Lecture 04.2 Logical Indexing

September 6, 2022

1 Logical indexing in Pandas

Duncan Callaway

```
[ ]: import pandas as pd
```

1.0.1 Logical indexing

Logical indexing is an extremely powerful way to pull data out of a frame.
For example, with the stacked data frame, let's pull out only wind generation.

To get started we're going to work with a different data set. Same as before, but the values are "stacked", as you can see here:

```
[ ]: caiso_data_stack = pd.read_csv('CAISO_2017to2018_stack.csv', index_col=0)
     caiso_data_stack.head()
```

First, I'll show you a boolean series based on comparisons to the 'Source' data column:

```
[ ]: wind_indx = (caiso_data_stack['Source']=='WIND TOTAL')
     wind_indx
```

Now we can embed that inside the `.loc` method:

```
[ ]: caiso_data_stack.loc[wind_indx,:]
```

1.1 Q: What hour in our data has the lowest average hourly wind generation?

First let's import numpy

```
[ ]: import numpy as np
```

```
[ ]: wind = caiso_data_stack.loc[caiso_data_stack['Source']=='WIND TOTAL',:]
```

In a moment we'll use pivots to do this better, but for now let's use a for loop to get information by hour.

First thing to do is figure out how to get the hour out of the index.

`datetime.strptime` is useful for this if you're working on individual dates.

But `pd.to_datetime` is even better, especially if you're working on a lot of values in a list (or as the case will be, values in a pandas series).

[ ]:
```
windex = pd.to_datetime(wind.index)
windex.hour
```

Now we'll do the real work. We're going to average all wind values with the same hour

[ ]:
```
wind_ave = [] # initalizes a list to populate

for i in range(0,24):
    hr_bool = windex.hour == i
    hr_vals = wind.loc[hr_bool,:]
    avgwind = np.mean(hr_vals)
    wind_ave.append(avgwind)
wind_ave
```

[ ]:
```
import matplotlib.pyplot as plt
```

[ ]:
```
plt.plot(wind_ave)
```

We can see pretty clearly that the min is 10 or 11…let's dig a little more.

One way to do this is to drop the data into a data frame and then *sort* the data frame.

[ ]:
```
df_wind = pd.DataFrame(wind_ave)
df_wind
```

I'm going to be adding more MWh values to the data frame in just a moment, so let's be clear that this is the average

[ ]:
```
df_wind.columns = ['Average MWh']
```

[ ]:
```
df_wind.sort_values(by='Average MWh',ascending=True).head()
```

Ok – so it looks as though mid-day is the minimum *average*.

1.1.1 Q: What's the range of wind values by hour?

[ ]:
```
wind_min = [] # initalizes a list to populate
wind_max = [] # initalizes a list to populate
for i in range(0,24):
    wind_min.append(np.min(wind.loc[windex.hour == i,:]))
    wind_max.append(np.max(wind.loc[windex.hour == i,:]))
```

[ ]:
```
df_wind['min MWh']=pd.DataFrame(wind_min)['MWh']
df_wind['max MWh']=pd.DataFrame(wind_max)['MWh']
```

[ ]:
```
df_wind
```

```
[ ]: plt.plot(df_wind)
```