時系列解析

モデルの推定と予測

村田 昇

講義概要

- 第 1 日 : 時系列の基本モデル
- **第 2 日 : 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

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

- **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

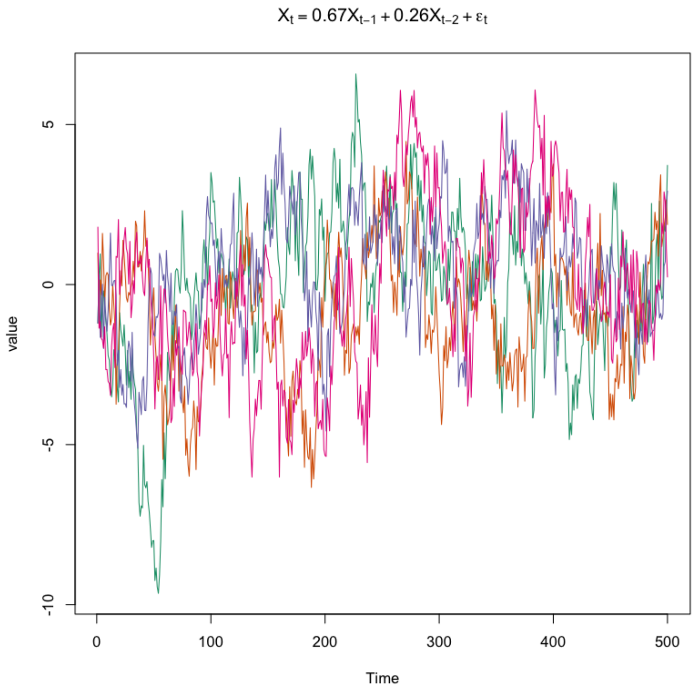

図 1: 同じモデルに従う AR 過程の例

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

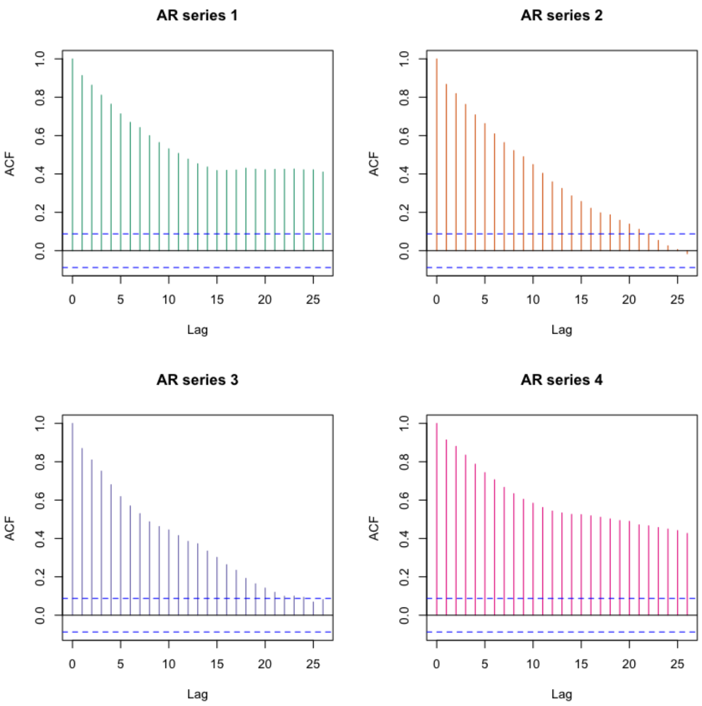

図 2: AR 過程の自己相関

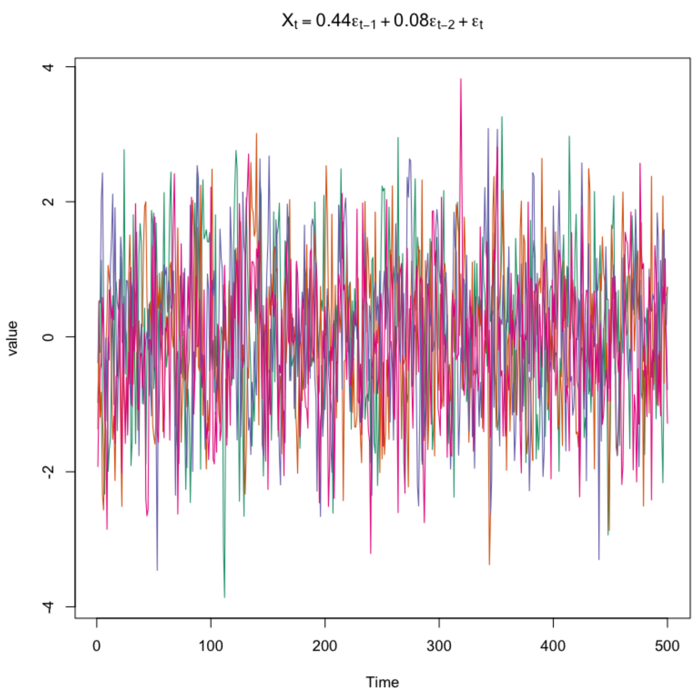

図 3: 同じモデルに従う MA 過程の例

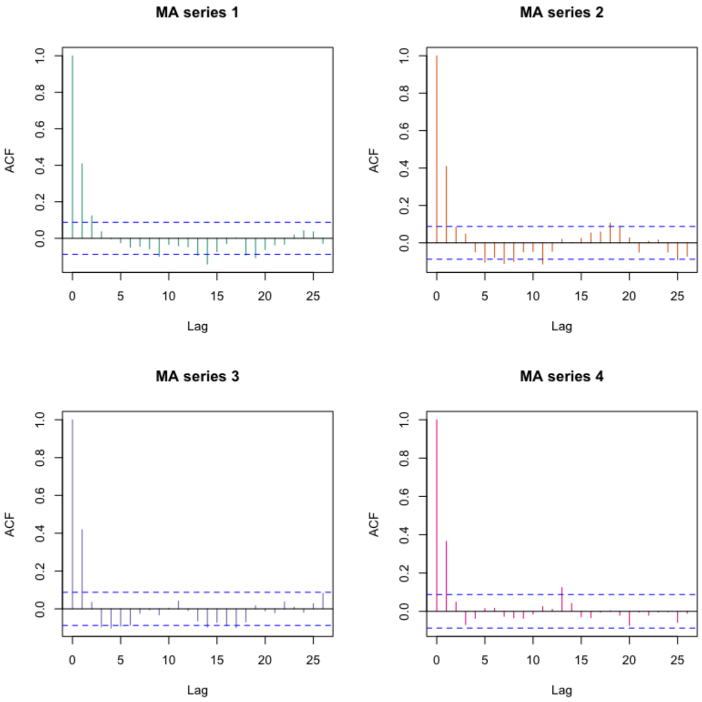

図 4: MA 過程の自己相関

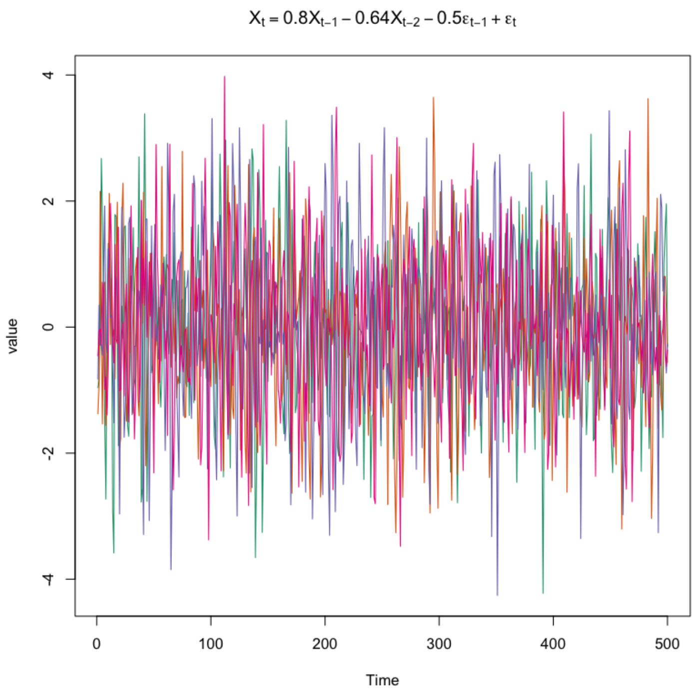

図 5: 同じモデルに従う ARMA 過程の例

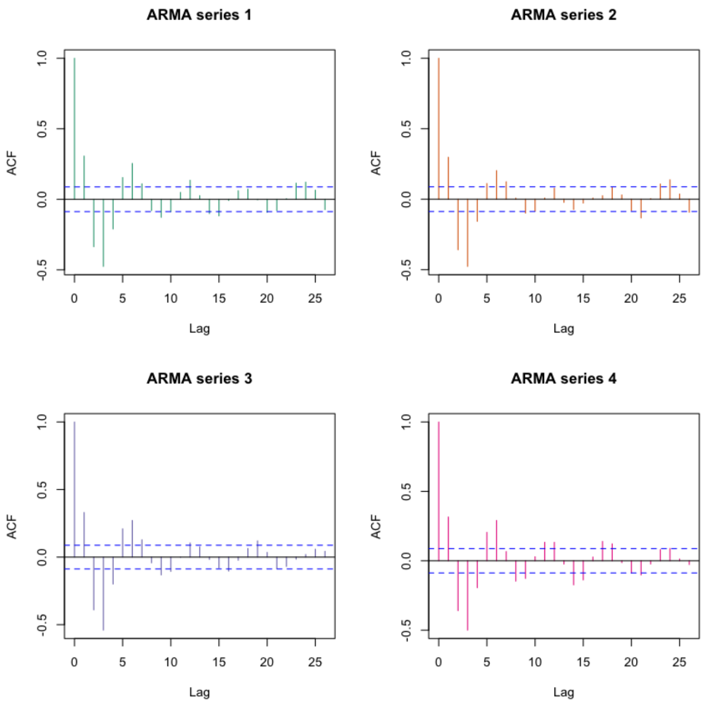

図 6: ARMA 過程の自己相関

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

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

  - 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**)

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

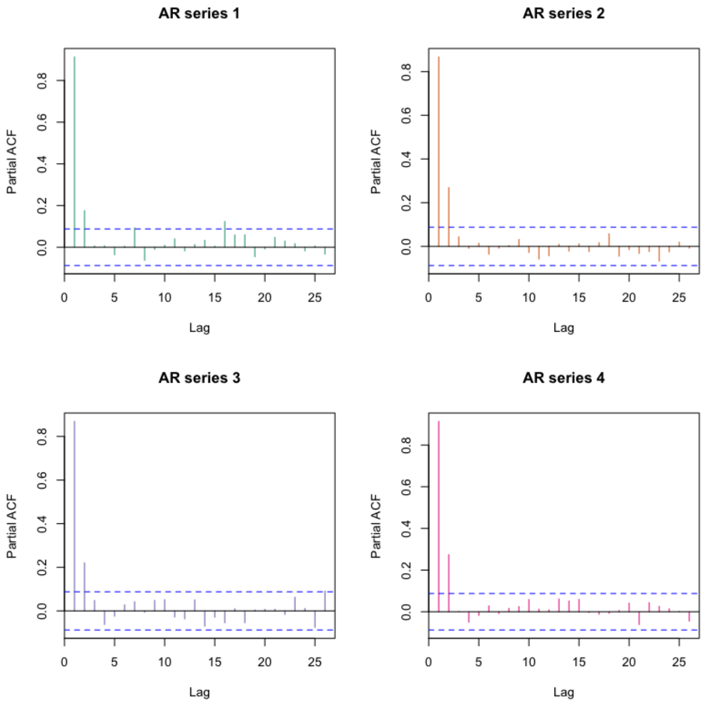

図 7: AR 過程の偏自己相関

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
  - 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

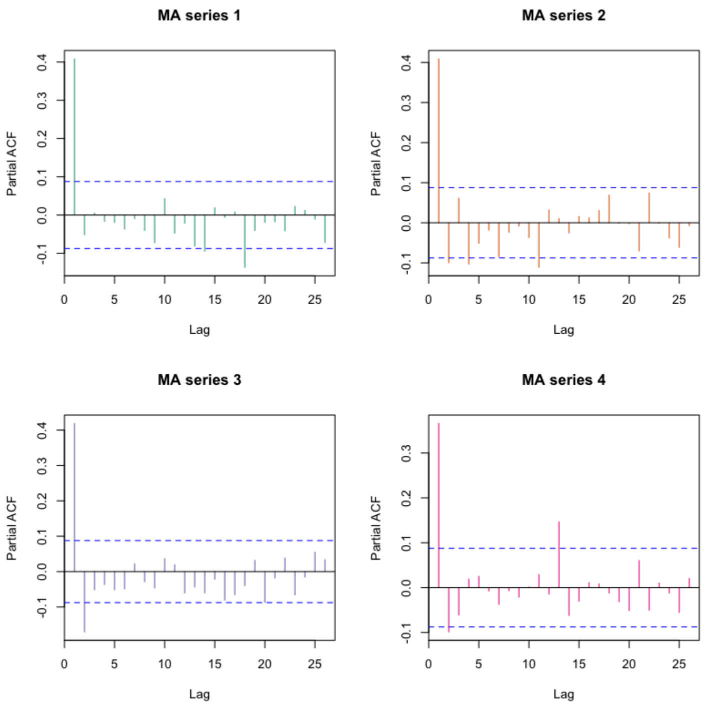

図 8: MA 過程の偏自己相関

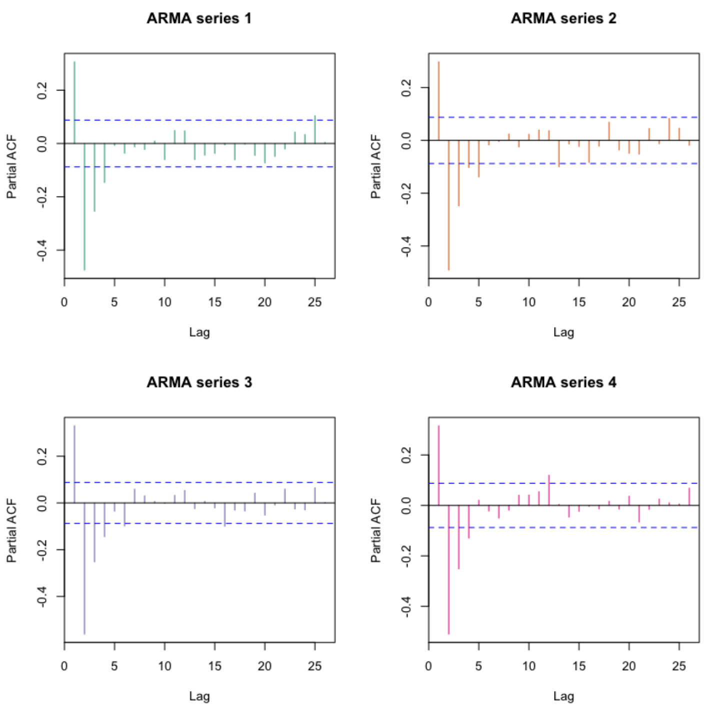

図 9: ARMA 過程の偏自己相関

実習

R：AR モデルの推定 `ar()`

- 定常 AR モデルのあてはめ

```
ar(x, aic = TRUE, order.max = NULL, method = "yule-walker")
## x: 時系列データ (ts クラスが望ましい)
## aic: AIC を計算するか否か
## order.max: 計算すべき最大次数 (既定値は時系列の長さから規則的に計算)
## method: 計算方法, 他に "burg", "ols", "mle" が指定できる
est <- ar(x) # 時系列 x の係数を推定する
acf(resid(est)) # 残差 (epsilon の推定値に相当) がホワイトノイズか確認する
```

  - AIC を用いて次数を自動決定することができる

R：ARIMA モデルの推定 `arima()`

- 階差系列への定常 ARMA モデルのあてはめ: `stats::arima()`

```
arima(x, order = c(0L, 0L, 0L),
      seasonal = list(order = c(0L, 0L, 0L), period = NA))
## x: 時系列データ (ts クラスが望ましい)
## order: 次数 c(AR,階差,MA)
## seasonal: 次数と期間 list(order=(AR,階差,MA),period=期間)
```

  - 関数 `arima()` には次数の決定機能はない
  - 試行錯誤による次数の決定が必要

R：ARIMA モデルの推定 `forecast::auto.arima()`

- パッケージ `forecast` の利用

```
## 右下ペインの package タブから forecast をインストール
## install.packages("forecast")
library(forecast)
auto.arima(x, d=1, D=1)
## x: 時系列データ
## d: 定常化のための階差回数
## D: 季節成分 (周期成分) の定常化のための階差回数
```

  - 次数を自動決定することができる

練習問題

- 先週作成した関数 `myARMA()` を利用して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()` および関数 `auto.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

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

R：関数 `zoo()`

- 時系列クラス `zoo` を作成する関数

```
library(zoo) # forecast を利用すると自動的に読み込まれる
zoo(x = NULL, order.by = index(x), frequency = NULL, ...)
## x: ベクトル，行列
## order.by: 成分の目盛
## frequency: 季節成分の周期
x.zoo <- zoo(x, # データに日付の情報を付加する例 (Date クラスで指定)
             order.by = seq(from=as.Date("2021-01-01"),
                            to=as.Date("2021-12-31"), by=1))
start(x.zoo) # index(x.zoo)[1] 最初の日付
end(x.zoo) # index(x.zoo)[length(x.zoo)] 最後の日付
```

  - その他の詳細は `?zoo` を参照

R：関数 `window()`

- 時系列から部分系列を切り出す関数

```
window(x, start = NULL, end = NULL)
## x: ベクトル，行列
## start: 開始時点
## end: 終了時点
window(x, # データに日付の情報が入っている場合 (zoo の例)
       start="2021-12-01", # Date クラスの標準の書き方
       end="2021/12/31") # Date クラスはこちらでも解釈可能
```

  - その他の詳細は `?stats::window` を参照

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

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

  - 気温のデータを `zoo` 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
  - 関数 `auto.arima()` を用いてモデルを作成しなさい

モデルによる予測

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

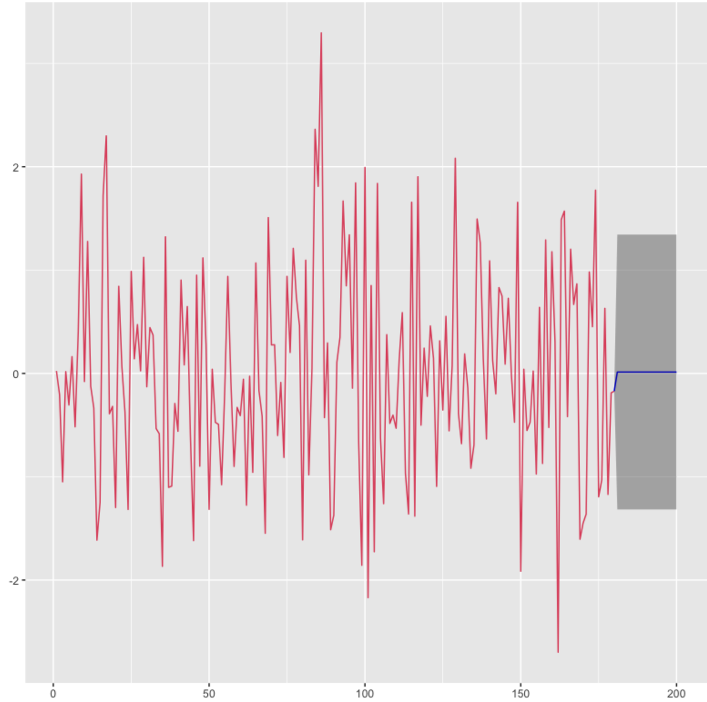

図 10: ホワイトノイズの予測

図 11: ランダムウォークの予測

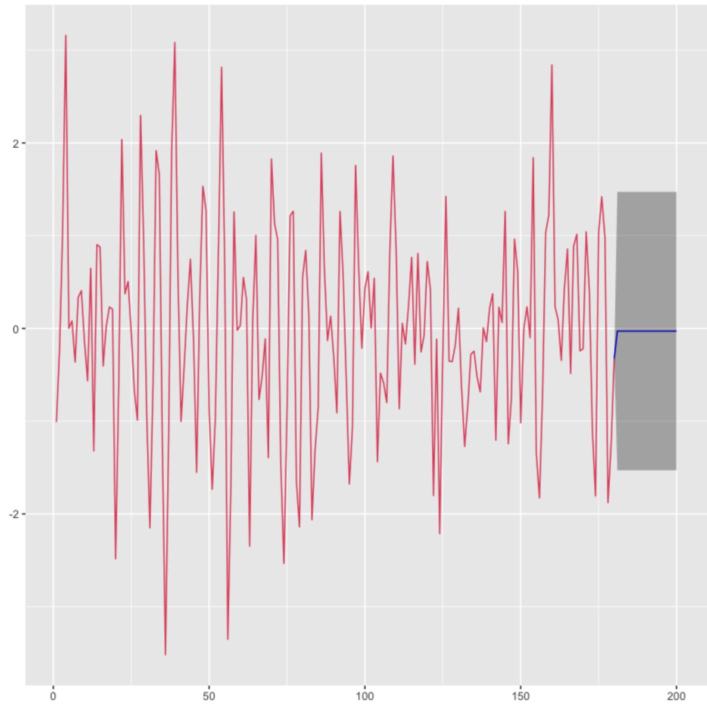

図 12: ARMA 過程の予測

分解による予測

- トレンド成分+季節成分+ランダム成分への分解

$$X_t = T_t + S_t + R_t$$

  - 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
  - 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

実習

R : 時系列の予測 `predict()`

- 推定されたモデルによる予測

```
predict(object, newdata, n.ahead = 1, se.fit = TRUE, ...)
## object: ar または arima による推定結果
## newdata: 予測対象のデータ (ar の場合のみ)
## n.ahead: n 期先の予測
## se.fit: 標準誤差を付加するか否か
x.fit <- arima(x, order=c(0,1,1),
               seasona=list(order=c(0,1,1), period=12))
x.prd <- predict(x.fit, n.ahead=10)
x.prd$pred # 予測値 (標準誤差は $se)
```

  - 詳細は `?predict.ar`, `?precidt.Arima` を参照

R：時系列の予測 forecast()

- パッケージ forecast の利用

```
forecast(object, h)
## object: ar または arima による推定結果
## h: h 期先の予測 (指定しないと 2 周期または 10 期先を予測)
x.fit <- auto.arima(x, d=1, D=1)
x.prd <- forecast(x.fit, h=10)
x.prd$mean # 予測値 (信頼区間は $upper/$lower)
plot(x.prd) # 全体を視覚化
```

  - 詳細は ?forecast を参照

R：時系列の分解 StructTS()

- トレンドの構造を仮定して分解する関数

```
StructTS(x, type = "level", fixed = NULL, ...)
## x: 時系列データ
## type: "level" 平均の変動をランダムウォークでモデル化
##       "trend" 平均と傾きをランダムウォークでモデル化
##       "BSM" 季節成分を含むモデル (frequency が必要)
## fixed: ホワイトノイズの分散の指定
x.sts <- StructTS(x, type = "trend", fixed = c(0.1,NA,NA))
## 平均のホワイトノイズの分散を 0.1, 傾きとランダム成分の分散は推定
forecast(x.sts, h=10) # predict を使うことも可
```

  - 詳細は ?StructTS を参照
  - 分解を行う関数は decompose, stl などもある

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
  - 6 月までのデータを用いて適切なモデルを推定しなさい
  - 7 月のデータの推定を行いなさい

練習問題

- 以下の問に答えなさい
  - AirPassengers データを用いて分析・予測を行いなさい
  - 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