

Защита лабораторной работы № 7

Введение в работу с данными

Компьютерный практикум по статистическому анализу данных

Работу Выполнил:
Саинт-Амур Измаэль
Группа: НПИбд-01-20

Цель работы

Основной целью работы является специализированных пакетов Julia для обработки данных.

В Julia для работы с такого рода структурами данных используют пакеты CSV, DataFrames, RDatasets, FileIO

```
julia [4]: # Считывание данных и их запись в структуру:  
P = CSV.File("programminglanguages.csv") |> DataFrame
```

```
out[4]: 73x2 DataFrame
```

Row	year	language
	Int64	String31
1	1951	Regional Assembly Language
2	1952	Autocode
3	1954	IPL
4	1955	FLOW-MATIC
5	1957	FORTRAN
6	1957	COMTRAN
7	1958	LISP
8	1958	ALGOL 58
9	1959	FACT
10	1959	COBOL
11	1959	RPG
12	1962	API

Здесь вы, вероятно, рассматривали способы записи данных в файлы различных форматов. CSV.jl и FileIO.jl могли использоваться для записи данных.

```
In [11]: # Запись данных в CSV-файл:
CSV.write("programming_languages_data2.csv", P)
```

```
Out[11]: "programming_languages_data2.csv"
```

```
In [12]: # Пример записи данных в текстовый файл с разделителем ',':
writedlm("programming_languages_data.txt", Tx, ',')
```

```
In [13]: # Пример записи данных в текстовый файл с разделителем '-':
writedlm("programming_languages_data2.txt", Tx, '-')
```

```
In [14]: # Построчное считывание данных с указанием разделителя:
P_new_delim = readdlm("programming_languages_data2.txt", '-')
```

```
Out[14]: 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"
1959     "COBOL"
```

DataFrames - это библиотека для работы с табличными данными. Она предоставляет функциональность, аналогичную тому, что предоставляют фреймы данных в других языках программирования.

RDatasets предоставляет доступ к наборам данных, которые часто используются в статистических исследованиях.

FileIO: - библиотека для работы с файлами различных форматов, предоставляя унифицированный интерфейс..

```
In [155]: # Подгружаем пакет DataFrames:  
using DataFrames
```

```
In [20]: # Задаём переменную со структурой DataFrame:  
df = DataFrame(year = P[:,1], language = P[:,2])
```

Out[20]: 73×2 DataFrame

Row	year	language
	Int64	String31
1	1951	Regional Assembly Language
2	1952	Autocode
3	1954	IPL
4	1955	FLOW-MATIC
5	1957	FORTRAN
6	1957	COMTRAN
7	1958	LISP
8	1958	ALGOL 58
9	1959	FACT

```
In [41]: # Загрузка изображения:  
X1 = load("logojulia.png")
```

Out[41]:



Метод главных компонент (Principal Components Analysis, PCA) позволяет уменьшить размерность данных, потеряв наименьшее количество полезной информации. Метод имеет широкое применение в различных областях знаний, например, при визуализации Данных

MultivariateStats предоставляет инструменты для анализа данных с использованием метода главных компонент.

```
In [120]: # Подключение пакета MultivariateStats:  
import Pkg  
Pkg.add("MultivariateStats")
```

```
Resolving package versions...  
No Changes to `C:\Users\Scorpion 1.0\.julia\environments\v1.9\Project.toml`  
No Changes to `C:\Users\Scorpion 1.0\.julia\environments\v1.9\Manifest.toml`
```

```
In [124]: using MultivariateStats  
using Plots
```

```
In [125]: # Приведение типов данных к распределению для PCA:  
M = fit(PCA, F)
```

```
Out[125]: PCA(indim = 2, outdim = 1, principalratio = 0.9999840784692097)
```

Pattern matrix (unstandardized loadings):

	PC1
1	460.52
2	1.19826e5

Importance of components:

В ходе полученные навыки в специализированных пакетов Julia для обработки данных.