**БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**ФАКУЛЬТЕТ ПРИКЛАДНОЙ МАТЕМАТИКИ И ИНФОРМАТИКИ**

**Кафедра теории вероятностей и математической статистики**

**ОТЧЕТ**

по лабораторной работе №3

«Подбор модели семивариограммы»

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 заданных значений мощности осадочного слоя в сантиметрах, измеренных в 20 точках, через каждые 100 м вдоль профиля, необходимо:

1. Нанести ряд данных на график; выполнить первичный визуальный анализ и сделать предположение о наличии тренда;
2. Сделать предварительный статистический анализ (вычислить описательные статистики: выборочное среднее, выборочную дисперсию, стандартное отклонение, минимум и максимум, коэффициент вариации; сделать вывод об однородности данных);
3. Проверить данные на наличие линейных трендов путем выполнения регрессионного анализа. При наличии тренда в исходных данных:

* Записать уравнение модели тренда;
* Вычислить коэффициент детерминации (величину достоверности аппроксимации) и сделать вывод;
* Добавить линию тренда на график исходных данных;
* Вычитая из исходных данных значения тренда, найти ряд остатков регрессии. Для ряда остатков построить график и сделать предварительный статистический анализ.

1. Для исходных данных и ряда остатков вычислить и представить графически оценки ковариационной функции и оценки семивариограммы для первых десяти шагов (лагов) h.

* Определить, оказывает ли влияние (и охарактеризовать его) присутствие тренда в исходных данных на вид оценок ковариационной функции и семивариограммы;
* По оценкам ковариационной функции и семивариограммы ряда остатков исследовать зависимость между наблюдениями на основе понятия интервала корреляции (ранга);

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

1. Визуальным методом подобрать модель семивариограммы для ряда остатков.

* Записать аналитический вид модели с указанием всех параметров. Объяснить выбор параметров;
* На одном рисунке представить графически оценку семивариограммы ряда остатков и предложенную модель семивариограммы.

**Исходные данные (алгоритм выполнения).**

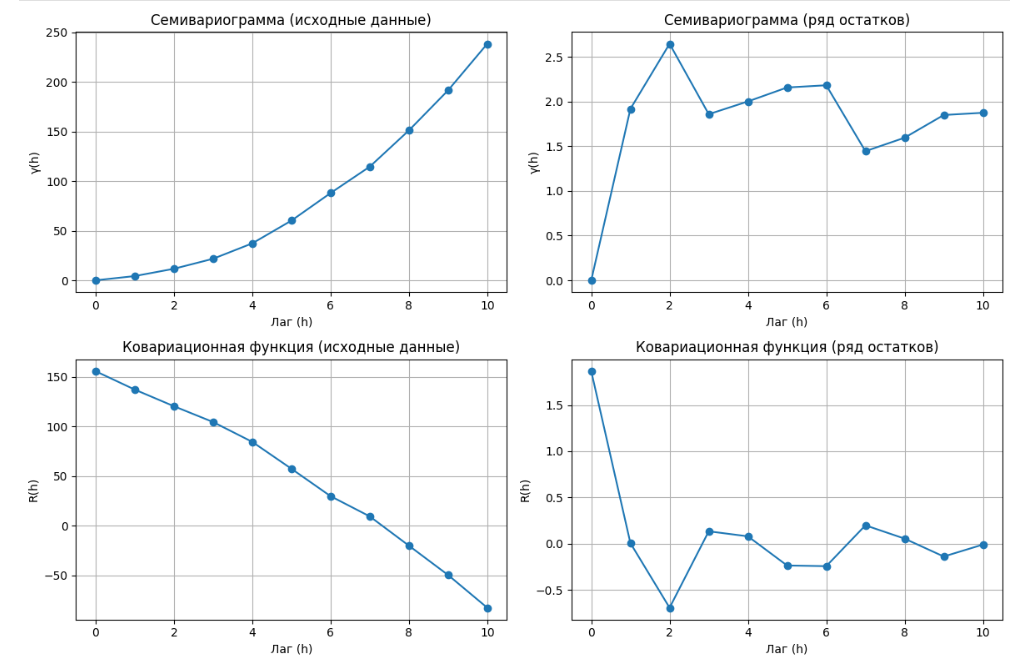
**Часть первая.**

Выполнение заданий 1-3 представлено в файле «Lab 3\_Var №\_3.xlsx».

**Вывод.** Выводы по заданиям 1-3 представлены в файле «Lab 3\_Var №\_3.xlsx».

4. График семивариограммы исходных данных с увеличением лага возрастает, что говорит о том, что с увеличением расстояния между точками дисперсия разностей значений становится больше. График ковариационной функции с увеличением лага убывает, что подтверждает сделанный вывод. Это говорит о наличии тренда в данных.

По графику семивариограммы и ковариационной функции для ряда остатков можно сказать, что интервал корреляции равен 1.

****

5. При выборе модели семивариограммы применялся визуальный метод. Значения быстро растут от h=1 до h=3, достигая максимума при h=2 (2.6), затем немного снижаются и стабилизируются. После h=3 значения колеблются вокруг ~2, имея волнообразный характер. Рассмотрим две модели: сферическую и смешанную (экспоненциальная + периодическая).

Параметры сферической модели:

= 0.0 (, = 1.96 (выборочная дисперсия), = 2.0 (после второго лага значения колеблются в пределах двойки)

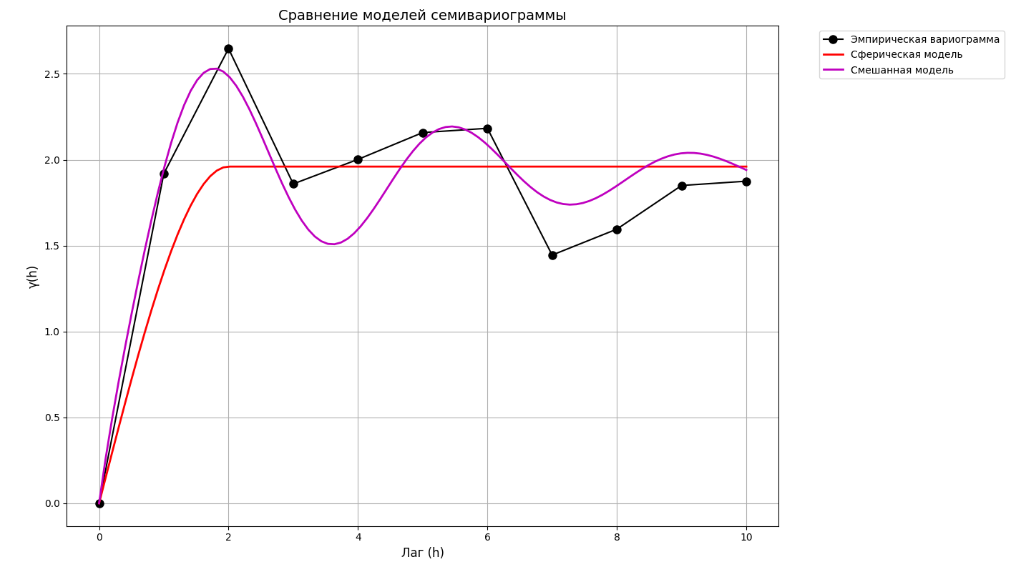
Сферическая модель семивариограммы:

Параметры смешанной модели:

= 0.0, = 0.99 (половина выборочной дисперсии), = 0.39 ( , = 0.93, = 3.68 (расстояние между пиками), = 4.48 (амплитуда колебаний уменьшается после h = 4)

Смешанная модель (Экспоненциальная + Периодическая):

Таким образом сферическая модель даёт простое и грубое приближение и не учитывает возможные колебания в данных. Смешанная модель семивариограммы лучше описывает ряд остатков, поскольку учитывает, как экспоненциальный рост дисперсии, так и возможные периодические колебания.

****

**Листинг программы**

#4. Оценка ковариационной функции и семивариограммы

import numpy as np

import matplotlib.pyplot as plt

original\_data = np.array([

6.7, 9.2, 11.9, 13.1, 15.9, 18.3, 20.7, 23.0, 25.3, 26.2,

29.4, 32.1, 35.3, 32.6, 39.7, 42.9, 41.2, 42.3, 44.4, 48.0

])

residuals = np.array([

-1.21, -0.71, -0.01, -0.81, -0.01, 0.39, 0.79, 1.09, 1.39, 0.29,

-0.51, 0.19, 1.39, -3.31, 1.79, 2.99, -0.71, -1.61, -1.51, 0.09

])

n = len(original\_data)

def semivariogram(data, h\_max):

gamma = np.zeros(h\_max + 1)

for h in range(h\_max + 1):

sum\_diff = 0.0

count = 0

for s in range(n - h):

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

count += 1

if count > 0:

gamma[h] = sum\_diff / (2 \* count)

return gamma

def covariance\_function(data, h\_max):

R = np.zeros(h\_max + 1)

mu = np.mean(data)

for h in range(h\_max + 1):

sum\_cov = 0.0

count = 0

for s in range(n - h):

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

count += 1

if count > 0:

R[h] = sum\_cov / count

return R

h\_max = 10

gamma\_original = semivariogram(original\_data, h\_max)

gamma\_residuals = semivariogram(residuals, h\_max)

R\_original = covariance\_function(original\_data, h\_max)

R\_residuals = covariance\_function(residuals, h\_max)

lags = np.arange(h\_max + 1)

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

# Графики семивариограммы

plt.subplot(2, 2, 1)

plt.plot(lags, gamma\_original, 'o-', label='Исходные данные')

plt.title('Семивариограмма (исходные данные)')

plt.xlabel('Лаг (h)')

plt.ylabel('γ(h)')

plt.grid()

plt.subplot(2, 2, 2)

plt.plot(lags, gamma\_residuals, 'o-', label='Ряд остатков')

plt.title('Семивариограмма (ряд остатков)')

plt.xlabel('Лаг (h)')

plt.ylabel('γ(h)')

plt.grid()

# Графики ковариационной функции

plt.subplot(2, 2, 3)

plt.plot(lags, R\_original, 'o-', label='Исходные данные')

plt.title('Ковариационная функция (исходные данные)')

plt.xlabel('Лаг (h)')

plt.ylabel('R(h)')

plt.grid()

plt.subplot(2, 2, 4)

plt.plot(lags, R\_residuals, 'o-', label='Ряд остатков')

plt.title('Ковариационная функция (ряд остатков)')

plt.xlabel('Лаг (h)')

plt.ylabel('R(h)')

plt.grid()

plt.tight\_layout()

plt.show()

def correlation\_interval(covariance):

h\_corr = 0

while h\_corr < len(covariance) and covariance[h\_corr] > 0:

h\_corr += 1

return h\_corr - 1

h\_corr\_residuals = correlation\_interval(R\_residuals)

print(h\_corr\_residuals)

#5. Сравнение моделей

spherical\_params = {

'nugget': 0.0,

'sill': 1.96,

'range\_': 2.0

}

mixed\_model\_params = {

'C0': 0.0,

'C1': 0.99,

'a1': 0.39,

'C2': 0.93,

'a2': 3.68,

'a3': 4.48

}

def spherical(h, nugget, sill, range\_):

return nugget + (sill - nugget) \* ((3\*h)/(2\*range\_) - (h\*\*3)/(2\*range\_\*\*3)) \* (h <= range\_) + (sill - nugget) \* (h > range\_)

def mixed\_model(h, C0, C1, a1, C2, a2, a3):

exp\_term = C1 \* (1 - np.exp(-h/a1))

periodic\_term = C2 \* (1 - np.cos(2\*np.pi\*h/a2) \* np.exp(-h/a3))

return C0 + exp\_term + periodic\_term

h\_plot = np.linspace(0, h\_max, 100)

spherical\_values = spherical(h\_plot, \*\*spherical\_params)

mixed\_values = mixed\_model(h\_plot, \*\*mixed\_model\_params)

plt.figure(figsize=(14, 8))

plt.plot(lags, gamma\_residuals, 'ko-', label='Эмпирическая вариограмма', markersize=8)

# Модельные вариограммы

plt.plot(h\_plot, spherical\_values, 'r-', linewidth=2,

label=f'Сферическая модель')

plt.plot(h\_plot, mixed\_values, 'm-', linewidth=2,

label=f'Смешанная модель')

plt.xlabel('Лаг (h)', fontsize=12)

plt.ylabel('γ(h)', fontsize=12)

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

plt.legend(fontsize=10, bbox\_to\_anchor=(1.05, 1), loc='upper left')

plt.grid(True)

plt.tight\_layout()

plt.show()

print("Параметры сферической модели:")

print(spherical\_params)

print("\nПараметры смешанной модели:")

print(mixed\_model\_params)

**Часть вторая.**

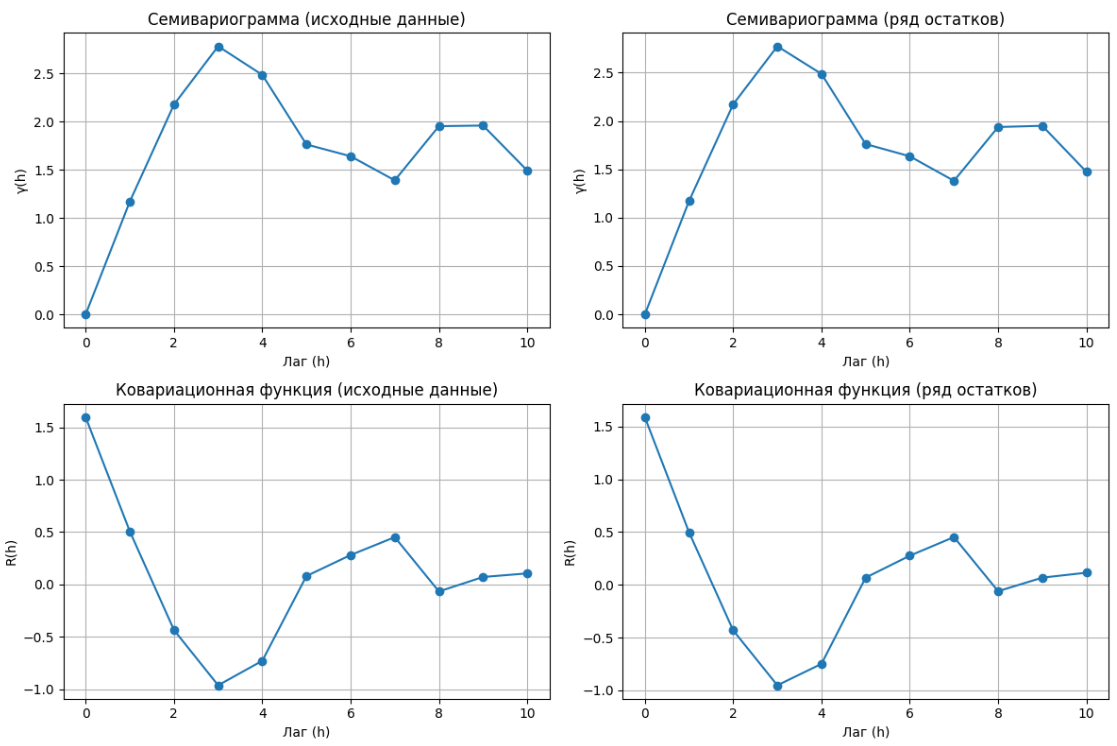
**Постановка задачи:** Удалить периодичность из серии 7 и посмотреть, какую форму примет оценка семивариограммы.

**Исходные данные (алгоритм выполнения).**

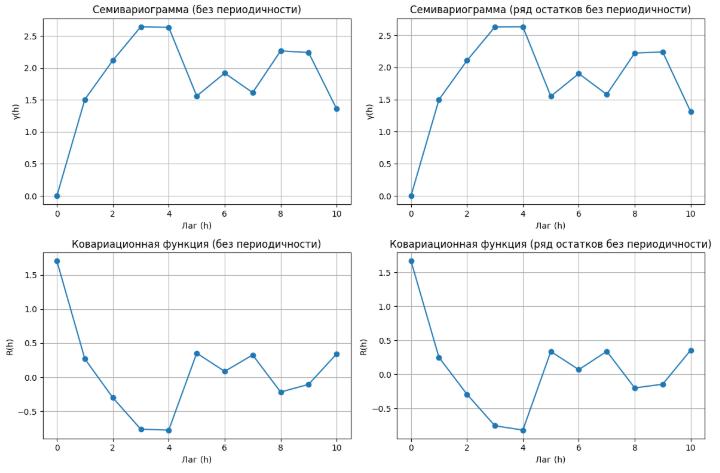
Выполнение заданий 1-3 для двух случаев и выводы представлены в файле «Lab 3\_Var №\_3.xlsx».

4. Семивариограмма исходных данных быстро возрастает до 2.5 при лаге 2, после чего рост замедляется. Это указывает на наличие пространственной зависимости в данных на малых расстояниях. Ковариационная функция убывает с увеличением лага, что подтверждает сделанный вывод. Это указывает на наличие периодичности в данных. Тренд практически незначим.

По графику семивариограммы и ковариационной функции для ряда остатков можно сказать, что интервал корреляции равен 1.



Удалим периодичность.



Семивариограмма данных без периодичности медленно возрастает с увеличением лага, не демонстрируя резких скачков или периодичности. Это указывает на отсутствие выраженной пространственной зависимости на малых расстояниях. Ковариационная функция также убывает плавно, без четких периодических колебаний, что подтверждает отсутствие периодической структуры в данных. Тренд все еще присутствует, но менее выражен, чем в исходных данных.

По графику семивариограммы и ковариационной функции для ряда остатков можно сказать, что интервал корреляции равен 1.

Таким образом: Удаление периодичности привело к более гладким и менее структурированным семивариограммам и ковариационным функциям, что указывает на отсутствие сильной периодической компоненты в данных. Оценки семивариограмм для исходных данных и данных без периодичности различаются, что подтверждает значительное влияние периодической компоненты на структуру данных.

Листинг программы:

#для первоначального слоя 7

import numpy as np

import matplotlib.pyplot as plt

original\_data = np.array([

5.2, 5.4, 6.1, 3.1, 5.4, 6.4, 7.2, 6.5, 4.5, 3.3,

2.2, 5.5, 6.2, 6.6, 5.5, 4.3, 6.3, 5.5, 4.2, 4.8

])

residuals = np.array([

-0.15, 0.06, 0.78, -2.21, 0.11, 1.12, 1.93, 1.25, -0.74, -1.92,

-2.99, 0.32, 1.03, 1.45, 0.36, -0.83, 1.19, 0.40, -0.89, -0.27

])

n = len(original\_data)

def semivariogram(data, h\_max):

gamma = np.zeros(h\_max + 1)

for h in range(h\_max + 1):

sum\_diff = 0.0

count = 0

for s in range(n - h):

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

count += 1

if count > 0:

gamma[h] = sum\_diff / (2 \* count)

return gamma

def covariance\_function(data, h\_max):

R = np.zeros(h\_max + 1)

mu = np.mean(data)

for h in range(h\_max + 1):

sum\_cov = 0.0

count = 0

for s in range(n - h):

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

count += 1

if count > 0:

R[h] = sum\_cov / count

return R

h\_max = 10

gamma\_original = semivariogram(original\_data, h\_max)

gamma\_residuals = semivariogram(residuals, h\_max)

R\_original = covariance\_function(original\_data, h\_max)

R\_residuals = covariance\_function(residuals, h\_max)

lags = np.arange(h\_max + 1)

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

# Графики семивариограммы

plt.subplot(2, 2, 1)

plt.plot(lags, gamma\_original, 'o-', label='Исходные данные')

plt.title('Семивариограмма (исходные данные)')

plt.xlabel('Лаг (h)')

plt.ylabel('γ(h)')

plt.grid()

plt.subplot(2, 2, 2)

plt.plot(lags, gamma\_residuals, 'o-', label='Ряд остатков')

plt.title('Семивариограмма (ряд остатков)')

plt.xlabel('Лаг (h)')

plt.ylabel('γ(h)')

plt.grid()

# Графики ковариационной функции

plt.subplot(2, 2, 3)

plt.plot(lags, R\_original, 'o-', label='Исходные данные')

plt.title('Ковариационная функция (исходные данные)')

plt.xlabel('Лаг (h)')

plt.ylabel('R(h)')

plt.grid()

plt.subplot(2, 2, 4)

plt.plot(lags, R\_residuals, 'o-', label='Ряд остатков')

plt.title('Ковариационная функция (ряд остатков)')

plt.xlabel('Лаг (h)')

plt.ylabel('R(h)')

plt.grid()

plt.tight\_layout()

plt.show()

def correlation\_interval(covariance):

h\_corr = 0

while h\_corr < len(covariance) and covariance[h\_corr] > 0:

h\_corr += 1

return h\_corr - 1

h\_corr\_residuals = correlation\_interval(R\_residuals)

print(h\_corr\_residuals)

#для слоя 7 без периода

import numpy as np

import matplotlib.pyplot as plt

original\_data = np.array([

5.43, 5.88, 6.05, 3.21, 5.88, 5.93, 7.40, 5.97, 3.96, 3.70,

1.91, 6.04, 5.98, 6.05, 6.04, 3.91, 5.94, 6.04, 3.95, 4.41

])

residuals = np.array([

-0.05, 0.43, 0.63, -2.18, 0.52, 0.60, 2.10, 0.69, -1.29, -1.51,

-3.24, 0.91, 0.89, 0.99, 1.00, -1.09, 0.96, 1.09, -0.97, -0.48

])

n = len(original\_data)

def semivariogram(data, h\_max):

gamma = np.zeros(h\_max + 1)

for h in range(h\_max + 1):

sum\_diff = 0.0

count = 0

for s in range(n - h):

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

count += 1

if count > 0:

gamma[h] = sum\_diff / (2 \* count)

return gamma

def covariance\_function(data, h\_max):

R = np.zeros(h\_max + 1)

mu = np.mean(data)

for h in range(h\_max + 1):

sum\_cov = 0.0

count = 0

for s in range(n - h):

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

count += 1

if count > 0:

R[h] = sum\_cov / count

return R

h\_max = 10

gamma\_original = semivariogram(original\_data, h\_max)

gamma\_residuals = semivariogram(residuals, h\_max)

R\_original = covariance\_function(original\_data, h\_max)

R\_residuals = covariance\_function(residuals, h\_max)

lags = np.arange(h\_max + 1)

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

# Графики семивариограммы

plt.subplot(2, 2, 1)

plt.plot(lags, gamma\_original, 'o-', label='Исходные данные')

plt.title('Семивариограмма (без периодичности)')

plt.xlabel('Лаг (h)')

plt.ylabel('γ(h)')

plt.grid()

plt.subplot(2, 2, 2)

plt.plot(lags, gamma\_residuals, 'o-', label='Ряд остатков')

plt.title('Семивариограмма (ряд остатков без периодичности)')

plt.xlabel('Лаг (h)')

plt.ylabel('γ(h)')

plt.grid()

# Графики ковариационной функции

plt.subplot(2, 2, 3)

plt.plot(lags, R\_original, 'o-', label='Исходные данные')

plt.title('Ковариационная функция (без периодичности)')

plt.xlabel('Лаг (h)')

plt.ylabel('R(h)')

plt.grid()

plt.subplot(2, 2, 4)

plt.plot(lags, R\_residuals, 'o-', label='Ряд остатков')

plt.title('Ковариационная функция (ряд остатков без периодичности)')

plt.xlabel('Лаг (h)')

plt.ylabel('R(h)')

plt.grid()

plt.tight\_layout()

plt.show()

def correlation\_interval(covariance):

h\_corr = 0

while h\_corr < len(covariance) and covariance[h\_corr] > 0:

h\_corr += 1

return h\_corr - 1

h\_corr\_residuals = correlation\_interval(R\_residuals)

print(h\_corr\_residuals)