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

```
/Users/kunalshriwas/opt/anaconda3/lib/python3.8/site-packages/pandas/core/computation/expressions.py:20
: UserWarning: Pandas requires version '2.7.3' or newer of 'numexpr' (version '2.7.1' currently install
ed).
  from pandas.core.computation.check import NUMEXPR_INSTALLED
```

In [2]:
```
df = pd.read_csv('Alcohol_Sales.csv',index_col = 'DATE' ,parse_dates=True)
df.head()
```

Out[2]:

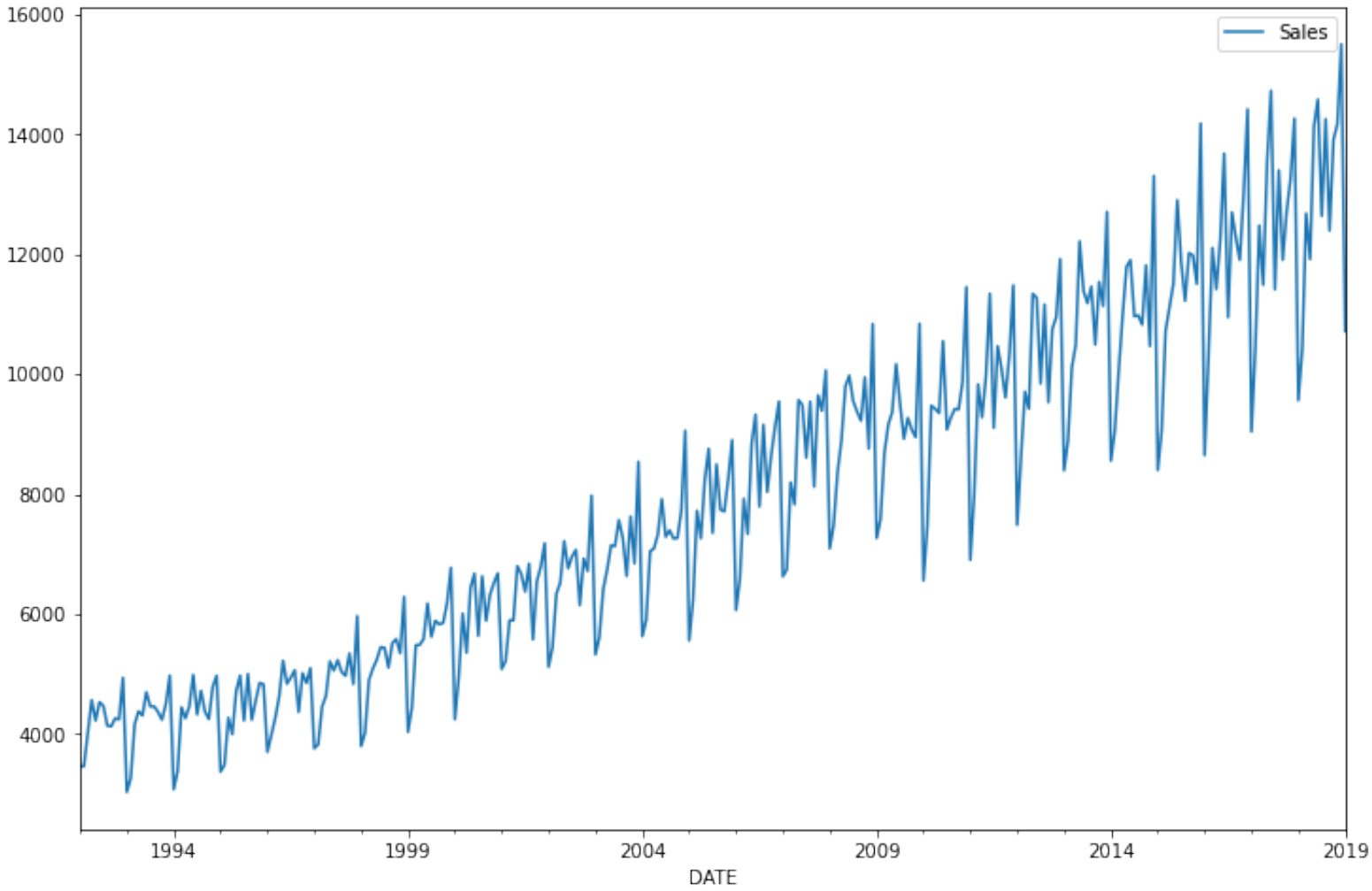
DATE	S4248SM144NCEN
1992-01-01	3459
1992-02-01	3458
1992-03-01	4002
1992-04-01	4564
1992-05-01	4221

In [3]:
```
df.shape
```

Out[3]:
```
(325, 1)
```

In [4]:
```
df.columns = ['Sales']
df.plot(figsize = (12,8))
```

Out[4]: <matplotlib.axes._subplots.AxesSubplot at 0x7feb0623d5e0>



In [5]:

```
# shifting data

df['sale_last_month'] = df['Sales'].shift(1)
df['sale_2months_back'] = df['Sales'].shift(2)
df['sale_3months_back'] = df['Sales'].shift(3)
df
```

Out[5]:

DATE	Sales	sale_last_month	sale_2months_back	sale_3months_back
1992-01-01	3459	NaN	NaN	NaN
1992-02-01	3458	3459.0	NaN	NaN
1992-03-01	4002	3458.0	3459.0	NaN
1992-04-01	4564	4002.0	3458.0	3459.0
1992-05-01	4221	4564.0	4002.0	3458.0
...	...	...	...	...
2018-09-01	12396	14257.0	12640.0	14583.0
2018-10-01	13914	12396.0	14257.0	12640.0
2018-11-01	14174	13914.0	12396.0	14257.0
2018-12-01	15504	14174.0	13914.0	12396.0
2019-01-01	10718	15504.0	14174.0	13914.0

325 rows × 4 columns

In [6]:
```
df.head()
```

Out[6]:

DATE	Sales	sale_last_month	sale_2months_back	sale_3months_back
1992-01-01	3459	NaN	NaN	NaN
1992-02-01	3458	3459.0	NaN	NaN
1992-03-01	4002	3458.0	3459.0	NaN
1992-04-01	4564	4002.0	3458.0	3459.0
1992-05-01	4221	4564.0	4002.0	3458.0

In [7]:
```
df = df.dropna()
df
```

Out[7]:

DATE	Sales	sale_last_month	sale_2months_back	sale_3months_back
1992-04-01	4564	4002.0	3458.0	3459.0
1992-05-01	4221	4564.0	4002.0	3458.0
1992-06-01	4529	4221.0	4564.0	4002.0
1992-07-01	4466	4529.0	4221.0	4564.0
1992-08-01	4137	4466.0	4529.0	4221.0
...	...	...	...	...
2018-09-01	12396	14257.0	12640.0	14583.0
2018-10-01	13914	12396.0	14257.0	12640.0
2018-11-01	14174	13914.0	12396.0	14257.0
2018-12-01	15504	14174.0	13914.0	12396.0

In [8]:
```
1 df.tail()
```

Out[8]:

DATE	Sales	sale_last_month	sale_2months_back	sale_3months_back
2018-09-01	12396	14257.0	12640.0	14583.0
2018-10-01	13914	12396.0	14257.0	12640.0
2018-11-01	14174	13914.0	12396.0	14257.0
2018-12-01	15504	14174.0	13914.0	12396.0
2019-01-01	10718	15504.0	14174.0	13914.0

In [20]:
```
import numpy as np
x1,x2,x3,y=df['sale_last_month'],df['sale_2months_back'],df['sale_3months_back'],df['Sales']
x1,x2,x3,y=np.array(x1),np.array(x2),np.array(x3),np.array(y)
x1,x2,x3,y=x1.reshape(-1,1),x2.reshape(-1,1),x3.reshape(-1,1),y.reshape(-1,1)
final_x=np.concatenate((x1,x2,x3),axis=1)
print(final_x)
```

```
[[ 4002.  3458.  3459.]
 [ 4564.  4002.  3458.]
 [ 4221.  4564.  4002.]
 [ 4529.  4221.  4564.]
 [ 4466.  4529.  4221.]
 [ 4137.  4466.  4529.]
 [ 4126.  4137.  4466.]
 [ 4259.  4126.  4137.]
 [ 4240.  4259.  4126.]
 [ 4936.  4240.  4259.]
 [ 3031.  4936.  4240.]
 [ 3261.  3031.  4936.]
 [ 4160.  3261.  3031.]
 [ 4377.  4160.  3261.]
 [ 4307.  4377.  4160.]
 [ 4696.  4307.  4377.]
 [ 4458.  4696.  4307.]
 [ 4457.  4458.  4696.]
 [ 4364.  4457.  4458.]
```

In [32]:
```
type(final_x)
```

Out[32]: numpy.ndarray

In [11]:
```
# separate out features and target

# x1 = df['sale_last_month']
# x2 = df['sale_2months_back']
# x3 = df['sale_3months_back']
# y = df['Sales']

# # converting into np array
# x1 = np.array(x1)
# x2 = np.array(x2)
# x3 = np.array(x3)
# y = np.array(y)

# #reshape
# x1 = x1.reshape(-1,1)
# x2 = x2.reshape(-1,1)
# x3 = x3.reshape(-1,1)
# y = y.reshape(-1,1)

# final_features_set = np.concatenate((x1,x2,x3),axis = 1)
# final_features_set
```

In [12]:
```
y.shape
```

Out[12]: (322, 1)

In [33]:
```
# Train test split

X_train,X_test,y_train,y_test = final_features_set[:-30],final_features_set[-30:],y[:-30],y[-30:]
```

In [34]:
```
#X_train,X_test,y_train,y_test=final_x[:-30],final_x[-30:],y[:-30],y[-30:]
```

In [35]:

```
from sklearn.linear_model import LinearRegression
lr = LinearRegression()
lr.fit(X_train,y_train)
pred_lr = lr.predict(X_test)
```

In [36]:

```
from sklearn.ensemble import RandomForestRegressor
rf = RandomForestRegressor(n_estimators=100)
rf.fit(X_train,y_train)
pred_rf = rf.predict(X_test)

```

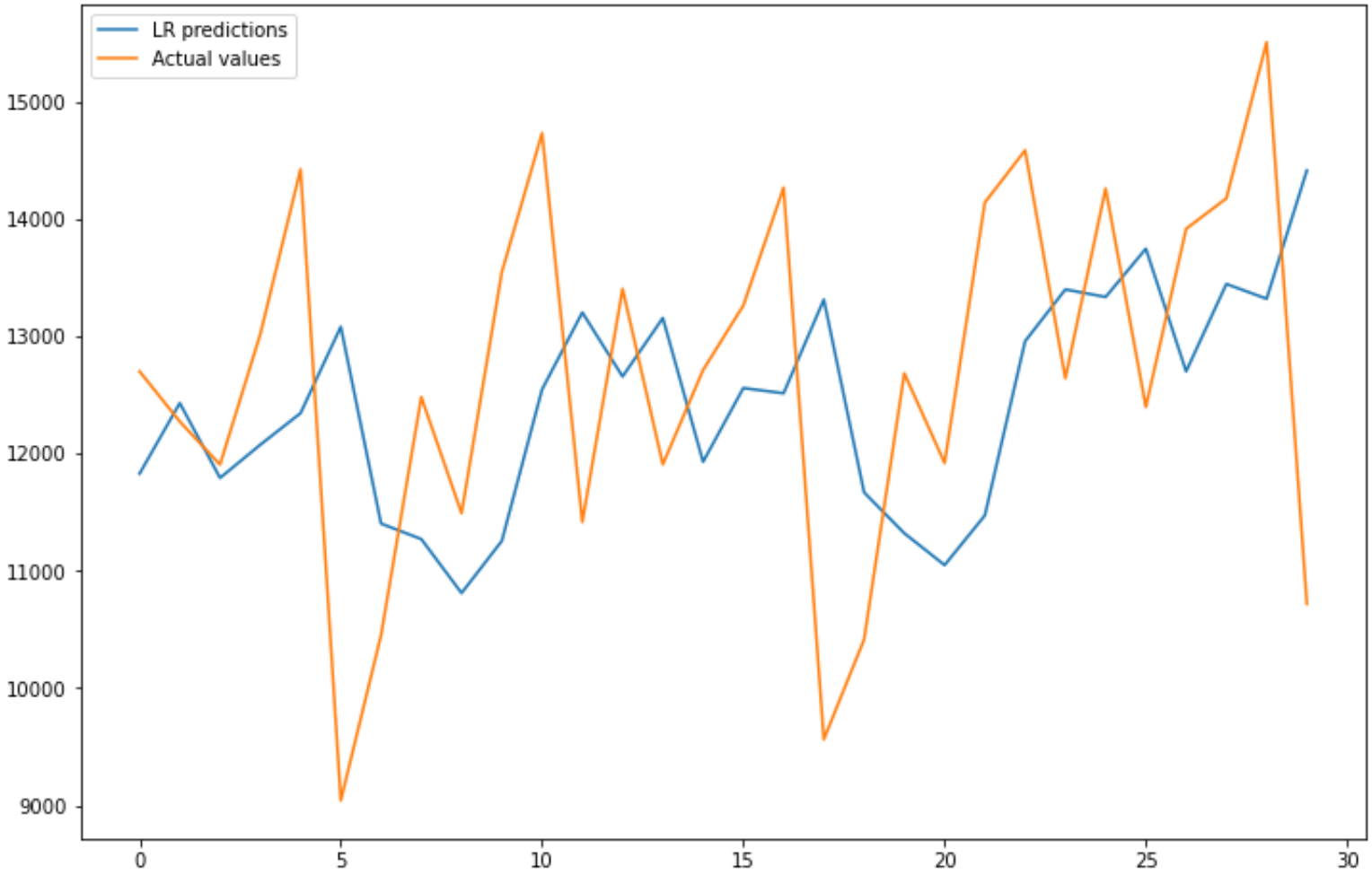
```
<ipython-input-36-4a2db3808926>:3: DataConversionWarning: A column-vector y was passed when a 1d array
was expected. Please change the shape of y to (n_samples,), for example using ravel().
  rf.fit(X_train,y_train)
```

In [37]:
```
1 print(pred_lr)
```

```
[[11827.38475478]
 [12429.62369271]
 [11791.52722406]
 [12074.3498315 ]
 [12343.48904265]
 [13079.54295242]
 [11402.58070872]
 [11270.45486831]
 [10810.70139556]
 [11254.70018415]
 [12545.65208275]
 [13201.55699274]
 [12655.70647517]
 [13154.28727827]
 [11928.87869712]
 [12557.27299237]
 [12513.06506434]
 [13311.56168518]
 [11668.38035137]
 [11320.13549056]
 [11048.93371063]
 [11472.09230302]
 [12956.38770063]
 [13397.71591856]
 [13333.20034719]
 [13744.6486356 ]
 [12699.60492899]
 [13443.98563967]
 [13318.08546261]
 [14412.26144533]]
```

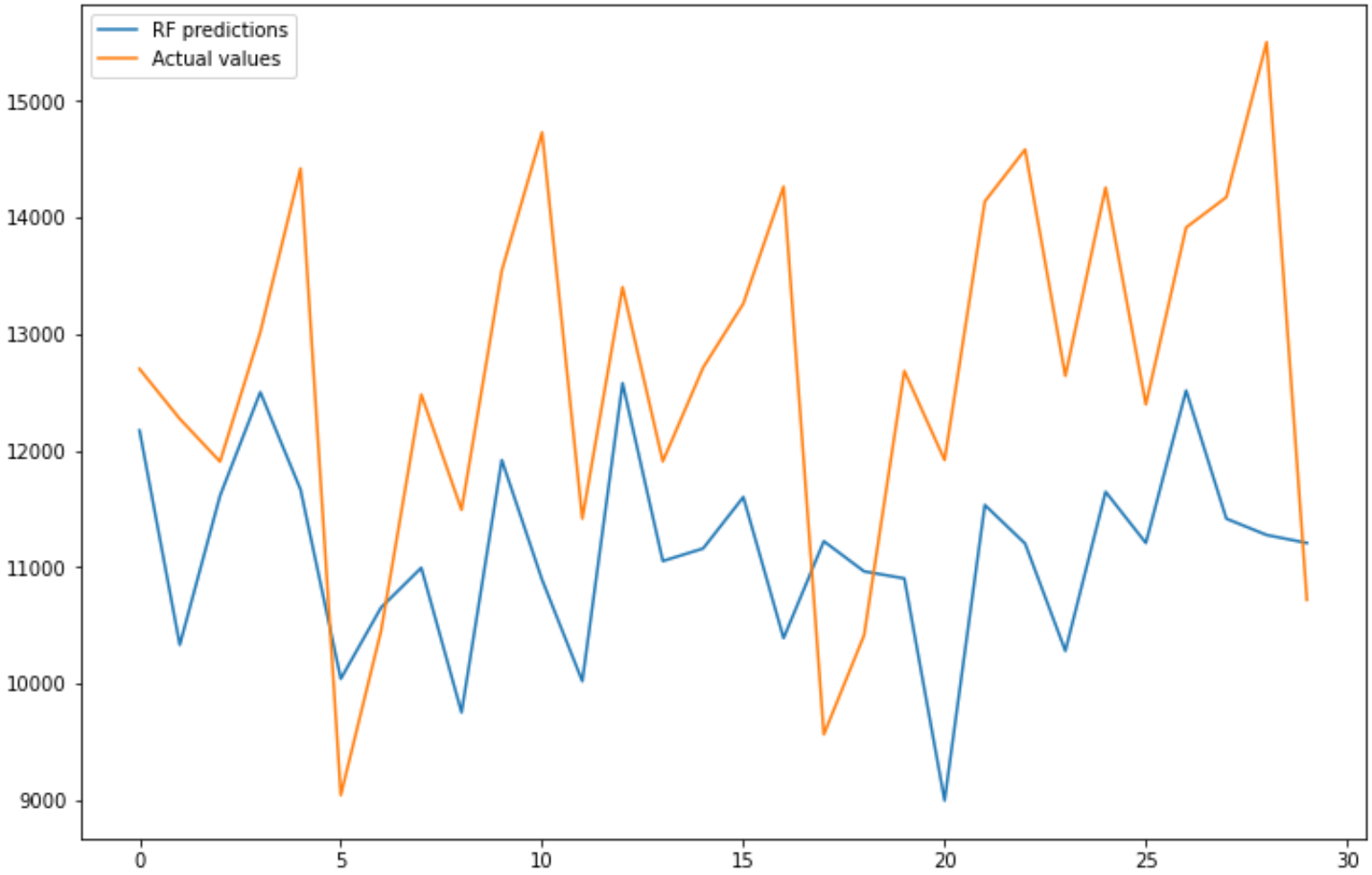
In [38]:

```
plt.figure(figsize = (12,8))
plt.plot(pred_lr,label = "LR predictions")
plt.plot(y_test, label = "Actual values")
plt.legend()
plt.show()
```



In [39]:

```
plt.figure(figsize = (12,8))
plt.plot(pred_rf,label = "RF predictions")
plt.plot(y_test, label = "Actual values")
plt.legend()
plt.show()
```



In [ ]:

```

```