

# 陽性患者数の変動分析

## Poisson モデルの母数の追跡

N. Murata

2020年9月2日

## 1 はじめに

陽性患者数は日々変動するが、検査機関の稼働状況により、その数は大きく変動する。このため、日報の値をそのまま用いるのではなく、週ごとあるいは移動平均などによる集計が多くの場合用いられる。また、増減を定量的に評価・予測するためには、SIR 系モデルに基づく実効再生産数の推定値などが用いられる。

本稿では、観測データが Poisson 分布に従うとして、その母数の変動を状態空間モデルで記述することにより、母数の変動と週単位の変動を分解し、実質的な患者数の増減の様子を取り出すを試みる。<sup>1</sup>

図示やモデルの構築に際しては以下の package を用いる。

```
1 ## パッケージの読み込み
2 library(tidyverse)
3 library(scales) # 年月日表示
4 library(plotly)
5 library(zoo)     # 移動平均のため
6 library(KFAS)    # 状態空間モデルの構成
```

<sup>1</sup> 状態空間モデルを扱うための package はいくつかあるが、本稿ではモデルの記述が明解であった **KFAS** を用いる。

## 2 データの視覚化

東京都が公開している COVID-19 のデータは、陽性患者ごとの属性情報なので、これを日毎の陽性患者数として集計する。<sup>2</sup>

<sup>2</sup> 東京都の陽性患者データ

```
1 ## データの取得と整理 (東京都)
2 myData <-
3   read.csv("https://stopcovid19.metro.tokyo.lg.jp/data/13")
4   ↳ 0001_tokyo_covid19_patients.csv
5   ↳ %>%
6   dplyr::select(公表_年月日) %>%
7   dplyr::rename(date=公表_年月日) %>%
8   dplyr::transmute(date=as.Date(date)) %>%
9   dplyr::group_by(date) %>%
10  dplyr::summarize(patients = n()) # 陽性者数
```

陽性患者数の推移を図示すると図 1 のようになる。各日の陽性患者数を点で、7 日および 14 日の移動平均をそれぞれ緑と橙の実線で表示している。

```
1 ## データの視覚化
2 p <-
3   ggplot(data = myData,
4     mapping = aes(x = date,
5                   y = patients)) +
6   geom_point() +
7   geom_line(data = myData %>% # 7日移動平均
```

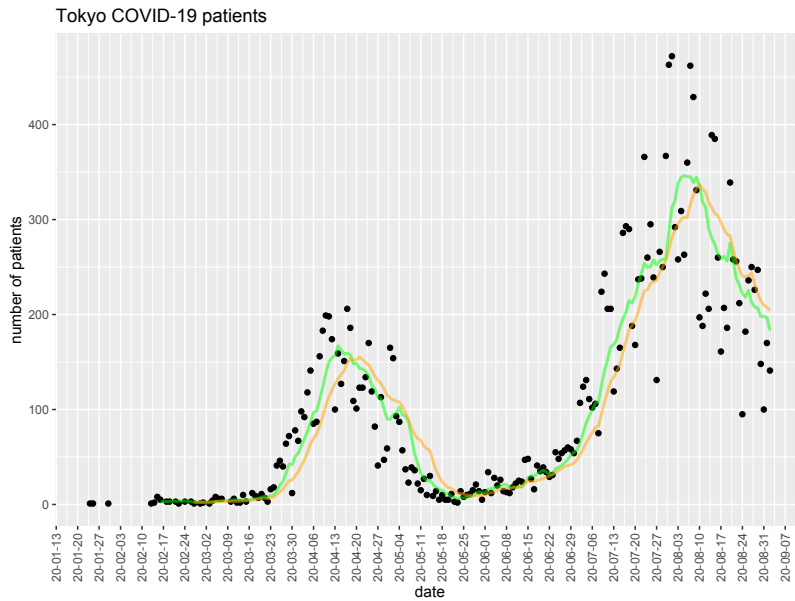


図 1: 東京都の陽性患者数の推移. 緑は 7 日移動平均, 橙は 14 日移動平均を表す.

```

8     dplyr::mutate(patients =
9                   zoo::rollmean(x = patients,
10                                k = 7,
11                                align = "right",
12                                fill = NA)),
13     color = "green", alpha = 0.5, size = 1,
14     na.rm = TRUE) +
15   geom_line(data = myData %>% # 14 日移動平均
16             dplyr::mutate(patients =
17                             zoo::rollmean(x = patients,
18                                              k = 14,
19                                              align = "right",
20                                              fill = NA)),
21             color = "orange", alpha = 0.5, size = 1,
22             na.rm = TRUE) +
23   scale_x_date(labels = date_format("%y-%m-%d"), # 年月日表示
24               breaks = date_breaks("1 week")) + # 週毎
25   theme(axis.text.x = element_text(angle = 90,
26                                     vjust = 0.5, hjust=1)) +
27   labs(title = "Tokyo COVID-19 patients",
28        x = "date",
29        y = "number of patients")
30   print(p) # グラフ出力
31   ggplotly() # plotly 表示 (browser)

```

### 3 モデルの構成

時刻  $t$  (年月日) の陽性患者数を  $X_t$  とし,  $X_t$  は母数  $\lambda_t$  の Poisson 分布に従うとする.

$$X_t \sim \mathcal{P}_o(\lambda_t),$$

$$\Pr(X_t = k) = \frac{\lambda_t^k e^{-\lambda_t}}{k!}$$

このとき母数  $\lambda_t$  の対数は以下の成分に分解されると仮定する.

$$\log \lambda_t = \mu_t + c_t$$

成分  $\mu_t$  は 2 次のトレンド成分で<sup>3</sup>

$$\begin{aligned}\mu_{t+1} &= \mu_t + \nu_t + \xi_t, & \xi_t &\sim \mathcal{N}(0, Q_{\text{level},t}) \\ \nu_{t+1} &= \nu_t + \zeta_t, & \zeta_t &\sim \mathcal{N}(0, Q_{\text{slope},t})\end{aligned}$$

に従うものとする。変数  $\mu_t$  はトレンド成分の水準 (level) を表し、変数  $\nu_t$  は勾配 (slope) を表しトレンド成分の増減を決定する。

成分  $c_t$  は周期成分で<sup>4</sup>

$$\begin{aligned}c_{t+1} &= c_t \cos \tau + c_t^* \sin \tau + \omega_t, \\ c_{t+1}^* &= -c_t \sin \tau + c_t^* \cos \tau + \omega_t^*, & \omega_t, \omega_t^* &\sim \mathcal{N}(0, Q_{\text{cycle},t})\end{aligned}$$

に従うものとする。ただし、 $\tau$  は定数で、周期を  $s$  としたとき  $\tau = 2\pi/s$  で与えられる。検査機関の稼働状況は週日・休日に依存すると考えられるので、以降  $s = 7$  を用いる。

KFAS package を用いて上記のモデルを構成する。トレンド成分においては、水準の変動の分散を  $Q_{\text{level}} = 0$  とし、勾配の変動の分散のみ未知母数  $Q_{\text{slope}} = NA$  として推定する。また、周期成分の変動の分散は  $Q_{\text{cycle}} = 0.05$  としている。<sup>5</sup>

まとめると、モデルは以下ようになる。

$$\begin{aligned}X_t &\sim \mathcal{P}_o(\lambda_t), \\ \log \lambda_t &= \mu_t + c_t \\ \mu_{t+1} &= \mu_t + \nu_t, \\ \nu_{t+1} &= \nu_t + \zeta_t, & \zeta_t &\sim \mathcal{N}(0, Q_{\text{slope},t}) \\ c_{t+1} &= c_t \cos \tau + c_t^* \sin \tau + \omega_t, \\ c_{t+1}^* &= -c_t \sin \tau + c_t^* \cos \tau + \omega_t^*, & \omega_t, \omega_t^* &\sim \mathcal{N}(0, Q_{\text{cycle},t})\end{aligned}$$

<sup>3</sup> 母数  $\lambda_t$  の時間変化を捉えるために、2 次の系を仮定して増減の推定を行う。

<sup>4</sup> 季節成分としてもよいが、KFAS では季節成分の推定方法として周期成分を半周期で平滑化したものと周期分の dummy 変数を用いたものが用意されており、前者を推奨しているの、ここでは簡単な周期成分を用いた。

<sup>5</sup> 補遺参照のこと。

```
1 ## 状態空間モデルの構成
2 myModel <-
3   SSMModel(data = myData,
4             formula = patients ~ # 目的変数
5               -1 + # 定数項を持たない
6               SSMtrend(degree = 2, #トレンド成分の定義
7                         Q = list(0, NA)) +
8               SSMcycle(period = 7, # 周期成分の定義
9                         Q = 0.05),
10             distribution = "poisson") # 目的変数は Poisson 分布
11 stateName <- colnames(myModel$Z) # 状態変数の名称
```

## 4 推定

モデルの母数推定 (トレンド成分の分散) および状態推定 (状態の smoothing / filtering) は以下で行うことができる。

```
1 ## 母数推定
2 fit <- fitSSM(myModel,
3               inits = 0, # 初期値
4               method = "BFGS") # 最適化法
5 ## 状態推定 (推定した母数を用いる)
6 out <- KFS(fit$model,
7            filtering = c("state", "mean"),
8            smoothing = c("state", "mean"))
```

推定した状態を、信頼区間付きで表示すると以下ようになる。

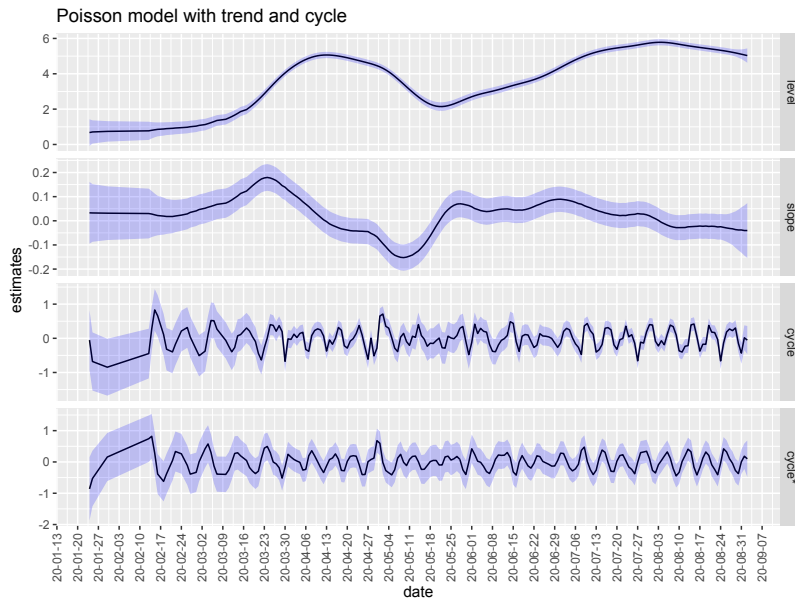


図 2: 状態空間モデルによる各成分の推定.

```

1  alpha <- 0.05 # 有意水準 (信頼区間の準備)
2  zq <- qnorm(1-alpha/2) # 正規分布の (1-alpha/2) 分位点
3  tmp <- # 必要な状態変数を取り出す
4      cbind(myData["date"],
5            out$alphahat, # 状態変数の平均
6            t(sqrt(apply(out$V,3,diag)))) # 標準偏差
7  names(tmp)[-1] <- # 名前を付与
8      paste(rep(c("value", "sd"), each = length(stateName)),
9            rep(stateName, times = 2),
10           sep = "_")
11  myState <- # tidy data 化
12      tmp %>%
13      tidyr::pivot_longer(
14          ~date,
15          names_to = c(".value", "name"),
16          names_pattern = "(.*)_(.*)") %>%
17      dplyr::mutate_at("name", ~factor(., levels = unique(.)))
18  p <-
19      ggplot(data = myState, group = name,
20            mapping = aes(x = date,
21                          y = value)) +
22      geom_line() +
23      geom_ribbon(mapping = aes(ymin = value-zq*sd,
24                              ymax = value+zq*sd),
25                fill = "blue", alpha = 0.2) +
26      facet_grid(name ~ ., scale = "free_y") +
27      scale_x_date(labels = date_format("%y-%m-%d"),
28                  breaks = date_breaks("1 week")) +
29      theme(axis.text.x = element_text(angle = 90,
30                                        vjust = 0.5, hjust=1)) +
31      labs(title = "Poisson model with trend and cycle",
32           x = "date",
33           y = "estimates")
34  print(p)
35  ggplotly()

```

さらに、状態推定をもとに周期成分を取り除いた結果を実データに重ねると以下ようになる。

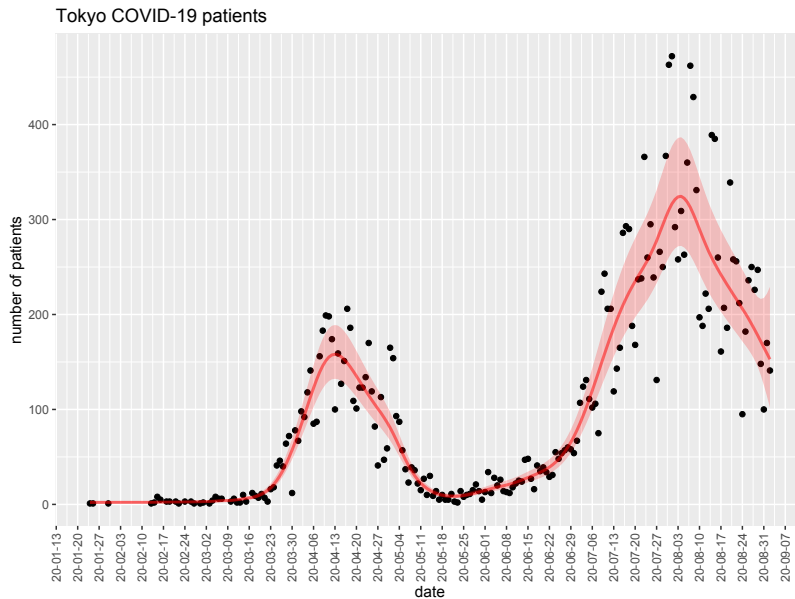


図 3: 状態空間モデルによる平均の推定.

```

1  ## 状態空間モデルにもとづく平均の推定
2  tmp <- KFAS::signal(out, states = "trend")
3  tmpa <- tmp$signal
4  tmpb <- sqrt(tmp$variance[1,1])
5  p <-
6    ggplot(data = myData %>%
7      dplyr::mutate(mean = exp(tmpa),
8                    lwr = exp(tmpa - zq*tmpb),
9                    upr = exp(tmpa + zq*tmpb)),
10     mapping = aes(x = date,
11                   y = patients)) +
12    geom_point() +
13    geom_line(mapping = aes(y = mean),
14              color = "red", alpha = 0.5, size = 1) +
15    geom_ribbon(mapping = aes(ymin = lwr, ymax = upr),
16              fill = "red", alpha = 0.2) +
17    scale_x_date(labels = date_format("%y-%m-%d"),
18                 breaks = date_breaks("1 week")) +
19    theme(axis.text.x = element_text(angle = 90,
20                                      vjust = 0.5, hjust=1)) +
21    labs(title = "Tokyo COVID-19 patients",
22          x = "date",
23          y = "number of patients")
24  print(p) # グラフ出力
25  ggplotly() # plotly 表示 (browser)

```

## 5 おわりに

移動平均でも窓幅を適切に選べば増減の傾向を捉えることは可能であるが、状態空間モデルでは背後にある確率的な力学系を仮定した上で、その分布を調べることができることにある。

8 月初旬から Poisson 分布の母数は減少を始めているが、信頼区間を考慮に入れると 8 月中旬から減少の速度が遅くなっていると考えられる。

## 6 補遺

### 6.1 状態空間モデルによる推定

状態空間モデルによる推定の仕組みを見るために、陽性患者数の対数値について簡略化した状態空間モデルを考えて説明する。

まず、図 4 に陽性患者数の対数値  $Y_t$  を示す。

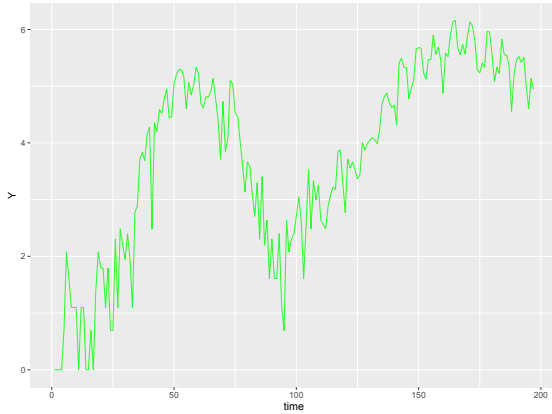


図 4: 陽性患者数の対数値の系列.

```
1 ## 簡単な設定で実験
2 Y <- log(myData$patients) # 陽性患者数の対数
3 tmp <- tibble(t = 1:length(Y),
4               Y = Y)
5 p <-
6   ggplot(data = tmp,
7           mapping = aes(x = t,
8                           y = Y)) +
9   geom_line(color = "green", alpha = 0.8) +
10  xlab(label = "time")
11 print(p)
```

観測値  $Y_t$  の状態空間モデルとして以下を考える。状態空間は 2 次のトレンド成分を考える。すなわち、勾配 ( $\nu$ , slope) がランダムウォークで連続に変化し、水準 ( $\mu$ , level) は単位時間あたり勾配分の増減を行う。<sup>6</sup> 観測値は水準に正規雑音が重畳されたものを考える。まとめると以下のようになる。

$$\begin{aligned} Y_t &= \mu_t + \epsilon_t, & \epsilon_t &\sim \mathcal{N}(0, H) \\ \mu_{t+1} &= \mu_t + \nu_t, \\ \nu_{t+1} &= \nu_t + \zeta_t, & \zeta_t &\sim \mathcal{N}(0, Q_{\text{slope}}) \end{aligned}$$

このモデルの母数を

$$\theta = (H, Q_{\text{slope}})$$

で表し、観測系列および状態系列をそれぞれ

$$\begin{aligned} Y &= \{Y_t; t = 1, \dots, T\} \\ Z &= \{(\mu_t, \nu_t); t = 1, \dots, T\} \end{aligned}$$

で表す。また、観測系列および状態系列の分布とその密度関数をそれぞれ

$$\begin{aligned} P_{\theta}^{Y|Z}, & \quad p^{Y|Z}(y|Z; \theta) & (\text{条件付}) \\ P_{\theta}^Z, & \quad p^Z(z; \theta) \end{aligned}$$

と書く。<sup>7</sup>

<sup>6</sup> 勾配を一定とし、水準が勾配と雑音の影響で変化するというモデルでも表すことができるが、この問題では勾配の変化を抽出したいので、勾配の方に確率変動を入れた形で記述し、可能な勾配系列をランダムウォーク (連続時間なら Wiener 過程 (Gauss 過程の一種)。微分はできないが連続な系列を網羅的に表す) でモデル化している。

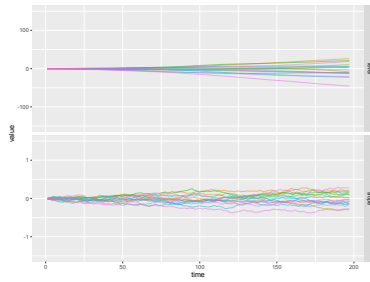
<sup>7</sup> このモデルでは母数  $\theta$  はそれぞれの分

母数  $\theta$  が未知の場合、その推定は観測系列から定義される尤度関数を最大化すればよい。すなわち、観測系列と状態系列の同時密度を状態系列について周辺すればよいので

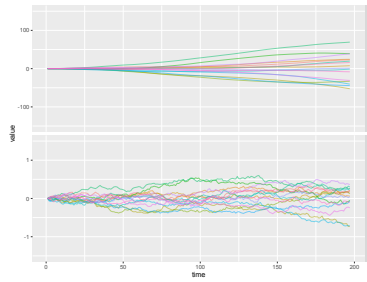
$$\begin{aligned}\ell(\theta) &= \ell(\theta; Y = y) \text{ (観測系列を } y \text{ とする)} \\ &= \int p^{Y|Z}(y|z; \theta) p^Z(z; \theta) dz \\ &= \mathbb{E}_{Z \sim P_\theta^Z} [p^{Y|Z}(y|Z; \theta)]\end{aligned}$$

となる。条件付密度  $p^{Y|Z}(y|z; \theta)$  は状態系列  $z$  の水準の項  $\mu$  に加法的雑音が重畳して観測系列  $y$  が出現する確率を表しているの、その差  $\|\mu - y\|$  が小さいほど大きな値となり、水準 (状態  $z$  の一部) と  $y$  の類似度と捉えることができる。したがって平均で書いた式は母数  $\theta$  を持つ状態系列の分布 (事前分布) からサンプリングされた状態系列と観測系列の類似度の平均と解釈される。直感的には観測系列  $y$  と似た  $\mu$  の系列を多く出力するように母数  $\theta$  を選ばばよい。

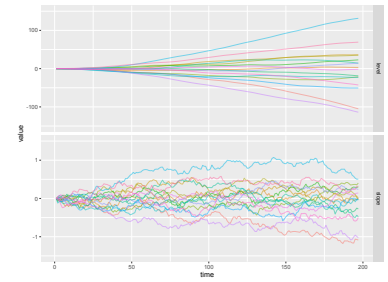
以下の実験では、勾配成分 (slope) の雑音の分散  $Q_{\text{slope}}$  (ランダムウォークの歩幅) としては、本文で推定された  $\hat{Q}_{\text{slope}}$  を用い、観測雑音の分散は  $H = 0.1$  に固定するが、その前に母数  $Q_{\text{slope}}$  の違いによってどのような状態系列が出現するか確認しておく。図 5 は雑音の分散  $Q$  を  $\lambda \times \hat{Q}_{\text{slope}}$ ,  $\lambda = 0.3, 1, 3$  とした事前分布から、それぞれ 16 個の状態変数系列  $\mu_t$  (水準; level) および  $\nu_t$  (勾配; slope) をサンプリングした結果を示す。



(a)  $\lambda = 0.3$



(b)  $\lambda = 1$



(c)  $\lambda = 3$

図 5: 事前分布のパラメタの違いによる状態変数の挙動。

```
1  ## いくつかのモデルで状態の事前分布を確認
2  Qs <- fit$model$Q[2,2,1] # 推定された Qslope を利用
3  tmpc <- tibble(t = 1,
4                 var = rep(c("level", "slope"), each = 2),
5                 name = "V1",
6                 value = c(150, -150, 1.5, -1.5))
7  for(lambda in c(0.3, 1, 3)){
8    model <- # モデルを作成
9      SSMModel(formula = Y ~ # 目的変数
10               -1 + # 定数項を持たない
11               SSMtrend(degree = 2, #トレンド成分の定義
12                        Q = list(0, lambda*Qs)), H = 0.1)
13    ## モデルにもとづいて状態を生成 (状態系列の事前分布)
14    tmpa <- simulateSSM(model,
15                       nsim = 16,
16                       conditional=FALSE)
17    tmpb <-
18      rbind(tibble(t = 1:length(Y),
19                  var = "level",
20                  as_tibble(tmpa[,1,])),
21            tibble(t = 1:length(Y),
22                  var = "slope",
```

```

23         as_tibble(tmpa[,2,])) %>%
24         tidyr::pivot_longer(-c(t,var))
25     p <-
26         ggplot(data = tmpb, group = var,
27               mapping = aes(x = t,
28                             y = value,
29                             color = name)) +
30         geom_blank(data = tmpc) +
31         geom_line(alpha = 0.5) +
32         facet_grid(var ~ ., scale = "free_y") +
33         theme(legend.position = "none") +
34         xlab(label = "time")
35     print(p)
36 }

```

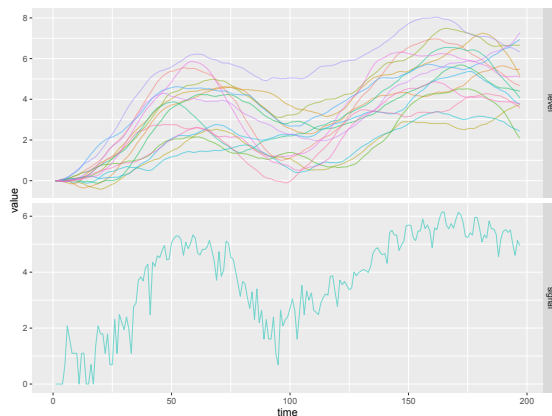
分散を大きくするほど、勾配のばらつきが大きくなり、その結果水準の増減幅も大きくなることがわかる。観測値  $Y_t$  の範囲は  $[0, 6]$  程度なので、それを記述する水準系列も同程度の範囲を動くように事前分布のパラメタが適切に選択される必要がある。以下では  $\lambda = 1$  のモデルを用いる。

事後分布の密度関数は、

$$p(z|y) = \frac{p^{Y|Z}(y|z; \theta) p^Z(z; \theta)}{p^Y(y; \theta)} \\ \propto p^{Y|Z}(y|z; \theta) p^Z(z; \theta)$$

となるので、直感的には事前分布に従う系列の中から観測値と似通った系列に条件付密度で重み付けして選び出す操作と考えることができる。<sup>8</sup>

図7では、事前分布からサンプリングされた多数 (30000 個) の水準系列の中から、観測値との残差平方和 (残差の分散) が小さいもの 16 個を抽出した結果を示す。この程度のサンプリング数の中でもある程度似通った系列があることがわかるが、同時にサンプリングにもとづく方法は非常に効率が悪いこともわかる。



<sup>8</sup> 愚直には、事前分布に従ってサンプリングされた状態系列から、条件付密度を使って棄却法で選び出せばよいが、相当効率は悪い。

図7: 事前分布からサンプリングされた変数  $\mu_t$  (level) のうち、観測データ  $Y_t$  と似たものを抽出した結果。

```

1  ## モデルの設定 (以降で使うモデル)
2  Qs <- fit$model$Q[2,2,1]
3  model <-
4      SSMModel(formula = Y ~ # 目的変数
5                -1 + # 定数項を持たない
6                SSMtrend(degree = 2, # トレンド成分の定義
7                          Q = list(0,Qs)), H = 0.1)
8  ## 事前分布からサンプリングした level から観測値 Y と相関の高いものを取り出す
9  tmpa <- simulateSSM(model,
10                      nsim = 30000,

```



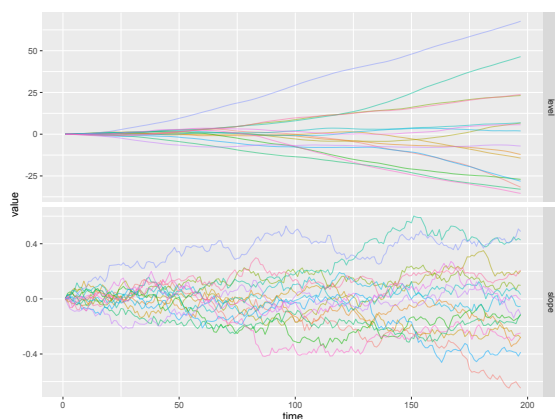
```

11         conditional = FALSE)
12 tmpb <- rbind(
13   tibble(t = 1:length(Y),
14     var = "level",
15     as_tibble(
16       tmpa[,1,rank(apply(tmpa[,1,],2,
17         function(x){var(x-Y)}))<17])) %>%
18   tidyr::pivot_longer(-c(t,var)),
19   tibble(t = 1:length(Y),
20     var = "signal",
21     name = "V17",
22     value=Y))
23 ## 図示
24 p <-
25   ggplot(data = tmpb, group = var,
26     mapping = aes(x = t,
27       y = value,
28       color = name)) +
29   geom_line(alpha = 0.5) +
30   facet_grid(var ~ ., scale = "free_y") +
31   theme(legend.position = "none") +
32   xlab(label = "time")
33 print(p)

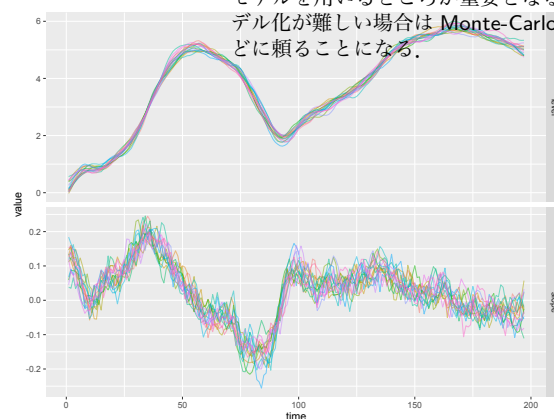
```

実際には、上記のように Bayes の定理にもとづいて事後分布を計算し、そこから適切な方法で効率良くサンプリングを行っている。この結果を図 8 に示す。<sup>9</sup>

<sup>9</sup> 事後分布の計算が容易に行えるようなモデルを用いるところが重要となる。モデル化が難しい場合は Monte-Carlo 法などに頼ることになる。



(a) 事前分布からサンプリングされた状態変数



(b) 事後分布からサンプリングされた状態変数

図 8: 状態空間モデルの推定。

```

1 ## モデルにもとづいて状態を生成 (状態系列の事前分布)
2 prior <- simulateSSM(model,
3   nsim = 16,
4   conditional = FALSE)
5 ## 観測データで条件付けて状態を生成 (状態系列の事後分布)
6 postr <- simulateSSM(model,
7   nsim = 16,
8   conditional = TRUE)
9 ## 図示
10 for(s in c("prior","postr")){
11   tmpa <- eval(parse(text=s))
12   tmpb <-
13     rbind(tibble(t = 1:length(Y),
14       var = "level",
15       as_tibble(tmpa[,1,])),
16     tibble(t = 1:length(Y),

```

```

17         var = "slope",
18         as_tibble(tmpa[,2,])) %>%
19     tidyr::pivot_longer(~c(t,var))
20 p <-
21     ggplot(data = tmpb, group = var,
22           mapping = aes(x = t,
23                         y = value,
24                         color = name)) +
25     geom_line(alpha = 0.5) +
26     facet_grid(var ~ ., scale = "free_y") +
27     theme(legend.position = "none") +
28     xlab(label = "time")
29     print(p)
30 }

```

## 6.2 周期成分のモデルについて

周期成分の変動の分散は既知としてモデルの構築を行ったが、これも本来は推定すべきである。単に  $Q_{\text{cycle}} = NA$  としても最適化してくれない。<sup>10</sup>

適当な範囲で  $Q_{\text{cycle}}$  の値を変えながら、解の尤度を検証した結果  $Q_{\text{cycle}} = 0.05$  としている。

<sup>10</sup> 最適化の設定を適切にするか、母数の更新関数をおそらく設定する必要があるが、検証できていない。

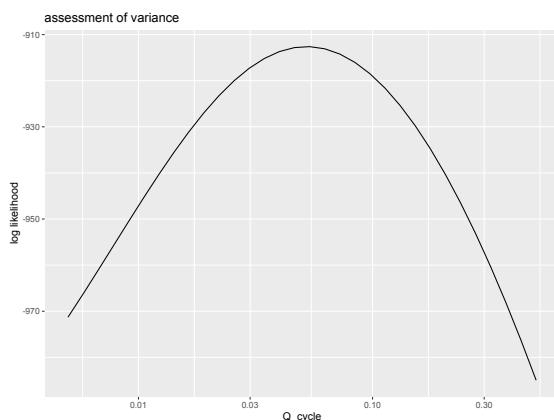


図 10:  $Q_{\text{cycle}}$  の検討について。

```

1  ## 周期成分の変動の分散の検討
2  Qc <- 10^seq(0,-2,length=32)/2
3  lL <- double(length(Qc))
4  for(i in 1:length(Qc)) {
5      tmp <-
6          SSMModel(data = myData,
7                    formula = patients ~
8                        -1 +
9                        SSMtrend(degree = 2,
10                               Q = list(0,NA)) +
11                        SSMcycle(period = 7,
12                               Q = Qc[i]), # 変更
13                    distribution = "poisson")
14      lL[i] <- logLik(fitSSM(tmp,
15                          inits = 0,
16                          method = "BFGS"))$model)
17  }
18  p <-
19      ggplot(data = data.frame(Q=Qc, logLik=lL),
20            mapping = aes(x = Q, y = logLik)) +
21      geom_line() +
22      scale_x_log10() +

```

```
23     labs(title = "assessment of variance",
24           x = "Q_cycle",
25           y = "log likelihood")
26   print(p)
27   print(Qc[which.max(1L)])
```

## 6.3 他のデータについて

全国のデータは厚生労働省から得られる。こちらは単純な集計データなので、例えば以下のようにすれば同様に利用できる。

```
1  ## データの取得と整理 (厚生労働省)
2  myData <- read.csv("https://www.mhlw.go.jp/content/pcr_posi_
   ↳ tive_daily.csv")
3  names(myData) <- c("date", "patients")
4  myData$date <- as.Date(myData$date)
```