

クレジット:

UTokyo Online Education 数理手法Ⅶ 2019 北川源四郎

ライセンス:

利用者は、本講義資料を、教育的な目的に限ってページ単位で利用することができます。特に記載のない限り、本講義資料はページ単位でクリエイティブ・コモンズ 表示-非営利-改変禁止 ライセンスの下に提供されています。

<http://creativecommons.org/licenses/by-nc-nd/4.0/>

本講義資料内には、東京大学が第三者より許諾を得て利用している画像等や、各種ライセンスによって提供されている画像等が含まれています。個々の画像等を本講義資料から切り離して利用することはできません。個々の画像等の利用については、それぞれの権利者の定めるところに従ってください。



R_code(2)_2019.txt

```
#####  
# 第2章 共分散関数  
#####  
  
### データの読み込み  
#  
hakusan <- as.ts(read.csv("hakusan_new.csv"))  
hakusan1 <- hakusan[,1]  
sunspot <- as.ts(read.csv("sunspot_new.csv"))  
maxtemp <- as.ts(read.csv("maxtemp.csv"))  
blsfood <- as.ts(read.csv("blsfood_new.csv"))  
whard <- as.ts(read.csv("whard_new.csv"))  
mye1f <- as.ts(read.csv("mye1f_new.csv"))  
  
# 図の余白を小さくする  
par(mar=c(2,2,1,1)+0.1)  
  
# 方向角データと横揺れデータ  
plot(hakusan[,c(1,3)],main="")  
  
#####  
# ヒストグラム  
hist(hakusan[,1],main="方向角速度",col="#539952")  
hist(hakusan[,3],main="横揺れ",breaks=seq(-6,12,1.2), col="#edae00")  
hist(hakusan[,4],main="縦揺れ",breaks=seq(-20,20,3))  
hist(hakusan[,7],main="舵角",breaks=seq(-15,5,1.5), col="#edae00")  
# 散布図  
y <- hakusan[,1]  
z <- hakusan[,3]  
plot(y,y2,pch=19,col="#539952",ylab="y(n-2)",xlab="y(n)")  
plot(y,lag(y,k=4),pch=19,col="#539952",ylab="y(n-4)",xlab="y(n)")  
plot(z,lag(z,k=2),pch=19,col="#539952",ylab="y(n-2)",xlab="y(n)")  
plot(z,lag(z,k=4),pch=19,col="#539952",ylab="y(n-4)",xlab="y(n)")  
  
#####  
# 散布図と相関係数  
x <- hakusan[,3]  
plot(x,x)  
plot(x,lag(x,k=1))  
plot(x,lag(x,k=2))
```

R_code(2)_2019.txt

```
# 相関係数
autcor <- acf(hakusan1)
autcor
#####
# 白色雑音の生成
plot(as.ts(rnorm(200)))

#自己相関関数
unicor(hakusan1) # TSSSパッケージの場合
unicor(subspot)
unicor(maxtemp)
unicor(blsfood)
unicor(whard)
unicor(mye1f)

acf(hakusan1) # 汎用関数でも計算できる
acf(sunspot)
acf(maxtemp,lag.max=50)

#####
# 多変量時系列
#####
# データ プロット
# 榛原データ 地下水位・気圧
haibara <- as.ts(read.csv("haibara_new.csv"))
plot(haibara[,c(2,4)])
# 船舶データ (横揺れ, 縦揺れ, 舵角)
hakusan <- as.ts(read.csv("hakusan_new.csv"))
plot(hakusan[,c(3,4,7)])

#####
# 多変量時系列の散布図
pairs(hakusan[,c(3,4,7)],col="blue")
# ヒストグラム
hist(hakusan[,3],breaks=seq(-6,12,1.2),ylim=c(0,200),col="#539952")
hist(hakusan[,4],breaks=seq(-20,20,2.4),ylim=c(0,200),col="#539952")
hist(hakusan[,7],breaks=seq(-16,8,1.6),ylim=c(0,200),col="#539952")
# 相関行列
cor(hakusan[,c(3,4,7)])
```

R_code(2)_2019.txt

```
#Package(TSSS)
crscor(hakusan[,c(3,4,7)]) # 船舶データ (横揺れ, 縦揺れ, 舵角)
crscor(hakusan[,c(1,3,4,7)]) # 方向角を加えた4変量

# 榛原地下水位データ
haibara <- read.csv("haibara_new.csv")
haibara2 <- as.ts(haibara[,c(2,4)])
crscor(haibara2)

haibara <- read.csv("haibara_new.csv") # 地下水位データ
haibara2 <- as.ts(haibara[,c(2,4)]) # 地下水位と気圧
crscor(haibara2)

#####
# 多変量正規乱数 MASSパッケージ mvrnorm
#####
# 独立ノイズ r=0
Sigma <- matrix(c(1,0,0,1),2,2)
Sigma
x <- mvrnorm(n = 1000, rep(0, 2), Sigma)
plot(x, pch=19)

# 正の相関 r=0.6
Sigma <- matrix(c(1,-0.6,-0.6,1),2,2)
Sigma
x <- mvrnorm(n = 1000, rep(0, 2), Sigma)
plot(x, pch=19)

#####
# AR process の生成
Sigma <- matrix(c(1,0,0,0,1,0,0,0,1),3,3)
Sigma

n <- 100
x <- mvrnorm(n = 100, rep(0, 3), Sigma)
y <- x
#x <- as.ts(x)
#plot(x)
```

R_code(2)_2019.txt

```
for(j in 1:3){
for(i in 3:n){
y[i,j] <- x[i,j] + 1.8*y[i-1,j] - 0.95*y[i-2,j]
}
}
y <- as.ts(y)
```

```
#####
# 第3章 スペクトルとピリオドグラム
#####
```

```
# periodogram
# hakusan data 船舶データ (方向角速度)
x <- period(hakusan1>window=0)
plot(x$smoothed,ylim=c(-3,2),type="n")
par(new=T)
y <- x$smooth+3
plot(y,ylim=c(0,5),type="h")
```

```
# sunspot data 太陽黒点データ
x <- period(sunspot>window=0)
plot(x$smoothed,ylim=c(0,5),type="h")
```

```
# maxtemp data
x <- period(maxtemp>window=0)
plot(x$smoothed,ylim=c(0,4),type="h")
```

```
# blsfood
x <- period(blsfood>window=0)
plot(x$smoothed,ylim=c(0,5),type="h")
```

```
# whard
x <- period(whard>window=0)
plot(x$smoothed,ylim=c(0,7),type="h")
```

```
# mye1f
period(mye1f)
```

```
#####
```

R_code(2)_2019.txt

```
# FFT によるピリオドグラム
fftper(mye1f)
period(mye1f,plot=FALSE)

# 正規乱数の生成
r <- as.ts(rnorm(200))
plot(r)

#####
# AR過程の生成, 自己相関関数, ピリオドグラム

### AR(1) a=0.9 ###
t <- 2:300
r <- as.ts(rnorm(300))
x <- rep(9,300)
a <- 0.9
for (i in t) x[i] = a*x[i-1] + r[i]
plot(as.ts(x[101:300]))

acf(x[101:300]) # 自己相関関数
period(x[101:300]) # ピリオドグラム

### AR(1) a=-0.9 ###
t <- 2:300
r <- as.ts(rnorm(300))
x <- rep(9,300)
a <- -0.9
for (i in t) x[i] = a*x[i-1] + r[i]
plot(as.ts(x[101:300]))

acf(x[101:300]) # 自己相関関数
period(x[101:300]) # ピリオドグラム

### AR(2) ###
t <- 3:300
r <- as.ts(rnorm(300))
x <- rep(9,300)
a1 <- 0.9*sqrt(3)
a2 <- -0.81
for (i in t) x[i] = a1*x[i-1] + a2*x[i-2] + r[i]
```

R_code(2)_2019.txt

```
plot(as.ts(x[101:300]))
#
acf(x[101:300]) # 自己相関関数
period(x[101:300]) # ピリオドグラム

#####
### 正規白色雑音の生成(n=3200) ###
#####
r <- as.ts(rnorm(3200))
plot(r)
acf(r)
plot(r,main="acf")

# ピリオドグラム (データ数3200)
r <- rnorm(3200)
r <- as.ts(r)
spec.pgram(r,log="yes")

# ピリオドグラム (データ数800)
#spectrum
spectrum(r,ylim=c(0.01,100))
spectrum(r[1:800],ylim=c(0.01,100))

# ピリオドグラム (データ数200)
spec.pgram(r[1:200],log="yes")

f <- 0:113
f <- f/226
x <- as.ts( period(r[1:200]) )
plot(f,x,type="l")

## Raw Spectrum
# autocovariance function
lag <- 50
lag <- length(r)
lag1 <- lag-1
x <- as.ts(acf(r,type="covariance",lag.max=lag))
```

R_code(2)_2019.txt

```
c <- x$acf

# raw spectrum
p <- rep(1:lag)
f <- 0:lag
g <- 1:lag1
for (i in f) {
  fj <- i/(2*lag)
  s <- c[1]
  for (k in g) {
    s <- s + 2*c[k+1]*cos(2*pi*k*fj)
  }
  p[i+1] <- s
}
p <- as.ts(p)
lp <- log10(p)
plot(lp,ylim=c(-4,1))

#####
# simulation of 2 cosine function + noise
t <- 1:400
r <- rnorm(400)
y <- rep(0,400)
for (i in t) {
  y[i] <- cos(2*pi*i/10) + cos(2*pi*i/4) + r[i]*0.1
}
y <- as.ts(y)
plot(y)

period(y>window=0) # ピリオドグラム

fftper(y>window=0) # FFT
```