時系列解析

モデルの推定と予測

村田 昇

講義概要

- 第 1 回 : 時系列の基本モデル
- **第 2 回 : モデルの推定と予測**

時系列解析の復習

時系列解析とは

- 時系列データ
  - 時間軸に沿って観測されたデータ
  - 観測の順序に意味がある
  - 異なる時点間での観測データの従属関係が重要
  - **独立性にもとづく解析は行えない**
    - そのままでは大数の法則や中心極限定理は使えない
- 時系列解析の目的
  - 時系列データの特徴を効果的に記述すること
  - 時系列モデルの推定と評価

時系列モデルと定常性

- **確率過程**

  時間を添え字として持つ確率変数列

  $$X_t,\ t = 1, \ldots, T$$

- **弱定常過程** : 以下の性質をもつ確率過程 X_t
  - X_t の平均は時点 t によらない
  - X_t と X_{t+h} の共分散は時点 t によらず時差 h のみで定まる
  - 特に X_t の分散は時点 t によらない ($h = 0$ の場合)
- 多くの場合，弱定常性を考えれば十分なので単に **定常** ということが多い
- 定常でない確率過程は **非定常** であるという

ホワイトノイズ

- 定義

  平均 0，分散 σ^2 である確率変数の確率分布 P からの独立かつ同分布な確率変数列

  $$X_t = \epsilon_t, \quad \epsilon_t \overset{i.i.d.}{\sim} P$$

  - 記号 $\mathrm{WN}(0, \sigma^2)$ で表記
  - **定常** な確率過程

トレンドのあるホワイトノイズ

- 定義

  μ, α を定数として

  $$X_t = \mu + \alpha t + \epsilon_t, \quad \epsilon_t \sim \mathrm{WN}(0, \sigma^2)$$

  で定義される確率過程

  - **非定常** な確率過程
  - トレンド項 (平均値の変化) は現象に応じて一般化される

ランダムウォーク

- 定義

  X_0 を定数もしくは確率変数として

  $$X_t = X_{t-1} + \epsilon_t, \quad \epsilon_t \sim \mathrm{WN}(0, \sigma^2)$$

  で帰納的に定義される確率過程

  - 分散が時間とともに増加・記憶のあるモデル
  - **非定常** な確率過程

自己回帰過程

- 定義 (次数 p の AR モデル)

  $a_1, \ldots, a_p$ を定数とし，$X_1, \ldots, X_p$ が初期値として与えられたとき,

  $$X_t = a_1 X_{t-1} + \cdots + a_p X_{t-p} + \epsilon_t, \quad \epsilon_t \sim \mathrm{WN}(0, \sigma^2)$$

  で帰納的に定義される確率過程

  - ランダムウォークの一般化
  - 無限長の記憶のある (忘却しながら記憶する) モデル
  - **定常にも非定常にもなる**

移動平均過程

- 定義 (次数 q の MA モデル)

  $b_1, \ldots, b_q$ を定数とし，$X_1, \ldots, X_q$ が初期値として与えられたとき

  $$X_t = b_1 \epsilon_{t-1} + \cdots + b_q \epsilon_{t-q} + \epsilon_t, \quad \epsilon_t \sim \mathrm{WN}(0, \sigma^2)$$

  で定義される確率過程

  - 有限長の記憶のあるモデル
  - **定常** な確率過程

自己回帰移動平均過程

- 定義 (次数 (p, q) の ARMA モデル)

  $a_1, \ldots, a_p, b_1, \ldots, b_q$ を定数とし，$X_1, \ldots, X_{\max\{p,q\}}$ が初期値として与えられたとき

  $$\begin{aligned} X_t = {} & a_1 X_{t-1} + \cdots + a_p X_{t-p} \\ & + b_1 \epsilon_{t-1} + \cdots + b_q \epsilon_{t-q} + \epsilon_t, \\ & \epsilon_t \sim \mathrm{WN}(0, \sigma^2) \end{aligned}$$

  で帰納的に定まる確率過程

  - AR・MA モデルの一般化・基本的な時系列モデル
  - **定常にも非定常にもなる**

自己共分散・自己相関

- 弱定常な確率過程 : $X_t,\ t = 1, \ldots, T$
  - X_t と X_{t+h} の共分散は時点 t によらずラグ h のみで定まる

    自己共分散 (定常過程の性質よりラグは $h \geq 0$ を考えればよい)

    $$\gamma(h) = \mathrm{Cov}(X_t, X_{t+h})$$

  - X_t と X_{t+h} の相関も t によらずラグ h のみで定まる

    自己相関

    $$\rho(h) = \gamma(h)/\gamma(0) = \mathrm{Cov}(X_t, X_{t+h})/\mathrm{Var}(X_t)$$

- 異なる時点間での観測データの従属関係を要約するための最も基本的な統計量

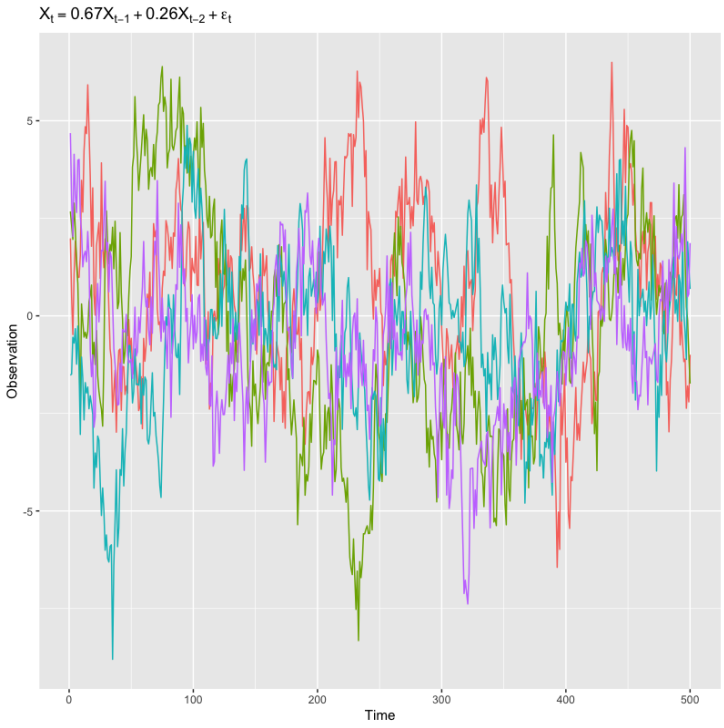

図 1: 同じモデルに従う AR 過程の例

実習 (復習)

R：自己相関・自己共分散の計算・描画

- 関数 feats::ACF()

```
ACF(.data,  y, ...,  lag_max = NULL,
  type = c("correlation", "covariance", "partial"),
  na.action = na.contiguous, demean = TRUE, tapered = FALSE)
#' .data: 時系列データ (tsibble クラス)
#' y: 計算対象の列名
#' type: 標準は相関，共分散と偏相関を選ぶこともできる
#' na.action: 欠損値の処理，標準は欠損を含むと計算しない
#' demean: 共分散の計算において平均を引くか否か
```

  - 関数 `acf()` と違い lag=0 を表示しない

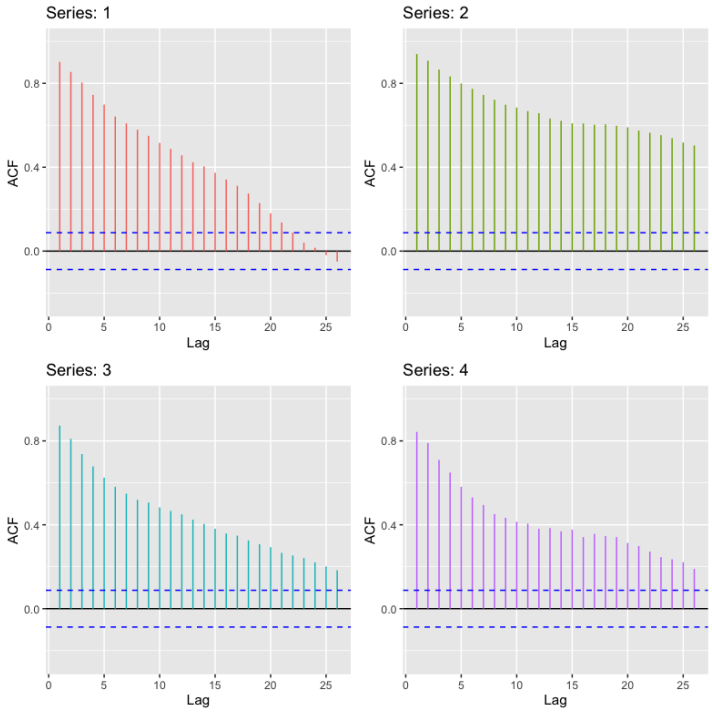

図 2: AR 過程の自己相関

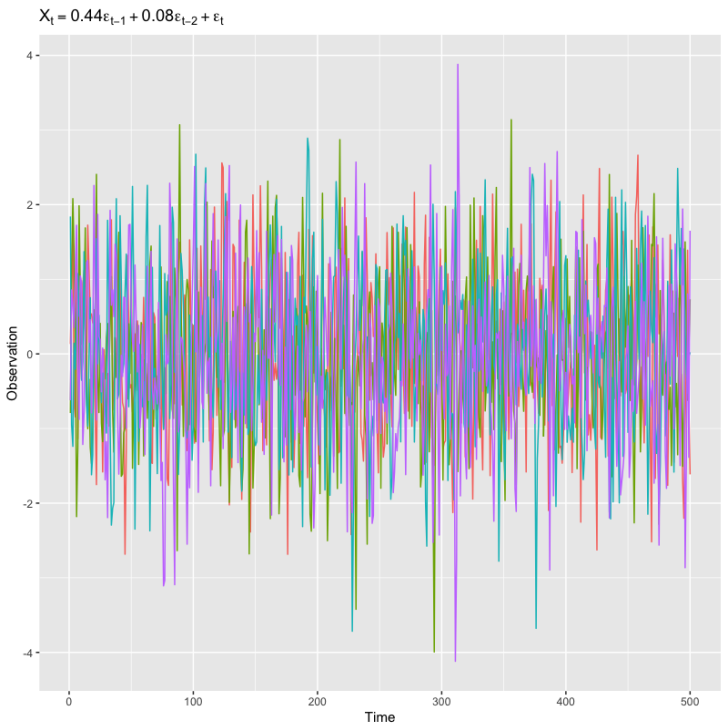

図 3: 同じモデルに従う MA 過程の例

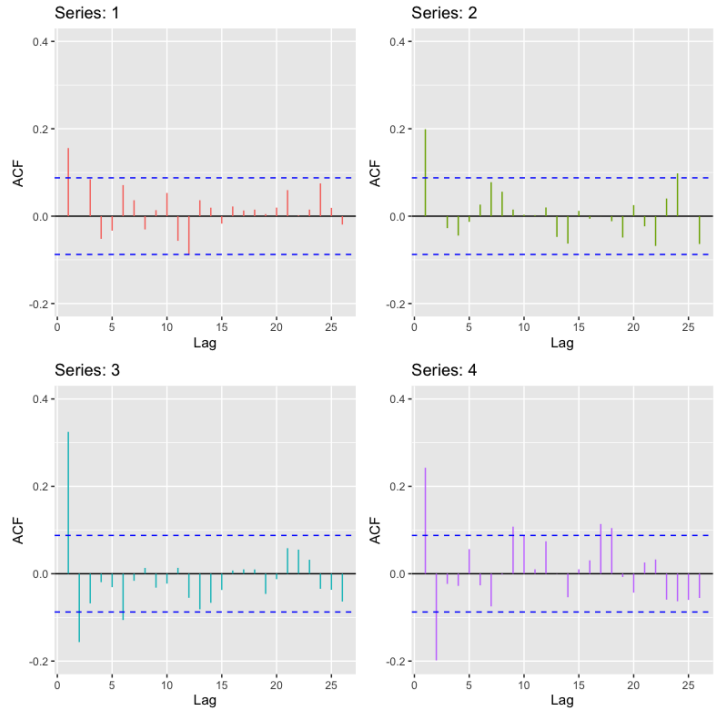

図 4: MA 過程の自己相関

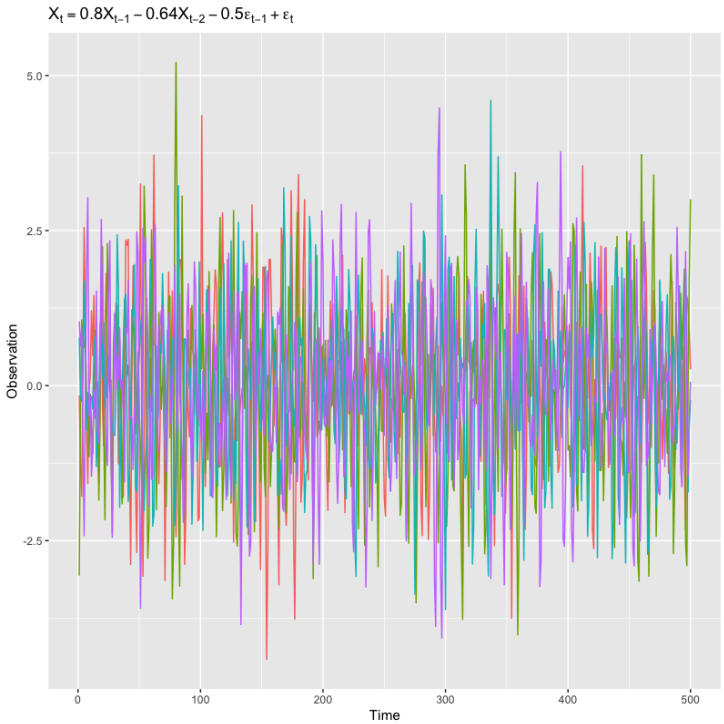

図 5: 同じモデルに従う ARMA 過程の例

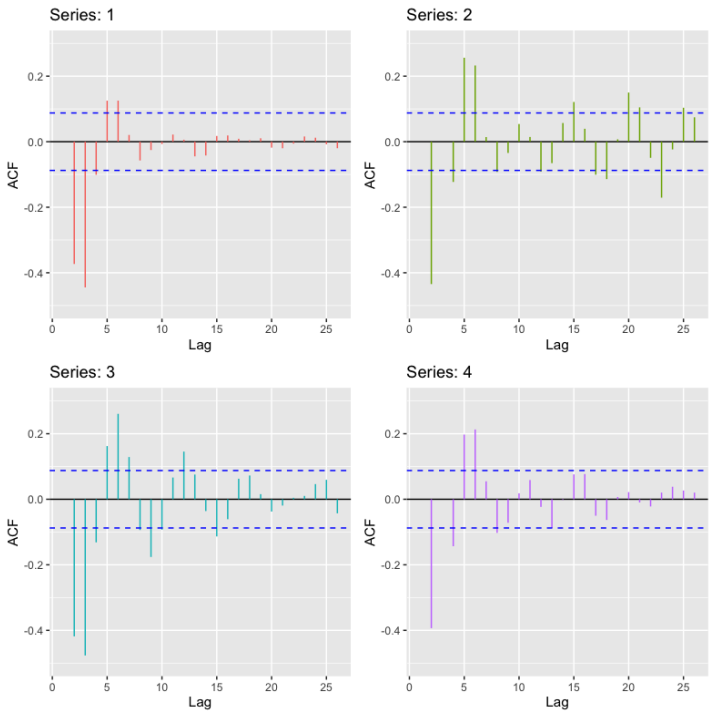

図 6: ARMA 過程の自己相関

- 返値を `autoplot()` に渡せばグラフを描画する
- 典型的な使い方

```
toy_acf <- arima.sim(model = list(ar = c(0.8, -0.64),
                                  ma = c(-0.5)),
                     n = 200) |>
  as_tsibble() |> ACF(value)
toy_acf |> autoplot()
```

練習問題

- 以下の問に答えなさい
  - 同じ AR 過程のモデルから生成した時系列の自己相関を比較しなさい (前回の練習問題を利用すればよい)
  - MA 過程についても同様な比較を行いなさい
  - ARMA 過程についても同様な比較を行いなさい

AR モデルの推定

自己共分散・自己相関

- 平均 0 の弱定常な確率過程 : $X_t,\ t = 1, \ldots, T$
  - X_t と X_{t+h} の共分散は時点 t によらずラグ h のみで定まる

    自己共分散

    $$\gamma(h) = \mathrm{Cov}(X_t, X_{t+h}) = \mathbb{E}[X_t X_{t+h}]$$

  - X_t と X_{t+h} の相関も t によらずラグ h のみで定まる

    自己相関係数

    $$\rho(h) = \mathrm{Cov}(X_t, X_{t+h})/\mathrm{Var}(X_t) = \gamma(h)/\gamma(0)$$

自己共分散と AR モデル

- AR(p) モデル :

$$X_t = a_1 X_{t-1} + a_2 X_{t-2} + \cdots + a_p X_{t-p} + \epsilon_t$$

- 係数と自己共分散の関係

$$\begin{aligned}\gamma(h) &= \mathbb{E}[X_t X_{t+h}] \\ &= \mathbb{E}[X_t (a_1 X_{t+h-1} + \cdots + a_p X_{t+h-p} + \epsilon_{t+h})] \\ &= a_1 \mathbb{E}[X_t X_{t+h-1}] + \cdots + a_p \mathbb{E}[X_t X_{t+h-p}] + \mathbb{E}[X_t \epsilon_{t+h}] \\ &= a_1 \gamma(h-1) + \cdots + a_p \gamma(h-p)\end{aligned}$$

Yule-Walker 方程式

- $1 \leq h \leq p$ を考えると以下の関係が成り立つ

$$\begin{pmatrix}\gamma(1) \\ \gamma(2) \\ \vdots \\ \gamma(p)\end{pmatrix} = \begin{pmatrix}\gamma(0) & \gamma(-1) & \dots & \gamma(-p+1) \\ \gamma(1) & \gamma(0) & \dots & \gamma(-p+2) \\ \vdots & \vdots & \ddots & \vdots \\ \gamma(p-1) & \gamma(p-2) & \dots & \gamma(0)\end{pmatrix}\begin{pmatrix}a_1 \\ a_2 \\ \vdots \\ a_p\end{pmatrix}$$

  - 行列は Toeplitz 行列と呼ばれる
  - 行列が正則ならば AR の係数は一意に求まる

偏自己相関

- AR(p) モデル

$$X_t = a_1 X_{t-1} + a_2 X_{t-2} + \cdots + a_p X_{t-p} + \epsilon_t$$

  - ラグ p の **偏自己相関係数**

    AR(p) モデルを仮定したときの a_p の推定値 (Yule-Walker 方程式の解)
  - ラグ p の特別な **自己相関係数**

    $a_1 = a_2 = \cdots = a_{p-1} = 0$ のときの a_p (**特殊なモデルにおける解釈**)

$$\mathbb{E}[X_t X_{t+p}] = a_p \mathbb{E}[X_t X_t] \;\Rightarrow\; \gamma(p) = a_p \gamma(0) \;\Rightarrow\; \rho(p) = a_p$$

モデルの推定に関する補足

- ARMA モデルの推定方法は主に以下の 3 つ
  - Yule-Walker 方程式 (AR 過程)
  - 最小二乗
    - 予測誤差の平方和の最小化
    - 回帰と同じだが，従属系列のため多重共線性に注意
  - 最尤推定
    - WN の分布を仮定して同時尤度関数を設定
    - 非線形最適化を行う
- 一般にモデルは近似なので，どの推定が良いかは問題による

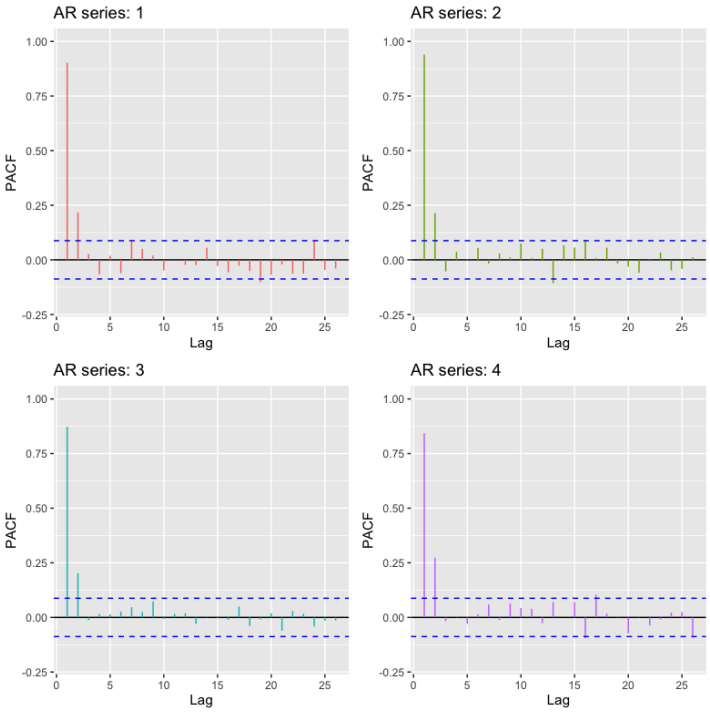

図 7: AR 過程の偏自己相関

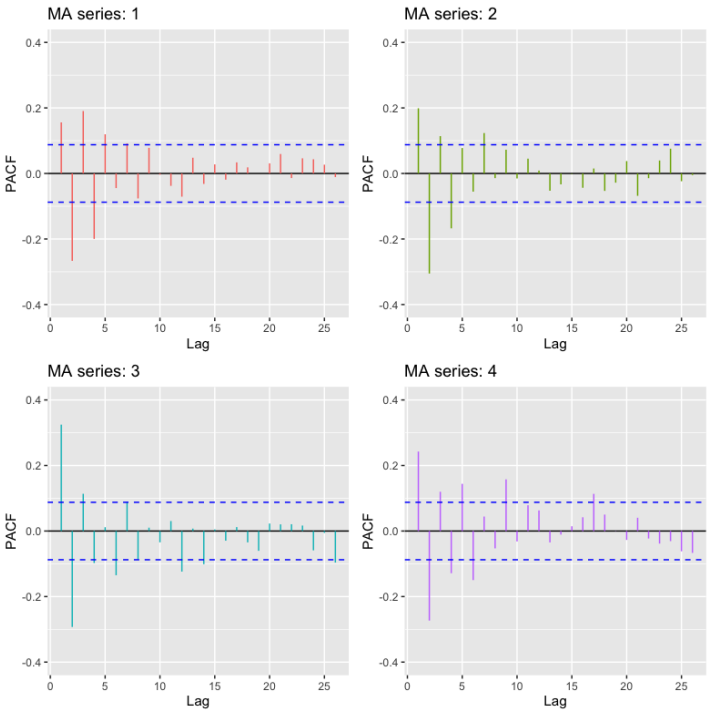

図 8: MA 過程の偏自己相関

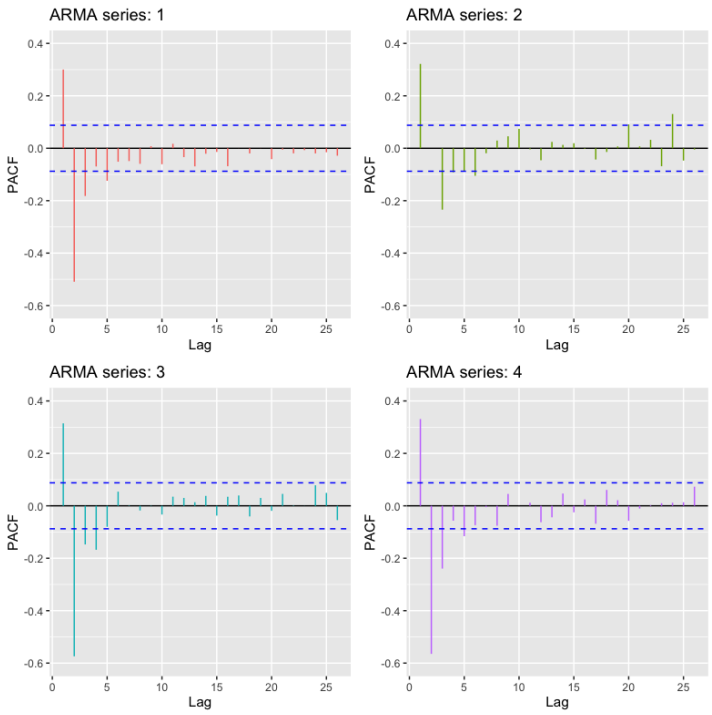

図 9: ARMA 過程の偏自己相関

非定常過程の変換

- 定常過程とみなせるように変換して分析
  - 階差の利用

$$X_t = X_{t-1} + \epsilon_t \quad \Rightarrow \quad Y_t = X_t - X_{t-1} = \epsilon_t$$

    - ランダムウォーク：階差をとるとホワイトノイズ (定常過程)
    - **ARIMA 過程**：階差をとると ARMA 過程になる確率過程
  - 対数変換の利用

$$X_t = (1 + \epsilon_t)X_{t-1} \quad \Rightarrow \quad Y_t = \log(X_t) - \log(X_{t-1}) = \log(1 + \epsilon_t) \simeq \epsilon_t$$

    - 対数変換と階差で微小な比率の変動を抽出

実習

R：AR モデルの推定

- 関数 `fabletools::model()`

```
model(.data, ...)
#' .data: 時系列データ (tsibble クラス)
#' ...: モデルを指定
```

- 関数 `fable::AR()`

```
AR(formula, ic = c("aicc", "aic", "bic"), ...)
#' formula: 時系列モデル
#' ic: モデル選択のための情報量規準
```

- stats::ar.ols() と同様の実装
- 典型的な使い方

```
toy_ar <- arima.sim(model = list(ar = c(0.7,-0.6, 0.5)),
                    n = 1000) |> as_tsibble()
toy_ar |> model(AR(value)) # モデルを自動選択する場合
toy_ar |> model(AR(value ~ order(3))) # モデルの次数を指定する場合
toy_ar |> model(AR(value ~ 0 + order(3))) # 平均項を含めない場合
```

  * 既定値では AIC(情報量規準) を用いて次数が自動的に決定される

R：ARIMA モデルの推定

- 関数 fable::ARIMA()

```
ARIMA(formula, ic = c("aicc", "aic", "bic"),
  stepwise = TRUE, greedy = TRUE, approximation = NULL, ...)
#' formula: 時系列モデル
#' ic: モデル選択のための情報量規準
#' stepwise/greedy/approximation: モデル探索のための設定
#' その他詳細は '?fable::ARIMA' を参照
```

- stats::arima() と同様の実装
- 典型的な使い方

```
toy_arima <- arima.sim(model = list(order = c(2,1,2),
                                    ar = c(0.8,-0.5),
                                    ma = c(-0.2,0.2)),
                       n = 1000) |> as_tsibble()
toy_arima |> model(ARIMA(value)) # 自動選択
toy_arima |> model(ARIMA(value ~ 0 + pdq(2,1,2))) # 次数を指定
```

R：モデルを扱うための補助的な関数

- 関数 fabletools::report()
  - モデルの情報

```
report(object, ...)
#' object: 推定された時系列モデル
```

  - fitted.values() でも可

- 関数 fabletools::tidy.mdl_df()
  - 係数の取得

```
tidy(object, ...)
#' object: 推定された時系列モデル
```

  - coef() でも可

- 関数 fabletools::augment.mdl_df()
  - あてはめ値・残差などの取得

```
augment(object, ...)
#' object: 推定された時系列モデル
```

- あてはめ値は `fitted()/fitted.values()` でも可
- 残差は `residuals()/resid()` でも可

R：モデルの評価

- 関数 `fabletools::accuracy.mdl_df()`
  - 精度評価の取得

```
accuracy(object, ...)
#' object: 推定された時系列モデル
```

- 関数 `fabletools::glance.mdl_df()`
  - 情報量規準などの取得

```
glance(object, ...)
#' object: 推定された時系列モデル
```

- 関数 `feasts::gg_tsresiduals()`

```
gg_tsresiduals(data, type = "innovation", ...)
#' data: 推定された時系列モデル
```

  - 描画には関数 `feasts::gg_tsdisplay()` が利用される
  - 典型的な使い方

```
toy_fit <- toy_arima |> model(ARIMA(value))
toy_fit |> accuracy()
toy_fit |> glance()
toy_fit |> gg_tsresiduals()
```

練習問題

- 以下の問に答えなさい
  - AR 過程を生成し，関数 `AR()` を用いて係数を推定しなさい
  - ARMA 過程を生成し，関数 `ARIMA()` を用いて係数を推定しなさい
  - 推定結果の妥当性を残差の自己相関係数を調べることによって確認しなさい

R：時系列クラスの作成 (再掲)

- 関数 `tsibble::tsibble()`

```
tsibble(..., key = NULL, index, regular = TRUE, .drop = TRUE)
#' ...: データ
#' key: index の補助情報 (同じ時間の異なるデータを表す)
#' index: 時間情報を表す列を設定
```

- 関数 `tsibble::as_tsibble()`

```
as_tsibble(x, key = NULL, index,
           regular = TRUE, validate = TRUE, .drop = TRUE, ...)
#' x: データ (時系列オブジェクトやデータフレーム)
```

  - 典型的な使い方

```r
tsibble(date = as_date("2024-01-01") + 0:9,
        value = rnorm(10))
tibble(year = 2001:2020,
       value = rnorm(20)) |>
  as_tsibble(index = year) # year を時間情報に指定
AirPassengers |> as_tsibble() # 時系列オブジェクトの変換
```

R：部分系列の切り出し

- 関数 `tsibble::filter_index()`

```r
filter_index(.data, ..., .preserve = FALSE)
#' .data: tsibble オブジェクト
#' ...: 区間を表す式 (~ end, start ~ end, start ~ .)
```

  - 典型的な使い方

```r
AirPassengers |>
  as_tsibble() |>
  filter_index("1955-10" ~ "1956-03")
```

練習問題

- 東京の気候データを用いて以下の問に答えなさい

```r
tw_data <- read_csv("data/tokyo_weather.csv")
```

  - 気温のデータを `tsibble` クラスに変換しなさい
  - 気温のデータおよびその階差の性質を検討しなさい
  - ARIMA モデルを作成しなさい

モデルによる予測

ARMA モデルによる予測

- 推定したモデルを用いて n 期先を予測
  - AR モデル：観測時点までの観測値を用いて回帰
  - MA モデル：観測時点までのホワイトノイズで回帰
  - ARMA モデル：上記の複合
- いずれも n が大きいと不確定性が増大
- 階差による変換は累積 (階差の逆変換) により推定

分解モデルによる予測

- トレンド成分+季節成分+ランダム成分への分解

$$X_t = T_t + S_t + R_t$$

あるいは

$$X_t = T_t \times S_t \times R_t \qquad (\log X_t = \log T_t + \log S_t + \log R_t)$$

  - トレンド成分：時間の関数やランダムウォークなどを想定
  - 季節成分：周期的な関数を想定

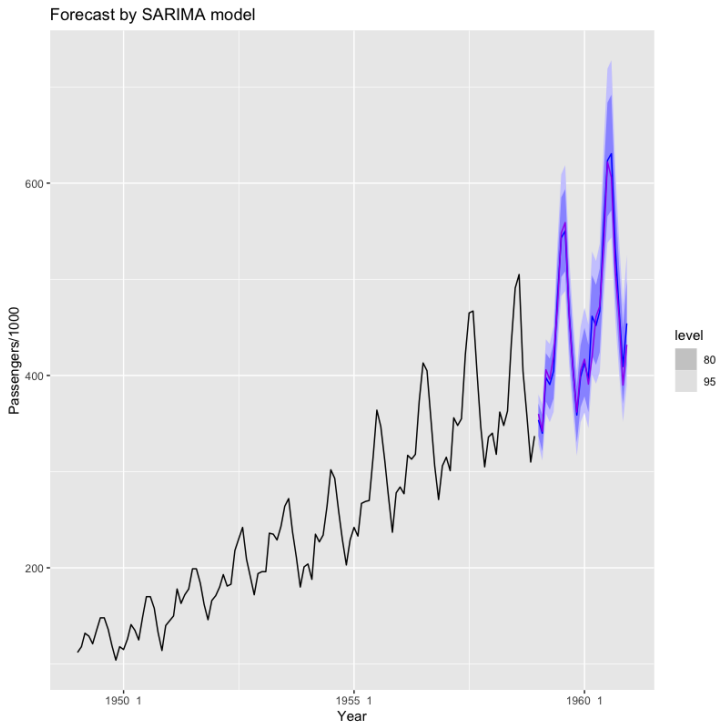

図 10: ARIMA モデル (階差あり ARMA) による予測

  - ランダム成分 : ARMA モデルなどを想定
- 分解の考え方
  - ランダム成分 : 適切な幅の移動平均が 0
  - 季節成分 : 1 周期の平均が 0

解析事例

COVID-19 の感染者数の分析

- 厚生労働省の COVID-19 のデータ
  - 陽性者数 (新規・累積)
  - 重症者数 (推移・性別・年齢別)
  - 死者数 (推移・性別・年齢別・累積)
  - 入院治療等を要する者等推移
  - 集団感染等発生状況
- 以下の解析で用いるデータ
  - 日毎の全国・各都道府県の新規陽性者数 (感染者数) https://covid19.mhlw.go.jp/public/opendata/newly_confirmed_cases_daily.csv

感染者数の推移

第 3 波における感染者数の推移

基礎分析 (分析対象 : 2020/9/15-11/30)

ARIMA モデルによる推定

- 推定された ARIMA モデル

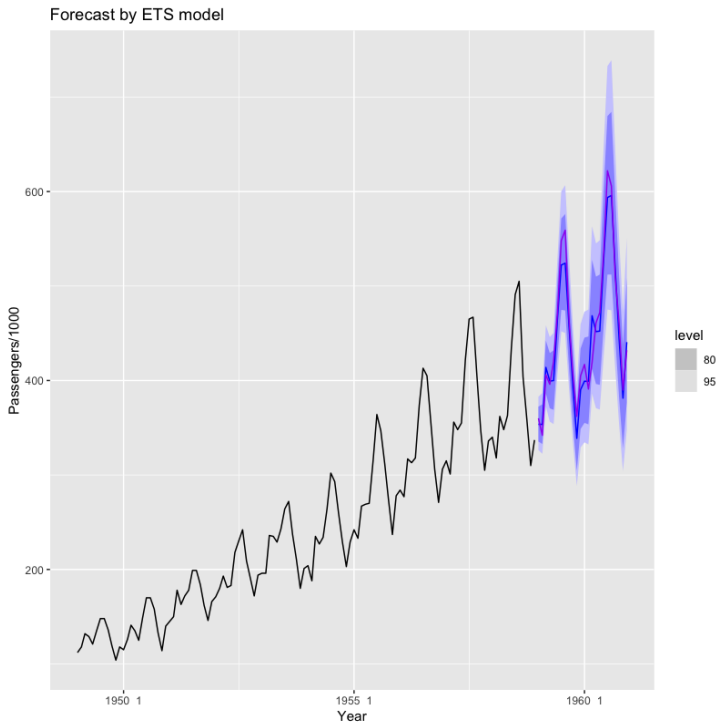

図 11: ETS モデル (expornential smoothing) による予測

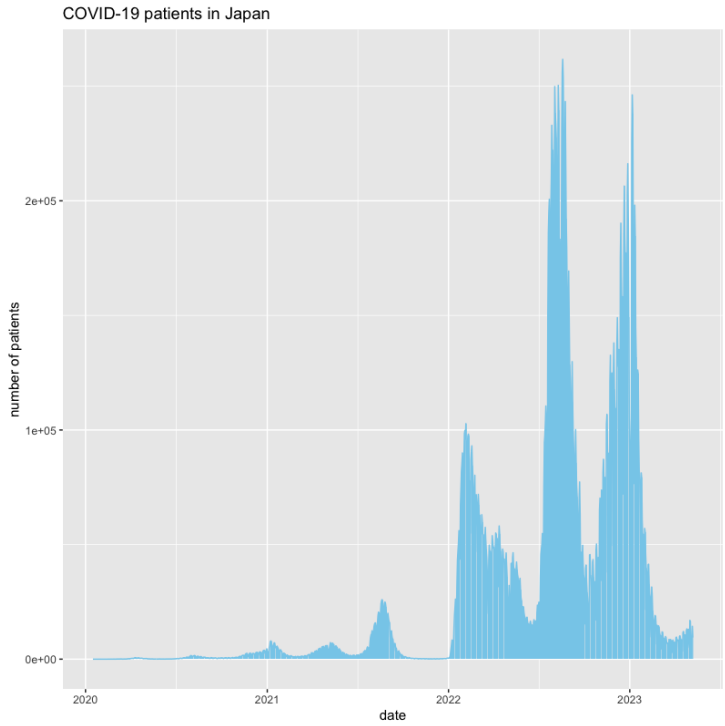

図 12: 全国の感染者数

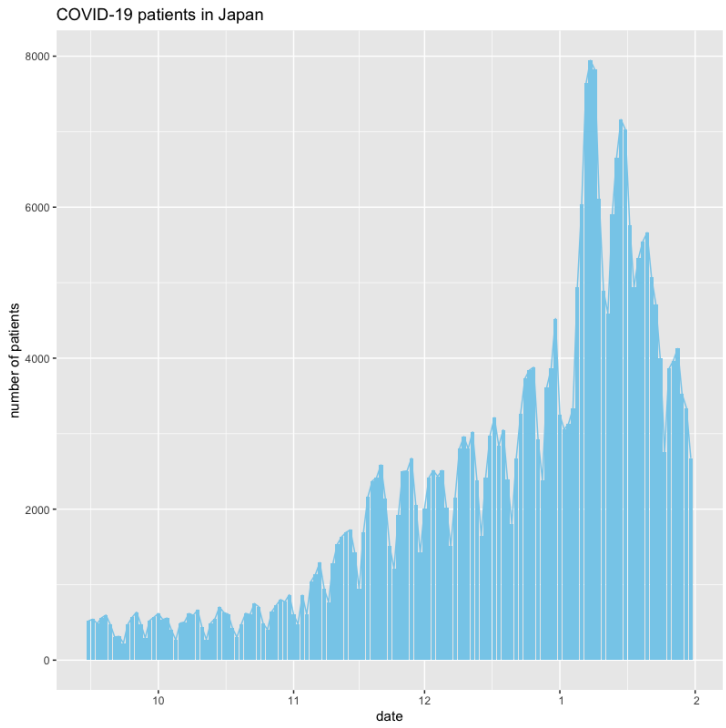

図 13: 第 3 波の感染者数

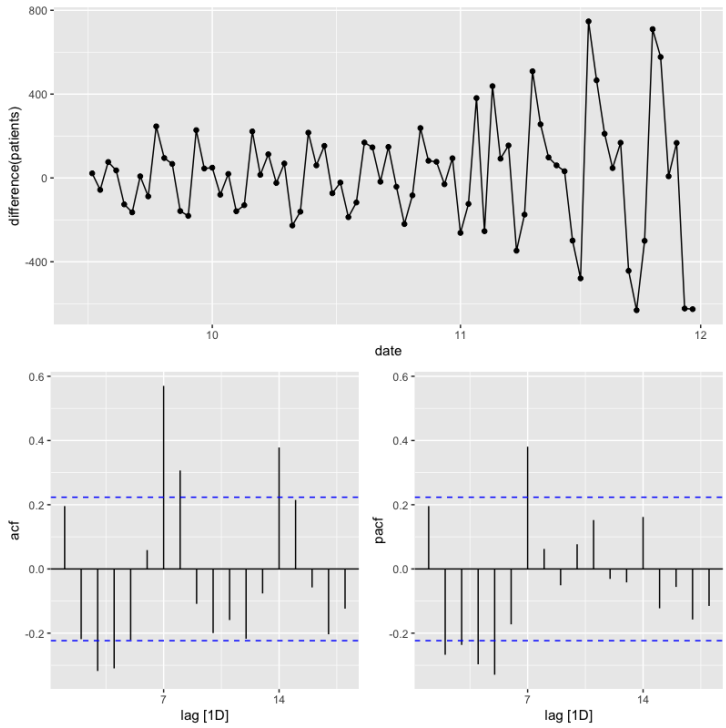

図 14: 時系列 (階差)

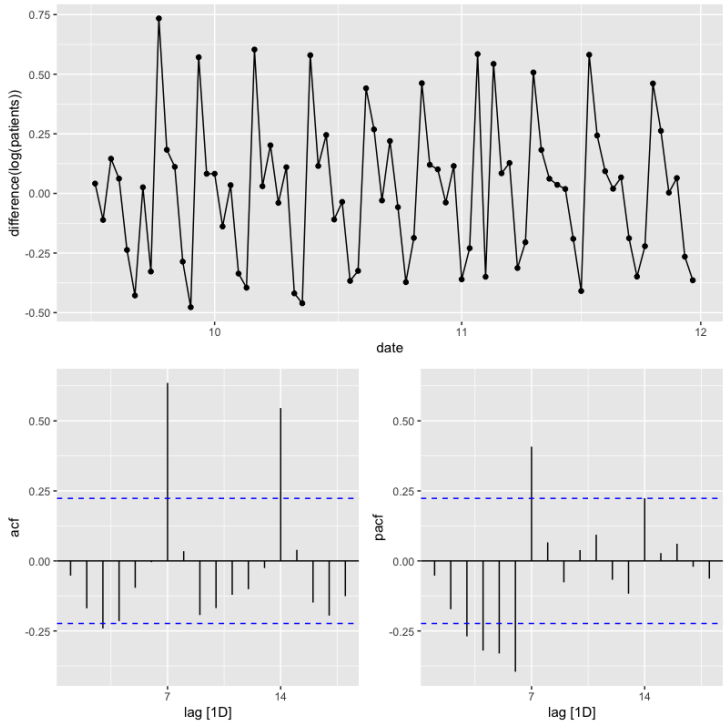

図 15: 時系列 (対数変換+階差)

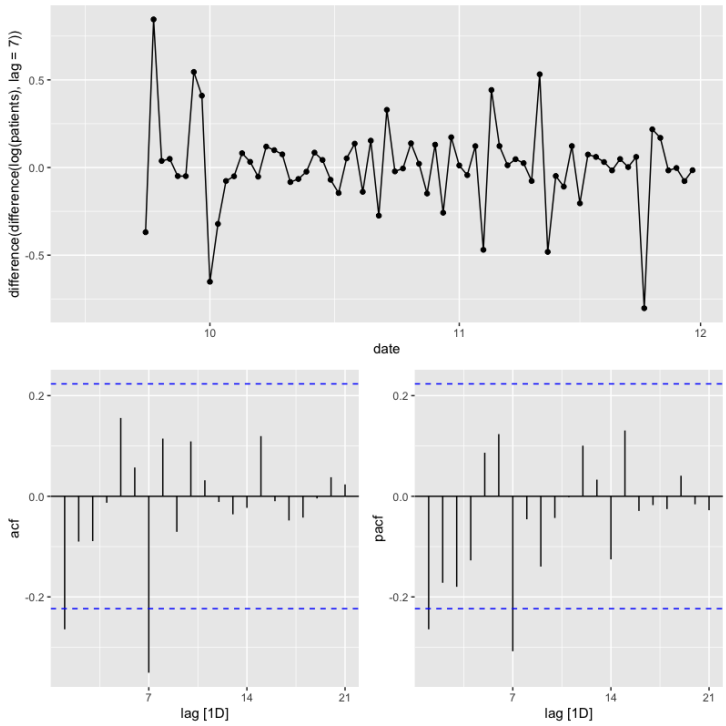

図 16: 時系列 (対数変換+階差+7 日階差)

```
Series: patients
Model: ARIMA(1,1,1)(2,0,0)[7]
Transformation: log(patients)

Coefficients:
         ar1      ma1    sar1    sar2
      0.4493  -0.8309  0.3709  0.4232
s.e.  0.1635   0.0981  0.1212  0.1353

sigma^2 estimated as 0.03811:  log likelihood=15.04
AIC=-20.07   AICc=-19.21   BIC=-8.42
```

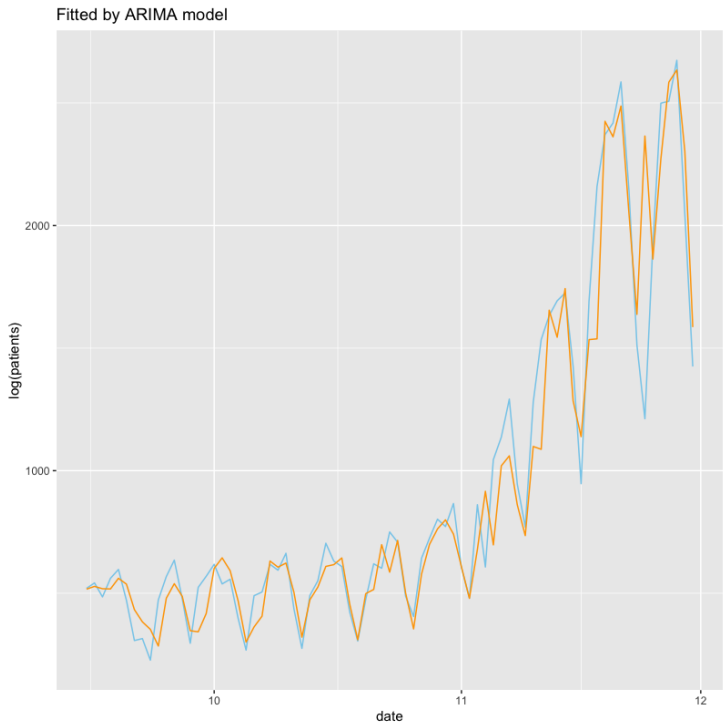

図 17: あてはめ値

まとめ

- 感染者数の推移は非定常なデータ
- 構造が時不変と考えられる区間を捉えれば
  - 時系列の適切な変換 (指数的な増大のため対数変換)
  - 基本的な ARMA モデル (階差系列に ARMA モデルを適用)

  の組み合わせである程度の分析は可能

実習

R：時系列の予測

- 関数 `fable::forecast()`

```
forecast(object, h, ...)
#' object: モデルの推定結果
```

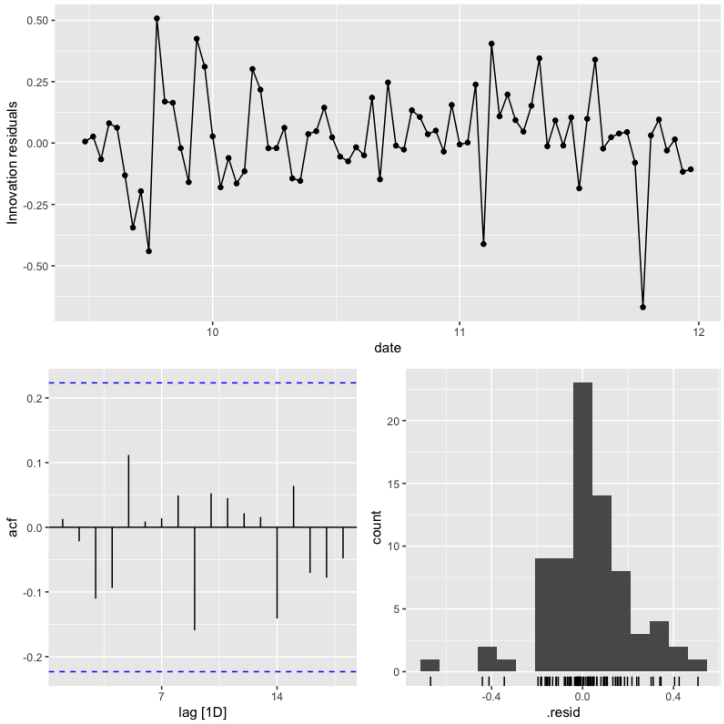

図 18: 診断プロット

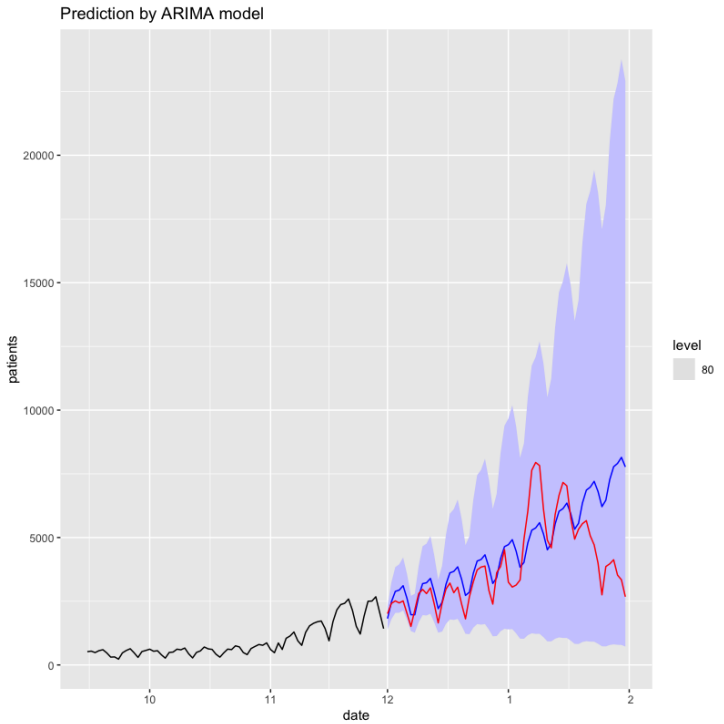

図 19: 予測値 (60 日分，80%信頼区間)

```
#' h: h 期先の予測
```

- 詳細は ?fable::forecast.X (X はモデル名) を参照
- 典型的な使い方

```
as_tsibble(AirPassengers) |>
  model(ARIMA(log(value))) |>
  forecast(h = 36) |> autoplot(AirPassengers)
```

  * 描画には fabletools::aotoplot.fbl_ts() が利用される

R：分解モデルの推定

- 関数 fable::ETS()

```
ETS(formula, opt_crit = c("lik", "amse", "mse", "sigma", "mae"),
    nmse = 3, bounds = c("both", "usual", "admissible"),
    ic = c("aicc", "aic", "bic"), restrict = TRUE, ...)
#' formula: 時系列モデル
#' opt_crit: モデルの最適化の指標
#' ic: モデル選択のための情報量規準
#' その他詳細は '?fable::ETS' を参照
```

- 典型的な使い方

```
as_tsibble(AirPassengers) |>
  model(ETS(value ~ season("M"))) |>
  components() |> autoplot()
```

練習問題

- 以下の問に答えなさい
  - COVID-19 のデータを用いて分析・予測を行いなさい https://covid19.mhlw.go.jp/public/opendata/newly_confirmed_cases_daily.csv
  - AirPassengers データを用いて分析・予測を行いなさい