Q-1 Select a dataset or datasets of your choice.

- Apply simple linear or multivariate linear regression and then evaluate it as it is done in "linear_regression.ipynb".

source: http://archive.ics.uci.edu/ml/datasets/ISTANBUL+STOCK+EXCHANGE (http://archive.ics.uci.edu/ml/datasets/ISTANBUL+STOCK+EXCHANGE)

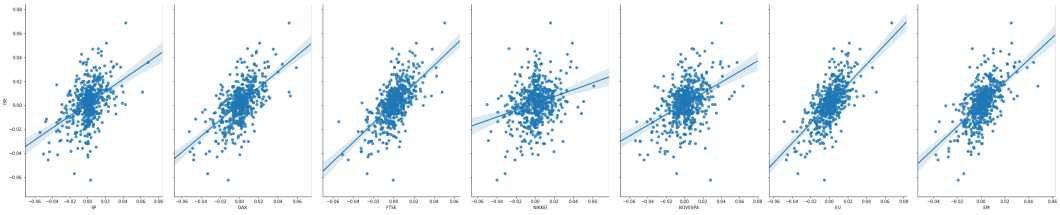
In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
% matplotlib inline

from IPython.core.display import display, HTML
display(HTML("<style>.container { width:90% !important; }</style>"))
pd.set_option('display.max_columns', 100)
```

In [2]:
```
df2 = pd.read_csv('data_akbilgic.csv', index_col=0)
```

In [3]:
```
sns.pairplot(df2, x_vars=['SP','DAX','FTSE','NIKKEI','BOVESPA',
'EU','EM'], y_vars='ISE', size=7, aspect=0.7, kind='reg')
```

Out[3]: <seaborn.axisgrid.PairGrid at 0x10d1a6358>



In [4]:
```
df2.sample()
```

Out[4]:

	ISE	ISE.1	SP	DAX	FTSE	NIKKEI	BOVESPA
date							
3-Nov-10	0.001202	0.007039	0.003671	-0.005502	-0.00146	0.021471	0.0

In [5]:
```
X = df2.loc[:,'SP':]
y = df2.ISE
```

In [6]:
```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, random
_state=1)
```

In [7]:
```
from sklearn.linear_model import LinearRegression

lr = LinearRegression()
lr.fit(X_train,y_train)
```

Out[7]:
```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=1, no
rmalize=False)
```

In [8]:
```
print(lr.intercept_)
print(lr.coef_)
```

```
0.000676600863895
[ 0.06037146 -0.06089133 -0.03375757 -0.00510386 -0.15754933
0.71117676
  0.53138047]
```

In [9]:
```
y_pred = lr.predict(X_test)
```

In [10]:
```
from sklearn import metrics

print('MAE:', metrics.mean_absolute_error(y_test,y_pred))
print('MSE:', metrics.mean_squared_error(y_test, y_pred))
print('RMSE:', np.sqrt(metrics.mean_squared_error(y_test, y_pred
)))
```

```
MAE: 0.0088625900406
MSE: 0.000138378089676
RMSE: 0.0117634216823
```