Lesson 7

Outliers

In [1]:
```
import pandas as pd
import sys
```

In [2]:
```
print 'Python version ' + sys.version
print 'Pandas version: ' + pd.__version__
```

```
Python version 2.7.5 |Anaconda 2.1.0 (64-bit)| (default, Jul  1 2013, 1
Pandas version: 0.15.2
```

In [3]:
```
# Create a dataframe with dates as your index
States = ['NY', 'NY', 'NY', 'NY', 'FL', 'FL', 'GA', 'GA', 'FL', 'FL']
data = [1.0, 2, 3, 4, 5, 6, 7, 8, 9, 10]
idx = pd.date_range('1/1/2012', periods=10, freq='MS')
df1 = pd.DataFrame(data, index=idx, columns=['Revenue'])
df1['State'] = States

# Create a second dataframe
data2 = [10.0, 10.0, 9, 9, 8, 8, 7, 7, 6, 6]
idx2 = pd.date_range('1/1/2013', periods=10, freq='MS')
df2 = pd.DataFrame(data2, index=idx2, columns=['Revenue'])
df2['State'] = States
```

In [4]:
```
# Combine dataframes
df = pd.concat([df1,df2])
df
```

Out[4]:

	Revenue	State
2012-01-01	1	NY
2012-02-01	2	NY
2012-03-01	3	NY
2012-04-01	4	NY
2012-05-01	5	FL
2012-06-01	6	FL
2012-07-01	7	GA
2012-08-01	8	GA
2012-09-01	9	FL
2012-10-01	10	FL
2013-01-01	10	NY
2013-02-01	10	NY
2013-03-01	9	NY
2013-04-01	9	NY
2013-05-01	8	FL
2013-06-01	8	FL
2013-07-01	7	GA
2013-08-01	7	GA
2013-09-01	6	FL
2013-10-01	6	FL

Ways to Calculate Outliers

Note: Average and Standard Deviation are only valid for gaussian distributions.

In [5]:
```
# Method 1

# make a copy of original df
newdf = df.copy()

newdf['x-Mean'] = abs(newdf['Revenue'] - newdf['Revenue'].mean())
newdf['1.96*std'] = 1.96*newdf['Revenue'].std()
newdf['Outlier'] = abs(newdf['Revenue'] - newdf['Revenue'].mean()) > 1.96
newdf
```

Out[5]:

	Revenue	State	x-Mean	1.96*std	Outlier
2012-01-01	1	NY	5.75	5.200273	True
2012-02-01	2	NY	4.75	5.200273	False
2012-03-01	3	NY	3.75	5.200273	False
2012-04-01	4	NY	2.75	5.200273	False
2012-05-01	5	FL	1.75	5.200273	False
2012-06-01	6	FL	0.75	5.200273	False
2012-07-01	7	GA	0.25	5.200273	False
2012-08-01	8	GA	1.25	5.200273	False
2012-09-01	9	FL	2.25	5.200273	False
2012-10-01	10	FL	3.25	5.200273	False
2013-01-01	10	NY	3.25	5.200273	False
2013-02-01	10	NY	3.25	5.200273	False
2013-03-01	9	NY	2.25	5.200273	False
2013-04-01	9	NY	2.25	5.200273	False
2013-05-01	8	FL	1.25	5.200273	False
2013-06-01	8	FL	1.25	5.200273	False
2013-07-01	7	GA	0.25	5.200273	False
2013-08-01	7	GA	0.25	5.200273	False
2013-09-01	6	FL	0.75	5.200273	False
2013-10-01	6	FL	0.75	5.200273	False

In [6]:
```
# Method 2
# Group by item

# make a copy of original df
newdf = df.copy()

State = newdf.groupby('State')

newdf['Outlier'] = State.transform( lambda x: abs(x-x.mean()) > 1.96*x.st
newdf['x-Mean'] = State.transform( lambda x: abs(x-x.mean()) )
newdf['1.96*std'] = State.transform( lambda x: 1.96*x.std() )
newdf
```

Out[6]:

	Revenue	State	Outlier	x-Mean	1.96*std
2012-01-01	1	NY	False	5.00	7.554813
2012-02-01	2	NY	False	4.00	7.554813
2012-03-01	3	NY	False	3.00	7.554813
2012-04-01	4	NY	False	2.00	7.554813
2012-05-01	5	FL	False	2.25	3.434996
2012-06-01	6	FL	False	1.25	3.434996
2012-07-01	7	GA	False	0.25	0.980000
2012-08-01	8	GA	False	0.75	0.980000
2012-09-01	9	FL	False	1.75	3.434996
2012-10-01	10	FL	False	2.75	3.434996
2013-01-01	10	NY	False	4.00	7.554813
2013-02-01	10	NY	False	4.00	7.554813
2013-03-01	9	NY	False	3.00	7.554813
2013-04-01	9	NY	False	3.00	7.554813
2013-05-01	8	FL	False	0.75	3.434996
2013-06-01	8	FL	False	0.75	3.434996
2013-07-01	7	GA	False	0.25	0.980000
2013-08-01	7	GA	False	0.25	0.980000
2013-09-01	6	FL	False	1.25	3.434996
2013-10-01	6	FL	False	1.25	3.434996

In [7]:
```
# Method 2
# Group by multiple items

# make a copy of original df
newdf = df.copy()

StateMonth = newdf.groupby(['State', lambda x: x.month])

newdf['Outlier'] = StateMonth.transform( lambda x: abs(x-x.mean()) > 1.96
newdf['x-Mean'] = StateMonth.transform( lambda x: abs(x-x.mean()) )
newdf['1.96*std'] = StateMonth.transform( lambda x: 1.96*x.std() )
newdf
```

Out[7]:

	Revenue	State	Outlier	x-Mean	1.96*std
2012-01-01	1	NY	False	4.5	12.473364
2012-02-01	2	NY	False	4.0	11.087434
2012-03-01	3	NY	False	3.0	8.315576
2012-04-01	4	NY	False	2.5	6.929646
2012-05-01	5	FL	False	1.5	4.157788
2012-06-01	6	FL	False	1.0	2.771859
2012-07-01	7	GA	False	0.0	0.000000
2012-08-01	8	GA	False	0.5	1.385929
2012-09-01	9	FL	False	1.5	4.157788
2012-10-01	10	FL	False	2.0	5.543717
2013-01-01	10	NY	False	4.5	12.473364
2013-02-01	10	NY	False	4.0	11.087434
2013-03-01	9	NY	False	3.0	8.315576
2013-04-01	9	NY	False	2.5	6.929646
2013-05-01	8	FL	False	1.5	4.157788
2013-06-01	8	FL	False	1.0	2.771859
2013-07-01	7	GA	False	0.0	0.000000
2013-08-01	7	GA	False	0.5	1.385929
2013-09-01	6	FL	False	1.5	4.157788
2013-10-01	6	FL	False	2.0	5.543717

In [8]:
```
# Method 3
# Group by item

# make a copy of original df
newdf = df.copy()

State = newdf.groupby('State')

def s(group):
    group['x-Mean'] = abs(group['Revenue'] - group['Revenue'].mean())
    group['1.96*std'] = 1.96*group['Revenue'].std()
    group['Outlier'] = abs(group['Revenue'] - group['Revenue'].mean()) >
    return group

Newdf2 = State.apply(s)
Newdf2
```

Out[8]:

	Revenue	State	x-Mean	1.96*std	Outlier
2012-01-01	1	NY	5.00	7.554813	False
2012-02-01	2	NY	4.00	7.554813	False
2012-03-01	3	NY	3.00	7.554813	False
2012-04-01	4	NY	2.00	7.554813	False
2012-05-01	5	FL	2.25	3.434996	False
2012-06-01	6	FL	1.25	3.434996	False
2012-07-01	7	GA	0.25	0.980000	False
2012-08-01	8	GA	0.75	0.980000	False
2012-09-01	9	FL	1.75	3.434996	False
2012-10-01	10	FL	2.75	3.434996	False
2013-01-01	10	NY	4.00	7.554813	False
2013-02-01	10	NY	4.00	7.554813	False
2013-03-01	9	NY	3.00	7.554813	False
2013-04-01	9	NY	3.00	7.554813	False
2013-05-01	8	FL	0.75	3.434996	False
2013-06-01	8	FL	0.75	3.434996	False
2013-07-01	7	GA	0.25	0.980000	False
2013-08-01	7	GA	0.25	0.980000	False
2013-09-01	6	FL	1.25	3.434996	False
2013-10-01	6	FL	1.25	3.434996	False

In [9]:
```
# Method 3
# Group by multiple items

# make a copy of original df
newdf = df.copy()

StateMonth = newdf.groupby(['State', lambda x: x.month])

def s(group):
    group['x-Mean'] = abs(group['Revenue'] - group['Revenue'].mean())
    group['1.96*std'] = 1.96*group['Revenue'].std()
    group['Outlier'] = abs(group['Revenue'] - group['Revenue'].mean()) >
    return group

Newdf2 = StateMonth.apply(s)
Newdf2
```

Out[9]:

	Revenue	State	x-Mean	1.96*std	Outlier
2012-01-01	1	NY	4.5	12.473364	False
2012-02-01	2	NY	4.0	11.087434	False
2012-03-01	3	NY	3.0	8.315576	False
2012-04-01	4	NY	2.5	6.929646	False
2012-05-01	5	FL	1.5	4.157788	False
2012-06-01	6	FL	1.0	2.771859	False
2012-07-01	7	GA	0.0	0.000000	False
2012-08-01	8	GA	0.5	1.385929	False
2012-09-01	9	FL	1.5	4.157788	False
2012-10-01	10	FL	2.0	5.543717	False
2013-01-01	10	NY	4.5	12.473364	False
2013-02-01	10	NY	4.0	11.087434	False
2013-03-01	9	NY	3.0	8.315576	False
2013-04-01	9	NY	2.5	6.929646	False
2013-05-01	8	FL	1.5	4.157788	False
2013-06-01	8	FL	1.0	2.771859	False
2013-07-01	7	GA	0.0	0.000000	False
2013-08-01	7	GA	0.5	1.385929	False
2013-09-01	6	FL	1.5	4.157788	False
2013-10-01	6	FL	2.0	5.543717	False

Assumign a non gaussian distribution (if you plot it, it will not look like a normal distribution)

In [10]:
```
# make a copy of original df
newdf = df.copy()

State = newdf.groupby('State')

newdf['Lower'] = State['Revenue'].transform( lambda x: x.quantile(q=.25)
newdf['Upper'] = State['Revenue'].transform( lambda x: x.quantile(q=.75)
newdf['Outlier'] = (newdf['Revenue'] < newdf['Lower']) | (newdf['Revenue'
newdf
```

Out[10]:

	Revenue	State	Lower	Upper	Outlier
2012-01-01	1	NY	-7.000	19.000	False
2012-02-01	2	NY	-7.000	19.000	False
2012-03-01	3	NY	-7.000	19.000	False
2012-04-01	4	NY	-7.000	19.000	False
2012-05-01	5	FL	2.625	11.625	False
2012-06-01	6	FL	2.625	11.625	False
2012-07-01	7	GA	6.625	7.625	False
2012-08-01	8	GA	6.625	7.625	True
2012-09-01	9	FL	2.625	11.625	False
2012-10-01	10	FL	2.625	11.625	False
2013-01-01	10	NY	-7.000	19.000	False
2013-02-01	10	NY	-7.000	19.000	False
2013-03-01	9	NY	-7.000	19.000	False
2013-04-01	9	NY	-7.000	19.000	False
2013-05-01	8	FL	2.625	11.625	False
2013-06-01	8	FL	2.625	11.625	False
2013-07-01	7	GA	6.625	7.625	False
2013-08-01	7	GA	6.625	7.625	False
2013-09-01	6	FL	2.625	11.625	False
2013-10-01	6	FL	2.625	11.625	False

Author: David Rojas (http://www.hedaro.com/)