ARIMA, SARIMAモデル

1. トレンド
2. ARIMA
3. SARIMA

トレンド

1次式で表されるトレンド(trend)が重畳するデータを考える

- データの大まかなトレンド(傾向)を知りたいことがある(気温の上昇, 株価上昇)
- 平均値が変化するから非定常過程(確率論)である。
- ARMAモデルは, 定常過程(確率論)の下で導かれたものである。

考え方

- 観測値y(k)＝信号s (k)＋(ak + b) (kは離散の時間, すなわち, 独立変数で一定の単調増加する数)
- 二つの面から考える。差分法, 1次式のカーブフィッティング(単回帰モデル)

差分法

- 差分　y ' (k) = y(k) – y(k－1)
- 例:y(k) = ak + b　(a, bは定数, kは自然数)とすると, 上記の　y ' (k)　は一定値aをとる。
- z変換として見ると, y ' (k) = (1－z^{－1}) y(k) , これはz変換における微分作用であるから, もとの信号s(k)の周波数成分および位相を変化させることとなる。
- この考え方を基にしたのがARIMAモデルである。

カーブフィッティング法(と, ここでは名付けているだけ)

- 観測値に対して, (ak + b)のカーブフィッティングを行い, これを観測値y(k)から差し引く。
- この値に対してARMAモデルを適用する。これと(ak+b)を重ね合わせる

ARIMA

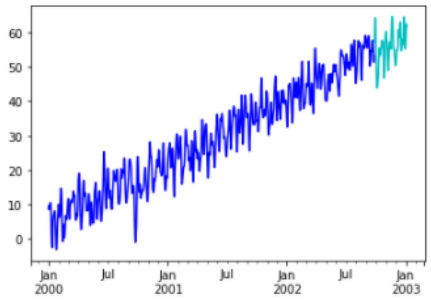
差分波形を見る

$$y'(k) = \left(1 - z^{-1}\right) y(k)$$

- 微分器を通して見るようなものであるから，
  振幅，位相は異なる。
- トレンドは除去できている

```
22 y = y_all[:nobs]      #観測データはy
23 y_test = y_all[nobs:] #予測精度を見るための実データはy_test
24 y.plot(color='b')
25 y_test.plot(color='c')
```

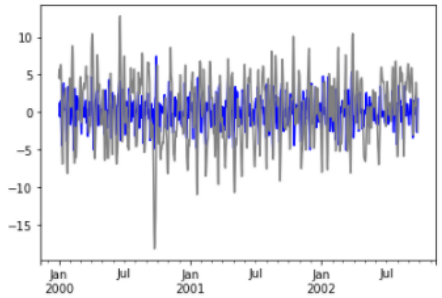
```
: <matplotlib.axes._subplots.AxesSubplot at 0x1a7ae4e6390>
```



yの差分系列を表示（青），元の信号（灰色）とは振幅，位相が異なることがわかる

```
: 1 diff = (y - y.shift()).dropna(axis=0) #先頭のデータは NaNとなるため
  2 diff.plot(color='b')
  3 sig_all[:nobs-1].plot(color='gray')
  4 len(diff)
```

```
: 999
```



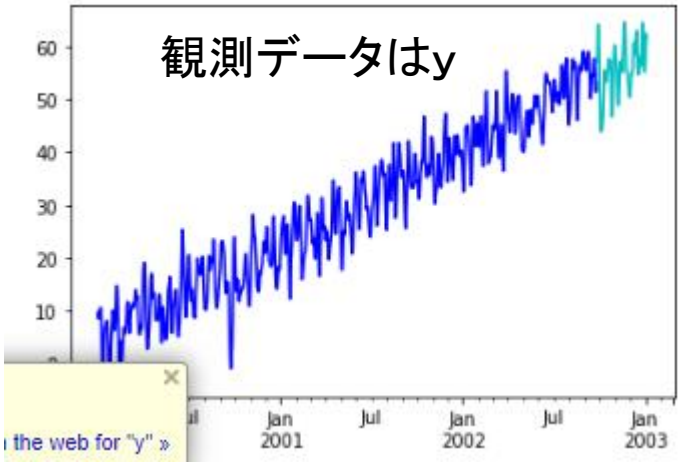
ARIMAモデルの構造

- Autoregressive integrated moving average
- 自己回帰和分移動平均

$$A\left(z^{-1}\right)\left(1 - z^{-1}\right)^{d} y(k) = B\left(z^{-1}\right)u(k)$$

- Box & Jenkinsらの研究で有名
- y ‘ (k) をARMAモデルのy(k)に代入したようなもの
- (1 – z^{-1}) ^d は単位円上にd個の特性根を配置
- これより非定常性を表している
- これは，d階差分と同じ操作である
- d≧2は微分器を2段直列結合と同じであるから，
  雑音を強調することにつながり、さらに位相がぐるっと回るため、2段直列結合は好ましくない。
- 1次式のトレンドは　d＝1で十分である。
- トレンドとARMAを同時に扱える

例：1次トレンド



ARIMA_Identification

```
arima_result = sm.tsa.ARIMA(y, order=(2,1,1)).fit(trend='nc')
fig, ax = plt.subplots(figsize=(12,4))
fig = arima_result.plot_predict(start='2002-07-31', end='2002-10-31', ax=ax)
y_test['2002-09-27':'2002-10-31'].plot(color='m', label='real')
```

例：1次トレンド

結果の評価

- y‘(k)に対するパラメータ同定を行っているのだから，パラメータ値がもとの値を異なっていることは当然である。
- 予測を右図に示す。定性的な傾向は合っている。

```
arima_result = sm.tsa.ARIMA(y, order=(2,1,1)).fit(trend='nc')
print(arima_result.summary())
```

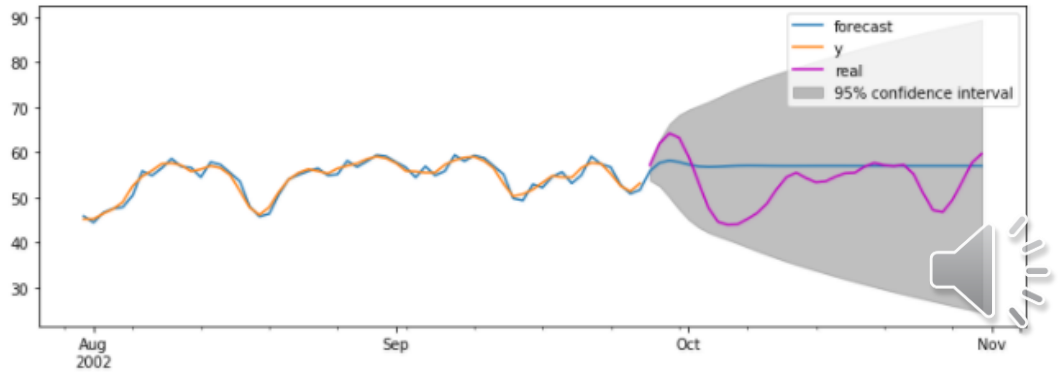
```
                             ARIMA Model Results
==============================================================================
Dep. Variable:                    D.y   No. Observations:                  999
Model:                 ARIMA(2, 1, 1)   Log Likelihood               -1466.858
Method:                       css-mle   S.D. of innovations              1.050
Date:                Fri, 09 Mar 2018   AIC                           2941.717
Time:                        09:04:41   BIC                           2961.344
Sample:                    01-02-2000   HQIC                          2949.177
                         - 09-26-2002
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1.D.y      0.9758      0.057     17.054      0.000       0.864       1.088
ar.L2.D.y     -0.4654      0.049     -9.497      0.000      -0.561      -0.369
ma.L1.D.y      0.3049      0.071      4.302      0.000       0.166       0.444
```

AIC = -2 log_e (L) + 2 (n+m+d+1)

ref: https://en.wikipedia.org/

Autoregressive integrated moving average

```
fig, ax = plt.subplots(figsize=(12,4))
fig = arima_result.plot_predict(start='2002-07-31', end='2002-10-31', ax=ax)
y_test['2002-09-27':'2002-10-31'].plot(color='m', label='real')
legend = ax.legend(loc='upper right')
```



アドバンス：差分法

手順

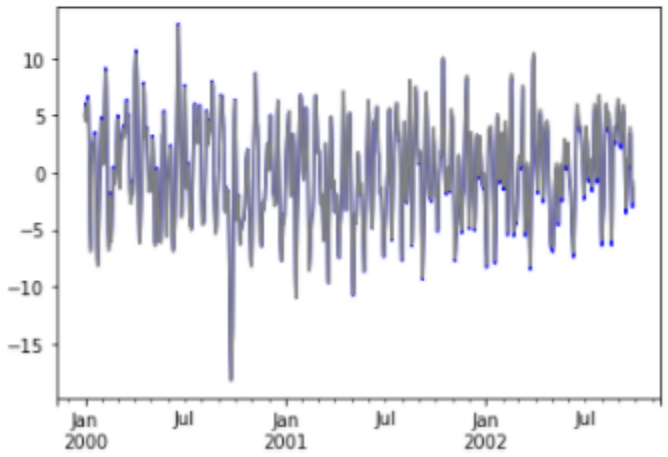
- トレンドを1次式とみなし，これを観測データから除去（左図，もとの信号とよく一致している）
- 除去したデータに対してARMAモデルを用いたパラメータ同定を行う
- 結果，パラメータ値は，真値に近い値を示している
- 予測は，ARMAモデルの予測値に求めた1次式を重畳すればよい

```
dummy_time = np.arange(nobs, dtype='float64')
est_a, est_b = np.polyfit(dummy_time,y,1)
print(est_a, est_b)
est_trend = est_a*np.arange(nobs, dtype='float64') + est_

y_remove = y.sub(est_trend)
y_remove.plot(color='b')
sig_all[:nobs].plot(color='gray')
```

```
0.0508554655868 3.64899960484

<matplotlib.axes._subplots.AxesSubplot at 0x1a7b5d142b0>
```




```
arma_result = sm.tsa.ARMA(y_remove, order=(2,1)).fit(trend='nc')
print(arma_result.summary())
```

```
                              ARMA Model Results
==============================================================================
Dep. Variable:                      y   No. Observations:                 1000
Model:                     ARMA(2, 1)   Log Likelihood               -1379.106
Method:                       css-mle   S.D. of innovations              0.959
Date:                Fri, 09 Mar 2018   AIC                           2766.211
Time:                        09:50:01   BIC                           2785.842
Sample:                    01-01-2000   HQIC                          2773.672
                         - 09-26-2002
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1.y        1.4804      0.025     58.095      0.000       1.430       1.530
ar.L2.y       -0.6804      0.025    -26.833      0.000      -0.730      -0.631
ma.L1.y        0.6089      0.028     21.828      0.000       0.554       0.664
```

最後に、予測値とトレンドを重ねるには、各自でそれ用のスクリプトを考えてください。

季節性変動（seasonal variation）

対象

- 社会科学や経済に関わる統計では，自然現象，社会の制度，慣習などに起因するものがある。例えば，自然現象では四季に基づき社会のエネルギー消費の変動，会社において3月，9月の決算期の企業活動や金融活動などに季節性変動が強く表れる。
- 大抵の場合は，週次，月次，四半期ごと，年次ごとなどの周期をいう。
- この季節性変動を除去してデータ処理を行う（http://www.stat.go.jp/koukou/trivia/careers/career9.htm），または，季節性変動も併せてデータ処理を行う，という考え方がある。

SARIMA

定式化

$$A\left(z^{-1}\right)A^{S}\left(z^{-1}\right)\left(1-z^{-1}\right)^{d}\left(1-z^{-s}\right)^{D}y(k)=B\left(z^{-1}\right)B^{S}\left(z^{-1}\right)u(k)$$

周期性AR項　$$A^{S}\left(z^{-1}\right)=1-a_1^{S}z^{-s}-a_2^{S}z^{-2*s}-\cdots-a_P^{S}z^{-P*s}$$

周期性MA項　$$B^{S}\left(z^{-1}\right)=1+b_1^{S}z^{-s}+b_2^{S}z^{-2*s}+\cdots+a_Q^{S}z^{-Q*s}$$

s：the number of Samples of Period，1周期あたりのサンプル数

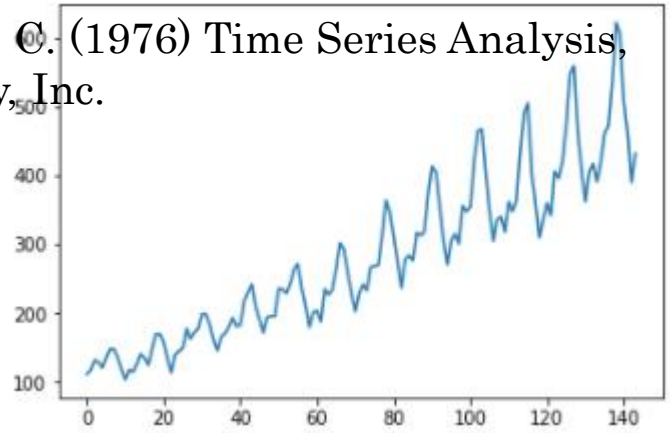
周期性差分項　$$\Delta_S^{D}=\left(1-z^{-s}\right)^{D}$$

例：AirPassengers

SARIMA_AirPassenger

ある航空会社の乗客数，月毎

- 1949－1960年の月毎の集計
- 単位は[千人]
- 原出典：Box, G. E. P., Jenkins, G. M. and Reinsel, G. C. (1976) Time Series Analysis, Forecasting and Control. Third Edition. Holden-Day, Inc.



データの取得法

- 下記の枠内のスクリプトのようにすればよい。
- ネットワークからデータ取得
- データのプロットを右に示す。

SARIMAの適用

- 参考文献：ただし，SARIMAは用いずにARIMAによる結果を示している。この結果と各自でSARIMAを用いた結果との比較には大変有用である。
- Aarshay Jain: A comprehensive beginner's guide to create a Time Series Forecast (with Codes in Python), https://www.analyticsvidhya.com/blog/2016/02/time-series-forecasting-codes-python/

```
import statsmodels.api as sm
url = 'https://sites.google.com/site/datasciencehiro/datasets/AirPassengers.csv'
df = pd.read_csv(url, index_col='Date', parse_dates=True)
df.head()
df['AirPassengers'].plot()
```

例：AirPassengers

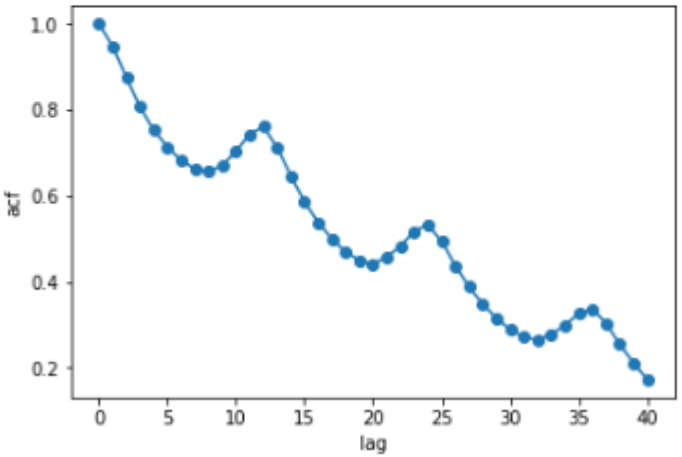
周期性を見るのに自己相関

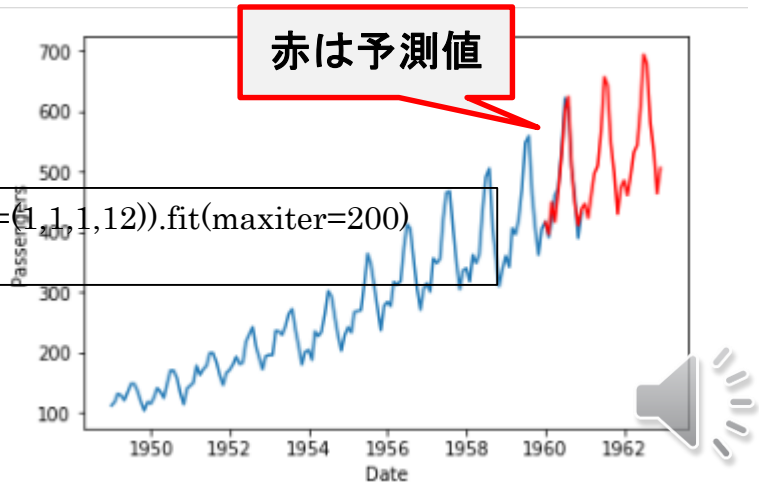
- 右図は自己相関
- 12点毎に周期性－＞1点が1か月より12か月周期が認められる

次数

- order = (p,d,q): ARMA(p,q), 差分の次数d
- seasonal_order = (P, D, Q, s), 季節性用のモデルの次数で, (P,D,Q)は(p,d,q)に類似したもの。sは季節調整に適用する周期を指定する。
- 12点毎の周期性があるので, s=12とする。
- 他の次数は試行錯誤

```
:
1  acf = sm.tsa.stattools.acf(df, nlags=40)
2  #fig, ax = plt.subplots(figsize=(4,4))
3  plt.plot(acf, marker='o')
4  plt.xlabel('lag')
5  plt.ylabel('acf')
6
7  if FLAG_fig: plt.savefig('fig_SARIMAX_Passenger_data_acf.png')
8  plt.show()
```






```
SARIMAX_model = sm.tsa.SARIMAX(df, order=(3,1,2), seasonal_order=(1,1,1,12)).fit(maxiter=200)
print(SARIMAX_model.summary())
```

SARIMA_AirPassenger

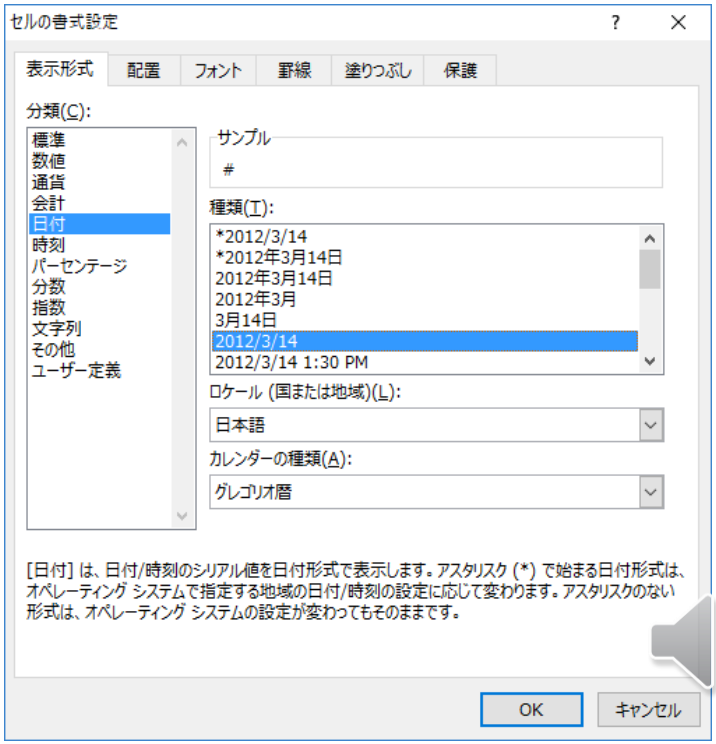
Google Trendの季節性データ

- https://trends.google.co.jp/
  - 検索キーワード，国，期間，カテゴリ，検索，を選択する
  - このキーワードの検索回数が示される。
  - ダウンロードアイコンをクリックすると，CSVファイルがダウンロードされる。
- **CSVファイルの前処理**
  - ファイルを開いたとき，文字化けは，GoogleエクスポートがUTF-8を用いているため。
  - 対処は，CSVファイルを，別のテキストエディタかメモ帳で開く。ここで見た日本語をCSVファイルに移す。
  - 日付のフォーマット
    - A列を全て選んで，CTRLキー＋“1”を押し，右のウィンドウで，

    日付→2012/3/14 を選ぶ。

    年/月/日　をスラッシュで区切る

    このフォーマットでpandasから読むものとする。

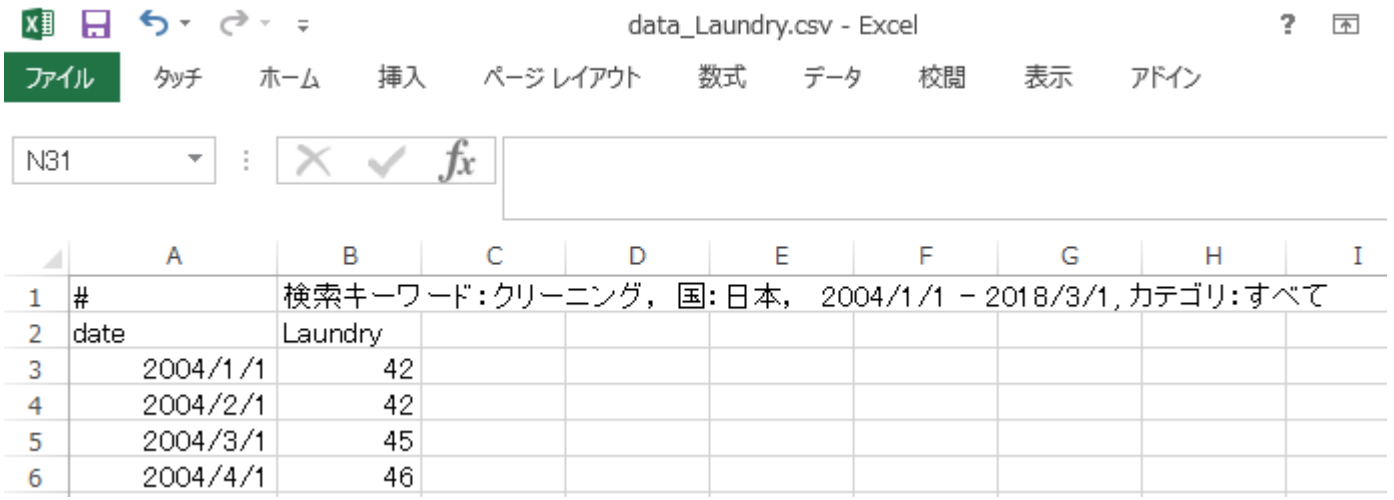


Google Trendの季節性データ

CSVファイル

- 1行目 “#” は，pandasで読むときにコメントとして読み飛ばす。念のため，日本語のコメントはB列から書き始める
- A2セル　date，pandasのtime indexとする
- B2セル　Laundry，適当に名称を与える。英語の方が日本語より扱いやすい。
- 1，2行目の途中で，別の行を入れないように。
- 3行目からデータが記述されていること

data_Laundry.csv

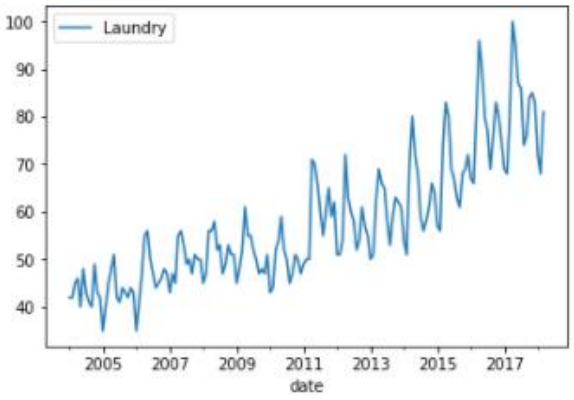


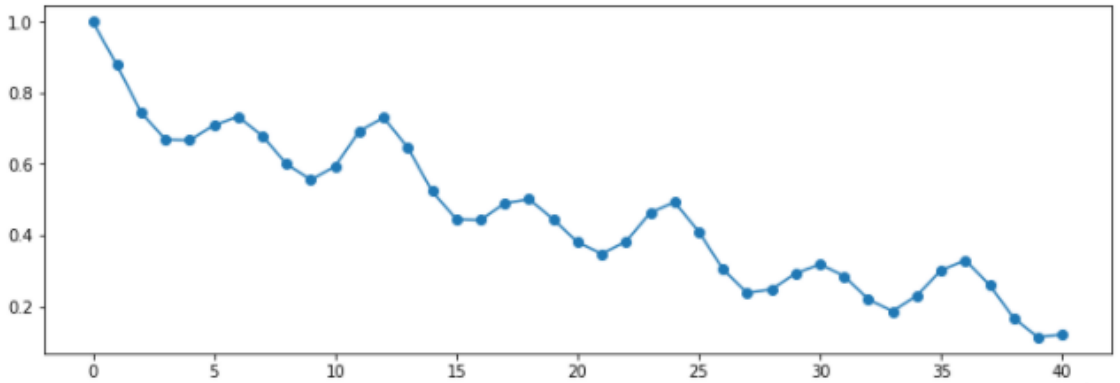
	A	B	C	D	E	F	G	H	I
1	#	検索キーワード：クリーニング，国：日本，　2004/1/1 － 2018/3/1，カテゴリ：すべて							
2	date	Laundry							
3	2004/1/1	42							
4	2004/2/1	42							
5	2004/3/1	45							
6	2004/4/1	46							

例：

検索キーワード：クリーニング

- 左は時系列データ（2004年1月～2018年3月），右は自己相関関数
- 時系列データを見ると，トレンドがあるように見える。
- 季節性（周期性）が4月が最も強く，次に年により9～12月にある（Google Trendで検索画面上から，時系列データにマウスを充てる操作で調べることができる）
- 推察：
  - 衣替え時期を迎えたとき，4月は年度初め，2回目は気候に依存している
  - 年々増えているのは，どう考える？
  - クリーニング業は，これから成長するのか？





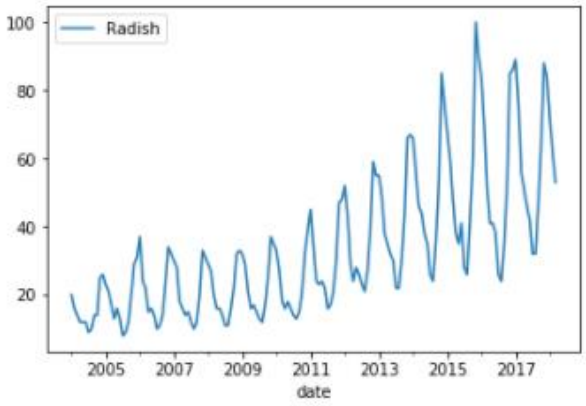
予測を試みてください，データは自前で作成も可

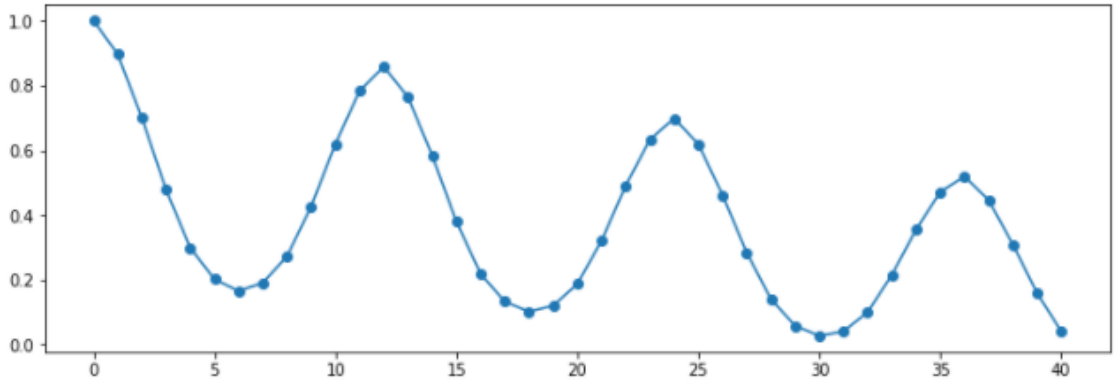
例：

検索キーワード：大根

- 季節性（周期性）が11月にある
- 鍋物，おでんの影響か？
- トレンドがあるのはなぜか？
- 先のクリーニングと相関はある？　因果関係は？
- どうやって調べる？
- もし，需要がこのまま伸びるならば，供給側としてどう対応する。
- ただし，農家は高齢化により，重たい野菜（大根，白菜，キャベツ）は作りにくく，このままでは供給量が維持できないという話がある。

data_Radish.csv





予測を試みてください，データは自前で作成も可

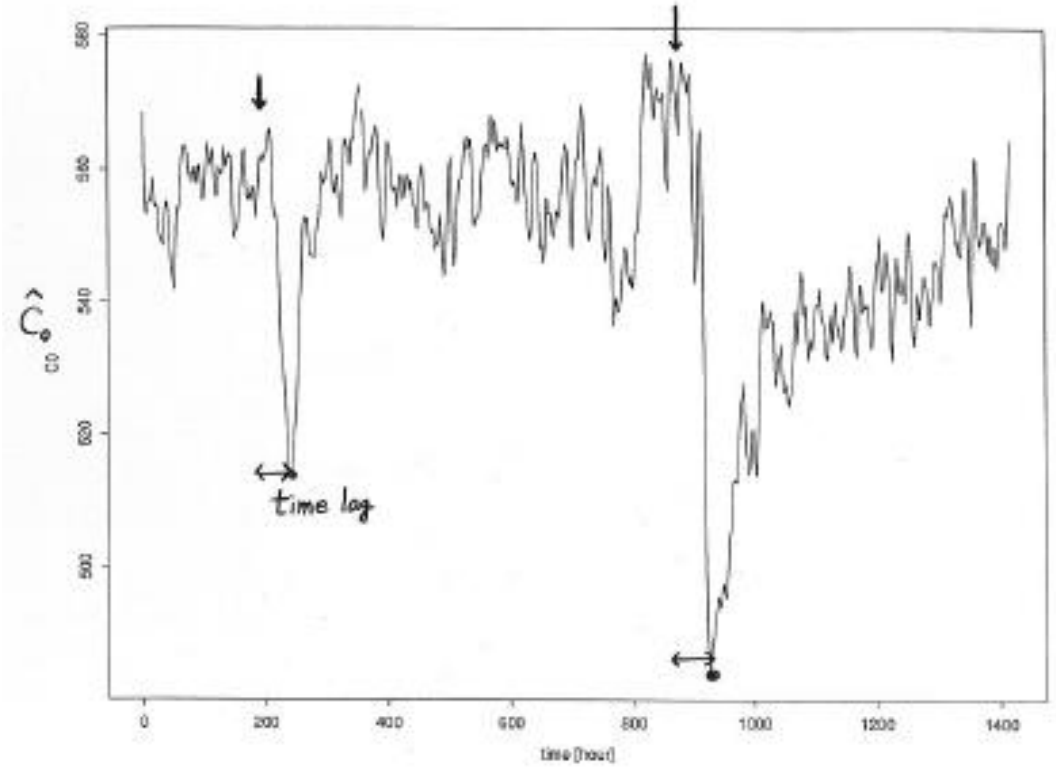
他に季節性を示すデータは？

- **例えば**
  - 衣食住関連（マフラー，ダウンジャケット，初カツオ，除湿，乾燥など）
  - 病気（花粉症，インフルエンザ）
  - 旅行関連（花見，温泉，Inboundなど）
  - 他に？
- **これらが予測できたら，どのようなメリットがある？**

備考：トレンドの考え方

区間で考えるトレンド

- 次のような時系列データのトレンドはどう考える？
- 全データのトレンドを取ると、例えば直線でトレンドを考えると、傾き0になるかもしれない。
- 下記のように、様相が変わるときには、区間に分けてトレンドを別々に求める考え方がある。例えば、time:0 – 200, 200 – 900, 950 - 1400



グラフの引用：https://www.ism.ac.jp/~higuchi/index_e/papers/Kouza-TSA-Higuchi.pdf

[7.1] 赤池弘次：情報量規範 AIC とは何か，数理科学, No.153, 5/11 (1976)

[7.2] 赤池，中川：ダイナミカルシステムの統計的解析と制御，サイエンス社 (1972)

SARIMAの参考文献

1. Seasonal ARIMA Models, Eberly College of Science, PennState https://onlinecourses.science.psu.edu/stat510/node/50, Chap.4.1, ペンシルベニア州立大学（同校はペン・ステート（Penn State）という愛称で親しまれている） Eberly 科学学部, Robert E. Eberly氏の名前に由来する。
2. SARIMAX: Introduction http://www.statsmodels.org/dev/examples/notebooks/generated/statespace_sarimax_stata.html
3. 初出は「Box, G. E. P., Jenkins, G. M. and Reinsel, G. C. (1976) Time Series Analysis, Forecasting and Control. Third Edition. Holden-Day. Series G.」
4. **Pythonによる時系列分析の基礎** https://logics-of-blue.com/python-time-series-analysis/