Bokeh 311 Dashboard

Serving Bokeh Apps

Write code in your favorite text editor:
Sublime, Atom, TextWrangler, BBEdit, etc.

311-dashboard-101.py × 311-dashboard-102.py × 311-dashboard-103.py ×

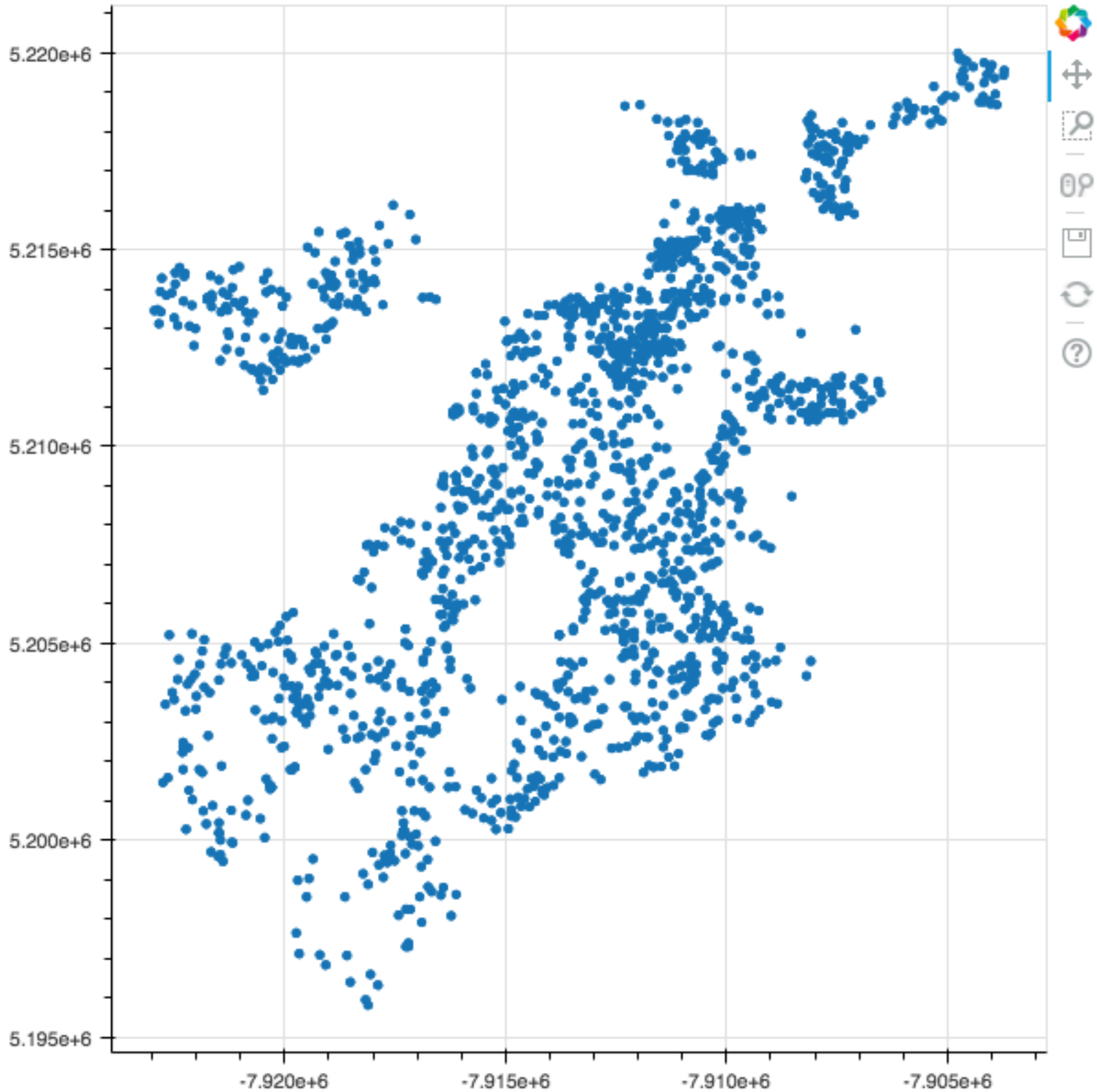
```python
# "current document" where all the the plots and layouts are held
from bokeh.io import curdoc

from bokeh.plotting import figure
from bokeh.models import ColumnDataSource
import pandas as pd

# read the dataset
service_requests = pd.read_csv('../datasets/service-requests.csv', index_col=0)

# convert the x and y coordinates from the DataFrame to a CDS
sr_cds = ColumnDataSource(data=dict(
    x = service_requests['wm_x'],
    y = service_requests['wm_y'],
    ))

# create the blank figure
p = figure()

# create circle glyphs with web mercator x and y coordinates
p.circle('x', 'y', source=sr_cds)

# add the plot to the current document
curdoc().add_root(p)
```

Use bokeh serve —show to display the html & js output

```
[[gmaclenn:~]$ bokeh serve --show my-app.py
```

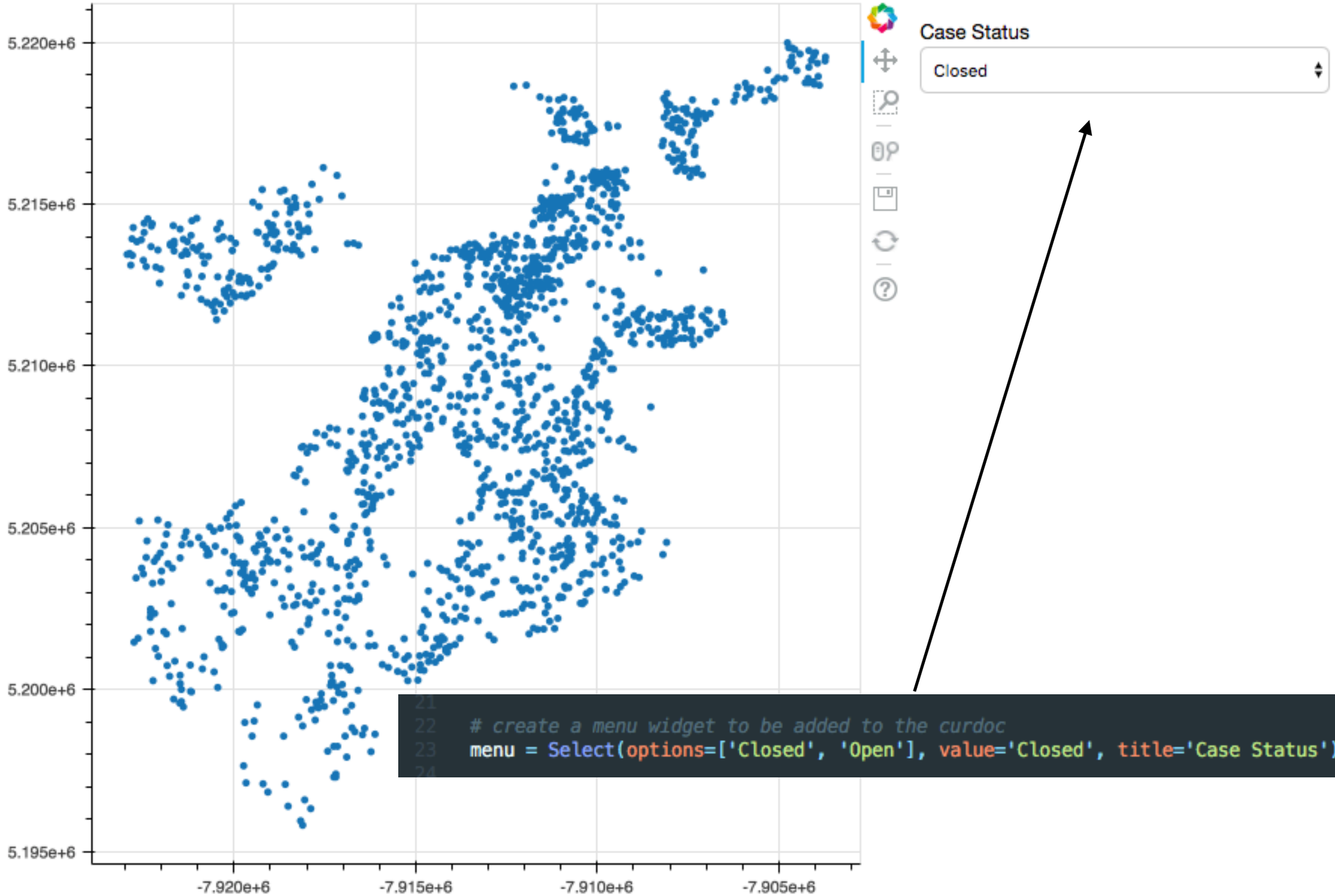
311-dashboard-101.py × 311-dashboard-102.py × 311-dashboard-103.py × 31

```python
# "current document" where all the the plots and layouts are held
from bokeh.io import curdoc

from bokeh.plotting import figure
from bokeh.models import ColumnDataSource
import pandas as pd

# read the dataset
service_requests = pd.read_csv('../datasets/service-requests.csv', index_col=0)

# convert the x and y coordinates from the DataFrame to a CDS
sr_cds = ColumnDataSource(data=dict(
    x = service_requests['wm_x'],
    y = service_requests['wm_y'],
    ))

# create the blank figure
p = figure()

# create circle glyphs with web mercator x and y coordinates
p.circle('x', 'y', source=sr_cds)

# add the plot to the current document
curdoc().add_root(p)
```


5.220e+6
5.215e+6
5.210e+6
5.205e+6
5.200e+6
5.195e+6
-7.920e+6
-7.915e+6
-7.910e+6
-7.905e+6

311-dashboard-101.py × 311-dashboard-102.py × 311-dashboard-103.py × 31

```python
from bokeh.io import curdoc
from bokeh.plotting import figure
from bokeh.models import ColumnDataSource, Select
from bokeh.layouts import row
import pandas as pd

# read the dataset
service_requests = pd.read_csv('../datasets/service-requests.csv', index_col=0)

# convert it to a ColumnDataSource
sr_cds = ColumnDataSource(data=dict(
    x = service_requests['wm_x'],
    y = service_requests['wm_y'],
    ))

# create the blank figure
p = figure()

# create circle glyphs with latitude and longitude coordinates as x and y
p.circle('x', 'y', source=sr_cds)

# create a menu widget to be added to the curdoc
menu = Select(options=['Closed', 'Open'], value='Closed', title='Case Status')

# create a layout with one row
layout = row(p, menu)

# add the layout to the current document
curdoc().add_root(layout)
```




```
# create a menu widget to be added to the curdoc
menu = Select(options=['Closed', 'Open'], value='Closed', title='Case Status')
```

311-dashboard-101.py × 311-dashboard-102.py × 311-dashboard-103.py × 311-da

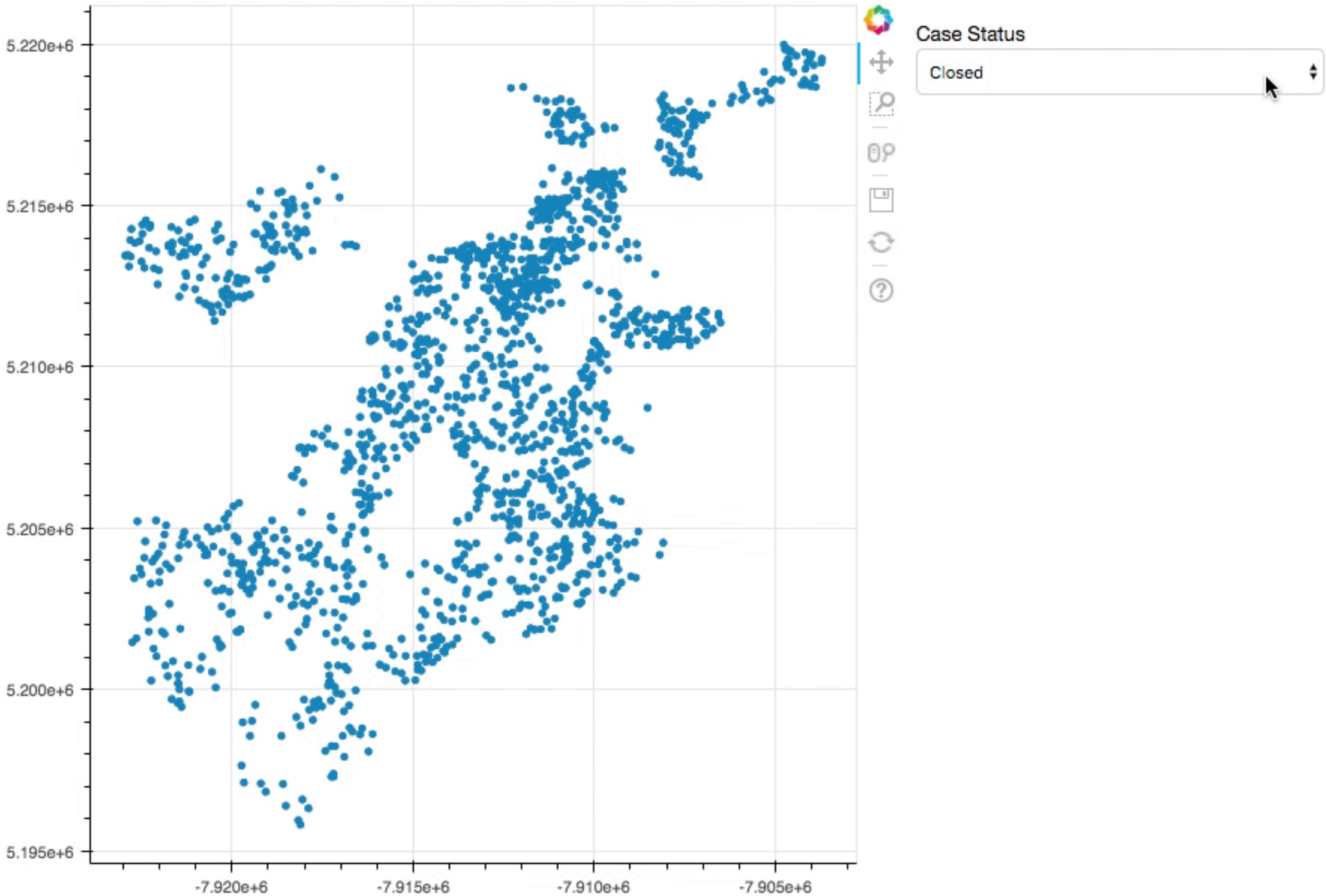
```python
from bokeh.io import curdoc
from bokeh.plotting import figure
from bokeh.models import ColumnDataSource, Select
from bokeh.layouts import row
import pandas as pd

# read the dataset
service_requests = pd.read_csv('../datasets/service-requests.csv', index_col=0)

# convert to CDS format
sr_cds = ColumnDataSource(data=dict(
    x = [],
    y = [],
    ))

# create the blank figure
p = figure()

# create circle glyphs with latitude and longitude coordinates as x and y
p.circle('x', 'y', source=sr_cds)

# create a menu widget to be added to the curdoc
menu = Select(options=['Closed', 'Open'], value='Closed', title='Case Status')

def select_data():
    menu_val = menu.value
    temp_df = service_requests
    filtered_df = temp_df[temp_df.CASE_STATUS.str.contains(menu_val) == True]
    return filtered_df

def update_plot():
    df = select_data()
    sr_cds.data = dict(
        x = df['wm_x'],
        y = df['wm_y'])

menu.on_change('value', lambda attr, old, new: update_plot())

# create a layout with one row
layout = row(p, menu)

update_plot() # run once to get data from the intial 'Closed' value

# add the layout to the current document
curdoc().add_root(layout)
```

* Create a blank ColumnDataSource

* Filter the DataFrame by the menu value

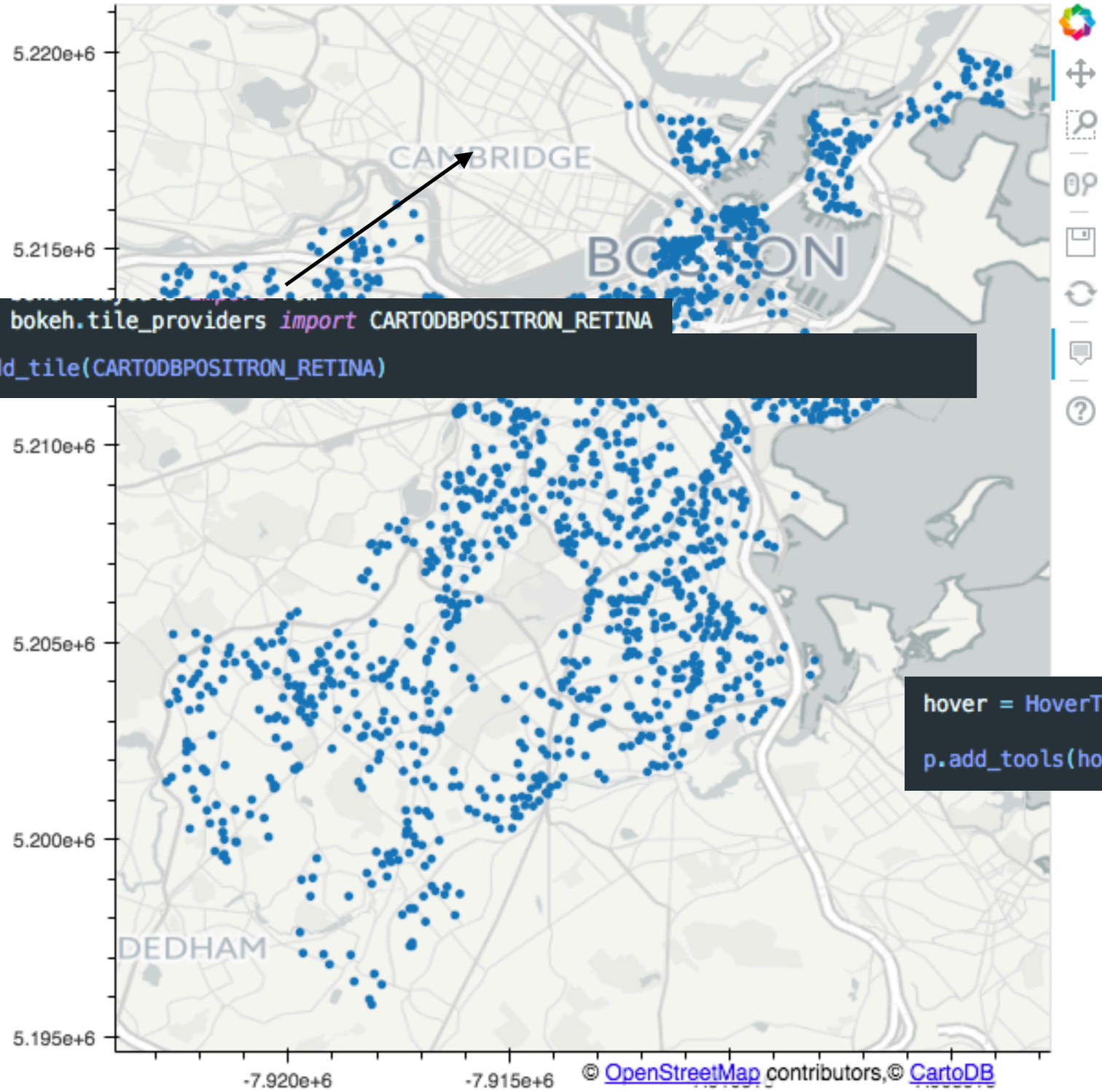
* Update the ColumnDataSource withe the new filtered DataFrame

* Update the plot when you change the value from the menu

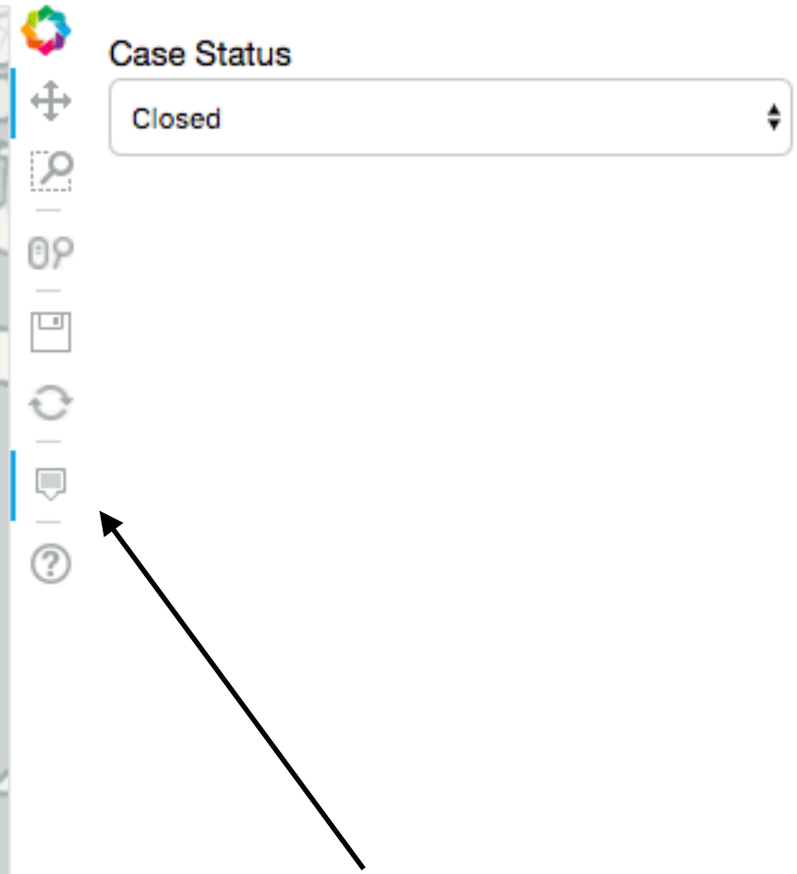

5.220e+6
5.215e+6
5.210e+6
5.205e+6
5.200e+6
5.195e+6
-7.920e+6
-7.915e+6
-7.910e+6
-7.905e+6
Case Status
Closed

311-dashboard-101.py × 311-dashboard-102.py × 311-dashboard-103.py ×

```python
from bokeh.io import curdoc
from bokeh.plotting import figure
from bokeh.models import ColumnDataSource, Select, HoverTool
from bokeh.layouts import row
from bokeh.tile_providers import CARTODBPOSITRON_RETINA
import pandas as pd

# read the dataset
service_requests = pd.read_csv('../datasets/service-requests.csv', index_col=0)

# convert to CDS format
sr_cds = ColumnDataSource(data=dict(
    x=[],
    y=[],
    source=[]))

# create the blank figure
p = figure(webgl=True)

# create circle glyphs with latitude and longitude coordinates as x and y
p.circle('x', 'y', source=sr_cds)

p.add_tile(CARTODBPOSITRON_RETINA)

hover = HoverTool(tooltips=[('Source', '@source')])

p.add_tools(hover)

# create a menu widget to be added to the curdoc
menu = Select(options=['Closed', 'Open'], value='Closed', title='Case Status')

def select_data():
    menu_val = menu.value
    temp_df = service_requests
    filtered_df = temp_df[temp_df.CASE_STATUS.str.contains(menu_val) == True]
    return filtered_df

def update_plot():
    df = select_data()
    sr_cds.data = dict(
        x = df['wm_x'],
        y = df['wm_y'],
        source = df['Source'])

menu.on_change('value', lambda attr, old, new: update_plot())

# create a layout with one row
layout = row(p, menu)

update_plot() # run once to get initial data

# add the layout to the current document
curdoc().add_root(layout)
```



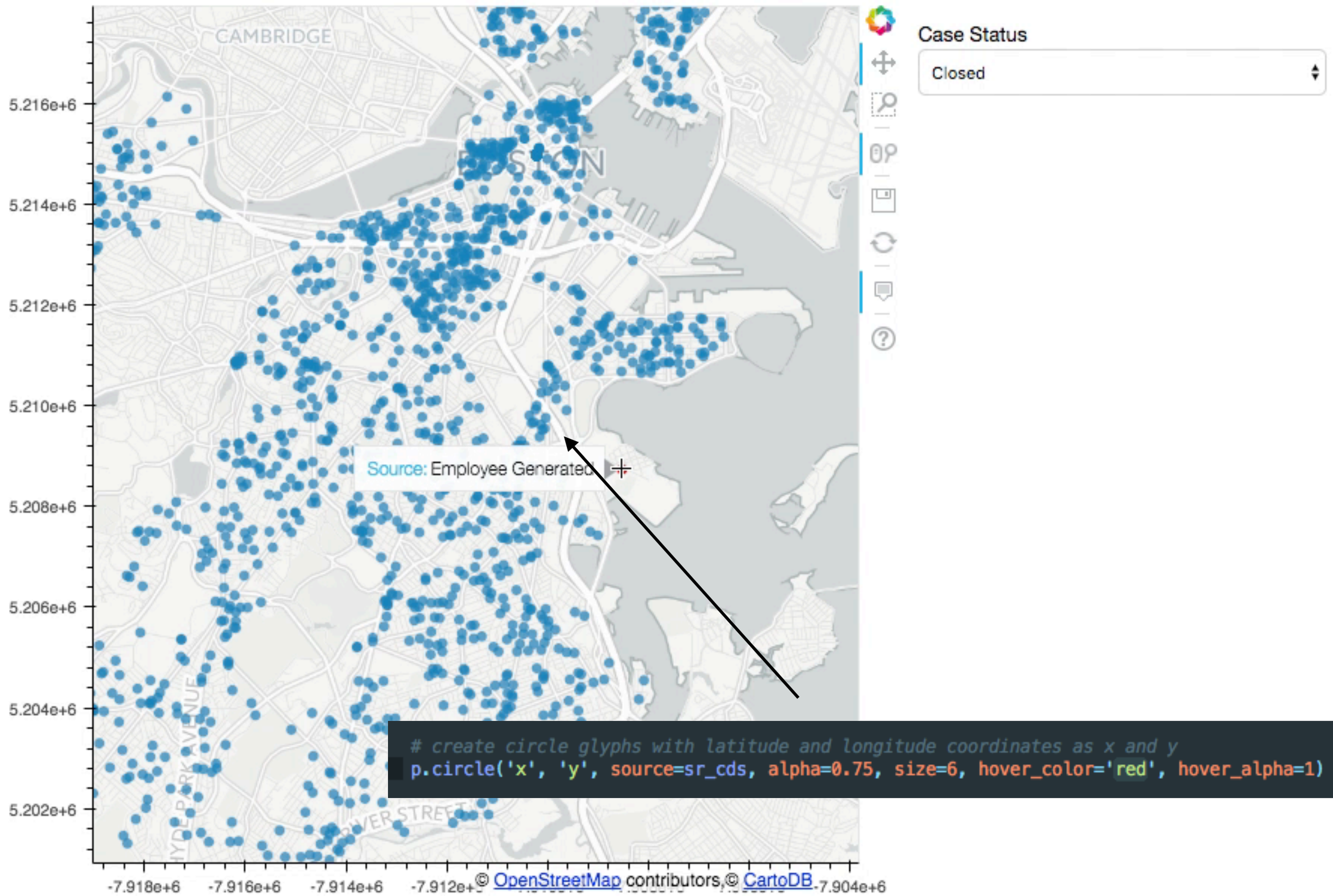

```
from bokeh.tile_providers import CARTODBPOSITRON_RETINA
p.add_tile(CARTODBPOSITRON_RETINA)
```




```
hover = HoverTool(tooltips=[('Source', '@source')])
p.add_tools(hover)
```

311-dashboard-101.py × 311-dashboard-102.py × 311-dashboard-103.py × 311-dashboard

```python
from bokeh.io import curdoc
from bokeh.plotting import figure
from bokeh.models import ColumnDataSource, Select, HoverTool
from bokeh.layouts import row
from bokeh.tile_providers import CARTODBPOSITRON_RETINA
import pandas as pd

# read the dataset
service_requests = pd.read_csv('../datasets/service-requests.csv', index_col=0)

# convert to CDS format
sr_cds = ColumnDataSource(data=dict(
    x=[],
    y=[],
    source=[]))

# create the blank figure
p = figure(webgl=True)

# create circle glyphs with latitude and longitude coordinates as x and y
p.circle('x', 'y', source=sr_cds, alpha=0.75, size=6, hover_color='red', hover_alpha=1)

# cr = p.circle('x', 'y', source=sr_cds, hover_color="firebrick")

p.add_tile(CARTODBPOSITRON_RETINA)

hover = HoverTool(tooltips=[('Source', '@source')])

p.add_tools(hover)

# create a menu widget to be added to the curdoc
menu = Select(options=['Closed', 'Open'], value='Closed', title='Case Status')

def select_data():
    menu_val = menu.value
    temp_df = service_requests
    filtered_df = temp_df[temp_df.CASE_STATUS.str.contains(menu_val) == True]
    return filtered_df

def update_plot():
    df = select_data()
    sr_cds.data = dict(
        x = df['wm_x'],
        y = df['wm_y'],
        source = df['Source'])
```


Case Status
Closed
CAMBRIDGE
BOSTON
Source: Employee Generated
5.216e+6
5.214e+6
5.212e+6
5.210e+6
5.208e+6
5.206e+6
5.204e+6
5.202e+6
-7.918e+6
-7.916e+6
-7.914e+6
-7.912e+6
-7.904e+6
© OpenStreetMap contributors,© CartoDB
# create circle glyphs with latitude and longitude coordinates as x and y
p.circle('x', 'y', source=sr_cds, alpha=0.75, size=6, hover_color='red', hover_alpha=1)

Dashboard 1 - Workshop