MOwNiT

Laboratorium 2 - Analiza danych DataFrames

In [6]:
```
using LinearAlgebra
using DataFrames
using BenchmarkTools
using CSV
using Plots
using Statistics
```

Zad1

In [7]:
```
function dot_product(vector_1, vector_2)
    return dot(vector_1, vector_2)
end

function multiply_sqr_matrix_by_vector(sqr_matrix, vector)
    return sqr_matrix * vector'
end
```

Out[7]:
```
multiply_sqr_matrix_by_vector (generic function with 1 method)
```

Zad2

In [37]:
```
dot_product_times = []
matrix_multiply_times = []

for i=100:100:10000
    vector = rand(Int32, (1,i))
    sqr_matrix = rand(Int32, (i,i))
    push!(dot_product_times, [])
    push!(matrix_multiply_times, [])
    for j=1:10
        push!(dot_product_times[i÷100], @timed dot_product(vector, vector))
        push!(matrix_multiply_times[i÷100], @timed multiply_sqr_matrix_by_vector(sqr_matrix, vector))
    end
end
```

In [70]:
```
df_times = DataFrame()
matrix_multiply_times[90][1]
df_times[:, :Rozmiar] = [i for i=100:100:10000 for j=1:10]
df_times[:, :DotProductTimes] = [dot_product_times[i][j].time for i=1:100 for j=1:10]
df_times[:, :MatrixMultiplicationTimes] = [matrix_multiply_times[i][j].time for i=1:100 for j=1:10]
show(df_times)
```

1000×3 DataFrame

Row	Rozmiar	DotProductTimes	MatrixMultiplicationTimes
	Int64	Float64	Float64
1	100	3.01e-7	1.9e-5
2	100	2.0e-7	1.12e-5
3	100	1.0e-7	1.1e-5
4	100	1.0e-7	1.11e-5
5	100	1.0e-7	1.1e-5
6	100	1.0e-7	1.12e-5
7	100	9.9e-8	1.25e-5
8	100	1.0e-7	1.1099e-5
9	100	0.0	1.11e-5
10	100	1.0e-7	1.1e-5
11	200	2.0e-7	4.4099e-5
⋮	⋮	⋮	⋮
991	10000	6.599e-6	0.362908
992	10000	7.101e-6	0.357223
993	10000	6.401e-6	0.34907
994	10000	5.8e-6	0.349905
995	10000	6.899e-6	0.364376
996	10000	6.099e-6	0.358263
997	10000	6.9e-6	0.357258
998	10000	6.7e-6	0.353786
999	10000	5.999e-6	0.36177
1000	10000	6.8e-6	0.362138

979 rows omitted

Zad3 Save to file

In [71]:
```
CSV.write("TimeTests.csv", df_times)
```

Out[71]:
```
"TimeTests.csv"
```

Zad4 Read file to DataFrame

In [72]:
```
my_data_frame = CSV.read("TimeTests.csv", delim=",", DataFrame)
```

Out[72]: 1,000 rows × 3 columns

	Rozmiar	DotProductTimes	MatrixMultiplicationTimes
	Int64	Float64	Float64
1	100	3.01e-7	1.9e-5
2	100	2.0e-7	1.12e-5
3	100	1.0e-7	1.1e-5
4	100	1.0e-7	1.11e-5
5	100	1.0e-7	1.1e-5
6	100	1.0e-7	1.12e-5
7	100	9.9e-8	1.25e-5
8	100	1.0e-7	1.1099e-5
9	100	0.0	1.11e-5
10	100	1.0e-7	1.1e-5
11	200	2.0e-7	4.4099e-5
12	200	2.0e-7	4.3399e-5
13	200	1.0e-7	4.3399e-5
14	200	1.0e-7	4.36e-5
15	200	1.0e-7	4.3399e-5
16	200	2.0e-7	4.35e-5
17	200	1.0e-7	4.3299e-5
18	200	1.0e-7	4.3299e-5
19	200	2.0e-7	4.3299e-5
20	200	1.0e-7	4.3499e-5
21	300	2.01e-7	0.0001023
22	300	2.01e-7	9.7801e-5
23	300	1.99e-7	9.6601e-5
24	300	2.0e-7	9.8401e-5
25	300	2.0e-7	9.6699e-5
26	300	2.0e-7	9.65e-5
27	300	2.01e-7	9.81e-5
28	300	2.0e-7	9.6701e-5
29	300	2.0e-7	9.6701e-5
30	300	2.0e-7	0.000105701
⋮	⋮	⋮	⋮

In [73]:
```
my_data_frame_grouped_by_size = groupby(my_data_frame, :Rozmiar)
```

Out[73]: **GroupedDataFrame with 100 groups based on key: Rozmiar**

First Group (10 rows): Rozmiar = 100

	Rozmiar	DotProductTimes	MatrixMultiplicationTimes
	Int64	Float64	Float64

1	100	3.01e-7	1.9e-5
2	100	2.0e-7	1.12e-5
3	100	1.0e-7	1.1e-5
4	100	1.0e-7	1.11e-5
5	100	1.0e-7	1.1e-5
6	100	1.0e-7	1.12e-5
7	100	9.9e-8	1.25e-5
8	100	1.0e-7	1.1099e-5
9	100	0.0	1.11e-5
10	100	1.0e-7	1.1e-5

⋮

Last Group (10 rows): Rozmiar = 10000

	Rozmiar	DotProductTimes	MatrixMultiplicationTimes
	Int64	Float64	Float64
1	10000	6.599e-6	0.362908
2	10000	7.101e-6	0.357223
3	10000	6.401e-6	0.34907
4	10000	5.8e-6	0.349905
5	10000	6.899e-6	0.364376
6	10000	6.099e-6	0.358263
7	10000	6.9e-6	0.357258
8	10000	6.7e-6	0.353786
9	10000	5.999e-6	0.36177
10	10000	6.8e-6	0.362138

Zad5

In [74]:
```
df_mean_times = combine(my_data_frame_grouped_by_size, "DotProductTimes"=>mean, "MatrixMultiplicationTimes"=>mean
 "DotProductTimes"=> std, "MatrixMultiplicationTimes"=>std)
```

Out[74]: 100 rows × 5 columns (omitted printing of 1 columns)

	Rozmiar	DotProductTimes_mean	MatrixMultiplicationTimes_mean	DotProductTimes_std
	Int64	Float64	Float64	Float64
1	100	1.2e-7	1.20199e-5	7.91637e-8
2	200	1.4e-7	4.34792e-5	5.16398e-8
3	300	2.002e-7	9.85505e-5	6.32456e-10
4	400	2.403e-7	0.00018122	5.13832e-8
5	500	3.0e-7	0.00027766	4.69089e-8
6	600	4.006e-7	0.00055604	1.15663e-7
7	700	4.202e-7	0.00055325	6.35292e-8
8	800	4.902e-7	0.00073097	8.77152e-8
9	900	5.701e-7	0.00093931	2.21676e-7
10	1000	6.299e-7	0.00133448	1.56584e-7
11	1100	6.899e-7	0.00167511	1.28575e-7
12	1200	8.003e-7	0.00216206	2.15974e-7
13	1300	8.602e-7	0.00234777	1.71452e-7
14	1400	8.799e-7	0.00317893	1.47332e-7
15	1500	1.1998e-6	0.00423752	6.82984e-7
16	1600	1.1502e-6	0.0062214	2.22262e-7
17	1700	1.1602e-6	0.00618735	2.91312e-7
18	1800	1.1803e-6	0.00737922	1.93483e-7
19	1900	1.3004e-6	0.00953796	2.15922e-7
20	2000	1.2499e-6	0.00981966	1.78108e-7

21	2100	1.56e-6	0.011053	4.92612e-7
22	2200	1.6497e-6	0.0120976	6.0208e-7
23	2300	1.7404e-6	0.0157251	3.72034e-7
24	2400	1.5001e-6	0.0150674	1.48626e-7
25	2500	1.5401e-6	0.0144167	1.7131e-7
26	2600	1.6403e-6	0.0145402	2.00987e-7
27	2700	1.6598e-6	0.0167451	2.7167e-7
28	2800	1.8603e-6	0.0209466	2.71875e-7
29	2900	2.7296e-6	0.0272873	2.0973e-6
30	3000	3.6101e-6	0.026904	4.97887e-6
⋮	⋮	⋮	⋮	⋮

Zad6

In [75]:
```
p1 = plot(df_mean_times.Rozmiar, df_mean_times.DotProductTimes_mean, colour = :blue,
        title="Dot product mean",label="",yerr=df_mean_times.DotProductTimes_std,
        seriestype=:scatter, layout = 1)

p2 = plot(df_mean_times.Rozmiar, df_mean_times.MatrixMultiplicationTimes_mean, colour = :blue,
        title="Matrix multiplication mean",label="",yerr=df_mean_times.MatrixMultiplicationTimes_std,seriestype=:
        layout = 1)

plot(p1, p2, ylabel = ["time [s]" "time [s]"],xlabel = ["size" "size"],layout = (1,2))
```

Out[75]: