jan16_2019_inclass

January 16, 2019

1 Basic info that should be at the beginning of every jupyter notebook

- Alex Weech
- ATMOS 6040/ Spring 2019
- January 16 inclass assignment
- Requires alta_snow.csv

January 16 2019
ATMOS 6040: Environmental Statistics
Alex Weech

Download this notebook and all images and data by downloading the ZIP file from GitHub, or use the git command:

```
git clone https://github.com/johnhorel/ATMOS_5040_2019.git
```

> Note: Windows users will have to install git for Windows and execute the git command from the PowerShell.

2 Using Python modules

`numpy` provides routines to handle arrays and many calculations efficiently and imported by convention as `np`. Numpy functions are very good at handling homogeneous data arrays (and similar in that respect to matlab functions).

`pyplot` is a *submodule* of matplotlib. It is typically imported as the alias `plt` to handle basic plotting

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

```
/usr/local/python/anaconda3/lib/python3.6/site-packages/matplotlib/font_manager.py:281: UserWa
  'Matplotlib is building the font cache using fc-list. '
```

3 January 16, 2019 Inclass assignment

4 Long-term variations in Alta snowfall

For info on the data http://utahavalanchecenter.org/alta-monthly-snowfall

On GitHub, look in the `data` folder for a file called `alta_snow.csv` and download it.

Open the `alta_snow.csv` file in the Jupyter Lab environment to see the column contents and the units.

-column 0 is the year and column 7 is the seasonal total

```
In [2]: #read the lake level data
        year = np.genfromtxt('../data/alta_snow.csv', delimiter=',', usecols=0)
        #convert seasonal totals from inches to cm
        tot = 2.54 * np.genfromtxt('../data/alta_snow.csv', delimiter=',', usecols=7)
```

```
In [3]: print(year)
        print(tot)
```

```
[1946. 1947. 1948. 1949. 1950. 1951. 1952. 1953. 1954. 1955. 1956. 1957.
 1958. 1959. 1960. 1961. 1962. 1963. 1964. 1965. 1966. 1967. 1968. 1969.
 1970. 1971. 1972. 1973. 1974. 1975. 1976. 1977. 1978. 1979. 1980. 1981.
 1982. 1983. 1984. 1985. 1986. 1987. 1988. 1989. 1990. 1991. 1992. 1993.
 1994. 1995. 1996. 1997. 1998. 1999. 2000. 2001. 2002. 2003. 2004. 2005.
 2006. 2007. 2008. 2009. 2010. 2011. 2012. 2013. 2014. 2015. 2016. 2017.
 2018.]
[1145.54    949.96   1394.46   1328.42   1211.58    886.46   1628.14   1043.94
  972.82   1198.88   1168.4     980.44   1421.13    980.44   1004.57    828.04
 1019.81   1018.54   1437.64   1455.42   1099.82   1381.76   1217.93   1437.894
 1165.86   1223.01   1185.164  1261.11   1512.824  1536.7    1116.33    798.83
 1332.23   1493.52   1305.56    993.14   1767.84   1617.98   1888.49   1160.78
 1521.46    969.772  1042.162  1477.01   1137.92   1473.708  1003.3    1652.016
 1245.362  1893.316  1427.48   1521.714  1460.246  1164.336  1132.84   1193.038
 1441.958  1014.476  1449.832  1406.144  1609.09    904.24   1661.16   1468.12
 1092.2    1404.62    836.93    971.55    908.05    679.45    998.22   1347.47
  731.52 ]
```

5 Do some basic calcs on the Alta snowfall data

```
In [4]: # determine the number of years
        ny = len(year)
        print('number of years= ',ny)
        #compute the mean of seasonal totals
        mmn = np.mean(tot)
        print('mean seasonal total= %8.2f cm'% (mmn))
        #find the max seasonal value and the year of that max
        maxs = np.max(tot)
        mxi = np.argmax(tot)
        print("year of max seasonal total and amount: %d %.1f cm" % (year[mxi],maxs))
        #find the min seasonal value and the year of that min
        mins = np.min(tot)
        mni = np.argmin(tot)
        print("year of min seasonal total and amount: %d %.1f cm" % (year[mni],mins))
```

```
number of years=  73
mean seasonal total=  1244.00 cm
year of max seasonal total and amount: 1995 1893.3 cm
year of min seasonal total and amount: 2015 679.5 cm
```

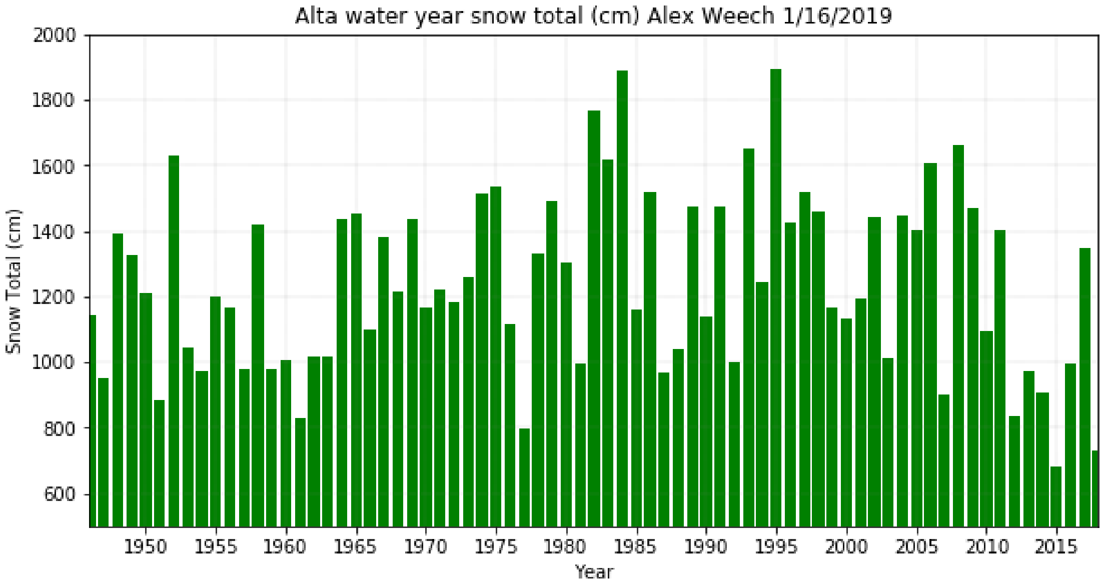
6 Figure 2.1

Create bar plot time series of Alta seasonal snowfall totals

```
In [5]: ticks = np.arange(1950,2020,5)

        fig,(ax1) = plt.subplots(1,1,figsize=(10,5))
        ax1.bar(year,tot,color='green')
        ax1.set(xlim=(1946,2018),ylim=(500,2000))
        ax1.set(xlabel="Year",ylabel='Snow Total (cm)')
        ax1.set(xticks=ticks)
        ax1.set(title="Alta water year snow total (cm) Alex Weech 1/16/2019")
        ax1.grid(linestyle='--', color='grey', linewidth=.2)

        plt.savefig('alta_anow_inclass_2019_python.png')
```


Alta water year snow total (cm) Alex Weech 1/16/2019
Snow Total (cm)
Year