In [1]:

```
%pylab inline
import pandas as pd
import numpy as np
TodasEstaciones = pd.read_csv('Est1_Est2_Est3.csv',index_col=0,parse_dates=True)
TodasEstaciones.head()
```

```
Populating the interactive namespace from numpy and matplotlib
```

Out[1]:

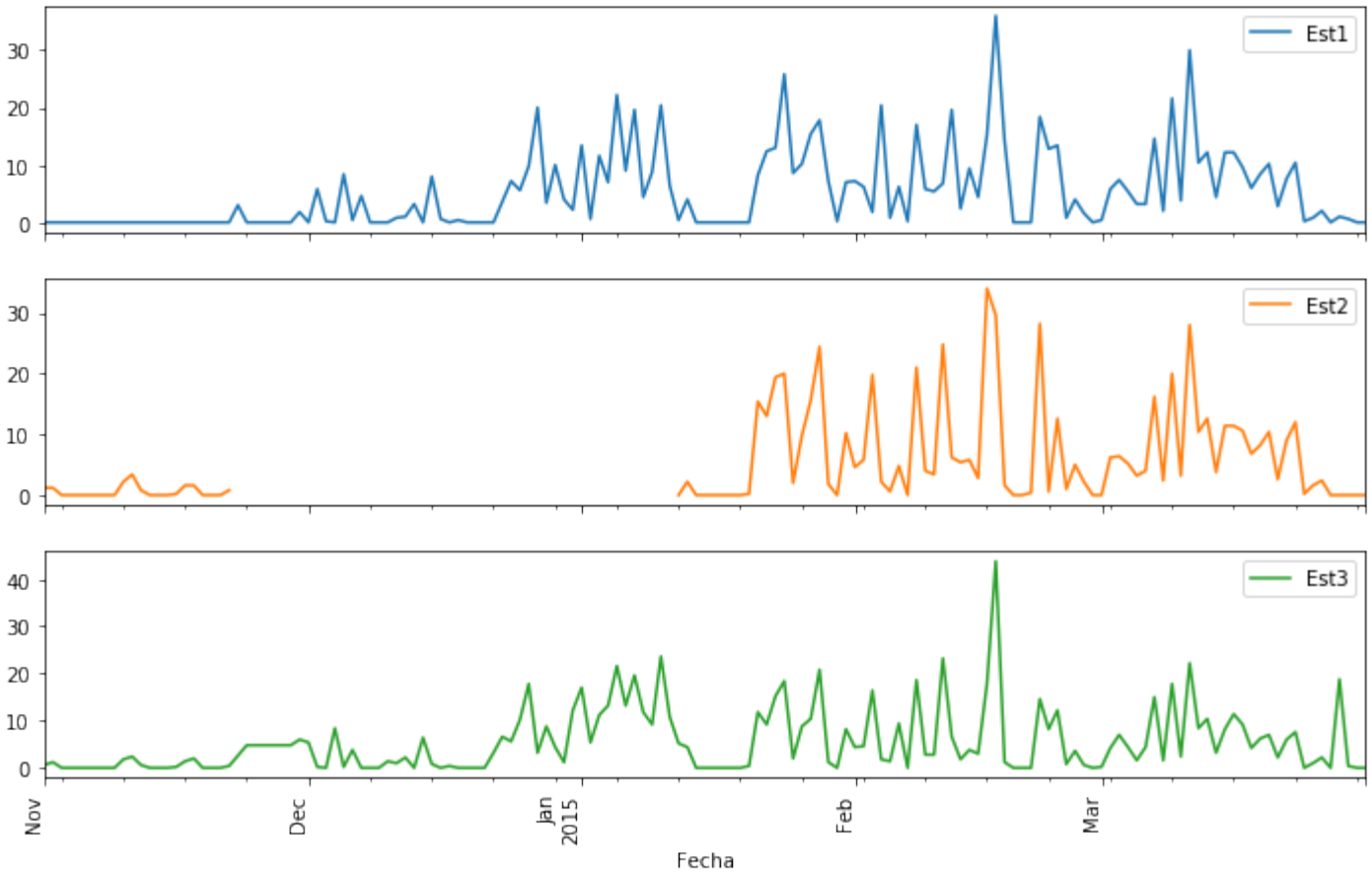
Fecha	Est1	Est2	Est3
2014-07-25	0.0	NaN	0.0
2014-07-26	0.6	NaN	0.0
2014-07-27	0.0	NaN	0.0
2014-07-28	0.0	NaN	0.0
2014-07-29	0.2	NaN	0.2

In [2]:

```
TodasEstaciones.loc['2014-11-01':'2015-03-31'].plot(subplots=True, figsize=(12, 8));
plt.legend(loc='best')
xticks(rotation='vertical')
```

Out[2]:

```
(array([16375, 16405, 16436, 16467, 16495, 16525], dtype=int64),
 <a list of 6 Text xticklabel objects>)
```



In [3]:

```
import datetime
#we create a date column to extract the week number
TodasEstaciones['date']=TodasEstaciones.index
#apply a lambda function to the whole panda dataframe column
TodasEstaciones['week'] = TodasEstaciones['date'].apply(lambda x: x.isocalendar()
[1])
#drop the date column because we dont need it
del TodasEstaciones['date']
#let see our dataframe
TodasEstaciones.head()
```

Out[3]:

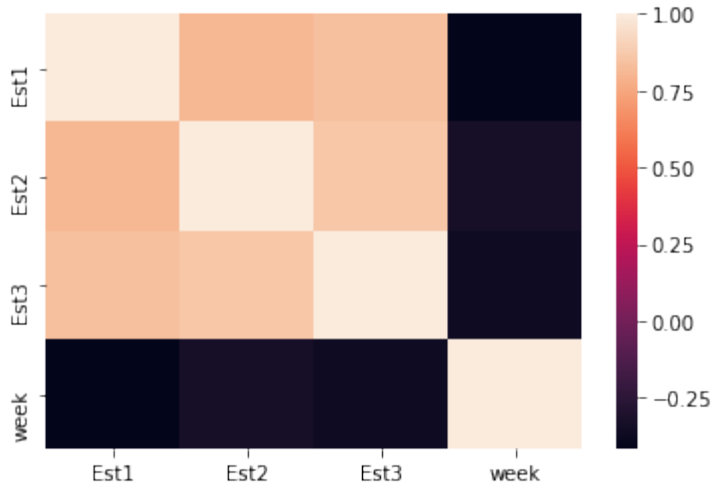
Fecha	Est1	Est2	Est3	week
2014-07-25	0.0	NaN	0.0	30
2014-07-26	0.6	NaN	0.0	30
2014-07-27	0.0	NaN	0.0	30
2014-07-28	0.0	NaN	0.0	31
2014-07-29	0.2	NaN	0.2	31

In [4]:

```
#creation of a correlation plot with seaborn
import seaborn as sns
corr = TodasEstaciones.corr()
sns.heatmap(corr,
            xticklabels=corr.columns.values,
            yticklabels=corr.columns.values)
```

Out[4]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1310f8aab70>
```



In [ ]:

```
#Definition of training sets
X_train = TodasEstaciones.loc['2015-01-20':'2015-03-27',
['Est1','Est3','week']].astype(float32).values#,'week']] # Est 1, 3 and #week
y_train = TodasEstaciones.loc['2015-01-20':'2015-03-
27','Est2'].astype(float32).values # Est 2
```

In [ ]:

```
# Import `StandardScaler` from `sklearn.preprocessing`
from sklearn.preprocessing import StandardScaler

# Define the scaler
scaler = StandardScaler().fit(X_train)

# Scale the train set
X_train = scaler.transform(X_train)
```

In [ ]:

```
X_train[:20]
```

In [ ]:

```
from keras.models import Sequential

from keras.layers import Dense

model = Sequential()

model.add(Dense(12, activation='linear', input_shape=(3,)))
model.add(Dense(8, activation='linear'))
```

```
model.add(Dense(1, activation='linear'))
model.summary()
```

In [ ]:

```
model.compile(loss='mean_squared_error',
              optimizer='adam',
              metrics=['accuracy'])

model.fit(X_train, y_train,epochs=200,verbose=0)
```

In [ ]:

```
y_pred = model.predict(X_train)
y_pred[:10]
```

In [ ]:

```
plot(TodasEstaciones.loc['2015-01-20':'2015-03-27'].index,y_pred,label='Predicted')
TodasEstaciones['Est2'].loc['2015-01-20':'2015-03-27'].plot()
figsize(12,8)
ylim(0,40)
legend(loc='best')
```

Predict the missing data in between 2014-11-23:2015-01-11¶

In [ ]:

```
#Get the prediction for the train set
X_missing = TodasEstaciones.loc['2014-11-23':'2015-01-11',
['Est1','Est3','week']].astype(float32).values
```

In [ ]:

```
# Import `StandardScaler` from `sklearn.preprocessing`
from sklearn.preprocessing import StandardScaler

# Define the scaler
scaler = StandardScaler().fit(X_missing)

# Scale the train set
X_missing = scaler.transform(X_missing)
```

In [ ]:

```
y_missing = model.predict(X_missing)
y_missing = y_missing.reshape([50]).tolist()
```

In [ ]:

```
TodasEstaciones['Est2_Completed']=TodasEstaciones['Est2']
TodasEstaciones['Est2_Completed'].loc['2014-11-23':'2015-01-11']=y_missing
```

In [ ]:

```
TodasEstaciones.loc['2014-11-01':'2015-03-31',
['Est1','Est2','Est2_Completed','Est3']].plot(subplots=True,
                                                    figsize=(15, 10));
plt.legend(loc='best')
xticks(rotation='vertical')
ylim(0,50)
```

In [ ]:

```
results = TodasEstaciones.loc['2012-09-28':'2020-01-31',
['Est1','Est2','Est2_Completed','Est3']]
```

In [ ]:

```
results.to_csv("results.csv")
```