

インサイト・ファクトリー 社内セミナー
2020年4月

マーケティング・ミックス・モデリング

VI. 動学的市場反応モデル (1)

小野 滋

統計学・データ解析	マーケティング・ミックス・モデリングへの適用
I. イントロダクション	
	II. 市場反応モデルとは
III. 回帰分析の基礎	
	IV. 静学的市場反応モデル
V. 時系列分析の基礎	
	VI. 動学的市場反応モデル(1)
VII. 状態空間モデルの基礎	
	VIII. 動学的市場反応モデル(2)

目次

この章の内容

1. 繰越効果と予期効果
2. 繰越効果の表現
3. 動的回帰モデル
4. ストック変数
5. ベクトル自己回帰モデル
6. 時間集計に伴うバイアス

この章の引用文献

この章に登場するRの関数

Appendix. 動的回帰モデルのためのRパッケージ

v章を踏まえ、時系列データに基づく市場反応モデルを構築する方法について考えます

特に、マーケティング・ミックス変数の効果の時間的遅延(繰越効果)に焦点を当てます

回帰モデル

$$Y_i = \beta_1 + \beta_2 X_{2i} + \dots + \beta_K X_{Ki} + U_i$$

[1] X_i は確率変数でないか
 U_i と統計的に独立

[2] $E(U_i) = 0$

[3] $\text{Var}(U_i) = \sigma^2$

[4] $\text{Cov}(U_i, U_j) = 0, i \neq j$

[5] U_i は正規分布に従う

[6] $\mathbf{X}$ の列ベクトルは一次独立。
列数は行数より小さい



市場反応の特性

H) マーケティング活動が与える効果は、即時的に現れることもあれば、時間的遅延とともに現れることもある

本資料作成に用いたすべてのRコードを、以下で公開しています:

https://rpubs.com/shig_ono/MMM_6

1. 繰越効果と予期効果

■ 繰越効果 carryover effect

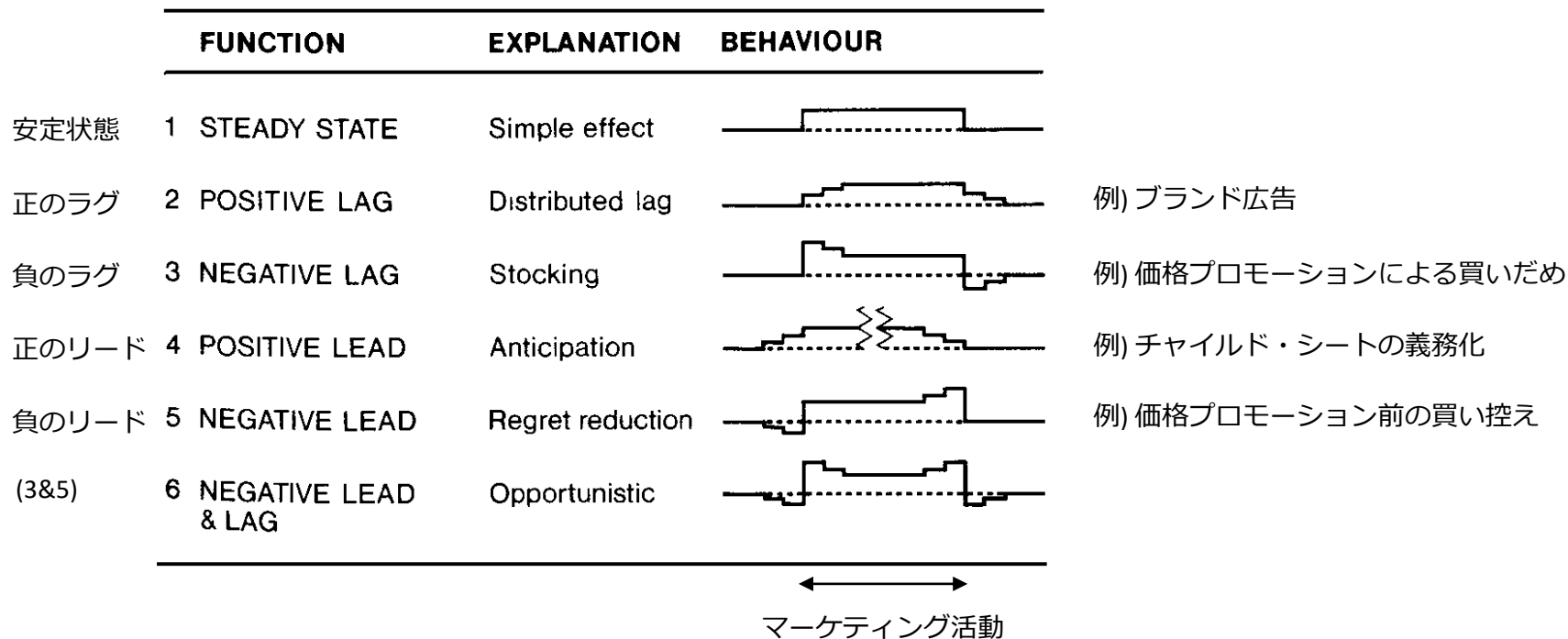
- マーケティング・ミックス変数の**現在の値が将来の売上に与える効果**
- 遅延反応 delayed response effect
 - マーケティング活動から、それによる売上の実現までに遅延が生じている
 1. 決定から実行までの遅延
 2. 実行から認知・態度変容までの遅延
 3. 態度変容から注文までの遅延
 4. 注文から売上までの遅延
- 顧客持越 customer holdover effect
 - マーケティング活動によって獲得した新規購買者が再購買している
 - マーケティング活動によって顧客の平均購買量が増大している

■ 予期効果 anticipatory response effect

- マーケティング・ミックス変数の**将来の値が現在の売上に与える効果**
- 将来のマーケティング活動を見越して、顧客や競合他社が行動を調整している

■ 繰越・予期の基本的パターン (Doyle & Saunders, 1985)

Figure 1
SIX DYNAMIC RESPONSE MODELS



- 以下では、正の繰越効果についてのみ考える
 - ・ 上の図でいう 1 と 2 についてのみ考える

2. 繰越効果の表現

以下では説明を簡単にするために、説明変数はひとつ (X_t) とする。

■ 現在効果モデル

定常時系列 Y_t, X_t について、

$$Y_t = \alpha + \beta X_t + \varepsilon_t$$

ただし ε_t はホワイトノイズ。

- X_t が1単位増えたときの、 $Y_t, Y_{t+1}, \dots$ への効果は？
 Y_t に対して β 。繰越効果なし

Y_{t+k} への効果



■ 以下では、繰越効果を表現するためのさまざまな工夫について紹介する：

1. 攪乱項に繰越効果を与える工夫

A. **ARMA誤差回帰モデル**

2. 説明変数に繰越効果を与える工夫

B. **分布ラグモデル** (直接ラグモデル)

C. **幾何級数型分布ラグモデル** (コイック・ラグモデル)

D. **負の二項分布型分布ラグモデル**

E. **多項分布ラグモデル** (アルモン・ラグモデル)

3. 攪乱項と説明変数の両方に繰越効果を与える工夫

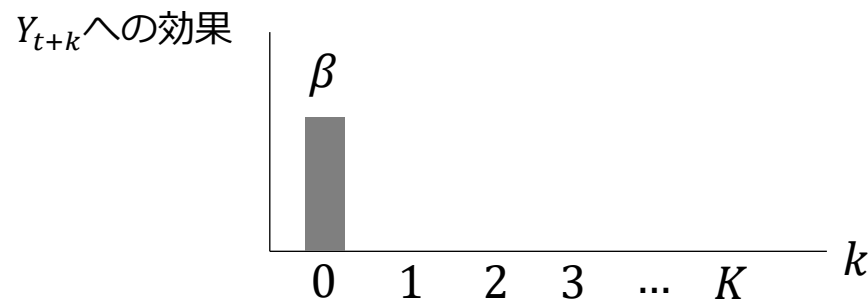
F. **部分調整モデル**

G. **自己回帰分布ラグモデル** (ADLモデル)

A. ARMA誤差回帰モデル (cf. V章)

$$Y_t = \alpha + \beta X_t + \frac{1 + \theta_1 L \dots + \theta_q L^q}{1 - \phi_1 L - \dots - \phi_p L^p} \varepsilon_t$$

- 撓乱項で表現されている未知の要因について繰越効果を表現していると解釈できる
- X_t が1単位増えたときの、 $Y_t, Y_{t+1}, \dots$ への効果は？
 - 現在効果モデルと同じく、 Y_t に対して β
 - 繰越効果なし



B. 分布ラグモデル distributed lag model

$$\begin{aligned} Y_t &= \alpha + \beta_0 X_t + \beta_1 X_{t-1} + \cdots + \beta_K X_{t-K} + \varepsilon_t \\ &= \alpha + \sum_{k=0}^K \beta_k X_{t-k} + \varepsilon_t \end{aligned}$$

- X_t の効果は、 $Y_t, Y_{t+1}, \dots, Y_{t+K}$ にまたがって分布する
- 繰越効果のもっとも一般的な表現
 - k 次繰越効果を直接に表現している
- **直接ラグモデル**とも呼ぶ

$y_3 = \alpha + \beta_0 x_3 + \beta_1$	x_2	$+$	β_2	x_1	$\dots + \varepsilon_3$
$y_4 = \alpha + \beta_0 x_4 + \beta_1$	x_3	$+$	β_2	x_2	$\dots + \varepsilon_4$
$y_5 = \alpha + \beta_0 x_5 + \beta_1$	x_4	$+$	β_2	x_3	$\dots + \varepsilon_4$
$\dots$					

ラグ変数

分布ラグモデルのイメージ

- ラグ演算子を使って書き直すと

$$Y_t = \alpha + \beta_0 \left(1 + \frac{\beta_1}{\beta_0} L + \frac{\beta_2}{\beta_0} L^2 + \cdots + \frac{\beta_K}{\beta_0} L^K \right) X_t + \varepsilon_t$$

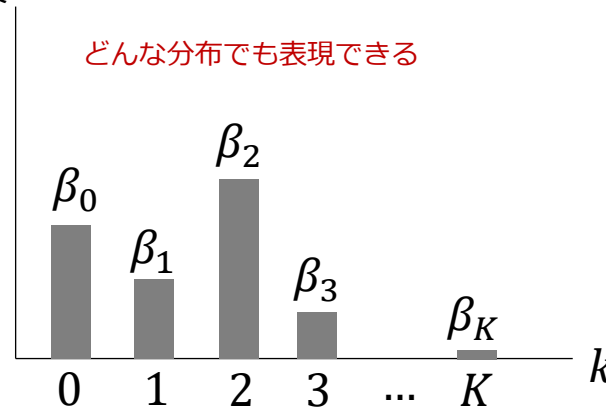
MAの形式になっている

- 参考：MA(q)過程

$$Y_t = \mu + (1 + \theta_1 L + \theta_2 L^2 + \cdots + \theta_q L^q) \varepsilon_t$$

- X_t が1単位増えたときの、 $Y_t, Y_{t+1}, \dots$ への効果は？

Y_{t+k} への効果



効果の合計は
(時間割引を無視すれば)

$$\sum_{k=0}^K \beta_k$$

- 直接ラグモデルの問題点
 - 最大ラグ数 K をいくつにすればよいかわからない
 - 長期的効果を表現する際、パラメータの数が多すぎる
 - 推定時に多重共線性が生じる

C. 幾何級数型分布ラグモデル geometric distributed lag model

$$\begin{aligned} Y_t &= \alpha + \beta(X_t + \lambda X_{t-1} + \lambda^2 X_{t-2} + \cdots) + \varepsilon_t \\ &= \alpha + \beta \sum_{k=0}^{\infty} \lambda^k X_{t-k} + \varepsilon_t \quad (0 < \lambda < 1) \end{aligned}$$

- λ を**繰越率** retention rate と呼ぶ
 - 効果は1時点ごとに λ 倍に減少する
- マーケティング活動の繰越効果の表現方法として、古くからもっともよく用いられている
- **コイック・ラグモデル**とも呼ばれる
- 特徴
 - パラメータ数が1増えるだけ
 - 最大ラグ数を決めなくてよい

コイック変換

- 幾何級数型分布ラグモデルは、一種の自己回帰モデルに書き換えることができる

前頁の式と同じ。
現在効果 βX_t を別の項に
している

$$Y_t = \alpha + \beta X_t + \beta \sum_{k=1}^{\infty} \lambda^k X_{t-k} + \varepsilon_t$$

前頁の式を1時点戻し
て $-\lambda$ を掛ける

$$-\lambda Y_{t-1} = -\lambda \alpha - \lambda \beta \sum_{k=0}^{\infty} \lambda^k X_{t-1-k} - \lambda \varepsilon_{t-1}$$

$$\begin{aligned} &= \beta \sum_{k=0}^{\infty} \lambda^{k+1} X_{t-1-k} \\ &= \beta \sum_{k=1}^{\infty} \lambda^k X_{t-k} \end{aligned}$$

上の2本の式を足す
と...

$$Y_t = (1 - \lambda)\alpha + \beta X_t + \lambda Y_{t-1} + (\varepsilon_t - \lambda \varepsilon_{t-1})$$

Y_t の自己回帰モデルに
なっている！



Leendert Marinus Koyck (1918–1962)
オランダの経済学者だそうです。
人名だったのか...知りませんでした

- ラグ演算子を使って書き直すと

$$(1 - \lambda L)Y_t = (1 - \lambda)\alpha + \beta X_t + (1 - \lambda L)\varepsilon_t$$

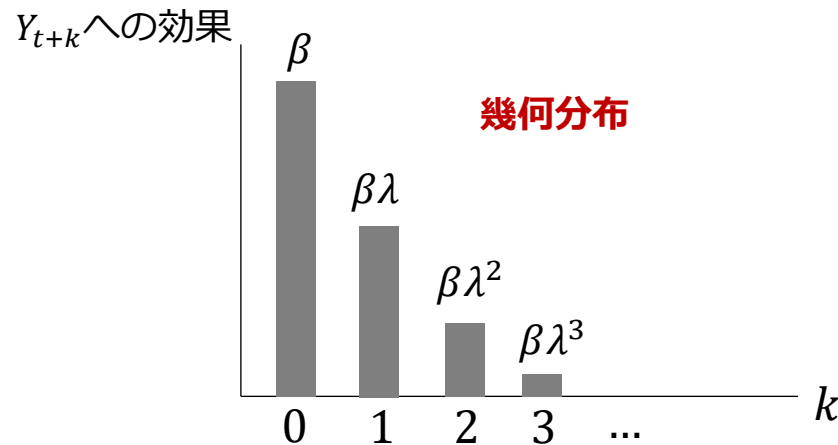
$$Y_t = \alpha + \beta \frac{1}{1 - \lambda L} X_t + \varepsilon_t$$

ARの形式になっている

- 参考: AR(1)過程

$$Y_t = \mu + \frac{1}{1 - \phi_1 L} \varepsilon_t$$

- X_t が1単位増えたときの、 $Y_t, Y_{t+1}, \dots$ への効果は？



効果の合計は

$$\beta \sum_{k=0}^{\infty} \lambda^k = \frac{\beta}{1 - \lambda}$$

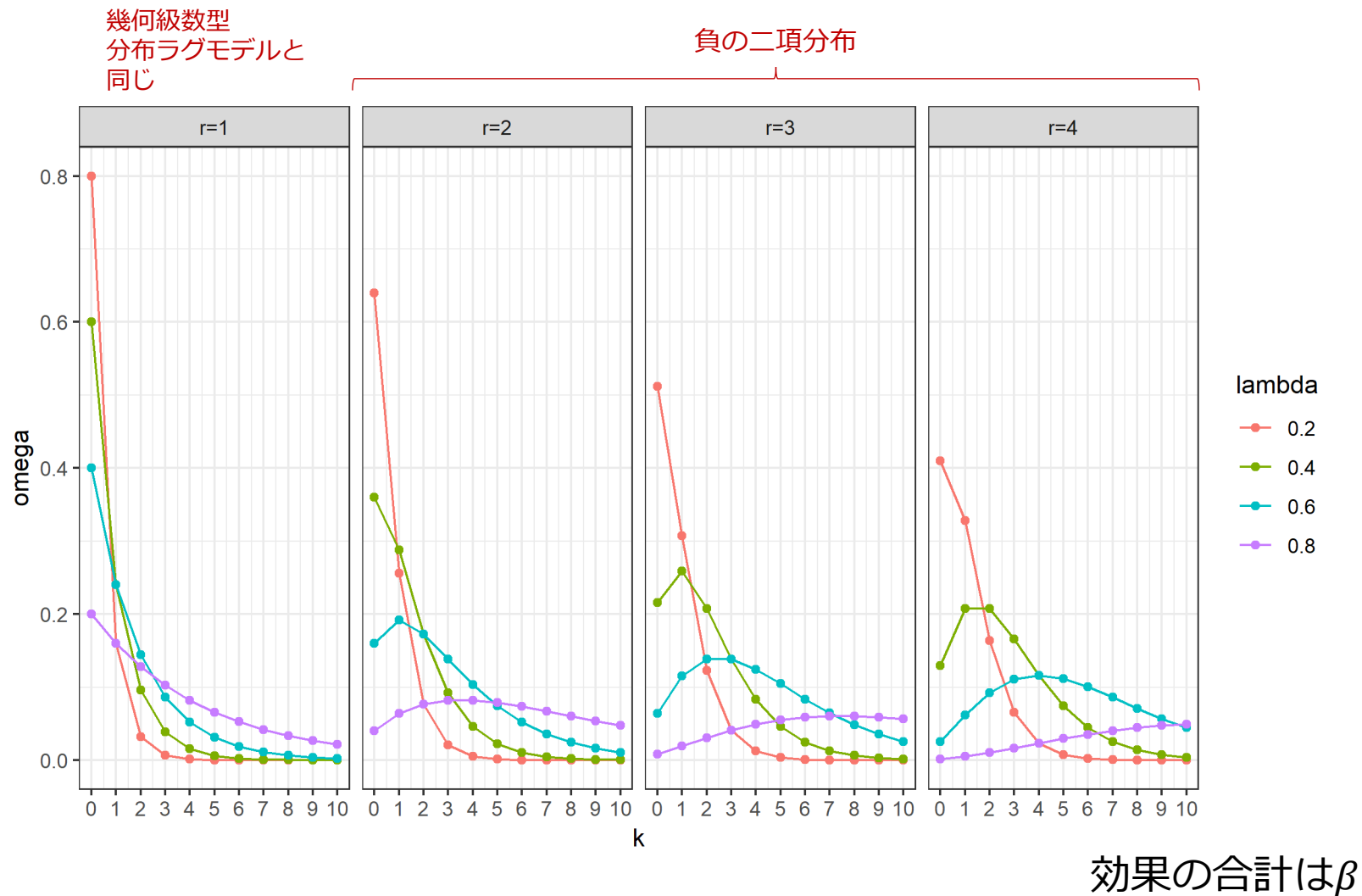
D. 負の二項分布ラグモデル negative binomial distributed lag model

$$Y_t = \alpha + \beta \sum_{k=0}^K \omega_k X_{t-k} + \varepsilon_t$$

$$\omega_k = \frac{(r+k-1)!}{(r-1)! k!} (1-\lambda)^r \lambda^k \quad (0 < \lambda < 1) \quad \text{負の二項分布}$$

- r は適当な整数にする (例, $r=2$)
- ラグとともに効果がいったん増大しその後減少する現象を表現できる

- X_t が1単位増えたときの、 $Y_t, Y_{t+1}, \dots$ への効果は？



E. 多項分布ラグモデル polynomial distributed lag model

$$Y_t = \alpha + \sum_{k=0}^K \beta_k X_{t-k} + \varepsilon_t$$

$$\beta_k = b_0 + b_1 k + b_2 k^2 + \dots b_p k^p \quad \text{P次多項式}$$

- PはKよりはるかに小さい整数とする。通常3~4程度まででよい
- アルモン・ラグモデルともいう
- X_t が1単位増えたときの、 $Y_t, Y_{t+1}, \dots, Y_{t+K}$ への効果は？
 - $\beta_0, \beta_1, \dots, \beta_K$
 - 効果の合計は $\sum_{k=0}^K \beta_k$

Shirley Montag Almon (1935-1975)
経済学者。40歳で早世したそうです
なんか美人ですね



F. 部分調整モデル partial adjustment model

$$Y_t = (1 - \lambda)(\alpha + \beta X_t) + \lambda Y_{t-1} + \varepsilon_t \quad (0 < \lambda < 1)$$

- 背後にある発想
 - 本来の売上水準は
$$Y'_t = \beta_0 + \beta_1 X_t$$
 - しかし、実際の売上水準はすぐには Y'_t にならず、
$$Y_t - Y_{t-1} = (1 - \lambda)(Y'_t - Y_{t-1}) + \varepsilon_t$$
だけしか変化できない
- $1 - \lambda$ を調整率 adjustment rate と呼ぶ
- どのマーケティング活動も同じ繰越率を持つと仮定している

- ラグ演算子を使って書き直すと

$$Y_t = \frac{\alpha}{1-\lambda} + \beta \frac{1}{1-\lambda L} X_t + \frac{1}{1-\lambda L} \varepsilon_t$$

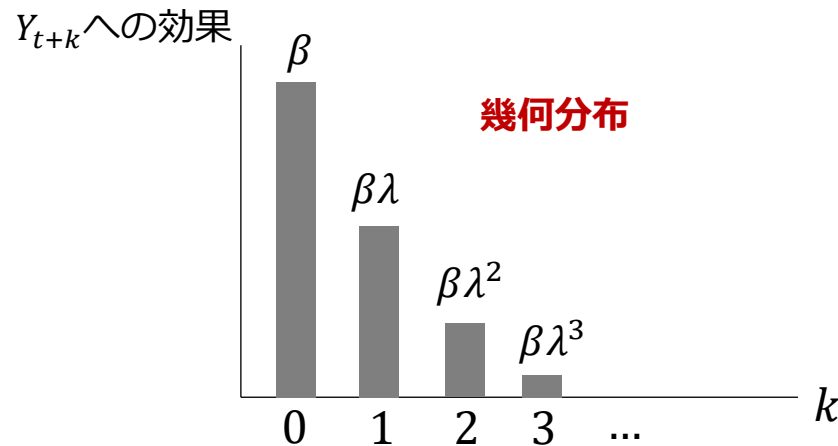
ARの形式になっている

いわゆる
ARXモデル

- 参考: AR(1)過程

$$Y_t = \mu + \frac{1}{1-\phi_1 L} \varepsilon_t$$

- X_t が1単位増えたときの、 $Y_t, Y_{t+1}, \dots, Y_{t+K}$ への効果は？
 - 幾何級数型分布ラグモデルと同じ



効果の合計は

$$\beta \sum_{k=0}^{\infty} \lambda^k = \frac{\beta}{1-\lambda}$$

G. 自己回帰分布ラグモデル(ADLモデル) autoregressive distributed lag model

$$Y_t = \alpha + \sum_{i=1}^p \gamma_i Y_{t-1} + \sum_{k=0}^q \beta_k X_{t-k} + \varepsilon_t$$

- $\gamma_1, \dots, \gamma_p$ は繰越効果のうち、マーケティング活動の種類に依存しない部分を表すと解釈できる
- $p=1, q=0$ とすると部分調整モデルになる
- 幾何級数型分布ラグモデルとの違いに注意！

- 幾何級数型分布ラグモデル

$$Y_t = \alpha + \beta X_t + \lambda Y_{t-1} + (\varepsilon_t - \lambda \varepsilon_{t-1})$$

撓乱項の効果は
繰り越されない

- 自己回帰分布ラグモデル($p=1, q=0$)

$$Y_t = \alpha + \beta X_t + \gamma Y_{t-1} + \varepsilon_t$$

撓乱項の効果も
繰り越される

- ラグ演算子を使って書き直すと

$$Y_t = \frac{\alpha}{1 - \gamma_1 L - \dots - \gamma_p L^p} + \beta_0 \frac{1 + \frac{\beta_1}{\beta_0} L + \dots + \frac{\beta_q}{\beta_0} L^q}{1 - \gamma_1 L - \dots - \gamma_p L^p} X_t + \frac{1}{1 - \gamma_1 L - \dots - \gamma_p L^p} \varepsilon_t$$

ARMAの形式になっている

- 参考: ARMA(p,q)過程

いわゆる
ARMAXモデル

$$Y_t = \mu + \frac{1 + \theta_1 L + \dots + \theta_q L^q}{1 - \phi_1 L - \dots - \phi_p L^p} \varepsilon_t$$

- 以上の工夫は組み合わせることができる

例) A,B,C,Fの組み合わせ (Leeflang, Mijatovic, Saunders, 1992)

	モデルの通称	モデル (切片は省略; ε_t はホワイトノイズ)	
B	直接ラグモデル	$Y_t = \sum_{k=0}^K \beta_k X_{t-k} + \varepsilon_t$	$Y_t = (\beta_0 + \beta_1 L + \dots + \beta_K L^K) X_t + \varepsilon_t$
A + B	自己回帰直接ラグモデル	$Y_t = \sum_{k=0}^K \beta_k X_{t-k} + E_t$ $E_t = \rho E_{t-1} + \varepsilon_t$	$Y_t = (\beta_0 + \beta_1 L + \dots + \beta_K L^K) X_t + \frac{1}{1 - \rho L} \varepsilon_t$
F + B	部分調整直接ラグモデル	$Y_t = \sum_{k=0}^K \beta_k X_{t-k} + \lambda Y_{t-K-1} + \varepsilon_t$	$Y_t = \frac{(\beta_0 + \beta_1 L + \dots + \beta_K L^K)}{1 - \lambda L^K} X_t + \frac{1}{1 - \lambda L^K} \varepsilon_t$
A + B + F	自己回帰部分調整直接ラグモデル	$Y_t = \sum_{k=0}^K \beta_k X_{t-k} + \lambda Y_{t-K-1} + E_t$ $E_t = \rho E_{t-1} + \varepsilon_t$	$Y_t = \frac{(\beta_0 + \beta_1 L + \dots + \beta_K L^K)}{1 - \lambda L^K} X_t + \frac{1}{(1 - \lambda L^K)(1 - \rho L)} \varepsilon_t$
	現在効果モデル	$Y_t = \beta X_t + \varepsilon_t$	
A	自己回帰現在効果モデル	$Y_t = \beta(X_t - \rho X_{t-1}) + \rho Y_{t-1} + \varepsilon_t$	$Y_t = \beta X_t + \frac{1}{1 - \rho L} \varepsilon_t$
F	部分調整モデル	$Y_t = \beta X_t + \lambda Y_{t-1} + \varepsilon_t$	$Y_t = \frac{\beta}{1 - \lambda L} X_t + \frac{1}{1 - \lambda L} \varepsilon_t$
A + F	自己回帰部分調整モデル	$Y_t = \beta X_t + \lambda Y_{t-1} + E_t$ $E_t = \rho E_{t-1} + \varepsilon_t$	$Y_t = \frac{\beta}{1 - \lambda L} X_t + \frac{1}{(1 - \lambda L)(1 - \rho L)} \varepsilon_t$
C	幾何級数型モデル	$Y_t = \beta \sum_{k=0}^{\infty} \lambda^k X_{t-k} + \varepsilon_t$	$Y_t = \frac{\beta}{1 - \lambda L} X_t + \varepsilon_t$
A + C	自己回帰幾何級数型モデル	$Y_t = \beta \sum_{k=0}^{\infty} \lambda^k X_{t-k} + E_t$ $E_t = \rho E_{t-1} + \varepsilon_t$	$Y_t = \frac{\beta}{1 - \lambda L} X_t + \frac{1}{1 - \rho L} \varepsilon_t$

■ 伝達関数モデル

定常時系列 Y_t, X_t について、

$$Y_t = \alpha + \frac{\omega_0 + \omega_1 L + \cdots + \omega_s L^s}{1 - \delta_1 L - \cdots - \delta_r L^r} L^b X_t + \frac{1 + \theta_1 L + \cdots + \theta_q L^q}{1 - \phi_1 L - \cdots - \phi_p L^p} \varepsilon_t$$

- 時系列回帰の一般的な枠組み
- ここまでに紹介した工夫はすべて、伝達関数モデルの特殊ケースとして捉えることができる

3. 動的回帰モデル

3. 動的回帰モデル

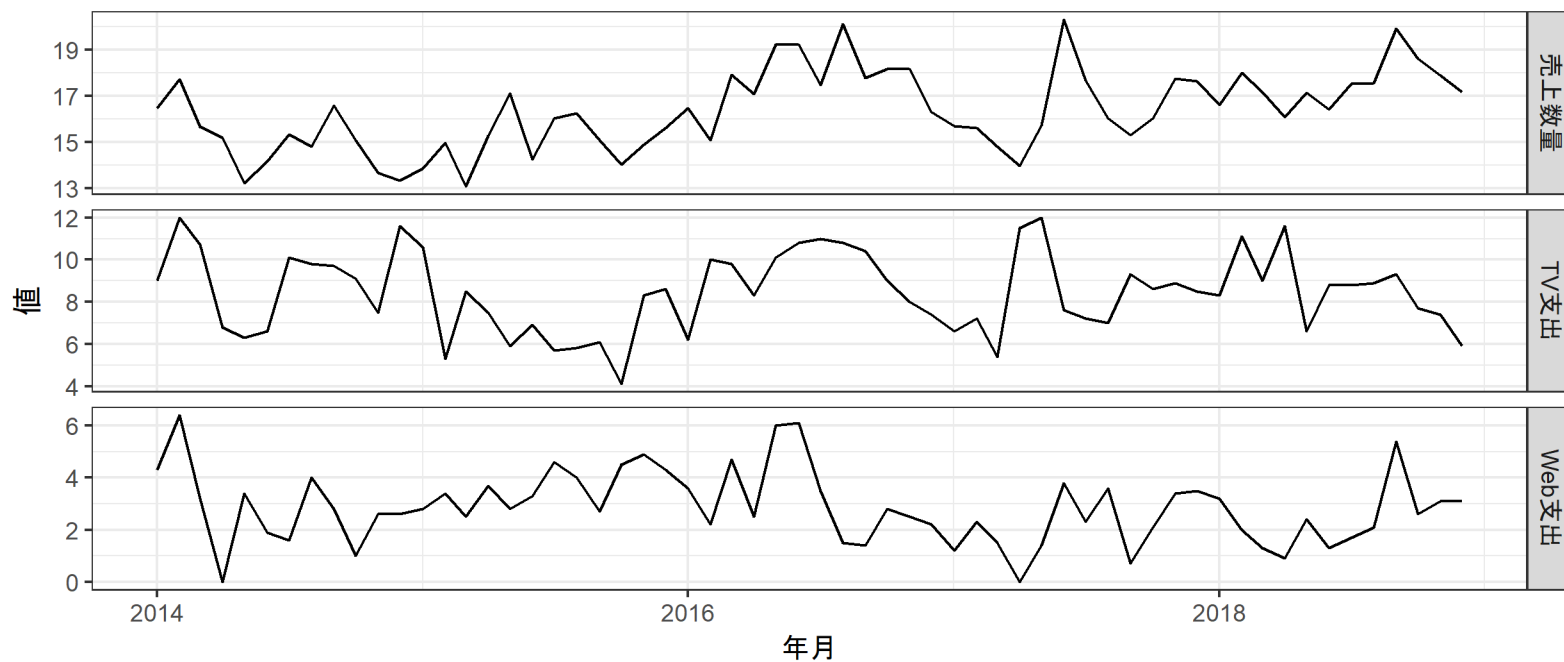
この節では、マーケティング活動の繰越効果を含む市場反応モデルを構築する手順について考えます

- こうしたモデルは、**動的回帰モデル** dynamic regression model と呼ばれることがあります
- ここでは、説明変数の効果が時間を通じて一定である場合について考えます
 - すなわち、(偏)回帰係数は定数

■ サンプルデータ： 売上と広告

Code 2

あるブランドの売上数量、TV広告支出、Web広告支出。
4年間の月次データ。



架空データ。ちょっと不自然ですが、ご容赦ください...

■ 動的回帰分析の手順

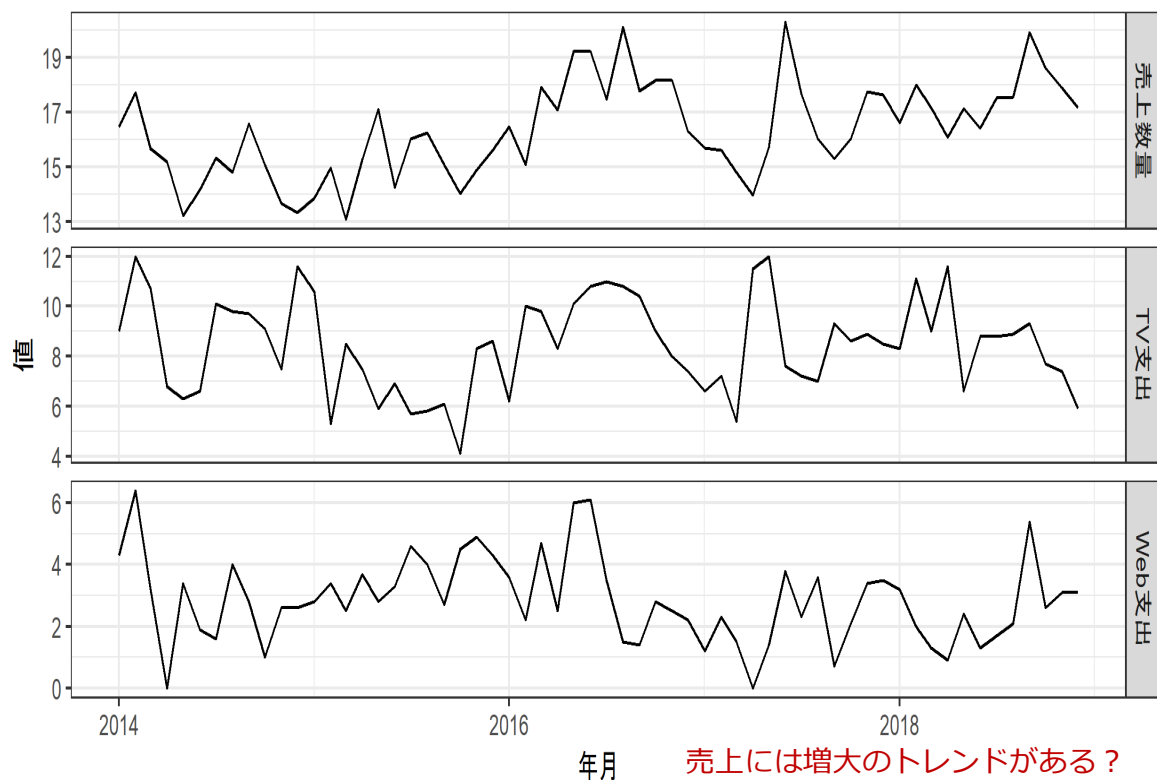
1. 時系列の観察
2. 時系列の変換
3. 定常性のチェック
4. 相関関係の観察
5. ラグ変数の次数を決める
6. モデル構築
7. 解釈

3-1. 時系列の観察

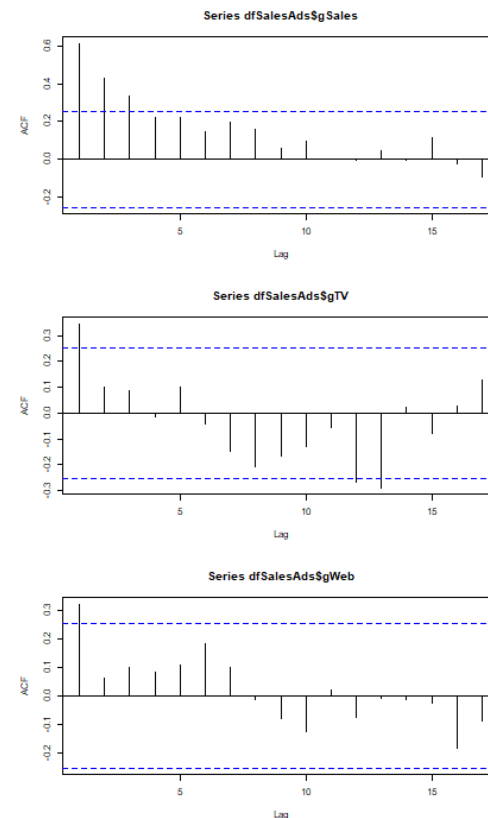
- トレンドはあるか？
- 季節変動は？なんらかの説明が可能な変動は？
- 分散は均一か？

Code 3

売上と広告



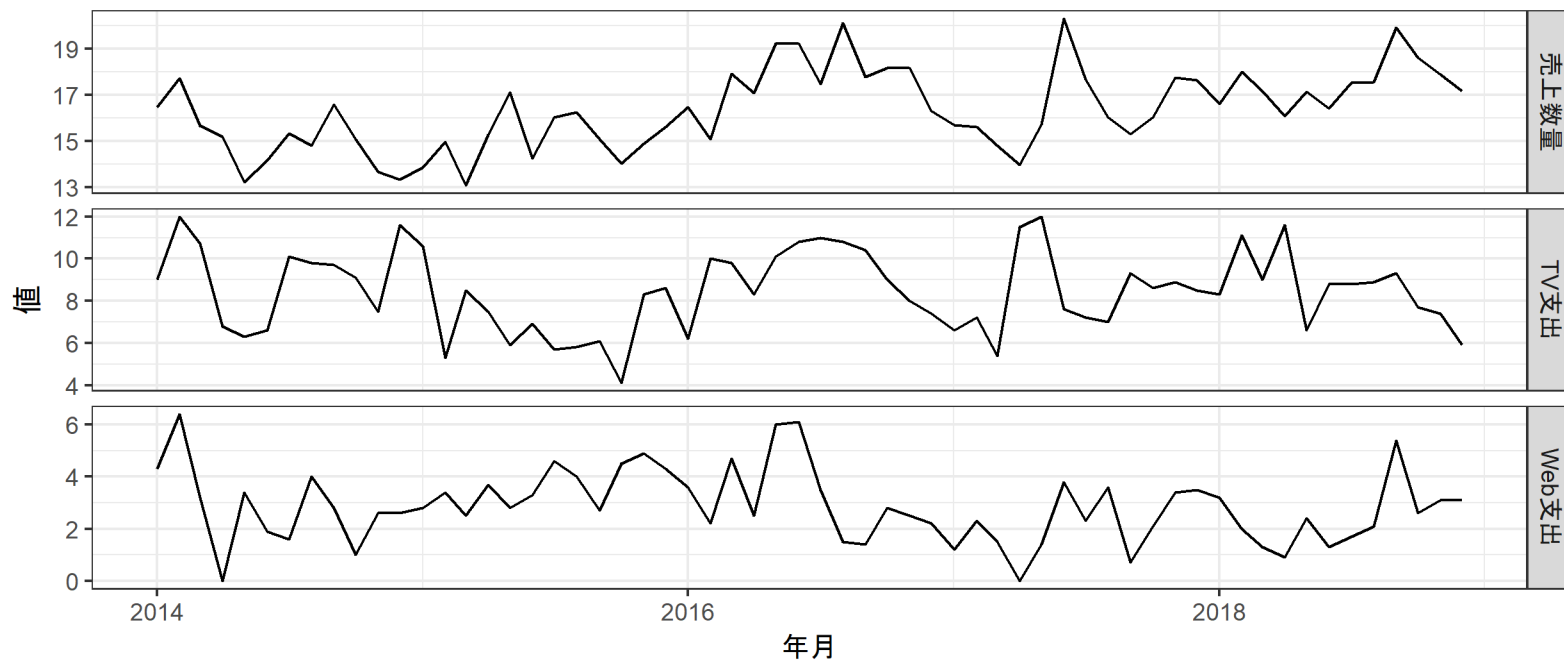
売上には増大のトレンドがある？
季節変動はなさそう。分散は均一な感じ



3-2. 時系列の変換

- 実質的な観点から
IV章を参照

- 定常な時系列に近づけるため
V章を参照



ま、原系列でいいか...

3-3. 定常性のチェック

- すべての時系列が単位根検定を通過することを確認する
 - いずれかの時系列が単位根を持っていたら、すべての時系列について差分をとり、再び単位根検定を行う
 - V章6-3.を参照

売上と広告

いずれの時系列も単位根検定を通過した

Code 4

```
> ndiffs(dfSalesAds$gSales, test = "adf", type = "trend")
[1] 0
> ndiffs(dfSalesAds$gweb, test = "adf", type = "trend")
[1] 0
> ndiffs(dfSalesAds$gTV, test = "adf", type = "trend")
[1] 0
```

■ 個々の説明変数について、目的変数との相関関係を観察する

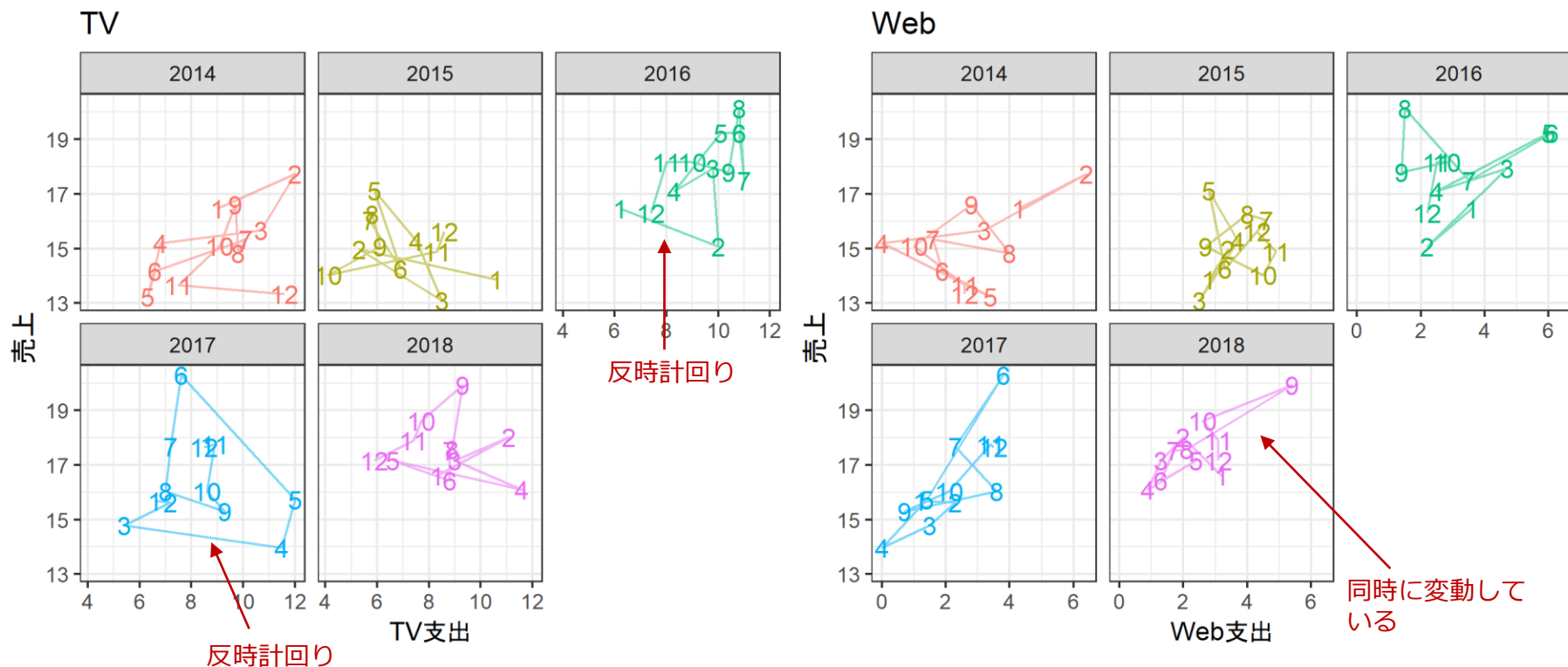
- 散布図
 - 標本相互相関関数
 - プレホワイト化標本相互相関関数
 - VARモデルで推定したグレンジャー因果性
 - VARモデルで推定したインパルス応答関数
- } 4節

散布図

- 横軸が縦軸の先行指標であれば、反時計回りの軌跡を描くはず
- 短い時系列を観察するときには便利 (Krider, Li, Liu, & Weinberg, 2005)

売上と広告

Code 5

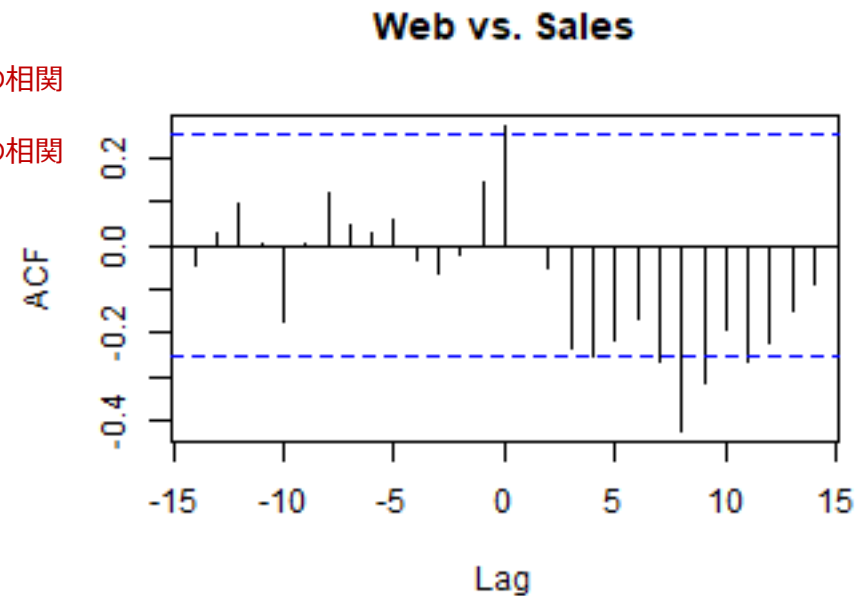
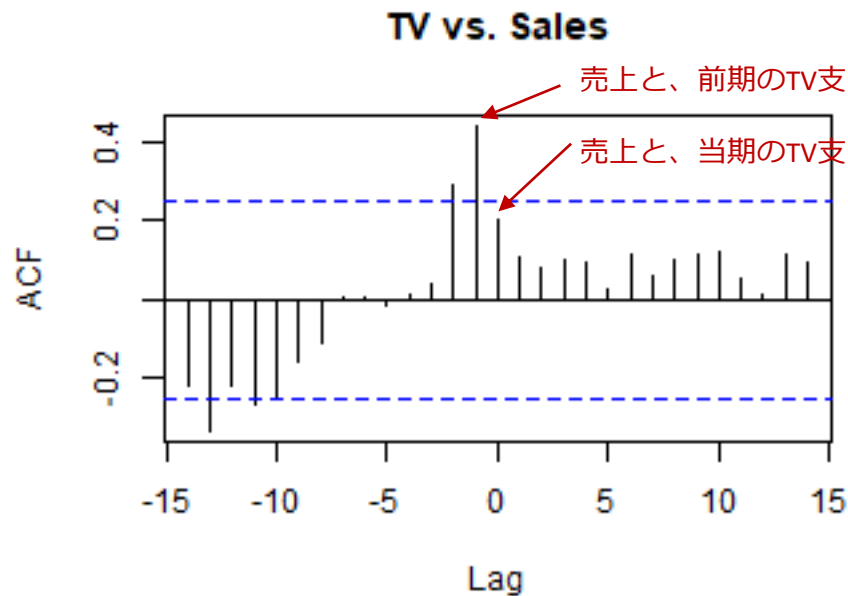


- **標本相互相関関数** (標本CCF) (HPS pp.267-268)
 - y_t と x_{t-k} の相関係数を、 k の関数と捉えたもの
 - わかりやすいが... 動的回帰モデル構築のための観察としては、あまり有用でない

売上と広告

Code 6

```
ccf(dfSalesAds$gTV, dfSalesAds$gSales, main = "TV vs. Sales")  
ccf(dfSalesAds$gweb, dfSalesAds$gSales, main = "Web vs. Sales")
```



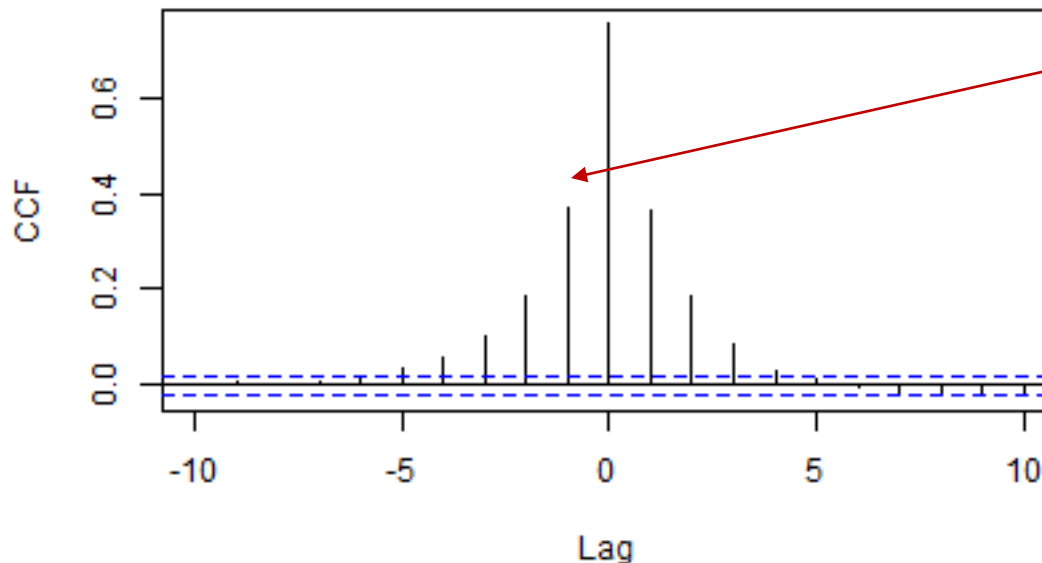
- なぜ相互相関関数 (CCF) は有用でないのか？
 - 自己相関関数 (ACF) を含んでいるから

Code 7

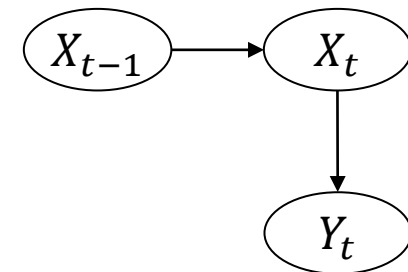
- 例) 真のモデルが

$$Y_t = X_t + \varepsilon_t, \quad X_t = 0.5X_{t-1} + \varepsilon'_t$$

であるときの、 Y_t と X_t の標本相互相関関数



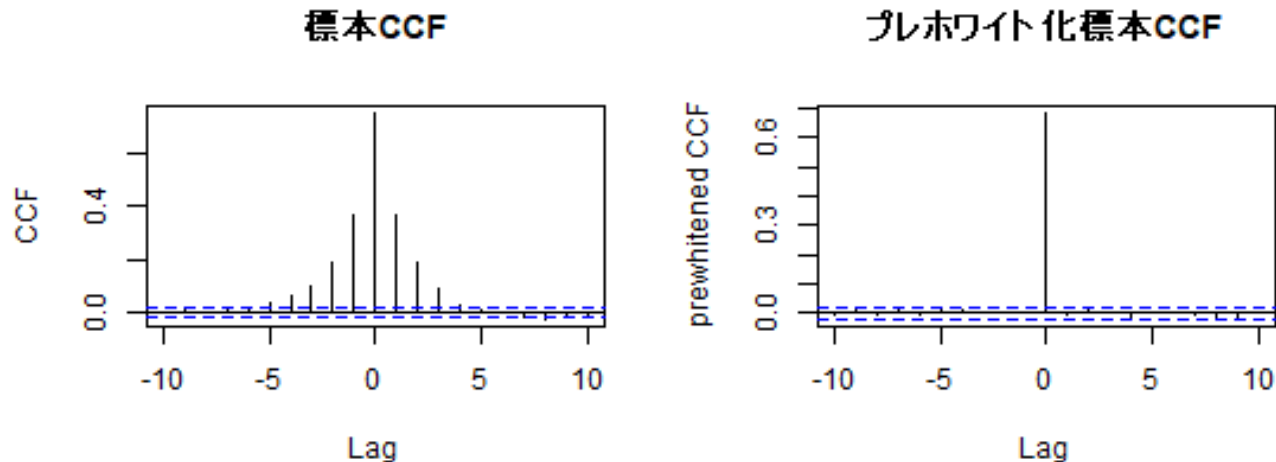
X_{t-1} は Y_t に影響していないが、
 X_{t-1} と X_t の間に自己相関があり
 X_t が Y_t に影響しているので
 X_{t-1} と Y_t の間に相互相関が生じる



- プレホワイト化標本相互相関関数 (HPS pp.269-270)

- 相互相関関数から自己相関関数を取り除いたもの
- 算出の手順
 1. 説明変数 X_t にARMAモデルをあてはめ、残差を求める
 - 残差はホワイトノイズとなるはず
 - この手順のことを**プレホワイト化**と呼ぶ
 2. 目的変数 Y_t に、**上と同じ**ARMAモデルをあてはめ、残差を求める
 - この手順はプレホワイト化ではない点に注意
 3. ふたつの残差時系列の相互相関関数を求める

Code 7



- プレホワイト化相互相関関数の意義

- 真のモデルが、**説明変数がひとつの**直接分布ラグモデル

$$Y_t = \alpha + \beta_0 X_t + \beta_1 X_{t-1} + \beta_2 X_{t-2} + \cdots + E_t$$

であるとき、プレホワイト化相互相関係数は $\beta_0, \beta_1, \beta_2, \dots$ と比例する

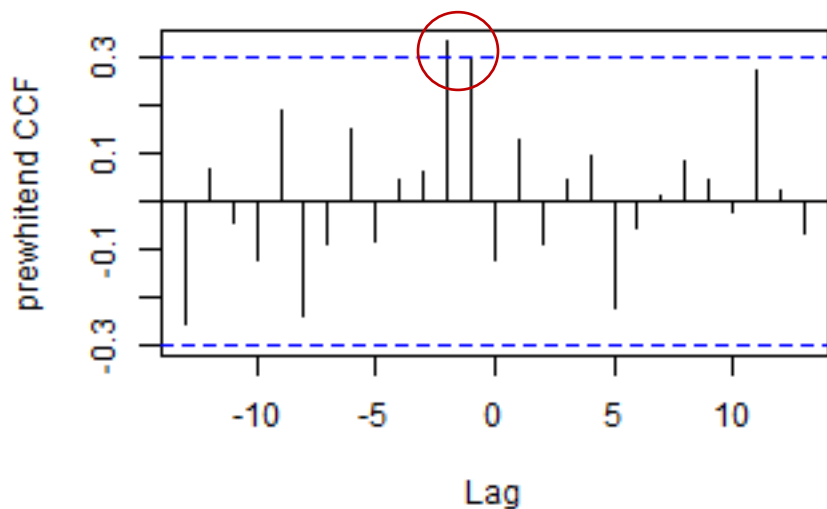
- モデルに含めるラグ変数の次数を決めるための手がかりとなる
 - 決め手にはならない。モデルはふつう複数の説明変数を含むため

売上と広告

Code 8

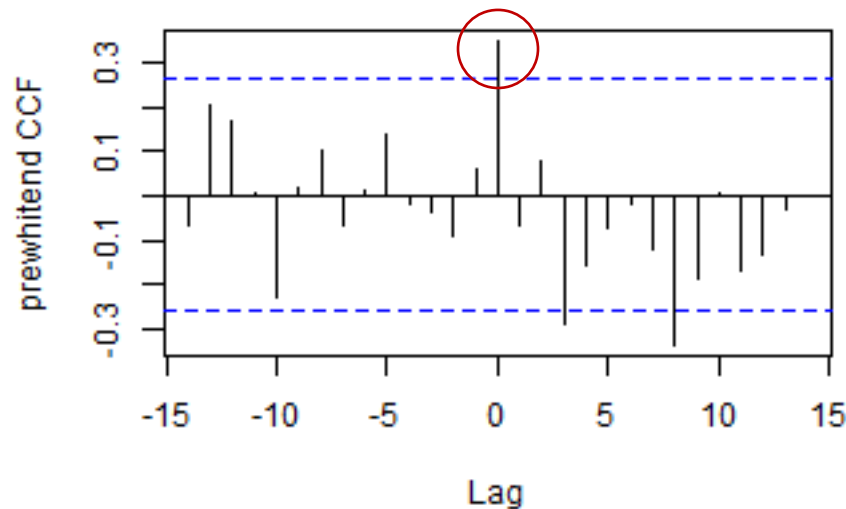
```
prewhiten(dfSalesAds$gTV, dfSalesAds$gSales, ylab = "prewhitend CCF", main = "TV vs. Sales")  
prewhiten(dfSalesAds$gweb, dfSalesAds$gSales, ylab = "prewhitend CCF", main = "Web vs. Sales")
```

TV vs. Sales



現在効果はない
繰越効果は2次くらい

Web vs. Sales



現在効果はある
繰越効果は小さい

3-5. ラグ変数の次数を決める

■ 説明変数のラグ効果を含めた回帰モデルを構築する

- 攪乱項の自己相関は無視する
- どのような手順で？
 - 「常にうまくいく」手順はない
 - クロス・セクション・データの回帰分析と同じく、試行錯誤が必要
- **当該領域についての実質的知識を活用する**
 - 繰越効果があるといわれているマーケティング活動は？
- **パラメータ数と時点数のバランスに気を配る**
 - 例) 説明変数が2つ、48時点のとき
 - 「サンプルサイズはパラメータ数の10倍必要」という言い伝えを信じるなら、パラメータ数は4～5まで
 - 現在効果モデルでも、攪乱項分散を含めるとパラメータ数は4
 - 説明変数ごとの繰越効果推定は断念し、繰越率はすべて共通と考えたほうがいいのかも
 - 部分調整モデルならばパラメータ数は5で済む

■ いくつかのお勧め手順

- HPS (pp.271-273)
 - 直接ラグモデルをOLS推定
 - 最初は十分にたくさんのラグ変数を含めておく
 - p値やAICの変化をみながら、徐々に削っていく
 - (→ 最終モデルは直接ラグモデルになりやすいだろう)
- Franses (1991)
 - まず Y_t についてARMAモデルを構築する
$$Y_t = \phi_1 Y_{t-1} + \dots + \phi_p Y_{t-p} + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \dots + \theta_q \varepsilon_{t-q}$$
 - 上のモデルの右辺に、すべての説明変数と、適切な次数のラグ変数を追加する
 - p値やAICの変化をみながら、追加・削除する
 - (→ 最終モデルは自己回帰分布ラグモデルになりやすいだろう)

売上と広告

まず、TVとWebについて4次ラグ変数までを含めてOLS回帰してみよう

Code 9

```
dfTemp <- dfSalesAds %>%
  mutate(
    nMonth = seq_along(dMonth),
    gTV_Lag1 = dplyr::lag(gTV, 1),
    gTV_Lag2 = dplyr::lag(gTV, 2),
    gTV_Lag3 = dplyr::lag(gTV, 3),
    gTV_Lag4 = dplyr::lag(gTV, 4),
    gweb_Lag1 = dplyr::lag(gweb, 1),
    gweb_Lag2 = dplyr::lag(gweb, 2),
    gweb_Lag3 = dplyr::lag(gweb, 3),
    gweb_Lag4 = dplyr::lag(gweb, 4)
  )
oModel1 <- lm(
  gSales ~ nMonth + gTV + gTV_Lag1 + gTV_Lag2 + gTV_Lag3 + gTV_Lag4
    + gweb + gweb_Lag1 + gweb_Lag2 + gweb_Lag3 + gweb_Lag4,
  data = dfTemp
)
print(summary(oModel1))
```

トレンド項をいれてみた

(中略)

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)	
(Intercept)	6.90296	1.42851	4.832	1.67e-05	***
nMonth	0.06689	0.00978	6.840	1.96e-08	***
gTV	0.04807	0.08728	0.551	0.584581	
gTV_Lag1	0.33001	0.09175	3.597	0.000811	***
gTV_Lag2	0.20652	0.09026	2.288	0.026999	*
gTV_Lag3	-0.07086	0.08925	-0.794	0.431473	
gTV_Lag4	0.05953	0.08633	0.690	0.494112	
gWeb	0.47568	0.13080	3.637	0.000721	***
gWeb_Lag1	0.21089	0.12524	1.684	0.099304	.
gWeb_Lag2	0.08791	0.12480	0.704	0.484910	
gWeb_Lag3	-0.05198	0.11593	-0.448	0.656088	
gWeb_Lag4	0.16409	0.11695	1.403	0.167634	

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(後略)

TVは2次のラグまで
含めたほうがよさそう

Webは1次のラグまでで
十分である模様

ラグ変数を削り、9個のモデルを推定してみた

Code 10

	パラメータ推定値								AIC
	切片	線形トレンド	TV	TV ラグ1	TV ラグ2	Web	Web ラグ1	Web ラグ2	
1	11.51	0.06	0.18			0.48			202.50
2	11.06	0.07	0.17			0.41	0.21		201.72
3	10.89	0.07	0.17			0.42	0.19	0.07	203.42
4	9.55	0.06	0.05	0.37		0.46			186.13
5	8.98	0.06	0.04	0.38		0.39	0.24		182.98
6	8.89	0.07	0.04	0.38		0.39	0.23	0.04	184.83
7	8.15	0.06	0.06	0.30	0.21	0.53			181.37
8	7.72	0.06	0.05	0.32	0.19	0.46	0.22		178.59
9	7.57	0.06	0.05	0.31	0.20	0.46	0.20	0.06	180.23

採用

TVの現在効果は0に近いが、
モデルから除外するのは常識に反すると思うので、
そのままにしておこう...

VIFは小さい

Code 10

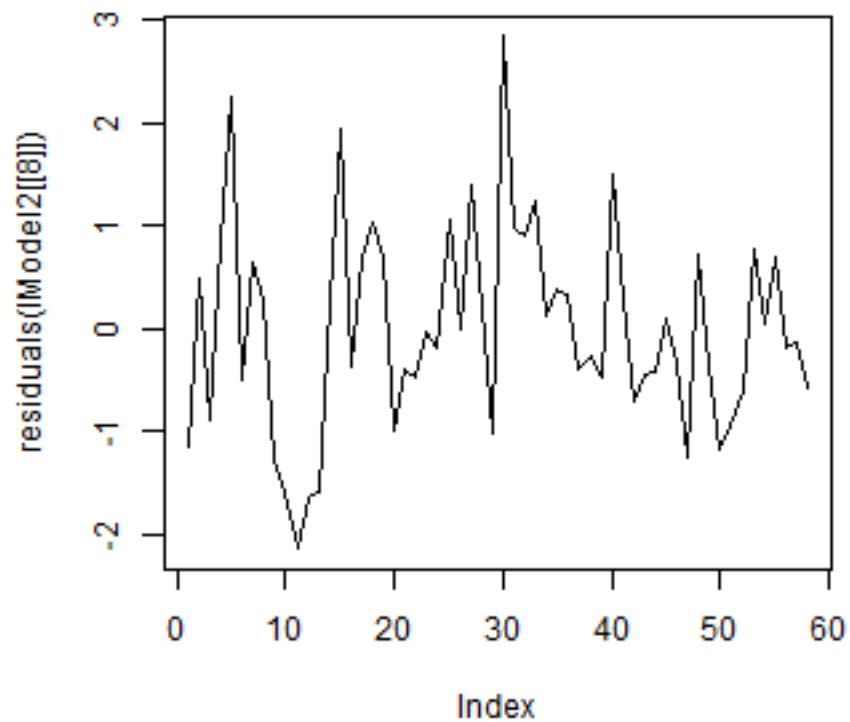
```
> car::vif(lmModel2[[8]])
      nMonth      gTV  gTV_Lag1  gTV_Lag2      gweb  gweb_Lag1
1.055391 1.181920 1.302087 1.202703 1.199167 1.147182
```

残差には自己相関がある模様

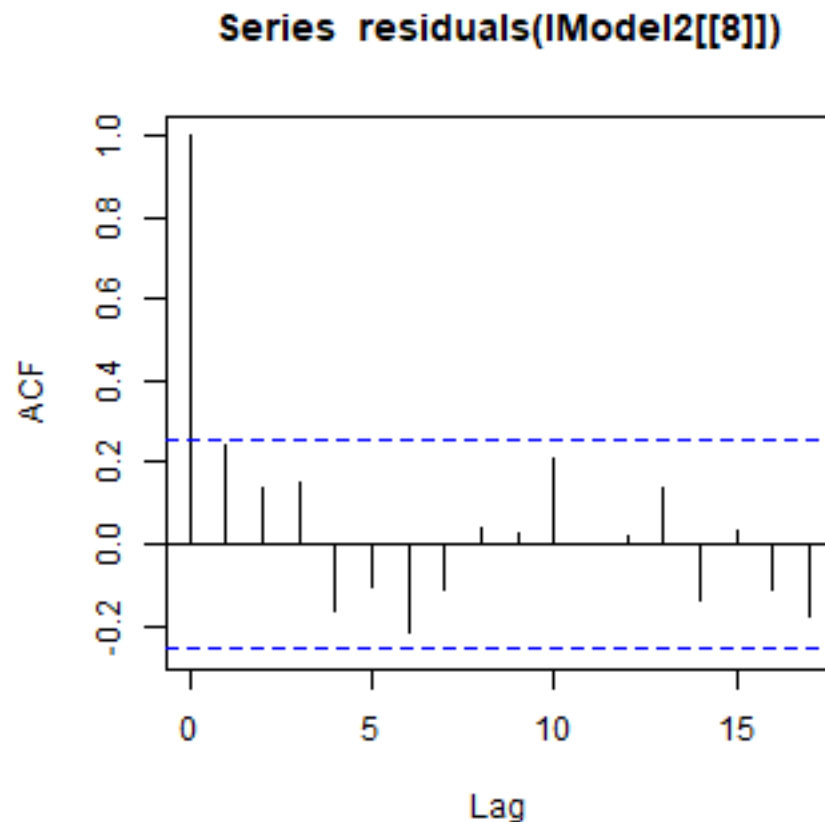
```
> lmtest::bgtest(lmModel2[[8]])

      Breusch-Godfrey test for serial correlation of order up to 1

data:  lmModel2[[8]]
LM test = 3.483, df = 1, p-value = 0.062
```



残差プロット



残差コレログラム

■ 最終的なモデルを構築する

- 2節で紹介した工夫を活用し、パラメータ数を減らす
- 攪乱項に自己相関を組み込む

■ 推定に用いるRパッケージ

- **付録にまとめました**
- 複雑なモデルを構築したい場合は、既存のパッケージを探すより、状態空間モデルとして表現したほうが早い (VII章)

- 直接ラグモデル+ARMA攪乱項

forecast::auto.arima()でARMA攪乱項の次数を自動選択。MA(3)が選択された

```
oModel3 <- auto.arima(
  dfTemp$gSales,
  d = 0,
  xreg = dfTemp %>%
    dplyr::select(nMonth, gTV, gTV_Lag1, gTV_Lag2, gweb, gweb_Lag1) %>%
    as.matrix(.),
  trace = TRUE, approximation = FALSE, stepwise = FALSE
)
print(summary(oModel3))
```

Series: dfTemp\$gSales
Regression with ARIMA(0,0,3) errors

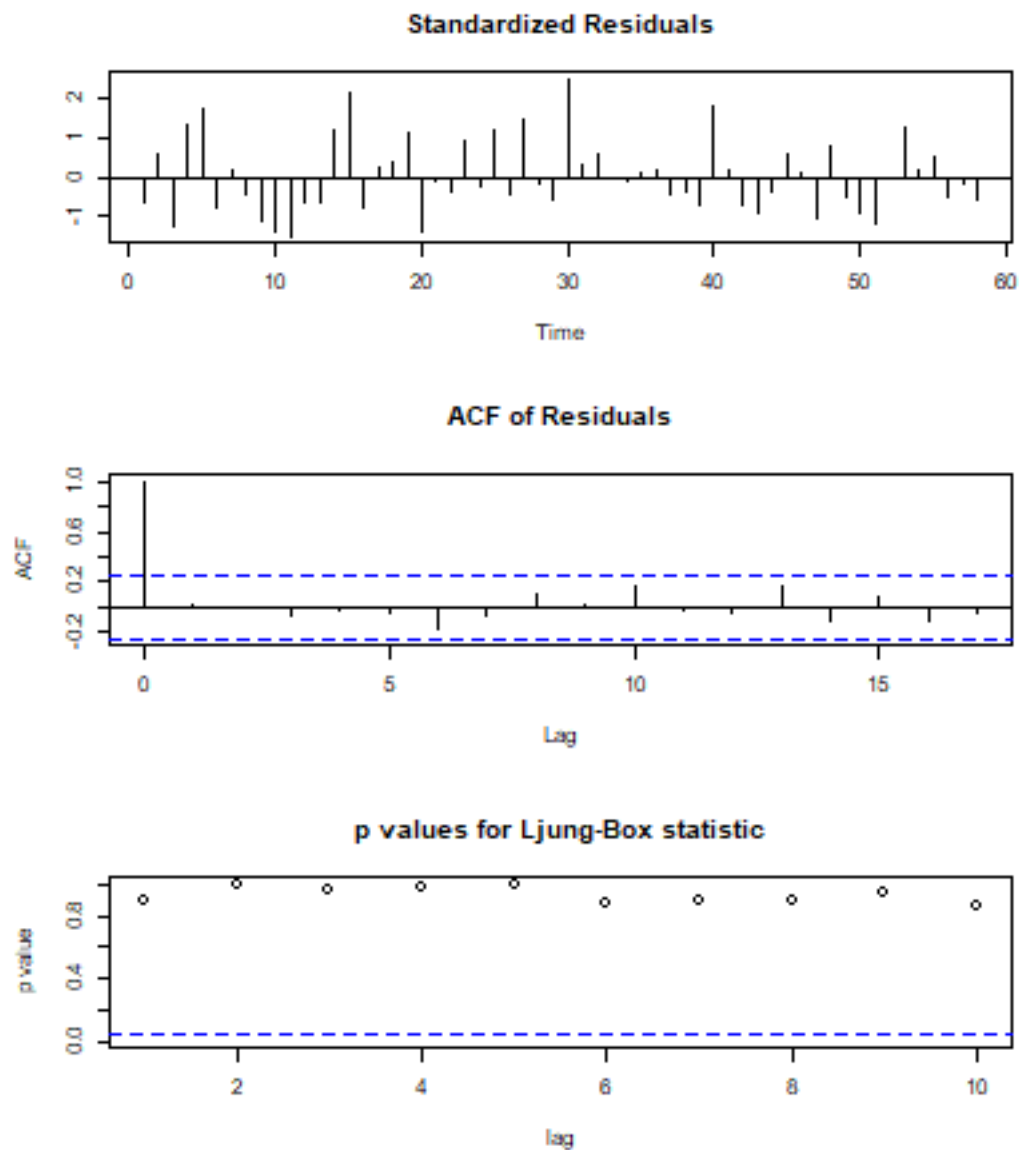
Coefficients:

	ma1	ma2	ma3	intercept	nMonth	gTV	gTV_Lag1	gTV_Lag2	gweb
	0.2529	0.1344	0.4397	7.9935	0.0620	0.0500	0.3119	0.2099	0.4481
s.e.	0.1259	0.1245	0.1680	1.1199	0.0124	0.0702	0.0702	0.0682	0.1005
	gweb_Lag1								
	0.0920								
s.e.	0.0983								

sigma^2 estimated as 0.9351: log likelihood=-76.4
AIC=174.8 AICC=180.54 BIC=197.46

モデル診断

```
tsdiag(oModel13)
```



- MA(3)撓乱項を指定してFGLS推定

Code 11

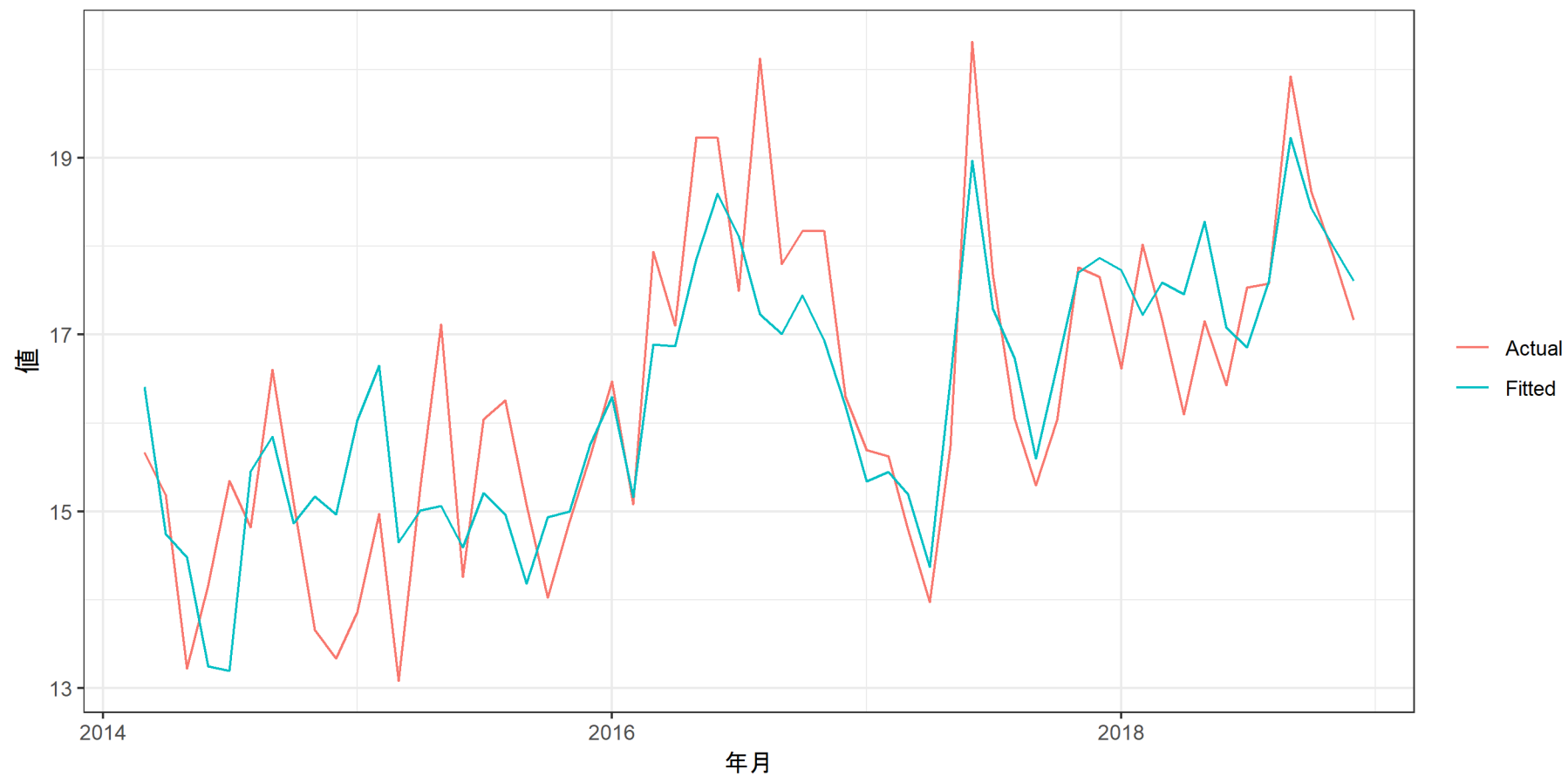
```
oModel4 <- gls(
  gSales ~ .,
  data = dfTemp %>% dplyr::select(-dMonth),
  corr = corARMA(q=3)
)
```

(中略)

Coefficients:

	Value	Std.Error	t-value	p-value
(Intercept)	7.946032	1.1809692	6.728399	0.0000
nMonth	0.062135	0.0133116	4.667781	0.0000
gTV	0.050873	0.0720590	0.705984	0.4834
gTV_Lag1	0.315700	0.0700079	4.509495	0.0000
gTV_Lag2	0.209146	0.0703625	2.972411	0.0045
gweb	0.445837	0.0973315	4.580608	0.0000
gweb_Lag1	0.098835	0.0917617	1.077078	0.2865

(中略)



- 自己回帰分布ラグモデル (次項の説明のために推定)

$$Y_t = \alpha + \gamma Y_{t-1} + \beta_{TV,0} X_{TV,t} + \beta_{TV,1} X_{TV,t-1} + \beta_{Web} X_{Web,t} + \varepsilon_t$$

$$Y_t = \frac{\alpha}{1 - \gamma L} + \frac{\beta_{TV,0} + \beta_{TV,1} L}{1 - \gamma L} X_{TV,t} + \frac{\beta_{Web}}{1 - \gamma L} X_{Web,t} + \frac{1}{1 - \gamma L} \varepsilon_t$$

Code 12

```
tsSales <- ts(dfSalesAds$gSales)
tsTV     <- ts(dfSalesAds$gTV)
tsweb    <- ts(dfSalesAds$gweb)
oModel5 <- dynlm(tsSales ~ L(tsTV, 0:1) + tsweb + L(tsSales))
print(summary(oModel5))
```

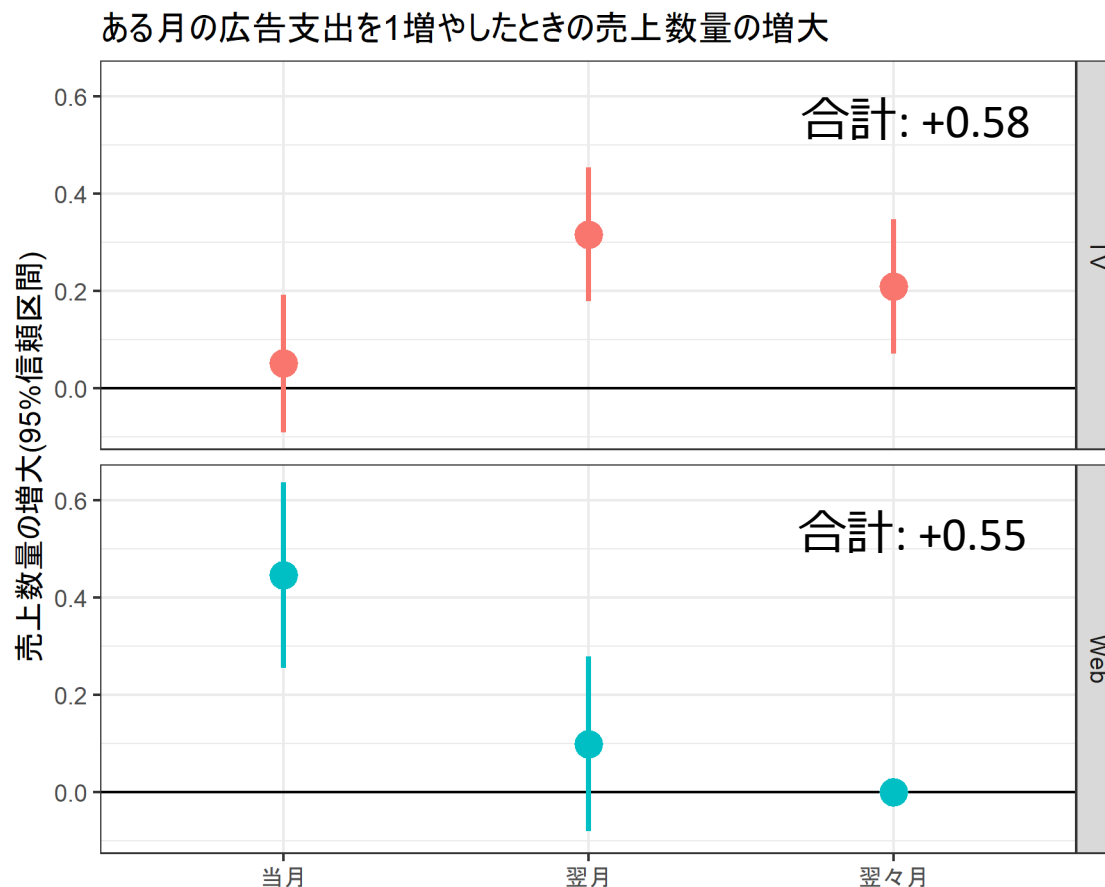
(中略)

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)	
(Intercept)	3.60775	1.63575	2.206	0.03169	*
L(tsTV, 0:1)0	0.04824	0.08799	0.548	0.58580	$\beta_{TV,0}$
L(tsTV, 0:1)1	0.29478	0.09085	3.245	0.00202	** $\beta_{TV,1}$
tsweb	0.35911	0.11328	3.170	0.00251	** β_{Web}
L(tsSales)	0.53968	0.09255	5.831	3.18e-07	*** γ

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

- 直接ラグモデルの場合
 - 係数は「説明変数を1増やしたときの目的変数の増大」を表す



- 自己回帰分布ラグモデルの場合

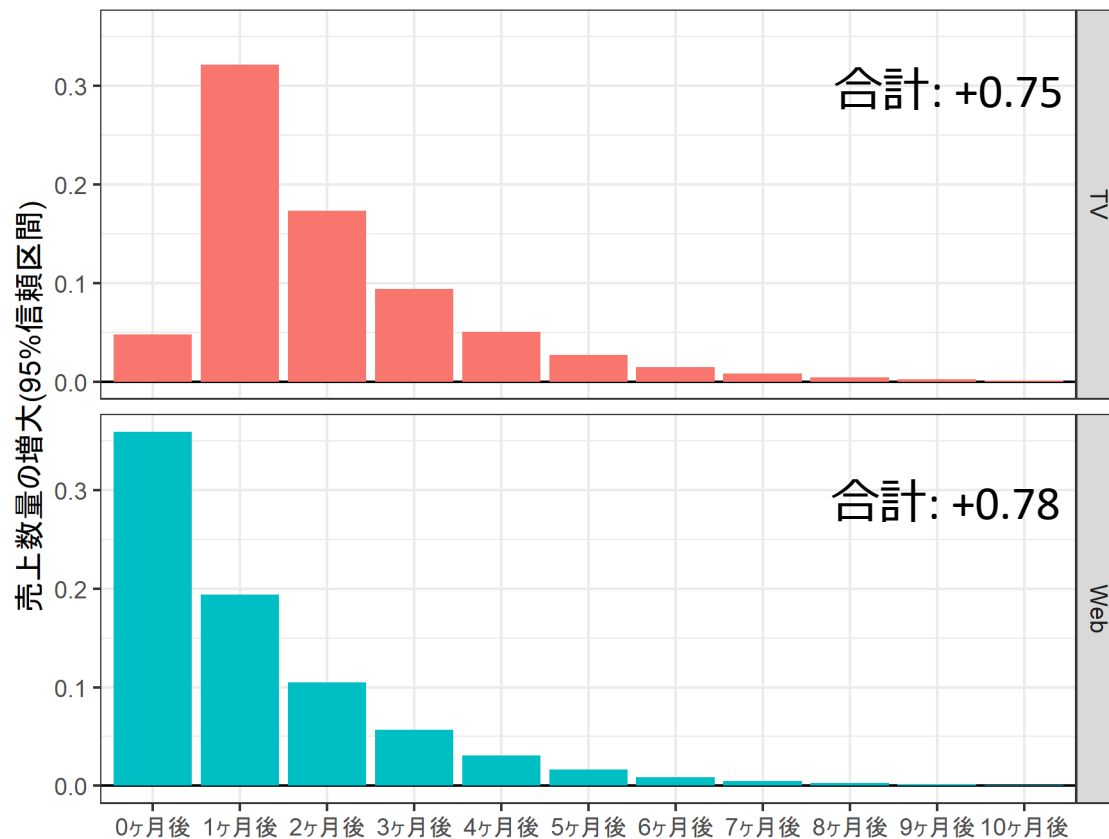
- TVの効果: $0.048 \frac{1 + \frac{0.295}{0.048}L}{1 - 0.540L} X_{TV,t}$
- Webの効果: $0.359 \frac{1}{1 - 0.540L} X_{Web,t}$

- 上記のARMA(p,q)形式の表現を、MA(∞)形式の表現 $(v_0 + v_1L + \dots)X_t$ に変形すればよい。V章4-2.を参照
- 面倒なので、ソフトウェアで数値的に計算したほうが早いかも

Code 12

```
out <- tibble(
  nLag = 0:1000, # 十分に大きな数
  gTV = 0.048 * c(1, ARMAtoMA(ar = 0.540, ma = c(0.295/0.048), lag.max = 1000)),
  gweb = 0.359 * c(1, ARMAtoMA(ar = 0.540, lag.max = 1000))
)
sum(out$gTV)
sum(out$gweb)
```

ある月の広告支出を1増やしたときの売上数量の増大



4. ストック変数

4. ストック変数 (HPS pp. 52-53, 130-131)

■ ストック変数

- 幾何級数型分布ラグモデルを仮定し、繰越率 λ として定数を仮定する
- 回帰モデルの説明変数を、マーケティング・ミックス変数 X_t のかわりに、下式で定義する $XStock_t$ とする

$$XStock_t = X_t + \lambda XStock_{t-1}$$

- ラグ演算子を使って書くと

$$XStock_t = \frac{1}{1 - \lambda L} X_t$$

- 利点：モデルが簡単になる

■ 繰越率をどうやって決めるか？

- 過去経験に基づいて決める
- 主観に基づいて決める
- 動的回帰モデルでの推定を参考にして決める

売上と広告

Code 13

例) 繰越率0.5のストック変数を用いた回帰モデル

- ・ 撓乱項の自己相関を無視してOLS推定してみた

```
dfTemp <- dfSalesAds %>%  
  mutate(  
    gTVStock = stats::filter(gTV, filter = 0.5, method = "recursive"),  
    gWebStock = stats::filter(gWeb, filter = 0.5, method = "recursive")  
  )  
summary(lm(gSales ~ gTVStock + gWebStock, data=dfTemp))
```

```
Call:  
lm(formula = gSales ~ gTVStock + gWebStock, data = dfTemp)  
  
Residuals:  
    Min       1Q   Median       3Q      Max   
-3.6398 -1.1932  0.1214  1.2154  3.7934   
  
Coefficients:  
            Estimate Std. Error t value Pr(>|t|)      
(Intercept) 10.62407    1.50452   7.061 2.51e-09 ***  
gTVStock      0.24902    0.07604   3.275  0.0018 **   
gWebStock     0.28244    0.11132   2.537  0.0139 *    
---  
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1  
(後略)
```

簡単でいいですね...

5. ベクトル自己回帰モデル

5. ベクトル自己回帰モデル (沖本 p.76-94)

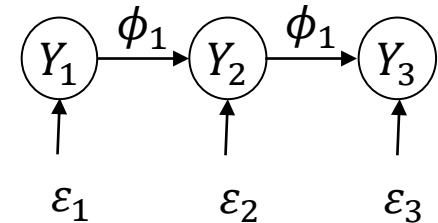
■ ベクトル自己回帰過程 (VAR過程)

- AR過程のベクトル版
- おさらい：自己回帰過程(AR過程) (V章4-2)
確率変数 Y_t について、
$$Y_t = c + \phi_1 Y_{t-1} + \dots + \phi_p Y_{t-p} + \varepsilon_t$$

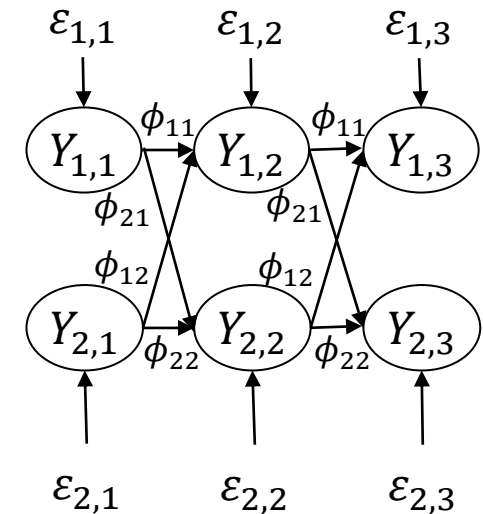
ただし ε_t はホワイトノイズ。
- ベクトル自己回帰過程(VAR過程)
確率変数ベクトル $\mathbf{Y}_t$ について、
$$\mathbf{Y}_t = \mathbf{c} + \boldsymbol{\Phi}_1 \mathbf{Y}_{t-1} + \dots + \boldsymbol{\Phi}_p \mathbf{Y}_{t-p} + \boldsymbol{\varepsilon}_t$$

ただし $\boldsymbol{\varepsilon}_t$ はベクトル・ホワイトノイズ。

AR(1)



二変量VAR(1)



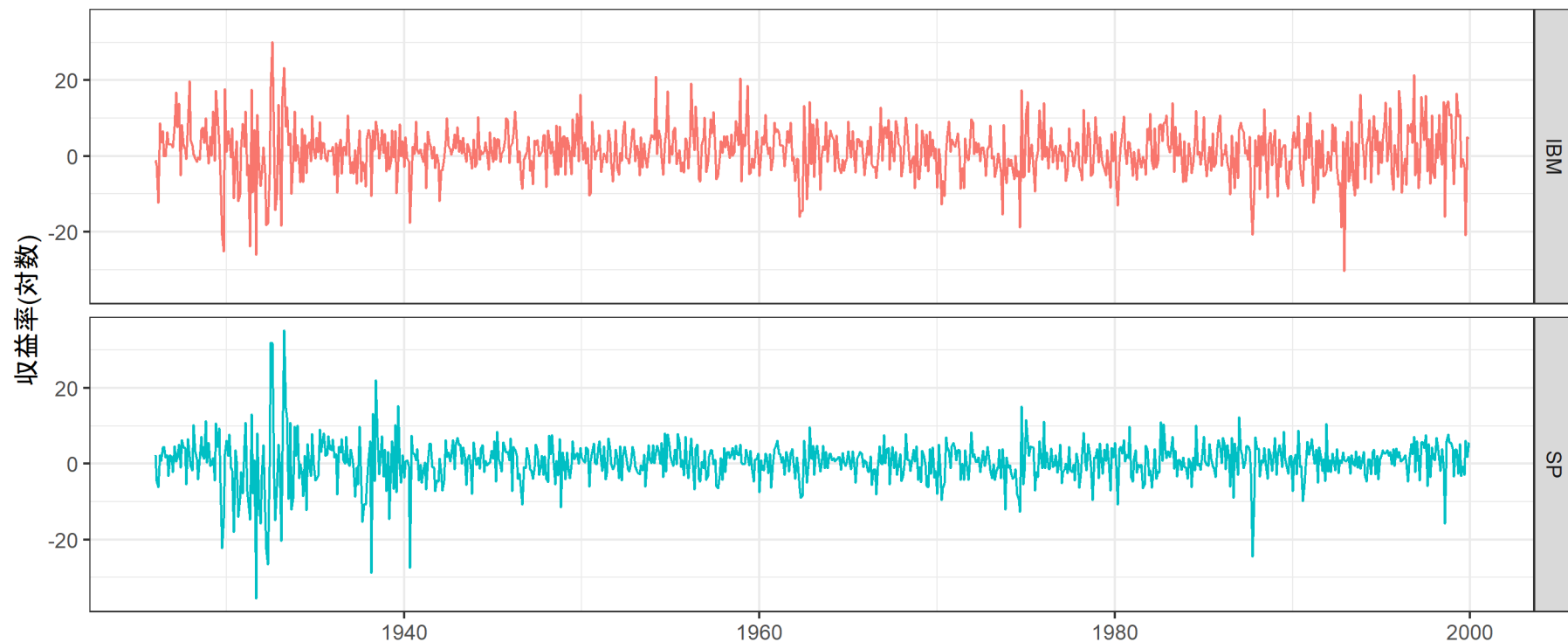
■ 市場反応分析におけるVARモデルの意義

- 市場反応モデルとして (HPS pp.278-282)
 - マーケティング・ミックス変数や環境変数の間の、長期的な相互関係をモデル化する際に有用
 - このセミナーでは割愛
- 動的回帰モデル構築の準備のため、時系列間の関係を観察するためのツールとして
 - **グレンジャー因果性**
 - **インパルス応答関数**

データ例

IBM-SP

- S&P500(USの株価指数)の対数収益率と、IBMの対数収益率



ここでは2変量VARモデルのみを扱う

1. 定常性のチェック
2. VAR次数の選択
3. VARモデルの推定

Step 1. 定常性のチェック

- VARモデルの性質
 - 単位根を持つ時系列についても推定できる
 - ただし、グレンジャー因果性検定(後述)は適用できない
 - 両方の時系列が単位根を持ち、さらにある条件(共和分関係)を満たしているとき、差分時系列にVARモデルをあてはめるのは誤り
- どちらか一方だけが単位根を持っていたら？
 - そのままVARモデルを適用
 - グレンジャー因果性検定は適用できない
- 両方が単位根を持っていたら？
 - 共和分関係の有無を調べる。このセミナーでは扱わない
 - cf. 沖本(pp.124-146), 福地・伊藤(pp.146-157), HPS(pp.183-185)

IBM-SP

Code 15

どちらの時系列も、単位根検定を通過した

```
> ndiffs(dfIBMSP$IBM, test = "adf", type = "trend")  
[1] 0  
  
> ndiffs(dfIBMSP$SP, test = "adf", type = "trend")  
[1] 0
```

Step 2. VAR次数の選択

Code 16

IBM-SP

```
# VAR次数選択
VARselect(
  dfIBMSP %>% dplyr::select(IBM, SP),
  lag.max = 5, ← 適当に大きめの数
  type = "const" ← 時系列を観察し、トレンドがあるときは"trend"とする
)
```

```
$selection
AIC(n)  HQ(n)  SC(n)  FPE(n) ← さまざまな基準に基づいて選択された、適切なVAR次数。
      3      1      1      3      1か3がよさそう

$criteria
      1      2      3      4      5
AIC(n) 6.760111 6.758874 6.753468 6.755548 6.753548
HQ(n)  6.772538 6.779587 6.782466 6.792831 6.799116
SC(n)  6.792614 6.813046 6.829308 6.853057 6.872725
FPE(n) 862.737822 861.671992 857.026761 858.811796 857.096638
```

Step 3. VARモデルの推定

Code 16

IBM-SP

```
# VARモデル推定
oVARModel <- VAR(
  dfIBMSP %>% dplyr::select(IBM, SP),
  p = 1,
  type = "const"
)
summary(oVARModel)
```

さきほど選択されたVAR次数。
ここでは1にします

VAR Estimation Results:

=====

(中略)

Estimation results for equation IBM:

=====

IBM = IBM.l1 + SP.l1 + const

	Estimate	Std. Error	t value	Pr(> t)
IBM.l1	0.01919	0.04334	0.443	0.6579
SP.l1	0.10616	0.05167	2.054	0.0402 *
const	1.16265	0.22897	5.078	4.66e-07 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(中略)

Estimation results for equation SP:

=====

SP = IBM.l1 + SP.l1 + const

	Estimate	Std. Error	t value	Pr(> t)
IBM.l1	-0.005419	0.036441	-0.149	0.88183
SP.l1	0.080189	0.043453	1.845	0.06531 .
const	0.499350	0.192541	2.593	0.00966 **

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(後略)

$$Y_{IBM,t} = 1.163 + 0.019Y_{IBM,t-1} + 0.106Y_{SP,t-1} + \varepsilon_{IBM,t}$$

$$Y_{SP,t} = 0.499 - 0.005Y_{IBM,t-1} + 0.080Y_{SP,t-1} + \varepsilon_{SP,t}$$

将来の Y の予測について、

- 現在と過去の Y の値だけに基づいた、将来の Y の予測
- 現在と過去の X, Y の値に基づいた、将来の Y の予測

を比べ、後者の平均平方誤差(MSE)のほうが小さかったとき、

「 X から Y への**グレンジャー因果性**が存在する」

と表現する

- 言いかえると、将来の Y の予測に過去の X が役立つこと
- 注意！ グレンジャー因果性は、いわゆる因果性ではない
 - 稲光は直後の雷鳴の予測に役立つが、雷鳴の原因ではない

- グレンジャー因果性検定
 - VARモデルに基づき、グレンジャー因果性を検定する
 - すべての時系列が定常であることが必要
- 市場反応モデリングにおける用途
 - 説明変数のスクリーニング
 - さまざまな環境変数のなかで、目的変数の予測に役立つものをピックアップする
 - 関連性がある2つの時系列について、どちらを目的変数とし、どちらを説明変数とするかを決めるための手がかりとして用いる
 - 例) 購入意向をクチコミ量で説明する？ それとも、クチコミ量を購入意向で説明する？

IBM-SP

Code 16

```
# SPからIBMへのグレンジャー原因は存在するか？  
causality(oVARModel, cause = "SP")
```

```
$Granger
```

```
Granger causality H0: SP do not Granger-cause IBM
```

```
data: VAR object oVARModel
```

```
F-Test = 4.2205, df1 = 1, df2 = 1768, p-value = 0.04009
```

帰無仮説「グレンジャー因果性はない」を検定。
有意水準5%で棄却されている。
SPからIBMへのグレンジャー因果性があると判断できる

```
$Instant
```

```
H0: No instantaneous causality between: SP and IBM
```

```
data: VAR object oVARModel
```

```
Chi-squared = 253.94, df = 1, p-value < 2.2e-16
```

```
# IBMからSPへのグレンジャー因果性は存在するか？  
causality(oVARModel, cause = "IBM")
```

```
$Granger
```

```
Granger causality H0: IBM do not Granger-cause SP
```

```
data: VAR object oVARModel
```

```
F-Test = 0.022111, df1 = 1, df2 = 1768, p-value = 0.8818
```

IBMからSPへのグレンジャー因果
性はみつからない

```
$Instant
```

```
H0: No instantaneous causality between: IBM and SP
```

```
data: VAR object oVARModel
```

```
Chi-squared = 253.94, df = 1, p-value < 2.2e-16
```

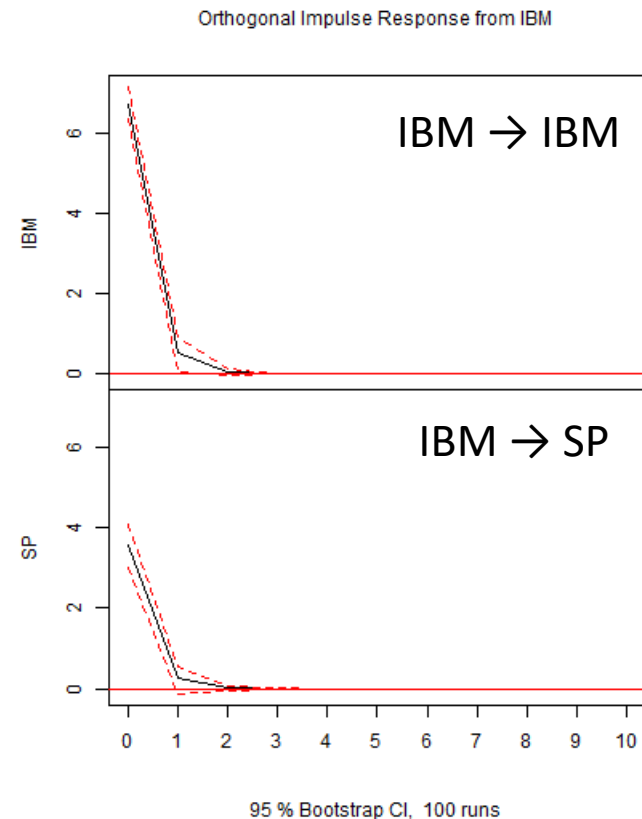
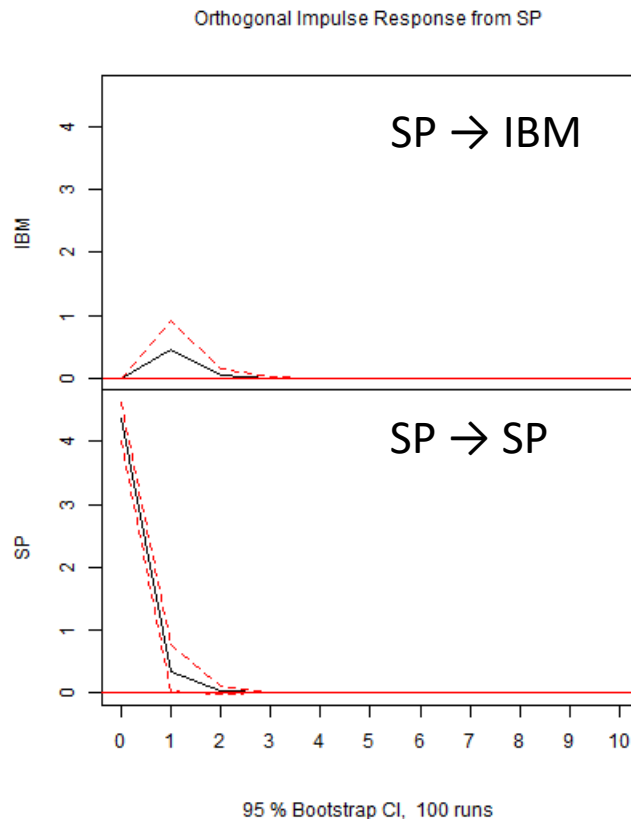
■ インパルス応答関数 (直交化インパルス応答関数)

- 「ある変数だけに衝撃(インパルス)が与えられたとき、他の変数はどのように反応するか」を、その衝撃からの時間の関数として捉えたもの
 - VARモデルにおいて、ある時点のある変数の攪乱項だけに1を与え、次の時点以降のすべての攪乱項を0としたときの挙動
- 市場反応モデリングにおける用途
 - 動的回帰モデルに含めるラグ効果の次数を決めるための手がかりとなる
- 注意点
 - 現在効果はモデルに含まれていない
 - 3変数以上のVARモデルの場合、直交化インパルス応答関数は変数の並べ方に依存する

IBM-SP

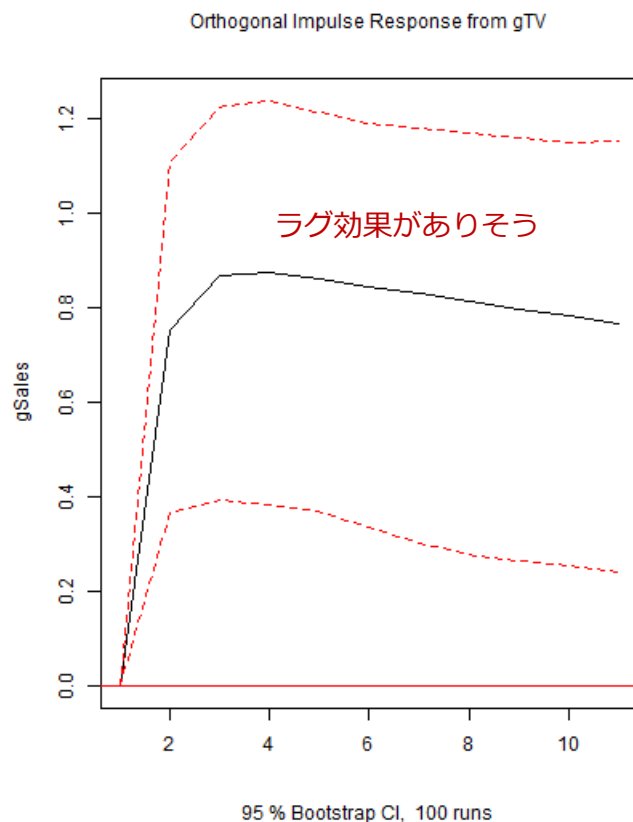
Code 16

```
plot(irf(oVARModel, impulse = "SP"))
plot(irf(oVARModel, impulse = "IBM"))
```

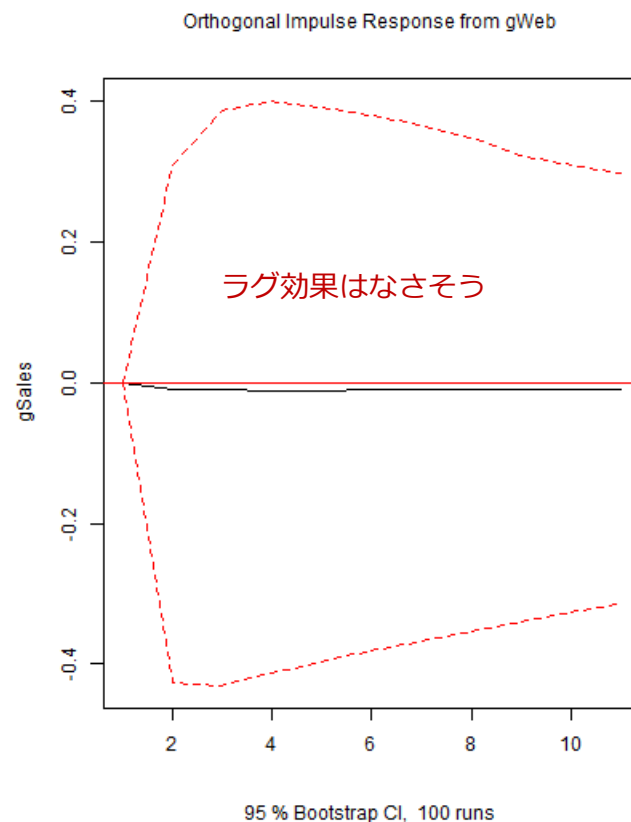


売上と広告

Code 17



売上とTVの2変量VAR(1)モデルで推定した、
TV→売上のインパルス応答関数



売上とWebの2変量VAR(1)モデルで推定した、
Web→売上のインパルス応答関数

6. 時間集計に伴うバイアス

■ 時間集計

- 通常、時系列データはなんらかの時間単位で集計されている
 - 年次、月次、週次...
- 集計の時間単位は、たいてい所与。分析者は時間単位を選べない

■ データ期間バイアス

- モデル構築の際に用いた集計データの時間単位に起因するバイアス
- 一般に、時間単位が長いほど (たとえば月次より年次のほうが)
 - マーケティング・ミックス変数の現在効果は大きめに推定される
 - マーケティング・ミックス変数の持続期間も長めに推定される

■ 異なる時間単位のあいだでのパラメータの関係 (Leone, 1995)

年 t , 月 m ($=1, \dots, 12$)の売上を Y_{tm} , 広告支出を X_{tm} とし、**月次モデル**を

$$Y_{tm} = \alpha + \beta X_{tm} + \gamma Y_{t,m-1} + \varepsilon_{tm}$$

としよう。

年次集計値を $Y_t = \sum_m Y_{tm}$, $X_t = \sum_m X_{tm}$ とし、**年次モデル**を

$$Y_t = \alpha' + \beta' X_t + \gamma' Y_{t-1} + E_t$$

としよう。

ふたつのモデルの間には、どのような関係があるだろうか？

月次モデルの両辺を $m = 1, \dots, 12$ について合計すると、 $\varepsilon_t = \sum_m \varepsilon_{tm}$ として

$$Y_t = 12\alpha + \beta X_t + \gamma(Y_t + Y_{\text{前年12月}} - Y_{\text{当年12月}}) + \varepsilon_t$$

仮に $Y_{t,12} = (1/12)Y_t$ とするならば、

$$Y_t = 12\alpha + \beta X_t + \gamma(Y_t + (1/12)Y_{t-1} - (1/12)Y_t) + \varepsilon_t$$

$$(1 - (11/12)\gamma)Y_t = 12\alpha + \beta X_t + (1/12)\gamma Y_{t-1} + \varepsilon_t$$

$$Y_t = \frac{12\alpha}{1 - (11/12)\gamma} + \frac{\beta}{1 - (11/12)\gamma} X_t + \frac{(1/12)\gamma}{1 - (11/12)\gamma} Y_{t-1} + \frac{1}{1 - (11/12)\gamma} \varepsilon_t$$

前頁のふたつのモデルは、どちらも

$$Y_t = \alpha + \beta X_t + \lambda Y_{t-1} + E_t$$

という形のモデルである。

X_t を1増やせば、 $Y_t, Y_{t+1}, Y_{t+2}, Y_{t+3}, \dots$ は $\beta, \beta\lambda, \beta\lambda^2, \beta\lambda^3, \dots$ だけ増える。

効果の持続期間の指標として、 $\lambda^c = 0.5$ となる c について考え、これを「半減期」と呼ぼう。 $\log(\lambda^c) = \log(0.5)$ より、半減期は下式となる：

$$c = \log(0.5) / \log(\lambda)$$

月次モデルでは、 $\lambda = \gamma$

- 仮に $\gamma = 0.6$ なら、半減期は $c = \log(0.5) / \log(0.6) = 1.3$ ヶ月

集計したせいで、効果の持続期間が長くなってしまふ

年次モデルでは、 $\lambda = \frac{(1/12)\gamma}{1 - (11/12)\gamma} = \frac{\gamma}{12 - 11\gamma}$

- 仮に $\gamma = 0.6$ なら、半減期は $c = \log(0.5) / \log(0.11) = 0.32$ 年 (=3.8ヶ月)

■ 教訓：

- できれば、短い時間単位のデータを使って分析したほうがよい
 - ただし、繰越効果や攪乱項の自己相関は複雑になるかも...
- 長い時間単位のデータから推定したマーケティング・ミックス変数の効果は、過大である可能性が高い

福地純一郎・伊藤有希(2011)「Rによる計量経済分析」, 朝倉書店,

Doyle, P., Saunders, J. (1985) The Lead Effect of Marketing Decisions. *Journal of Marketing Research*, 22(1), 54-65.

Leeflang, P.S.H., Mijatovic, G.M., Saunders, J. (1992) Identification and Estimation of Complex Multivariate Lag Structures: A Nesting Approach. *Applied Economics*, 24, 273-283.

Franses, P.H. (1991) Primary Demand for Beer in The Netherlands: An Application of ARMAX Model Specification. *Journal of Marketing Research*, 240-245.

Krider, R.E., Li, T., Liu, Y, Weinberg, C.B. (2005) The Lead-Lag Puzzle of Demand and Distribution: A Graphical Method Applied to Movies. *Marketing Science*, 24(4), 635-645.

Leone, R.P. (1995) Generalizing What Is Known About Temporal Aggregation and Advertising Carryover. *Marketing Science*, 14(3_supplement), G141-G150.

- `stats::filter`(入力ベクトル, `filter` = 係数ベクトル, `method`=手法)
 - 入力ベクトルにARフィルタないしMAフィルタを掛ける
- `stats::ccf`(ベクトル1, ベクトル2)
 - ふたつのベクトルの標本相互相関関数を返し、図示する
- `TSA::prewhiten`(ベクトル1, ベクトル2)
 - ふたつのベクトルのプレホワイト化標本相互相関関数を返し、図示する
- `dynlm::dynlm`(フォーミュラ)
 - 動的回帰モデルを推定する。データはtsオブジェクトとする
- `vars::VARselect`(データ行列)
 - VARモデルの次数を選択する
- `vars::VAR`(データ行列, 次数, `type` = タイプ)
 - VARモデルを推定する
- `vars::causality`(VARオブジェクト)
 - グレンジャー因果性検定を行う
- `vars::irf`(VARオブジェクト)
 - 直交化インパルス応答関数を推定する

Appendix. 動的回帰モデルのためのRパッケージ

以下では、動的回帰モデルを推定する関数を提供しているRパッケージについて紹介します

- Y_t, X_t は定常時系列、 ε_t はホワイトノイズを表します
- 詳細は各パッケージのマニュアルを参照してください
- 抜け漏れや誤りがあると思います。ご容赦ください

```
## 動作確認用データ
set.seed(123)
nLength <- 10000

# dfDirectLag:
# omega[0]=0.3, omega[1]=0.2, theta[1]=0.25, phi[1]=0.1
dfDirectLag <- tibble(gX = rnorm(nLength)) %>%
  mutate(
    gXLag1 = dplyr::lag(gX),
    gState = as.vector(
      0.3 * arima.sim(list(ma=0.2/0.3), n=nLength, innov=gX)
    ),
    gErr = as.vector(
      arima.sim(list(ma=0.25, ar=0.1), n=nLength)
    ),
    gY = gState + gErr
  ) %>%
  drop_na()
mgDirectLagX <- dfDirectLag %>%
  dplyr::select(gX, gXLag1) %>%
  as.matrix(.)
```

```
# dfADL:
# omega[0]=0.4, omega[1]=0.3, phi[1]=0.2
dfADL <- tibble(gX = rnorm(nLength)) %>%
  mutate(
    gState = as.vector(
      0.4 * arima.sim(list(ma=0.3/0.4, ar=0.2), n=nLength, innov=gX)
    ),
    gErr = as.vector(
      arima.sim(list(ar=0.2), n=nLength)
    ),
    gY = gState + gErr
  ) %>%
  drop_na()

# dfTransfer:
# omega[0]=0.3, omega[1]=0.2, delta[1]=0.2, theta[1]=0.25, phi[1]=0.1
dfTransfer <- tibble(gX = rnorm(nLength)) %>%
  mutate(
    gState = as.vector(
      0.3 * arima.sim(list(ma=0.2/0.3, ar=0.2), n=nLength, innov=gX)
    ),
    gErr = as.vector(
      arima.sim(list(ma=0.25, ar=0.1), n=nLength)
    ),
    gY = gState + gErr
  ) %>%
  drop_na()
```

■ forecast::Arima()

- 推定できるモデル：直接ラグモデル+ARMA攪乱項

$$Y_t = \alpha + \frac{\omega_0 + \omega_1 L + \dots + \omega_s L^s}{1} X_t + \frac{1 + \theta_1 L + \dots + \theta_q L^q}{1 - \phi_1 L - \dots - \phi_p L^p} \varepsilon_t$$

- ARIMA攪乱項, SARIMA攪乱項も指定できる
- $s > 0$ の場合、ラグ変数をデータとして与える
- デフォルトでは最尤推定

■ forecast::auto.arima()

- 推定できるモデル：Arima()と同じ
- 攪乱項のARMA次数 (ARIMA次数, SARIMA次数)について自動選択できる

```
# forecast パッケージ - - - - -
library(forecast)

# forecast::Arima()
oModel <- Arima(
  dfDirectLag$gY,      # 目的変数
  order = c(1, 0, 1),  # 攪乱項のARMA次数
  xreg = mgDirectLagX  # 説明変数行列
)
print(oModel)
```

```
Series: dfDirectLag$gY
Regression with ARIMA(1,0,1) errors

Coefficients:
      ar1      ma1  intercept      gX  gXLag1
  0.1185  0.2372   -0.0144  0.2971  0.2139
s.e.  0.0285  0.0278    0.0141  0.0100  0.0100

sigma^2 estimated as 1.004:  log likelihood=-14203.07
AIC=28418.15   AICc=28418.16   BIC=28461.41
```

```
## forecast::auto.arima()
oModel <- auto.arima(
  dfDirectLag$gY,      # 目的変数
  d = 0,               # 差分階数
  xreg = mgDirectLagX, # 攪乱項のARMA次数行列
  trace = TRUE, approximation = FALSE
)
print(oModel)
```

```
Series: dfDirectLag$gY
Regression with ARIMA(0,0,2) errors

Coefficients:
      ma1      ma2      gX  gXLag1
0.3558  0.0414  0.2972  0.214
s.e.  0.0100  0.0100  0.0100  0.010

sigma^2 estimated as 1.004:  log likelihood=-14203.53
AIC=28417.05   AICc=28417.06   BIC=28453.11
```

攪乱項のARMA次数として、データ生成時とは異なる次数が選択されたため、
Arima()とは異なる推定結果となっている

■ nlme::gls()

- 推定できるモデル：直接ラグモデル+ARMA攪乱項

$$Y_t = \alpha + \frac{\omega_0 + \omega_1 L + \cdots \omega_s L^s}{1} X_t + \frac{1 + \theta_1 L + \cdots + \theta_q L^q}{1 - \phi_1 L - \cdots - \phi_p L^p} \varepsilon_t$$

- $s > 0$ の場合、ラグ変数をデータとして与える
- ARIMA攪乱項は指定できない。差分系列をデータとして与える
- FGLS推定

```
## nlmeパッケージ - - - - -
library(nlme)

## nlme::gls()
oModel <- gls(
  gY ~ gX + gXLag1,
  corr = corARMA(p = 1, q = 1),      # ARMA攪乱項の次数
  data = dfDirectLag %>% head(1000) # 計算時間節約のため時点数を減らした
)
print(summary(oModel))
```

Generalized least squares fit by REML

Model: gY ~ gX + gXLag1

Data: dfDirectLag %>% head(1000)

AIC	BIC	logLik
2863.836	2893.264	-1425.918

Correlation Structure: ARMA(1,1)

Formula: ~1

Parameter estimate(s):

Phi1	Theta1	
0.1353544	0.1875080	Phi: AR係数 Theta: MA係数

Coefficients:

	value	Std.Error	t-value	p-value
(Intercept)	0.02957195	0.04347242	0.680246	0.4965
gX	0.30609021	0.03186142	9.606922	0.0000
gXLag1	0.25455108	0.03187183	7.986711	0.0000

(後略)

■ dynlm::dynlm()

- 推定できるモデル：自己回帰分布ラグモデル

$$Y_t = \alpha + \frac{\omega_0 + \omega_1 L + \cdots \omega_s L^s}{1 - \phi_1 L - \cdots - \phi_p L^p} X_t + \frac{1}{1 - \phi_1 L - \cdots - \phi_p L^p} \varepsilon_t$$

- ARIMA攪乱項、SARIMA攪乱項も指定できるが、いずれもMAは指定できない
- データはtsオブジェクトとして与える
 - tsオブジェクトについてはv章付録を参照
- 推定方法は最小二乗法を拡張した方法 (二段階最小二乗法)

```
## dynlmパッケージ - - - - -
library(dynlm)

# 使用するデータはtsオブジェクトにする
tsY <- ts(dfADL$gY)
tsX <- ts(dfADL$gX)

## dynlm::dynlm()
## L(変数名, 次数) でラグ変数を指定
oModel <- dynlm(tsY ~ L(tsX, 0:1) + L(tsY, 1))
print(summary(oModel))
```

(前略)

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-0.006581	0.010037	-0.656	0.512
L(tsX, 0:1)0	0.393989	0.009930	39.675	<2e-16 ***
L(tsX, 0:1)1	0.290703	0.010541	27.578	<2e-16 ***
L(tsY, 1)	0.211777	0.009153	23.138	<2e-16 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(後略)

■ TSA::arimax()

- 推定できるモデル：伝達関数モデルの一般形

$$Y_t = \alpha + \frac{\omega_0 + \omega_1 L + \cdots + \omega_s L^s}{1 - \delta_1 L - \cdots - \delta_r L^r} X_t + \frac{1 + \theta_1 L + \cdots + \theta_q L^q}{1 - \phi_1 L - \cdots - \phi_p L^p} \varepsilon_t$$

- ARIMA攪乱項, SARIMA攪乱項も指定できる
- 説明変数ごとに繰越率が異なる幾何級数型ラグモデルを指定できるのは、このパッケージだけかも
- 部分調整モデル・自己回帰分布ラグモデルのように $\delta_1 = \phi_1, \delta_2 = \phi_2, \dots$ と制約することはできない模様
- デフォルトでは最尤推定

```
## TSAパッケージ - - - - -
library(TSA)

## TSA::arimax()
oModel <- arimax(
  dfTransfer$gY[1:1000],
  order      = c(1, 0, 1), # 攪乱項のARMA次数
  xtransf    = as.matrix(dfTransfer$gX[1:1000]), # 説明変数行列
  transfer   = list(c(1,1)), # 説明変数のARMA次数。
                                # 説明変数が複数あるときはその順に並べる
  method     = "ML"
)
print(oModel)
```

(前略)

説明変数のARMA係数

Coefficients:			説明変数のARMA係数		
ar1	ma1	intercept	T1-AR1	T1-MA0	T1-MA1
-0.0468	0.3742	-0.0057	0.1917	0.2938	0.190
s.e. 0.0980	0.0914	0.0412	0.1208	0.0313	0.045

sigma^2 estimated as 0.9843: log likelihood = -1409.67, aic = 2831.34

- dynパッケージ
 - 推定できるモデル：自己回帰分布ラグモデル
- getsパッケージ
 - 推定できるモデル：自己回帰分布ラグモデル
- marimaパッケージ
 - 推定できるモデル：自己回帰分布ラグモデル
- sysidパッケージ
 - 推定できるモデル：自己回帰分布ラグモデル+ARMA攪乱項
- dseパッケージ
 - 推定できるモデル：自己回帰分布ラグモデル+ARMA攪乱項
 - 使い方が難しそう...
- dlsemパッケージ
 - 推定できるモデル：直接ラグモデル, 負の二項分布ラグモデル, 多項分布ラグモデル
- dlnmパッケージ
 - 推定できるモデル：直接ラグモデル
 - 非線形的な関数形を指定できる模様
- dLagMパッケージ
 - 推定できるモデル：直接ラグモデル、幾何級数型分布ラグモデル、自己回帰分布ラグモデル、多項分布ラグモデル
 - 説明変数は1つしか扱えない
- dlmパッケージ, KFASパッケージ
 - 状態空間モデリングのための汎用的パッケージ
 - VII章で紹介します