

Российский университет дружбы народов
Факультет физико-математических и естественных наук

Отчёт по лабораторной работе №7

Москва 2023

1032203967
Быстров Глеб

Цель работы (задание)

- Освоение специализированных пакетов Julia для обработки данных.

Задачи (метод выполнения)

- Считывание данных

```
Ввод [8]: # Функция определения по названию языка программирования  
# года его создания (без учёта регистра):  
function language_created_year_v2(P,language::String)  
    loc = findfirst(lowercase.(P[:,2]).==lowercase.(language))  
    return P[loc,1]  
end
```

```
Out[8]: language_created_year_v2 (generic function with 1 method)
```

```
Ввод [9]: # Пример вызова функции и определение даты создания языка julia:  
language_created_year_v2(P,"julia")
```

```
Out[9]: 2012
```

```
Ввод [*]: # Построчное считывание данных с указанием разделителя:  
Tx = readdlm("programminglanguages.csv", ',',')
```


Задачи (метод выполнения)

- Запись данных в файл

Запись данных в CSV-файл:

```
CSV.write("programming_languages_data2.csv", P)
```

```
"programming_languages_data2.csv"
```

Пример записи данных в текстовый файл с разделителем ',':

```
writedlm("programming_languages_data.txt", Tx, ',')
```

Пример записи данных в текстовый файл с разделителем '-':

```
writedlm("programming_languages_data2.txt", Tx, '-')
```

Построчное считывание данных с указанием разделителя:

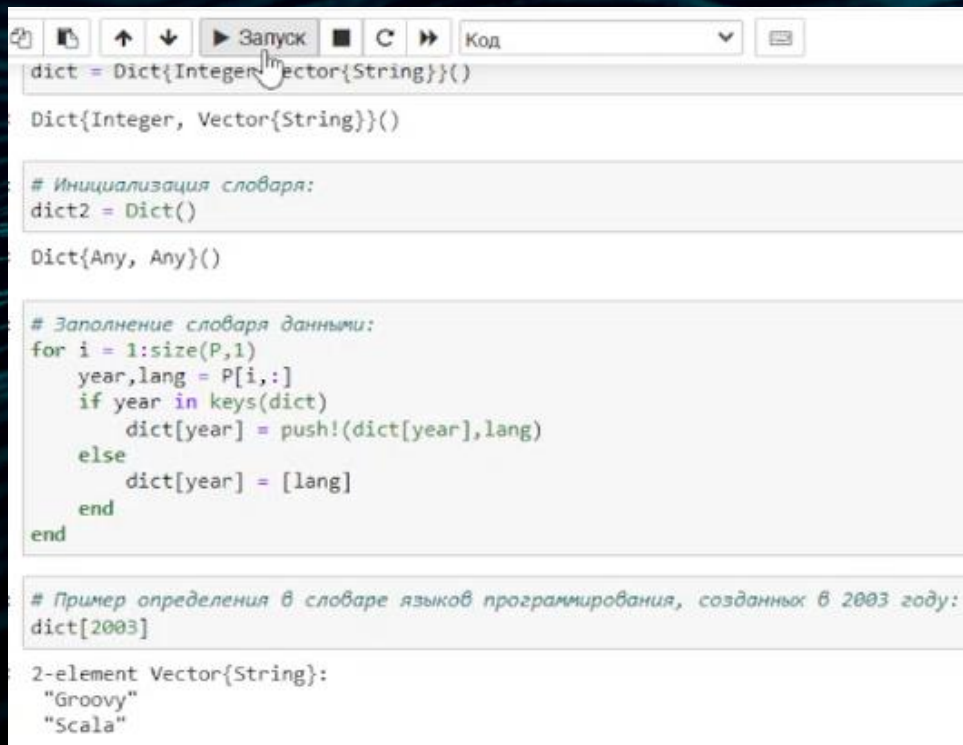
```
P_new_delim = readall("programming_languages_data2.txt", '-')
```

74x2 Matrix{Any}:

"year"	"language"
1951	"Regional Assembly Language"
1952	"Autocode"
1954	"IPL"
1955	"FLOW-MATIC"
1957	"FORTRAN"
1957	"COMTRAN"
1958	"LISP"
1958	"ALGOL 58"
1959	"FACT"

Задачи (метод выполнения)

- Словари



The screenshot shows a code editor with a toolbar at the top containing icons for file operations and a 'Запуск' (Run) button. The code is written in Scala and demonstrates dictionary operations. It includes comments in Russian for initialization and data filling. The final output shows a dictionary entry for the year 2003, containing a vector of programming languages: Groovy and Scala.

```
dict = Dict{Integer, Vector{String}}{}  
Dict{Integer, Vector{String}}()  
  
# Инициализация словаря:  
dict2 = Dict()  
Dict{Any, Any}()  
  
# Заполнение словаря данными:  
for i = 1:size(P,1)  
  year, lang = P[i,:]  
  if year in keys(dict)  
    dict[year] = push!(dict[year], lang)  
  else  
    dict[year] = [lang]  
  end  
end  
  
# Пример определения в словаре языков программирования, созданных в 2003 году:  
dict[2003]  
  
2-element Vector{String}:  
"Groovy"  
"Scala"
```

Задачи (метод выполнения)

- DataFrames

```
# Подгружаем пакет DataFrames:  
using DataFrames
```

```
# Задаём переменную со структурой DataFrame:  
df = DataFrame(year = P[:,1], language = P[:,2])
```

```
# Получение статистических сведений о фрейме:  
describe(df)
```

2x7 DataFrame

Row	variable	mean	min	median	max	nmissing	eltype
	Symbol	Union...	Any	Union...	Any	Int64	DataType
1	year	1982.99	1951	1986.0	2014	0	Int64
2	language		ALGOL 58		dBase III	0	String31

Задачи (метод выполнения)

- RDatasets

```
# Определения типа переменной:  
typeof(iris)
```

DataFrame

```
describe(iris)
```

5x7 DataFrame

Row	variable	mean	min	median	max	nmissing	eltype
	Symbol	Union...	Any	Union...	Any	Int64	DataType
1	SepalLength	5.84333	4.3	5.8	7.9	0	Float64
2	SepalWidth	3.05733	2.0	3.0	4.4	0	Float64
3	PetalLength	3.758	1.0	4.35	6.9	0	Float64
4	PetalWidth	1.19933	0.1	1.3	2.5	0	Float64
5	Species		setosa		virginica	0	CategoricalValue{String, UInt8}

Задачи (метод выполнения)

- Работа с переменными отсутствующего типа (Missing Values)

```
# Формирование данных о калориях:  
dataframe_calories = DataFrame(item=foods,calories=calories)
```

4x2 DataFrame

Row	item	calories
	String	Int64?
1	apple	missing
2	cucumber	47
3	tomato	22
4	banana	105

```
# Формирование данных о ценах:  
dataframe_prices = DataFrame(item=foods,price=prices)  
# Объединение данных о калориях и ценах:  
DF = join(dataframe_calories,dataframe_prices,on=:item)
```


Задачи (метод выполнения)

- FileIO

```
RGBA{N0f8}(0.0,0.0,0.0,0.0)  -- RGBA{N0f8}(0.0,0.0,0.0,0.0)
RGBA{N0f8}(0.0,0.0,0.0,0.0)  -- RGBA{N0f8}(0.0,0.0,0.0,0.0)
RGBA{N0f8}(0.0,0.0,0.0,0.0)  -- RGBA{N0f8}(0.0,0.0,0.0,0.0)
RGBA{N0f8}(0.0,0.0,0.0,0.0)  -- RGBA{N0f8}(0.0,0.0,0.0,0.0)
RGBA{N0f8}(0.0,0.0,0.0,0.0)  -- RGBA{N0f8}(0.0,0.0,0.0,0.0)
RGBA{N0f8}(0.0,0.0,0.0,0.0)  -- RGBA{N0f8}(0.0,0.0,0.0,0.0)
RGBA{N0f8}(0.0,0.0,0.0,0.0)  -- RGBA{N0f8}(0.0,0.0,0.0,0.0)
RGBA{N0f8}(0.0,0.0,0.0,0.0)  -- RGBA{N0f8}(0.0,0.0,0.0,0.0)
```

Определение типа и размера данных:

```
@show typeof(X1);
```

```
@show size(X1);
```

```
typeof(X1) = Matrix{ColorTypes.RGBA{FixedPointNumbers.N0f8}}
```

```
size(X1) = (828, 1280)
```

Загрузка пакетов:

```
import Pkg
```

```
Pkg.add("DataFrames")
```

```
Pkg.add("Statistics")
```

```
using DataFrames
```

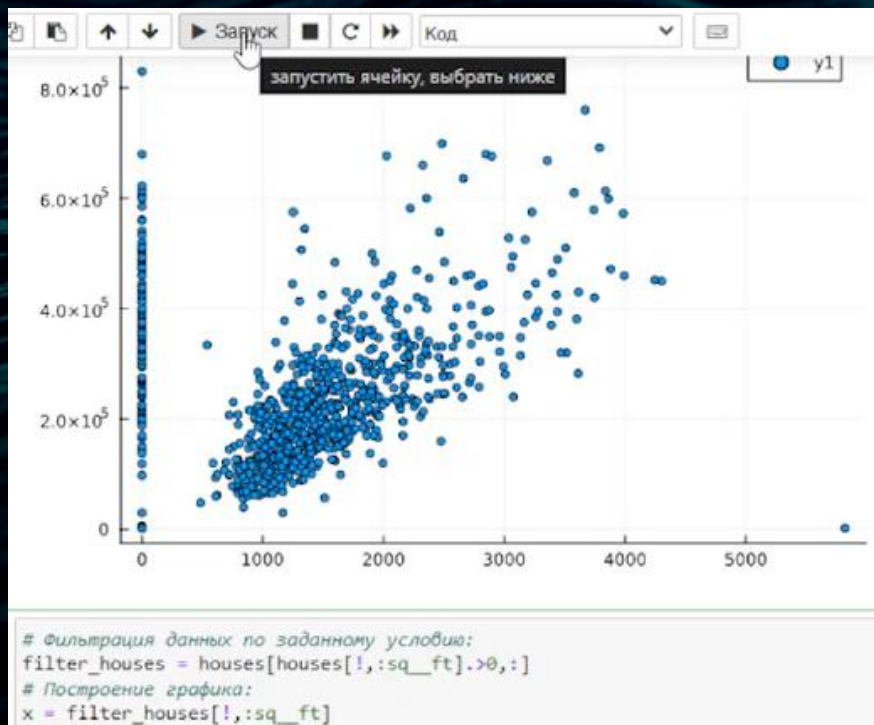
```
using CSV
```

```
import Pkg
```

```
Pkg.add("Plots") ]
```

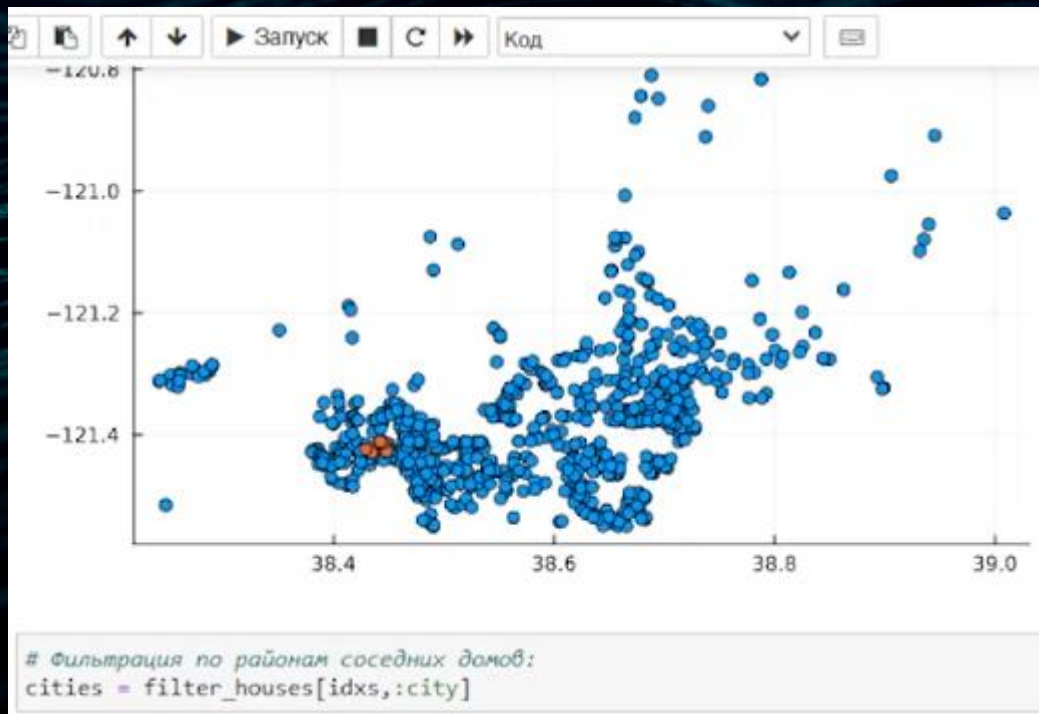
Задачи (метод выполнения)

- Кластеризация данных. Метод k-средних



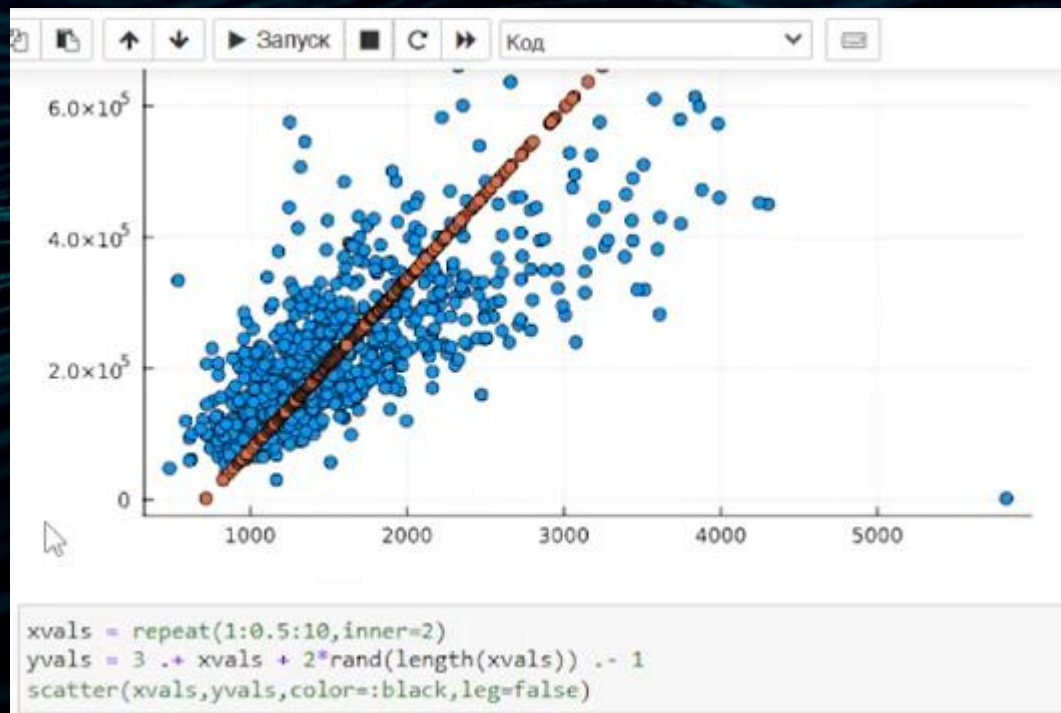
Задачи (метод выполнения)

- Кластеризация данных. Метод k ближайших соседей



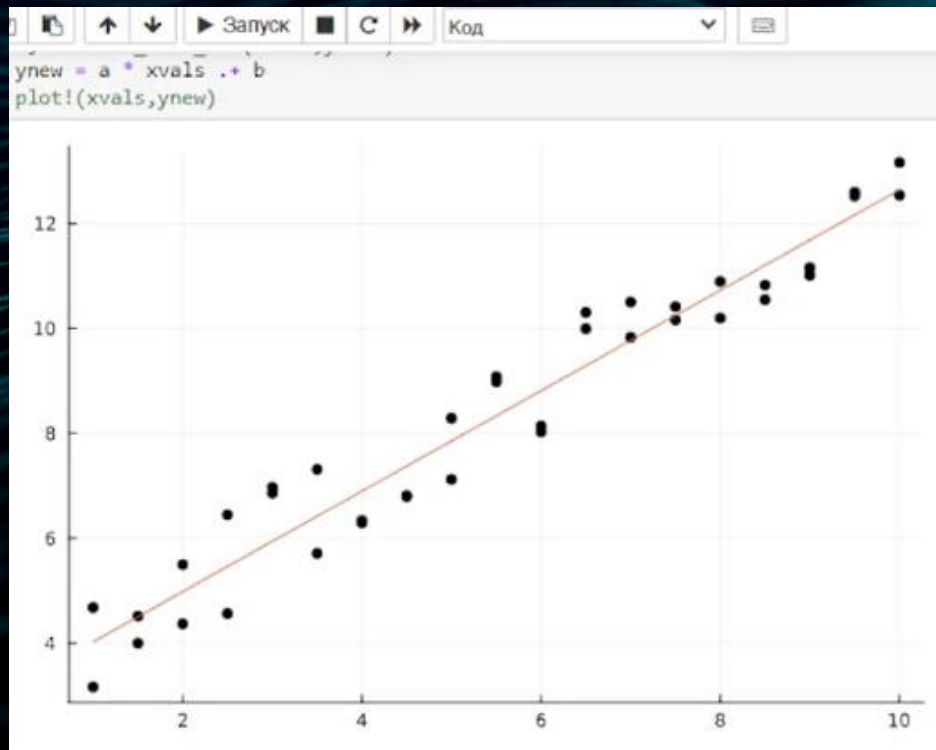
Задачи (метод выполнения)

- Обработка данных. Метод главных компонент



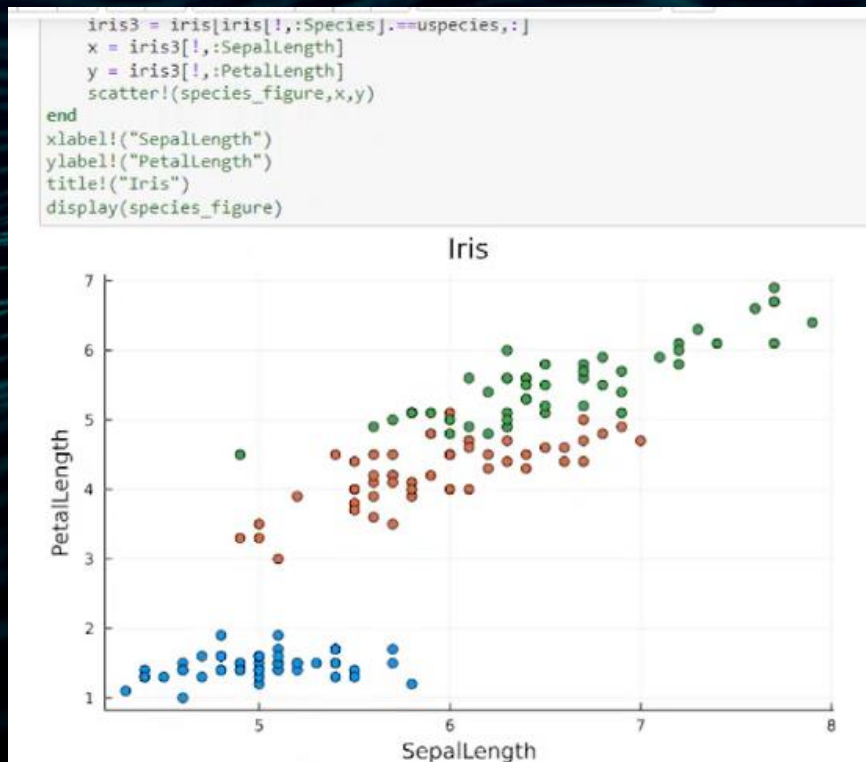
Задачи (метод выполнения)

- Обработка данных. Линейная регрессия



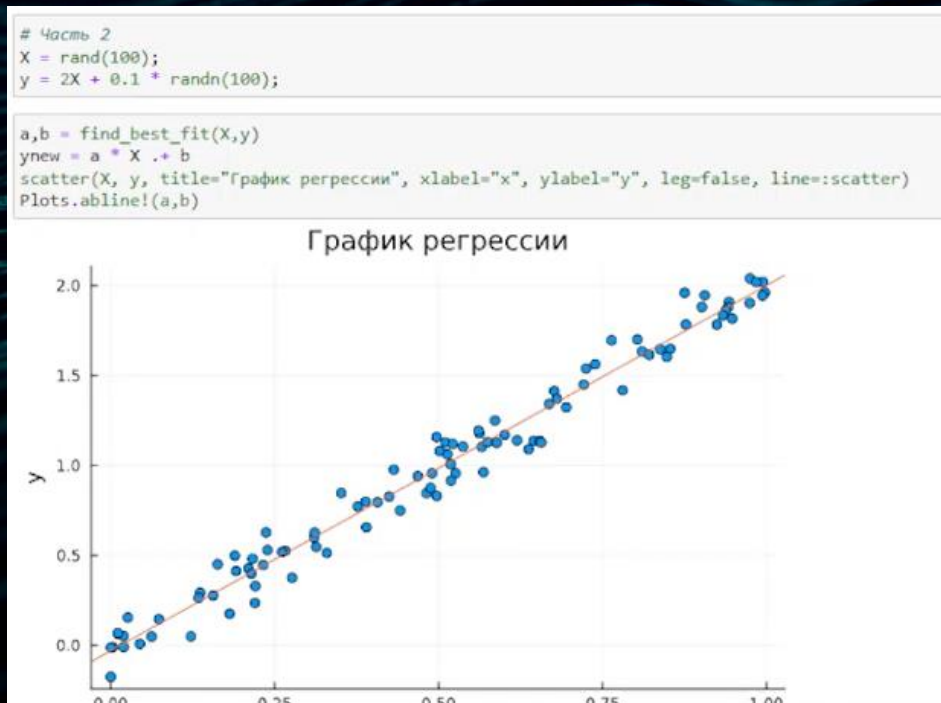
Задачи (метод выполнения)

- Задания для самостоятельного выполнения



Задачи (метод выполнения)

- Задания для самостоятельного выполнения



Результаты и их анализ

- Успешно удалось освоить специализированные пакеты Julia для обработки данных.



Благодарю за внимание