Введение в работу с данными

Гань Чжаолун

26 декабрь, 2024, Москва, Россия

Российский Университет Дружбы Народов

Цели и задачи работы

Цель лабораторной работы

Основной целью работы является специализированных пакетов Julia для обработки данных.

Процесс выполнения лабораторной работы

Используя Jupyter Lab, повторите примеры из раздела 7.2

Я повторю все задание 7.2 целиком

Выполните задания для самостоятельной работы

```
using RDatasets
iris = dataset("datasets", "iris")
```

150 rows × 5 columns

	SepalLength	SepalWidth	PetalLength	PetalWidth	Species
	Float64	Float64	Float64	Float64	Cat...
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3.0	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
4	4.6	3.1	1.5	0.2	setosa

```
# Построение графика:
plot(size=(500,500),leg=false)

x = iris[!,:SepalLength]
y = iris[!,:PetalLength]
scatter(x, y, markersize=3, title="Распределение признаков SepalLength и PetalLength",
    xlabel="SepalLength", ylabel="PetalLength", leg=false)
```

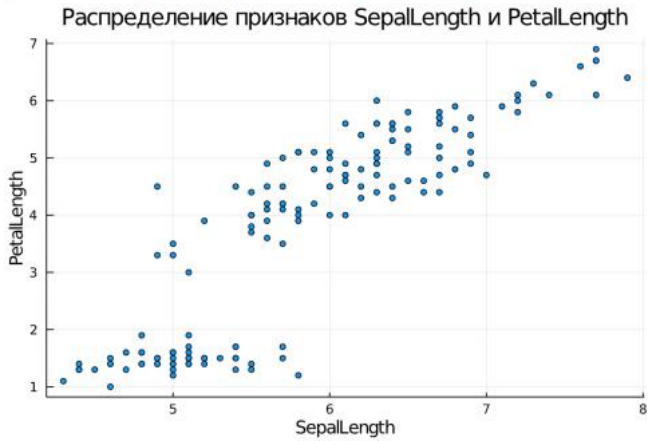
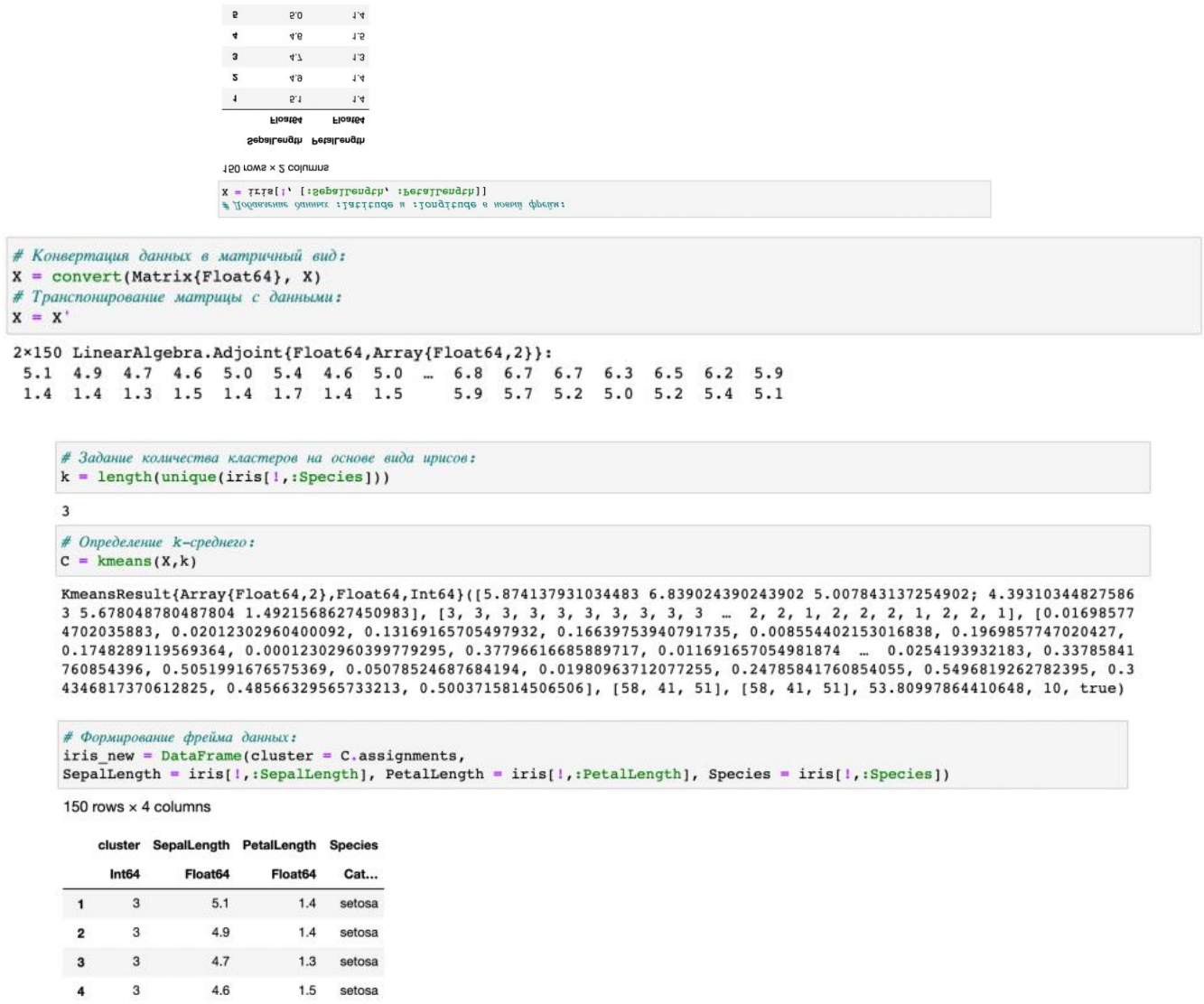


Рисунок 1. Код и результат Задания 1.1–1.2

Выполните задания для самостоятельной работы



```
# Конвертация данных в матричный вид:
X = convert(Matrix{Float64}, X)
# Транспонирование матрицы с данными:
X = X'
```

```
2×150 LinearAlgebra.Adjoint{Float64,Array{Float64,2}}:
 5.1  4.9  4.7  4.6  5.0  5.4  4.6  5.0  …  6.8  6.7  6.7  6.3  6.5  6.2  5.9
 1.4  1.4  1.3  1.5  1.4  1.7  1.4  1.5     5.9  5.7  5.2  5.0  5.2  5.4  5.1
```

```
# Задание количества кластеров на основе вида ирисов:
k = length(unique(iris[!,:Species]))
```

```
3
```

```
# Определение k-среднего:
C = kmeans(X,k)
```

```
KmeansResult{Array{Float64,2},Float64,Int64}([5.874137931034483 6.839024390243902 5.007843137254902; 4.39310344827586
3 5.678048780487804 1.4921568627450983], [3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3  …  2, 2, 1, 2, 2, 2, 2, 1, 2, 2, 1], [0.01698577
4702035883, 0.02012302960400092, 0.13169165705497932, 0.16639753940791735, 0.008554402153016838, 0.1969857747020427,
0.1748289119569364, 0.00012302960399779295, 0.37796616685889717, 0.011691657054981874  …  0.0254193932183, 0.33785841
760854396, 0.5051991676575369, 0.05078524687684194, 0.01980963712077255, 0.24785841760854055, 0.5496819262782395, 0.3
4346817370612825, 0.48566329565733213, 0.5003715814506506], [58, 41, 51], [58, 41, 51], 53.80997864410648, 10, true)
```

```
# Формирование фрейма данных:
iris_new = DataFrame(cluster = C.assignments,
SepalLength = iris[!,:SepalLength], PetalLength = iris[!,:PetalLength], Species = iris[!,:Species])
```

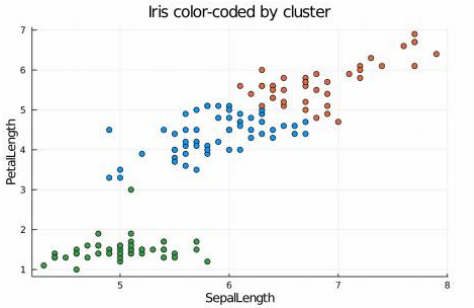
150 rows × 4 columns

	cluster	SepalLength	PetalLength	Species
	Int64	Float64	Float64	Cat...
1	3	5.1	1.4	setosa
2	3	4.9	1.4	setosa
3	3	4.7	1.3	setosa
4	3	4.6	1.5	setosa

Рисунок 2. Код и результат Задания 1.3–1.6

Выполните задания для самостоятельной работы

```
unique_species = unique(iris[!,:Species])
species_figure = plot(legend = false)
for uspecies in unique_species
    iris_sp = iris[iris[!,:Species].==uspecies,:]
    x = iris_sp[!,:SepalLength]
    y = iris_sp[!,:PetalLength]
    scatter!(species_figure, x, y)
end
xlabel!("SepalLength")
ylabel!("PetalLength")
title!("Iris color-coded by species")
display(species_figure)
```



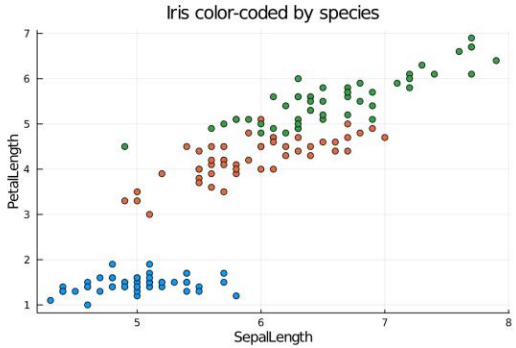
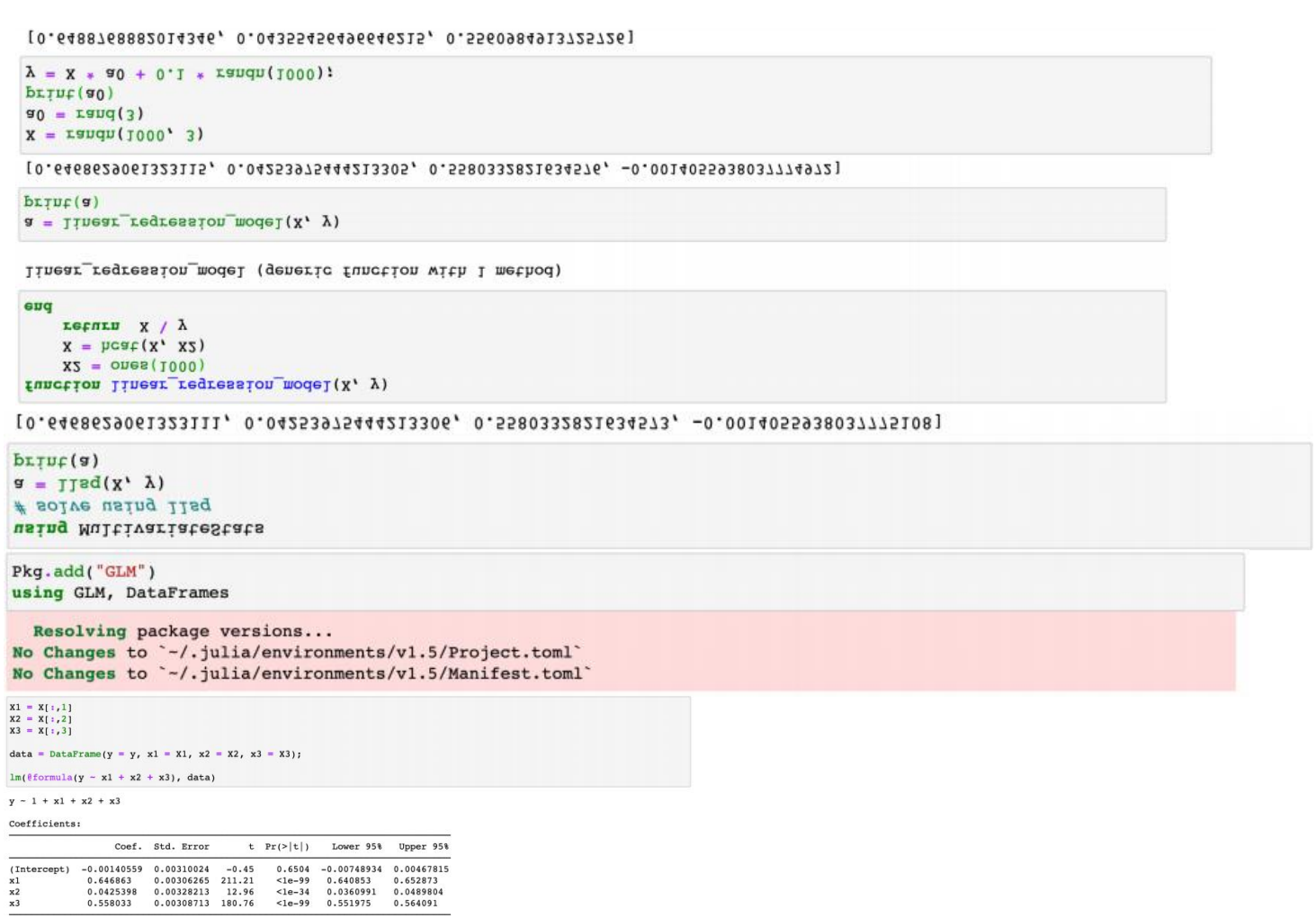


Рисунок 3. Код и результат Задания 1.7–1.8

Выполните задания для самостоятельной работы



```
Pkg.add("GLM")
using GLM, DataFrames
```

```
  Resolving package versions...
No Changes to `~/.julia/environments/v1.5/Project.toml`
No Changes to `~/.julia/environments/v1.5/Manifest.toml`
```

```
X1 = X[:,1]
X2 = X[:,2]
X3 = X[:,3]

data = DataFrame(y = y, x1 = X1, x2 = X2, x3 = X3);

lm(@formula(y ~ x1 + x2 + x3), data)
```

```
y ~ 1 + x1 + x2 + x3

Coefficients:
```

	Coef.	Std. Error	t	Pr(>|t|)	Lower 95%	Upper 95%
(Intercept)	-0.00140559	0.00310024	-0.45	0.6504	-0.00748934	0.00467815
x1	0.646863	0.00306265	211.21	<1e-99	0.640853	0.652873
x2	0.0425398	0.00328213	12.96	<1e-34	0.0360991	0.0489804
x3	0.558033	0.00308713	180.76	<1e-99	0.551975	0.564091

Рисунок 4. Код и результат Задания 2.1

Выполните задания для самостоятельной работы

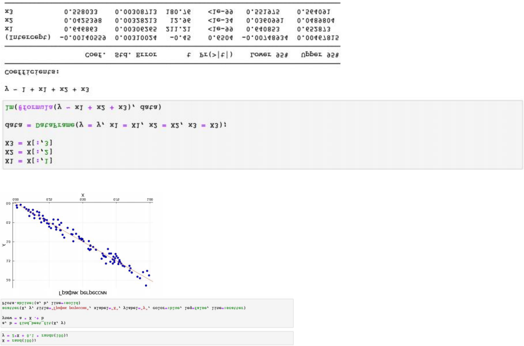
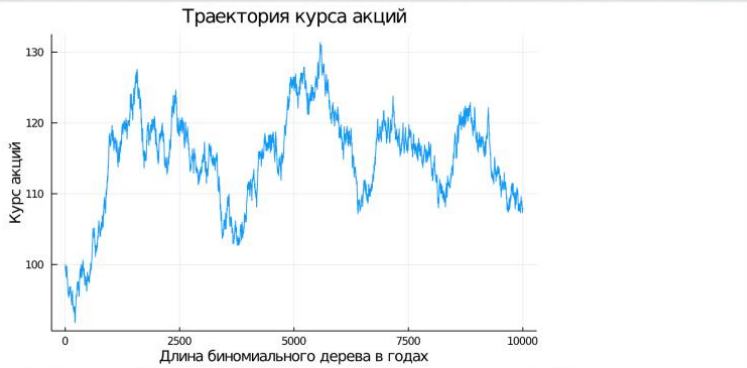


Рисунок 5. Код и результат Задания 2.2

Выполните задания для самостоятельной работы

```
S = 100
T = 1
n = 10000
sigma = 0.3
r = 0.08

h = T / n # длина одного периода;
u = exp(r*h + sigma * sqrt(h))
d = exp(r*h - sigma * sqrt(h))
p = (exp(r*h) - d)/(u - d)

j = 0
stockTree = []
append!(stockTree, S)
for i in 1:n
    k = rand()
    if k < p
        append!(stockTree, S * (u^(i - j)) * (d ^ j))
    else
        append!(stockTree, S * (u^(i - j)) * (d ^ (j + 1)))
        j = j + 1
    end
end
```

```
using Plots
plot(stockTree, title="Траектория курса акций", xlabel="Длина биномиального дерева в годах", ylabel="Курс акций", leg=false)
```

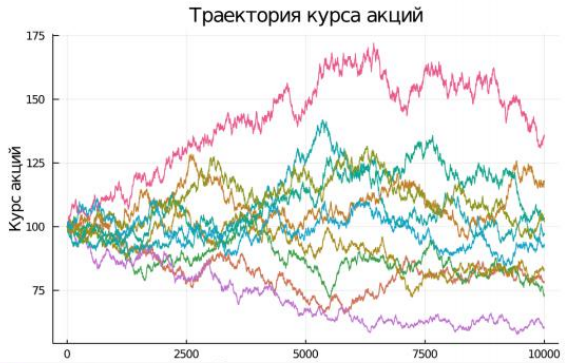



```
function createPath(S::Int64, T::Int64, n::Int64, sigma::Float64, r::Float64)
    # S - начальная цена акции;
    # T - длина биномиального дерева в годах;
    # n - длина одного периода;
    # sigma - волатильность акции;
    # r — годовая процентная ставка;

    h = T / n # длина одного периода;
    u = exp(r*h + sigma * sqrt(h))
    d = exp(r*h - sigma * sqrt(h))
    p = (exp(r*h) - d)/(u - d)
    stockTree = []
    append!(stockTree, S)
    j = 0

    for i in 1:n
        k = rand()
        if k < p
            append!(stockTree, S * (u^(i - j)) * (d ^ j))
        else
            j = j + 1
            append!(stockTree, S * (u^(i - j)) * (d ^ j))
        end
    end
    return stockTree
end
```

Рисунок 6. Код и результат Задания 3.1

Выполните задания для самостоятельной работы

```
for i in 1:10
    IJulia.clear_output(true)
    traj = createPath(100, 1, 10000, 0.3, 0.08)
    if i == 1
        p = plot(traj, title="Траектория курса акций", xlabel="Длина биномиального дерева в годах",
            ylabel="Курс акций", leg=false)
    end
    p = plot!(traj)
    display(p)
end
```




```
using Base.Threads

Threads.@threads for i in 1:10
    IJulia.clear_output(true)
    traj = createPath(100, 1, 10000, 0.3, 0.08)
    if i == 1
        g = plot(traj, title="Траектория курса акций", xlabel="Длина биномиального дерева в годах",
            ylabel="Курс акций", leg=false)
    end
    g = plot!(traj)
    display(g)
end
```

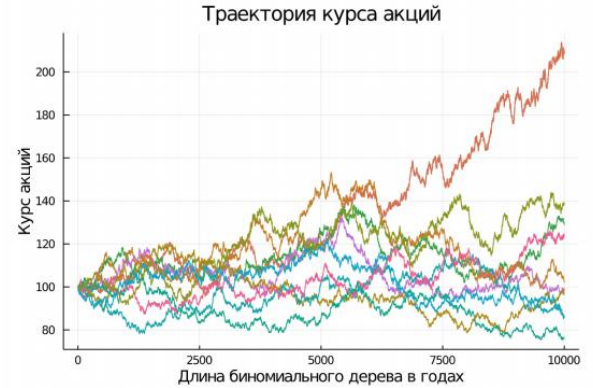


Рисунок 7. Код и результат Задания 3.2

Выводы по проделанной работе

Вывод

Я освоил специализированные пакеты для решения задач в непрерывном и дискретном времени