Julia

Is a relative new language (introduced in 2012).
Ultimately may be a replacement for both matlab and R.

- Efficient and fast
- Replace matlab? Yes
- Replace R?
  Lots of tools for data handling,
  but still not up to R's level for advanced econometrics.

Julia as a matlab replacement

- Julia – can easily replace matlab if you are using the basic language
- Starting from fresh:
  - Go straight to Julia
- If you are thinking of switching to Julia from Matlab
  - Warning: There are some annoying differenes in syntax.
  - If you are a power user of matlab i a finance setting: Missing some of the toolboxes developed for finance

Bond pricing

```
C = [80 1080; 100 0 ]
B = [982.5;90]
d = inv(C)*B

println("C=",C)
println("B=",B)
println("d=",d)
```

```
C=[80 1080; 100 0]
B=[982.5, 90.0]
d=[0.9, 0.8430555555555556]
```

Bond pricing with term structure

```
r = [5.0 5.4 5.7 5.9 6.0]/100.0
t = [1 2 3 4 5]
R = r.+1
#note difference from matlab when adding scalars to matrix
d = R.^(-t)
println("d = ", d)
f = r
for i=2:5
    f[i] = ((1+r[i])^i)/((1+r[i-1])^(i-1))-1
end
println("f=",f)

C=[5.0 105.0 0.0 0.0 0.0]
b=C*d’
println("b = ",b)
```

Bond pricing with term structure ctd

```
b=[5 5 5 5 105]*d’
println("b = ",b)

b=[10 10 10 10 110]*d’
println("b = ",b)

C1=[5 5 5 5 105]
B1=C1*d’
Dur1 = (1.0/B1[]) * (t.*C1)*d’
println("B1   = ", B1[])
println("Dur1 = ", Dur1[])

C2 = [10 10 10 10 110]
B2 = C2*d’
Dur2 =  1.0/B2[] * (t.*C2)*d’
println("B2   = ", B2[])
println("Dur2 = ", Dur2[])
```

Bond pricing with term structure ctd

```
d = [0.9523809523809523 0.900158067756698 0.8467888669093383
f=[0.05 0.05801523809523812 0.05497244544621038 0.071175154
b = [99.27850187635805]
b = [95.93419539138128]
b = [117.14257349615684]
B1   = 95.93419539138128
Dur1 = 4.530998041254895
B2   = 117.14257349615684
Dur2 = 4.231819411058618
```

Linear Algebra example

```
using LinearAlgebra
a = 1
b = 2
c = 3
y = [1;2;3]
x = [1 2 3]

A = [1 2 3;4 5 6]

B = [c x]

#C = [A;x]

# D = [A y]
println("a+b = ", a + b)
println("b+c = ", b + c)
```

Linear Algebra example

```
x=[1 2 3 4 ]
 y=[4 3 2 1]
 x+y
 y-x
println("ax+by =", a*x + b*y)

A=[1 2 3; 4 5 6]
B=[6 5 4; 3 2 1]
A+B
A-B
a*A + b*B

A = [1 2 3;4 5 6]
B = [1 2;3 4; 5 6]
A*B
B*A
```

Linear Algebra example

```
B=[1 2;3 4]
# A*B  # not defined
B*A

A
A’

null = zeros(3,3)
#ident = eye(3,3)
ident = one(null)
#ident = Matrix(I,3,3)
ident

rank(A)
```

Linear Algebra example

```
D = [3 4;4 6]
D^(-1)
D * D^(-1)

F = [0.5 0.5; 0.5 0.5]
F*F

det(B)

A= [ 1  2  3;  3  2  1;  1  1  1]
det(A)
rank(A)
#A^(-1)
```

Linear Algebra example

```
A = [3 4;4 6]
b=[5;8]
rank(A)
rank([A b])
A^(-1)
x = A^(-1) * b
x = A\b

A = [1 1 ; 1 1]
A^2
A.^2
```

Doing a plot

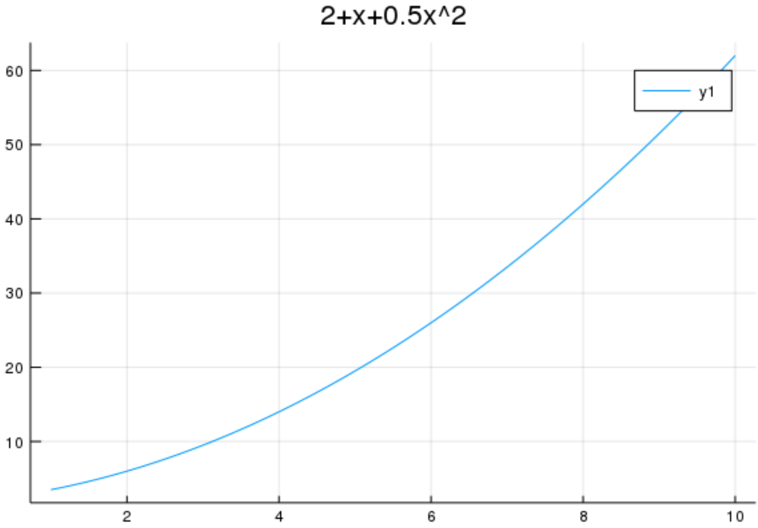
Plotting the function

$$y = 2 + x + \frac{1}{2}x^2$$

Julia Prog:

```
outdir = "../../results/2019_02_plot/"
using Plots
x = collect(1.0:0.1:10.0)
y = 2.0 .+ x
y = y + 0.5 .* x.^2;
p = plot(x,y,title="2+x+0.5x^2")
filename = join((outdir,"plot_square_fct.png"),"")
savefig(filename)
```

Doing a plot


2+x+0.5x^2
60
50
40
30
20
10
2
4
6
8
10
y1