Esercitazione 8

November 29, 2024

1 Gaussian Processes

1.1 Processo Spaziale

Facciamo delle simulazioni di un processo Gaussiano. Assumiamo che

$$Y(\mathbf{s})|W(\mathbf{s}) \overset{iid}{\sim} GP(\mu + W(\mathbf{s}), \tau^2)$$

$$W(\mathbf{s}) \sim GP(0, C(||\mathbf{s}-\mathbf{s}'||;\theta))$$

con

$$C(||\mathbf{s}-\mathbf{s}'||;\theta) = \sigma^2 \exp(-\phi||\mathbf{s}-\mathbf{s}'||)$$

e

$$\mathbf{s} \in \mathbb{R}^2$$

La realizzazione del GP sono su una griglia regolare di m punti in $[0,1]^2$.

0. valutate quanto crece il tempo computazionale che ci si mette per invertire una matrice di covarianza (o il calcolo della decomposizione di Cholesky) per dimensione crescente
1. Disegnate la funzione di correlazione per diversi valori di ϕ, ta cui anche $\frac{3}{\text{min dist}}$ e $\frac{3}{\text{max dist}}$
2. Simulate W e Y e rappresentateli graficamente
3. Assumete noti i parametri e simulate dalla a posteriori di $\mathbf{W}|\mathbf{y}$, e confrontate media a posteriori con il valore vero
4. fate uno scatterplot tra i valori delle distanze e i valori della correlazione, sia a priori che a posteriori

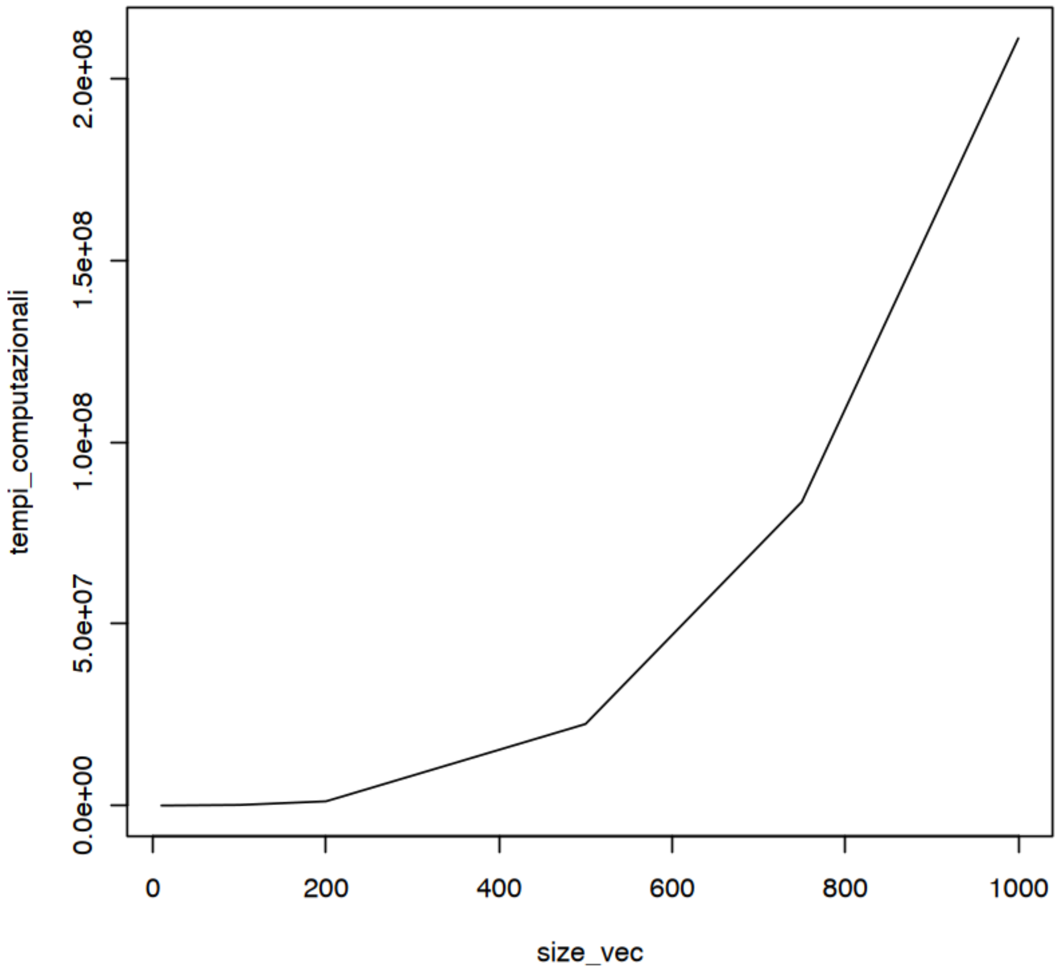
[ ]:

```

```

[92]:

```
# punto 0
library(microbenchmark)
size_mat = 1000
C = matrix(runif(size_mat^2), ncol=size_mat)
Cov = C%*%t(C)

# adesso prendo una parte di cov e provo a invertirla 20 volte. Lo faccio per
↪diverse dimensioni
tempi_computazionali <- c()
size_vec <- c(10,100,200,500,750,1000)
n_test <- 20
size <- size_vec[1]
```

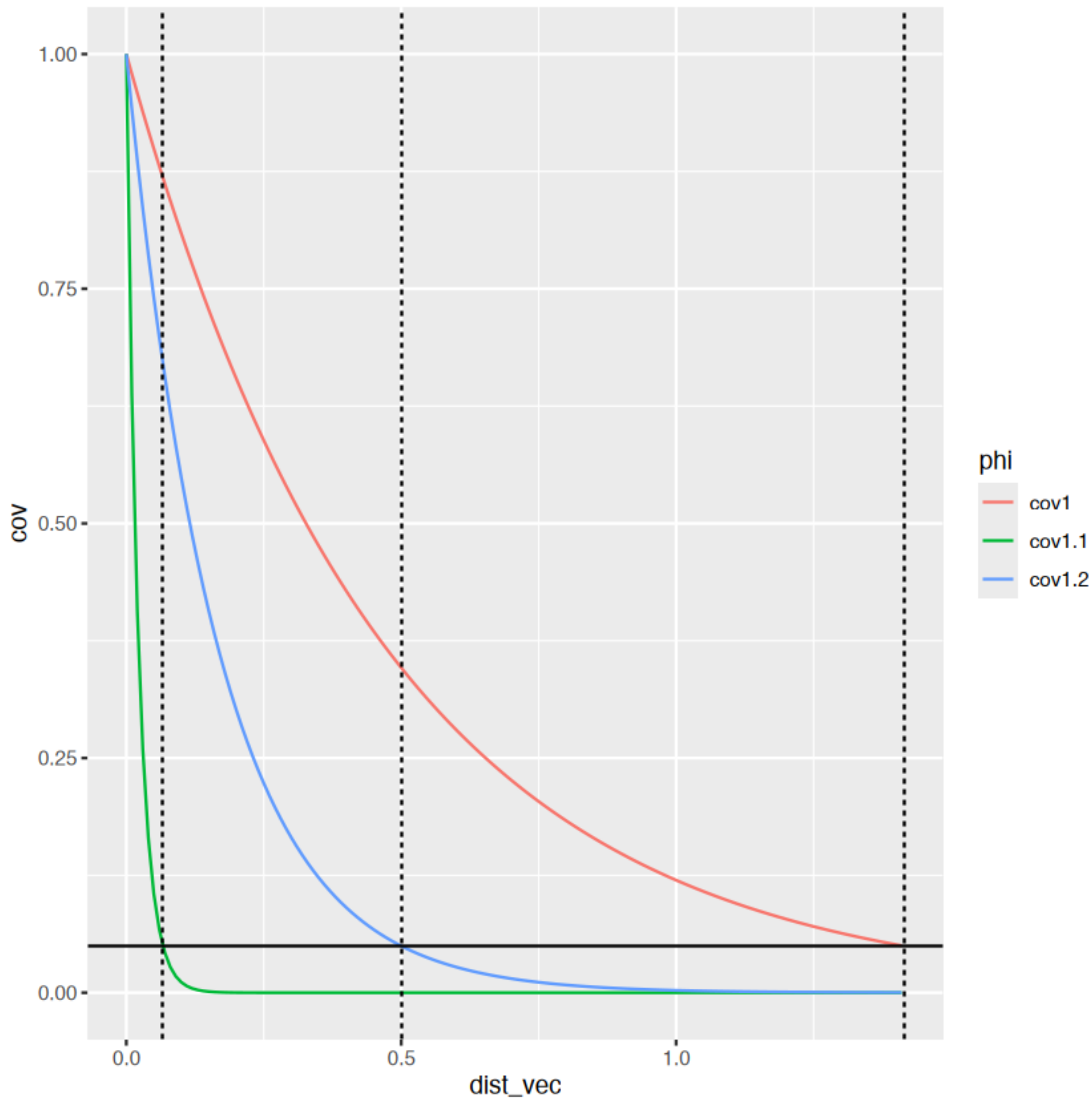
```
tempi_computazionali[1] <- mean(microbenchmark(chol(Cov[1:size, 1:size]), times = n_test)[, 2])
size <- size_vec[2]
tempi_computazionali[2] <- mean(microbenchmark(chol(Cov[1:size, 1:size]), times = n_test)[, 2])
size <- size_vec[3]
tempi_computazionali[3] <- mean(microbenchmark(chol(Cov[1:size, 1:size]), times = n_test)[, 2])
size <- size_vec[4]
tempi_computazionali[4] <- mean(microbenchmark(chol(Cov[1:size, 1:size]), times = n_test)[, 2])
size <- size_vec[5]
tempi_computazionali[5] <- mean(microbenchmark(chol(Cov[1:size, 1:size]), times = n_test)[, 2])
size <- size_vec[6]
tempi_computazionali[6] <- mean(microbenchmark(chol(Cov[1:size, 1:size]), times = n_test)[, 2])

plot(size_vec, tempi_computazionali, type="l")
```



[93]:

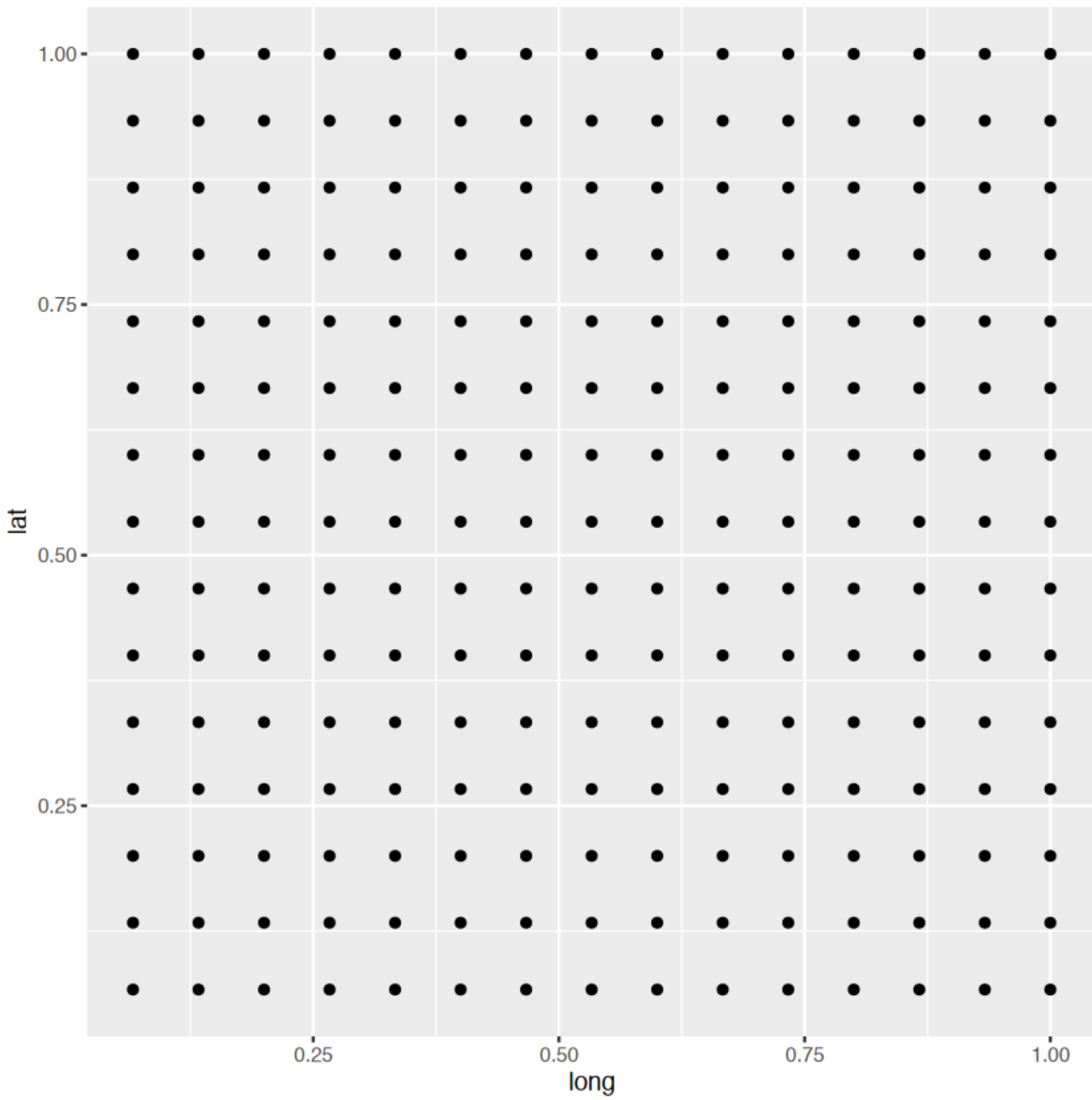
```
library(ggplot2)
library(tidyverse)
m <- 15
min_dist <- 1/m
max_dist <- sqrt(2)
dist_vec <- seq(0, max_dist, by = 0.01)
phi_vec = c(3 / max_dist, 3 / min_dist, 3/0.5)
data_plot = data.frame(
    dist_vec = dist_vec,
    cov1 =  exp(-phi_vec[1] * dist_vec),
    cov1 =  exp(-phi_vec[2] * dist_vec),
    cov1 =   exp(-phi_vec[3] * dist_vec)
    )
```

```
data_plot%>% pivot_longer(cols = 2:4, names_to = "phi", values_to = "cov")%>%␣
  ↪ggplot(aes(x = dist_vec, y = cov, color = phi)) +geom_line()+
geom_hline(yintercept = 0.05) + geom_vline(xintercept = c(min_dist, max_dist, 0.
  ↪5), linetype = "dashed")
```



[94]:
```
n = m^2
coords <- data.frame(long = rep(NA, n), lat = rep(NA, n))
h = 1
for(i in 1:m)
{
```

```
  for(j in 1:m)
  {
    coords[h,] <- c(i,j)/m
    h <- h + 1
  }
}
coords%>% ggplot(aes(x = long, y = lat)) + geom_point()
```

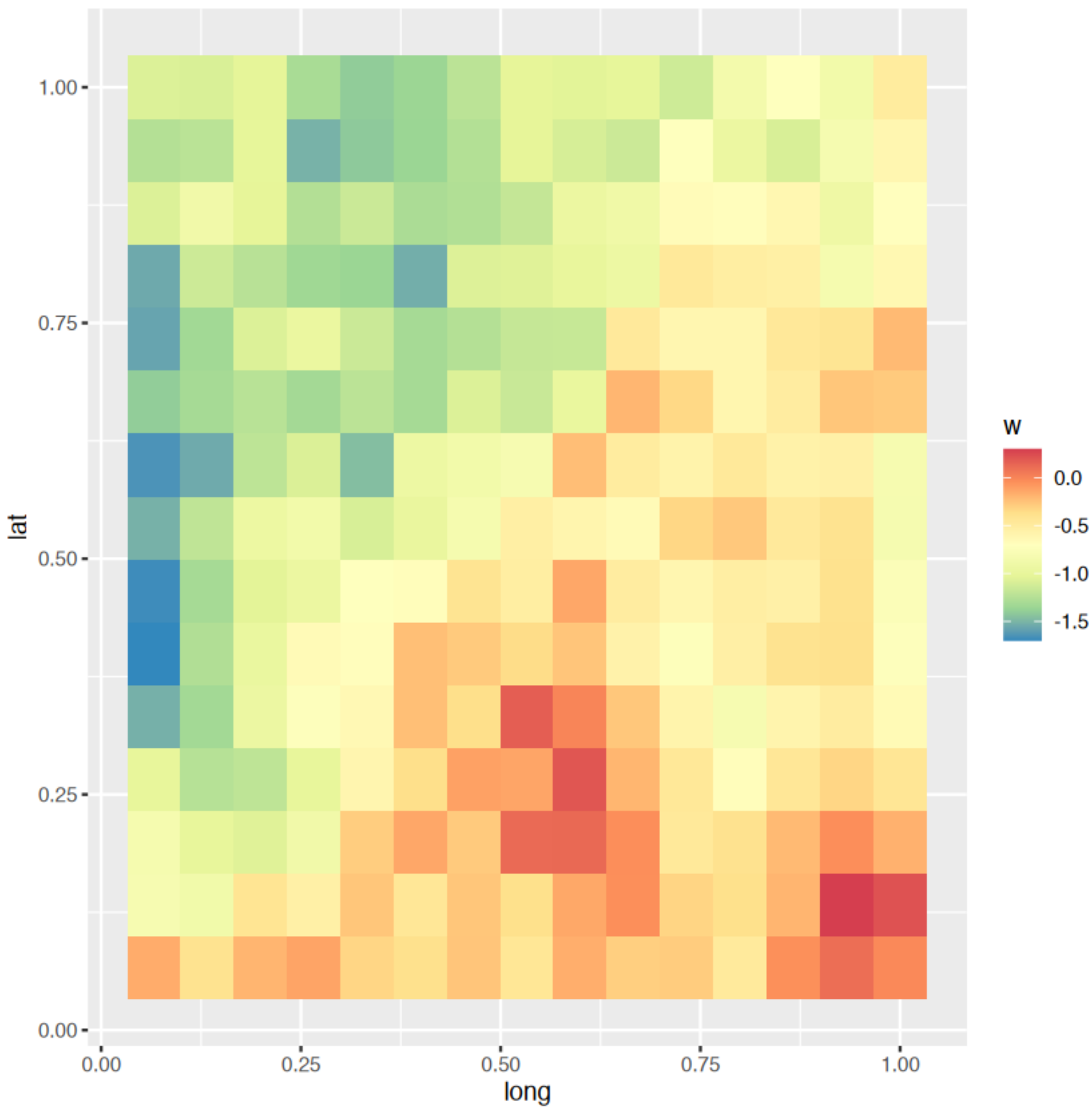


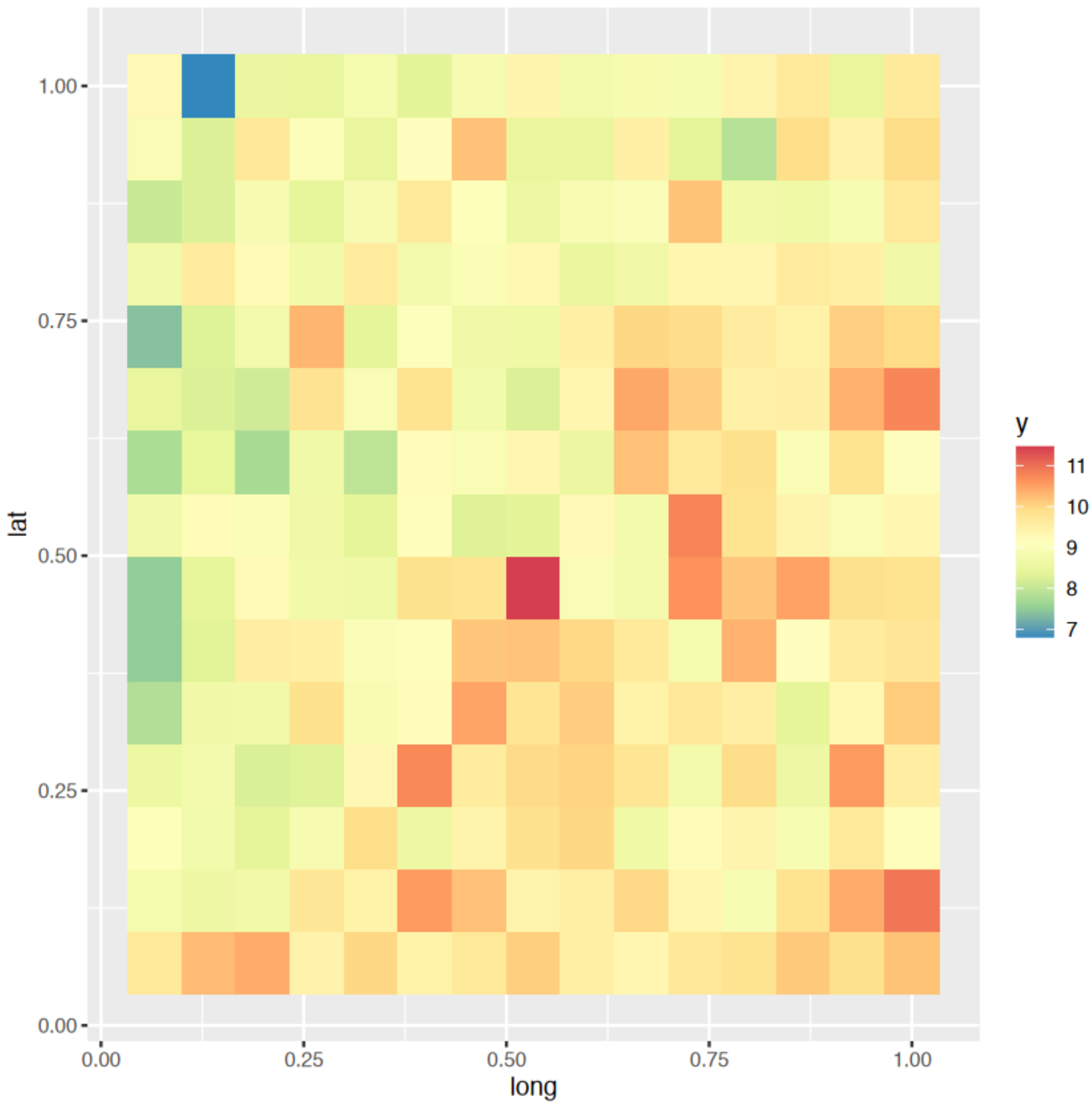
[95]:

```
mu <- 10
sigma2 <- 0.5
tau2 <- 0.3
phi <- 1
mu <- 10
```

```
dist_mat = as.matrix(dist(coords))
cov_w <- sigma2 * exp(-phi*dist_mat)
w_sim <- t(chol(cov_w))%*%matrix(rnorm(n), ncol=1)
y_sim <- mu + w_sim + rnorm(n, 0, tau2^0.5)
```

[96]:
```
data_sim <- data.frame(long = coords$long, lat = coords$lat, w = w_sim, y =␣
 ↪y_sim)
data_sim%>% ggplot(aes(x = long, y = lat, fill = w)) + geom_tile() +␣
 ↪scale_fill_distiller(palette = "Spectral")
data_sim %>% ggplot(aes(x = long, y = lat, fill = y)) +
  geom_tile() +
  scale_fill_distiller(palette = "Spectral")
```





Per il calcolo della a posteriori sappimao che

$$f(\mathbf{y}|\mathbf{w}) = (2\pi\tau^2)^{-0.5n} \exp\left(-\frac{(\mathbf{y} - \mathbf{1}\mu - \mathbf{w})^T(\mathbf{y} - \mathbf{1}\mu - \mathbf{w})}{2\tau^2}\right)$$

e

$$f(\mathbf{w}) \propto \exp\left(-\frac{\mathbf{w}^T\Sigma^{-1}\mathbf{w}}{2}\right)$$

d dove gli elementi di Σ sono calcolati con la funzione di covarianza.

La a posteriori è quindi proporzionale a

$$\exp\left(-\frac{\mathbf{w}^T\Sigma^{-1}\mathbf{w}}{2} - \frac{\mathbf{w}^T\mathbf{w}^T - 2\mathbf{w}^T(\mathbf{y} - \mathbf{1}\mu)}{2\tau^2}\right)$$

e quindi

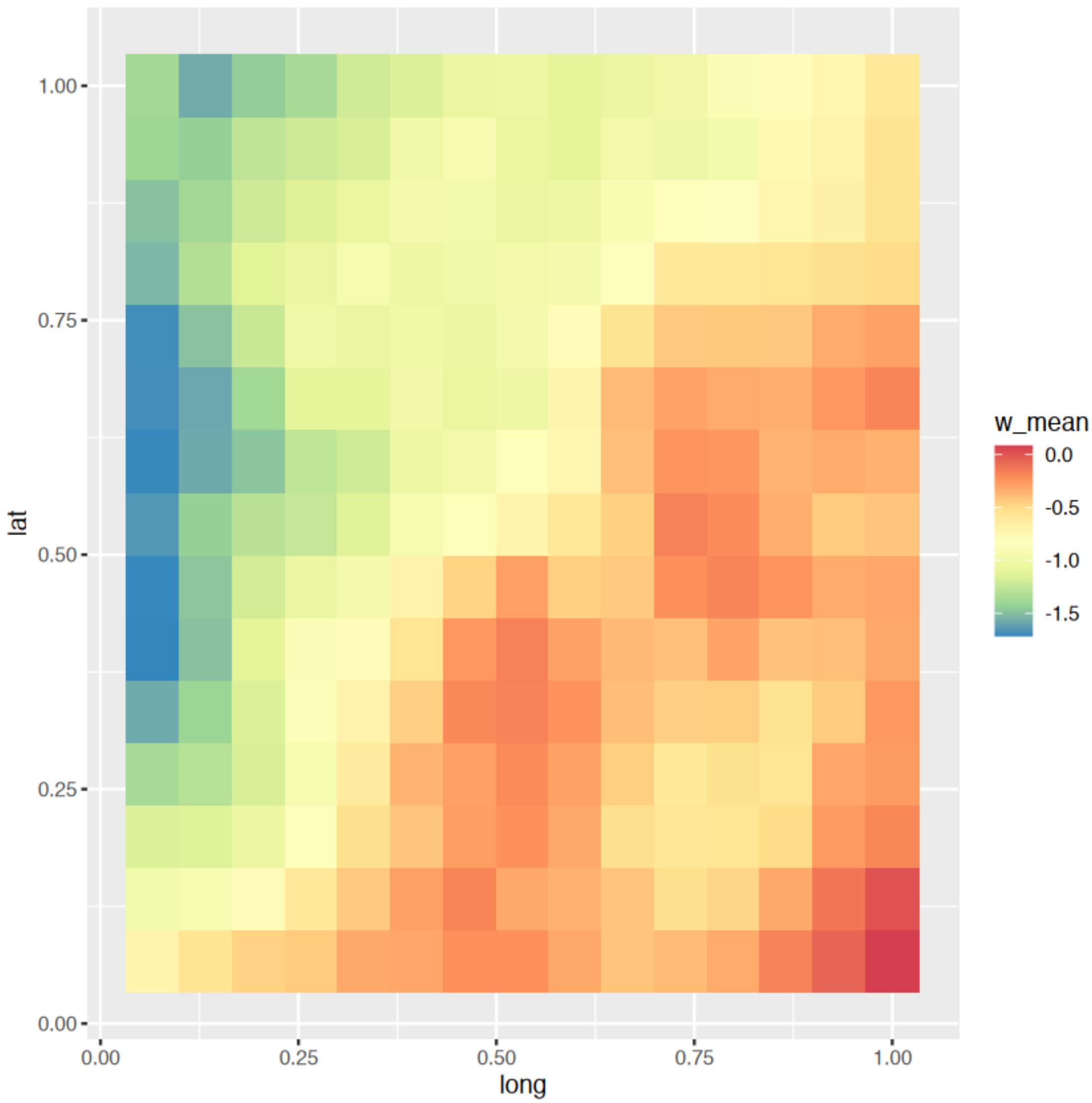
$$\mathbf{W}|\mathbf{y} \sim N\left(\left(\Sigma^{-1}+\frac{1}{\tau^2}\mathbf{I}_n\right)^{-1}\frac{(\mathbf{y}-\mathbf{1}\mu)}{\tau^2}, \left(\Sigma^{-1}+\frac{1}{\tau^2}\mathbf{I}_n\right)^{-1}\right)$$

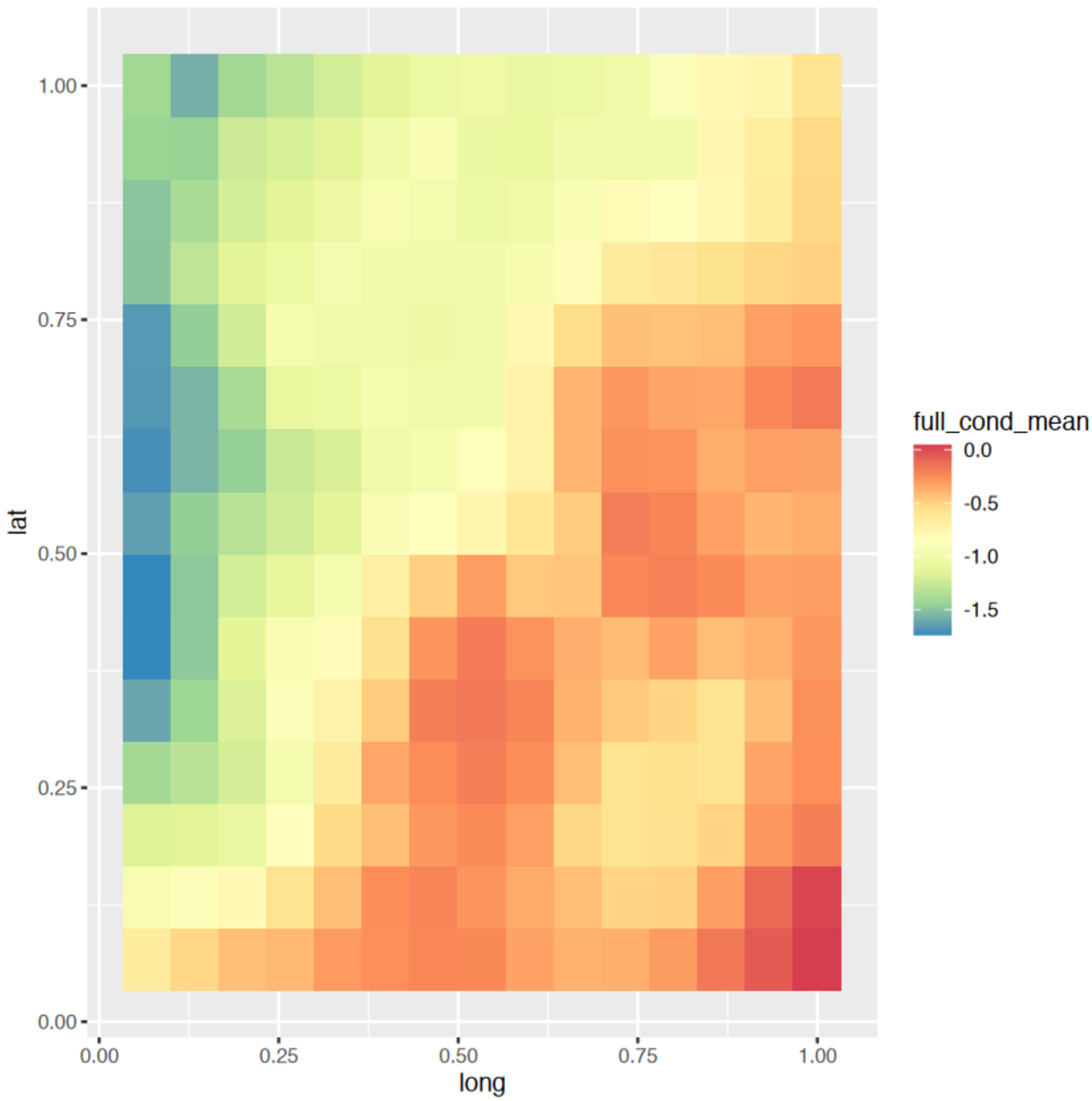
[97]:
```
cov_w_post = solve(solve(cov_w) + 1 / tau2 * diag(n))
chol_cov_w_post = t(chol(cov_w_post))
mean_w_post = cov_w_post %*% matrix((y_sim - mu) / tau2, ncol=1)
```

[98]:
```
n_post <- 100
w_post = matrix(NA, nrow = n, ncol=n_post)
for(isim in 1:n_post)
{
  w_post[,isim]<- chol_cov_w_post%*%matrix(rnorm(n), ncol=1) + mean_w_post
}
w_post_mean <- rowMeans(w_post)
```

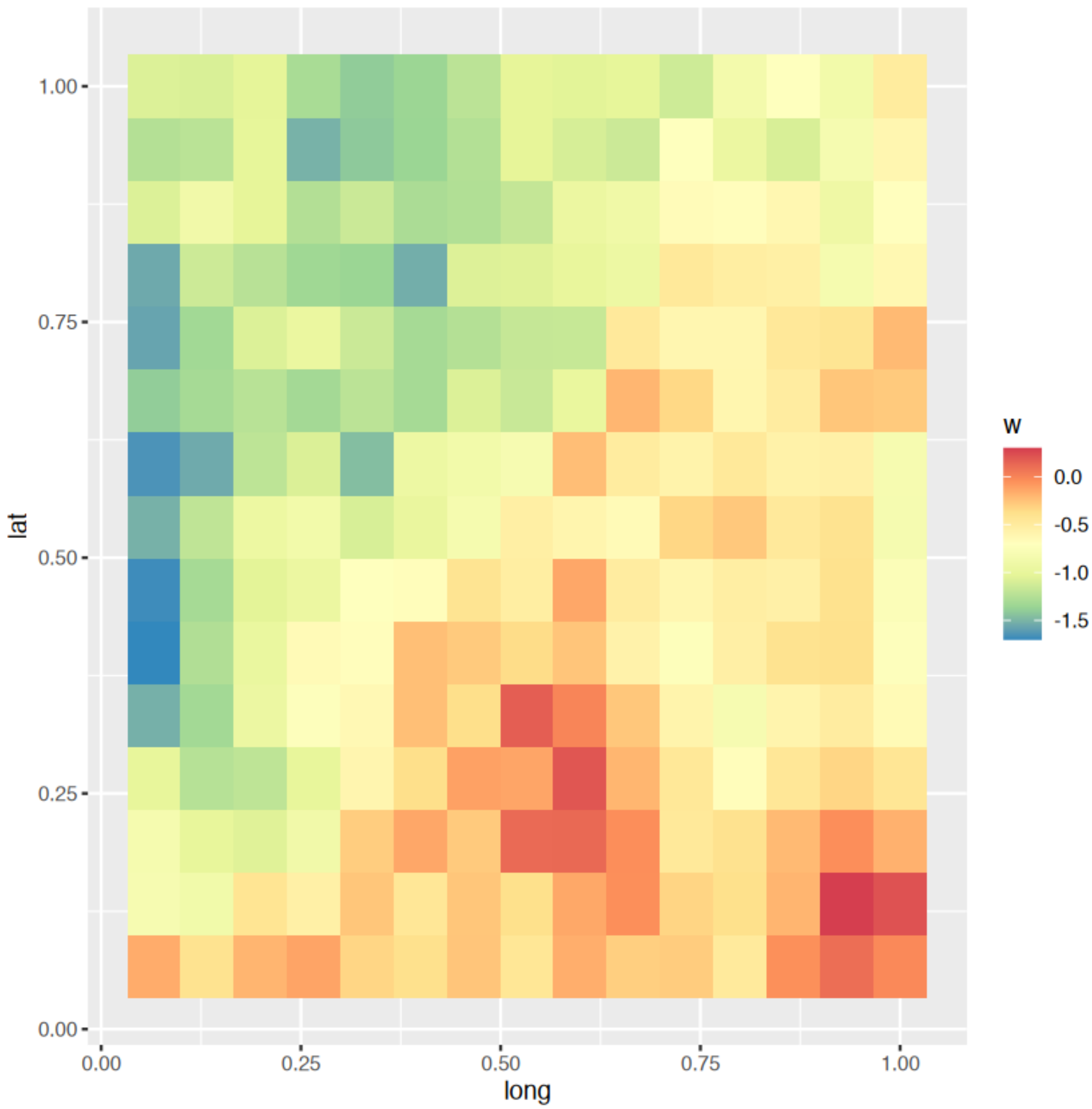
Per completezza mostro anche la media della full conditional

[99]:
```
data_sim <- data.frame(long = coords$long, lat = coords$lat, w = w_sim, y =␣
 ↪y_sim, w_mean = w_post_mean, full_cond_mean = mean_w_post)
data_sim %>% ggplot(aes(x = long, y = lat, fill = w_mean)) +
  geom_tile() +
  scale_fill_distiller(palette = "Spectral")
data_sim %>% ggplot(aes(x = long, y = lat, fill = full_cond_mean)) +
  geom_tile() +
  scale_fill_distiller(palette = "Spectral")
data_sim %>% ggplot(aes(x = long, y = lat, fill = w)) +
  geom_tile() +
  scale_fill_distiller(palette = "Spectral")
```


w_mean
0.0
-0.5
-1.0
-1.5
lat
1.00
0.75
0.50
0.25
0.00
long
0.00
0.25
0.50
0.75
1.00

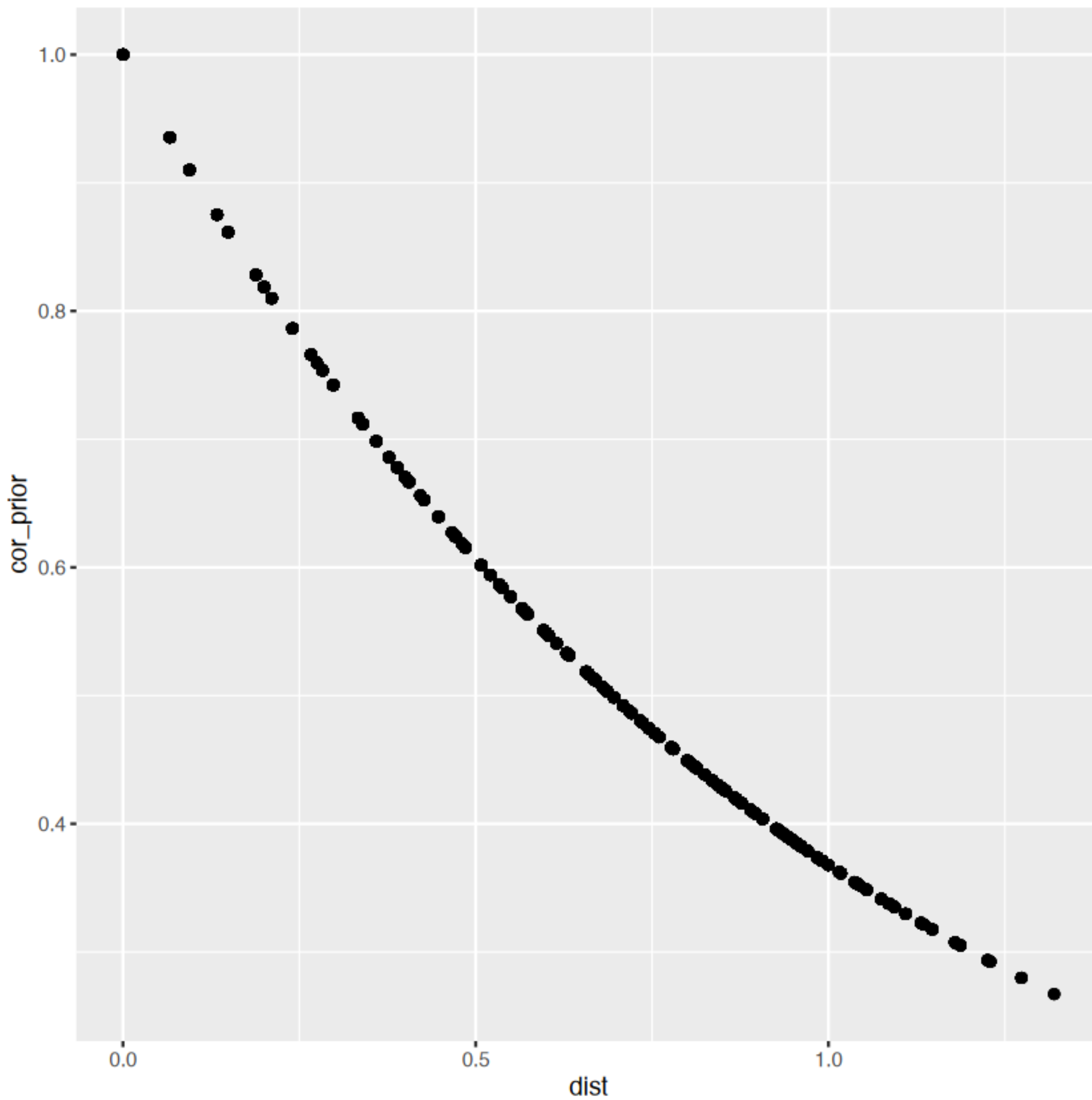


1.00
0.75
0.50
0.25
0.00
lat
0.00
0.25
0.50
0.75
1.00
long
full_cond_mean
0.0
-0.5
-1.0
-1.5

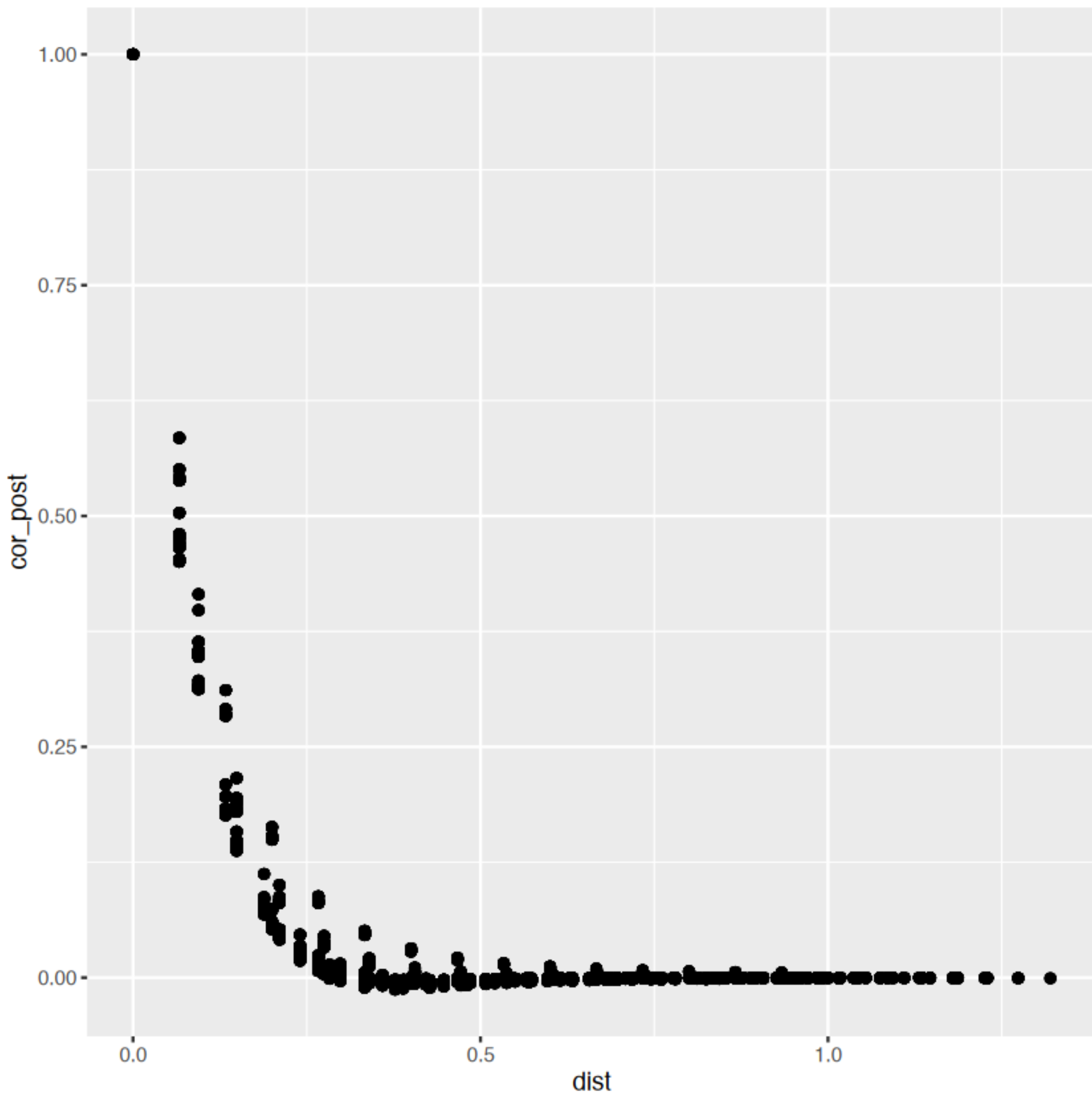


Adesso posso vedere come cambia la covarianza rispetto alla distanza

[100]:
```
data_plot_cov <- data.frame(dist = c(dist_mat), cor_prior = c(cov2cor(cov_w)),␣
↪cor_post = c(cov2cor(cov_w_post)))

data_plot_cov %>% ggplot(aes(x = dist, y = cor_prior)) +
  geom_point()

data_plot_cov %>% ggplot(aes(x = dist, y = cor_post)) +
  geom_point()
```



1.0
0.8
0.6
0.4
cor_prior
0.0
0.5
1.0
dist

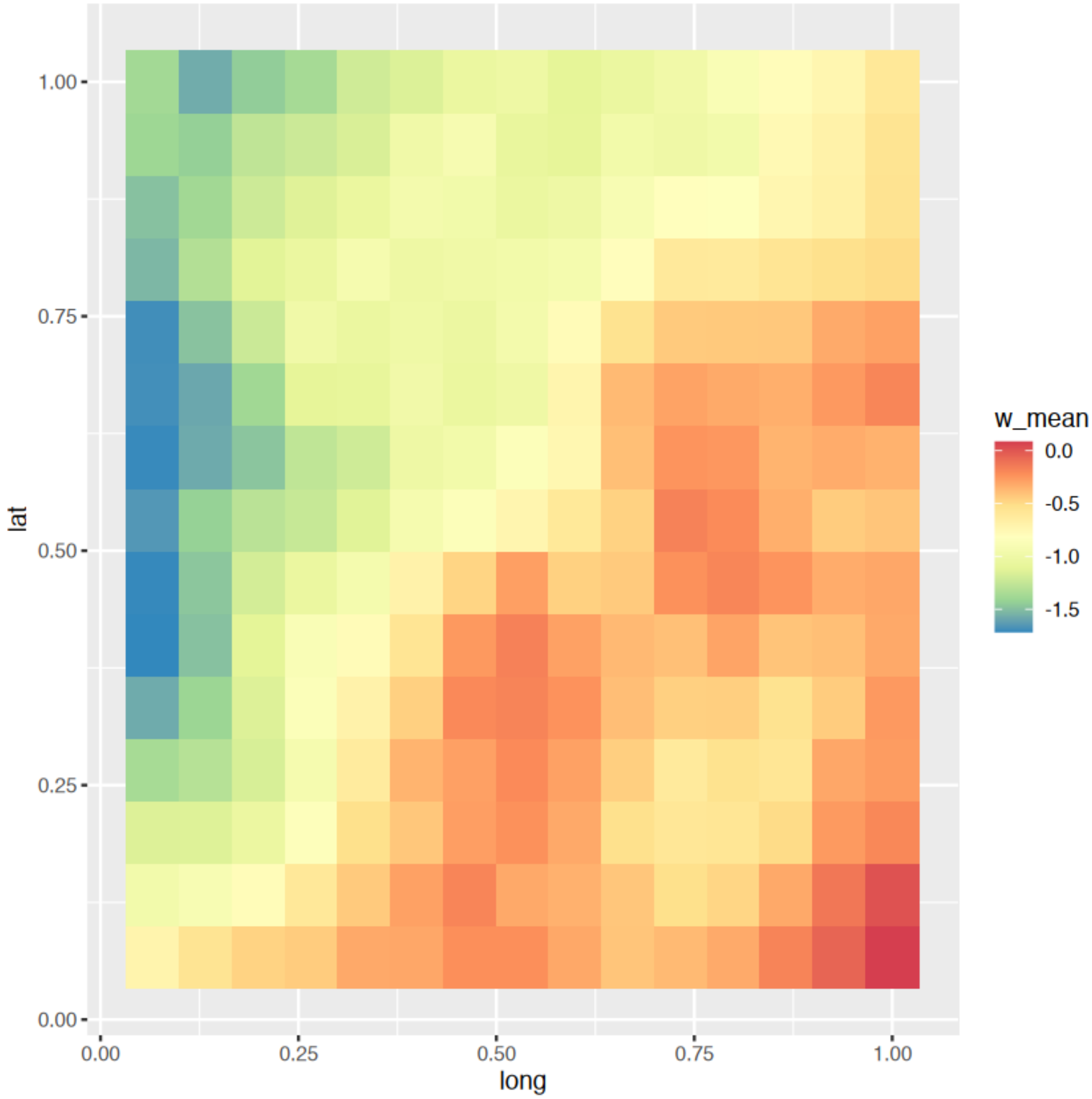


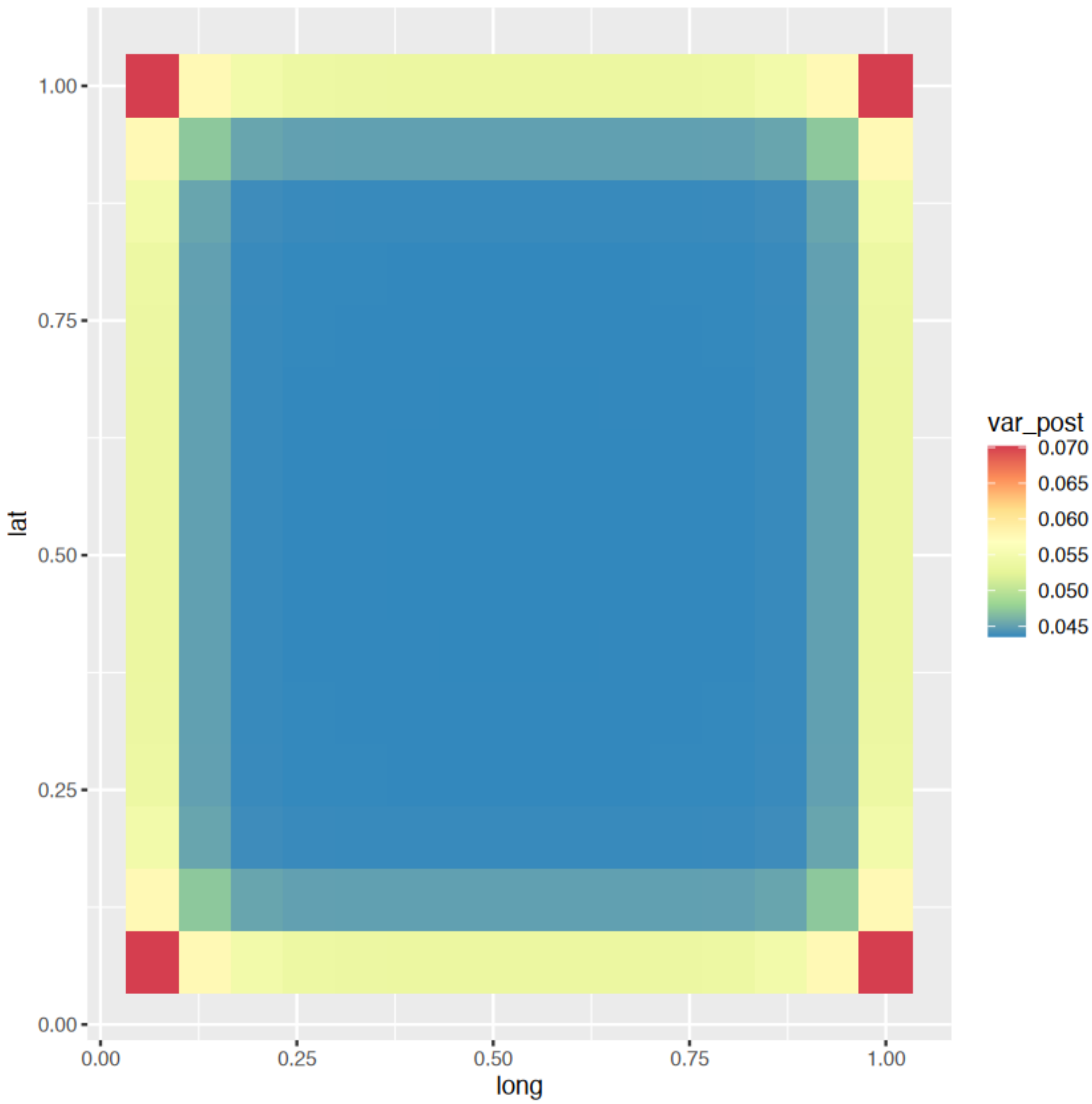
A prioi la covarianzaè stazionaria (a parità di distanze, abbiamo la stessa covarianza), mentre a posteriori questa cosa non è più vera. Inoltre c'è meno dipendenza visto che la covarianza va a zero molto più velocemente, è la variabilità viene spiegata dalla media che non è più zero, ma strutturata spazialmente

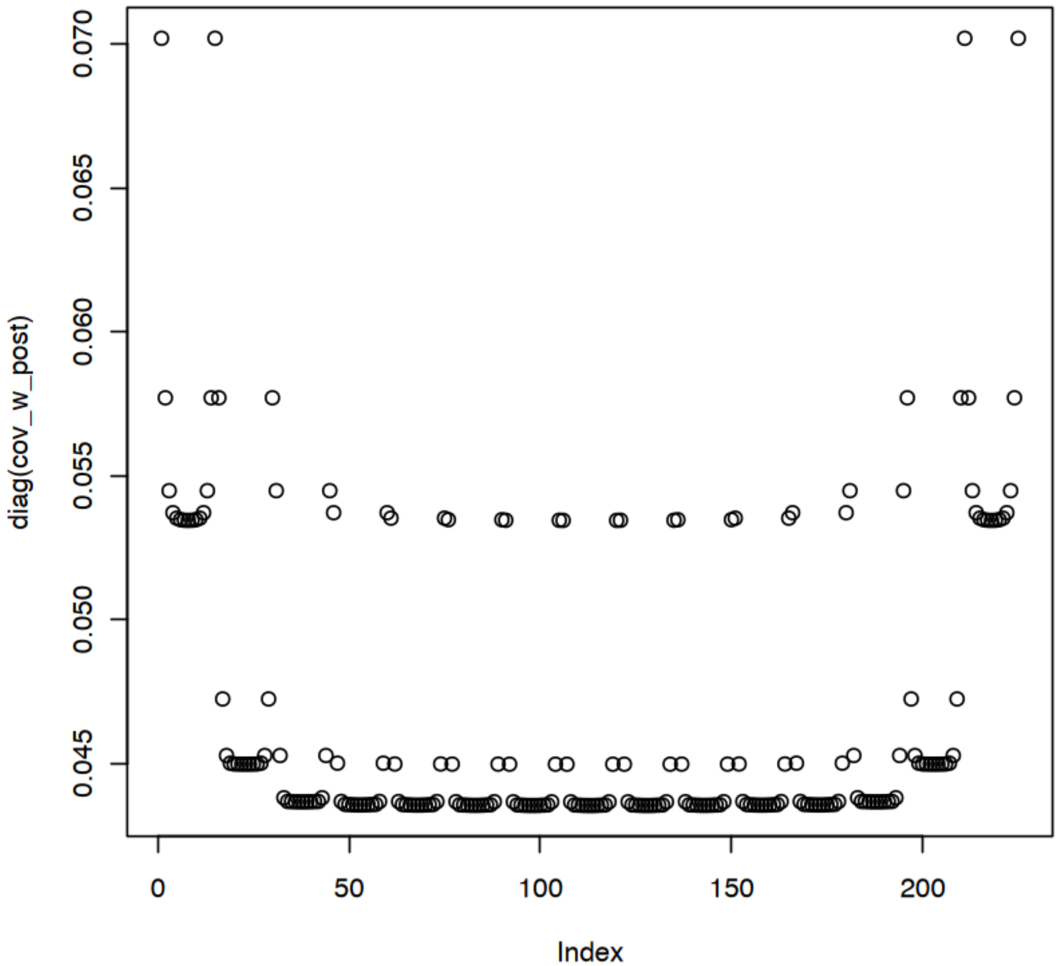
Per completezza facciamo anche un plot della varianza nello spazio

[101]:
```
data_sim <- data.frame(long = coords$long, lat = coords$lat, w = w_sim, y =␣
 ↪y_sim, w_mean = w_post_mean, var_post = diag(cov_w_post))
data_sim %>% ggplot(aes(x = long, y = lat, fill = w_mean)) +
  geom_tile() +
  scale_fill_distiller(palette = "Spectral")
data_sim %>% ggplot(aes(x = long, y = lat, fill = var_post)) +
  geom_tile() +
```

```
scale_fill_distiller(palette = "Spectral")
plot(diag(cov_w_post))
```




var_post
0.070
0.065
0.060
0.055
0.050
0.045
lat
1.00
0.75
0.50
0.25
0.00
long
0.00
0.25
0.50
0.75
1.00



1.2 Processo Spazio- Spaziale

Estendiamo l'esempio di prima con

$$Y(\mathbf{s},t)|W(\mathbf{s}) \overset{iid}{\sim} GP(\mu + W(\mathbf{s},t), \tau^2)$$

$$W(\mathbf{s},t) \sim GP(0, C(|\mathbf{s}-\mathbf{s}'|, |t-t'|\theta))$$

con

$$C(||\mathbf{s}-\mathbf{s}'||, |t-t'|\theta) = \sigma^2(\eta_1 \exp(-\phi_s||\mathbf{s}-\mathbf{s}'||)+\eta_2 \exp(-\phi_t|t-t'|)+\eta_3 \exp(-\phi_s||\mathbf{s}-\mathbf{s}'||)\exp(-\phi_t|t-t'|))$$

con

$$\eta_1 + \eta_2 + \eta_3 = 1 \qquad \eta_j > 0$$

2. Prendete dei valori ragionevoli per i decay ϕ_s e ϕ_t e simualte il processo spazio temporale sua una griglia spaziele di m punti e di g punti temporali. Poi plottati i GP per differenti tempi.

Adesso assumete

$$Y(\mathbf{s}, t) \overset{iid}{\sim} GP(\mu + W(\mathbf{s}, t), \tau^2)$$

con

$$W(\mathbf{s}, t) = W(\mathbf{s}) + W(t)$$

$$W(\mathbf{s}) \sim GP(0, C(||\mathbf{s} - \mathbf{s}'||; \theta))$$

e

$$W(t) \sim GP(0, C(|t - t'|; \theta))$$

entrambi con funzioni di cavarinza esponenziale

$$C(||\mathbf{s} - \mathbf{s}'||; \theta) = \sigma_1^2 \exp(-\phi_{sp}||\mathbf{s} - \mathbf{s}'||)$$

$$C(|t - t'|; \theta) = \sigma_1^2 \exp(-\phi_t|t - t'|)$$

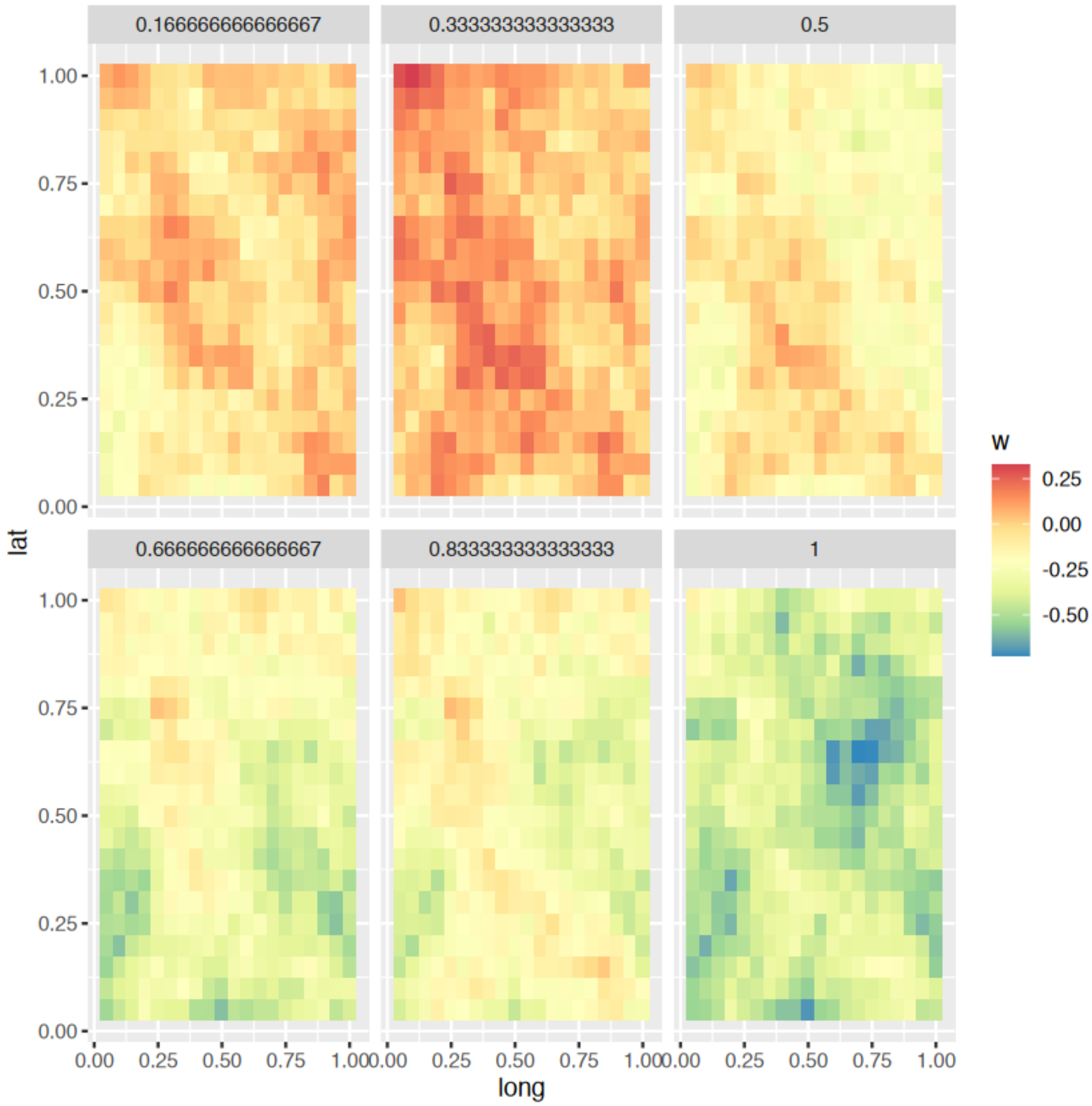
2. simulate $W(\mathbf{s}, t)$ e confrontate i risultati con il punto precedente

[102]:
```
m <- 20
g <- 6
n_sp = m^2
n <- n_sp * g
coords <- data.frame(long = rep(NA, n), lat = rep(NA, n), time = rep(NA, n))
h <- 1
for(itime in 1:g)
{
  for (i in 1:m)
  {
    for (j in 1:m)
    {
      coords[h, 1:2] <- c(i, j) / m
      coords[h, 3] <- itime / g
      h <- h + 1
    }
  }
}
```

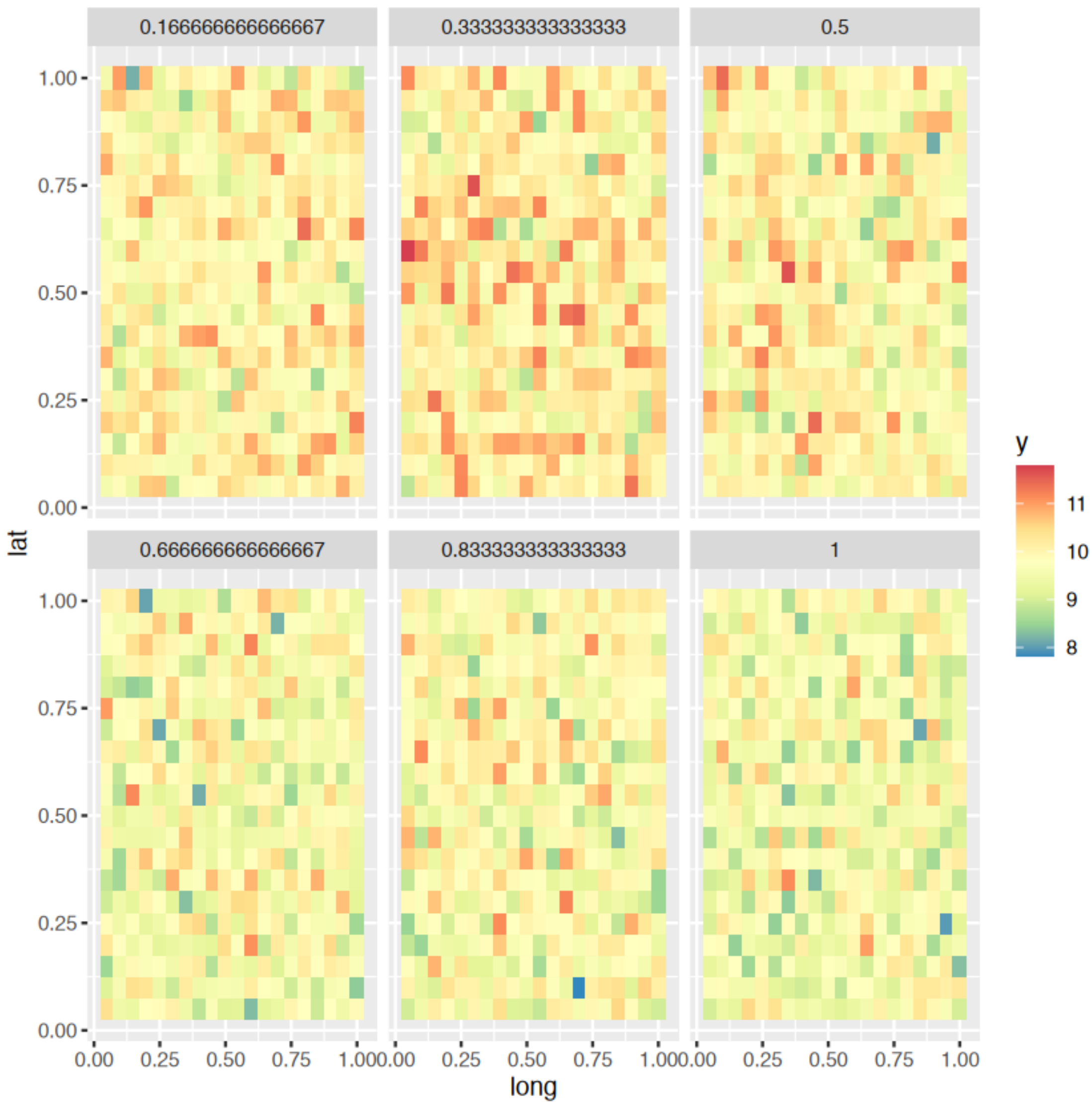
[103]:
```
set.seed(1)
mu <- 10
sigma2 <- 0.2
eta1 <- 0.25
eta2 <- 0.25
eta3 <- 0.5
tau2 <- 0.3
phi_sp <- 0.4
phi_t <- 0.8
mu <- 10
```

```r
dist_mat_sp <- as.matrix(dist(coords[,1:2]))
dist_mat_t <- as.matrix(dist(coords[,3]))

cov_w <- sigma2 * (eta1 * exp(-phi_sp * dist_mat_sp) + eta2 * exp(-phi_t *
 ↪dist_mat_t) + eta3* exp(-phi_sp * dist_mat_sp)* exp(-phi_t * dist_mat_t))

#cov_w <- sigma2 *   exp(-phi_t * dist_mat_t)
w_sim <- t(chol(cov_w)) %*% matrix(rnorm(n), ncol = 1)
y_sim <- mu + w_sim + rnorm(n, 0, tau2^0.5)
```

[104]:
```r
data_sim <- data.frame(long = coords$long, lat = coords$lat, time =
 ↪coords$time,  w = w_sim, y = y_sim)
data_sim %>%
ggplot(aes(x = long, y = lat, fill = w)) +
  geom_tile() +
  scale_fill_distiller(palette = "Spectral") + facet_wrap(~time )
data_sim %>%  ggplot(aes(x = long, y = lat, fill = y)) +
  geom_tile() +
  scale_fill_distiller(palette = "Spectral") + facet_wrap(~time )
```



0.166666666666667
0.333333333333333
0.5
0.666666666666667
0.833333333333333
1
lat
long
w
0.25
0.00
-0.25
-0.50

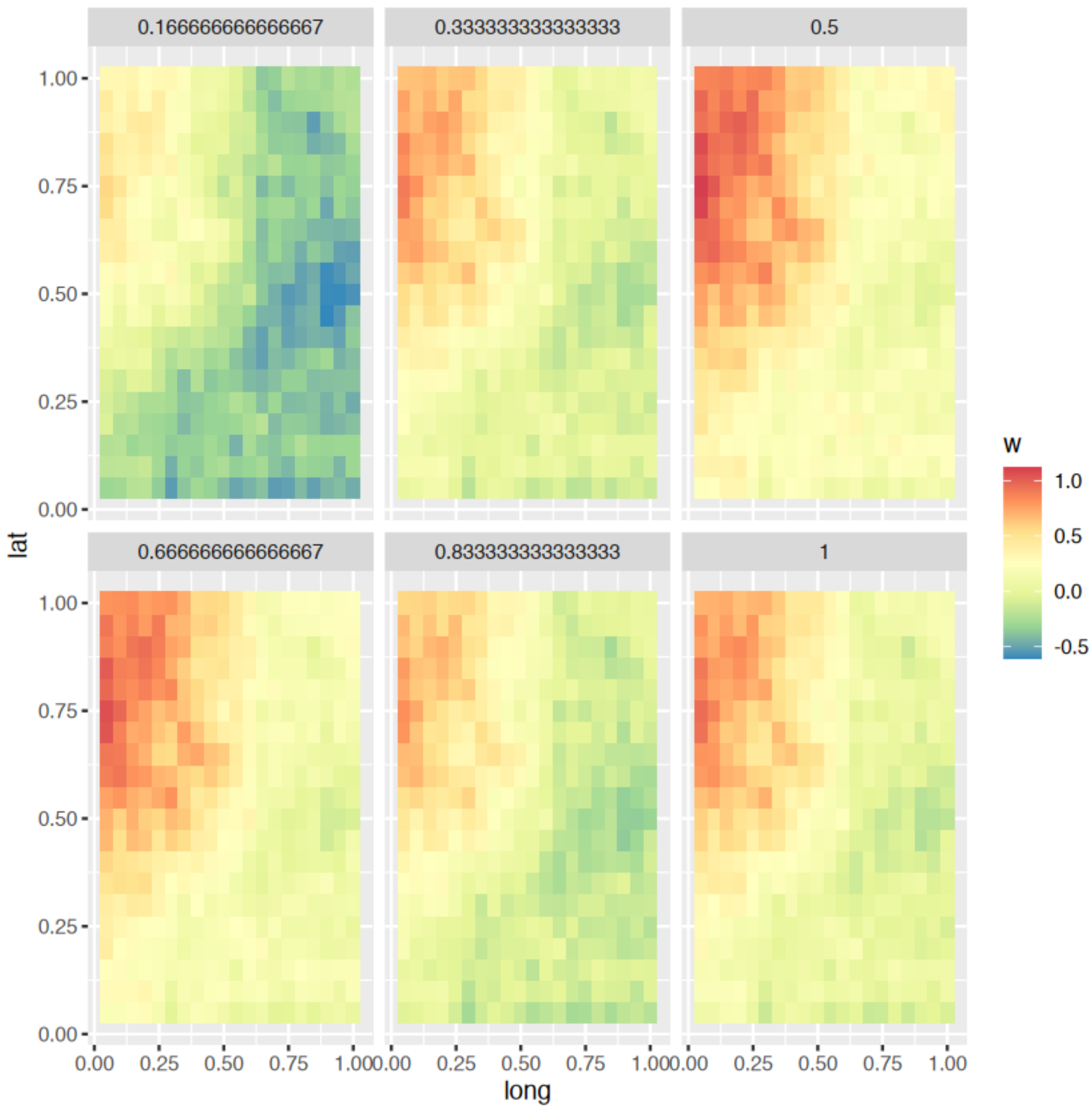


per il punto due simulo indipendentemente i due processi e poi li sommo. Dovete far attenzione che la funzione di covarianza èp definita positiva solo se due punti non hanno la stessa coordinata.

[105]:
```
m <- 20
g <- 6
n_sp <- m^2
n <- n_sp * g
coords_sp <- data.frame(long = rep(NA, n_sp), lat = rep(NA, n_sp))
coords_time <- data.frame(time = (1:g)/g)
h <- 1
for (i in 1:m)
{
  for (j in 1:m)
```

```r
  {
    coords_sp[h, 1:2] <- c(i, j) / m
    h <- h + 1
  }
}

mu <- 10
sigma2_sp <- 0.2
sigma2_time <- 0.2
eta1 <- 0.25
eta2 <- 0.25
eta3 <- 0.5
tau2 <- 0.3
phi_sp <- 0.4
phi_t <- 0.8
mu <- 10


dist_mat_sp <- as.matrix(dist(coords_sp))
dist_mat_time <- as.matrix(dist(coords_time))
cov_sp <- sigma2_sp*exp(-phi_sp * dist_mat_sp)
cov_time <- sigma2_time * exp(-phi_t * dist_mat_time)

#cov_w <- sigma2 * (eta1 * exp(-phi_sp * dist_mat_sp) + eta2 * exp(-phi_t *␣
↪dist_mat_t) + eta3 * exp(-phi_sp * dist_mat_sp) * exp(-phi_t * dist_mat_t))

## cov_w <- sigma2 *   exp(-phi_t * dist_mat_t)
w_sp_sim <- t(chol(cov_sp)) %*% matrix(rnorm(n_sp), ncol = 1)
w_time_sim <- t(chol(cov_time)) %*% matrix(rnorm(g), ncol = 1)

w_sim_2 = rep(w_sp_sim, times = g) + rep(w_time_sim, each = n_sp)
#y_sim <- mu + w_sim + rnorm(n, 0, tau2^0.5)
```

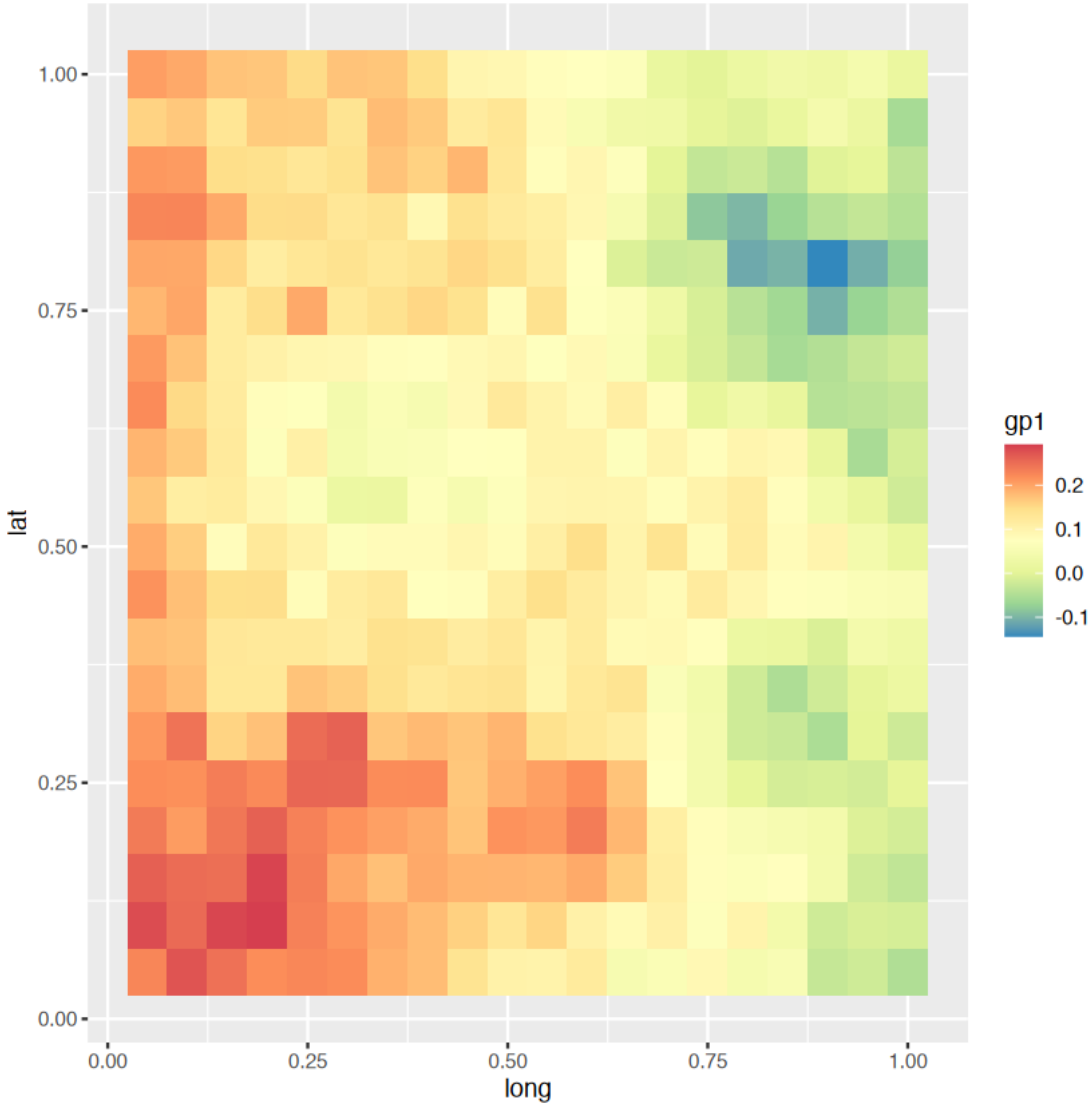
[106]:
```r
data_sim <- data.frame(long = rep(coords_sp$long, times = g), lat =␣
↪rep(coords_sp$lat, times = g), time = rep(coords_time$time, each = n_sp), w␣
↪= w_sim_2)
data_sim %>%
  ggplot(aes(x = long, y = lat, fill = w)) +
  geom_tile() +
  scale_fill_distiller(palette = "Spectral") +
  facet_wrap(~time)
```

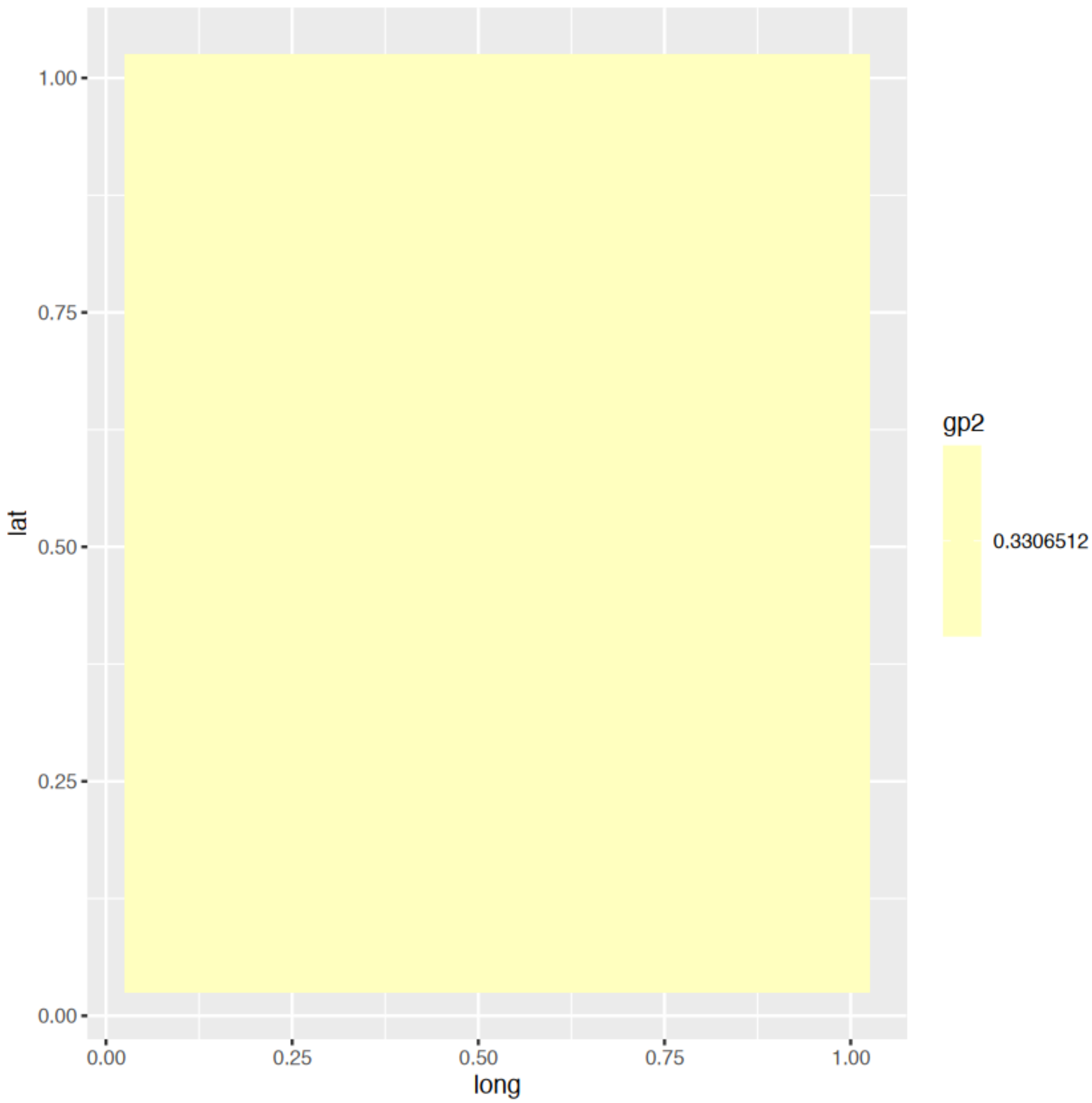


La differenza tra i due è che nel secondo caso, per ogni tempo il processo spaziale è lo stesso, solo con un valor medio diverso. Per vederlo, possiamo calcolare la differenza tra i GP spaziali tra due tempi consecutivi

[107]:
```
data_confronto <- data.frame(gp1 = w_sim[1:n_sp + n_sp] - w_sim[1:n_sp], gp2 =
  ↪w_sim_2[1:n_sp + n_sp] - w_sim_2[1:n_sp], long = coords_sp$long, lat =
  ↪coords_sp$lat)

data_confronto %>% ggplot(aes(x = long, y = lat, fill = gp1)) +
  geom_tile() +
  scale_fill_distiller(palette = "Spectral")

data_confronto %>% ggplot(aes(x = long, y = lat, fill = gp2)) +
```

```
  geom_tile() +
  scale_fill_distiller(palette = "Spectral")
```



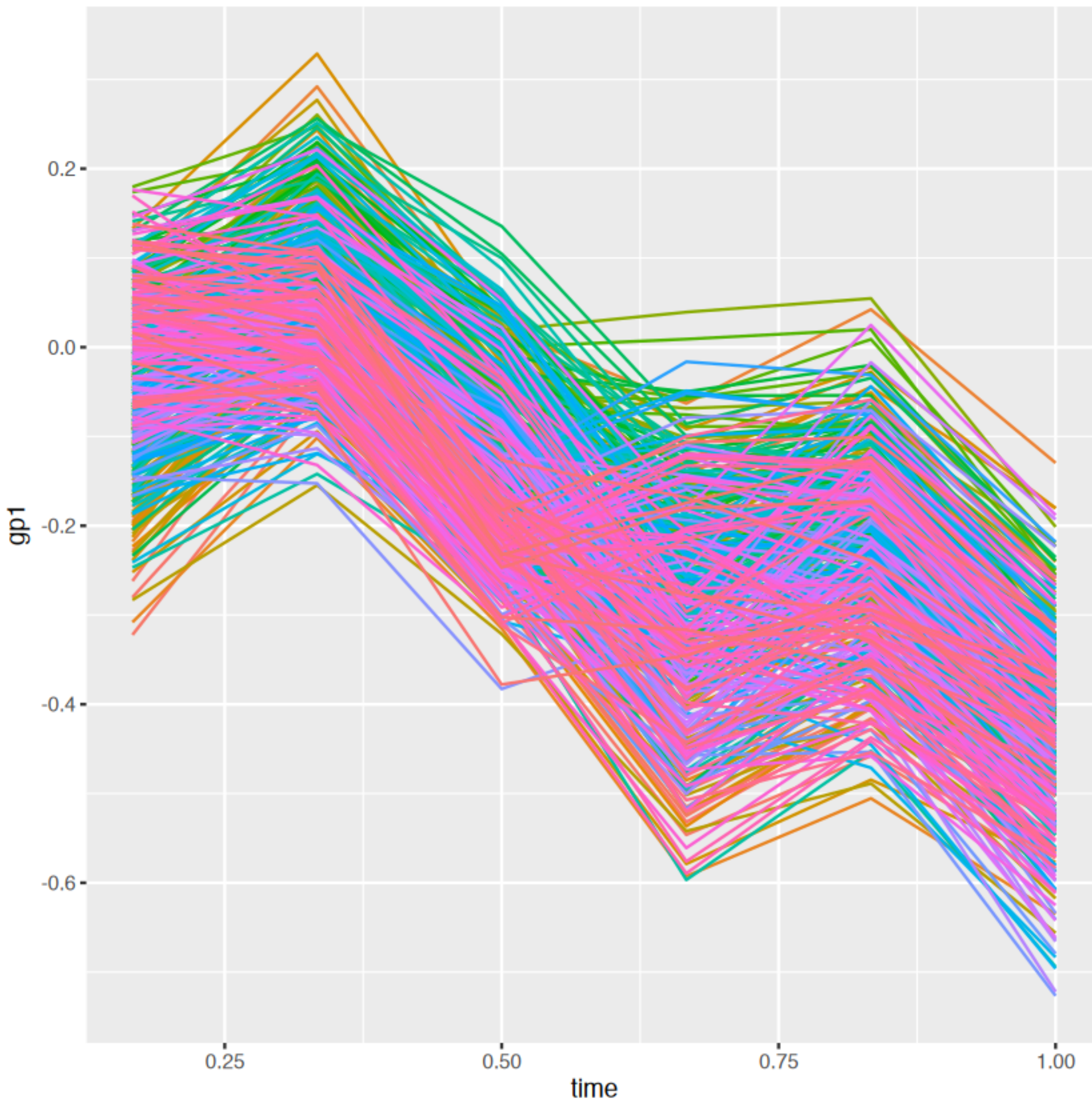


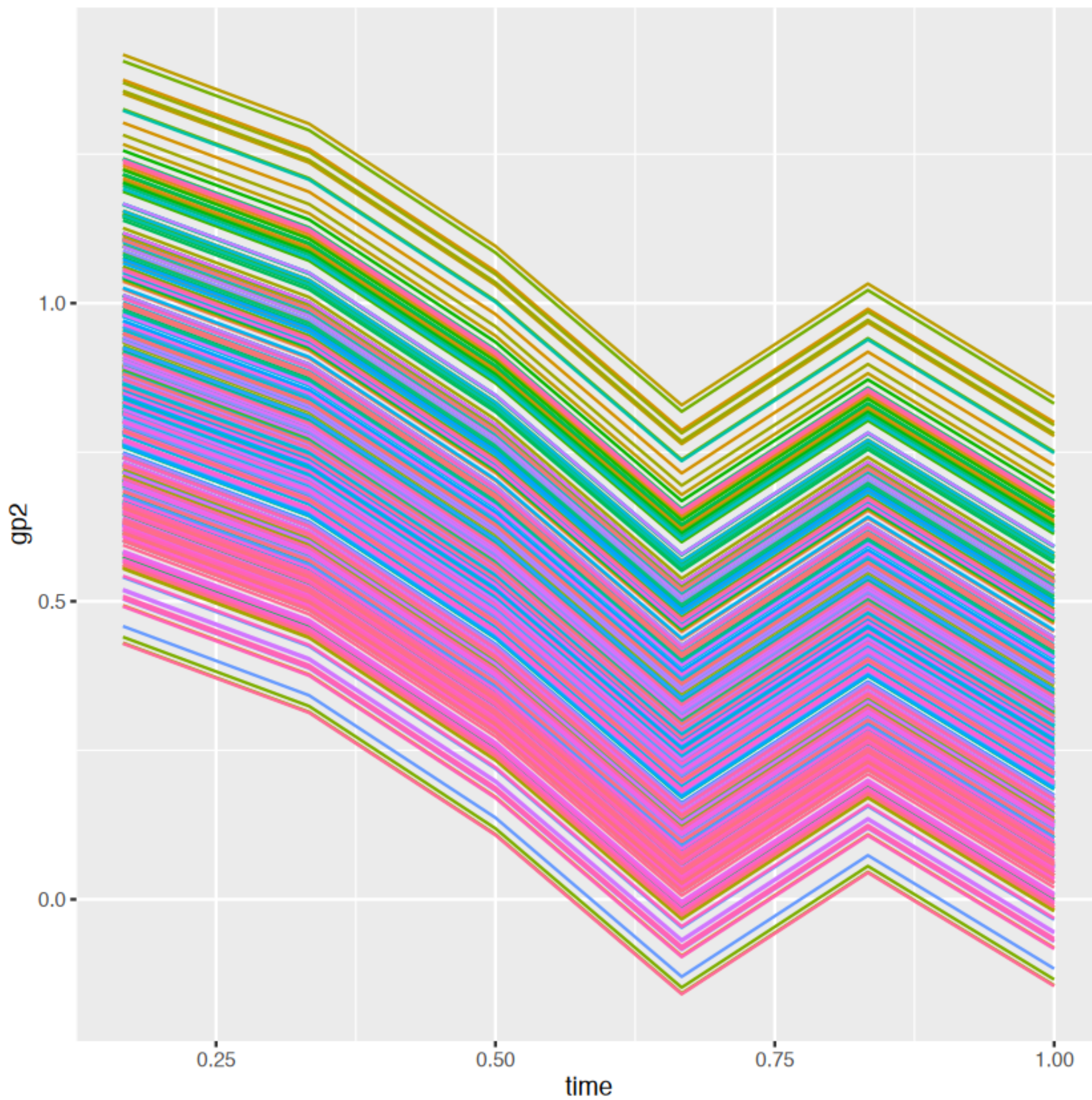
[108]:
```
Cosa simile se volesismo confrontare le serie storiche divise per sito spaziale.
↪ Per fare questo vediamo direttamente le serie storiche
```

```
Error in parse(text = input): <text>:1:6: simbolo inatteso
1: Cosa simile
         ^

Traceback:
```

[ ]:
```
data_sim <- data.frame(long = rep(coords_sp$long, times = g), lat =␣
↪rep(coords_sp$lat, times = g), time = rep(coords_time$time, each = n_sp),␣
↪gp2 = w_sim_2, gp1 = w_sim)
```

```r
data_sim %>%
  ggplot(aes(x = time, y  = gp1, col= paste(long, lat))) +
  geom_line() + theme(legend.position = "none")

data_sim %>%
  ggplot(aes(x = time, y = gp2, col = paste(long, lat))) +
  geom_line() +
  theme(legend.position = "none")
```




gp2
1.0
0.5
0.0
0.25
0.50
0.75
1.00
time