Description

- Here we used the “Daily new confirmed cases of COVID-19 per people (rolling 7-day average)” for 201 countries from Our World In Data
- The objectives of this analysis are:
  - Estimate the kernel values according to the proposed approach (describe in equation 3 in the paper).
  - Evaluate the utility of λ to explore other kernels values to improve the waves detection while keeping the possible kernel values in a delimited range.
  - Evaluate the robustness of the method when the reporting frequencies change.

Initialization

Functions

Data

Effect of the reporting frequency

- The kernel value was estimated (as proposed) for each country, set of data (i.e., daily, weekly, monthly), and the three proposed λ : 0.5, 1, 1.5
- Initialization

In[452]:=

```
countriesS = Import["kernelByCountryW.csv"]⟦2 ;; , 1⟧;
countries = Interpreter["Country"][#] & /@ countriesS;
```

Data

Upload

Kernel for data

- Daily
- Weekly
- Monthly
- All

Waves detected daily

Waves detected weekly

Waves detected monthly

Waves detected daily plots

Waves detected weekly plots

Waves detected monthly plots

Comparing Waves detected daily , weekly, and monthly

- Here we make a different calculation of the variance in date detection
- firstDateByCountry
- Exploring
- Final countries

In[535]:=

```
countriesSFinal = DeleteCases[countriesS, "SaintVincentGrenadines" |
    "PitcairnIslands" | "SaintHelena" | "Samoa" | "SanMarino" | "Micronesia"];
```

- Results λ : 1

In[533]:=

```
kernels = allKernel[All, {"Country", "KD", "KW", "KM"}];
```

In[536]:=

```
results = Association[wavesByKByCountryGranularity2[#, wavesDaily,
      wavesWeekly, wavesMonthly, kernels] & /@ countriesSFinal];
```

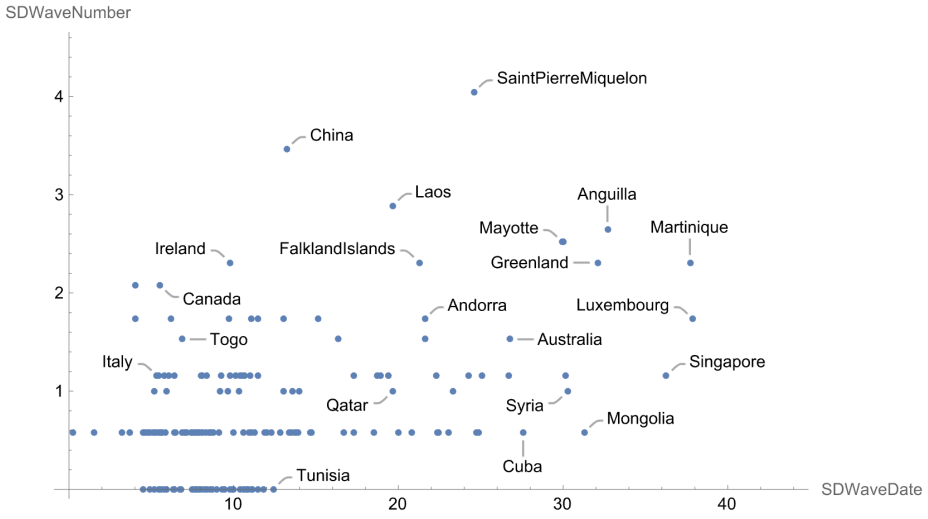
In[537]:=

```
resultsCluster2var = DeleteCases[results⟦All, {2, 4}⟧, {___, {}, ___}];
```

In[539]:=

```
ListPlot[resultsCluster2var, AxesLabel → {"SDWaveDate", "SDWaveNumber"}]
```

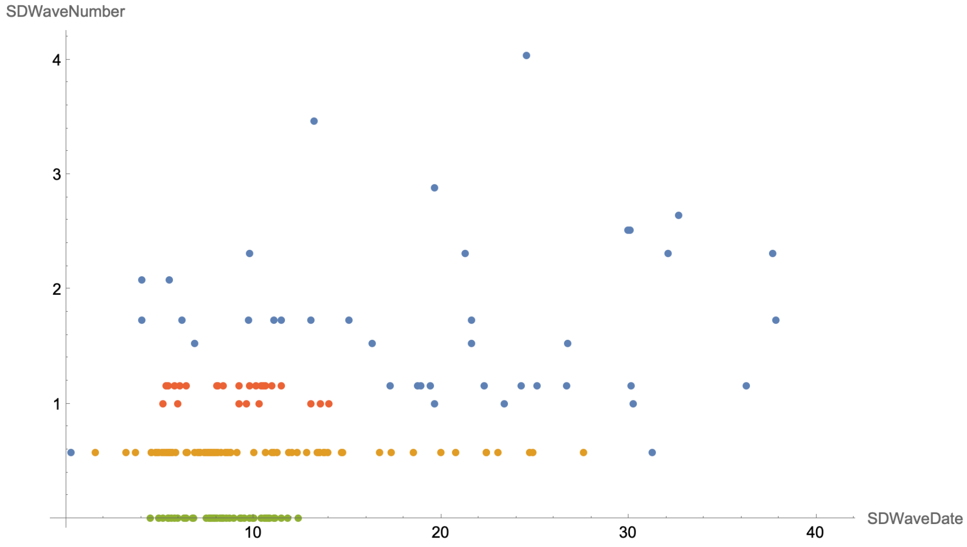
Out[•]=



In[540]:=

```
countriesGroupsPlotm12var = ListPlot[FindClusters[resultsCluster2var // Values],
   AxesLabel → {"SDWaveDate", "SDWaveNumber"}]
countriesGroupsm12var = FindClusters[resultsCluster2var];
```

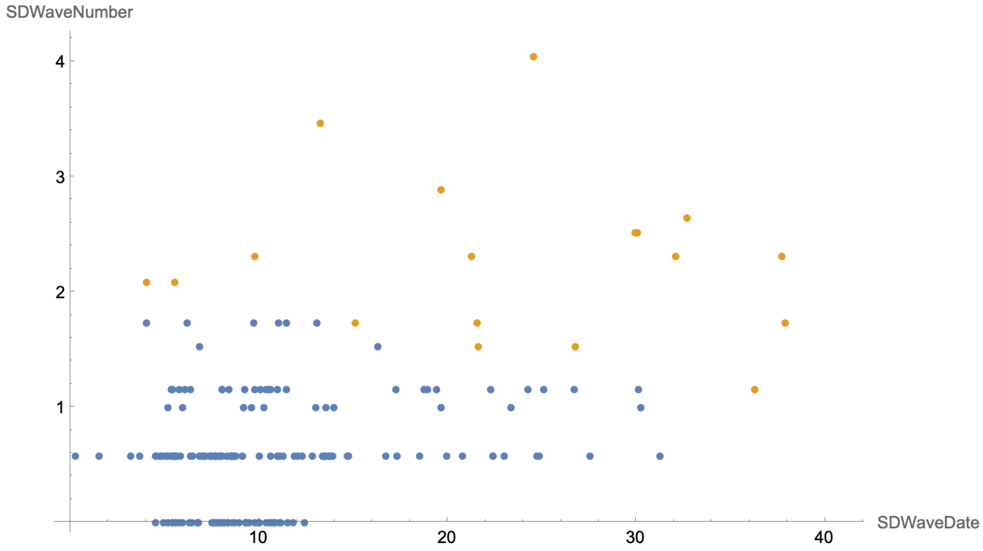
Out[•]=



In[544]:=

```
countriesGroupsPlot2m12var =
 ListPlot[FindClusters[resultsCluster2var // Values, 2, Method → "KMeans"],
  AxesLabel → {"SDWaveDate", "SDWaveNumber"}, ImageSize → Large]
countriesGroups2m12var = FindClusters[resultsCluster2var, 2, Method → "KMeans"];
```

Out[∘]=



In[∘]:=

```
Manipulate[
 {
  results[country],
  Show[countriesGroupsPlot2m12var,
   ListPlot[{country} /. Normal[resultsCluster2var], PlotStyle → Black]],
  wavesDailyPlot[country],
  wavesWeeklyPlot[country],
  wavesMonthlyPlot[country]
 },
 {country, countriesGroups2m12var⟦2⟧}
]
```

■ Results λ : 1.5

In[546]:=

```
kernelsλ15 = allKernel[All, {"Country", "KD1.5", "KW1.5", "KM1.5"}];
```

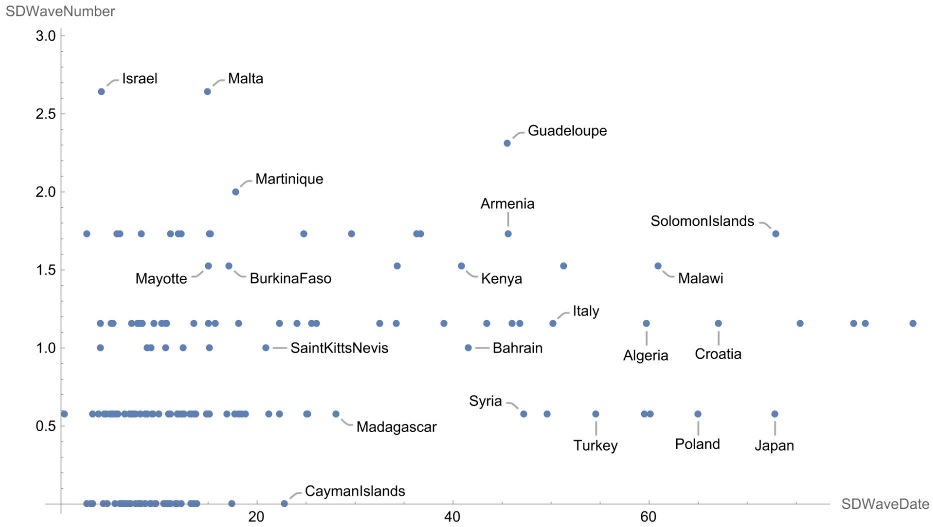
```
In[547]:=
resultsλ15 =
  Association[wavesByKByCountryGranularity2[#, wavesDaily15, wavesWeekly15,
      wavesMonthly15, kernelsλ15] & /@ countriesSFinal];
```

```
••• Part : Part 1 of {} does not exist.

••• Part : Part 1 of {} does not exist.

••• Part : Cannot take positions 2 through −1 in {}.

••• Part : Cannot take positions 1 through −2 in {}.

••• Part : Part 1 of {} does not exist.

••• General : Further output of Part::partw will be suppressed during this calculation.
```

```
In[548]:=
resultsλ15Cluster2var = DeleteCases[resultsλ15⟦All, {2, 4}⟧, {___, {}, ___}];
```

```
In[550]:=
ListPlot[resultsλ15Cluster2var, AxesLabel → {"SDWaveDate", "SDWaveNumber"}]
```

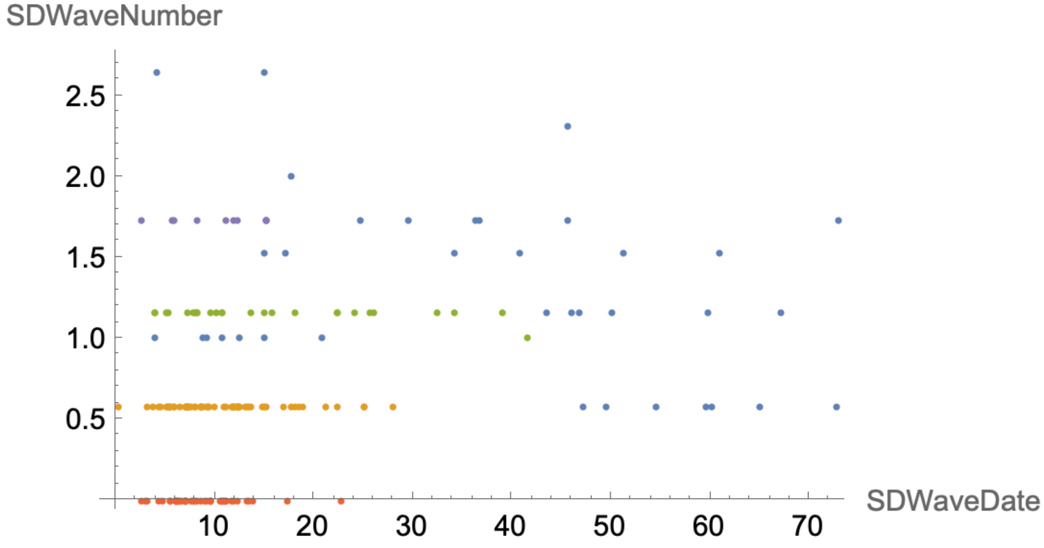
Out[ ]=



In[553]:=

```
countriesGroupsPlotm12varλ15 =
 ListPlot[FindClusters[resultsλ15Cluster2var // Values],
  AxesLabel → {"SDWaveDate", "SDWaveNumber"}]
countriesGroupsm12varλ15 = FindClusters[resultsλ15Cluster2var];
```

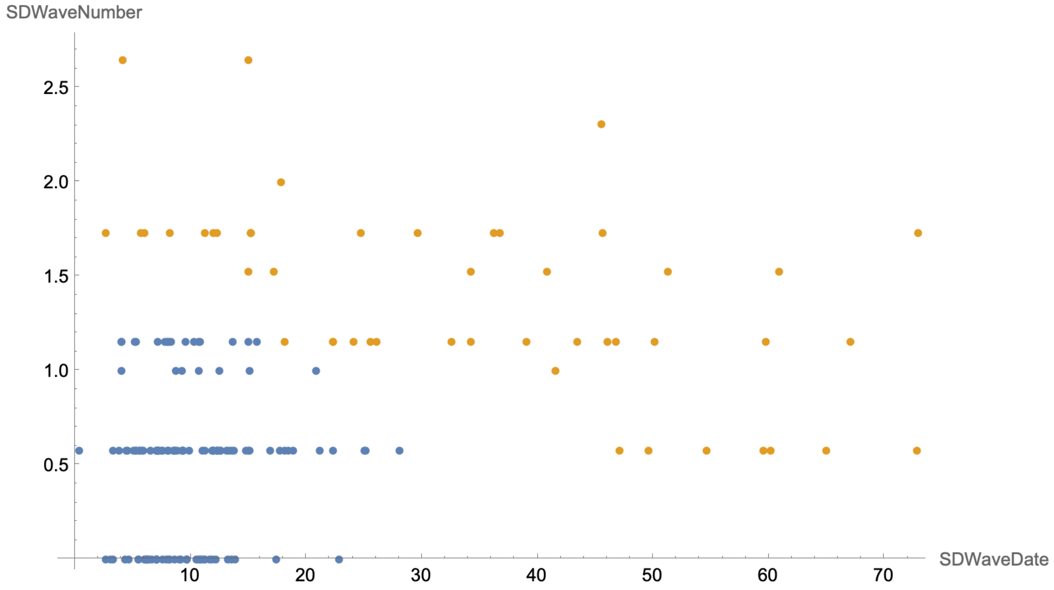
Out[∘]=



In[557]:=

```
countriesGroupsPlot2m12varλ15 =
 ListPlot[FindClusters[resultsλ15Cluster2var // Values, 2, Method → "KMeans"],
  AxesLabel → {"SDWaveDate", "SDWaveNumber"}, ImageSize → Large]
countriesGroups2m12varλ15 =
  FindClusters[resultsλ15Cluster2var, 2, Method → "KMeans"];
```

Out[∘]=



■ Results λ : 0.5

In[559]:=

```
kernelsλ05 = allKernel[All, {"Country", "KD0.5", "KW0.5", "KM0.5"}];
```

In[560]:=

```
resultsλ05 =
  Association[wavesByKByCountryGranularity2[#, wavesDaily05, wavesWeekly05,
      wavesMonthly05, kernelsλ05] & /@ countriesSFinal];
```

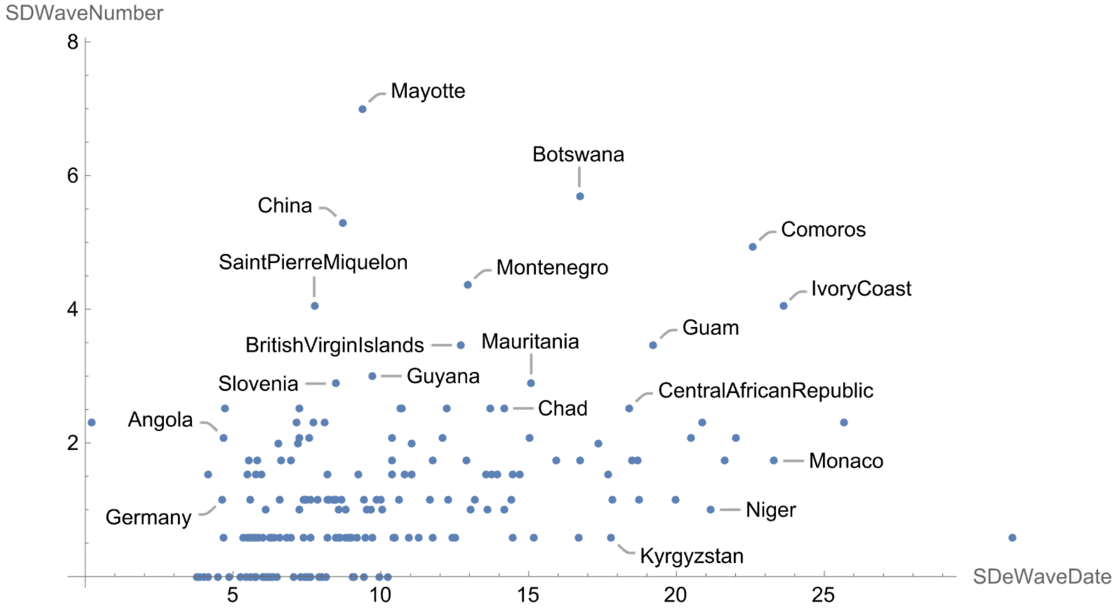
In[561]:=

```
resultsλ05Cluster2var =
  DeleteCases[resultsλ05⟦All, {2, 4}⟧, {Transpose[__], _} | {{}, 0.`}];
```

In[563]:=

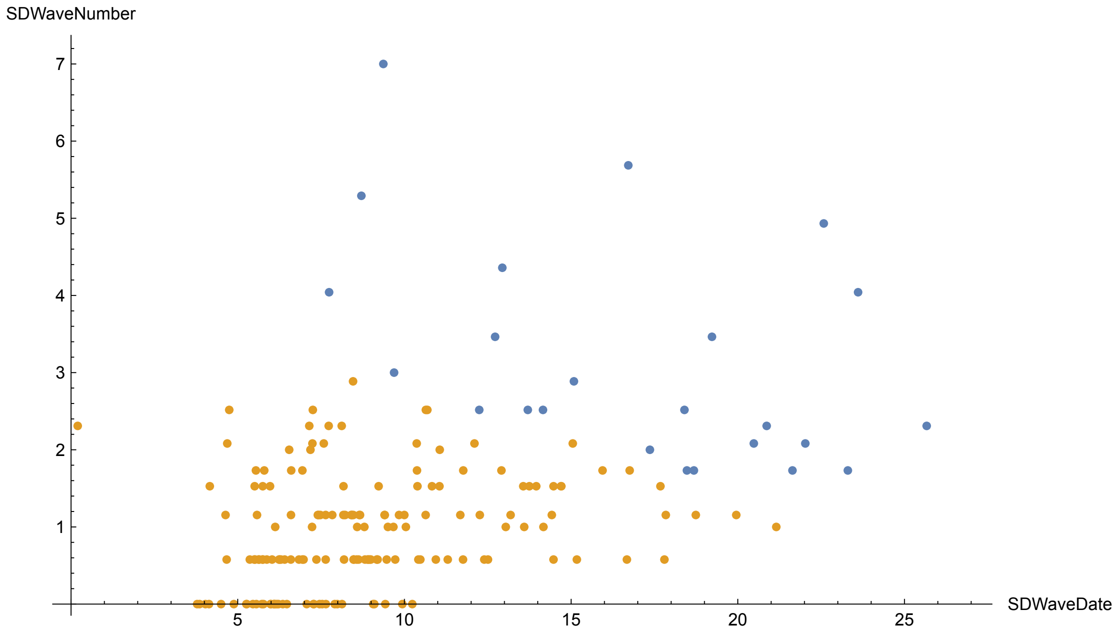
```
ListPlot[resultsλ05Cluster2var, AxesLabel → {"SDeWaveDate", "SDWaveNumber"}]
```

Out[ ]=



```
In[566]:=
countriesGroupsPlotm12varλ05 =
  ListPlot[FindClusters[resultsλ05Cluster2var // Values],
   AxesLabel → {"SDWaveDate", "SDWaveNumber"}, ImageSize → Large]
countriesGroupsm12varλ05 = FindClusters[resultsλ05Cluster2var];
```

Out[566]=



■ Summary

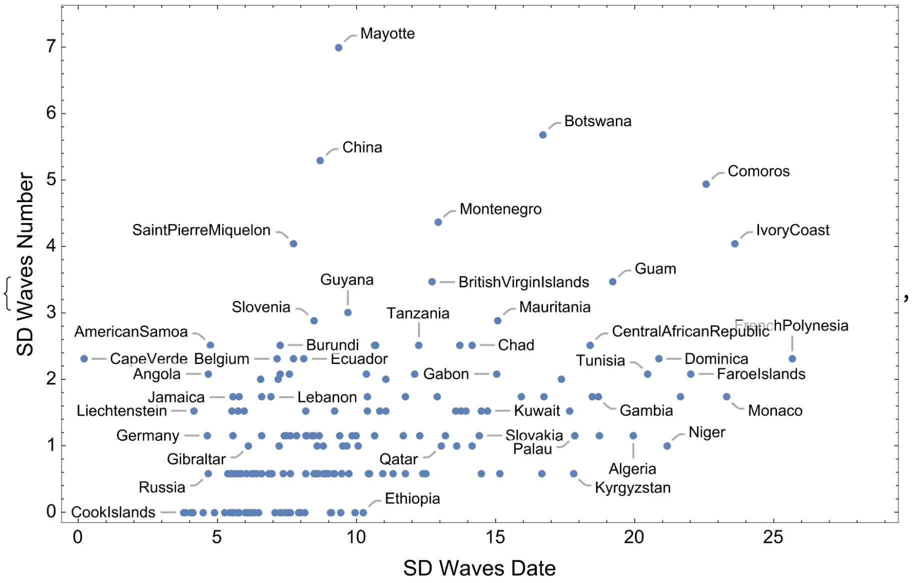
```
In[568]:=
kernelsV = Association@Normal[kernels[All, (#Country /. x_Entity ⧴ x⟦2⟧) →
       {"KD: " <> ToString[#KD], "KW: " <> ToString[#KW], "KM: " <> ToString[#KM],
        "VarK: " <> ToString[Variance[{#KD, #KW, #KM}]]} &]];

In[569]:=
kernelsVλ05 = Association@Normal[
    kernelsλ05[All, (#Country /. x_Entity ⧴ x⟦2⟧) → {"KD05: " <> ToString[#"KD0.5"],
        "KW05: " <> ToString[#"KW0.5"], "KM05: " <> ToString[#"KM0.5"],
        "VarK: " <> ToString[Variance[{#"KD0.5", #"KW0.5", #"KM0.5"}]]} &]];

In[570]:=
kernelsVλ15 = Association@Normal[
    kernelsλ15[All, (#Country /. x_Entity ⧴ x⟦2⟧) → {"KD15: " <> ToString[#"KD1.5"],
        "KW15: " <> ToString[#"KW1.5"], "KM15: " <> ToString[#"KM1.5"],
        "VarK: " <> ToString[Variance[{#"KD1.5", #"KW1.5", #"KM1.5"}]]} &]];

In[571]:=
opts = {FrameLabel →
    {Style["SD Waves Date", Black, 14], Style["SD Waves Number", Black, 14]},
   ImageSize → Large, Frame → True, FrameTicksStyle → Directive[Black, 12]};
```

In[572]:=

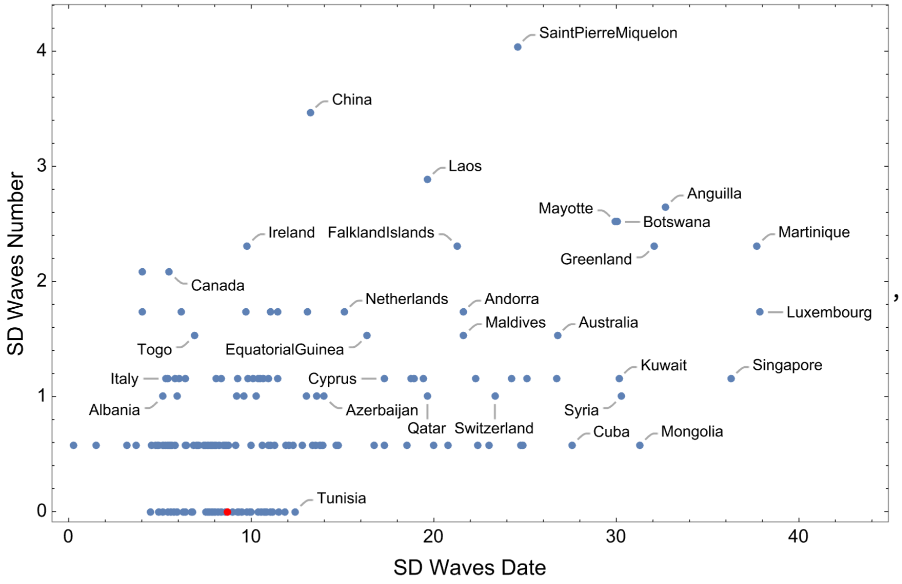
```
{
 ListPlot[resultsλ05Cluster2var, Sequence @@ opts],
 ListPlot[resultsCluster2var /. {8.710597298913347`, 0.`} →
    Style[{8.710597298913347`, 0.`}, Red], Sequence @@ opts],
 ListPlot[resultsλ15Cluster2var, Sequence @@ opts]
}
```

Out[ ]=

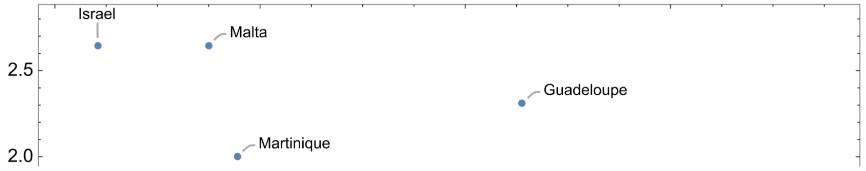
{

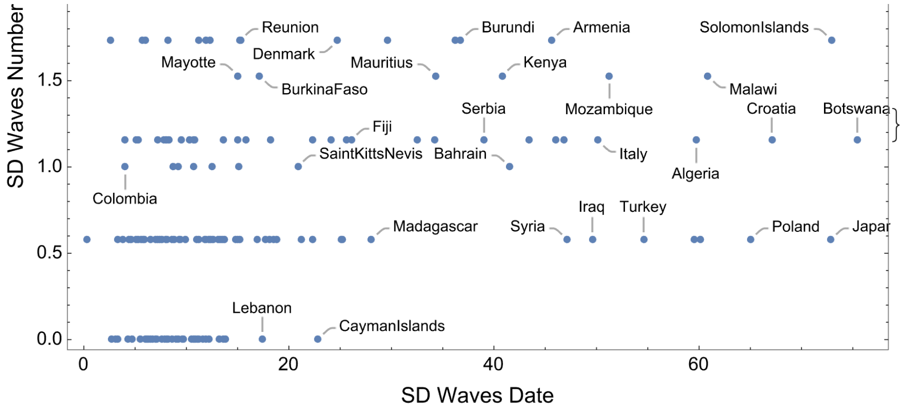


,



,





- what is the best lambda for each country

In[573]:=

```
theBestλ = # → SortBy[{{0.5, Reverse@resultsλ05Cluster2var[#]},
        {1, Reverse@resultsCluster2var[#]}, {1.5, Reverse@
          resultsλ15Cluster2var[#]}}, Last]⟦1⟧ & /@ Keys[resultsλ05Cluster2var];
```

In[574]:=

```
countryAndλ = Thread[theBestλ⟦All, 1⟧ → theBestλ⟦All, 2, 1⟧];
```

In[575]:=

```
resultAss = <|0.5 → resultsλ05Cluster2var,
   1 → resultsCluster2var, 1.5 → Reverse@resultsλ15Cluster2var|>;
```

In[576]:=

```
plotStyle = countryAndλ /. {0.5 → Yellow, 1 → Black, 1.5 → Red}
```

Out[576]=

```
{Afghanistan → 🟨, Albania → 🟥, Algeria → ⬛, AmericanSamoa → 🟨, Andorra → 🟨,
 Angola → ⬛, Anguilla → 🟨, AntiguaBarbuda → 🟨, Argentina → 🟥, Armenia → 🟨,
 Aruba → ⬛, Australia → 🟥, Austria → 🟨, Azerbaijan → 🟨, Bahamas → ⬛, Bahrain → ⬛,
 Bangladesh → 🟨, Barbados → 🟨, Belarus → ⬛, Belgium → ⬛, Belize → 🟥, Bermuda → ⬛,
 Bhutan → ⬛, Bolivia → 🟨, BosniaHerzegovina → ⬛, Botswana → 🟥, Brazil → 🟥,
 BritishVirginIslands → ⬛, Brunei → 🟨, Bulgaria → ⬛, BurkinaFaso → ⬛, Burundi → ⬛,
 Cambodia → ⬛, Cameroon → ⬛, Canada → 🟥, CapeVerde → ⬛, CaymanIslands → 🟥,
 CentralAfricanRepublic → 🟥, Chad → 🟥, Chile → 🟥, China → 🟥, Colombia → 🟨,
 Comoros → ⬛, CookIslands → 🟨, CostaRica → 🟨, Croatia → 🟨, Cuba → 🟨,
 Cyprus → 🟨, CzechRepublic → 🟥, DemocraticRepublicCongo → ⬛, Denmark → 🟨,
 Djibouti → 🟥, Dominica → ⬛, DominicanRepublic → 🟥, Ecuador → 🟥, Egypt → 🟥,
 ElSalvador → 🟥, EquatorialGuinea → 🟥, Eritrea → ⬛, Estonia → 🟥, Ethiopia → 🟨,
 FalklandIslands → 🟥, FaroeIslands → 🟥, Fiji → ⬛, Finland → ⬛, France → 🟨,
 FrenchGuiana → ⬛, FrenchPolynesia → ⬛, Gabon → ⬛, Gambia → ⬛, Georgia → 🟨,
 Germany → 🟥, Ghana → ⬛, Gibraltar → ⬛, Greece → 🟨, Greenland → 🟥, Grenada → 🟨,
 Guadeloupe → ⬛, Guam → ⬛, Guatemala → ⬛, Guinea → 🟨, GuineaBissau → 🟨,
 Guyana → ⬛, Haiti → 🟨, Honduras → 🟨, Hungary → ⬛, Iceland → 🟥, India → ⬛,
 Indonesia → ⬛, Iran → 🟥, Iraq → 🟨, Ireland → 🟥, Israel → ⬛, Italy → 🟨,
 IvoryCoast → 🟥, Jamaica → ⬛, Japan → 🟨, Kazakhstan → 🟥, Kenya → 🟨, Kiribati → ⬛,
 Kuwait → 🟥, Kyrgyzstan → 🟥, Laos → 🟨, Latvia → 🟥, Lebanon → 🟥, Lesotho → ⬛,
 Libya → ⬛, Liechtenstein → 🟥, Lithuania → 🟨, Luxembourg → 🟨, Macedonia → ⬛,
 Madagascar → ⬛, Malawi → 🟨, Malaysia → 🟨, Maldives → 🟥, Mali → 🟥, Malta → 🟨,
 MarshallIslands → 🟥, Martinique → 🟨, Mauritania → 🟥, Mauritius → ⬛,
 Mayotte → 🟥, Mexico → 🟨, Moldova → ⬛, Monaco → 🟥, Mongolia → 🟥, Montenegro → 🟥,
 Montserrat → ⬛, Morocco → 🟨, Mozambique → ⬛, Myanmar → 🟨, Namibia → ⬛,
 Nauru → 🟨, Nepal → 🟥, Netherlands → 🟥, NewCaledonia → 🟥, NewZealand → 🟥,
 Nicaragua → 🟥, Niger → 🟥, Nigeria → 🟨, NorthernMarianaIslands → ⬛,
 Norway → 🟥, Oman → 🟥, Pakistan → 🟨, Palau → 🟨, Panama → 🟨, PapuaNewGuinea → ⬛,
 Paraguay → ⬛, Peru → 🟥, Philippines → ⬛, Poland → ⬛, Portugal → ⬛,
 PuertoRico → 🟨, Qatar → 🟥, RepublicCongo → 🟥, Reunion → ⬛, Russia → 🟥,
 Rwanda → ⬛, SaintKittsNevis → 🟥, SaintLucia → 🟥, SaintPierreMiquelon → 🟥,
 SaudiArabia → 🟨, Senegal → ⬛, Serbia → 🟨, SierraLeone → ⬛, Singapore → 🟨,
 Slovakia → 🟥, Slovenia → 🟥, SolomonIslands → 🟨, Somalia → 🟥, SouthAfrica → ⬛,
 SouthKorea → 🟥, Spain → 🟥, SriLanka → ⬛, Sudan → 🟥, Suriname → ⬛,
 Swaziland → ⬛, Switzerland → 🟥, Syria → 🟨, Tajikistan → ⬛, Tanzania → ⬛,
 Thailand → 🟥, Togo → 🟨, Tonga → 🟨, TrinidadTobago → ⬛, Tunisia → ⬛, Turkey → 🟨}
```

In[577]:=

```
Tally[plotStyle⟦All, 2⟧]
```

Out[577]=

```
{{🟨, 56}, {🟥, 65}, {⬛, 66}}
```

In[578]:=

```
dataByCountry = Association@
   MapThread[#2 → resultAss[#1][#2] &, {theBestλ〚All, 2, 1〛, theBestλ〚All, 1〛}];
```

In[579]:=

```
dataByCountryStylized = MapThread[
   Callout[Style[#1, #2〚2〛], #2〚1〛] &, {dataByCountry // Values, plotStyle}];
```

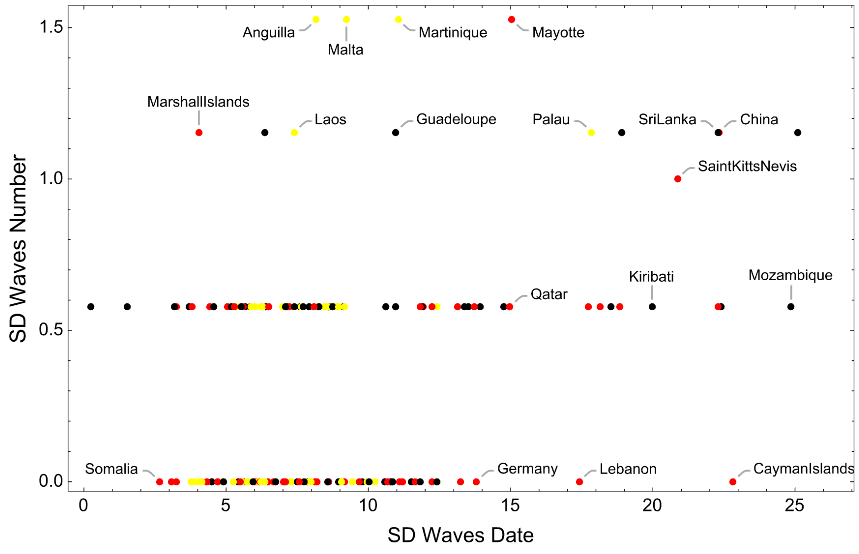
In[581]:=

```
opts = {FrameLabel →
     {Style["SD Waves Date", Black, 14], Style["SD Waves Number", Black, 14]},
   ImageSize → Large, Frame → True, FrameTicksStyle → Directive[Black, 12]};
```

In[582]:=

```
ListPlot[dataByCountryStylized, PlotStyle → plotStyle〚All, 2〛, FrameLabel →
  {Style["SD Waves Date", Black, 14], Style["SD Waves Number", Black, 15]},
 ImageSize → Large, Frame → True, FrameTicksStyle → Directive[Black, 12]]
```

Out[ ]=



Plots variations due to different kernels

SD Waves number

- Exploring

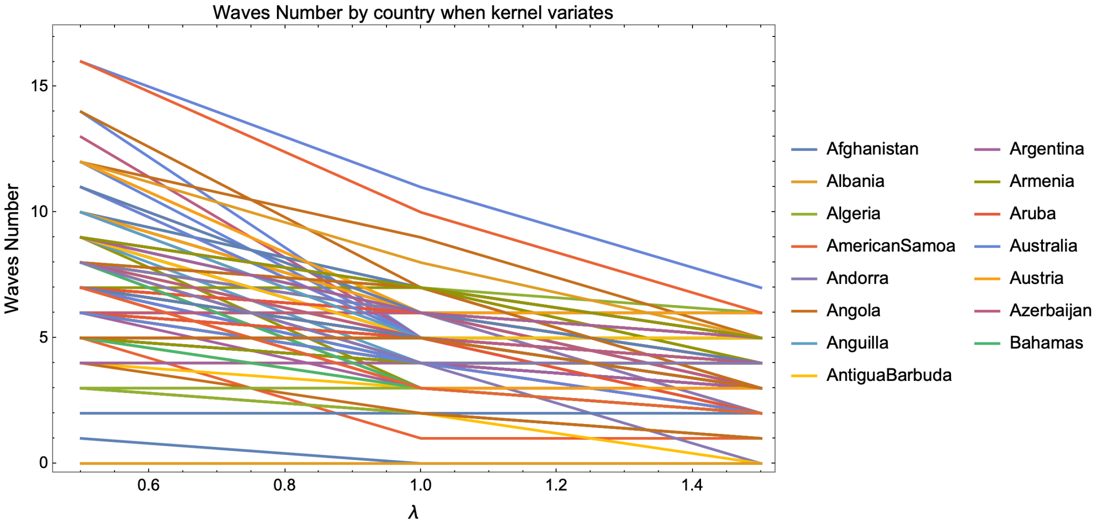
In[461]:=

```
wavesNumberλ = # → {{0.5, Count[wavesWeekly05[#], DateObject[__]]},
       {1, Count[wavesWeekly[#], DateObject[__]]},
       {1.5, Count[wavesWeekly15[#], DateObject[__]]}} & /@ countriesS〚 ;; 〛;
```

```
In[462]:=
opts = {FrameLabel → {Style["λ", Black, 14], Style["Waves Number", Black, 14]},
   ImageSize → Large, Frame → True, FrameTicksStyle → Directive[Black, 12],
   PlotLabel → Style["Waves Number by country when kernel variates", Black, 14]};
```

```
In[ ]:= Length@wavesNumberλ
Out[ ]=
201
```

```
In[ ]:= ListLinePlot[Association@wavesNumberλ⟦ ;; ⟧, Sequence @@ opts]
Out[ ]=
```

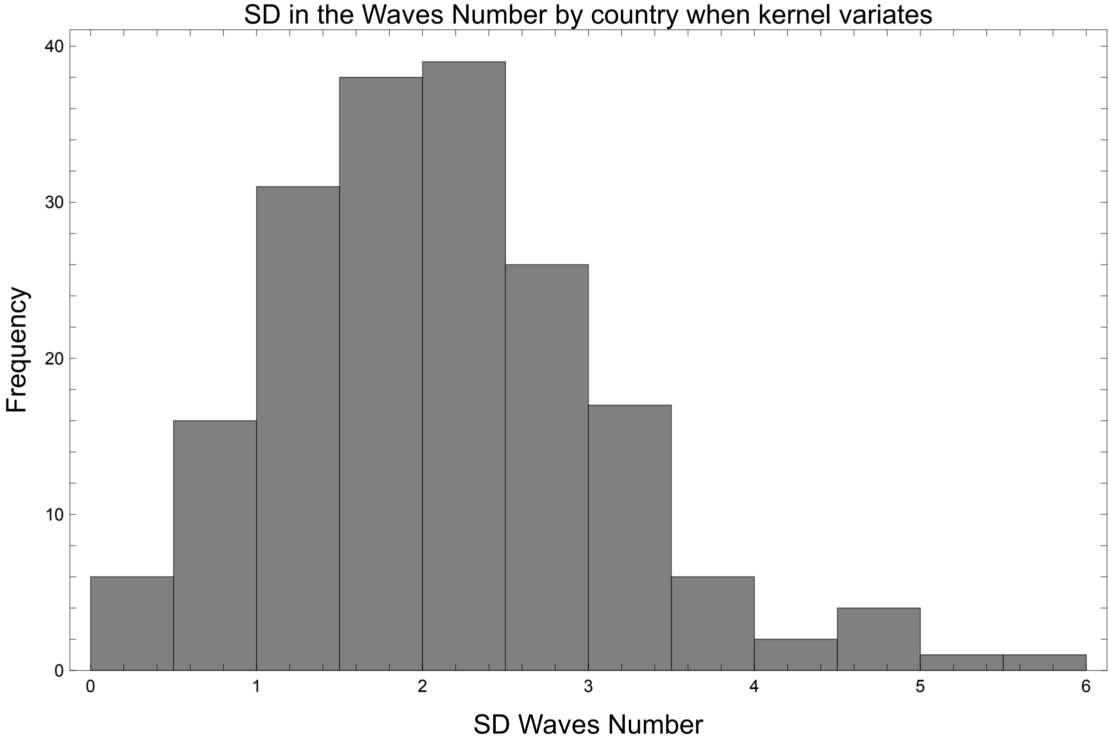


```
In[464]:=
wavesNumberλSD = # → StandardDeviation[{
       Count[wavesWeekly05[#], DateObject[__]],
       Count[wavesWeekly[#], DateObject[__]],
       Count[wavesWeekly15[#], DateObject[__]]
      }] & /@ countriesS⟦ ;; ⟧;
```

In[465]:=

```
opts = {
   Frame → True,
   FrameLabel → {Style["SD Waves Number", Black, 14],
     Style["Frequency", Black, 14]}, ImageSize → Large,
   PlotLabel → Style["SD in the Waves Number by country when kernel variates",
     Black, 14], ChartStyle → Gray};
Histogram[wavesNumberλSD⟦All, 2⟧, 10, Sequence @@ opts]
```

Out[466]=



In[468]:=

```
{N@Mean[#], N@StandardDeviation[#]} &@wavesNumberλSD⟦All, 2⟧
```

Out[468]=

```
{1.94958, 1.02715}
```

SD Waves date

- firstDateByCountry

In[469]:=

```
covid19ByCountryToCSVW2 = SemanticImport["covid19ByCountryToCSVW2.csv"];
```

In[470]:=

```
firstDateByCountryW = Quiet@KeyMap[#⟦2⟧ &,
    Normal[covid19ByCountryToCSVW2[GroupBy[#Country &]][All, 1][All, #Date &]]];
```

- Results

In[510]:=

```
resultsVariationλ = Function[country,
    wavesByCountryRaw = Function[{waves},
       resta = waves⟦1⟧[country] - firstDateByCountryD[country];
       Transpose[
        {Range[1, Length[resta]], Table[waves⟦2⟧, Length[resta]], resta}]] /@
      {{wavesWeekly05, 0.5}, {wavesWeekly, 1}, {wavesWeekly15, 1.5}};
    country → meanOfWavesDateSDFun[wavesByCountryRaw]] /@ countriesS⟦ ;; ⟧;
```

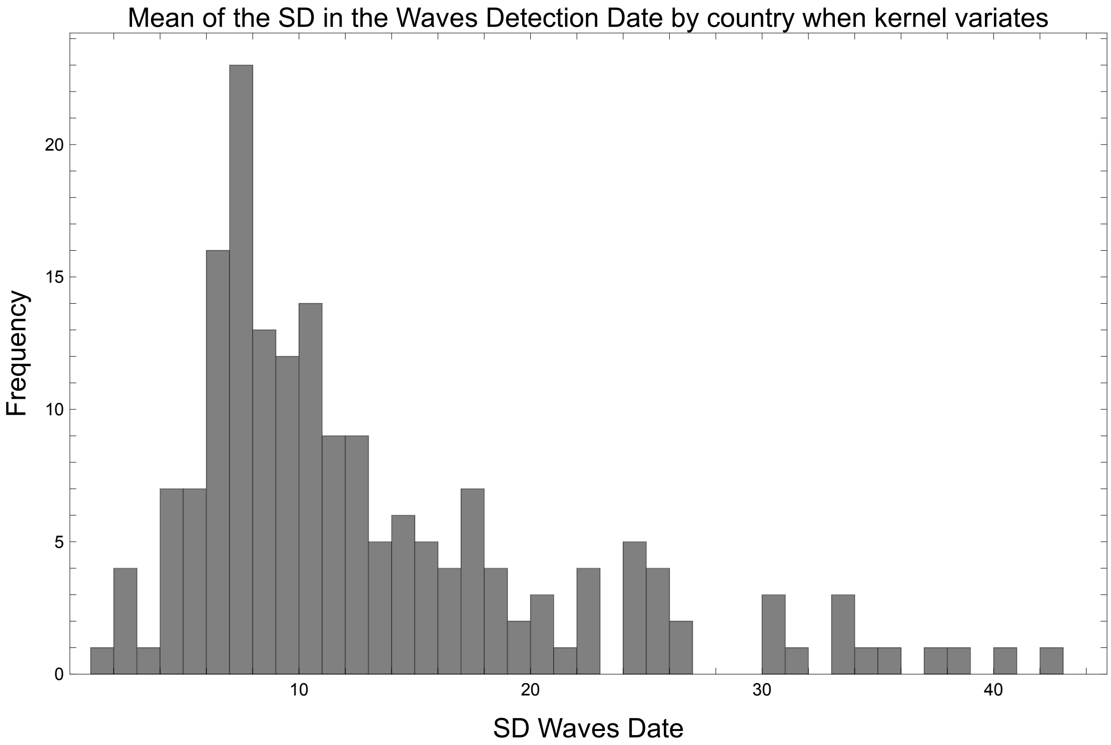
In[511]:=

```
MeanSDOfWavesDateλ = Select[resultsVariationλ, MatchQ[#⟦2⟧, _Quantity] &];
```

In[514]:=

```
opts = {
   Frame → True,
   FrameLabel → {Style["SD Waves Date", Black, 14], Style["Frequency", Black, 14]},
   ImageSize → Large,
   PlotLabel → Style["Mean of the SD in the Waves Detection Date by
       country when kernel variates", Black, 14], ChartStyle → Gray};
Histogram[MeanSDOfWavesDateλ⟦All, 2⟧, 100, Sequence @@ opts]
```

Out[515]=



In[516]:=

```
{N@Mean[#], N@StandardDeviation[#]} &@MeanSDOfWavesDateλ⟦All, 2⟧
```

Out[516]=

```
{15.0844 days , 15.6934 days}
```

Conclusions

- It is important to note that the kernel changes every time the reporting frequency change:

```
(*    KD: Kernel for daily frequency and λ : 1
      KW: Kernel for weekly frequency and λ : 1
      KM: Kernel for monthly frequency and λ : 1
      KD0.5: Kernel for daily frequency and λ : 0.5
      KW0.5: Kernel for weekly frequency and λ : 0.5
      KM0.5: Kernel for monthly frequency and λ : 0.5
      KD1.5: Kernel for daily frequency and λ : 1.5
      KW1.5: Kernel for weekly frequency and λ : 1.5
      KM1.5: Kernel for monthly frequency and λ : 1.5
*)
```

```
In[∘]:= allKernel = SemanticImport["allKernels.csv"]
Out[∘]=
```

Country	KD	KW	KM	KD0.5	KW0.5	KM0.
Afghanistan	35.6113	35.5604	42.9251	17.8056	17.7802	21.46
Albania	35.1548	37.0304	52.3067	17.5774	18.5152	26.15
Algeria	31.3452	34.55	54.3196	15.6726	17.275	27.15
American Samoa	23.5167	26.2611	44.8678	11.7584	13.1305	22.43
Andorra	29.934	31.6932	35.5762	14.967	15.8466	17.78
Angola	22.5871	25.567	36.0506	11.2935	12.7835	18.02
Anguilla	33.8655	38.6342	56.166	16.9327	19.3171	28.08
Antigua and Barbuda	38.8358	37.355	47.9265	19.4179	18.6775	23.96
Argentina	28.5147	29.21	35.0293	14.2574	14.605	17.51
Armenia	33.001	36.2901	49.7647	16.5005	18.145	24.88
Aruba	22.8663	25.6988	33.5059	11.4331	12.8494	16.75
Australia	40.406	46.7278	56.0518	20.203	23.3639	28.02
Austria	36.7972	37.8751	42.8198	18.3986	18.9376	21.40
Azerbaijan	34.8628	35.8169	47.5717	17.4314	17.9085	23.78
Bahamas	24.2461	26.6267	36.6397	12.1231	13.3133	18.31
Bahrain	28.3409	30.6391	36.4655	14.1704	15.3196	18.23
Bangladesh	39.7792	42.5081	50.1234	19.8896	21.2541	25.06
Barbados	48.3264	47.9056	59.9082	24.1632	23.9528	29.95
Belarus	29.3239	29.2166	39.2582	14.6619	14.6083	19.62
Belgium	29.0078	28.5788	42.6232	14.5039	14.2894	21.31

rows 1–20 of **218**