Esercitazione 3

October 9, 2024

1 Esercitazione 3

1.1 Simulazioni di variabili troncate

Assumete di avere

$$X \sim TN(\mu, \sigma^2, l, u)$$

dove $TN(\mu, \sigma^2, l, u)$ è una normale troncata, definita in $[l, u]$ con $l < u$, i.e., $X \in [l, u]$, con densità

$$f(x) = \frac{\phi(x|\mu, \sigma^2)}{\Phi(u|\mu, \sigma^2) - \Phi(l|\mu, \sigma^2)}$$

dove

$$\phi(x|\mu, \sigma^2)$$

è la densità di una normale di media μ e varianza σ^2 valutata in x, e

$$\Phi(y|\mu, \sigma^2)$$

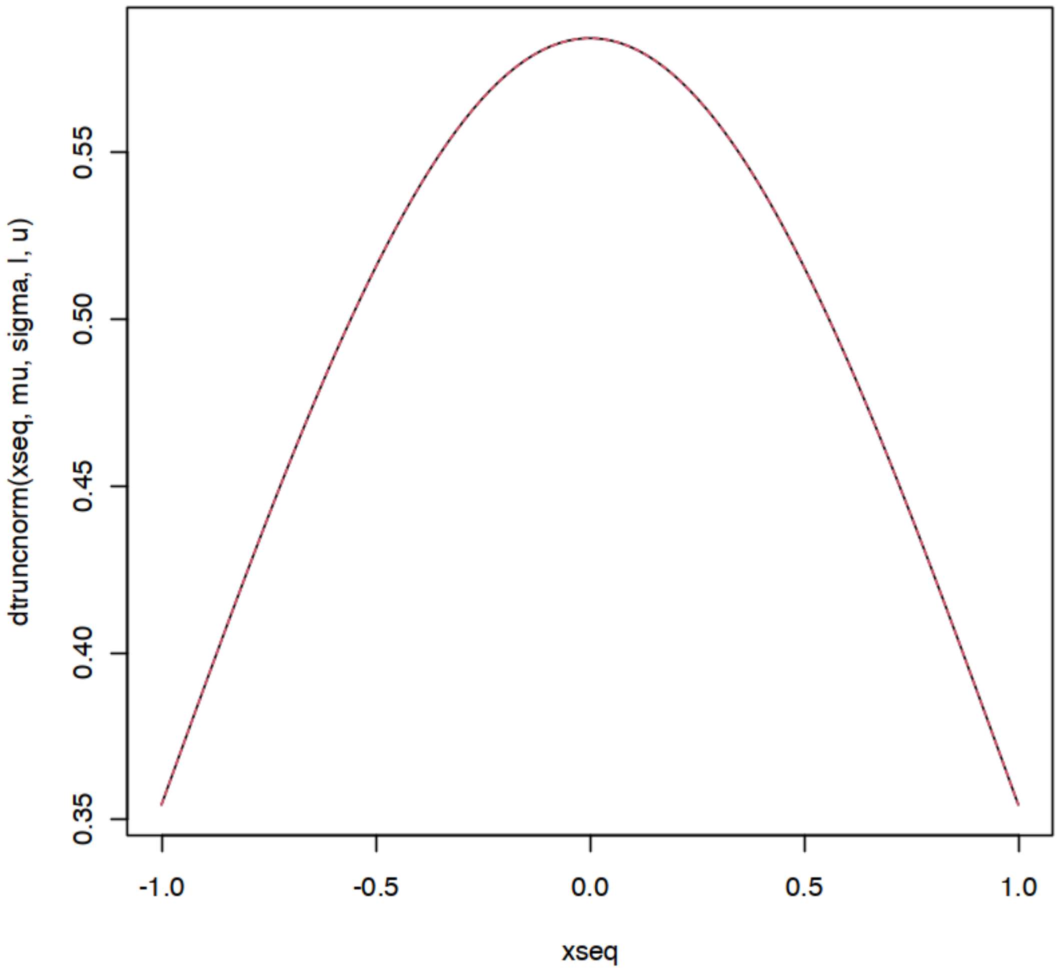
è la cumulata di una normale di media μ e varianza σ^2 valutata in y. Notate come $\phi(x|\mu, \sigma^2)$ è il kernel della distribuzione

Usando il metodo Accept Reject, simulate dalla normale troncata, usando l'M più piccolo possibile e come densità g usate $\phi(.|\mu, \sigma^2)$. Disegnate coi dei punti colorati, i valori che accettate e quelli che rifiutate (usate valori simili a $\mu = 0$, $l = -1$ $u = 1$, $\sigma^2 = 1$). Vi consiglio di disegnare prima i punti i poi, se volete, le densità.

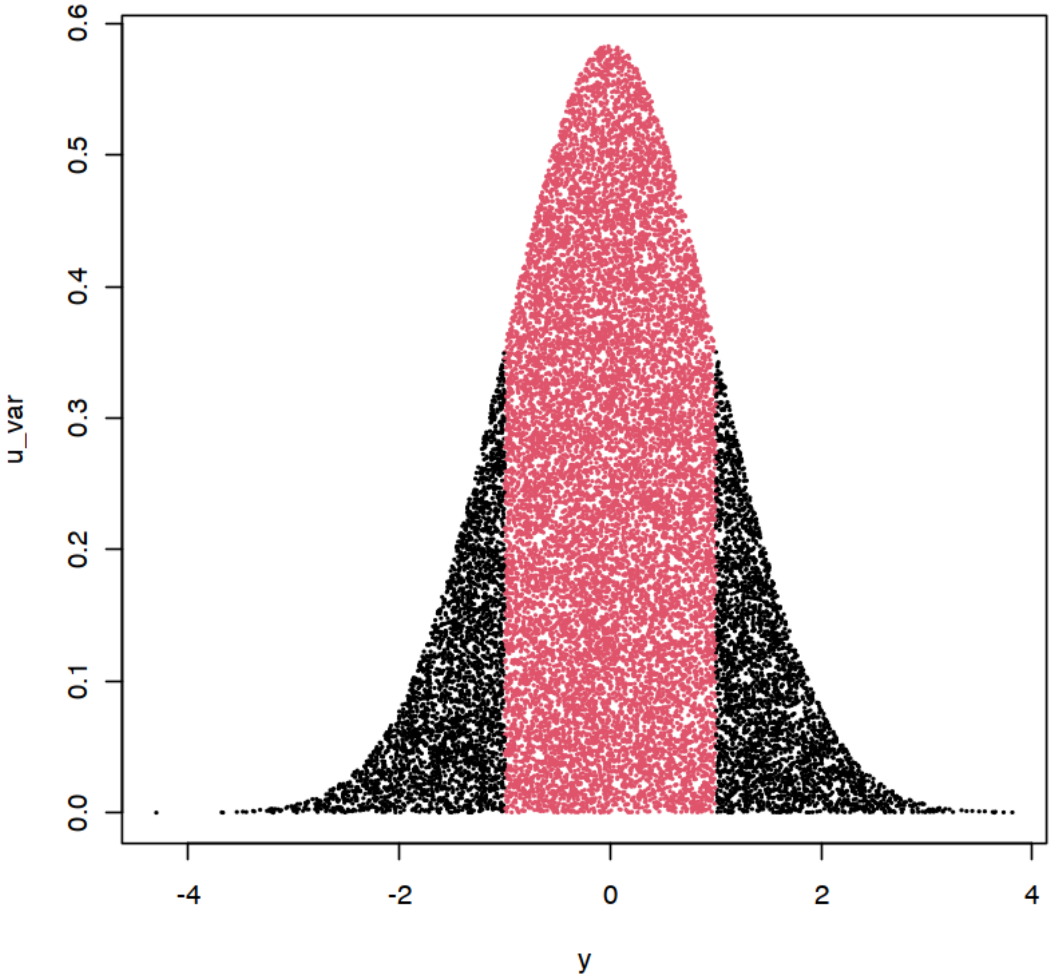
Fate attenzione che se scrivete la densità della normale troncata, dovete dire che la densità è 0 per valori fuori dai limiti

[104]:
```
set.seed(1)
# punto 1
dtruncnorm = function(x, mu, sigma, l,u)
{
    pl = pnorm(l, mu,sigma)
    pu = pnorm(u, mu,sigma)
    ret = rep(NA, length(x))
    for(i in 1:length(x))
    {
        if((x[i]<l)| (x[i]>u))
        {
            ret[i] = 0
```

```
        }else{
            ret[i] =  dnorm(x[i],mu,sigma)/(pu-pl)
        }

    }
    return( ret)
}

# parametri
mu = 0
sigma=1
l = -1
u = 1
xseq = seq(l,u,length.out=100)



if((mu>l) & (mu<u))
{
    M = dtruncnorm(mu, mu, sigma, l,u)/dnorm(mu, mu, sigma)
}else{
    M = max(c(dtruncnorm(l, mu, sigma, l,u)/dnorm(l, mu, sigma), dtruncnorm(u, 
 ↪mu, sigma, l,u)/dnorm(u, mu, sigma)))
}

plot(xseq, dtruncnorm(xseq, mu, sigma, l,u), type="l")
lines(xseq, M*dnorm(xseq, mu, sigma), type="l", col=2, lty=2)
```



[105]:
```
# simul0
nsim = 20000
y = rnorm(nsim, mu, sigma)
u_var = runif(nsim,0,M*dnorm(y, mu, sigma))
w_acc = u_var<= dtruncnorm(y, mu, sigma, l,u)

plot(y,u_var, col=w_acc+1, pch=20, cex=0.1)
```



1.2 Variabili Miste

Assumete di avere delle variabili

$$X \sim N(\mu, \sigma^2)$$

e

$$W sim TN(\mu, \sigma^2, 0, \infty)$$

e di definire Y_i come

$$Y = X \text{ se } X > 0 \qquad Y = 0 \text{ se } X = 0$$

e

$$Z = W \text{ se } K = 1 \qquad Z = 0 \text{ se } K = 0$$

con

$$K \sim Bern(p)$$

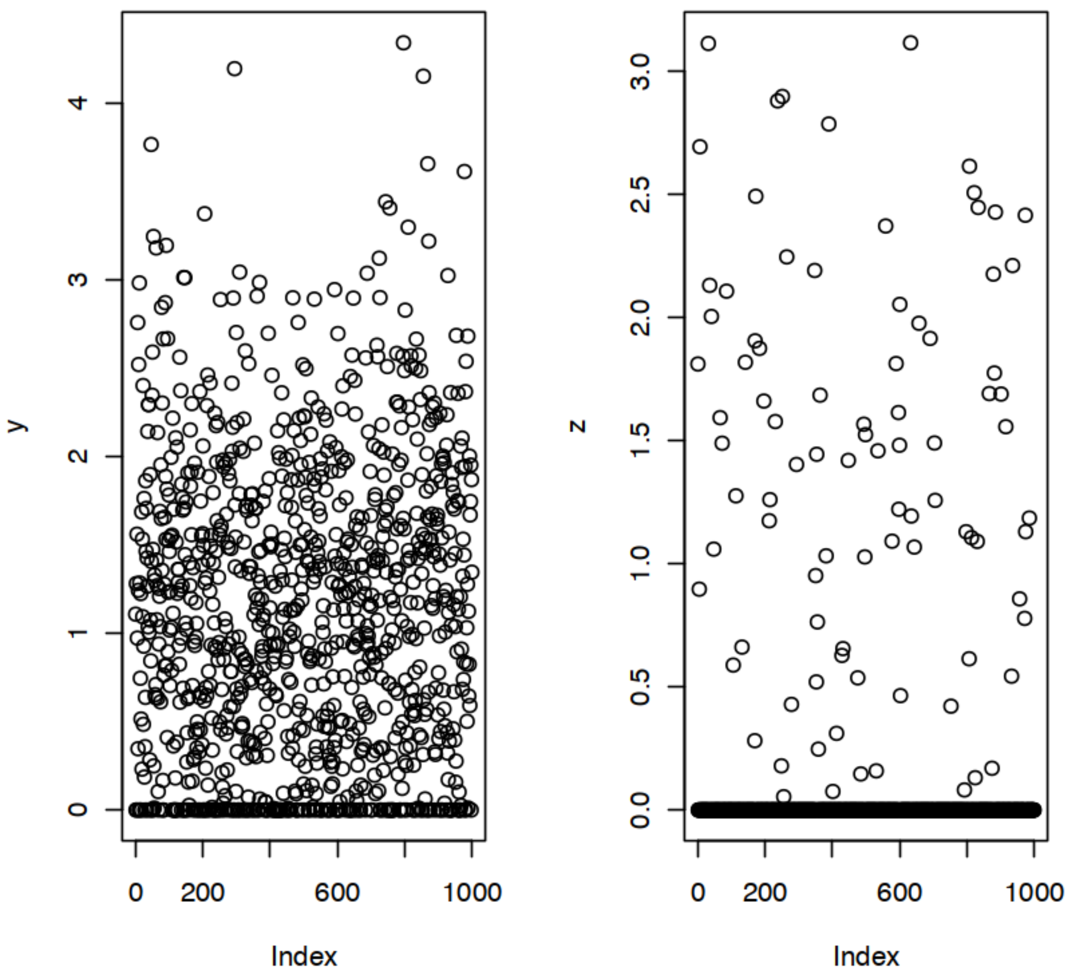
Sia Z che Y sono variabili miste

1. Simulate un campione delle due variabili aleatorie (usate il pacchetto truncnorm per simulare dalla normale troncata). Provate diversi valori dei parametri per capire cosa fanno.
2. Calcolate quanto vale $P(Y = 0)$ e P(Z = 0) (matematicamente e stimatelo tramite MC)

Soluzioni Punto 1

[123]:

```
library(truncnorm)
nsim = 1000
mu = 1
sigma2 = 1
p = 0.1
x = rnorm(nsim, mu, sigma2^0.5)
k = rbinom(nsim, 1,p)
w = rtruncnorm(nsim,0, Inf, mu, sigma2^0.5)

y = ifelse(x<=0,0,x)
z = ifelse(k==0,0,w)
par(mfrow=c(1,2))
plot(y)
plot(z)
par(mfrow=c(1,1))
```



per il punto due dobbiamo calcolare

$$P(Y=0) = P(X<0) = \Phi(0|\mu, \sigma^2)$$

e

$$P(Z=0) = P(K=0) = 1-p$$

[126]:
```
p_y0 = mean(y==0)
p_w0 = mean(z==0)
p_y0
p_w0
pnorm(0, mu, sigma2^0.5)
1-p
```

0.152

0.911

0.158655253931457

0.9

[ ]:

1.3 Indipendenza condizionata

Assumete di avere

$$Z = A + W_1 \quad B = A + W_1$$

con

$$W_1 \sim Bern(p_1) \quad W_2 \sim Bern(p_2) \quad A \sim Bern(p_3)$$

1. Valutate con MC quanto vale $P(B=0)$, $P(B=1)$ $P(B=2)$, $P(B=3)$
2. Valutate quanto vale $P(B=0|Z=1)$. $P(B=1|Z=1)$ $P(B=2|Z=1)$, $P(B=3|Z=1)$. **Note**: Se questi sono diverse (fate sempre attenzione che sono stime) dai valori del primo punto, allora Z e B sono dipendenti
3. Valutate quanto vale $P(B=0|A=j_a, Z=j_z)$. $P(B=1|A=j_a, Z=j_z)$ $P(B=2|A=j_a, Z=j_z)$, $P(B=3|A=j_a, Z=j_z)$ almeno per alcuni valori di j_a e j_z (fate attenzione che ci sono valori di j_a e j_z che hanno probabilità 0)

Soluzioni

Per il punto 1 potete semplicemente simulare le varie variabili aleatorie, costruire B, e poi calcolare la frequenza relativa di B=0 per stimare $P(B=0)$. Questo perchè

$$P(B=j) = P(A+W_2=j) = \int 1_j(a+w_2)f(a)f(w_2)d\lambda(a)d\lambda(w_2) \approx \frac{\sum 1_j(a_i+w_{2,i})}{n} = \frac{\sum 1_j(b_i)}{n}$$

[7]:

```
# punto 1
nsim = 10000

p1 = 0.3
p2 = 0.3
p3 = 0.5

w1 = rbinom(nsim,1,p1)
w2 = rbinom(nsim,1,p2)
a = rbinom(nsim,1,p3)

b = a+w1
z = a+w2

# P(B=0)
mean(b==0)
# P(B=1)
mean(b==1)
```

```
# P(B=2)
mean(b==2)
# P(B=3)
mean(b==3)
```

0.345

0.5022

0.1528

0

Per il secondo punto possiamo calcolare

$$P(B=j|Z=1)=\frac{P(B=j,Z=1)}{P(Z=1)}$$

e abbiamo che

$$P(B=j,Z=1)\approx\frac{\sum 1_j(b_i)1_1(z_i)}{n}$$

e

$$P(Z=1)\approx\frac{\sum 1_1(z_i)}{n}$$

Abbiamo allora che

$$P(B=j|Z=1)\approx\frac{\sum 1_j(b_i)1_1(z_i)}{\sum 1_1(z_i)}$$

[19]:
```
# punto 2
# P(B=0|Z=1)
sum((b==0) & (z==1))/sum((z==1))
# P(B=1|Z=1)
sum((b==1) & (z==1))/sum((z==1))
# P(B=2|Z=1)
sum((b==2) & (z==1))/sum((z==1))
# P(B=3|Z=1)
sum((b==3) & (z==1))/sum((z==1))
```

0.205096839959225

0.577981651376147

0.216921508664628

0

Per il punto 3, si può fare come i punti precedenti

$$P(B=j|A=j_a,Z=j_z)\approx\frac{\sum 1_j(b_i)1_{j_z}(z_i)1_{j_a}(a_i)}{\sum 1_{j_z}(z_i)1_{j_a}(a_i)}$$

[26]:
```
# punto 3
# P(B=1|A = 1,Z=1)
```

```
sum((b==1) & (z==1)& (a==1))/sum((z==1)& (a==1))
# P(B=1|A = 1,Z=2)
sum((b==1) & (z==2)& (a==1))/sum((z==2)& (a==1))

# P(B=2|A = 1,Z=1)
sum((b==2) & (z==1)& (a==1))/sum((z==1)& (a==1))
# P(B=2|A = 1,Z=2)
sum((b==2) & (z==2)& (a==1))/sum((z==2)& (a==1))
```

0.689705453484981

0.708908406524467

0.310294546515019

0.291091593475533

1.4 DAG e modelli gerarchici

Abbiamo il seguente modello

$$w_i|w_{i-1} \sim N(\mu + \alpha(w_{i-1} - \mu), \sigma^2), i = 2 \dots, n$$

e

$$w_1 = \mu$$

questo modello è un chiamato autoregressivo 1 (AR(1))

1. Scrivere il DAG associato al modello
2. valutate graficamente la variabilità e la media della serie per ogni punto n sia con $|\alpha| < 1$ che con $|\alpha| > 1$

Soluzioni Punto 2

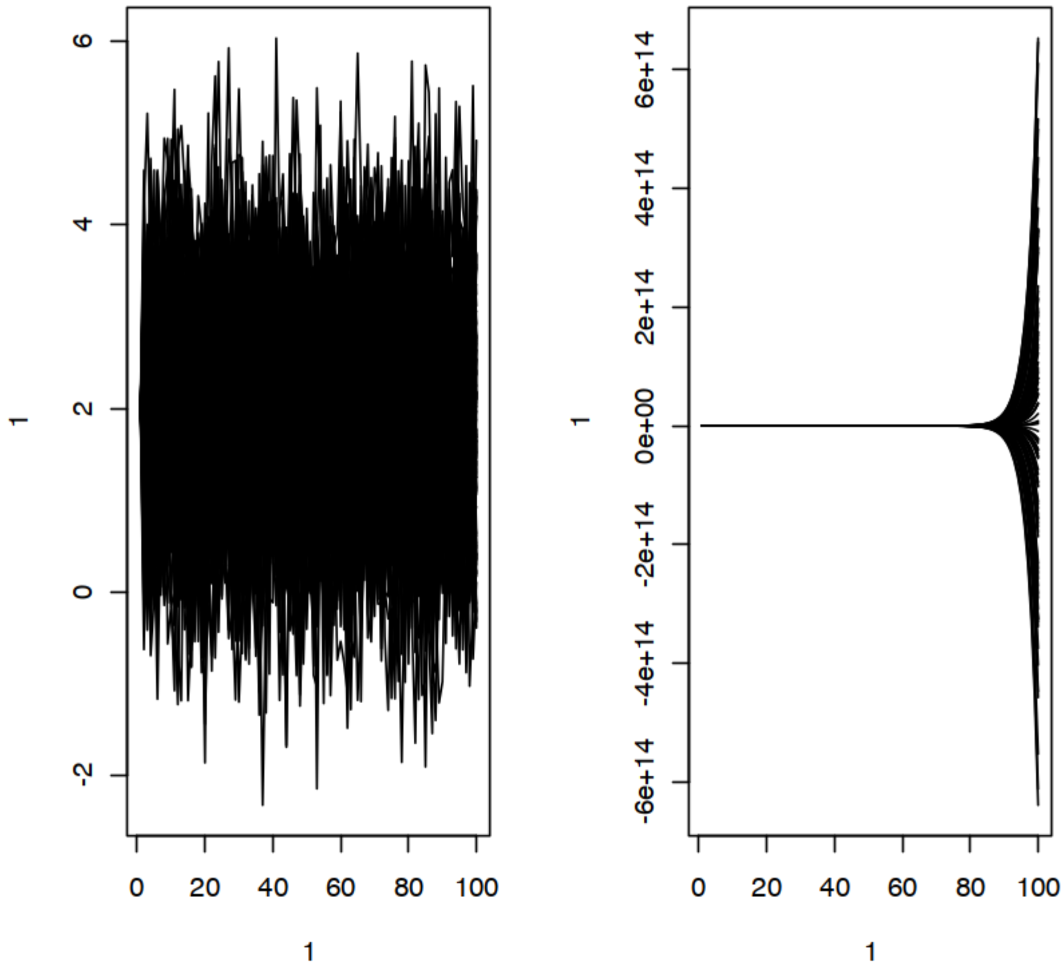
Per ogni valore di α, simuliamo diverse serie e ne valutiamo il comportamento

[34]:
```
n = 100
nsim = 100

alpha_1 = 0.4
alpha_2 = 1.4
sigma2 = 1
mu = 2
sim_w_1 = matrix(NA, nrow = n, ncol = nsim)
sim_w_2 = matrix(NA, nrow = n, ncol = nsim)
for(isim in 1:nsim)
{
    sim_w_1[1,isim] = mu
    sim_w_2[1,isim] = mu
    for(i in 2:n)
    {
        sim_w_1[i,isim] = rnorm(1,mu + alpha_1*(sim_w_1[i-1,isim] - mu),␣
 ↪sigma2^0.5)
```

```
        sim_w_2[i,isim] = rnorm(1,mu + alpha_2*(sim_w_2[i-1,isim] - mu),
 ↪sigma2^0.5)
    }
}
```

[35]:

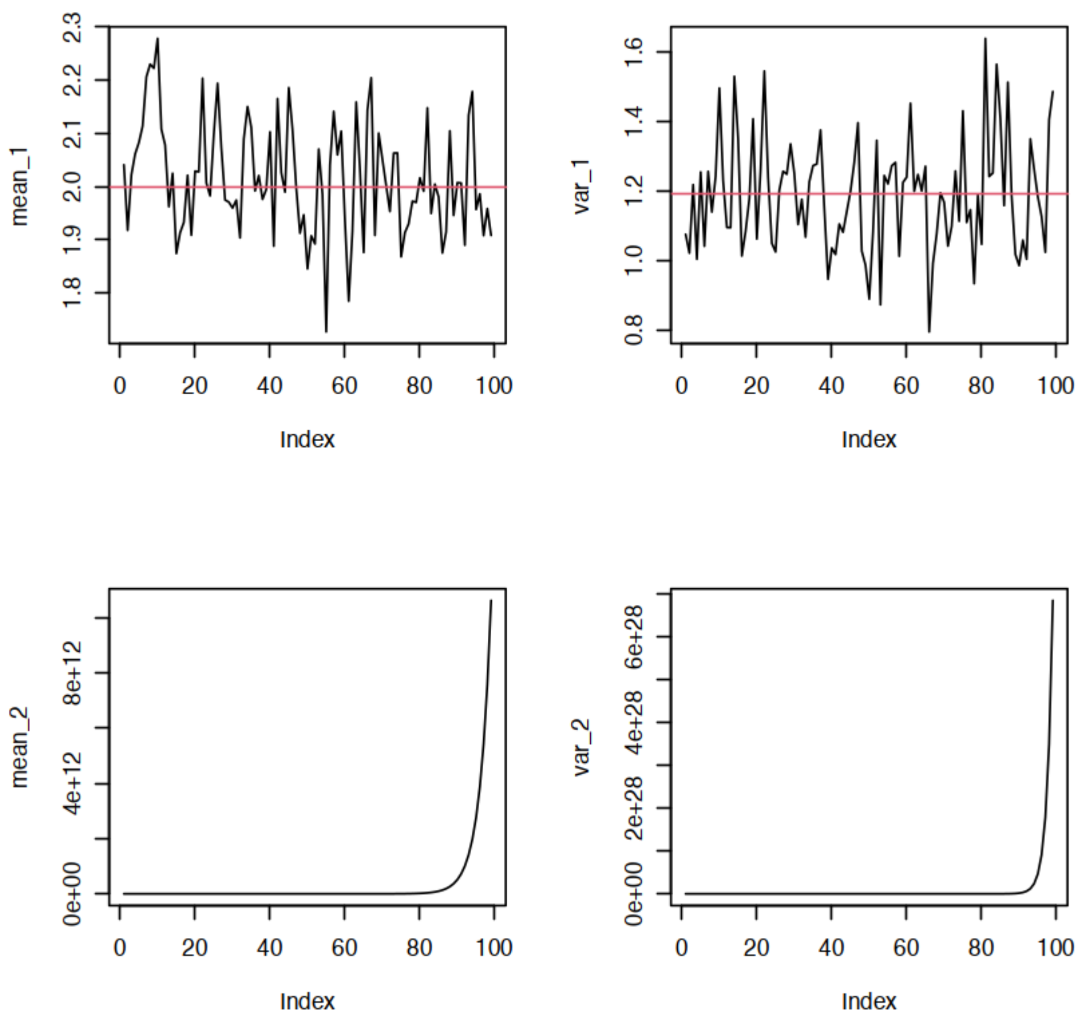
```
par(mfrow=c(1,2))
plot(1,1, type="n", ylim=range(c(sim_w_1)), xlim= c(1,n))
for(isim in 1:nsim)
{
    lines(sim_w_1[,isim], type="l")
}
plot(1,1, type="n", ylim=range(c(sim_w_2)), xlim= c(1,n))
for(isim in 1:nsim)
{
    lines(sim_w_2[,isim], type="l")
}
par(mfrow=c(1,1))
```



possiamo calcolare stime della media e della varianza

[40]:
```
mean_1 = apply(sim_w_1[-1,],1,mean)
mean_2 = apply(sim_w_2[-1,],1,mean)

var_1 = apply(sim_w_1[-1,],1,var)
var_2 = apply(sim_w_2[-1,],1,var)

par(mfrow=c(2,2))
plot(mean_1, type="l")
abline(h = mu, col=2)
plot(var_1, type="l")
abline(h = sigma2/(1-alpha_1^2), col=2)
```

```
plot(mean_2, type="l")
plot(var_2, type="l")
par(mfrow=c(1,1))
```



vediamo che nel primo caso, medie e varianza sembrano costanti (il loro vero valore lo è). E la media è μ mentre la varianza $\sigma^2/(1-\alpha^2)$. Con $|\alpha| \leq 1$ la serie 'e stazionaria (vedremo in seguito che significa), altrimenti no

1.5 Simulazione processo spaziale

Assumete di avere una griglia regolare con $i = 1, \dots, n$ righe e $i = 1, \dots, n$ colonne (prendete piccoli, per esempio $n = 30$). Per ogni punto griglia avete una realizzazioni di una variabile aleatoria w_{ij}.

Il vettore $\mathbf{w} = (w_1, w_{1,2}, \dots w_{nn})'$ di tutte le osservazioni ha distribuzione normale

$$\mathbf{w} \sim N(\mathbf{0}, \Sigma)$$

Assumete poi che

$$y_{ij}|w_{ij} \sim Pois(\exp(\mu + w_{ij}))$$

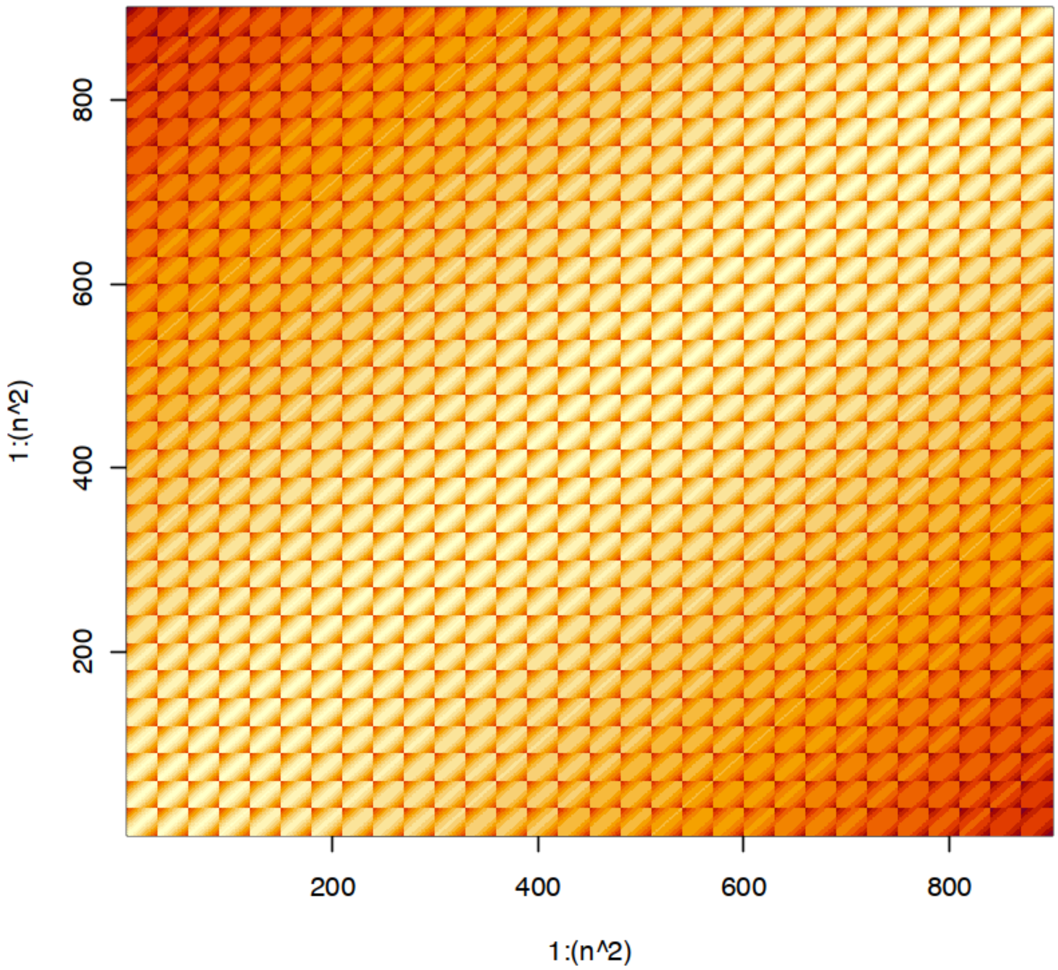
Gli elementi di Σ sono definiti come

$$Cov(W_{ij}, W_{i',j'}) = \sigma^2 \exp(-\rho\sqrt{(i-i')^2 + (j-j')^2})$$

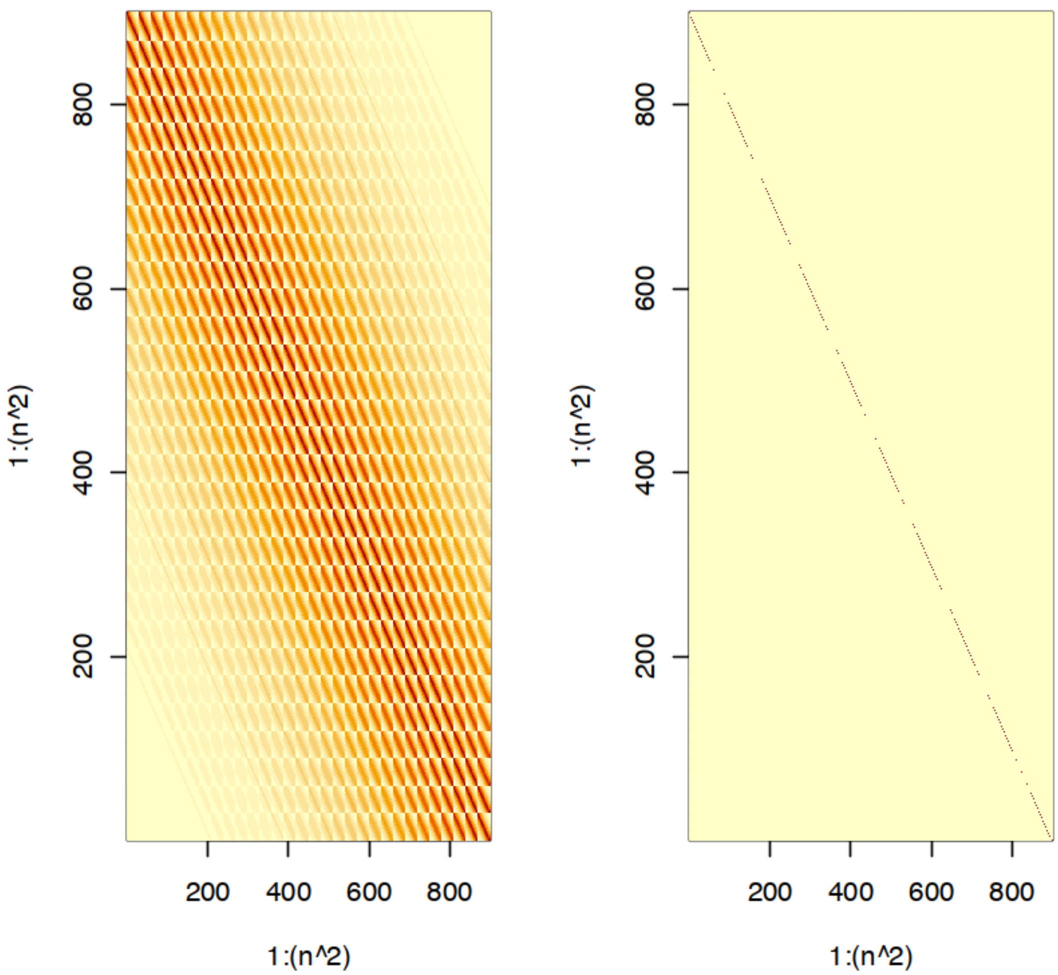
1. Scrivete una DAG che rappresenti il modello (fatolo con n = 2, o 3)
2. simulate una realizzazioni del modello e mostrate, con un heatmap (comando image di R), i valori di w e y. Vedete cosa cambia con $\rho \in \{3/n, 3/1\}$ e provate diversi μ e σ^2

Soluzioni Punto 2. Per semplicità, assumo che il vettore **w** contanga prima tutti i valori della prima riga, poi quelli dellla seconda etc

[61]:
```
library(MCMCpack)

n = 30
phi_1 = 3/n
phi_2 = 3/1




### per riempire sigma, creo una matrice delle coordinate che poi uso per␣
 ↪calcolare le distanze \sqrt{(i-i')^2   + (j-j')^2}
coords = matrix(NA, nrow=n^2, ncol=2)
h = 0
for(i in 1:n)
{
    for(j in 1:n)
    {
        h = h+1
        coords[h, ] = c(i,j)
    }
}
dist_mat = as.matrix(dist(coords))

## mostor i vlaori delle distanza tra le n^2 osservvazioni
image(dist_mat, x = 1:(n^2), y = 1:(n^2))
```



[68]:
```
# calcolo i valori della covarianza, senza sigma^2
sigma_2_1 = matrix(NA, ncol = n^2, nrow=n^2)
sigma_2_2 = matrix(NA, ncol = n^2, nrow=n^2)
sigma_2_1 = 1*exp(- phi_1* dist_mat)
sigma_2_2 = 1*exp(- phi_2* dist_mat)

## facico un plot delle matrici
par(mfrow=c(1,2))
image(sigma_2_1[(n^2):1,], x = 1:(n^2), y = 1:(n^2))
image(sigma_2_2[(n^2):1,], x = 1:(n^2), y = 1:(n^2))
par(mfrow=c(1,1))
```



[73]:
```
# aggiungo la varianza e simulo
sigma2 = 2
chol_mat_1 = t(chol(sigma2*sigma_2_1))
w_1 = chol_mat_1%*%matrix(rnorm(n^2), ncol=1)
chol_mat_2 = t(chol(sigma2*sigma_2_2))
w_2 = chol_mat_2%*%matrix(rnorm(n^2), ncol=1)


mu = 1

y_1 = exp(mu + w_1)
y_2 = exp(mu + w_2)
```

```r
## plotto i campi Gaussiani e quello poissono

par(mfrow=c(2,2))
image(matrix(w_1,ncol=n, nrow=n))
image(matrix(y_1,ncol=n, nrow=n))
image(matrix(w_2,ncol=n, nrow=n))
image(matrix(y_2,ncol=n, nrow=n))
par(mfrow=c(1,1))
```