 0:

R[h] = np.var(data, ddof=1)

else:

sum\_prod = 0

count = 0

for s in range(n - h):

sum\_prod += (data[s + h] - mu) \* (data[s] - mu)

count += 1

R[h] = sum\_prod / count if count > 0 else 0

return R

# Вычисляем оценки

gamma = semivariogram\_estimate(all\_temps, max\_lag)

R = covariance\_estimate(all\_temps, max\_lag)

lags = np.arange(max\_lag + 1)

plt.figure(figsize=(14, 6))

plt.subplot(1, 2, 1)

plt.plot(lags, gamma, marker='o', linestyle='-', color='b')

plt.title(r'Оценка семивариограммы')

plt.xlabel('Лаг (h)')

plt.ylabel(r'$\hat{\gamma}(h)$')

plt.grid(True)

plt.subplot(1, 2, 2)

plt.plot(lags, R, marker='o', linestyle='-', color='r')

plt.title(r'Оценка ковариационной функции')

plt.xlabel('Лаг (h)')

plt.ylabel(r'$\hat{R}(h)$')

plt.grid(True)

plt.tight\_layout()

plt.show()

# 5. Сравнение моделей семивариограммы

# Периодическая модель (Hole-Effect)

def hole\_effect(h, C0, C, a):

with np.errstate(divide='ignore', invalid='ignore'):

return C0 + C \* (1 - np.sin(np.pi\*h/a) / (np.pi\*h/a + 1e-10)) # Защита от деления на 0

# Смешанная экспоненциально-периодическая модель

def exp\_periodic(h, C0, C1, a1, C2, a2, a3):

return C0 + C1\*(1 - np.exp(-h/a1)) + C2\*(1 - np.cos(2\*np.pi\*h/a2)\*np.exp(-h/a3))

# Гауссова модель с периодичностью

def gaussian\_periodic(h, C0, C, a, p):

return C0 + C\*(1 - np.exp(-(h\*\*2)/(a\*\*2)) \* np.cos(2\*np.pi\*h/p))

# Функция для подгонки параметров

def fit\_model(model\_func, lags, gamma, p0):

try:

params, \_ = curve\_fit(model\_func, lags[1:], gamma[1:], p0=p0, maxfev=10000)

return params

except RuntimeError:

return None

# Начальные приближения для параметров

p0\_hole = [10, 45, 10] # C0, C, a

p0\_exp\_per = [10, 30, 5, 20, 10, 5] # C0, C1, a1, C2, a2, a3

p0\_gauss = [10, 45, 5, 10] # C0, C, a, p

params\_hole = fit\_model(hole\_effect, lags, gamma, p0\_hole)

params\_exp\_per = fit\_model(exp\_periodic, lags, gamma, p0\_exp\_per)

params\_gauss = fit\_model(gaussian\_periodic, lags, gamma, p0\_gauss)

plt.figure(figsize=(12, 8))

plt.scatter(lags, gamma, color='black', s=50, label='Эмпирическая семивариограмма')

if params\_hole is not None:

fit\_hole = hole\_effect(lags, \*params\_hole)

plt.plot(lags, fit\_hole, 'r-', linewidth=2, label=f'Hole-Effect')

if params\_exp\_per is not None:

fit\_exp\_per = exp\_periodic(lags, \*params\_exp\_per)

plt.plot(lags, fit\_exp\_per, 'g--', linewidth=2, label='Экспоненциально-периодическая')

if params\_gauss is not None:

fit\_gauss = gaussian\_periodic(lags, \*params\_gauss)

plt.plot(lags, fit\_gauss, 'b:', linewidth=2, label=f'Гауссова')

plt.title('Сравнение моделей семивариограммы', fontsize=14)

plt.xlabel('Лаг (h)', fontsize=12)

plt.ylabel('γ(h)', fontsize=12)

plt.legend(fontsize=12)

plt.grid(True, linestyle='--', alpha=0.7)

plt.xticks(np.arange(0, max\_lag+1, 4))

plt.show()

# Вывод параметров моделей

print("Параметры моделей:")

if params\_hole is not None:

print(f"\nHole-Effect: C0={params\_hole[0]:.2f}, C={params\_hole[1]:.2f}, a={params\_hole[2]:.2f}")

if params\_exp\_per is not None:

print(f"\nЭкспоненциально-периодическая:")

print(f"C0={params\_exp\_per[0]:.2f}, C1={params\_exp\_per[1]:.2f}, a1={params\_exp\_per[2]:.2f}")

print(f"C2={params\_exp\_per[3]:.2f}, a2={params\_exp\_per[4]:.2f}, a3={params\_exp\_per[5]:.2f}")

if params\_gauss is not None:

print(f"\nГауссова: C0={params\_gauss[0]:.2f}, C={params\_gauss[1]:.2f}, a={params\_gauss[2]:.2f}, p={params\_gauss[3]:.2f}")

# Сравнение по MSE

def calculate\_mse(model\_func, params, lags, gamma):

if params is None:

return np.inf

pred = model\_func(lags[1:], \*params)

return np.mean((pred - gamma[1:])\*\*2)

mse\_hole = calculate\_mse(hole\_effect, params\_hole, lags, gamma)

mse\_exp\_per = calculate\_mse(exp\_periodic, params\_exp\_per, lags, gamma)

mse\_gauss = calculate\_mse(gaussian\_periodic, params\_gauss, lags, gamma)

print("\nСравнение MSE:")

print(f"Hole-Effect: {mse\_hole:.2f}")

print(f"Экспоненциально-периодическая: {mse\_exp\_per:.2f}")

print(f"Гауссова: {mse\_gauss:.2f}")

best\_model = np.argmin([mse\_hole, mse\_exp\_per, mse\_gauss])

models = ["Hole-Effect", "Экспоненциально-периодическая", "Гауссова"]

print(f"\nЛучшая модель: {models[best\_model]} (MSE = {min(mse\_hole, mse\_exp\_per, mse\_gauss):.2f})")