How to create a color bar in an osmnx plot

Asked 1 year, 6 months ago Active 1 year, 4 months ago Viewed 748 times

10

2

Currently I have created a color map based on the distance of the nodes in the network to a specific target. The one thing I am not being able to do is a color bar. I would like the color bar to show me how much time the color indicates.

The time data is in `data['time']`.

Each color will indicate how long it will take to go from the node to the target. I have defined the speed of the car.

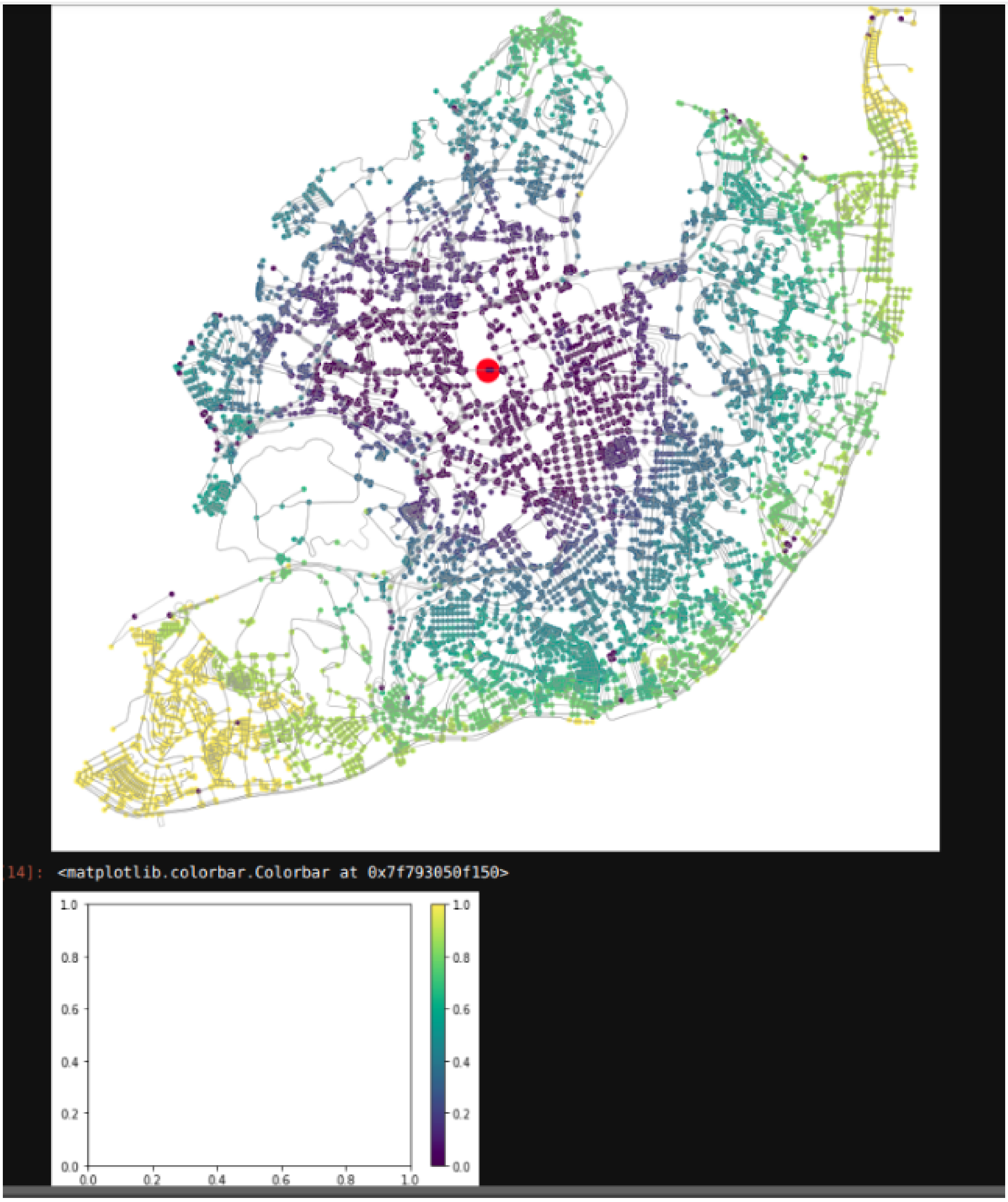
For example, a color bar that ranges from 0 to 60 min. But in this case it would go to the maximum value of `data['time']`.

Here is what I have tried:

```
import networkx as nx
import matplotlib.pyplot as plt
import osmnx as ox
import pandas as pd
from shapely.wkt import loads as load_wkt
import numpy as np
import matplotlib.cm as cm
ox.config(log_console=True, use_cache=True)

place = {'city': 'Lisbon', 'country': 'Portugal'}
G = ox.graph_from_place(place, network_type='drive')
hospitals = ox.pois_from_place(place, amenities=['hospital'])

hosp_1 = hospitals.iloc[21]['geometry']  # Hospital Santa Maria

coord_1 = (38.74817825481225, -9.160815118526642)  # Coordinate Hospital Santa Maria
target_1 = ox.get_nearest_node(G, coord_1)

nodes, edges = ox.graph_to_gdfs(G, nodes=True, edges=True)  # Transforms nodes and
edges into Geodataframes

travel_speed = 20  # km/h
meters_per_minute = travel_speed * 1000 / 60

nodes['shortest_route_length_to_target'] = 0
route_lengths = []
i = 0
# print(G.edges(data=True))
for u, v, k, data in G.edges(data=True, keys=True):

    data['time'] = data['length'] / meters_per_minute


for node in G.nodes:
    try:
        route_length = nx.shortest_path_length(G, node, target_1, weight='time')
        route_lengths.append(route_length)
        nodes['shortest_route_length_to_target'][node] = route_length
```

```
def get_colors(n, cmap='viridis', start=0., stop=1., alpha=1.):

    colors = [cm.get_cmap(cmap)(x) for x in np.linspace(start, stop, n)]
    colors = [(r, g, b, alpha) for r, g, b, _ in colors]
    return colors


def get_node_colors_by_attr(G, attr, num_bins=None, cmap='viridis', start=0, stop=1,
na_color='none'):

    if num_bins is None:
        num_bins = len(G.nodes())
    bin_labels = range(num_bins)
    # attr_values = pd.Series([data[attr] for node, data in G.nodes(data=True)])
    attr_values = pd.Series(nodes[attr].values)  # Cretaes a dataframe ith the
attribute of each node
    # print(attr_values)
    cats = pd.qcut(x=attr_values, q=num_bins, labels=bin_labels)  # Puts the values in
bins
    # print(cats)
    colors = get_colors(num_bins, cmap, start, stop)  #List of colors of each bin
    node_colors = [colors[int(cat)] if pd.notnull(cat) else na_color for cat in cats]

    return node_colors


nc = get_node_colors_by_attr(G, attr='shortest_route_length_to_target', num_bins=10)
ns = [80 if node == target_1 else 20 for node in G.nodes()]
k = 0

for node in G.nodes():

    if node == target_1:

        nc[k] = str('red')
        k += 1
    else:
        k += 1


G = ox.project_graph(G)
cmap = plt.cm.get_cmap('viridis')
norm=plt.Normalize(vmin=0, vmax=1)
sm = mpl.cm.ScalarMappable(norm=norm, cmap=cmap)
sm.set_array([])

fig, ax = ox.plot_graph(G, node_color=nc, node_size=ns, edge_linewidth=0.5, fig_height
= 13, fig_width =13, bgcolor = 'white')
plt.colorbar(sm)
```

The graph I obtain is the following:



```
[14]: <matplotlib.colorbar.Colorbar at 0x7f793050f150>
```

python matplotlib networkx osmnx

Share Improve this question Follow

edited Jun 23 '20 at 12:09
crusy
1,240 2 20 45

asked May 19 '20 at 0:02
DPM
513 2 17

1 Answer

Active Oldest Votes

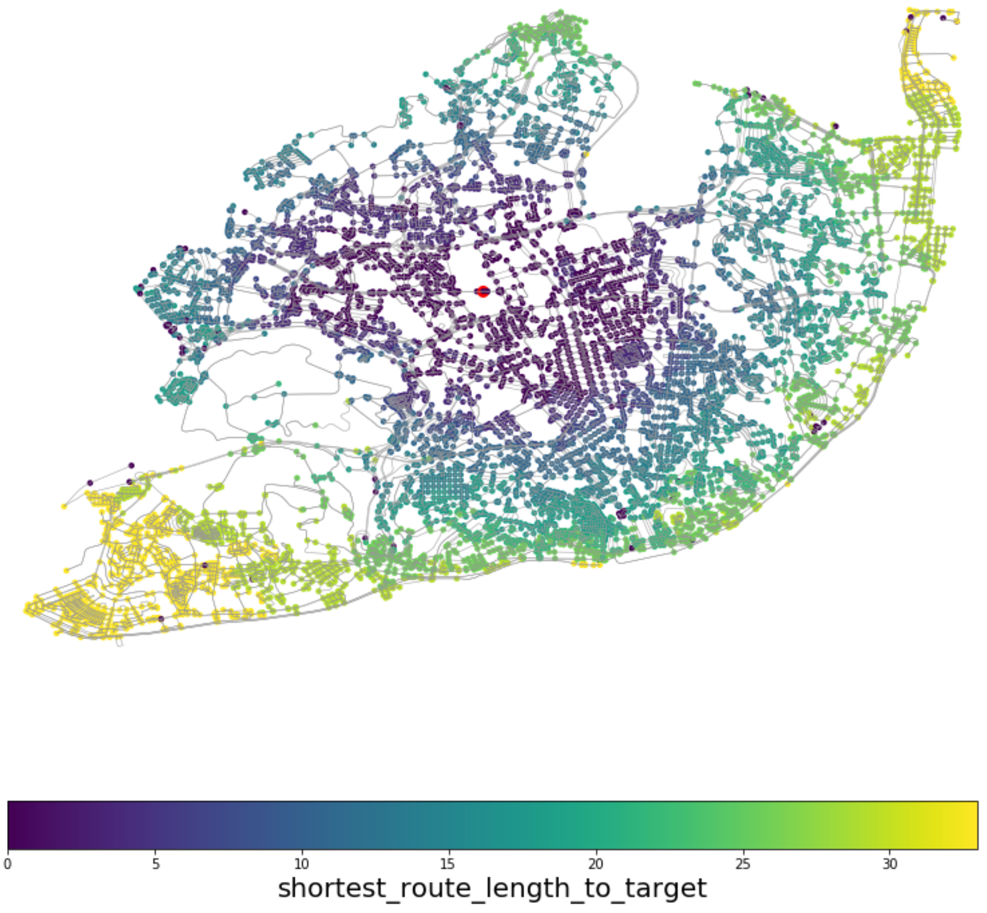
8

+100

again. I had faced this same issue earlier without having enough motivation to solve it. But somehow I have managed to do it this time (and your trial has helped a lot as well, given that my coding knowledge is limited to say the least).

See that I have changed the normalised values so that it means something on the figure instead of just ranging from 0 to 1.

```
import matplotlib as mpl
G = ox.project_graph(G)
cmap = plt.cm.get_cmap('viridis')
norm=plt.Normalize(vmin=nodes['shortest_route_length_to_target'].min(),
vmax=nodes['shortest_route_length_to_target'].max())
sm = mpl.cm.ScalarMappable(norm=norm, cmap=cmap)
sm.set_array([])

fig, ax = ox.plot_graph(G, node_color=nc, node_size=ns, edge_linewidth=0.5, fig_height
= 13, fig_width =13, bgcolor = 'white', show=False)
cb = fig.colorbar(cm.ScalarMappable(norm=norm, cmap=cmap), ax=ax,
orientation='horizontal')
cb.set_label('shortest_route_length_to_target', fontsize = 20)
fig.savefig('demo.png')
```



Join Stack Overflow to learn, share knowledge, and build your career. Sign up

Share Improve this answer Follow edited May 22 '20 at 3:33 answered May 22 '20 at 3:24

Debjit Bhowmick
712 3 17

Thank you! I can now see what I was missing – DPM May 22 '20 at 10:42