Visualisation

You are required to produce a **visualisation** of food hygiene ratings for different establishments across London.

Your visualisation should contain a map-based display of the ratings, placed according to their geolocation data. A user should be able to intuitively see which businesses are ‘safe’ to eat at, and those which have not scored so well.

You will supplement the map with additional charts and graphics that you deem appropriate to tell a coherent story from the data available.

Question 0

In this assignment, you will use Bokeh, in particular `WMTSTileSource` class to add points to a map. You will use a convenience package `bokeh.tile_providers` (http://bokeh.pydata.org/en/latest/docs/reference/tile_providers.html) which creates a `WMTSTileSource` (like used in the guided exercise) instance with the `url` and `attribution` already set. So, instead of manually creating a tile, you can just use one of the variables already created. See the Bokeh source code (https://github.com/bokeh/bokeh/blob/master/bokeh/tile_providers.py) to look at how they have done it.

The available tiles supported by Bokeh use *Web Mercator* (https://en.wikipedia.org/wiki/Web_Mercator) format to represent location, so a function `wgs84_to_web_mercator` to convert the two is provided.

Run the code in the cell below to set up the Notebook.

In [84]:

```
# You don't need to write anything here
# Set up MongoDB
from pymongo import MongoClient

client = MongoClient('mongodb://cpduser:M13pV5woDW@mongodb/health_data', 27017)
db = client.health_data

from nose.tools import *

# # Numpy, Pandas and Bokeh imports
import numpy as np
import pandas as pd
from bokeh.palettes import Spectral6
from bokeh.io import output_notebook, show
from bokeh.models.sources import ColumnDataSource
from bokeh.models import *
from bokeh.io import curdoc
from bokeh.tile_providers import *
from bokeh.models.tiles import WMTSTileSource
import ipywidgets
from ipywidgets import interact, interactive
from ipywidgets import HBox, Label, IntSlider

from bokeh.plotting import figure
from bokeh.models import TapTool, CustomJS




def wgs84_to_web_mercator(df, lon="lon", lat="lat"):
    """
    Converts decimal longitude/latitude to Web Mercator format
    Source https://github.com/bokeh/bokeh-notebooks/blob/master/tutorial/11%20-%20ge
    """
    k = 6378137
    df["x"] = df[lon] * (k * np.pi/180.0)
    df["y"] = np.log(np.tan((90 + df[lat]) * np.pi/360.0)) * k
    return df

# from ipywidgets import *
# from bokeh.layouts import *
from IPython.display import display
from bokeh.io import output_file, output_notebook, show, push_notebook

output_notebook()
```

(http://bokeh.pydata.org)

BokehJS successfully loaded.

In [54]:

```
# You don't need to write anything here
# Check it's set up correctly
try:
    imports = [MongoClient, db, np, pd, output_notebook, show, ColumnDataSource,
               output_notebook, show, ColumnDataSource, STAMEN_TERRAIN, figure
              ]
    assert True
    print('Successfully imported required libraries')
except NameError as e:
    print(e)
    assert False
```

```
Successfully imported required libraries
```

Question 1: Create Map

In this question, you will create functions which will **return** the different objects required for the visualisation of a map on: A `DataFrame` (http://pandas.pydata.org/pandas-docs/stable/generated/pandas.DataFrame.html), a `ColumnDataSource` and a `Figure`.

Question 1(a) [4 marks]

Create a function **`get_data()`** to extract data from the MongoDB database for all institutions which are in the **London** region with the following constraints:

- The results should include: `Lat`, `Lng`, `BusinessType`, `AddressLine1`, `BusinessName`, `RatingValue` but **NOT** the `_id` field
- The results should **only include businesses which have a RatingValue** (N.B. A value of 0 is a RatingValue)
- The results returned should **only include businesses which have a Geocode**
- The returned values should be **limited to 200** institutions
- **Add fields `x` and `y` in *Web Mercator* format** to specify co-ordinates on the map
- **Return** the result as as `DataFrame` (http://pandas.pydata.org/pandas-docs/stable/generated/pandas.DataFrame.html).

Hint: Week 3, Guided Exercise 4, Cursors
Hint: Week 4, Guided Exercise 2, Importing Data

In [139]:

```
dff = db.uk.find({'Region' : 'london'})
listy = list(dff)
df1 = pd.DataFrame(listy).dropna(subset=[['Geocode', 'RatingValue']])
df2 = df1.head(200)
df3 = df2[['Lat', 'Lng', 'BusinessType', 'AddressLine1', 'BusinessName', 'RatingValu
print(df3)
```

```
           Lat       Lng                            BusinessType  \
0    51.406561  0.028132    Hospitals/Childcare/Caring Premises
1    51.431513  0.076540                Restaurant/Cafe/Canteen
2    51.411260  0.104858                      Retailers - other
3    51.416480 -0.072576                      Retailers - other
4    51.359719  0.072717                      Retailers - other
5    51.369514  0.093848    Hospitals/Childcare/Caring Premises
6    51.406208 -0.015839    Hospitals/Childcare/Caring Premises
7    51.414499  0.007386    Hospitals/Childcare/Caring Premises
8    51.369514  0.093848    Hospitals/Childcare/Caring Premises
9    51.404352 -0.000290    Hospitals/Childcare/Caring Premises
10   51.369514  0.093848    Hospitals/Childcare/Caring Premises
11   51.369514  0.093848    Hospitals/Childcare/Caring Premises
12   51.404444 -0.012033    Hospitals/Childcare/Caring Premises
13   51.414901  0.008094    Hospitals/Childcare/Caring Premises
15   51.343307  0.031628                      Retailers - other
16   51.416154 -0.072978                Restaurant/Cafe/Canteen
17   51.378695  0.101438                      Retailers - other
18   51.384779 -0.022270    Hospitals/Childcare/Caring Premises
```

In [140]:

```
cursor = db.uk.find({'Region' : 'london', 'Geocode' : {'$ne' : None }, 'RatingValue
                   {'Lat' : 1, 'Lng' : 1, 'BusinessType' : 1,  'AddressLine1' : 1,

listy = list(cursor)
#rating_values = pd.DataFrame())['RatingValue']
#cursor.
#df = wgs84_to_web_mercator(df, 'lon', 'lat')
first_200 = pd.DataFrame(listy)
#dropna(subset=[['RatingValue','Geocode']])

print(first_200)
#first_200[first_200['RatingValue'] != None]
#f200 = fi_200[fi_200['Geocode'] > -1]
#first_200[['Geocode', 'Lat', 'Lng', 'BusinessType', 'AddressLine1', 'BusinessName',
```

```
                          AddressLine1  \
0                              Bromley
1                  The 19th Hole Cafe
2                  80 Grovelands Road
3                     61 Anerley Road
4                     143 High Street
5                       10 Tower Road
6                   114 Bromley Road
7                     118 London Road
8                       18 Tower Road
9                  20 Shortlands Road
10                      22 Tower Road
11                      24 Tower Road
12                   8 Oakwood Avenue
13                   Abbeyfield House
14                          Vale Farm
15               112 - 114 Anerley Road
16                      138 High Street
17              29 South Eden Park Road
18        Glebe Housing Association Ltd
```

In [ ]:

```
states = []
lons = []
lats = [] #리스트 작성하기
# The sample data is in a slightly difficult format, so we will change it to be in a
# We don't mind about the State being repeated, as long as we have all the latitudes
#     Lat       Lon         State
# 0 -82.88318   -82.88318   FL
# 1 -82.87484   -82.87484   FL
# 2 -82.86562   -82.86562   FL

for s in us_states: #US에서 시작
    # The amount of longitudes is the same as the latitudes, so this is safe
    # Iterate through each lat/lon pair

    for data in range(len(us_states[s]['lons'])):
        #범위 또는 길이는 우리가 여기에 있는 수 또는 길이를 기반으로 한다.
        states.append(s)
        lons.append(us_states[s]['lons'][data]) #경도 데이터 추가
        lats.append(us_states[s]['lats'][data]) #위도 데이터 추가


# We created 3 lists of equal length, now we create a
df = pd.DataFrame({'state': states, 'lat': lats, 'lon': lons})#dict(state=states, lo
df.head()

df = wgs84_to_web_mercator(df, 'lon', 'lat')
df.head()
```

In [141]:

```
def get_data():
    dff = db.uk.find({'Region' : 'london'})
    listy = list(dff)
    df1 = pd.DataFrame(listy).dropna(subset=[['Geocode', 'RatingValue']])
    df2 = df1.head(200)
    df3 = df2[['Lat', 'Lng', 'BusinessType', 'AddressLine1', 'BusinessName', 'Rating
    df = wgs84_to_web_mercator(df3, 'Lng', 'Lat')

    return df

    #return businesses (RV와 Geocode가 존재하고 200개 기관으로 제한 되어있어야함 비지니스만 +데이터
    raise NotImplementedError()
get_data().head()
```

```
/opt/conda/lib/python3.5/site-packages/ipykernel/__main__.py:36: Setti
ngWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-
docs/stable/indexing.html#indexing-view-versus-copy (http://pandas.pyd
ata.org/pandas-docs/stable/indexing.html#indexing-view-versus-copy)
/opt/conda/lib/python3.5/site-packages/ipykernel/__main__.py:37: Setti
ngWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-
docs/stable/indexing.html#indexing-view-versus-copy (http://pandas.pyd
ata.org/pandas-docs/stable/indexing.html#indexing-view-versus-copy)
```

Out[141]:

	Lat	Lng	BusinessType	AddressLine1	BusinessName	Rating
0	51.406561	0.028132	Hospitals/Childcare/Caring Premises	Bromley	118 Widmore Road	3.0
1	51.431513	0.076540	Restaurant/Cafe/Canteen	The 19th Hole Cafe	19th Hole Cafe	1.0
2	51.411260	0.104858	Retailers - other	80 Grovelands Road	A & A Stores	2.0
3	51.416480	-0.072576	Retailers - other	61 Anerley Road	A Butterfly	3.0
4	51.359719	0.072717	Retailers - other	143 High Street	Aambal Local Store	3.0

In [85]:

```
def get_data():
    dff = db.uk.find({'Region' : 'london', 'Geocode' : {'$ne' : None }, 'RatingValue
                    {'Lat' : 1, 'Lng' : 1, 'BusinessType' : 1,  'AddressLine1' : 1,

    listy = list(dff)
    df1 = pd.DataFrame(listy)
    df = wgs84_to_web_mercator(df1, 'Lng', 'Lat')

    return df

    #return businesses (RV와 Geocode가 존재하고 200개 기관으로 제한 되어있어야함 비지니스만 +데이터
    raise NotImplementedError()
get_data().head()
```

Out[85]:

	AddressLine1	BusinessName	BusinessType	Lat	Lng	Rating
0	Bromley	118 Widmore Road	Hospitals/Childcare/Caring Premises	51.406561	0.028132	3
1	The 19th Hole Cafe	19th Hole Cafe	Restaurant/Cafe/Canteen	51.431513	0.076540	1
2	80 Grovelands Road	A & A Stores	Retailers - other	51.411260	0.104858	2
3	61 Anerley Road	A Butterfly	Retailers - other	51.416480	-0.072576	3
4	143 High Street	Aambal Local Store	Retailers - other	51.359719	0.072717	3

In [56]:

```
# You don't need to write anything here
test_data =  get_data()
assert_equal(type(test_data),pd.DataFrame)

for index, row in test_data.iterrows():
    try:
        i = int(row['RatingValue'])
    except:
        raise AssertionError('There is a row which is not an integer.  '
                             'Make sure you exclude all those without a RatingValue
        assert False

#print('All tests passed successfully')
assert_equal(len(test_data.index), 200)
print('All tests passed successfully')
```

```
All tests passed successfully
```

Question 1(b) [2 marks]

Create a function **`get_source`** which takes a **`DataFrame`** as a parameter and manipulates it to prepare for addition to the plot. The function should:

- **Contain a column `Colour`**, which contains a hex string of the colour with which to display the establishment on the map, e.g., #d53e4f. This should be used to distinguish different RatingValue values of the businesses.
- **RatingValues will be displayed by different colours** using an appropriate palette such as **`Spectral6`** from the standard Bokeh palettes (http://bokeh.pydata.org/en/latest/docs/reference/palettes.html)
- The function should **accept an integer** as a number to filter the businesses by `RatingValue`.
- If the rating value is equal to **`-1`**, then **all businesses should be included**. Otherwise, the data should be filtered to only include businesses with a `RatingValue` of the value passed.
- The function should **return** the result as a **`DataFrame`**

Hint: Week 5, Guided Exercise 2, Data Sources

In [201]:

```
Colour = Spectral6
Colour1 = []
test_source = get_data()
#test_source['RatingValue'] =
#test_source
test_source[colour1]

Colour1 = Spectral6
test_source = pd.DataFrame({'Colour': [1,2,3,4,5], 'RatingValue' : [1,2,3,4,5], 'x'
#test_source['RatingValue'] =
i = 0
#print(test_source)
for s in Colour1:
    test_source['Colour'][i] = Colour1[i]
    i += 1

print(test_source)

#R = 1 -> 빨 2-> 주 ..
#rating_value_1 = test_source.loc[test_source['RatingValue'] == 1]
##rating_value_2 = test_source.loc[test_source['RatingValue'] == 2]
rating_value_3 = test_source.loc[test_source['RatingValue'] == 3]
#rating_value_4 = test_source.loc[test_source['RatingValue'] == 4]
#rating_value_5 = test_source.loc[test_source['RatingValue'] == 5]
#print(rating_value_1)
#print(rating_value_2)
print(rating_value_3)
print(rating_value_3['Colour'].values[0])
#print(rating_value_4)
#print(rating_value_5)
#fields = test_source.columns.values
#print(fields)
```

```
/opt/conda/lib/python3.5/site-packages/ipykernel/__main__.py:36: Setti
ngWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-
docs/stable/indexing.html#indexing-view-versus-copy (http://pandas.pyd
ata.org/pandas-docs/stable/indexing.html#indexing-view-versus-copy)
/opt/conda/lib/python3.5/site-packages/ipykernel/__main__.py:37: Setti
ngWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-
docs/stable/indexing.html#indexing-view-versus-copy (http://pandas.pyd
ata.org/pandas-docs/stable/indexing.html#indexing-view-versus-copy)

    Colour  RatingValue  x  y
0  #3288bd            1  1  1
1  #99d594            2  2  2
2  #e6f598            3  3  3
3  #fee08b            4  4  4
4  #fc8d59            5  5  5
    Colour  RatingValue  x  y
2  #e6f598            3  3  3
#e6f598
```

```
/opt/conda/lib/python3.5/site-packages/ipykernel/__main__.py:13: Setti
ngWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: http://pandas.pydata.org/pandas-
docs/stable/indexing.html#indexing-view-versus-copy (http://pandas.pyd
ata.org/pandas-docs/stable/indexing.html#indexing-view-versus-copy)
```

In [5]:

```
Colour2 = Spectral6
test_source = get_data()
test_source['Colour'] = '#ffffff'
#test_source['RatingValue'] =
#test_source
for e in range(len(test_source)):
    RV = test_source['RatingValue'][e]
    Colour = ''
    if RV == 0:
        test_source['Colour'] == Colour2[0]
    elif RV == 1.0:
        test_source['Colour'] == Colour2[1]
    elif RV == 2.0:
        test_source['Colour'] == Colour2[2]
    elif RV == 3.0:
        test_source['Colour'] == Colour2[3]
    elif RV == 4.0:
        test_source['Colour'] == Colour2[4]
    elif RV == 5.0:
        test_source['Colour'] == Colour2[5]
    elif test_source['RatingValue'] == -1.0:
        pass
test_source.loc[test_source['RatingValue'] == 0, 'Colour'] = Colour2[0]
test_source.loc[test_source['RatingValue'] == 1, 'Colour'] = Colour2[1]
test_source.loc[test_source['RatingValue'] == 2, 'Colour'] = Colour2[2]
test_source.loc[test_source['RatingValue'] == 3, 'Colour'] = Colour2[3]
test_source.loc[test_source['RatingValue'] == 4, 'Colour'] = Colour2[4]
test_source.loc[test_source['RatingValue'] == 5, 'Colour'] = Colour2[5]
test_source.head()
```

Out[5]:

	AddressLine1	BusinessName	BusinessType	Lat	Lng	Rating
0	Bromley	118 Widmore Road	Hospitals/Childcare/Caring Premises	51.406561	0.028132	3
1	The 19th Hole Cafe	19th Hole Cafe	Restaurant/Cafe/Canteen	51.431513	0.076540	1
2	80 Grovelands Road	A & A Stores	Retailers - other	51.411260	0.104858	2
3	61 Anerley Road	A Butterfly	Retailers - other	51.416480	-0.072576	3
4	143 High Street	Aambal Local Store	Retailers - other	51.359719	0.072717	3

In [86]:

```
def get_source(df, data_filter=-1):
    # YOUR CODE HERE
    Colour2 = Spectral6
    df['Colour'] = '#ffffff'
    for e in range(len(df)):
        RV = df['RatingValue'][e]
        Colour = ''
        if RV == 0:
            df['Colour'] == Colour2[0]
        elif RV == 1.0:
            df['Colour'] == Colour2[1]
        elif RV == 2.0:
            df['Colour'] == Colour2[2]
        elif RV == 3.0:
            df['Colour'] == Colour2[3]
        elif RV == 4.0:
            df['Colour'] == Colour2[4]
        elif RV == 5.0:
            df['Colour'] == Colour2[5]

    df.loc[df['RatingValue'] == 0, 'Colour'] = Colour2[0]
    df.loc[df['RatingValue'] == 1, 'Colour'] = Colour2[1]
    df.loc[df['RatingValue'] == 2, 'Colour'] = Colour2[2]
    df.loc[df['RatingValue'] == 3, 'Colour'] = Colour2[3]
    df.loc[df['RatingValue'] == 4, 'Colour'] = Colour2[4]
    df.loc[df['RatingValue'] == 5, 'Colour'] = Colour2[5]

    if data_filter == -1:
        return df
    else:
        filter1 = df['RatingValue'] == data_filter
        filter2 = df[filter1]

    return filter2

    raise NotImplementedError()
get_source(get_data(), data_filter=5).head()
```

Out[86]:

	AddressLine1	BusinessName	BusinessType	Lat	Lng	Ratin
6	114 Bromley Road	Abbeyfield	Hospitals/Childcare/Caring Premises	51.406208	-0.015839	5
7	118 London Road	Abbeyfield	Hospitals/Childcare/Caring Premises	51.414499	0.007386	5
9	20 Shortlands Road	Abbeyfield	Hospitals/Childcare/Caring Premises	51.404352	-0.000290	5
12	8 Oakwood Avenue	Abbeyfield	Hospitals/Childcare/Caring Premises	51.404444	-0.012033	5
13	Abbeyfield House	Abbeyfield Bromley Society Limited	Hospitals/Childcare/Caring Premises	51.414901	0.008094	5

In [58]:

```
# You don't need to write anything here
test_source = get_source(get_data())
assert_equal(type(test_source),pd.DataFrame)
# Check the colours are different
rating_value_1 = test_source.loc[test_source['RatingValue'] == 1]
rating_value_2 = test_source.loc[test_source['RatingValue'] == 2]
rating_value_3 = test_source.loc[test_source['RatingValue'] == 3]
rating_value_4 = test_source.loc[test_source['RatingValue'] == 4]
rating_value_5 = test_source.loc[test_source['RatingValue'] == 5]

colour_list = [rating_value_1['Colour'].values[0],
               rating_value_2['Colour'].values[0],
               rating_value_3['Colour'].values[0],
               rating_value_4['Colour'].values[0],
               rating_value_5['Colour'].values[0]]
# If they are all different, then the length of the set should be 5
assert_equal(len(set(colour_list)), 5)

fields = test_source.columns.values
assert 'x' in fields
assert 'y' in fields

# Test that the x and y are the correct web mercator format
test_lng = rating_value_1['Lng'].values[0]
test_lat = rating_value_1['Lat'].values[0]

test_x = rating_value_1['x'].values[0]
test_y = rating_value_1['y'].values[0]

k = 6378137
# print(test_lat * (k * np.pi/180.0))

assert_equal(test_x, test_lng * (k * np.pi/180.0))
assert_equal(test_y, np.log(np.tan((90 + test_lat) * np.pi/360.0)) * k)
print('All tests passed successfully')
```

```
All tests passed successfully
```

Question 1(c) [2 marks]

Create a function **`get_map`**, which **returns** a map of London using the `STAMEN_TERRAIN` (http://bokeh.pydata.org/en/latest/docs/reference/tile_providers.html#bokeh.tile_providers.STAMEN_TERRAIN) tile

- The function should **return** a **type** `Figure`
- The figure should include **all** the available **Pan/Drag tools (http://bokeh.pydata.org/en/latest/docs/user_guide/tools.html#pan-drag-tools)**, the **reset** tool, and the **mouse wheel zoom** tool
- The map should display London, at an appropriate zoom level for the data

Hint: To get the `x,y` values surrounding London, look at the smallest and largest `x` and `y` values in the data
Hint: Week 4, Guided Exercise 3, Residual Analysis (attributes)
Hint: Week 5, Guided Exercise 2, Data Sources (Bokeh Map Tiling)

In [248]:

```
test1 = get_source(get_data())
print(test1['y'].max())
print(test1['y'].min())
test2 = test1.loc[test1['RatingValue'] == 1 ]
print(test2)
```

```
6699112.66832
6675917.19275
                  AddressLine1                       BusinessName  \
1           The 19th Hole Cafe                     19th Hole Cafe
45         4 - 6 Market Parade                 Aqua Bar And Grill
46               23 Genoa Road  Archangels Montessori Day Nursery
61   305 Upper Elmers End Road                       Auntie May's
63              33 Rodway Road                         Avis Hotel
75              32 East Street                         Bar Du Vin
83           251 Croydon Road           Beckenham Charcoal Grill
94            205 High Street                    Belash Tandoori
97           59 Beckenham Road                    Belleberry Deli
136           113 Croydon Road                            Biter's
146            101 High Street                       Blue Bengal
155          53 Beckenham Road                  Bon Appetit Cafe
180        49 Chislehurst Road              Bridges Fish & Chips

                           BusinessType        Lat       Lng  RatingV
alue  \
1                Restaurant/Cafe/Canteen  51.431513  0.076540
1
45               Restaurant/Cafe/Canteen  51.407575  0.016300
1
46   Hospitals/Childcare/Caring Premises  51.409998 -0.059058
1
61                Takeaway/sandwich shop  51.392033 -0.037639
1
63     Hotel/bed & breakfast/guest house  51.412501  0.023161
1
75               Restaurant/Cafe/Canteen  51.407131  0.016468
1
83                Takeaway/sandwich shop  51.402668 -0.036839
1
94               Restaurant/Cafe/Canteen  51.413343 -0.050906
1
97                Takeaway/sandwich shop  51.408289 -0.038496
1
136              Restaurant/Cafe/Canteen  51.398829 -0.042323
1
146              Restaurant/Cafe/Canteen  51.376692 -0.020003
1
155              Restaurant/Cafe/Canteen  51.408289 -0.038496
1
180               Takeaway/sandwich shop  51.407255  0.055900
1

               x             y   Colour
1    8520.393825  6.697981e+06  #99d594
45   1814.507700  6.693708e+06  #99d594
46  -6574.306487  6.694141e+06  #99d594
61  -4189.954314  6.690935e+06  #99d594
63   2578.270726  6.694587e+06  #99d594
```

```
75   1833.209374  6.693629e+06  #99d594
83  -4100.898721  6.692833e+06  #99d594
94  -5666.829998  6.694738e+06  #99d594
97  -4285.355118  6.693836e+06  #99d594
136 -4711.374809  6.692147e+06  #99d594
146 -2226.723774  6.688199e+06  #99d594
155 -4285.355118  6.693836e+06  #99d594
180  6222.759535  6.693651e+06  #99d594
```

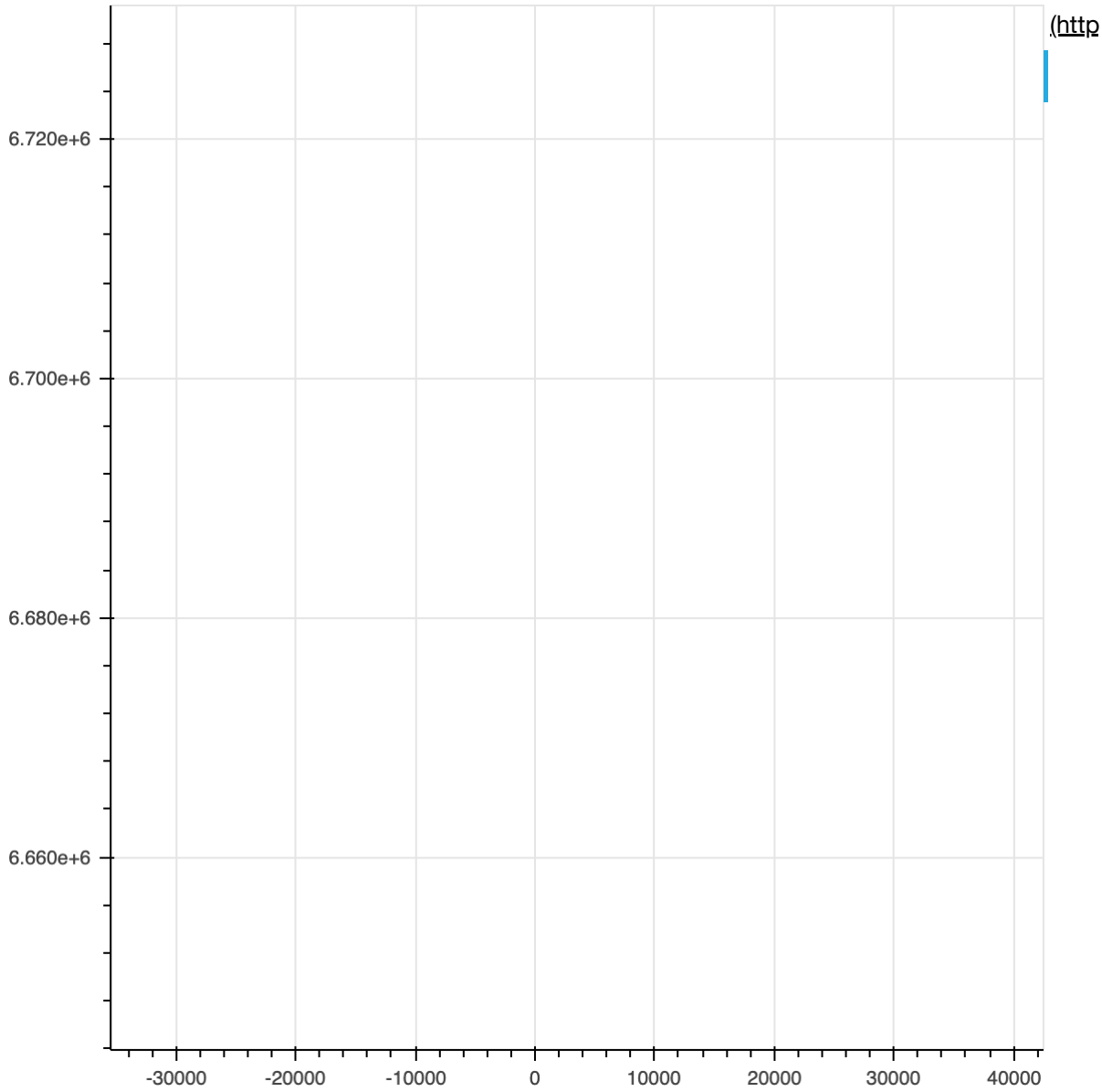
In [105]:

```
url = 'http://a.basemaps.cartocdn.com/STAMEN_TERRAIN/{Z}/{X}/{Y}.png'
x_range,y_range = ((-8123.87379911,15202.9028576), (6675917.19275,6699112.66832))
fig = figure(tools='pan, reset, wheel_zoom', x_range=x_range, y_range=y_range)
fig.add_tile(WMTSTileSource(tiles = STAMEN_TERRAIN))
show(fig)
```

```
---------------------------------------------------------------------------
-----
AttributeError                            Traceback (most recent call
 last)
<ipython-input-105-c890be898621> in <module>()
      2 x_range,y_range = ((-8123.87379911,15202.9028576), (6675917.19
275,6699112.66832))
      3 fig = figure(tools='pan, reset, wheel_zoom', x_range=x_range,
 y_range=y_range)
----> 4 fig.add_tile(WMTSTileSource(tiles = STAMEN_TERRAIN))
      5 show(fig)

/opt/conda/lib/python3.5/site-packages/bokeh/model.py in __init__(sel
f, **kwargs)
    143         self._id = kwargs.pop("id", make_id())
    144         self._document = None
--> 145         super(Model, self).__init__(**kwargs)
    146         default_theme.apply_to_model(self)
    147

/opt/conda/lib/python3.5/site-packages/bokeh/core/has_props.py in __in
it__(self, **properties)
    220
    221         for name, value in properties.items():
--> 222             setattr(self, name, value)
    223
    224     def __setattr__(self, name, value):

/opt/conda/lib/python3.5/site-packages/bokeh/core/has_props.py in __se
tattr__(self, name, value)
    256
    257             raise AttributeError("unexpected attribute '%s' to
%s, %s attributes are %s" %
--> 258                  (name, self.__class__.__name__, text, nice_joi
n(matches)))
    259
    260     def __str__(self):

AttributeError: unexpected attribute 'tiles' to WMTSTileSource, simila
r attributes are tile_size
```

In [87]:

```
from bokeh.io import output_file, show
from bokeh.layouts import column
from bokeh.plotting import figure


def get_map(data):
    """
    In this function you return a figure with a map background.  The background shou
    on London at an appropriate zoom level
    """
    # YOUR CODE HERE
    url = 'http://tile.stamen.com/terrain/{z}/{x}/{y}.jpg'
    x_range,y_range = ((-8123.87379911,15202.9028576), (6675917.19275,6699112.66832)
    fig = figure(tools='pan, reset, wheel_zoom', x_range=x_range, y_range=y_range)
    fig.add_tile(WMTSTileSource(url = url))



    return fig
    raise NotImplementedError()

show(get_map(get_source(get_data())))
```



Question 1(d) [3 marks]

Write code which creates and shows a figure `london_map` using the `get_map()` function, obtains a data source for the figure using `get_data()`, then uses the `circle` method to add the data to the map.

- You should call the output of the `circle` function `data_points`.
- The dots you add to the map should have a size of 10, no border, and a `fill_alpha` of 0.8
- You should call your map `london_map`
- Your code should contain a variable name for the dots added to the map

N.B. You are not being asked to create a function for this question

Hint: Week 4, Guided Exercise 3, Fitting a Model - Residual Analysis
Hint: Week 5, Guided Exercise 2, Widgets

In [26]:

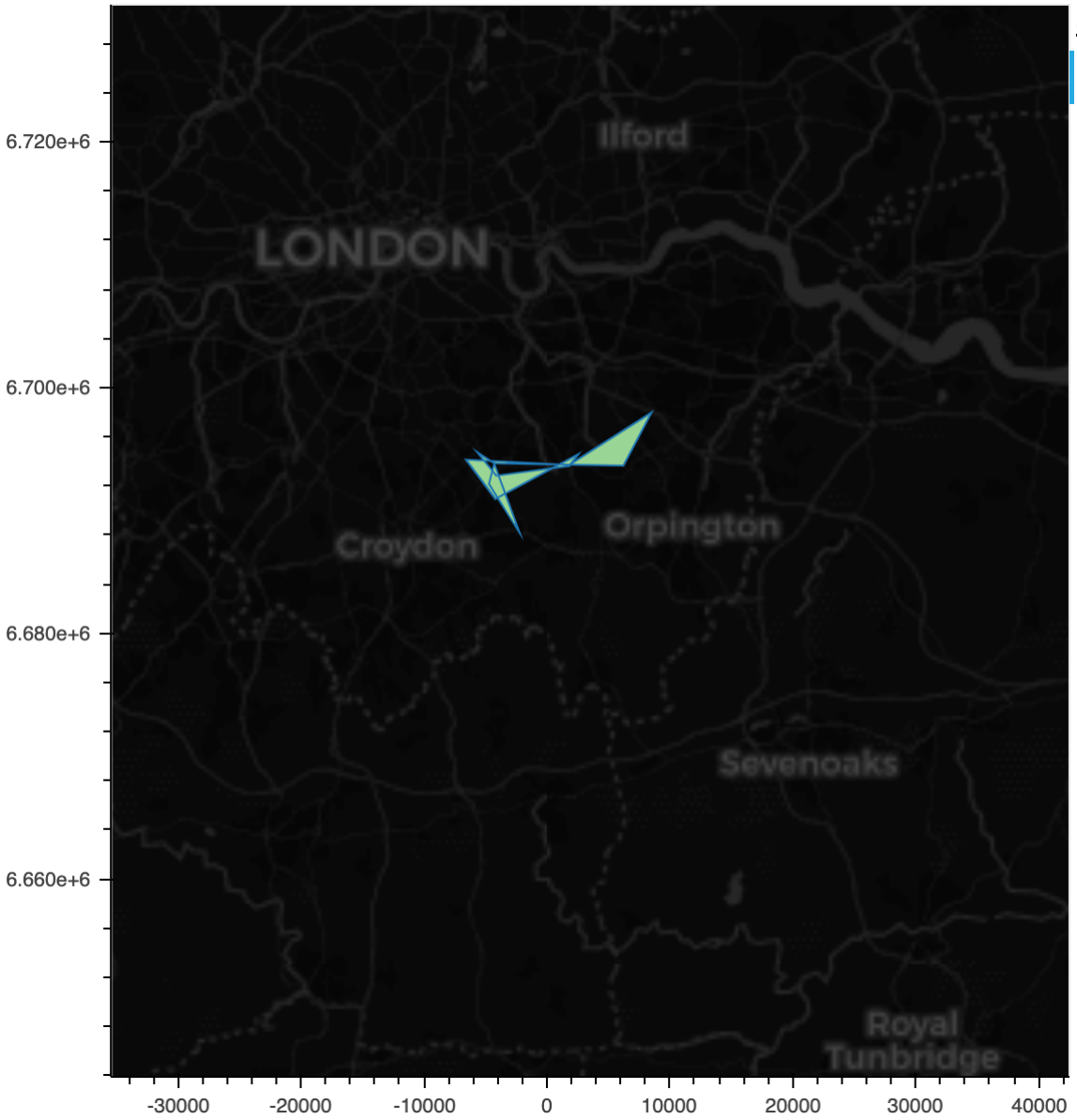
```
get_source(get_data())
```

Out[26]:

	AddressLine1	BusinessName	BusinessType	Lat	Lng	RatingValue	x
0	Bromley	118 Widmore Road	Hospitals/Childcare/Caring Premises	51.406561	0.028132	3	3131
1	The 19th Hole Cafe	19th Hole Cafe	Restaurant/Cafe/Canteen	51.431513	0.076540	1	8520
2	80 Grovelands Road	A & A Stores	Retailers - other	51.411260	0.104858	2	1167
3	61 Anerley Road	A Butterfly	Retailers - other	51.416480	-0.072576	3	-807
4	143 High Street	Aambal Local Store	Retailers - other	51.359719	0.072717	3	8094
5	10 Tower	Abbeyfield	Hospitals/Childcare/Caring	51.369514	0.093848	4	1044

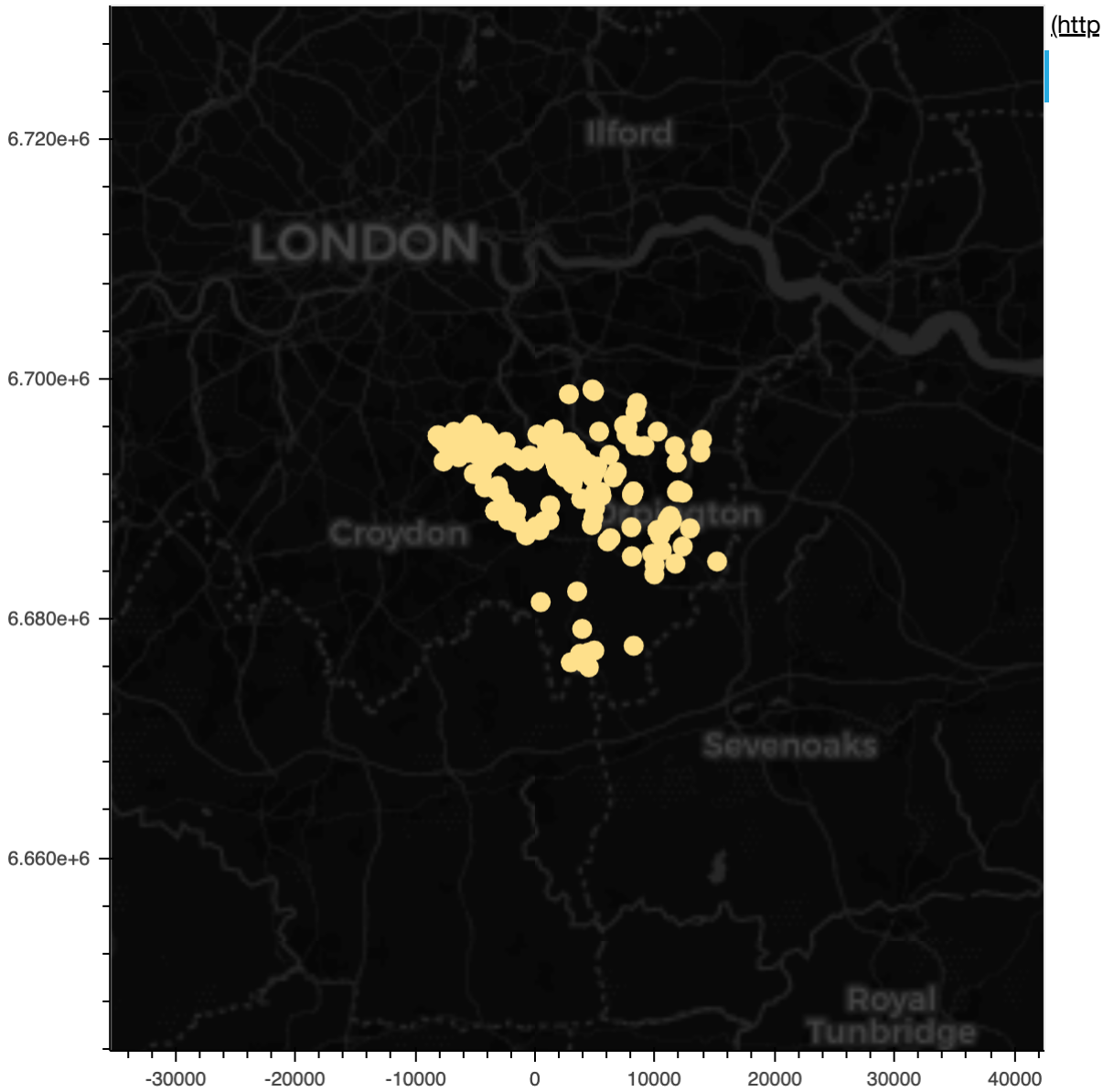
In [158]:

```
fig1 = get_map(get_source(get_data()))
test1 = get_source(get_data())
data = test1.loc[test1['RatingValue'] == 1]
colour = test1.loc[test1['RatingValue'] == 1]['Colour'].iloc[0]
ca = fig1.patch(data['x'], data['y'], fill_color= colour)
show(fig1)
```



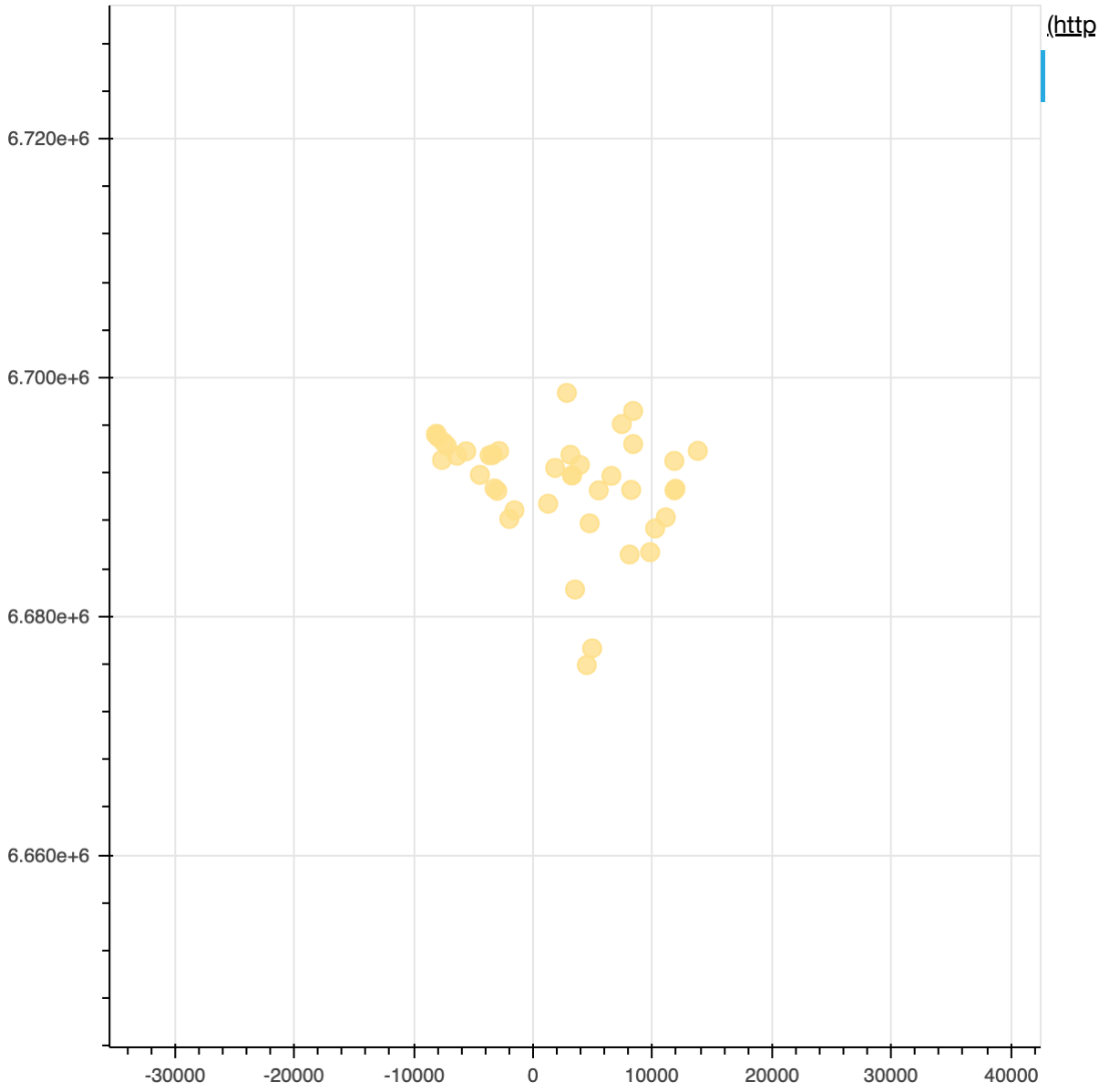
In [177]:

```
fig1 = get_map(get_source(get_data()))
test1 = get_source(get_data())
#for i in range(0):
data = test1.loc[test1['RatingValue'] == 3]
colour = test1.loc[test1['RatingValue'] == 3]['Colour'].iloc[0]
ca = fig1.circle(test1['x'], test1['y'] , size = 10, color = colour)
show(fig1)
```



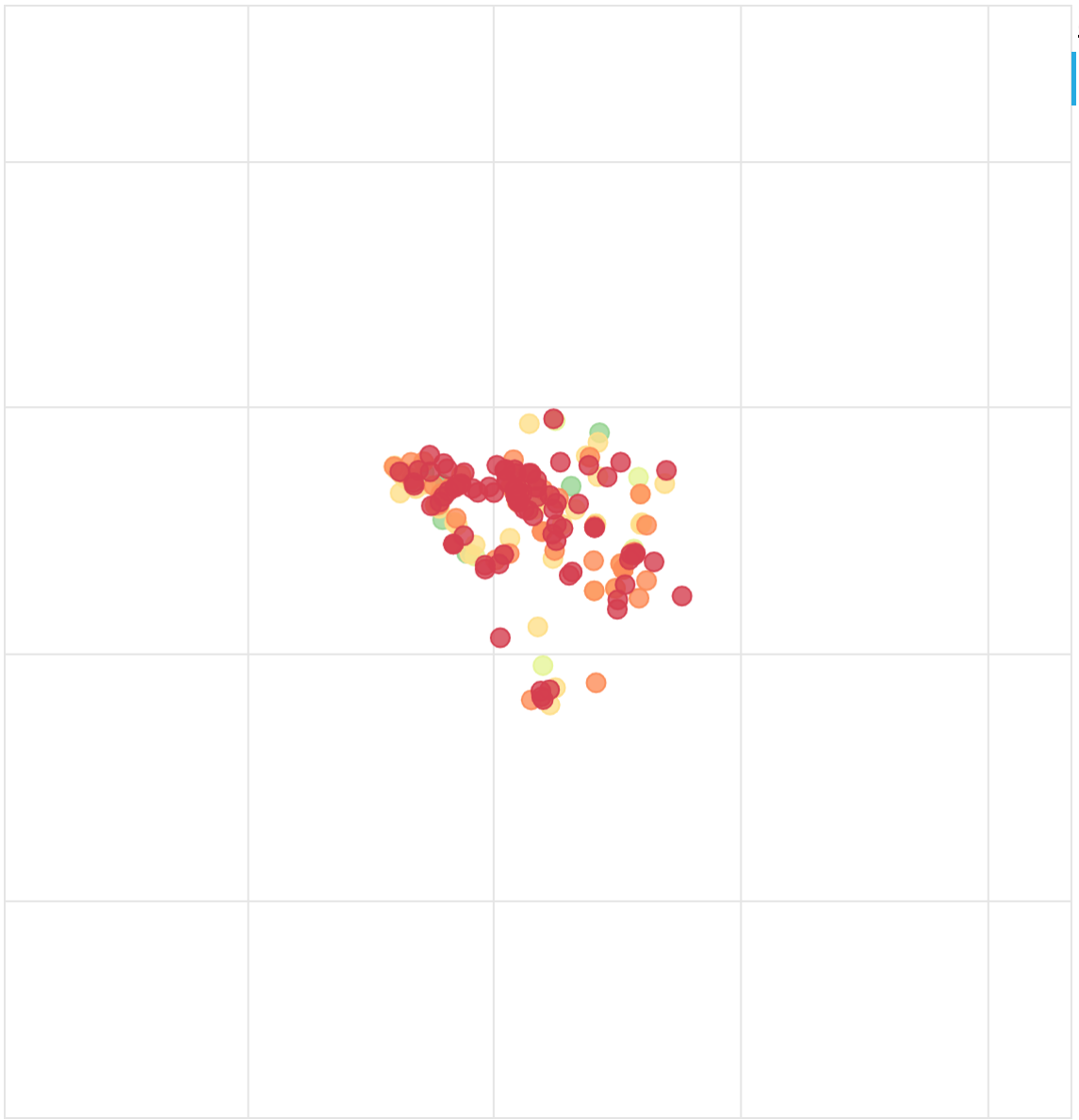
In [25]:

```
london_map = get_map(get_source(get_data()))
data = get_source(get_data())

source = data.loc[data['RatingValue'] == 3]
Colour = data.loc[test1['RatingValue'] == 3]['Colour'].iloc[0]
k = london_map.circle(source['x'], source['y'] , size = 10, color = Colour, fill_alp


show(london_map)
```



In [35]:

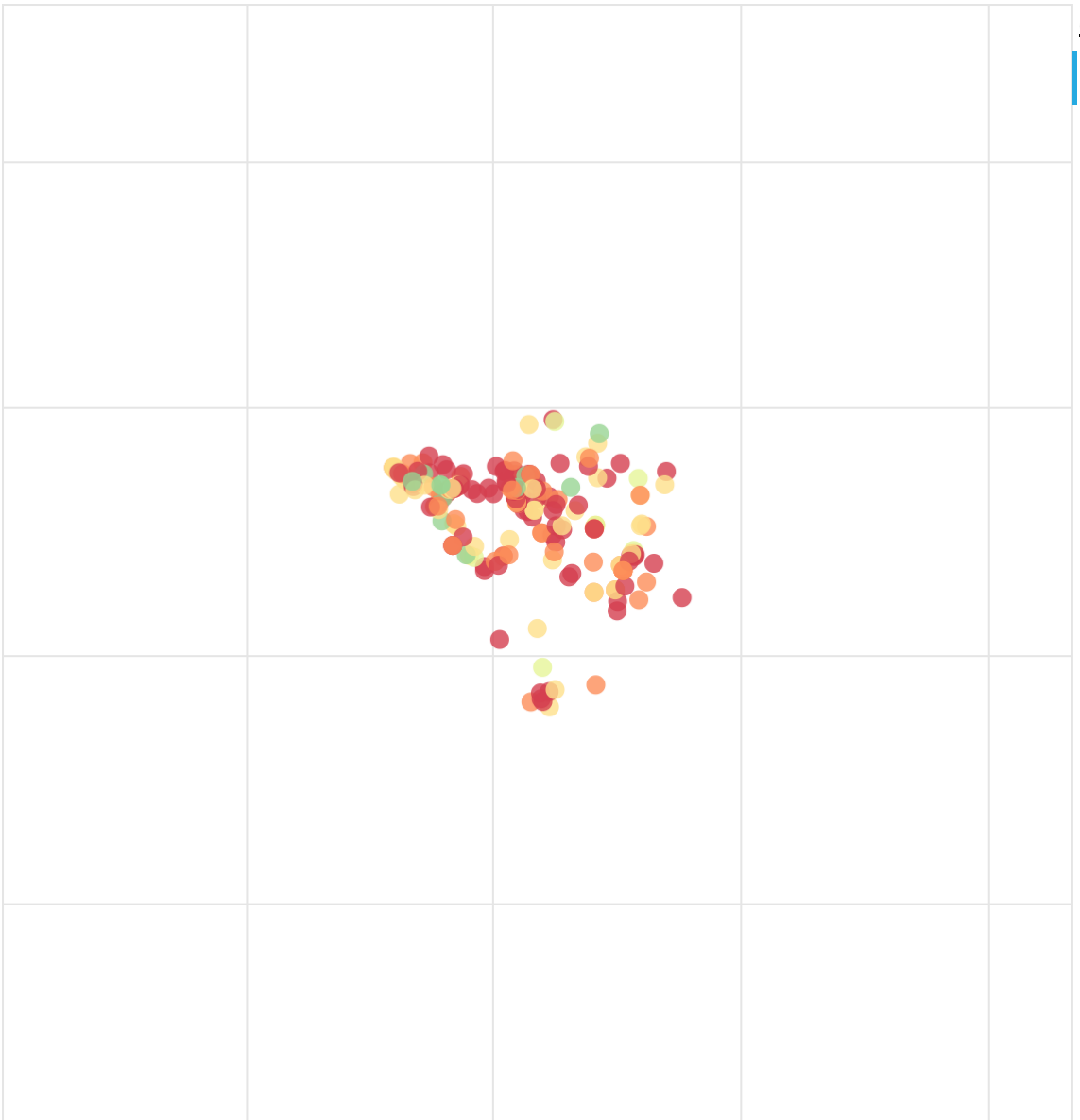
```
london_map = get_map(get_source(get_data()))
test1 = get_source(get_data())
for i in range(1,6):
    data = test1.loc[test1['RatingValue'] == i]
    colour = test1.loc[test1['RatingValue'] == i]['Colour'].iloc[0]
    data_points = london_map.circle(data['x'], data['y'] , size = 10, color = colour

london_map.axis.visible = False
show(london_map)
```



(http

In [167]:

```
source = get_source(get_data())
london_map = get_map(source)


data_points = london_map.circle(source = source, x = 'x', y = 'y',
                                fill_color = 'Colour', fill_alpha = 0.8, size = 10, l

london_map.axis.visible = False
show(london_map)
```



(http

In [75]:

```
fig_source = london_map.select(GlyphRenderer)[0].data_source
#print(vars(fig_source))
assert_equal(fig_source.data['x'][0], source['x'][0])
assert_equal(fig_source.data['y'][0], source['y'][0])
assert_equal(fig_source.data['Colour'][0], source['Colour'][0])
#assert_equal(fig_source.data['fill_color'][0], ds.data['Colour'][0])


glyph = london_map.select(GlyphRenderer)[0].glyph
assert_equal(glyph.line_color, None)
assert_equal(glyph.size, 10)
assert_equal(glyph.fill_alpha, 0.8)
#glyph.fill_color == (ds.data['Colour'][0])
print('All tests passed successfully')
```

```
All tests passed successfully
```

Question 2: Make it Interactive

Question 2(a) [2 marks]

Create a function **`callback`** for later use, which updates the visible businesses on the map according to their **`RatingValue`**. The function should have parameter **`rating`** which specifies the value to filter by, calling **`get_source`**. Use the **`source`** variable from Question 1(d) to update the map.

Hint: Week 4, Guided Exercise 2, Bokeh Charts
Hint: Week 5, Guided Exercise 2, Data Sources - Widgets (update figure)

In [169]:

```
source = get_source(get_data())
london_map = get_map(source)

data = source.loc[source['RatingValue'] == 1]
data_points.data_source.data['x'] = data['x']
data_points.data_source.data['y'] = data['y']
colour = source.loc[test1['RatingValue'] == 1]['Colour'].iloc[0]
data_points = london_map.circle(source = source, x = 'x', y = 'y',
                                fill_color = 'Colour', fill_alpha = 0.8, size = 10, l

push_notebook()
london_map.axis.visible = False
show(london_map)
```

```
/opt/conda/lib/python3.5/site-packages/bokeh/models/sources.py:81: Bok
ehUserWarning: ColumnDataSource's columns must be of the same length
  lambda: warnings.warn("ColumnDataSource's columns must be of the sam
e length", BokehUserWarning))
/opt/conda/lib/python3.5/site-packages/bokeh/models/sources.py:81: Bok
ehUserWarning: ColumnDataSource's columns must be of the same length
  lambda: warnings.warn("ColumnDataSource's columns must be of the sam
e length", BokehUserWarning))

---------------------------------------------------------------------
-----
NameError                                 Traceback (most recent call
 last)
<ipython-input-169-cac1a65ef2cb> in <module>()
      5 data_points.data_source.data['x'] = data['x']
      6 data_points.data_source.data['y'] = data['y']
----> 7 colour = source.loc[test1['RatingValue'] == 1]['Colour'].iloc[
0]
      8 data_points = london_map.circle(source = source, x = 'x', y =
 'y',
      9                                 fill_color = 'Colour', fill_alp
ha = 0.8, size = 10, line_color = None)

NameError: name 'test1' is not defined
```

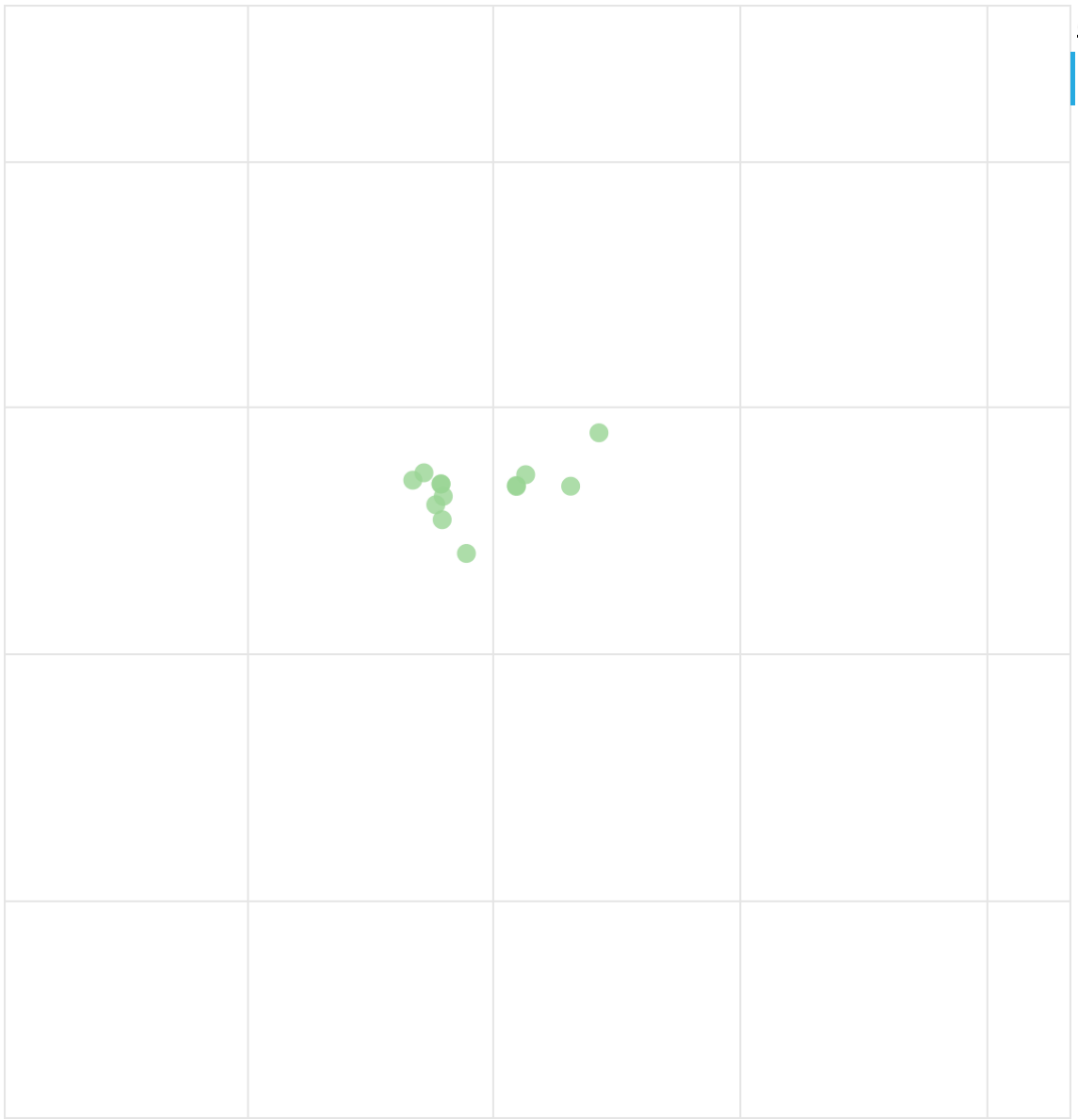
In [171]:

```
def callback(rating):
    # YOUR CODE HERE
    source = get_source(get_data())
    london_map = get_map(source)

    data = source.loc[source['RatingValue'] == rating]
    colour = source.loc[source['RatingValue'] == rating]['Colour'].iloc[0]
    data_points = london_map.circle(source = source, x = 'x', y = 'y',
                                    fill_color = colour, fill_alpha = 0.8, size = 10,

    data_points.data_source.data['x'] = data['x']
    data_points.data_source.data['y'] = data['y']
    push_notebook()
    london_map.axis.visible = False
    show(london_map, notebook_handle=True)
    return london_map
    raise NotImplementedError()
```

In [172]:

```
callback(1)
```

```
/opt/conda/lib/python3.5/site-packages/bokeh/models/sources.py:81: Bok
ehUserWarning: ColumnDataSource's columns must be of the same length
  lambda: warnings.warn("ColumnDataSource's columns must be of the sam
e length", BokehUserWarning))
/opt/conda/lib/python3.5/site-packages/bokeh/models/sources.py:81: Bok
ehUserWarning: ColumnDataSource's columns must be of the same length
  lambda: warnings.warn("ColumnDataSource's columns must be of the sam
e length", BokehUserWarning))
/opt/conda/lib/python3.5/site-packages/bokeh/io.py:578: UserWarning: C
annot find a last shown plot to update. Call output_notebook() and sho
w(..., notebook_handle=True) before push_notebook()
  warnings.warn("Cannot find a last shown plot to update. Call output_
notebook() and show(..., notebook_handle=True) before push_notebook
()")
```



(http

Out[172]:

Figure(id = '14dd3e02-375c-4823-addc-7cbb2d149a9f', …)

Question 2(b) [2 marks]

Using **`ipywidgets`**, create an interactive (http://ipywidgets.readthedocs.io/en/latest/examples/Using%20Interact.html) `IntSlider` widget, which calls the **`callback`** function when it updates.

Return the interactive widget in the function **`set_interactive()`**

Hint: Week 5, Guided Exercise 2, Widgets

In [162]:

```
def set_interactive():
    # YOUR CODE HERE
    i = interactive(callback, rating=3)
    i

    return i
    raise NotImplementedError()

set_interactive()
```

```
/opt/conda/lib/python3.5/site-packages/bokeh/models/sources.py:81: Bok
ehUserWarning: ColumnDataSource's columns must be of the same length
  lambda: warnings.warn("ColumnDataSource's columns must be of the sam
e length", BokehUserWarning))
/opt/conda/lib/python3.5/site-packages/bokeh/models/sources.py:81: Bok
ehUserWarning: ColumnDataSource's columns must be of the same length
  lambda: warnings.warn("ColumnDataSource's columns must be of the sam
e length", BokehUserWarning))
/opt/conda/lib/python3.5/site-packages/bokeh/io.py:578: UserWarning: C
annot find a last shown plot to update. Call output_notebook() and sho
w(..., notebook_handle=True) before push_notebook()
  warnings.warn("Cannot find a last shown plot to update. Call output_
notebook() and show(..., notebook_handle=True) before push_notebook
()")
```


(http

Figure(id = '154b6642-fd86-4331-aa0b-1f9cf2389c96', …)

In [145]:

```
# You don't need to write anything here
old_callback = callback
del callback

try:
    set_interactive()
except NameError as e:
    pass
else:
    raise AssertionError('You have not called the callback function in your code')
finally:
    callback = old_callback
    del old_callback
assert_equal(type(set_interactive()), ipywidgets.widgets.widget_box.Box)
```

Question 3: Extend the Visualisation

Applying question 2 solutions

Now you have created an initial visualisation, you are going to add the following components to it:

- **Hover** text, so that each dot will give information about the business when you hover
- A **drop down menu** to limit the type of business

You will also be asked to explain a possible use-case for this chart, and offer a suggestion as to how it could be improved.

NOTE: There are discretionary marks available for good visualisation practice

Question 3(a) [5 marks]

Create a function **`get_hover`**, which returns a HoverTool (http://bokeh.pydata.org/en/latest/docs/user_guide/tools.html#hover-tool) to be added to the map. When the cursor hovers over any circle, the following information should be displayed:

- The **name** of the establishment
- The **type** of the establishment
- The **`RatingValue`** of the establishment

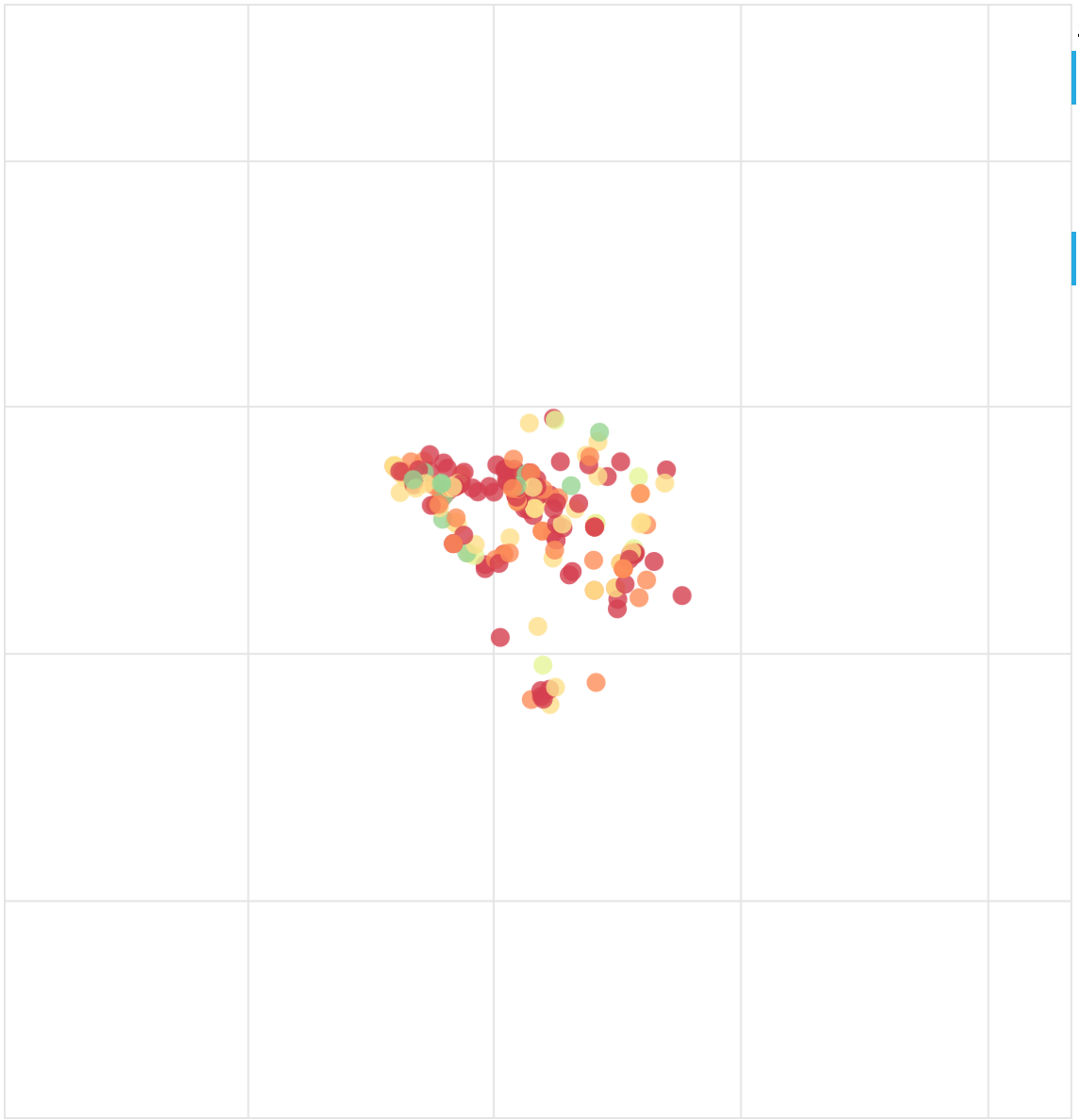
Your function should **return** the HoverTool (http://bokeh.pydata.org/en/latest/docs/user_guide/tools.html#hover-tool).

N.B You may need to read the documentation available to ensure that you match a suitable title for the **field** with the **field value** using **`@`**.

Hint: Week 5, Guided Exercise 2, Tools

In [168]:

```
def get_hover():
    # YOUR CODE HERE
    hover_tool = HoverTool(
    tooltips=[
        ( 'BusinessName',   '@BusinessName'            ),
        ( 'BusinessType',  '@BusinessType' ), # use @{ } for field names with spaces
        ( 'RatingValue', '@RatingValue'      )
    ],

    # display a tooltip whenever the cursor is vertically in line with a glyph
    mode='vline'
    )

    return hover_tool
    raise NotImplementedError()

london_map.add_tools(get_hover())
show(london_map)
```



(http

In [165]:

```
from bokeh.plotting import Figure
tools_fig_1 = Figure(tools=[HoverTool(
    tooltips=[
        ( 'BusinessName',   '@BusinessName'            ),
        ( 'BusinessType',  '@BusinessType' ), # use @{ } for field names with spaces
        ( 'RatingValue', '@RatingValue'      )
    ],

    # display a tooltip whenever the cursor is vertically in line with a glyph
    mode='vline'
    ), BoxSelectTool()], height=100)
show(tools_fig_1)
```


(http

Question 3(b) [5 marks]

Using `interactive` (http://ipywidgets.readthedocs.io/en/latest/examples/Using%20Interact.html), create a **dropdown menu** which allows the user to choose between different **types of business**. The map should update automatically as you select.

- You should create two functions **`filter_business_types`**, which the dropdown menu will call when it changes. This function should update the data source
- You will need to obtain a list of the different types of business in the database
- You should **return** an **`interactive`** object, which should call the function **`filter_business_types`**, which updates the **`source`** variable accordingly

N.B. Should you use the original data set?

Hint: Week 5, Guided Exercise 2, Data Sources - Tools

In [315]:

```
#source.loc[source['BusinessType'] == 'Retailers - other']
source['BusinessType'].drop_duplicates()
j = source['BusinessType'].drop_duplicates()
listie = j[0], j[1], j[2], j[24], j[26], j[28], j[39], j[40], j[63], j[78], j[189]
print(listie)
```

```
('Hospitals/Childcare/Caring Premises', 'Restaurant/Cafe/Canteen', 'Re
tailers - other', 'Other catering premises', 'Retailers - supermarket
s/hypermarkets', 'School/college/university', 'Pub/bar/nightclub', 'Ta
keaway/sandwich shop', 'Hotel/bed & breakfast/guest house', 'Mobile ca
terer', 'Distributors/Transporters')
```

In [317]:

```
def filter_business_types(business_type):
    # YOUR CODE HERE
    #get_source(get_data())
    k = source.loc[source['BusinessType'] == business_type]


    return k

j = source['BusinessType'].drop_duplicates()
listie = j[0], j[1], j[2], j[24], j[26], j[28], j[39], j[40], j[63], j[78], j[189]
kkk = interact(filter_business_types, business_type = (listie) )
```

	AddressLine1	BusinessName	BusinessType	Lat	Lng	RatingValue	x
0	Bromley	118 Widmore Road	Hospitals/Childcare/Caring Premises	51.406561	0.028132	3	3131.6
5	10 Tower Road	Abbeyfield	Hospitals/Childcare/Caring Premises	51.369514	0.093848	4	10447
6	114 Bromley Road	Abbeyfield	Hospitals/Childcare/Caring Premises	51.406208	-0.015839	5	-1763.
7	118 London Road	Abbeyfield	Hospitals/Childcare/Caring Premises	51.414499	0.007386	5	822.20
8	18 Tower Road	Abbeyfield	Hospitals/Childcare/Caring Premises	51.369514	0.093848	4	10447
9	20 Shortlands Road	Abbeyfield	Hospitals/Childcare/Caring Premises	51.404352	-0.000290	5	-32.28

In [313]:

```
def filter_business_types(business_type):
    # YOUR CODE HERE
    source = get_source(get_data())
    london_map = get_map(source)

    data = source.loc[source['BusinessName'] == business_type]

    return data

    raise NotImplementedError()

def get_dropdown_list():
    # YOUR CODE HERE

    j = source['BusinessType'].drop_duplicates()
    listie = j[0], j[1], j[2], j[24], j[26], j[28], j[39], j[40], j[63], j[78], j[18
    print(listie)
    kkk = interact(filter_business_types, business_type = (listie) )
    return kkk
    raise NotImplementedError()

# DISPLAY THE MAP
data = get_data()
source = get_source(data)
mappy = get_map(data)
dots = mappy.circle(source=source, x='x', y='y',fill_color='Colour', size=10, fill_a
mappy.add_tools(get_hover())
dropdown = get_dropdown_list()
slider = set_interactive()
show(mappy, notebook_handle=True)
```

	AddressLine1	BusinessName	BusinessType	Lat	Lng	RatingValue	x	y	Colour

In [311]:

```
# You don't have to write anything here
# Display the widgets
HBox([slider, dropdown])

# Why do you think we might get a Bokeh warning diplayed when we use our data?
```

```
---------------------------------------------------------------------------
-----
ValueError                                Traceback (most recent call
 last)
<ipython-input-311-9a490881eaa4> in <module>()
      1 # You don't have to write anything here
      2 # Display the widgets
----> 3 HBox([slider, dropdown])
      4
      5 # Why do you think we might get a Bokeh warning diplayed when
 we use our data?

/opt/conda/lib/python3.5/site-packages/ipywidgets/widgets/widget_box.p
y in HBox(*pargs, **kwargs)
     90 def HBox(*pargs, **kwargs):
     91     """Displays multiple widgets horizontally using the flexib
le box model."""
---> 92     box = Box(*pargs, **kwargs)
     93     box.layout.display = 'flex'
     94     box.layout.align_items = 'stretch'

/opt/conda/lib/python3.5/site-packages/ipywidgets/widgets/widget_box.p
y in __init__(self, children, **kwargs)
     42     def __init__(self, children = (), **kwargs):
     43         kwargs['children'] = children
---> 44         super(Box, self).__init__(**kwargs)
     45         self.on_displayed(Box._fire_children_displayed)
     46

/opt/conda/lib/python3.5/site-packages/ipywidgets/widgets/domwidget.py
in __init__(self, *pargs, **kwargs)
     88
     89     def __init__(self, *pargs, **kwargs):
---> 90         super(DOMWidget, self).__init__(*pargs, **kwargs)
     91
     92         # Deprecation added in 5.0.  TODO: Remove me and corre
sponging traits.

/opt/conda/lib/python3.5/site-packages/ipywidgets/widgets/widget.py in
__init__(self, **kwargs)
    182
    183         Widget._call_widget_constructed(self)
--> 184         self.open()
    185
    186     def __del__(self):

/opt/conda/lib/python3.5/site-packages/ipywidgets/widgets/widget.py in
open(self)
    201                 args['comm_id'] = self._model_id
    202
--> 203             self.comm = Comm(**args)
    204             if buffers:
    205                 # FIXME: workaround ipykernel missing binary m
```

```
essage support in open-on-init

/opt/conda/lib/python3.5/site-packages/ipykernel/comm/comm.py in __ini
t__(self, target_name, data, metadata, buffers, **kwargs)
     55             if self.primary:
     56                 # I am primary, open my peer.
---> 57                 self.open(data=data, metadata=metadata, buffer
s=buffers)
     58             else:
     59                 self._closed = False

/opt/conda/lib/python3.5/site-packages/ipykernel/comm/comm.py in open
(self, data, metadata, buffers)
     92                                 data=data, metadata=metadata, bu
ffers=buffers,
     93                                 target_name=self.target_name,
---> 94                                 target_module=self.target_module
,
     95                                 )
     96             self._closed = False

/opt/conda/lib/python3.5/site-packages/ipykernel/comm/comm.py in _publ
ish_msg(self, msg_type, data, metadata, buffers, **keys)
     63         data = {} if data is None else data
     64         metadata = {} if metadata is None else metadata
---> 65         content = json_clean(dict(data=data, comm_id=self.comm
_id, **keys))
     66         self.kernel.session.send(self.kernel.iopub_socket, msg
_type,
     67             content,

/opt/conda/lib/python3.5/site-packages/ipykernel/jsonutil.py in json_c
lean(obj)
    165         out = {}
    166         for k,v in iteritems(obj):
--> 167             out[unicode_type(k)] = json_clean(v)
    168         return out
    169     if isinstance(obj, datetime):

/opt/conda/lib/python3.5/site-packages/ipykernel/jsonutil.py in json_c
lean(obj)
    165         out = {}
    166         for k,v in iteritems(obj):
--> 167             out[unicode_type(k)] = json_clean(v)
    168         return out
    169     if isinstance(obj, datetime):

/opt/conda/lib/python3.5/site-packages/ipykernel/jsonutil.py in json_c
lean(obj)
    151
    152     if isinstance(obj, list):
--> 153         return [json_clean(x) for x in obj]
    154
    155     if isinstance(obj, dict):

/opt/conda/lib/python3.5/site-packages/ipykernel/jsonutil.py in <listc
omp>(.0)
    151
    152     if isinstance(obj, list):
--> 153         return [json_clean(x) for x in obj]
    154
```

```
    155     if isinstance(obj, dict):

/opt/conda/lib/python3.5/site-packages/ipykernel/jsonutil.py in json_c
lean(obj)
    171
    172     # we don't understand it, it's probably an unserializable
 object
--> 173     raise ValueError("Can't clean for JSON: %r" % obj)

ValueError: Can't clean for JSON: <function filter_business_types at 0
x7fd7db058b70>
```

Question 3(c) [5 marks]

Describe a use case for which an application like this would be useful, and suggest one way which it could be improved.

이와 같은 유용한 어플리케이션 사례는 코로나 맵이 있습니다. 요즈음 코로나 바이러스 문제로 인하여 확진자가 발생하면 동선을 나타나는 맵을 나라별, 지역별 코로나 맵을 만들어서 어플리케이션 이용하는 사람들에게 확진자가 어디를 다녀왔는지 시각적으로 알아 볼 수 있습니다. 이를 개선시킬 방법으로는 레이팅 벨류가 0부터 5까지 구분 되는 것 처럼, 일자별로 구분되게 필터링을 만들어서 좀 더 이용자가 알아보기 쉽고 유용하게 어플리케이션을 사용하는 방법이 있습니다.

In [ ]: