



To make Medium work, we log user data.
By using Medium, you agree to our
Privacy Policy, including cookie policy.

Open in app

Get started



Published in Coding Entropy



Hugo Larcher

[Follow](#)

Oct 20, 2015 · 4 min read · [Listen](#)

Save



Flight data visualisation with Pandas and Matplotlib

Learn how to quickly plot aircraft flight routes on a map for data visualisation.

I recently had to play with aircraft flight data to assess CO₂ emissions along flight routes. Given the data points I found myself a little puzzled when trying to find an adequate way to easily communicate the results. Data tables are boring and do not allow understand the connections between the different routes. I gave a try to a bubble chart overlay but I was only getting a bloated map. I then remembered a famous [Facebook visualisation](#) that presented all friends connections with “flight paths” and created a world map. Really nice, and allowed to represent a third dimension using





To make Medium work, we log user data.
By using Medium, you agree to our
Privacy Policy, including cookie policy.

[Open in app](#)

[Get started](#)



I can haz CSV? — (By Isa2886)

When it comes to data manipulation, Pandas is the library for the job. It allows easy manipulation of structured data with high performances. My dataset being quite small, I directly used Pandas' CSV reader to import it. I called the `read_csv()` function to import my dataset as a Pandas DataFrame object. I just needed to escape the first row which contained some headers and to define the delimiter (autodetection does not work here...):

```
1 import pandas as pd
2
3 # define the expected CSV columns
4 CSV_COLS = ('airline', 'dep_lat', 'dep_lon', 'arr_lat',
5             'arr_lon', 'nb_flights', 'route')
6
7 routes = pd.read_csv(in_filename, names=CSV_COLS, na_values=['\\N'],
8                      sep=';', skiprows=1)
```

[read_csv.py](#) hosted with ❤ by GitHub

[view raw](#)

Using GMap

A little googling led me to [GMap](#), a great tool ported to Python by [paulgb](#) and based on Facebook visualisations from 2010.

My dataset only contains departure and arrival airport. Luckily, GMap computes the flight routes using a [Great Circle Distance calculation](#). This is not the real aircraft trajectory but will be perfect for our use.





To make Medium work, we log user data.
By using Medium, you agree to our
Privacy Policy, including cookie policy.

[Open in app](#)

[Get started](#)

```
5
6 # define CSV column names
7 CSV_COLS = ('dep_lat', 'dep_lon', 'arr_lat', 'arr_lon', 'nb_flights', 'CO2')
8
9 routes = pd.read_csv('data.csv', names=CSV_COLS,
10                      na_values=['\\N'], sep=';', skiprows=1)
11
12 # create gradient to color the routes according to the number of flights
13 grad = Gradient(((0, 0, 0, 0), (0.5, 204, 0, 153), (1, 255, 204, 230)))
14 # initialize GCMapper and set data
15 gcm = GCMapper(cols=grad, height=2000, width=4000)
16 gcm.set_data(routes['dep_lon'], routes['dep_lat'], routes['arr_lon'],
17               routes['arr_lat'], routes['nb_flights'])
18 # run & save
19 img = gcm.draw()
20 img.save('flights_map_gcmap.png')
```

gcm.py hosted with ❤ by GitHub

[view raw](#)

The above script seemed to work but with my first dataset I could only see a few lines on a black background! The script does not plot country borders so it needs a dataset with enough world coverage. Trying with another dataset I got a much more decent result.



Worldwide flights during 24 hours in September. Color scale is based on the number of flights





To make Medium work, we log user data.
By using Medium, you agree to our
Privacy Policy, including cookie policy.

[Open in app](#)

[Get started](#)

According to documents

comparison between two coordinates pair. This is clearly not suitable for accurate data analysis.

which does not allow

Matplotlib to the rescue

Matplotlib is the default choice for data visualisation in Python. It comes with a handful basemap plotting toolkit which easily allows to add country boundaries. It also includes a useful function to compute the Great Circle path between two geographic points : neat!

The script is similar to GCmap: it estimates the flight path between departure and arrival airports using great circle distance and plots it with a colour depending on the number of flights. I also implemented a little hack that detects when a route intersects the edge of the map: matplotlib's default behaviour is to link the two opposite points, resulting in a straight line crossing the whole map.

I implemented two approaches to compute the route color. The first one is based on the absolute value of the number of flights. It is perfect for comparison but your distribution must not have an excessive standard deviation. Otherwise you'll only see the routes with the greater number of flights. In that case, use a power-law normalizer to compute the color (matplotlib PowerNorm). The result is great even with small datasets:

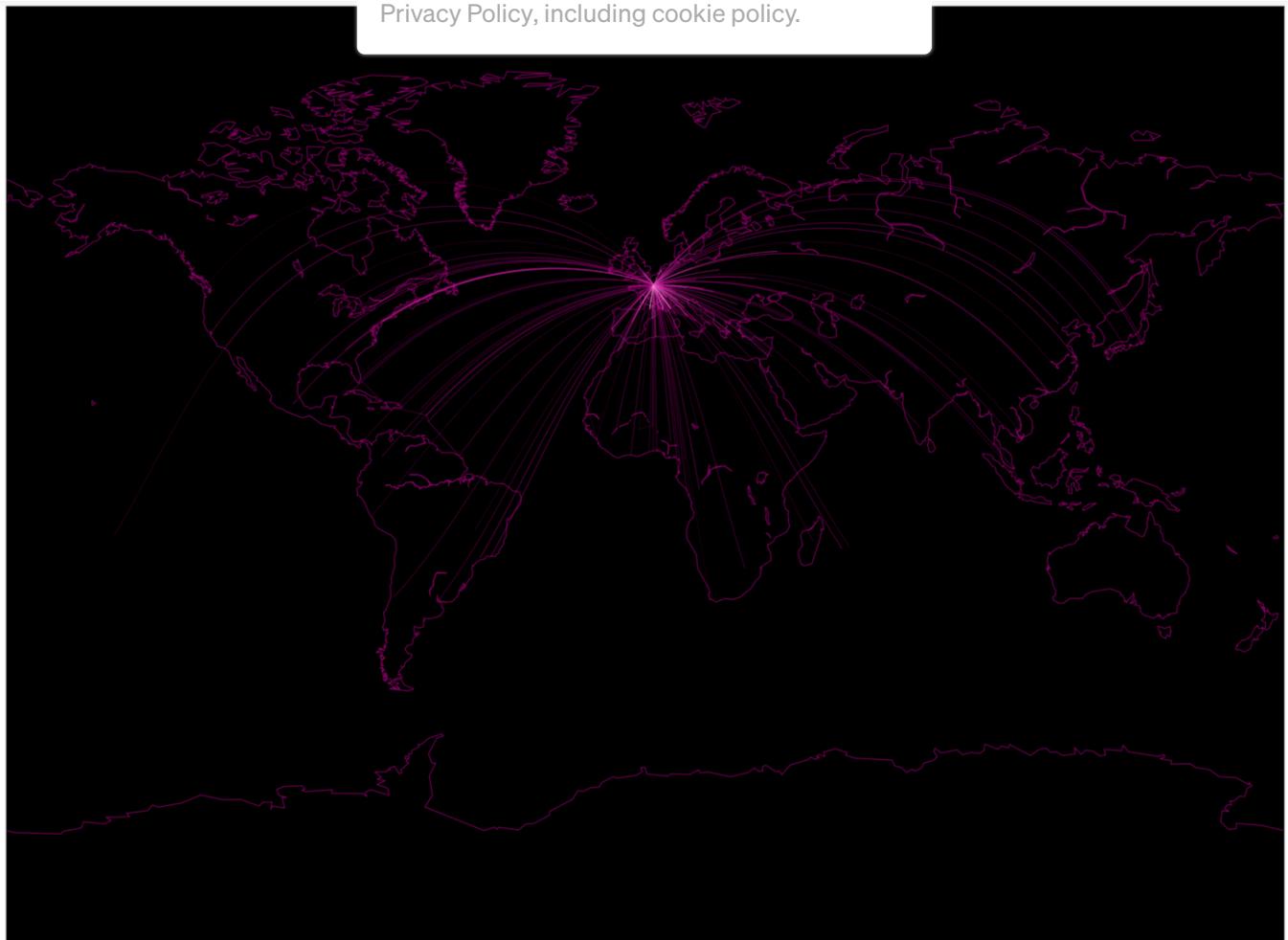




To make Medium work, we log user data.
By using Medium, you agree to our
Privacy Policy, including cookie policy.

[Open in app](#)

[Get started](#)



Visualisation of flights for Air France during 24 hours

The second approach sorts the routes by their number of flights and attributes one color to each route ranging for black (less flights) to pink (more flights). The result is quite similar to GCmap, with countries boundaries added:





To make Medium work, we log user data.
By using Medium, you agree to our
Privacy Policy, including cookie policy.

[Open in app](#)

[Get started](#)



Visualisation of the dataset with matplotlib using relative coloring

Take care, if you want to compare multiple visualisations you must use the same normalisation.

Here is the script I used:

```
1  #!/usr/bin/env python
2  # -*- coding: utf-8 -*-
3
4  import numpy as np
5  import pandas as pd
6  from mpl_toolkits.basemap import Basemap
7  import matplotlib.pyplot as plt
8  from matplotlib.colors import Normalize, LinearSegmentedColormap, PowerNorm
9
10
11 def plot_map(in_filename, color_mode='screen',
```





To make Medium work, we log user data.

By using Medium, you agree to our

[Open in app](#)

[Get started](#)

```
17     in_filename Privacy Policy, including cookie policy.
18     out_filename. output image filename
19     color_mode: Use 'screen' if you intend to use the visualisation for
20         on screen display. Use 'print' to save the visualisation
21         with printer-friendly colors.
22     absolute: set to True if you want coloring to depend on your dataset
23         parameter value (ie for comparison).
24             When set to false, each coordinate pair gets a different
25             color.
26
27     """
28
29     if color_mode == 'screen':
30         bg_color = (0.0, 0.0, 0, 1.0)
31         coast_color = (204/255.0, 0, 153/255.0, 0.7)
32         color_list = [(0.0, 0.0, 0.0, 0.0),
33                         (204/255.0, 0, 153/255.0, 0.6),
34                         (255/255.0, 204/255.0, 230/255.0, 1.0)]
35     else:
36         bg_color = (1.0, 1.0, 1.0, 1.0)
37         coast_color = (10.0/255.0, 10.0/255.0, 10/255.0, 0.8)
38         color_list = [(1.0, 1.0, 1.0, 0.0),
39                         (255/255.0, 204/255.0, 230/255.0, 1.0),
40                         (204/255.0, 0, 153/255.0, 0.6)
41                     ]
42
43     # define the expected CSV columns
44     CSV_COLS = ('dep_lat', 'dep_lon', 'arr_lat', 'arr_lon',
45                 'nb_flights', 'CO2')
46
47     routes = pd.read_csv(in_filename, names=CSV_COLS, na_values=['\\N'],
48                           sep=';', skiprows=1)
49
50     num_routes = len(routes.index)
51
52     # normalize the dataset for color scale
53     norm = PowerNorm(0.3, routes['nb_flights'].min(),
54                       routes['nb_flights'].max())
55     # norm = Normalize(routes['nb_flights'].min(), routes['nb_flights'].max())
56
57     # create a linear color scale with enough colors
58     if absolute:
59         n = routes['nb_flights'].max()
```





To make Medium work, we log user data.
By using Medium, you agree to our
Privacy Policy, including cookie policy.

Open in app

Get started

```
65 plt.figure(figsize=(10, 6))
66 m = Basemap(projection='mill', lon_0=0)
67 m.drawcoastlines(color=coast_color, linewidth=1.0)
68 m.fillcontinents(color=bg_color, lake_color=bg_color)
69 m.drawmapboundary(fill_color=bg_color)
70
71 # plot each route with its color depending on the number of flights
72 for i, route in enumerate(routes.sort_values(by='nb_flights',
73                                ascending=True).iterrows()):
74     route = route[1]
75     if absolute:
76         color = cmap(norm(int(route['nb_flights'])))
77     else:
78         color = cmap(i * 1.0 / num_routes)
79
80     line, = m.drawgreatcircle(route['dep_lon'], route['dep_lat'],
81                               route['arr_lon'], route['arr_lat'],
82                               linewidth=0.5, color=color)
83
84     # if the path wraps the image, basemap plots a nasty line connecting
85     # the points at the opposite border of the map.
86     # we thus detect path that are bigger than 30km and split them
87     # by adding a NaN
88     path = line.get_path()
89     cut_point, = np.where(np.abs(np.diff(path.vertices[:, 0])) > 30000e3)
90     if len(cut_point) > 0:
91         cut_point = cut_point[0]
92         vertices = np.concatenate([path.vertices[:cut_point, :],
93                                   [[np.nan, np.nan]],
94                                   path.vertices[cut_point+1:, :]])
95         path.codes = None # treat vertices as a serie of line segments
96         path.vertices = vertices
97
98     # save the map
99     plt.savefig(out_filename, format='png', bbox_inches='tight')
100
101 if __name__ == '__main__':
102     # use 'screen' color mode for on-screen display. Use 'print' if you intend
103     # to print the map
```

cartridge killer. Let's save the planet and add a printing colour mode :

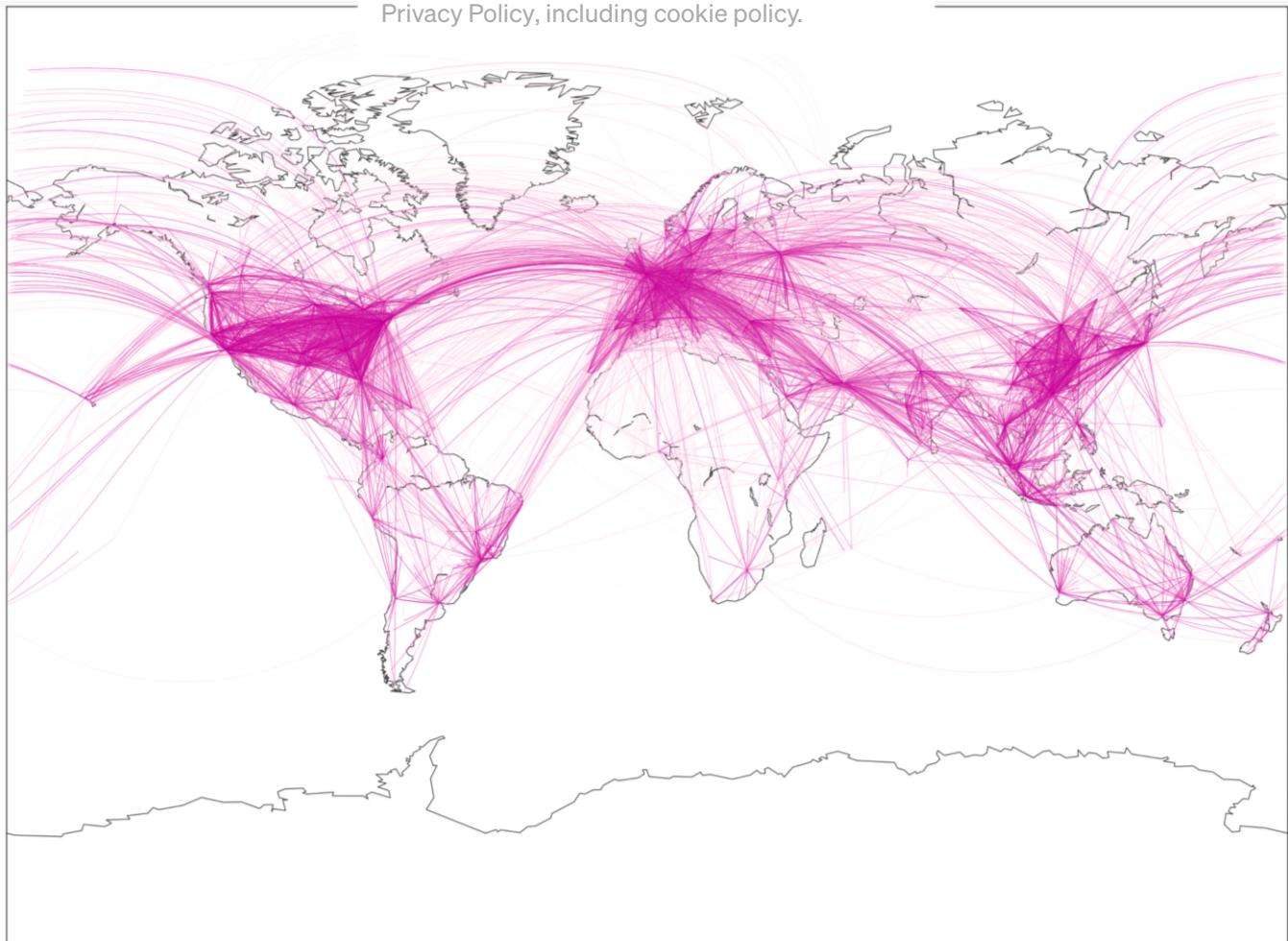




To make Medium work, we log user data.
By using Medium, you agree to our
Privacy Policy, including cookie policy.

[Open in app](#)

[Get started](#)



Saving the planet

There we are : we can easily create a nice and really comprehensive visualisation either for on screen display or printer friendly. You can now use this script to plot more useful data (like CO2 impact of the route).

[Get the code on GitHub](#) and feel free to comment!





To make Medium work, we log user data.
By using Medium, you agree to our
Privacy Policy, including cookie policy.

Open in app

Get started

Download on the
App Store

Get it on Google Play

