

# Ensayo de métodos de monitorización acústica en el Jardín Botánico de Maracaibo, municipio San Francisco, estado Zulia

---

JR Ferrer-Paris  
Laboratorio de Ecología Espacial, Centro de Estudios Botánicos y  
Agroforestales  
Instituto Venezolano de Investigaciones Científicas

Borrador  
Versión de 1 de noviembre de 2014

## A quien pueda interesar

Este documento es generado utilizando las funciones de **Sweave** desde una sesión de *R* (R Development Core Team, 2010), por tanto todas las tablas y figuras se generan y actualizan automáticamente a partir de los datos suministrados. Para acceso al código fuente en *R* y los archivos de datos contacte al primer autor. Dentro de *R* utilizamos los paquetes *gstat* (Pebesma, 2004); *plotrix* (Lemon, 2006); *rgeos* (Bivand & Rundel, 2012); *igraph0* (Csardi & Nepusz, 2006); *rgdal* (Keitt *et al.*, 2012); *RColorBrewer* (Neuwirth, 2011); *raster* (Hijmans & van Etten, 2012); *sp* (Pebesma & Bivand, 2005); *chron* (James & Hornik, 2011).

## 1. Puntos de muestreo

Bajamos los datos del archivo en *Google Drive* en formato **csv** y los leemos en una sesión de *R*.

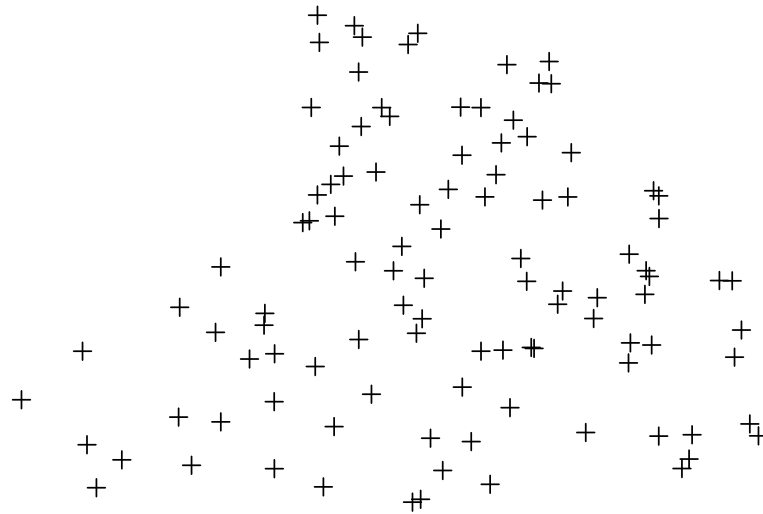
```
> Puntos <- read.csv("~/CEBA/data/JardinBotanicoMaracaibo/20141028_MA_Puntos.csv")
> Muestreo <- read.csv("~/CEBA/data/JardinBotanicoMaracaibo/20141028_MA_Muestreo.csv")
> Grabaciones <- read.csv("~/CEBA/data/JardinBotanicoMaracaibo/20141028_MA_Grabaciones.csv")
```

Vamos a empezar con la lista de puntos caracterizados. Debemos calcular la incertidumbre de la georeferencia, que en este caso es la suma de la distancia al punto (DAP), el error de posicionamiento o error de GPS (`Error.GPS`) y la extensión del punto del muestreo (`Extent`) que equivale a la distancia máxima de las fuentes de sonido que pueden ser detectadas con el micrófono, según Arlene esta es de aproximadamente 10 m. La incertidumbre (`Incertidumbre`) está expresada en metros.

```
> Puntos$DAP <- as.numeric(sub(",", ".", as.character(Puntos$Distancia.al.punto)))
> Puntos$DAP <- ifelse(is.na(Puntos$DAP), mean(Puntos$DAP, na.rm=T), Puntos$DAP)
> Puntos$Error.GPS <- Puntos$EPE
> Puntos$Error.GPS <- ifelse(is.na(Puntos$Error.GPS), mean(Puntos$Error.GPS, na.rm=T), Puntos$Error.GPS)
> Puntos$Extent <- 10
> Puntos$Incertidumbre <- Puntos$DAP + Puntos$Error.GPS + Puntos$Extent
>
```

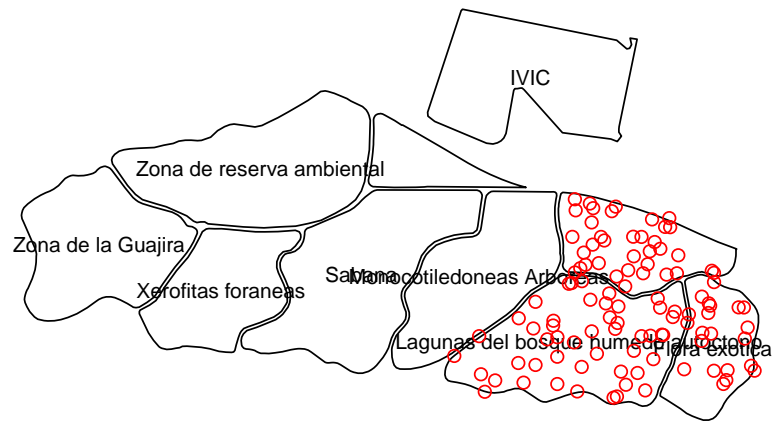
Para visualizar estos puntos en un mapa debemos definir sus atributos espaciales. Las funciones `coordinates` y `proj4string` del paquete *sp* permiten definir las coordenadas y proyección geográfica de los datos:

```
> coordinates(Puntos) <- c("Longitude", "Latitude")
> proj4string(Puntos) <- "+proj=longlat +datum=WGS84"
> plot(Puntos)
```



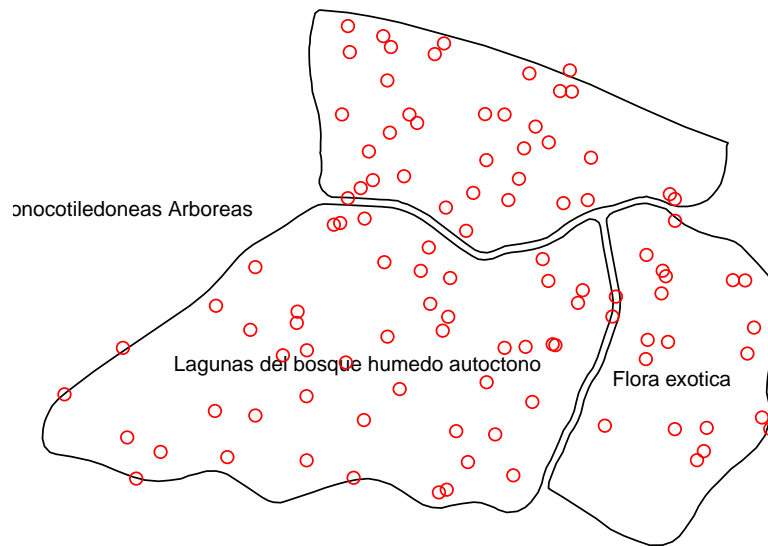
Añadimos un mapa de la zonificación del jardín para observar su ubicación relativa en el JBM:

```
> JBM.11 <- shapefile("~/CEBA/data/JardinBotanicoMaracaibo/ZonificacionJardin.shp")
> plot(JBM.11)
> text(JBM.11,"nombre",cex=.75)
> points(Puntos,col=2)
```



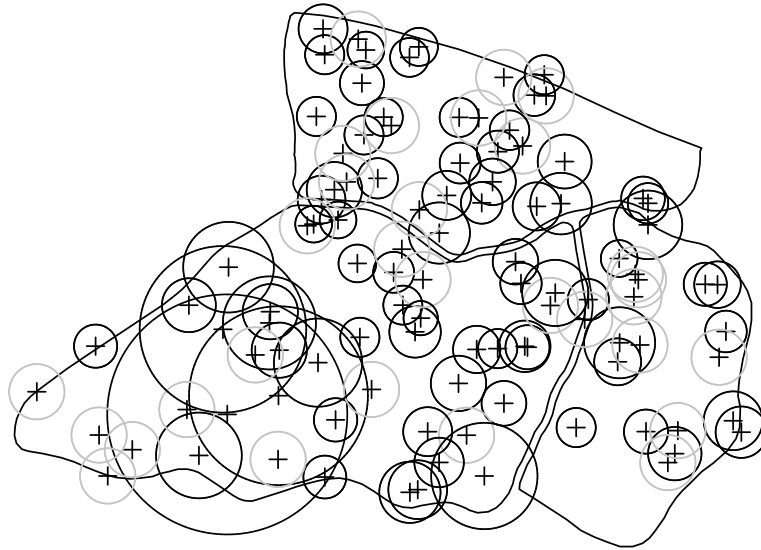
Vamos a enfocarnos en la sección oeste del JBM, donde están los puntos:

```
> plot(JBM.11[JBM.11@data$id %in% c(7,8,11),])
> text(JBM.11,"nombre",cex=.75)
> points(Puntos,col=2)
```



Para visualizar la incertidumbre (expresada en metros), vamos primero a proyectar los datos a UTM (función `spTransform`), hacer el gráfico con la función `plot`, y añadir círculos proporcionales a la incertidumbre con la función `draw.arc` del paquete *plotrix*. Los círculos oscuros fueron puntos caracterizados durante la actividad de campo, y los grises fueron los puntos que no pudimos caracterizar (columna `Anotador` con valor “No muestreado”).

```
> pts <- spTransform(Puntos, CRS("+proj=utm +zone=19n +datum=WGS84"))
> JBM.xy <- spTransform(JBM.ll, CRS("+proj=utm +zone=19n +datum=WGS84"))
> plot(JBM.xy[JBM.xy@data$id %in% c(7,8,11),])
> points(pts, pch=3)
> draw.arc(x=pts@coords[,1],
+         y=pts@coords[,2],
+         radius=pts@data$Incertidumbre,
+         deg1=0, deg2=360, lty=2,
+         col=c("grey77", 1)[1+(pts@data$Anotador != "No muestreado")])
>
```

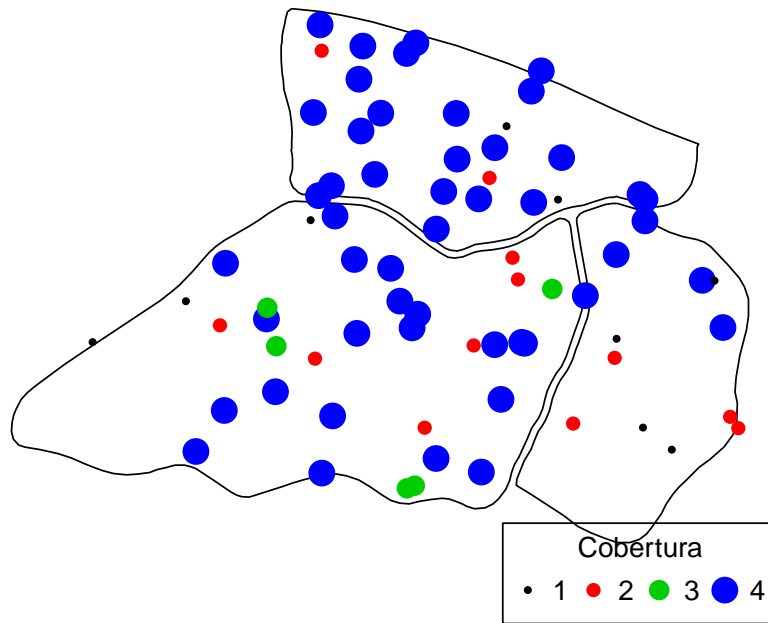


## 2. Caracterización de los puntos

Para visualizar las variables caracterizadas usamos las opciones de la función `points` para modificar el tamaño y color de los puntos según la variable de interés. Posteriormente se pueden simplificar las categorías para hacer más sencillo el análisis.

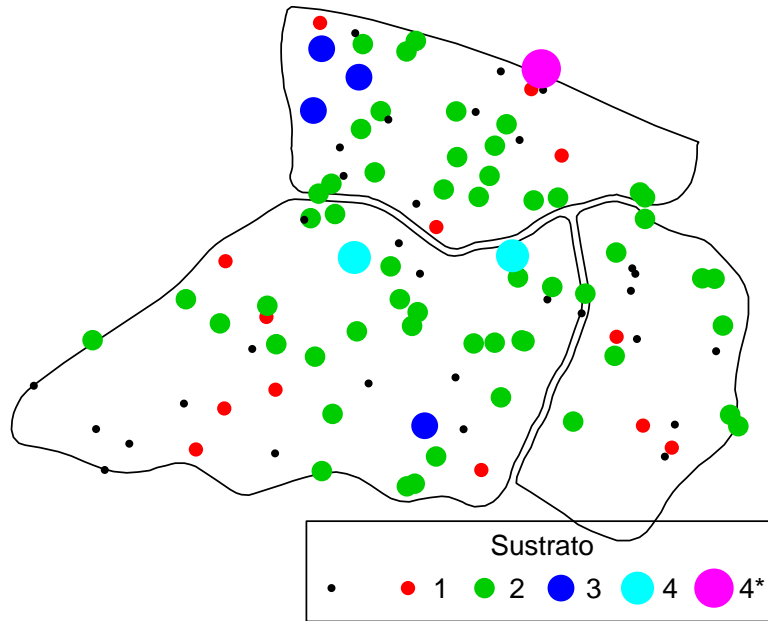
En el caso de la cobertura hay cuatro niveles de la variable:

```
> plot(JBM.xy[JBM.xy@data$id %in% c(7,8,11),])
> points(pts,col=pts@data$Cobertura,cex=pts@data$Cobertura/2,pch=19)
> legend("bottomright",legend=1:4,col=1:4,pch=19,pt.cex=c(1:4)/2,title="Cobertura",
```



En la columna de sustrato hay un punto caracterizado como “4\*”.

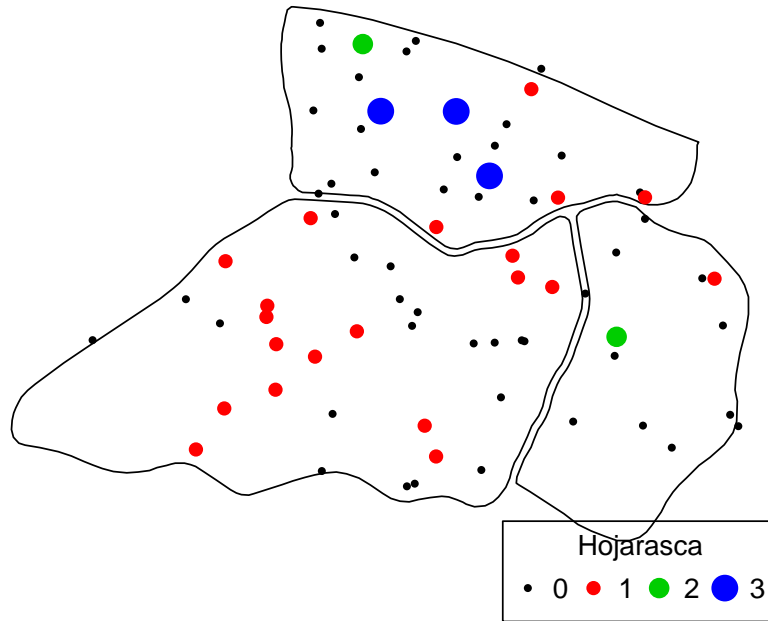
```
> plot(JBM.xy[JBM.xy@data$id %in% c(7,8,11),])
> points(pts,col=pts@data$Sustrato,cex=as.numeric(pts@data$Sustrato)/2,pch=19)
> legend("bottomright",legend=levels(pts@data$Sustrato),col=1:6,pch=19,pt.cex=c(1:6))
```



En hojarasca la escala empieza con cero.

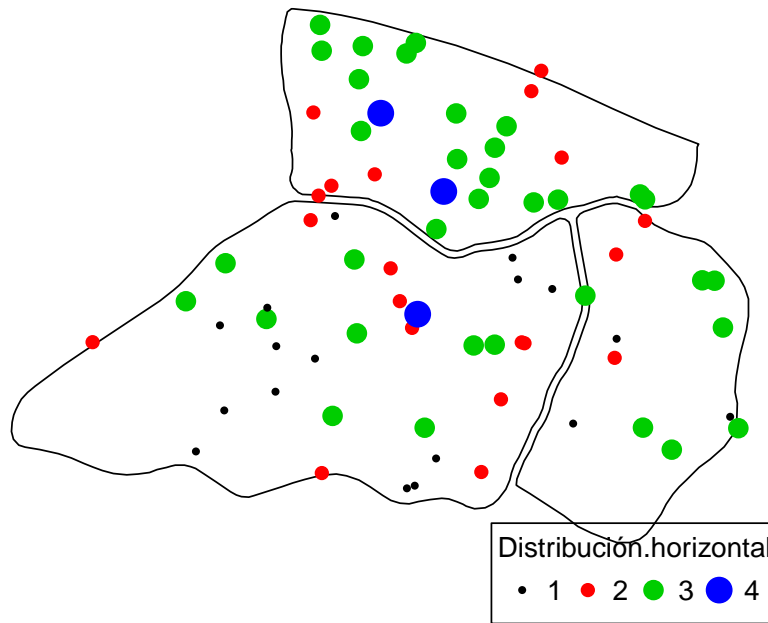
```
> plot(JBM.xy[JBM.xy@data$id %in% c(7,8,11),])
> points(pts,col=1+pts@data$Hojarasca,cex=(1+pts@data$Hojarasca)/2,pch=19)
> legend("bottomright",legend=0:3,col=1:4,pch=19,pt.cex=c(1:4)/2,title="Hojarasca",
```





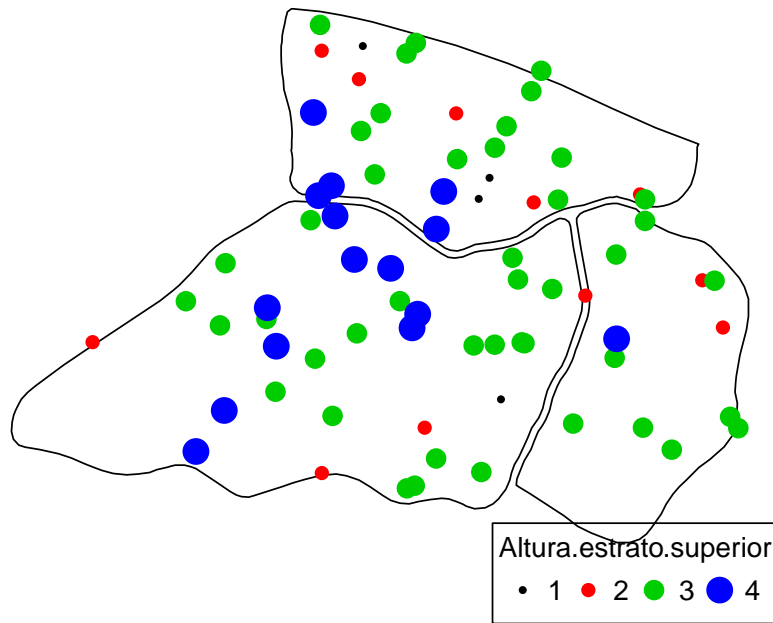
Para Distribución.horizonta hay cuatro categorías.

```
> plot(JBM.xy[JBM.xy@data$id %in% c(7,8,11),])
> points(pts,col=pts@data$Distribución.horizonta,cex=(pts@data$Distribución.horizonta)/2)
> legend("bottomright",legend=1:4,col=1:4,pch=19,pt.cex=c(1:4)/2,title="Distribución.horizonta")
```



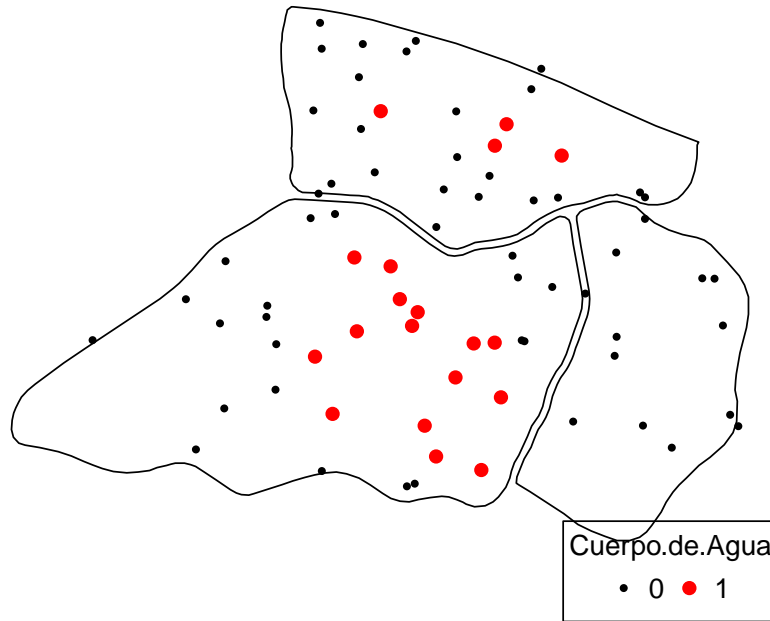
Altura.estrato.superior

```
> plot(JBM.xy[JBM.xy@data$id %in% c(7,8,11),])
> points(pts,col=pts@data$Altura.estrato.superior,cex=(pts@data$Altura.estrato.superior/2))
> legend("bottomright",legend=1:4,col=1:4,pch=19,pt.cex=c(1:4)/2,title="Altura.estrato.superior")
```



En Cuerpo.de.Agua solo hay dos categorías.

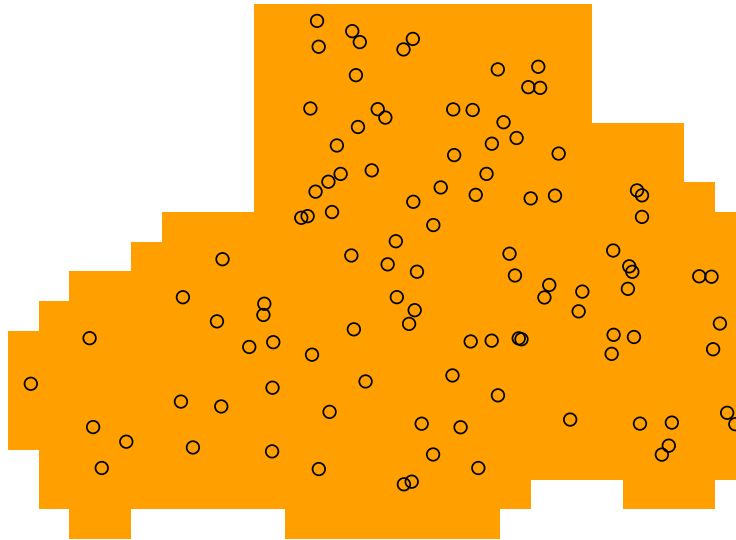
```
> plot(JBM.xy[JBM.xy@data$id %in% c(7,8,11),])
> points(pts,col=1+pts@data$Cuerpo.de.Agua,cex=(1+pts@data$Cuerpo.de.Agua)/2,pch=19)
> legend("bottomright",legend=c("0","1"),col=1:2,pch=19,pt.cex=c(1:2)/2,title="Cuerpo.de.Agua")
```



### 3. Predicción espacial

A partir de los puntos caracterizados realizamos una predicción espacial para todo el área de estudio. Primero generamos una capa espacial alrededor de los puntos muestreados:

```
> rnull <- raster(extent(JBM.xy[JBM.xy@data$id %in% c(7,8,11),]),
+                 nrows=18, ncols=24)
> mask <- distanceFromPoints(rnull,pts)<50
> JBM.grd <- SpatialPixelsDataFrame(points=coordinates(mask)[values(mask)==1,],
+                                   data=data.frame(values(mask)[values(mask)==1]),
+                                   proj4string=JBM.xy@proj4string)
> image(JBM.grd)
> points(pts)
```

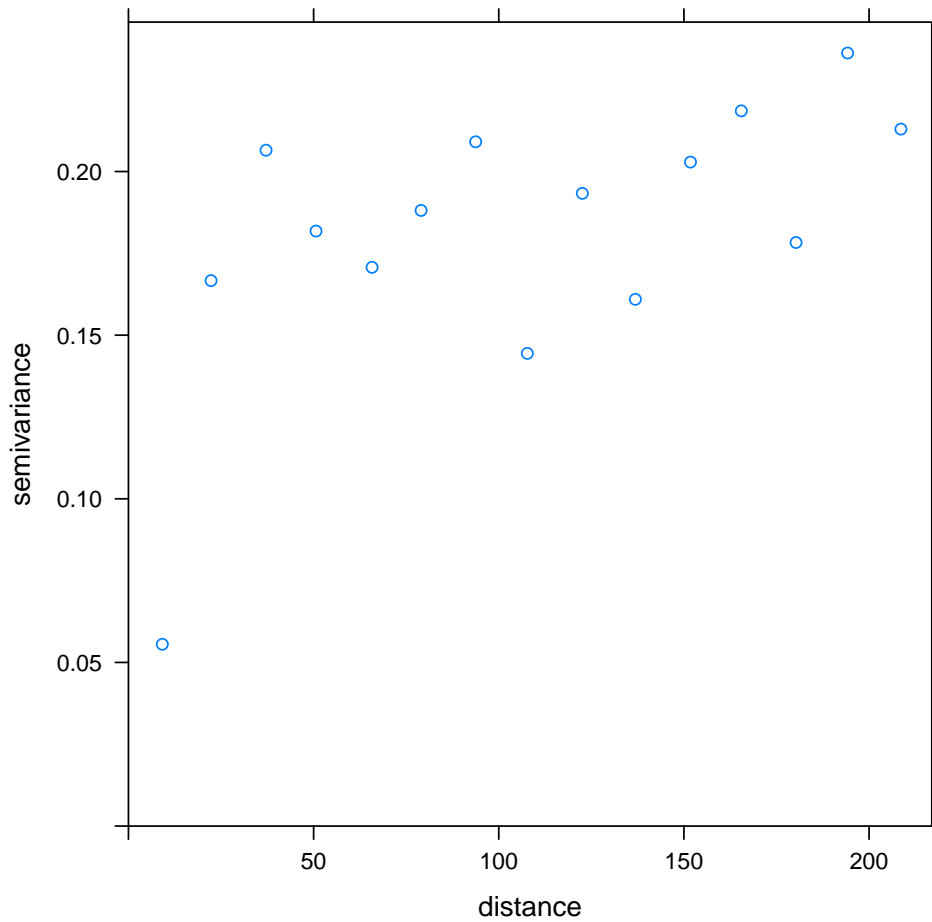


Para la predicción espacial vamos a utilizar el método de “kriging” de una variable indicadora. Vamos a usar las funciones en el paquete *gstats*.

```
> require(gstat)
```

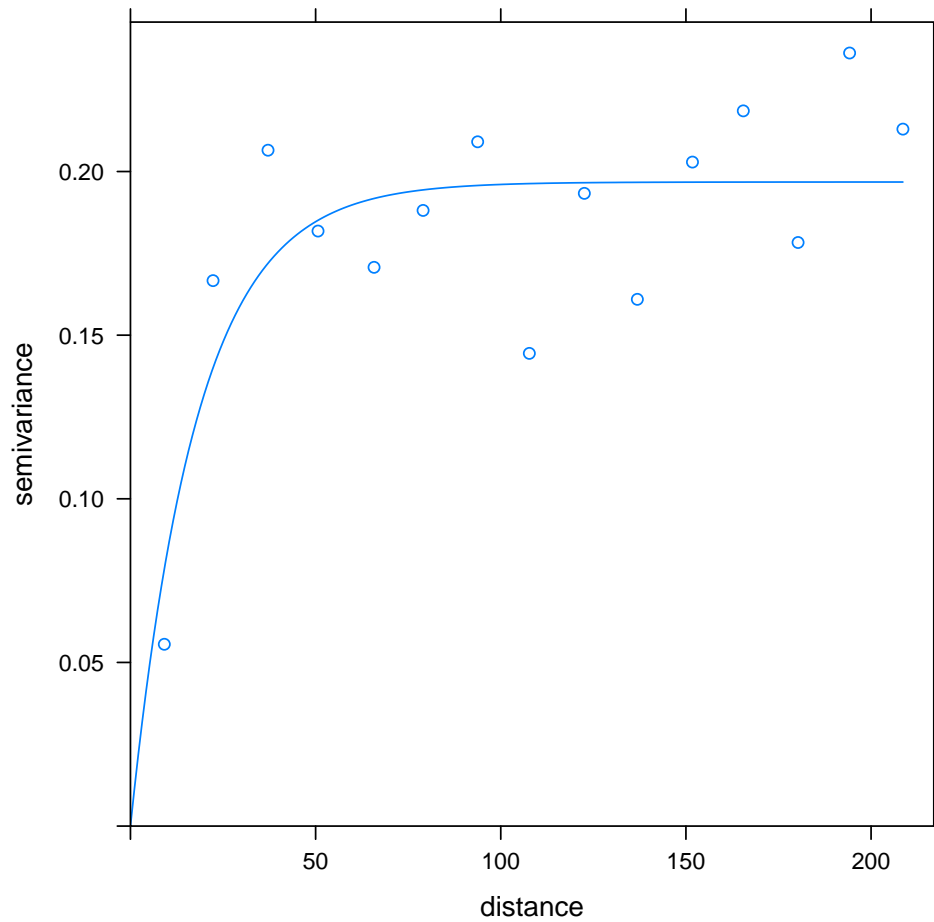
Vamos a simplificar nuestras variables para considerar solamente dos categorías. Por ejemplo en la Cobertura, vamos a considerar valores mayores a dos. Aplicamos la función `variogram` para observar la dependencia espacial de los datos en forma de un semi-variograma. Usamos un filtro (`ss`) para excluir las observaciones sin valores.

```
> ss <- !is.na(pts@data$Cobertura)
> v = variogram(I(Cobertura > 2)~1,pts[ss,])
> plot(v)
```



Observamos que la semi-varianza es baja menor cuando la distancia es menor a 20 metros, y luego se mantiene alrededor de 0,15 y 0,25. Para hacer la predicción espacial necesitamos definir la estructura espacial en terminos de un modelo. Aplicamos las funciones `fit.variogram` y `vgm` para ajustar el modelo a partir de unos parámetros iniciales. En este caso utilizamos un modelo exponencial.

```
> vm = fit.variogram(v, vgm(1, "Exp", 70, .1))  
> plot(v,vm)
```



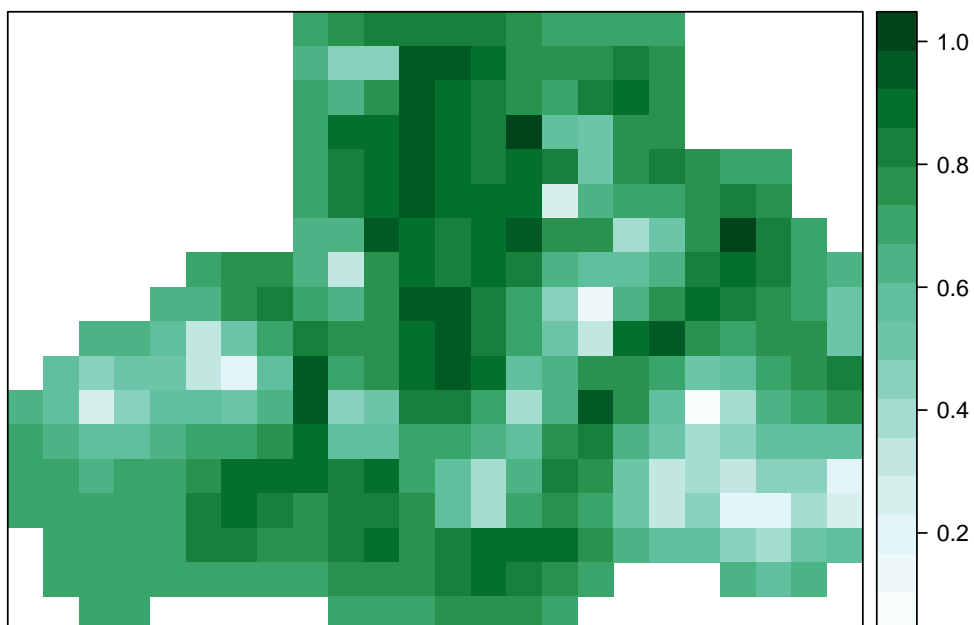
Ahora podemos utilizar la función `krige` para la interpolación espacial basada en “kriging” de una variable indicadora según el modelo de variograma estimado anteriormente.

```
> ik = krige(I(Cobertura > 2)~1, pts[ss,], JBM.grd, vm)
```

```
[using ordinary kriging]
```

```
> colores <- colorRampPalette(brewer.pal(9,"BuGn"))
```

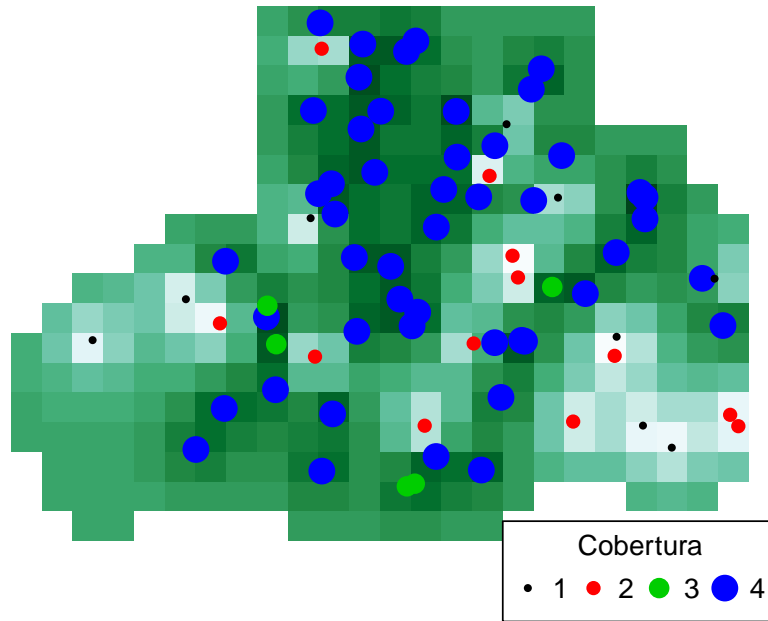
```
> print(spplot(ik[1],col.regions=colores(30)))
```



El resultado indica la probabilidad de observar valores de cobertura mayores a dos según los puntos caracterizados en campo. En el siguiente gráfico podemos ver la correspondencia entre los valores observados y los predichos. Notese que la predicción en lugares alejados de los sitios muestreados toma un valor promedio.

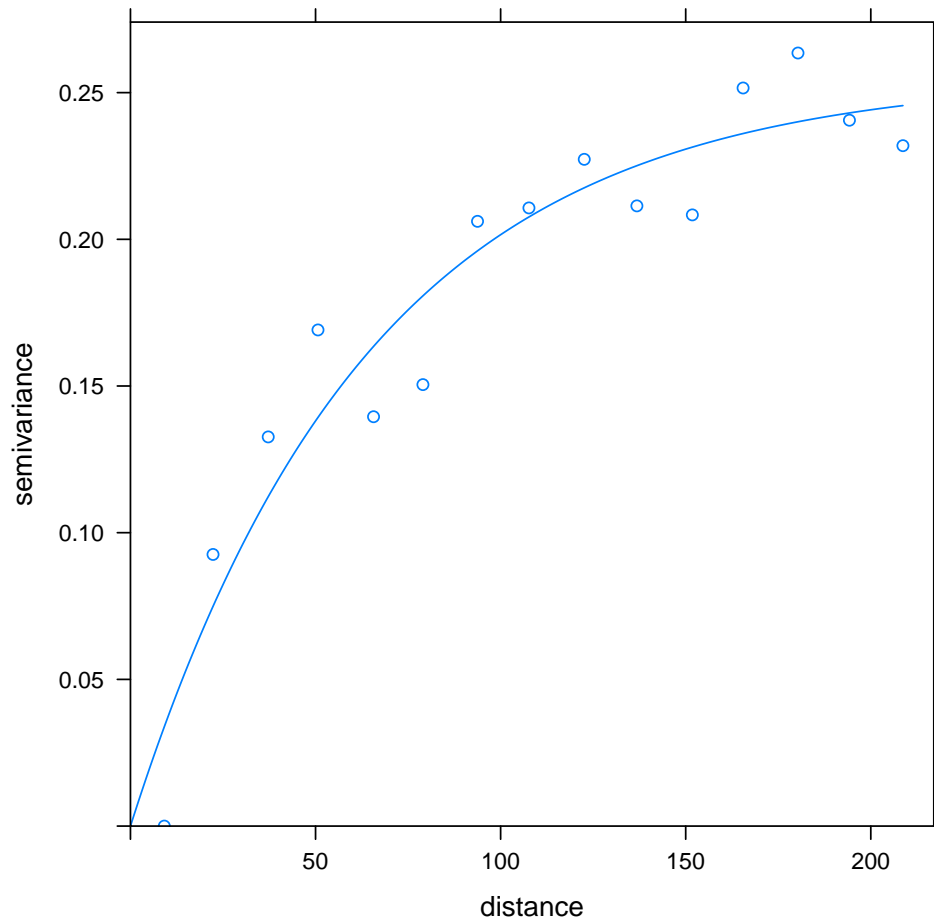
```
> image(ik[1],col=colores(30))
> points(pts,col=pts@data$Cobertura,cex=pts@data$Cobertura/2,pch=19)
> legend("bottomright",legend=1:4,col=1:4,pch=19,pt.cex=c(1:4)/2,title="Cobertura",
```





Aplicamos este mismo procedimiento a otras variables, por ejemplo la variable Cuerpo.de.Agua. En este caso el semi-variograma muestra una tendencia creciente.

```
> ss <- !is.na(pts@data$Cuerpo.de.Agua)
> v = variogram(I(Cuerpo.de.Agua > 0)~1,pts[ss,])
> vm = fit.variogram(v, vgm(.3, "Exp", 100, 0.1))
> plot(v,vm)
```

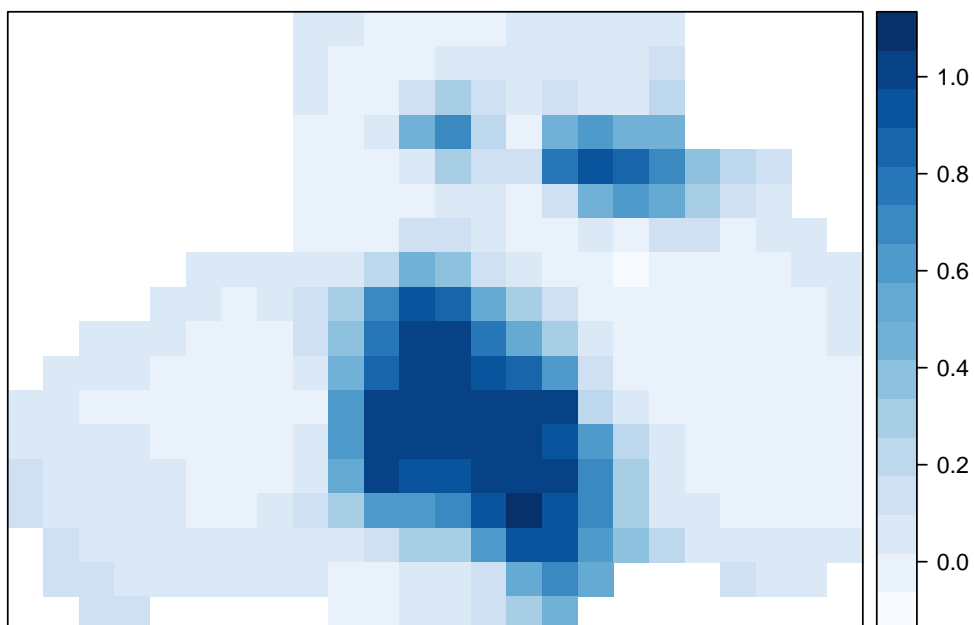


Aplicamos la función de `krige` y observamos la predicción espacial.

```
> colores <- colorRampPalette(brewer.pal(9,"Blues"))
> ik = krige(I(Cuerpo.de.Agua > 0)~1, pts[ss,], JBM.grd, vm)

[using ordinary kriging]

> spplot(ik[1],col.regions=colores(30))
```



En este caso observamos valores ligeramente menores a cero y mayores a uno. Redondeamos estos valores para ajustarlos al intervalo de probabilidades.

```
> summary(ik[[1]])
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
-0.06459	0.01008	0.05156	0.21450	0.30080	1.05500

```
> ik[[1]][ik[[1]]<0] = 0
```

```
> ik[[1]][ik[[1]]>1] = 1
```

```
> summary(ik[[1]])
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
0.00000	0.01008	0.05156	0.21570	0.30080	1.00000

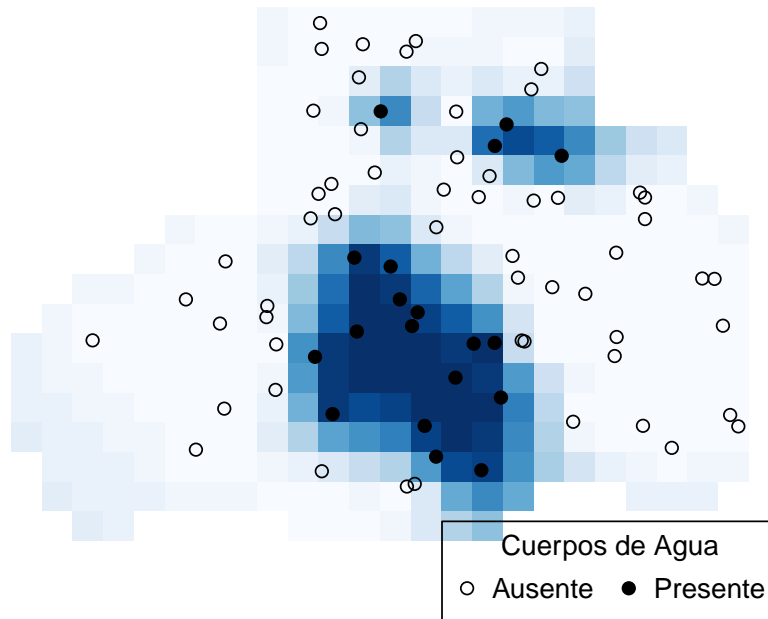
```
>
```

El mapa resultante nos indica la presencia de cuerpos de agua en el área de estudio.

```

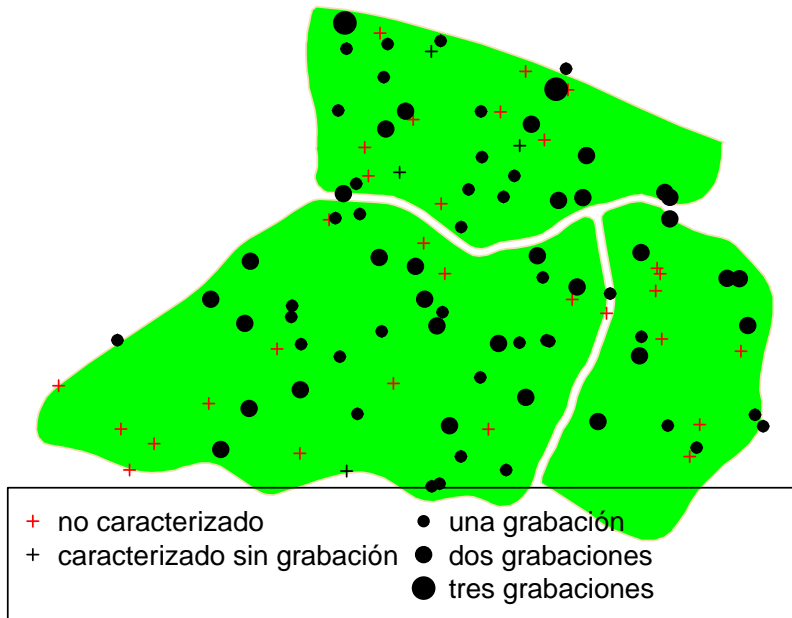
> image(ik[1],col=colores(30))
> points(pts,col=1,pch=c(1,19)[pts@data$Cuerpo.de.Agua+1])
> legend("bottomright",legend=c("Ausente","Presente"),col=1,pch=c(1,19),pt.cex=1,ti

```



## 4. Puntos muestreados y grabaciones

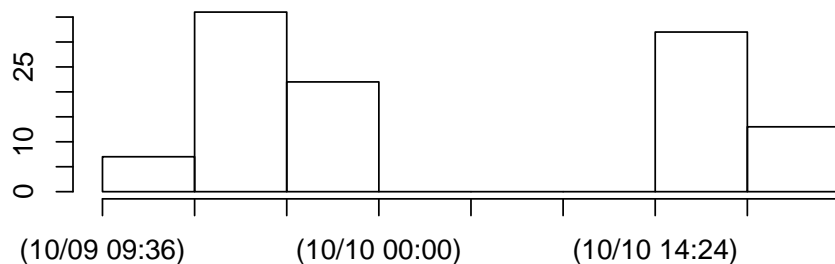
De los 105 puntos seleccionados originalmente se caracterizaron y muestrearon 75, pero sólo se cuentan con grabaciones para 71 puntos, pues hubo algunas grabaciones que no se realizaron correctamente.



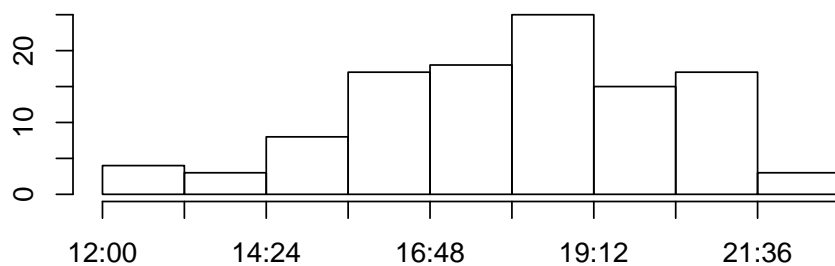
Para ver la distribución del esfuerzo de muestreo entre los dos días de trabajo de campo utilizamos la función `chron` del paquete *chron*

```
> require(chron)
> fechas <- chron(dates.=as.character(Muestreo$Fecha),
+               times.=sprintf("%s:00",as.character(Muestreo$Hini)),
+               format = c(dates = "ymd", times = "h:m:s"))
> horas <- chron(times.=sprintf("%s:00",as.character(Muestreo$Hini)),
+               format = c(dates = "ymd", times = "h:m:s"))
> layout(1:2)
> hist(fechas)
> title(main="Muestreos por fechas")
> hist(horas)
> title(main="Muestreos por horas")
```

### Muestreos por fechas

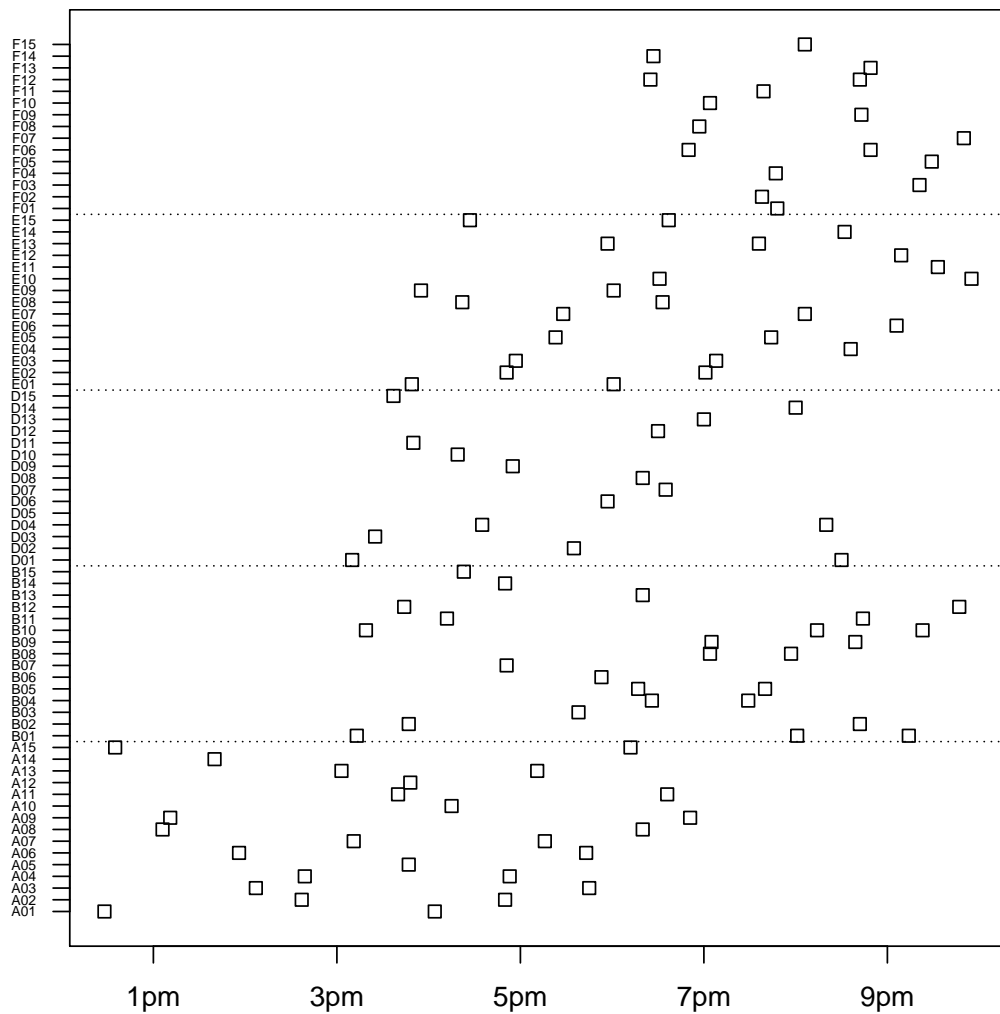


### Muestreos por horas



También podemos ver a que horas se realizó cada muestreo, por ejemplo el grupo A fue muestreado desde la 1 hasta las 7, y el grupo F sólo fue muestreado de 6 a 10, pero los otros tres grupos fueron muestreados entre las 3 y 10 pm.

```
> par(mar=c(2,2,0,0))
> stripchart(horas~Muestreo$Punto,axes=F)
> axis(1,c(13,15,17,19,21)/24,c("1pm","3pm","5pm","7pm","9pm"))
> axis(2,1:75,levels(Muestreo$Punto),
+      las=2, cex.axis=.5)
> box()
> abline(h=15*c(1,2,3,4)+.5,lty=3)
```



En las planillas se anotaron también las condiciones del muestreo:

```
> table(Muestreo$Viento)
```

```
0  1  2
73 35  2
```

```
> table(Muestreo$Ruido)
```

```
0  1  2
28 49 33
```

```
> table(Muestreo$Cielo)
```

```
0  1  2
49 44 17
```

La posición y sustrato del micrófono no se anotó consistentemente:

```
> table(Muestreo$Posición.mic)
```

	F	F	I	N/I no muestreado
	64	1	3	42
No muestreado				37
	7			

```
> table(Muestreo$Sustrato.mic)
```

	A	C	N/I
44	10	56	42
*SISTRATO HUMANO			
2			

La humedad y temperatura se anotaron con las unidades de medición, lo cual no es recomendable, además para la temperatura se mezclan grados Fahrenheit y Centígrados.

```
> table(Muestreo$Humedad)
```

	71,0%	71,4%	71,6%	71,8%	72,8%	7,3%	73,8%	74,2%	74,4%	74,5	74,5%	74,6%
44	1	1	2	1	1	1	1	1	1	1	1	1
74,7%	75,1%	75,6%	76,2%	76,4%	76,5%	76,9%	77,4%	77,7%	78,6%	78,8%	79,6	79,6%
1	1	1	1	1	1	1	1	2	1	1	1	1
79,8%	80,1%	80,3%	80,7	81,0%	81,1%	82,7%	82,8%	83,1%	83,3%	83,6%	84,3%	84,7%
1	1	1	1	1	1	1	1	1	1	1	1	1
85,8%	86,2%	86,5	86,6%	87,4%	N/A							
1	1	1	1	1	65							

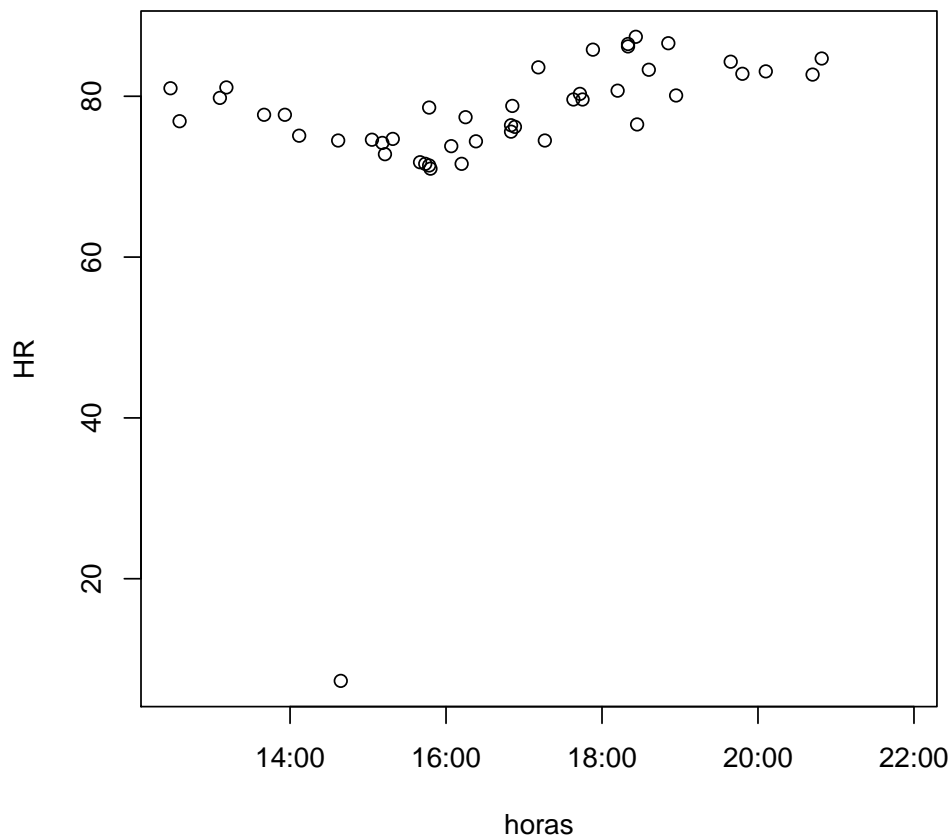
```
> table(Muestreo$Temperatura)
```

	25,6	25,9	27,0°C	27,1	27,2°C	27,4	27,4°C	27,8	28,8°C
44	1	2	1	1	2	1	1	1	1
29,5 °C	29,8	30,7	31,2	31,9	32,2	32,3	33,3	72,9	81,1°F
1	1	1	1	1	1	1	1	1	1
82,4	82,7°F	83,1	84,7	84,9 °F	85,1°F	86,1 °F	86,3	86,5	86,5°F
1	1	1	1	1	1	1	1	1	1
871,9	87,2 °F	87,4°F	87,4 °F	87,6°F	87,8 °F	87,9°F	88,3	88,3°F	88,5°F
1	1	2	1	1	1	1	2	1	1
88,7°F	N/A								
2	65								

Hacemos una variable nueva para la humedad relativa (HR) removiendo las anotaciones de porcentaje y cambiando las comas por puntos, y luego graficamos su comportamiento con respecto a la hora de muestreo:



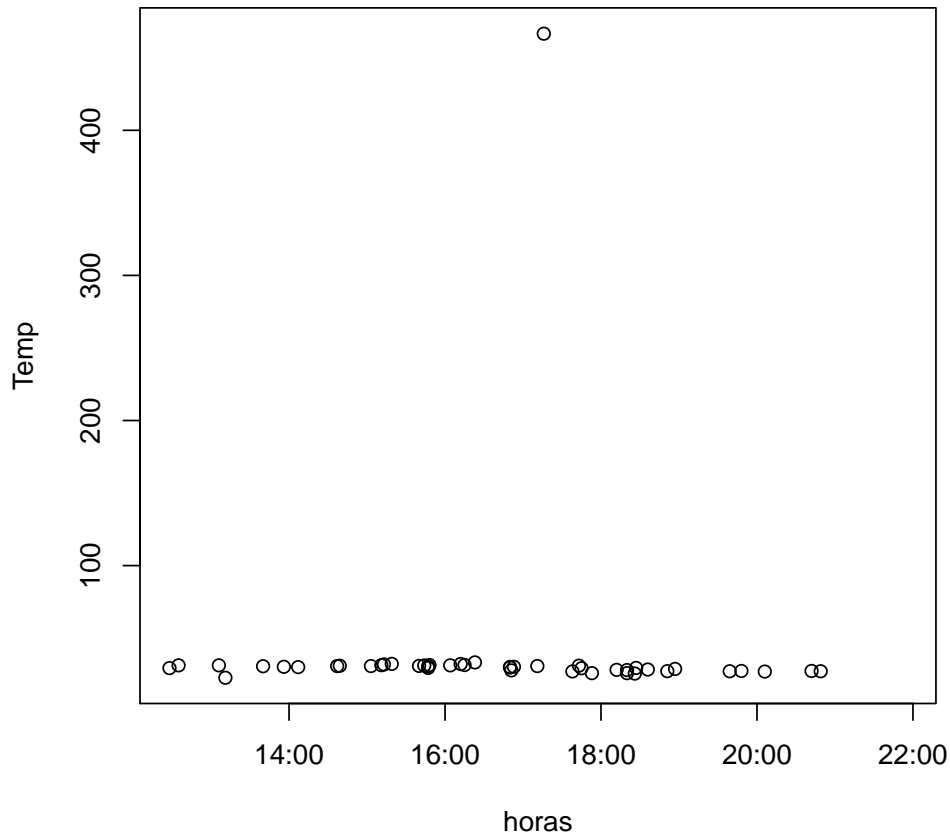
```
> HR <- as.numeric(sub("",".",sub("%","",Muestreo$Humedad)))
> plot(HR~horas)
>
```



Vemos que hay una observación atípica, con un valor muy bajo, probablemente se trata de un error de tipeo.

Hacemos lo mismo con la temperatura, aplicando una transformación de Fahrenheit a Celsius para las que tienen valores muy altos:

```
> Temp <- as.numeric(sub("",".",sub("[CF]","",Muestreo$Temperatura)))
> Temp[!is.na(Temp) & Temp>50] <- (Temp[!is.na(Temp) & Temp>50]-32)*5/9
> plot(Temp~horas)
```



Igualmente hay una observación atípica por un error de tipeo.

## 5. Pendientes

- Normalizar columnas en las tablas (temperatura, humedad) y revisar errores de tipeo
- Procesar grabaciones
- Utilizar *Raven* para detectar los cantos de una especie en las grabaciones
- Utilizar “krigging” para la predicción espacial de probabilidad de presencia de la especie seleccionada.
- Utilizar una regresión logística para la predicción de probabilidad de presencia tomando en cuenta covariables espaciales y temporales
- Aplicar un modelo mixto de presencia y detección para estimar la probabilidad de presencia y probabilidad de detección tomando en cuenta covariables espaciales y temporales

## Referencias

- BIVAND, R. & RUNDEL, C., 2012. *rgeos: Interface to Geometry Engine - Open Source (GEOS)*. URL <http://CRAN.R-project.org/package=rgeos>. R package version 0.2-7.
- CSARDI, G. & NEPUSZ, T., 2006. *The igraph software package for complex network research*. InterJournal, Complex Systems:1695. URL <http://igraph.sf.net>.
- HIJMANS, R. J. & VAN ETTEN, J., 2012. *raster: Geographic analysis and modeling with raster data*. URL <http://CRAN.R-project.org/package=raster>. R package version 2.0-08.
- JAMES, D. & HORNIK, K., 2011. *chron: Chronological Objects which Can Handle Dates and Times*. URL <http://CRAN.R-project.org/package=chron>. R package version 2.3-42. S original by David James, R port by Kurt Hornik.
- KEITT, T. H., BIVAND, R., PEBESMA, E. & ROWLINGSON, B., 2012. *rgdal: Bindings for the Geospatial Data Abstraction Library*. URL <http://CRAN.R-project.org/package=rgdal>. R package version 0.7-16.
- LEMON, J., 2006. *Plotrix: a package in the red light district of r*. R-News, 6:8–12.
- NEUWIRTH, E., 2011. *RColorBrewer: ColorBrewer palettes*. URL <http://CRAN.R-project.org/package=RColorBrewer>. R package version 1.0-5.
- PEBESMA, E. & BIVAND, R., 2005. *Classes and methods for spatial data in r*. R News, 5. URL <http://cran.r-project.org/doc/Rnews/>.
- PEBESMA, E. J., 2004. *Multivariable geostatistics in s: the gstat package*. Computers & Geosciences, 30:683–691.
- R DEVELOPMENT CORE TEAM, 2010. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL <http://www.R-project.org/>. ISBN 3-900051-07-0.