# 11.11

## Given

$$f(t) = (\frac{c_1+c_2t+c_3t^2}{1+d_1t+d_2t^2})$$

let,
f(t) = y
$\therefore y + d_1ty + d_2t^2y = c_1 + c_2t + c_3t^2$

$y = c_1 + c_2t + c_3t^2 - d_1ty - d_2t^2y$

Now we use the given interpolation conditions :

### For f(1)=2,

$c_1 + c_2 + c_3 - 2d_1 - 2d_2 = 2$ ..... (i)

### For f(2)=5,

$c_1 + 2c_2 + 4c_3 - 10d_1 - 20d_2 = 5$ ..... (ii)

### For f(3)=9,

$c_1 + 3c_2 + 9c_3 - 27d_1 - 81d_2 = 9$ ..... (iii)

### For f(4)=-1,

$c_1 + 4c_2 + 16c_3 + 4d_1 + 16d_2 = -1$ ..... (iv)

### For f(5)=-4,

$c_1 + 5c_2 + 25c_3 + 20d_1 + 100d_2 = -4$ ..... (v)

### Equations (i),(ii),(iii),(iv) and (v) can be written as a set of linear equations in 5 variables :

$$\begin{pmatrix} 1 & 1 & 1 & -2 & -2 \\ 1 & 2 & 4 & -10 & -20 \\ 1 & 3 & 9 & -27 & -81 \\ 1 & 4 & 16 & 4 & 16 \\ 1 & 5 & 25 & 20 & 100 \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \\ c_3 \\ d_1 \\ d_2 \end{pmatrix} = \begin{pmatrix} 2 \\ 5 \\ 9 \\ -1 \\ -4 \end{pmatrix}$$

In [24]:

```
# Here we consider the above set of linear equations in the form AX = B, so we define
A = [1 1 1 -2 -2 ; 1 2 4 -10 -20; 1 3 9 -27 -81; 1 4 16 4 16; 1 5 25 20 100]
display(A)
```

```
B = [2 ; 5 ; 9 ; -1 ; -4]
display(B)
```

```
5×5 Matrix{Int64}:
 1  1   1   -2   -2
 1  2   4  -10  -20
 1  3   9  -27  -81
 1  4  16    4   16
 1  5  25   20  100
5-element Vector{Int64}:
  2
  5
  9
 -1
 -4
```

In [25]:

```
# Thus we find the required value of X
X = A\B
display(X)
```

```
5-element Vector{Float64}:
  0.6296296296296255
  0.6049382716049432
 -0.19753086419753174
 -0.5679012345679012
  0.08641975308641975
```

### Solving the set of linear equations we got :

$c_1 = 0.6296296296296255$
$c_2 = 0.6049382716049432$
$c_3 = -0.19753086419753174$
$d_1 = -0.5679012345679012$
$d_2 = 0.08641975308641975$

# Now we plot the function f(t) for t $\in$ [0,6]

In [26]:

```
using Pkg
Pkg.add("Plots")
Pkg.add("CalculusWithJulia")
using CalculusWithJulia
using Plots
```

```
   Resolving package versions...
  No Changes to `C:\Users\Malvika\.julia\environments\v1.6\Project.toml`
  No Changes to `C:\Users\Malvika\.julia\environments\v1.6\Manifest.toml`
```

In [27]:

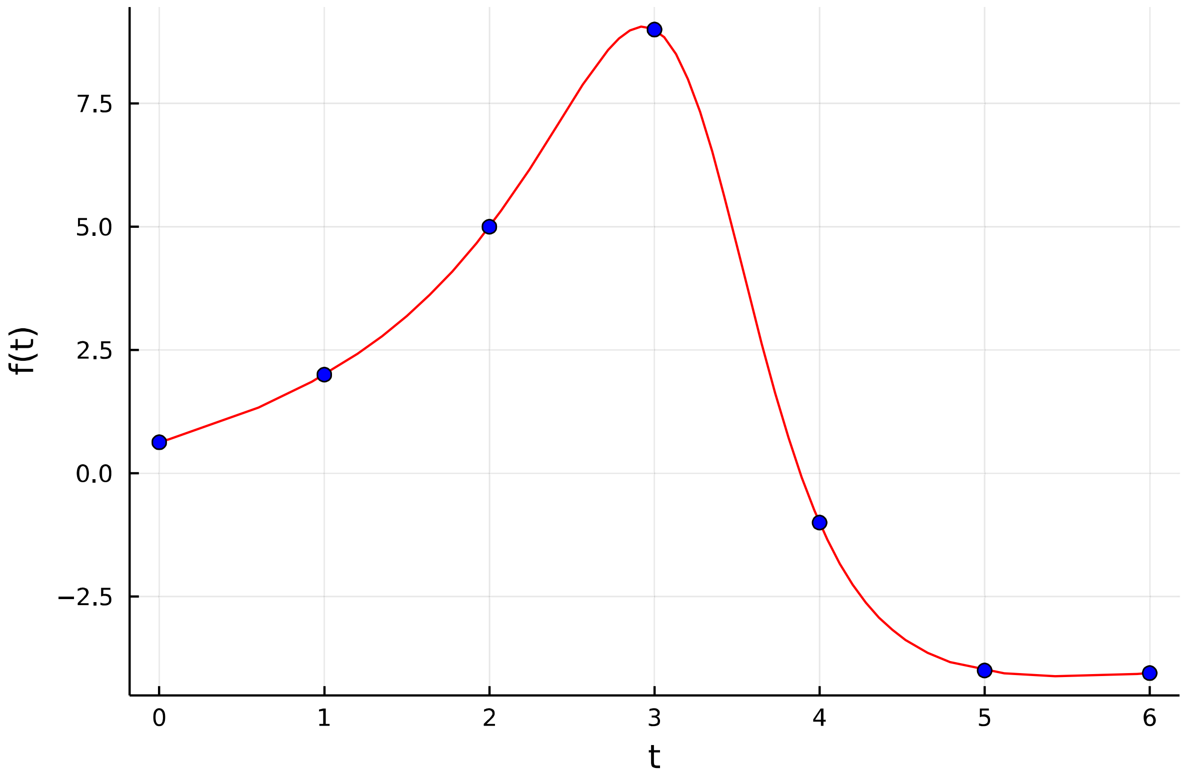
```
c_1 = 0.6296296296296255
c_2 = 0.6049382716049432
c_3 = -0.19753086419753174
d_1 = -0.5679012345679012
d_2 = 0.08641975308641975
```

Out[27]:

```
0.08641975308641975
```

In [28]:
```
f(t) = (c_1 + (c_2 * t) + (c_3 * (t^2)))/(1 + (d_1 * t) + (d_2 * (t^2)))
Plots.plot(f, 0, 6,xlabel="t",ylabel="f(t)",c = "red",legend = false)
scatter!(0:6,f, c = "blue", legend = false)
```

Out[28]:




# 11.21

## We calculate inverse of matrix A using Julia

### The transpose of given matrix is a vandermonde matrix hence we know that the given matrix A is invertible.

In [58]:
```
A = [1 1 1 1 ; -0.6 -0.2 0.2 0.6 ; 0.36 0.04 0.04 0.36 ; -0.216 -0.008 0.008 0.216]
A_inv = inv(A)
p = [2 ; 0 ; 2/3 ; 0]
display(A_inv)
```

```
4×4 Matrix{Float64}:
 -0.0625   0.104167   1.5625  -2.60417
  0.5625  -2.8125    -1.5625   7.8125
  0.5625   2.8125    -1.5625  -7.8125
 -0.0625  -0.104167   1.5625   2.60417
```

In [59]:
```
w = A_inv * p
```

Out[59]:
```
4-element Vector{Float64}:
 0.9166666666666667
 0.08333333333333481
 0.0833333333333326
 0.9166666666666667
```