Geo Heal

ASGR

2022-06-29

```
library(arrow)
```

```
##
## Attaching package: 'arrow'

## The following object is masked from 'package:utils':
##
##     timestamp
```

```
library(magicaxis)
library(data.table)
```

Overview

This document mostly describes the various inputs that went into the geo_heal_10.parquet data. This was put together by Aaron Robotham, and for any usage of this data beyond the course you should request permission.

The base grid for this data is a Heal Pix projection with Nside=10. I used Heal Pix since it is equal area per cell, has efficient properties in terms of matching positions to cells and locally still tends to be quite grid like in nature.

This Nside=10 grid has the following attributes:

```
sqdeg_sky = 4*pi*(180/pi)^2
area_earth = 510.1e6 #km^2

pixcount = function(k=10){12*(2^k)^2} # Pixels in specified Heal Pix system
cellsqdeg = function(k=10){sqdeg_sky/pixcount(k)} #Sky area per cell in sq. deg.
celldeg = function(k=10){sqrt(sqdeg_sky/pixcount(k))} #Sky angular size per cell in deg.
cellarea = function(k=10){area_earth/pixcount(k)} #Earth land area per cell in sq. km
cellsize = function(k=10){sqrt(area_earth/pixcount(k))} #Earth land size per cell in km
cellsqdeg()
```

```
## [1] 0.003278491
```

```
celldeg()
```

```
## [1] 0.05725811
```

```
cellarea()
```

```
## [1] 40.53911
```

```
cellsize()
```

```
## [1] 6.367033
```

cellsize is effectively the resolution of the grid. The value of 6km is well matched to the resolution of various public datasets I managed to combine, i.e. going higher than this would mean over-resolving the base data (so a waste).

Format

The geo_heal_10.parquet file is (fairly obviously) a Parquet file format file. In **R** we can load this with the *read_parquet* function in the **arrow** package (in Python you can use a similar function in the **pyarrow** package.

```r
geo_heal_10 = as.data.table(read_parquet('geo_heal_10.parquet'))
dim(geo_heal_10)
```

```
## [1] 12582912        7
```

```r
geo_heal_10[1:10,]
```

```
##         long         lat country  land  elev pop Tmax
##  1: 45.00000 -0.03730194    <NA> FALSE -3600   0   NA
##  2: 45.04395 -0.07460390    <NA> FALSE -3617   0   NA
##  3: 44.95605 -0.07460390    <NA> FALSE -3574   0   NA
##  4: 45.00000 -0.11190589    <NA> FALSE -3636   0   NA
##  5: 45.08789 -0.11190589    <NA> FALSE -3684   0   NA
##  6: 45.13184 -0.14920793    <NA> FALSE -3707   0   NA
##  7: 45.04395 -0.14920793    <NA> FALSE -3697   0   NA
##  8: 45.08789 -0.18651003    <NA> FALSE -3682   0   NA
##  9: 44.91211 -0.11190589    <NA> FALSE -3543   0   NA
## 10: 44.95605 -0.14920793    <NA> FALSE -3709   0   NA
```

The columns are largely obvious, but here we go:

long: The Earth Longitude (deg, from -180 to 180) of the Heal Pix cell centre. This is the position when assigning a position to a cell (since in Heal Pix the closest position on the sphere will be the correct cell).

lat: The Earth Latitude (deg, from -90 to 90) of the Heal Pix cell centre. This is the position when assigning a position to a cell (since in Heal Pix the closest position on the sphere will be the correct cell).

country: The best matching country for each Heal Pix cell (character). If no country is assigned then left as NA (e.g. oceans). This uses the *map.where* function and the base "world" map taken from the Natural Earth project in 2013 (1:50m resolution, see https://www.naturalearthdata.com/downloads/). Some odd errors have been fixed (such as latitudes above 85 and below -85 not being assigned a country). Also individual unassigned cells that were found to have populations above 1,000 from the towns/cities database were assigned the respective countries for said town/city. These are mostly islands that sit below the nominal 6 km resolution of the grid, or are sometimes even missing from the "world" map used (given its resolution).

land: A flag to say a given Heal Pix cell contains significant land. This is largely TRUE where-ever the *country* is not NA, however large lakes have been set to land=FALSE (e.g. the North American Great Lakes are visible).

elev: Elevation data for each Heal Pix cell (m from sea-level). The data here is taken from AWS Terrain Tiles (https://registry.opendata.aws/terrain-tiles/). The base data is effectively 4-5 km resolution, so quite well matched to our Heal Pix cells. Note the sea is not simply 0, it has negative values corresponding to the sea floor level.

pop: Propagated population data taken from the opendatasoft Geonames data base of All Cities with a population above 1,000 (https://public.opendatasoft.com/explore/dataset/geonames-all-cities-with-a-population-1000/). This contain 139,765 towns and cities, and 3.5 billions people (i.e. it roughly represents half the world's population. There was no obvious way to get more data than this expect in very ad-hoc local ways (e.g. just for USA) so I decided to stick with this. The consequence is most population results are indicative and relative. For very urbanised countries this should be quite accurate (e.g. we find 85m million people in Germany), although some seem to end up too high (Australia registers 39m, so there must be some double counting). China and India are far too low (630m and 308m respectively). So, lots of caveats here basically! Nearly every cell with a non-zero population will have a country assigned, but there are 14 '?' which appear to be islands that are hard to assign.

Tmax: Mean daily maximum temperature data in 2015 for each Heal Pix cell (C) taken created by Abatzoglou, J.T, 2017 (https://data.nkn.uidaho.edu/dataset/monthly-climate-and-climatic-water-balance-global-terrestrial-surfaces-1958-2015). The original data provided values per month, and these were trivially avergaed to get a yearly average daily maximum temperature. The base data is effectively 4 km resolution, so quite well matched to our Heal Pix cells.