```
using Plots
```

```
data =  Int64[1,  0,  1,  1,  0,  1,  2,  1,  5,  4,  3,  3,  4,  3,  17,  6,  6,  11,  18,  15
```

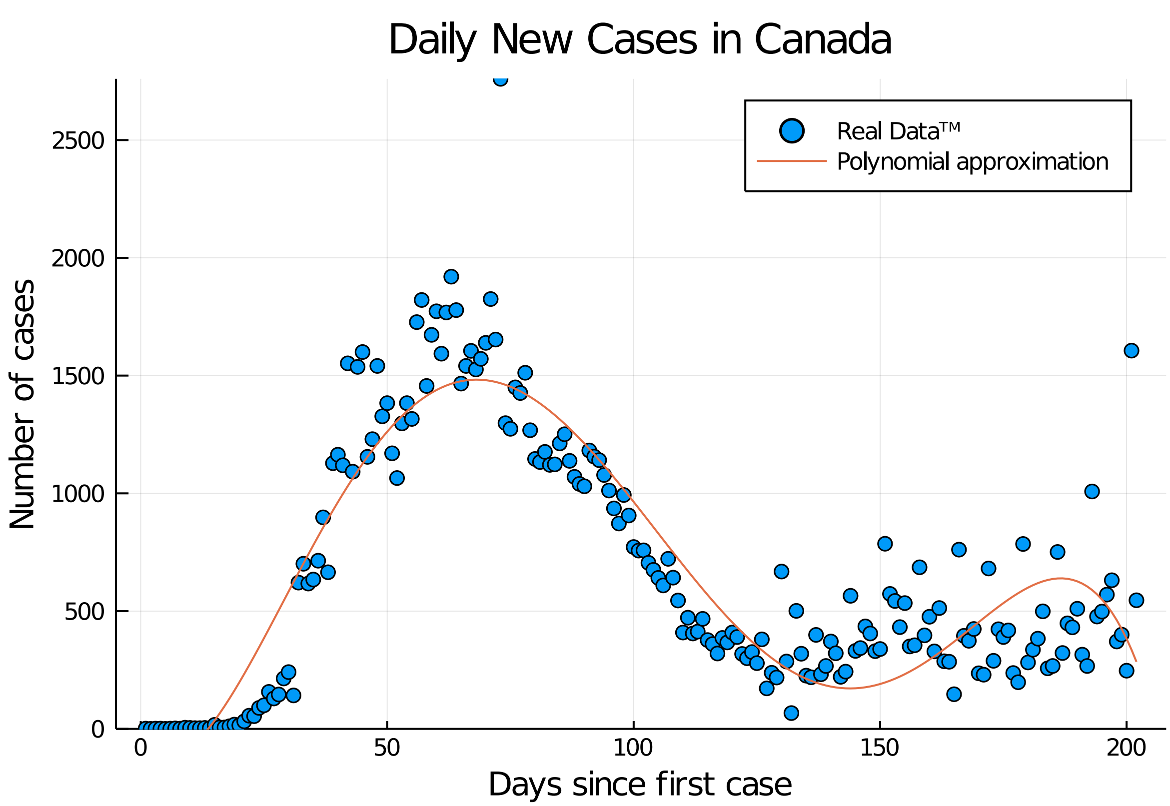
```
data =
[1,0,1,1,0,1,2,1,5,4,3,3,4,3,17,6,6,11,18,15,32,56,54,89,100,157,129,146,214,241,142,621,701,617,634,714
,898,665,1128,1164,1119,1552,1092,1537,1600,1155,1230,1541,1327,1383,1170,1065,1297,1383,1316,1727,1821,
1456,1673,1773,1593,1768,1920,1778,1466,1541,1605,1526,1571,1639,1825,1653,2760,1298,1274,1450,1426,1512
,1268,1146,1133,1176,1121,1123,1212,1251,1138,1070,1040,1030,1182,1156,1141,1078,1012,936,872,993,906,77
2,757,758,705,675,641,609,722,642,545,409,472,405,413,467,377,360,320,386,367,409,390,318,300,326,279,38
0,172,238,218,668,286,67,501,319,226,219,399,232,267,371,321,221,243,565,331,343,435,405,330,339,786,573
,543,432,534,350,355,686,397,476,329,513,287,285,147,761,395,374,424,236,230,681,289,423,390,418,237,198
,785,282,336,383,499,257,267,751,322,448,431,510,315,267,1008,477,498,570,631,371,400,247,1606,546]
```

```
N = 202
```

```
N = length(data)
```

```
Float64[-1.0,  -0.99005,  -0.9801,  -0.970149,  -0.960199,  -0.950249,  -0.940299,  -0.93
```

```
begin
    days = 1:N
    scaled_days = -1 .+ 2/(N-1).*(days.-1)
end
```




```
begin
    scatter(days, data;label="Real Data™")
    ylims!(extrema(data))
```

```
    plot!(days, map(x->horner(x,c), scaled_days); label="Polynomial approximation")
    xlabel!("Days since first case")
    ylabel!("Number of cases")
    title!("Daily New Cases in Canada")
end
```

```
@bind deg html"""<input type="range" min="0" max="201" value=5>"""
```

5

```
deg
```

Float64[923.939, -2688.18, -665.863, 7569.68, -171.545, -4683.32]

```
begin
    A = scaled_days.^(0:deg)'
    c = A\data
end
```

f = #3 (generic function with 1 method)

```
f = x-> c[1] + x*(c[2] + x*(c[3] + x*(c[4] + x*(c[5] + x*c[6])))) # A degree-5 polynomial with
coefficients `c` evaluated at `x`.
```

horner (generic function with 1 method)

```
function horner(x, c)
    N = length(c)
    ret = c[N]
    for k in N-1:-1:1
        ret = x*ret+c[k]
    end
    return ret
end
```

Linear algebra is all about $Ax = b$.

Calculus is all about $f(x)$, $f'(x)$ and $\int_a^b f(x)\,\mathrm{d}x$

```
using LinearAlgebra
```

For least-squares problems, the norm of the residual, $r = b - Ax$, is not usually small. But if we multiply it by A^*, then it is near machine precision multiplied by the condition number of the matrix and the norm of the right-hand side.

3427.395858285415

```
norm(A*c-data) # Pretty huge
```

true

```
norm(A'*(A*c-data)) ≤ 2*eps()*cond(A)*norm(data)
```