

```
#####
##第十五回「時系列解析」
#####

library(tseries)

#####
## 季節成分の除去
# 古典的時系列分解関数 decompose
# 時系列データを、季節(周期)成分、傾向(大域的平均)、不規則(ランダム)成分に分解
#  $X_t = p_t + m_t + e_t$  (加法モデル)
#  $X_t = p_t * m_t + e_t$  (乗法モデル)
# 炭酸ガス濃度データ co2 を使用
co2
m <- decompose(co2) # 既定の加法モデルの当てはめ
m$figure # 推定された季節成分
plot(m) # 原データ、傾向成分、季節成分、不規則成分のグラフ
m <- decompose(co2, type="multiplicative") # 乗法モデルの当てはめ
m$figure # 推定された季節成分
plot(m) # 原データ、傾向成分、季節成分、不規則成分のグラフ
#####
# 系列分解関数 stl
# ノッティングラム城の気温データ nottem を使用
op <- par(mfrow=c(4,1))
plot(stl(nottem, "per"), set.pars=NULL) # 周期条件で分解
# 炭酸ガス濃度データ co2 を使用. 対数値を分解
plot(stl1c <- stl(log(co2), s.window=21), set.pars=NULL)
par(mfrow=c(1,1))
#####
## 核関数平滑化
## 大局的変動を見るため平滑化操作で不規則成分を除去
k1 <- kernel("daniell", 50) # Daniell核関数を用いた長期の移動平均用重み係数
k2 <- kernel("daniell", 10) # その短期版
k3 <- kernel("dirichlet", 10, r=1) # Dirichlet核関数を用いた長期の移動平均用重み係数
k4 <- kernel("dirichlet", 50, r=3) # Dirichlet核関数を用いた長期の移動平均用重み係数
# 核関数には"daniell", "dirichlet", "fejer", "modified.daniell"がある.

plot(k1)
plot(k2)
plot(k3)
plot(k4)

# ヨーロッパ株価指標データ EuStockMarkets を使用
x <- EuStockMarkets[,1] # そのうちのドイツの株価指数 DAX 系列
x1 <- kernapply(x, kernel("daniell", 50)) # 長期の移動平均
x2 <- kernapply(x, kernel("daniell", 10)) # 短期版
x3 <- kernapply(x, kernel("dirichlet", 50, r=3)) # Dirichlet, 長期
x4 <- kernapply(x, kernel("dirichlet", 10, r=1)) # Dirichlet, 短期

plot(x)
lines(x1, col = "red")
lines(x2, col = "green")
lines(x3, col = "blue")
lines(x4, col = "cyan")

#####
## TOPIX CORE 30銘柄の時系列解析

#TOPIX CORE 30に含まれる銘柄のインデックスと名前を取得
names <- read.csv("./topix/TOPIX_CORE_30_NAMES.csv", header=TRUE)

#各銘柄の時系列データの取得, 2013/6/28--2014/7/4の250日分
topix30 <- matrix(1:(250*30), ncol=30)
logrt <- matrix(1:(250*30), ncol=30)

for(i in 1:30){
  index <- names[i,2]
  tmp <- read.csv(paste("./topix/", index, "-T.csv", sep=""), skip=2, header=FALSE)
```

```

    topix30[, i] <- as.matrix(tmp[, 5]) #終値の値
}

# 株式分割などを修正
# アステラス製薬と三菱UFJ
topix30[70:250, 6] <- topix30[70:250, 6]/5 #アステラス2014年3月31日に5分割
topix30[190:250, 29] <- topix30[190:250, 29]/100 #三菱UFJ
# plot(topix30[, 6])
# plot(topix30[, 29])

# 当日と前日の株価の比を取り、log変換（日足収益率）
logrt <- log(topix30[2:250, ]/topix30[1:249, ])
colnames(topix30) <- names[, 3] #列に銘柄の名前を付与
colnames(logrt) <- names[, 3] #列に銘柄の名前を付与

# 自己相関
acf(logrt[, 20]) #三菱UFJ: 自己相関は小さいが直近の値との相関が見て取れる.
acf(logrt[, 1]) #日本たばこ産業ややsinカーブ的な相関

# ARモデルのあてはめ、230日分、三菱UFJ
ufjts <- ts(logrt[1:230, 20]) #時系列データ構造に変換
plot(ufjts)
ar.ufj <- ar(ufjts) #AR
plot(0:(length(ar.ufj$aic)-1), ar.ufj$aic, type="l") # AICのプロット、AR次数4で最小

ar.ufj$ar #AR係数
qqnorm(ar.ufj$resid) #残差の正規性確認
qqline(ar.ufj$resid) #残差の正規性検定
acf(ar.ufj$resid[(ar.ufj$order+1):230]) #ノイズの自己相関

shapiro.test(ar.ufj$resid[5:230]) #正規性の検定、棄却はされない.
jarque.bera.test(ar.ufj$resid[5:230]) #別の正規性検定

# 予測
(ar.ufj.pred<-predict(ar.ufj, n.ahead=20))
predu<-ar.ufj.pred$pred+2*ar.ufj.pred$se
predl<-ar.ufj.pred$pred-2*ar.ufj.pred$se
ts.plot(ts(logrt[, 20]), ar.ufj.pred$pred, predu, predl, gpars=list(lty=c(1, 2, 3, 3), col
=c(1, 2, 4, 4), lwd=c(1, 2, 2, 2)))

legend(x=1, y=0.05, c("実測値", "予測値", "2*SE"), lty=c(1, 2, 3), col=c(1, 2, 4))

# ベクトル自己回帰モデル
library(vars)

tstmp <- ts(logrt[, 10:14])
varsel <- VARselect(tstmp, lag.max=5) #モデル選択
varsel$selection #AR次数は1が選択される
var.topix <- VAR(tstmp, p=varsel$selection[1]) #AICによるモデル選択結果を使って推定
var.topix$varresult

summary(var.topix$varresult[[1]])
summary(var.topix$varresult[[2]])
summary(var.topix$varresult[[3]])
summary(var.topix$varresult[[4]])

#####
## おまけ
## 収益率が上がるか下がるかを判別
set.seed(101)
index <- 20 #UFJを予測、lags=30
xindex <- 18:25 #UFJを予測するのに用いる変数

## 過去3日分を説明変数とする判別問題を構成
lags = 3
stackmat <- logrt[lags:249, xindex]
for(i in 2:lags){
stackmat <- cbind(stackmat, logrt[(lags-i+1):(249-i+1), xindex])

```

```

}
ynumel <- logrt[lags:249,]
y <- ifelse(ynumel>0, 1, 0)

trnum <- 200
trdata <- list(y=y[(lags+1):(trnum+1), index], x=as.matrix(stackmat[(lags):trnum, ]
)) #訓練データ
tedata <- list(y=y[(trnum+2):(249-lags+1), index], x=as.matrix(stackmat[(trnum+1):
(248-lags+1), ])) #テストデータ

library("glmnet")
# glmnetでL1正則化ロジスティック回帰, クロスバリデーション
glmtop <- cv.glmnet(x = trdata$x, y = trdata$y, nfolds=10, lambda = exp(seq(-7, -
2, length.out=40)))
plot(glmtop)
(p<-predict(glmtop, newx = tedata$x, s="lambda.min")) #株価が上昇する確率の予測値

ypred <- ifelse(p > 0.5, 1, 0)
mean(tedata$y == ypred)

# どの係数が効いているか?
predict(glmtop, s="lambda.min", type="coefficient")

# random guess
mean(tedata$y == 1)
mean(tedata$y == 0)
# 予測値はランダムな解答よりはましな正答率

# 確率の予測値と実際の上下
plot(tedata$y, type="l")
lines(p, col=2)
abline(0.5, 0)

```