03_01_pyplot_end

January 5, 2020

[1]:
```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
```

[2]:
```
df = pd.read_csv("../../inputs/Environmental_Data_Deep_Moor_2015.csv")
```

[3]:
```
?plt.plot
```

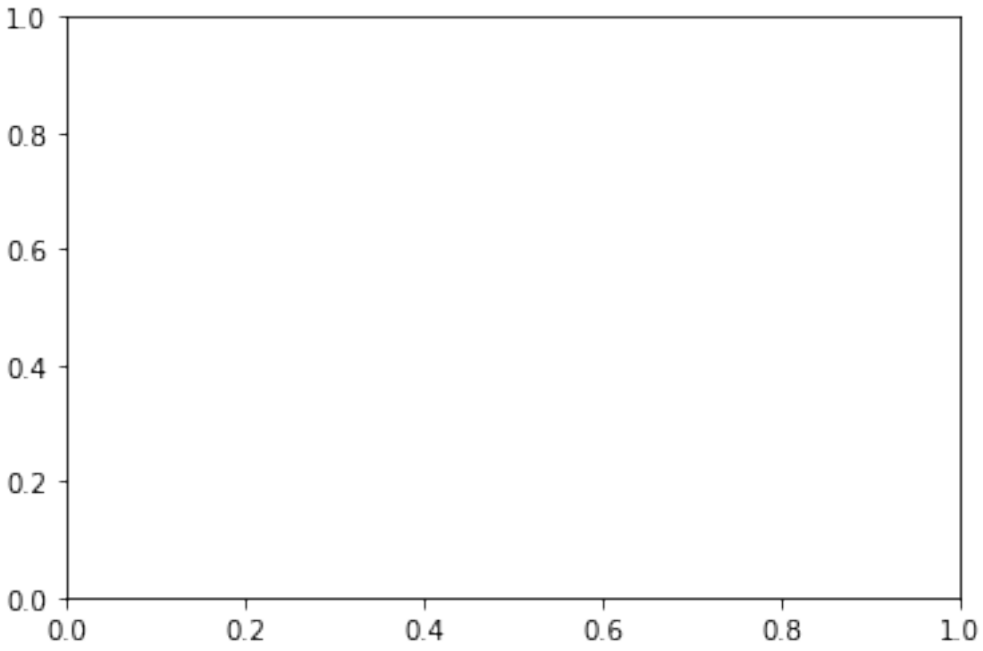
[4]:
```
# %load hours_dict.py

def hours_dict(date):
    day = df[df['date']==date]
    hours = [time.split(':')[0] for time in day['time']]
    hours_dict = {i:hours.count(str(i)) for i in np.arange(24)}
    return hours_dict
```

[5]:
```
feb15 = hours_dict('2015_02_15')
plt.plot(feb15.keys(),feb15.values())
plt.show()
```

```
␣
↪---------------------------------------------------------------------------

TypeError                                 Traceback (most recent call␣
↪last)

<ipython-input-5-df3f003c0ae7> in <module>
  1 feb15 = hours_dict('2015_02_15')
----> 2 plt.plot(feb15.keys(),feb15.values())
  3 plt.show()


~/anaconda3/lib/python3.7/site-packages/matplotlib/pyplot.py in␣
↪plot(scalex, scaley, data, *args, **kwargs)
2793     return gca().plot(
2794         *args, scalex=scalex, scaley=scaley, **({"data": data} if␣
↪data
```

```
   -> 2795         is not None else {}), **kwargs)
      2796
      2797


       ~/anaconda3/lib/python3.7/site-packages/matplotlib/axes/_axes.py in␣
↪plot(self, scalex, scaley, data, *args, **kwargs)
      1666         lines = [*self._get_lines(*args, data=data, **kwargs)]
      1667         for line in lines:
   -> 1668             self.add_line(line)
      1669         self.autoscale_view(scalex=scalex, scaley=scaley)
      1670         return lines


       ~/anaconda3/lib/python3.7/site-packages/matplotlib/axes/_base.py in␣
↪add_line(self, line)
      1900             line.set_clip_path(self.patch)
      1901
   -> 1902         self._update_line_limits(line)
      1903         if not line.get_label():
      1904             line.set_label('_line%d' % len(self.lines))


       ~/anaconda3/lib/python3.7/site-packages/matplotlib/axes/_base.py in␣
↪_update_line_limits(self, line)
      1922         Figures out the data limit of the given line, updating self.
↪dataLim.
      1923         """
   -> 1924         path = line.get_path()
      1925         if path.vertices.size == 0:
      1926             return


       ~/anaconda3/lib/python3.7/site-packages/matplotlib/lines.py in␣
↪get_path(self)
      1025         """
      1026         if self._invalidy or self._invalidx:
   -> 1027             self.recache()
      1028         return self._path
      1029


       ~/anaconda3/lib/python3.7/site-packages/matplotlib/lines.py in␣
↪recache(self, always)
       668         if always or self._invalidx:
       669             xconv = self.convert_xunits(self._xorig)
   --> 670             x = _to_unmasked_float_array(xconv).ravel()
```

```
        671         else:
        672             x = self._x


        ~/anaconda3/lib/python3.7/site-packages/matplotlib/cbook/__init__.py in␣
↪_to_unmasked_float_array(x)
       1388         return np.ma.asarray(x, float).filled(np.nan)
       1389     else:
    -> 1390         return np.asarray(x, float)
       1391
       1392


        ~/anaconda3/lib/python3.7/site-packages/numpy/core/_asarray.py in␣
↪asarray(a, dtype, order)
         83
         84     """
    ---> 85     return array(a, dtype, copy=False, order=order)
         86
         87


        TypeError: float() argument must be a string or a number, not 'dict_keys'
```



[ ]:
```
feb15 = hours_dict('2015_02_15')
feb16 = hours_dict('2015_02_16')
plt.plot(feb15.keys(),feb15.values())
plt.plot(feb16.keys(),feb16.values())
plt.show()
```

[ ]:
```
feb15 = hours_dict('2015_02_15')
feb16 = hours_dict('2015_02_16')
plt.plot(feb15.keys(),feb15.values())
plt.plot(feb16.keys(),feb16.values())
plt.xticks(np.arange(0,24,6))
plt.xlim(-3,26)
plt.ylim(-3,28)
plt.xlabel('Hour of the day')
plt.ylabel('Measurements Taken')
plt.title('Measurements By Hour')
plt.show()
```

[ ]:
```

```