10月26日　コンピュータ言語授業

学習メモ

In [ ]:

```
・回帰 ある属性で別の属性値を推定すること
・線形回帰 一次式
・回帰残渣を最小にするのが目標
・定常時系列 時刻で分布に変化が出ないこと
・相関係数は定常時系列を仮定して出す意味がある
・今の時刻から次の時刻を回帰するのが自己回帰モデル
・自己回帰モデルの自己相関係数はxtと(xt-1)の相関件数
```

In [2]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
import warnings
import xgboost as xgb
warnings.filterwarnings('ignore')

from sklearn.model_selection import train_test_split
from sklearn import linear_model
from sklearn.model_selection import GridSearchCV
pd.set_option("max_columns", 100)
pd.set_option('max_rows',1000)
pd.set_option('max_info_columns',100)
```

In [5]:

```
!pip install tqdm
```

```
Collecting tqdm
  Downloading https://files.pythonhosted.org/packages/91/55/8cb23a97301b177e
9c8e3226dba45bb454411de2cbd2574763267f226c2/tqdm-4.28.1-py2.py3-non
e-any.whl (45kB)
    100% |████████████████████████████████| 51kB 3.
6MB/s ta 0:00:01
Installing collected packages: tqdm
Successfully installed tqdm-4.28.1
```

In [6]:

```
from tqdm import tqdm_notebook as tqdm
```

In [130]:

```
ts7 = np.array([1,2,3,4,5,6])
```

In [131]:

```
lag = 1
tslen = len(ts7)
print(tslen)
xvec = ts7[0:tslen-lag]
yvec = ts7[lag:tslen]
xdevvec = xvec-xvec.mean()
ydevvec = yvec-yvec.mean()
cov = xdevvec.dot(ydevvec)/(tslen-1)
rho1 =cov/(xdevvec.std()*ydevvec.std())
rho1ver2 = np.corrcoef(ts7[0:tslen-lag],ts7[lag:tslen])[0][1] #2行2列の行列の左上だけとる
```

```
6
```

In [132]:

```
rho1
```

Out[132]:

```
0.9999999999999998
```

In [133]:

```
for lag in range(26):
    rho1 = np.corrcoef(ts7[0:tslen-lag], ts7[lag:tslen][0][1])
```

```
---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
<ipython-input-133-9b51e71f6072> in <module>()
      1 for lag in range(26):
----> 2     rho1 = np.corrcoef(ts7[0:tslen-lag], ts7[lag:tslen][0][1])

IndexError: invalid index to scalar variable.
```

In [134]:

```
np.corrcoef(ts7[0:tslen-lag],ts7[lag:tslen])
```

Out[134]:

```
array([[1., 1.],
       [1., 1.]])
```

In [17]:

```
xdevvec = xvec-xvec.mean()
ydevvec = yvec-yvec.mean()
print(xdevvec)
print(ydevvec)
```

```
[-2. -1.  0.  1.  2.]
[-2. -1.  0.  1.  2.]
```

In [18]:

```
print(cov)
```

```
2.0
```

In [73]:

```
print(ts7)
```

```
[ 0.       -4.66888802  1.59396894  4.07553885  6.08543859  2.64062009
  1.2827888   0.93315014 -0.80281971 -0.80599685 -0.98999554 -1.20779249
 -3.0660186  -4.16017481 -1.07740826 -0.80453259  1.42119527  3.33242433
  0.48247084  0.17070919 -3.28785803 -4.77790612 -3.66295499 -2.1010445
5
 -1.97815916  0.24251713 -0.02432449  0.63883508  1.74630566  2.2848840
7
  2.47745872 -1.71443062 -1.25993739 -0.94679488 -0.09105049 -4.0682379
2
  0.94991714 -0.99391643  4.11305363  1.8331009   2.68385792  3.76020283
  0.84374786  2.96763028  0.95683449 -0.1988117  -3.76362377 -5.70744067
 -7.51167611 -5.76255773 -2.55197459 -1.72611217  0.97552101  1.9828499
7
  1.38202155 -0.97022762 -4.4136221  -7.23033826 -2.94401626 -2.23176955
  0.85615224  0.95121693  1.3597485   2.7969223   0.47496831 -1.8915654
 -1.98298137 -2.38045172 -1.90814677  0.24046475  0.290926   -0.2295634
  0.22659485  1.22363965  1.68536814  2.5659099   1.89973632  2.75130583
  1.16668742  2.63729036  0.64436308  5.41700848  8.04511616  4.62373076
  5.62156036  2.68289412 -0.86363304 -0.818515   -0.83986183 -3.37540629
 -3.74487816 -0.33060466  3.0556269  -0.55504864  1.32547764  3.07546377
  3.17384911  4.13438243  2.59742915  1.70306734]
```

In [108]:

```
len(rho)
print(rho)
```

```
[1.0, 0.6913721883387566, 0.44078189996061046, 0.13229298723750985, -
0.09028271824635123, -0.22302762850005733, -0.2507929190626818, -0.1
52870475974372, -0.05035721630315227, 0.08075703234952834, 0.174587
0350571928, 0.2334361055954945, 0.19203647942493077, 0.17521436485
951378, 0.06920751773437223, -0.029541082628178488, -0.170763519016
93093, -0.2211006786819057, -0.22283162264871811, -0.104247915815614
8, 0.006289847960187607, 0.09765818641971798, 0.18585362250900608,
0.13534983033980671, 0.08689340833951464, -0.023307696495037337]
```

演習問題

In [357]:

```
ts7 = np.loadtxt("week7ar.csv",delimiter=",")
ts7.shape
```

Out[357]:

(100,)

In [233]:

```
lag = []
for i in range(0,26):
    lag.append(i)
```

In [234]:

```
import statsmodels.api as sm

rho = sm.tsa.stattools.acf(ts7, nlags=25)
```

In [372]:

```
rho.shape
```

Out[372]:

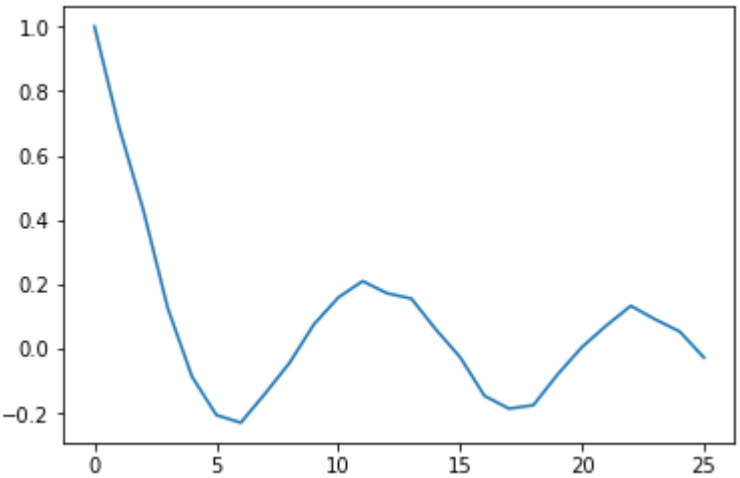
(26,)

In [235]:

```
plt.plot(lag, rho)
```

Out[235]:

[<matplotlib.lines.Line2D at 0x1a134532e8>]



lagが5ずつで相関係数が上がり下がりする

In [236]:

```
tslen = len(ts7)
tslen
```

Out[236]:

100

In [237]:

```
a = np.corrcoef(ts7[1:tslen],ts7[0:tslen-1])[0][1]
print(np.corrcoef(ts7[1:tslen],ts7[0:tslen-1])[0][1])
```

0.6913721883387566

In [304]:

```
xt_1 = []
for i in range(0,tslen):
    xt_1.append(ts7[i]*a)
```

In [305]:

```
len(xt_1)
```

Out[305]:

100

In [306]:

```
print(xt_1)
```

```
[0.0, -3.227939329679387, 1.102025797526876, 2.8177142167237568, 4.20
7302992558953, 1.8256512906483309, 0.8868844990514051, 0.645154051
5322575, -0.5550472195145643, -0.5572438061137284, -0.68445538395944
03, -0.8350341393332444, -2.119759990572408, -2.876229163281623, -0.74
48901044958366, -0.5562314597325247, 0.9825748872451219, 2.30394549
9378422, 0.33356692198844934, 0.11802358577435895, -2.2731335981339
06, -3.3033114107663577, -2.5324652060611803, -1.4526037689504434, -1.
3676442251548295, 0.1676695963590232, -0.01681727446286377, 0.44167
28100229928, 1.2073471641408322, 1.5797052974816024, 1.71284605345
2834, -1.1853096460776622, -0.8710856687261932, -0.6545876454362776,
-0.06294977451117163, -2.8126665556069668, 0.6567462950958366, -0.68
71661800426307, 2.843650887205368, 1.267354983400854, 1.8555447239
090588, 2.599699660483311, 0.58334380754479, 2.051737040757428, 0.66
1528756667539, -0.13745287781283136, -2.6020648048358357, -3.9459657
46705576, -5.1933639486729675, -3.984072147466468, -1.76436425626823
7, -1.193385948677735, 0.6744480932953915, 1.37088731993395, 0.955491
2630196107, -0.6707883916604485, -3.0514555699628643, -4.99885478731
9127, -2.035410962981217, -1.5429833988446595, 0.5919198462274532, 0.
6576449337295208, 0.9400922939564027, 1.9337142887341388, 0.328379
87907097405, -1.3077757098802307, -1.370978166868814, -1.64577811663
59189, -1.3192396109327598, 0.16625063772146376, 0.2011381457149350
2, -0.15871375169504784, 0.15666137418102596, 0.845990421479898, 1.16
5216655751999, 1.7739987420881436, 1.3134248578063108, 1.902176332
787106, 0.8066152350900279, 1.82334920688509, 0.44549471439180355,
3.745169007632244, 5.562169566853734, 3.1967188532821598, 3.8865904
911902156, 1.8548783773259, -0.5970918639152483, -0.565898509146891
5, -0.5806571131411041, -2.3336620314486236, -2.5891046088707483, -0.2
2857086689409895, 2.112575459791862, -0.38374518983594225, 0.916398
376893791, 2.126290116261287, 2.19431100354718, 2.858397029452354,
1.7957902726267596, 1.1774533926537558]
```

In [312]:

```
diff = []
for i in range(0,tslen):
    diff.append((xt_1[i] - ts7[i])**2)
diff_sum = sum(diff)
```

In [313]:

```
diff_sum
```

Out[313]:

77.8521089757165

In [337]:

```
xt_0 = []
xt_1 = []
xt_2 = []
diff_0 = []
diff_1 = []
diff_2 = []
for i in range(0,tslen):
    xt_0.append(ts7[i]*0)
    xt_1.append(ts7[i]*1)
    xt_2.append(ts7[i]*2)
    diff_0.append((xt_0[i] - ts7[i])**2)
    diff_1.append((xt_1[i] - ts7[i])**2)
    diff_2.append((xt_2[i] - ts7[i])**2)

diff_sum0 = sum(diff_0)
diff_sum1 = sum(diff_1)
diff_sum2 = sum(diff_2)
```

In [339]:

```
print(diff_sum0)
print(diff_sum1)
print(diff_sum2)
```

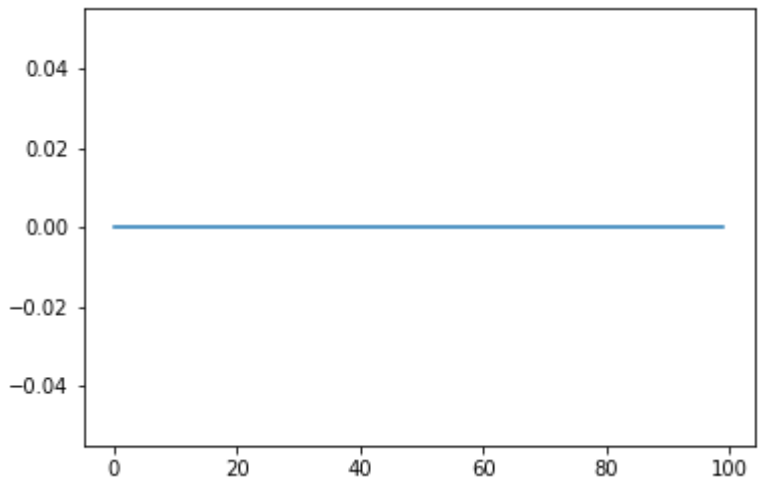
```
817.3353128529164
0.0
817.3353128529164
```

In [346]:

```
plt.plot(xt_0)
```

Out[346]:

[<matplotlib.lines.Line2D at 0x1c14169a20>]

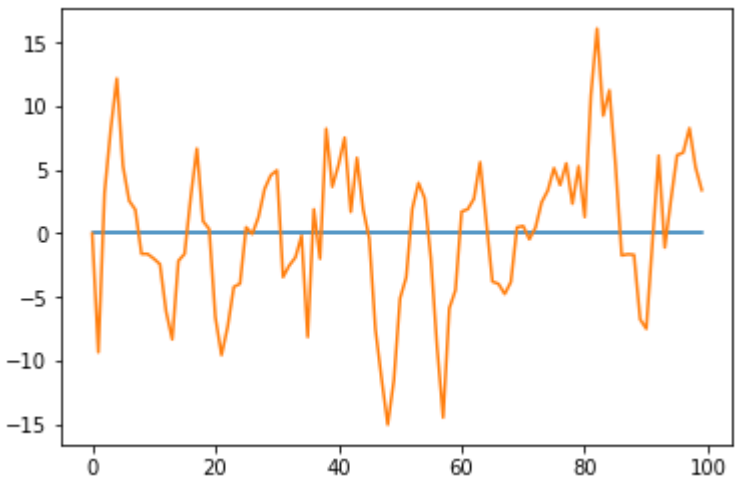


In [345]:

```
plt.plot(xt_2)
```

Out[345]:

[<matplotlib.lines.Line2D at 0x102525358>]



In [385]:

```
ts0= np.loadtxt("tokyo_monthly_average_max_1950_2017.csv",delimiter="," ,usecols=(1,2))
ts0.shape
```

Out[385]:

(816, 2)

In [386]:

```
print(ts0)
```

```
[[ 5.  19.8]
 [ 4.7 21.4]
 [ 7.7 19.1]
 ...
 [16.8 29. ]
 [11.9 21.9]
 [ 6.6 16. ]]
```

In [388]:

```
import statsmodels.api as sm

autocorr = sm.tsa.stattools.acf(ts0[:,1], nlags=25)
```

In [371]:

```
lag = []
for i in range(0,26):
    lag.append(i)
```

In [390]:

```
autocorr.shape
```

Out[390]:

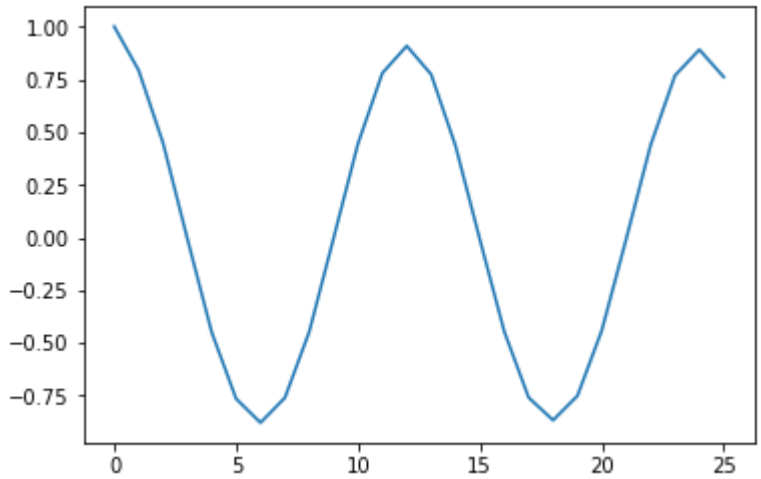
(26,)

In [389]:

```
plt.plot(lag, autocorr)
```

Out[389]:

[<matplotlib.lines.Line2D at 0x1c14535320>]



In [392]:

```
print(np.corrcoef(ts0[12:tslen],ts0[0:tslen-12])[0][1])
```

0.9999999999999998

In [405]:

```
ts0= pd.read_csv("tokyo_monthly_average_max_1950_2017.csv", header=None)
```

In [414]:

```
data = ts0[2]
```

In [467]:

```
data.head()
```

Out[467]:

```
0    19.8
1    21.4
2    19.1
3    24.4
4    29.1
Name: 2, dtype: float64
```

In [449]:

```
diff = data.diff(12)
```

In [453]:

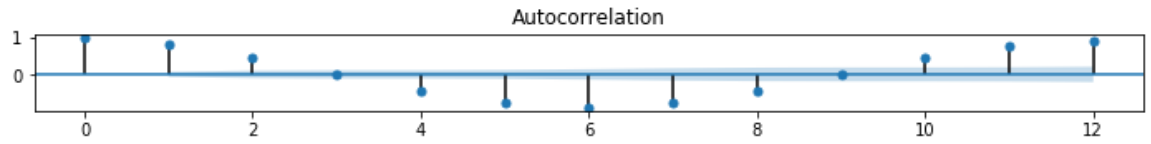
```
diff1 = data.diff()
```

In [421]:

```
ts_acf = sm.tsa.stattools.acf(data, nlags=25)
```

In [439]:

```
fig = plt.figure(figsize=(12,8))
ax1 = fig.add_subplot(816)
fig = sm.graphics.tsa.plot_acf(data, lags=12, ax=ax1)
```

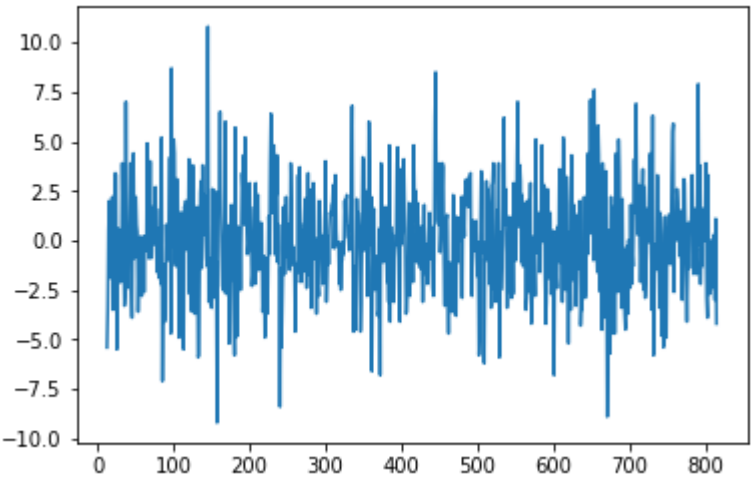


In [450]:

```
plt.plot(diff)
```

Out[450]:

[<matplotlib.lines.Line2D at 0x1c1a991a20>]



In [455]:

```
difflen = len(diff)
```