

# Лабораторная работа № 7

Введение в работу с данными

---

Герра Гарсия Паола Валентина

Российский университет дружбы народов, Москва, Россия

## Информация

---

- Герра Гарсия Паола Валентина
- студентка
- Российский университет дружбы народов
- 1032225472@pfur.ru

## Цель работы

---

Основной целью работы является освоение специализированных пакетов Julia для обработки данных.

## Задание

---

1. Используя JupyterLab, повторите примеры. При этом дополните графики обозначениями осей координат, легендой с названиями траекторий, названиями графиков и т.п.
2. Выполните задания для самостоятельной работы.

# Выполнение лабораторной работы

The screenshot shows a Jupyter Notebook interface running on a Linux desktop. The browser window title is "jupyter Лаб7\_статистический\_анализ.ipyb". The notebook has two cells:

In [1]:

```
# Обработка данных
using Pkg
Pkg.add("DataFrames")
# Установка пакетов
for p in ["CSV", "DataFrames", "RDatasets", "FileIO"]
    Pkg.add(p)
end

using CSV, DataFrames, DelimitedFiles
```

This cell outputs a message about successful precompilation:

```
8 dependencies successfully precompiled in 99 seconds. 403 already precompiled.
8 dependencies were added to `Project.toml`:
For a report of the errors see `Julia-err`. To retry use `#(pkgs) precompile`.
```

In [2]:

```
# Считывание данных и их запись в структуру DataFrame
df = CSV.File("programminglanguages.csv") |> DataFrame
```

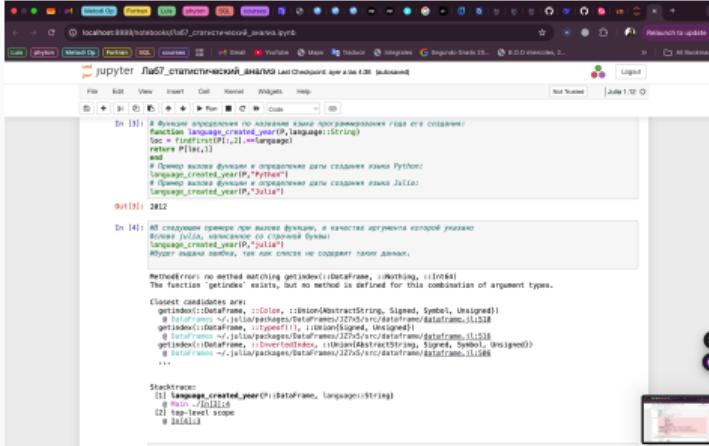
This cell outputs the structure of the DataFrame:

```
Out[2]: 70 rows × 2 columns
```

year	language
Int64	String{1}

Рис. 1: Считывание данных

# Выполнение лабораторной работы



The screenshot shows a Jupyter Notebook interface running on a local host. The notebook has two cells:

- Cell 13:** Contains Python code defining a function `language_created_year` that takes a string argument `language`. It prints the year 2012 and returns a tuple containing the year and language name.
- Cell 14:** Contains Python code attempting to call the `language_created_year` function with the argument "Julia". This results in a runtime error because the function expects a string argument, but "Julia" is a DataFrame object.

The error message from Cell 14 is:

```
MethodError: no method matching getindex!(DataFrame, Int64, Int64)
The function getindex! exists, but no method is defined for this combination of argument types.
Closest candidates are:
getindex!(DataFrame, ColRef, Union{AbstractString, Symbol, Unsigned})
#巴拉巴拉...  
getindex!(DataFrame, Int64), Union{Signed, Unsigned}  
getindex!(DataFrame, Int64, Int64), Union{AbstractString, Symbol, Unsigned}  
getindex!(DataFrame, Int64, Int64), Union{AbstractString, Symbol, Unsigned}  
#巴拉巴拉...
```

Stacktrace:

```
[1] language_created_year!(::DataFrame, ::String)
#巴拉巴拉...
[2] top-level scope
#巴拉巴拉...
```

Рис. 2: Запись данных в файл

# Выполнение лабораторной работы

The screenshot shows a Jupyter Notebook interface running on a local host. The notebook has tabs for 'Lab', 'Python', 'Kernel', and 'SQL'. The current tab is 'Python'. The code cell (In 181) contains Python code for reading a CSV file and writing its contents to a text file. The output cell (Out[17]) displays the contents of the 'programming\_languages\_data2.txt' file, which lists various programming languages and their years of creation. The code cell (In 191) shows the command to read the CSV file. The output cell (Out[191]) shows the first few lines of the CSV data. The code cell (In 190) shows the creation of a dictionary from the data. The output cell (Out[190]) shows the created dictionary. The code cell (In 111) shows the creation of a second dictionary. The output cell (Out[111]) shows the created second dictionary.

```
In [181]: # Пример записи данных в текстовый файл с разделителем ","
#writefile("programming_languages_data2.csv", Tx)
# Пример записи данных в текстовый файл с разделителем "-"
#writefile("programming_languages_data2.txt", Tx, "-")
Out[17]: "programming_languages_data2.csv"
In [191]: # Построение сгруппированных данных с указанием разделяителя
P_new_dflim = readfile("programming_languages_data2.txt", "-", 1)
Out[191]: 7442 DataFrame:
   *year*    *language*
1951    "Assembly Language"
1952    "Fortran"
1954    "TPL"
1955    "FLOW-MATIC"
1957    "COBOL"
1958    "LISP"
1958    "Algol 68"
1959    "PASCAL"
1959    "FACT"
1959    "COCOMO"
1960    "Logo"
1962    "APL"
2003    "Scala"
2005    "F#"
2006    "PowerShell"
2007    "Clojure"
In [190]: # Анализирующая команда
dict = Dict(Integer,Vector[String])()
Out[190]: Dict{Integer, Vector[String}}()
In [111]: # Анализирующая команда
dict2 = Dict{String, Vector[String]}()
Out[111]: Dict{String, Vector[String]}()
```

Рис. 3: Словари

## Выполнение лабораторной работы

Рис. 4: DataFrames

# Выполнение лабораторной работы

The screenshot shows a Jupyter Notebook interface running on a local host. The title bar indicates it's a Julia environment. The notebook has three cells:

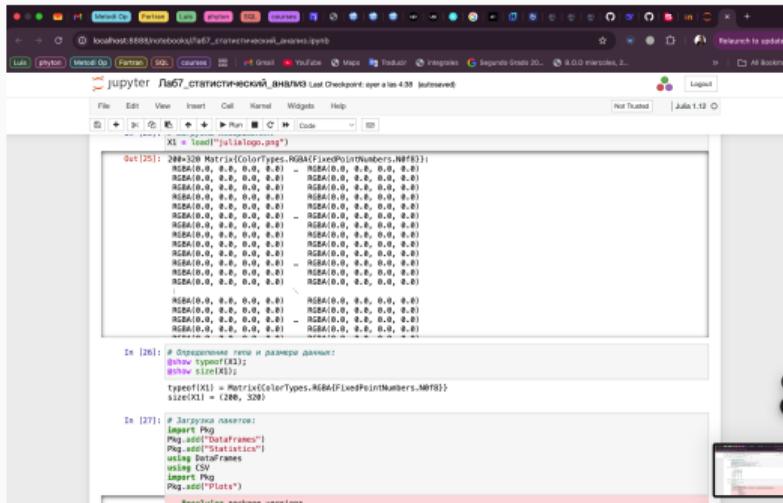
- In [22]:** Contains code to calculate the ratio of price to calories for four items: apple, cucumber, tomato, and banana. The output is a DataFrame with columns 'item', 'calories', and 'price'.
- Out [22]:** Shows the resulting DataFrame:

item	calories	price
apple	100	0.08
cucumber	47	1.6
tomato	28	0.8
banana	106	0.8

- In [23]:** A comment indicating the use of the FileIO package.
- In [24]:** A comment indicating the use of the ImageIO package, followed by code to import the package and add 'ImageIO' to the path.

Рис. 5: RDatasets

# Выполнение лабораторной работы



The screenshot shows a Jupyter Notebook interface running on a Mac OS X desktop. The browser window title is "jupyter Лаб7\_статистический\_анализ MissingValues.ipynb". The notebook has three cells:

- In [25]:** Displays the output of a command that generates a 200x200 matrix of RGB values. The output is a large block of text showing a grid of color values.
- In [26]:** Displays the input code for checking the type and size of the matrix X3. The code includes `#show type(X3)`, `#show size(X3)`, `typeof(X3) = Matrix{ColorTypes,RGB{FixedPointNumbers,M8}}`, and `size(X3) = (200, 200)`.
- In [27]:** Displays the input code for loading packages. The code includes `using Pkg`, `Pkg.add("DataFrames")`, `using DataFrames`, `using CSV`, `using Plots`, and `Pkg.add("Plots")`. A small error message "RunTimeError: unknown variable" is visible at the bottom of this cell.

Рис. 6: Работа с переменными отсутствующего типа (MissingValues)

# Выполнение лабораторной работы

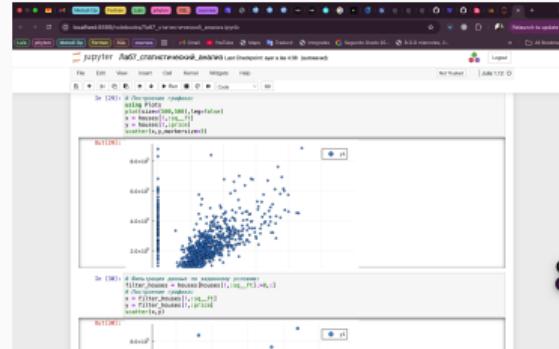
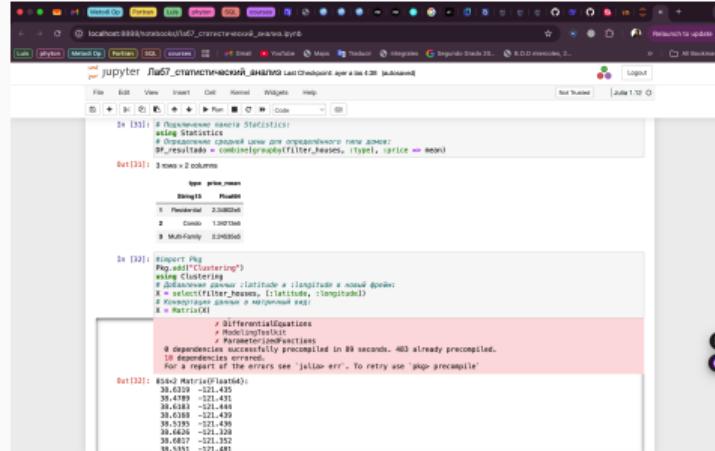


Рис. 7: FileIO

# Выполнение лабораторной работы



The screenshot shows a Jupyter Notebook interface running on a local server at `localhost:8888/notebooks/167_статистический_анализ.ipynb`. The notebook has one cell executed, with the output displayed below it.

```
In [31]: # Помеченные данные Statistics
# Использование спиральной кривой для определения новых домов:
DF_results = pd.concat([panda[filter_houses], (type, nprice - mean)])
Out[31]: 3 rows x 2 columns
```

Type	price_mean
Single-Family	2.382024
Condo	1.307194
Mult-Family	3.249565

```
In [32]: import Pg
Pg.add("Clustering")
using Clustering
# Данные о домах по координатам в новый файл:
# filter_houses, [Latitude, Longitude]
# Конвертируем данные в матричный вид
# X = Matrix(X)
```

```
In [33]: # Импортируем библиотеку
# DifferentialEquations
# ModelingToolkit
# ParameterizedFunctions
# dependencies already precompiled in 89 seconds. 403 already precompiled.
# 10 dependencies errors
# For more see the errors see 'julia> err'. To retry use 'julia> precompile'
Out[32]: # 10 dependencies errors
38.6319 -121.435
38.6319 -121.435
38.6319 -121.435
38.6319 -121.435
38.6319 -121.435
38.5335 -121.436
38.6319 -121.435
38.6319 -121.435
38.6319 -121.435
38.6319 -121.435
38.6319 -121.435
```

Рис. 8: Кластеризация данных. Метод k-средних

# Выполнение лабораторной работы

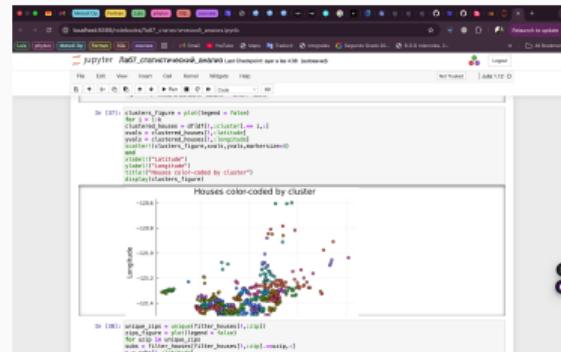


Рис. 9: Кластеризация данных. Метод k-средних

# Выполнение лабораторной работы

The screenshot shows a Jupyter Notebook interface running on a Mac OS X desktop. The browser window title is "jupyter Лаб7\_статистический\_анализ Notebook Checkpoint saved at 4:38 [untrusted]". The notebook has two cells:

```
In [59]: #7.4.2. Регрессия (метод наименьших квадратов в случае линейной регрессии)
#Файл 1
#Генерация данных
N=1000
X = rand(N, 3)
a0 = rand(3)
y = X*a0 + 0.1 * rand(N)
X_design = hcat(ones(N), X)
a_model = X_design\y
println("Истинные коэффициенты:", a0)
println("оценки коэффициентов:", a_model[1])
истинные коэффициенты: [6.169948862529427, 0.3775688417819259, 0.585537849258362]
оценки коэффициентов: [-6.0035441361874373]
```

```
In [60]: #add MultivariateStats
Pkg.add("MultivariateStats")
using MultivariateStats
#Adress para MultivariateStats
#https://github.com/JuliaStats/MultivariateStats.jl
Y_RS = Y
a_multivariate = llsq(Matrix(X), y, biwisefalse)
println("оценки MultivariateStats:", a_multivariate)
```

A red box highlights the error message from cell [60]:

```
Resolving package versions...
Project No packages added to or removed from `~/.julia/environments/v1.12/Project.toml'
Manifest No packages added to or removed from `~/.julia/environments/v1.12/Manifest.toml'
#Precompiling
  x LinearSolve
  x Groebner
  x Interpolations
  x OrdinaryDiffEq
  x DelayDiffEq
  x StochasticDiffEq
  x Compat
```

Рис. 10: Кластеризация данных. Метод k ближайших соседей

## Выполнение лабораторной работы

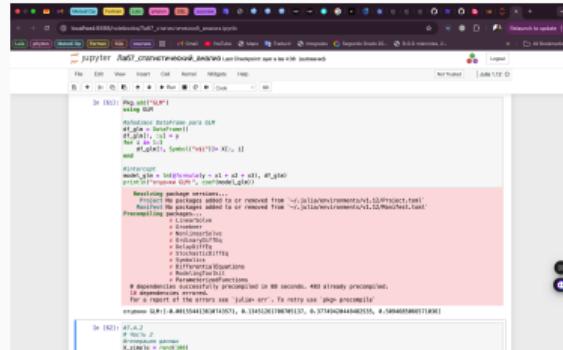


Рис. 11: Обработка данных. Метод главных компонент

# Выполнение лабораторной работы

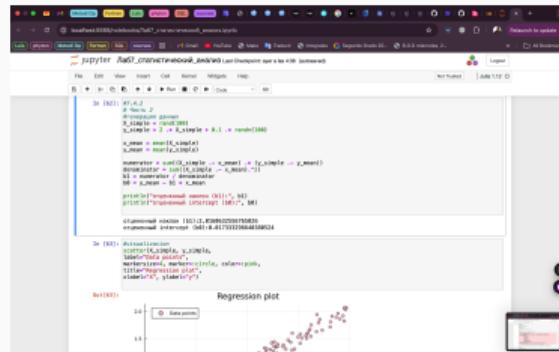


Рис. 12: Обработка данных. Линейная регрессия

# Выполнение лабораторной работы

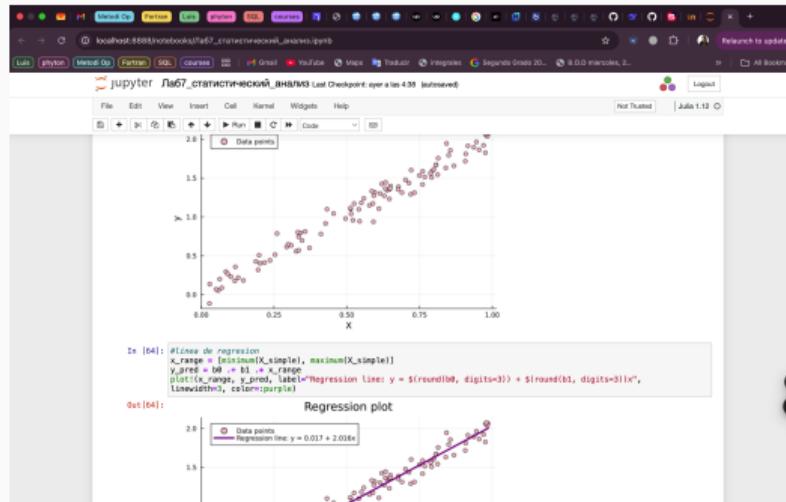
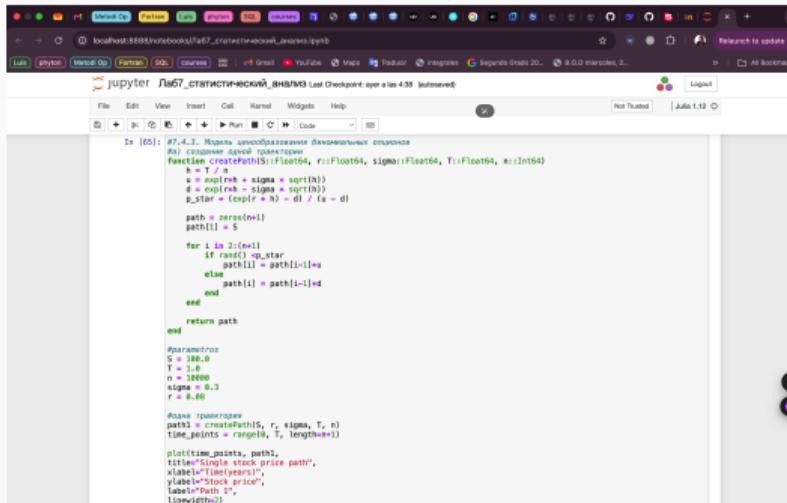


Рис. 13: Обработка данных. Линейная регрессия

# Выполнение лабораторной работы



The screenshot shows a Jupyter Notebook interface running in a web browser. The title bar indicates the notebook is titled "jupyter\_Лаб7\_статистический\_анализ.ipynb". The code cell contains the following Python script:

```
In [65]: #7.4.3. Модель ценообразования биномиальных опционов
# со временем ячейкой траектории
def createPath(S0:float64, r:float64, sigma:float64, T:float64, n:Int64):
    path = np.zeros(n+1)
    path[0] = S0
    u = exp(r*T + sigma * sqrt(T))
    d = exp(r*T - sigma * sqrt(T))
    p_star = (exp(r*T) - d) / (u - d)

    for i in range(1, n+1):
        if random() < p_star:
            path[i] = path[i-1]*u
        else:
            path[i] = path[i-1]*d
    return path

#параметры
S = 300.0
r = 0.05
n = 30000
sigma = 0.3
T = 0.05

#сама траектория
path1 = createPath(S, r, sigma, T, n)
print(path1[:10], path1[-10:])

plotTime_price(path1,
               title="Stock price path",
               xlabel="Time(years)",
               ylabel="Stock price",
               label="Path 1",
               lineStyle="--")
```

Рис. 14: Кластеризация

# Выполнение лабораторной работы

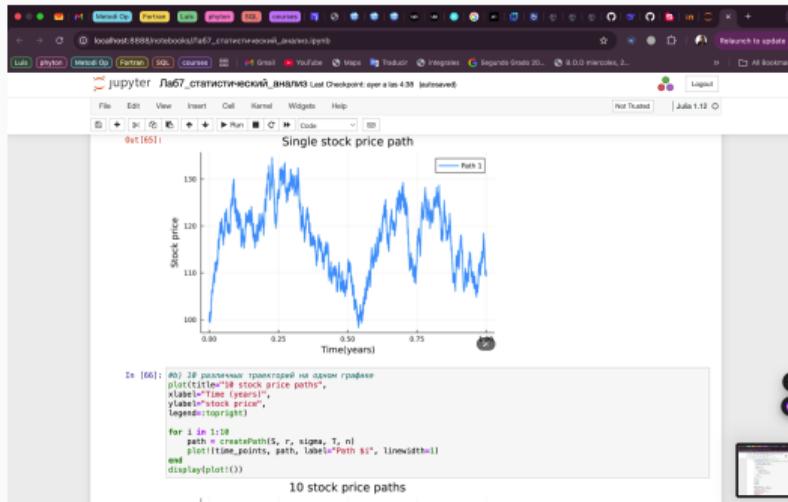


Рис. 15: Кластеризация

# Выполнение лабораторной работы

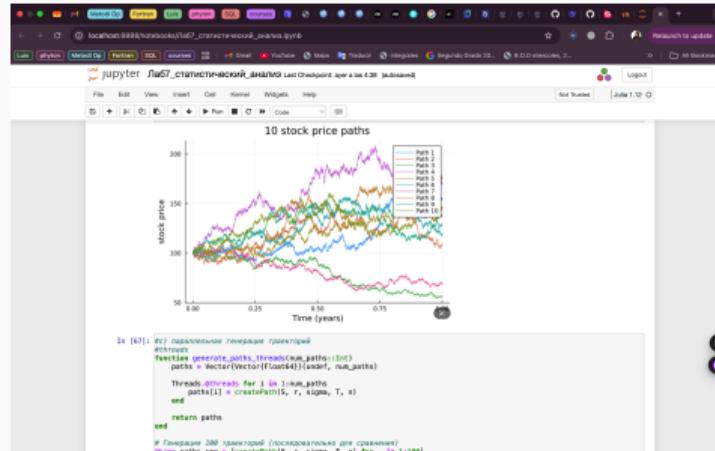
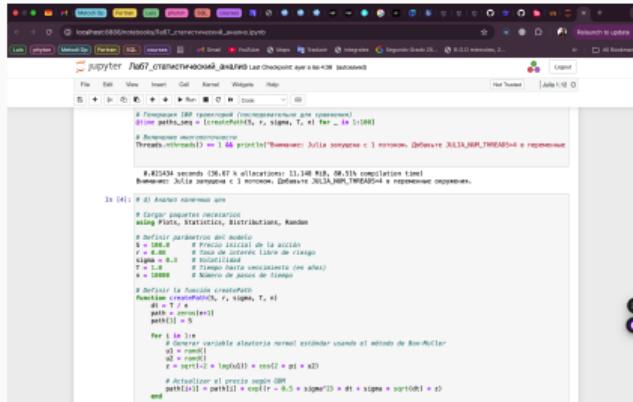


Рис. 16: Регрессия

# Выполнение лабораторной работы



The screenshot shows a Jupyter Notebook interface with a single cell containing Python code. The code is for a regression analysis, specifically a Poisson regression model. It includes importing libraries, defining parameters, and implementing a loop to calculate predictions. A status message at the top indicates the execution time and completion.

```
# Funcion para calcular los errores
def error(p, r):
    return np.sqrt(np.sum((p - r) ** 2))

# Funcion para calcular la probabilidad de que una variable aleatoria normal esténdar cuando el método de Box-Muller
def normal():
    u1 = random()
    u2 = random()
    r = np.sqrt(-2 * np.log(u1)) * np.cos(2 * np.pi * u2)
    return r

# Actualizar el vector regresión
def update(p, r, sigma):
    p[0] = p[0] + r[0] * sigma
    for i in range(1, len(r)):
        p[i] = p[i] + r[i] * sigma
    return p

# Funcion para calcular la probabilidad de que una variable aleatoria normal esténdar cuando el método de Box-Muller
def normal():
    u1 = random()
    u2 = random()
    r = np.sqrt(-2 * np.log(u1)) * np.cos(2 * np.pi * u2)
    return r

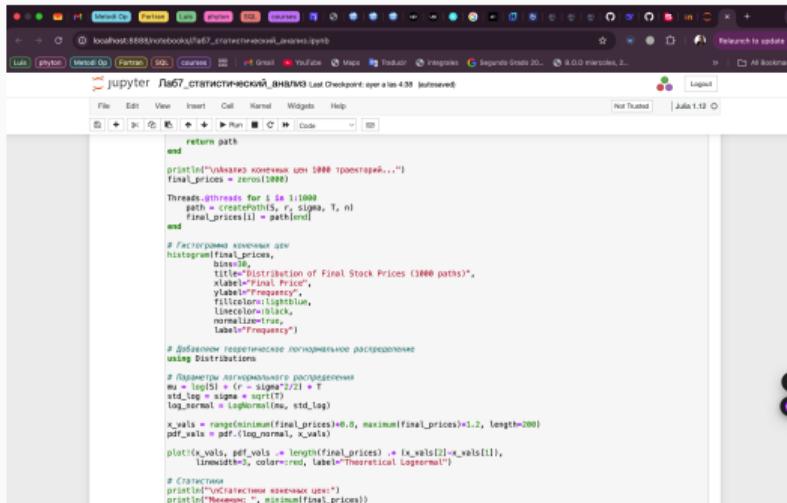
# Actualizar el vector regresión
def update(p, r, sigma):
    p[0] = p[0] + r[0] * sigma
    for i in range(1, len(r)):
        p[i] = p[i] + r[i] * sigma
    return p

# Cargar paquetes necesarios
using Plots, Statistics, Distributions, Random

# Definir parámetros del modelo
k = 1000 # Número de observaciones de la variable de respuesta
r = 4.00 # Tasa de ocurrencia libre de riesgo
lambda = 0.3 # Tiempo hasta visualizarse (en años)
T = 1.0 # Número de pasos de tiempo
n = 10000 # Número de pasos de tiempo

# Definir la función de logaritmo natural
function creacionRho(k, r, sigma, T, e)
    dt = T / e
    p[0] = normal()
    p[1] = 0
    p[2] = 0
    p[3] = 0
    p[4] = 0
    p[5] = 0
    p[6] = 0
    p[7] = 0
    p[8] = 0
    p[9] = 0
    p[10] = 0
    p[11] = 0
    p[12] = 0
    p[13] = 0
    p[14] = 0
    p[15] = 0
    p[16] = 0
    p[17] = 0
    p[18] = 0
    p[19] = 0
    p[20] = 0
    p[21] = 0
    p[22] = 0
    p[23] = 0
    p[24] = 0
    p[25] = 0
    p[26] = 0
    p[27] = 0
    p[28] = 0
    p[29] = 0
    p[30] = 0
    p[31] = 0
    p[32] = 0
    p[33] = 0
    p[34] = 0
    p[35] = 0
    p[36] = 0
    p[37] = 0
    p[38] = 0
    p[39] = 0
    p[40] = 0
    p[41] = 0
    p[42] = 0
    p[43] = 0
    p[44] = 0
    p[45] = 0
    p[46] = 0
    p[47] = 0
    p[48] = 0
    p[49] = 0
    p[50] = 0
    p[51] = 0
    p[52] = 0
    p[53] = 0
    p[54] = 0
    p[55] = 0
    p[56] = 0
    p[57] = 0
    p[58] = 0
    p[59] = 0
    p[60] = 0
    p[61] = 0
    p[62] = 0
    p[63] = 0
    p[64] = 0
    p[65] = 0
    p[66] = 0
    p[67] = 0
    p[68] = 0
    p[69] = 0
    p[70] = 0
    p[71] = 0
    p[72] = 0
    p[73] = 0
    p[74] = 0
    p[75] = 0
    p[76] = 0
    p[77] = 0
    p[78] = 0
    p[79] = 0
    p[80] = 0
    p[81] = 0
    p[82] = 0
    p[83] = 0
    p[84] = 0
    p[85] = 0
    p[86] = 0
    p[87] = 0
    p[88] = 0
    p[89] = 0
    p[90] = 0
    p[91] = 0
    p[92] = 0
    p[93] = 0
    p[94] = 0
    p[95] = 0
    p[96] = 0
    p[97] = 0
    p[98] = 0
    p[99] = 0
    p[100] = 0
    p[101] = 0
    p[102] = 0
    p[103] = 0
    p[104] = 0
    p[105] = 0
    p[106] = 0
    p[107] = 0
    p[108] = 0
    p[109] = 0
    p[110] = 0
    p[111] = 0
    p[112] = 0
    p[113] = 0
    p[114] = 0
    p[115] = 0
    p[116] = 0
    p[117] = 0
    p[118] = 0
    p[119] = 0
    p[120] = 0
    p[121] = 0
    p[122] = 0
    p[123] = 0
    p[124] = 0
    p[125] = 0
    p[126] = 0
    p[127] = 0
    p[128] = 0
    p[129] = 0
    p[130] = 0
    p[131] = 0
    p[132] = 0
    p[133] = 0
    p[134] = 0
    p[135] = 0
    p[136] = 0
    p[137] = 0
    p[138] = 0
    p[139] = 0
    p[140] = 0
    p[141] = 0
    p[142] = 0
    p[143] = 0
    p[144] = 0
    p[145] = 0
    p[146] = 0
    p[147] = 0
    p[148] = 0
    p[149] = 0
    p[150] = 0
    p[151] = 0
    p[152] = 0
    p[153] = 0
    p[154] = 0
    p[155] = 0
    p[156] = 0
    p[157] = 0
    p[158] = 0
    p[159] = 0
    p[160] = 0
    p[161] = 0
    p[162] = 0
    p[163] = 0
    p[164] = 0
    p[165] = 0
    p[166] = 0
    p[167] = 0
    p[168] = 0
    p[169] = 0
    p[170] = 0
    p[171] = 0
    p[172] = 0
    p[173] = 0
    p[174] = 0
    p[175] = 0
    p[176] = 0
    p[177] = 0
    p[178] = 0
    p[179] = 0
    p[180] = 0
    p[181] = 0
    p[182] = 0
    p[183] = 0
    p[184] = 0
    p[185] = 0
    p[186] = 0
    p[187] = 0
    p[188] = 0
    p[189] = 0
    p[190] = 0
    p[191] = 0
    p[192] = 0
    p[193] = 0
    p[194] = 0
    p[195] = 0
    p[196] = 0
    p[197] = 0
    p[198] = 0
    p[199] = 0
    p[200] = 0
    p[201] = 0
    p[202] = 0
    p[203] = 0
    p[204] = 0
    p[205] = 0
    p[206] = 0
    p[207] = 0
    p[208] = 0
    p[209] = 0
    p[210] = 0
    p[211] = 0
    p[212] = 0
    p[213] = 0
    p[214] = 0
    p[215] = 0
    p[216] = 0
    p[217] = 0
    p[218] = 0
    p[219] = 0
    p[220] = 0
    p[221] = 0
    p[222] = 0
    p[223] = 0
    p[224] = 0
    p[225] = 0
    p[226] = 0
    p[227] = 0
    p[228] = 0
    p[229] = 0
    p[230] = 0
    p[231] = 0
    p[232] = 0
    p[233] = 0
    p[234] = 0
    p[235] = 0
    p[236] = 0
    p[237] = 0
    p[238] = 0
    p[239] = 0
    p[240] = 0
    p[241] = 0
    p[242] = 0
    p[243] = 0
    p[244] = 0
    p[245] = 0
    p[246] = 0
    p[247] = 0
    p[248] = 0
    p[249] = 0
    p[250] = 0
    p[251] = 0
    p[252] = 0
    p[253] = 0
    p[254] = 0
    p[255] = 0
    p[256] = 0
    p[257] = 0
    p[258] = 0
    p[259] = 0
    p[260] = 0
    p[261] = 0
    p[262] = 0
    p[263] = 0
    p[264] = 0
    p[265] = 0
    p[266] = 0
    p[267] = 0
    p[268] = 0
    p[269] = 0
    p[270] = 0
    p[271] = 0
    p[272] = 0
    p[273] = 0
    p[274] = 0
    p[275] = 0
    p[276] = 0
    p[277] = 0
    p[278] = 0
    p[279] = 0
    p[280] = 0
    p[281] = 0
    p[282] = 0
    p[283] = 0
    p[284] = 0
    p[285] = 0
    p[286] = 0
    p[287] = 0
    p[288] = 0
    p[289] = 0
    p[290] = 0
    p[291] = 0
    p[292] = 0
    p[293] = 0
    p[294] = 0
    p[295] = 0
    p[296] = 0
    p[297] = 0
    p[298] = 0
    p[299] = 0
    p[300] = 0
    p[301] = 0
    p[302] = 0
    p[303] = 0
    p[304] = 0
    p[305] = 0
    p[306] = 0
    p[307] = 0
    p[308] = 0
    p[309] = 0
    p[310] = 0
    p[311] = 0
    p[312] = 0
    p[313] = 0
    p[314] = 0
    p[315] = 0
    p[316] = 0
    p[317] = 0
    p[318] = 0
    p[319] = 0
    p[320] = 0
    p[321] = 0
    p[322] = 0
    p[323] = 0
    p[324] = 0
    p[325] = 0
    p[326] = 0
    p[327] = 0
    p[328] = 0
    p[329] = 0
    p[330] = 0
    p[331] = 0
    p[332] = 0
    p[333] = 0
    p[334] = 0
    p[335] = 0
    p[336] = 0
    p[337] = 0
    p[338] = 0
    p[339] = 0
    p[340] = 0
    p[341] = 0
    p[342] = 0
    p[343] = 0
    p[344] = 0
    p[345] = 0
    p[346] = 0
    p[347] = 0
    p[348] = 0
    p[349] = 0
    p[350] = 0
    p[351] = 0
    p[352] = 0
    p[353] = 0
    p[354] = 0
    p[355] = 0
    p[356] = 0
    p[357] = 0
    p[358] = 0
    p[359] = 0
    p[360] = 0
    p[361] = 0
    p[362] = 0
    p[363] = 0
    p[364] = 0
    p[365] = 0
    p[366] = 0
    p[367] = 0
    p[368] = 0
    p[369] = 0
    p[370] = 0
    p[371] = 0
    p[372] = 0
    p[373] = 0
    p[374] = 0
    p[375] = 0
    p[376] = 0
    p[377] = 0
    p[378] = 0
    p[379] = 0
    p[380] = 0
    p[381] = 0
    p[382] = 0
    p[383] = 0
    p[384] = 0
    p[385] = 0
    p[386] = 0
    p[387] = 0
    p[388] = 0
    p[389] = 0
    p[390] = 0
    p[391] = 0
    p[392] = 0
    p[393] = 0
    p[394] = 0
    p[395] = 0
    p[396] = 0
    p[397] = 0
    p[398] = 0
    p[399] = 0
    p[400] = 0
    p[401] = 0
    p[402] = 0
    p[403] = 0
    p[404] = 0
    p[405] = 0
    p[406] = 0
    p[407] = 0
    p[408] = 0
    p[409] = 0
    p[410] = 0
    p[411] = 0
    p[412] = 0
    p[413] = 0
    p[414] = 0
    p[415] = 0
    p[416] = 0
    p[417] = 0
    p[418] = 0
    p[419] = 0
    p[420] = 0
    p[421] = 0
    p[422] = 0
    p[423] = 0
    p[424] = 0
    p[425] = 0
    p[426] = 0
    p[427] = 0
    p[428] = 0
    p[429] = 0
    p[430] = 0
    p[431] = 0
    p[432] = 0
    p[433] = 0
    p[434] = 0
    p[435] = 0
    p[436] = 0
    p[437] = 0
    p[438] = 0
    p[439] = 0
    p[440] = 0
    p[441] = 0
    p[442] = 0
    p[443] = 0
    p[444] = 0
    p[445] = 0
    p[446] = 0
    p[447] = 0
    p[448] = 0
    p[449] = 0
    p[450] = 0
    p[451] = 0
    p[452] = 0
    p[453] = 0
    p[454] = 0
    p[455] = 0
    p[456] = 0
    p[457] = 0
    p[458] = 0
    p[459] = 0
    p[460] = 0
    p[461] = 0
    p[462] = 0
    p[463] = 0
    p[464] = 0
    p[465] = 0
    p[466] = 0
    p[467] = 0
    p[468] = 0
    p[469] = 0
    p[470] = 0
    p[471] = 0
    p[472] = 0
    p[473] = 0
    p[474] = 0
    p[475] = 0
    p[476] = 0
    p[477] = 0
    p[478] = 0
    p[479] = 0
    p[480] = 0
    p[481] = 0
    p[482] = 0
    p[483] = 0
    p[484] = 0
    p[485] = 0
    p[486] = 0
    p[487] = 0
    p[488] = 0
    p[489] = 0
    p[490] = 0
    p[491] = 0
    p[492] = 0
    p[493] = 0
    p[494] = 0
    p[495] = 0
    p[496] = 0
    p[497] = 0
    p[498] = 0
    p[499] = 0
    p[500] = 0
    p[501] = 0
    p[502] = 0
    p[503] = 0
    p[504] = 0
    p[505] = 0
    p[506] = 0
    p[507] = 0
    p[508] = 0
    p[509] = 0
    p[510] = 0
    p[511] = 0
    p[512] = 0
    p[513] = 0
    p[514] = 0
    p[515] = 0
    p[516] = 0
    p[517] = 0
    p[518] = 0
    p[519] = 0
    p[520] = 0
    p[521] = 0
    p[522] = 0
    p[523] = 0
    p[524] = 0
    p[525] = 0
    p[526] = 0
    p[527] = 0
    p[528] = 0
    p[529] = 0
    p[530] = 0
    p[531] = 0
    p[532] = 0
    p[533] = 0
    p[534] = 0
    p[535] = 0
    p[536] = 0
    p[537] = 0
    p[538] = 0
    p[539] = 0
    p[540] = 0
    p[541] = 0
    p[542] = 0
    p[543] = 0
    p[544] = 0
    p[545] = 0
    p[546] = 0
    p[547] = 0
    p[548] = 0
    p[549] = 0
    p[550] = 0
    p[551] = 0
    p[552] = 0
    p[553] = 0
    p[554] = 0
    p[555] = 0
    p[556] = 0
    p[557] = 0
    p[558] = 0
    p[559] = 0
    p[560] = 0
    p[561] = 0
    p[562] = 0
    p[563] = 0
    p[564] = 0
    p[565] = 0
    p[566] = 0
    p[567] = 0
    p[568] = 0
    p[569] = 0
    p[570] = 0
    p[571] = 0
    p[572] = 0
    p[573] = 0
    p[574] = 0
    p[575] = 0
    p[576] = 0
    p[577] = 0
    p[578] = 0
    p[579] = 0
    p[580] = 0
    p[581] = 0
    p[582] = 0
    p[583] = 0
    p[584] = 0
    p[585] = 0
    p[586] = 0
    p[587] = 0
    p[588] = 0
    p[589] = 0
    p[590] = 0
    p[591] = 0
    p[592] = 0
    p[593] = 0
    p[594] = 0
    p[595] = 0
    p[596] = 0
    p[597] = 0
    p[598] = 0
    p[599] = 0
    p[600] = 0
    p[601] = 0
    p[602] = 0
    p[603] = 0
    p[604] = 0
    p[605] = 0
    p[606] = 0
    p[607] = 0
    p[608] = 0
    p[609] = 0
    p[610] = 0
    p[611] = 0
    p[612] = 0
    p[613] = 0
    p[614] = 0
    p[615] = 0
    p[616] = 0
    p[617] = 0
    p[618] = 0
    p[619] = 0
    p[620] = 0
    p[621] = 0
    p[622] = 0
    p[623] = 0
    p[624] = 0
    p[625] = 0
    p[626] = 0
    p[627] = 0
    p[628] = 0
    p[629] = 0
    p[630] = 0
    p[631] = 0
    p[632] = 0
    p[633] = 0
    p[634] = 0
    p[635] = 0
    p[636] = 0
    p[637] = 0
    p[638] = 0
    p[639] = 0
    p[640] = 0
    p[641] = 0
    p[642] = 0
    p[643] = 0
    p[644] = 0
    p[645] = 0
    p[646] = 0
    p[647] = 0
    p[648] = 0
    p[649] = 0
    p[650] = 0
    p[651] = 0
    p[652] = 0
    p[653] = 0
    p[654] = 0
    p[655] = 0
    p[656] = 0
    p[657] = 0
    p[658] = 0
    p[659] = 0
    p[660] = 0
    p[661] = 0
    p[662] = 0
    p[663] = 0
    p[664] = 0
    p[665] = 0
    p[666] = 0
    p[667] = 0
    p[668] = 0
    p[669] = 0
    p[670] = 0
    p[671] = 0
    p[672] = 0
    p[673] = 0
    p[674] = 0
    p[675] = 0
    p[676] = 0
    p[677] = 0
    p[678] = 0
    p[679] = 0
    p[680] = 0
    p[681] = 0
    p[682] = 0
    p[683] = 0
    p[684] = 0
    p[685] = 0
    p[686] = 0
    p[687] = 0
    p[688] = 0
    p[689] = 0
    p[690] = 0
    p[691] = 0
    p[692] = 0
    p[693] = 0
    p[694] = 0
    p[695] = 0
    p[696] = 0
    p[697] = 0
    p[698] = 0
    p[699] = 0
    p[700] = 0
    p[701] = 0
    p[702] = 0
    p[703] = 0
    p[704] = 0
    p[705] = 0
    p[706] = 0
    p[707] = 0
    p[708] = 0
    p[709] = 0
    p[710] = 0
    p[711] = 0
    p[712] = 0
    p[713] = 0
    p[714] = 0
    p[715] = 0
    p[716] = 0
    p[717] = 0
    p[718] = 0
    p[719] = 0
    p[720] = 0
    p[721] = 0
    p[722] = 0
    p[723] = 0
    p[724] = 0
    p[725] = 0
    p[726] = 0
    p[727] = 0
    p[728] = 0
    p[729] = 0
    p[730] = 0
    p[731] = 0
    p[732] = 0
    p[733] = 0
    p[734] = 0
    p[735] = 0
    p[736] = 0
    p[737] = 0
    p[738] = 0
    p[739] = 0
    p[740] = 0
    p[741] = 0
    p[742] = 0
    p[743] = 0
    p[744] = 0
    p[745] = 0
    p[746] = 0
    p[747] = 0
    p[748] = 0
    p[749] = 0
    p[750] = 0
    p[751] = 0
    p[752] = 0
    p[753] = 0
    p[754] = 0
    p[755] = 0
    p[756] = 0
    p[757] = 0
    p[758] = 0
    p[759] = 0
    p[760] = 0
    p[761] = 0
    p[762] = 0
    p[763] = 0
    p[764] = 0
    p[765] = 0
    p[766] = 0
    p[767] = 0
    p[768] = 0
    p[769] = 0
    p[770] = 0
    p[771] = 0
    p[772] = 0
    p[773] = 0
    p[774] = 0
    p[775] = 0
    p[776] = 0
    p[777] = 0
    p[778] = 0
    p[779] = 0
    p[780] = 0
    p[781] = 0
    p[782] = 0
    p[783] = 0
    p[784] = 0
    p[785] = 0
    p[786] = 0
    p[787] = 0
    p[788] = 0
    p[789] = 0
    p[790] = 0
    p[791] = 0
    p[792] = 0
    p[793] = 0
    p[794] = 0
    p[795] = 0
    p[796] = 0
    p[797] = 0
    p[798] = 0
    p[799] = 0
    p[800] = 0
    p[801] = 0
    p[802] = 0
    p[803] = 0
    p[804] = 0
    p[805] = 0
    p[806] = 0
    p[807] = 0
    p[808] = 0
    p[809] = 0
    p[810] = 0
    p[811] = 0
    p[812] = 0
    p[813] = 0
    p[814] = 0
    p[815] = 0
    p[816] = 0
    p[817] = 0
    p[818] = 0
    p[819] = 0
    p[820] = 0
    p[821] = 0
    p[822] = 0
    p[823] = 0
    p[824] = 0
    p[825] = 0
    p[826] = 0
    p[827] = 0
    p[828] = 0
    p[829] = 0
    p[830] = 0
    p[831] = 0
    p[832] = 0
    p[833] = 0
    p[834] = 0
    p[835] = 0
    p[836] = 0
    p[837] = 0
    p[838] = 0
    p[839] = 0
    p[840] = 0
    p[841] = 0
    p[842] = 0
    p[843] = 0
    p[844] = 0
    p[845] = 0
    p[846] = 0
    p[847] = 0
    p[848] = 0
    p[849] = 0
    p[850] = 0
    p[851] = 0
    p[852] = 0
    p[853] = 0
    p[854] = 0
    p[855] = 0
    p[856] = 0
    p[857] = 0
    p[858] = 0
    p[859] = 0
    p[860] = 0
    p[861] = 0
    p[862] = 0
    p[863] = 0
    p[864] = 0
    p[865] = 0
    p[866] = 0
    p[867] = 0
    p[868] = 0
    p[869] = 0
    p[870] = 0
    p[871] = 0
    p[872] = 0
    p[873] = 0
    p[874] = 0
    p[875] = 0
    p[876] = 0
    p[877] = 0
    p[878] = 0
    p[879] = 0
    p[880] = 0
    p[881] = 0
    p[882] = 0
    p[883] = 0
    p[884] = 0
    p[885] = 0
    p[886] = 0
    p[887] = 0
    p[888] = 0
    p[889] = 0
    p[890] = 0
    p[891] = 0
    p[892] = 0
    p[893] = 0
    p[894] = 0
    p[895] = 0
    p[896] = 0
    p[897] = 0
    p[898] = 0
    p[899] = 0
    p[900] = 0
    p[901] = 0
    p[902] = 0
    p[903] = 0
    p[904] = 0
    p[905] = 0
    p[906] = 0
    p[907] = 0
    p[908] = 0
    p[909] = 0
    p[910] = 0
    p[911] = 0
    p[912] = 0
    p[913] = 0
    p[914] = 0
    p[915] = 0
    p[916] = 0
    p[917] = 0
    p[918] = 0
    p[919] = 0
    p[920] = 0
    p[921] = 0
    p[922] = 0
    p[923] = 0
    p[924] = 0
    p[925] = 0
    p[926] = 0
    p[927] = 0
    p[928] = 0
    p[929] = 0
    p[930] = 0
    p[931] = 0
    p[932] = 0
    p[933] = 0
    p[934] = 0
    p[935] = 0
    p[936] = 0
    p[937] = 0
    p[938] = 0
    p[939] = 0
    p[940] = 0
    p[941] = 0
    p[942] = 0
    p[943] = 0
    p[944] = 0
    p[945] = 0
    p[946] = 0
    p[947] = 0
    p[948] = 0
    p[949] = 0
    p[950] = 0
    p[951] = 0
    p[952] = 0
    p[953] = 0
    p[954] = 0
    p[955] = 0
    p[956] = 0
    p[957] = 0
    p[958] = 0
    p[959] = 0
    p[960] = 0
    p[961] = 0
    p[962] = 0
    p[963] = 0
    p[964] = 0
    p[965] = 0
    p[966] = 0
    p[967] = 0
    p[968] = 0
    p[969] = 0
    p[970] = 0
    p[971] = 0
    p[972] = 0
    p[973] = 0
    p[974] = 0
    p[975] = 0
    p[976] = 0
    p[977] = 0
    p[978] = 0
    p[979] = 0
    p[980] = 0
    p[981] = 0
    p[982] = 0
    p[983] = 0
    p[984] = 0
    p[985] = 0
    p[986] = 0
    p[987] = 0
    p[988] = 0
    p[989] = 0
    p[990] = 0
    p[991] = 0
    p[992] = 0
    p[993] = 0
    p[994] = 0
    p[995] = 0
    p[996] = 0
    p[997] = 0
    p[998] = 0
    p[999] = 0
    p[1000] = 0
    p[1001] = 0
    p[1002] = 0
    p[1003] = 0
    p[1004] = 0
    p[1005] = 0
    p[1006] = 0
    p[1007] = 0
    p[1008] = 0
    p[1009] = 0
    p[1010] = 0
    p[1011] = 0
    p[1012] = 0
    p[1013] = 0
    p[1014] = 0
    p[1015] = 0
    p[1016] = 0
    p[1017] = 0
    p[1018] = 0
    p[1019] = 0
    p[1020] = 0
    p[1021] = 0
    p[1022] = 0
    p[1023] = 0
    p[1024] = 0
    p[1025] = 0
    p[1026] = 0
    p[1027] = 0
    p[1028] = 0
    p[1029] = 0
    p[1030] = 0
    p[1031] = 0
    p[1032] = 0
    p[1033] = 0
    p[1034] = 0
    p[1035] = 0
    p[1036] = 0
    p[1037] = 0
    p[1038] = 0
    p[1039] = 0
    p[1040] = 0
    p[1041] = 0
    p[1042] = 0
    p[1043] = 0
    p[1044] = 0
    p[1045] = 0
    p[1046] = 0
    p[1047] = 0
    p[1048] = 0
    p[1049] = 0
    p[1050] = 0
    p[1051] = 0
    p[1052] = 0
    p[1053] = 0
    p[1054] = 0
    p[1055] = 0
    p[1056] = 0
    p[1057] = 0
    p[1058] = 0
    p[1059] = 0
    p[1060] = 0
    p[1061] = 0
    p[1062] = 0
    p[1063] = 0
    p[1064] = 0
    p[1065] = 0
    p[1066] = 0
    p[1067] = 0
    p[1068] = 0
    p[1069] = 0
    p[1070] = 0
    p[1071] = 0
    p[10
```

# Выполнение лабораторной работы

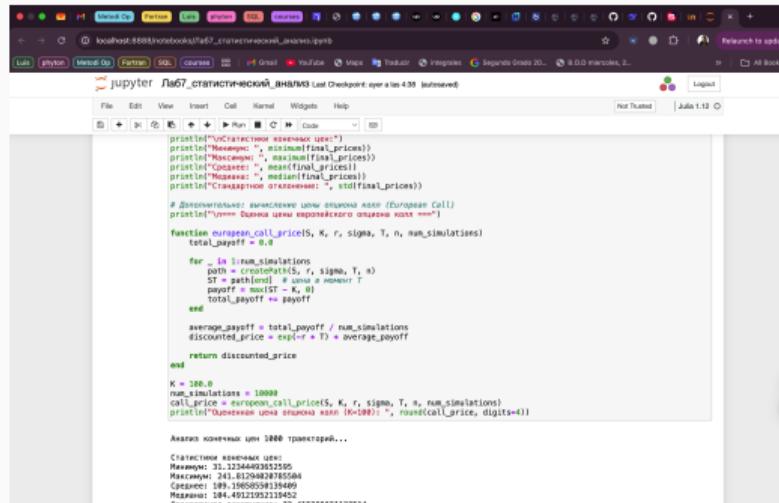


The screenshot shows a Jupyter Notebook interface running on a local host. The notebook has a single cell containing Python code. The code is used to simulate stock prices using a binomial model and then compare the distribution of final prices with a theoretical lognormal distribution.

```
return path
and
print("увидим, сколько цен 1000 траекторий...")
final_prices = zeros(1000)
Threads,threads for i in range(1,1000)
    path = createPath(S, r, sigma, T, n)
    final_prices[i] = path[nd]
and
# Гистограмма полученных цен
histogram(final_prices,
          bins=50,
          title="Distribution of Final Stock Prices (1000 paths)",
          xlabel="Final Price",
          ylabel="Frequency",
          fillcolor="lightblue",
          linecolor="black",
          normalize=True,
          label="Frequency")
# добавлено гетерогенное логнормальное распределение
using Distributions
# Параметры логнормального распределения
mu = log(S) + (r - sigma^2/2) * T
std_log = sigma * sqrt(T)
log_normal = LogNormal(mu, std_log)
log_norm = lognormal(mu, std_log)
x_vals = range(min(final_prices)>0.0, maximum(final_prices)<1.2, length=200)
pdf_x_vals = pdf(log_normal, x_vals)
plot(x_vals, pdf_x_vals, x_label="lognormal", y_label="Theoretical Frequency",
      linecolor=red, label="Theoretical Lognormal")
# Сравнение
print("Сколько цен...")
print("Максимум: ", maximum(final_prices))
```

Рис. 18: Модель ценообразования биномиальных опционов

# Выполнение лабораторной работы

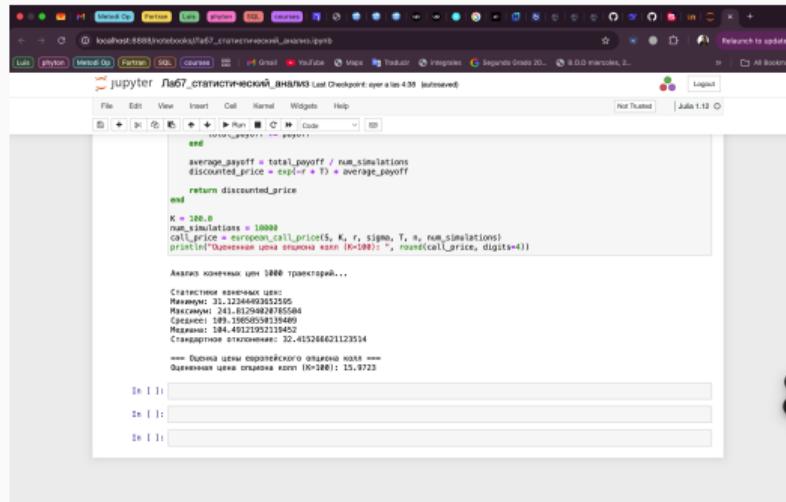


The screenshot shows a Jupyter Notebook interface running on a local host. The notebook title is "jupyter\_Лаб7\_статистический\_анализ.ipynb". The code cell contains Python code for calculating the price of a European call option using a binomial model. The code includes imports for numpy, random, and math, and defines functions for binomial pricing and a function to calculate the average payoff. It also includes a section for analyzing the final prices of 1000 trajectories. The output cell shows the results of the analysis.

```
print("Анализ конечных цен 1000 траекторий...")
статистика конечных цен:
    Минимум: 243.81294824815598
    Максимум: 243.81294824815598
    Среднее: 189.19850598198499
    Медиана: 184.491210952119452
```

Рис. 19: Модель ценообразования биномиальных опционов

# Выполнение лабораторной работы



The screenshot shows a Jupyter Notebook interface running on a local host. The code cell contains Python code for calculating the price of a European call option using a binomial model. The code includes imports for numpy, math, and scipy.stats, defines a function for binomial payoff, calculates average payoff and discounted price, and prints the final call price. The output cell displays the analysis of 5000 final prices, including minimum, maximum, mean, median, and standard deviation, followed by the final call price.

```
import numpy as np
import math
from scipy.stats import norm

def binomial_payoff(s, K):
    if s > K:
        return s - K
    else:
        return 0

K = 300.0
num_simulations = 10000
call_price = european_call_price(S, K, r, sigma, T, n, num_simulations)
print("Оцененная цена опциона колл (n=100): ", round(call_price, digits=4))

Анализ конечных цен 5000 траекторий...
Статистики конечных цен:
Минимум: 122.00000000000002
Максимум: 248.00000000000003
Среднее: 189.1985059195499
Средний: 184.40121952118452
Стандартное отклонение: 32.415208621123514
*** Оценка цены европейского опциона колл ***
Оцененная цена опциона колл (n=200): 15.9723
```

In [1]:  
In [1]:  
In [1]:

Рис. 20: Модель ценообразования биномиальных опционов

## Выводы

---

В результате выполнения данной лабораторной работы я освоила специализированные пакеты Julia для обработки данных.

## Список литературы

---

1. JuliaLang [Электронный ресурс]. 2024 JuliaLang.org contributors.  
URL:<https://julialang.org/>(дата обращения: 11.10.2024).
2. Julia 1.11 Documentation [Электронный ресурс]. 2024 JuliaLang.orgcontributors.  
URL:<https://docs.julialang.org/en/v1/>(дата обращения:11.10.2024).