

Exploring Pandemic Data

March 24, 2020 Livestream

Christopher Wolfram, Stephen Wolfram, Bob Nachbar, etc.

Basic Data Exploration

Here is the epidemiological data from the Wolfram Data Repository:

```
In[ ]:= rawCovidData = ResourceData["Epidemic Data for Novel Coronavirus COVID-19"];
```

Show the data available for each region:

```
In[ ]:= rawCovidData[1 /* Keys]
```

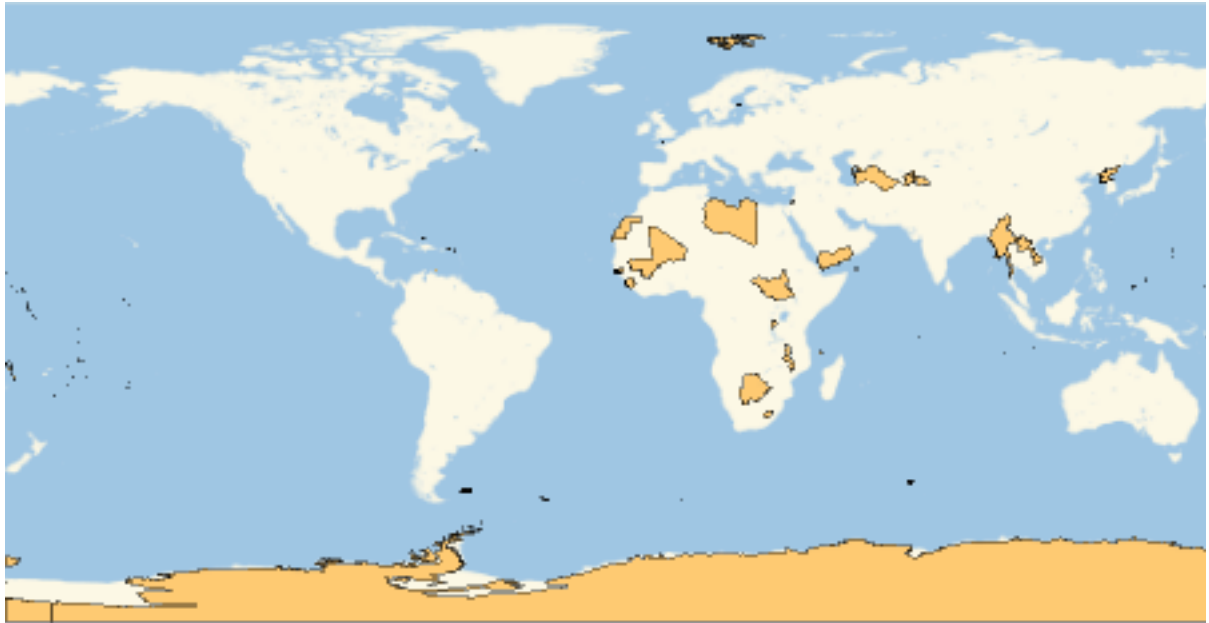
Out[]:=

AdministrativeDivision
Country
GeoPosition
ConfirmedCases
RecoveredCases
Deaths

Show which countries have not reported cases so far:

```
In[ ]:= GeoListPlot[Complement[EntityList["Country"],
  Normal@Keys@rawCovidData[GroupBy["Country"], Total, "ConfirmedCases"]]]
```

Out[]:=



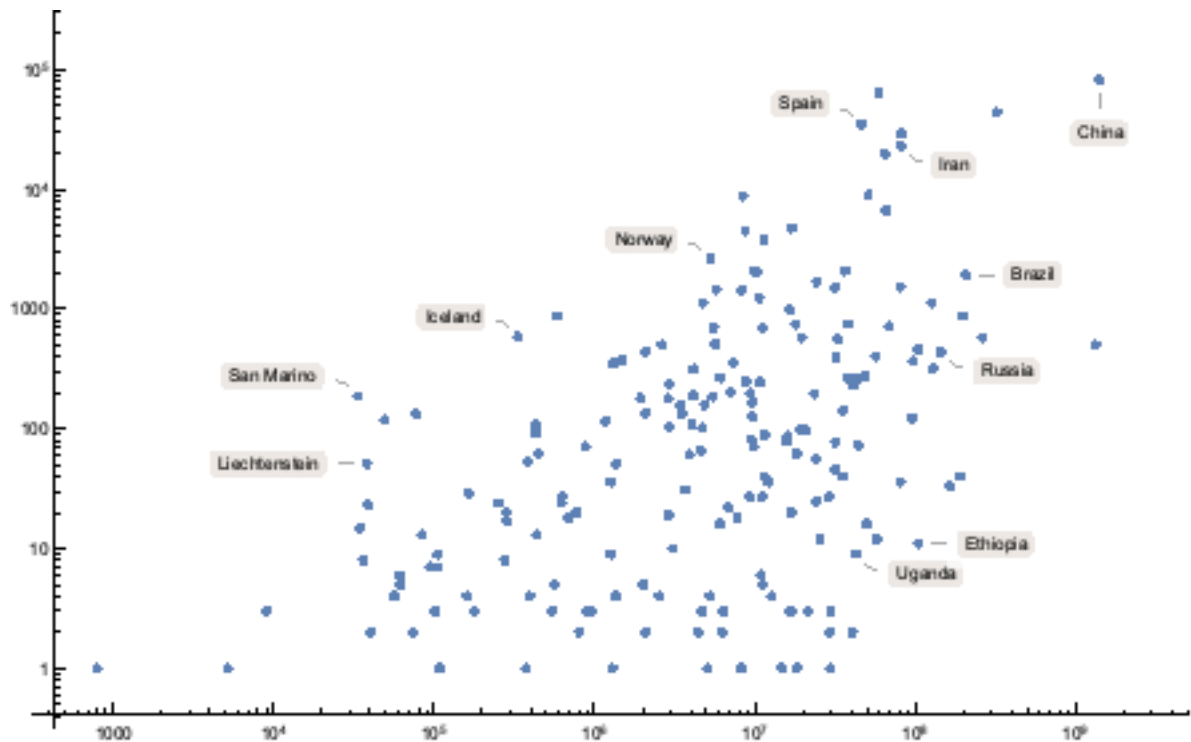
Generate the time series of confirmed cases, totaled by country:

```
In[ ]:= countryTs = Normal[rawCovidData[GroupBy["Country"], Total, "ConfirmedCases"]];
```

Make a log-log plot of the latest reported number of cases as a function of population:

```
In[ ]:= ListLogLogPlot[AssociationThread[Keys[countryTs],
  Transpose[{EntityValue[Keys[countryTs], "Population"],
    #["LastValue"] & /@ Values[countryTs]}]], PlotRange -> All]
```

Out[]:=



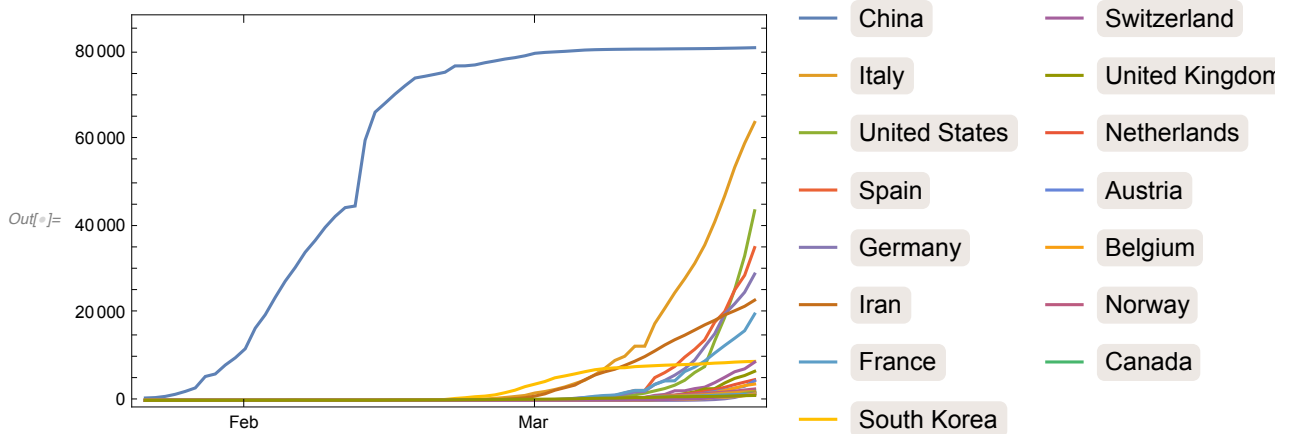
Basic Time Series

Find the raw numbers of confirmed cases for each country:

```
In[ ]:= countryValues = #["Values"] & /@ countryTs;
```

Show growth for the 25 countries with the largest current number of reported cases:

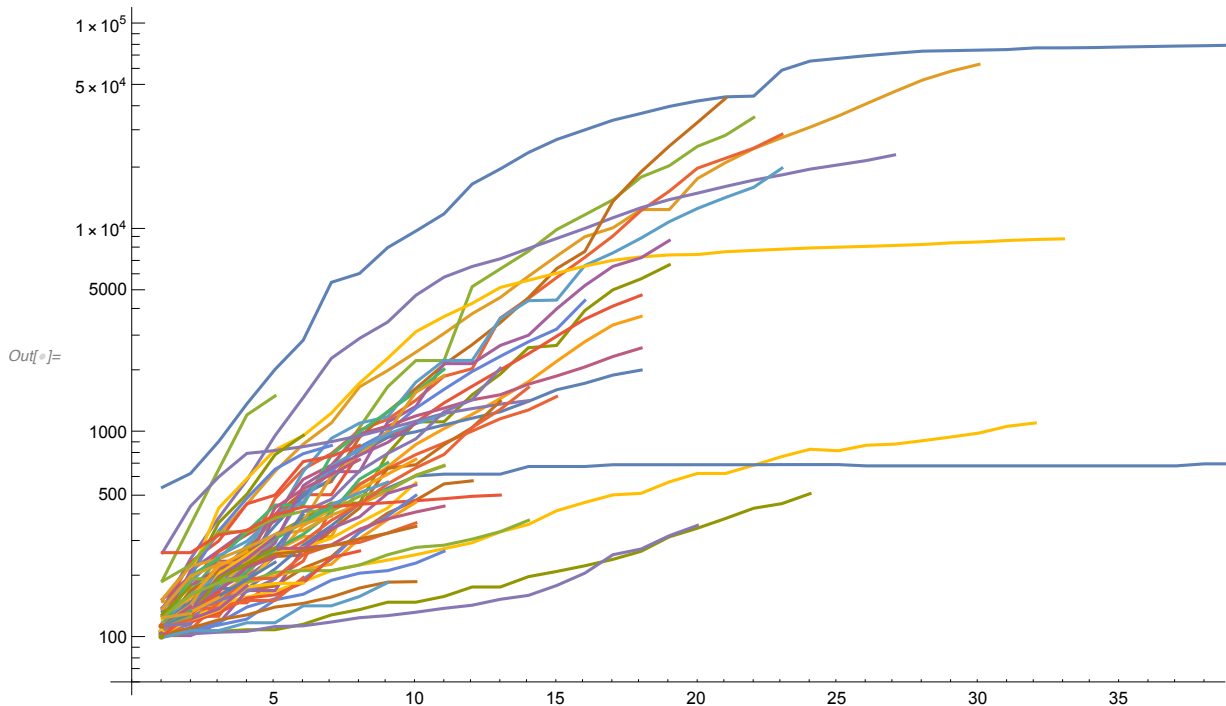
```
In[ ]:= DateListPlot[TakeLargestBy[countryTs, #["LastValue"] &, 25], PlotRange -> All]
```



Make a log plot of the number of cases as a function of time, in each case starting when the number of cases first exceeded 100:

(Countries are indicated by tooltips)

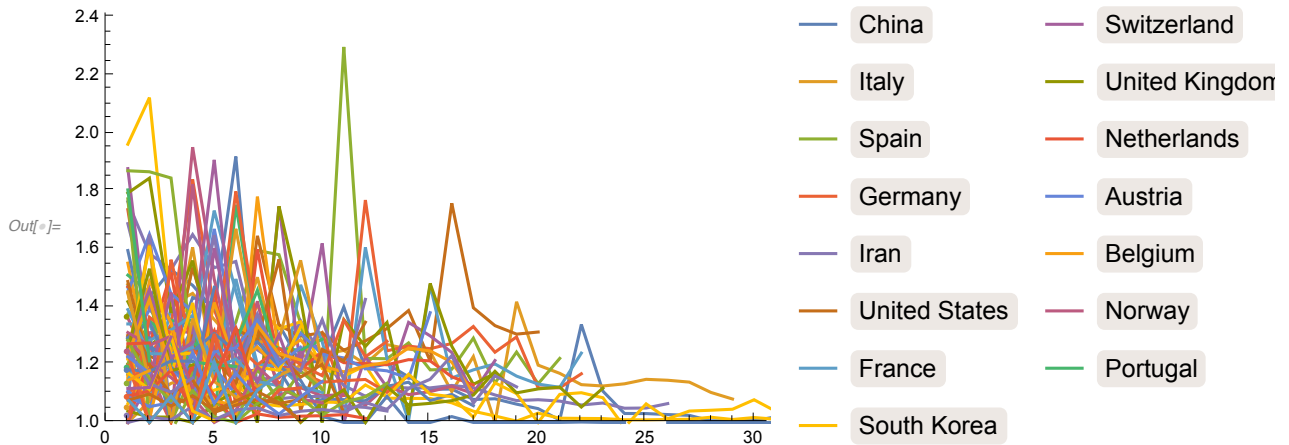
```
In[ ]:= ListLogPlot[KeyValueMap[Tooltip[#2, #1] &,
DeleteCases[Select[GreaterThan[100]] /@ N@countryValues, {}]], Joined -> True]
```



Daily Growth Rates

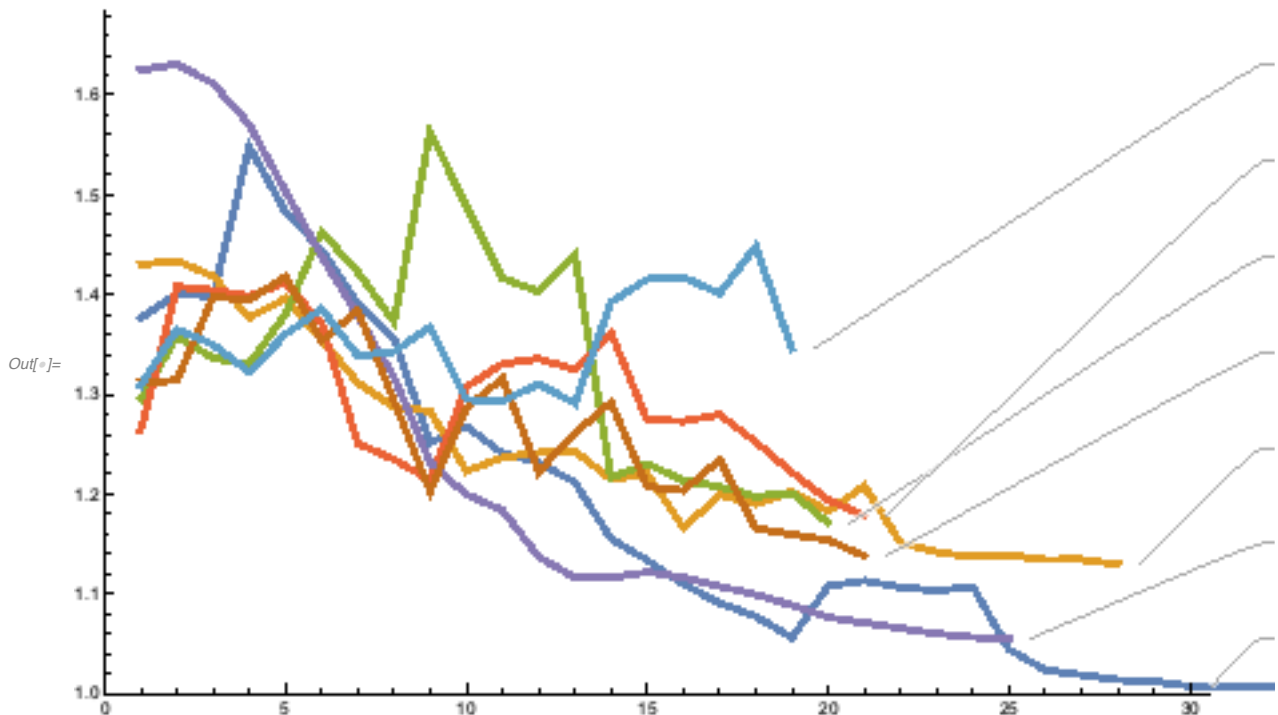
Show the ratio of cases for each country on successive days (starting when each country first identified more than 100 cases):

```
In[ ]:= ListLinePlot[
  DeleteCases[Ratios /@ Select[GreaterThan[100]] /@ N@countryValues, {}],
  PlotRange -> {{0, 30}, {1, All}}
```



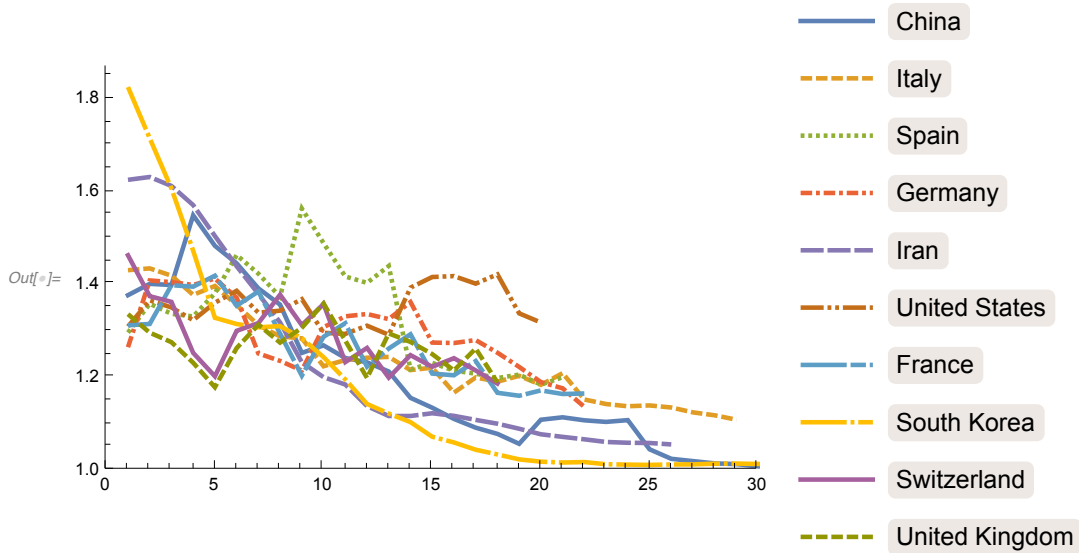
Show daily ratios for countries with more than 10000 cases, smoothed with a radius of 2 days:

```
In[ ]:= ListLinePlot[
  KeyValueMap[Callout[#2, #1] &, DeleteCases[MeanFilter[Ratios[#], 2] & /@
    Select[GreaterThan[100]] /@ N@Select[countryValues, Max[#] > 10 000 &], {}]],
  PlotRange -> {{0, 30}, {1, All}}, PlotStyle -> Thickness[0.007]]
```



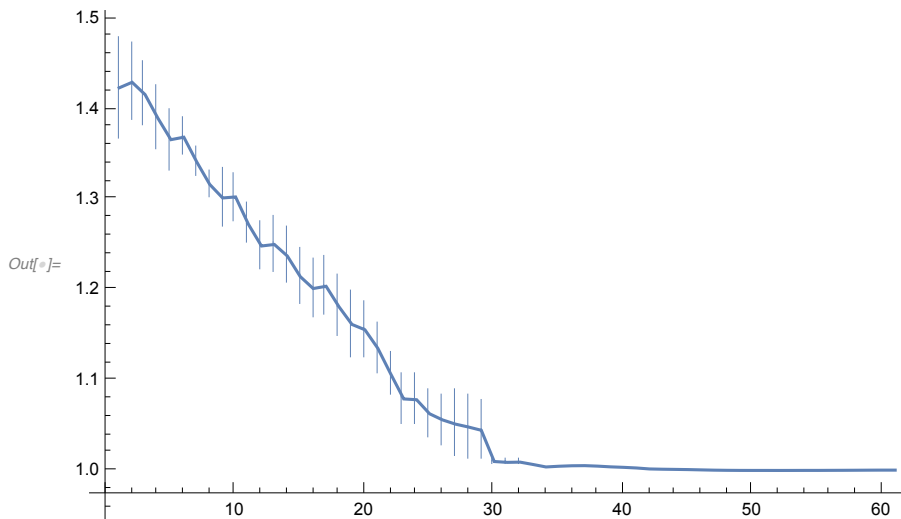
Include all countries with more than 5000 cases:

```
In[ ]:= ListLinePlot[
  KeyValueMap[Legended[#, #1] &, DeleteCases[MeanFilter[Ratios[#], 2] & /@
    Select[GreaterThan[100]] /@ N@Select[countryValues, Max[#] > 5000 &], {}]],
  PlotRange -> {{0, 30}, {1, All}}, PlotTheme -> "DashedLines",
  PlotStyle -> Thickness[0.007]
```



Find the mean daily ratio across countries with more than 5000 cases:

```
In[ ]:= ListLinePlot[
  RaggedMeanAround[Values@DeleteCases[MeanFilter[Ratios[#], 2] & /@
    Select[GreaterThan[100]] /@ N@Select[countryValues, Max[#] > 5000 &], {}]]]
```

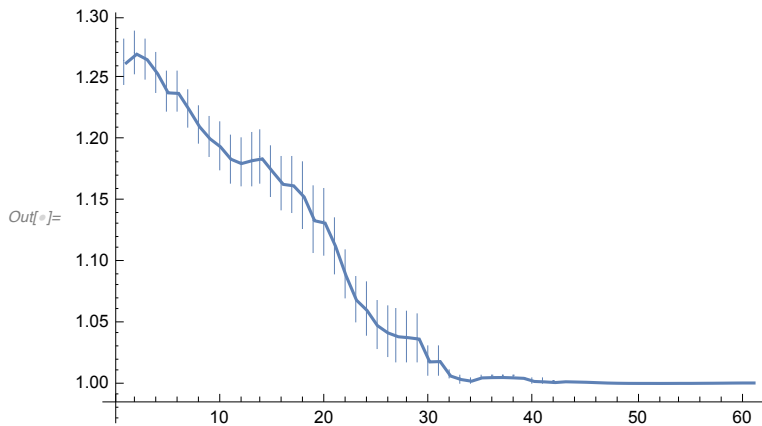


Find the mean daily ratio across all countries reporting cases:

```

In[ ]:= ListLinePlot[
  RaggedMeanAround [Values@DeleteCases[MeanFilter[Ratios[#], 2] & /@
    Select[GreaterThan[100]] /@ N@countryValues, {}]]]

```

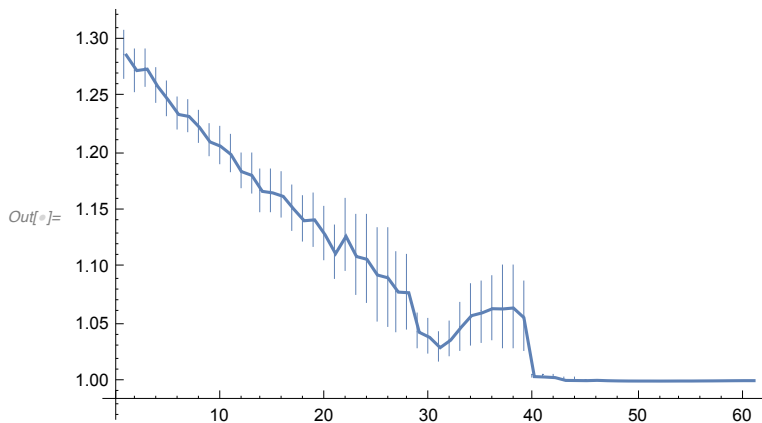


Find the mean across all countries, with the values for each country starting with the country first reported more than 50 cases:

```

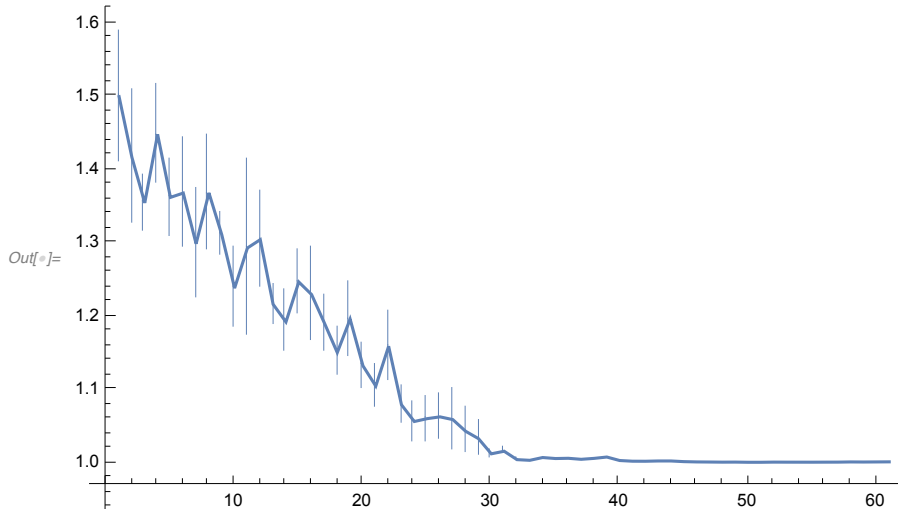
In[ ]:= ListLinePlot[
  RaggedMeanAround [Values@DeleteCases[MeanFilter[Ratios[#], 2] & /@
    Select[GreaterThan[50]] /@ N@countryValues, {}]]]

```



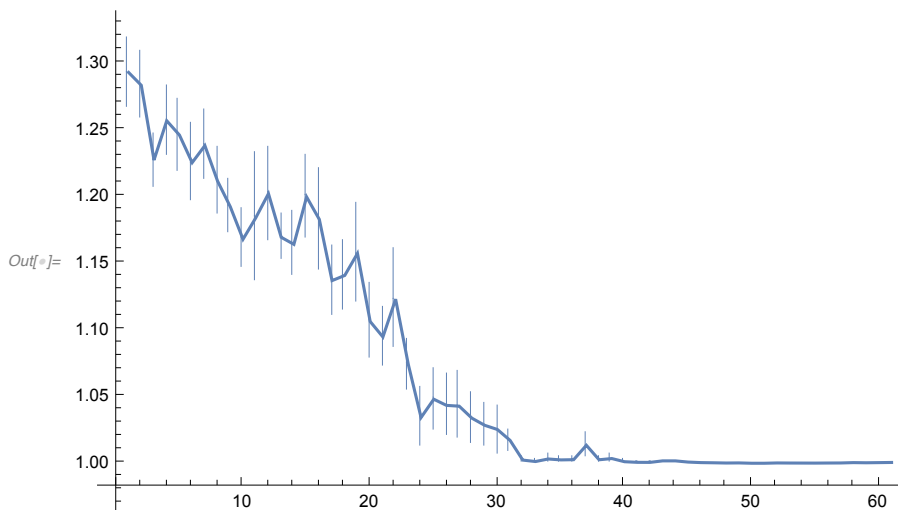
Average over countries with more than 5000 cases, but do not take the mean across days:

```
In[ ]:= ListLinePlot [ RaggedMeanAround [ Values@DeleteCases [ Ratios /@
Select [ GreaterThan [ 100 ] ] /@ N@Select [ countryValues, Max [ # ] > 5000 & ], { } ] ]
```



Average over all countries reporting cases, but do not take the mean across days:

```
In[ ]:= ListLinePlot [ RaggedMeanAround [
Values@DeleteCases [ Ratios /@ Select [ GreaterThan [ 100 ] ] /@ N@countryValues, { } ] ]
```



Investigating Results

We wanted to understand the seemingly linear decrease in average daily ratios.

Find the linear term in a fit of the first 30 days of the data:

```
In[ ]:= fitSlope = m /. FindFit [ RaggedMeanAround [
Values [ DeleteCases [ Ratios /@ Select [ GreaterThan [ 100 ] ] /@ N@countryValues,
{ } ] ] [ [ All, ;; UpTo [ 30 ] ] ] ], m * x + b, { m, b }, x ]
```

Out[]:= -0.0090 ± 0.0005

This corresponds to change of average daily ratio with a slope of about 1 in 111 days:

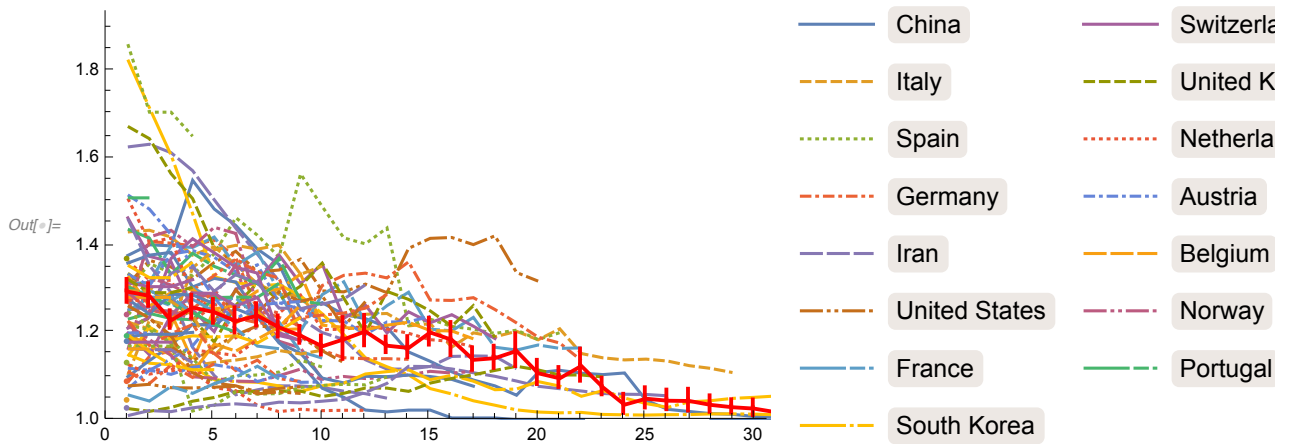
```
In[ ]:= 1 / fitSlope
```

```
Out[ ]:= -111. ± 6.
```

Summarizing Country Daily Ratio Data

Show daily ratio by country, together with the average over all reporting countries:

```
In[ ]:= Show[ {
  ListLinePlot[DeleteCases[MeanFilter[Ratios[#], 2] & /@
    Select[GreaterThan[100]] /@ N@countryValues, {}],
    PlotRange -> {{0, 30}, {1, All}}, PlotTheme -> "DashedLines"],
  ListLinePlot[  RaggedMeanAround  [Values@
    DeleteCases[Ratios /@ Select[GreaterThan[100]] /@ N@countryValues, {}]],
    PlotStyle -> Directive[Thick, Red] ]
}
```

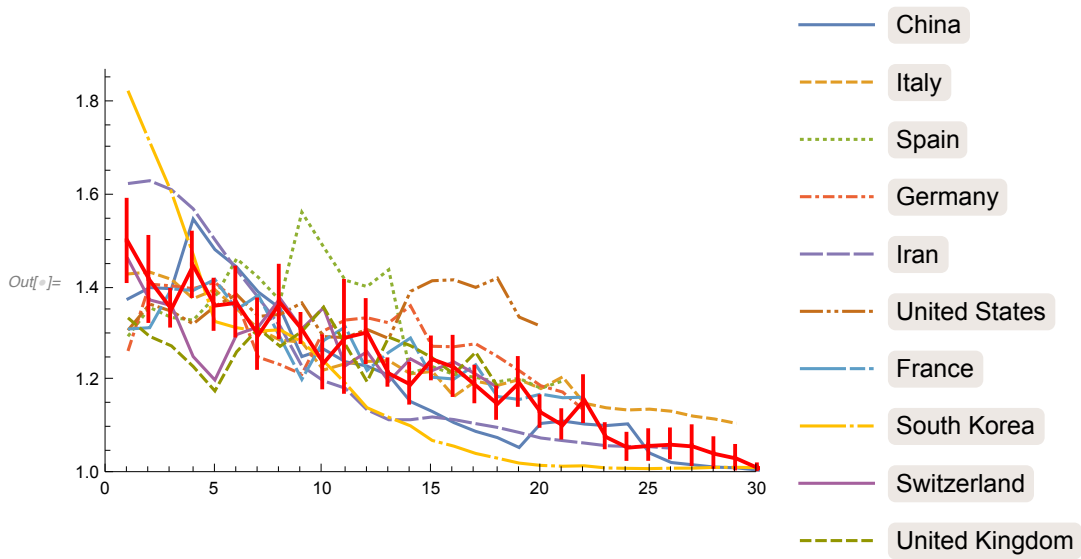


Include only countries with more than 5000 cases reported:


```

In[ ]:= Show[ {
  ListLinePlot[
    KeyValueMap[Legended[#2, #1] &, DeleteCases[MeanFilter[Ratios[#], 2] & /@
      Select[GreaterThanOrEqual[100]] /@ N@Select[countryValues, Max[#] > 5000 &], {}]],
    PlotRange -> {{0, 30}, {1, All}}, PlotTheme -> "DashedLines"],
  ListLinePlot[
    RaggedMeanAround[Values@DeleteCases[Ratios /@
      Select[GreaterThanOrEqual[100]] /@ N@Select[countryValues, Max[#] > 5000 &],
      {}]], PlotStyle -> Directive[Thick, Red]
  ]
} ]

```

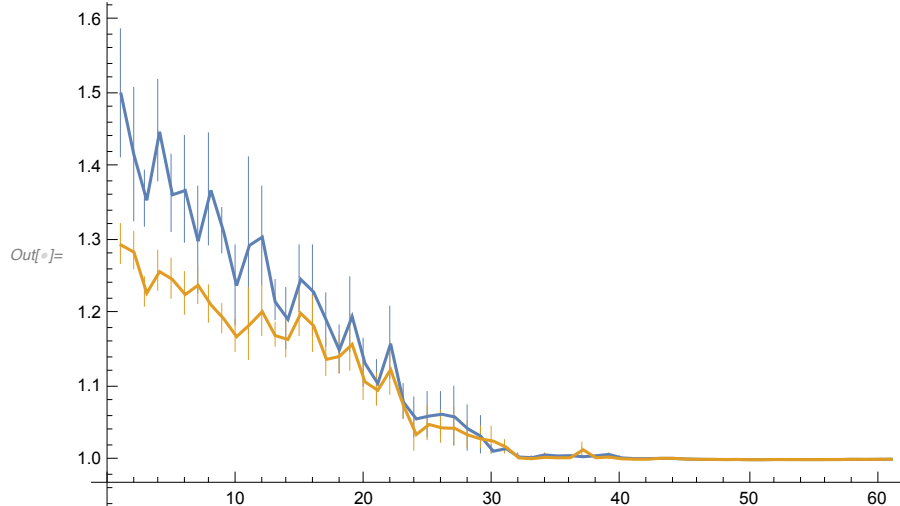


Compare results for all countries, and countries with more than 5000 cases:

```

In[ ]:= ListLinePlot[ {
  RaggedMeanAround [Values@DeleteCases[Ratios /@ Select[GreaterThan[100]] /@
    N@Select[countryValues, Max[#] > 5000 &], {}]],
  RaggedMeanAround [Values@DeleteCases [
    Ratios /@ Select[GreaterThan[100]] /@ N@countryValues, {}]]
} ]

```

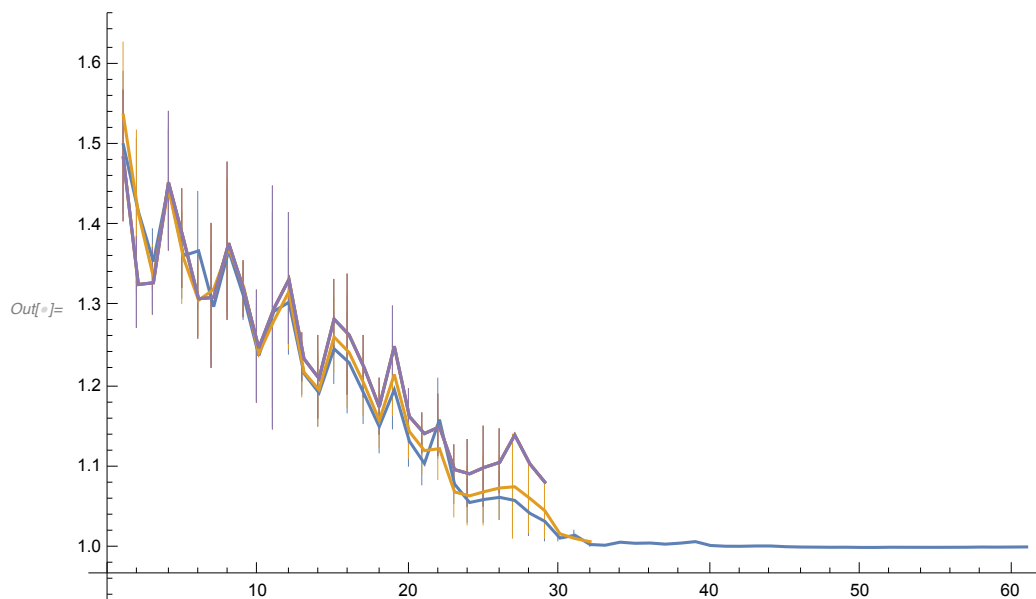


Show results successively dropping certain countries:

```

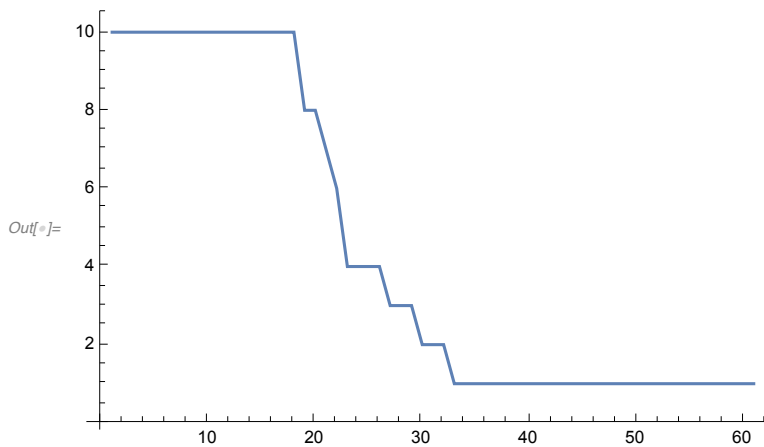
In[ ]:= ListLinePlot [ RaggedMeanAround [
  Values@DeleteCases[Ratios /@ Select[GreaterThan[100]] /@ #, {}]] & /@
  N@FoldList[KeyDrop, Select[countryValues, Max[#] > 5000 &],
    { China COUNTRY, South Korea COUNTRY, Singapore COUNTRY, Japan COUNTRY }]]

```



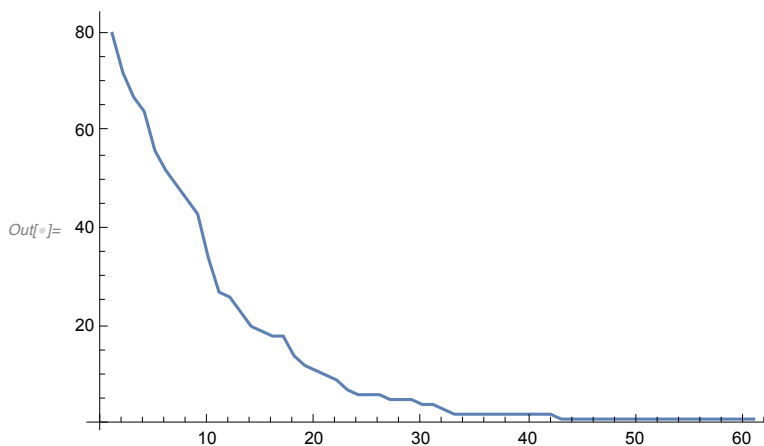
Show how many countries are included in the averages for each day, including only countries with more than 5000 current cases:

```
In[ ]:= ListLinePlot@
  Total@PadRight[Sign[Values@DeleteCases[Ratios /@ Select[GreaterThanOrEqual[100]] /@
    N@Select[countryValues, Max[#] > 5000 &], {}]]]
```



Show how many countries are included in the averages for each day, including all countries reporting cases:

```
In[ ]:= ListLinePlot@Total@PadRight[Sign[Values@
  DeleteCases[Ratios /@ Select[GreaterThanOrEqual[100]] /@ N@countryValues, {}]]]
```



Possible Model for Results

In the standard SIR continuum epidemiological model, the number of infected people is $i[t]$, and there is a “force of infection” β .

Solve assuming an infinite supply of susceptible people, and fixed force of infection; the result is a pure exponential:

```
In[ ]:= DSolve[i'[t] == \beta * i[t], i[t], t]
```

$$Out[]:= \left\{ \left\{ i[t] \rightarrow e^{t\beta} c_1 \right\} \right\}$$

Solve assuming a force of infection that varies linearly with time:

```
In[ ]:= DSolve[i'[t] == (m * t + b) * i[t], i[t], t]
```

$$Out[]:= \left\{ \left\{ i[t] \rightarrow e^{b t + \frac{m t^2}{2}} c_1 \right\} \right\}$$

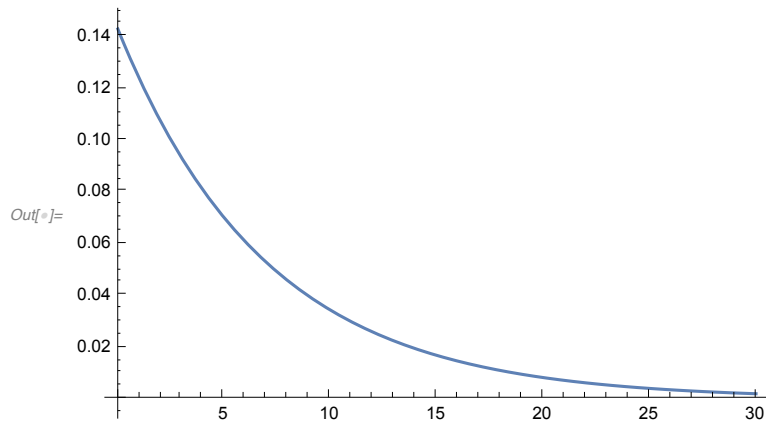
`In[]:= D[Log[eb t + $\frac{m t^2}{2}$ c1], t]`

`Out[]:= b + m t`

A typical model is that the distribution of times between becoming infected and showing symptoms is an exponential distribution.

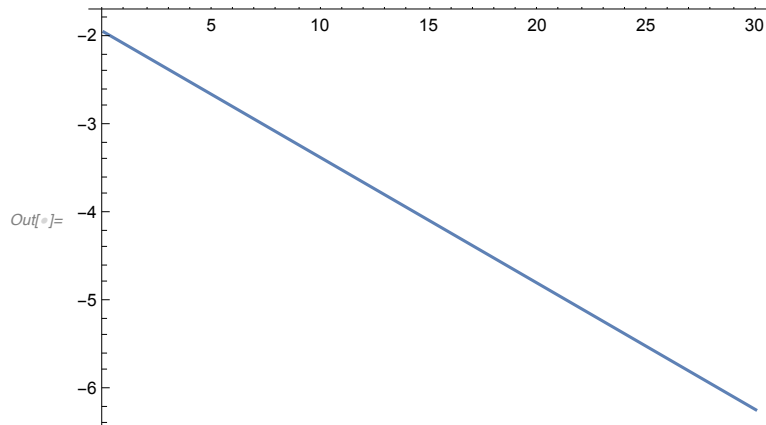
Show the PDF for an exponential distribution:

`In[]:= Plot[PDF[ExponentialDistribution[1 / 7], t], {t, 0, 30}]`



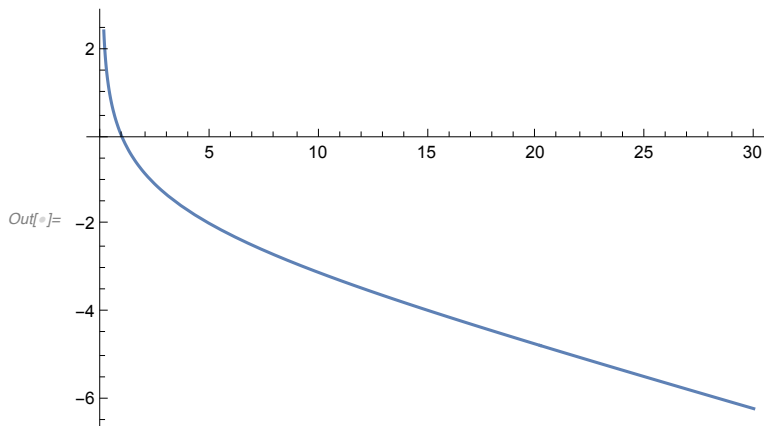
The ratio of successive values will be given roughly by the log of the PDF:

`In[]:= Plot[Log[PDF[ExponentialDistribution[1 / 7], x]], {x, 0, 30}]`



More accurately, it is the ratio of PDF to CDF:

```
In[ ]:= Plot[Log[PDF[ExponentialDistribution[1 / 7], x] /
CDF[ExponentialDistribution[1 / 7], x]], {x, 0, 30}]
```

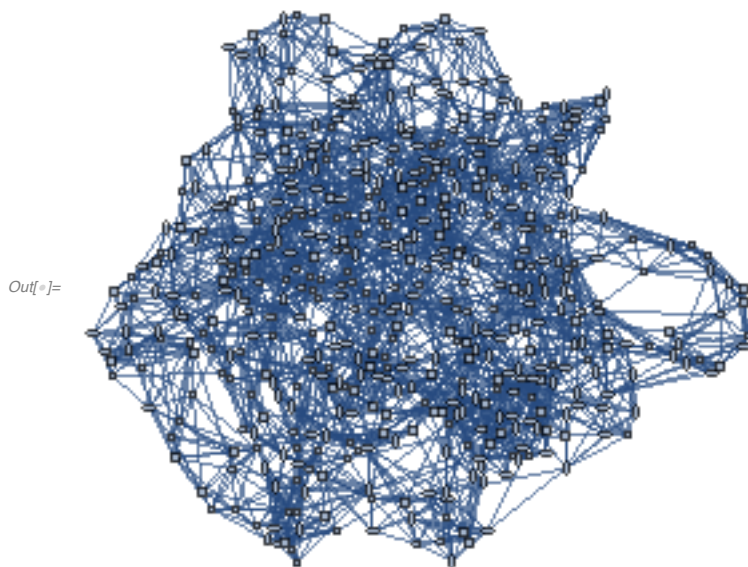


Network-Based Modeling

The continuum SIR model does not accurately represent human contacts, especially when they are limited by social distancing. It is better to consider a network, although it is not clear what the correct network should be.

Generate a typical example of network that models certain features of human networks:

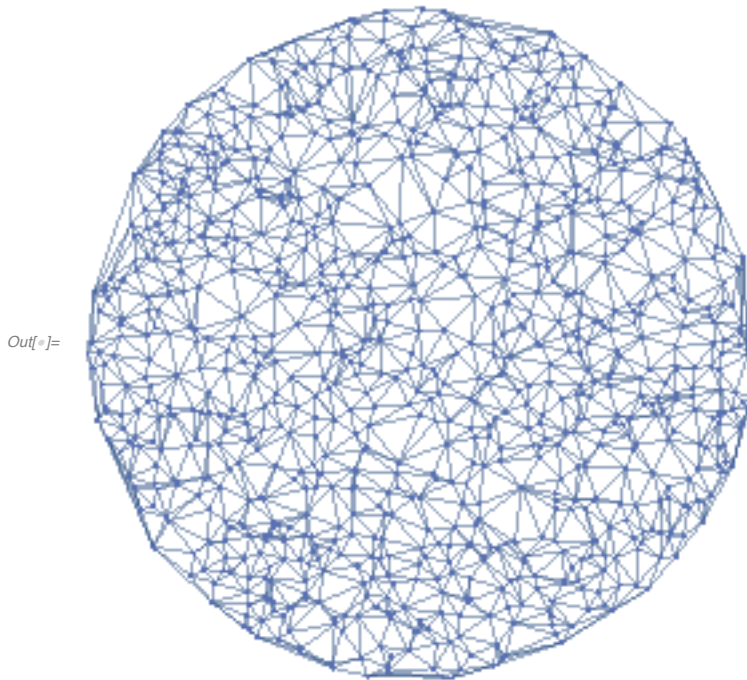
```
In[ ]:= RandomGraph[WattsStrogatzGraphDistribution[500, 0.1, 5]]
```



At larger scales, human networks will tend to reflect actual geographical (i.e. spatial) relations, and so will have features of random planar networks.

Generate an example of a random planar network:

```
In[ ]:= MeshConnectivityGraph[DeLaunayMesh[RandomPoint[Disk[]], 1000]]
```




Make larger examples of these types of graphs:

```
In[ ]:= g1 = RandomGraph[WattsStrogatzGraphDistribution[10 000, 0.1, 5]]
```

Out[]:= Graph [ Vertex count: 10 000
Edge count: 50 000]

```
In[ ]:= g2 = MeshConnectivityGraph[DeLaunayMesh[RandomPoint[Disk[]], 100 000]]
```

Out[]:= Graph [ Vertex count: 100 000
Edge count: 299 826]

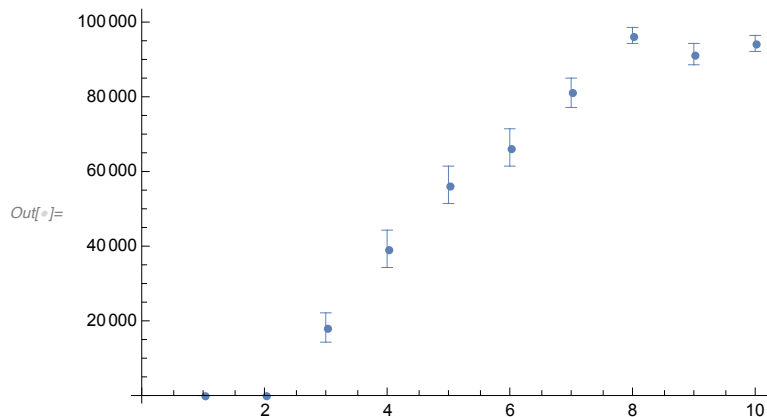
For the model human network, the graph diameter is still quite small:

```
In[ ]:= GraphDiameter [g1]
```

Out[]:= 10

Starting from one node in the graph (i.e. one person) this shows the number of nodes reached after n steps in the random planar graph:

```
In[ ]:= ListPlot@Table[MeanAround@N@Table[Length@
VertexOutComponent[g2, RandomChoice@VertexList[g2], n], 100], {n, 10}]
```



Data from Actual Contagion Networks

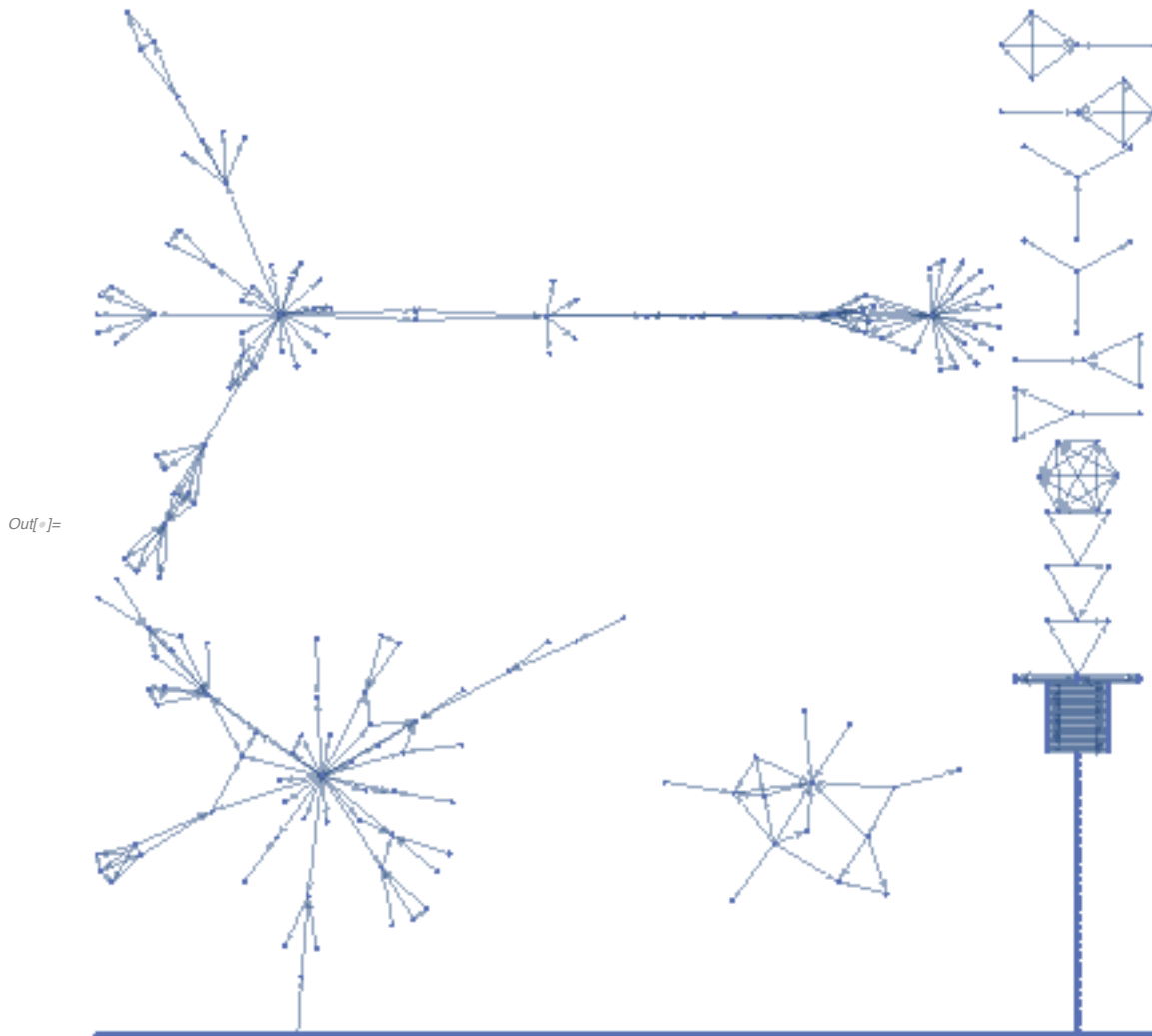
Singapore has carefully tracked cases, and generated a network giving information on contagion.

Import the data:

```
In[ ]:= rawGraph = Import["https://co.vid19.sg/api/cases/network_graph", "RawJSON"];
```

Show the network from this data:

```
In[ ]:= g = VertexDelete[Graph[rawGraph["nodes"]][[All, "id"]],
  #from -> #to & /@ rawGraph["edges"], "unknown"]
```

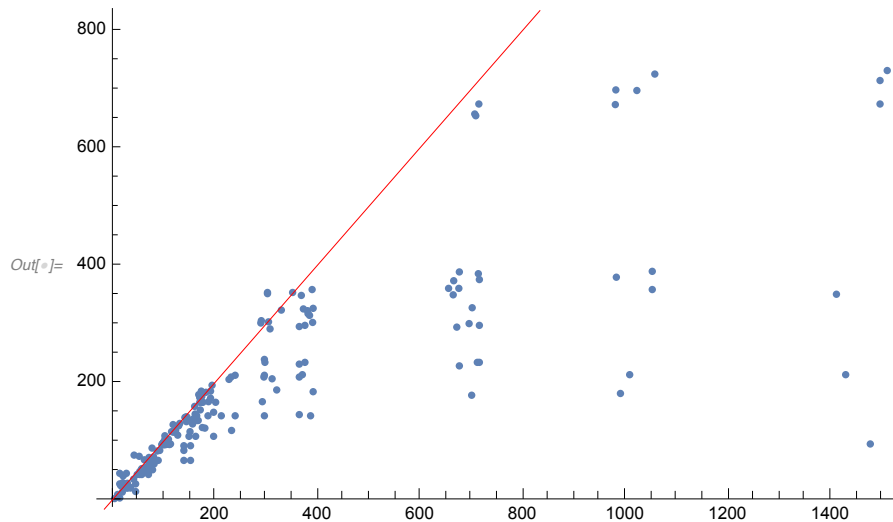


Find cases of person-to-person transmission, giving the case numbers involved:

```
In[ ]:= transmissions = Map[ToExpression, StringDrop[#, 5] & /@
  Select[DeleteMissing[Lookup[{"from", "to"}] /@ rawGraph["edges"], 1, 2],
    AllTrue[StringMatchQ["Case " ~~ ___]]]]
```

Plot case numbers for transmission pairs:


```
In[ ]:= ListPlot[transmissions, Epilog -> {Red, InfiniteLine[{0, 0}, {1, 1}]}]
```



This data seems to indicate that most transmissions are found by backtracing. The analysis should be repeated with actual report times included.

Analysis of Government Responses

Import a dataset of measures implemented by governments:

```
In[ ]:= governmentMeasures = ResourceData[ResourceObject[CloudObject[
  "https://www.wolframcloud.com/obj/christianp/DeployedResources/Data/
  Coronavirus-COVID-19-Pandemic-Government-Measures"]]];

```

Show counts of measures implemented:

In[*]:= `governmentMeasures[Counts /* ReverseSort, "Measure"]`

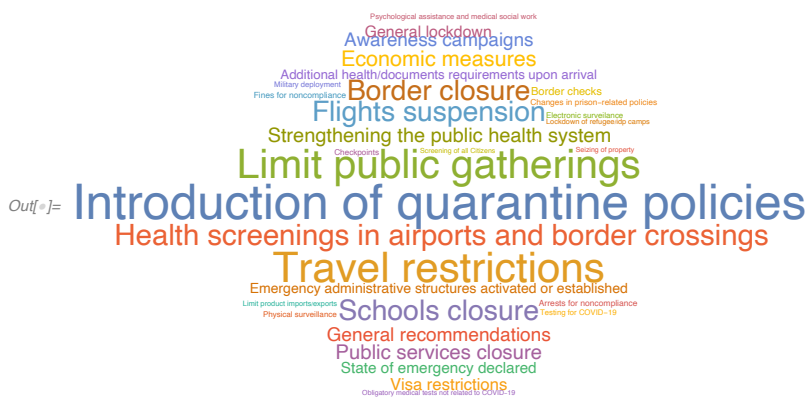
Out[*]=

Introduction of quarantine policies
Travel restrictions
Limit public gatherings
Health screenings in airports and border crossings
Schools closure
Flights suspension
Border closure
Economic measures
Public services closure
Strengthening the public health system
General recommendations
Awareness campaigns
Visa restrictions
General lockdown
State of emergency declared
Emergency administrative structures activated or established
Additional health/documents requirements upon arrival
Border checks
Arrests for noncompliance
Fines for noncompliance

rows 1-20 of 32

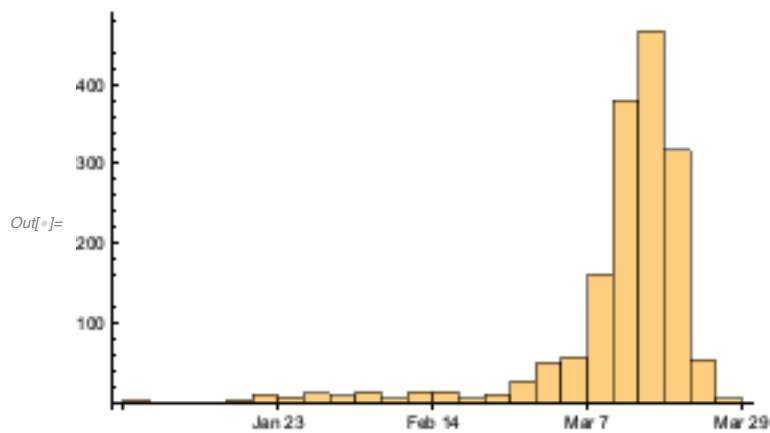
Show a word cloud of measures taken:

In[*]:= `governmentMeasures[WordCloud, "Measure"]`



Show a date histogram of when measures were implemented:

```
In[ ]:= governmentMeasures[DateHistogram, "ImplementationDate"]
```

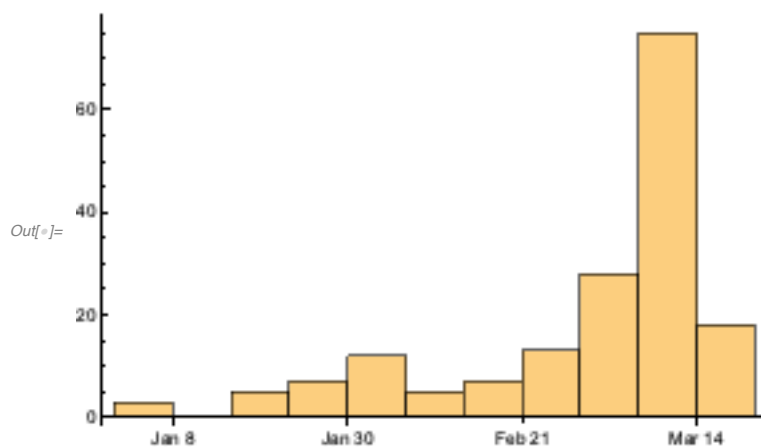


```
In[ ]:= minImplementationTime =
```

```
    DateObject /@ Select[Normal[governmentMeasures[GroupBy["Country"], All,
    "ImplementationDate"][All, DeleteMissing /* Min, AbsoluteTime]], NumberQ];
```

Show a histogram of when the first measure was implemented for each country:

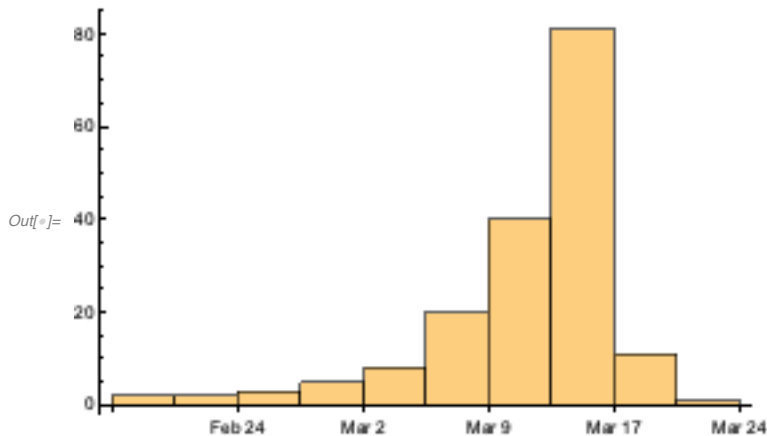
```
In[ ]:= DateHistogram[minImplementationTime]
```



```
In[ ]:= meanImplementationTime = DateObject /@ Select[
    Normal[governmentMeasures[GroupBy["Country"], All, "ImplementationDate"][
    All, DeleteMissing /* Mean, AbsoluteTime]], NumberQ];
```

Show a histogram of the average time when measures were implemented:

```
In[ ]:= DateHistogram[meanImplementationTime]
```



Compute when general lockdown was implemented in each country:

```
In[ ]:= lockdownTimes =
  Select[Normal@governmentMeasures[Select[#Measure === "General lockdown" &] /*
    GroupBy[Lookup["Country"] → Lookup["ImplementationDate"]],
    DeleteMissing /* Map[AbsoluteTime] /* Mean /* DateObject], DateObjectQ];
```

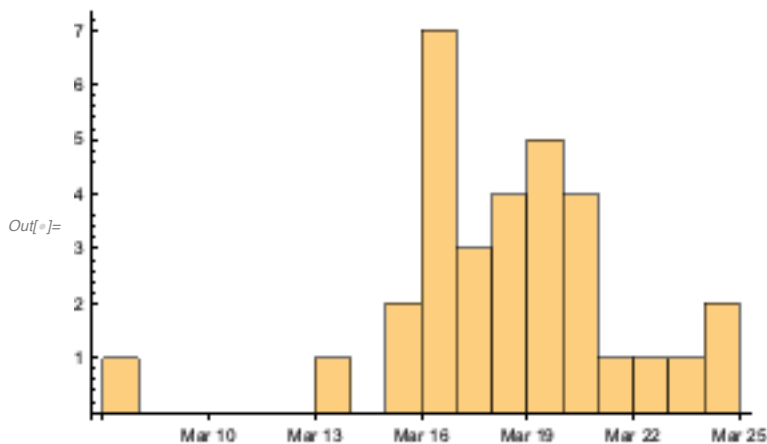
Find how many countries have implemented general lockdown so far:

```
In[ ]:= Length[lockdownTimes]
```

```
Out[ ]:= 32
```

Make a histogram of when general lockdown was implemented:

```
In[ ]:= DateHistogram[lockdownTimes, 1 day]
```



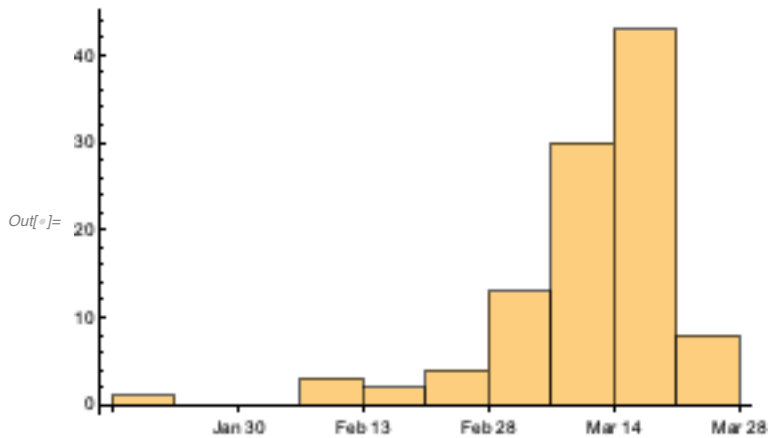
Comparing with Cases

Compute when each country first reported more than 100 cases:

```
In[ ]:= over100 = DeleteMissing[
  SelectFirst[#[["DatePath"]], #[[2]] > 100 &] & /@ countryTs][[All, 1]];
```

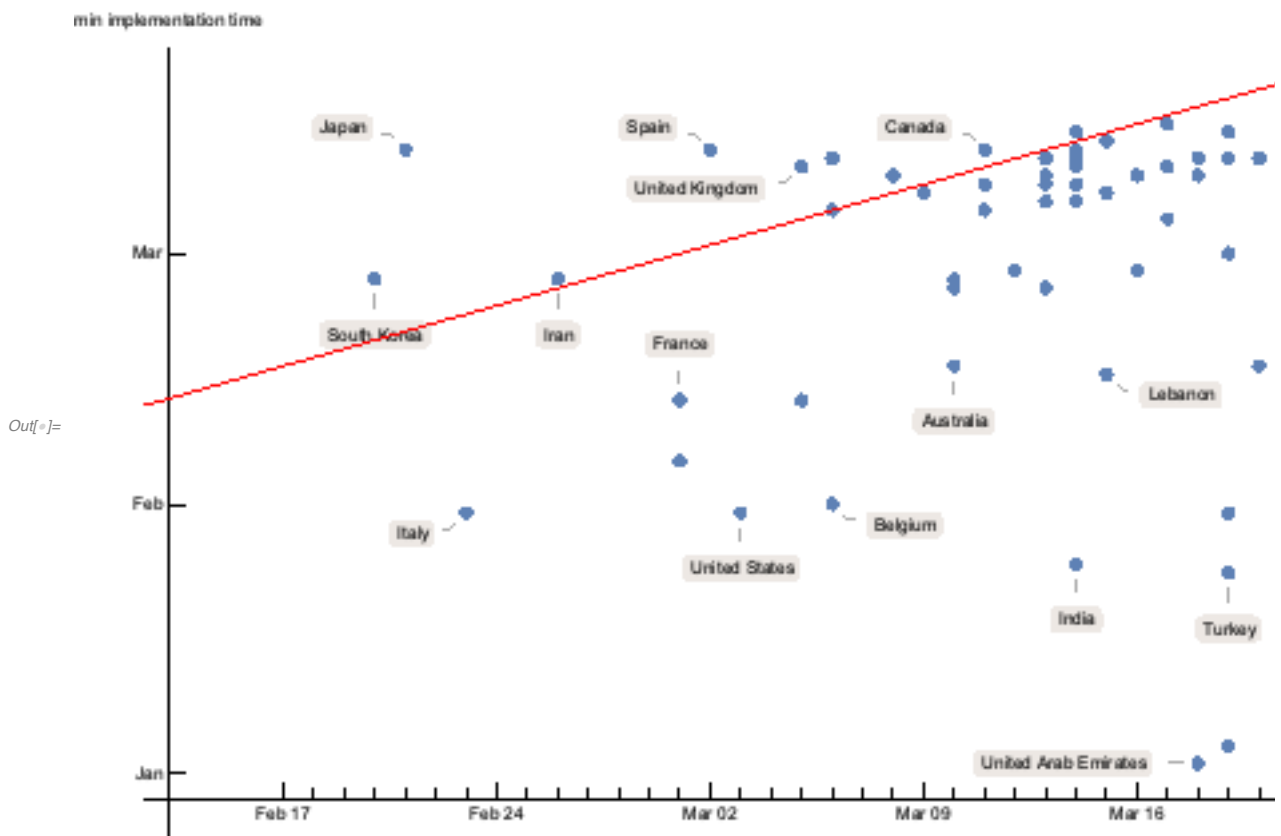
Make a date histogram of when more than 100 cases were first reported:

In[]:= DateHistogram[over100]



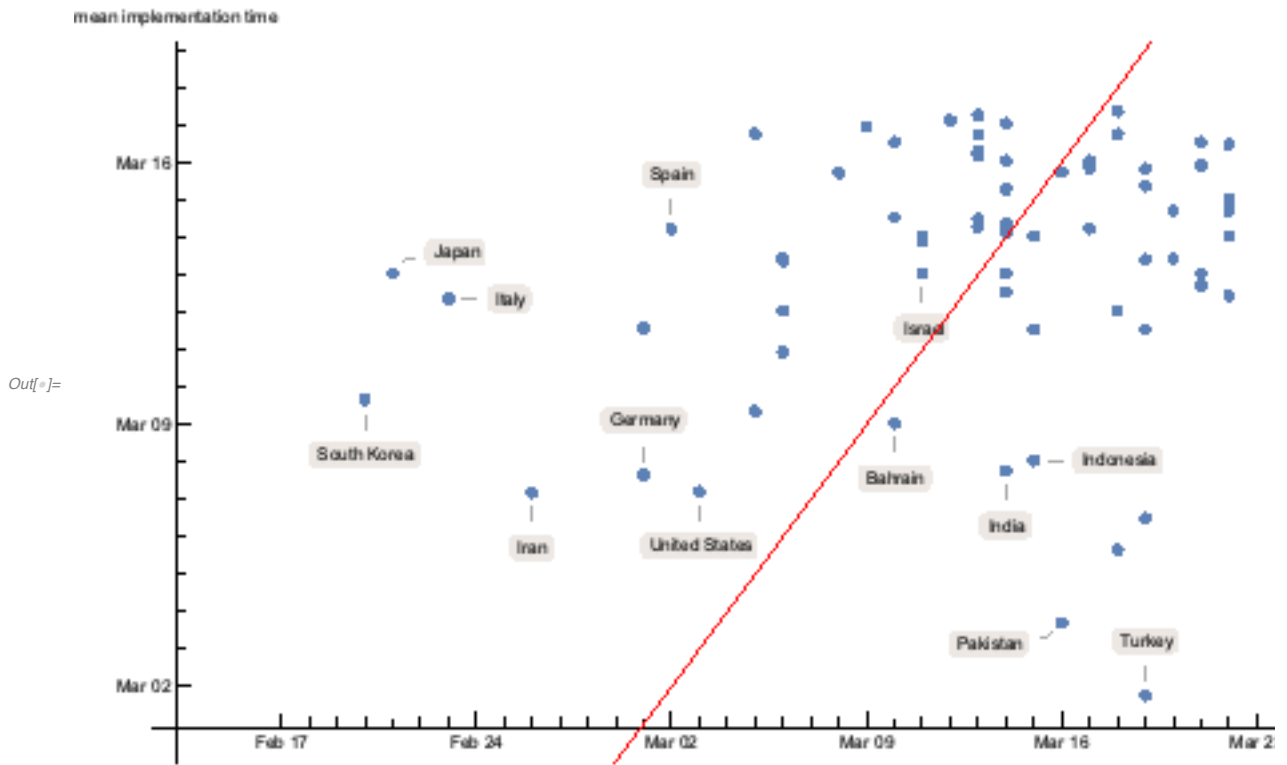
Compare when countries first implemented measures vs. when they first reported 100 cases:

In[]:= ListPlot[Map[AbsoluteTime, Merge[KeyIntersection[{over100, minImplementationTime}], Identity], {2}], AxesLabel -> {"over 100", "min implementation time"}, ...]



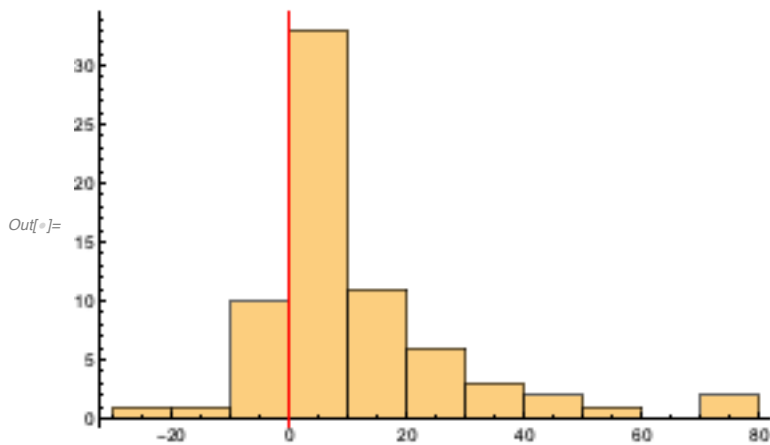
Compare the average of when countries implemented their various measures vs. when they first reported 100 cases:

```
In[ ]:= ListPlot[Map[AbsoluteTime,
  Merge[KeyIntersection[{over100, meanImplementationTime}], Identity], {2}],
  AxesLabel -> {"over 100", "mean implementation time"}, ...]
```



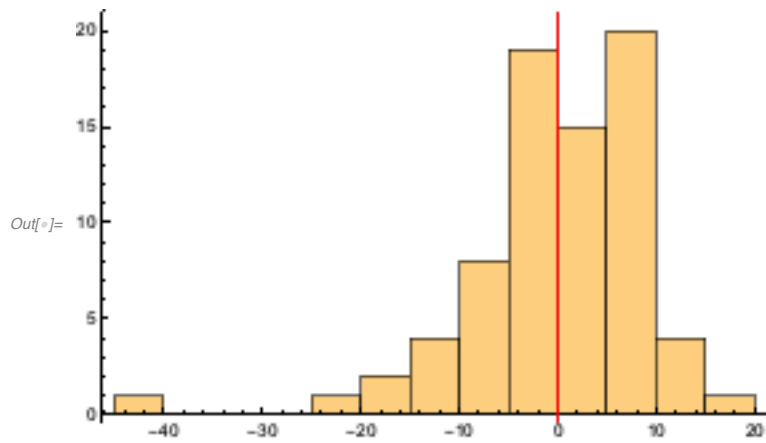
Compare the difference in time between reporting 100 cases, and the first measure being implemented:

```
In[ ]:= Histogram[Merge[KeyIntersection[{over100, minImplementationTime}],
  Apply[Subtract]], Epilog -> {Red, InfiniteLine[{0, 0}, {0, 1}]}]
```






Compare the difference in time between reporting 100 cases, and the average of when measures were implemented:

```
In[ ]:= Histogram[Merge[KeyIntersection[{over100, meanImplementationTime}],  
Apply[Subtract]], Epilog -> {Red, InfiniteLine[{0, 0}, {0, 1}]}]
```



Posted Graphic

```

In[ ]:= With[{countryValues = #["Values"] & /@
  Normal[ResourceData["Epidemic Data for Novel Coronavirus COVID-19"][
    GroupBy["Country"], Total, "ConfirmedCases"]]}, Show[{
  ListLinePlot[KeyValueMap[Labeled[#2, ImageResize[#1["FlagImage"], 20],
    {...} +] &, DeleteCases[MeanFilter[Ratios[#], 2] & /@ Select[
    GreaterThan[100]] /@ N@Select[countryValues, Max[#] > 1000 &], {}]]],
  ListLinePlot[ RaggedMeanAround + [Values@DeleteCases[
    Ratios /@ Select[GreaterThan[100]] /@
    N@Select[countryValues, Max[#] > 1000 &], {}]],  Rule[... +]
  },
   +]]]

```