РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ №4

дисциплина: «Компьютерный практикум по статистическому анализу данных»

Работу выполнил:

Снимщиков Иван Игоревич

Группа: НПИбд-02-21

МОСКВА

2024 г.

Цели работы: Основной целью работы является изучение возможностей специализированных пакетов Julia для выполнения и оценки эффективности операций над объектами линейной алгебры.

Ход работы:

Первым делом я повторил примеры из файла лабораторной работы.

Массивы:

```
[51]: # Массив 4x3 со случайными целыми числами (от 1 до 20):
      a = rand(1:20, (4, 3))

      # Поэлементная сумма:
      sum(a)
```

```
[51]: 98
```

```
[ ]:
```

```
[55]: # Массив 4x3 со случайными целыми числами (от 1 до 20):
      a = rand(1:20, (4, 3))

      # Поэлементное произведение по столбцам:
      prod(a, dims=1)
```

```
[55]: 1×3 Matrix{Int64}:
       20900  24480  800
```

```
[ ]:
```

```
[57]: # Средние значения:
import Pkg
Pkg.add("Statistics")
using Statistics

# Массив 4x3 со случайными целыми числами (от 1 до 20):
a = rand(1:20, (4, 3))

# Среднее значение массива:
mean(a)
```

```
  Resolving package versions...
 No Changes to `F:\Users\vnaq2\.julia\environments\v1.10\Project.toml`
 No Changes to `F:\Users\vnaq2\.julia\environments\v1.10\Manifest.toml`
```

```
[57]: 10.166666666666666
```

Работа после подключения пакета Линейной Алгебры

```
•[59]: # Подключение пакета:
Pkg.add("LinearAlgebra")
using LinearAlgebra

# Массив 4x3 со случайными целыми числами (от 1 до 20):
a = rand(1:20, (4, 3))

# Среднее по строкам:
mean(a, dims=2)
```

```
  Resolving package versions...
 No Changes to `F:\Users\vnaq2\.julia\environments\v1.10\Project.toml`
 No Changes to `F:\Users\vnaq2\.julia\environments\v1.10\Manifest.toml`
```

```
[59]: 4×1 Matrix{Float64}:
  6.666666666666667
 14.666666666666666
 17.0
 14.333333333333334
```

```
[ ]:
```

```
[64]: # Подключение пакета:
Pkg.add("LinearAlgebra")
using LinearAlgebra

# Массив 4x4 со случайными целыми числами (от 1 до 20):
b = rand(1:20, (4, 4))

# Инверсия:
inv(b)
```

```
  Resolving package versions...
 No Changes to `F:\Users\vnaq2\.julia\environments\v1.10\Project.toml`
 No Changes to `F:\Users\vnaq2\.julia\environments\v1.10\Manifest.toml`
```

```
[64]: 4×4 Matrix{Float64}:
  0.0860487   0.0172675  -0.057149   -0.0224695
 -0.0552344  -0.0341738  -0.0082364   0.115599
 -0.0915758   0.0584495  -0.0260819   0.0327288
  0.0166173  -0.0184958   0.0737663  -0.0177733
```

```
[ ]:
```

Работа с векторами

```
[67]: # Вектор:
      X = [2, 4, -5]

      # Евклидова норма:
      norm(X)
```

```
[67]: 6.708203932499369
```

```
[ ]:
```

```
[73]: # Вектор:
      X = [2, 4, -5]
      # Евклидово расстояние между векторами:
      Y = [1, -1, 3]
      norm(X - Y)
```

```
[73]: 9.486832980505138
```

```
[ ]:
```

Произведение матриц

```
[76]: # Матрицы:
      A = rand(1:10, (2, 3))
      B = rand(1:10, (3, 4))

      # Произведение матриц:
      A * B
```

```
[76]: 2×4 Matrix{Int64}:
       69  93  93  66
       48  75  61  44
```

```
[ ]:
```

Самостоятельная работа

Скалярное произведение

```
[78]: # Вектор v:
      v = [1, 2, 3]

      # Скалярное произведение:
      dot_v = dot(v, v)
```

```
[78]: 14
```

```
[ ]:
```

Внешнее произведение

```
[79]: # Вектор v:
      v = [1, 2, 3]

      # Внешнее произведение:
      outer_v = v * transpose(v)
```

```
[79]: 3×3 Matrix{Int64}:
       1  2  3
       2  4  6
       3  6  9
```

```
[ ]:
```

Системы линейных уравнений

[80]:
```
# a) {x + y = 2, x - y = 3}
A = [1 1; 1 -1]
b = [2; 3]
x = A \ b
```

[80]:
```
2-element Vector{Float64}:
  2.5
 -0.5
```

[ ]:

```
if rank(A) < rank(hcat(A, b))
    println("Система несовместна (нет решений).")
else
    x = A \ b
    println("Решение:", x)
end
```

```
Система недоопределена (бесконечно много решений).
```

[85]:
```
# c) Система без решений
A = [1 1; 2 2]
b = [2; 5]

if rank(A) < rank(hcat(A, b))
    println("Система несовместна (нет решений).")
else
    x = A \ b
    println("Решение:", x)
end
```

```
Система несовместна (нет решений).
```

[ ]:

[81]:
```
# d) {x + y = 1, 2x + 2y = 2, 3x + 3y = 3}
A = [1 1; 2 2; 3 3]
b = [1; 2; 3]
x = A \ b
```

[81]:
```
2-element Vector{Float64}:
 0.4999999999999999
 0.5
```

[ ]:

[82]:
```
# e) {x + y = 2, 2x + y = 1, x - y = 3}
A = [1 1; 2 1; 1 -1]
b = [2; 1; 3]
x = A \ b
```

[82]:
```
2-element Vector{Float64}:
  1.5000000000000004
 -0.9999999999999997
```

[ ]:

[83]:
```
# f) {x + y = 2, 2x + y = 1, 3x + 2y = 3}
A = [1 1; 2 1; 3 2]
b = [2; 1; 3]
x = A \ b
```

[83]:
```
2-element Vector{Float64}:
 -0.9999999999999989
  2.9999999999999982
```

[ ]:

Приведение к диагональному виду

```
[87]: # Приведение к диагональному виду:
      A1 = [1 -2; -2 1]
      eigvals(A1)

      A2 = [1 -2; -2 3]
      eigvals(A2)

      A3 = [1 -2 0; -2 1 2; 0 2 0]
      eigvals(A3)
```

```
[87]: 3-element Vector{Float64}:
       -2.1413361156553643
        0.51513804712807
        3.6261980685272945
```

```
[ ]:
```

Проверка на продуктивность

```
[89]: using LinearAlgebra

      A = [1 2; 3 1]
      eigenvalues = eigvals(A)
      println("Собственные значения: ", eigenvalues)

      # Проверка продуктивности:
      if all(abs.(eigenvalues) .< 1)
          println("Матрица продуктивна.")
      else
          println("Матрица НЕ продуктивна.")
      end
```

```
Собственные значения: [-1.4494897427831779, 3.4494897427831783]
Матрица НЕ продуктивна.
```

```
[ ]:
```

[90]:
```
A = 1/2 * [1 2; 3 1]
eigenvalues = eigvals(A)
println("Собственные значения: ", eigenvalues)

# Проверка продуктивности:
if all(abs.(eigenvalues) .< 1)
    println("Матрица продуктивна.")
else
    println("Матрица НЕ продуктивна.")
end
```

```
Собственные значения: [-0.7247448713915892, 1.724744871391589]
Матрица НЕ продуктивна.
```

[ ]:

[91]:
```
using LinearAlgebra

A = [1 2; 3 1]
eigenvalues = eigvals(A)
println("Собственные значения: ", eigenvalues)

# Проверка продуктивности:
if all(abs.(eigenvalues) .< 1)
    println("Матрица продуктивна.")
else
    println("Матрица НЕ продуктивна.")
end
```

```
Собственные значения: [-1.4494897427831779, 3.4494897427831783]
Матрица НЕ продуктивна.
```

[ ]:

[92]:
```
A = [0.1 0.2 0.3; 0 0.1 0.2; 0 0.1 0.3]
eigenvalues = eigvals(A)
println("Собственные значения: ", eigenvalues)

# Проверка продуктивности:
if all(abs.(eigenvalues) .< 1)
    println("Матрица продуктивна.")
else
    println("Матрица НЕ продуктивна.")
end
```

```
Собственные значения: [0.02679491924311228, 0.1, 0.37320508075688774]
Матрица продуктивна.
```

[ ]:

Вывод: Я изучил возможности специализированных пакетов Julia для выполнения и оценки эффективности операций над объектами линейной алгебры.