# FEBRUARY 2021

Mathematica Notebook by Lee Altenberg

---

## DATA

```
may2020 = {X, X, X, X, X, X, X, X, X, X, X,
   X, X, X, X, X, X, X, X, X, X, X, X, X, X, X, 1, 3, 3, 3, 1};
```

In[901]:=
```
apr2020 // Dim
```

Out[901]=
```
{30}
```

In[897]:=
```
Map[Total, jun2020]
```

Out[897]=
```
{0, 1, 1, 2, 9, 9, 2, 1, 6, 4, 7, 15, 17, 5,
 8, 4, 4, 18, 27, 14, 11, 4, 3, 16, 17, 17, 6, 27, 2, 18}
```

In[917]:=
```
mar2020 = {0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 2, 3,
   3, 1, 5, 6, 17, 10, 10, 20, 14, 9, 17, 17, 35, 28, 31, 22, 31};
apr2020 = {28, 39, 22, 15, 18, 22, 21, 9, 11, 20, 14, 4, 6, 11,
   11, 9, 20, 8, 4, 2, 6, 5, 3, 2, 2, 1, 1, 4, 4, 1};
may2020 = {0, 3, 1, 3, 1, 2, 0, 2, 1, 2, 1, 3, 0, 0, 2, 1,
   0, 1, 1, 4, 0, 1, 0, 0, 0, 0, 2, 3, 3, 0, 0};
jun2020 = {0, 1, 1, 2, 9, 9, 2, 1, 6, 4, 7, 15, 17, 5, 8, 4, 4,
   18, 27, 14, 11, 4, 3, 16, 17, 17, 6, 27, 2, 18};
jul2020 = {9, 9, 9, 24, 25, 7, 41, 23, 36, 28, 42, 21, 23, 22, 29, 19,
   23, 20, 28, 12, 25, 17, 55, 60, 73, 64, 28, 47, 109, 124, 123};
aug2020 = { 87, 45, 207, 144, 173, 152, 201, 231, 152, 140,
   118, 202, 355, 233, 284, 220, 174, 134, 261, 236,
   230, 284, 244, 169, 215, 277, 306, 265, 310, 200, 133};
sep2020 = {181, 339, 211, 271, 221, 164, 105, 66, 100, 169, 167, 131, 114, 80,
   66, 102, 160, 114, 110, 77, 56, 63, 168, 90, 112, 127, 98, 90, 87, 121};
oct2020 = {108, 87, 133, 70, 52, 83, 110, 101, 155, 73,
   103, 42, 62, 101, 91, 89, 96, 83, 39, 91,
   78, 102, 131, 90, 121, 38, 66, 62, 77, 94, 68};
nov2020 = {83, 78, 89, 155, 100, 122, 128, 128, 64, 78,
   78, 97, 110, 108, (* 95 *) 53.5, 41.5 , 53, 71, 107, 95,
   163, 123, 114, 61, 108, 120, 92, 76, 57, 85};
dec2020 = {44, 63, 144, 106, 133, 105, 81, 53, 80, 123,
   89, 198, 90, 190, 57, 110, 142, 130, 156 , 204,
   134, 66, 107, 129, 120, 120, 95, 46, 76, 108, 188};
jan2021 = {241, 171, 149, 89, 124, 143, 322, 264, 250, 200,
   172, 114, 106, 179, 150, 165, 132, 129, 65, 75,
   119, 132, 134, 153, 123, 71, 103, 100, 115, 116,
   82};
feb2021 = {90};
sinceMar2020 = Flatten[{mar2020, apr2020, may2020, jun2020, jul2020,
    aug2020, sep2020, oct2020, nov2020, dec2020, jan2021, feb2021}];
sinceJune2020 = Flatten[{jun2020, jul2020, aug2020, sep2020,
    oct2020, nov2020, dec2020, jan2021, feb2021}];
dataN = Length[sinceJune2020];
dataAllN = Length[sinceMar2020];
```

In[786]:=
```
Dim@jan2021
```

Out[786]=
```
{30}
```

# TOOLS

In[817]:=
```
Tp = Transpose;
Dim = Dimensions;
Idm = IdentityMatrix;
MF = MatrixForm;
TF = TableForm;
FS = FullSimplify;
LP[ar_] := ListPlot[ar, PlotRange → All];
LP3[ar_] := ListPlot3D[ar, PlotRange → All];
LPJ[ar_] := ListPlot[ar, PlotRange → All, Joined → True];
MP[mat_] := MatrixPlot[mat];
specRad[A_] := Max[Abs[Eigenvalues[A]]];
eVec[n_] := ConstantArray[1, n];
iVec[i_, n_] := Table[If[j == i, 1, 0], {j, n}];
Lg[ar_] := Log[2, ar];
```

In[830]:=
```
upperCycleMatrix[n_] := Table[If[Mod[i + 1, n] == Mod[j, n], 1, 0], {i, n}, {j, n}];
lowerCycleMatrix[n_] := Table[If[Mod[i, n] == Mod[j - 1, n], 1, 0], {i, n}, {j, n}]
```

In[832]:=
```
runningMean[ar_, n_] :=
  Table[Mean@Take[ar, {i, i + n - 1}], {i, 1, Length[ar] - n + 1}];
runningSymMean[ar_, k_] := Block[{n = Length[ar]},
  Table[Mean@Take[ar, {i - Min[i - 1, n - i, Floor[k / 2]],
      i + Min[i - 1, n - i, Ceiling[k / 2]]}], {i, 1, n}]]
```

In[834]:=
```
rToRt[r_] := Exp[r / .29];
rToDoublingTime[r_] := Log[2.] / r
```

In[•]:=
```
cJP[ar_] := DateListPlot[TimeSeries[ar, {"Jun 1, 2020"}], PlotRange → All ]
```

In[1034]:=
```
dateAllPlot[ar_] := DateListPlot[TimeSeries[ar, {"Mar 1, 2020"}], PlotRange → All ]
```

In[•]:=
```
Optimization`NMinimizeDump`$Methods
```

Out[•]=
```
{Automatic, DifferentialEvolution, MeshSearch, NelderMead,
 SimulatedAnnealing, RandomSearch, NonlinearInteriorPoint}
```

# EXPONENTIAL FIT FUNCTIONS

In[836]:=
```
twoExpFunc[t_, date_, c1_, r1_, r2_] :=
 If[t < date, c1 Exp[t r1], c1 Exp[date r1] Exp[(t - date) r2] ]
```

In[837]:=
```
threeExpFunc[t_, date1_, date2_, c1_, r1_, r2_, r3_] :=
 ({d1, d2} = Sort[{date1, date2}];
  If[t < d1, c1 Exp[t r1],
   If[t < d2, c1 Exp[d1 r1] Exp[(t - d1) r2],
    c1 Exp[d1 r1] Exp[(d2 - d1) r2] Exp[(t - d2) r3] ]])
```

In[838]:=
```
fourExpFunc[t_, date1_, date2_, date3_, c1_, r1_, r2_, r3_, r4_] :=
 ({d1, d2, d3} = Sort[{date1, date2, date3}];
  If[t < d1, c1 Exp[t r1],
   If[t < d2, c1 Exp[d1 r1] Exp[(t - d1) r2],
    If[t < d3 ,
     c1 Exp[d1 r1] Exp[(d2 - d1) r2] Exp[(t - d2) r3],
     c1 Exp[d1 r1] Exp[(d2 - d1) r2] Exp[(d3 - d2) r3] Exp[(t - d3) r4] ]] ])
```

## 5 PIECE FITS

In[839]:=
```
fiveExpFunc[t_, date1_, date2_, date3_, date4_, c1_, r1_, r2_, r3_, r4_, r5_] :=
 ({d1, d2, d3, d4} = Sort[{date1, date2, date3, date4}];
  If[t < d1, c1 Exp[t r1],
   If[t < d2, c1 Exp[d1 r1] Exp[(t - d1) r2],
    If[t < d3 ,
     c1 Exp[d1 r1] Exp[(d2 - d1) r2] Exp[(t - d2) r3],
     If[t < d4,
      c1 Exp[d1 r1] Exp[(d2 - d1) r2] Exp[(d3 - d2) r3] Exp[(t - d3) r4], c1 Exp[d1 r1]
       Exp[(d2 - d1) r2] Exp[(d3 - d2) r3] Exp[(d4 - d3) r4 ] Exp[(t - d4) r5]]
    ] ]])
```

In[840]:=
```
fiveRtFunc[t_, date1_, date2_, date3_, date4_, c1_, r1_, r2_, r3_, r4_, r5_] :=
 ({d1, d2, d3, d4} = Sort[{date1, date2, date3, date4}];
  If[t < d1, rToRt[r1],
   If[t < d2, rToRt[r2],
    If[t < d3 ,
     rToRt[r3],
     If[t < d4,
      rToRt[ r4], rToRt[ r5]]
    ] ]])
```

## 5+1 PIECE FITS

In[841]:=
```
fiveExpMigFunc[t_, date1_, date2_, date3_,
  date4_, date5_, c1_, c2_, r1_, r2_, r3_, r4_, r5_] :=
 ({d1, d2, d3, d4, d5} = Sort[{date1, date2, date3, date4, date5}];
  If[t < d1, c1 Exp[t r1],
   If[t < d2, c1 Exp[d1 r1] Exp[(t - d1) r2],
    If[t < d3 ,
     c1 Exp[d1 r1] Exp[(d2 - d1) r2] Exp[(t - d2) r3],
     If[t < d4,
      c1 Exp[d1 r1] Exp[(d2 - d1) r2] Exp[(d3 - d2) r3] Exp[(t - d3) r4],
      If[t < d5, c1 Exp[d1 r1] Exp[(d2 - d1) r2]
          Exp[(d3 - d2) r3] Exp[(d4 - d3) r4 ] Exp[(t - d4) r5] ,
        c1 Exp[d1 r1] Exp[(d2 - d1) r2] Exp[(d3 - d2) r3] Exp[(d4 - d3) r4 ]
         Exp[(t - d4) r5]] + c2 (1 - Exp[(t - d5) r5]) / (1 - Exp[r5])]
    ] ]])
```

In[842]:=
```
period5Data = Take[sinceJune2020, {101, Length[sinceJune2020]}];
period5Cycle = Take[
   Flatten[Table[{w1, w2, w3, w4, w5, w6, w7}, Ceiling[Length[period5Data] / 7]]],
   Length[period5Data]];
```

## PLOTS

In[844]:=
```
triCalPlot[ar_] := DateListPlot[
  Map[TimeSeries[#, {"Jun 1, 2020"}] &, ar], PlotRange → All , Joined → True,
  PlotStyle → {Thickness[0.0051], Thickness[0.003], Thickness[0.01]} ]
```

In[937]:=
```
triCalPlotAll[ar_] := DateListPlot[
  Map[TimeSeries[#, {"Mar 1, 2020"}] &, ar], PlotRange → All , Joined → True,
  PlotStyle → {Thickness[0.0051], Thickness[0.003], Thickness[0.01]} ]
```

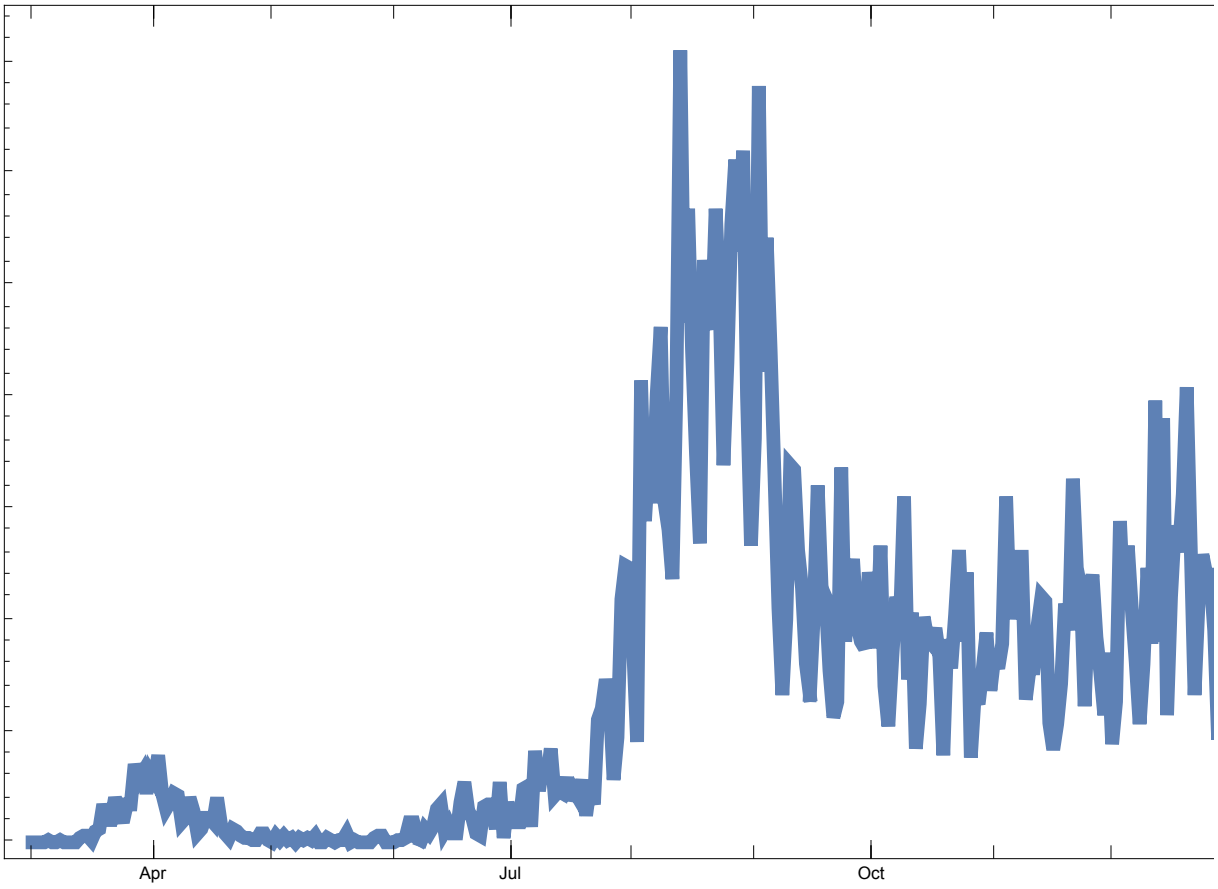
In[845]:=
```
calJunPlotL[ar_] :=
 DateListPlot[Map[TimeSeries[#, {"Jun 1, 2020"}] &, ar], PlotRange → All ]
```

In[846]:=
```
cJP[ar_] := DateListPlot[TimeSeries[ar, {"Jun 1, 2020"}], PlotRange → All ]
```

# THE CASE DATA TIME SERIES

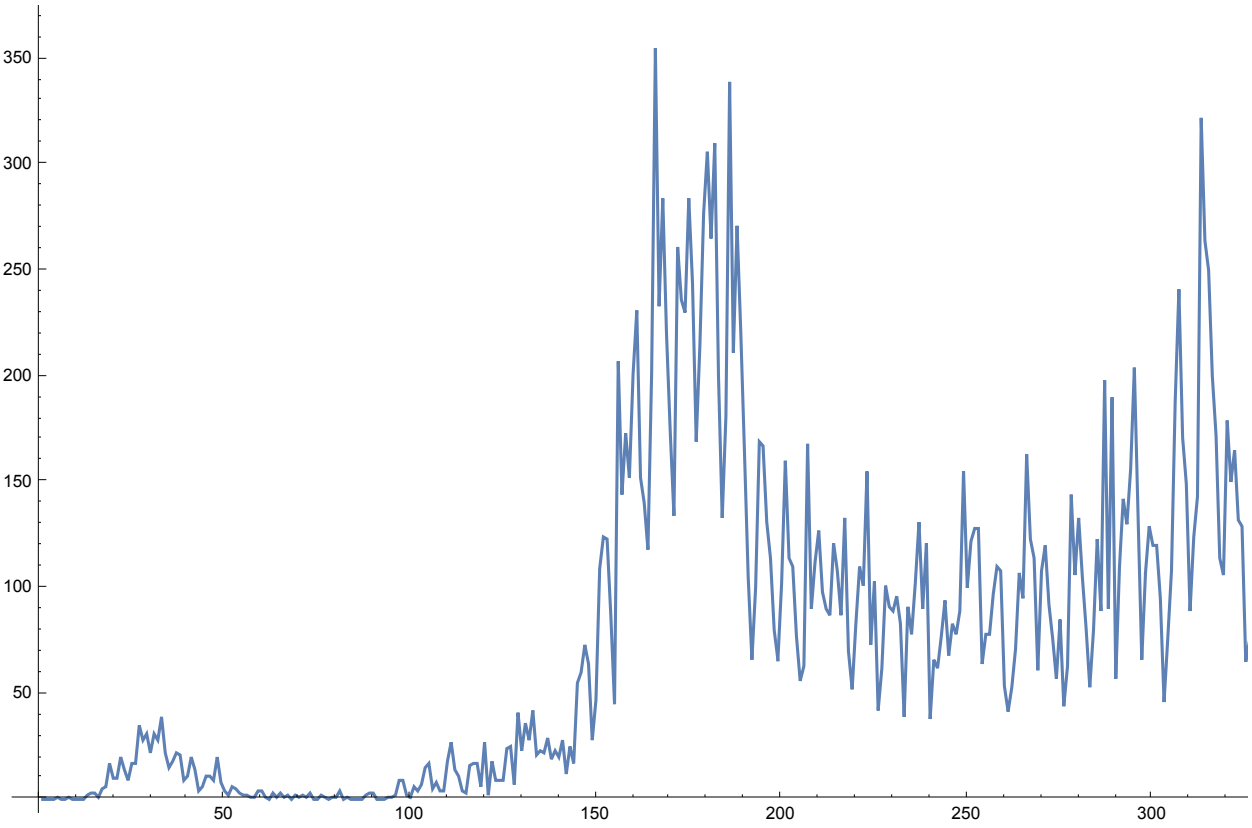
In[1133]:=
```
triCalPlotAll[{sinceMar2020}]
```

Out[1133]=



In[1030]:=
```
LPJ@sinceMar2020
```

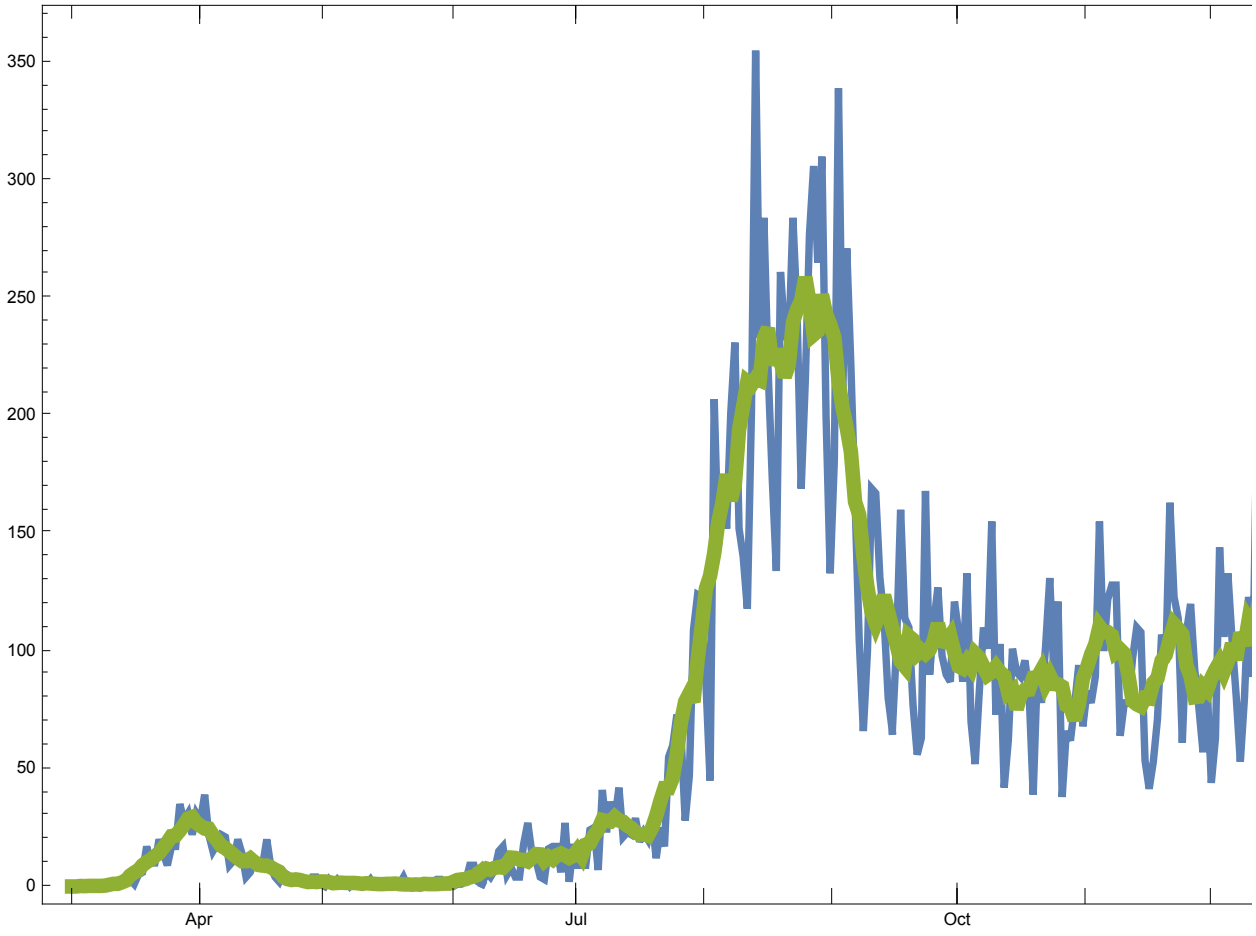
Out[1030]=



# 7-DAY RUNNING MEAN

In[1032]:=
```
triCalPlotAll[
 {sinceMar2020, runningSymMean[sinceMar2020, 7], runningSymMean[sinceMar2020, 7]}]
```

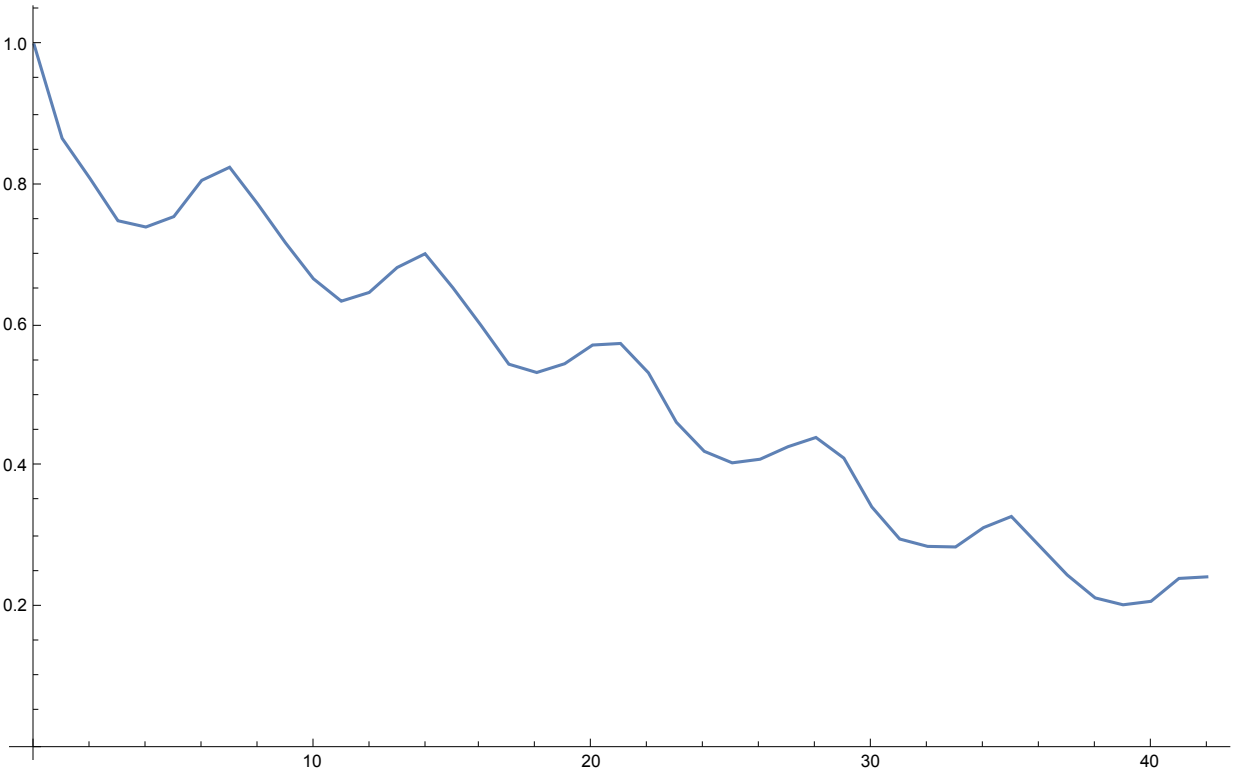
Out[1032]=



# AUTOCORRELATION

In[1029]:=
```
Table[{gap, CorrelationFunction[sinceMar2020, gap]}, {gap, 0, 42}] // LPJ
```

Out[1029]=




We see peaks at multiples of 7.
Is there a weekly periodicity?

# DO GROUP

In[1139]:=
```
cycleTableAll * runningSymMean[sinceMar2020, 7] 1. /. bestMeanCycleFitAll[[2]]
```

Out[1139]=
```
{0., 0., 0.1352967888, 0.1475387993, 0.3110610576, 0.3001881876, 0.3009658636,
 0.2330220582, 0.2847670063, 0.5073629579, 1.161868044, 1.399774759,
 2.101317313, 3.009658636, 4.310908077, 4.461349765, 4.650827114, 9.294944354,
 12.90903389, 13.65856253, 15.49974197, 13.28225732, 12.52974828, 12.68407395,
 22.07549284, 26.90678149, 28.51787782, 31.45093274, 26.91404772, 22.40167116,
 18.26506649, 26.59386857, 30.63951418, 29.41844238, 26.18403013, 18.29223157,
 13.09928229, 10.99286409, 15.36247747, 16.64176658, 14.409033, 12.94153213,
 10.0199485, 9.017621865, 7.018520918, 9.424040803, 11.04266755, 10.65668066,
 9.781390567, 6.641128659, 4.746116771, 2.705935776, 3.227411234, 3.421671634,
 3.602258251, 3.310624499, 2.097198524, 1.423835031, 1.352967888, 1.93644674,
```

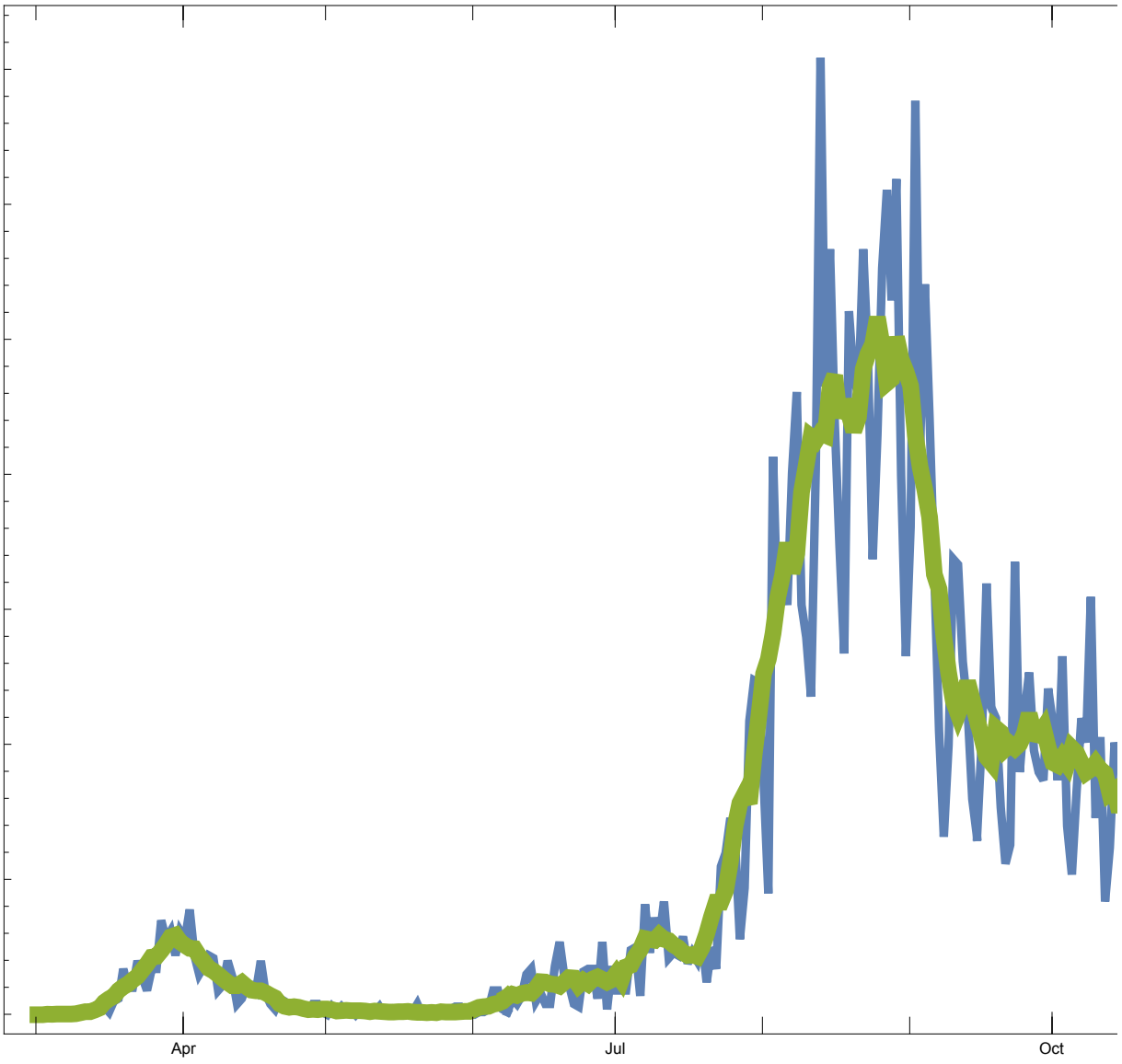
```
2.64401899, 2.551599594, 2.257243977, 1.28162132, 1.139068025, 1.099286409,
1.549157392, 1.866366346, 1.801129125, 1.65531225, 1.048599262, 1.04414569,
0.8456049299, 1.161868044, 1.244244231, 1.20075275, 1.354346386,
1.048599262, 0.9492233542, 0.6764839439, 0.9036751455, 1.088713702,
0.9005645627, 1.053380523, 0.6990661746, 0.8543010188, 0.6764839439,
1.032771595, 1.244244231, 1.350846844, 1.504829318, 1.165110291,
1.518757367, 1.860330846, 3.098314785, 3.88826322, 4.652916907, 5.116419681,
4.660441164, 5.030883777, 5.158190072, 7.358497613, 9.798423315, 9.90621019,
9.630907635, 9.08786027, 9.302388871, 8.20236782, 11.74777689, 13.99774759,
12.75799797, 14.59684438, 12.8162132, 10.34653456, 7.441323383, 13.03874138,
14.30880865, 15.90997394, 16.85408836, 12.23365806, 9.207466536, 8.794291271,
15.87886327, 16.01964447, 21.31336132, 22.12099097, 20.27291906, 18.32001074,
19.11067142, 28.78850821, 34.37224687, 35.42220613, 33.70817672, 25.6324264,
19.64892343, 16.8275381, 23.88284313, 27.37337307, 26.71674869, 26.0335472,
23.18569479, 22.7813605, 24.52254297, 43.11821409, 51.94719662, 55.38472061,
68.1687681, 65.24617629, 59.61122664, 55.38712291, 80.94347375, 119.7585072,
132.9833671, 152.288727, 122.9191357, 107.4520837, 104.8550113, 168.4708664,
217.7427403, 196.773357, 206.0111336, 180.7086061, 154.9132514, 145.0212455,
219.9803497, 268.4456927, 258.1618413, 280.3497019, 221.0214222, 168.2023784,
154.1537787, 230.1789692, 269.3788759, 266.1168283, 288.3252973, 228.4781281,
188.8954475, 175.0402205, 256.3855484, 291.6197415, 283.227555, 302.6211758,
226.6139516, 181.3016607, 157.7898799, 222.0458929, 252.7371093, 233.8465981,
222.2632903, 152.279915, 119.8869096, 94.96143363, 131.1619925, 144.9544529,
134.0340258, 139.7986436, 115.2294078, 88.65746128, 74.15955235, 106.2463778,
118.9808545, 112.2703822, 127.910492, 97.63624238, 74.98864498, 67.90207587,
102.1152914, 125.0465452, 125.3285683, 134.3812581, 97.05368724, 78.7855384,
71.96097953, 102.5025808, 116.3368356, 111.2197235, 114.9689599, 86.68420565,
75.08356732, 65.70350305, 96.43504767, 111.8264502, 109.4185944, 112.41075,
84.82002918, 67.96439216, 55.55624389, 86.10733172, 93.78490888, 97.86134915,
100.5225984, 77.94587847, 67.29993581, 60.03795002, 94.88589028, 107.3160649,
107.6174652, 103.5322571, 80.04307699, 64.45226575, 52.08926368, 78.61973766,
88.03027931, 92.60805587, 106.2409498, 86.68420565, 74.89372265, 69.59328573,
113.9921648, 134.3783769, 129.681297, 128.3619408, 92.62626813, 76.41248001,
66.88734995, 92.49760596, 97.98423315, 92.90824405, 92.09555426, 74.68356965,
60.65537233, 58.51586115, 91.27118969, 119.3696809, 118.1240518, 126.7066286,
103.8113269, 83.15196583, 72.46834249, 96.95143347, 110.893267, 96.51050231,
97.06149101, 79.34401082, 63.31319773, 59.86882904, 95.14408318, 118.3587324,
109.4185944, 115.1194428, 96.12159901, 73.09019827, 72.89114496, 105.729992,
140.5995981, 132.0828025, 141.0025071, 116.3945181, 95.49186943, 90.73340898,
139.2950689, 174.6607839, 149.9439997, 157.8565955, 124.4337791, 99.28876285,
87.60467074, 125.8690381, 127.068442, 113.9214172, 120.5368284, 102.7627277,
94.35280141, 88.36571517, 138.6495866, 166.1066048, 172.0078315, 182.5357963,
166.2612385, 142.6682701, 127.8554654, 198.9376285, 243.2497471, 238.499515,
236.4086858, 187.2332238, 136.2135513, 112.9728186, 157.2394753, 178.3935166,
```

```
156.0978575, 150.6334147, 118.1421835, 91.78989835, 80.41702883, 121.2215659,
144.6433918, 130.8820498, 136.9394679, 108.9378122, 88.37269428, 77.37285108,
111.4102358, 124.4244231, 116.1299446, 121.1086635, 89.48047035, 68.3440815}
```

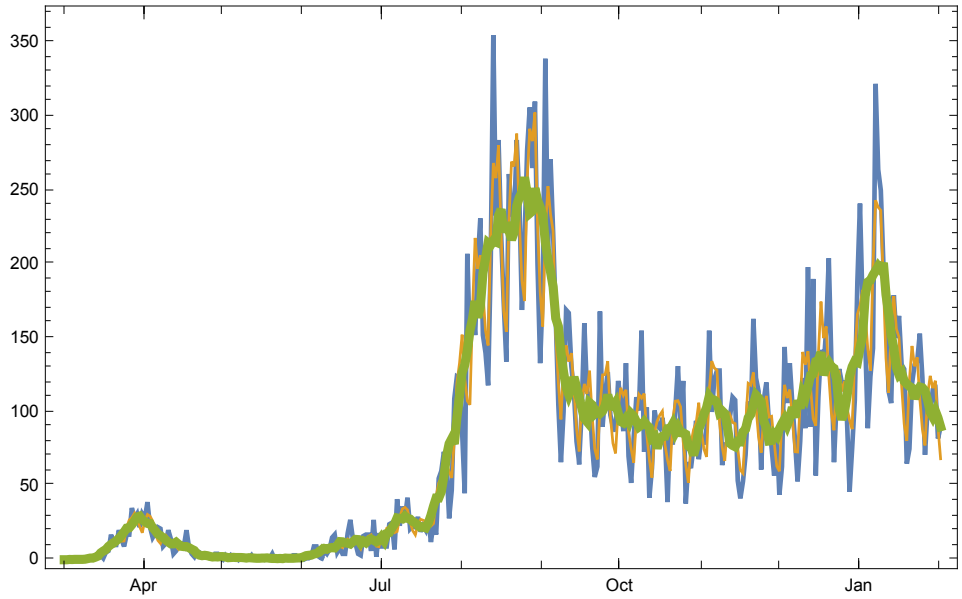
In[1016]:=
```
cycleTableAll = Take[
   Flatten[Table[{w1, w2, w3, w4, w5, w6, w7}, Ceiling[Length[sinceMar2020] / 7]]],
   Length[sinceMar2020]];
bestMeanCycleFitAll = NMinimize[
  Total[(sinceMar2020 - cycleTableAll * runningSymMean[sinceMar2020, 7])^2],
  {w1, w2, w3, w4, w5, w6, w7}]
cycleTableAllVals = cycleTableAll /. bestMeanCycleFitAll[[2]];
meanFitDataAll =
  cycleTableAll * runningSymMean[sinceMar2020, 7] /. bestMeanCycleFitAll[[2]];
triCalPlotAll[{sinceMar2020, runningSymMean[sinceMar2020, 7],
  runningSymMean[sinceMar2020, 7]}]
triCalPlotAll[{sinceMar2020, meanFitDataAll, runningSymMean[sinceMar2020, 7]}]
dailyFactorsRunM = {w1, w2, w3, w4, w5, w6, w7} / Mean@{w1, w2, w3, w4, w5, w6, w7} /.
  bestMeanCycleFitAll[[2]]
ListPlot[dailyFactorsRunM → {"SUN", "MON", "TUE", "WED", "THU", "FRI", "SAT"},
 LabelingFunction → Above,
 PlotStyle → {PointSize[0.0251], FontSize → Large}, GridLines → Automatic]
```

Out[1017]=
```
{176 835.4369, {w1 → 0.9320882328, w2 → 0.7593786834, w3 → 0.6764839439,
  w4 → 1.032771595, w5 → 1.244244231, w6 → 1.20075275, w7 → 1.203863454}}
```

Out[1020]=

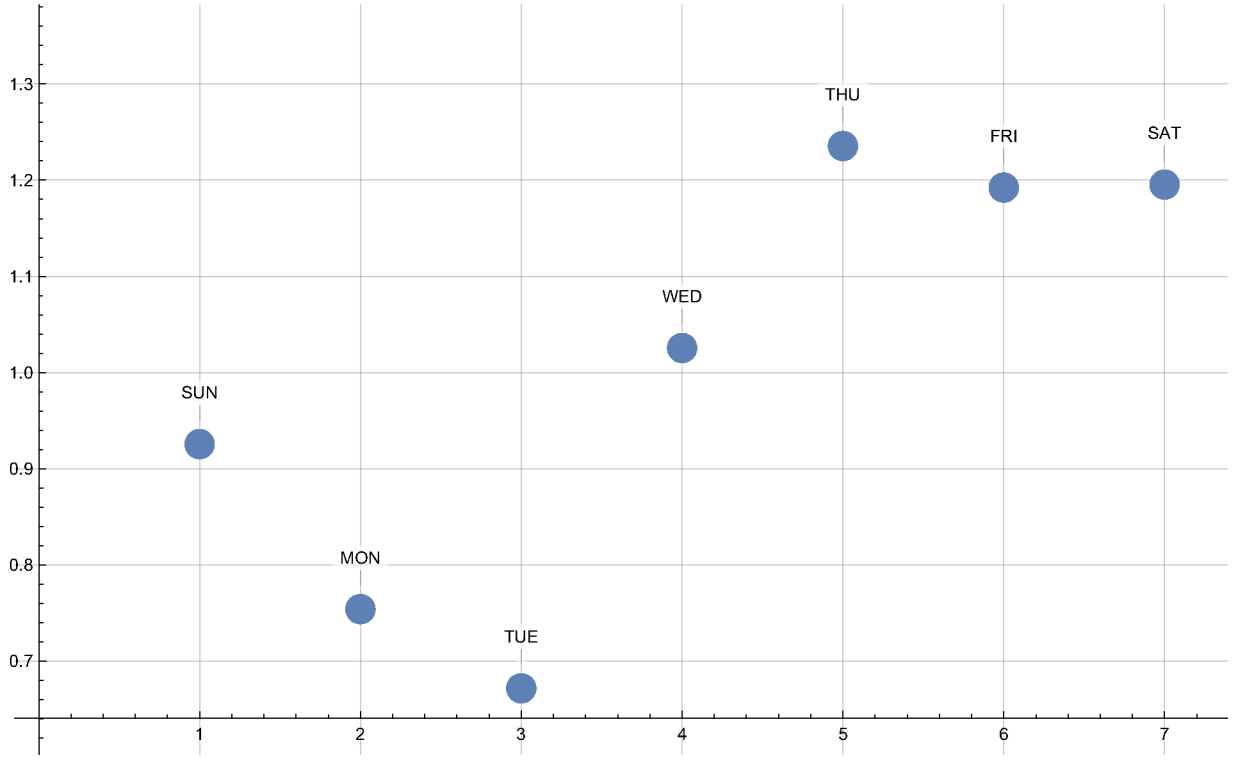


Out[1021]=



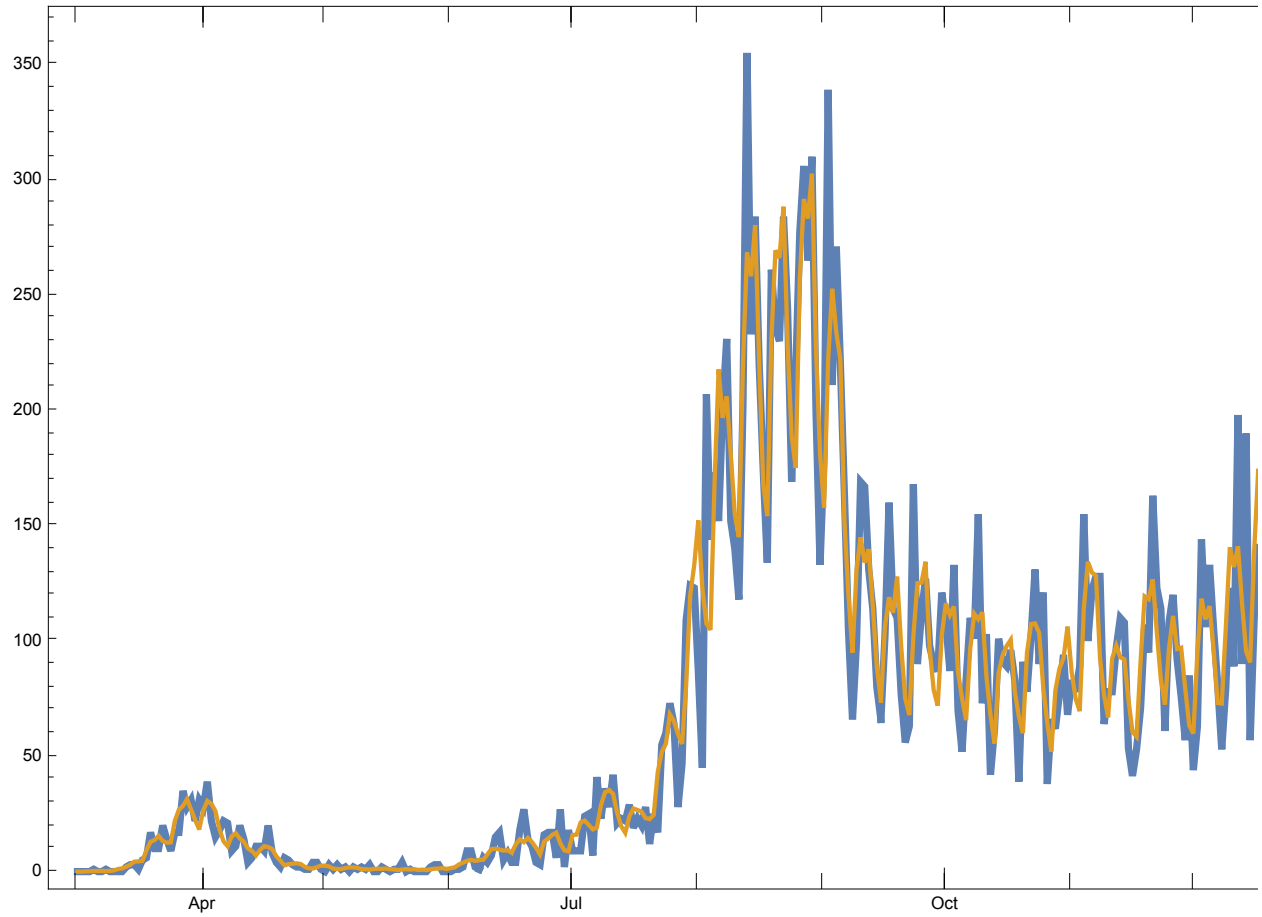

Out[1022]= {0.9255324366, 0.7540376312, 0.6717259278,
1.025507647, 1.235492901, 1.192307316, 1.195396141}

Out[1023]=



In[973]:=
```
triCalPlotAll[{sinceMar2020, meanFitDataAll}]
```

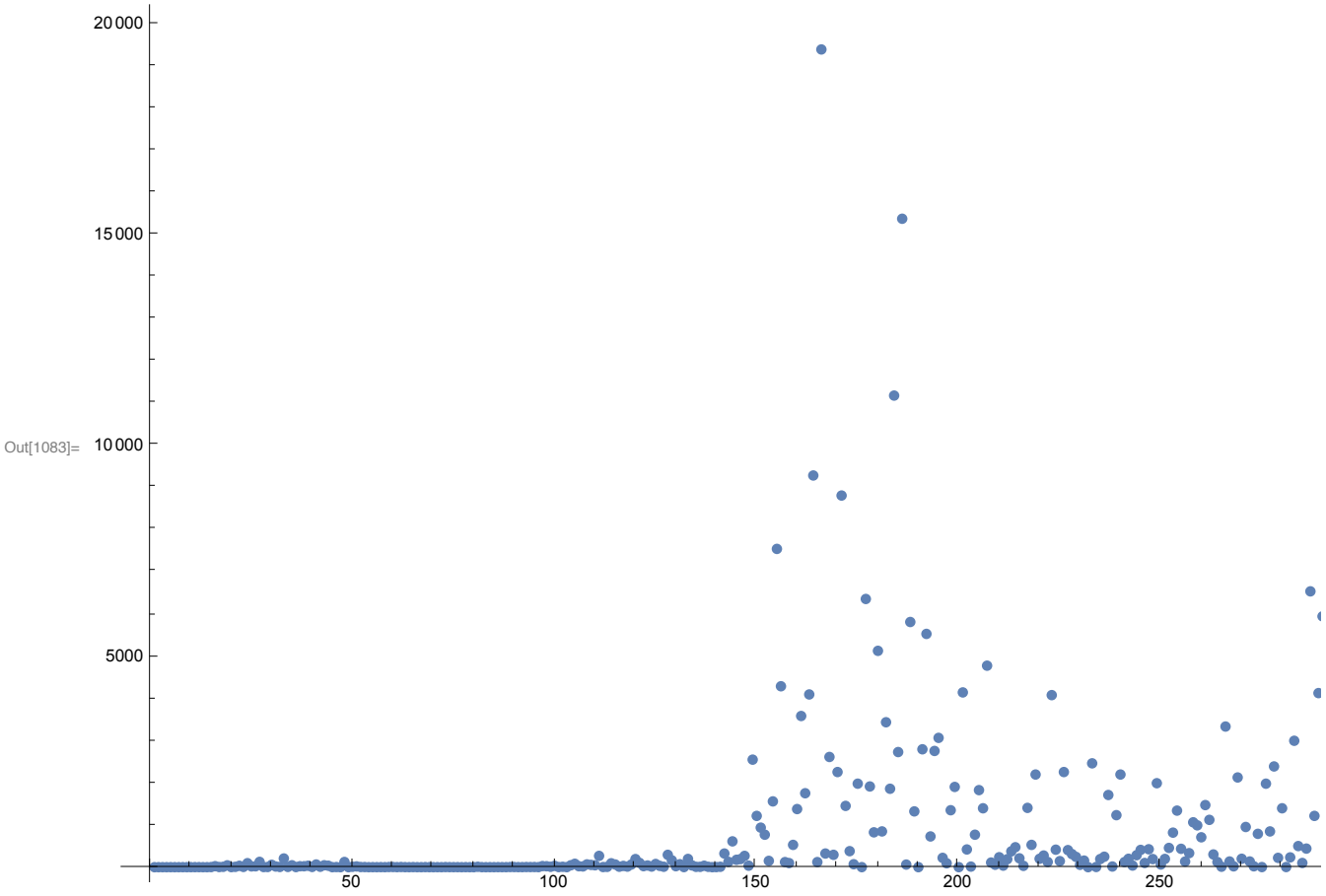
Out[973]=



# VARIANCE — Squared Departures from the 7-day running mean

In[1083]:=
```
LP[(sinceMar2020 - runningSymMean[sinceMar2020, 7])^2]
```

Out[1083]=




## HALF THE VARIANCE IS DUE TO THE DAY-OF-THE-WEEK FACTORS

In[1003]:=
```
squaresTotal = Total[(sinceMar2020 - runningSymMean[sinceMar2020, 7])^2]
```

Out[1003]= 337 809.3768

In[1004]:=
```
squaresCycleTotal =
 Total[(sinceMar2020 - cycleTableAllVals * runningSymMean[sinceMar2020, 7])^2]
```

Out[1004]= 176 835.4369

In[1005]:=
```
squaresCycleTotal / squaresTotal
```

Out[1005]= 0.5234769934

In[1082]:=
```
LP[{(sinceMar2020 - runningSymMean[sinceMar2020, 7])^2,
   (sinceMar2020 - cycleTableAllVals * runningSymMean[sinceMar2020, 7])^2}]
```

In[1060]:=
```
meanFitDataAllEd = meanFitDataAll;
meanFitDataAllEd[[1]] = 0.13529678878200513`;
meanFitDataAllEd[[2]] = 0.13529678878200513`;
```

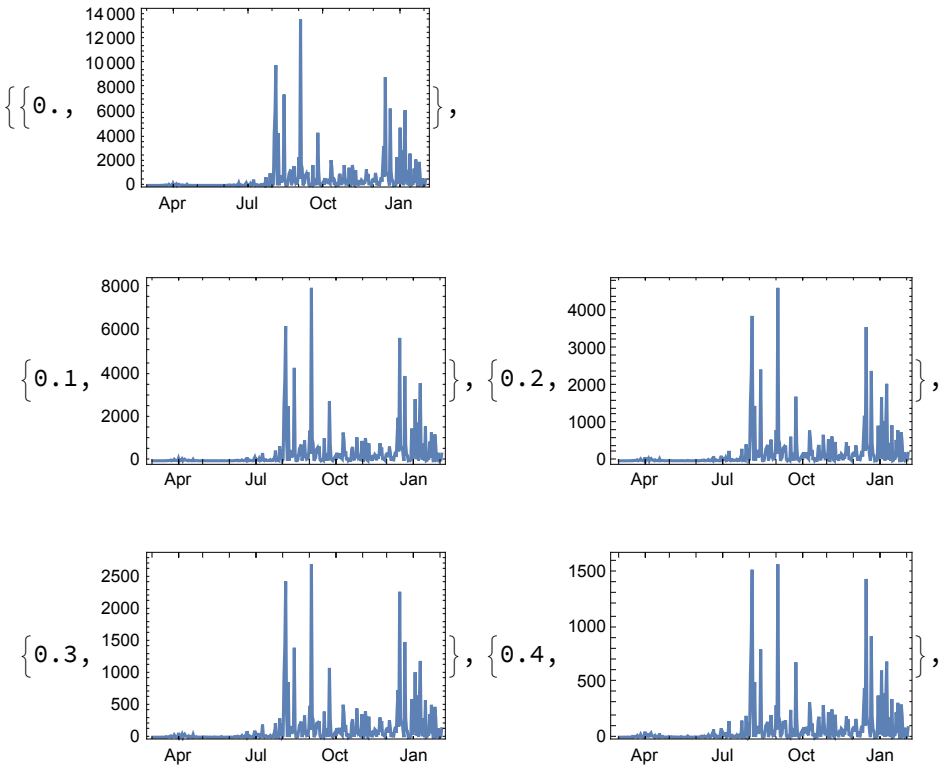
# TAYLOR'S LAW

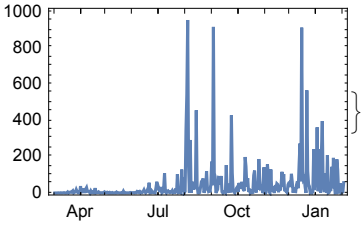
In[1141]:=
```
rMean = runningSymMean[sinceMar2020, 7];
```

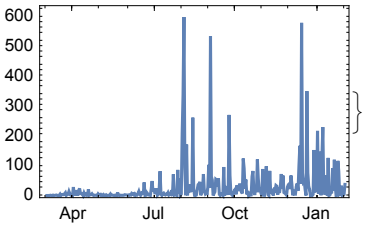
In[•]:=
```
powerPlots = ParallelTable[{k,
    dateAllPlot@Accumulate[(sinceMar2020 - rMean)^2 / rMean^(k)]}, {k, 0, 2, 0.1} ]
```

In[1087]:=
```
powerPlots = ParallelTable[
  {k, dateAllPlot[(sinceMar2020 - meanFitDataAll)^2 / meanFitDataAllEd^(k)]},
  {k, 0, 2, 0.1} ]
```

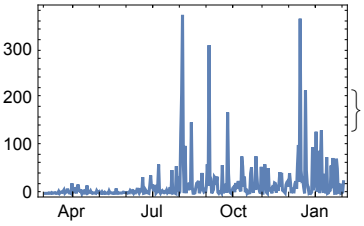
Out[1087]=

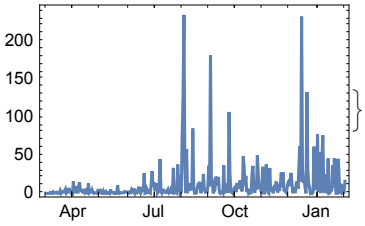


{0.5, 

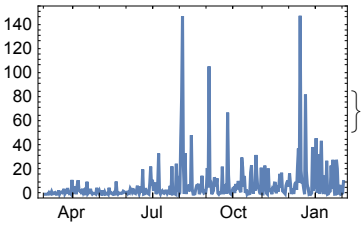
}, {0.6, 

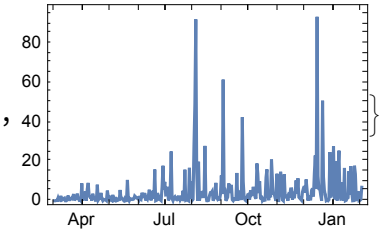
},

{0.7, 

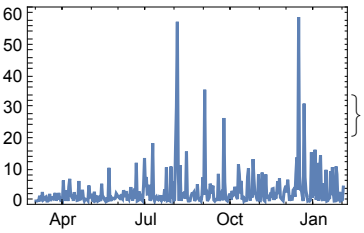
}, {0.8, 

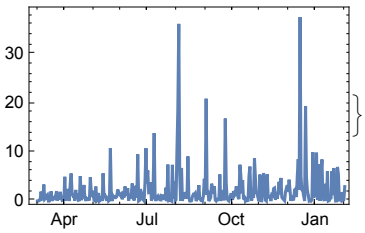
},

{0.9, 

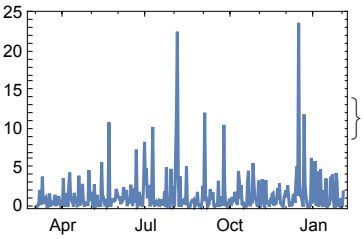
}, {1., 

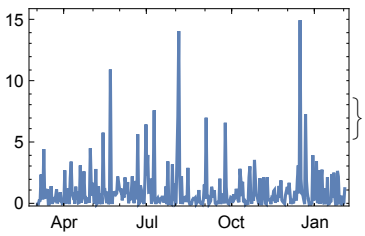
},

{1.1, 

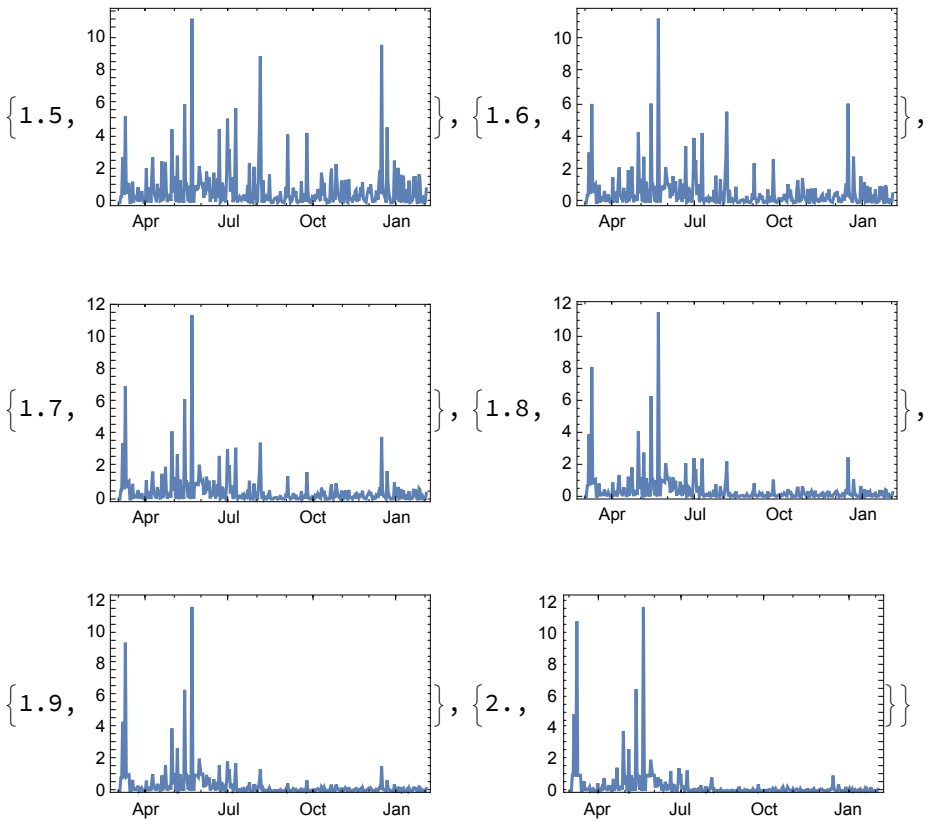
}, {1.2, 

},

{1.3, 

}, {1.4, 

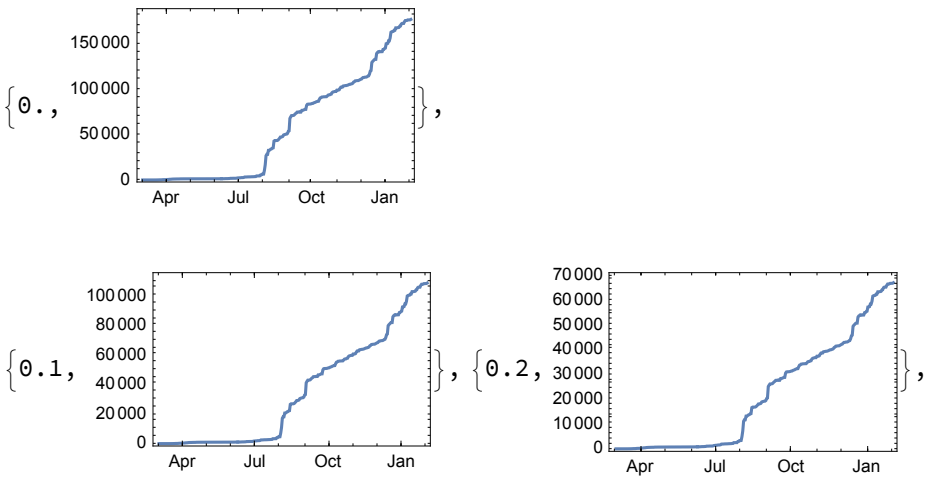
},

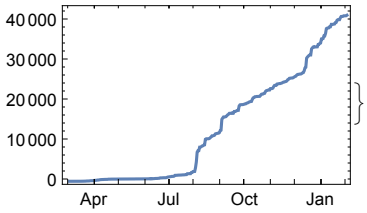


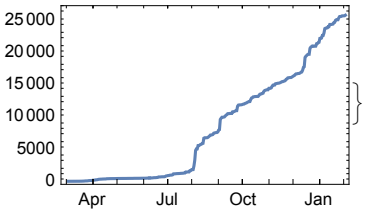

## How to see uniformity: Accumulate.

In[1084]:=
```
powerPlots = ParallelTable[{k, dateAllPlot@Accumulate[
      (sinceMar2020 - meanFitDataAll)^2 / meanFitDataAllEd^(k)]}, {k, 0, 2, 0.1}]
```

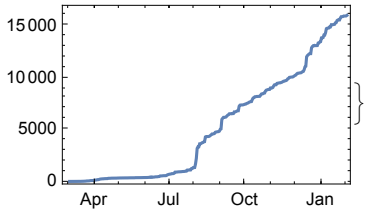
Out[1084]=

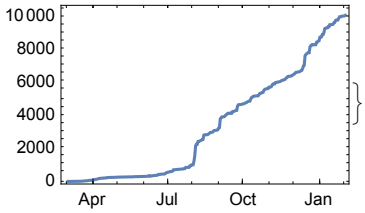


{0.3, 

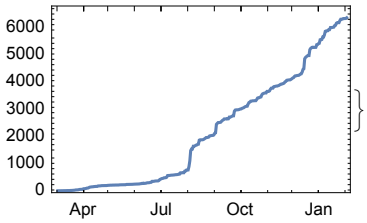
}, {0.4, 

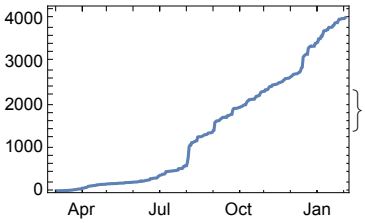
},

{0.5, 

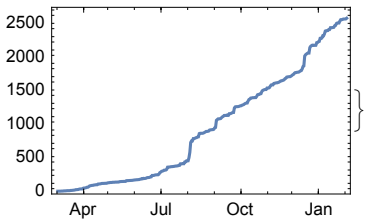
}, {0.6, 

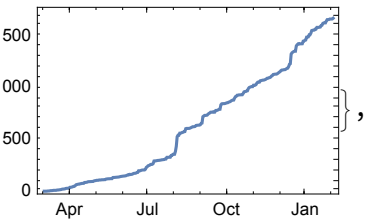
},

{0.7, 

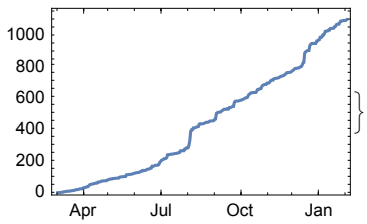
}, {0.8, 

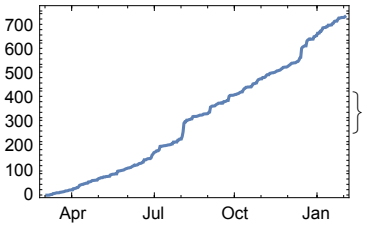
},

{0.9, 

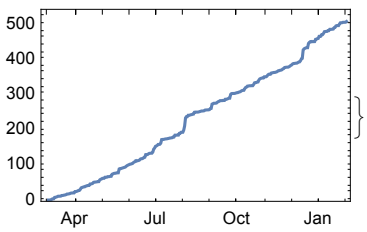
}, {1., 

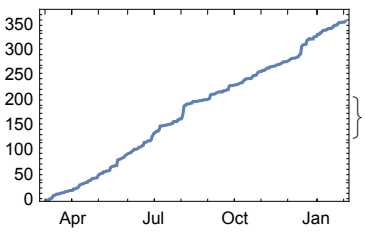
},

{1.1, 

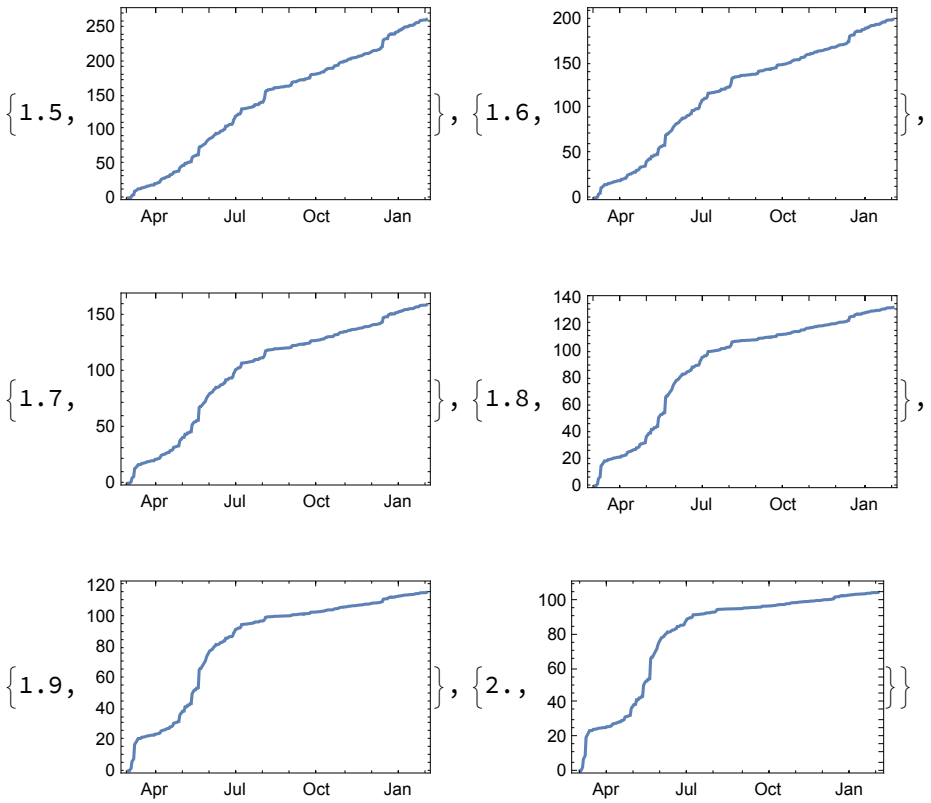
}, {1.2, 

},

{1.3, 

}, {1.4, 

},




## Measure the straightness of the curves using Correlation:

In[1086]:=

```
Table[{k, Correlation[Range[Length@sinceMar2020],
    Accumulate[(sinceMar2020 - meanFitDataAll)^2 / meanFitDataAllEd^(k)]]},
  {k, 0, 2, 0.1}] // TF
```

Out[1086]//TableForm=

| | |
|---|---|
| 0. | 0.9472536227 |
| 0.1 | 0.9482333694 |
| 0.2 | 0.9494626049 |
| 0.3 | 0.9510164862 |
| 0.4 | 0.9529847153 |
| 0.5 | 0.9554716837 |
| 0.6 | 0.9585936579 |
| 0.7 | 0.9624696631 |
| 0.8 | 0.9671999705 |
| 0.9 | 0.972821912 |
| 1. | 0.9792278926 |
| 1.1 | 0.9860292822 |
| 1.2 | 0.99236518 |
| 1.3 | 0.9967117884 |
| 1.4 | 0.9968647519 |
| 1.5 | 0.9903755717 |
| 1.6 | 0.9755763608 |
| 1.7 | 0.9527257463 |
| 1.8 | 0.9242838372 |
| 1.9 | 0.8938815636 |
| 2. | 0.8647930116 |

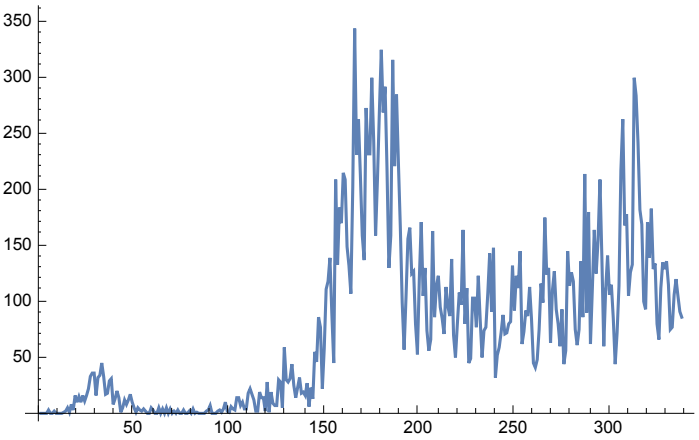
Correlation maximized for exponent 1.4.

# BOOTSTRAP

In[1164]:=
```
simData := Table[
   RandomVariate@PoissonDistribution@(0.001 * sinceMar2020 [[ii]]),
   {ii, Length[sinceMar2020]}];
```

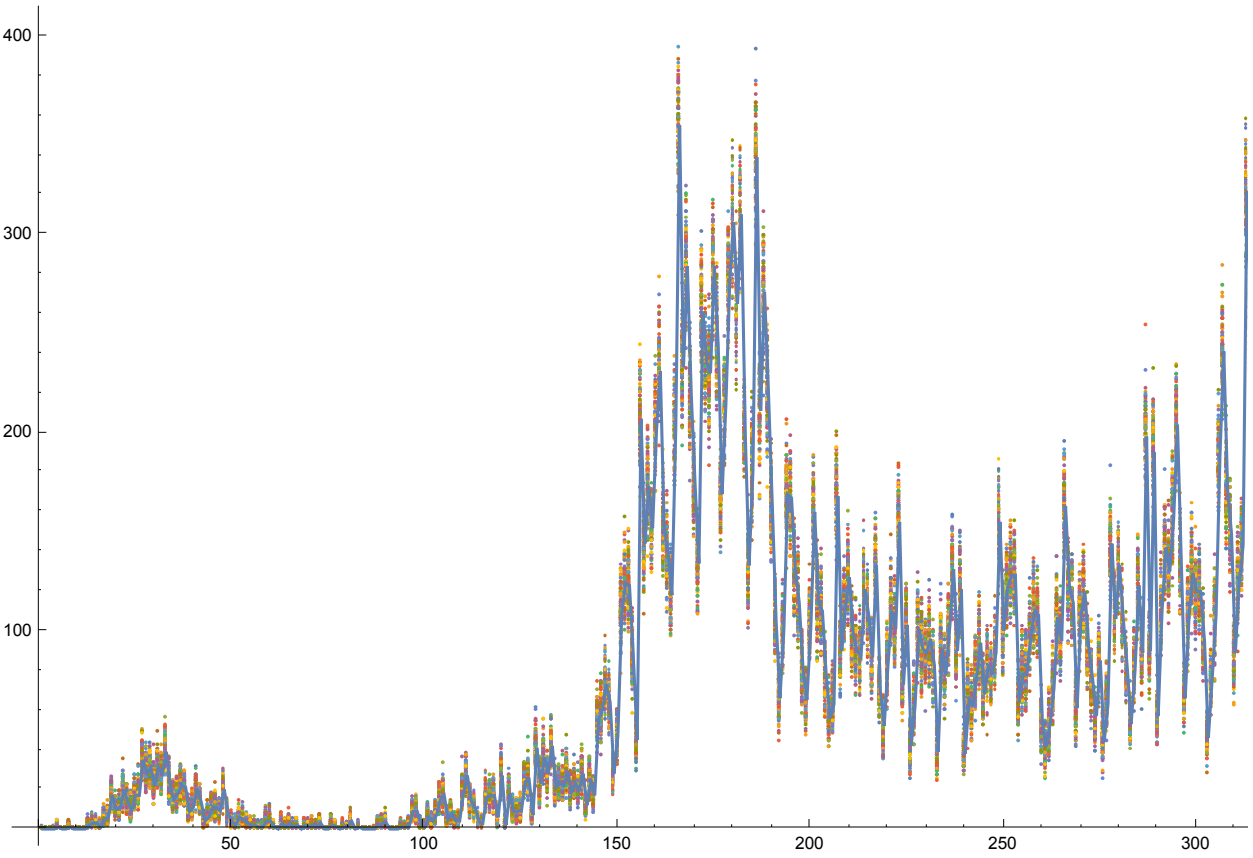
In[1165]:=
```
simData // LPJ
```

Out[1165]=



In[1162]:=
```
simRuns = Table[simData, 100];
Show[ListPlot[simRuns, PlotRange → All],
 ListPlot[sinceMar2020, Joined → True, PlotRange → All] ]
```

Out[1163]=



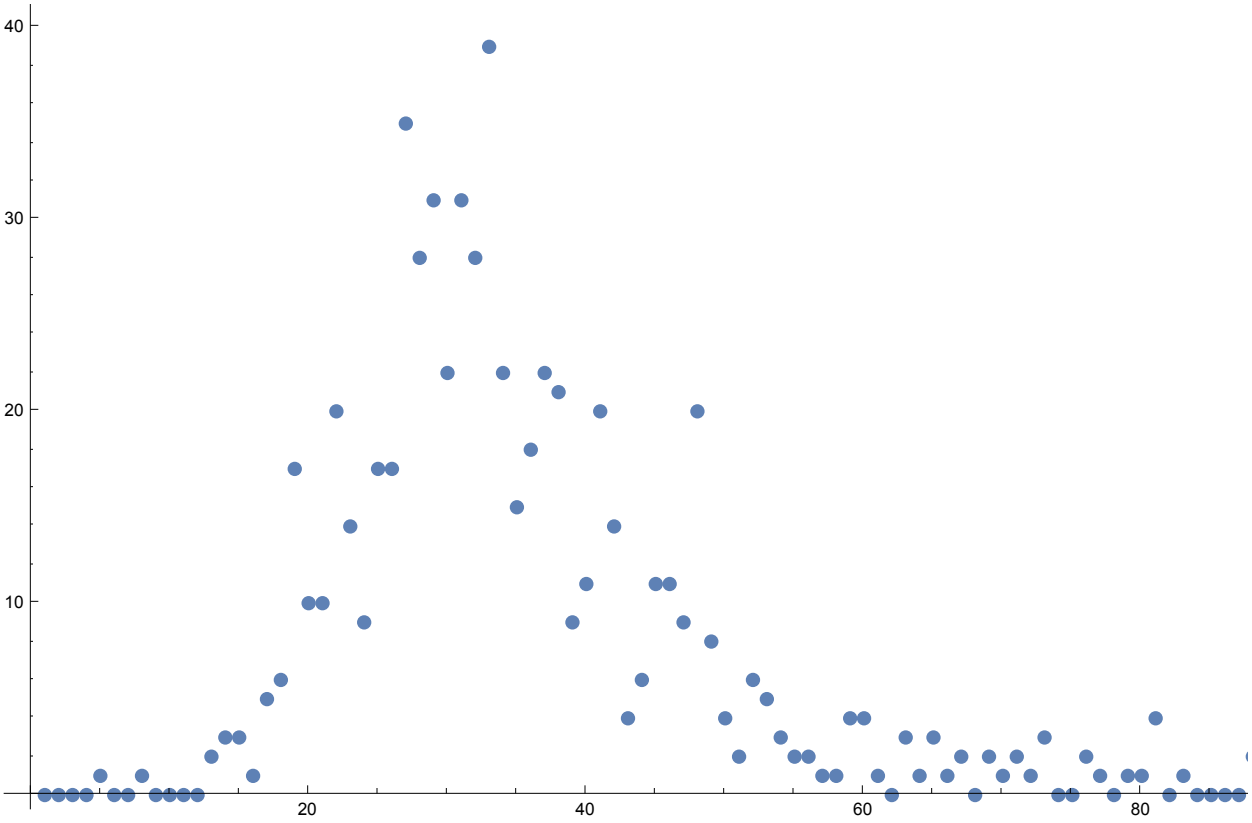

# TWO-PERIOD FIT OF “FIRST WAVE”

In[•]:=
```
twoExpFunc[t_, date_, c1_, r1_, r2_] :=
 If[t < date, c1 Exp[t r1], c1 Exp[date r1] Exp[(t - date) r2] ]
```

In[1108]:=
```
first90 = Take[sinceMar2020, 90]
```

In[1143]:=
```
LP@first90
```

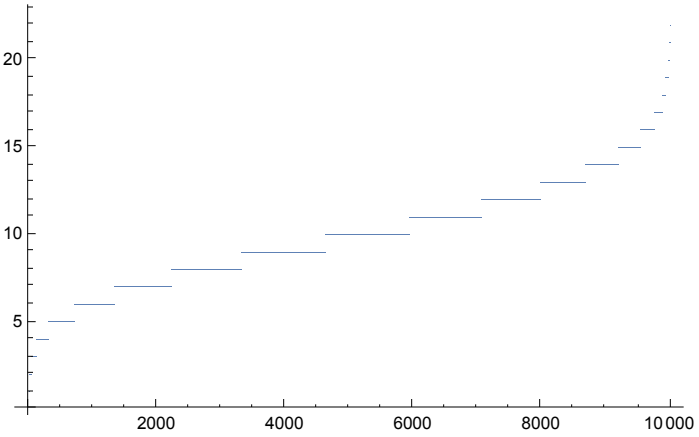
Out[1143]=




In[1111]:=
```
bootStrap[ar_] := Table[
  RandomVariate@PoissonDistribution@(0.001 + ar [[ii]]), {ii, Length[ar]}]
```

In[1148]:=
```
LP@Sort@Table[RandomVariate@PoissonDistribution[10], 10 000]
```

Out[1148]=




In[1156]:=
```
Variance[Sort@Table[RandomVariate@PoissonDistribution[10], 10 000]] 1.
```

Out[1156]=
```
10.17478819
```

In[1155]:=
```
Mean[Sort@Table[RandomVariate@PoissonDistribution[10], 10 000]] 1.
```

Out[1155]=
```
10.0365
```

In[1112]:=
```
bootStrap[first90]
```

Out[1112]=
```
{0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0, 1, 4, 4, 0, 4, 6, 18, 7, 10, 23,
 16, 10, 14, 9, 37, 33, 37, 19, 38, 32, 31, 20, 15, 6, 20, 22, 9, 9, 18,
 15, 2, 5, 13, 15, 14, 16, 8, 1, 0, 6, 2, 3, 1, 2, 1, 1, 2, 4, 1, 0, 3, 1, 4,
 0, 5, 0, 3, 2, 3, 1, 4, 0, 0, 1, 1, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 2, 1, 3}
```

# BEST FIT OF A 2-PIECE EXPONENTIAL MODEL

In[1122]:=
```
bestFit90 = Table[NMinimize[Total[
    (first90 - Table[twoExpFunc[t, break, c1, r1, r2], {t, Length[first90]}])^2],
  {c1, r1, r2}, Method → "DifferentialEvolution"], {break, 10, 40}]
```

Out[1122]=
```
{{5986.993125, {c1 → 8.694059303 * 10^-6, r1 → 1.43660045, r2 → -0.01703737675}},
 {5755.428768, {c1 → 0.00008495848623, r1 → 1.103807912, r2 → -0.01866899602}},
 {5471.174853, {c1 → 4.386571179 * 10^-6, r1 → 1.263395862, r2 → -0.02027062446}},
 {5167.564649, {c1 → 4.204146215 * 10^-6, r1 → 1.173958898, r2 → -0.02205284428}},
 {4893.167687, {c1 → 0.00003747641204, r1 → 0.9378358638, r2 → -0.02402808059}},
 {4593.934247, {c1 → 6.147215222 * 10^-6, r1 → 0.9991464833, r2 → -0.02571499527}},
 {4299.983623, {c1 → 0.00006998282323, r1 → 0.7877900617, r2 → -0.02765511495}},
 {3870.514104, {c1 → 3.355922202 * 10^-6, r1 → 0.9242109413, r2 → -0.03041881171}},
 {3470.610576, {c1 → 6.458269763 * 10^-11, r1 → 1.480193831, r2 → -0.03349356328}},
 {3137.549474, {c1 → 0.0000318547571, r1 → 0.7146014714, r2 → -0.03597892284}},
 {3021.500936, {c1 → 0.0003910044328, r1 → 0.5548204056, r2 → -0.03809966199}},
 {2813.475102, {c1 → 0.002479213708, r1 → 0.4420164281, r2 → -0.04068119559}},
 {2549.154335, {c1 → 0.005958326135, r1 → 0.3839412218, r2 → -0.04389268342}},
 {2412.757839, {c1 → 0.02858397752, r1 → 0.2998353935, r2 → -0.04645519447}},
 {2230.792158, {c1 → 0.06157782483, r1 → 0.2564667715, r2 → -0.04972785508}},
 {1926.878044, {c1 → 0.076624764, r1 → 0.2393293728, r2 → -0.05454286897}},
 {1629.06923, {c1 → 0.1039847717, r1 → 0.2200871788, r2 → -0.06010432556}},
 {1303.917648, {c1 → 0.1339021128, r1 → 0.2043873075, r2 → -0.06718748362}},
 {1216.833507, {c1 → 0.2649189365, r1 → 0.1730792077, r2 → -0.0726576603}},
 {1190.784269, {c1 → 0.4444512035, r1 → 0.1494270751, r2 → -0.07860998989}},
 {1252.934369, {c1 → 0.6893519095, r1 → 0.1296053427, r2 → -0.08444515202}},
 {1267.661854, {c1 → 0.9157502686, r1 → 0.1164101266, r2 → -0.0928611771}},
 {1363.674454, {c1 → 1.199321977, r1 → 0.1041454365, r2 → -0.101620958}},
 {1498.891128, {c1 → 1.508878726, r1 → 0.09371707577, r2 → -0.1117693015}},
 {1826.784697, {c1 → 1.968657282, r1 → 0.08180221674, r2 → -0.1157189418}},
 {2135.312036, {c1 → 2.412816313, r1 → 0.07240037526, r2 → -0.1198495321}},
 {2366.979401, {c1 → 2.78523167, r1 → 0.06554606424, r2 → -0.127799264}},
 {2574.748115, {c1 → 3.128578665, r1 → 0.05989974273, r2 → -0.1383545891}},
 {2818.482625, {c1 → 3.500368921, r1 → 0.05440385226, r2 → -0.1467972425}},
 {3102.386015, {c1 → 3.909256351, r1 → 0.0489105843, r2 → -0.1496865753}},
 {3311.710209, {c1 → 4.236809154, r1 → 0.04479358787, r2 → -0.1591817593}}}

r1 → 0.14942707507311212`, r2 → -0.0786099898926045`
```

In[1158]:=
```
Solve[Exp[t 0.14942707507311212`] == 2, t]
```

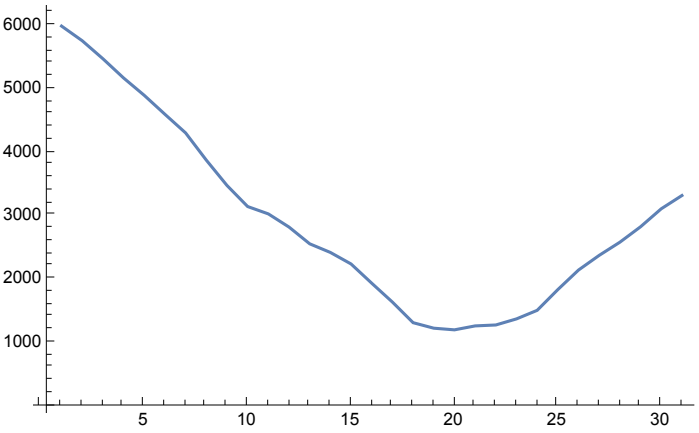
Out[1158]=
```
{{t → 4.638698711}}
```

In[1160]:=
```
Solve[Exp[t * 0.0786099898926045`] == 2, t]
```

Out[1160]=
$$\{\{t \to 8.817545728\}\}$$

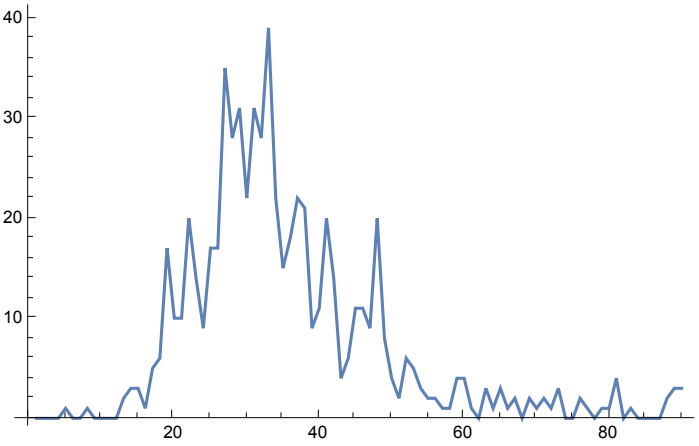
In[1123]:=
```
LPJ@Map[First, bestFit90]
```

Out[1123]=




In[1157]:=
```
LPJ@first90
```

Out[1157]=



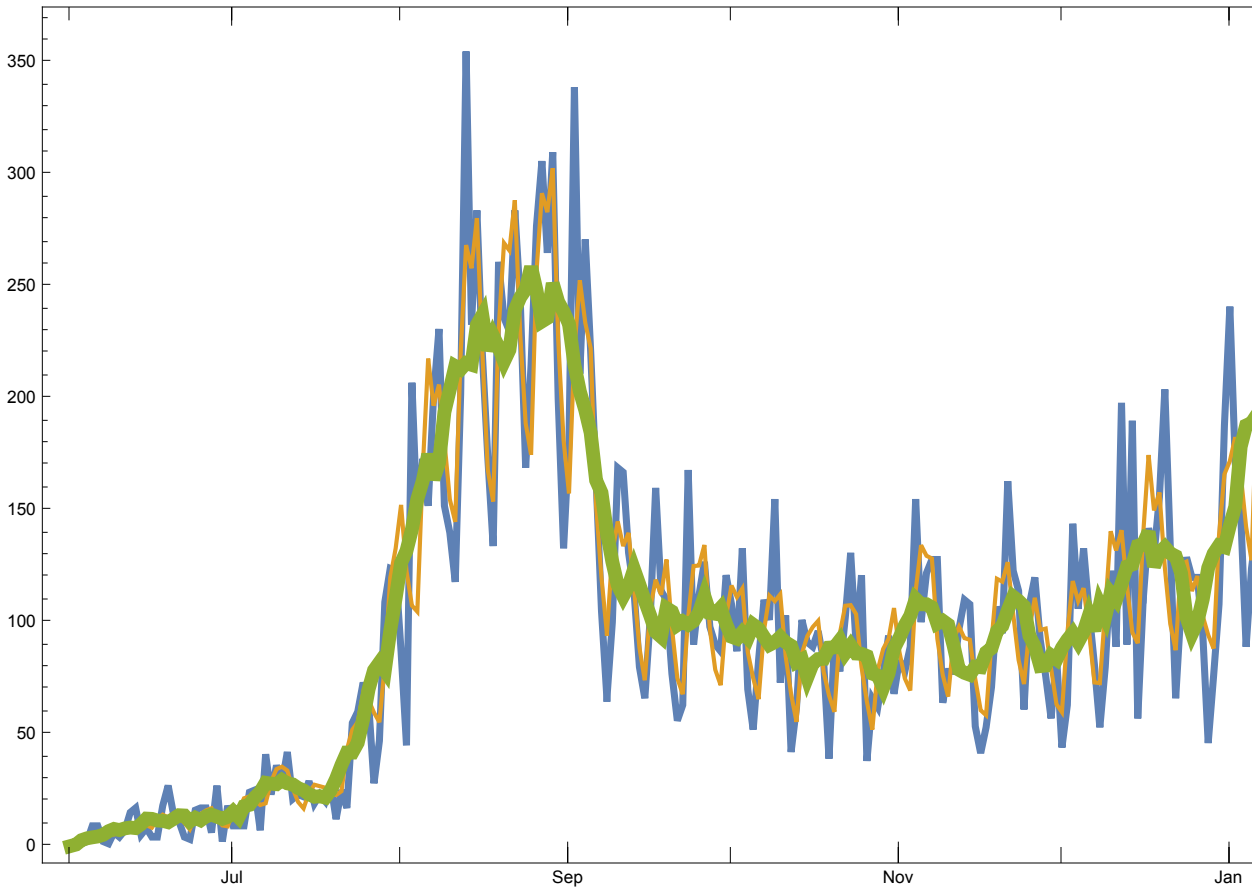

In[1116]:=
```
NMinimize[
 (first90 - Table[twoExpFunc[t, break, c1, r1, r2], {t, Length[first90]}])^2,
 {break, c1, r1, r2}]
```

# JUNE and AFTER

In[983]:=
```
cycleTable = Take[
    Flatten[Table[{w1, w2, w3, w4, w5, w6, w7}, Ceiling[Length[sinceJune2020] / 7]]],
    Length[sinceJune2020]];
bestMeanCycleFit = NMinimize[Total[
    (sinceJune2020 - cycleTable * runningSymMean[sinceJune2020, 7]) ^ 2],
   {w1, w2, w3, w4, w5, w6, w7}]
cycleTableVals = cycleTable /. bestMeanCycleFit[[2]];
meanFitData =
  cycleTable * runningSymMean[sinceJune2020, 7] /. bestMeanCycleFit[[2]];
triCalPlot[{sinceJune2020, meanFitData, runningSymMean[sinceJune2020, 7]}]
```

Out[984]=
```
{175 532.1757, {w1 → 0.7592100903, w2 → 0.6755566565, w3 → 1.033112053,
  w4 → 1.244381427, w5 → 1.200599959, w6 → 1.204656409, w7 → 0.9318796048}}
```

Out[987]=



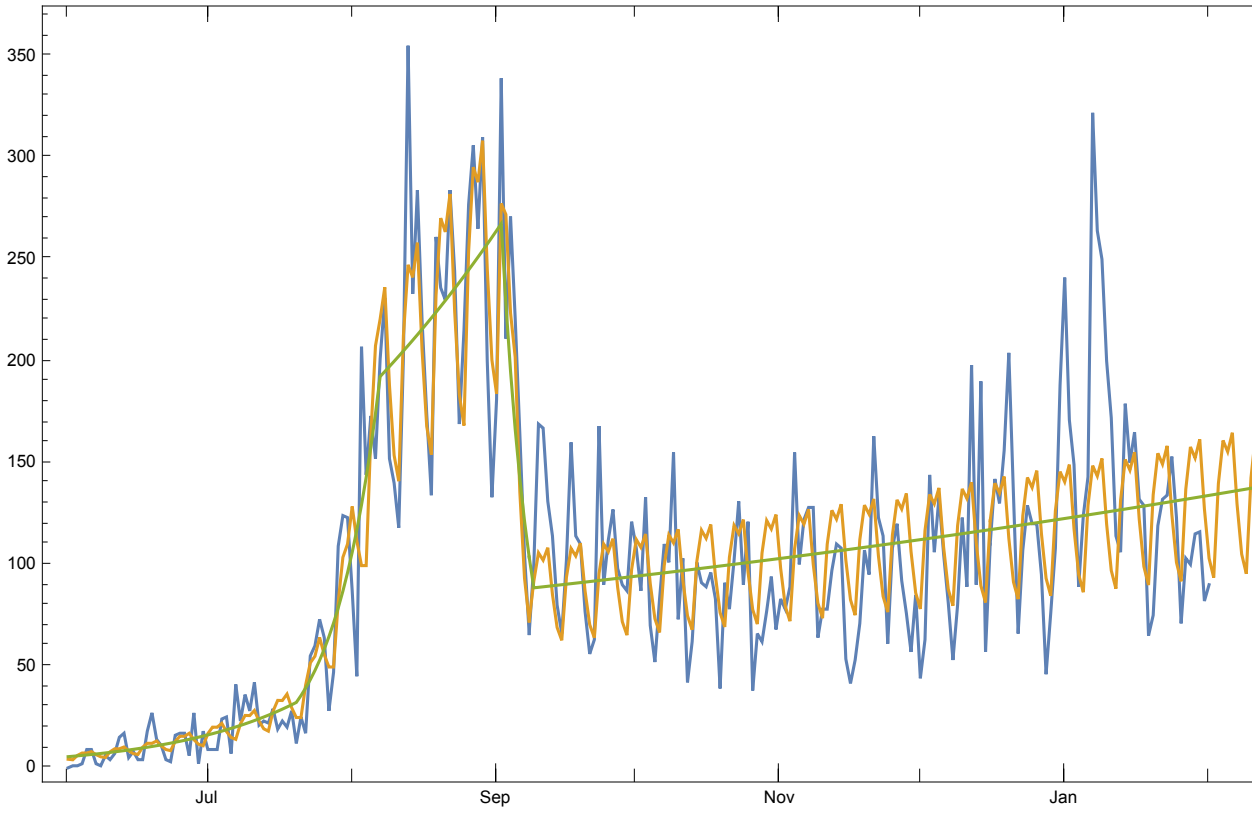
In[851]:=
```
bestFit94101 =
 NMinimize[Total[(sinceJune2020 - cycleTable * Table[fiveExpFunc[t, 50, 68, 94,
         101, c1, r1, r2, r3, r4, r5], {t, Length[sinceJune2020]}])^2],
  {c1, r1, r2, r3, r4, r5, w1, w2, w3, w4, w5, w6, w7}]
Log[2.] / {r1, r2, r3, r4, r5} /. bestFit94101[[2]]
cycleTable = Take[
   Flatten[Table[{w1, w2, w3, w4, w5, w6, w7}, Ceiling[Length[sinceJune2020] / 7]]],
   Length[sinceJune2020]];
cycleTableProj = Take[Flatten[Table[{w1, w2, w3, w4, w5, w6, w7},
    2 + Ceiling[Length[sinceJune2020] / 7]]], Length[sinceJune2020] + 14];
fit5TabCyc = (cycleTable * Table[fiveExpFunc[t, 50, 68, 94, 101, c1,
      r1, r2, r3, r4, r5], {t, 1, dataN}]) /. bestFit94101[[2]];
fit5TabCycProj = (cycleTableProj * Table[fiveExpFunc[t, 50, 68, 94, 101,
       c1, r1, r2, r3, r4, r5], {t, 1, dataN + 14}]) /. bestFit94101[[2]];
fit5TabCycMean = (Mean@cycleTableProj * Table[fiveExpFunc[t, 50, 68, 94,
       101, c1, r1, r2, r3, r4, r5], {t, 1, dataN + 14}]) /. bestFit94101[[2]];
(* {LPJ@{sinceJune2020,fit5TabCycProj,fit5TabCycMean}, *)
Round[Take[fit5TabCycProj, -15], 1]
calJunPlotL[{sinceJune2020, fit5TabCycProj, fit5TabCycMean}]
```

Out[851]=
```
{278 267.9004, {c1 → 2.815860724, r1 → 0.03612549442,
  r2 → 0.09939612856, r3 → 0.01267513692, r4 → -0.1580322157,
  r5 → 0.002858870484, w1 → 1.441923431, w2 → 1.305737665, w3 → 1.944890841,
  w4 → 2.233973878, w5 → 2.150688097, w6 → 2.273855733, w7 → 1.787107054}}
```

Out[852]=
```
{19.1872026, 6.973583283, 54.68557734, -4.386113156, 242.4549081}
```

Out[857]=
```
{103, 94, 140, 161, 155, 165, 130, 105, 95, 143, 164, 159, 168, 132, 107}
```

Out[858]=



# BEST BREAKPOINT

In[859]:=
```
Timing[period5Data = Take[sinceJune2020, {101, Length[sinceJune2020]}];
 fitPeriod5Table =
  ParallelTable[{date, NMinimize[Total[(period5Data - period5Cycle *
           Table[twoExpFunc[t, date, c1, r1, r2], {t, Length[period5Data]}])^2],
     {c1, r1, r2, w1, w2, w3, w4, w5, w6, w7}, Method → "DifferentialEvolution"]},
   {date, 2, Length[period5Data] - 1}];]
squaredErrors = Map[First, Map[Last, fitPeriod5Table]];
bestPeriod5break = Position[squaredErrors, Min[squaredErrors]][[1, 1]]
bestPeriod5Params = fitPeriod5Table[[bestPeriod5break, 2, 2]]
squaredErrors[[ bestPeriod5break ]];
doubleHalfTimes = Log[2.] / {r1, r2} /. bestPeriod5Params
{ListPlot[ Map[First, Map[Last, fitPeriod5Table]], PlotRange → Automatic,
  Joined → True , AxesLabel → {"Days After Sept. 9", "Squared Error"}],
 ListPlot[ Map[First, Map[Last, fitPeriod5Table]],
  PlotRange → {0, All}, Joined → True ]}
```

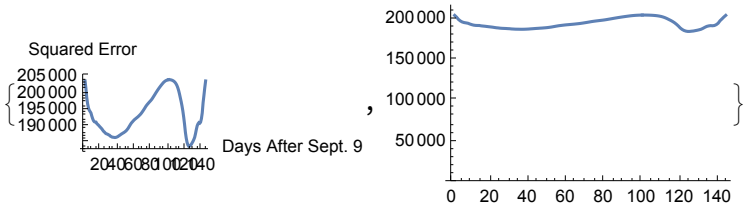
Out[859]= {9.902575, Null}

Out[861]= 124

Out[862]= {c1 → 4.371207212, r1 → 0.005134304353, r2 → -0.02162214448,
 w1 → 15.95079549, w2 → 21.40991504, w3 → 20.78337827,
 w4 → 21.15570269, w5 → 17.83149814, w6 → 14.24156278, w7 → 11.49106217}

Out[864]= {135.0031344, -32.05728188}

Out[865]=

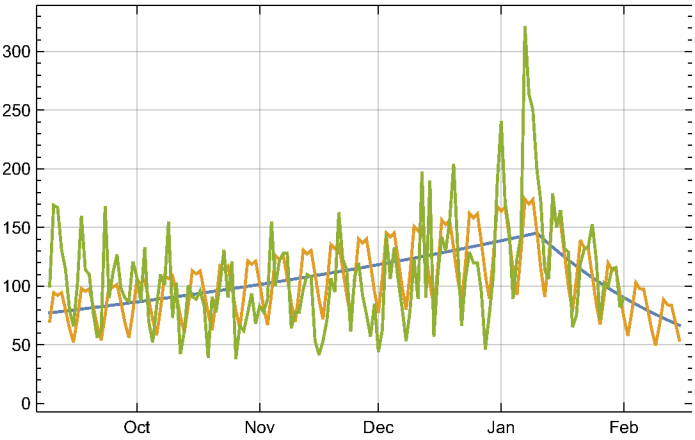


# FIT USING BEST BREAKPOINT

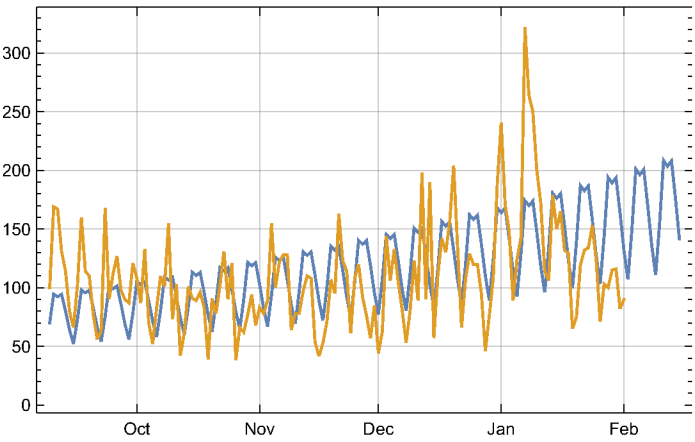
In[866]:=
```
period5CycProj = Take[Flatten[Table[{w1, w2, w3, w4, w5, w6, w7},
     2 + Ceiling[Length[period5Data] / 7]]], Length[period5Data] + 14];
period5FitTabProj = (period5CycProj * Table[twoExpFunc[t, bestPeriod5break,
       c1, r1, r2], {t, Length[period5Data] + 14}]) /. bestPeriod5Params;
period5FuncProj = (Mean@period5CycProj * Table[twoExpFunc[t, bestPeriod5break,
       c1, r1, r2], {t, Length[period5Data] + 14}]) /. bestPeriod5Params;
period5FitTabCont = (period5CycProj * Table[twoExpFunc[t, bestPeriod5break,
       c1, r1, r1], {t, Length[period5Data] + 14}]) /. bestPeriod5Params;
{{r1, r2} /. bestPeriod5Params, TF@Map[rToRt, {r1, r2} /. bestPeriod5Params],
 TF@Round[Map[rToDoublingTime, {r1, r2} /. bestPeriod5Params], 0.1],
 Round[Take[period5FitTabProj, -15], 1]}
(* LPJ@{period5FuncProj,period5FitTabProj,period5Data} *)
DateListPlot[Map[TimeSeries[#, {"Sep 9, 2020"}] &,
  {period5FuncProj, period5FitTabProj, period5Data}],
 PlotRange → All, GridLines → True ]
DateListPlot[Map[TimeSeries[#, {"Sep 9, 2020"}] &,
  {period5FitTabCont, period5Data}], PlotRange → All, GridLines → True ]
DateListPlot[TimeSeries[runningSymMean[
   period5Data - Take[period5FitTabCont, Length[period5Data]], 7],
  {"Sep 9, 2020"}], PlotRange → All, GridLines → {True, {0}}]
(* ListPlot[{period5Data,period5FitTabCont},Joined→True,
  AxesLabel→{"Days after Sept. 9","Cases"}]
 ListPlot[period5Data -Take[period5FitTabCont,Length[period5Data]],
  Joined→True,AxesLabel→{"Days after Sept. 9","Excess in cases"}] *)
(* ListPlot[runningSymMean[period5Data -
   Take[period5FitTabCont,Length[period5Data]],7],Joined→True,
 AxesLabel→{"Days after Sept. 9","Smoothed excess in cases"}] *)
```

Out[870]=
$$\left\{\{0.005134304353, -0.02162214448\}, \begin{matrix} 1.017862151 & 135. \\ 0.9281526011 & -32.1 \end{matrix}, \{73, 58, 78, 103, 98, 97, 80, 63, 50, 67, 89, 84, 84, 69, 54\}\right\}$$

Out[871]=




Out[872]=




Out[873]=