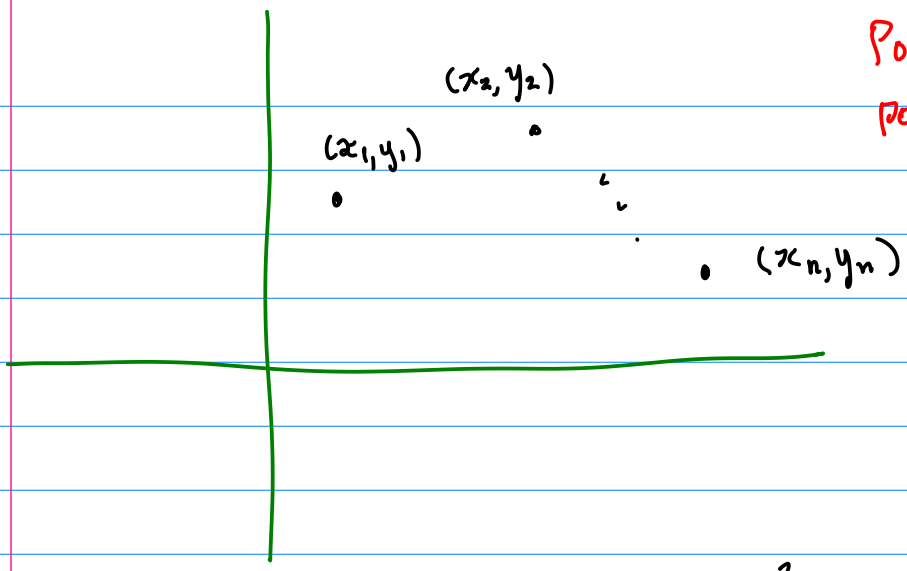
Polynomial of degree $n-1$ passing through n points

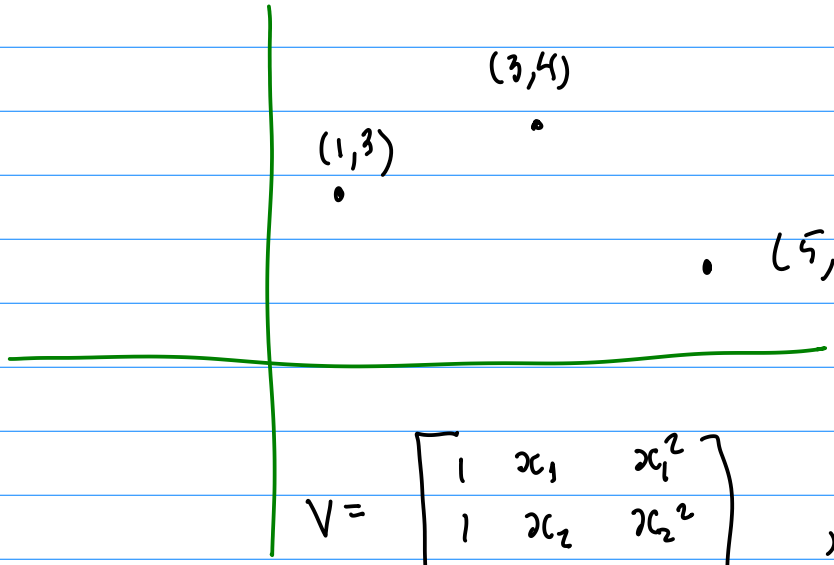


$$p(x_i) = y_i \quad \text{for } i = 1, \dots, n$$

$$p(t) = a_0 + a_1 t + a_2 t^2 + \cdots + a_{n-1} t^{n-1}$$

$$\begin{aligned}
a_0 + a_1 x_1 + a_2 x_1^2 + \cdots + a_{n-1} x_1^{n-1} &= y_1 \\
a_0 + a_1 x_2 + a_2 x_2^2 + \cdots + a_{n-1} x_2^{n-1} &= y_2 \\
&\vdots \\
a_0 + a_1 x_n + a_2 x_n^2 + \cdots + a_{n-1} x_n^{n-1} &= y_n
\end{aligned}$$

$$\begin{bmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^{n-1} \\ 1 & x_2 & x_2^2 & & x_2^{n-1} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_n & x_n^2 & & x_n^{n-1} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_{n-1} \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$



$$p(t) = a_0 + a_1 t + a_2 t^2$$

$$V = \begin{bmatrix} 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \\ 1 & x_3 & x_3^2 \end{bmatrix}, \quad c = \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix}, \quad y = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}$$

↑ solve for c

$$V = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 3 & 9 \\ 1 & 5 & 25 \end{bmatrix} \qquad y = \begin{bmatrix} 3 \\ 4 \\ 1 \end{bmatrix}$$

Solve this $Vc = y$

```
julia> V=[1 1 1; 1 3 9; 1 5 25]
3×3 Matrix{Int64}:
 1  1   1
 1  3   9
 1  5  25
```

semicolon separates rows

```
julia> y=[3,4,1]
3-element Vector{Int64}:
 3
 4
 1
```

Comma used for a column vector

Left (matrix) division "\"

← backslash (above enter key)

$$Vc = y$$

$$c = V \backslash y$$

```
julia> c=V\y
3-element Vector{Float64}:
  1.0
  2.5
 -0.5
```

Use Gaussian elimination, QR factorization or singular value decomposition to solve for c

$$c = \begin{bmatrix} 1 \\ 2.5 \\ -0.5 \end{bmatrix}$$

$a_0 = 1$

$a_1 = 2.5$

$a_2 = -0.5$

$p(t) = 1 + 2.5t - 0.5t^2$

Plot the points and check that p(t) passes through them:

$$x = \begin{bmatrix} 1 \\ 3 \\ 5 \end{bmatrix} \qquad y = \begin{bmatrix} 3 \\ 4 \\ 1 \end{bmatrix}$$

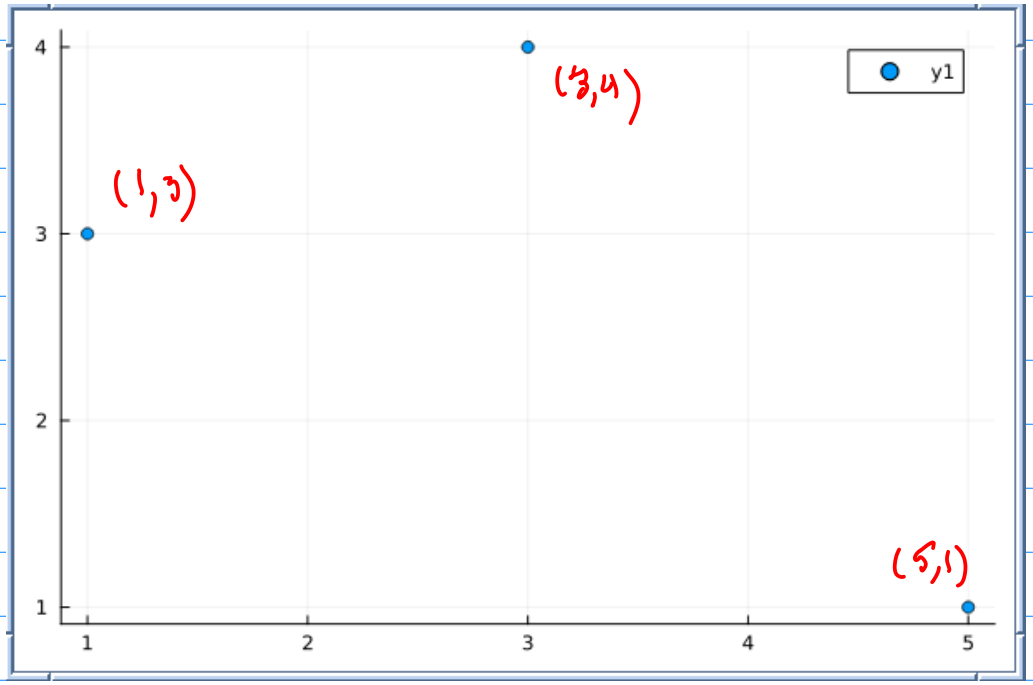
```
julia> x=[1,3,5]
3-element Vector{Int64}:
 1
 3
 5
```

Load plotting library

```
julia> using Plots
```

Plot points

```
julia> scatter(x,y)
```



$p(t) = 1 + 2.5t - 0.5t^2$

Nested multiplication for evaluating a polynomial

mult mult

$p(t) = 1 + 2.5t - 0.5t^2$ ← mult

3 multiplications before

$= 1 + (2.5 - 0.5t)t$

mult mult

2 multiplications

Easy way to reduce computational effort and reduce rounding errors...

Most compilers don't perform transformation to nested multiplication automatically because it changes the answer (slight by rounding differences).

```
julia> p(t)=1+(2.5-0.5*t)*t
p (generic function with 1 method)
```

Overlay the graph of p(t) on the points...

```
julia> ts=0:0.01:6
0.0:0.01:6.0
```

← range of points equally spaced a distance 0.01 apart from 0 to 6

```
julia> ts[1]
0.0

julia> ts[2]
0.01

julia> ts[3]
0.02
```

← an equally spaced grid...

modify or add to the previous plot

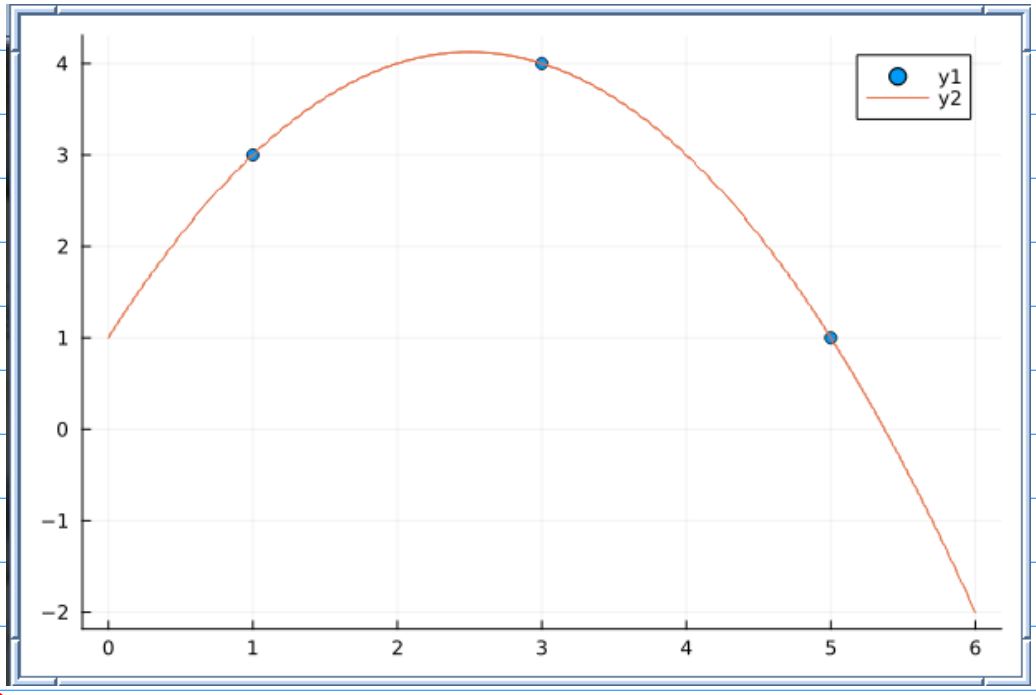
means this is not a pure function but the state of the graph changes

```
julia> plot!(ts,p.(ts))
```

In Julia the "." is called a broadcast

apply p pointwise to each entry of ts.

x values on graph



missing the broadcast ...

```
julia> p(ts)
ERROR: MethodError: no method matching -(::Float64, ::StepRangeLen{Float64, Base
.TwicePrecision{Float64}, Base.TwicePrecision{Float64}, Int64})
For element-wise subtraction, use broadcasting with dot syntax: scalar .- array

Closest candidates are:
  -(::Real, ::Complex{Bool})
   @ Base complex.jl:321
  -(::Number, ::AbstractGray{Bool})
   @ ColorVectorSpace ~/.julia/packages/ColorVectorSpace/JXxVe/src/ColorVectorSp
ace.jl:331
  -(::Number, ::AbstractGray)
   @ ColorVectorSpace ~/.julia/packages/ColorVectorSpace/JXxVe/src/ColorVectorSp
ace.jl:329
  ...

Stacktrace:
 [1] p(t::StepRangeLen{Float64, Base.TwicePrecision{Float64}, Base.TwicePrecisio
n{Float64}, Int64})
   @ Main ./REPL[7]:1
 [2] top-level scope
   @ REPL[13]:1
```

error happend when trying to evaluate p.

In linear algebra matrix multiplication means composition of linear functions.

$$f(x) = Ax \qquad g(x) = Bx$$

$$(f \circ g)(x) = f(g(x)) = f(Bx) = A(Bx) = (AB)x$$

matrix mult

Since it makes a difference whether a matrix is multiplies on the left or right, .. that's why Julia has a left matrix division "\"
and a right matrix division "/"

$$Vc = y$$

$$c = V \backslash y$$ left matrix division

$$(Vc)^T = (y)^T$$

$$c^T V^T = y^T$$ by right matrix division I can solve for c^T

$$c^T = y^T / V^T$$

```
julia> yt=y'
1×3 adjoint(::Vector{Int64}) with eltype Int64:
 3  4  1

julia> Vt=V'
3×3 adjoint(::Matrix{Int64}) with eltype Int64:
 1  1   1
 1  3   5
 1  9  25
```

transpose

transpose...

$$c^T = y^T / V^T$$

```
julia> ct=yt/Vt
1×3 adjoint(::Vector{Float64}) with eltype Float64:
 1.0  2.5  -0.5
```

$a_0 = 1 \quad a_1 = 2.5 \quad a_2 = -0.5$

same polynomial as before...

Next time inner and outer products and representation of matrix products.