

```

1 import matplotlib.pyplot as plt
2 import pandas as pd
3 from datetime import datetime
4
5 '''
6 This is an example to generate the Philadelphia, PA weather chart.
7
8 If you want to make the chart for another city, you will have to modify
9 this code slightly to read that city's data in, change the title, and
10 likely change the y-axis of the chart to fit your city's temperature range.
11
12 I also use a custom matplotlib style as the basis for these charts, which you
13 can find here:
14 https://gist.githubusercontent.com/rhiever/d0a7332fe0beebfdc3d5/raw/223d70799b48131d5ce2723cd5784f39d7a3a653/tableau10.mplstyle
15 '''
16 weather_data = pd.read_csv('KPHL.csv', parse_dates=['date'])
17 print(weather_data.describe())
18
19 # Generate a bunch of histograms of the data to make sure that all of the data
20 # is in an expected range.
21 with
plt.style.context('https://gist.githubusercontent.com/rhiever/d0a7332fe0beebfdc3d5/raw/223d70799b48131d5ce2723cd5784f39d7a3a653/tableau10.mplstyle'):
22     for column in weather_data.columns:
23         if column in ['date']:
24             continue
25         plt.figure()
26         plt.hist(weather_data[column].values)
27         plt.title(column)
28         plt.savefig('{} .png'.format(column))
29
30     # Make sure we're only plotting temperatures for July 2014 - June 2015
31     weather_data_subset = weather_data[weather_data['date'] >= datetime(year=2014,
month=7, day=1)]
32     weather_data_subset = weather_data_subset[weather_data_subset['date'] <
datetime(year=2015, month=7, day=1)].copy()
33     weather_data_subset['day_order'] = range(len(weather_data_subset))
34
35     day_order = weather_data_subset['day_order']
36     record_max_temps = weather_data_subset['record_max_temp'].values
37     record_min_temps = weather_data_subset['record_min_temp'].values
38     average_max_temps = weather_data_subset['average_max_temp'].values
39     average_min_temps = weather_data_subset['average_min_temp'].values
40     actual_max_temps = weather_data_subset['actual_max_temp'].values
41     actual_min_temps = weather_data_subset['actual_min_temp'].values
42
43     fig, ax1 = plt.subplots(figsize=(15, 7))
44
45     # Create the bars showing all-time record highs and lows
46     plt.bar(day_order, record_max_temps - record_min_temps, bottom=record_min_temps,
47            edgecolor='none', color='#C3BBA4', width=1)
48
49     # Create the bars showing average highs and lows
50     plt.bar(day_order, average_max_temps - average_min_temps, bottom=average_min_temps,

```

```

51         edgecolor='none', color='#9A9180', width=1)
52
53     # Create the bars showing this year's highs and lows
54     plt.bar(day_order, actual_max_temps - actual_min_temps, bottom=actual_min_temps,
55             edgecolor='black', linewidth=0.5, color='#5A3B49', width=1)
56
57     new_max_records = weather_data_subset[weather_data_subset.record_max_temp <=
58     weather_data_subset.actual_max_temp]
59     new_min_records = weather_data_subset[weather_data_subset.record_min_temp >=
60     weather_data_subset.actual_min_temp]
61
62     # Create the dots marking record highs and lows for the year
63     plt.scatter(new_max_records['day_order'].values + 0.5,
64                 new_max_records['actual_max_temp'].values + 1.25,
65                 s=15, zorder=10, color='#d62728', alpha=0.75, linewidth=0)
66
67     plt.scatter(new_min_records['day_order'].values + 0.5,
68                 new_min_records['actual_min_temp'].values - 1.25,
69                 s=15, zorder=10, color='#1f77b4', alpha=0.75, linewidth=0)
70
71     plt.ylim(-15, 111)
72     plt.xlim(-5, 370)
73
74     plt.yticks(range(-10, 111, 10), [r'{}$^\circ$'.format(x)
75                                     for x in range(-10, 111, 10)], fontsize=10)
76     plt.ylabel(r'Temperature ($^\circ$F)', fontsize=12)
77
78     month_beginning_df = weather_data_subset[weather_data_subset['date']].apply(lambda x:
79     True if x.day == 1 else False)]
80     month_beginning_indeces = list(month_beginning_df['day_order'].values)
81     month_beginning_names = list(month_beginning_df['date'].apply(lambda x:
82     x.strftime("%B")).values)
83     month_beginning_names[0] += '\n\14'
84     month_beginning_names[6] += '\n\15'
85
86     # Add the Last month Label manually
87     month_beginning_indeces += [weather_data_subset['day_order'].values[-1]]
88     month_beginning_names += ['July']
89
90     plt.xticks(month_beginning_indeces,
91                 month_beginning_names,
92                 fontsize=10)
93
94     ax2 = ax1.twinx()
95     plt.xticks(month_beginning_indeces,
96                 month_beginning_names,
97                 fontsize=10)
98
99     plt.xlim(-5, 370)
100    plt.grid(False)
101
102    ax3 = ax1.twinx()
103    plt.yticks(range(-10, 111, 10), [r'{}$^\circ$'.format(x)
104                                    for x in range(-10, 111, 10)], fontsize=10)
105    plt.ylim(-15, 111)
106    plt.grid(False)

```

```
104 plt.title('Philadelphia, PA\'s weather, July 2014 - June 2015\\n\\n', fontsize=20)
105
106 plt.savefig('philadelphia-weather-july14-june15.png')
107
```