```julia
using Plots
```

```julia
data =
```

[1, 0, 1, 1, 0, 1, 2, 1, 5, 4, 3, 3, 4, 3, 17, 6, 6, 11, 18, 15,   more ,6444, 2906, 3856

```julia
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, 772, 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, 380, 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, 630, 702, 515, 518, 1351, 793,
944, 1120, 1044, 863, 875, 1766, 1248, 1090, 1341, 1362, 1215, 1454, 2176, 1660,
1797, 1777, 2124, 1812, 1685, 2804, 2363, 1800, 2436, 2558, 2062, 1685, 975, 4042,
2506, 2345, 2374, 2215, 1827, 3289, 2251, 2266, 3194, 2584, 2227, 2145, 4109, 2674,
2699, 2956, 3457, 2512, 2330, 3422, 4672, 2768, 3635, 4471, 4246, 4058, 4622, 4302,
4024, 5516, 4741, 4613, 4146, 6115, 4276, 4641, 4645, 4965, 4992, 6131, 5713, 4889,
5022, 5631, 5967, 6495, 6477, 6103, 5329, 6307, 6495, 6299, 6352, 6261, 7872, 5981,
6295, 6739, 6772, 6011, 5891, 8119, 6352, 6416, 7008, 6708, 6248, 6201, 7519, 6195,
6845, 6858, 1, 6403, 10404, 3187, 10299, 7476, 8413, 1302, 7583, 11383, 9761, 7222,
8154, 8334, 9214, 8125, 7816, 7892, 6292, 6855, 7563, 6816, 6476, 6436, 6453, 4679,
5744, 5955, 5957, 5124, 4852, 5628, 4011, 4205, 4876, 4690, 4255, 3924, 4617, 2831,
3231, 4083, 4022, 3301, 3203, 3860, 2677, 3185, 3181, 3143, 3047, 2432, 1139, 4653,
2605, 3315, 3089, 2715, 2351, 3865, 2752, 2857, 3094, 3252, 2724, 2307, 3530, 2714,
2812, 2832, 3370, 2325, 2488, 4124, 2820, 3221, 3018, 3476, 2968, 2956, 3890, 2821,
3371, 3609, 4216, 3452, 3269, 4934, 3601, 4050, 5192, 5093, 5364, 5126, 4573, 4880,
5518, 5802, 2686, 11041, 2343, 10386, 6519, 7148, 7982, 9255, 7261, 7619, 10858,
7546, 8590, 9564, 9346, 7845, 7591, 10275, 7270, 8421, 8370, 8753, 7417, 6983,
8931, 7071, 7748, 8346, 8342, 7610, 7146, 9076, 6693, 7378, 7992, 7849, 6802, 6695,
7520, 5386, 6198, 6644, 5991, 5276, 4901, 5526, 4033, 4247, 5057, 4676, 3644, 3415,
2384, 3952, 2590, 2967, 3202, 2459, 2237, 2611, 1633, 2063, 2167, 2063, 1666, 1389,
1583, 1264, 1388, 1477, 1441, 1290, 1122, 1157, 808, 1053, 1107, 1016, 854, 713,
771, 599, 721, 707, 592, 542, 435, 931, 602, 548, 26, 1007, 344, 308, 670, 441,
552, 564, 768, 314, 253, 577, 339, 384, 415, 395, 277, 259, 701, 342, 495, 382,
605, 285, 228, 1155, 547, 766, 896, 904, 342, 279, 501, 2177, 955, 1445, 1519, 518,
524, 3344, 1346, 1872, 2141, 2405, 1141, 977, 4002, 1727, 2477, 2731, 2960, 1442,
1505, 4811, 2528, 3158, 3349, 3778, 1799, 1577, 6444, 2906, 3856, 4034, 4159, 4107,
4228, 2792, 3307, 3654]
```
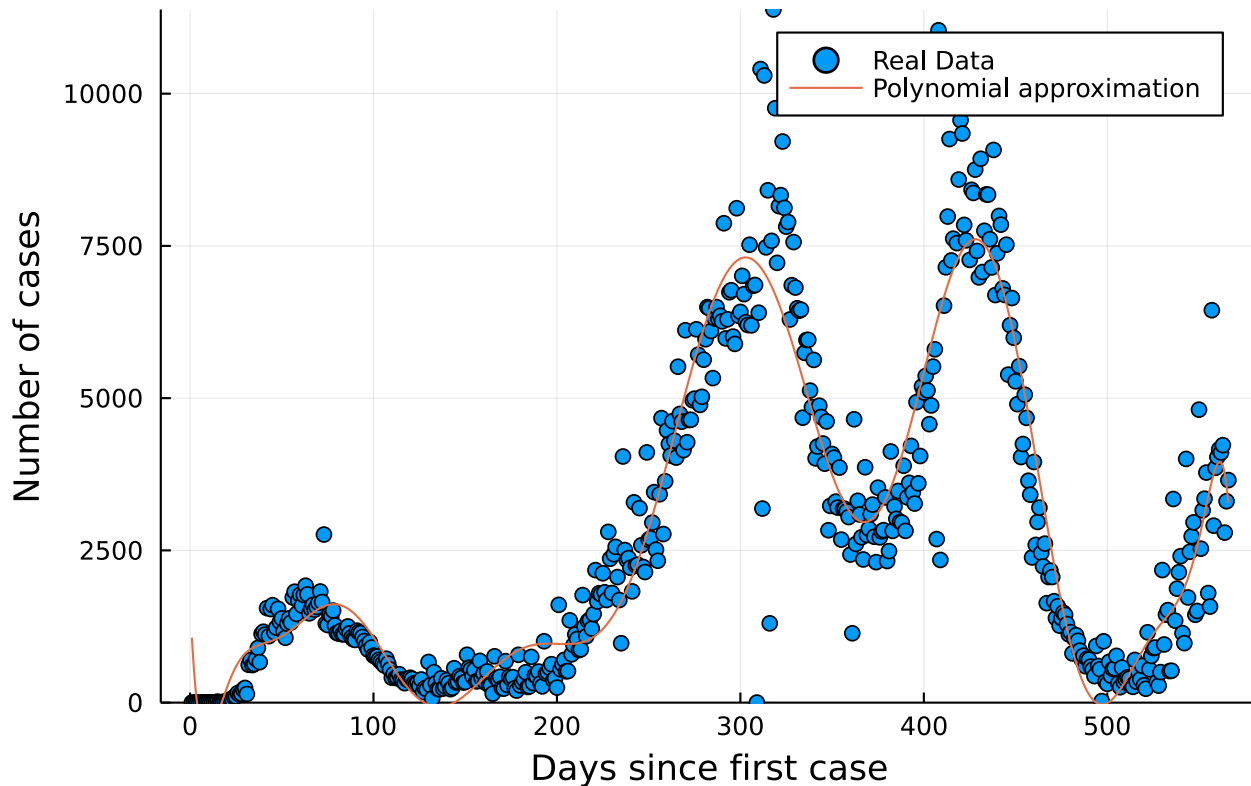
```
N = 566
```

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

```
-1.0:0.0035398230088495575:1.0
```

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

Daily New Cases in Canada


```
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="565" value="15">"""
```

```
15
```

```
· deg
```

```
[6450.9, 23689.3, -1.31499e5, -5.67376e5, 1.3278e6, 5.61654e6, -5.64353e6, -2.48165e7,
```

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

`f = #1 (generic function with 1 method)`

```
· f = x -> c[1] + x*(c[2] + x*(c[3] + x*(c[4] + x*(c[5] + x*(c[6] + x*
·   (c[7]+x*c[8])))))))   # A degree-7 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 num,ber of the matrix and the norm of the right-hand side.

`22104.023939898925`

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

`true`

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