

# Trabajo Fin de Máster: Smart Bike - Sevilla.

## Anexo 2. Clasificación de estaciones y patrones espacio-temporales.

Máster en Data Science y Big Data - Universidad de Sevilla, 2016/2017.

*Jerónimo Carranza Carranza*

*1 de marzo de 2018*

### Índice

<b>1. Análisis de correlación entre estaciones</b>	<b>2</b>
<b>2. Clasificación de las estaciones</b>	<b>10</b>
<b>3. Patrones espacio-temporales</b>	<b>23</b>

### Índice de cuadros

1. Estructura <i>sevicip5m</i> . . . . .	2
2. Muestra de datos de bicicletas disponibles <i>sevicip5mBS</i> . . . . .	3

### Índice de figuras

1. Datos válidos estaciones. Matriz de correlación ( $ corr >0.5$ ) entre estaciones. . . . .	4
2. Datos válidos estaciones. Grafo espacial de correlaciones $ corr > 0.5$ . . . . .	5
3. Datos válidos estaciones. Suma de correlaciones positivas por estación ( $corr>0.5$ ). . . . .	7
4. Datos válidos estaciones. Suma de correlaciones negativas por estación ( $corr<-0.5$ ). . . . .	9
5. Datos válidos estaciones. Dendrograma de estaciones basado en correlación. . . . .	20
6. Datos válidos estaciones. Clasificación de estaciones. . . . .	22
7. Datos estaciones. %Bicis disponibles por clase de estación, hora del día y día de la semana. .	25
8. Datos estaciones. %Bicis disponibles por hora del día y día de la semana. Patrones por clase de estación. . . . .	27
9. Datos estaciones. Bicis disponibles por clase de estación, hora del día y día de la semana. Coeficiente de Variación (%). . . . .	28
10. Datos estaciones. Bicis disponibles por hora del día y día de la semana. Patrones por clase de estación. Coeficiente de Variación (%). . . . .	30
11. Datos estaciones. %Bicis disponibles por hora del día y día de la semana. Muestra de estaciones por clase. . . . .	33

```
library(RPostgreSQL)
library(tidyverse)
library(tidyr)
library(dplyr)
library(dbplyr)
library(knitr)
library(sp)
library(sf)
library(ggplot2)
library(ggcrrplot)
library(ggspatial)
library(lubridate)
library(scales)
library(factoextra)
library(reshape2)
library(igraph)
library(ggraph)
library(ggdendro)
```

```
set.seed(12345)
```

## 1. Análisis de correlación entre estaciones

Para el análisis de correlación y posteriores partimos de *sevicip5m*, este dataframe tiene las 105132 filas correspondientes a los periodos de cinco minutos entre inicio y fin y las 522 columnas correspondientes a:

Cuadro 1: Estructura *sevicip5m*

Variable	Descripción
p5min	Periodo de 5min (Datetime)
hueco	Hueco global (Boolean)
si	Estacionamientos disponibles estación i
bi	Bicicletas disponibles estación i
:	para i en 1:260

Se ha utilizado anteriormente *sevicip5m* para el análisis de huecos, excluyendo los datos duplicados, pero no todos los casos no válidos, en particular, para ‘ok in 3:5’. Para anular dichos casos se implementa el siguiente procedimiento:

```
# Se dispone en una fila el máximo número de estacionamientos por
# estación
seviesta_MaxS = resumen_datos_por_estacion[, 8]

# Se extrean las columnas de estacionamientos y bicicletas disponibles
# Se trasponen (=> a matriz) para permitir la comparación posterior: el
# vector se compara con la matriz columna por columna
sevicip5m_B = t(sevicip5m %>% select(starts_with("b")))
sevicip5m_S = t(sevicip5m %>% select(starts_with("s")))

# Se identifican los casos no válidos con -1 matriz auxiliar
sevicip5mL = (sevicip5m_B + sevicip5m_S > seviesta_MaxS) * -1
```

```

# Se anulan los casos no válidos en matriz auxiliar
sevicip5mL[sevicip5mL == -1] = NA

# Se anulan en estacionamientos y bicicletas disponibles los casos no
# válidos y se re-traspone
sevicip5mS = t(sevicip5m_S + sevicip5mL)
sevicip5mB = t(sevicip5m_B + sevicip5mL)

# Se anteponen las columnas p5min y hueco
sevicip5mBS <- bind_cols(sevicip5m[, 1:2], as.data.frame(sevicip5mB), as.data.frame(sevicip5mS))
# Se ordena temporalmente
sevicip5mBS <- sevicip5mBS %>% arrange(p5min)

# Limpia
rm(sevicip5m_B)
rm(sevicip5m_S)
rm(sevicip5mL)
rm(sevicip5mB)
rm(sevicip5mS)

```

Seguidamente una muestra de los datos de bicicletas disponibles en *sevicip5mBS*

```
kable(sevicip5mBS[, c(1:10, 261, 262, 521, 522)] %>% sample_n(10), caption = "Muestra de datos de bicicletas")
```

Cuadro 2: Muestra de datos de bicicletas disponibles *sevicip5mBS*.

	p5min	hueco	b1	b2	b3	b4	b5	b6	b7	b8	b259	b260	s259	s260
75791	2016-08-20 03:50:00	FALSE	19	1	9	13	35	6	7	10	3	10	27	7
92071	2016-10-15 16:30:00	FALSE	8	1	2	3	1	4	0	4	1	2	29	17
80003	2016-09-03 18:50:00	FALSE	6	8	26	31	23	3	17	2	3	18	27	2
93158	2016-10-19 11:05:00	FALSE	6	1	23	38	34	19	20	15	2	2	28	18
47989	2016-05-15 15:00:00	FALSE	10	0	11	22	3	2	12	3	0	16	30	4
17491	2016-01-30 17:30:00	FALSE	5	0	5	0	9	4	4	2	7	1	22	19
34176	2016-03-28 15:55:00	FALSE	13	18	7	18	37	16	4	2	20	2	10	18
53533	2016-06-03 21:00:00	FALSE	NA	NA	NA	NA								
76500	2016-08-22 14:55:00	FALSE	8	10	16	20	38	17	9	17	2	6	28	14
104045	2016-11-26 06:20:00	FALSE	12	3	2	12	7	1	0	0	7	11	23	9

Calculamos la matriz de correlación (Pearson) entre estaciones para la variable número de bicicletas disponibles.

```
sevici_bcorr = cor(select(sevicip5mBS, starts_with("b")), use = "pairwise")
```

La matriz de correlación obtenida la segregamos en un dataframe con todos los pares y el valor de correlación para su tratamiento posterior como grafo (con nodos geoposicionados).

```

sevici_bcorr_melted = tbl_df(melt(sevici_bcorr))
sevici_bcorr_melted <- mutate(sevici_bcorr_melted, from = as.integer(substr(Var1,
    2, 5)), to = as.integer(substr(Var2, 2, 5)))
sevici_bcorr_melted <- sevici_bcorr_melted[, c(4, 5, 3, 1, 2)]

```

```

sevici_bcorr_melted <- sevici_bcorr_melted %>% rename(corr = value)
sevici_bcorr_melted

```

```
## # A tibble: 67,600 x 5
```

```
##      from      to      corr Var1  Var2
##      <int> <int>    <dbl> <fct> <fct>
##  1      1       1  1.00   b1    b1
##  2      2       1  0.0982  b2    b1
##  3      3       1  0.0710  b3    b1
##  4      4       1 -0.0696  b4    b1
##  5      5       1 -0.0120  b5    b1
##  6      6       1 -0.161   b6    b1
##  7      7       1 -0.143   b7    b1
##  8      8       1 -0.118   b8    b1
##  9      9       1 -0.143   b9    b1
## 10     10      1  0.225   b10   b1
## # ... with 67,590 more rows
sevici_bcorr_melted %>% filter(abs(corr) > 0.5) %>% ggplot(aes(x = from,
  y = to, fill = corr)) + scale_fill_gradientn(colors = rainbow(6)) +
  geom_tile()
```

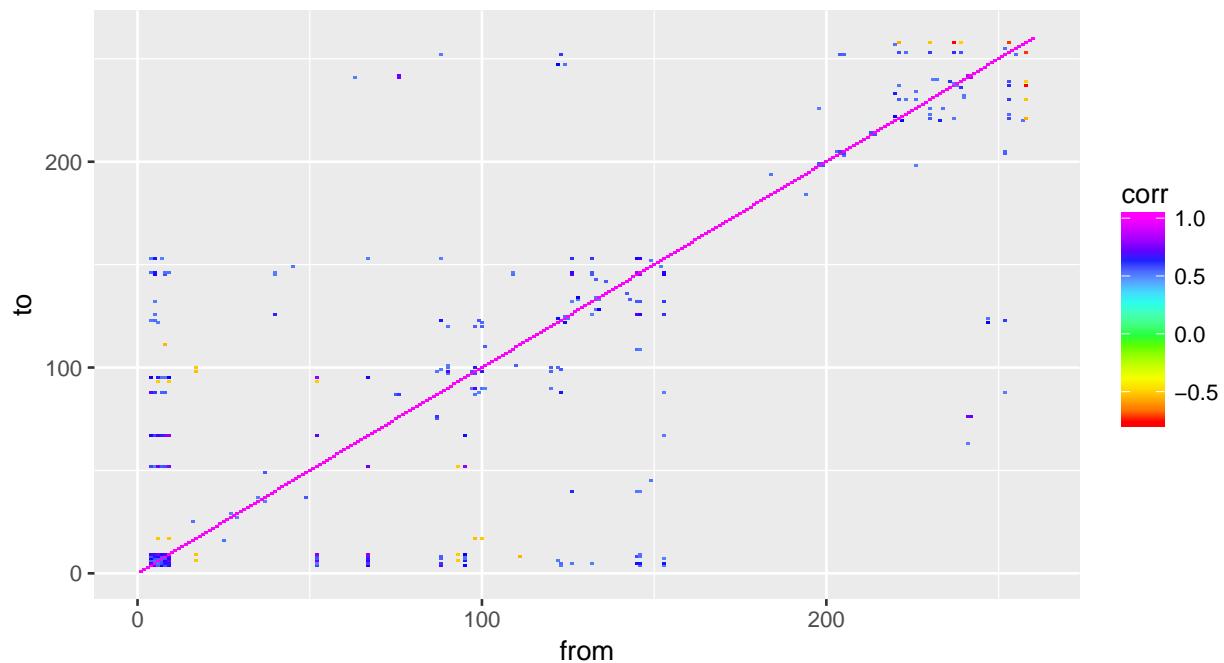


Figura 1: Datos válidos estaciones. Matriz de correlación ( $|corr|>0.5$ ) entre estaciones.

```
g <- sevici_bcorr_melted %>% filter(abs(corr) > 0.5) %>% filter(to > from) %>%
  graph_from_data_frame(directed = FALSE, vertices = seviesta)

g$layout = cbind(V(g)$longitude, V(g)$latitude)

ggraph(g, fullpage = TRUE) + geom_osm(type = "cartolight", quiet = TRUE) +
  geom_node_point(color = "black", size = 0.5, alpha = 0.5) + geom_edge_arc(aes(color = corr),
  edge_alpha = 0.6, curvature = 0.2) + scale_edge_color_gradient2(low = "red",
  high = "blue", mid = "white", midpoint = 0) + labs(x = "", y = "") +
  coord_map() + theme_bw()
```

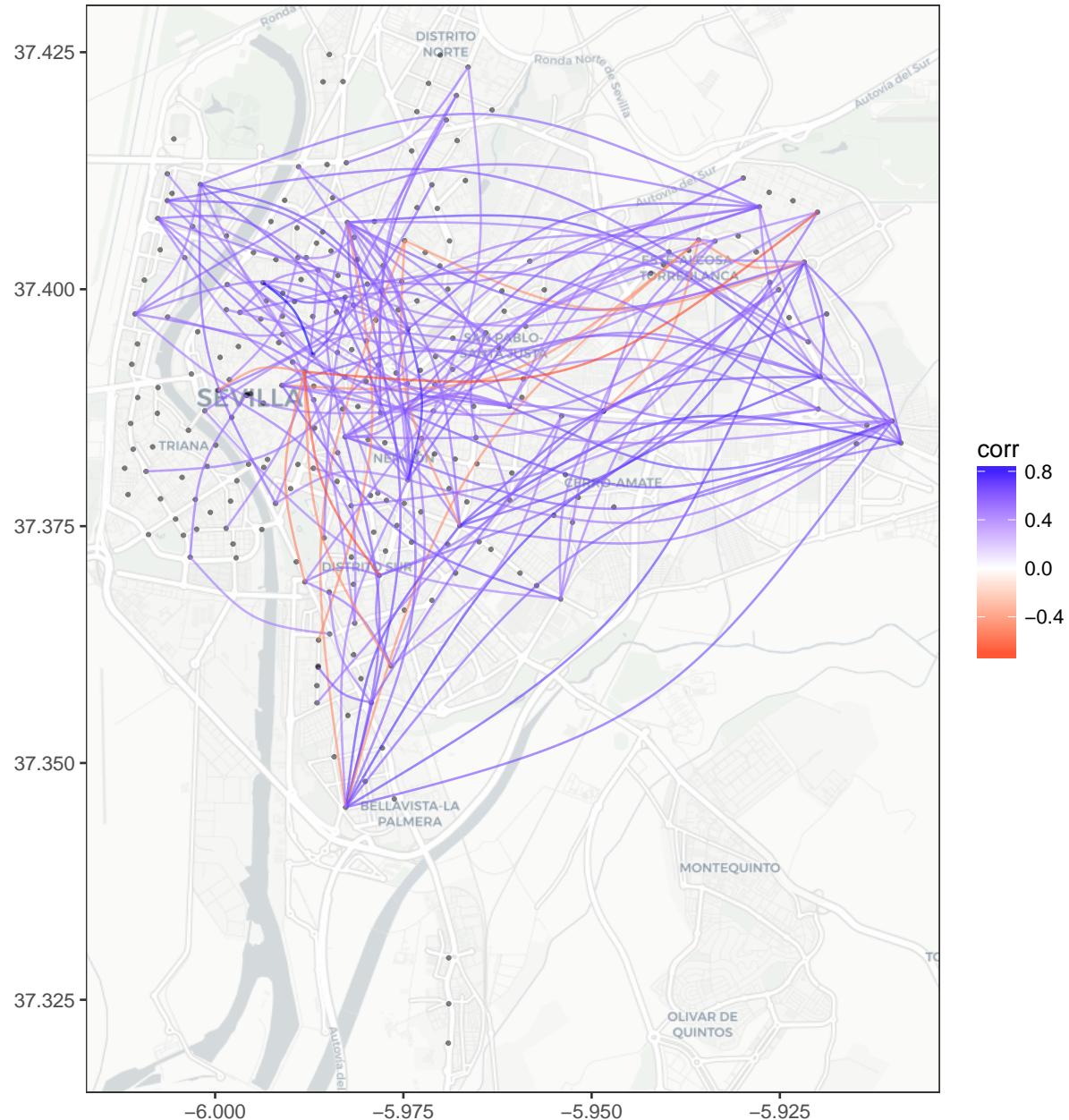


Figura 2: Datos válidos estaciones. Grafo espacial de correlaciones  $|\text{corr}| > 0.5$ .

```
sumcorrp <- sevici_bcorr_melted %>% filter(corr > 0.5) %>% filter(corr <  
1) %>% group_by(from) %>% summarise(sump = sum(corr)) %>% select(num = from,  
sump) %>% inner_join(seviesta, by = "num")  
  
sumcorrн <- sevici_bcorr_melted %>% filter(corr < -0.5) %>% group_by(from) %>%  
summarise(sumn = sum(corr)) %>% select(num = from, sumn) %>% inner_join(seviesta,  
by = "num")  
  
ggraph(g, fullpage = TRUE) + geom_osm(type = "cartolight", quiet = TRUE) +  
geom_node_point(color = "black", size = 0.5, alpha = 0.5) + geom_edge_arc(aes(color = corr),  
edge_alpha = 0.3, curvature = 0.2) + scale_edge_color_gradient2(low = "red",  
high = "blue", mid = "white", midpoint = 0) + geom_point(data = sumcorrp,  
aes(longitude, latitude, color = sump), size = 2.5, alpha = 0.5) +  
scale_color_gradientn(colours = c("green", "cyan", "blue")) + labs(x = "",  
y = "") + coord_map() + theme_bw()
```

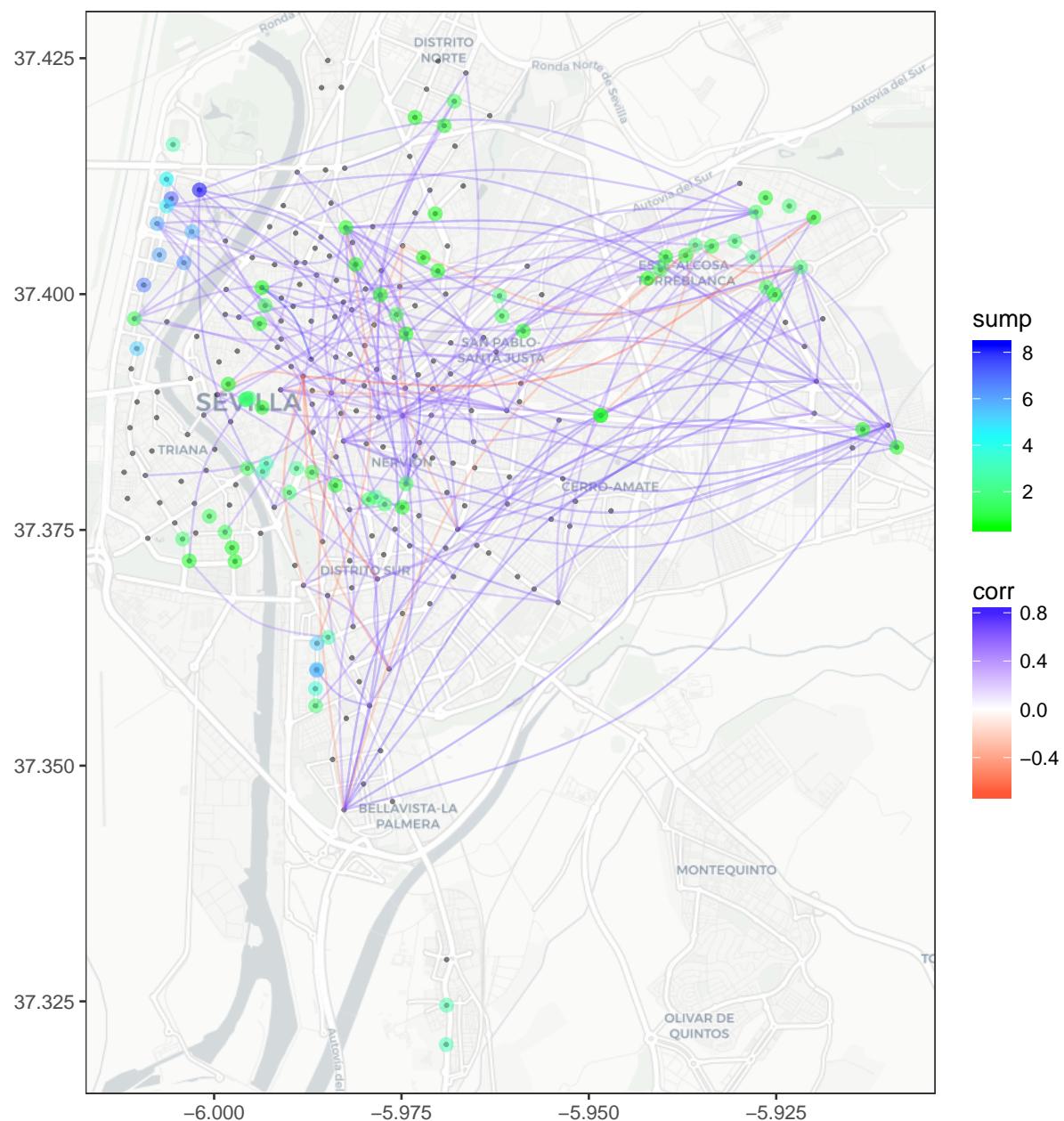


Figura 3: Datos válidos estaciones. Suma de correlaciones positivas por estación ( $\text{corr} > 0.5$ ).

```
ggraph(g, fullpage = TRUE) + geom_osm(type = "cartolight", quiet = TRUE) +
  geom_node_point(color = "black", size = 0.5, alpha = 0.5) + geom_edge_arc(aes(color = corr),
  edge_alpha = 0.3, curvature = 0.2) + scale_edge_color_gradient2(low = "red",
  high = "blue", mid = "white", midpoint = 0) + geom_point(data = sumcorr,
  aes(longitude, latitude, color = sumn), size = 2.5, alpha = 0.4) +
  scale_color_gradientn(colours = c("red", "orange", "green")) + labs(x = "",
  y = "") + coord_map() + theme_bw()
```

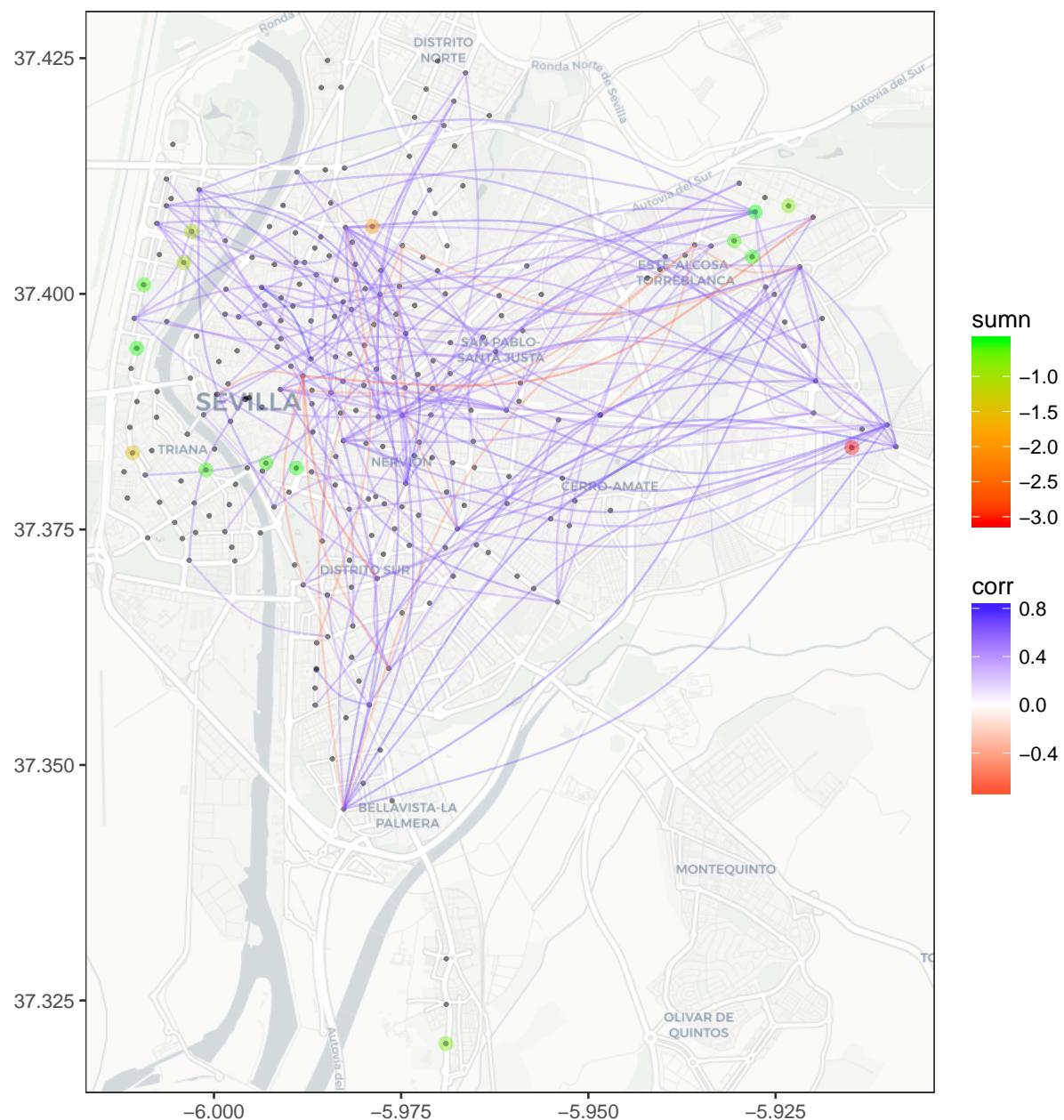


Figura 4: Datos válidos estaciones. Suma de correlaciones negativas por estación ( $\text{corr} < -0.5$ ).

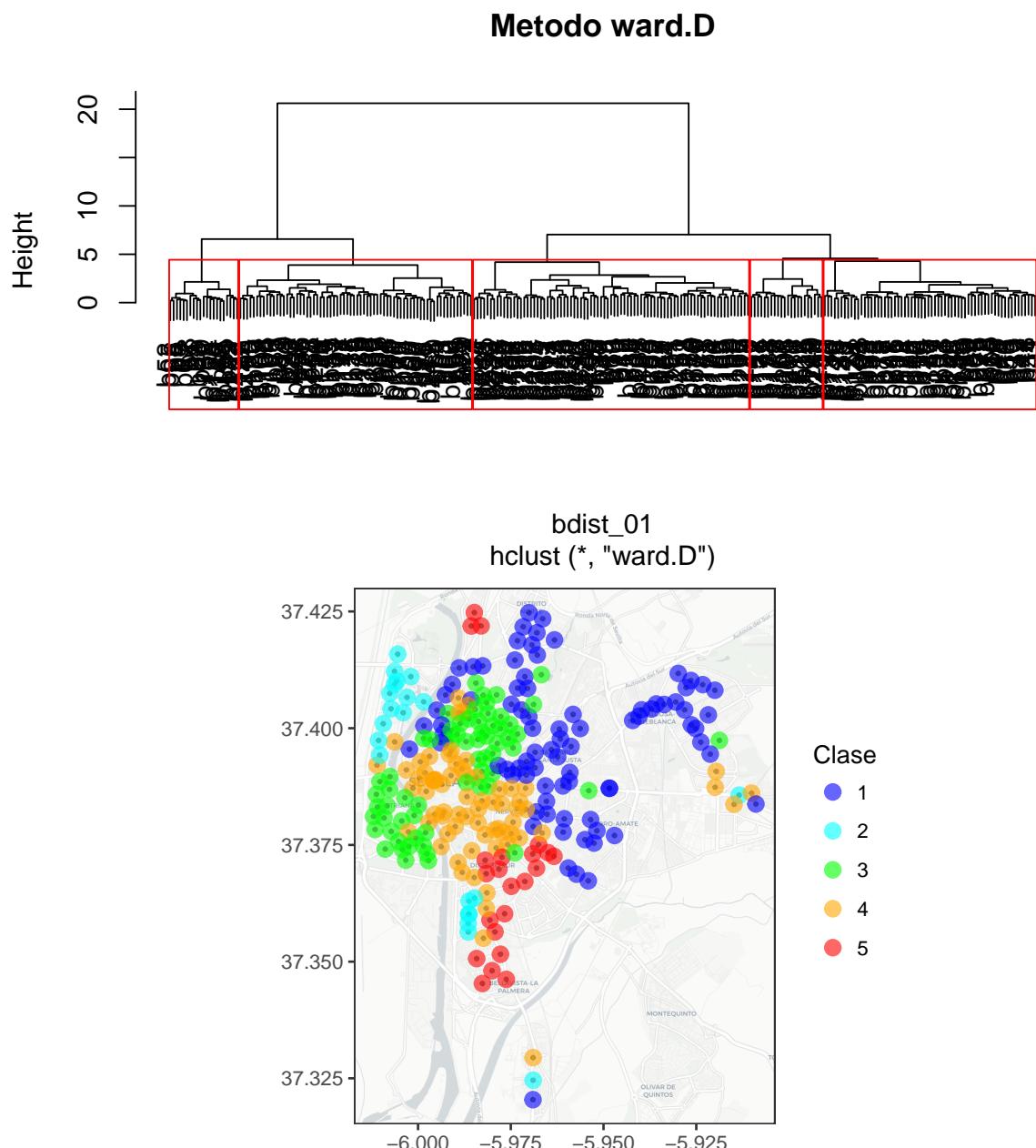
## 2. Clasificación de las estaciones

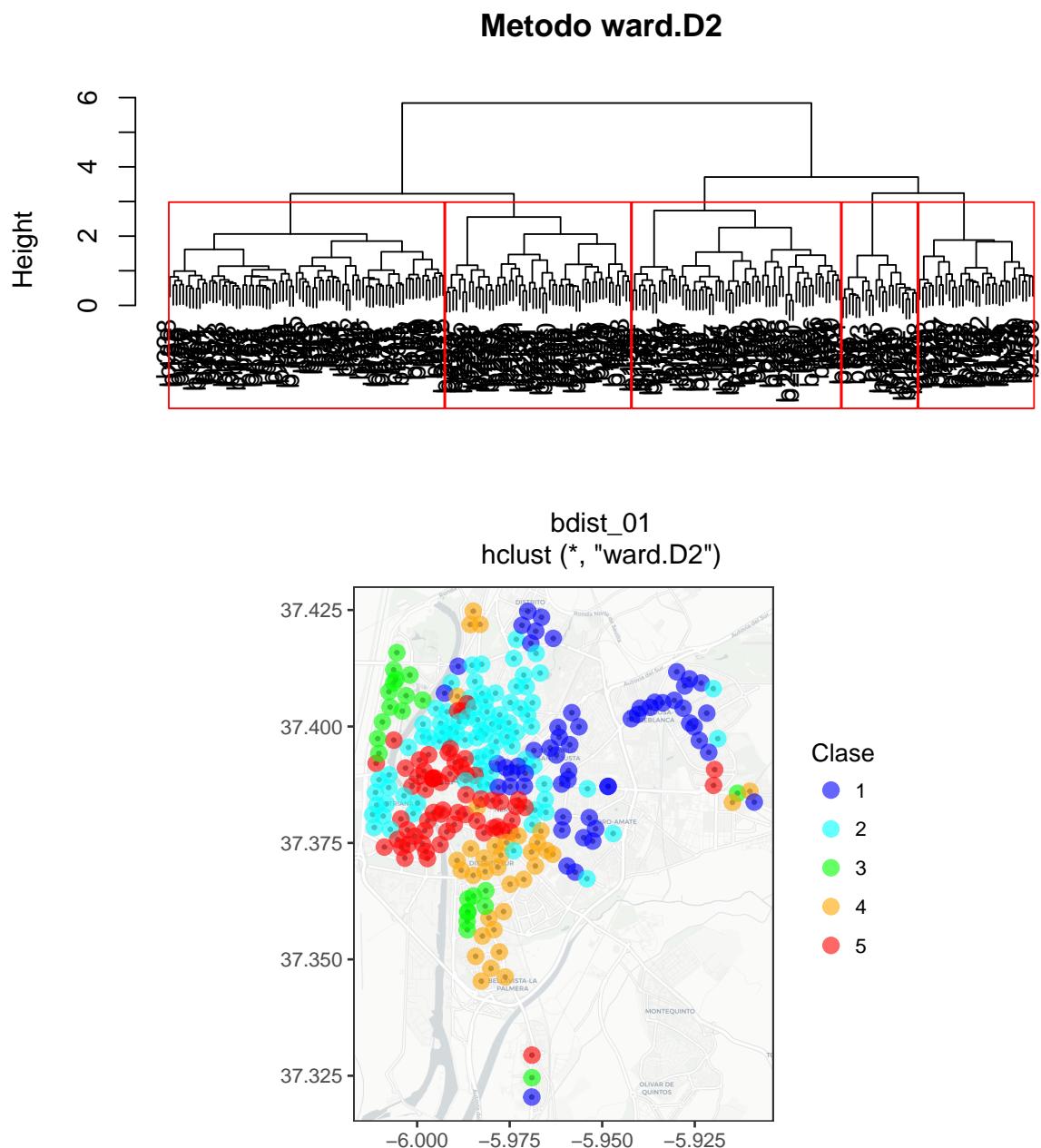
Utilizamos la matriz de correlación como base para la clasificación de las estaciones. Para ello en primer lugar convertimos los coeficientes de correlación en disimilaridades y éstas son tratadas como distancias.

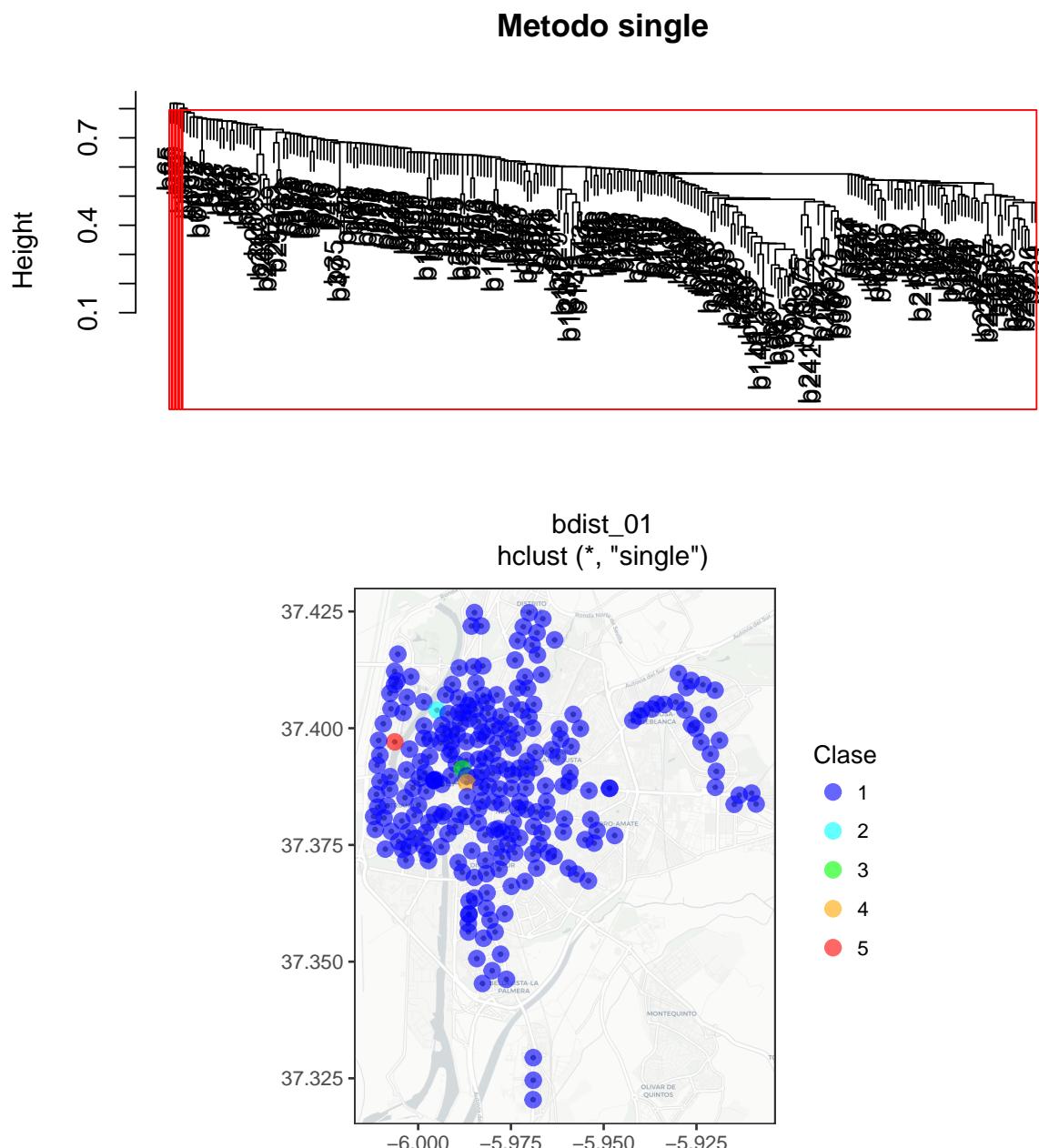
```
bdisim_01 = 1 - sevici_bcorr  
bdist_01 = as.dist(bdisim_01)
```

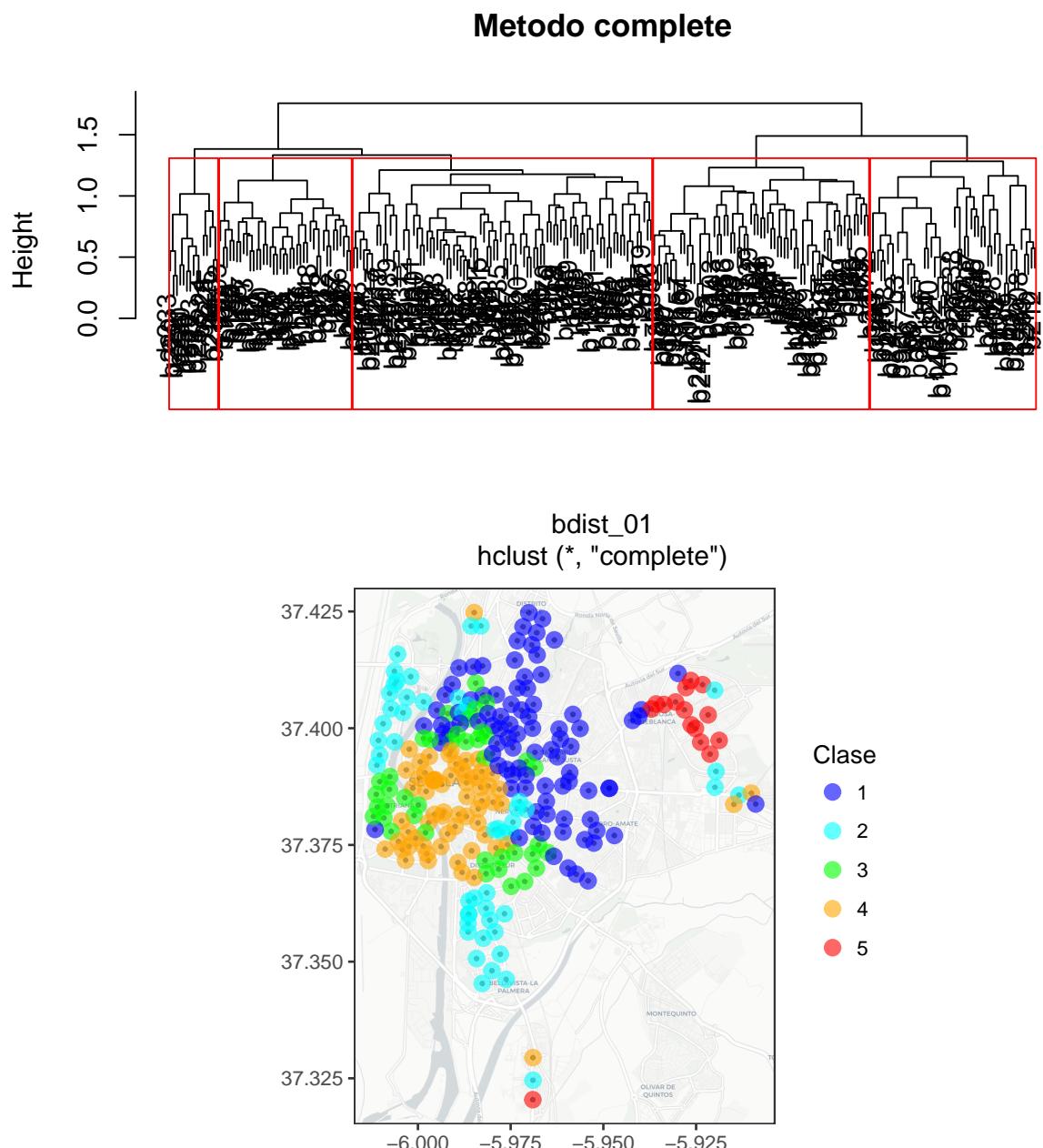
Probamos los métodos de agregación disponibles en *hclust* mostrando sus resultados en forma de dendrograma y su expresión espacial para la segmentación en 5 clases.

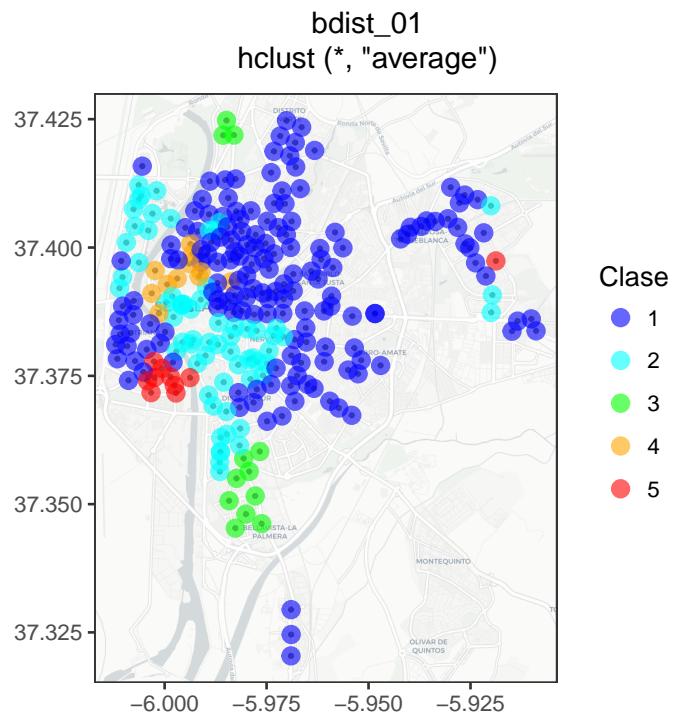
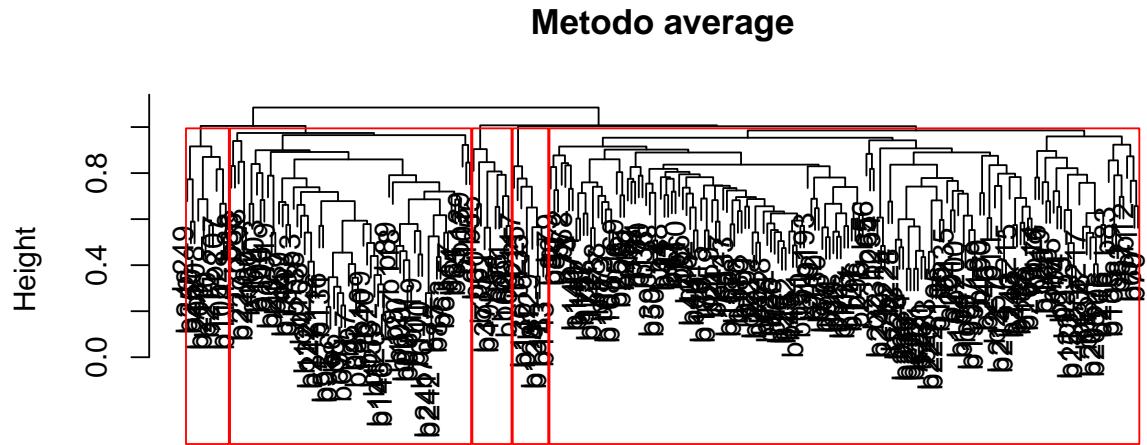
```
metodos.hclust = c("ward.D", "ward.D2", "single", "complete", "average",  
"mcquitty","median","centroid")  
  
for (m in metodos.hclust){  
  clus = hclust(bdist_01, method = m)  
  plot(clus, main=paste0("Metodo ",m))  
  rect.hclust(clus,k=5)  
  clus_class = tbl_df(cutree(clus, k = 5))  
  clus_class <- seviesta %>% arrange(num) %>% bind_cols(clus_class)  
  
  pltclas = ggraph(g, fullpage = TRUE) +  
    geom_osm(type = 'cartolight', quiet = TRUE) +  
    geom_node_point(color='black', size=0.5, alpha=0.5) +  
    geom_point(data=clus_class, aes(longitude, latitude,  
                                    color=as.factor(value)),  
               size=3, alpha=0.6) +  
    scale_color_manual(name = 'Clase',  
                      values = c('blue'  
                                , 'cyan'  
                                , 'green'  
                                #, 'dark green'  
                                #, 'yellow'  
                                , 'orange'  
                                , 'red'  
                                , 'brown'  
                                , 'black'  
                                )) +  
    labs(x="",y="") + coord_map() + theme_bw()  
  print(pltclas)  
}  
}
```

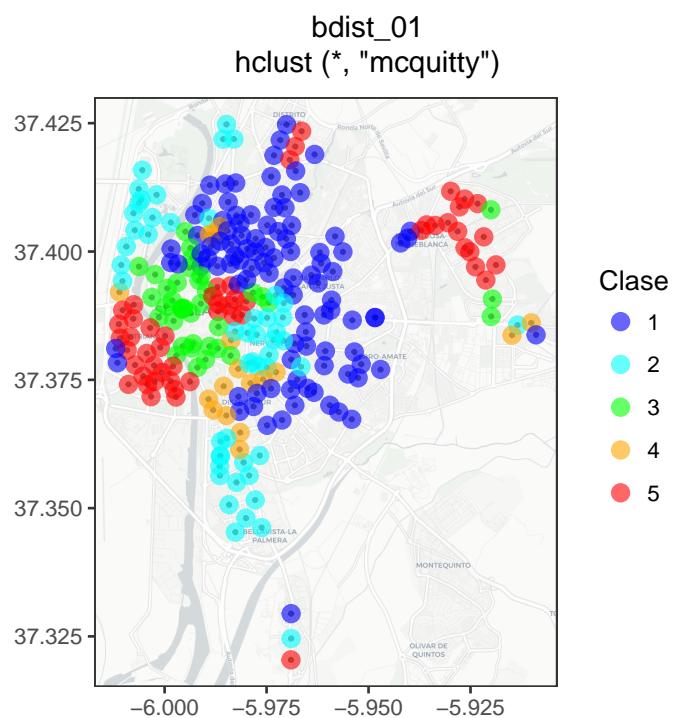
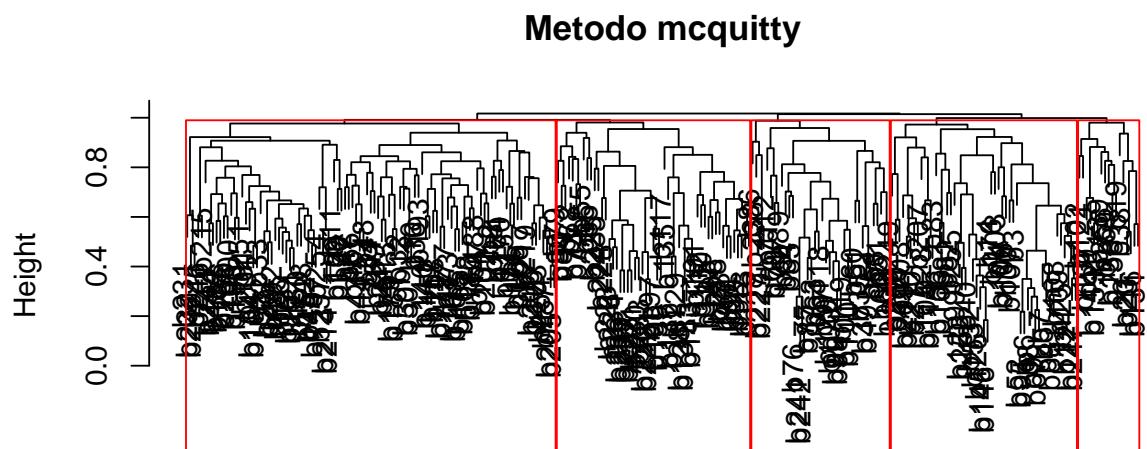


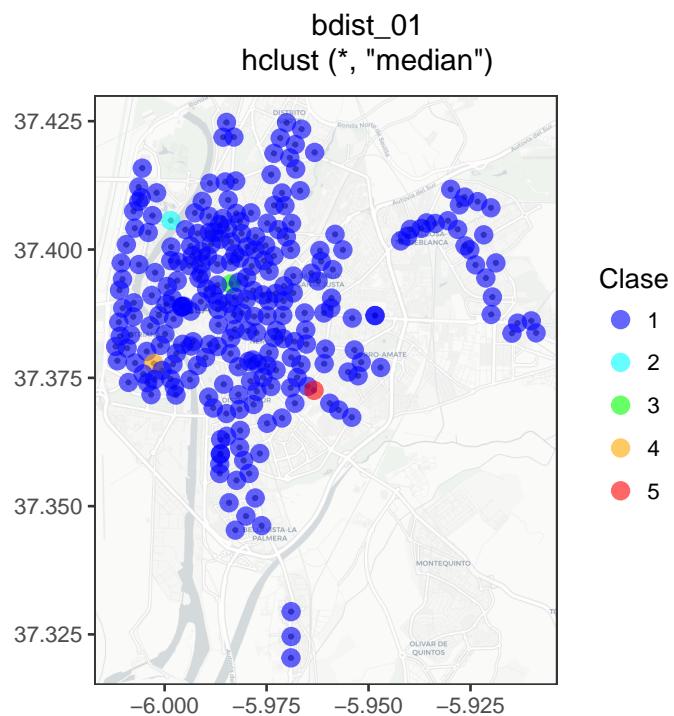
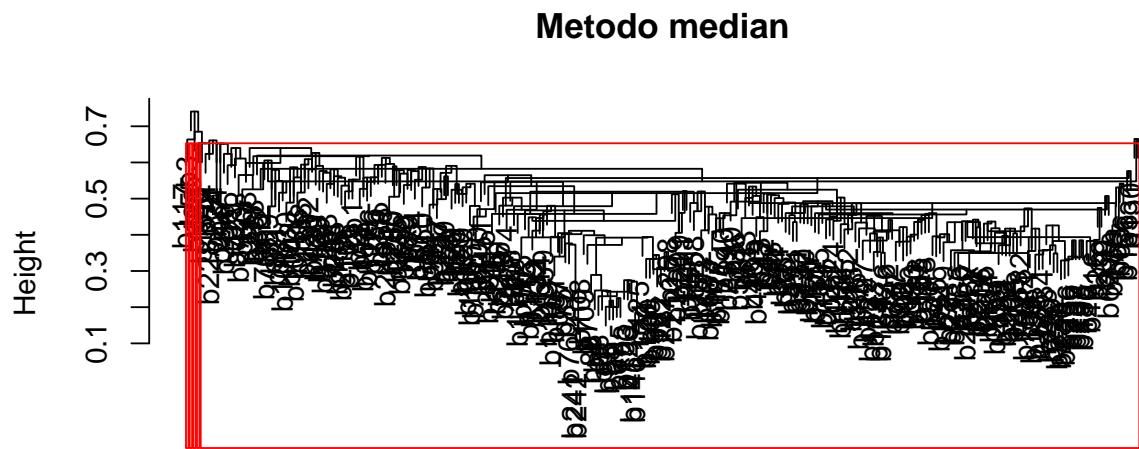


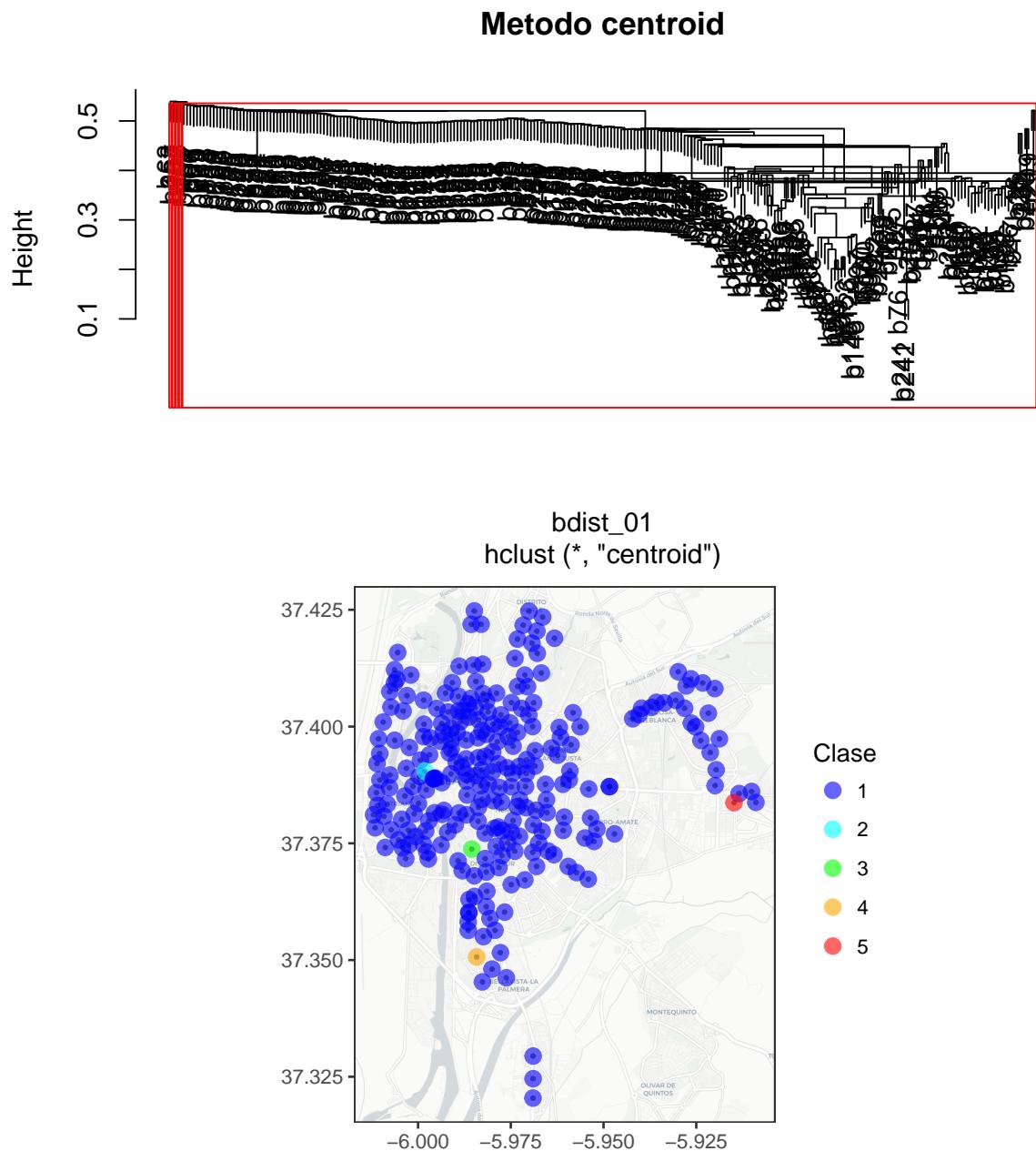












Seleccionamos el método ‘complete’ que tanto en dendrograma como espacialmente presenta buen aspecto y es espacialmente más coherente.

```
clus_01 = hclust(bdist_01, method = metodos.hclust[4])

dclus_01 = dendro_data(clus_01, type = "rectangle")

ggplot() + geom_segment(data = segment(dclus_01), aes(x = x, y = y, xend = xend,
yend = yend), size = 0.2) + geom_text(data = label(dclus_01), aes(x = x,
y = y, label = label, hjust = 2), size = 1.5) + geom_hline(aes(yintercept = 1.3),
color = "red", size = 0.2) + coord_flip() + labs(x = "", y = "") +
theme(axis.line.y = element_blank(), axis.ticks.y = element_blank(),
```

```
axis.text.y = element_blank(), axis.title.y = element_blank(),
panel.background = element_rect(fill = "white"), panel.grid = element_blank())
```

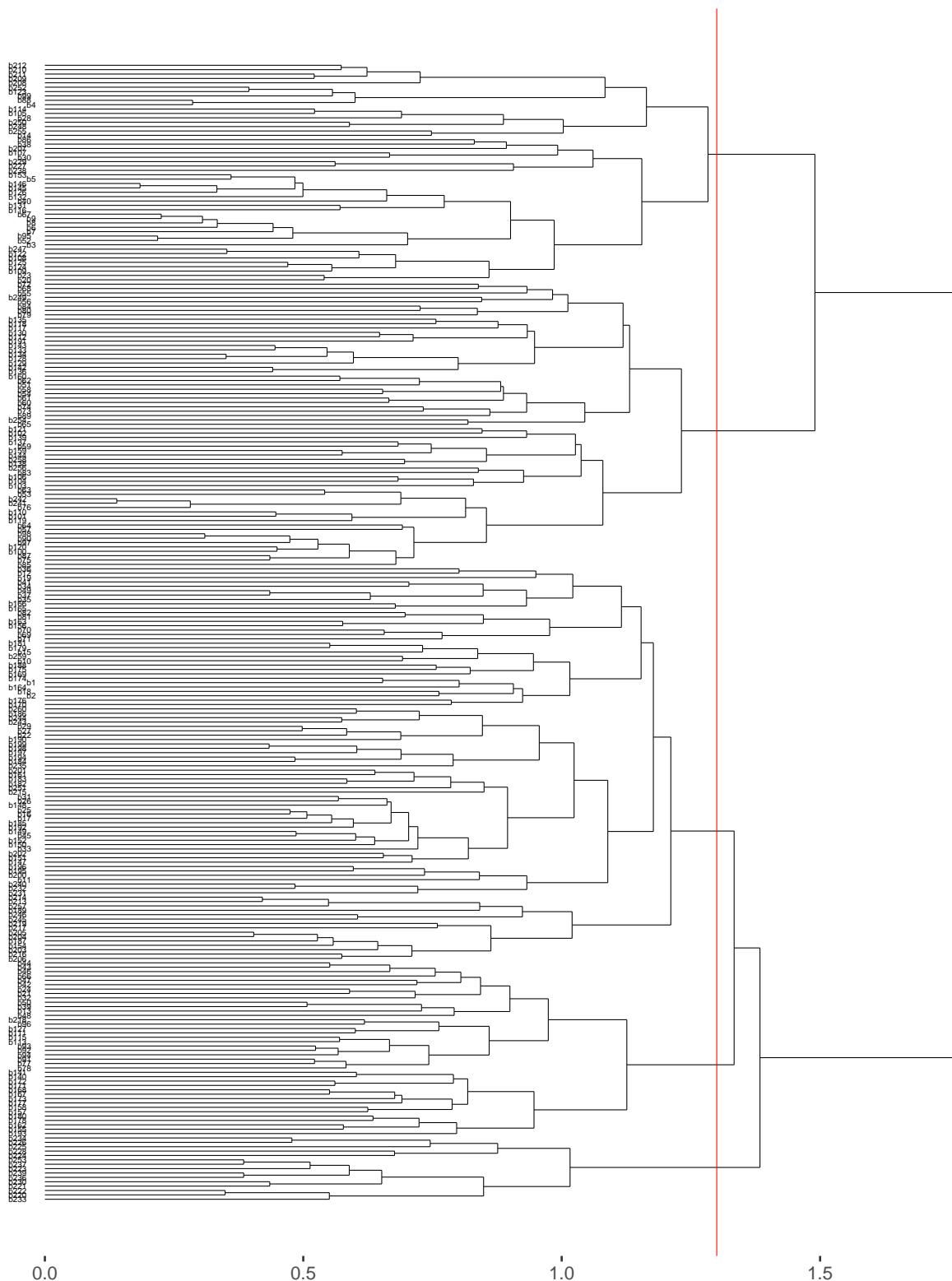


Figura 5: Datos válidos estaciones. Dendrograma de estaciones basado en correlación.

```
clus_01_class = tbl_df(cutree(clus_01, k = 5))
clus_01_class <- seviesta %>% arrange(num) %>% bind_cols(clus_01_class)
# clus_01_class = bind_cols(clus_01_class, seviesta)

ggraph(g, fullpage = TRUE) +
  geom_osm(type = 'cartolight', quiet = TRUE) +
  geom_node_point(color='black', size=0.5, alpha=0.5) +
  geom_point(data=clus_01_class, aes(longitude, latitude,
                                      color=as.factor(value)),
             size=3, alpha=0.6) +
  scale_color_manual(name = 'Clase',
                     values = c('blue'
                               , 'cyan'
                               , 'green'
                               #, 'dark green'
                               #, 'yellow'
                               , 'orange'
                               , 'red'
                               , 'brown'
                               , 'black'
                               )) +
  labs(x="",y="") + coord_map() + theme_bw()
```

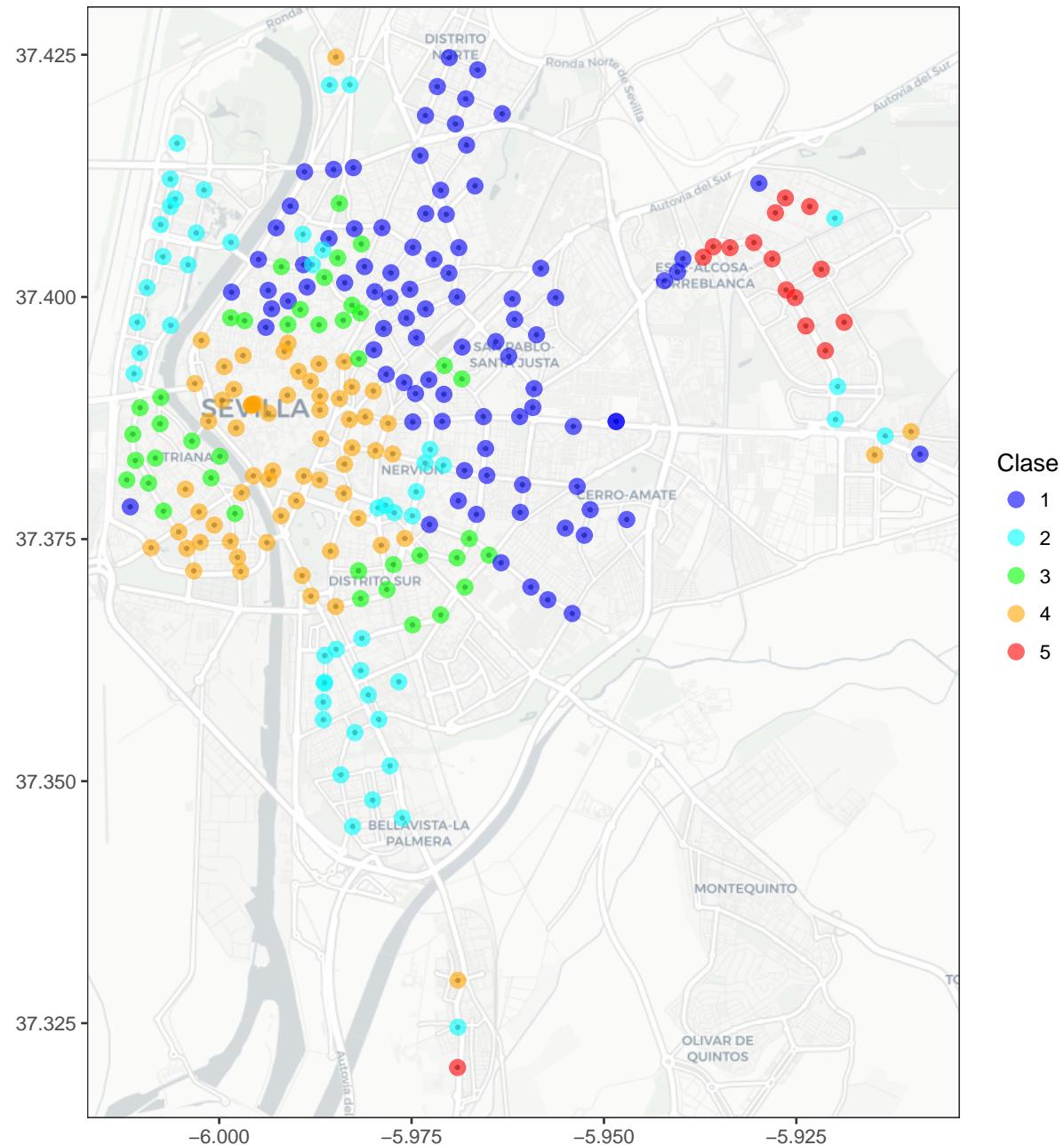


Figura 6: Datos válidos estaciones. Clasificación de estaciones.

Puede apreciarse una distribución espacial de las cinco clases identificadas muy concentrada o compacta, esto es, sería posible establecer una zonificación con un número de zonas casi homogéneas relativamente bajo (entre los vecinos de cada estación son en general mayoría los de su misma clase). Se aprecia así mismo una disposición con cierto carácter concéntrico para las clases.

- La clase 4 ocupa una posición central extendiéndose por parte del casco histórico de la ciudad, los barrios de Nervión, Los Remedios, Felipe II.
- La clase 3 ocupa la primera corona entorno a la clase 4, en Triana, centro Norte - Macarena, Santa Justa, Provenir, Tiro de Linea - La Paz.
- La clase 2 ocupa toda la Isla de la Cartuja, todo el entorno de La Palmera (Sur) y núcleos menores en Nervión, Macarena, Alcosa-Torreblaca y Bellavista.
- La clase 1 ocupa la periferia Norte y Este adentrándose hacia el centro sobre todo por el norte (Macarena, Alameda).
- La clase 5 está en exclusiva en Parque Alcosa-Torreblanca y una estación en Bellavista.

Las zonas de Parque Alcosa-Torreblanca (al Este), Bellavista (al Sur) y posiblemente también San Jerónimo (al Norte) están suficientemente distanciadas del resto como para generar dinámicas propias con patrones de centralidad distintos, lo que podría explicar la distribución de clases que se observan en las mismas.

### 3. Patrones espacio-temporales

```

resumen_por_estacion_dsem_hora = dbQueryIf("resumen_por_estacion_dsem_hora",
  con, "SELECT EXTRACT(ISODOW FROM add_date) as dsem,
    EXTRACT(HOUR FROM add_date) as hora,
    num,
    count(*) as n,
    avg(availablestands) as avgs,
    avg(availablebikes) as avgb,
    stddev(availablestands) as stds,
    stddev(availablebikes) as stdb,
    min(availablestands) as mins,
    min(availablebikes) as minb,
    max(availablestands) as maxs,
    max(availablebikes) as maxb
  FROM sevidata WHERE ok = 1 or ok = 6 GROUP BY dsem, hora, num;")

resumen_por_estacion_dsem_hora$dsem = factor(resumen_por_estacion_dsem_hora$dsem,
  labels = c("L", "M", "X", "J", "V", "S", "D"))

resumen_por_estacion_dsem_hora =tbl_df(resumen_por_estacion_dsem_hora %>%
  inner_join(clus_01_class, by = c(num = "num")))

resumen_clase_dsem_hora = resumen_por_estacion_dsem_hora %>% group_by(dsem,
  hora, value) %>% summarise(means = mean(avgb), minb = min(avgb), maxb = max(avgb),
  sdb = sd(avgb), means = mean(avgs), mins = min(avgs), maxs = max(avgs),
  sda = sd(avgs))

resumen_clase_dsem_hora <- resumen_clase_dsem_hora %>% mutate(pctb = 100 *
  meanb/(meanb + means))

resumen_clase_dsem_hora <- resumen_clase_dsem_hora %>% mutate(pctCVb = 100 *
  sdb/means)

```

```
ggplot(resumen_clase_dsem_hora) + geom_tile(aes(y = as.factor(hora), x = as.factor(dsem),
  fill = pctb)) + scale_fill_gradientn(colors = c("cyan", "green", "yellow",
  "red")) + # scale_x_continuous(breaks = c(0,2,4,6,8,10,12,14,16,18,20,22))+  
  labs(x = "Día de la semana", y = "Hora del día") + facet_grid(~value) +  
  theme(legend.position = "bottom")
```

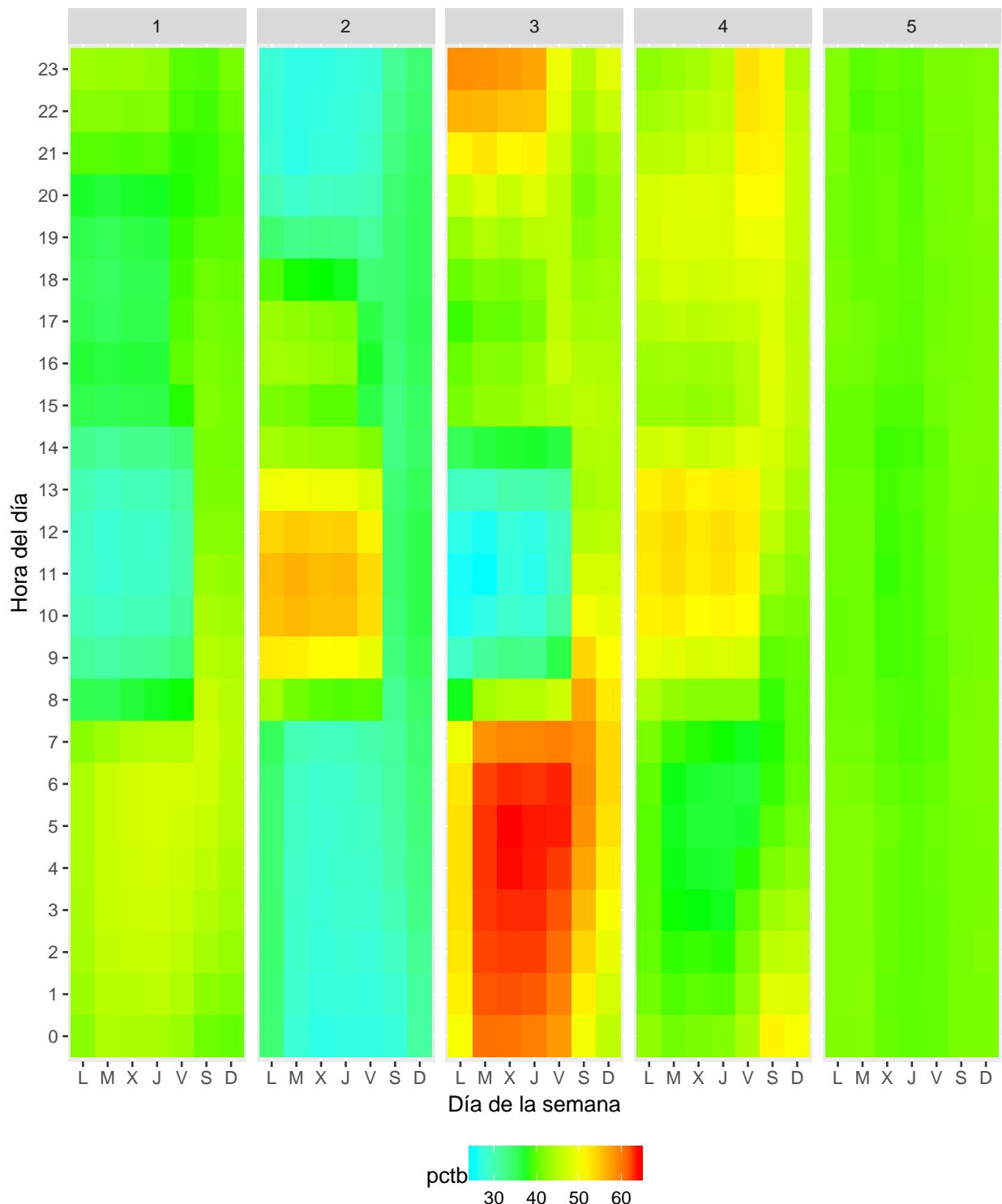


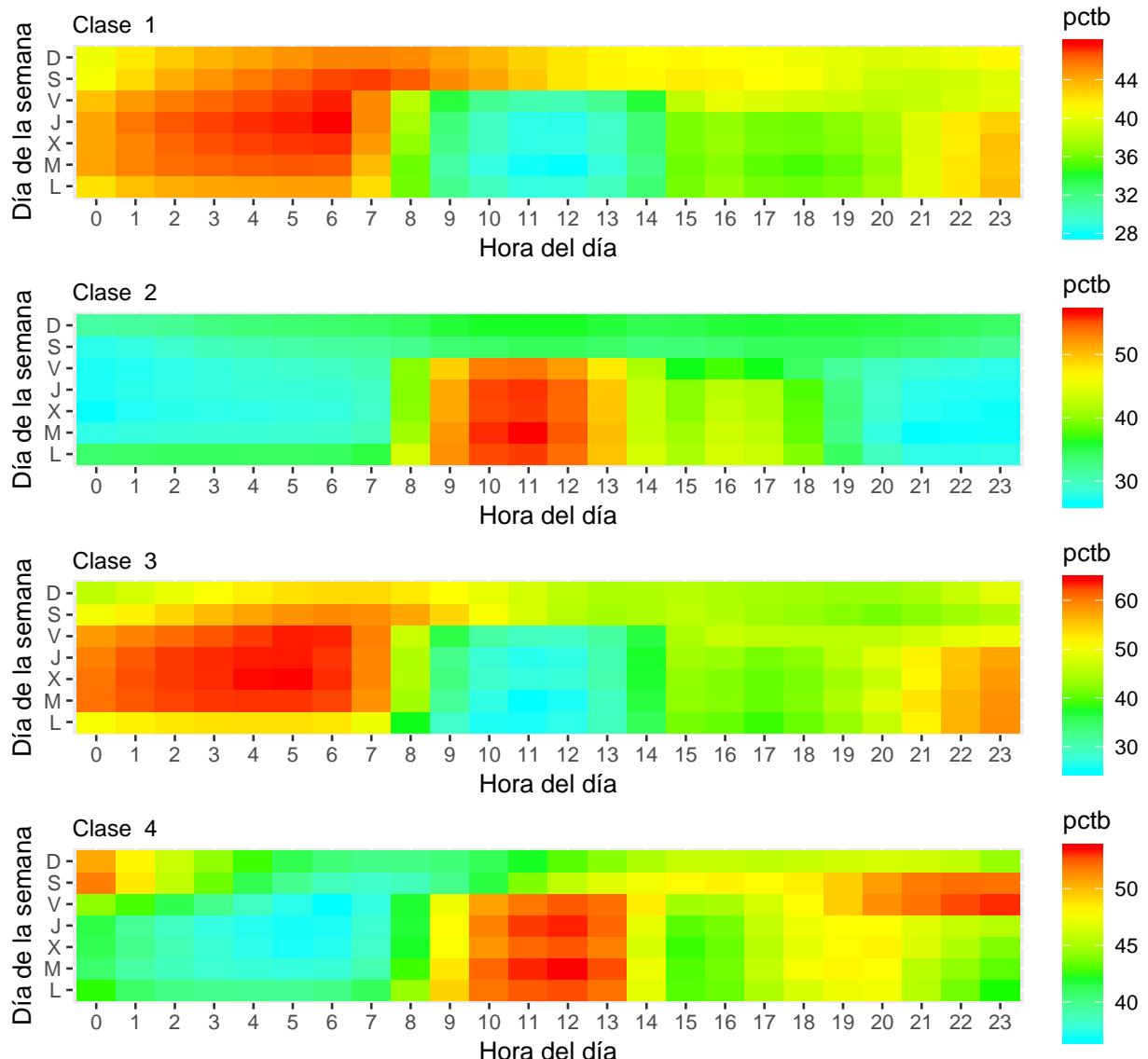
Figura 7: Datos estaciones. %Bicis disponibles por clase de estación, hora del día y día de la semana.

```
for (i in 1:4) {
  pltfill = ggplot(resumen_clase_dsem_hora %>% filter(value == i)) +
```

```

geom_tile(aes(x = as.factor(hora), y = as.factor(dsem), fill = pctb)) +
  scale_fill_gradientn(colors = c("cyan", "green", "yellow", "red")) +
  labs(y = "Día de la semana", x = "Hora del día", subtitle = paste("Clase ", i))
print(pltfill)
}

```



```

ggplot(resumen_clase_dsem_hora %>% filter(value == 5)) + geom_tile(aes(x = as.factor(hora),
  y = as.factor(dsem), fill = pctb)) + scale_fill_gradientn(colors = c("cyan",
  "green", "yellow", "red")) + labs(y = "Día de la semana", x = "Hora del día",
  subtitle = paste("Clase ", 5))

```

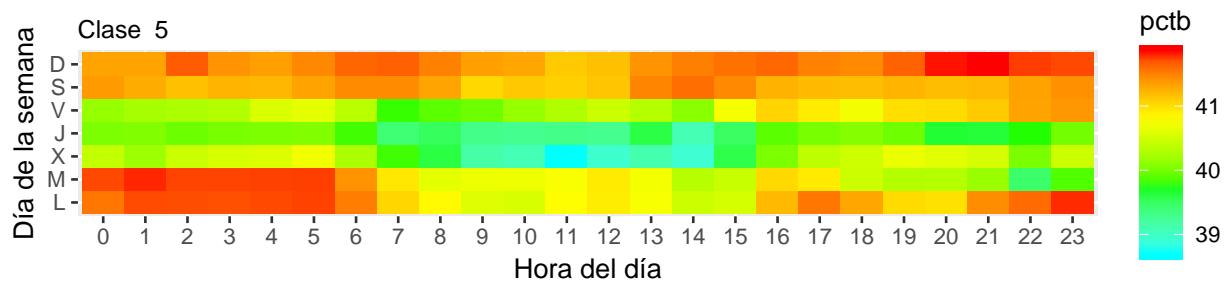


Figura 8: Datos estaciones. %Bicis disponibles por hora del día y día de la semana. Patrones por clase de estación.

La distribución por día de la semana y hora del día del porcentaje de bicis disponibles entre las distintas clases de estaciones muestra:

- 1) Los patrones para las clases 1 y 4 son claramente complementarios, correspondiendo la clase 1 a estaciones con concentración de bicicletas disponibles todos los días de noche y madrugada y la clase 4 a estaciones con máxima presencia de bicis disponibles entre las 9:00 y las 13:00 horas de Lunes a Viernes. Lo que se correspondería a desplazamientos entre residencia (1) y trabajo o estudio (4).
- 2) Los patrones para las clases 2 y 3 son igualmente complementarios y muy parecidos a los indicados para 4 y 1, respectivamente.
- 3) La clase 5 presenta un comportamiento temporal bien distinto al de las otras clases, con máximos en las madrugadas de lunes y martes, niveles relativamente altos durante todo el fin de semana, y mínimos en la parte central del día de los días centrales de la semana (X,J,V).
- 4) Las clases 2 y 4 aunque presentan patrones generales muy parecidos, según lo dicho, se diferencian sobre todo por su comportamiento los viernes y sábados a partir de las 20:00, con niveles altos en la clase 4, posiblemente inducida por desplazamientos a actividades de tipo lúdico.
- 5) La distinción entre los patrones 1 y 3 está vinculada al comportamiento en fin de semana relacionada con la extensión de niveles altos hasta horas más tardías en la clase 1.

La variabilidad interna dentro de las clases puede apreciarse mediante el coeficiente de variación. Se muestra seguidamente expresada en %.

```
ggplot(resumen_clase_dsem_hora) + geom_tile(aes(y = as.factor(hora), x = as.factor(dsem),
fill = pctCVb)) + scale_fill_gradientn(colors = c("cyan", "green",
"yellow", "red")) + # scale_x_continuous(breaks = c(0,2,4,6,8,10,12,14,16,18,20,22))+ 
labs(x = "Día de la semana", y = "Hora del día") + facet_grid(~value) +
theme(legend.position = "bottom")
```

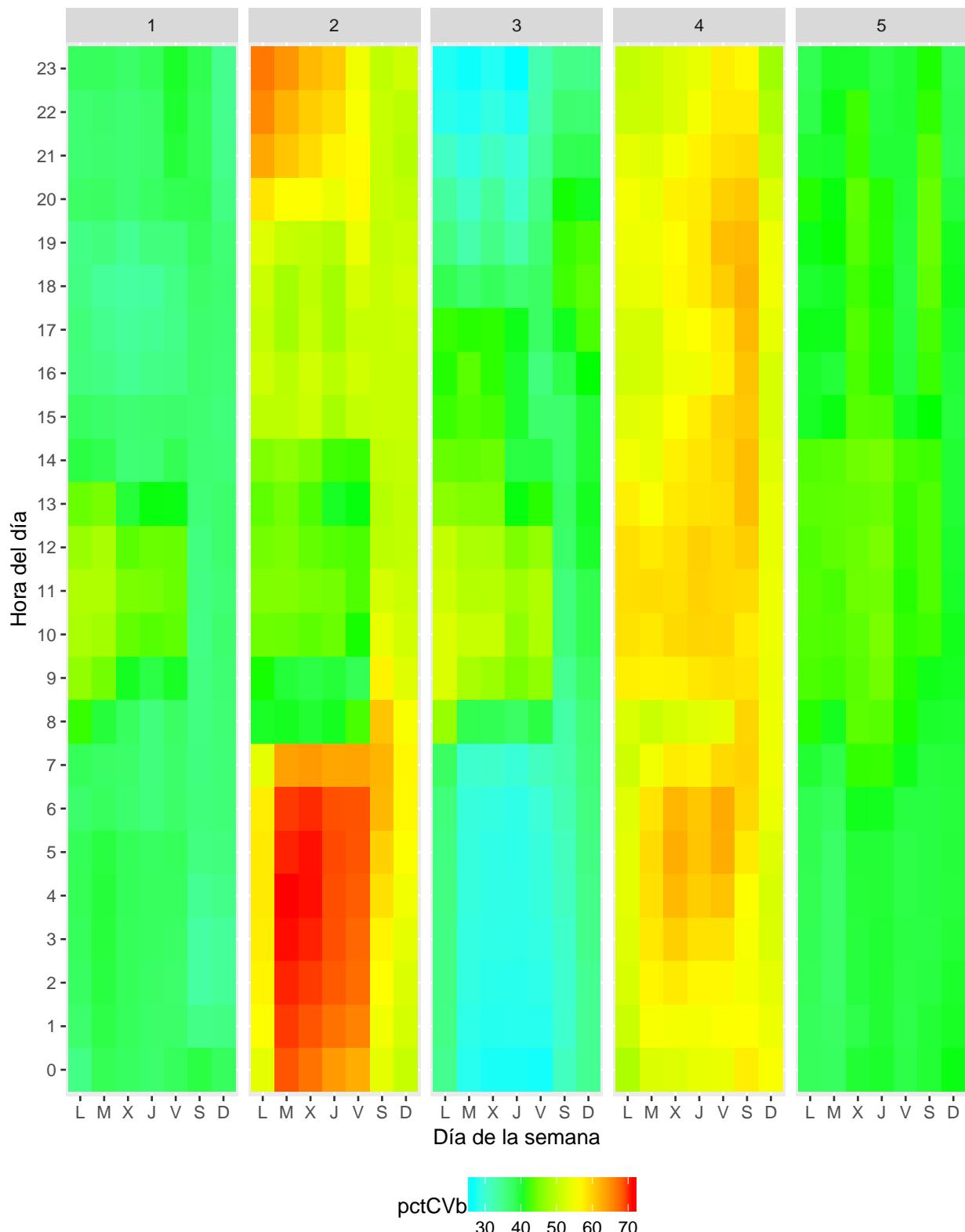
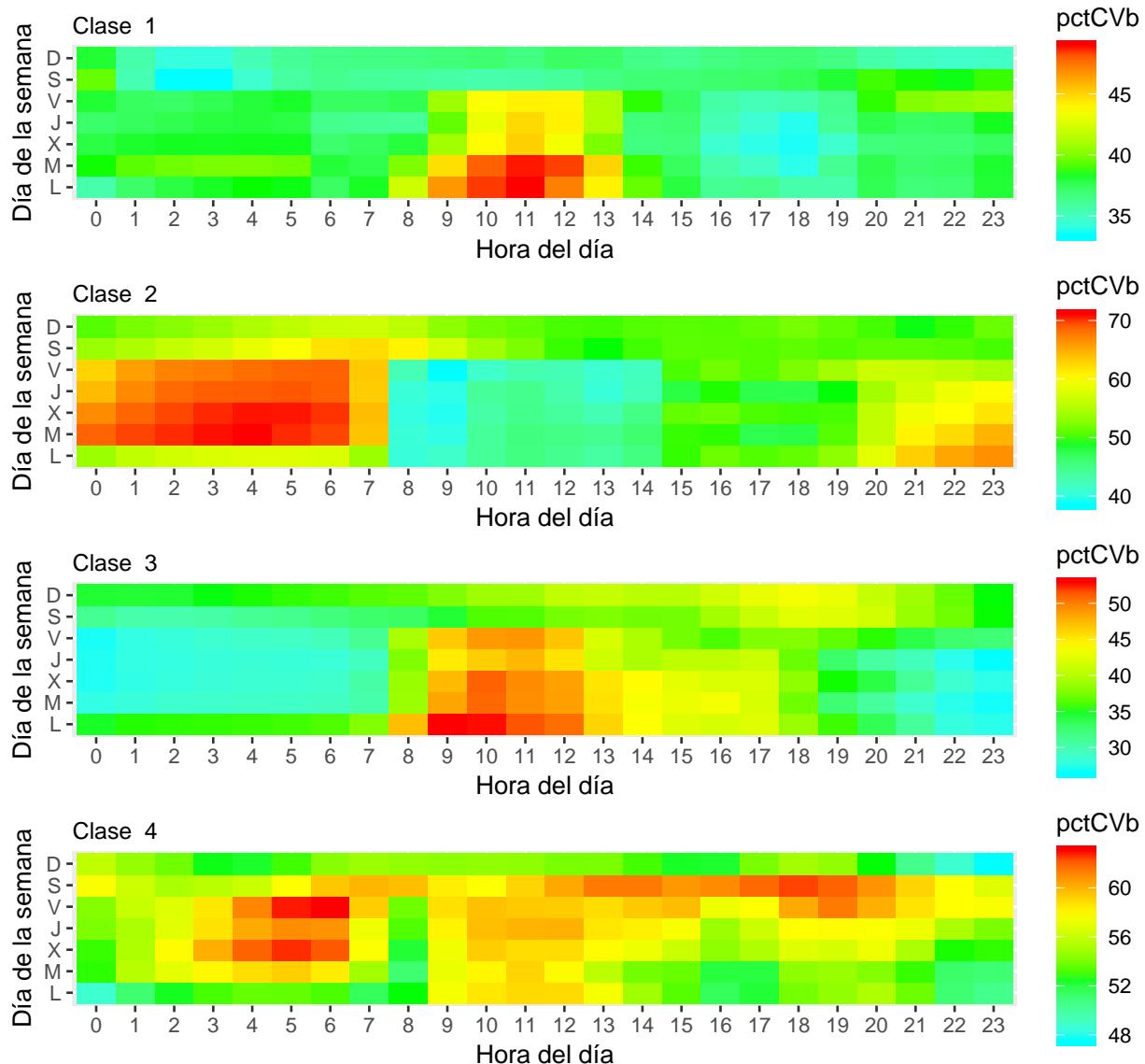


Figura 9: Datos estaciones. Bicis disponibles por clase de estación, hora del día y día de la semana. Coeficiente de Variación (%).

```
for (i in 1:4) {
  pltfill = ggplot(resumen_clase_dsem_hora %>% filter(value == i)) +
    geom_tile(aes(x = as.factor(hora), y = as.factor(dsem), fill = pctCVb)) +
    scale_fill_gradientn(colors = c("cyan", "green", "yellow", "red")) +
    labs(y = "Día de la semana", x = "Hora del día", subtitle = paste("Clase ", i))
  print(pltfill)
}
```



```
ggplot(resumen_clase_dsem_hora %>% filter(value == 5)) + geom_tile(aes(x = as.factor(hora),
y = as.factor(dsem), fill = pctCVb)) + scale_fill_gradientn(colors = c("cyan",
"green", "yellow", "red")) + labs(y = "Día de la semana", x = "Hora del día",
subtitle = paste("Clase ", 5))
```

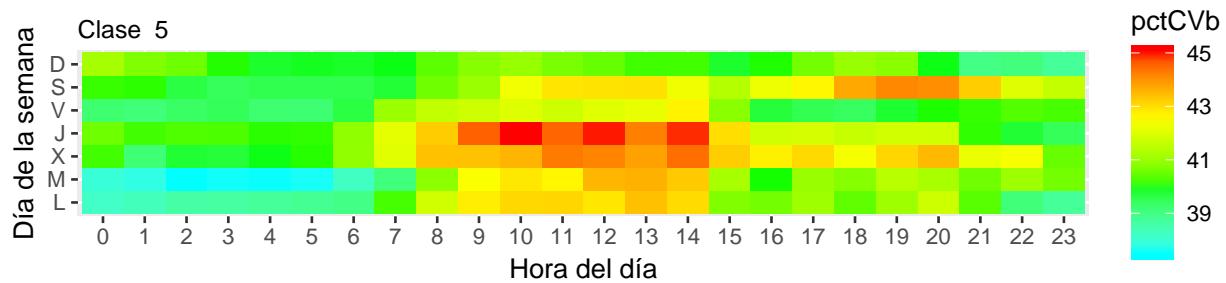


Figura 10: Datos estaciones. Bicis disponibles por hora del día y día de la semana. Patrones por clase de estación. Coeficiente de Variación (%).

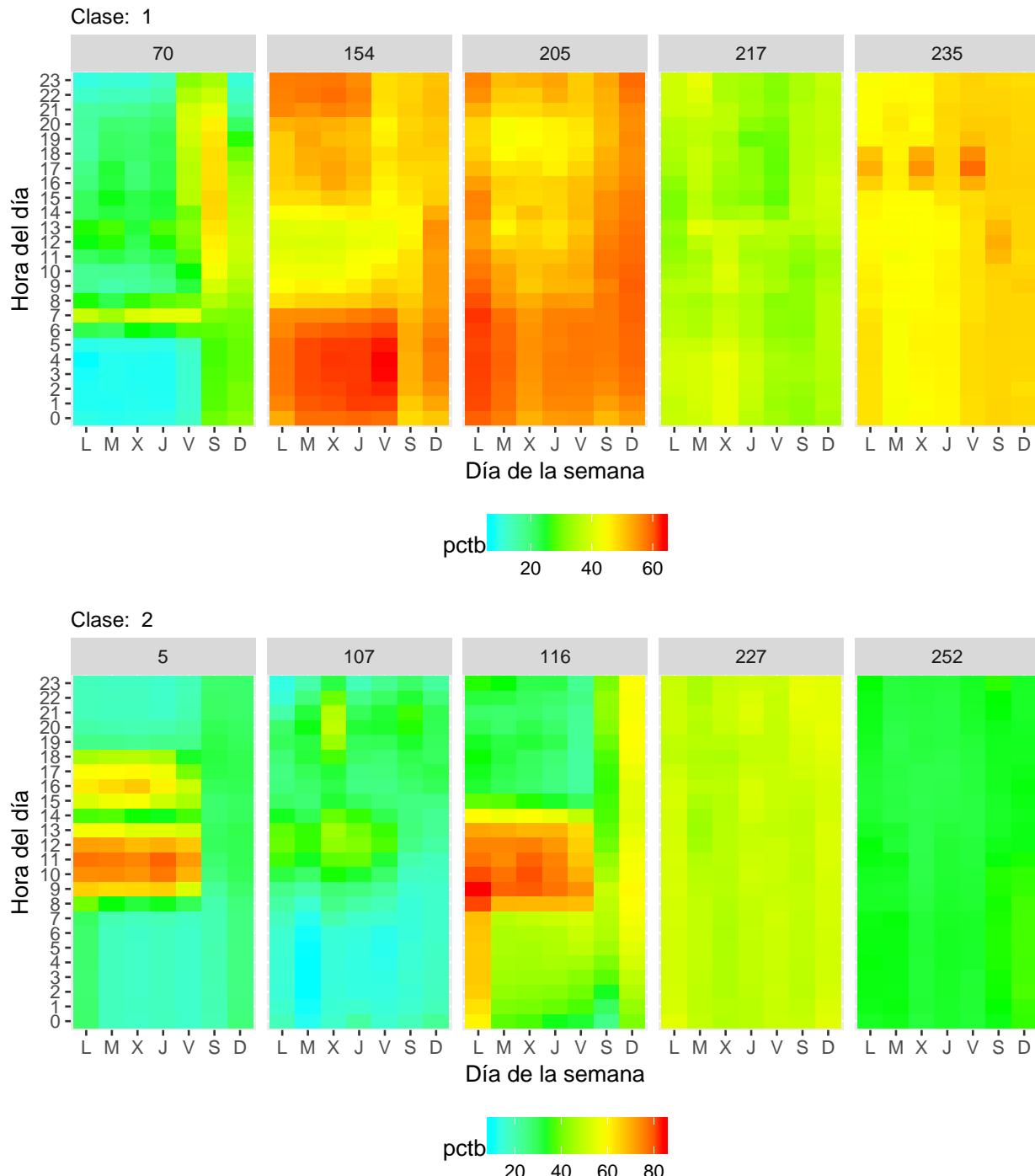
Se presenta seguidamente el comportamiento individual de una muestra de estaciones en cada una de las clases identificadas.

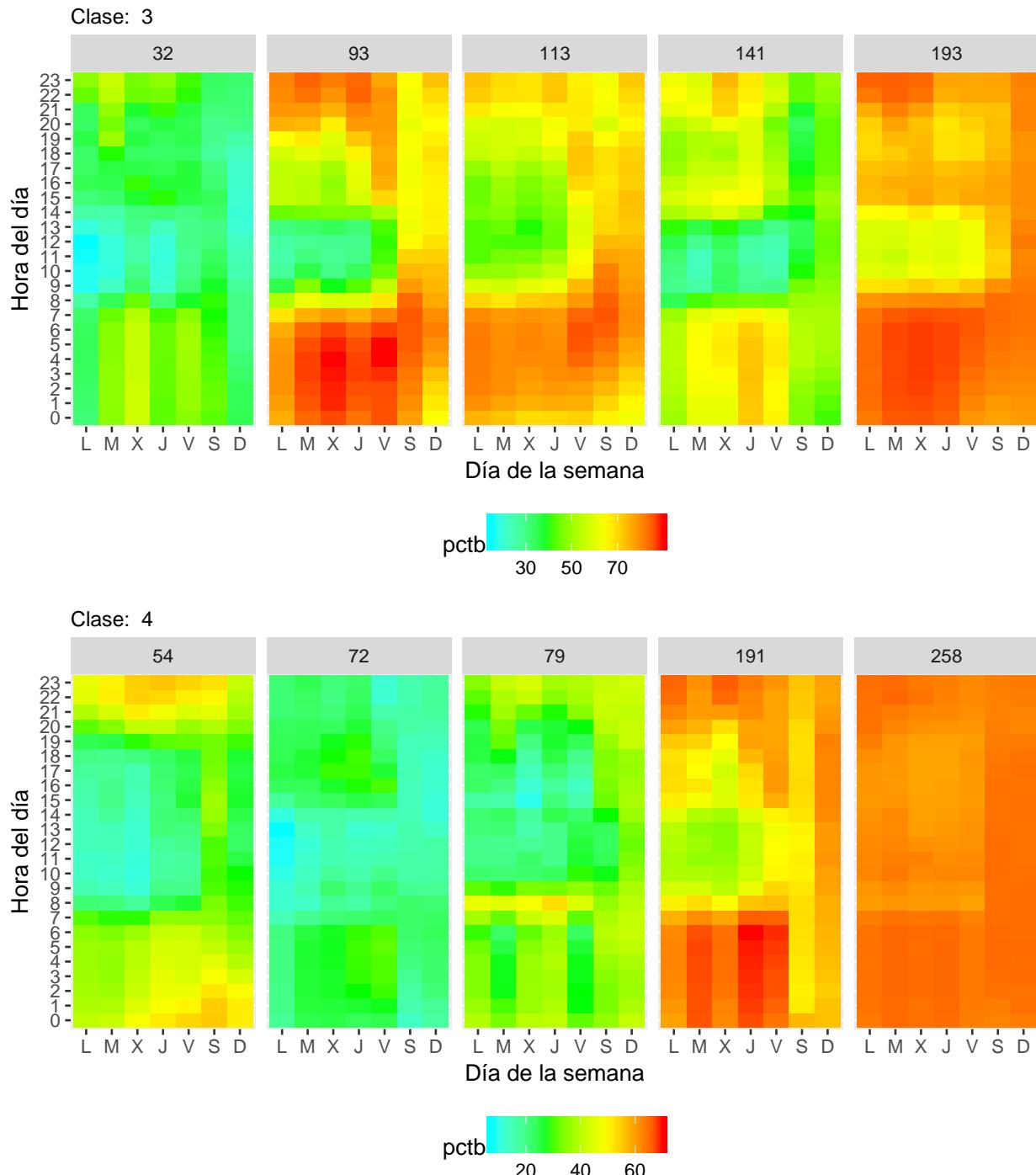
```
set.seed(123)
sample_stations_class <- clus_01_class %>% group_by(value) %>% sample_n(5) %>%
    arrange(value, num)

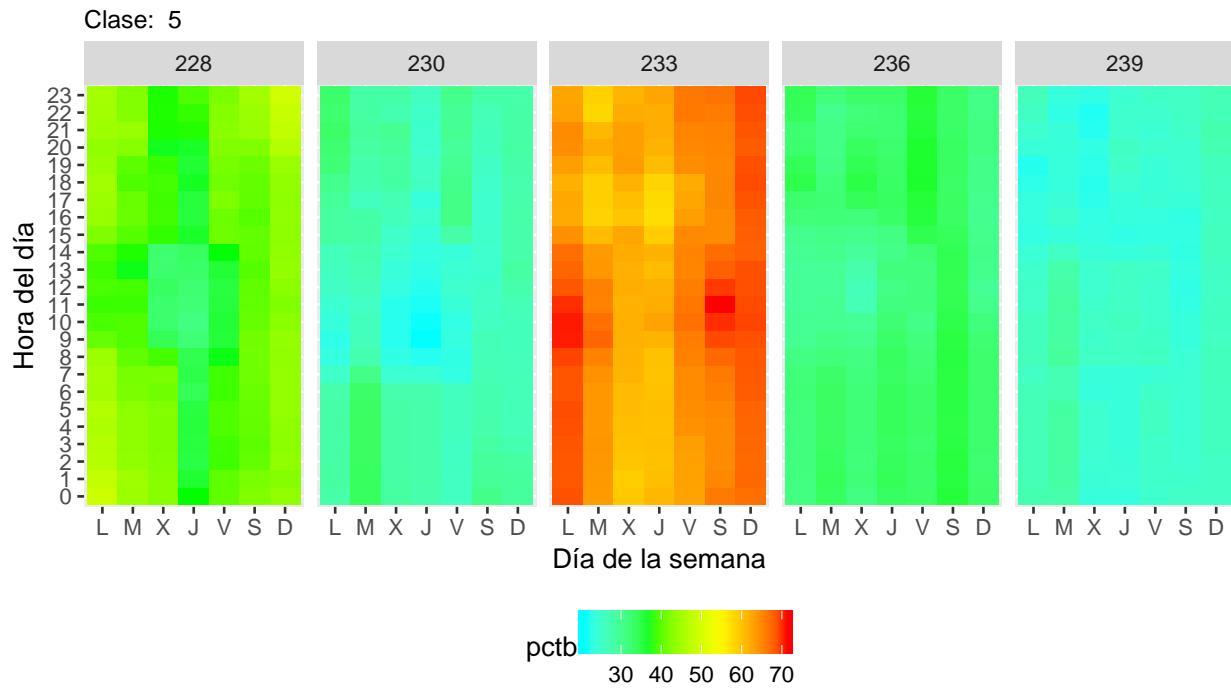
# dim(sample_stations_class) head(sample_stations_class)

resumen_por_estacion_dsem_hora <- resumen_por_estacion_dsem_hora %>% mutate(pctb = 100 * 
    avgb/(avgb + avgs))

for (k in 1:5) {
    pltfillmu = ggplot(resumen_por_estacion_dsem_hora %>% semi_join(sample_stations_class,
        by = "num") %>% filter(value == k)) + geom_tile(aes(y = as.factor(hora),
            x = as.factor(dsem), fill = pctb)) + scale_fill_gradientn(colors = c("cyan",
                "green", "yellow", "red")) + labs(x = "Día de la semana", y = "Hora del día",
                subtitle = paste("Clase: ", k)) + facet_grid(~num) + theme(legend.position = "bottom")
    print(pltfillmu)
}
```







```
ggplot()
```

Figura 11: Datos estaciones. %Bicis disponibles por hora del día y día de la semana. Muestra de estaciones por clase.

Con objeto de contrastar la validez de los patrones espacio-temporales observados construimos un modelo con las variables independientes clase de estación, día de la semana y hora del día y variable dependiente el número de bicicletas disponibles.

Los datos para construir el modelo no son los datos completamente desagregados sino que se utilizan las medias del número de bicicletas disponibles por estación, fecha y hora. Este conjunto de datos tiene más de 2 millones de registros.

```
data_glm = seidata_num_fecha_hora_dsem %>% inner_join(clus_01_class, by = "num") %>%
  mutate(pctb = avgb/(avgb + avgc) * 100) %>% select(one_of("pctb", "value",
  "dsem", "hora")) %>% rename(cls = value) %>% filter(pctb >= 0)

summary(data_glm)
```

	pctb	cls	dsem	hora
## Min.	: 0.00	Min. :1.000	Min. :1.000	Min. : 0.0
## 1st Qu.:	12.92	1st Qu.:1.000	1st Qu.:2.000	1st Qu.: 5.0
## Median :	36.11	Median :2.000	Median :4.000	Median :11.0
## Mean :	42.35	Mean :2.479	Mean :3.998	Mean :11.5
## 3rd Qu.:	70.56	3rd Qu.:4.000	3rd Qu.:6.000	3rd Qu.:18.0
## Max. :	100.00	Max. :5.000	Max. :7.000	Max. :23.0

```
data_glm$cls = factor(data_glm$cls)
data_glm$dsem = factor(data_glm$dsem)
```

```
# data_glm$hora = factor(data_glm$hora)

summary(data_glm)

##      pctb        cls       dsem        hora
##  Min.   : 0.00   1:768923  1:309673  Min.   : 0.0
##  1st Qu.: 12.92  2:420639  2:320174  1st Qu.: 5.0
##  Median : 36.11  3:341150  3:321659  Median :11.0
##  Mean   : 42.35  4:552836  4:316642  Mean   :11.5
##  3rd Qu.: 70.56  5:127038  5:313220  3rd Qu.:18.0
##  Max.   :100.00          6:315232  6:313986  Max.   :23.0
##                                         7:313986

mod_glm = glm(pctb ~ ., data = data_glm)

summary(mod_glm)

Call:
glm(formula = pctb ~ ., data = data_glm)

Deviance Residuals:
    Min      1Q   Median      3Q      Max 
-59.961 -28.289 -6.174  27.173  72.344 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 41.59419   0.12184 341.396 < 2e-16 ***
cls2        -11.04681   0.08421 -131.190 < 2e-16 ***
cls3         5.04646   0.05738   87.954 < 2e-16 ***
cls4        -1.83015   0.05618  -32.575 < 2e-16 ***
cls5         16.17585   0.08258  195.876 < 2e-16 ***
dsem2        0.22705   0.08029   2.828 0.004685 **  
dsem3        0.19638   0.08020   2.449 0.014338 *   
dsem4        0.29428   0.08051   3.655 0.000257 *** 
dsem5        0.57196   0.08072   7.085 1.39e-12 ***
dsem6        1.16314   0.08060  14.432 < 2e-16 ***
dsem7        0.97045   0.08068  12.029 < 2e-16 ***
hora1        0.38757   0.14839   2.612 0.009003 **  
hora2        0.68545   0.14848   4.616 3.91e-06 ***
hora3        0.86640   0.14839   5.839 5.26e-09 ***
hora4        0.97805   0.14844   6.589 4.43e-11 ***
hora5        1.02817   0.14844   6.926 4.32e-12 ***
hora6        0.98977   0.14844   6.668 2.60e-11 ***
hora7        0.12919   0.14844   0.870 0.384115  
hora8        -1.34364   0.14838  -9.055 < 2e-16 ***
hora9        -1.65884   0.14849 -11.172 < 2e-16 ***
hora10       -1.78680   0.14845 -12.036 < 2e-16 ***
hora11       -2.12190   0.14845 -14.294 < 2e-16 ***
hora12       -2.36827   0.14855 -15.942 < 2e-16 ***
hora13       -2.53617   0.14854 -17.074 < 2e-16 ***
hora14       -2.89099   0.14843 -19.477 < 2e-16 ***
hora15       -2.20202   0.14864 -14.815 < 2e-16 ***
hora16       -1.51862   0.14858 -10.221 < 2e-16 ***
hora17       -1.75505   0.14841 -11.826 < 2e-16 ***
hora18       -1.97019   0.14835 -13.281 < 2e-16 ***
```

```

hora19      -2.08439   0.14839  -14.047 < 2e-16 ***
hora20      -2.03999   0.14849  -13.738 < 2e-16 ***
hora21      -1.42718   0.14849  -9.611 < 2e-16 ***
hora22      -0.58673   0.14849  -3.951 7.77e-05 ***
hora23      -0.08916   0.14849  -0.600  0.548195
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

```

(Dispersion parameter for gaussian family taken to be 1014.714)

```

Null deviance: 2330727744 on 2210585 degrees of freedom
Residual deviance: 2243078554 on 2210552 degrees of freedom
AIC: 21575884

```

Number of Fisher Scoring iterations: 2

Salvo hora23 y hora7, todos los niveles de los factores considerados son altamente significativos.

Veamos ahora interacciones sobre una muestra.

```

sampledata_glm = data_glm %>% sample_n(5e+05)

mod2_glm = glm(pctb ~ cls * dsem * hora, data = sampledata_glm)

summary(mod2_glm)

```

```

##
## Call:
## glm(formula = pctb ~ cls * dsem * hora, data = sampledata_glm)
##
## Deviance Residuals:
##    Min      1Q      Median      3Q      Max
## -56.936 -29.062   -5.969    27.910    66.537
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 41.38198   0.40283 102.729 < 2e-16 ***
## cls2        3.13324   0.67606  4.635 3.58e-06 ***
## cls3        2.92635   0.72582  4.032 5.54e-05 ***
## cls4        0.59559   0.62282  0.956 0.338929
## cls5       -1.39448   1.07800 -1.294 0.195809
## dsem2       1.60853   0.56368  2.854 0.004323 **
## dsem3       3.01701   0.56098  5.378 7.53e-08 ***
## dsem4       2.86736   0.56627  5.064 4.11e-07 ***
## dsem5       3.27764   0.56369  5.815 6.08e-09 ***
## dsem6       5.32019   0.56909  9.349 < 2e-16 ***
## dsem7       3.92685   0.56661  6.930 4.20e-12 ***
## hora        -0.24397   0.02987 -8.168 3.13e-16 ***
## cls2:dsem2 -6.35098   0.94650 -6.710 1.95e-11 ***
## cls3:dsem2  9.26215   1.01480  9.127 < 2e-16 ***
## cls4:dsem2 -2.75286   0.86989 -3.165 0.001553 **
## cls5:dsem2 -1.04009   1.49367 -0.696 0.486221
## cls2:dsem3 -9.94800   0.94085 -10.573 < 2e-16 ***
## cls3:dsem3  9.13286   1.01229  9.022 < 2e-16 ***
## cls4:dsem3 -4.55586   0.86941 -5.240 1.60e-07 ***
## cls5:dsem3 -3.11927   1.49601 -2.085 0.037064 *

```

```

##  cls2:dsem4      -9.17938   0.94748  -9.688 < 2e-16 ***
##  cls3:dsem4       8.25864   1.01881   8.106 5.24e-16 ***
##  cls4:dsem4      -4.50699   0.87613  -5.144 2.69e-07 ***
##  cls5:dsem4      -1.52428   1.51019  -1.009 0.312817
##  cls2:dsem5      -8.91542   0.94787  -9.406 < 2e-16 ***
##  cls3:dsem5       9.34993   1.01720   9.192 < 2e-16 ***
##  cls4:dsem5      -5.22500   0.87472  -5.973 2.33e-09 ***
##  cls5:dsem5      -4.43132   1.50607  -2.942 0.003258 **
##  cls2:dsem6     -16.37231   0.95299 -17.180 < 2e-16 ***
##  cls3:dsem6       6.99303   1.02039   6.853 7.22e-12 ***
##  cls4:dsem6      -5.53161   0.87512  -6.321 2.60e-10 ***
##  cls5:dsem6      -3.56216   1.50672  -2.364 0.018071 *
##  cls2:dsem7     -11.96496   0.94772 -12.625 < 2e-16 ***
##  cls3:dsem7       2.20202   1.02119   2.156 0.031058 *
##  cls4:dsem7      -2.86179   0.87532  -3.269 0.001078 **
##  cls5:dsem7      -1.71387   1.50362  -1.140 0.254358
##  cls2:hora        0.12309   0.05026   2.449 0.014329 *
##  cls3:hora        0.12791   0.05396   2.371 0.017762 *
##  cls4:hora        0.47209   0.04603  10.256 < 2e-16 ***
##  cls5:hora        0.41117   0.08049   5.108 3.25e-07 ***
##  dsem2:hora       -0.10401   0.04194  -2.480 0.013140 *
##  dsem3:hora       -0.18436   0.04176  -4.415 1.01e-05 ***
##  dsem4:hora       -0.15410   0.04209  -3.661 0.000251 ***
##  dsem5:hora       -0.12707   0.04197  -3.028 0.002463 **
##  dsem6:hora       -0.05499   0.04225  -1.302 0.193044
##  dsem7:hora        0.02264   0.04219   0.537 0.591556
##  cls2:dsem2:hora    0.33542   0.07042   4.763 1.91e-06 ***
##  cls3:dsem2:hora    -0.48033   0.07564  -6.350 2.15e-10 ***
##  cls4:dsem2:hora    0.20153   0.06457   3.121 0.001803 **
##  cls5:dsem2:hora    -0.06113   0.11185  -0.547 0.584658
##  cls2:dsem3:hora    0.50168   0.07003   7.164 7.84e-13 ***
##  cls3:dsem3:hora    -0.47726   0.07529  -6.339 2.31e-10 ***
##  cls4:dsem3:hora    0.25180   0.06449   3.904 9.45e-05 ***
##  cls5:dsem3:hora    0.06787   0.11177   0.607 0.543699
##  cls2:dsem4:hora    0.44478   0.07053   6.306 2.87e-10 ***
##  cls3:dsem4:hora    -0.41171   0.07588  -5.426 5.77e-08 ***
##  cls4:dsem4:hora    0.27415   0.06496   4.220 2.44e-05 ***
##  cls5:dsem4:hora    -0.12382   0.11219  -1.104 0.269721
##  cls2:dsem5:hora    0.24678   0.07061   3.495 0.000474 ***
##  cls3:dsem5:hora    -0.55655   0.07581  -7.341 2.12e-13 ***
##  cls4:dsem5:hora    0.35879   0.06498   5.521 3.37e-08 ***
##  cls5:dsem5:hora    0.11128   0.11252   0.989 0.322677
##  cls2:dsem6:hora    0.41098   0.07089   5.798 6.72e-09 ***
##  cls3:dsem6:hora    -0.52837   0.07593  -6.959 3.44e-12 ***
##  cls4:dsem6:hora    0.12036   0.06484   1.856 0.063412 .
##  cls5:dsem6:hora    -0.11742   0.11280  -1.041 0.297891
##  cls2:dsem7:hora    0.24440   0.07084   3.450 0.000561 ***
##  cls3:dsem7:hora    -0.21326   0.07632  -2.794 0.005200 **
##  cls4:dsem7:hora    -0.16202   0.06508  -2.489 0.012795 *
##  cls5:dsem7:hora    -0.25277   0.11265  -2.244 0.024845 *
##  ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 1038.962)

```

```
##  
## Null deviance: 527349947 on 499999 degrees of freedom  
## Residual deviance: 519408115 on 499930 degrees of freedom  
## AIC: 4891999  
##  
## Number of Fisher Scoring iterations: 2
```