UWyo_Soundings

November 24, 2017

1 University of Wyoming Radiosonde Data and Soundings

1.0.1 Justin Richling 4/18/2017

http://weather.uwyo.edu/upperair/sounding.html

1.0.2 All the station cities and states

In [1]:
```
StationsList = ["Wallops Island, VA","Upton, NY","Chatham, MA","Albany, NY","Sterling,
"Buffalo, NY","Pittsburgh, PA","Wilmington, OH","White Lake, MI","Blacksburg, VA","Gree
"Newport, NC","Charleston, SC","Jacksonville, FL","Tampa Bay, FL","Miami, FL","Tallahas
"Shelby Cnty. Airport, AL","Peachtree City, GA","Nashville, TN","Gaylord, MI","Green Ba
"Lincoln, IL","Springfield, MO","Little Rock, AR","Jackson Thomas, MS","Slidell Muni.,
"Shreveport, LA","Norman, OK","Dodge City, KS","Topeka, KS","Omaha, NE","Chanhassen, M
"Aberdeen, SD","Bismarck, ND","Rapid City, SD","North Platte, NE","Amarillo, TX","Midla
"Corpus Christi, TX","Brownsville, TX","Ft. Worth, TX","Santa Teresa, NM","Albuquerque
"Riverton, WY","Glasgow, MT","Great Falls, MT","Salt Lake City, UT","Flagstaff, AZ","Tu
"Elko, NV","Boise, ID","Spokane, WA","Quillayute, WA","Salem, OR","Medford, OR","Reno,
"Oakland, CA","Vandenberg Air Force Base, CA","San Diego, CA"]
```

In [2]:
```
StationsList2 = ["Wallops Island","Upton","Chatham","Albany","Sterling","Gray",\
"Buffalo","Pittsburgh","Wilmington","White Lake","Blacksburg","Greensboro",\
"Newport","Charleston","Jacksonville","Tampa Bay","Miami","Tallahassee",\
"Shelby Cnty. Airport","Peachtree City","Nashville","Gaylord","Green Bay","Davenport",\
"Lincoln","Springfield","Little Rock","Jackson Thomas","Slidell Muni.","Lake Charles",\
"Shreveport","Norman","Dodge City","Topeka","Omaha","Chanhassen","International Falls"
"Aberdeen","Bismarck","Rapid City","North Platte","Amarillo","Midland","Del Rio",\
"Corpus Christi","Brownsville","Ft. Worth","Santa Teresa","Albuquerque","Denver","Gran
"Riverton","Glasgow","Great Falls","Salt Lake City","Flagstaff","Tuscon",\
"Yuma Prarie Grnds","Las Vegas","Elko","Boise","Spokane","Quillayute","Salem","Medford
"Oakland","Vandenberg Air Force Base","San Diego"]
```

In [3]:
```
# just making sure the lenth of the list so we can use it later in the
# range argument of the for loops
len(StationsList)
```

Out[3]: 69

1.0.3 All the station code numbers

In [4]:
```
StationNumList = [72402,72501,74494,72518,72403,74389,72528,72520,72426,72632,72318,723
72202,72214,72230,72215,72327,72635,72645,74455,74560,72440,72340,72235,72233,72240,722
72558,72649,72747,72659,72764,72662,72562,72363,72265,72261,72251,72250,72249,72364,723
72768,72776,72572,72376,72274,74004,72388,72582,72681,72786,72797,72694,72597,72489,724
```

In [5]:
```
# Double checking I entered all the station numbers, they need to have a matching
# station city
len(StationNumList)
```

Out[5]: 69

StationInfo2[0][0],StationInfo2[1][0]

In [6]:
```
#StationFinal = dict(zip(StationsList,StationNumList))
StationFinal2 = dict(zip(StationsList2,StationNumList))
StationFinal_1 = dict(zip(StationNumList,StationsList))
#StationFinal2_1 = dict(zip(StationNumList,StationsList2))
```

In [7]:
```
StationFinal_1
```

Out[7]:
```
{72202: 'Miami, FL',
 72206: 'Jacksonville, FL',
 72208: 'Charleston, SC',
 72210: 'Tampa Bay, FL',
 72214: 'Tallahassee, FL',
 72215: 'Peachtree City, GA',
 72230: 'Shelby Cnty. Airport, AL',
 72233: 'Slidell Muni., LA',
 72235: 'Jackson Thomas, MS',
 72240: 'Lake Charles, LA',
 72248: 'Shreveport, LA',
 72249: 'Ft. Worth, TX',
 72250: 'Brownsville, TX',
 72251: 'Corpus Christi, TX',
 72261: 'Del Rio, TX',
 72265: 'Midland, TX',
 72274: 'Tuscon, AZ',
 72293: 'San Diego, CA',
 72305: 'Newport, NC',
 72317: 'Greensboro, NC',
 72318: 'Blacksburg, VA',
 72327: 'Nashville, TN',
 72340: 'Little Rock, AR',
 72357: 'Norman, OK',
 72363: 'Amarillo, TX',
 72364: 'Santa Teresa, NM',
 72365: 'Albuquerque, NM',
 72376: 'Flagstaff, AZ',
```

```
 72388: 'Las Vegas, NV',
 72393: 'Vandenberg Air Force Base, CA',
 72402: 'Wallops Island, VA',
 72403: 'Sterling, VA',
 72426: 'Wilmington, OH',
 72440: 'Springfield, MO',
 72451: 'Dodge City, KS',
 72456: 'Topeka, KS',
 72469: 'Denver, CO',
 72476: 'Grand Juncion, CO',
 72489: 'Reno, NV',
 72493: 'Oakland, CA',
 72501: 'Upton, NY',
 72518: 'Albany, NY',
 72520: 'Pittsburgh, PA',
 72528: 'Buffalo, NY',
 72558: 'Omaha, NE',
 72562: 'North Platte, NE',
 72572: 'Salt Lake City, UT',
 72582: 'Elko, NV',
 72597: 'Medford, OR',
 72632: 'White Lake, MI',
 72635: 'Gaylord, MI',
 72645: 'Green Bay, WI',
 72649: 'Chanhassen, MN',
 72659: 'Aberdeen, SD',
 72662: 'Rapid City, SD',
 72672: 'Riverton, WY',
 72681: 'Boise, ID',
 72694: 'Salem, OR',
 72747: 'International Falls, MN',
 72764: 'Bismarck, ND',
 72768: 'Glasgow, MT',
 72776: 'Great Falls, MT',
 72786: 'Spokane, WA',
 72797: 'Quillayute, WA',
 74004: 'Yuma Prarie Grnds, AZ',
 74389: 'Gray, ME',
 74455: 'Davenport, IA',
 74494: 'Chatham, MA',
 74560: 'Lincoln, IL'}
```

StationFinal2

```
In [8]: # Since we have dicts for the station cities and numbers, you can
        # search with either city or code
        print StationFinal2["Aberdeen"]
        print StationFinal_1[72786]
```

```
72659
Spokane, WA
```

```
In [9]: import os
```

```
In [10]: os.chdir("/Users/chowdahead/Documents/stuff from old macbook/SkewT-1.1.0/")
```

1.0.4 This is necessary for me because I probably need to change the path for the SkewT library...

```
In [11]: %pylab
         from matplotlib.pyplot import imshow
         import matplotlib.image as mpimg
         import time,urllib,urllib2,cStringIO,logging,datetime,webbrowser,\
         IPython.display,shutil
         import numpy as np
         #from mpl_toolkits.basemap import Basemap, cm
         #from scipy.io import netcdf
         import matplotlib.pyplot as plt
         from skewt import SkewT
         from bs4 import BeautifulSoup
         from IPython.core.display import Image
         from PIL import Image as PILImage
         import sys, time, datetime
```

```
Using matplotlib backend: MacOSX
Populating the interactive namespace from numpy and matplotlib
```

```
In [12]: def UWyoRadiosonde(stn,year,month,day,hour,path,plot_title):
             #Denver = 72469

             #if hour == 0:
                 #hour = "0"+str(hour)

             try:
         # 1)
         # Wyoming URL to download Sounding from
                 url = "http://weather.uwyo.edu/cgi-bin/sounding?region=naconf"+\
             "&TYPE=TEXT%3ALIST&YEAR="+str(year)+"&MONTH="+str(month)+"&FROM="+\
             str(day)+str(hour)+"&TO="+str(day)+str(hour)+"&STNM="+str(stn)
                 #print url
         #url = "http://weather.uwyo.edu/cgi-bin/sounding?region=naconf&TYPE=\
         #TEXT%3ALIST&YEAR=2015&MONTH=06&FROM=1400&TO=1400&STNM=72469"
                 content = urllib2.urlopen(url).read()

         # 2)
         # Remove the html tags
```

```
            soup = BeautifulSoup(content)
            data_text = soup.get_text()

        # 3)
        # Split the content by new line.
            splitted = data_text.split("\n",data_text.count("\n"))

        # 4)
        # Write this splitted text to a .txt document
            Sounding_filename = str(stn)+'.'+str(year)+str(month)+\
        str(day)+str(hour)+"Z"+'.txt'
            UWyoFilename = str(year)+"_"+str(month)+"_"+str(day)+"_"+\
            str(hour)+"_"+str(stn)+"_Sounding.png"

            f = open(path+"/"+Sounding_filename,'w')
            for line in splitted[4:]:      # This is the key part of the function
                f.write(line+'\n')
            f.close()

        # 5)
            S = SkewT.Sounding(path+"/"+Sounding_filename)
            S.plot_skewt(title=plot_title+"Z")
            #fig = plt.figure()
            #plt.show()
            plt.axis('off')
            savefig(path+"/"+UWyoFilename)
            plt.close()
        except ValueError:
            logging.exception("No Data from website: "+Sounding_filename)
            pass

        print url
        return UWyoFilename

In [13]: Time = raw_input("(C)urrent time or (a)rchive? ")
        if Time == "C":
            now = datetime.datetime.now()
            Day = now.day
            Year = now.year
            Hour = now.hour
            Month = now.month

        if Time == "a":
            Year = input("Year: ")
            Month = input("Month: ")
            Day = input("Day: ")
            Hour = input("Would you like 0 or 12Z? ")
            if Hour == 0:
```

```
        Hour = "0"+str(Hour)
print "Current Local Hour: "+str(Hour)
if 6 < Hour < 18:
    Hour = 12

if 18 < Hour < 25:
    Hour = 0
    Hour = "0"+str(Hour)
if 0 < Hour < 6:
    Hour = 0
    Hour = "0"+str(Hour)

if Month < 10:
    Month = "0"+str(Month)
if Day < 10:
    Day = "0"+str(Day)

print "Fixed Local Hour: "+str(Hour)
#mypath = raw_input("Where would you like the file? \
#(For now you need to provide the full path file)")

print datetime.date.today().strftime("%B")[:3]
month = datetime.date.today().strftime("%B")[:3]

location = raw_input("(D)aily map or (O)ther location: ")
if location == "D":
    mypath ="/Users/ChowdaHead/Desktop/Weather_Blog/"+month+"_"+str(Day)+"/"
if location == "O":
    mypath = raw_input("Where would you like the file? \
    (For now you need to provide the full path file)")
#path = "/Users/ChowdaHead/Desktop/Weather_Blog/UWyo_Soundings/"
#mypath = path+str(Year)+"_"+str(Month)+"_"+str(int(Day))+"_"+str(Hour)+"Z"

if not os.path.isdir(mypath):
    os.makedirs(mypath)
os.chdir(mypath)

multiplemaps = raw_input("Multiple Maps? (y) or (n): ")
if multiplemaps == "y":

    Input = raw_input("The default will run Denver and Grand Junction soundings. \
Run these (y) or choose other station (n)? ")

# UWyoRadiosonde(station#,year,month(numerical),day(numerical),time(00 or 12 Z),file

    if Input == "y":
# Denver
        UWyoRadiosonde(72476,Year,Month,Day,Hour,mypath,"Grand Junction Sounding: ")
```

```
        # Grand Junction
                UWyoRadiosonde(72469,Year,Month,Day,Hour,mypath,"Denver Sounding: ")

            if Input == "n":
                mapzz = raw_input("Enter number of different station maps: ")
                print mapzz
                for i in range(int(mapzz)):
                    mapp = raw_input("Which map: (please provide a 5-digit station code) ")
                    if mapp in StationNumList:

                        j = StationNumList.index(mapp)
                        print "\n"+"Index Number: "+str(j)
                        UWyoRadiosonde(int(mapp),Year,Month,Day,Hour,mypath,str(mapp)+" Soundi
                #time.sleep(45)

        if multiplemaps == "n":

            StnInput = input("Choose 5 digit station number: ")
            if StnInput in StationNumList:
            # If Input is in our list, then set j equal to that numbered index
            # j will be an int!!
                j = StationNumList.index(StnInput)
                print "\n"+"Index Number: "+str(j)

                #UWyoRadiosonde(station number, year, month, day, hour, file path, plot title

            UWyoRadiosonde(StnInput,Year,Month,Day,Hour,mypath,str(StnInput)+" Sounding: "+sti
```

```
(C)urrent time or (a)rchive? C
Current Local Hour: 9
Fixed Local Hour: 12
Nov
(D)aily map or (O)ther location: D
Multiple Maps? (y) or (n): y
The default will run Denver and Grand Junction soundings. Run these (y) or choose other statio


/Users/chowdahead/anaconda/lib/python2.7/site-packages/bs4/__init__.py:181: UserWarning: No pa

The code that caused this warning is on line 174 of the file /Users/chowdahead/anaconda/lib/py

 BeautifulSoup(YOUR_MARKUP})

to this:

 BeautifulSoup(YOUR_MARKUP, "lxml")

  markup_type=markup_type))
```

```
---- Lifted Parcel Quantities ----
Parcel: SB
Ps  : 853.0hPa
TCs :   4.6C
TDs :   0.0C
-------------
Plcl: 794.6hPa
Tlcl:  -1.0C
Plfc:   nanhPa
P_el:   nanhPa
CAPE:   0.0J
CIN:    0.0J
http://weather.uwyo.edu/cgi-bin/sounding?region=naconf&TYPE=TEXT%3ALIST&YEAR=2017&MONTH=11&FRO

---- Lifted Parcel Quantities ----
Parcel: SB
Ps  : 832.0hPa
TCs :  15.4C
TDs :  -8.6C
-------------
Plcl: 600.0hPa
Tlcl:  -10.3C
Plfc:   nanhPa
P_el:   nanhPa
CAPE:   0.0J
CIN:    0.0J
http://weather.uwyo.edu/cgi-bin/sounding?region=naconf&TYPE=TEXT%3ALIST&YEAR=2017&MONTH=11&FRO
```

In [ ]: