Spatio-temporal Data Analysis HW1

2020311198 Dongook Son

Spring 2020

Q1

(a)

Prove $E(Y) = E(\mu + LZ)$ and $V(Y) = V(\mu + LZ)$

Proof.

$$
\begin{aligned}
E(Y) &= E(\mu + LZ) \\
&= \mu + E(LZ) \\
&= \mu + LE(Z) && \text{(linearity property)} \\
&= \mu \\
V(Y) &= E([(Y - E(Y))(Y - E(Y))']) \\
&= E([(Y - \mu)(Y - \mu)']) \\
&= E([(LZ)(LZ)']) \\
&= E([LZZ'L']) \\
&= LE(ZZ')L' \\
&= LV(Z)L' \\
&= LIL' \\
&= LL' \\
&= \Sigma
\end{aligned}
$$

□

(b)

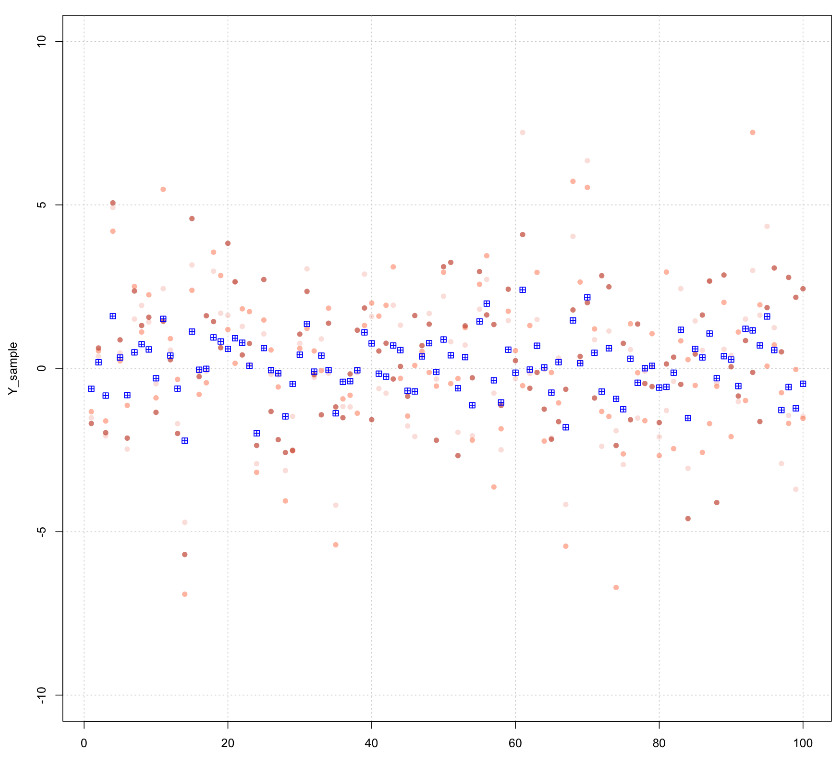
```
create_multivariate_normal <- function (mu, Sigma) {
lower_triangle_matrix <- t(chol(Sigma))
dimension <- nrow(Sigma)
Z <- rmvnorm(n = 1, mean = rep(0, dimension), sigma = diag(dimension))
```

```r
  Y = mu + lower_triangle_matrix %*% Z
  return(Y)
}
```

(c)

```r
library(mvtnorm)
library(clusterGeneration)
library(yarrr)
generate_z <- function(dimension, seed= 1) {
  set.seed(seed)
  Z <- rmvnorm(n = 1, mean = rep(0, dimension), sigma = diag(dimension))
  return(Z)
}

generate_multivariate_normal_with_Z <- function (mu, Sigma, Z, seed=1) {
  set.seed(seed)
  lower_triangle_matrix <- t(chol(Sigma))
  Y = mu + (lower_triangle_matrix %*% t(Z))
  return(Y)
}


dimension <- 100
Z_sample <- generate_z(dimension)

plot(x=1,
     type="n",
     xlim = c(1,100),
     ylim = c(-10,10),
     pch = 16,
     xlab="Sample Index",
     ylab="Y_sample",
     )
grid()

set.seed(1)
Sigma <- genPositiveDefMat(dimension, covMethod="eigen")$Sigma
Y_sample <- generate_multivariate_normal_with_Z(mu = rep(0,100), Sigma = Sigma, Z = Z_sample)
points(1:100, Y_sample,
       pch = 16,
       col = transparent("coral2", trans.val = .8))

set.seed(1)
```

```r
Sigma <- genPositiveDefMat(dimension, covMethod="onion")$Sigma
Y_sample <- generate_multivariate_normal_with_Z(mu = rep(0,100), Sigma =
    Sigma, Z = Z_sample)
points(1:100, Y_sample,
       pch = 16,
       col = transparent("coral", trans.val = .5))

set.seed(1)
Sigma <- genPositiveDefMat(dimension, covMethod="unifcorrmat")$Sigma
Y_sample <- generate_multivariate_normal_with_Z(mu = rep(0,100), Sigma =
    Sigma, Z = Z_sample)
points(1:100, Y_sample,
       pch = 16,
       col = transparent("coral3", trans.val = .3))

Sigma <- diag(dimension)
Y_sample <- generate_multivariate_normal_with_Z(mu = rep(0,100), Sigma =
    Sigma, Z = Z_sample)
points(1:100, Y_sample,
       pch = 12,
       col = transparent("blue", trans.val = .1))
```



The blue squares are samples from an identity matrix sigma and others are from correlated sigmas.

Q2

(a)

(b)

(c)

(d)