Plotting Temperature data - 4 European Regions

Here we take temperature data observed in Western Europe during the period of 2005-2015. We examine how many days during that period had their daily max and min temperatures exceeded in 2015.

Using `%matplotlib` rather than `%matplotlib notebook` because its better to zoom in on the figures

Basic Notebook Setup

```
%matplotlib
```

In [1]:

```
%matplotlib notebook
```

In [2]:

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import mplleaflet
```

In [3]:

```
RED = sns.xkcd_rgb["pale red"]
#RED = '#e74c3c' #sns.xkcd_rgb["pale red"]
```

Orientation

Below we see the geographic location of the data colection points

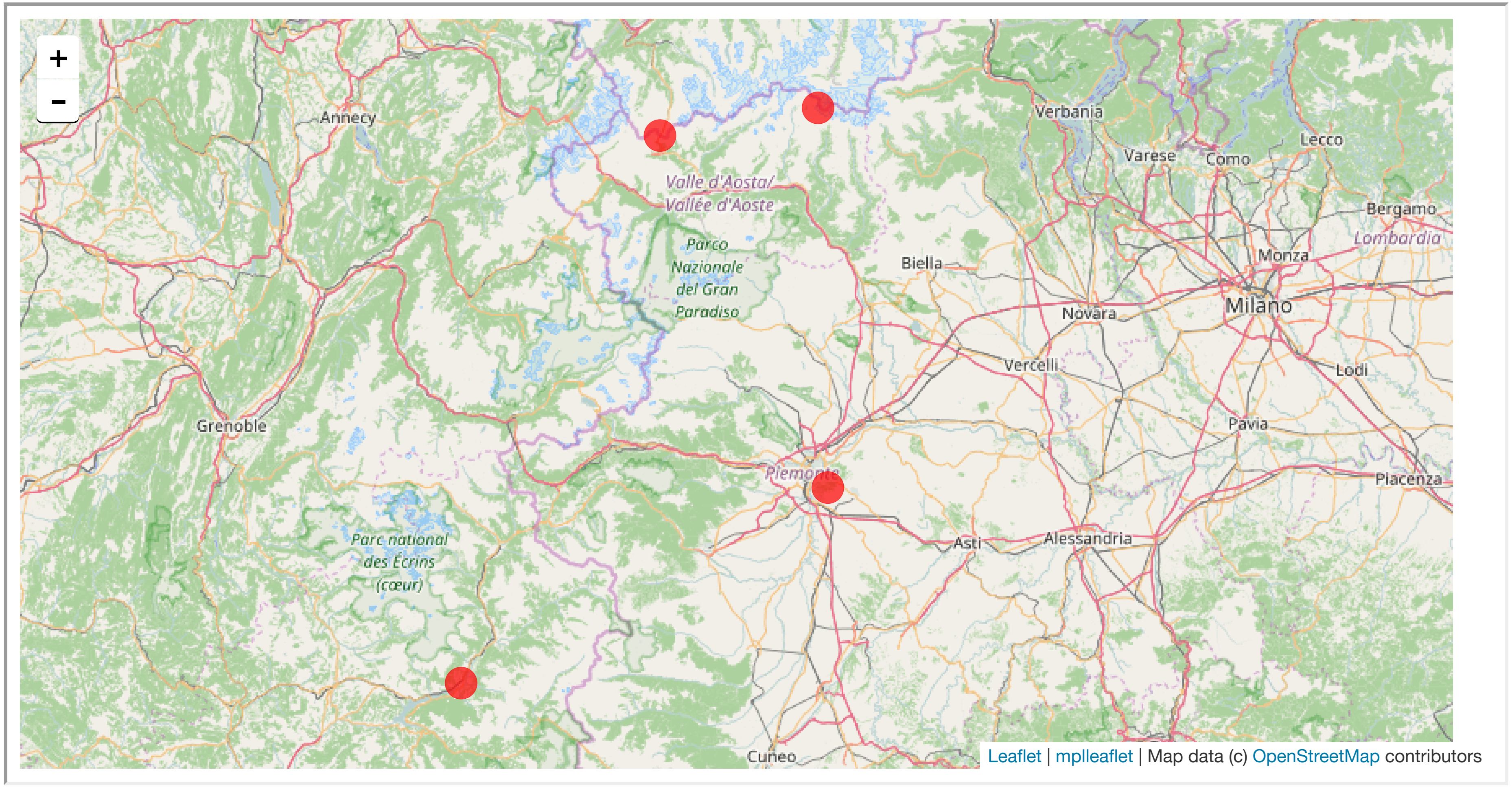
In [4]:

```
binsize = 200
hashid = 'f730b3be5c0ea89e0421a2cf1f9c2fd2c1bebafae51d00fa2775d250'
reg_df = pd.read_csv('BinSize_d{}.csv'.format(binsize))
reg_df.set_index('ID', inplace=True)
```

In [5]:

```
def leaflet_plot_stations(df):

    station_locations_by_hash = df[df['hash'] == hashid]
    lons = station_locations_by_hash['LONGITUDE'].tolist()
    lats = station_locations_by_hash['LATITUDE'].tolist()

    plt.figure(figsize=(8,8))

    plt.scatter(lons, lats, c='r', alpha=0.7, s=200)

    return mplleaflet.display()

leaflet_plot_stations(reg_df)
```

```
Out[5]:
```



Process Data

Read and process the temperature data

In [6]:

```
# For online version
#df = pd.read_csv('data/C2A2_data/BinnedCsvs_d200/f730b3be5c0ea89e0421a2cf1f9c2fd2c1bebafae51d00fa2775d250.csv')

# Offline version
df = pd.read_csv('f730b3be5c0ea89e0421a2cf1f9c2fd2c1bebafae51d00fa2775d250.csv')
```

In [7]:

```
# Take a look
df.head()
```

Out[7]:

	ID	Date	Element	Data_Value
0	FRM00007591	2009-06-20	TMIN	97
1	FRM00007591	2006-06-11	TMIN	110
2	FRM00007591	2010-06-12	TMIN	146
3	FRM00007591	2013-02-27	TMAX	98
4	ITM00016061	2011-04-24	TMAX	184

In [8]:

```
# Convert temp to Celcius
df['Data_Value'] = df['Data_Value'] / 10

# Format date
df.Date = pd.to_datetime(df.Date)

# Set up the table better
df = df.pivot_table(values='Data_Value', index=['Date', 'ID'], columns='Element')
df = df.reset_index().set_index('Date')

# Remove leap days
leap_days = df[(df.index.month == 2) & (df.index.day == 29)].index
df.drop(leap_days, inplace=True)

# Rename Columns
df.columns = ['Region', 'Max', 'Min']
```

In [9]:

```
df.head()
```

Out[9]:

	Region	Max	Min
Date			
2005-01-01	SZ000006717	-3.3	-7.8
2005-01-02	SZ000006717	-2.2	-11.3
2005-01-03	SZ000006717	-7.2	-12.6
2005-01-04	SZ000006717	0.3	-7.4
2005-01-05	SZ000006717	0.2	-6.3

Get Region Data

In [10]:

```
# Split into Regions
regions = df.Region.unique()
regions
```

Out[10]:

```
array(['SZ000006717', 'FRM00007591', 'ITM00016052', 'ITM00016061'], dtype=object)
```

In [12]:

```python
region_data = {}

for region in regions:

    ## 1. Prepare the data

    # Get region subset
    region_temps = df[df.Region == region].drop('Region', axis=1)

    # Foward fill Nan values
    region_temps.fillna(method='ffill', inplace=True)

    # Create columns to compare yearly values with
    # Could be better acheived with day_of_year attribute
    region_temps['Month'] = region_temps.index.month
    region_temps['DoM'] = region_temps.index.day

    # Divide pre and post 2015 Data
    region_pre_2015 = region_temps.loc[:'2014']
    region_2015 = region_temps.loc['2015']

    # Merged Data
    region_df = pd.merge(region_pre_2015.reset_index(), region_2015,
             how='left',
             left_on=['Month', 'DoM'],
             right_on=['Month', 'DoM'],
             suffixes=['','_2015']
            ).set_index('Date')

    ## 2. We want to especially note the days of the year in which
    ## the temperature was exceeded in 2015.

    # Set lower values for 2015 to null for Max's
    region_df.loc[(region_df.Max > region_df.Max_2015), 'Max_2015'] = np.NaN
    # Set lower values for 2015 to null for Min's
    region_df.loc[(region_df.Min > region_df.Min_2015), 'Min_2015'] = np.NaN

    ## 3. Save data
    reg_name, reg_elev = reg_df.loc[region, ['NAME', 'ELEVATION']].values
    region_data["{} ({}m)".format(reg_name, reg_elev)] = region_df
```

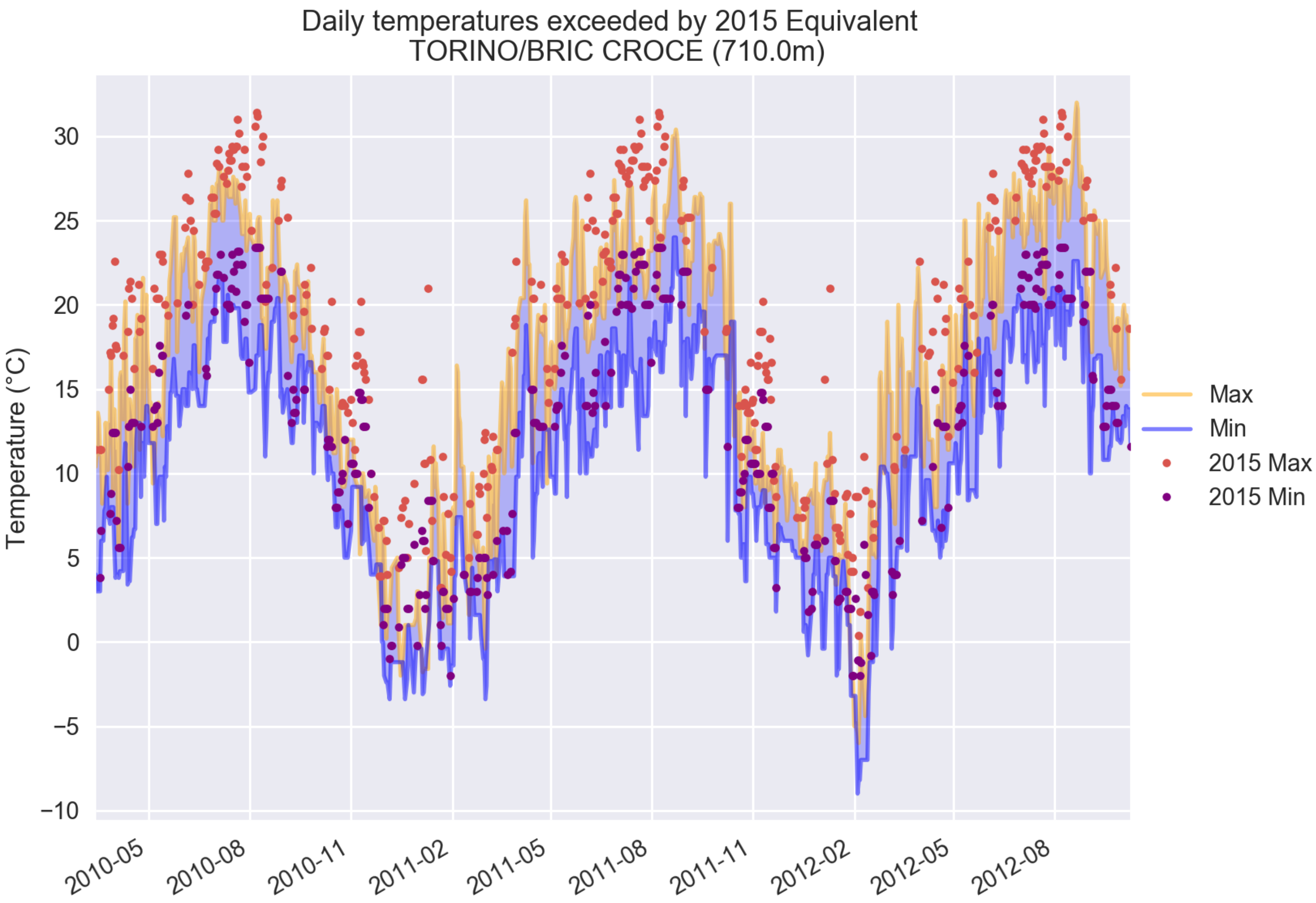
Set up Figure

Create 4 seperate figures.

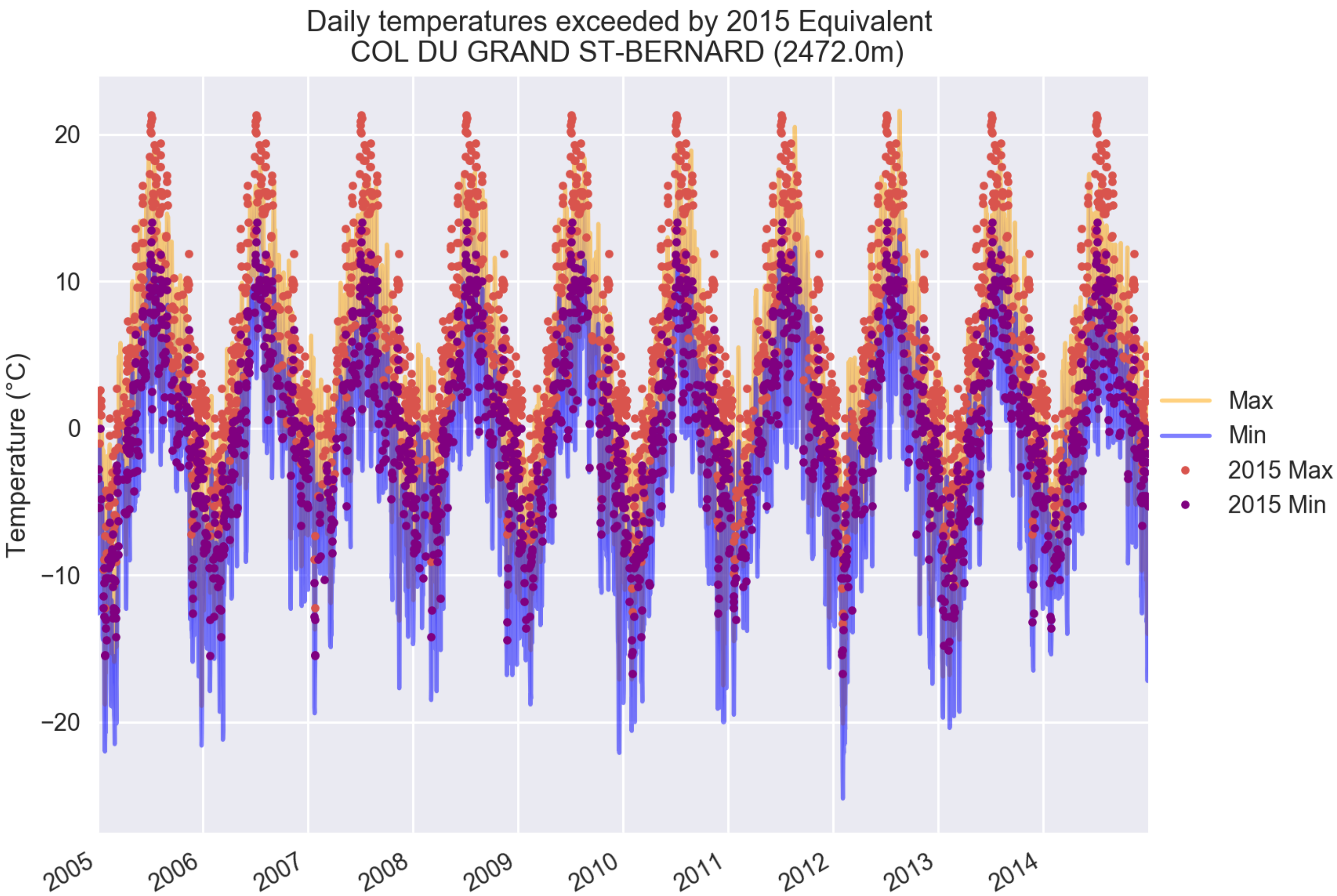
With simple modification we could create a figure of 2x2 or 1x4 subplots but seperate figures allow us to better zoom and pan

In [16]:

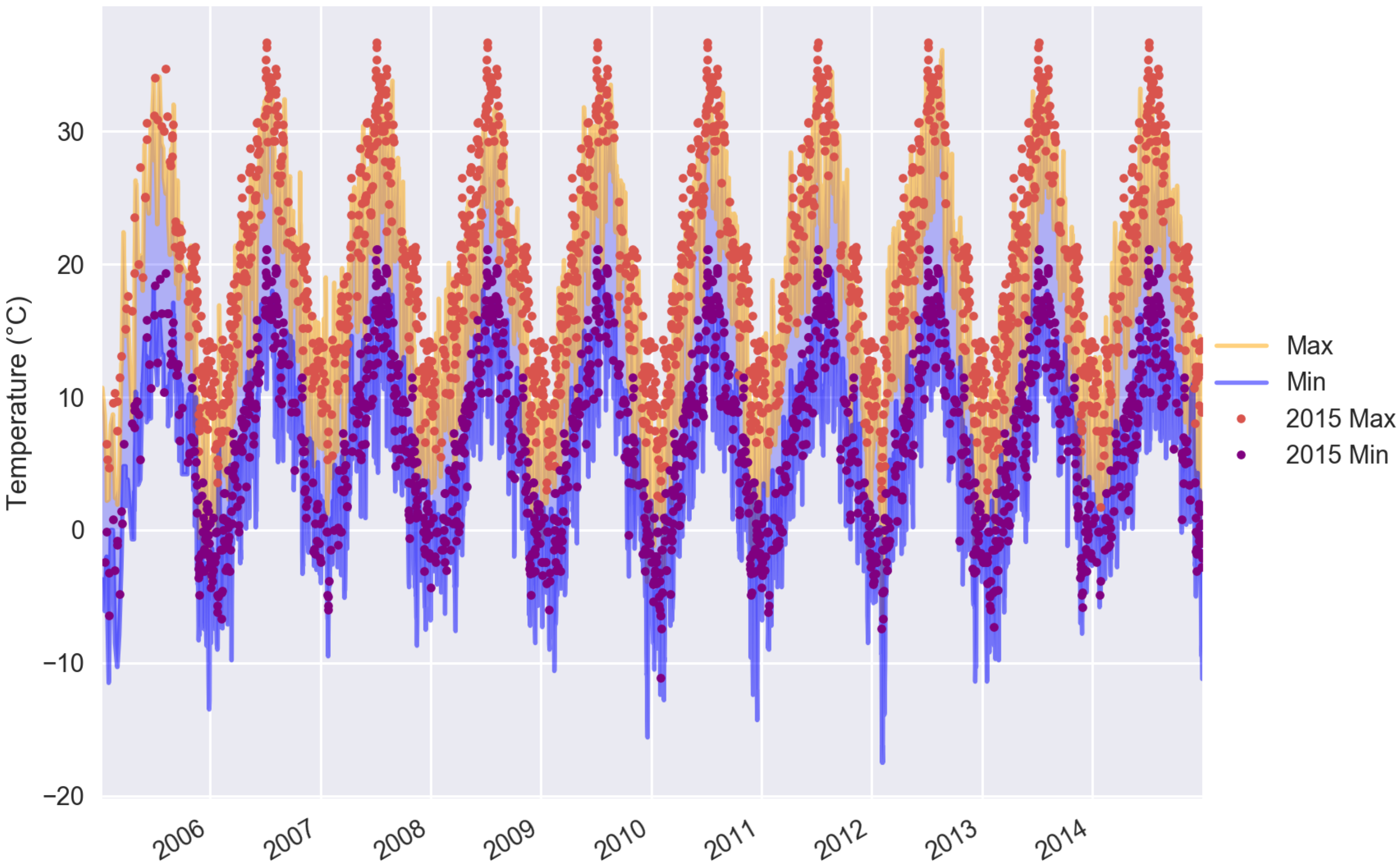
```
for region_name, region_df in region_data.items():

    ## 1. Plot the data
    # Plot daily temperatures
    fig, axes = plt.subplots()
    axes = region_df[['Max', 'Min']].plot.line(color=['orange', 'blue'], alpha=0.5, ax=axes)

    # Fill between Max and Min
    axes.fill_between(region_df.index,
                     region_df.Min, region_df.Max,
                     facecolor='blue',
                     alpha=0.25
                    )

    # Plot corresponding 2015 Values
    region_df[['Max_2015','Min_2015']].plot(style='.', color=[RED, 'purple'], ax=axes)

    # Cleanup Legend
    handles, labels = axes.get_legend_handles_labels()
    labels = ['Max', 'Min', '2015 Max', '2015 Min']
    axes.legend(handles, labels)

    # Labels and Titles
    axes.set_xlabel('')
    axes.set_ylabel('Temperature (°C)')
    # Update the Title and labels
    axes.set_title('Daily temperatures exceeded by 2015 Equivalent \n {}'.format(region_name))

    handles, labels = axes.get_legend_handles_labels()

    labels = ['Max', 'Min', '2015 Max', '2015 Min']
    axes.legend_.remove()

    # shift subplots down:
    fig.tight_layout()
    fig.subplots_adjust(right=0.85)
    leg = fig.legend(handles, labels, loc="center right")
```



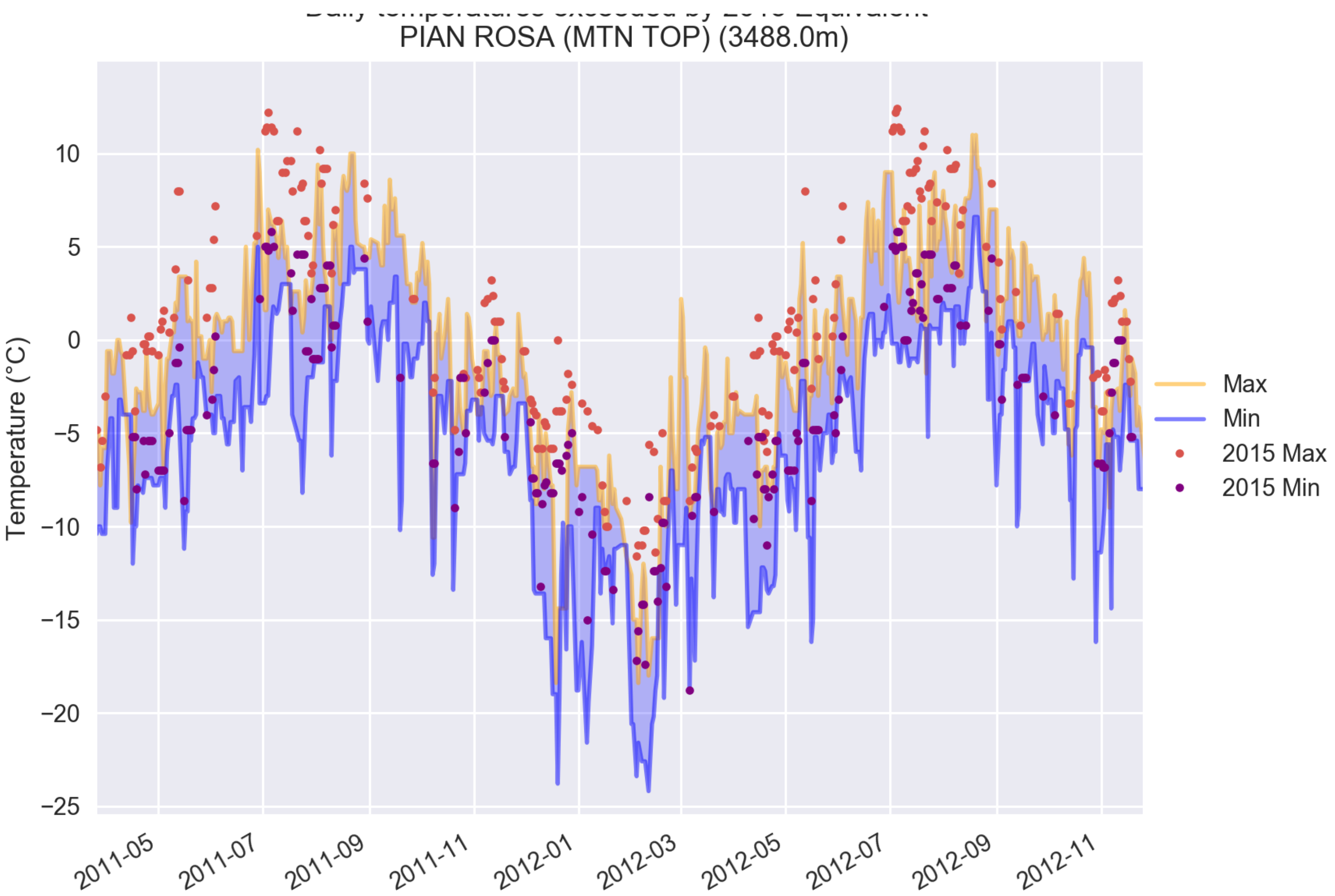
Daily temperatures exceeded by 2015 Equivalent
TORINO/BRIC CROCE (710.0m)
Temperature (°C)
30
25
20
15
10
5
0
−5
−10
2010-05
2010-08
2010-11
2011-02
2011-05
2011-08
2011-11
2012-02
2012-05
2012-08
Max
Min
2015 Max
2015 Min


Daily temperatures exceeded by 2015 Equivalent
COL DU GRAND ST-BERNARD (2472.0m)
Temperature (°C)
20
10
0
−10
−20
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
Max
Min
2015 Max
2015 Min

Daily temperatures exceeded by 2015 Equivalent
EMBRUN (876.0m)



Daily temperatures exceeded by 2015 Equivalent


Daily temperatures exceeded by 2015 Equivalent
PIAN ROSA (MTN TOP) (3488.0m)
Temperature (°C)
10
5
0
−5
−10
−15
−20
−25
2011-05
2011-07
2011-09
2011-11
2012-01
2012-03
2012-05
2012-07
2012-09
2012-11
Max
Min
2015 Max
2015 Min

In [ ]: