Brian — PS 7 — 2025-02-11 — Solution

EIWL3 Sections 18 and 19

Exercises from *EIWL3* Section 18

```
In[•]:= options = GeoServer → {Automatic, "GlobalTimeout" → 180, "ConnectionRetryCount" → 6};

In[•]:= (* 18.1 *) GeoDistance[New York City CITY, London CITY]

Out[•]=
        3453.71 mi

In[•]:= (* 18.2 *) GeoDistance[New York City CITY, London CITY] /
         GeoDistance[New York City CITY, San Francisco CITY]

Out[•]=
        1.35109

In[•]:= (* 18.3 *) UnitConvert[GeoDistance[Sydney CITY, Moscow CITY], km]

Out[•]=
        14 387. km
```

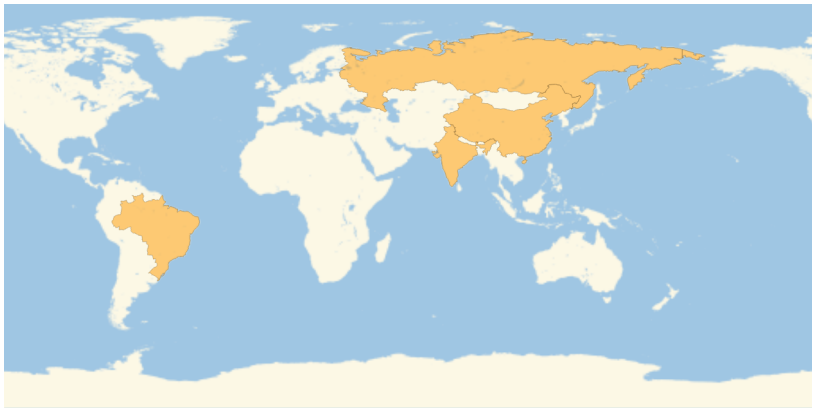
In[ ]:=
```
(* 18.4 *) GeoGraphics[Luxembourg COUNTRY, options]
```

Out[ ]=

I had repeated timeouts downloading the whole United States, so I downloaded a little country instead.

In[ ]:=
```
(* 18.5 *) GeoListPlot[{Brazil COUNTRY, Russia COUNTRY, India COUNTRY, China COUNTRY}]
```

Out[ ]=



In[ ]:=
```
(* 18.6 *) GeoGraphics[GeoPath[{New York City CITY, Boston CITY}], options]
```

Out[ ]=

Same problem with timeouts as above, so I chose a shorter route.

In[ ]:=
```
(* 18.7 *) GeoGraphics[GeoDisk[Great Pyramid of Giza HISTORIC SITE, 10 mi]]
```

Out[ ]=

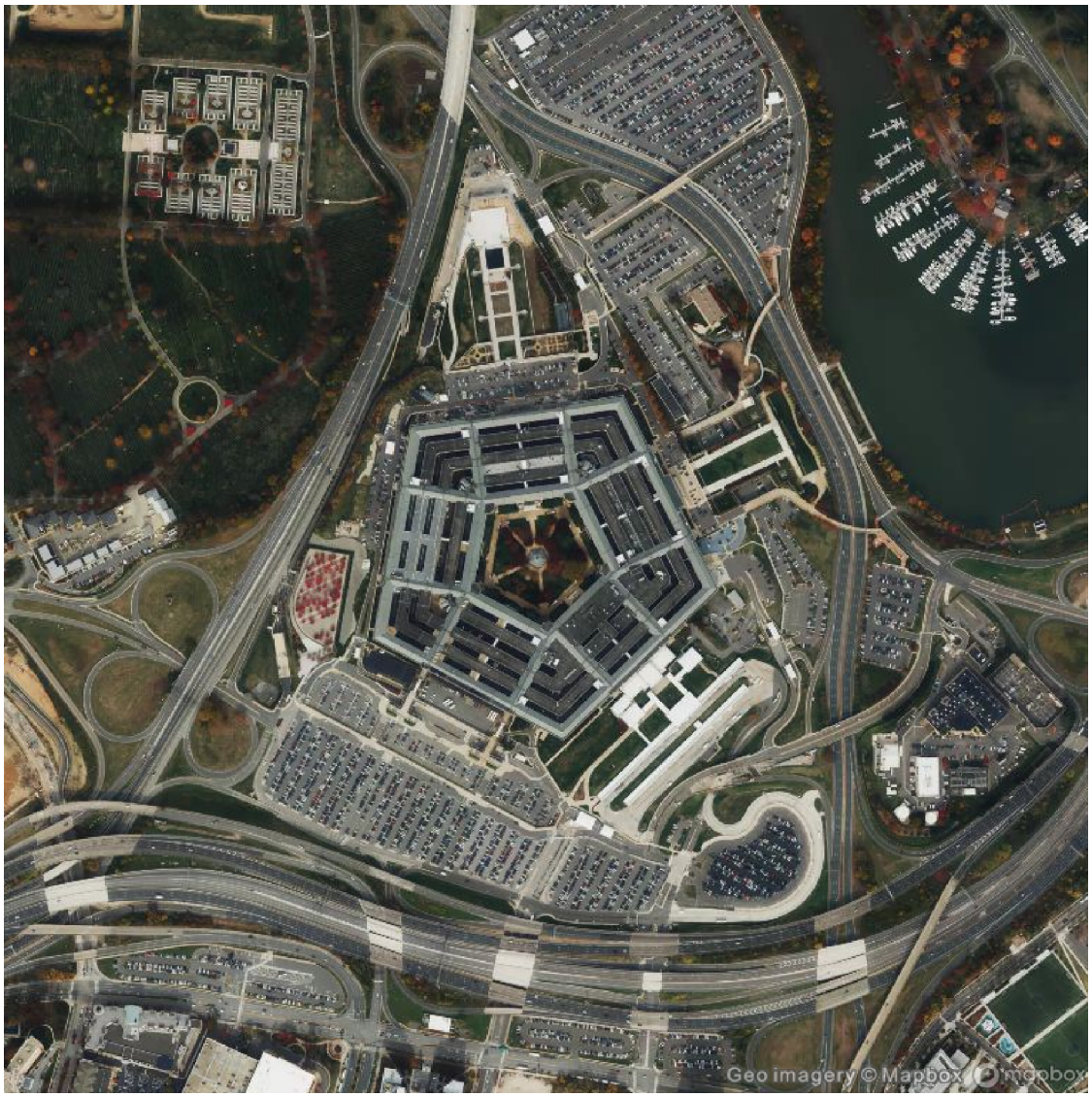
```
(* 18.8 *) GeoGraphics[
 GeoDisk[ New York City CITY , GeoDistance[ New York City CITY , Amherst CITY ]], options]
```

Out[ ]=

Yet more timeout issues, so I chose Amherst, MA as the second city.

In[•]:=
```
(* 18.9 *) GeoImage[GeoDisk[ The Pentagon BUILDING , 0.4 mi ]]
```

Out[•]=



In[•]:=
```
(* 18.10 *) GeoNearest["Country", GeoPosition["NorthPole"], 5]
```

Out[•]=

{ Greenland , Canada , Russia , Svalbard , United States }

```
(* 18.11 *) EntityValue[GeoNearest["Country", GeoPosition[{45, 0}], 3], "Flag"]
```

Out[•]=

```
(* 18.12 *) GeoListPlot[GeoNearest["Volcano", Rome CITY, 25]]
```

Out[•]=

```
(* 18.13 *)
EntityValue[New York City CITY, "Latitude"] - EntityValue[Los Angeles CITY, "Latitude"]
```

Out[•]=

6.64488°

Exercises from *EIWL3* Section 19

```
(* 19.1 *) Today - Mon 1 Jan 1900
```

Out[•]=

45 697 days

```
(* 19.2 *) DayName[Sat 1 Jan 2000]
```

Out[•]=

Saturday

```
(* 19.3 *) Today - 100 000 days
```

Out[•]=

Thu 29 Apr 1751

In[•]:=
```
(* 19.4 *) LocalTime[ Delhi CITY ]
```

Out[•]=

Tue 11 Feb 2025 19:47:14 GMT+5.5

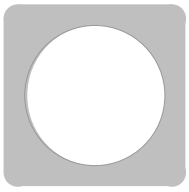
```
(* 19.5 *) Sunset[ Bishop CITY , Today] - Sunrise[ Bishop CITY , Today]
```

Out[•]=

10.7219 h

In[•]:=
```
(* 19.6 *) MoonPhase[Today, "Icon"]
```

Out[•]=



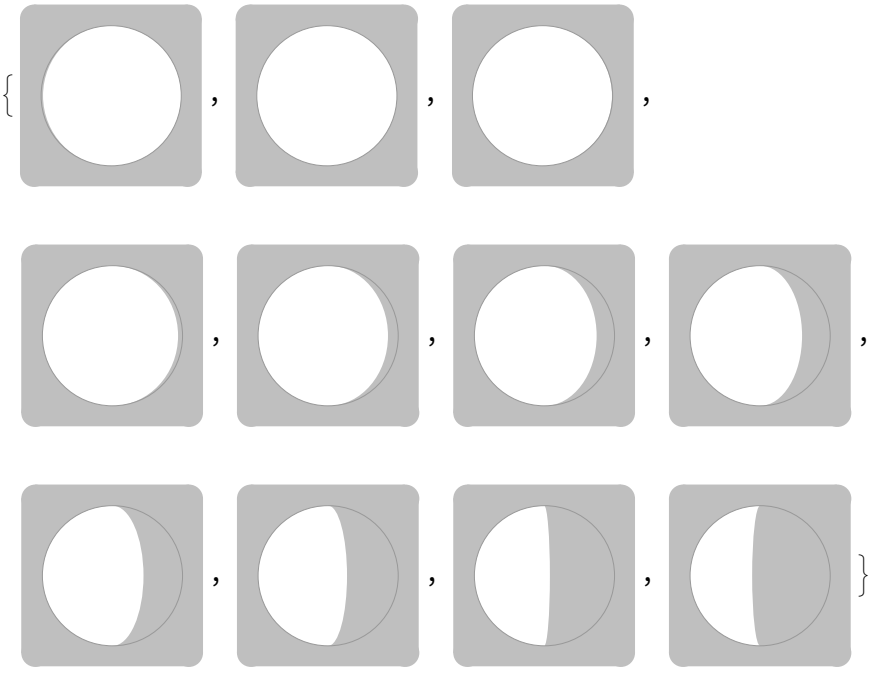
In[•]:=
```
(* 19.7 *) Table[MoonPhase[Today + i days ], {i, 10}]
```

Out[•]=

```
{0.998647, 0.993907, 0.969673, 0.927894,
 0.870882, 0.801093, 0.720987, 0.632977, 0.539487, 0.443082}
```

```
(* 19.8 *) Table[MoonPhase[Today + i days , "Icon"], {i, 0, 10}]
```

Out[•]=



```
(* The next one gave an annoyingly wrong answer until I *)
(* forced it to do something better by changing the date *)
(* for London. Until I did that, it was *)
(* getting tomorrow's sunrise in London, *)
(* because tomorrow's sunrise in London occurs just *)
(* before midnight in California. *)
(* Ptooey. *)

(* 19.9 *)
Sunrise[ New York City CITY , Tue 11 Feb 2025 ] - Sunrise[ London CITY , Mon 10 Feb 2025 ]
```

Out[•]=

```
4.54772 h
```

```
(* 19.10 *)UnitConvert[Today - Apollo 11 MANNED SPACE MISSION [ lunar landing date ], yr ]
```

Out[•]=

$$\frac{278}{5}\ \text{yr}$$

In[•]:=

```
(* 19.11 *)
yesterdayNoonInFrance = LocalTime[ France COUNTRY , Mon 10 Feb 2025 12:00:00 CET ];
AirTemperatureData[ Eiffel Tower BUILDING , yesterdayNoonInFrance]
```

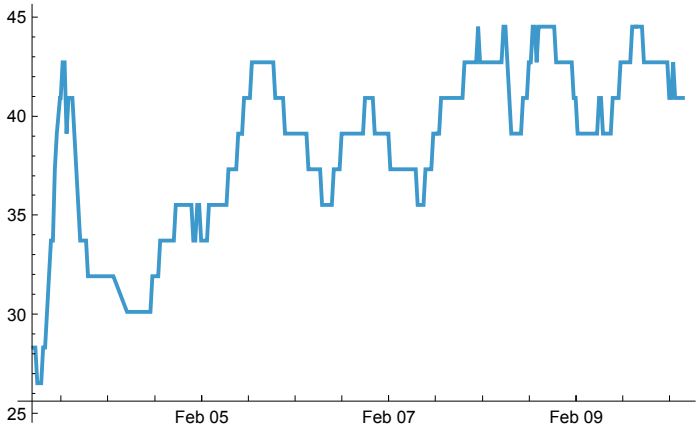
Out[•]=

```
41. °F
```

In[•]:=

```
(* 19.12 *) ListLinePlot[AirTemperatureData[ Eiffel Tower BUILDING ,
  {yesterdayNoonInFrance - 7 days , yesterdayNoonInFrance}]]
```

Out[•]=

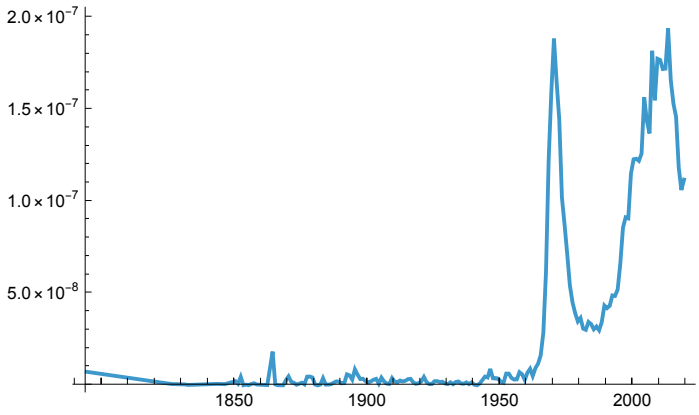


In[ ]:=
```
(* 19.13 *) AirTemperatureData[Los Angeles CITY, Now] -
 AirTemperatureData[New York City CITY, Now]
```

Out[ ]=
```
20. ° △ F
```

```
(* 19.14 *) ListLinePlot[WordFrequencyData["groovy", "TimeSeries"]]
```

Out[ ]=



In[ ]:=
```
(* 19.15 *) United States COUNTRY [Dated["Population", 2000]] -
 United States COUNTRY [Dated["Population", 1900]]
```

Out[ ]=
```
2.04604 × 10^8 people
```

I guess I didn't read carefully. We were supposed to do UK, not US.