陽性患者数の変動分析

Poisson モデルの母数の追跡

N. Murata

*2020*年 *8*月 *29*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ことを試みる．[1]

図示やモデルの構築に際しては以下の package を用いる．

[1] 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．

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

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．[2]

[2] 東京都の陽性患者データ

```
## データの取得と整理（東京都）
myData <-
    read.csv("https://stopcovid19.metro.tokyo.lg.jp/data/13」
    ↪  0001_tokyo_covid19_patients.csv")
    ↪  %>%
    dplyr::select(公表_年月日) %>%
    dplyr::rename(date=公表_年月日) %>%
    dplyr::transmute(date=as.Date(date)) %>%
    dplyr::group_by(date) %>%
    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．

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

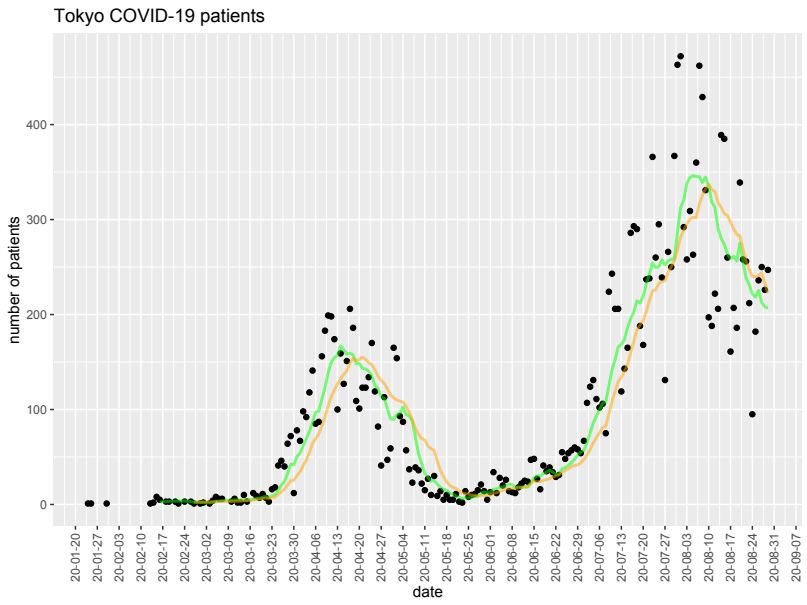


図 1: 東京都の陽性患者数の推移．緑は 7 日移動平均，橙は 14 日移動平均を表す．

```
                dplyr::mutate(patients =
                                zoo::rollmean(x = patients,
                                              k = 7,
                                              align = "right",
                                              fill = NA)),
              color = "green", alpha = 0.5, size = 1,
              na.rm = TRUE) +
    geom_line(data = myData %>% # 14 日移動平均
                dplyr::mutate(patients =
                                zoo::rollmean(x = patients,
                                              k = 14,
                                              align = "right",
                                              fill = NA)),
              color = "orange", alpha = 0.5, size = 1,
              na.rm = TRUE) +
    scale_x_date(labels = date_format("%y-%m-%d"), # 年月日表示
                 breaks = date_breaks("1 week")) + # 週毎
    theme(axis.text.x = element_text(angle = 90,
                                     vjust = 0.5, hjust=1)) +
    labs(title = "Tokyo COVID-19 patients",
         x = "date",
         y = "number of patients")
  print(p) # グラフ出力
  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 λ_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 λ_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 μ_t は 2 次のトレンド成分で [3]

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

に従うものとする．変数 μ_t はトレンド成分の水準 (level) を表し，変数 ν_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 [4]

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

に従うものとする．ただし，τ 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．[5]

まとめると，モデルは以下のようになる．

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

```
## 状態空間モデルの構成
myModel <-
  SSModel(data = myData,
          formula = patients ~ # 目的変数
            -1 + # 定数項を持たない
            SSMtrend(degree = 2, # トレンド成分の定義
                     Q = list(0,NA)) +
            SSMcycle(period = 7, # 周期成分の定義
                     Q = 0.05),
          distribution = "poisson") # 目的変数は Poisson 分布
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．

```
## 母数推定
fit <- fitSSM(myModel,
                    inits = 0, # 初期値
                    method = "BFGS") # 最適化法
## 状態推定 (推定した母数を用いる)
out <- KFS(fit$model,
                  filtering = c("state","mean"),
                  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のようになる．

[3] 母数 λ_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．

[4] 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で，ここでは簡単な周期成分を用いた．

[5] 補遺参照のこと．

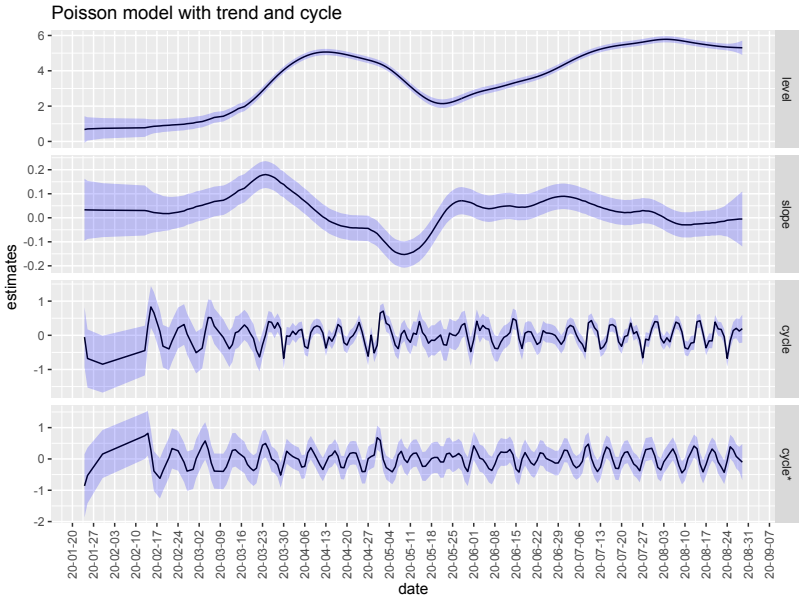


図 2: 状態空間モデルによる各成分の推定.

```
alpha <- 0.05 # 有意水準 (信頼区間の準備)
zq <- qnorm(1-alpha/2) # 正規分布の (1-alpha/2) 分位点
tmp <- # 必要な状態変数を取り出す
  cbind(myData["date"],
        out$alphahat, # 状態変数の平均
        t(sqrt(apply(out$V,3,diag)))) # 標準偏差
names(tmp)[-1] <- # 名前を付与
  paste(rep(c("value","sd"), each = length(stateName)),
        rep(stateName, times = 2),
        sep = "_")
myState <- # tidy data 化
  tmp %>%
  tidyr::pivot_longer(
    -date,
    names_to = c(".value", "name"),
    names_pattern = "(.*)_(.*)") %>%
  dplyr::mutate_at("name", ~factor(., levels = unique(.)))
p <-
  ggplot(data = myState, group = name,
         mapping = aes(x = date,
                       y = value)) +
  geom_line() +
  geom_ribbon(mapping = aes(ymin = value-zq*sd,
                            ymax = value+zq*sd),
              fill = "blue", alpha = 0.2) +
  facet_grid(name ~ ., scale = "free_y") +
  scale_x_date(labels = date_format("%y-%m-%d"),
               breaks = date_breaks("1 week")) +
  theme(axis.text.x = element_text(angle = 90,
                                   vjust = 0.5, hjust=1)) +
  labs(title = "Poisson model with trend and cycle",
       x = "date",
       y = "estimates")
print(p)
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のようになる.

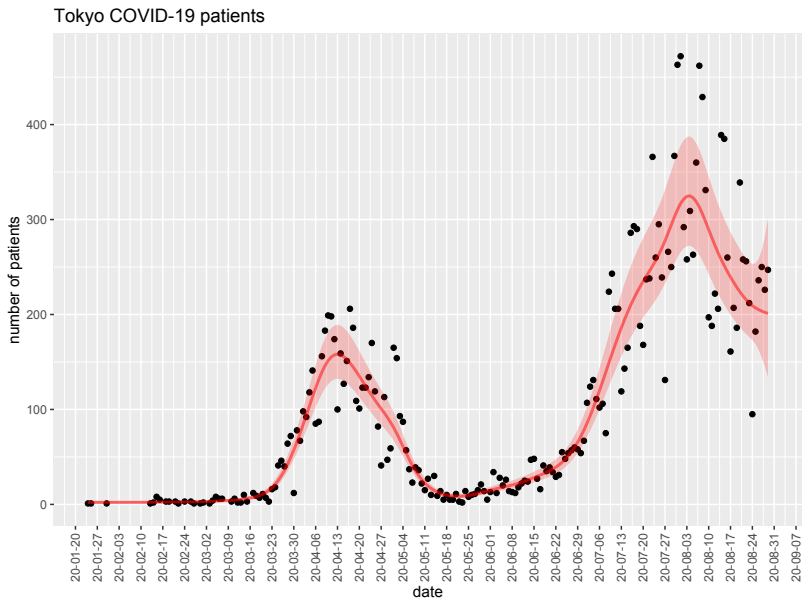


図 3: 状態空間モデルによる平均の推定.

```
## 状態空間モデルにもとづく平均の推定
tmp <- KFAS::signal(out, states = "trend")
tmpa <- tmp$signal
tmpb <- sqrt(tmp$variance[1,1,])
p <-
  ggplot(data = myData %>%
           dplyr::mutate(mean = exp(tmpa),
                         lwr = exp(tmpa - zq*tmpb),
                         upr = exp(tmpa + zq*tmpb)),
         mapping = aes(x = date,
                       y = patients)) +
  geom_point() +
  geom_line(mapping = aes(y = mean),
            color = "red", alpha = 0.5, size = 1) +
  geom_ribbon(mapping = aes(ymin = lwr, ymax = upr),
              fill = "red", alpha = 0.2) +
  scale_x_date(labels = date_format("%y-%m-%d"),
               breaks = date_breaks("1 week")) +
  theme(axis.text.x = element_text(angle = 90,
                                   vjust = 0.5, hjust=1)) +
  labs(title = "Tokyo COVID-19 patients",
       x = "date",
       y = "number of patients")
print(p) # グラフ出力
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の状態空間モデルとして簡略化したもの考える．

まず，図 4 に陽性患者数の対数値 Y_t を示す．

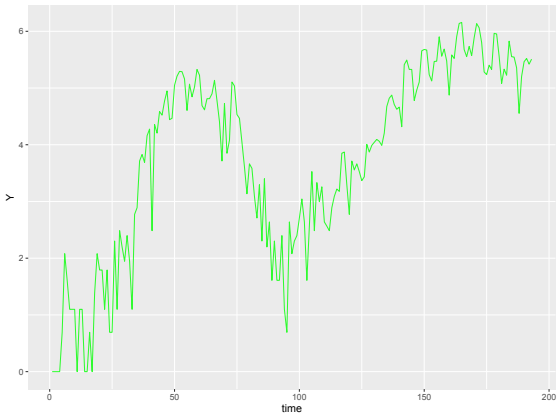


図 4: 陽性患者数の対数値の系列．

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

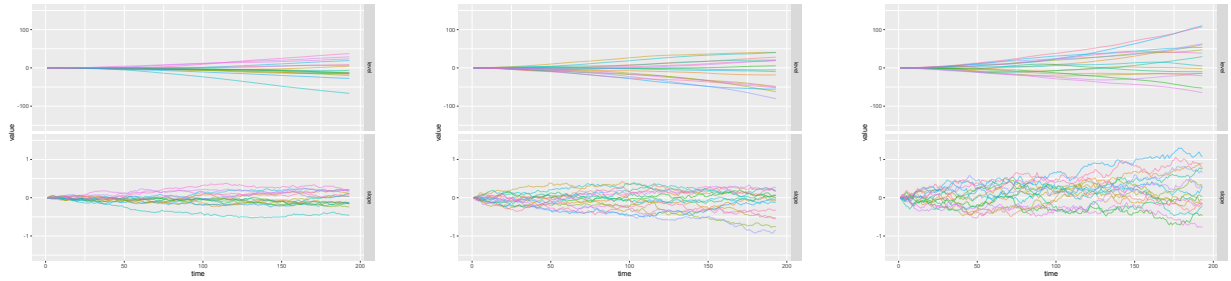
陽性患者数の対数値 Y_t の状態空間モデルとして，2 次のトレンド成分のみからなるモデルを，観測は単純な加法雑音を考える．

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

以下では勾配成分 (slope) の雑音の分散としては，本文で推定された Q_{slope} を用いるが，まず，このパラメタの違いによってどのような状態変数が出現するか確認する．図 5 は勾配成分の雑音の分散を $\lambda \times Q_{\mathrm{slope}}$, $\lambda = 0.3, 1, 3$ とした事前分布から，それぞれ 16 個の状態変数系列 μ_t (水準; level) および ν_t (勾配; slope) をサンプリングした結果を示す．

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のばらつきも同程度になる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．

```
## いくつかのモデルで状態の事前分布を確認
Qs <- fitModel$model$Q[2,2,1] # 推定された Qslope を利用
tmpc <- tibble(t = 1,
               var = rep(c("level","slope"),each = 2),
               name = "V1",
               value = c(150,-150,1.5,-1.5))
for(lambda in c(0.3,1,3)){
```



(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.

```
    model <- # モデルを作成
        SSModel(formula = Y ~ # 目的変数
                    -1 + # 定数項を持たない
                    SSMtrend(degree = 2, # トレンド成分の定義
                             Q = list(0,lambda*Qs)), H = 0.1)
    ## モデルにもとづいて状態を生成 (状態系列の事前分布)
    tmpa <- simulateSSM(model,
                        nsim = 16,
                        conditional=FALSE)
    tmpb <-
        rbind(tibble(t = 1:length(Y),
                     var = "level",
                     as_tibble(tmpa[,1,])),
              tibble(t = 1:length(Y),
                     var = "slope",
                     as_tibble(tmpa[,2,]))) %>%
        tidyr::pivot_longer(-c(t,var))
    p <-
        ggplot(data = tmpb, group = var,
               mapping = aes(x = t,
                             y = value,
                             color = name)) +
        geom_blank(data = tmpc) +
        geom_line(alpha = 0.5) +
        facet_grid(var ~ ., scale = "free_y") +
        theme(legend.position = "none") +
        xlab(label = "time")
    print(p)
}
```

事後分布の計算は，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を選び出す過程と考えることができる．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．

```
## モデルの設定 (以降で使うモデル)
Qs <- fitModel$model$Q[2,2,1]
model <-
  SSModel(formula = Y ~ # 目的変数
                    -1 + # 定数項を持たない
                    SSMtrend(degree = 2, # トレンド成分の定義
                             Q = list(0,Qs)), H = 0.1)
## 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
tmpa <- simulateSSM(model,
```

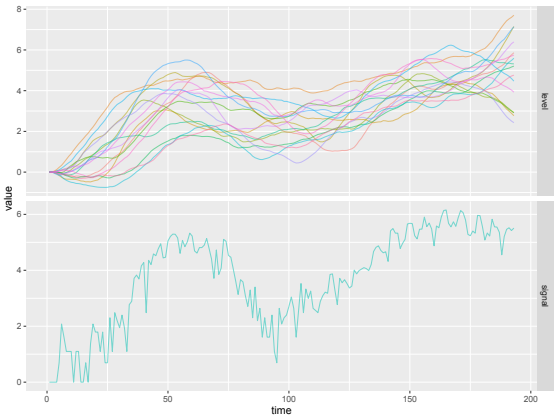
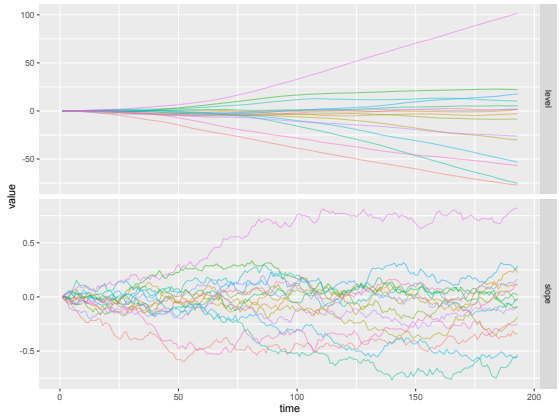


図 7: 事前分布からサンプリングされた変数 μ_t (level) のうち，観測データ Y_t と似たものを抽出した結果.

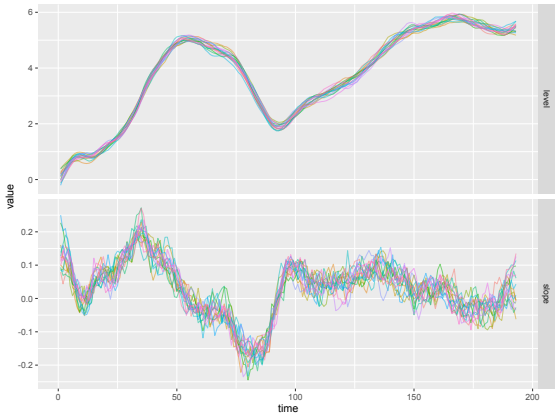
```
                    nsim = 30000,
                    conditional = FALSE)
tmpb <- rbind(
    tibble(t = 1:length(Y),
           var = "level",
           as_tibble(
               tmpa[,1,rank(apply(tmpa[,1,],2,
                                  function(x){var(x-Y)}))<17])) %>%
    tidyr::pivot_longer(-c(t,var)),
    tibble(t = 1:length(Y),
           var = "signal",
           name = "V17",
           value=Y))
## 図示
p <-
    ggplot(data = tmpb, group = var,
           mapping = aes(x = t,
                         y = value,
                         color = name)) +
    geom_line(alpha = 0.5) +
    facet_grid(var ~ ., scale = "free_y") +
    theme(legend.position = "none") +
    xlab(label = "time")
print(p)
```

実際の推定は，Bayes の定理にもとづいて事後分布を計算し，そこからサンプリングを行うことになる．この結果を図 8 に示す．[6]

[6] 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．



(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.

```r
## モデルにもとづいて状態を生成（状態系列の事前分布）
prior <- simulateSSM(model,
                     nsim = 16,
                     conditional = FALSE)
## 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）
postr <- simulateSSM(model,
                     nsim = 16,
                     conditional = TRUE)
## 図示
for(s in c("prior","postr")){
    tmpa <- eval(parse(text=s))
    tmpb <-
        rbind(tibble(t = 1:length(Y),
                     var = "level",
                     as_tibble(tmpa[,1,])),
              tibble(t = 1:length(Y),
                     var = "slope",
                     as_tibble(tmpa[,2,]))) %>%
        tidyr::pivot_longer(-c(t,var))
    p <-
        ggplot(data = tmpb, group = var,
               mapping = aes(x = t,
                             y = value,
                             color = name)) +
        geom_line(alpha = 0.5) +
        facet_grid(var ~ ., scale = "free_y") +
        theme(legend.position = "none") +
        xlab(label = "time")
    print(p)
}
```

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 $Q_{\mathrm{cycle}} = NA$ としても最適化してくれない．[7]

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

[7] 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．

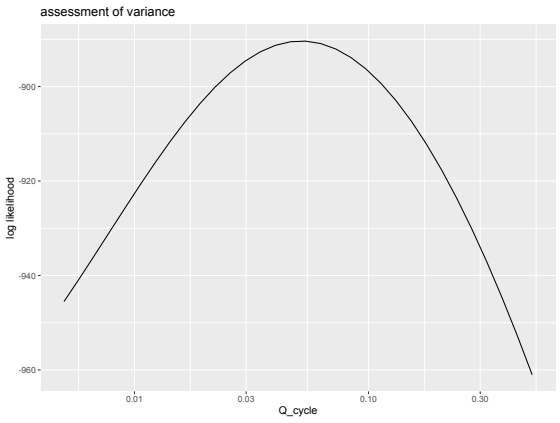


図 10: Q_{cycle} の検討について．

```r
## 周期成分の変動の分散の検討
Qc <- 10^seq(0,-2,length=32)/2
lL <- double(length(Qc))
for(i in 1:length(Qc)) {
    tmp <-
        SSModel(data = myData,
```

```
                  formula = patients ~
                      -1 +
                      SSMtrend(degree = 2,
                               Q = list(0,NA)) +
                      SSMcycle(period = 7,
                               Q = Qc[i]), # 変更
                  distribution = "poisson")
    lL[i] <- logLik(fitSSM(tmp,
                           inits = 0,
                           method = "BFGS")$model)
}
p <-
    ggplot(data = data.frame(Q=Qc, logLik=lL),
           mapping = aes(x = Q, y = logLik)) +
    geom_line() +
    scale_x_log10() +
    labs(title = "assessment of variance",
       x = "Q_cycle",
       y = "log likelihood")
print(p)
print(Qc[which.max(lL)])
```

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．

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