

ANEXOS

SISTEMA AUTOMÁTICO DE DETECCIÓN TEMPRANA DE AFECCIONES EN VIÑEDO: MILDIU Y OIDIO

Autor: Maialen Ausin Martin

Director: Marcos Rodrigues Mimbreno

**Máster Universitario en
Tecnologías de la información geográfica para la ordenación del
territorio: sistemas de información geográfica y teledetección**

Diciembre de 2014



**Universidad
Zaragoza**

**Departamento de Geografía
y Ordenación del Territorio**



Índice de anexos

Anexo 1: Cronograma de las prácticas	pág. 1
Anexo 2: Script crecimiento de mildiu	pág. 2
Anexo 3: Script Índice de Oídio	pág. 8
Anexo 4: Diagrama de flujo del modelo de mildiu	pág. 14
Anexo 5: Diagrama de flujo del modelo de oídio.....	pág. 15
Anexo 6: Script gráfico de crecimiento	pág. 16
Anexo 7: Mapas de crecimiento de mildiu e Índice de Oídio, Zaragoza	pág. 17
Anexo 8: Zoom a parcelas de crecimiento acumulado de mildiu en 9 días	pág. 23
Anexo 9: Metadatos de los <i>shapes</i> finales.....	pág. 24

ANEXO 1: CRONOGRAMA DE LAS PRÁCTICAS

Las prácticas se han desarrollado de la siguiente manera, con una media de 6 horas al día.

JULIO

L	M	M	J	V	S	D
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

AGOSTO

L	M	M	J	V	S	D
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

SEPTIEMBRE


L	M	M	J	V	S	D
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					


OCTUBRE


L	M	M	J	V	S	D
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		


NOVIEMBRE


L	M	M	J	V	S	D
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30


 Búsqueda de información de mildiu y descarga de software: gvSIG y R


 Ejemplo práctico en gvSIG y redacción del tutorial


 Script:SIG

 Redacción de teoría de mildiu


 Script: selección de porcentaje de crecimiento


 Comprensión y redacción del script del crecimiento

 Script Oidio

 Búsqueda y tratamiento de datos de Zaragoza

 Visor

 Redacción del proyecto

 Correcciones

ANEXO 2: SCRIPT CRECIMIENTO DE MILDUI

```
#####SELECCION DE PORCENTAJE DE CRECIMIENTO#####

#Establecimiento del directorio de trabajo

setwd("C:/Users/MAIALEN/Desktop/PROYECTO MASTER/R_zaragoza_mildiu")

#PASO 1: CÁLCULO DE CRECIMIENTO DE OIDIO

#Lectura de la tabla Goidanich, tabla base del Modelo de predicción de Mildiu.

G <- read.table("Goidanich.csv", header=TRUE, sep=";")

##Se listan todos los archivos cuyos nombres terminen en datos.csv.
#Estos ficheros serán los que contengan la Temperatura y la humedad promedio registrada por cada estación meteorológica por día

lista <- list.files(pattern="*datos.csv")

#Por tanto, habrá dos archivos de entrada: la tabla Goidanich que se utilizará para establecer los valores de crecimiento y
#la tabla con la temperatura y la humedad.

primerfichero<-1

#Para todos los archivos contenidos en la lista, se lee la tabla y para los días en los que no haya datos, se pondrá un cero.

for (i in 1:length(lista)){

datos <- read.table(lista[i], header=TRUE, sep=";")

datos[is.na(datos)]=0

#El establecimiento del valor de crecimiento de mildiu se realiza en función de dos variables: TEMPERATURA Y HUMEDAD

#TEMPERATURA:

#A continuación, se presenta el bucle que va a asignar un valor de crecimiento de mildiu (de la tabla Goidanich) por cada valor
#de temperatura registrada (una al día).
#Para ello, se procede a leer cada tabla contenida en la lista creada, guardando esa lectura en un objeto llamado datos.
#Los valores de crecimiento se van a asignar en función de la diferencia en temperatura existente entre los valores de las tablas.
```

#El funcionamiento del bucle es el siguiente: Se recorren las filas del objeto datos y las filas del objeto G. Se resta el valor de temperatura de la tabla de datos con el valor de temperatura de la tabla G.

#Para cada fila de datos y de G, se observa la diferencia de temperatura con respecto a la anterior y se genera el valor absoluto para que los valores sean siempre positivos. La diferencia con respecto a la fila anterior se almacena y se compara con la siguiente.

#Si la fila anterior es menor a la fila que se está observando, la ANTERIOR A LA que se está observando es la más parecida. Por tanto, nos quedamos con esa fila.

##diffT= diferencia anterior

##Diff.temp= diferencia temporal (la fila que se esta observando)

#Este mecanismo funciona cuando hay una fila anterior con la que comparar; para la primera fila esto no se cumple.

#Para que funcione para la primera fila, se inicia la variable con la que registramos la diferencia (diffT) con un valor muy alto.

#Por tanto, la fila G sólo cambia cuando la diferencia de la fila anterior es mayor a la actual ya que cuando eso sucede, la variable G se actualiza.

#Así, el resultado que se obtiene es el siguiente:para cada temperatura de datos, se sabe en qué columna de Tmax esta la temperatura que mas se le parece.

```
primerafila<-1

for (i in 1:nrow(datos))
{
diffT<-999
diffT.temp<-0
filaG<-0
  for (n in 1:nrow(G))
  {
    diffT.temp <- abs(datos[i,1]-G[n,1])    #Valor absoluto de la diferencia entre las temperaturas
    if (diffT>diffT.temp)
    {
      filaG <- n
      diffT <- diffT.temp
    }
  }
  if (primerafila==1){
    crecimiento<-G[filaG,2:3]
    primerafila<-0}
  else {crecimiento<-rbind(crecimiento,G[filaG,2:3])}
}
```

#Se va a generar una tabla que contiene el valor de cada temperatura de datos junto con el valor de la temperatura de la tabla Goidanich que
#más se le parece junto con los valores que corresponden a esa misma fila. Es decir la tabla tendrá las siguientes columnas: las dos columnas
#del objeto datos (temperatura y humedad) y el valor de crecimiento de mildiu cuando la humedad es superior e inferior a 75 con esa temperatura.

```
datosCrec <- cbind(datos[,1:2],crecimiento)
```

```
#HUMEDAD
```

#Este bucle añade una nueva columna a la tabla anteriormente mencionada. Este bucle recorrerá todas las filas del objeto datosCrec y en función
#de si la temperatura es mayor o menor de 75, seleccionará un valor de crecimiento y lo copiará a la nueva columna

```
crecimiento<-0  
for(j in 1:nrow(datosCrec)){  
  if(datosCrec[j,2]<75){crecimiento<-rbind(crecimiento,datosCrec[j,3])}  
  else{crecimiento<-rbind(crecimiento,datosCrec[j,4])}  
}
```

```
datosCrec <- cbind(datosCrec,crecimiento[2:nrow(crecimiento),])
```

#Se crea una tabla que contiene las siguientes columnas: temperatura, humedad, valor de crecimiento con la humedad inferior al 75%, valor de
#crecimiento con la humedad superior al 75% y el valor de crecimiento definitivo (teniendo en cuenta el dato de humedad)

```
write.table(datosCrec,"datosCrec.csv",row.names=FALSE,sep=";",dec=".")
```

```
#PASO 2: CÁLCULO DE CRECIMIENTO ACUMULADO
```

#A continuación se procede a calcular el porcentaje de acumulado. Este bucle recorre todas las filas del objeto datosCrec y va sumando el
#crecimiento de cada fila. Así, se obtiene la suma total es decir, el crecimiento acumulado total. Este valor será el que se utilizará para
#realizar la interpolación.

```
acumulado=datosCrec[1,5]  
  
for (i in 2:nrow(datosCrec))  
{  
  acumulado<-rbind(acumulado,datosCrec[i,5]+acumulado[length(acumulado)])  
}  
  
prueba<-cbind(datosCrec,acumulado)  
if(primerfichero==1){
```

```

estaciones<-sum(prueba[5])
primerfichero<-0
}
else{
estaciones<-rbind(estaciones,sum(prueba[5]))
}
}
print(estaciones)

#Se crea una tabla que recogerá el valor de crecimiento acumulado de Mildiu para cada estacion

write.table(estaciones,"estaciones.csv",row.names=FALSE,sep=";",dec=".")

#####PARTE SIG#####

#PASO 3: CREACIÓN DE LA TABLA A UTILIZAR EN LA INTERPOLACIÓN

#Se cargan las librerías necesarias

library(sp)
library(rgdal)
library(gstat)
library(raster)
library(maptools)

#Se lee la capa de puntos
puntos<-readShapeSpatial("estaciones_zaragoza.shp")

#Se exporta la tabla de atributos de la capa de puntos a un archivo .csv
write.table(puntos@data,"estacionestabla.csv",row.names=FALSE,sep=";",dec=".")

#Se lee el archivo CSV creado.
estacionestabla<-read.table("estacionestabla.csv", header=TRUE, sep=";")

#Se crea un nuevo objeto que contenga el valor de crecimiento acumulado y la información de la tabla de atributos del shape.
#Es decir, las coordenadas.
prueba3<-cbind(estacionestabla,estaciones)

#Se le cambia el nombre a las columnas

```

```

colnames(prueba3)=c("Estacion", "X", "Y", "X.mildiu")

#Se crea la tabla con la información anteriormente mencionada
write.table(prueba3,"tabla.csv",row.names=FALSE,sep=";",dec=".")

#PASO 4: INTERPOLACIÓN

#Lectura de la tabla con los datos a interpolar y su conversión en SpatialPointsDataFrame (coordinates())
tabla <- read.table("tabla.csv", header=TRUE, sep=";")
coordinates(tabla)<-c("X", "Y")

#Se crea un data frame con los puntos
coords<-data.frame(tabla@coords)

#Lectura del poligono de parcelas
parcelas<-readShapeSpatial("parcelas.shp")
#plot(parcelas)

#Obtención del area de estudio a partir del poligono de ZARAGOZA

zaragoza<-readShapeSpatial("Zaragoza.shp")
ext<-extent(zaragoza)

#Se establece la resolución del pixel del raster de salida
resol<-250

#Se crea el raster destino a partir del extent de la nube de puntos a interpolar
r <- raster(extent(ext),ncol=round((ext@xmax-ext@xmin)/resol), nrow=round((ext@ymax-ext@ymin)/resol))
res(r) <- resol

#El raster se convierte en SpatialPointsDataFrame para usarlo en la interpolación
p <- data.frame(rasterToPoints(r))
#Extraemos las coordenadas
coordinates(p)<-c("x", "y")

#Se ajusta el modelo IDW. El argumento idp controla el exponente de la función IDW
idw.out <- idw(X.mildiu ~ 1,tabla, p, idp = 2)

#Se dibuja el mapa en R

```



```

#spplot(idw.out, "var1.pred")

#Se crea la capa raster para poder usarla en SIG
rasterize(idw.out,r,idw.out@data$var1.pred,filename="idw.tif",overwrite=TRUE)

#SE RECORTA EL RASTER SEGUN LA PROVINCIA
r2<-raster("idw.tif")
r2.mask<-mask(r2, zaragoza,"C:/ms4w/apps/pmapper/pmapper_demodata/idw_mildiu.tif",overwrite=TRUE)

#Se obtiene la altura media de cada parcela
mildiu.parc<-extract(r2, parcelas, fun=mean, small=TRUE)

#Se une el resultado del zonal a la capa de parcelas
parcelas2<-spCbind(parcelas,mildiu.parc)

#Se establece una paleta de colores para el mapa
rgb.palette <- colorRampPalette(c("darkgreen", "yellow","red"), space = "rgb")
#Ploteamos el mapa en R estableciendo unas rupturas, utilizando la paleta de colores anteriormente creada y establecemos el titulo del mapa.
spplot(parcelas2, "mildiu.parc",breaks=c(75,80,85,90,95,100,105),col.regions=rgb.palette(20),main="PORCENTAJE DE CRECIMIENTO DE MILDIU")

#se crea el shape con el atributo que contiene el porcentaje de crecimiento de mildiu
writePolyShape(parcelas2, "C:/ms4w/apps/pmapper/pmapper_demodata/parcelas_mildiu.shp")

```

ANEXO 3: SCRIPT INDICE OIDIO

```
#####SELECCION DEL INDICE DE OIDIO#####

#Establecimiento del directorio de trabajo
setwd("C:/Users/MAIALEN/Desktop/PROYECTO MASTER/R_zaragoza_Oidio_9dias")

#Lectura de la tabla oidio, tabla base del Modelo de predicción de Oidio.
O <- read.table("oidio.csv", header=TRUE, sep=";")

#PASO1: CÁLCULO DE COORDENADAS

#Lectura de los datos necesarios: Estos dos archivos contienen una única columna cada uno que corresponde con la Tmax
#y Tmin de referencia de la tabla de Oidio respectivamente.

Tmax <- read.table("tmax.csv", header=TRUE, sep=";")
Tmin <- read.table("tmin.csv", header=TRUE, sep=";")

###Se listan todos los archivos cuyos nombres terminen en datos.csv.
#Estos ficheros serán los que contengan la Tmax y la Tmin registrada por cada estación meteorológica por día

lista <- list.files(pattern="*datos.csv")

#Para todos los archivos contenidos en la lista, se lee la tabla y para los días en los que no haya datos, se pondrá un cero.

primerfichero<-1

for (i in 1:length(lista)){

datos <- read.table(lista[i], header=TRUE, sep=";")

datos[is.na(datos)]=0

#El establecimiento del índice de Oidio se da a partir del cruce de dos variables de la tabla oidio: Tmax (fila) y Tmin (columna)
#Para poder establecer el índice de oidio para cada día, en primer lugar, es necesario buscar las coordenadas que ocupan los
#índices que nos interesan en la tabla base.
#Para ello, se dispone de dos tablas. Cada una contiene una unica columna; en una estará la Tmax base y en la otra la Tmin base.
#Se van a construir dos bucles, uno por cada tabla

#PARA LA TEMPERATURA MÁXIMA
#El funcionamiento del bucle es el siguiente:se recorren las filas del objeto datos y las filas del objeto Tmax.A cada valor de
```

#Tmax registrado en datos se le resta el valor de Tmax y se genera el valor absoluto para que los valores sean siempre positivos.
 #Así, se va observando la diferencia de temperatura con respecto a la fila anterior y se almacena, permitiendola comparar con la
 #siguiente. Esta diferencia entre filas va a servir para encontrar el valor de Tmax base que más se le parezca. Esto permitirá
 #seleccionar la fila en la que se va a encontrar el índice correspondiente a esa temperatura.

#Al igual que en mildiu, si la fila anterior es menor a la fila que se está observando, la ANTERIOR A LA que se está observando
 #es la más parecida.
 #Por tanto, nos quedamos con esa fila.
 #**diffT= diferencia anterior
 #**Diff.temp= diferencia temporal (la fila que se esta observando)

#Este mecanismo funciona cuando hay una fila anterior con la que comparar; pero para la primera fila esto no se cumple.
 #Para este caso, se inicia la variable con la que se registra la diferencia (diffT) con un valor muy alto.
 #Por tanto, la fila G sólo cambia cuando la diferencia de la fila anterior es mayor a la actual ya que cuando eso sucede,
 #la variable G se actualiza

```
primerafila<-1

for (i in 1:nrow(datos))
{
  diffT<-999
  diffT.temp<-0
  filaG<-0
  for (n in 1:nrow(Tmax))
  {
    diffT.temp <- abs(datos[i,1]-Tmax[n,1])
    if (diffT>diffT.temp)
    {
      filaG <- n
      diffT <- diffT.temp
    }
  }
  if (primerafila==1){
    crecimiento<-filaG
    primerafila<-0}
  else {crecimiento<-rbind(crecimiento,filaG)}
}
```

#El objeto resultante de este bucle contiene las coordenadas de las Tmax. Con ellas, se crea una tabla que posteriormente se lee.

```

write.table(crecimiento,"crec_max.csv",row.names=FALSE,sep=";",dec=".")

crec_max<- read.table("crec_max.csv", header=TRUE, sep=";")

#PARA LA TEMPERATURA MINIMA se sigue exactamente el mismo proceso.Sólo que en este caso, los datos de Tmin están en la columna
#2 y no en la 1 del objeto datos.

primerafila<-1

for (i in 1:nrow(datos))
{
diffT<-999
diffT.temp<-0
filaG<-0
  for (n in 1:nrow(Tmin))
  {
    diffT.temp <- abs(datos[i,2]-Tmin[n,1])
    if (diffT>diffT.temp)
    {
      filaG <- n
      diffT <- diffT.temp
    }
  }
  if (primerafila==1){
    crecimiento<-filaG
    primerafila<-0}
  else {crecimiento<-rbind(crecimiento,filaG)}
}

write.table(crecimiento,"crec_min.csv",row.names=FALSE,sep=";",dec=".")
crec_min<- read.table("crec_min.csv", header=TRUE, sep=";")

#Una vez ejecutados los dos bucles, se crea un objeto que contenga las coordenadas tanto para las Tmax como para las Tmin
coord<-cbind(crec_max,crec_min)

#PASO 2:BUSQUEDA DEL VALOR DEL ÍNDICE DEL OIDIO
#El siguiente bucle va a recorrer las filas y las columnas de la tabla base en busca del indice correspondiente.
#El resultado va a ser un listado con el indice correspondiente por cada temperatura.

```

```

primerafila<-1
for (i in 1:nrow(coord))
{if(primerafila==1){
indice<-O[coord[i,1],coord[i,2]]
primerafila<-0}
else{indice<-rbind(indice, (O[coord[i,1],coord[i,2]]))}
}

#PASO 3: CÁLCULO DEL ÍNDICE ACUMULADO

#A continuación se procede a calcular el porcentaje de acumulado. Este bucle recorre todas las filas del objeto indice
#y va sumando el crecimiento de cada fila. Así, se obtiene la suma total es decir, el crecimiento acumulado total.
#Este valor será el que se utilizará para realizar la interpolación.

{if(primerfichero==1){
indice_oidio<-sum(indice)
primerfichero<-0
}
else{
indice_oidio<-rbind(indice_oidio,sum(indice))
}
}
}
print(indice_oidio)

#Se crea una tabla con los indices acumulados

write.table(indice_oidio,"estaciones_indice_oidio.csv",row.names=FALSE,sep=";",dec=".")

#####PARTE SIG#####

#PASO 4: PREPARACIÓN DE LA TABLA A UTILIZAR EN LA INTERPOLACIÓN

#Se cargan las librerías necesarias
library(sp)
library(rgdal)
library(gstat)
library(raster)
library(maptools)

```

```

#Se procede a la extracción de las coordenadas a utilizar en la interpolación a partir del shape de puntos
#y se crea un nuevo archivo que contiene las coordenadas

puntos<-readShapeSpatial("puntos.shp")
write.table(puntos@data,"estacionestabla.csv",row.names=FALSE,sep=";",dec=".")

estacionestabla<-read.table("estacionestabla.csv", header=TRUE, sep=";")

#Se lee la tabla que contiene los índices de oídio
estaciones_indice_oidio<-read.table("estaciones_indice_oidio.csv", header=TRUE, sep=";")

#Se crea un nuevo objeto que combine la tabla que contiene las coordenadas y la tabla de los índices.
prueba3<-cbind(estacionestabla,estaciones_indice_oidio)

#Se renombran las columnas

colnames(prueba3)=c("Estacion","X","Y","X.Oidio")

#Se crea una tabla que va a tener la siguiente información: nombre de estación, coordenadas e índice de oídio.
write.table(prueba3,"tabla.csv",row.names=FALSE,sep=";",dec=".")

PASO 5: INTERPOLACION

#Leemos la tabla con los datos a interpolar y la convertimos en SpatialPointsDataFrame (coordinates())
tabla <- read.table("tabla.csv", header=TRUE, sep=";")
coordinates(tabla)<-c("X","Y")

#Creamos un data frame con los puntos
coords<-data.frame(tabla@coords)

#Leemos poligono de parcelas
parcelas<-readShapeSpatial("parcelas.shp")
#plot(parcelas)

#Obtenemos area de estudio del poligono de ZARAGOZA

zaragoza<-readShapeSpatial("Zaragoza.shp")
ext<-extent(zaragoza)

```

```

#Resolucion del pixel del raster de salida
resol<-250

#el raster de destino a partir del extent de la nube de puntos a interpolar
r <- raster(extent(ext),ncol=round((ext@xmax-ext@xmin)/resol), nrow=round((ext@xmax-ext@xmin)/resol))
res(r) <- resol

#Convertimos el raster r en SpatialPointsDataFrame para usarlo en la interpolacion
p <- data.frame(rasterToPoints(r))
coordinates(p)<-c("x","y")

#Ajustamos el modelo IDW. El argumento idp controla el exponente de la funcion IDW
idw.out <- idw(X.Oidio ~ 1,tabla, p, idp = 2)

#Dibujamos el mapa en R
#spplot(idw.out, "var1.pred")

#Creamos la capa raster para poder usarla en SIG
rasterize(idw.out,r,idw.out@data$var1.pred,filename="idw.tif",overwrite=TRUE)

#Se recorta el raster IDW utilizando el shape de la provincia de Zaragoza
r2<-raster("idw.tif")

r2.mask<-mask(r2, zaragoza,"C:/ms4w/apps/pmapper/pmapper_demodata/idw_oidio.tif",overwrite=TRUE)

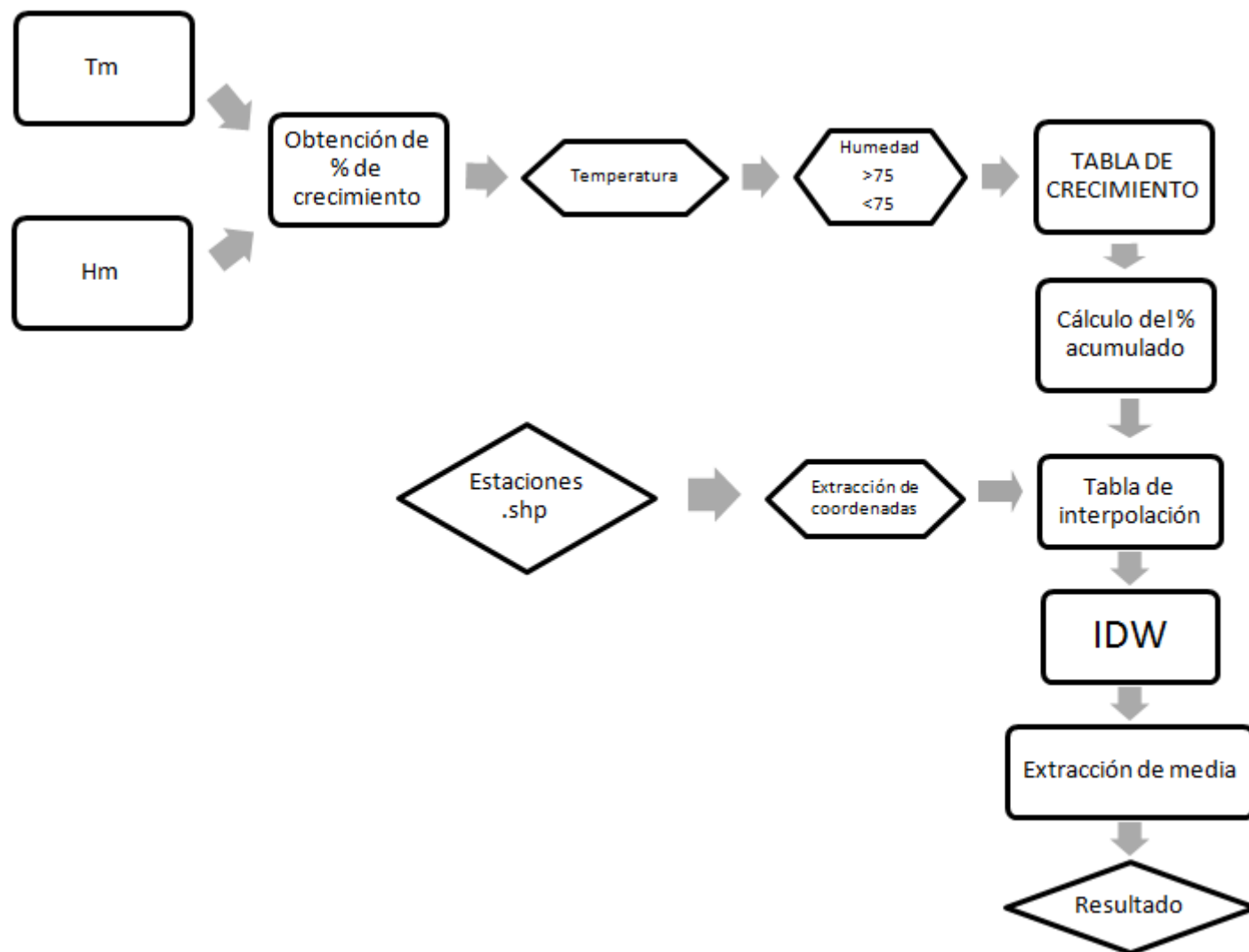
#Obtenemos la altura media de cada parcela
oidio.parc<-extract(r2, parcelas, fun=mean, small=TRUE)

#Unimos el resultado del zonal a la capa de parcelas
parcelas2<-spCbind(parcelas,oidio.parc)
rgb.palette <- colorRampPalette(c("darkgreen", "yellow","red"), space = "rgb")
spplot(parcelas2, "oidio.parc",breaks=c(75,80,85,90,95,100,105),col.regions=rgb.palette(20),main="PORCENTAJE DE CRECIMIENTO DE OIDIO")

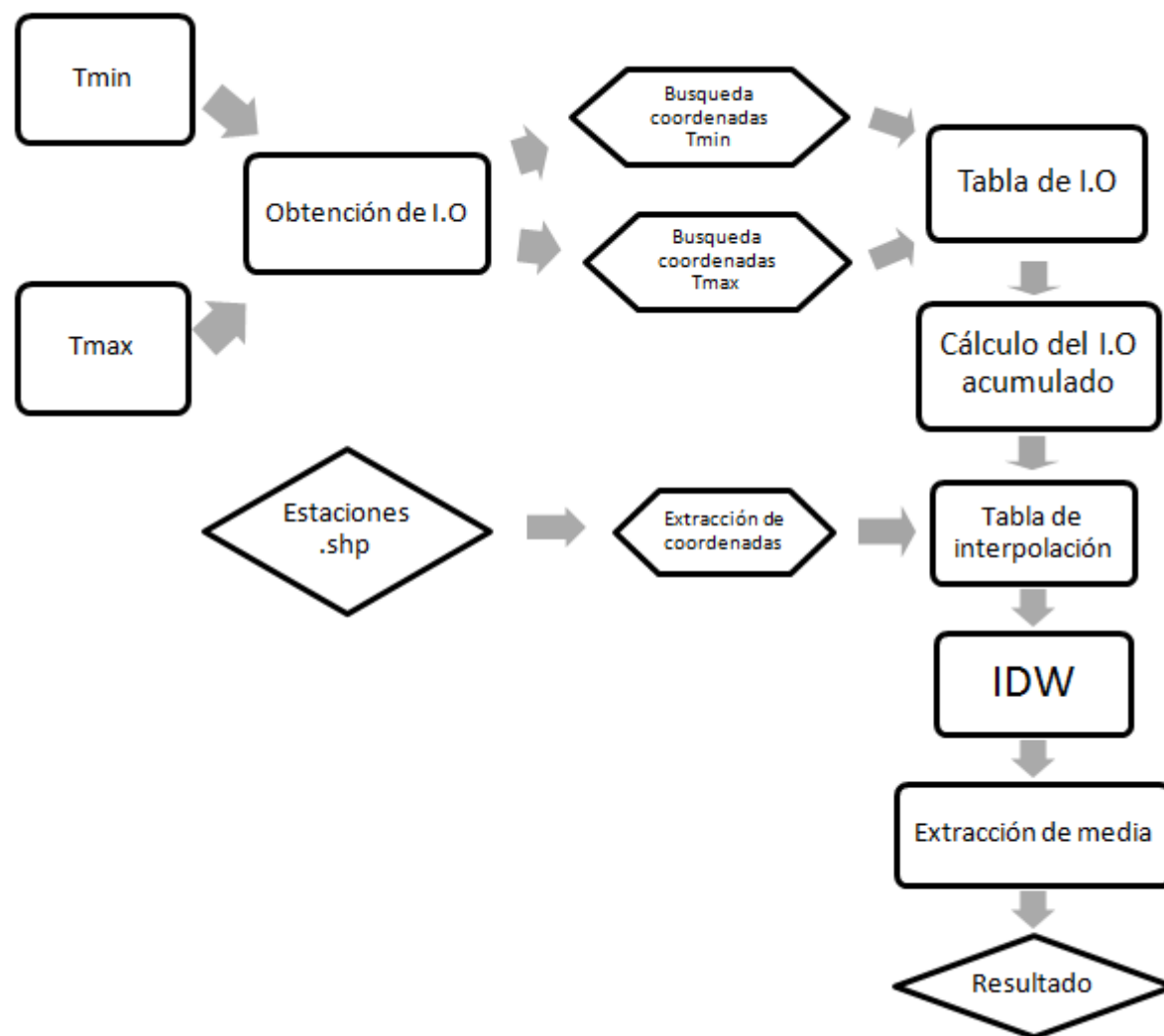
#Se crea el shp que va a contener los indices de oidio
writePolyShape(parcelas2, "C:/ms4w/apps/pmapper/pmapper_demodata/parcelas_oidio.shp")

```

ANEXO 4: DIAGRAMA DE FLUJO DEL MODELO DE MILDUI



ANEXO 5: DIAGRAMA DE FLUJO DEL MODELO DE OIDIO



ANEXO 6: SCRIPT GRÁFICO DE CRECIMIENTO

El código que se añade a continuación, se ejecuta tras el cálculo del crecimiento acumulado, y es el mismo para cada enfermedad y para cada estación. En este caso, se muestra el código de la estación de Hueta para la enfermedad de Mildiu y Oidio

```
#Se establecen los días
dias<-1:31

#Se crea un objeto con los días y el porcentaje acumulado
crecimiento<-cbind(dias,acumulado)

#Se organiza la tabla, poniendoles nombres a las columnas
colnames(crecimiento)=c("Dias","crecimiento")

#Se crea la tabla
write.table(crecimiento,"tabla_crecimiento.csv",row.names=FALSE,sep=";",dec=".")

#Se dibuja el grafico en R
plot(crecimiento, xlab="Días", ylab="Crecimiento (%)", main="Crecimiento de Mildiu, Hueta")
```

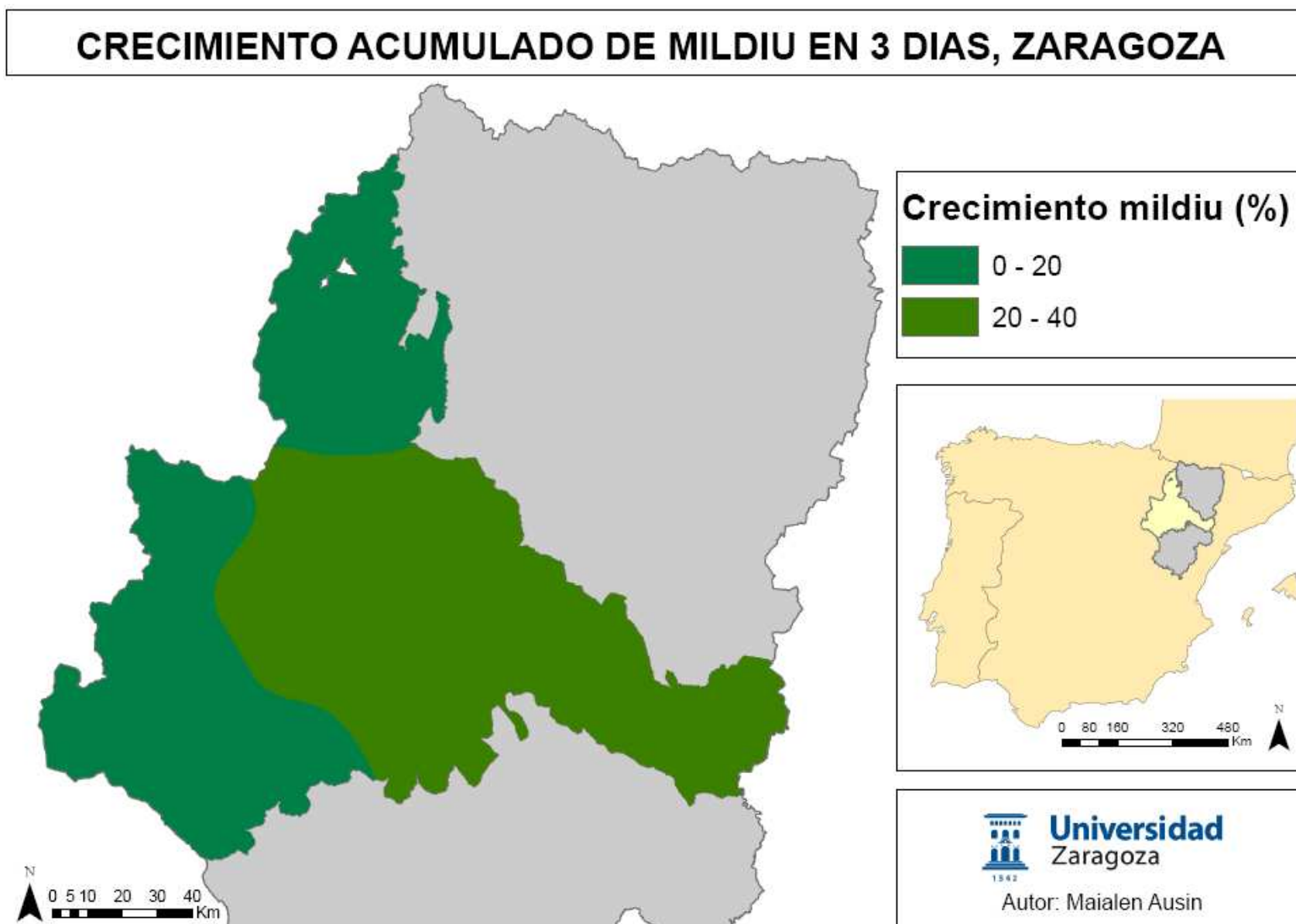
```
#Se establecen los días
dias<-1:31

#Se crea un objeto con los días y el porcentaje acumulado
crecimiento<-cbind(dias,acumulado)

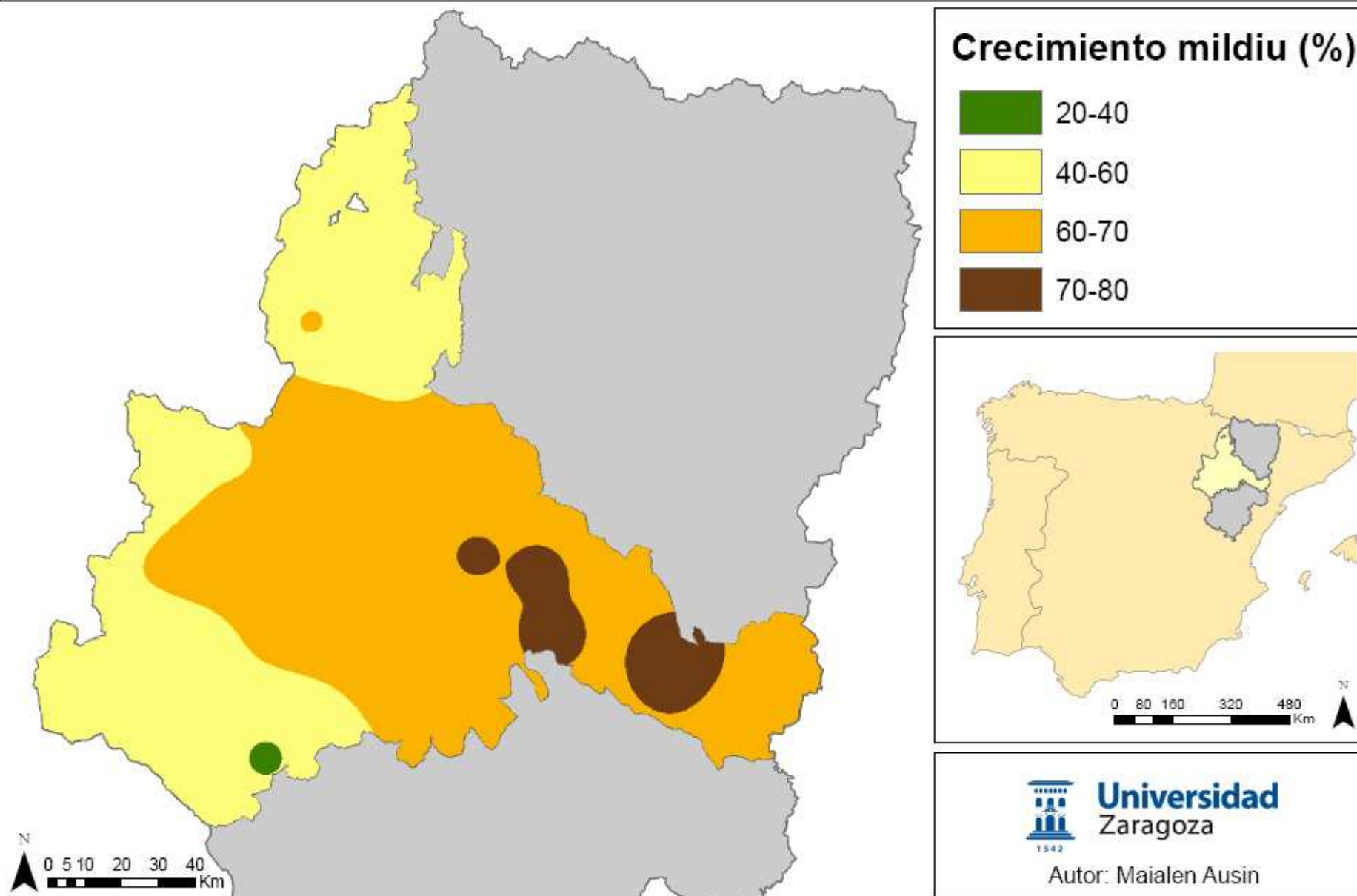
#Se organiza la tabla, poniendoles nombres a las columnas
colnames(crecimiento)=c("Dias","crecimiento")

#Se crea la tabla
write.table(crecimiento,"tabla_crecimiento.csv",row.names=FALSE,sep=";",dec=".")

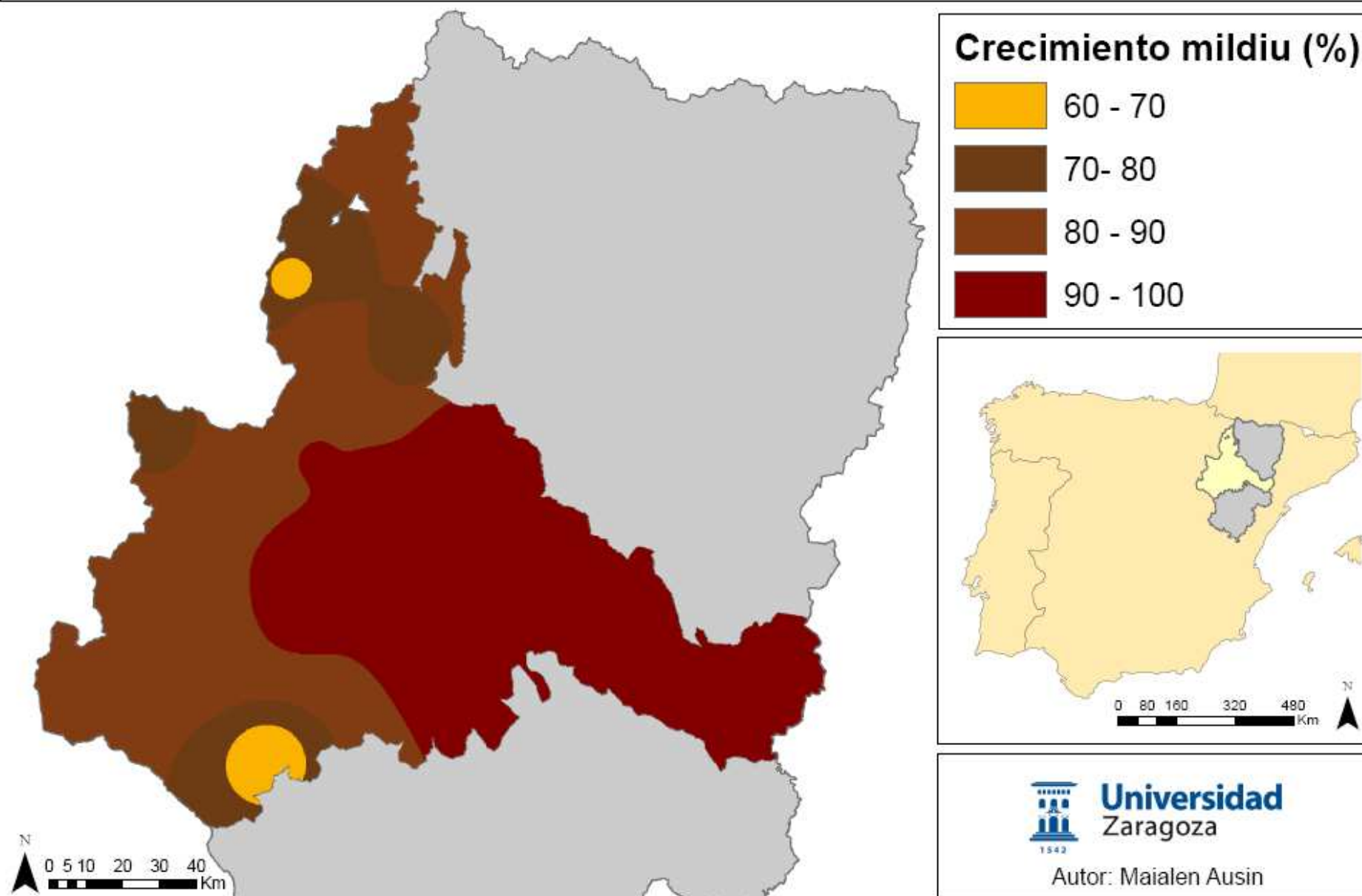
#Se dibuja el grafico en R
plot(crecimiento, xlab="Días", ylab="Crecimiento (°/1)", main="Crecimiento de Oidio, Hueta")
```



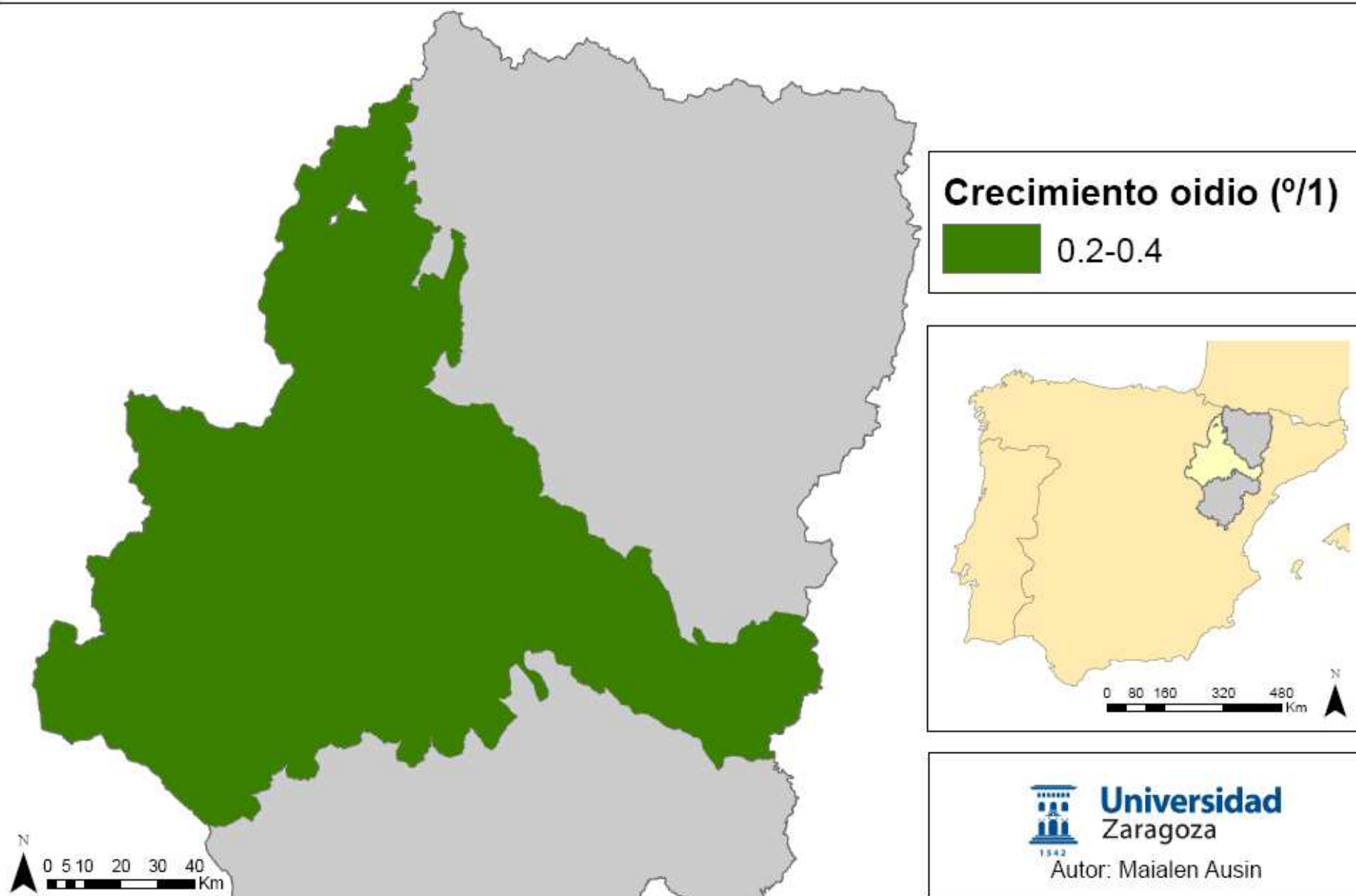
CRECIMIENTO ACUMULADO DE MILDIU EN 7 DIAS, ZARAGOZA



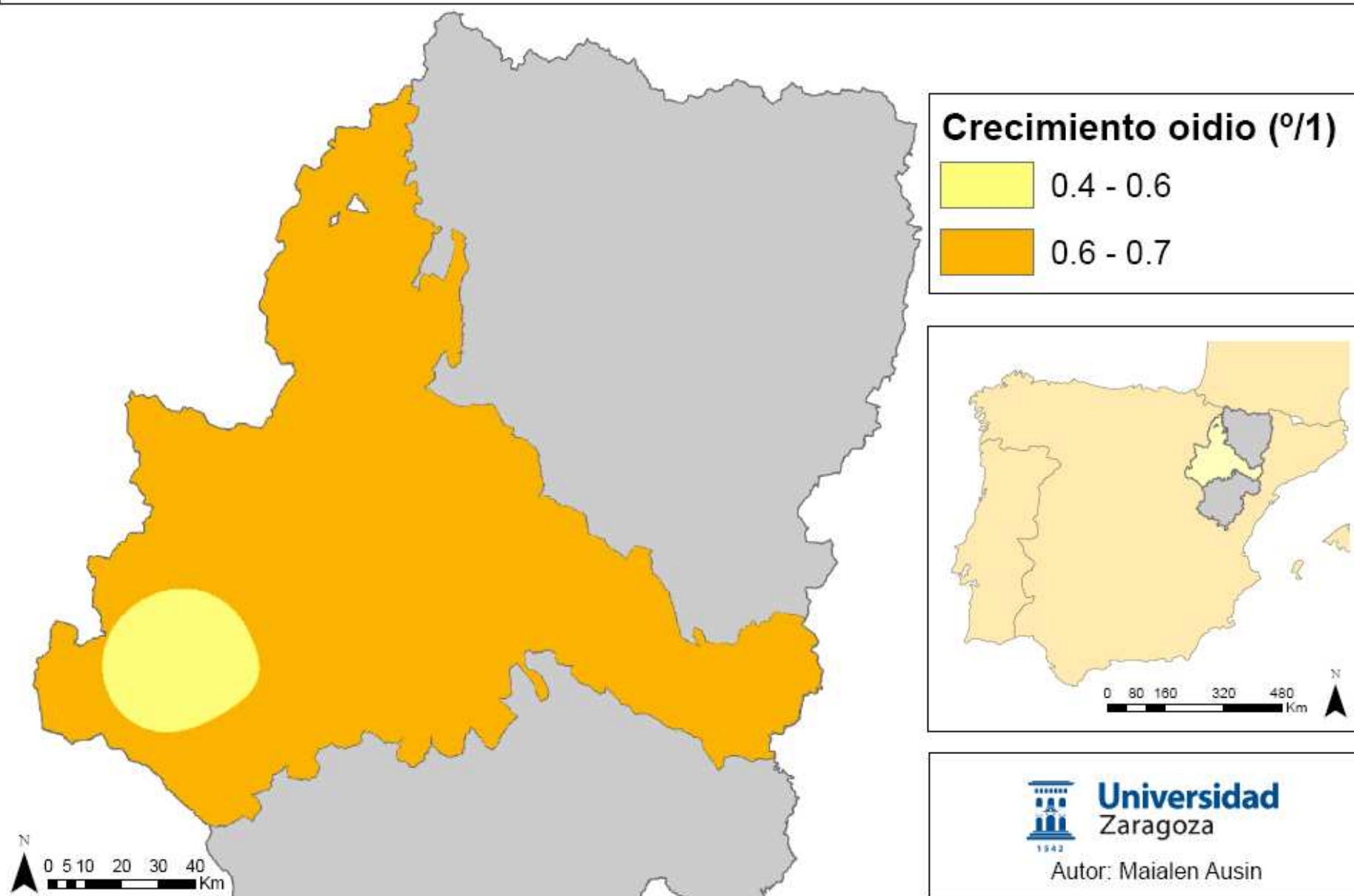
CRECIMIENTO ACUMULADO DE MILDIU EN 9 DIAS, ZARAGOZA



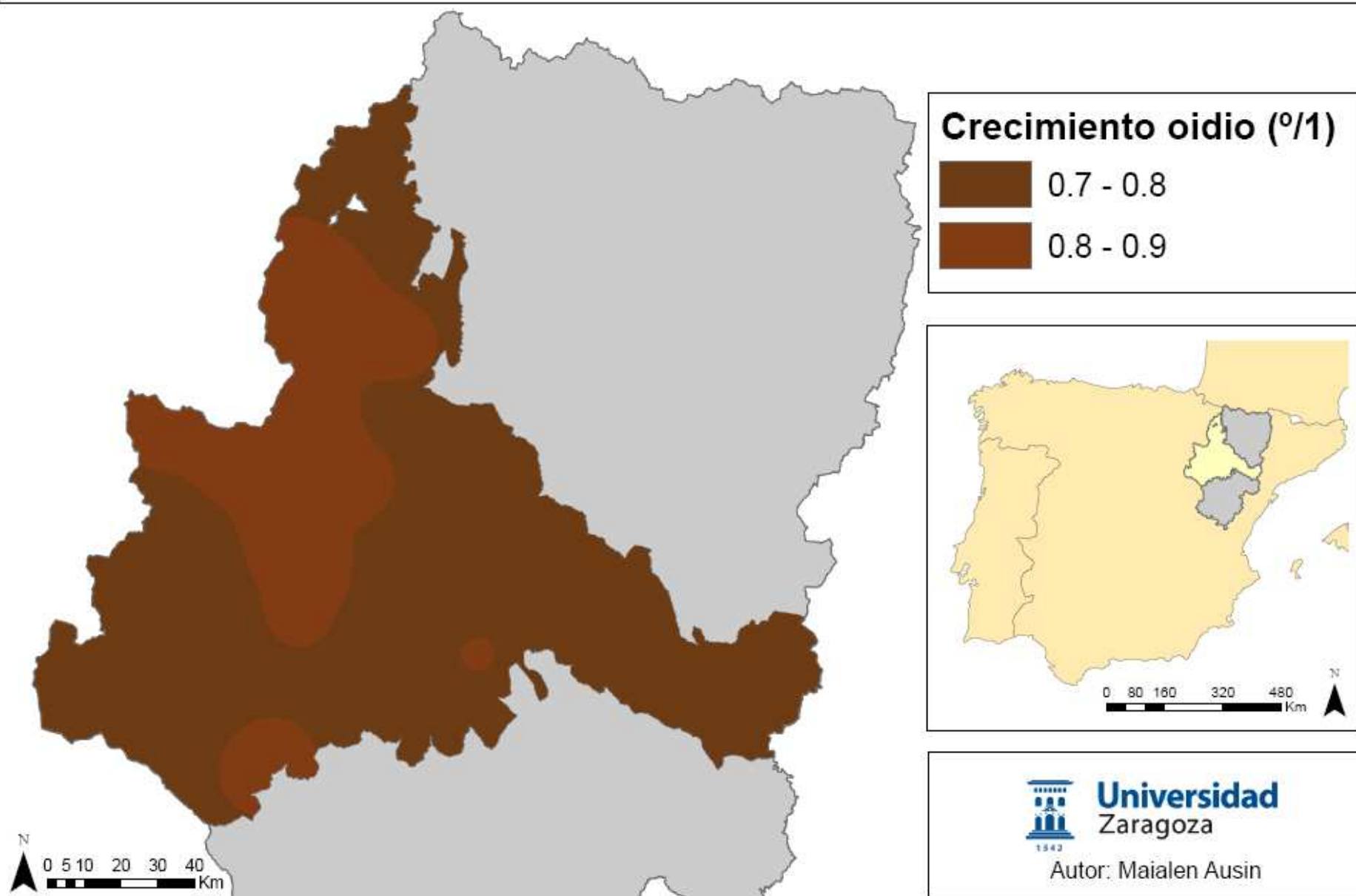
INDICE DE OIDIO ACUMULADO EN 3 DIAS, ZARAGOZA

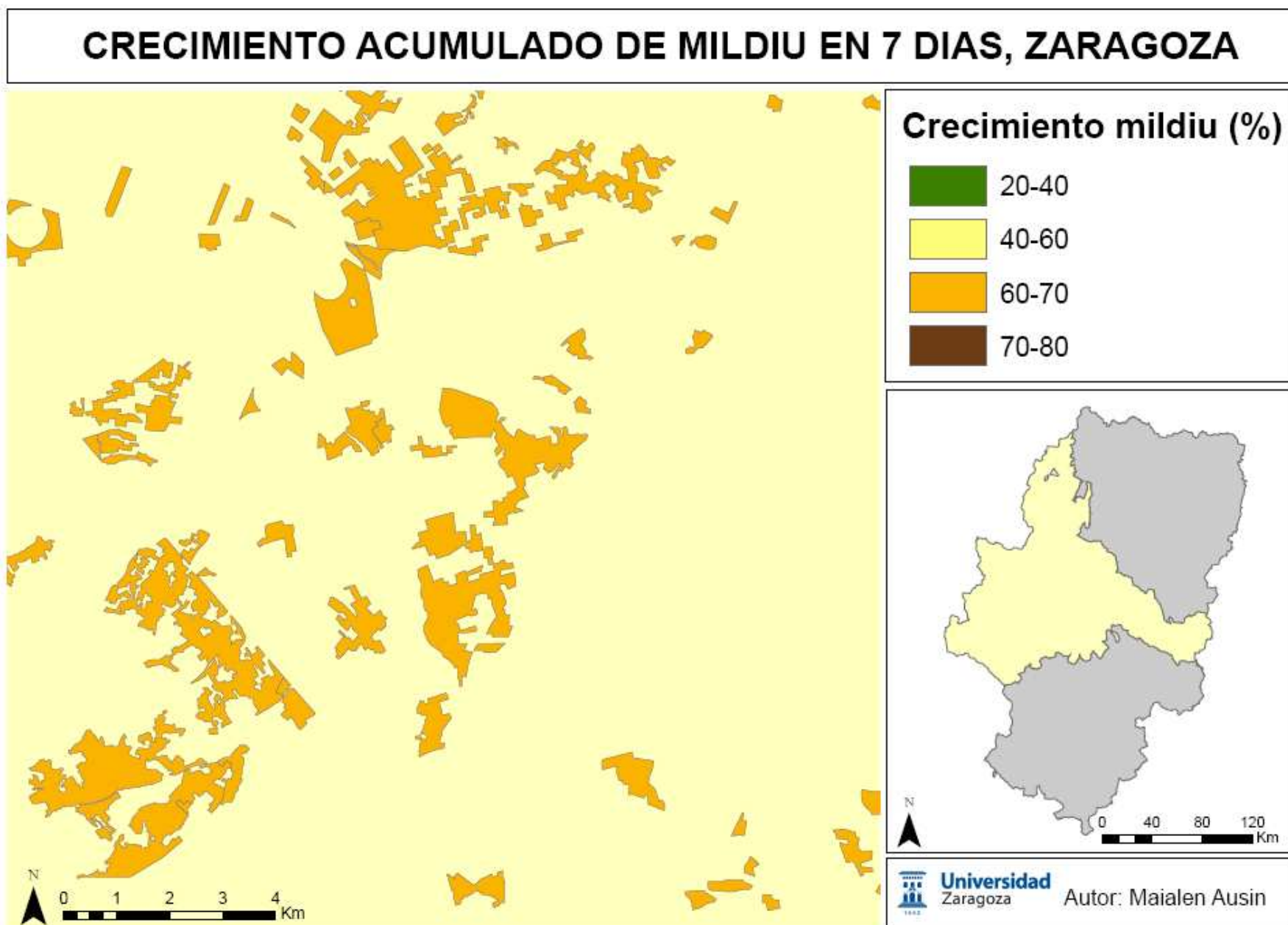


INDICE DE OIDIO ACUMULADO EN 7 DIAS, ZARAGOZA



INDICE DE OIDIO ACUMULADO EN 9 DIAS, ZARAGOZA





Parcelas de mildiu

Índice

[Identificador del fichero](#)
[Idioma](#)
[Contacto \(1, 2\)](#)
[Fecha de Creación](#)
[Norma de Metadatos](#)
[Versión de la Norma de Metadatos](#)
[Información sobre la Representación Espacial](#)
[Información de identificación](#)
[Información del Contenido](#)
[Información de Distribución](#)
[Mantenimiento de los metadatos](#)

Identificador del fichero

Parcelas de mildiu

[Volver al índice](#)

Idioma

gmd:LanguageCode: Español

[Volver al índice](#)

Contacto

Nombre individual

Maialen Ausin

Nombre del cargo

Técnico en SIG y teledetección

Información de contacto

Dirección

Ciudad

Donostia

Código postal

20012

Dirección electrónica

maialen.ausin@gmail.com

Horario de atención

9.00-15.00

Instrucciones para contacto

Contactar preferiblemente vía e-mail

Rol

Distribuidor

[Volver al índice](#)

Fecha de Creación

2014-11-18

[Volver al índice](#)

Norma de Metadatos

ISO 19115

[Volver al índice](#)

Versión de la Norma de Metadatos

ISO19115:2003/Cor 1 2006

[Volver al índice](#)

Información sobre la Representación Espacial

Objetos geométricos

Tipo de Objeto Geométrico

Superficie

Recuento de Objetos Geométricos

365

[Volver al índice](#)

Información de identificación

Mención

Título

Parcelas de mildiu

Fecha

Fecha

2014-11-18

Tipo de Fecha

creation

Resumen

Resultado del modelo predictivo de la evolución de mildiu en zaragoza: Parcelas que contienen el porcentaje de crecimiento de mildiu. La modelización de esta enfermedad tiene como objetivo el desarrollo final de un sistema de alerta agrícola.El sistema necesitaría una entrada de datos diarios y la ejecución automática del modelo (a través de batch-scripts) en busca del crecimiento de la enfermedad. Con este crecimiento se crearían mapas de riesgo que se irían actualizando día a día

Tipo de Representación espacial

Vector

Idioma

gmd:LanguageCode: Español

Extensión

Elemento geográfico

Límite de longitud Oeste

590461.9711999996

Límite de longitud Este

776584.4179999996

Límite de latitud Sur

4547232.3347

Límite de latitud Norte

4681684.5765

[Volver al índice](#)

Información del Contenido

Incluido en el conjunto de datos

false

Tipos de fenómeno

parcelas_mildiu

Mención del catálogo de fenómenos

Título

parcelas_mildiu

Fecha

Fecha

2014-12-01

Tipo de Fecha

publication

[Volver al índice](#)

Información de Distribución

Formato de distribución

Nombre

SHP - ArcView ShapeFile

Versión

Opciones de transferencia

Tamaño de transferencia (Mb)

0.5327758

Fuentes en línea

Enlace

parcelas_mildiu.shp

Opciones de transferencia

Tamaño de transferencia (Mb)

0.00288

Fuentes en línea

Enlace

parcelas_mildiu.shx

Opciones de transferencia

Tamaño de transferencia (Mb)

0.106779

Fuentes en línea

Enlace

parcelas_mildiu.dbf

[Volver al índice](#)

Mantenimiento de los metadatos

Frecuencia de mantenimiento y actualización

Según necesidad

Contacto

Nombre individual

Maialen Ausin

Rol

Creador

[Volver al índice](#)

Parcelas Oidio

Índice

[Identificador del fichero](#)

[Idioma](#)

[Contacto \(1, 2\)](#)

[Fecha de Creación](#)

[Norma de Metadatos](#)

[Versión de la Norma de Metadatos](#)

[Información sobre la Representación Espacial](#)

[Información de identificación](#)

[Información del Contenido](#)

[Información de Distribución](#)

Identificador del fichero

Parcelas de Oidio

[Volver al índice](#)

Idioma

gmd:LanguageCode: Español

[Volver al índice](#)

Contacto

Nombre individual

Maialen Ausin

Nombre del cargo

Técnico en SIG y Teledetección

Información de contacto

Dirección

Ciudad

Donostia
Código postal
20012
Dirección electrónica
maialen.ausin@gmail.com

Rol

Creador

[Volver al índice](#)

Contacto

Nombre individual

Maialen Ausin

Nombre del cargo

Técnico en SIG y Teledetección

Información de contacto

Dirección

Ciudad

Donostia

Código postal

20012

Dirección electrónica

maialen.ausin@gmail.com

Horario de atención

9.00-15.00

Instrucciones para contacto

Contactar preferiblemente por e-mail

Rol

Conservador

[Volver al índice](#)

Fecha de Creación

2014-11-18

[Volver al índice](#)

Norma de Metadatos

ISO 19115

[Volver al índice](#)

Versión de la Norma de Metadatos

ISO19115:2003/Cor 1 2006

[Volver al índice](#)

Información sobre la Representación Espacial

Objetos geométricos

Tipo de Objeto Geométrico

Superficie

Recuento de Objetos Geométricos

365

[Volver al índice](#)

Información de identificación

Mención

Título

Parcelas Oidio

Fecha

Fecha

2014-11-18

Tipo de Fecha

creation

Resumen

Resultado del modelo predictivo de la evolución del Oidio en Zaragoza: Páreas que contienen el porcentaje de crecimiento de Oidio La modelización de estas enfermedades tiene como objetivo el desarrollo final de un sistema de alerta agrícola. El sistema necesitaría una entrada de datos diarios y la ejecución automática del modelo (a través de batch-scripts) en busca del crecimiento de la enfermedad. Con este crecimiento se crearían mapas de riesgo que se irían actualizando día a día

Tipo de Representación espacial

Vector

Idioma

gmd:LanguageCode: spa

Categorías de Temas

Agricultura

Extensión

Elemento geográfico

Límite de longitud Oeste

590461.9711999996

Límite de longitud Este

776584.4179999996

Límite de latitud Sur

4547232.3347

Límite de latitud Norte

4681684.5765

[Volver al índice](#)

Información del Contenido

Incluido en el conjunto de datos

false

Tipos de fenómeno

parcelas_oidio

Mención del catálogo de fenómenos

Título

parcelas_oidio

Fecha

Fecha

2014-12-01

Tipo de Fecha

publication

[Volver al índice](#)

Información de Distribución

Formato de distribución

Nombre

SHP - ArcView ShapeFile

Versión

Opciones de transferencia

Tamaño de transferencia (Mb)

0.5327758

Fuentes en línea

Enlace

[parcelas_oidio.shp](#)

Opciones de transferencia

Tamaño de transferencia (Mb)

0.00288

Fuentes en línea

Enlace

[parcelas_oidio.shx](#)

Opciones de transferencia

Tamaño de transferencia (Mb)

0.1158905

Fuentes en línea

Enlace

[parcelas_oidio.dbf](#)

[Volver al índice](#)