

Pensieve header: Program to produce population density maps of planet Earth.

See <http://katlas.math.toronto.edu/drorbn/AcademicPensieve/Projects/FairMapping/>, and within that directory, especially the files

<http://katlas.math.toronto.edu/drorbn/AcademicPensieve/Projects/FairMapping/PoliticalMap.png>
<http://katlas.math.toronto.edu/drorbn/AcademicPensieve/Projects/FairMapping/PopulationDensityMap.png>
<http://katlas.math.toronto.edu/drorbn/AcademicPensieve/Projects/FairMapping/PopulationDensitySummary.m>

The last file is a list of triples of the form {n, d, c}, where n is the colour used for a country/region (0=white, 255=black), d is its population density as a float number between 0.026 for Greenland and 19432.6 for Macau, and c is the country/region name. The country with colour code 89 is "Water" and it is populated as the average land area population density of the world.

```
In[1]:= SetDirectory["C:\\drorbn\\AcademicPensieve\\Projects\\FairMapping"];
In[2]:= size = 2400;
SetOptions[Rasterize, {RasterSize -> size, ImageSize -> size}];
SetOptions[Graphics, PlotRange -> {{-Pi, Pi}, {-2, 2}}, ImageSize -> size];
SetAttributes[MakeImage, HoldRest];
MakeImage[name_String, g_] := Timing[Module[{s},
  Export[name <> ".png", s = Rasterize[img[name] = g]];
  ImageResize[s, 512]
]];
In[7]:= AllCountries = Union[
  CountryData["Countries"],
  {"FormerNetherlandsAntilles"}
];
Length[AllCountries]
Out[8]:= 240
In[9]:= MakeColourScheme[ac_List, newborders_List] := Module[
  {c, nbd, cnbd, addborders, k, oc, nc, changes, change},
  (nbd[#] = CountryData[#, "BorderingCountries"]) & /@ ac;
  addborders[{c1_, c2_}] := (
    nbd[c1] = Union[nbd[c1], {c2}];
    nbd[c2] = Union[nbd[c2], {c1}];
  );
  addborders[l_List] := Do[
    addborders[{l[[i]], l[[j]]}],
    {i, 2, Length[l]}, {j, i}
  ];
  addborders /@ newborders;
  (CountryColour[#] = RandomReal[]) & /@ AllCountries;
  changes = {}; change = 0;
  Do[
    Do[
      cnbd = Sort[CountryColour /@ nbd[c]];
      If[cnbd != {},
        AppendTo[cnbd, 1 + First[cnbd]];
      ]
    ]
  ]
];
```

```

    {k} = Ordering[Rest[RotateRight[cnbd] - cnbd], 1];
    oc = CountryColour[c];
    nc = FractionalPart[(cnbd[[k]] + cnbd[[k + 1]]) / 2];
    oc = Abs[oc - nc]; If[oc > 0.5, oc = 1 - oc];
    change += oc;
    CountryColour[c] = nc
  ],
  {c, AllCountries}
];
AppendTo[changes, change];
change = 0,
{32}
];
Print["Country colouring process changes: ", changes]
];
MakeColourScheme[ac_] := MakeColourScheme[ac, {}];
LoadColourScheme[s_String] := Module[
  {rule},
  rule = Get["ColourSchemes/" <> s <> ".m"];
  (CountryColour[#] = (# /. rule)) & /@ (First /@ rule);
];
SaveColourScheme[] := SaveColourScheme[
  DateString[{"YearShort", "Month", "Day", "-", "Hour24", "Minute", "Second"}]
];
SaveColourScheme[dt_String] := Module[
  {},
  Put[
    (# → CountryColour[#]) & /@ AllCountries,
    "ColourSchemes/" <> dt <> ".m"
  ];
  Export["ColourSchemes/" <> dt <> ".png",
  Graphics[{
    EdgeForm[White],
    {
      Hue[CountryColour[#]],
      CountryShape[#] /.
        {CountryShape → List, Region[pts___] ⇒ Polygon[{pts} /. LonLat → List]}
    } & /@
    (AllCountries)
  },
  Background → Black, ImageSize → 640
  ]
  ]
  ]
  ]

```

```

In[14]:= MakeColourScheme[AllCountries, {
  {"Canada", "Greenland"}, {"Canada", "Russia"},
  {"UnitedStates", "Russia"}, {"Canada", "Mexico"},
  {"Malaysia", "PapuaNewGuinea"}, {"Japan", "Russia"}, {"Japan", "SouthKorea"},
  {"Kazakhstan", "Mongolia"}, {"Australia", "NewZealand"},
  {"Australia", "Indonesia"}, {"Australia", "PapuaNewGuinea"},
  {"China", "Japan"}, {"China", "SouthKorea"}, {"NorthKorea", "Japan"},
  {"UnitedStates", "Greenland"}, {"Greenland", "Russia"},
  {"Pakistan", "Tajikistan"}, {"Canada", "Japan"}, {"Canada", "Italy"},
  {"Italy", "Japan"}, {"Italy", "Germany"}, {"France", "UnitedKingdom"}
}];

Country colouring process changes:
{30.9711, 8.03328, 2.98875, 1.55946, 1.45645, 1.25493, 0.863846, 0.573624, 0.426908,
 0.342159, 0.288242, 0.250271, 0.222947, 0.200688, 0.180401, 0.162428, 0.147007,
 0.133765, 0.12237, 0.112486, 0.104393, 0.0972188, 0.0905531, 0.0844003, 0.0787269,
 0.0734949, 0.0686679, 0.0642119, 0.0600948, 0.0562871, 0.0527617, 0.0494937}

In[15]:= Area[reg_Region] := Abs[Dot[
  Sin[List @@ Last /@ reg],
  ((RotateLeft[#] - #) &) @ (List @@ First /@ reg)
]];
Area[c_CountryShape] := (Plus @@ Area /@ c);
Area[s_String] := Area[CountryShape[s]];

A "CountryShape" is a sequence of "Region"s, each of which is a sequence of "Point"s:

In[18]:= RussianShift = 10.4`;
CountryShape[c_String] := Module[
  {shape},
  shape = Sort[
    Select[
      Map[LonLat[(#[[1]] - RussianShift) Degree, Sin#[[2] Degree]] &,
        Apply[Region,
          CountryShape @@ First[CountryData[c, "FullPolygon"]],
          {1}
        ],
      {2}
    ],
    Area[#] > 10^(-5) &
  ],
  Area[#] &
];
If[c != "Russia", shape,
  shape /. LonLat[ln_, lt_] /; ln < -Pi => LonLat[ln + 2 Pi, lt]
]
]
(* /. LonLat[ln_, lt_] => LonLat[-Pi/2+ln, lt] *)

```

```

In[20]:= MakeImage["PoliticalMap", Graphics[{
  EdgeForm[White],
  {
    Hue[CountryColour[#]],
    CountryShape[#] /. {
      CountryShape → List,
      Region[pts___] => Polygon[{pts} /. LonLat[ln_, lt_] => {ln, 2 lt}}]
    } & /@
    (AllCountries)
  },
  Background → LightBlue
]]

```

Out[20]= {11.809, 

```

In[21]:= PopulationDensity[c_String] := CountryData[c, "Population"] / CountryData[c, "Area"];
PopulationDensity["Water"] := CountryData["World", "Population"] /
  (CountryData["World", "LandArea"] - CountryData["Antarctica", "LandArea"]);
AllDensities = Sort[
  PopulationDensity /@ Append[AllCountries, "Water"]
];
PopulationDensityRank[c_String] :=
  Position[AllDensities, PopulationDensity[c]][[1, 1]] - 1;
PopulationDensityRank["Water"]

```

Out[25]= 89

```

In[26]:= MakeImage["PopulationDensityMap", Graphics[{
  EdgeForm[],
  {
    GrayLevel[1 - PopulationDensityRank[#] / 256],
    CountryShape[#] /. {
      CountryShape → List,
      Region[pts___] => Polygon[{pts} /. LonLat[ln_, lt_] => {ln, 2 lt}}]
    } & /@
    (AllCountries)
  },
  Background → GrayLevel[1 - PopulationDensityRank["Water"] / 256]
]]

```

Out[26]= {8.159, 

```
In[27]:= PopulationDensitySummary = Sort[
  {PopulationDensityRank[#], PopulationDensity[#], #} & /@
  Append[AllCountries, "Water"]
];
f = OpenWrite["PopulationDensitySummary.m"];
Write[f, #] & /@ PopulationDensitySummary;
Close[f]
```

```
Out[30]= PopulationDensitySummary.m
```