Box-Jenkins 法というフレームワーク

R による ARIMA モデル

柳樂 優太 (1260404)

2024-09-02

目次

準備 2

1 **分析の対象** 3

2 **対数変化** 3

3 **差分系列の作成方法** 4

4 **季節成分の取り扱い** 6

5 **自己相関とコレログラム** 8

6 **訓練データとテストデータに分ける** 10

7 **ARIMA モデルの推定** 10

8 **補足** 11

8.1 差分系列と ARIMA の次数の関係 11

9 **自動選択モデル auto.arima 関数** 13

10 **定常性・反転可能性のチェック** 14

11 **残差のチェック** 15

12 **ARIMA による予測** 16

13 **ナイーブ予測** 18

14 **予測の評価** 18

15 発展 19
15.1 非定常過程への分析 19

準備

```
## PDF に出力する際は cairo を使用する
if (knitr::is_latex_output()) {
  knitr::opts_chunk$set(dev = "cairo_pdf")
}

#パッケージの読み込み
pacman::p_load(tidyverse,
               broom,
               coefplot,
               texreg,
               bayesplot,
               rstan,
               rstanrm,
               parallel,
               posterior,
               cmdstanr,
               patchwork,
               ggplot2,
               tidybayes,
               ggfortify,
               gridExtra,
               forecast,
               tseries,
               summarytools,
               forecast
               )


#日本語の設定
if (.Platform$OS.type == "windows") {
  if (require(fontregisterer)) {
    my_font <- "Yu Gothic"
  } else {
    my_font <- "Japan1"
```

```
  }
} else if (capabilities("aqua")) {
  my_font <- "HiraginoSans-W3"
} else {
  my_font <- "IPAexGothic"
}

theme_set(theme_gray(base_size = 9,
                     base_family = my_font))
```

```
#計算の高速化
rstan_options(auto_write = TRUE)
options(mc.cores = parallel::detectCores())
```

1　分析の対象

```
front <- Seatbelts[, "front"]
```

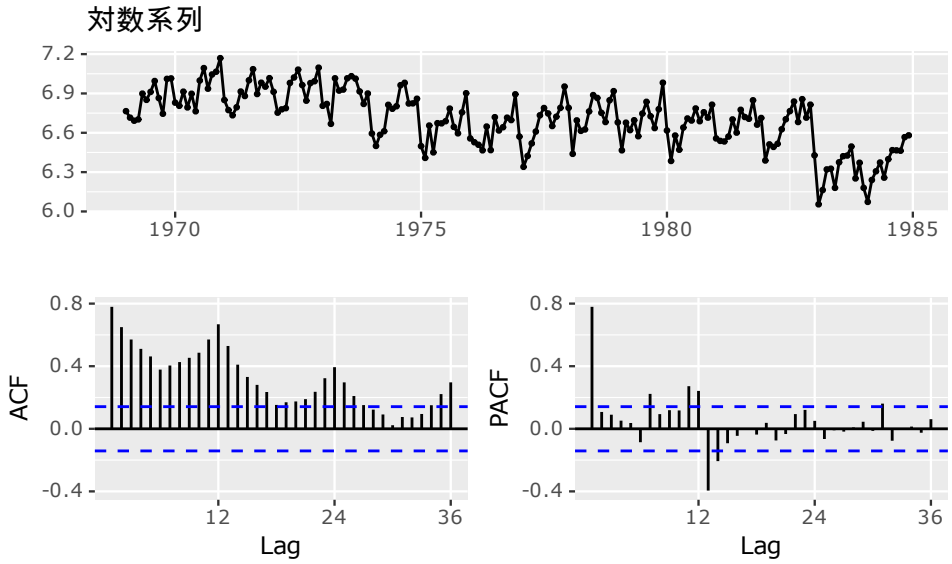
交通事故の死傷者をモデル化する

季節性は当然あるだろう

また, ガソリンの値段や法案によって死傷者も変化するだろう

2　対数変化

```
#対数系列
log_front <- log(front)

#図示
ggtsdisplay(log_front ,main = "対数系列" )
```



右下の偏自己相関でも一年単位での大きな自己相関が見られる.

3　差分系列の作成方法

```
#原型列
front
```

```
      Jan  Feb  Mar  Apr  May  Jun  Jul  Aug  Sep  Oct  Nov  Dec
1969  867  825  806  814  991  945 1004 1091  958  850 1109 1113
1970  925  903 1006  892  990  866 1095 1204 1029 1147 1171 1299
1971  944  874  840  893 1007  973 1097 1194  988 1077 1045 1115
1972 1005  857  879  887 1075 1121 1190 1058  939 1074 1089 1208
1973  903  916  787 1114 1014 1022 1114 1132 1111 1008  916  992
1974  731  665  724  744  910  883  900 1057 1076  919  920  953
1975  664  607  777  633  791  790  803  884  769  732  859  994
1976  704  684  671  643  771  644  828  748  767  825  810  986
1977  714  567  616  678  742  840  888  852  774  831  889 1046
1978  889  626  808  746  754  865  980  959  856  798  942 1010
1979  796  643  794  750  809  716  851  931  834  762  880 1077
1980  748  593  720  646  765  820  807  885  803  860  825  911
1981  704  691  688  714  814  736  876  829  818  942  782  823
1982  595  673  660  676  755  815  867  933  798  950  825  911
1983  619  426  475  556  559  483  587  615  618  662  519  585
1984  483  434  513  548  586  522  601  644  643  641  711  721
```

```
#ラグをとった
stats::lag(front,-1)
```

```
      Jan  Feb  Mar  Apr  May  Jun  Jul  Aug  Sep  Oct  Nov  Dec
1969       867  825  806  814  991  945 1004 1091  958  850 1109
1970 1113  925  903 1006  892  990  866 1095 1204 1029 1147 1171
1971 1299  944  874  840  893 1007  973 1097 1194  988 1077 1045
1972 1115 1005  857  879  887 1075 1121 1190 1058  939 1074 1089
1973 1208  903  916  787 1114 1014 1022 1114 1132 1111 1008  916
1974  992  731  665  724  744  910  883  900 1057 1076  919  920
1975  953  664  607  777  633  791  790  803  884  769  732  859
1976  994  704  684  671  643  771  644  828  748  767  825  810
1977  986  714  567  616  678  742  840  888  852  774  831  889
1978 1046  889  626  808  746  754  865  980  959  856  798  942
1979 1010  796  643  794  750  809  716  851  931  834  762  880
1980 1077  748  593  720  646  765  820  807  885  803  860  825
1981  911  704  691  688  714  814  736  876  829  818  942  782
1982  823  595  673  660  676  755  815  867  933  798  950  825
1983  911  619  426  475  556  559  483  587  615  618  662  519
1984  585  483  434  513  548  586  522  601  644  643  641  711
1985  721
```

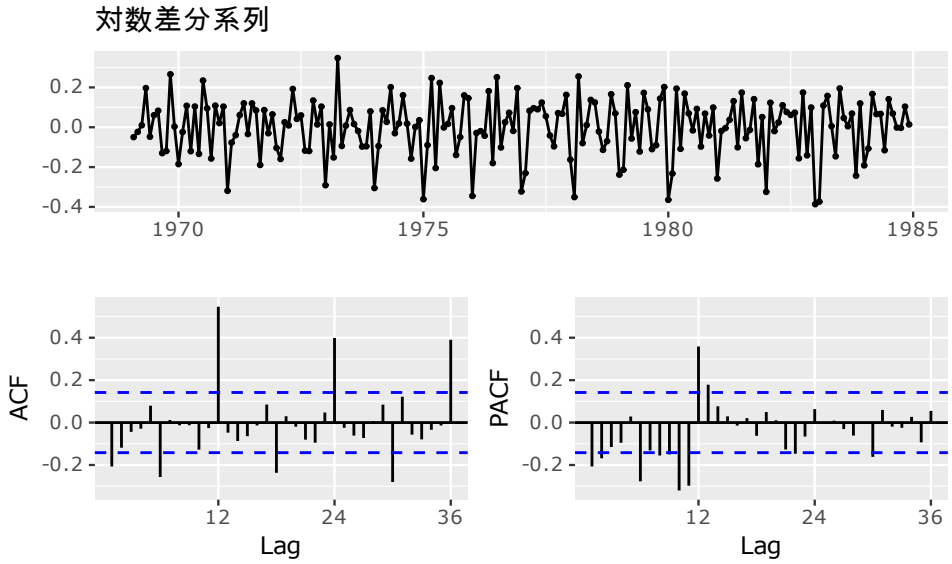
ラグから現系列を引くここで差分系列が手に入る

```
front - stats::lag(front,-1)
```

```
      Jan  Feb  Mar  Apr  May  Jun  Jul  Aug  Sep  Oct  Nov  Dec
1969       -42  -19    8  177  -46   59   87 -133 -108  259    4
1970 -188  -22  103 -114   98 -124  229  109 -175  118   24  128
1971 -355  -70  -34   53  114  -34  124   97 -206   89  -32   70
1972 -110 -148   22    8  188   46   69 -132 -119  135   15  119
1973 -305   13 -129  327 -100    8   92   18  -21 -103  -92   76
1974 -261  -66   59   20  166  -27   17  157   19 -157    1   33
1975 -289  -57  170 -144  158   -1   13   81 -115  -37  127  135
1976 -290  -20  -13  -28  128 -127  184  -80   19   58  -15  176
1977 -272 -147   49   62   64   98   48  -36  -78   57   58  157
1978 -157 -263  182  -62    8  111  115  -21 -103  -58  144   68
1979 -214 -153  151  -44   59  -93  135   80  -97  -72  118  197
1980 -329 -155  127  -74  119   55  -13   78  -82   57  -35   86
1981 -207  -13   -3   26  100  -78  140  -47  -11  124 -160   41
1982 -228   78  -13   16   79   60   52   66 -135  152 -125   86
```

```
1983 -292 -193   49   81    3  -76  104   28    3   44 -143   66
1984 -102  -49   79   35   38  -64   79   43   -1   -2   70   10
```

```
#対数差分系列
log_diff <- diff(log_front)

#図示

ggtsdisplay(log_diff,main = "対数差分系列")
```

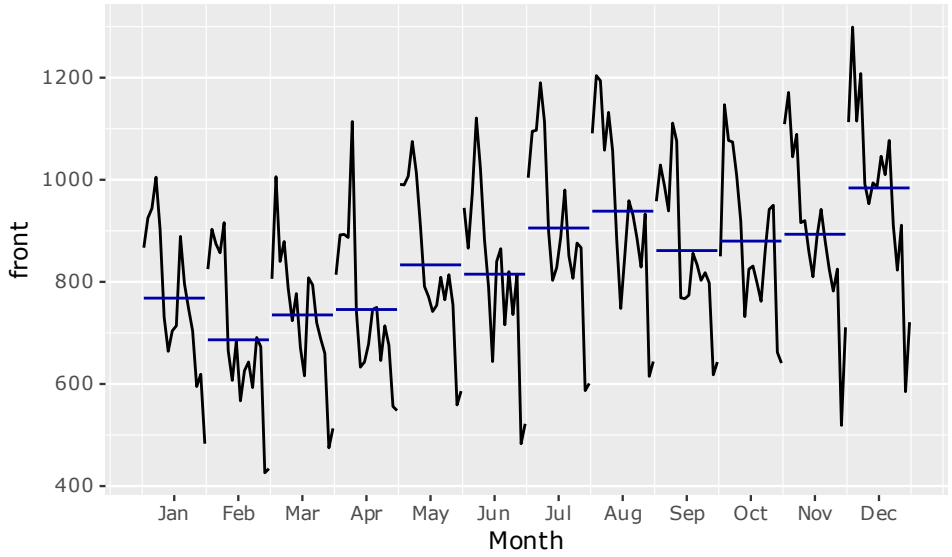


差分系列は長期にわたって平均値が変化せず, 単位根がないことグラフからも分かる.

4　季節成分の取り扱い

1 年周期での自己相関が目立った. 季節性があると言うことである.

```
#1 月毎の図

ggsubseriesplot(front)
```




```
#季節差分をとってみる
frequency(front)
```

```
[1] 12
```

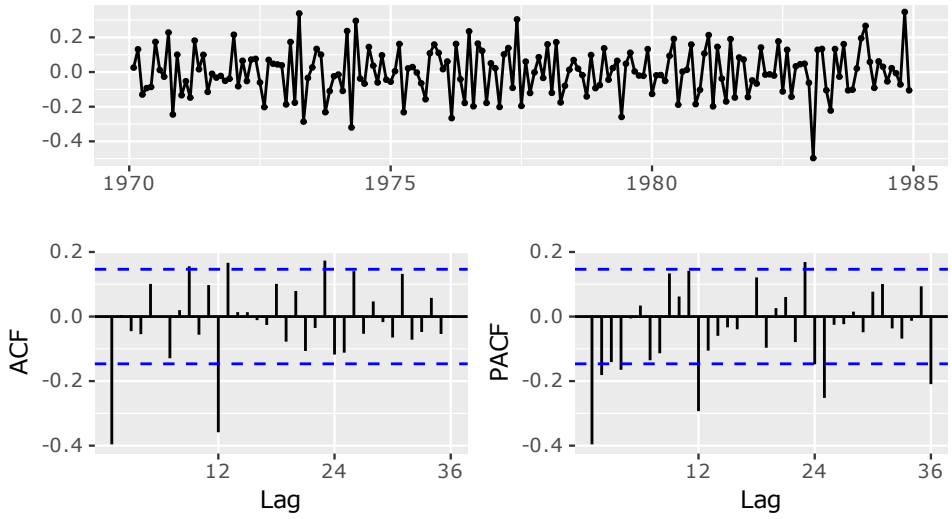
```
diff(front,lag = frequency(front))
```

```
      Jan  Feb  Mar  Apr  May  Jun  Jul  Aug  Sep  Oct  Nov  Dec
1970   58   78  200   78   -1  -79   91  113   71  297   62  186
1971   19  -29 -166    1   17  107    2  -10  -41  -70 -126 -184
1972   61  -17   39   -6   68  148   93 -136  -49   -3   44   93
1973 -102   59  -92  227  -61  -99  -76   74  172  -66 -173 -216
1974 -172 -251  -63 -370 -104 -139 -214  -75  -35  -89    4  -39
1975  -67  -58   53 -111 -119  -93  -97 -173 -307 -187  -61   41
1976   40   77 -106   10  -20 -146   25 -136   -2   93  -49   -8
1977   10 -117  -55   35  -29  196   60  104    7    6   79   60
1978  175   59  192   68   12   25   92  107   82  -33   53  -36
1979  -93   17  -14    4   55 -149 -129  -28  -22  -36  -62   67
1980  -48  -50  -74 -104  -44  104  -44  -46  -31   98  -55 -166
1981  -44   98  -32   68   49  -84   69  -56   15   82  -43  -88
1982 -109  -18  -28  -38  -59   79   -9  104  -20    8   43   88
1983   24 -247 -185 -120 -196 -332 -280 -318 -180 -288 -306 -326
1984 -136    8   38   -8   27   39   14   29   25  -21  192  136
```

次は対数差分系列に対して, 季節性とってみる

```
#対数差分にさらに季節差分をとる
```

```
seas_log_diff <- diff(log_diff,lag = frequency(log_front))

#図示

ggtsdisplay(seas_log_diff)
```



季節階差をとっても, 影響を全て取り除けるわけではない.

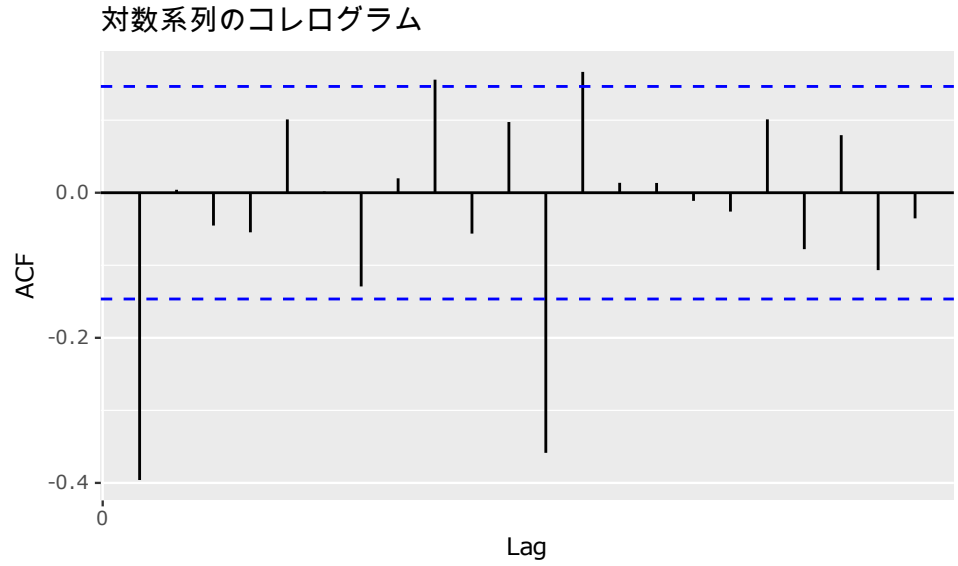
5 自己相関とコレログラム

自己相関の図示はしてきたが, 数値で欲しいときもある

```
#自己相関
acf(seas_log_diff,plot = F, lag.max = 12)
```

```
Autocorrelations of series 'seas_log_diff', by lag

0.0000 0.0833 0.1667 0.2500 0.3333 0.4167 0.5000 0.5833 0.6667 0.7500 0.8333
 1.000 -0.396  0.004 -0.045 -0.055  0.101  0.002 -0.129  0.020  0.156 -0.056
0.9167 1.0000
 0.097 -0.359
```

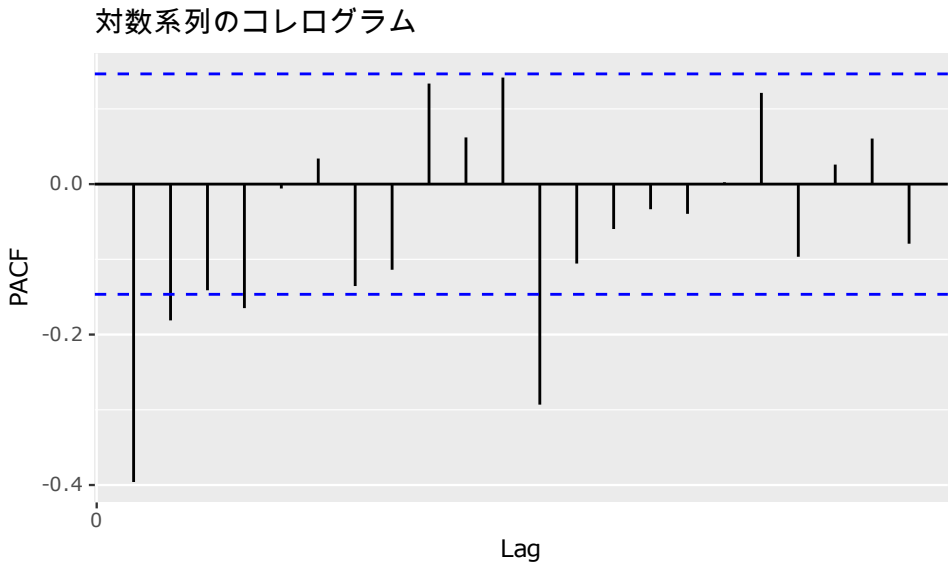
```
#図示
autoplot(
  acf(seas_log_diff,plot = F),
  main = "対数系列のコレログラム"
)
```




```
#編相関係数
pacf(seas_log_diff,plot = F, lag.max = 12)
```

```
Partial autocorrelations of series 'seas_log_diff', by lag

0.0833 0.1667 0.2500 0.3333 0.4167 0.5000 0.5833 0.6667 0.7500 0.8333 0.9167
-0.396 -0.181 -0.141 -0.165 -0.006  0.034 -0.135 -0.114  0.134  0.062  0.142
1.0000
-0.293
```

```
#図示
autoplot(
  pacf(seas_log_diff,plot = F),
  main = "対数系列のコレログラム"
)
```



6　訓練データとテストデータに分ける

予測のために, 訓練データとテストデータに分割する

まずは対数変換したグラフを作る

```
Seatbelts_log <- Seatbelts[,c("front", "PetrolPrice", "law")]

Seatbelts_log[,"front"] <- log(Seatbelts[,"front"])

Seatbelts_log[,"PetrolPrice"] <- log(Seatbelts[,"PetrolPrice"])
```

最後の一年をテストデータとする

```
train <- window(Seatbelts_log,end = c(1983,12))

test <- window(Seatbelts_log,start = c(1984,1))
```

今回のモデルでは front が予測対象であり, 応答変数である.

```
#説明変数だけを切り出す
petro_law <- train[,c("PetrolPrice","law")]
```

7　ARIMA モデルの推定

```
model_sarimax <- Arima(
  y = train[,"front"],
```

```
  order = c(1,1,1),                    #(p,d,q)
  seasonal = list(order = c(1,0,0)),    #(P,D,Q)
  xreg = petro_law
)

model_sarimax
```

```
Series: train[, "front"]
Regression with ARIMA(1,1,1)(1,0,0)[12] errors

Coefficients:
         ar1      ma1    sar1  PetrolPrice      law
      0.2589  -0.9503  0.6877      -0.3464  -0.3719
s.e.  0.0826   0.0303  0.0548       0.0955   0.0467

sigma^2 = 0.009052:  log likelihood = 165.33
AIC=-318.66   AICc=-318.18   BIC=-299.54
```

8 補足

8.1 差分系列と ARIMA の次数の関係

平易にするために, 定数項は入れない

- 差分系列と ARIMA の字数を確認する

```
Arima(
  y = log_diff, order = c(1, 0, 0),
  include.mean = F
)
```

```
Series: log_diff
ARIMA(1,0,0) with zero mean

Coefficients:
          ar1
      -0.2058
s.e.   0.0706

sigma^2 = 0.0202:  log likelihood = 102.1
AIC=-200.21   AICc=-200.15   BIC=-193.7
```

この結果は実質アリマ (1,1,0) であることを確認する

```
Arima(
  y = log_front, order = c(1, 1, 0),

  include.mean = F         #定数項なし
)
```

```
Series: log_front
ARIMA(1,1,0)

Coefficients:
          ar1
      -0.2058
s.e.   0.0706

sigma^2 = 0.0202:  log likelihood = 102.1
AIC=-200.21   AICc=-200.15   BIC=-193.7
```

- SARIMA と季節階差の関係を確認する

対数差分系列に季節階差を導入したデータに ARIMA(1,0,0) を適応する

```
Arima(
  y = seas_log_diff,order = c(1,0,0),

  include.mean = F           #定数項なし
)
```

```
Series: seas_log_diff
ARIMA(1,0,0) with zero mean

Coefficients:
          ar1
      -0.3951
s.e.   0.0685

sigma^2 = 0.01569:  log likelihood = 118.26
AIC=-232.52   AICc=-232.45   BIC=-226.15
```

これは実質 SARIMA(1,1,0)(0,1,0) であることを確認する

```
Arima(
  y = log_front,order = c(1,1,0),
  seasonal = list(order = c(0,1,0))
)
```

```
Series: log_front
ARIMA(1,1,0)(0,1,0)[12]

Coefficients:
          ar1
      -0.3951
s.e.   0.0685

sigma^2 = 0.0157:  log likelihood = 118.26
AIC=-232.52   AICc=-232.45   BIC=-226.15
```

9　自動選択モデル auto.arima 関数

字数の決定は手順は AIC を比較することだが, 時間がかかるので自動化する.

```
sarimax_petro_law <- auto.arima(
  y = train[,"front"],
  xreg = petro_law,
  ic = "aic",
  max.order = 7,              #p+q+P+Q
  stepwise = F,
  approximation = F,
  parallel = T,
  num.cores = 4
)
```

```
sarimax_petro_law
```

```
Series: train[, "front"]
Regression with ARIMA(2,0,1)(0,1,1)[12] errors

Coefficients:
         ar1      ar2      ma1     sma1  PetrolPrice      law
      1.1225  -0.1322  -0.8690  -0.8183      -0.3748  -0.3431
s.e.  0.0906   0.0876   0.0443   0.1129       0.1000   0.0473
```

```
sigma^2 = 0.007624:  log likelihood = 168.12
AIC=-322.23   AICc=-321.53   BIC=-300.36
```

最良とされるモデルを数式に起こしてみる

$$\left(1-\sum_{i=1}^{2}\phi_i B^i\right)\left(1-\sum_{i=1}^{0}\Phi_i B^i\right)\Delta^0\Delta_{12}^1 y_t = \left(1+\sum_{j=1}^{1}\theta_j B^j\right)\left(1+\sum_{j=1}^{1}\Theta_j B^j\right)\epsilon_t + \sum_{k=1}^{2}\beta_k x_{k,t}$$

つまりは, まとめると以下のようになる

$$\left(1-\phi_1 B^1-\phi_2 B^2\right)\Delta_{12}^1 y_t = \left(1+\theta_1 B^1\right)\left(1+\Theta_1 B^1\right)\epsilon_t + \beta_1 x_{1,t}+\beta_2 x_{2,t}$$

$$\left(1-1.1225B^1+0.1322B^2\right)\Delta_{12}^1 y_t = \left(1-0.8690B^1\right)\left(1-0.8183B^1\right)\epsilon_t - 0.3748x_{1,t}-0.3431x_{2,t}$$

10　定常性・反転可能性のチェック

特性方程式の解の絶対値 $|z|$ が 1 異常であれば反転可能性と定常性を持つ

$$1-\phi_1 z-\phi_2 z^2=0$$
$$1+\theta_1 z=0$$
$$1+\Theta_1 z=0$$

$$1-1.1225z+0.1322z^2=0$$
$$1-0.8690z=0$$
$$1-0.8183z=0$$

このチェックは `auto.arima` **関数の中ですでに行われている**

特定方程式の絶対値を求めるコード

```
#AR 項
abs(polyroot(c(1, -coef(sarimax_petro_law)[c("ar1","ar2")])))
```

```
[1] 1.011397 7.477827
```

```
#MA 項
abs(polyroot(c(1, coef(sarimax_petro_law)[c("ma1")])))
```

```
[1] 1.150755
```

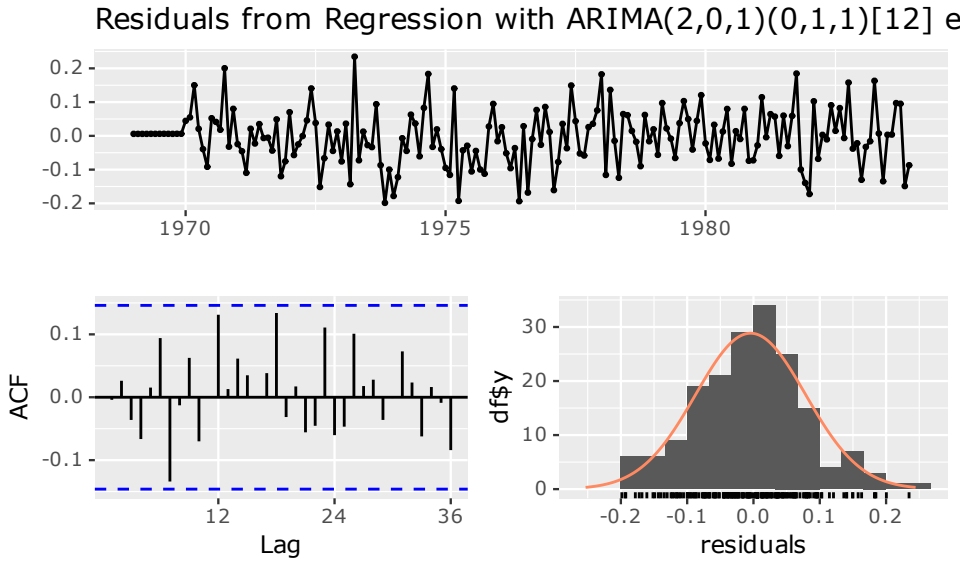
```
#SAR 項

#今回はなかった

#SMA 項
abs(polyroot(c(1, coef(sarimax_petro_law)[c("sma1")])))
```

```
[1] 1.222057
```

11　残差のチェック

- まずは残差の自己相関の検定を行う

```
checkresiduals(sarimax_petro_law)
```




```
    Ljung-Box test

data:  Residuals from Regression with ARIMA(2,0,1)(0,1,1)[12] errors
Q* = 20.99, df = 20, p-value = 0.3977

Model df: 4.   Total lags used: 24
```

統計的優位ではない. 何もわからなかった. 異常の発見ができなかっただけで, 良いモデルの保証はない.

- 残差の正規性の検定

```
jarque.bera.test(resid(sarimax_petro_law))
```

```
    Jarque Bera Test

data:  resid(sarimax_petro_law)
X-squared = 0.39938, df = 2, p-value = 0.819
```

正規分布と有意に異なっているとは言えない. 何もわからなかった.

12 ARIMA による予測

同定されたモデルを使って予測をします

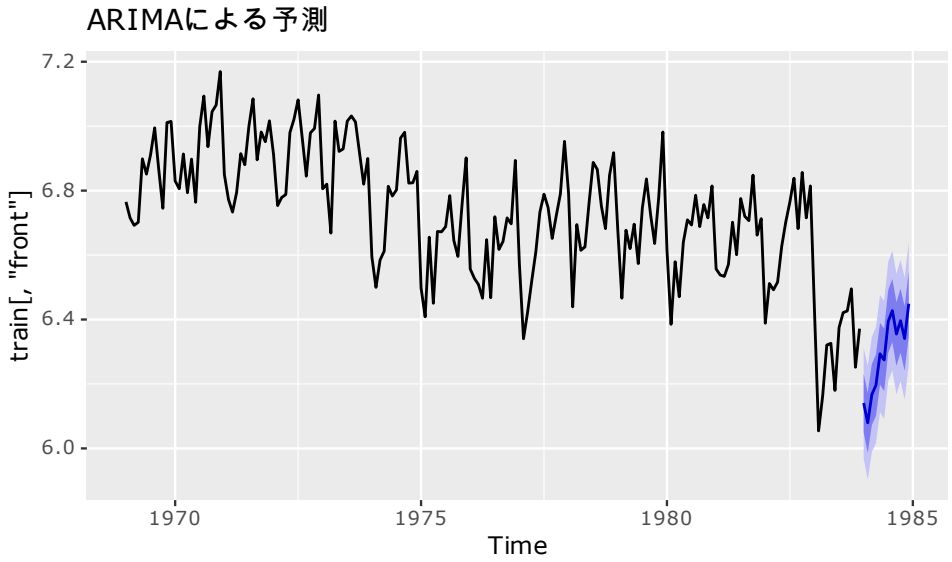
予測制度の評価にはテストデータを使う. 説明変数のデータを作った上で予測を行う

```
petro_law_test <- test[, c("PetrolPrice","law")]

sarimax_f <- forecast(
   sarimax_petro_law,
   xreg = petro_law_test,
   h = 12,                  #12 時点先まで予測する
  level = c(95,70)         #95 と 70 信頼区間んも出す
  )

sarimax_f
```

```
         Point Forecast    Lo 70    Hi 70    Lo 95    Hi 95
Jan 1984       6.140922 6.050389 6.231454 5.969719 6.312124
Feb 1984       6.079469 5.986077 6.172861 5.902858 6.256080
Mar 1984       6.167368 6.072965 6.261771 5.988846 6.345890
Apr 1984       6.196306 6.101088 6.291523 6.016243 6.376368
May 1984       6.293745 6.197758 6.389732 6.112227 6.475262
Jun 1984       6.274492 6.177762 6.371222 6.091568 6.457416
Jul 1984       6.394263 6.296812 6.491714 6.209976 6.578550
Aug 1984       6.427685 6.329534 6.525836 6.242075 6.613294
Sep 1984       6.354929 6.256100 6.453759 6.168036 6.541823
Oct 1984       6.396352 6.296863 6.495840 6.208212 6.584491
Nov 1984       6.340966 6.240838 6.441095 6.151616 6.530317
Dec 1984       6.448938 6.348188 6.549689 6.258413 6.639464
```

結果の図示

```
autoplot(sarimax_f,predict.clour = 1,main = "ARIMA による予測")
```



将来の石油価格は本来わからない

将来の石油価格の代理変数を考えないといけない

- 過去の石油価格を予測に使う

```
petro_law_mean <- data.frame(
     PetrolPrice = rep(mean(train[,"PetrolPrice"]),12),
     law = rep(1,12)
   )

   petro_law_mean <- as.matrix(petro_law_mean)

   sarimax_f_mean <- forecast(sarimax_petro_law, xreg = petro_law_mean)
```

- 直前の値を予測に使う

```
petro_law_tail <- data.frame(
     PetrolPrice = rep(tail(train[,"PetrolPrice"], n = 1),12),
    law = rep(1,12)
    )

   petro_law_tail <- as.matrix(petro_law_tail)

   sarimax_f_tail <- forecast(sarimax_petro_law, xreg = petro_law_tail)
```

13　ナイーブ予測

- 過去の平均を予測値として使う

```
naive_f_mean <- meanf(train[,"front"], h = 12)
```

- 前時点の値を予測値として使う

```
naive_f_latest <- rwf(train[,"front"], h = 12)
```

14　予測の評価

RMSE は以下のように定義されている

$$\sqrt{\frac{1}{T}\sum_{t=1}^{T} e_t^2} = \sqrt{\frac{1}{T}\sum_{t=1}^{T}(y_t - \hat{y}_t)^2}$$

- 将来の石油価格が既知

R での実装は以下の通り

```
sarimax_rmse <- sqrt(
    sum((sarimax_f$mean - test[,"front"])^2) /
       length(sarimax_f$mean)
   )


 sarimax_rmse
```

```
[1] 0.09674572
```

または

```
accuracy(sarimax_f,x = test[, "front"])[,"RMSE"]
```

```
Training set     Test set
  0.08283297   0.09674572
```

RMSE は 1 に近いほど良い指標であることを考慮すると, 予測制度は当てはめ精度に劣っていることがわかる.

- 価格が未知である.

```
# 石油価格の平均を使用する
accuracy(sarimax_f_mean, x = test[, "front"])["Test set", "RMSE"]
```

```
[1] 0.06945114
```

```
# 直近の石油価格を使用する

accuracy(sarimax_f_tail, x = test[, "front"])["Test set", "RMSE"]
```

```
[1] 0.1018344
```

- ナイーブ予測

```
# ナイーブ予測　過去の平均値

accuracy(naive_f_mean, x = test[, "front"])["Test set","RMSE"]
```

```
[1] 0.3949872
```

```
# ナイーブ予測　直近の値

accuracy(naive_f_latest, x = test[, "front"])["Test set","RMSE"]
```

```
[1] 0.1498196
```

15　発展

15.1　非定常過程への分析

詳細は note