GHCNpy: Using Python to Analyze and Visualize Daily Weather Station Data in Near Real Time

Jared Rennie
Cooperative Institute for Climate and Satellites – North Carolina
National Centers For Environmental Information (NCEI)
Asheville, NC

Sam Lillo
University of Oklahoma
Norman, OK

Python Symposium
AMS Annual Meeting
January 12th, 2016

Special Thanks
Randy Olson – FiveThirtyEight.com
Andrew Buddenburg – CICS-NC
Jim Biard – CICS-NC
HOMR Team – NOAA/NCEI and ERT

GHCN

- Global Historical Climatology Network
  - Consolidated global dataset used to monitor and assess the state of the climate
- GHCN-Daily
  - Integrated database of daily climate summaries
    - Temperature, Precipitation, Snowfall, Other Weather Data
  - 100,000 stations worldwide
  - Updates each night with new data
  - Subjected to a common suite of quality assurance

Accessing GHCN-Daily

- NCEI FTP
  - Text files (one per station), and csv files (one per year)
  - Requires knowledge of file location, formats, readmes
- NCEI “Climate Data Online” Portal
  - Mapping interface
  - Runs on Oracle Database
  - Custom Text / CSV files
- xmACIS
  - Custom Text / CSV files, Visualizations
  - US Data Only
  - “Only for NWS employees”

Accessing GHCN-Daily

	Global Data	Text Files	CSV Files	netCDF Files	Visualizations Of Data
NCEI FTP	X	X	X		
NCEI CDO	X	X	X		
xmACIS		X	X		X
GHCNpy	**X**	**X**	**X**	**X**	**X**

GHCNpy

- Python package for downloading, analyzing and visualizing data from GHCN-Daily
- Requires no knowledge about formats or location of the data
- Open Source
- Free!
- On GitLab
  - https://github.com/jjrennie/GHCNpy.git

GHCNpy

https://github.com/jjrennie/GHCNpy.git

- Utilizes Python 2.7 Anaconda Distribution
- Major packages include NumPy, matplotlib, netCDF4
- Three major programs
  - io.py
  - metadata.py
  - plotting.py

GHCNpy: io.py

https://github.com/jjrennie/GHCNpy.git

- **get_ghcnd_version()**
  - Gets version number of GHCN-D
- **get_data_station(station_id)**
  - Given station ID, get the file from FTP
- **get_data_year(year)**
  - Given year, get the yearly csv file from FTP
- **get_ghcnd_stations()**
  - Grabs the latest Station Metadata file from FTP
- **get_ghcnd_inventory()**
  - Grabs the latest Station Inventory file from FTP
- **output_to_csv(station_id)**
  - Given station Id, output 6 major elements to CSV format (one day for each line)
- **output_to_netcdf(station_id)**
  - Given station Id, output 6 major elements to CF Compliant netcdf file

GHCNpy: metadata.py

https://github.com/jjrennie/GHCNpy.git

- **get_metadata(station_id)**
  - given station id, tap into the Historical Metadata Observing Repository (HOMR) and grab station metadata:
    - Station ID, Name, Lat, Lon, Elev, State, Climate Division, County, NWS Office, COOP ID, WBAN ID
- **find_station(*args)**
  - attempts to search for stations in inventory file
  - 1 Argument: Search By Name
  - 3 Arguments: Search by lat/lon/distance limit

GHCNpy: plotting.py (Timeseries)

https://github.com/jjrennie/GHCNpy.git

- **plot_temperature(station_id,begin_date,end_date)**
  - Plots NY Times style plots for stations reporting temperature. For a given station and period, plots the following data:
    - Raw TMAX/TMIN for each day
    - Average TMAX/TMIN for each day
    - Record TMAX/TMIN for each day
    - Daily records (if Raw meets or exceeds Record)
- **plot_precipitation(station_id)**
  - Given Station ID, plots accumulated precipitation for each year in its period of record (January-December). Also highlights record max, record min, average for each day, and also current year.
- **plot_snowfall(station_id)**
  - Given Station ID, plots accumulated snowfall for each year in its period of record (October-September). Also highlights record max, record min, average for each day, and also current year.

GHCNpy: plotting.py (Spatial)

https://github.com/jjrennie/GHCNpy.git

- **plot_spatial**(year,month,day,element)
  - Plots data specifically for a given date
  - Uses GHCN-D's major elements
    - TMAX/TMIN/TAVG/PRCP/SNOW/SNWD
    - Special color maps made depending on element
  - Able to specify projection, lat/lon boxes,dpi

- **plot_spatial_derived**(year,element)
  - Special version of plot_spatial where derived temperature elements are plotted
    - Heating Degree Days, Cooling Degree days, Growing Degree Days

- **plot_spatial_freeze**(year,element)
  - Special version of plot_spatial where given minimum temperatures for a defined year, determine freeze characteristics
    - First Freeze Date, Last Freeze Date

EXAMPLES

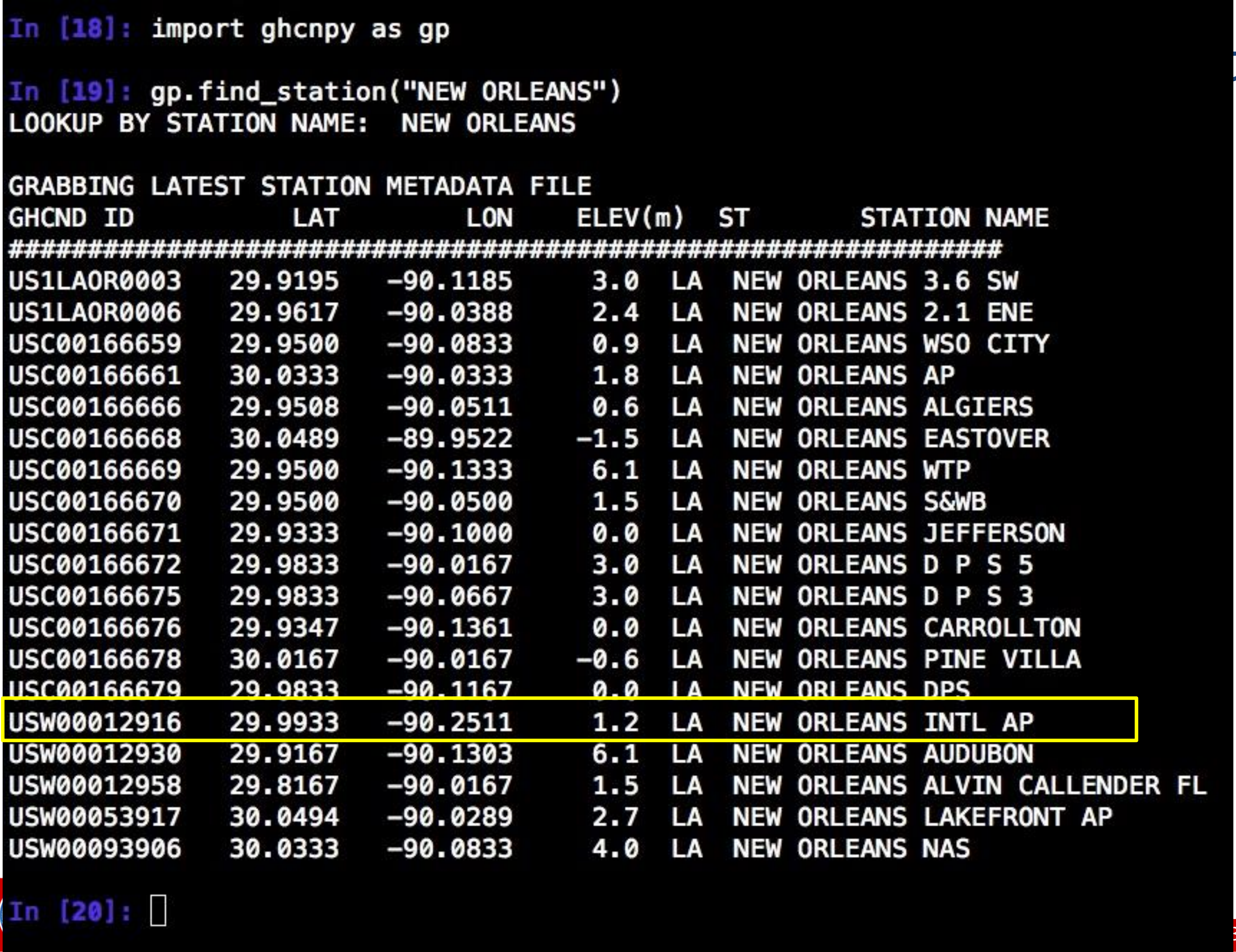
http://github.com/jjrennie/GHCNpy.git

Plot 2015 temperatures for New Orleans Airport

plot_temperature(station_id,begin_date,end_date)

- We don't know the GHCN-D ID.
  - Not a problem!

http://github.com/jjrennie/GHCNpy.git

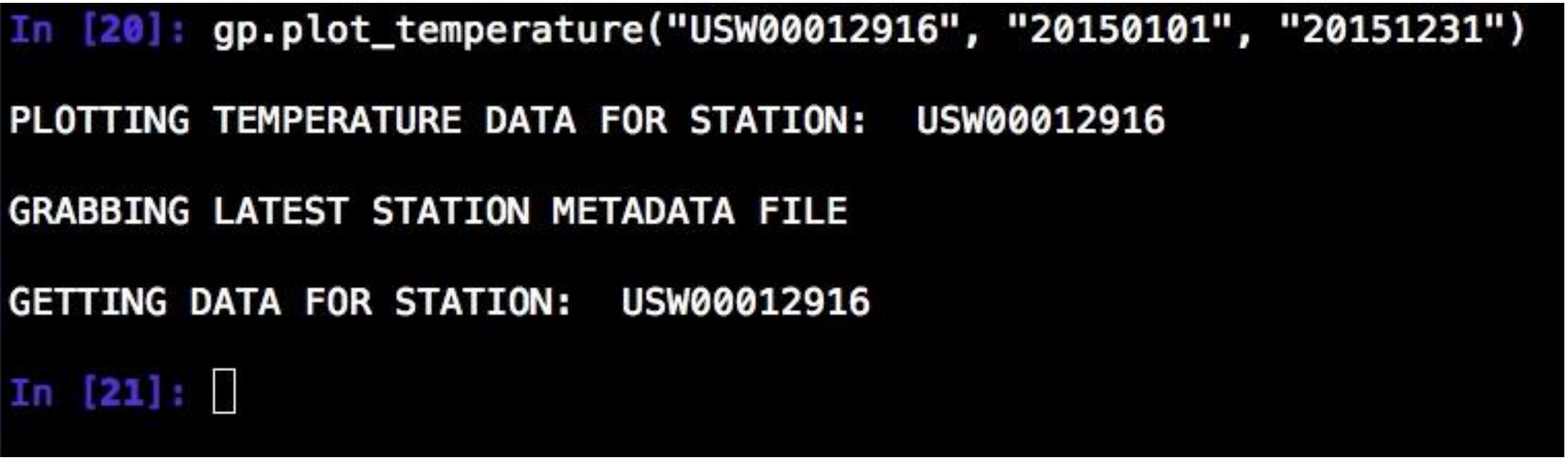


```
In [18]: import ghcnpy as gp

In [19]: gp.find_station("NEW ORLEANS")
LOOKUP BY STATION NAME:  NEW ORLEANS

GRABBING LATEST STATION METADATA FILE
GHCND ID          LAT         LON     ELEV(m)  ST       STATION NAME
###############################################################
US1LAOR0003   29.9195   -90.1185     3.0   LA   NEW ORLEANS 3.6 SW
US1LAOR0006   29.9617   -90.0388     2.4   LA   NEW ORLEANS 2.1 ENE
USC00166659   29.9500   -90.0833     0.9   LA   NEW ORLEANS WSO CITY
USC00166661   30.0333   -90.0333     1.8   LA   NEW ORLEANS AP
USC00166666   29.9508   -90.0511     0.6   LA   NEW ORLEANS ALGIERS
USC00166668   30.0489   -89.9522    -1.5   LA   NEW ORLEANS EASTOVER
USC00166669   29.9500   -90.1333     6.1   LA   NEW ORLEANS WTP
USC00166670   29.9500   -90.0500     1.5   LA   NEW ORLEANS S&WB
USC00166671   29.9333   -90.1000     0.0   LA   NEW ORLEANS JEFFERSON
USC00166672   29.9833   -90.0167     3.0   LA   NEW ORLEANS D P S 5
USC00166675   29.9833   -90.0667     3.0   LA   NEW ORLEANS D P S 3
USC00166676   29.9347   -90.1361     0.0   LA   NEW ORLEANS CARROLLTON
USC00166678   30.0167   -90.0167    -0.6   LA   NEW ORLEANS PINE VILLA
USC00166679   29.9833   -90.1167     0.0   LA   NEW ORLEANS DPS
USW00012916   29.9933   -90.2511     1.2   LA   NEW ORLEANS INTL AP
USW00012930   29.9167   -90.1303     6.1   LA   NEW ORLEANS AUDUBON
USW00012958   29.8167   -90.0167     1.5   LA   NEW ORLEANS ALVIN CALLENDER FL
USW00053917   30.0494   -90.0289     2.7   LA   NEW ORLEANS LAKEFRONT AP
USW00093906   30.0333   -90.0833     4.0   LA   NEW ORLEANS NAS

In [20]: 
```

Plot 2015 temperatures for New Orleans Airport

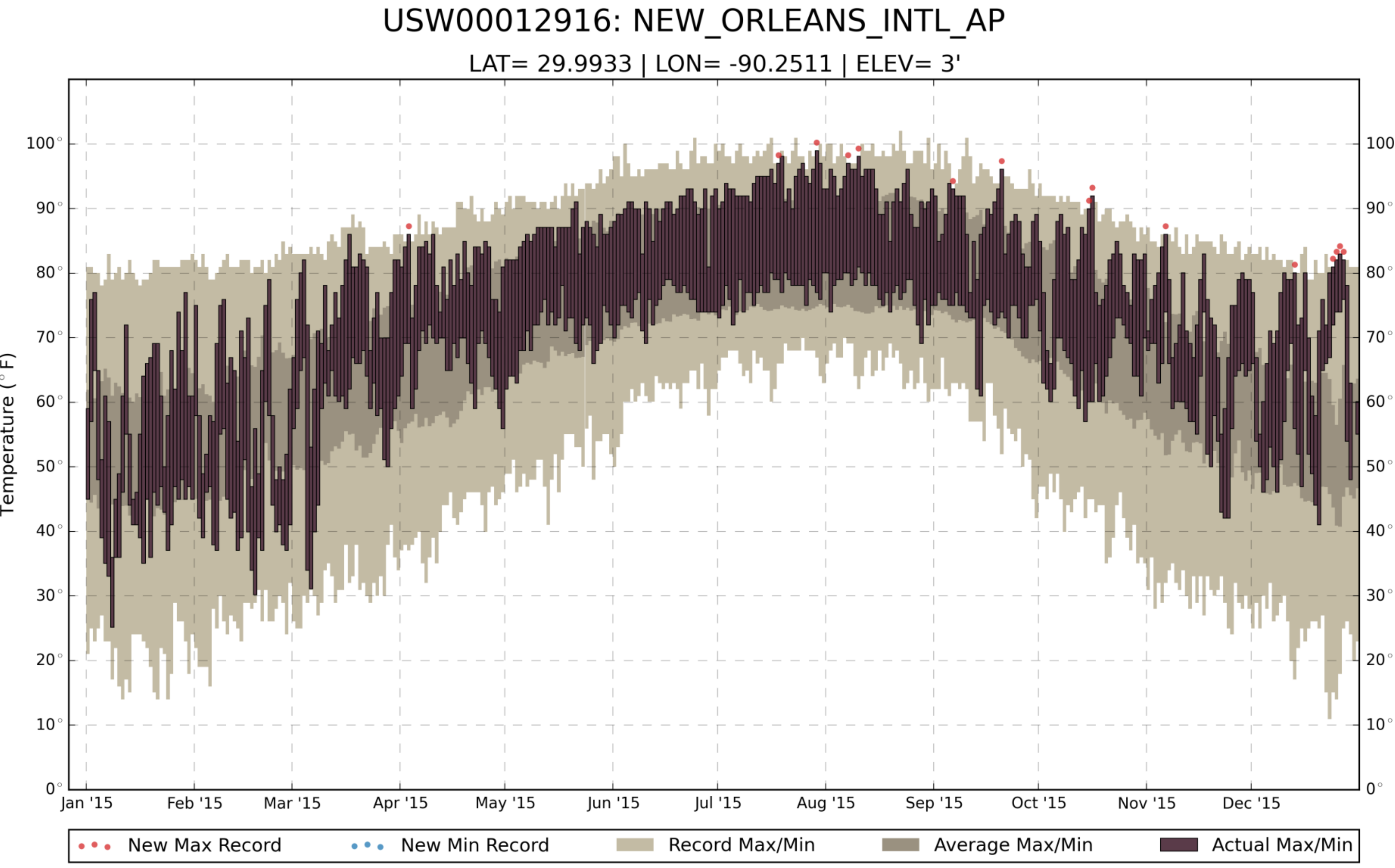
plot_temperature(station_id,begin_date,end_date)

- Now we have ID and a POR



```
In [20]: gp.plot_temperature("USW00012916", "20150101", "20151231")

PLOTTING TEMPERATURE DATA FOR STATION:  USW00012916

GRABBING LATEST STATION METADATA FILE

GETTING DATA FOR STATION:  USW00012916

In [21]: 
```

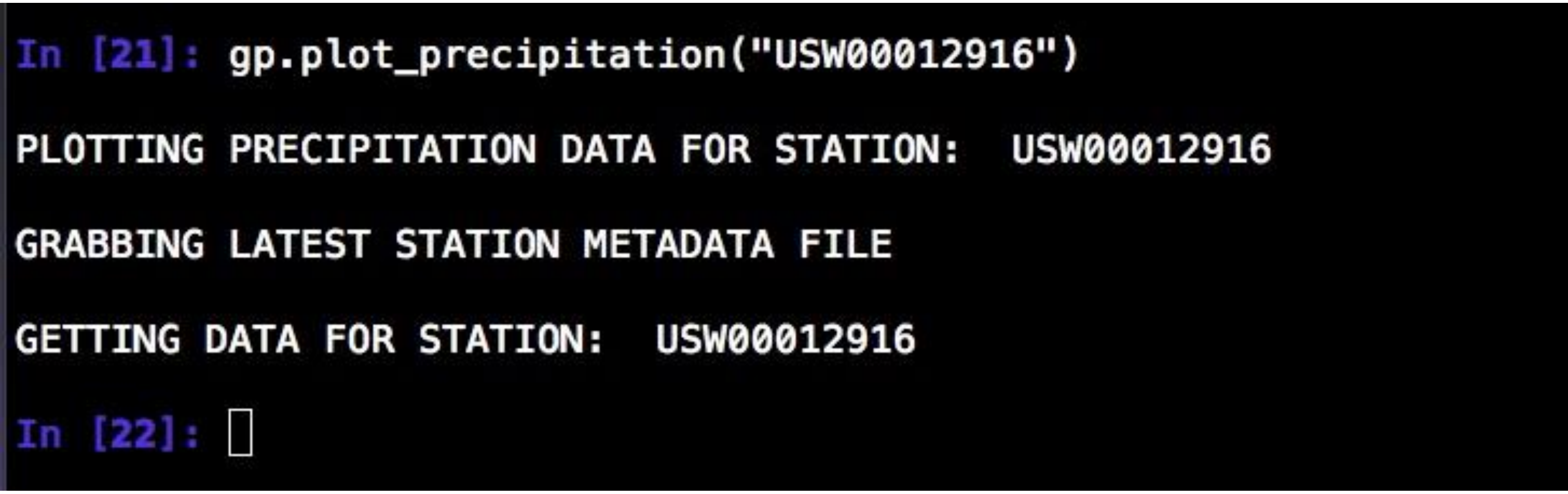
http://github.com/jjrennie/GHCNpy.git


USW00012916: NEW_ORLEANS_INTL_AP
LAT= 29.9933 | LON= -90.2511 | ELEV= 3'
Temperature (°F)
0°
10°
20°
30°
40°
50°
60°
70°
80°
90°
100°
Jan '15
Feb '15
Mar '15
Apr '15
May '15
Jun '15
Jul '15
Aug '15
Sep '15
Oct '15
Nov '15
Dec '15
New Max Record
New Min Record
Record Max/Min
Average Max/Min
Actual Max/Min

Get Accumulated Precipitation for Same Station

plot_precipitation(station_id)

- Already have station

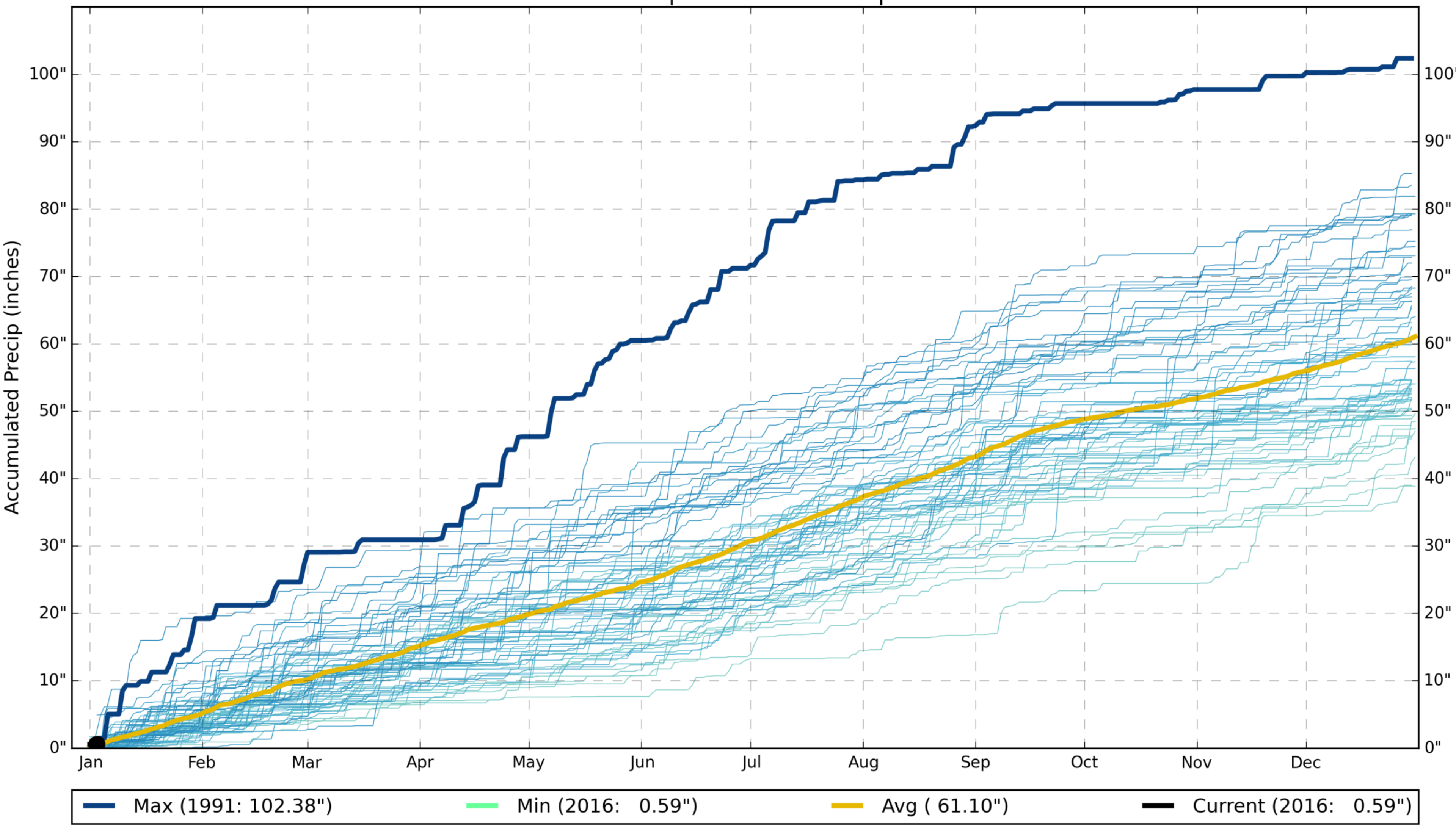


```
In [21]: gp.plot_precipitation("USW00012916")

PLOTTING PRECIPITATION DATA FOR STATION:  USW00012916

GRABBING LATEST STATION METADATA FILE

GETTING DATA FOR STATION:  USW00012916

In [22]: 
```

http://github.com/jjrennie/GHCNpy.git

USW00012916: NEW_ORLEANS_INTL_AP

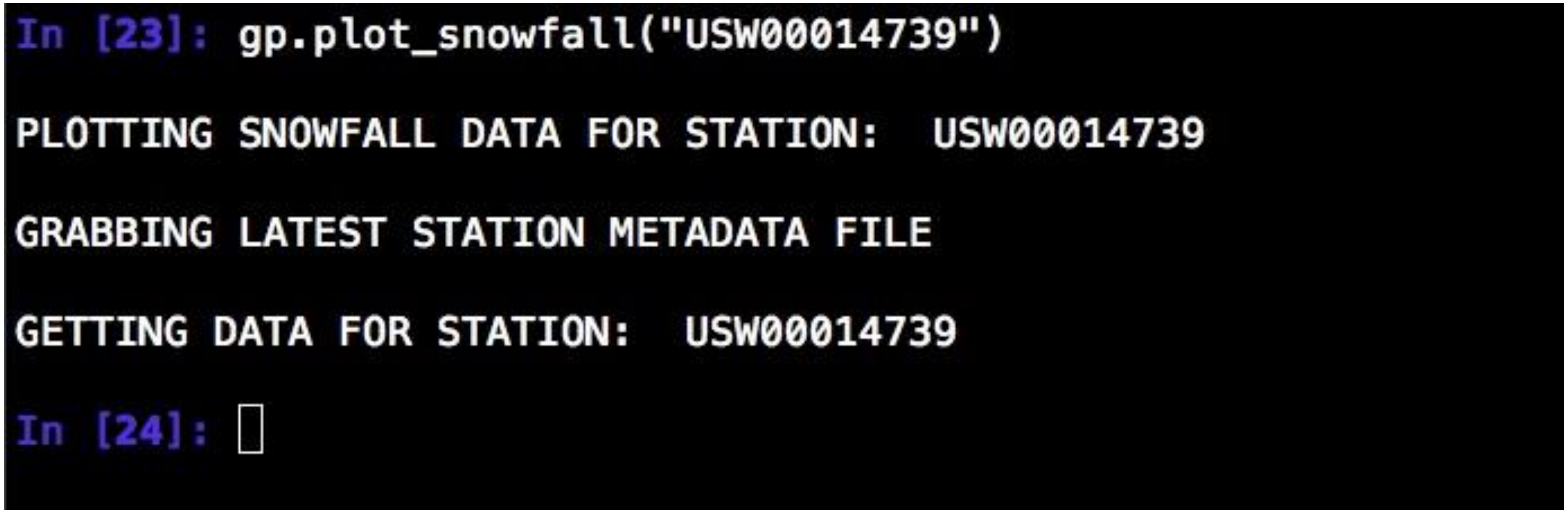
LAT= 29.9933 | LON= -90.2511 | ELEV= 3'



Which winter had the most snow in Boston?

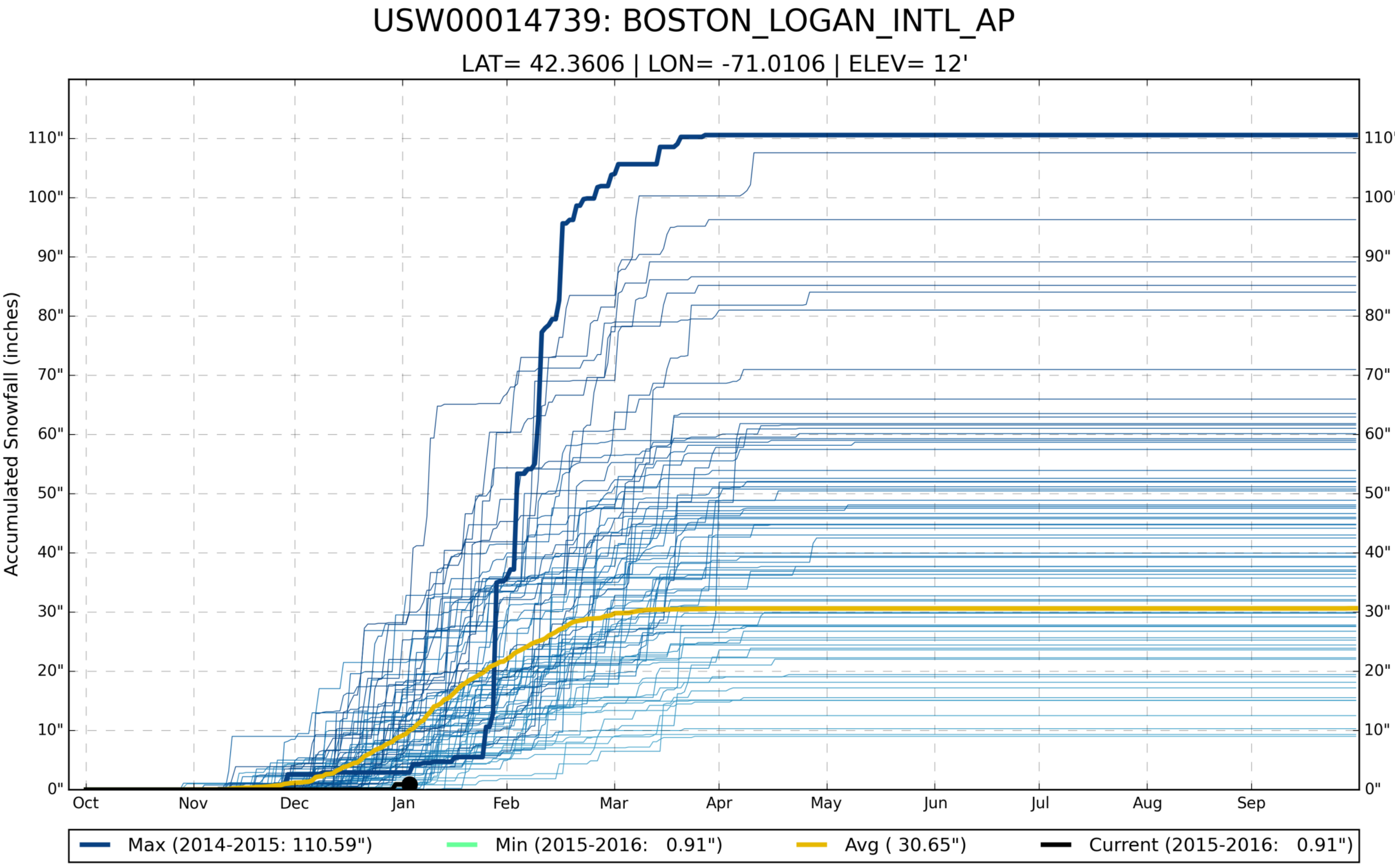
plot_snowfall(station_id)

- Same construct as Precipitation, but different enough to have own function



```
In [23]: gp.plot_snowfall("USW00014739")

PLOTTING SNOWFALL DATA FOR STATION:  USW00014739

GRABBING LATEST STATION METADATA FILE

GETTING DATA FOR STATION:  USW00014739

In [24]: 
```

http://github.com/jjrennie/GHCNpy.git

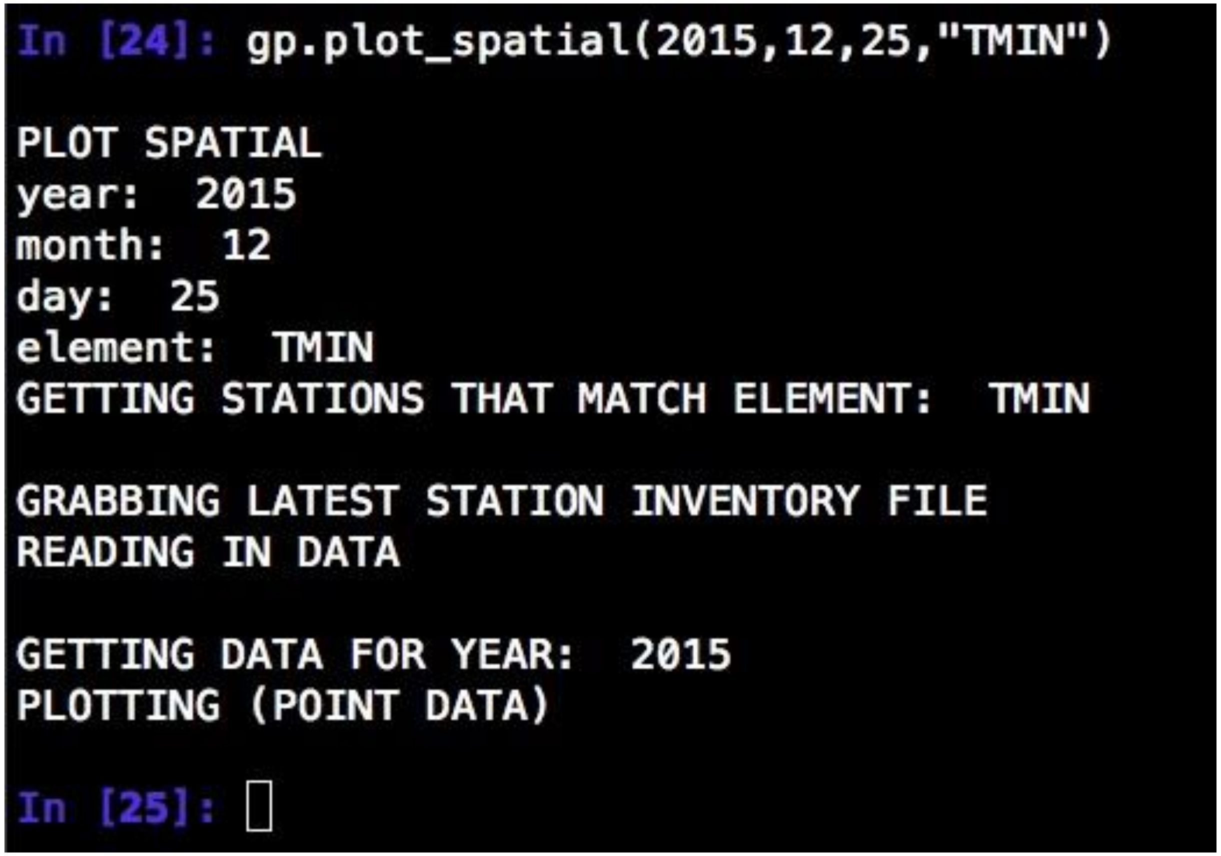


USW00014739: BOSTON_LOGAN_INTL_AP
LAT= 42.3606 | LON= -71.0106 | ELEV= 12'
Accumulated Snowfall (inches)
0"
10"
20"
30"
40"
50"
60"
70"
80"
90"
100"
110"
Oct
Nov
Dec
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Max (2014-2015: 110.59")
Min (2015-2016: 0.91")
Avg (30.65")
Current (2015-2016: 0.91")

What were the morning lows for the US on Christmas Day?

plot_spatial(year,month,day,element)

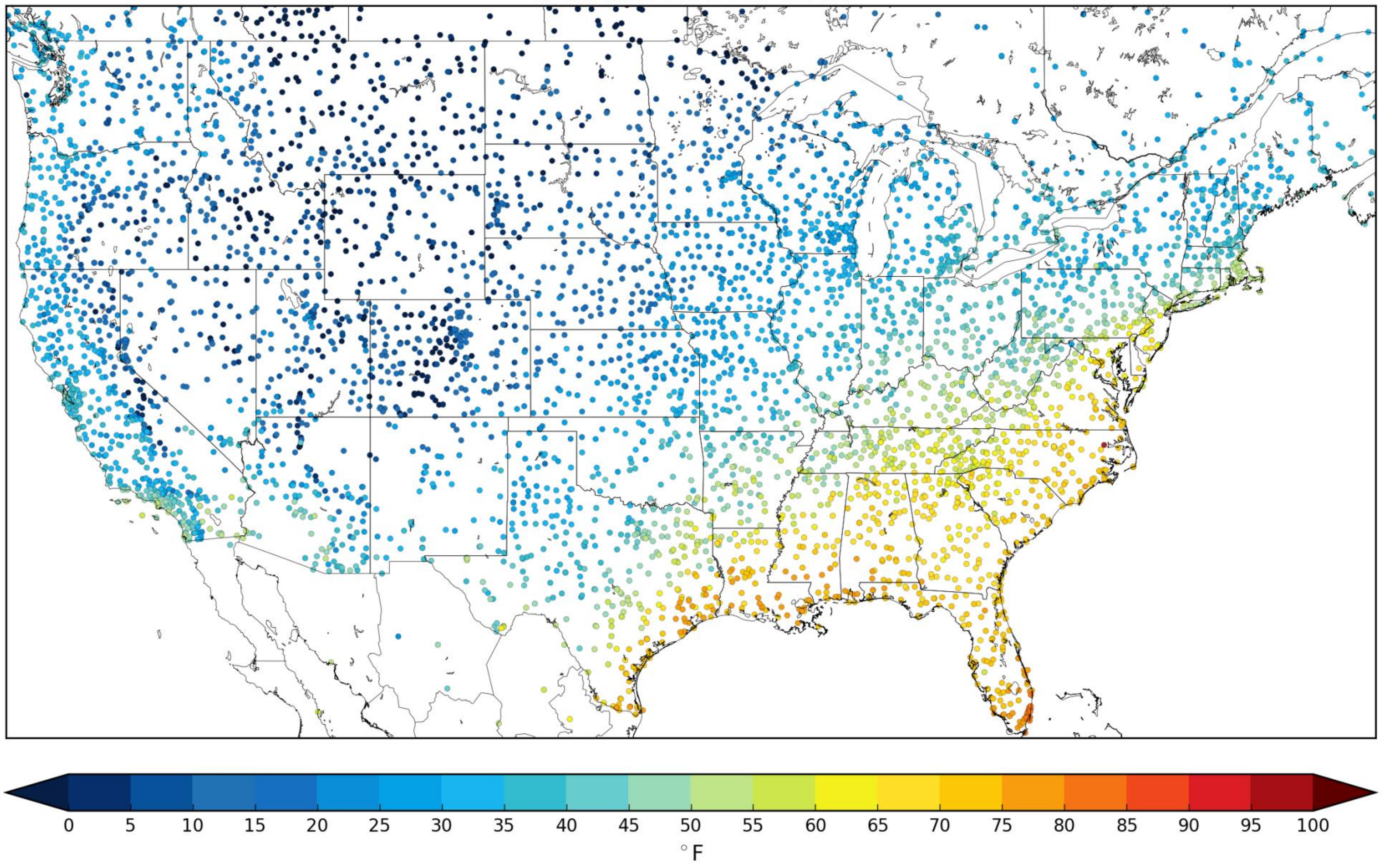
- Element is TMIN



```
In [24]: gp.plot_spatial(2015,12,25,"TMIN")

PLOT SPATIAL
year:  2015
month:  12
day:  25
element:  TMIN
GETTING STATIONS THAT MATCH ELEMENT:  TMIN

GRABBING LATEST STATION INVENTORY FILE
READING IN DATA

GETTING DATA FOR YEAR:  2015
PLOTTING (POINT DATA)

In [25]: 
```

http://github.com/jjrennie/GHCNpy.git

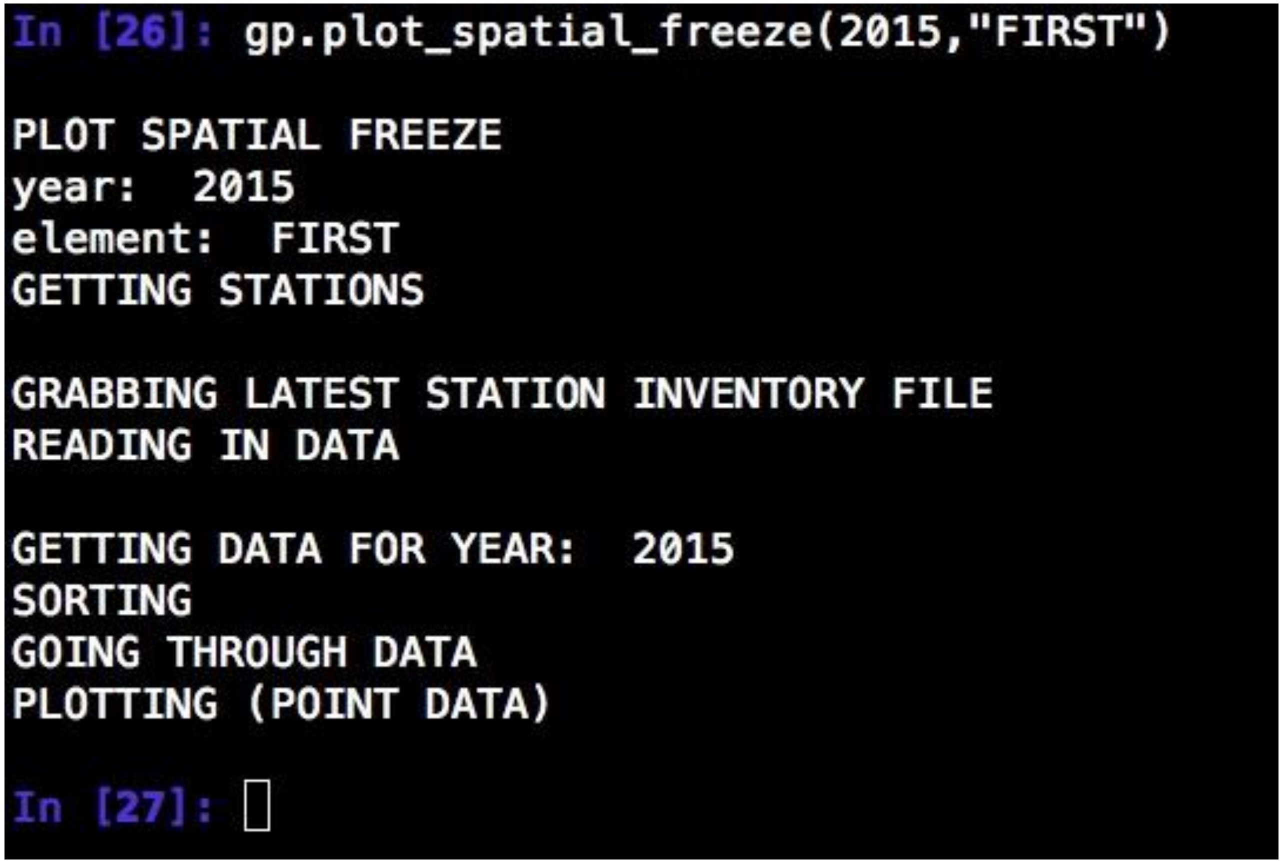
TMIN data for 2015 12 25



First Freeze Date for 2015?

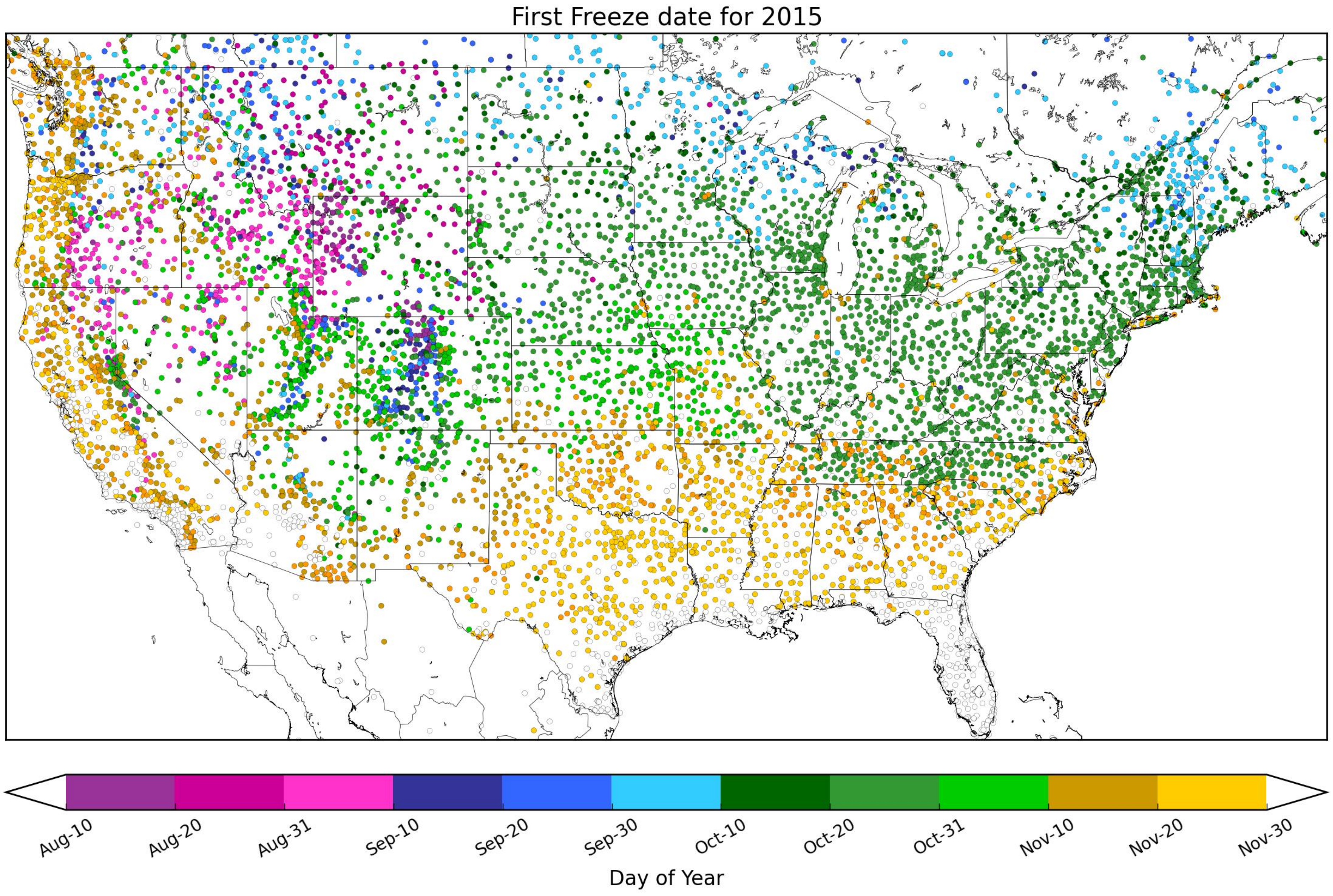
plot_spatial_freeze(year,element)

- Element="FIRST"



```
In [26]: gp.plot_spatial_freeze(2015,"FIRST")

PLOT SPATIAL FREEZE
year:  2015
element:  FIRST
GETTING STATIONS

GRABBING LATEST STATION INVENTORY FILE
READING IN DATA

GETTING DATA FOR YEAR:  2015
SORTING
GOING THROUGH DATA
PLOTTING (POINT DATA)

In [27]: 
```

http://github.com/jjrennie/GHCNpy.git


First Freeze date for 2015
Aug-10
Aug-20
Aug-31
Sep-10
Sep-20
Sep-30
Oct-10
Oct-20
Oct-31
Nov-10
Nov-20
Nov-30
Day of Year

Next Steps

- Accessing more of GHCN-Daily elements
- More visualizations and derived products
- More statistical calculations
  - SciPy, RPy, others?
- Incorporate GIS?
- Consider using Pandas instead of NumPy?
- Faster processing
  - All functions run in < 10 seconds, with the exception of the spatial plotting
- Utilize a database

Please "break" my program

http://github.com/jjrennie/GHCNpy.git

E-mail: jared@cicsnc.org

Twitter: @jjrennie