loess-regression

September 23, 2014

```
In []: import numpy as np
       import pandas as pd
       import statsmodels.api as sm
       import statsmodels.formula.api as smf
```

```
In []: %matplotlib inline
       from matplotlib import pyplot as plt
       from matplotlib import dates
```

```
In []: obama = pd.read_csv('https://github.com/pmagwene/Bio723/raw/master/datasets/obama-favorable-ratin
```

```
In []: obama.columns
```

```
In []: obama.drop(['Pollster URL', 'Source URL', 'Source URL.1'], axis=1, inplace=True)
```

```
In []: obama.columns = ['Pollster', 'Start', 'End', 'Release', 'NumObs',
                         'Popn', 'Mode','Favorable','Unfavorable','Undecided']
```

1 Specifying Date Columns in Pandas

If a data set contains date values, Pandas needs to be told about this. Here's one way to set variables of interest to dates.

```
In []: obama['Start'] = pd.to_datetime(obama['Start'], utc=True)
       obama['End'] = pd.to_datetime(obama['End'], utc=True)
       obama['Release'] = pd.to_datetime(obama['Release'], utc=True)
```

Pandas and Matplotlib have different notions of time zones. There should be a more elegant way to handle this date conversion, but I haven't tracked it down yet.

```
In []: # need to convert Pandas dates to Matplotlib compatible dates
       end_dates = [dates.datestr2num(str(i)) for i in obama.End]

       # note the use of the plot_date function to make
       plt.plot_date(edates, obama.Favorable, 'k.')
       plt.ylabel('Favorability')
```

1.1 Fitting Loess (Lowess) Models in StatsModels

```
In []: from statsmodels.nonparametric.smoothers_lowess import lowess
```

```
In []: fav_lowess = lowess(obama.Favorable, edates)
```

```
In []: plt.plot_date(end_dates, obama.Favorable, 'k.')
       plt.plot_date(fav_lowess[:,0], fav_lowess[:,1], 'r-')
       plt.ylabel('Favorability')
```

1.1.1 Modifying the Loess Model

Our loess fit above looks to be oversmoothed. We're not capturing the trends of interest. To make our loess fit more "local" we can use the `frac` argument to specify the fraction of the data used for each local regression.

```
In []: fav_lowess = lowess(obama.Favorable, edates, frac=0.15)
       plt.plot_date(edates, obama.Favorable, 'k.')
       plt.plot_date(fav_lowess[:,0], fav_lowess[:,1], 'r-')
       plt.ylabel('Favorability')
```

```
In []:
```