

放電加工機の時系列解析（その5）

1.1 はじめに

放電加工機の時系列解析に取り組みます。

時系列解析は、以下の4ステップから構成されます。

① ディスクリプション（description）

時系列の特徴を把握します。定常性の確認、成分分解等を行います。

② モデリング（modeling）

時系列モデルを構成し、パラメータを推定します。

③ 予測（prediction）

現在までの情報から今後の変動を予測します。

④ 信号抽出

必要な信号や情報を取り出す。（異常検知とか）

今回の報告書は、上記の **モデリング（modeling）** と **予測（prediction）** についての取組内容です。モデルについて、代表的な手法として、**自己回帰モデル** と **状態空間モデル**がありますが、今回は、**自己回帰モデル**についての報告です。

1.2 どのようなデータを扱ったか

放電加工機は、加工中に障害が発生すると自己診断して3段階のレベルに分け、レベルに応じたメッセージを表示します。内容は以下です。

- ・エラーメッセージ：続行不可能な障害が発生したときに表示し、動作を中断する。
- ・ハルトメッセージ：再開可能な障害が発生したときに表示し、一時停止する。
- ・コメントメッセージ：続行可能な障害で発生し、注意を促す。

また、メッセージの記録は、USBによりCSVデータとして取り出すことが可能です。

レベルに対応してデータ処理できるように、3段階のレベルについて障害のレベルが大きいと、点数は大きくなるように点数を設けました。

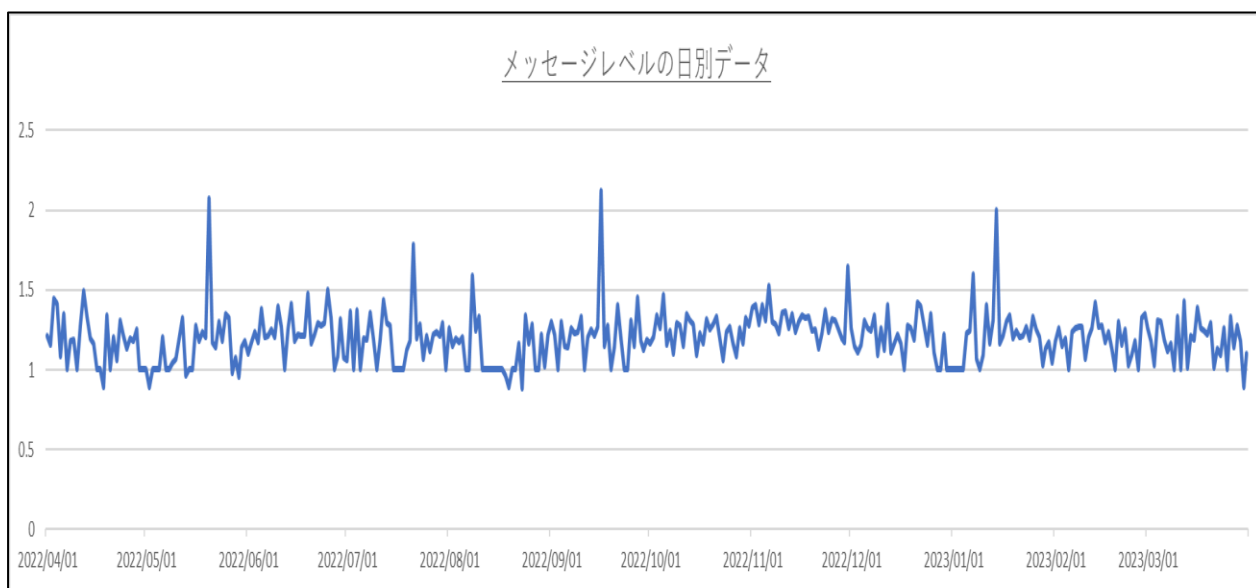
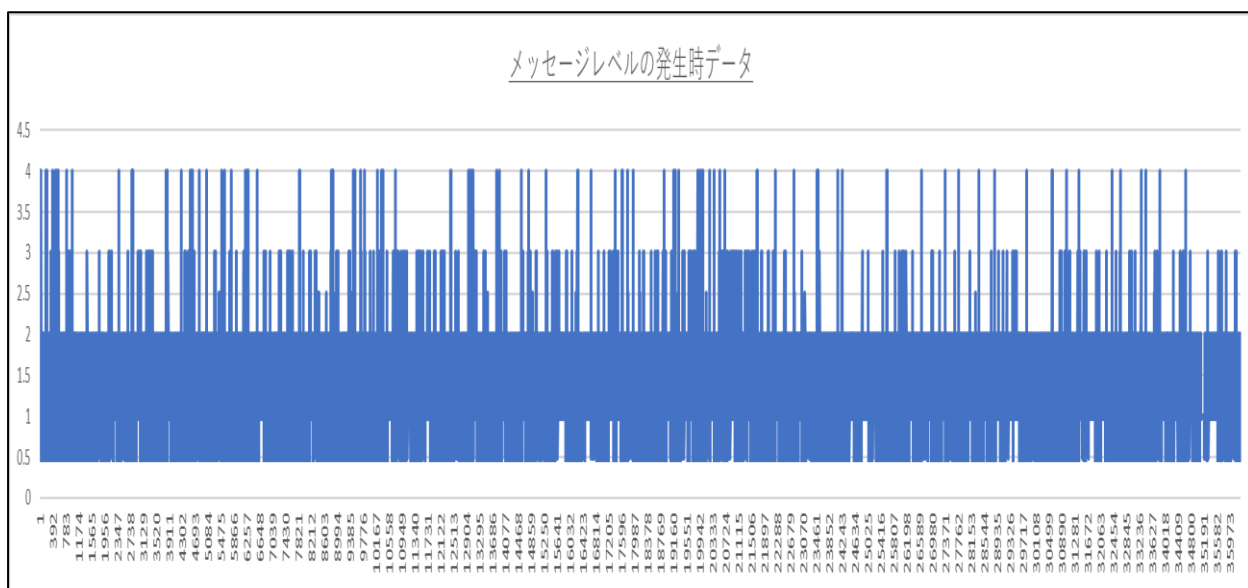
これらは、メッセージが発生した時の時間毎のデータで、点過程データと呼ばれるもので、時系列データではありません。故に、一日分の平均値を取って時系列データとして取り出す

ようしました。本データを以下に示します。2022年4月から2023年3月までのデータです。

上段が発生時毎のデータ（点過程データ）で、1年で37000回発生しています。

1点の発生は、一定時間の間隔ではありません。

下段は点過程データを、一日毎に平均して、平均したデータです。点の間隔は1日毎です。時系列データと言えます。



1.3 プログラムについて

Python の statsmodels.tsa.ar_model

というライブラリー内の以下の関数で計算しています。

予測 → forecast , predict

最適な次数の選択 → ar_select_order

詳細は以下で確認ください。

<https://www.statsmodels.org/stable/api.html#statsmodels-api>

本報告書にはプログラムは記載していません。

～参考文献、WEB～

時系列解析 -自己回帰モデル・状態空間モデル・異常検知- 島田直希 著 共立出版(株)

・ARIMA 系モデルで予測する方法

<https://www.salesanalytics.co.jp/datascience/datascience087/>

2.1 自己回帰型モデルについて、

AR、ARIMA、SARIMA の各モデルについて、説明します。

2.1.1 AR モデル(Auto-Regressive)

以前の報告書で取り上げたモデルです。自己回帰モデルという意味になり、基本形という
か、このモデルに、成分を追加して、ARIMA、SARIMA モデルが構成されます。

AR モデルは、被説明変数の過去の値に説明力があるものと考え、ある変数の過去の値か
ら現在や将来の値を予測します。

一次 AR(p)モデル (AR(1)) を数式にすると以下のようになります。

$$Y_t = \alpha + \phi_1 Y_{t-1} + \epsilon_t$$

時系列データは、過去の値に依存する場合があります。自己相関があるということです。

(今日の売上が 1 日前 (1 期前) と似たような値を取るケースです。)

AR (Auto-Regressive) でモデル化するとき、何期前 (p) までのデータを用いるかを考えな
くてはなりません。それを自己回帰パラメータ p と言います。

2.2.2 ARIMA モデル(Auto-Regressive Integrated Moving Average)

このモデルは、次の 3 つの成分から構成されます。

- ・ AR component : 自己回帰成分, (上項で説明した内容です)

以下の 2 項目を追加します。

- ・ I (Integrated) component : 和分成分

時系列データは、明らかに定常でない場合があります。

I (Integrated) 成分で定常化を目指します。モデルを構築するというよりも、前処理と
表現したほうがいいです。この場合の前処理とは、過去との差分を取り階差系列を作るこ
とです。例えば、今日の売上と昨日の売上の差分を求め、その差分を新たな時系列データ
とする、ということです。その新たなデータを階差系列と呼びます。上昇傾向や下降傾向
などのトレンドがなくなるまで、この前処理を繰り返します。

多くの場合、1, 2 回で十分です。この繰り返し数を、差分の階数 d で表現します。

・ MA (Moving Average) component : 移動平均成分

MA モデル (移動平均モデル) は、ある変数とその期待値を中心にして推移するものと仮定し、現在の値は過去の攪乱項から説明できると考えたモデルです。

一次 MA(q)モデル (MA(1)) の計算式は以下のようになります。

$$Y_t = \mu + \varepsilon_t + \theta_1 \varepsilon_{t-1}$$

この数式からわかるように、期待値は一定であることから、 Y_t は現在の攪乱項と過去の攪乱項から決定されることとなります。

自己相関は、MA (Moving Average)でもモデル化することができます。

どこまでの過去を考慮するのかを考えなくてはなりません。

それを移動平均パラメータ q と言います。(移動平均は、残差の移動平均です)

パラメータについておさらいします。

時系列データは、AR (Auto-Regressive)だけで自己相関を十分にモデル化できません。

要は、以下の3つのパラメータがあります。

自己回帰パラメータ p

差分の階数 d

移動平均パラメータ q

どのようにパラメータ設定したかを明示的に示すために、ARIMA(p,d,q)という形式で表現します。これらのパラメータは非負の整数値 (0,1,2,...) の値を取ります。パラメータの値を「次数」という表現することもあります。

問題は、この(p,d,q)の次数の値をどう設定するのか? ということになります。

2.2.3 SARIMA モデル (Seasonal ARIMA) 季節自己回帰和分移動平均

ARIMA は、季節性を効果的に捉えることができないという問題があります。

この問題を解決するのが、SARIMA モデルです。

SARIMA モデルのパラメータは、以下のようになります。

非季節性パラメータ (p,d,q)

季節性パラメータ (P, D, Q, m)

非季節性パラメータ (p,d,q) は、ARIMA モデルのパラメータと同じです。

ARIMA モデルと大きく異なるのは、以下の季節性パラメータ (P, D, Q, m)の存在です。

m (Seasonal Period) : 季節性の周期

P (Seasonal AR component) : 季節性の AR component の次数

D (Seasonal I component) : 季節性の I component の次数

Q (Seasonal MA Component) : 季節性の MA component の次数

m は季節性の周期です。

例えば、月単位の時系列データに対し 12 ヶ月周期がある場合、m=12 となります。

P は、季節性の AR component の次数です。

例えば、月単位の時系列データに対し m=12 かつ P=2 の場合、12 ヶ月前と 24 ヶ月前のデータの値を考慮するということです。

D は、季節性の I component の次数です。

例えば、月単位の時系列データに対し m=12 かつ D=1 の場合、12 ヶ月前とデータとの差分をとるということです。

Q は、季節性の MA component の次数です。

例えば、月単位の時系列データに対し m=12 かつ Q=2 の場合、12 ヶ月前と 24 ヶ月前の残差を考慮するということです。

どのようにパラメータ設定したかを明示的に示すために、SARIMA(p,d,q)(P, D, Q, m) もしくは SARIMA(p,d,q)(P,D,Q)[m]という形式で表現します。

2.2.4 モデルの表し方

ARIMA 系のモデルは、以下の表し方になります。

AR モデル : SARIMA(p,0,0)(0,0,0)[0]

MA モデル : SARIMA(0,0,q)(0,0,0)[0]

ARMA モデル : SARIMA(p,0,q)(0,0,0)[0]

ARIMA モデル : SARIMA(p,d,q)(0,0,0)[0]

SARIMA モデル : SARIMA(p,d,q)(P,D,Q)[m]

2.3 予測の結果

パラメータ(p,d,q) (P,D,Q)[m]の設定を説明しながら、AR モデル、ARIMA モデル、SARIMA モデルの予測を行います。

2.3.1 差分の階数の設定

ADF 検定 (augmented Dicker-Fuller test) により、時系列が、定常過程になるように、階数を設定します。

プログラムは、statsmodels.tsa.stattools の adfuller 関数で計算します。

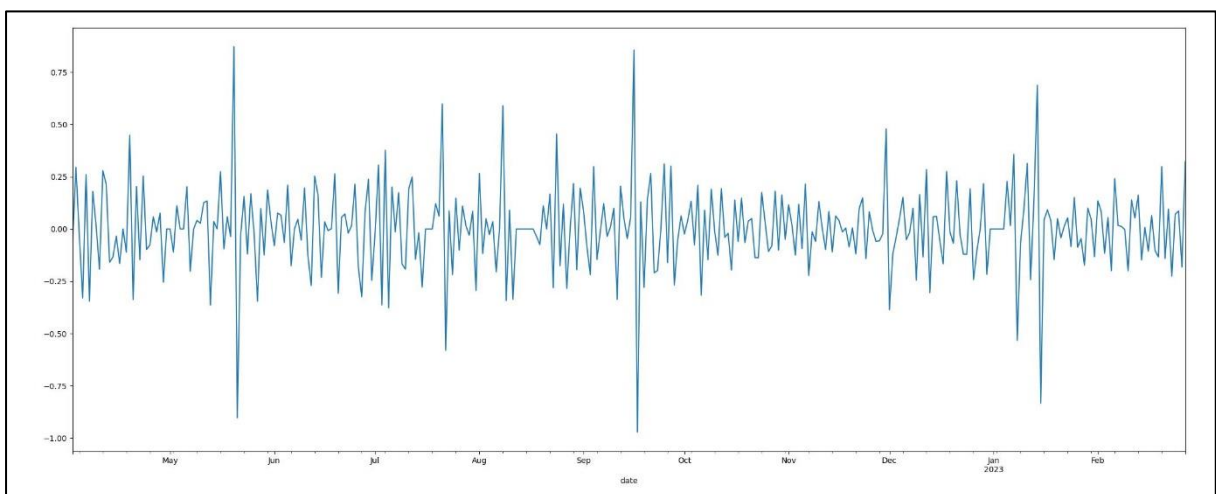
戻り値の中の p-value で判定し、0.05 以下であることが、定常過程の条件です。

結果は、以下で、差分を 1 階数に設定することとなります。d=1 となります。

	データ原系列				1差分したデータ			
	ctt	ct	c	nc	ctt	ct	c	nc
p-value	6.83E-16	5.57E-16	6.85E-18	0.560122	2.16E-16	1.49E-16	6.45E-19	4.76E-19

ctt, ct, c, nc は AR モデルのトレンド項、定数項の有無による 違いです。

1 差分した時系列データが以下です。(原データは、2 ページの下のグラフです。)



2.3.2 AR モデル

AR モデルは、 $(p,d,q) = (p,0,0)$ ですので、 p の設定を行います。

(ここでは、差分化したモデルを原モデルとして扱います)

プログラムは、`statsmodels.tsa.stattools` の `fit` 関数で AIC(Akaike information criterion) を、 p を 1 から 20 の値を入力して、各 AIC 値を比較します。

最小値を示す p 値を選びます。結果は以下でした。

$p = 1$ aic : -183.29025618292812

----- (省略)

$p = 5$ aic : -234.90874481796163

$p = 6$ aic : -242.09425915418785 <= 最小値

$p = 7$ aic : -240.91014470541194

$p = 8$ aic : -241.34164587993274

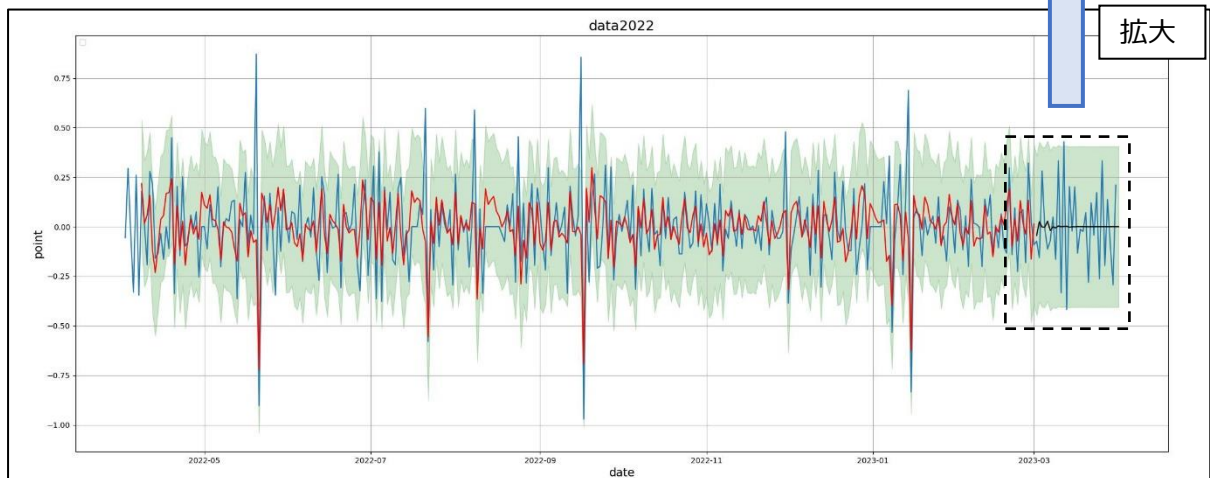
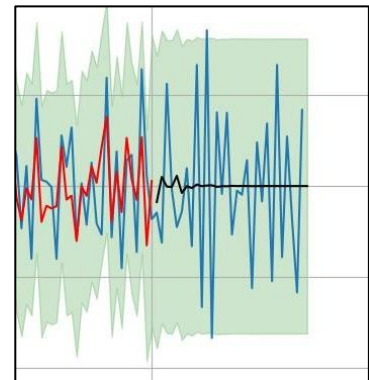
$p = 9$ aic : -240.3340226300007

----- (省略)

$P = 20$ aic : -229.56623910754445

$p=6$ となりました。

AutoReg 関数で計算し、グラフにしました。



2.3.3 ARIMA モデル

AR モデルは、 (p,d,q) については、 $d=1$ は検証済として、 p と q の設定を行います。

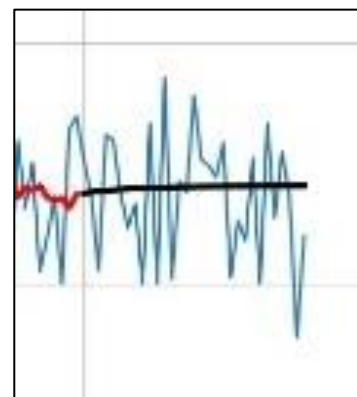
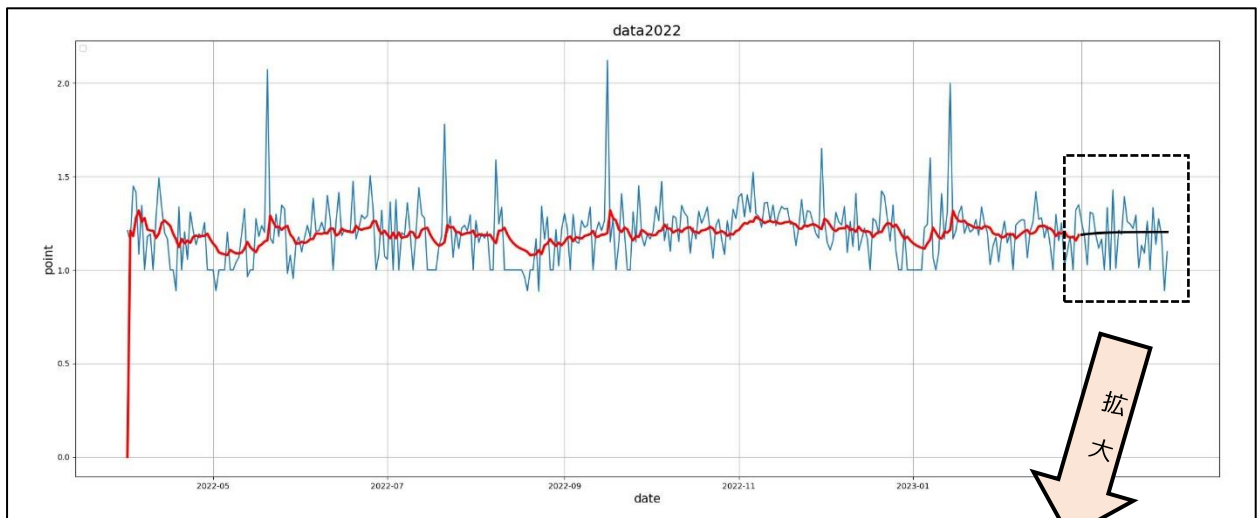
p, q の算出は、statsmodels.api の `arma_order_select_ic` 関数で計算します。

結果は以下です。数値は、AIC の値で、最小値が最適値となります。

		p		
		0	1	2
q	0	-97.757	-267.946	-268.107
	1	-186.517	-268.648	-271.728
	2	-208.273	-269.833	-265.809
	3	-224.675	-267.920	-268.001
	4	-232.792	-266.742	-263.925

$(p,d,q) = (1,2,1)$ となり、この値を入力して作成したグラフが以下です。

簡単なプログラム (`ARIMA(y, order=(1, 1, 2)).fit()`) で作成可能です。



2.3.3 SARIMA モデル

パラメータ(p,d,q) (P,D,Q)[m]の設定ですが、ARIMA モデルのような計算してくれる関数はありません。パラメータを変えて、繰り返し処理を行い、AIC が最小になる設定を選びます。

AIC の算出は、`sm.tsa.SARIMAX().fit()` の関数で可能です。

パラメータを多数にすると、計算時間がかかり、回数の限界になることもあります。

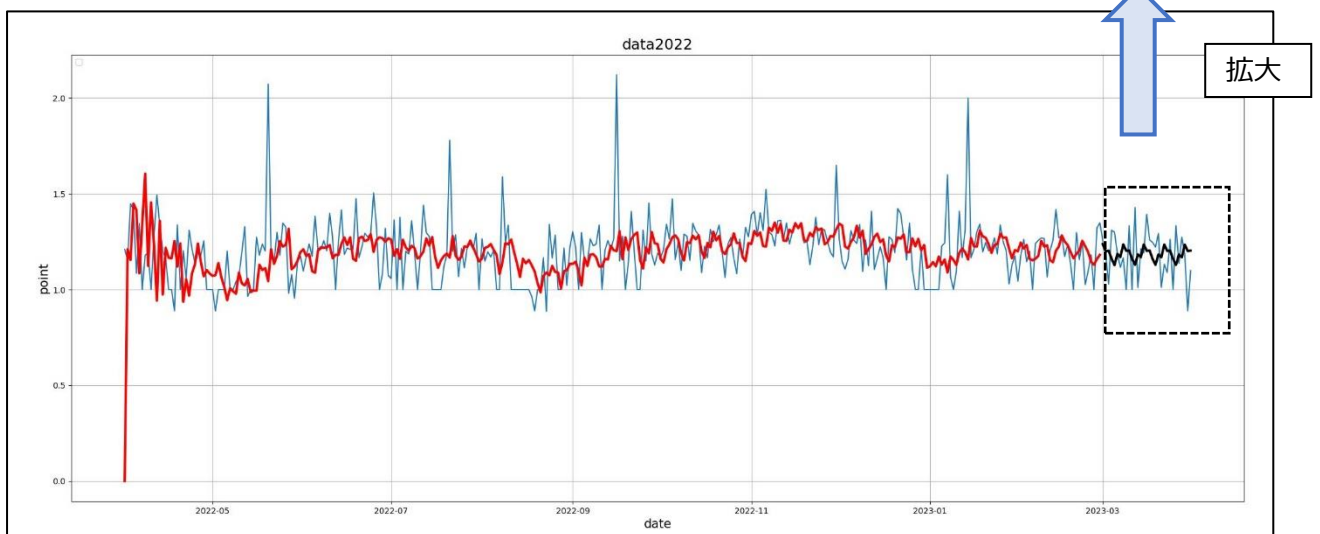
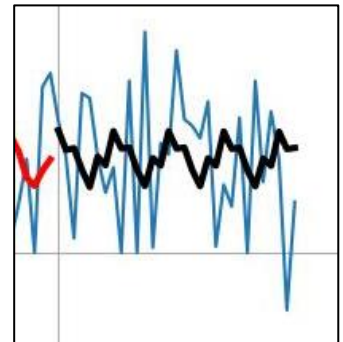
今回は、p: 1,2,3 d:0,1 q: 0,1,2,3 P,D,Q : 0,1 と変化させ

[m] は7として計算しました。

結果は、以下でした。(AIC の小さい順番に5個表示させました)

```
40 order=(1,1,1), season=(0,0,0) -267.313822
56 order=(1,1,3), season=(0,0,0) -266.249487
104 order=(2,1,1), season=(0,0,0) -265.740003
168 order=(3,1,1), season=(0,0,0) -265.031735
24 order=(1,0,3), season=(0,0,0) -264.747002
```

season=(0,0,0) は、ARIMA モデルと同じですので、
order=(1,1,1), season=(1,1,1) の設定でグラフを作成しました。



3 まとめ

SARIMA モデルにより、予測がそれらしいグラフとなりました。

ただ 訓練データでの推定値は、状態空間モデルと比較すると、観測値から乖離しています。

以下のことは、見えてきたので、さらに、検討していきます。

- ・ 時系列データは原データではなく、差分データでないと、定常性は確保できない。
- ・ 季節性を配慮することが、予測精度は上がる。

以上 です。