陽性患者数の予測

ARIMA モデルと局所線形構造モデル

N. Murata

*2021*年 *1*月 *8*日

1 はじめに

本稿では，時系列の基本的なモデルを用いて今後の患者数の予測を試みる．

図示やモデル化に際しては以下の package を用いる．

```
  ## パッケージの読み込み
library(forecast)
library(tidyverse)
library(scales) # 年月日表示
library(plotly)
library(zoo)    # 時系列表示
library(ggfortify)
```

2 データの取得

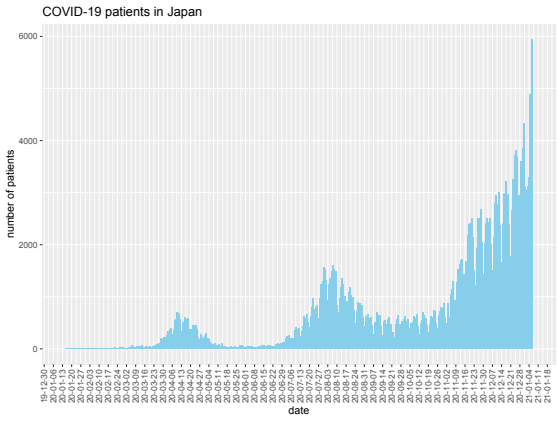
以下では厚生労働省が公開している COVID-19 の全国の感染者数データを利用する．[1]

[1] 厚生労働省の患者数データ

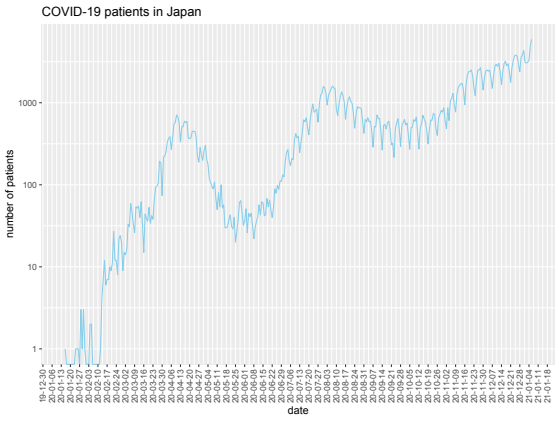
```
## データの取得と整理
patients <-
    read.csv("https://www.mhlw.go.jp/content/pcr_positive_d」
    ↪  aily.csv")
    ↪  %>%
    dplyr::rename(date=1, patients=2) %>%
    dplyr::mutate(date=as.Date(date))
## 時系列データ(zooクラス)への変更
patients <- with(patients,
                  zoo(x=patients, order.by=date))
```

図 1 は全国の陽性患者数の推移を図示したものである．左図は横軸に日付を，縦軸に観測された患者数を表示したものである．SIR などの感染症の基本的なモデルによれば，感染が拡大しはじめる際の人数の増加は指数関数的であるので，右図では縦軸を患者数の常用対数として表示している．週日・休日での検査数に波はあるが，9 月以降はほぼ単調に増加していることがわかる．

```
## データの視覚化
p <-
    ggplot(data = fortify(patients, melt = TRUE),
           mapping = aes(x = Index,
                         y = Value)) +
    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)) +
```



(a) 日毎の患者数の変遷



(b) 患者数の対数表示

図 1: 全国の陽性患者数の推移.

```
    labs(title = "COVID-19 patients in Japan",
         x = "date",
         y = "number of patients")
## 棒グラフ
print(p + geom_col(fill="skyblue")) # グラフ出力
ggplotly() # plotly 表示 (browser)
## 折れ線グラフ+常用対数表示
print(p + geom_line(colour="skyblue") + scale_y_log10())
ggplotly()
```

3 基礎分析

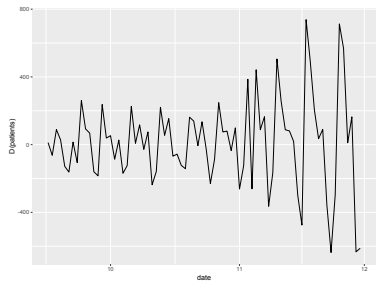
次節以降で確認するモデルの精度を確認するために，推定と予測のために系列を分けて考える．[2]　　9 月中旬から 11 月のデータをモデルの推定に用い，12 月以降のデータで予測の精度を検証することとする．

[2] 東京都の分析結果から 9 月中旬あたりが第 3 波の開始と考えらるので，ここでは 9 月 15 日以降のデータを用いることとする．

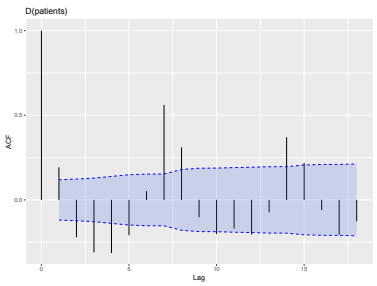
```
## 9月以降の第3波を対象とする
train <- window(patients,
                start="2020-09-15",
                end="2020-11-30")
test <- window(patients,
               start="2020-12-01")
```

まず，9-11 月のデータを用いて時系列の基礎的な性質を確認する．

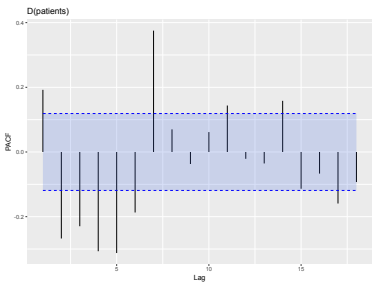
データは明らかに定常ではないので，まず階差系列の性質を確認する．図 3 に階差系列の推移，および自己相関と偏自己相関を示す．系列の分散は時間とともに増大しており，また相関関係から 7 日毎の関係が強いことがわかる．これは検査機関の稼働状況の影響だと考えられる．



(a) 階差系列



(b) 自己相関

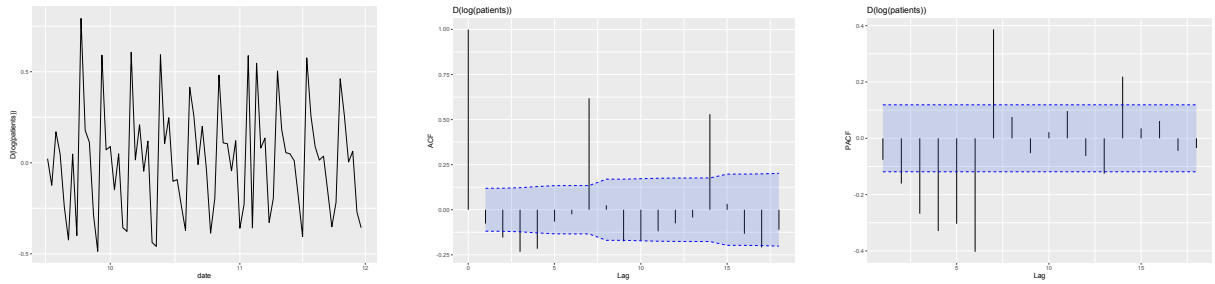


(c) 偏自己相関

図 3: 階差系列の基礎的な性質.

```r
## 階差系列の性質
autoplot(diff(train)) +
  labs(x = "date",
       y = "D(patients)")
autoplot(acf(diff(train), plot = FALSE), # 自己相関
         conf.int.fill = "royalblue",
         conf.int.alpha =0.2,
         conf.int.value = 0.7,
         conf.int.type = "ma") +
  labs(title = "D(patients)")
autoplot(pacf(diff(train), plot = FALSE), # 偏自己相関
         conf.int.fill = "royalblue",
         conf.int.alpha =0.2,
         conf.int.value = 0.7) +
  labs(title = "D(patients)",
       y = "PACF")
```

SIR のような感染症のモデルの解析から，感染者数は指数型の増加を示すことが知られているので，次に感染者数の対数を取ったものの階差系列の性質を調べる．この結果を図 5 に示す．系列の推移より，分散の変動が定常化されていることがわかる．自己相関からは 7 日毎の関係が同様に確認される．



(a) 階差系列　(b) 自己相関　(c) 偏自己相関

図 5: 対数変換後の階差系列の性質．

```r
## 対数変換を確認する
ltrain <- log(train)
autoplot(diff(ltrain)) +
    labs(x = "date",
         y = "D(log(patients))")
autoplot(acf(diff(ltrain), plot = FALSE), # 自己相関
         conf.int.fill = "royalblue",
         conf.int.alpha =0.2,
         conf.int.value = 0.7,
         conf.int.type = "ma") +
    labs(title = "D(log(patients))")
autoplot(pacf(diff(ltrain), plot = FALSE), # 偏自己相関
         conf.int.fill = "royalblue",
         conf.int.alpha =0.2,
         conf.int.value = 0.7) +
    labs(title = "D(log(patients))",
         y = "PACF")
```

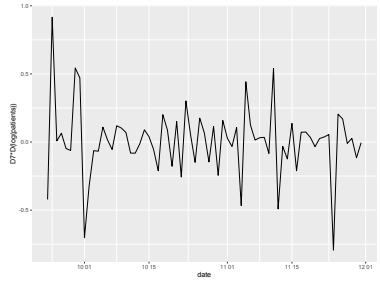
また，この階差系列の基本統計量 (平均と中央値) を見ると，若干正に偏っており，元の系列が増大する傾向があることが確認される．

```r
## 基本統計量の確認
summary(as.numeric(diff(ltrain)))
```

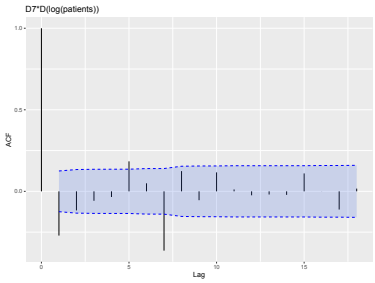
```
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
-0.48603 -0.23086  0.02945  0.01303  0.17210  0.79224
```

次に 7 日毎の階差を取り，7 日毎の相関の性質がどのように変化するか確認する．これを図 7 に示す．相関関係の正負が逆転し，強い相関が残ることから，単純な周期成分ではないことが示唆される．[3]

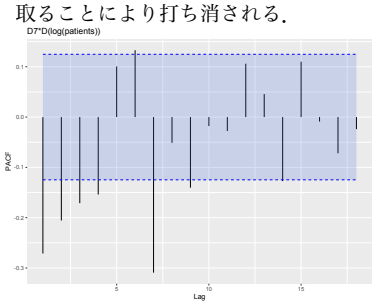
[3] 7 日毎の周期成分であれば 7 日階差を取ることにより打ち消される．



(a) 階差系列



(b) 自己相関



(c) 偏自己相関

図 7: 1 日および 7 日の階差を取った系列の性質．

```
## 7日の周期性を確認する
autoplot(diff(diff(ltrain), lag=7)) +
    labs(x = "date",
         y = "D7*D(log(patients))")
autoplot(acf(diff(diff(ltrain), lag=7), plot = FALSE), # 自己相関
         conf.int.fill = "royalblue",
         conf.int.alpha =0.2,
         conf.int.value = 0.7,
         conf.int.type = "ma") +
    labs(title = "D7*D(log(patients))")
autoplot(pacf(diff(diff(ltrain), lag=7), plot = FALSE), # 偏自己相関
         conf.int.fill = "royalblue",
         conf.int.alpha =0.2,
         conf.int.value = 0.7) +
    labs(title = "D7*D(log(patients))",
         y = "PACF")
```

4 ARIMA モデル

まず ARIMA モデルによる推定を行う．ARIMA モデルは階差系列に対して ARMA モデルを用いたものである．直感的には ARMA モデルに従う時系列のランダムウォークであり，非定常な時系列の基本的なモデルの一つである．

時系列を $X_t,\ t = 1, 2, \ldots$ とし，ラグ作用素 L (lag operator, backshift operator) を以下で定義する．

$$LX_t = X_{t-1}$$

これを用いると，階差系列は

$$Y_t = X_t - X_{t-1} = (1 - L)X_t$$

と書け，高階の階差は作用素 $(1 - L)$ の冪で書くことができる．また次数 (p, d, q) の ARIMA モデルは以下で定義される．

$$\begin{aligned}&(1 - a_1 L - a_2 L^2 - \cdots - a_p L^p)(1 - L)^d X_t \\ &\quad = (1 + b_1 L + b_2 L^2 + \cdots + b_q L^q)\epsilon_t,\ \epsilon_t \sim \mathcal{N}(0, \sigma^2)\end{aligned}$$

今回対象とするデータは 1 階の階差で定常となっていると考えられるため，$d = 1$ のモデルを考えることになる．また，増大する傾向を一種のトレンドとしてモデル化するためにドリフト (drift) 項を加えて以下のモデルを考える．

$$\begin{aligned}&(1 - a_1 L - a_2 L^2 - \cdots - a_p L^p)(1 - L)(X_t + \beta t) \\ &\quad = (1 + b_1 L + b_2 L^2 + \cdots + b_q L^q)\epsilon_t,\ \epsilon_t \sim \mathcal{N}(0, \sigma^2)\end{aligned}$$

次数 (p, d, q) と対応する係数とドリフト，およびホワイトノイズの分散 σ^2 を推定した結果は以下のようになる．[4]

[4] 季節成分 (seasonal) を加えた SARIMA モデルを考えることもできるが，このデータでは周期性が曖昧なため，モデルの推定はうまくいかなかった．コードのコメント部分を参照．

```
## drift 付きの ARIMA モデルの次数を自動推定
est.arima <- forecast::auto.arima(ltrain)
## 推定されたモデルを表示
print(est.arima)
## SARIMA モデルを当て嵌める場合は周期を指定する．
## frequency(ltrain) <- 7 # 7日周期の成分を仮定
## est.arima <- auto.arima(ltrain)
## このデータではモデルの推定はうまくいかない
```

```
Series: ltrain
ARIMA(5,1,1) with drift

Coefficients:
          ar1      ar2      ar3      ar4      ar5      ma1   drift
      -0.0394  -0.3175  -0.3379  -0.3265  -0.2757  -0.4083  0.0191
s.e.   0.1722   0.1102   0.1083   0.1120   0.1263   0.1532  0.0075

sigma^2 estimated as 0.06513:  log likelihood=-1.28
AIC=18.55   AICc=20.7   BIC=37.2
```

図 9 は推定されたモデルの良さを診断するためのプロットで，上から順に残差の推移，残差の自己相関，自己相関の Ljung-Box 統計量 (自己相関が 0 であるか検定) を表している．残差は無作為になっているように見えるが，7 日毎の自己相関は若干残っており，7 日以降のラグの無相関性は棄却されない．

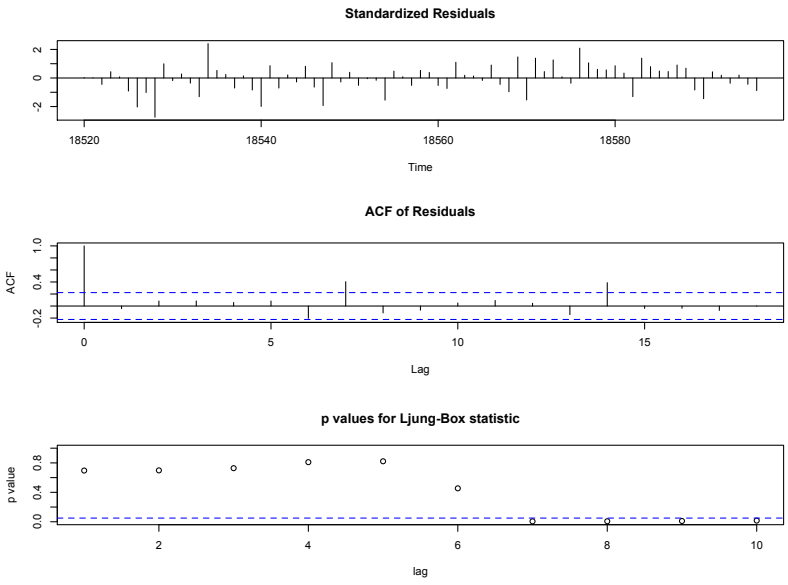


図 9: ARIMA モデルの診断．

```
## 診断プロット
tsdiag(est.arima)
## 残差に相関が残っているので，優れたモデルという訳ではない
```

推定されたモデルによる当て嵌め結果 (モデルによる 1 期先の予測) を図 10 に示す．概周期成分による誤差 (遅れ) はあるものの，それなりに良く追従していることがわかる．

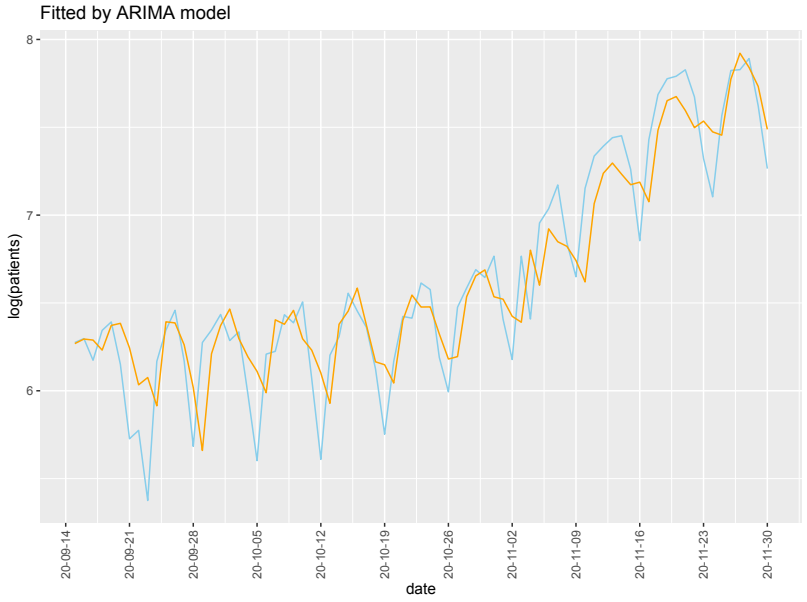


図 10: ARIMA モデルによる当て嵌め結果.

```
## モデルによる当て嵌めの視覚化
p <-
    ggplot(data = fortify(est.arima) %>%
               dplyr::mutate(Index=as.Date(Index)),
           mapping = aes(x = Index,
                         y = Data)) +
    geom_line(colour = "skyblue") +
    geom_line(mapping = aes(y = Fitted),
              colour = "orange") +
    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 = "Fitted by ARIMA model",
         x = "date",
         y = "log(patients)")
print(p)
ggplotly()
```

このモデルを用いて，12 月以降の患者数を予測した結果を 80%信頼区間とともに示したのが図 11 である.

```
## 12月以降（最大 60 日）を予測してみる
p <-
    ggplot(data = fortify(forecast(est.arima,
                                   h=min(length(test),60))) %>%
               dplyr::mutate(Index=as.Date(Index)) %>%
               left_join(fortify(test), by = "Index"),
           mapping = aes(x = Index,
                         y = exp(Data)),
           na.rm = TRUE) +
    geom_line(colour = "skyblue",
              na.rm = TRUE) +
    geom_line(mapping = aes(y = test),
              colour = "red",
              na.rm = TRUE) +
    geom_line(mapping = aes(y = exp(`Point Forecast`)),
              colour = "royalblue",
```

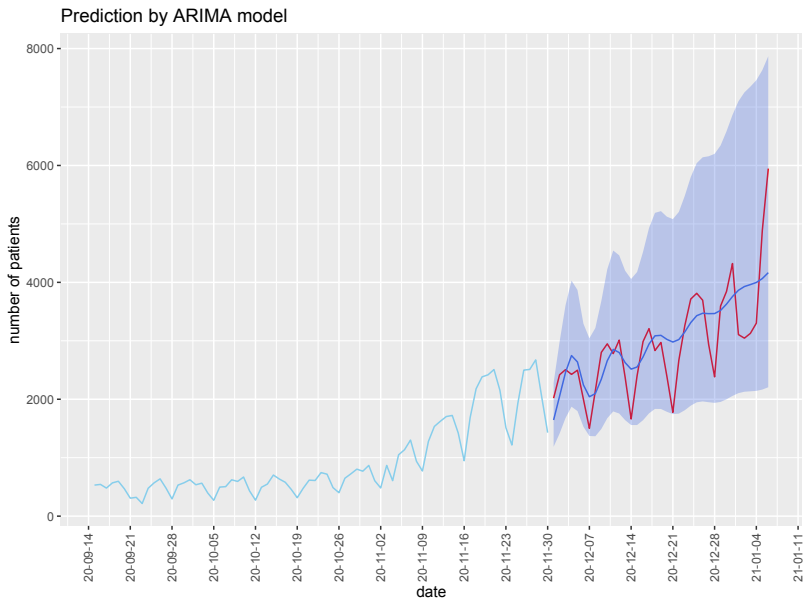


図 11: ARIMA モデルによる予測.

```
                na.rm = TRUE) +
      geom_ribbon(mapping = aes(ymin = exp(`Lo 80`),
                                ymax = exp(`Hi 80`)),
                  fill = "royalblue", alpha = 0.3,
                  na.rm = TRUE) +
      ## geom_ribbon(mapping = aes(ymin = exp(`Lo 95`),
      ##                           ymax = exp(`Hi 95`)),
      ##             fill = "royalblue", alpha = 0.1,
      ##             na.rm = TRUE) +
      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 = "Prediction by ARIMA model",
           x = "date",
           y = "number of patients")
  print(p)
  ggplotly()
```

5 局所線形構造モデル

もう一つは局所線形構造モデルと呼ばれるもので，時系列に以下の構造を仮定したもである．

観測された時系列 X_t は，トレンド μ_t とホワイトノイズ ϵ の 2 つの成分からなると仮定する．

$$X_t = \mu_t + \epsilon_t,\ \epsilon_t \sim \mathcal{N}(0, \sigma_\epsilon^2)$$

さらにトレンドは以下のような力学系に従うと仮定する．

$$\mu_{t+1} = \mu_t + \nu_t + \xi_t,\ \xi_t \sim \mathcal{N}(0, \sigma_\xi^2)$$
$$\nu_{t+1} = \nu_t + \zeta_t,\ \zeta_t \sim \mathcal{N}(0, \sigma_\zeta^2)$$

直感的には ν_t はトレンドの傾きに相当する．傾きがランダムウォークにより増減することにより，トレンドの変化の速度が変わるため，多項式より複雑な形状のトレンドを表すことができる．

このモデルには 3 つのホワイトノイズが含まれているが，その分散の最尤推定を行うことによって，モデルの推定が行われる．

```r
## StructTS による方法
est.sts <- StructTS(ltrain)
## 推定されたモデルを表示
print(est.sts)
```

```
Call:
StructTS(x = ltrain)

Variances:
  level    slope  epsilon
0.07039  0.00000  0.01102
```

ARIMA モデルと同様に，推定されたモデルの診断プロットを図 12 に示す．ARIMA より残差に相関が残っていることが確認できる．

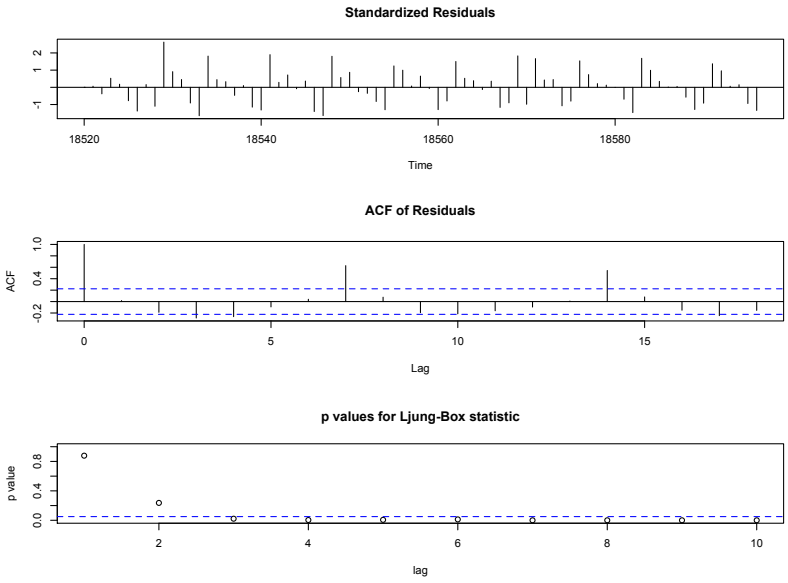


図 12: 局所線形構造モデルの診断．

```r
## 診断プロット
tsdiag(est.sts)
## こちらも残差に相関が残っているので，優れたモデルという訳ではない
```

推定されたモデルによる元の時系列の分解結果は図 13 のようになる．

```r
## StructTS による時系列の分解結果
autoplot(est.sts)
```

モデルによる当て嵌め結果を図 14 に示す．ARIMA モデルより推定のラグが大きいことが見て取れる．

```r
## モデルによる当て嵌めの視覚化
p <-
    ggplot(data = fortify(forecast(est.sts)) %>%
               dplyr::mutate(Index=as.Date(Index)),
           mapping = aes(x = Index,
                         y = Data),
           na.rm = TRUE) +
    geom_line(colour = "skyblue",
              na.rm = TRUE) +
    geom_line(mapping = aes(y = Fitted),
              colour = "orange",
              na.rm = TRUE) +
```

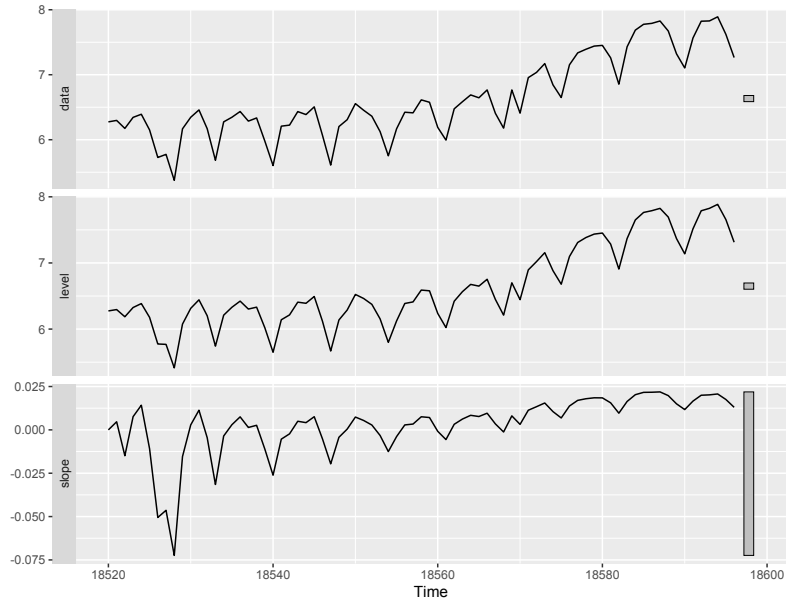


図 13: 局所線形構造モデルによる時系列の分解．level が μ，slope が ν に対応する．

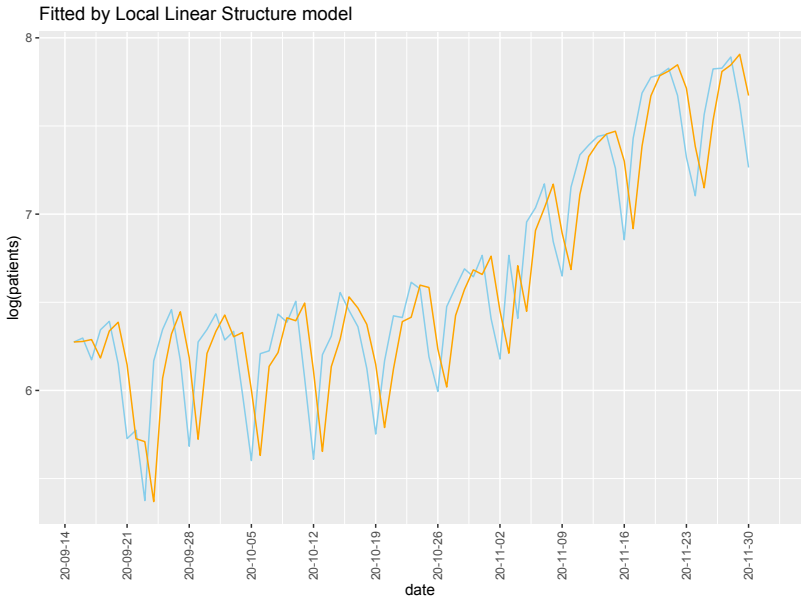


図 14: 局所線形構造モデルによる当て嵌め結果．

```
    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 = "Fitted by Local Linear Structure model",
         x = "date",
         y = "log(patients)")
print(p)
ggplotly()
```

予測の結果は図 15 に示す．増加傾向は予測できており，信頼区間の中に実際の値は含まれているが，信頼区間は非常に広く，予測の精度は ARIMA に劣ると考えられる．

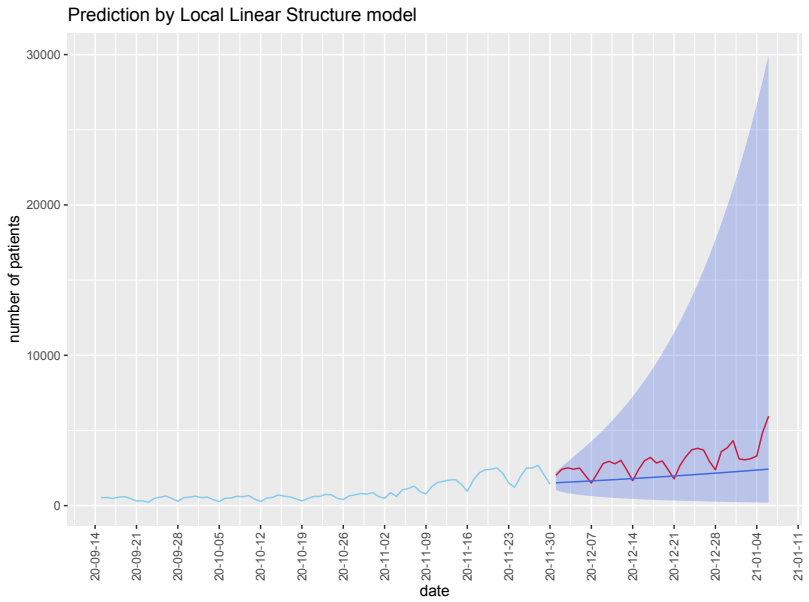


図 15: 局所線形構造モデルによる予測．

```
## 12月以降（最大 60 日）を予測してみる
p <-
    ggplot(data = fortify(forecast(est.sts,
                                   h=min(length(test),60))) %>%
               dplyr::mutate(Index=as.Date(Index)) %>%
               left_join(fortify(test), by = "Index"),
           mapping = aes(x = Index,
                         y = exp(Data)),
           na.rm = TRUE) +
    geom_line(colour = "skyblue",
              na.rm = TRUE) +
    geom_line(mapping = aes(y = test),
              colour = "red",
              na.rm = TRUE) +
    geom_line(mapping = aes(y = exp(`Point Forecast`)),
              colour = "royalblue",
              na.rm = TRUE) +
    geom_ribbon(mapping = aes(ymin = exp(`Lo 80`),
                              ymax = exp(`Hi 80`)),
                fill = "royalblue", alpha = 0.3,
                na.rm = TRUE) +
    ## geom_ribbon(mapping = aes(ymin = exp(`Lo 95`),
    ##                           ymax = exp(`Hi 95`)),
    ##             fill = "royalblue", alpha = 0.1,
```

```
    ##              na.rm = TRUE) +
    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 = "Prediction by Local Linear Structure model",
         x = "date",
         y = "number of patients")
print(p)
ggplotly()
```

6 ARIMA モデルによる予測

前 2 節の結果から，このデータに対しては ARIMA モデルによる予測の方が精度が良いと考えられる．階差系列が ARMA モデルで良く近似されるということは，9 月中旬以降の感染拡大の動特性があまり変化していないということであり，集団の行動がこの期間でほとんど変様していないことが示唆される．

現在までのデータを用いて ARIMA モデルを再度構築し，これを用いて 60 日間の予測を行った結果を図 16 に示す．内側は 80%信頼区間，外側は 95%信頼区間である．

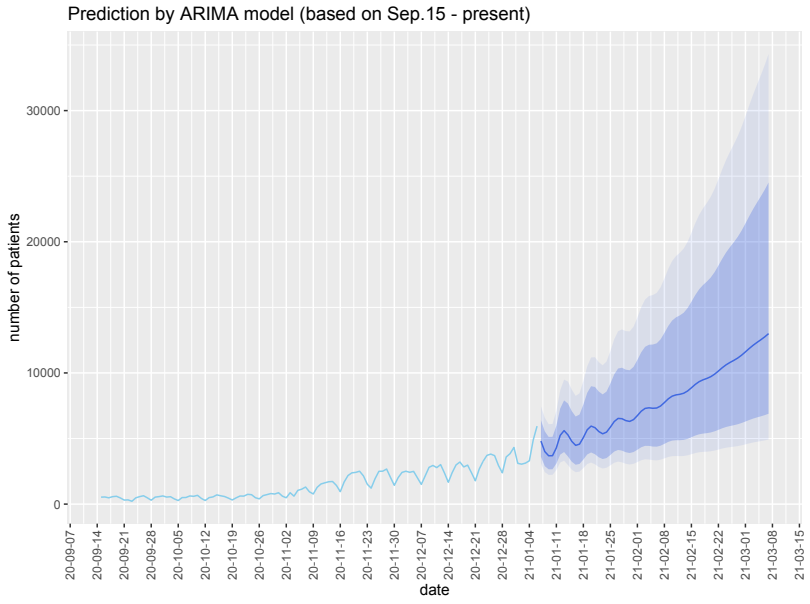


図 16: ARIMA モデルによる今後 60 日間の予測．

```
## モデルの推定
est <- auto.arima(log(window(patients,start="2020-09-15")))
## 推定されたモデルの表示
print(est)
## 現在から 60 日先まで予測してみる
p <-
    ggplot(data = fortify(forecast(est,h=60)) %>%
               dplyr::mutate(Index=as.Date(Index)),
           mapping = aes(x = Index,
                         y = exp(Data)),
           na.rm = TRUE) +
    geom_line(colour = "skyblue",
              na.rm = TRUE) +
    geom_line(mapping = aes(y = exp(`Point Forecast`)),
```

```
                colour = "royalblue",
                na.rm = TRUE) +
      geom_ribbon(mapping = aes(ymin = exp(`Lo 80`),
                                ymax = exp(`Hi 80`)),
                  fill = "royalblue", alpha = 0.3,
                  na.rm = TRUE) +
      geom_ribbon(mapping = aes(ymin = exp(`Lo 95`),
                                ymax = exp(`Hi 95`)),
                  fill = "royalblue", alpha = 0.1,
                  na.rm = TRUE) +
      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 = "Prediction by ARIMA model (based on Sep.15 - present)",
           x = "date",
           y = "number of patients")
  print(p)
  ggplotly()
```

```
Series: log(window(patients, start = "2020-09-15"))
ARIMA(5,1,1) with drift

Coefficients:
          ar1      ar2      ar3      ar4      ar5      ma1   drift
      -0.0594  -0.3813  -0.3579  -0.3409  -0.3318  -0.4022  0.0191
s.e.   0.1324   0.0869   0.0891   0.0886   0.1006   0.1193  0.0051

sigma^2 estimated as 0.04994:  log likelihood=11.54
AIC=-7.07   AICc=-5.69   BIC=14.74
```

7 おわりに

対象とした患者数の推移は極めて非定常なデータであるが，時系列を適切に変換し，生成モデルがある程度定常な区間を捉えることができれば，基本的な ARMA モデルでも良い推定を行うことができる．