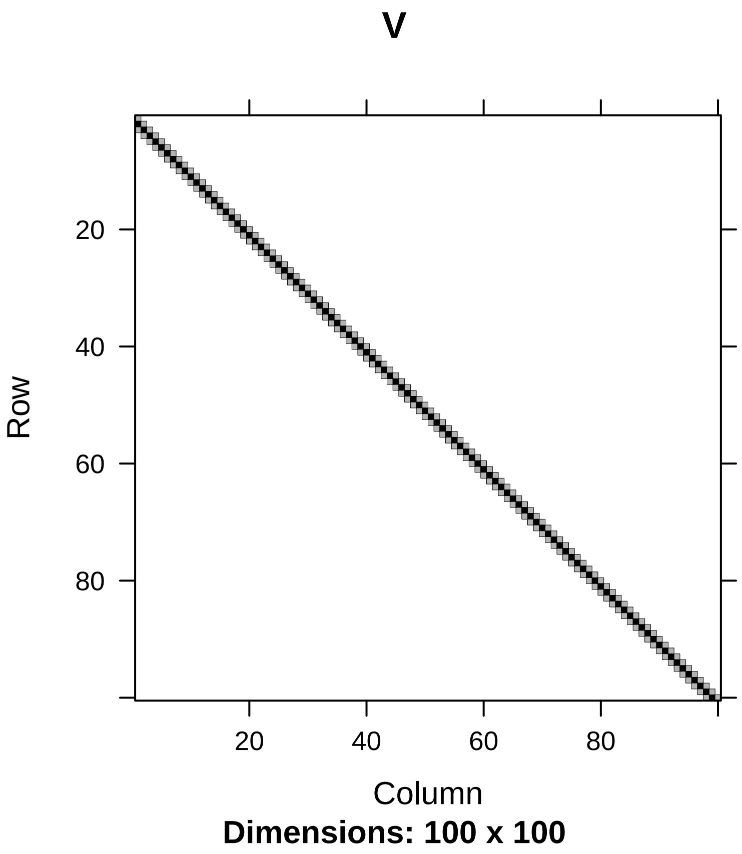
Covariance Model for Teleconnection

```
source('../utils_cov.R')
```

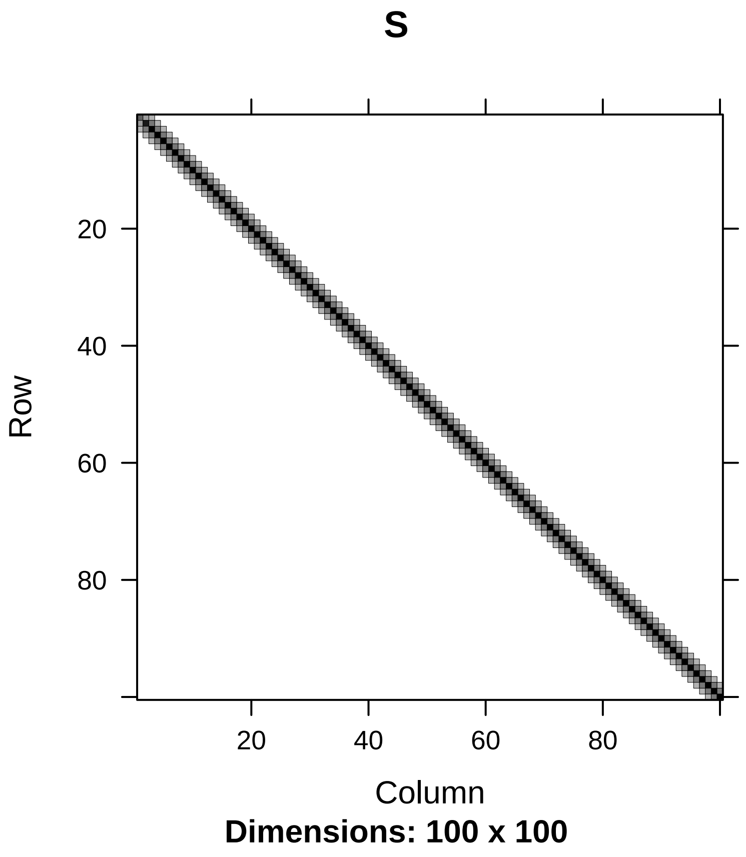
```
## Warning: package 'pROC' was built under R version 3.3.3

## Type 'citation("pROC")' for a citation.

##
## Attaching package: 'pROC'

## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

1. NO teleconnection case

```
# True model (NO teleconnection)
p=100; r=0.5
V=genV(p=p,r=0.5,tele=F) # only neighbour effect
S=V%*%t(V)+diag(rep(1,p)) # true cov, generated from factor model S=VV'+I
printM(V)
```



```r
printM(S)
```

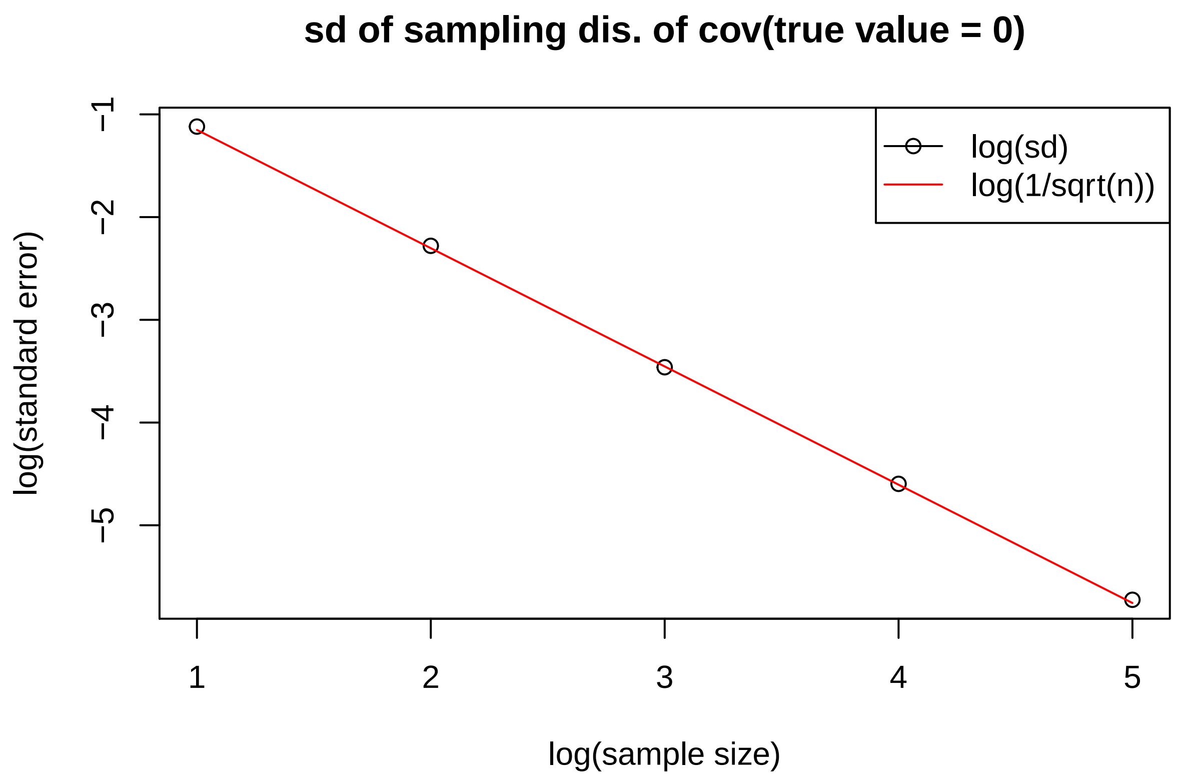


1.0 sampling dis. of cov of two (univariate) ind. normal

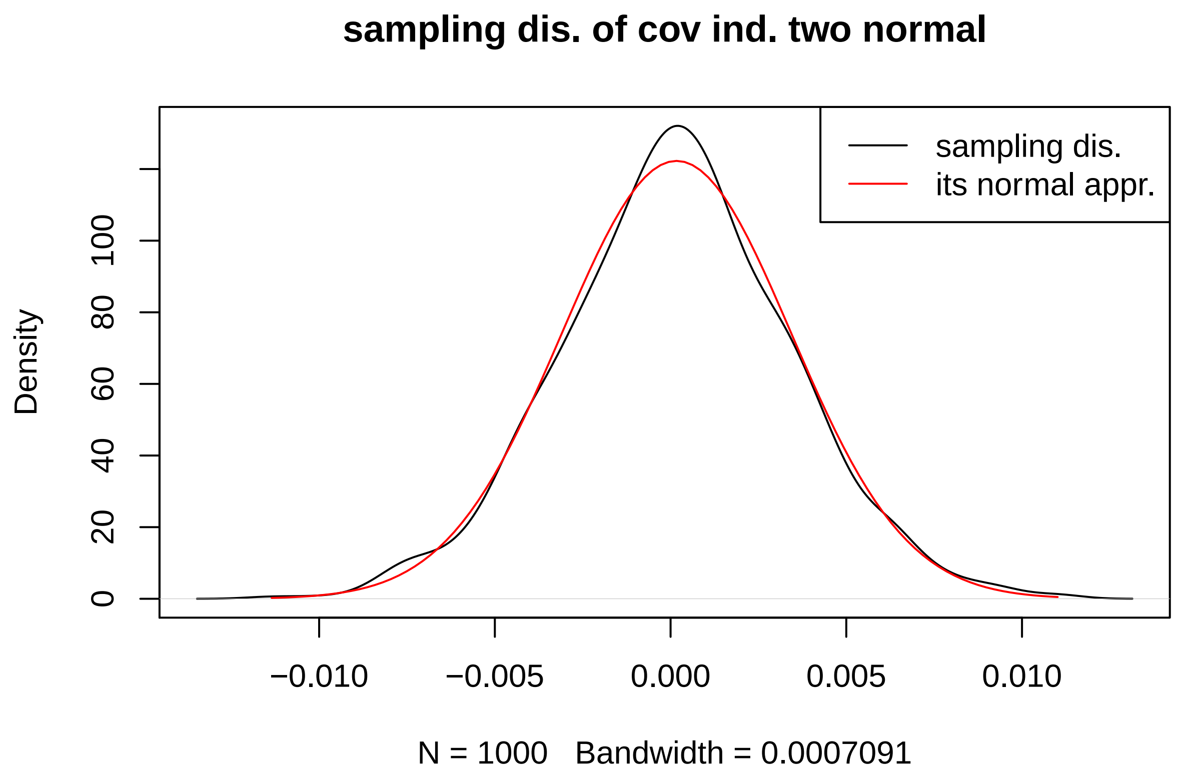
Given X, and Y are two independent normal (sample data with sample size 10^N2), what is the distribution of cov(X,Y)? (of course, the true value cov(X,Y)=0). This part simply repeat it N times, and find the sampling distribution of cov(X,Y). This is a preparation of the zero(true value) elements in the sample covariance matrix.

```r
N=1000 # repeat size
N2=5 # sample size: 10 to the power
res=rep(0,N) # each iteration, save the covariance. all repeatants forms a sample
res2=rep(0,N2) # each sample size, save sd

for(j in 1:N2){
  n=10^j # sample size
  for(i in 1:N){
    x=rnorm(n)
    y=rnorm(n)
    res[i]=cov(x,y)
  }
  res2[j]=sd(res) # standard error of the "sample covariance"
}

# sample size vs sd (of sampling dis.) and 1/sqrt(n)
```

```r
plot(1:N2,log(res2),main="sd of sampling dis. of cov(true value = 0)",
     xlab="log(sample size)",ylab="log(standard error)")
lines(1:N2,log(1/sqrt(10^(1:N2))),type="l",col="red")
legend("topright",c("log(sd)","log(1/sqrt(n))"),lty=c(1,1),pch = c(1,NA),col=c("black","red"))
```

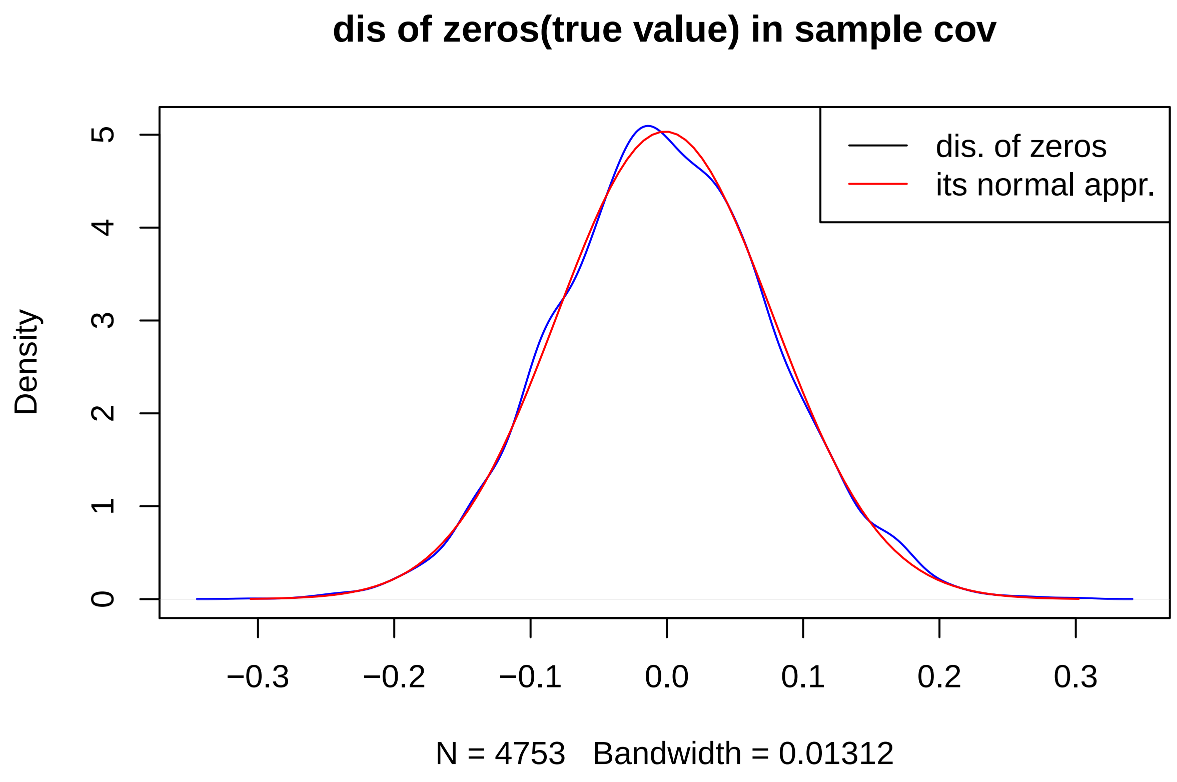


```r
# sampling distribution vs its normal version(same mean & sd)
plot(density(res),main="sampling dis. of cov ind. two normal")
x=seq(min(res),max(res),length=100)
y=dnorm(x,mean=mean(res), sd=sd(res))
lines(x,y, type="l", lwd=1,col="red")
legend("topright",c("sampling dis.","its normal appr."),lty=c(1,1),col=c("black","red"))
```



1.1 dis of zero/non-0s in sample cov

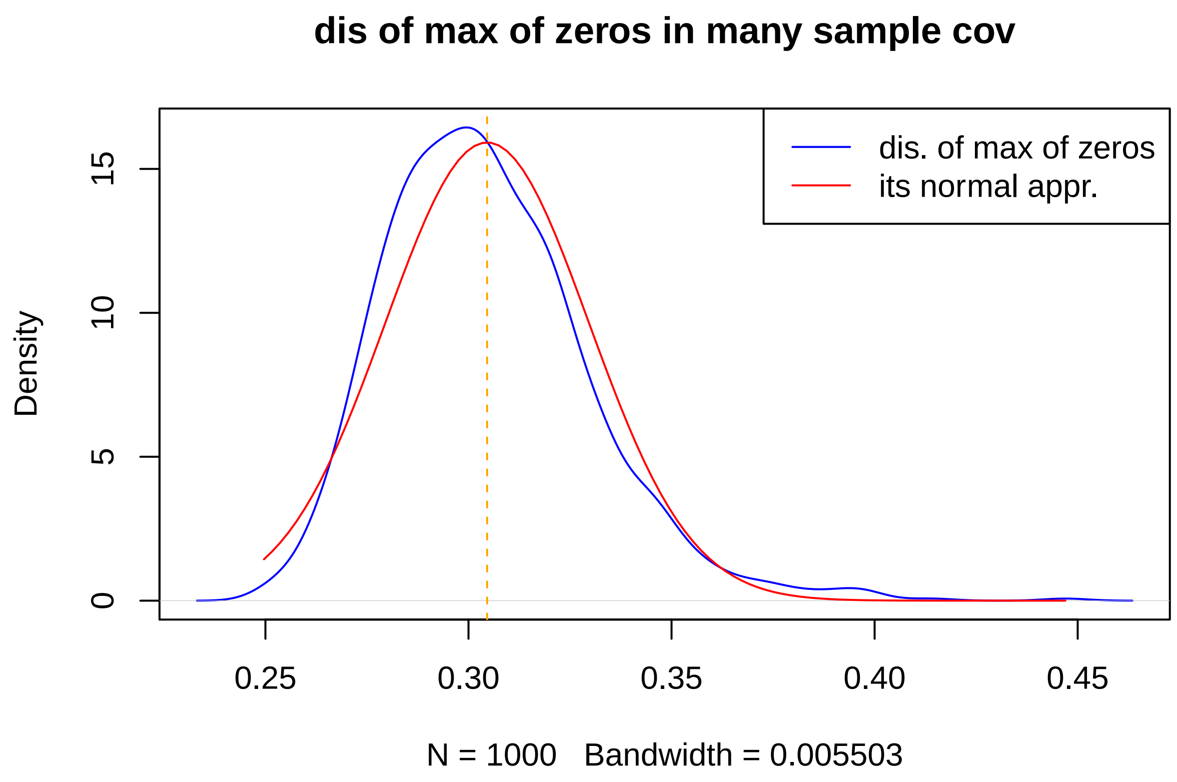
```r
# (for one fixed sample cov) zero elements(off-diagonals) are the sample data
n=1000
data = mvrnorm(n, mu = rep(0,p), Sigma = S)
Shat=cov(data)

res=NULL
for(i in 1:p){
  for(j in 1:p){
    if((i-j)>2){
      res=c(res,Shat[i,j])
    }
  }
}

# sampling distribution vs its normal version(same mean & sd)
plot(density(res),col="blue",main="dis of zeros(true value) in sample cov")
x=seq(min(res),max(res),length=100)
y=dnorm(x,mean=mean(res), sd=sd(res))
lines(x,y, type="l", lwd=1,col="red")
legend("topright",c("dis. of zeros","its normal appr."),lty=c(1,1),col=c("black","red"))
```



1.2 dis of max zero/non-0 in many sample cov

```
N=1000
res2=rep(0,N)
for(k in 1:N){
  data = mvrnorm(n, mu = rep(0,p), Sigma = S)
  Shat=cov(data)
  # for(i in 1:p){
  #   Shat[i,i]=0
  # }
  # for(i in 1:(p-1)){
  #   Shat[i,i+1]=Shat[i+1,i]=0
  # }
  # for(i in 1:(p-2)){
  #   Shat[i,i+2]=Shat[i+2,i]=0
  # }
  res2[k]=max(abs(Shat*(S==0)))
}

plot(density(res2),col="blue",main="dis of max of zeros in many sample cov")
x=seq(min(res2),max(res2),length=100)
y=dnorm(x,mean=mean(res2), sd=sd(res2))
lines(x,y, type="l", lwd=1,col="red")
abline(v=mean(res2),lty=2,col="orange")
```

```r
legend("topright",c("dis. of max of zeros","its normal appr."),lty=c(1,1),col=c("blue","red"))
```



```r
summary(res2)
```

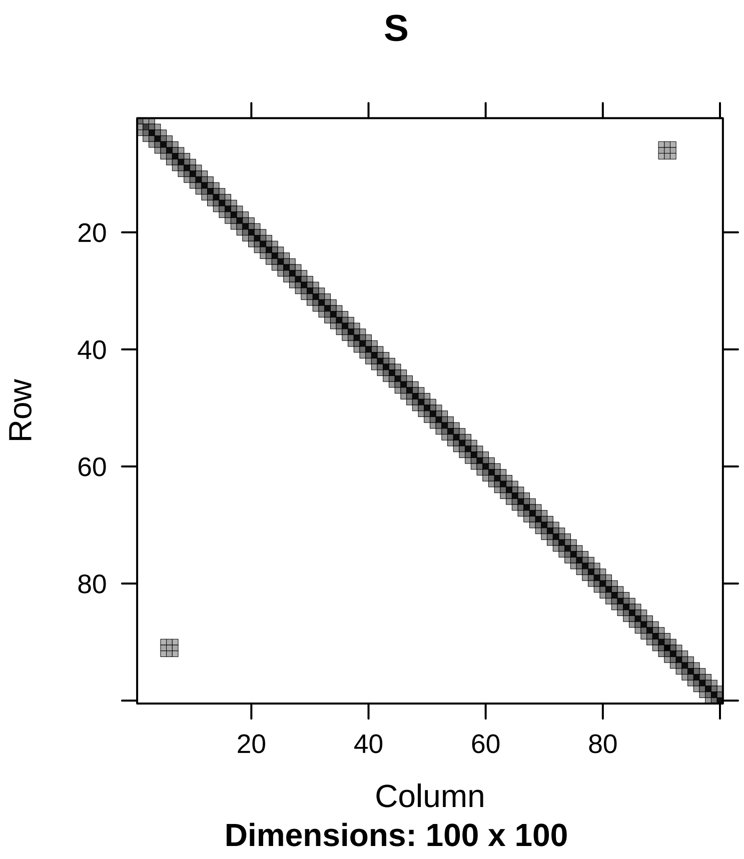
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
## 0.2496  0.2865  0.3022  0.3046  0.3191  0.4470
```

```r
quantile(res2,0.99)
```

```
##     99% 
## 0.38277
```

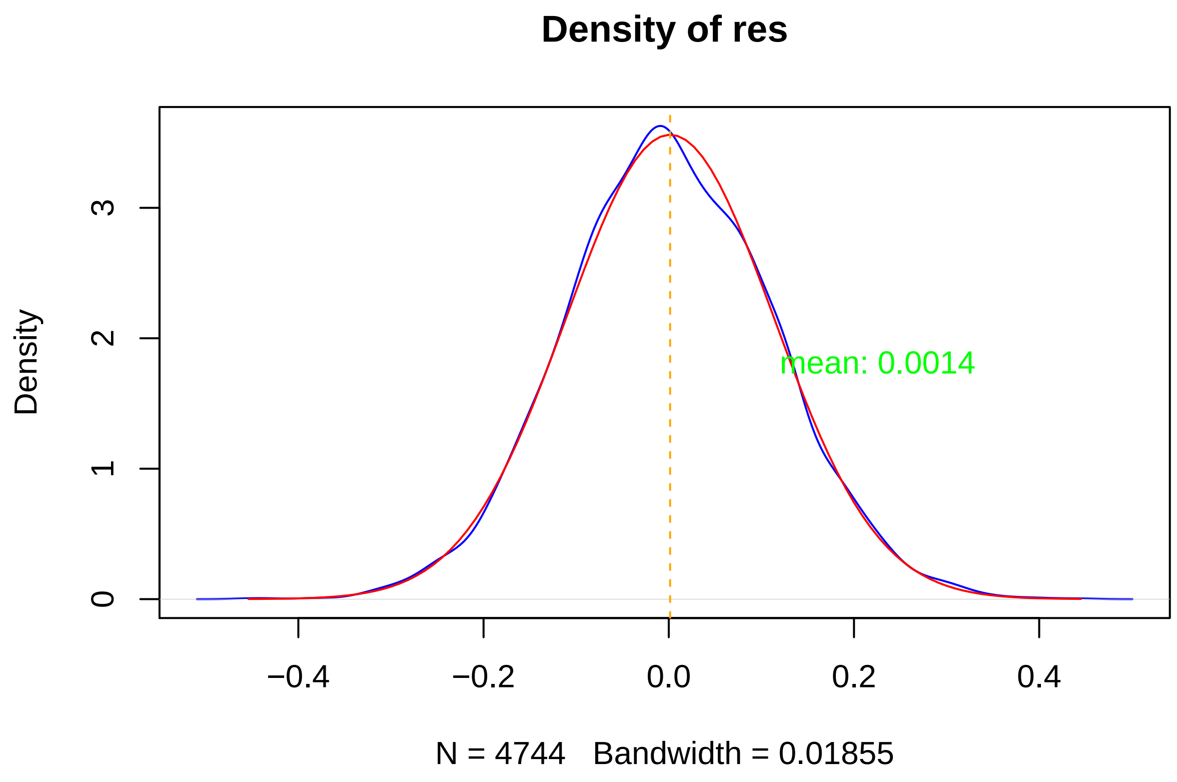
2. WITH teleconnection case

```r
# True model (WITH teleconnection)
p=100; n=1000; r=0.9; s=0.5; tele=1; sigma=1
Ip=diag(rep(1,p))
V=genV(p=p,r=r,s=s,tele=tele)
S=V%*%t(V)+diag(rep(1,p))
M=(S==0)
printM(S)
```



2.1 one sample cov, distr of zero elements

```
n=1000
data = mvrnorm(n, mu = rep(0,p), Sigma = S)
Shat=cov(data)
# take zero elements in true cov
res=NULL

for(i in 1:p){
  for(j in 1:p){
    if(M[i,j] & (i-j>2)){
      res=c(res,Shat[i,j])
    }
  }
}

# result: dis. of zeros like normal
plotDen(res)
```



2.2 multi sample cov, distr of max elements

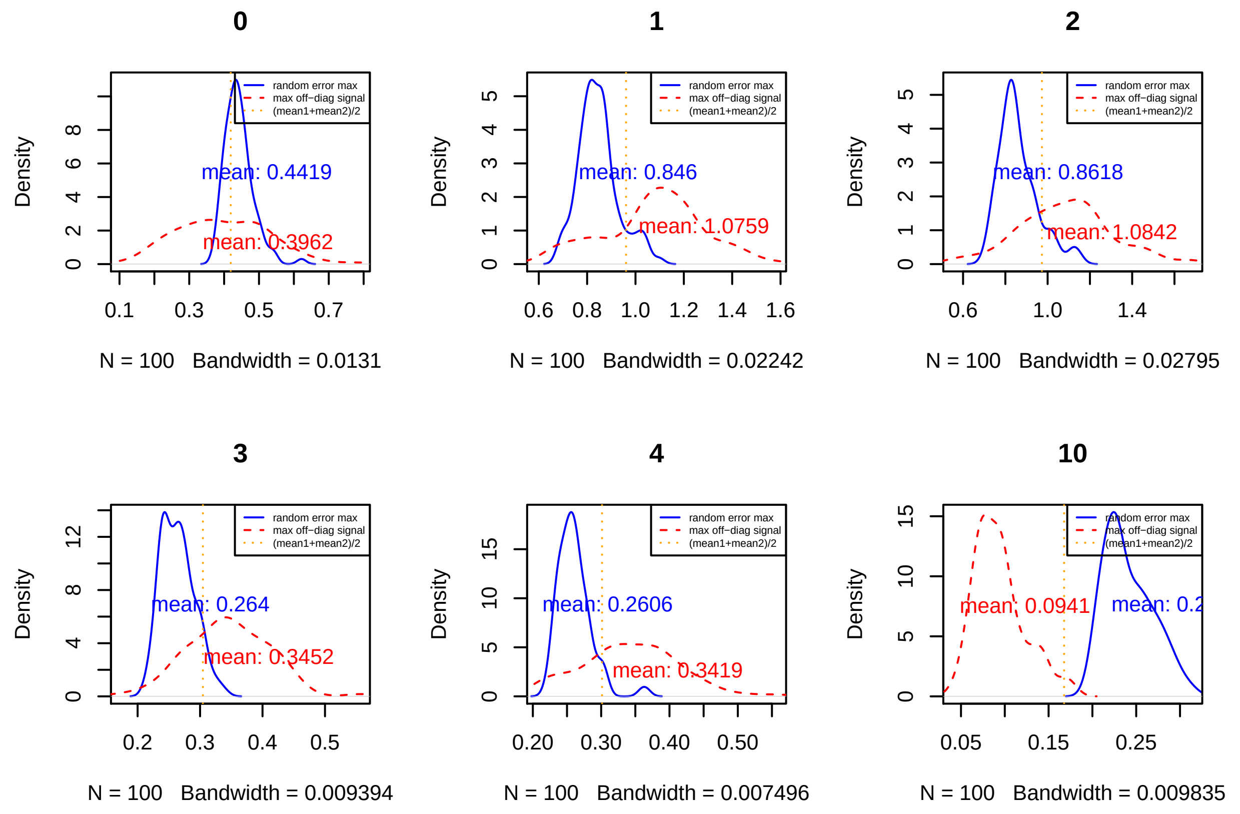
compare the distribution of:

- max of zeros in true cov (Well, I exclude the margin 2<|i-j|<5, this part maybe large in the 3 by 3 block version)
- max of teleconnection signal (max off-diagonal non-zero center)

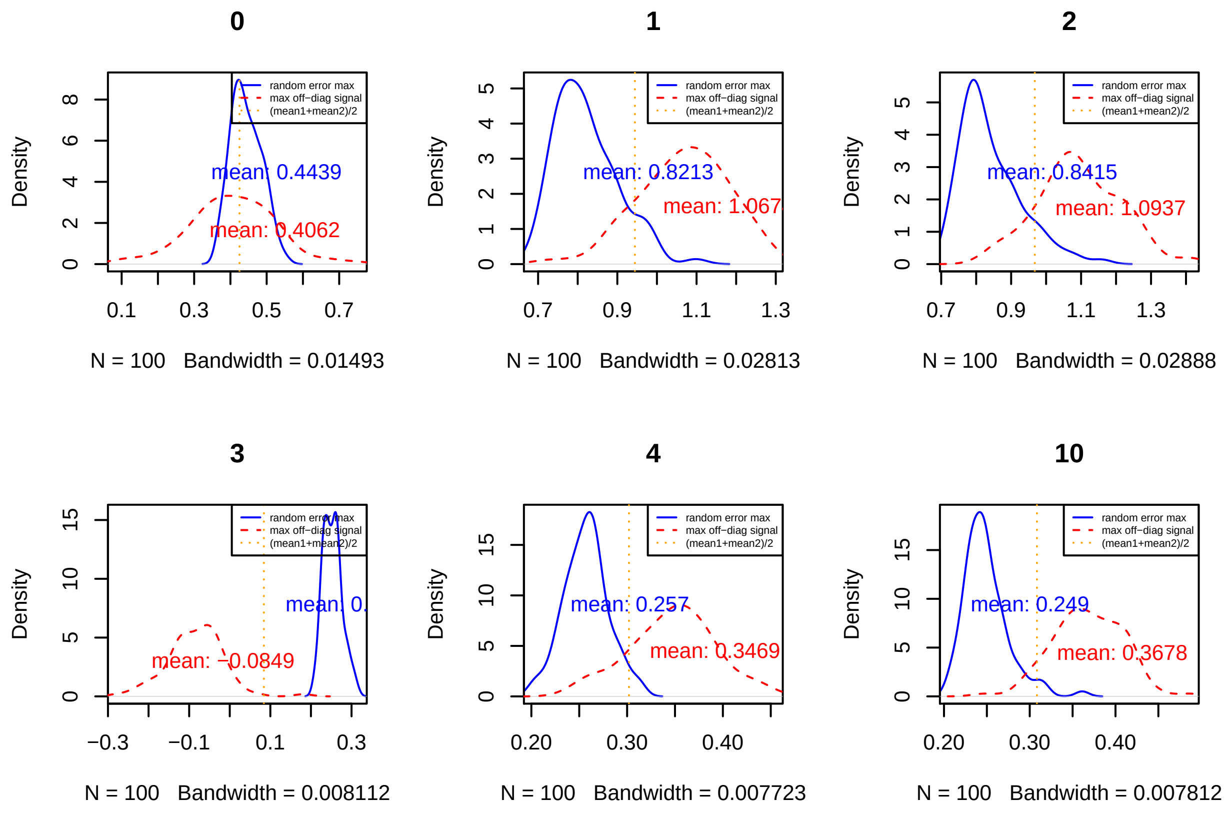
type: 0: sample cov; 1: spectual norm; 2: Frobenius norm; 3: mean; 4: abs(mean); 5: prod; 6: 2nd largest; 8: abs(prod(m0)); 10: sd; 11: skewness; 12: kurtosis;

r: rho; s: scale of the teleconnection s*(rho,1,rho);

```
# rho=1
p=100; n=1000; r=0.9; s=0.4; tele=1; sigma=1

par(mfrow=c(2,3))
for(i in c(0,1,2,3,4,10)){
  maxnoise(p,type=i, n=1000, N=100,r=r,s=s,tele=tele, sigma=sigma)
}
```



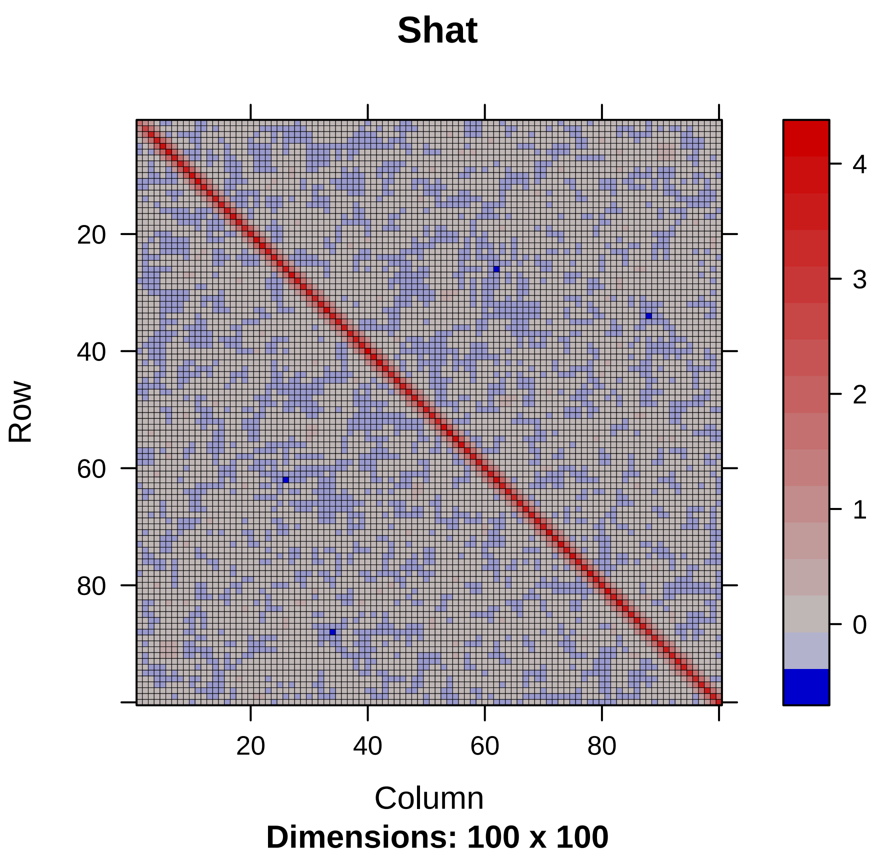
```r
# rho=-1
p=100; n=1000; r=0.9; s=0.4; tele=-1; sigma=1

par(mfrow=c(2,3))
for(i in c(0,1,2,3,4,10)){
  maxnoise(p,type=i, n=1000, N=100,r=r,s=s,tele=tele, sigma=sigma)
}
```



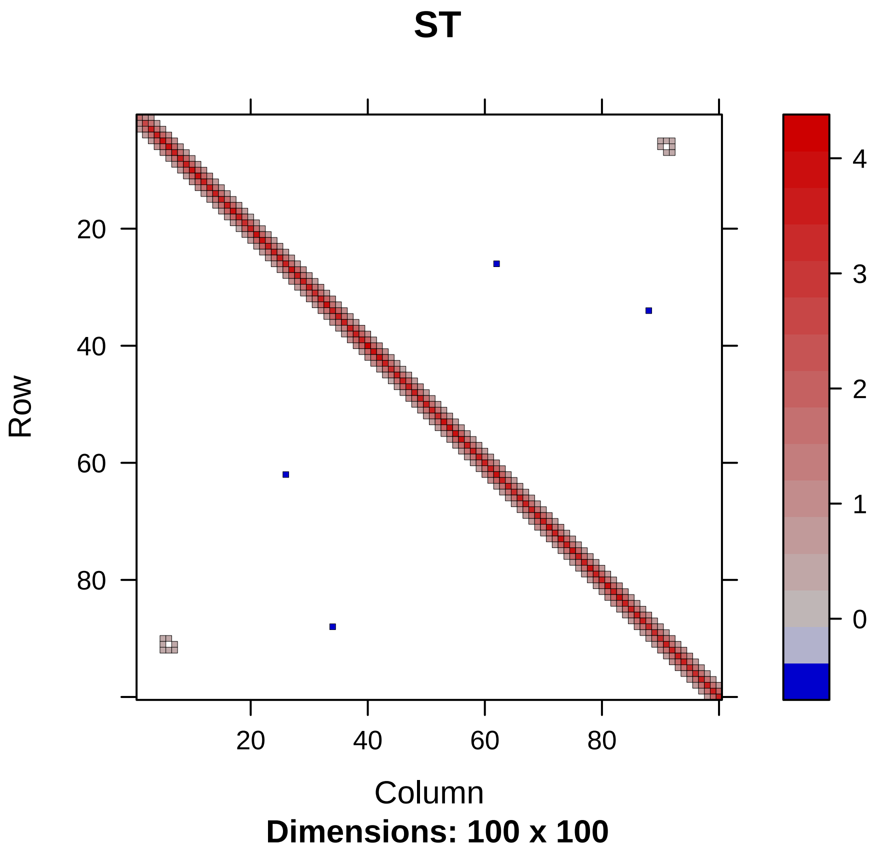
2.3 covariance estimation

```r
# rho=1
p=100; n=1000; r=0.9; s=0.4; tele=1; sigma=1
Ip=diag(rep(1,p))
V=genV(p=p,r=r,s=s,tele=tele)
S=V%*%t(V)+diag(rep(1,p))

# Step I: Generate data
# method 1: generate data directly with sigma
data = mvrnorm(n, mu = rep(0,p), Sigma = S)
# method 2: generate according to factor model
V=genV(p=p,r=r,s=s,tele=tele)
f=mvrnorm(n, mu = rep(0,p), Sigma = Ip)
E=mvrnorm(n, mu = rep(0,p), Sigma = Ip)
data = V%*%t(f)+sigma*t(E) #mvrnorm(n, mu = rep(0,p), Sigma = S)
data=t(data)

# Step II: Calculate Different Covariance Estimator
# NOTE:  each one has a threshould to tune based on different model

# TYPE 1: sample covariance
Shat=cov(data)
# TYPE 2: (universal) threshoulding sample covariance
```

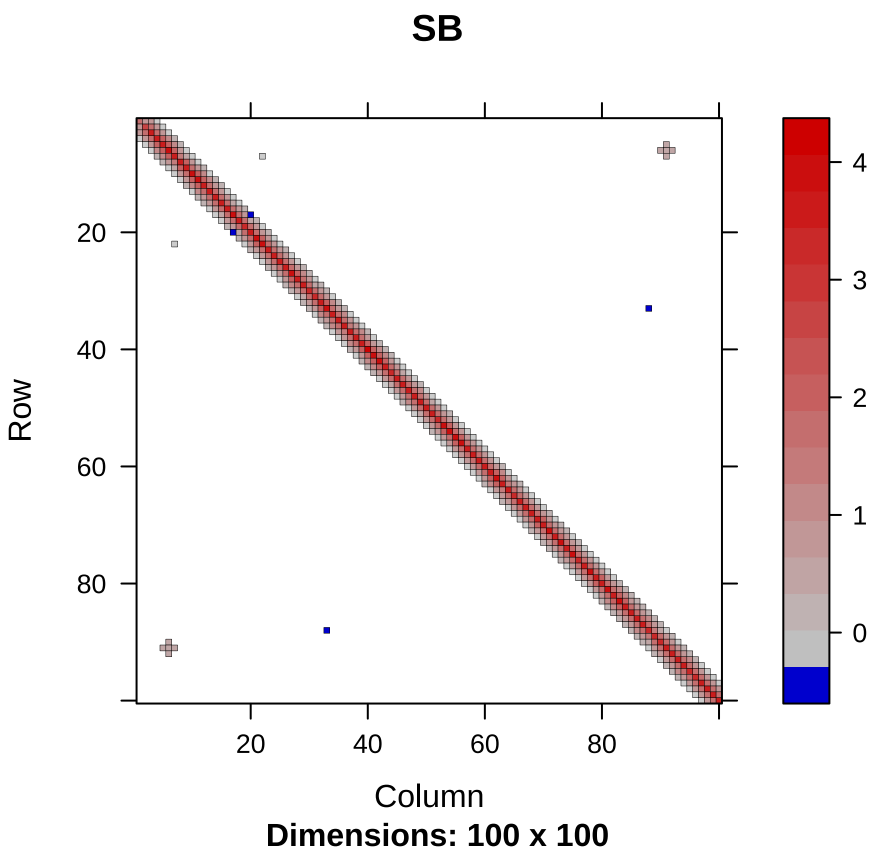
```
ST=Shat*(abs(Shat)>0.38)
# note: large sample size n=1e5, thre=0.1, frob_thre=0.2, could find all, better than thresoulding (no
# TYPE 3: use (3 by 3 block) variance threshoulding
Snorm=MnormM(Shat,h=2,type=4,RM=F)
B1=(Snorm>0.38)
B2=matrix(0,p,p)
for(i in 1:p){
  for(j in 1:p){
    if(abs(i-j)<=10){
      B1[i,j]=0
    }
    if(abs(i-j)<=2){
      B2[i,j]=1
    }
  }
}
# SB=Shat*(B1 | B2) # B1: teleconnection; B2: banding, cut nearest-neighbours (too good)
SB=Shat*(Snorm>0.26)
# TYPE 4: Tony Cai's Adaptive threshould
SAT=MnormMvar(data) # Calculate variance of covariance elements
SA=regAT(Shat,sqrt(SAT),n,const=1.47)

printM(Shat)
```



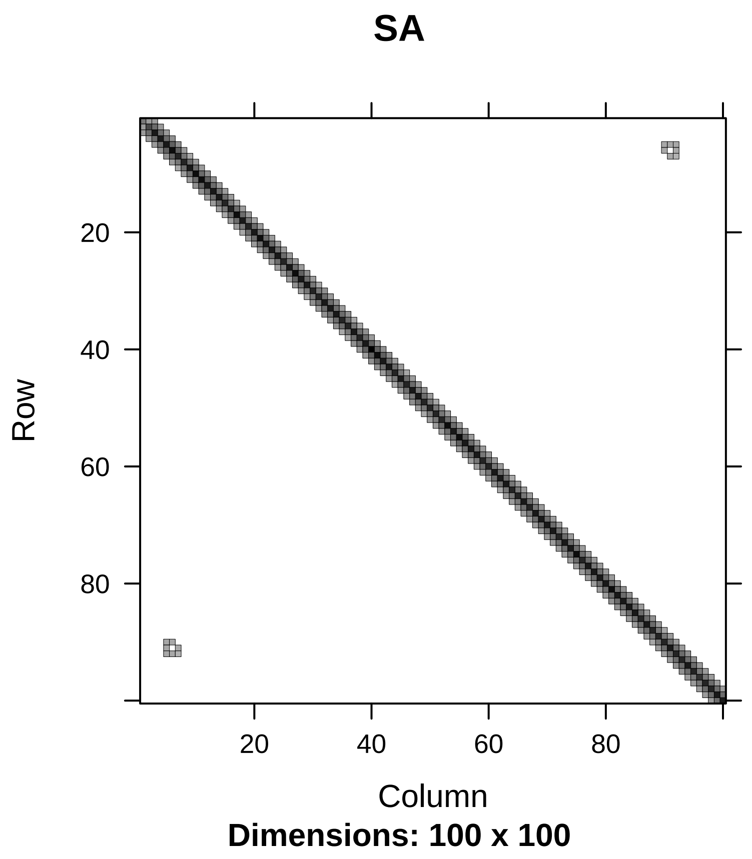
```r
printM(ST)
```



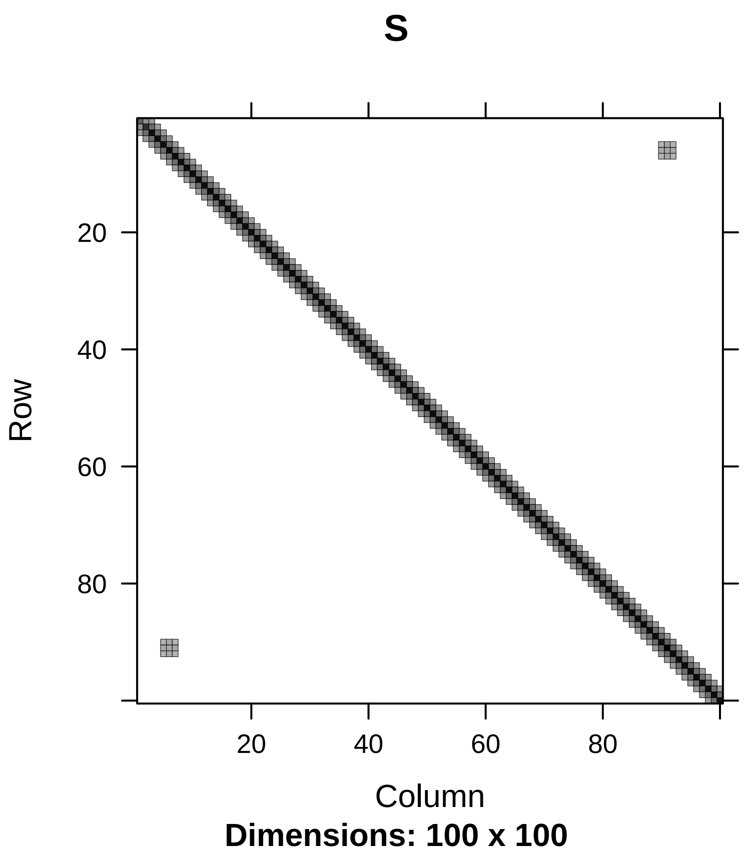
```r
printM(SB)
```



```r
printM(SA)
```



```r
printM(S)
```



```
base::norm(S-Shat,"F")
```

```
## [1] 11.31562
```

```
base::norm(S-ST,"F")
```

```
## [1] 3.134372
```

```
base::norm(S-SB,"F")
```

```
## [1] 3.474059
```

```
base::norm(S-SA,"F")
```

```
## [1] 3.035231
```

```
round(S[4:8,89:93],2)
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    0 0.00 0.00 0.00    0
## [2,]    0 0.32 0.36 0.32    0
## [3,]    0 0.36 0.40 0.36    0
## [4,]    0 0.32 0.36 0.32    0
## [5,]    0 0.00 0.00 0.00    0
```

```
round(Shat[4:8,89:93],2)
```

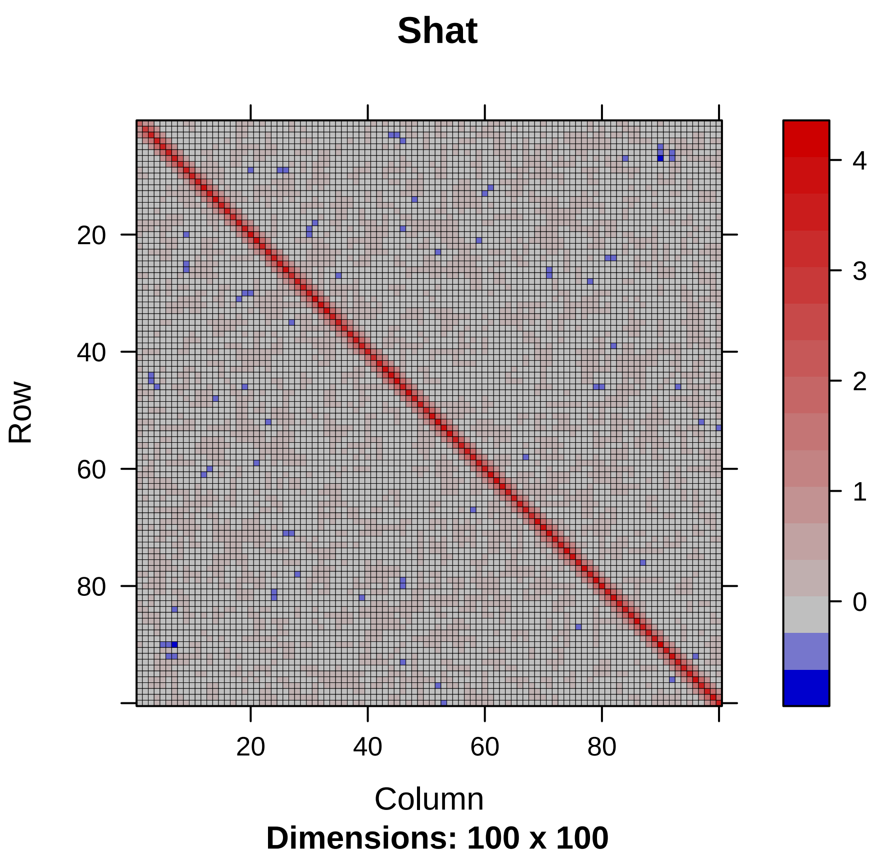
```
##       [,1]  [,2] [,3] [,4]  [,5]
## [1,]  0.14  0.06 0.08 0.05 -0.01
## [2,]  0.14  0.48 0.49 0.49 -0.02
## [3,]  0.15  0.41 0.36 0.43  0.16
## [4,]  0.16  0.28 0.48 0.40  0.18
## [5,] -0.04 -0.06 0.00 0.03  0.20
```

```
round(Snorm[4:8,89:93],2)
```

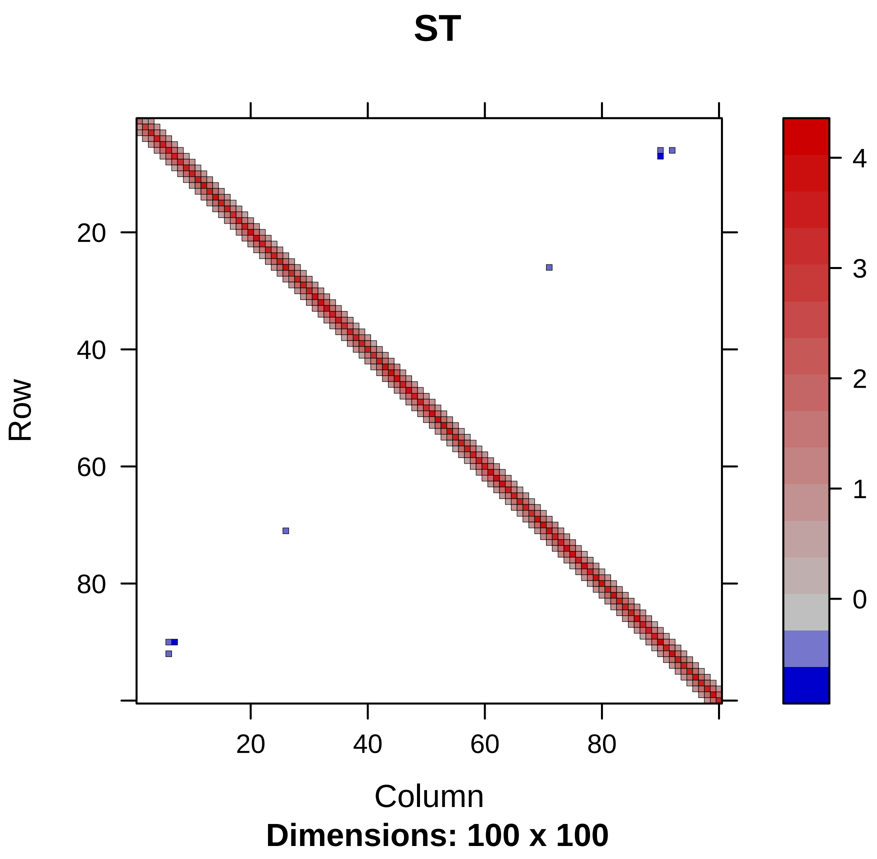
```
##      [,1] [,2] [,3] [,4] [,5]
## [1,] 0.12 0.19 0.21 0.16 0.10
## [2,] 0.17 0.26 0.32 0.23 0.15
## [3,] 0.20 0.33 0.42 0.33 0.20
## [4,] 0.14 0.22 0.27 0.25 0.16
## [5,] 0.10 0.16 0.19 0.18 0.11
```

```
# rho=-1
p=100; n=1000; r=0.9; s=0.4; tele=-1; sigma=1
Ip=diag(rep(1,p))
V=genV(p=p,r=r,s=s,tele=tele)
S=V%*%t(V)+diag(rep(1,p))

# Step I: Generate data
# method 1: generate data directly with sigma
data = mvrnorm(n, mu = rep(0,p), Sigma = S)
# method 2: generate according to factor model
V=genV(p=p,r=r,s=s,tele=tele)
f=mvrnorm(n, mu = rep(0,p), Sigma = Ip)
E=mvrnorm(n, mu = rep(0,p), Sigma = Ip)
data = V%*%t(f)+sigma*t(E) #mvrnorm(n, mu = rep(0,p), Sigma = S)
data=t(data)

# Step II: Calculate Different Covariance Estimator
# NOTE:  each one has a threshould to tune based on different model

# TYPE 1: sample covariance
Shat=cov(data)
# TYPE 2: (universal) threshoulding sample covariance
ST=Shat*(abs(Shat)>0.38)
# note: large sample size n=1e5, thre=0.1, frob_thre=0.2, could find all, better than threshoulding (no
# TYPE 3: use (3 by 3 block) variance threshoulding
Snorm=MnormM(Shat,h=2,type=4,RM=T)
B1=(Snorm>0.35)
B2=matrix(0,p,p)
for(i in 1:p){
  for(j in 1:p){
    if(abs(i-j)<=10){
      B1[i,j]=0
    }
    if(abs(i-j)<=2){
      B2[i,j]=1
    }
  }
}
# SB=Shat*(B1 | B2) # B1: teleconnection; B2: banding, cut nearest-neighbours (too good)
SB=Shat*(Snorm>0.26)
# TYPE 4: Tony Cai's Adaptive threshould
SAT=MnormMvar(data) # Calculate variance of covariance elements
```

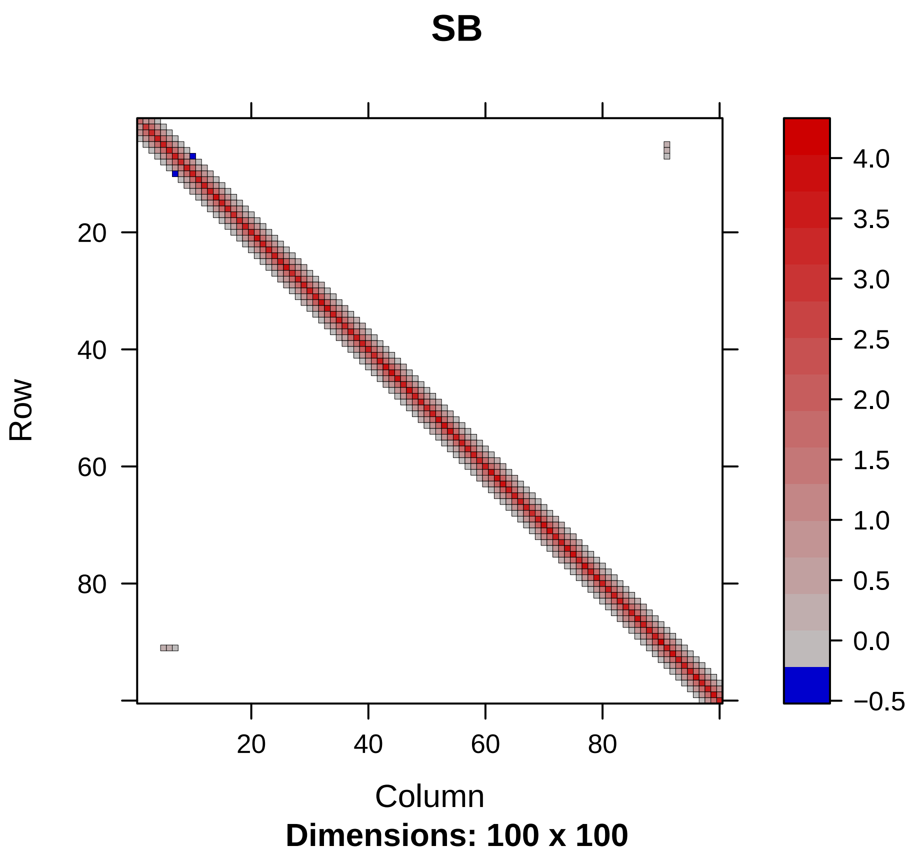
```
SA=regAT(Shat,sqrt(SAT),n,const=1.47)

printM(Shat)
```



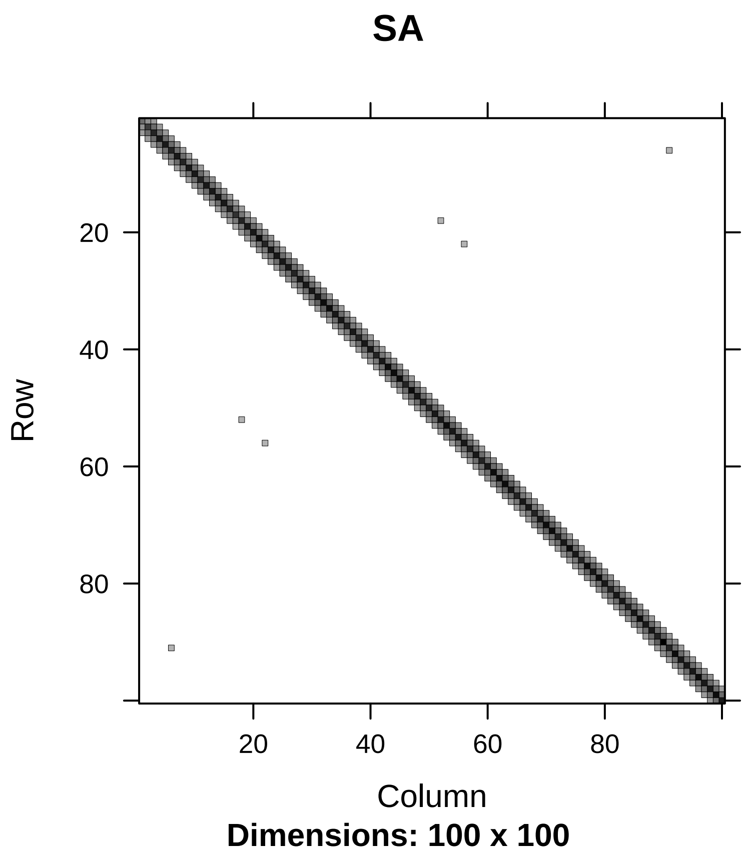
```
printM(ST)
```



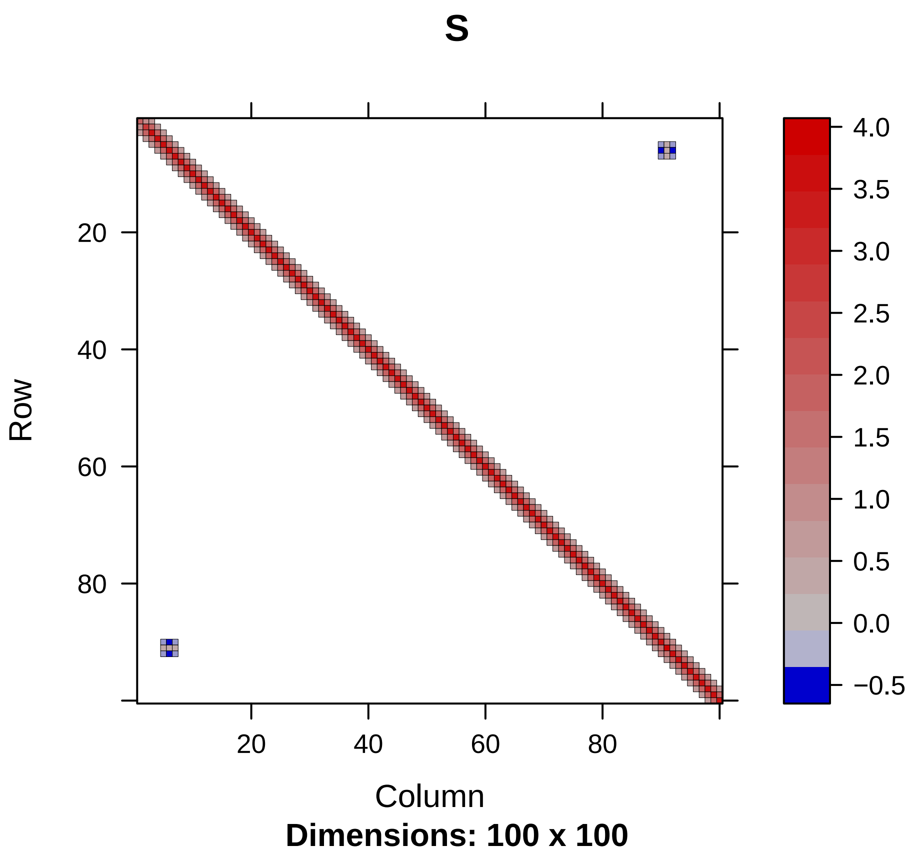
```
printM(SB)
```


SB
Row
Column
Dimensions: 100 x 100
20
40
60
80
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
−0.5

```
printM(SA)
```



```r
printM(S)
```



```
base::norm(S-Shat,"F")
```

```
## [1] 11.35312
```

```
base::norm(S-ST,"F")
```

```
## [1] 3.126592
```

```
base::norm(S-SB,"F")
```

```
## [1] 3.366839
```

```
base::norm(S-SA,"F")
```

```
## [1] 3.193212
```

```
round(S[4:8,89:93],2)
```

```
##      [,1]  [,2] [,3]  [,4] [,5]
## [1,]    0  0.00 0.00  0.00    0
## [2,]    0 -0.32 0.36 -0.32    0
## [3,]    0 -0.36 0.40 -0.36    0
## [4,]    0 -0.32 0.36 -0.32    0
## [5,]    0  0.00 0.00  0.00    0
```

```
round(Shat[4:8,89:93],2)
```

```
##       [,1]  [,2]  [,3]  [,4] [,5]
## [1,]  0.04  0.04  0.04  0.21 0.08
## [2,] -0.11 -0.37  0.34 -0.17 0.20
## [3,] -0.04 -0.42  0.36 -0.40 0.07
## [4,] -0.20 -0.62  0.02 -0.38 0.10
## [5,] -0.15 -0.01 -0.17  0.16 0.10
```

```
round(Snorm[4:8,89:93],2)
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,] 0.09 0.11 0.14 0.11 0.09
## [2,] 0.15 0.20 0.26 0.17 0.15
## [3,] 0.18 0.26 0.28 0.22 0.20
## [4,] 0.16 0.20 0.26 0.20 0.17
## [5,] 0.12 0.12 0.20 0.16 0.10
```