

PAC 4

Exercicis

Exercici 1: Mapes de variabilitat. Generar un mapa de variabilitat de precipitació mensual acumulada durant el gener de 2024. Generar també un mapa de temperatures mínimes

Precipitació

Primer de tot cal baixar les dades del mes de gener de 2024, d'AEMET:

```
library(meteospain)
library(keyring)

api_options <- aemet_options(
  resolution = 'daily',
  start_date = as.Date('2024-01-01'),
  end_date = as.Date('2024-01-31'),
  api_key = key_get('aemet')
)

datos <- get_meteo_from('aemet', options = api_options)
```

Podem desar aquestes dades en un fitxer per tal de no fer consultes constantment.

El primer que podem fer és inspeccionar les dades que hem baixat, (quantes entrades, quines variables, quines unitats).

```
# La llibreria units ens mostra les unitats de les variables
library(units)

nrow(datos) # nombre de files
head(datos) # primeres dades
```

Ens hem baixat un total de 27516 entrades o files i les variables que tenim les podem veure en el següent resum de les primeres entrades:

Tot seguit canviem la variable *precipitation* de unit a numeric per tal de fer operacions amb ella. També eliminem els valors NAs d'aquesta variable.

```
if('units' %in% class(datos$precipitation)) {
  datos$precipitation <- as.numeric(datos$precipitation)
}

datos_pluja <- datos[!is.na(datos$precipitation), ]
```

```
## timestamp service station_id station_name station_province altitude
## 1 2024-01-01 aemet 0009X ALFORJA TARRAGONA 406 [m]
## 2 2024-01-01 aemet 0016A REUS AEROPUERTO TARRAGONA 71 [m]
## 3 2024-01-01 aemet 0016B REUS (CENTRE LECTURA) TARRAGONA 118 [m]
## 4 2024-01-01 aemet 0034X VALLS TARRAGONA 233 [m]
## 5 2024-01-01 aemet 0042Y TARRAGONA TARRAGONA 55 [m]
## 6 2024-01-01 aemet 0061X PONTONS BARCELONA 632 [m]
## mean_temperature min_temperature max_temperature precipitation
## 1 8.8 [°C] 4.0 [°C] 13.6 [°C] 0 [L/m^2]
## 2 8.9 [°C] 2.5 [°C] 15.3 [°C] 0 [L/m^2]
## 3 12.7 [°C] 8.9 [°C] 16.5 [°C] 0 [L/m^2]
## 4 8.6 [°C] 2.0 [°C] 15.3 [°C] 0 [L/m^2]
## 5 10.6 [°C] 5.7 [°C] 15.5 [°C] 0 [L/m^2]
## 6 7.9 [°C] 3.4 [°C] 12.4 [°C] 0 [L/m^2]
## mean_wind_speed insolation geometry
## 1 4.2 [m/s] NA [h] 0.9633333, 41.2138889
## 2 2.2 [m/s] 6.3 [h] 1.163611, 41.145000
## 3 NA [m/s] 5.9 [h] 1.108889, 41.154167
## 4 NA [m/s] NA [h] 1.260833, 41.293056
## 5 NA [m/s] NA [h] 1.249167, 41.123889
## 6 2.5 [m/s] NA [h] 1.519167, 41.416944
```

Volem la pluja acumulada durant el mes de gener per cada estació meteorològica, per obtenir-ho fem servir la llibreria `dplyr` i agrupem pels camps que volem mantenir i apliquem la funció `suma`. En aquest cas considerem l'identificador de l'estació, el nom d'aquesta (potser no caldria) i la geometria que ens dona la ubicació de cada estació.

```
library(dplyr)

pluja_acumulada <- datos_pluja |>
  group_by(station_id, station_name, geometry) |>
  summarise(pluja = sum(precipitation))
```

Ara volem fer una representació gràfica d'aquesta pluja acumulada a les diferents estacions, abans hem de convertir el resultat obtingut a un objecte `sf`, per tal d'obtenir les geometries com a objectes `sfc_POINT`. Això ho fem amb la llibreria `sf`.

```
library(sf)

pluja_acumulada <- st_as_sf(pluja_acumulada)
```

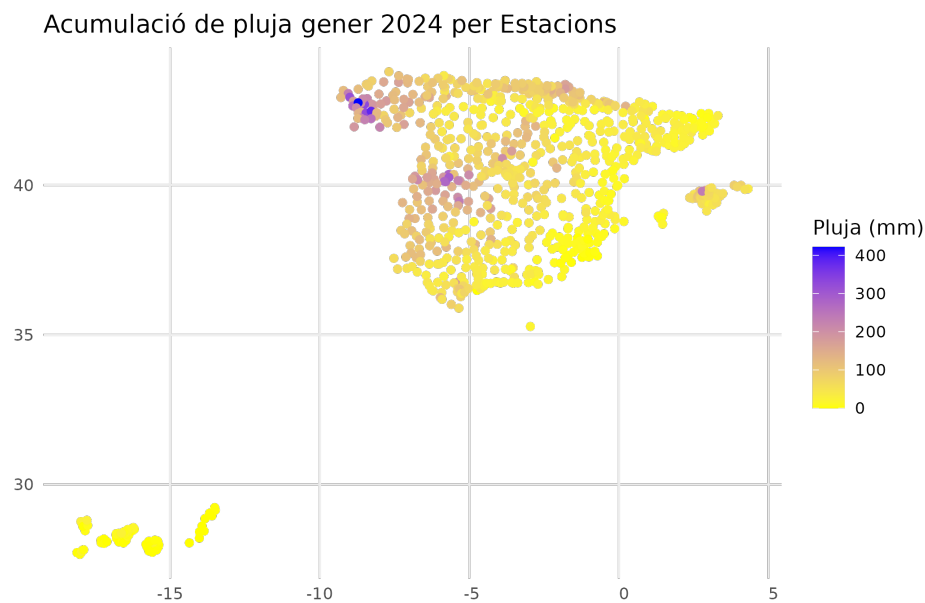
Ja podem plotejar amb la llibreria `ggplot2`. Fem servir un gradient de color que va de groc (menys precipitació) a blau (més precipitació).

```
library(ggplot2)

ggplot(pluja_acumulada) +
  geom_sf(aes(color = pluja), size = 2) +
```

```
scale_color_gradient(low = 'yellow', high = 'blue') +
labs(
  title = 'Acumulació de pluja gener 2024 per Estacion',
  color = 'Pluja (mm)'
) +
theme_minimal()
```

El resultat és el següent:



Podem veure que les zones on ha plogut més és la part nord-est de la Península.

Temperatura mínima

D'una manera semblant generem un mapa per la temperatura mínima enregistrada per les diferents estacions. Canviem la variable *min_temperature* de unit a numeric per tal de fer operacions amb ella i també eliminem els possibles valors NAs d'aquesta variable.

```
if('units' %in% class(datos$min_temperature)) {
  datos$min_temperature <- as.numeric(datos$min_temperature)
}

datos_min_temp <- datos[!is.na(datos$min_temperature), ]
```

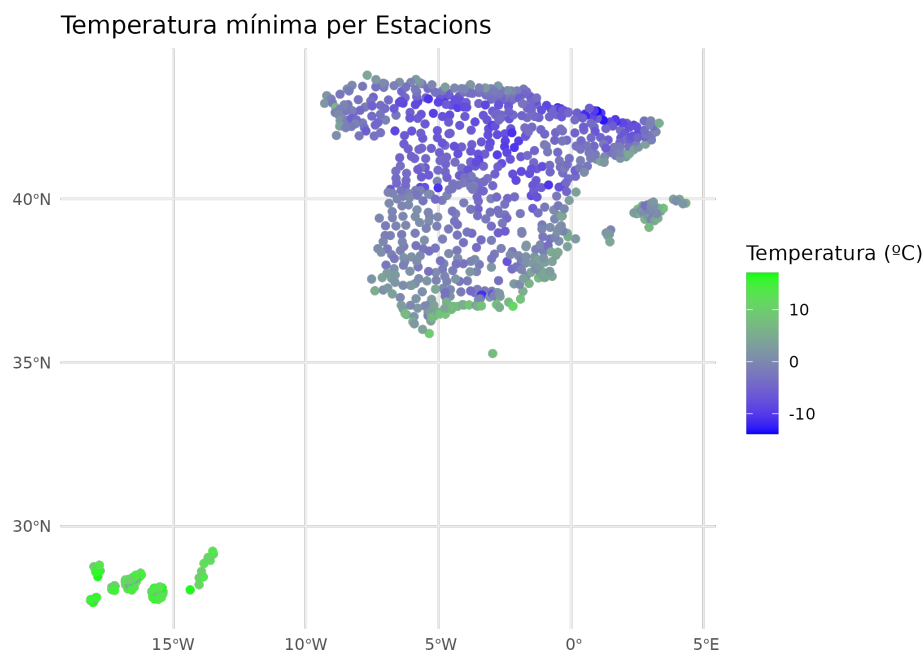
Tot seguit calculem la temperatura mínima enregistrada, fent una agrupació com hem fet abans per la precipitació, però ara apliquem la funció *min*. També convertim a un objecte *sf*.

```
temp_min <- datos_min_temp |>
  group_by(station_id, station_name, geometry) |>
  summarise(temp = min(min_temperature))

temp_min <- st_as_sf(temp_min)
```

Fem una gràfica com l'anterior, amb un gradient de blau (temperatura més baixa) a verd (temperatura més alta):

```
ggplot(temp_min) +
  geom_sf(aes(color = temp), size = 2) +
  scale_color_gradient(low = 'blue', high = 'green') +
  labs(
    title = 'Temperatura mínima per Estacions',
    color = 'Temperatura (°C)',
  ) +
  theme_minimal()
```



Podem veure que les temperatures mínimes més altes, a part de Les Illes Canàries, es troben al llarg de la costa mediterrània, sobretot cap al sud, i les més baixes les trobem al Pirineu i interior de la Península.

Exercici 2: Mapa temàtic de precipitació per comunitat.

Fem servir les dades de l'exercici anterior, recordem que tenim `pluja_acumulada` que conté els camps ID de l'estació, nom de l'estació, la geometria i la pluja acumulada.

Primer de tot llegim el *shapefile* de les províncies:

```
provincias <- st_read('./recintos_provinciales_inspire_peninbal_etr89/  
recintos_provinciales_inspire_peninbal_etr89.shp')
```

Tot seguit amb la llibreria *sp* generem un objecte *SpatialPointsDataFrame* per facilitar l'anàlisi. Per obtenir-lo seguim els següents passos:

1. extraïem les coordenades de les estacions,
2. creem un objecte *SpatialPoints* a partir de les coordenades extreïdes, amb el datum *WGS84*
3. convertim tot seguit les dades de l'exercici anterior a *dataframe* per poder facilitar la manipulació de les seves dades,
4. eliminem la columna *geometry* que ja hem salvat,
5. generem finalment un objecte *SpatialPointsDataFrame*

```
library(sp)  
  
coords <- st_coordinates(pluja_acumulada$geometry) # 1  
SP_puntos <- SpatialPoints(coords,  
                           proj4string = CRS('+proj=longlat +datum=WGS84')) # 2  
pluja_acumulada_SPDF <- as.data.frame(pluja_acumulada) # 3  
pluja_acumulada_SPDF$geometry <- NULL # 4  
SP_puntos_df <- SpatialPointsDataFrame(SP_puntos, pluja_acumulada_SPDF) # 5
```

Ara hem d'ajustar la projecció de les dades provincials perquè coincideixin amb la de les estacions, la del *spatialpointsdataframe*. Abans canviem l'objecte *provincias* de *sf* a *sp*.

```
proj4string(SP_puntos_df)  
provincias_sp <- as(provincias, 'Spatial')  
proj4string(provincias_sp)  
  
provincias_geo <- spTransform(provincias_sp,  
                             CRSobj = CRS(proj4string(SP_puntos_df)))
```

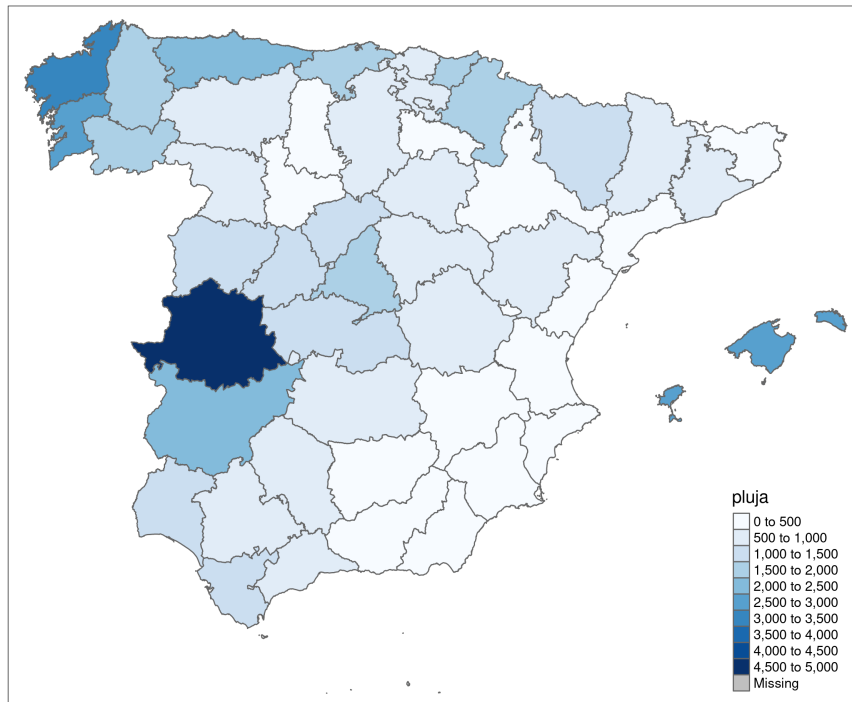
A l'exercici anterior ja hem eliminat valors *NA*s, per tant seleccionem les dades que ens interessin, en el nostre cas, l'acumulació de pluja de cada estació. I tot seguit fem una agregació per calcular l'acumulació de pluja per cada província, amb la funció *suma*.

```
SP_puntos_df2 <- SP_puntos_df[, c(3)]  
provincias_agg <- aggregate(SP_puntos_df2, by = provincias_geo, FUN = sum)
```

Finalment visualitzem el mapa temàtic amb el paquet *tmap*, amb un gradient de blaus que indica més o menys pluja acumulada, (blau fosc o blau clar).

```
library(tmap)

qtm(provincias_agg, fill = 'pluja', fill.n=8, fill.palette = 'Blues')
```



Podem veure que el resultat és correspon al mapa que hem fet a l'exercici 1 per l'acumulació de pluja per les estacions.

Exercici 3: Anàlisi espacial de les estacions meteorològiques de Catalunya.

Volem calcular i visualitzar les mesures centrogràfiques per Catalunya, en particular, el centre mig, el centre mitjà, la desviació estàndard i l'el·lipse de desviació estàndard.

Primer de tot seleccionem les províncies de Catalunya de l'objecte que hem creat a l'exercici anterior, `provincias_geo`.

```
cat <- subset(provincias_geo, (provincias_geo@data$NAMEUNIT %in%
  c('Barcelona', 'Tarragona', 'Lleida', 'Girona')))
```

Calculem la intersecció de l'objecte `SP_puntos_df2` que conté les ubicacions de les estacions, amb l'objecte de polígons `cat`, que conté les províncies catalanes.

Per fer-ho fem servir el paquet `rgeos`, que hem instal·lat, en el cas de Linux amb la instrucció `sudo apt install r-cran-rgeos`.

```
library(rgeos)
estaciones.cat <- gIntersection(SP_puntos_df2, cat)
```

Tot seguit calculem les mesures abans esmentades fent servir el paquet *aspace*.

```
library(aspace)

centro_medio_g <- calc_mnc(id = 1, weighted = FALSE, weights = NULL,
  points = estaciones.cat@coords)

centro_mediano_g <- calc_mdc(id = 1, points = estaciones.cat@coords)

distancia_tipica_g <- calc_sdd(id = 1, centre.xy = NULL,
  calccentre = TRUE, weighted = FALSE, weights = NULL,
  points = estaciones.cat@coords, verbose = FALSE)

elipse_desviacion_estandar_g <- calc_sde(id = 1, centre.xy = NULL,
  calccentre = TRUE, weighted = FALSE, weights = NULL,
  points = estaciones.cat@coords, verbose = FALSE)
```

Preparem les dades per poder-les visualitzar amb *ggplot2*. Considerem les coordenades de les estacions meteorològiques i creem un *dataframe*, això ho fem per cada mesura.

```
centro_medio_data <- centro_medio_g$LOCATIONS
centro_medio_df <- data.frame(lon = centro_medio_data[[2]],
  lat = centro_medio_data[[3]])

centro_mediano_data <- centro_mediano_g$LOCATIONS
centro_mediano_df <- data.frame(lon = centro_mediano_data[[2]],
  lat = centro_mediano_data[[3]])

elipse_data <- elipse_desviacion_estandar_g$LOCATIONS
elipse_df <- data.frame(lon = elipse_data[[2]],
  lat = elipse_data[[3]])

distancia_tipica_data <- distancia_tipica_g$LOCATIONS
distancia_tipica_df <- data.frame(lon = distancia_tipica_data[[2]],
  lat = distancia_tipica_data[[3]])
```

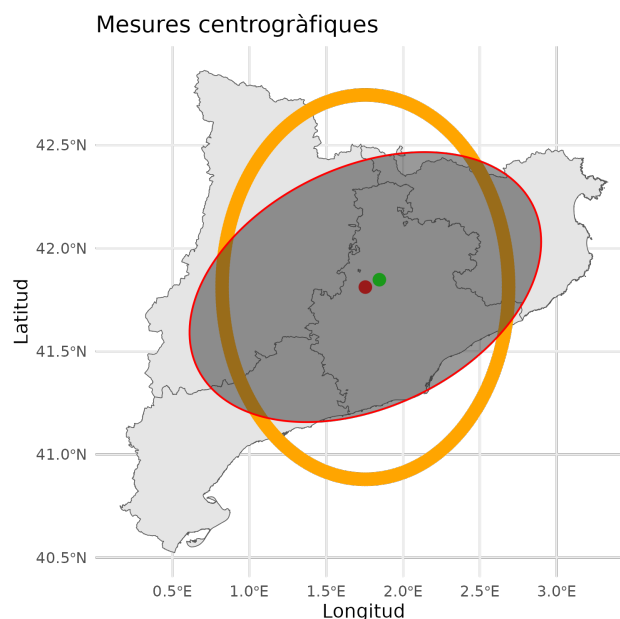
Finalment, llegim l'arxiu *shapefile* de les províncies i seleccionem només les catalanes. Aquestes són la base del mapa per situar les mesures calculades.

```
shapefile <- st_read('./recintos_provinciales_inspire_peninbal_etr89/
  recintos_provinciales_inspire_peninbal_etr89.shp')
```

```
# https://www.r-bloggers.com/2019/04/zooming-in-on-maps-with-sf-and-ggplot2/
shp_cat <- shapefile[shapefile$NAMEUNIT %in%
  c('Barcelona', 'Tarragona', 'Lleida', 'Girona'), ]
```

Només ens queda la visualització del mapa:

```
ggplot() +
  geom_sf(data = shp_cat) +
  geom_point(data = centro_mediano_df, aes(x = lon, y = lat),
    color = 'green', size = 3) +
  geom_point(data = centro_medio_df, aes(x = lon, y = lat),
    color = 'red', size = 3) +
  geom_point(data = distancia_tipica_df, aes(x = lon, y = lat),
    color = 'orange', size = 3) +
  geom_polygon(data = ellipse_df, aes(x = lon, y = lat),
    color = 'red', alpha = 0.5, color = 'red') +
  theme_minimal() +
  labs(title = 'Mesures centrogràfiques', x = 'Longitud', y = 'Latitud')
```



Exercici 4: Anàlisi de densitat de punts de Catalunya.

Volem fer un estudi de la densitat de punts a Catalunya, incloent l'anàlisi de quadrants i la densitat de Kernel, per tal de descriure la variació espacial de la distribució i intensitat de punts.

Comencem amb l'enfocament *quadratcount*, i fem el càlcul de la mida de les cel·les de la malla base. Generem un sol polígon de Catalunya, però a diferència del guió de la PAC4, fem servir la

funció `st_union` de la llibreria `sf`.

Els passos que seguim venen descrits en el següent codi:

```
# Verifica duplicats
row.names(estaciones.cat) <- make.unique(row.names(estaciones.cat))

# Transformem a UTM 31 i WGS84
estacionesUTM <- spTransform(estaciones.cat,
  CRS = '+proj=utm +zone=31 +datum=WGS84 +units=m +no_defs')

# unim regions (fem servir sf i passem a sp per la posterior transformació)
region <- as_Spatial(st_union(shp_cat))

# transformem a UTM 31 i WGS84
regionUTM <- spTransform(region,
  CRS = '+proj=utm +zone=31 +datum=WGS84 +units=m +no_defs')

# Calculem àrea
areap <- raster::area(regionUTM)
areap

# Comptem estacions meteorològiques
n.estaciones <- nrow(estacionesUTM)
n.estaciones

# Calculem grandària de les celes
l.celda <- sqrt((2 * areap) / n.estaciones)
l.celda
```

Obtenim una àrea de Catalunya igual a $32\,107\,530\,548\text{ m}^2$, valor correcte si ho cerquem a la web, tenim 72 estacions i la grandària de les cel·les és de $29\,864.29\text{ m}^2$.

Seguidament determinem el nombre de cel·les i columnes que farem servir.

```
# Carreguem la llibreria 'raster'
library(raster)

# Convertim la regió a quadrícula
r <- raster(regionUTM)
res(r) <- l.celda
r
```

El resultat és:

Tenim una malla base de 9×9 cel·les quadrades. Per calcular *quadratcount* necessitem un objecte tipus *ppp* amb els punts mostrals. Projectem a coordenades UTM i obtenim també l'objecte *window*.

```
## class      : RasterLayer
## dimensions : 9, 9, 81  (nrow, ncol, ncell)
## resolution : 29864.29, 29864.29  (x, y)
## extent     : 260159.6, 528938.2, 4479197, 4747976  (xmin, xmax, ymin, ymax)
## crs        : +proj=utm +zone=31 +datum=WGS84 +units=m +no_defs
```

```
# Carreguem llibreries necessàries
library(sp)
library(maptools)

# Convertim regionUTM a un objecte de la classe owin
regionUTM_owin <- as(regionUTM, 'owin')

library(spatstat)
aoi <- as.owin(regionUTM_owin)

# Creem objecte ppp que representa estacions meteorològiques
# x, y són les coordenades de les estacions
# window especifica la finestra d'àrea d'observació per l'objecte puntual
estaciones.ppp <- ppp(x=estacionesUTM@coords[,1],
                     y=estacionesUTM@coords[,2],
                     window = aoi)
```

Finalment obtenim el *quadratcount* i podem generar els mapes d'estacions per cel·la i el mapa de densitat.

```
# Comptem punts
estaciones.q1 <- quadratcount(estaciones.ppp,
                             nx=r@ncols, ny=r@nrows, as.test(aoi))

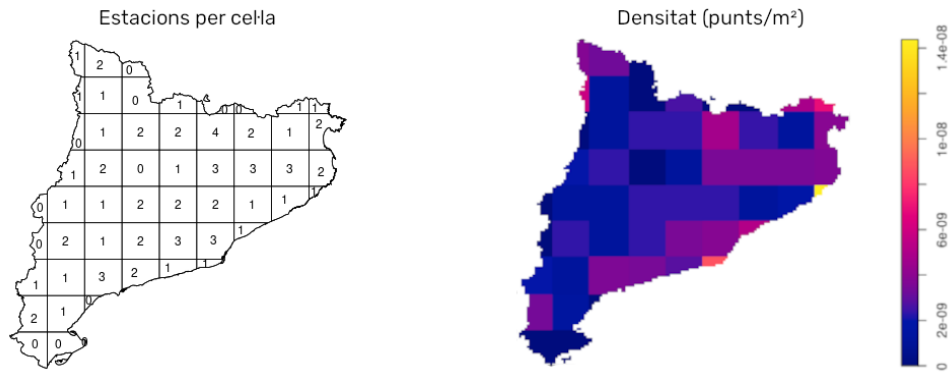
# Dibuixem mapa d'estacions per cel·la
plot(estaciones.q1, main = "Estacions per cel·la")

# Dibuixem mapa de densitat
plot(intensity(estaciones.q1, image = TRUE), main = "Densitat (punts/m2)")
```

Podem veure aquests mapes en les següents figures.

Seguim amb l'anàlisi de la densitat de punts i tot seguit volem veure la relació entre la mitjana i la variància amb la intenció de determinar si tenim agrupació o clúster. Aquest anàlisi es basa en les freqüències del nombre de punts per cel·la i permet confirmar visualment les característiques de distribució dels punts a través del mapa d'intensitat.

Comencem amb la taula de freqüències del nombre de punts per cel·la.



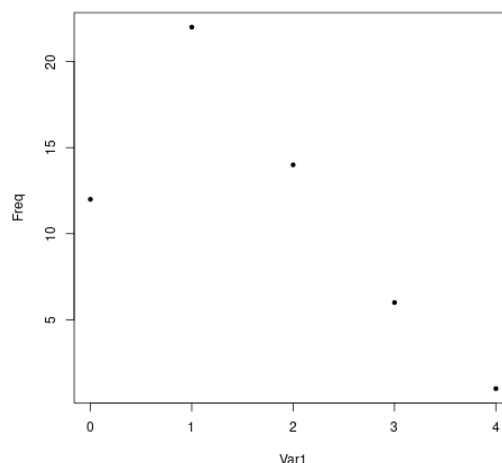
```
# Convertim estaciones.q1 a dataframe per facilitar la manipulació
estaciones.q1.df <- as.data.frame(estaciones.q1)

# Creem taula de freqüència de num punts per quadrat
# Comptem les freqüències de la columna Freq
estaciones.count.table <- data.frame(table(estaciones.q1.df$Freq, exclude = NULL))
```

Convertim la la columna *Freq* a numèric i generem el gràfic.

```
# Freq a numèric
estaciones.count.table[, 1] <- as.numeric(levels(estaciones.count.table[, 1]))

# Gràfica
plot(estaciones.count.table, pch = 20)
```



Podem comprovar que aquestes freqüències s'ajusten amb el mapa d'estacions per cel·la anterior. Calculem ara la mitjana del nombre de punts per quadrant i la variància per cada quadrant.

```
# Calcul de quadrants
cuadrantes <- sum(estaciones.count.table[, 2])
```

```
# Mitjana de punts per quadrat dividit pel nombre d'estacions
media <- n.estaciones / cuadrantes

estaciones.count.table$difsqr <- (
  (estaciones.count.table$Var1 - media)^2) * estaciones.count.table$Freq

# Càlcul de la variancia
s2 <- sum(estaciones.count.table$difsqr) / (sum(estaciones.count.table$Freq) - 1)
```

Finalment calculem el valor per a VMR (*Variance / Mean Ratio*).

```
VMR <- s2 / media
VMR
```

Obtenim un valor de 0.7602881, la qual cosa ens indica que tenim poca dispersió, podem dir que estem entre un patró regular ($VMR = 0$) i un patró aleatori ($VMR = 1$). El mapa de densitat anterior ja ens mostra que no tenim agrupacions.

Exercici 5: Tècniques d'interpolació espacial.

Primera part: precipitacions a tot l'estat

El que volem fer ara és, considerant les precipitacions de tot l'estat que hem fet servir a l'exercici 1, apliquem la distància inversa ponderada i el krigatge com a mètodes d'interpolació espacial.

Recordem que tenim un *SpatialPointsDataFrame* amb el nom de `SP_puntos_df`, net de valors NAs, amb una columna que es diu `pluja` que conté la pluja acumulada.

Comencem amb el mètode de la distància inversa ponderada (*IDW*) per tal d'estimar els valors en una malla regular a partir de l'acumulació de pluja de les estacions. Ho fem amb diferents valors de potència inversa.

Com hem dit, ja tenim les dades netes, per tant executem les següents instruccions:

```
# Carreguem gstat
library(gstat)

# Treiem les Balears i creem raster amb resolució de 0.05
esp_pen <- subset(provincias_geo,
  !(provincias_geo@data$NAMEUNIT == 'Illes Balears'))
esp.raster <- raster(esp_pen, res = 0.05)

# Layout dels mapes a crear: 2 files x 3 columnes
par(mfrow=c(2, 3), mai=c(0.5, 0.5, 0.5, 0.5))

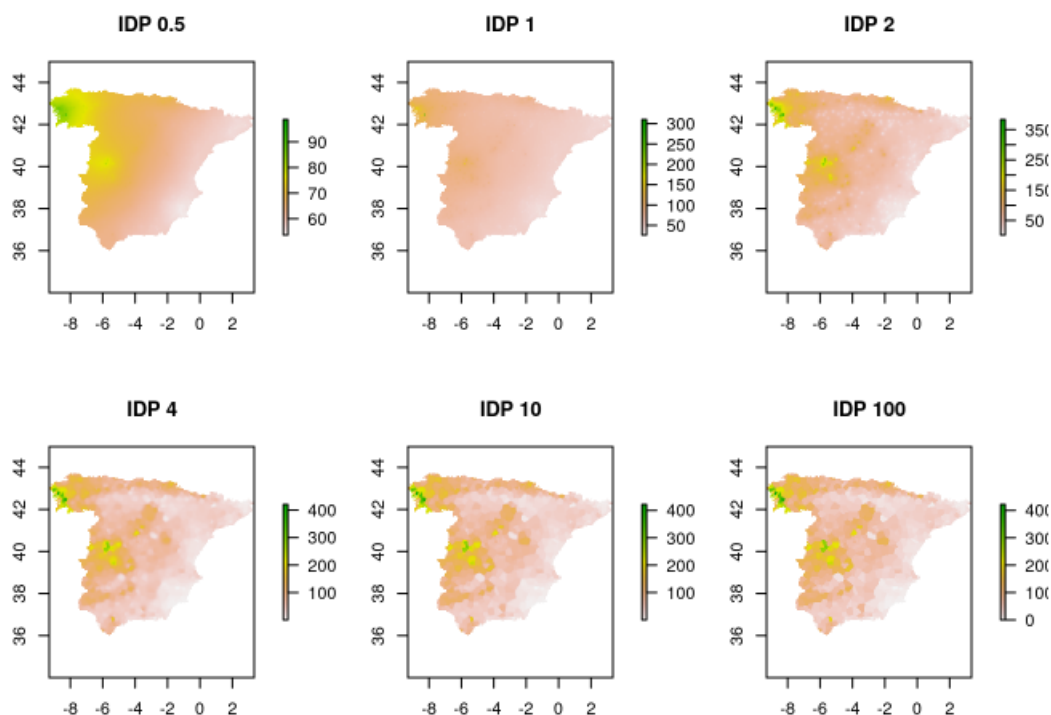
# Bucle d'interpolació per diferents valors de potència inversa
for (idp.val in c(0.5, 1, 2, 4, 10, 100)) {
```

```
# Creem objecte gstat amb la variable a estimar: pluja
gstat.parametros <- gstat(formula = pluja~1,
  locations = SP_puntos_df,
  set = list(idp = idp.val))

# Interpolació IDW
estaciones.idw <- interpolate(esp.raster, gstat.parametros)

# Mascara al raster per retallar límits
estaciones.idwr <- raster::mask(estaciones.idw, esp_pen)

# Visualització
plot(estaciones.idwr) + title(paste('IDP', idp.val))
}
```



Tal com podem veure, per valors més alts de *idp*, el pes dels punts més llunyans disminueix, apropant els valors màxim i mínim.

Pel que fa al *krigatge*, volem fer una estimació de valors desconeguts a partir de les dades que tenim de la pluja acumulada. Novament executem el següent codi:

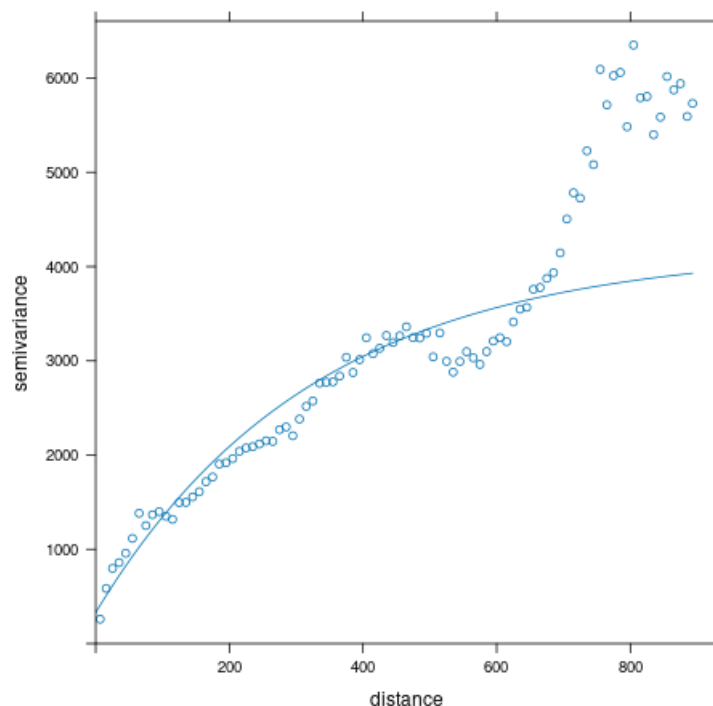
```
# Creem objecte gstat amb la variable a estimar: pluja
gstat.parametros <- gstat(formula = pluja~1,
  locations = SP_puntos_df)
```

```
# Calculem variograma empíric
variograma.gas <- variogram(gstat.parametros, width = 10)

# Ajustem variogrames teòric i empíric amb diferents models
var.teorico <- fit.variogram(variograma.gas,
  vgm(c("Exp", "Ste", "Sph", "Mat", "Gau", "Spl")))

# Visualització dels variogrames, avalua la qualitat de l'ajust
# Els punts = real, línia = teòric
plot(variograma.gas, var.teorico)
var.teorico
}
```

```
##   model    psill    range
## 1   Nug    328.8368    0.0000
## 2   Exp   3851.2879  327.1101
}
```

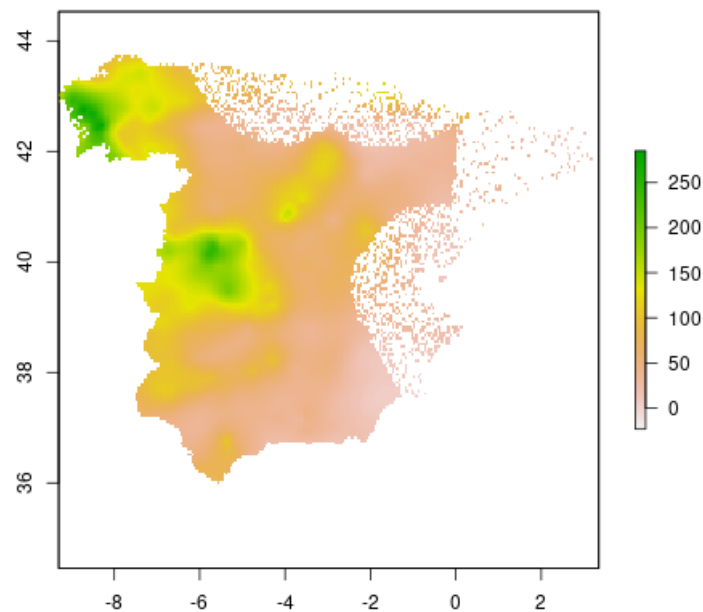


Tot seguit apliquem el *krigatge* utilitzant el model de variograma ajustat, per l'acumulació de pluja.

```
# Apliquem krigatge per la pluja
estaciones.ordkg <- krige(formula = pluja~1,
  locations = SP_puntos_df,
  newdata <- as(esp.raster, "SpatialGrid"),
  model = var.teorico, nmax = 100)
```

```
# Creem objecte raster
estaciones.okgr <- raster::mask(raster(estaciones.ordkg), esp_pen)

# Visualització
plot(estaciones.okgr)
```



Observem que tenim zones sense dades que corresponen a la gran quantitat de missatges *warnings* generats, probablement deguts a la presència de punts duplicats.

Repetim la visualització però eliminant els punts duplicats.

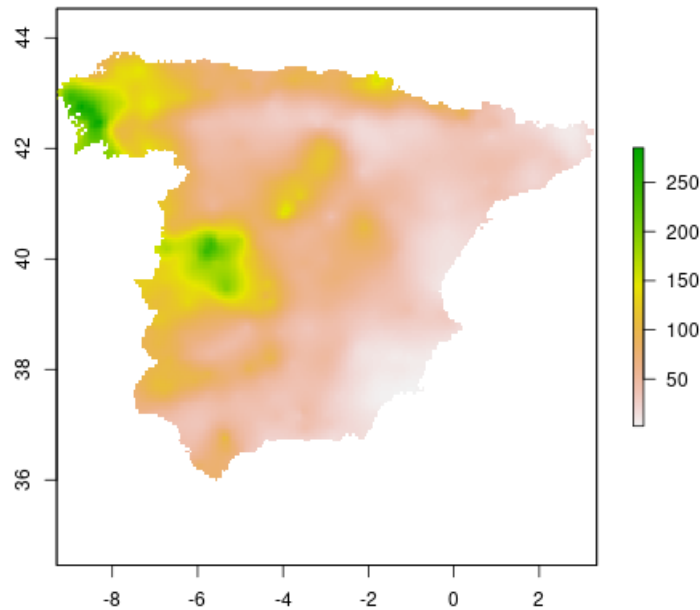
```
# Eliminem duplicats
estaciones.unique <- SP_puntos_df[-zerodist(SP_puntos_df)[, 1],]

# Krigatge amb el dataframe anterior
estaciones.ordkg <- krige(formula = pluja~1,
  locations = estaciones.unique,
  newdata <- as(esp.raster, "SpatialGrid"),
  model = var.teorico, nmax = 100)

# Creem objecte raster
estaciones.okgr <- raster::mask(raster(estaciones.ordkg), esp_pen)

# Visualització
```

```
plot(estaciones.okgr)
```



Obtenim un resultat millorat. Podem observar que aquesta visualització obtinguda comparant-la amb els resultats de la interpolació *IDW* que:

- per krigatge tenim una superfície més suavitzada,
- els valors màxims i mínims en la interpolació per *IDW* són més extrems, reflectint una major influència dels valors pròxims a cada punt interpolat.
- la superfície generada per interpolació per *IDW* presenta els denominats “ulls de bou”, àrees de major influència de punts amb valors extrems locals.

Segona part: temperatura mínima per Catalunya

Volem fer el mateix que la primera part però ara considerant la temperatura mínima enregistrada per les estacions a Catalunya, apliquem la distància inversa ponderada i el krigatge com a mètodes d'interpolació espacial.

Necessitem un *SpatialPointsDataFrame* que obtenim a partir de `temp_min` de l'exercici 1 i que té un camp que és `temp`, (recordem també que ja està neta de NAs):

```
# Obtenim les coordenades de les estacions
coords <- st_coordinates(temp_min$geometry)
SP_puntos <- SpatialPoints(coords, proj4string = CRS('+proj=longlat +datum=WGS84'))
temp_min_SPDF <- as.data.frame(temp_min)
```



```
temp_min_SPDF$geometry <- NULL
SP_puntos_df <- SpatialPointsDataFrame(SP_puntos, temp_min_SPDF)
```

Recordem que a cat tenim les províncies catalanes. Seguint els mateixos passos que a l'apartat 1 i comencem amb el mètode de la distància inversa ponderada (*IDW*) per tal d'estimar els valors en una malla regular a partir de la temperatura mínima de les estacions. Ho fem amb diferents valors de potència inversa.

```
# Fem servir Catalunya i creem raster amb resolució de 0.05
cat.raster <- raster(cat, res = 0.05)

# Layout dels mapes a crear: 2 files x 3 columnes
par(mfrow=c(2, 3), mai=c(0.5, 0.5, 0.5, 0.5))

# Bucle d'interpolació per diferents valors de potència inversa
for (idp.val in c(0.5, 1, 2, 4 , 10, 100)) {

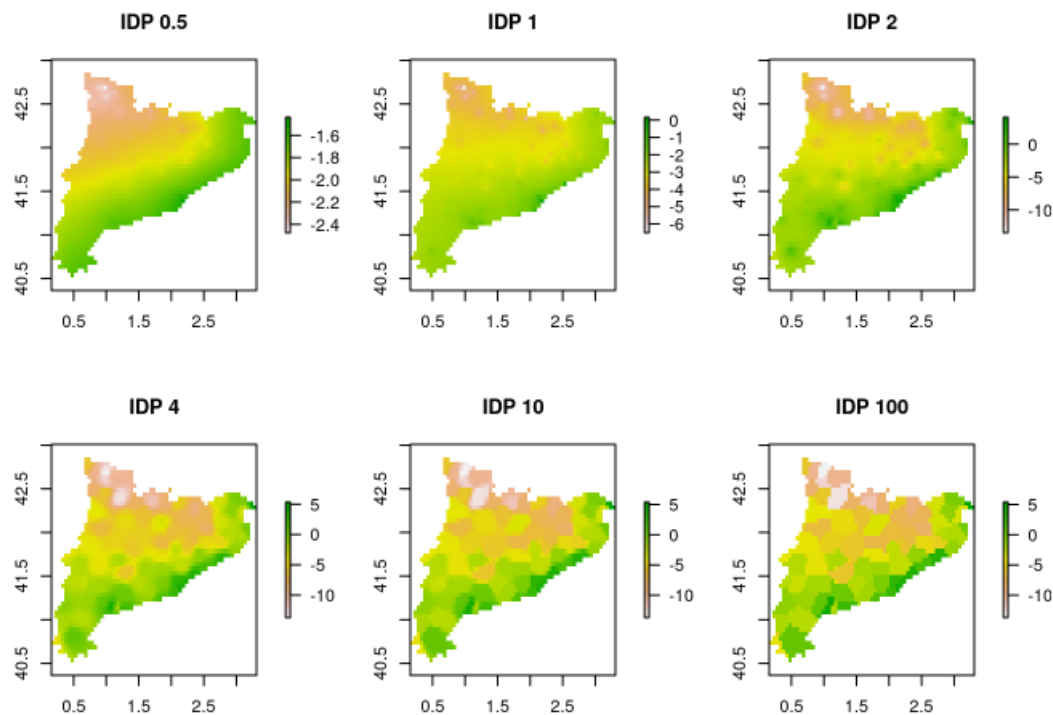
  # Creem objecte gstat amb la variable a estimar: temp
  gstat.parametros <- gstat(formula = temp~1,
    locations = SP_puntos_df,
    set = list(idp = idp.val))

  # Interpolació IDW
  estaciones.idw <- interpolate(cat.raster, gstat.parametros)

  # Mascara al raster per retallar límits
  estaciones.idwr <- raster::mask(estaciones.idw, cat)

  # Visualització
  plot(estaciones.idwr) + title(paste('IDP', idp.val))
}
```

Igual que abans, per valors més alts de *idp*, el pes dels punts més llunyans disminueix, apropant els valors màxim i mínim.



Pel que fa al *krigatge*, volem fer una estimació de valors desconeguts a partir de les dades que tenim de les temperatures mínimes. Novament executem el següent codi:

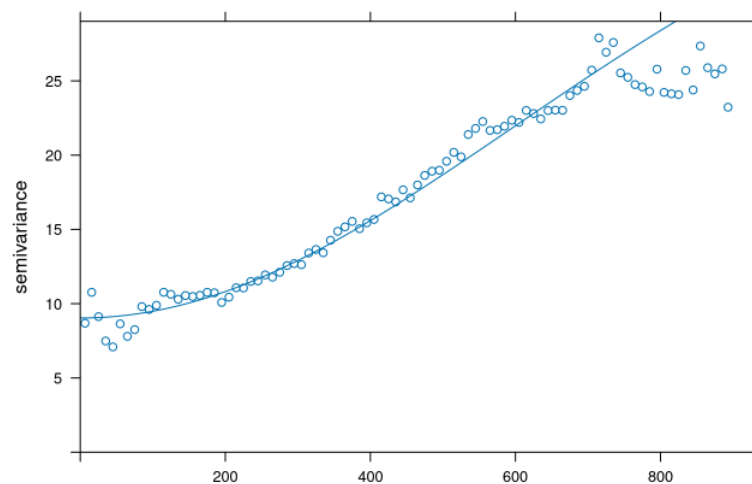
```
# Creem objecte gstat amb la variable a estimar: temp
gstat.parametros <- gstat(formula = temp~1,
  locations = SP_puntos_df)

# Calculem variograma empíric
variograma.gas <- variogram(gstat.parametros, width = 10)

# Ajustem variogrames teòric i empíric amb diferents models
var.teorico <- fit.variogram(variograma.gas,
  vgm(c("Exp", "Ste", "Sph", "Mat", "Gau", "Spl")))

# Visualització dels variogrames, avalua la qualitat de l'ajust
# Els punts = real, línia = teòric
plot(variograma.gas, var.teorico)
var.teorico
}
```

```
##   model    psill  range
## 1  Nug  9.034956  0.000
## 2  Gau 32.773675 846.581
}
```

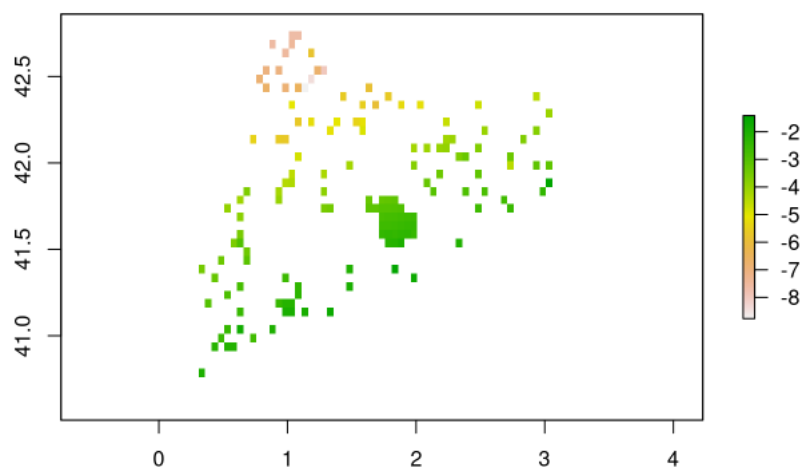


Tot seguit apliquem el *krigatge* utilitzant el model de variograma ajustat, per la temperatura mínima.

```
# Apliquem krigatge per la temperatura mínima
estaciones.ordkg <- krige(formula = temp~1,
  locations = SP_puntos_df,
  newdata <- as(esp.raster, "SpatialGrid"),
  model = var.teorico, nmax = 100)

# Creem objecte raster
estaciones.okgr <- raster::mask(raster(estaciones.ordkg), cat)

# Visualització
plot(estaciones.okgr)
```



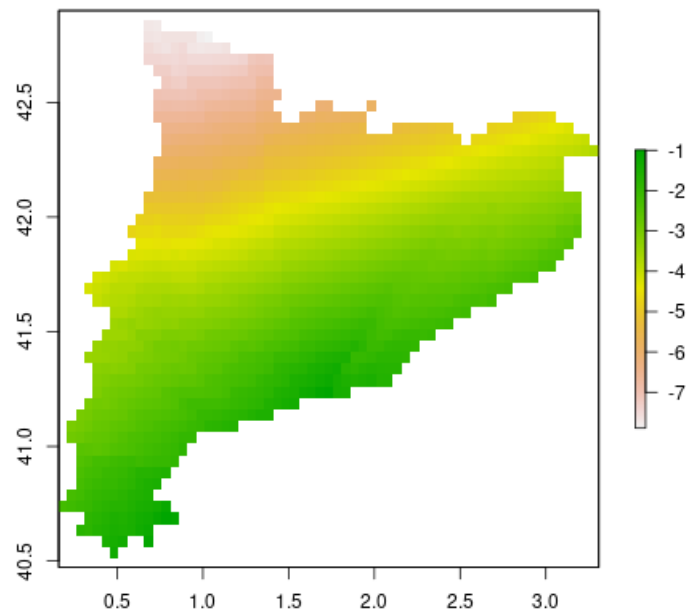
Observem que, igual que abans, tenim zones sense dades, fins i tot costa distingir la forma de Catalunya. Així que repetim la visualització però eliminant els punts duplicats.

```
# Eliminem duplicats
estaciones.unique <- SP_puntos_df[~zerodist(SP_puntos_df)[, 1],]

# Krigatge amb el dataframe anterior
estaciones.ordkg <- krige(formula = temp~1,
  locations = estaciones.unique,
  newdata <- as(esp.raster, "SpatialGrid"),
  model = var.teorico, nmax = 100)

# Creem objecte raster
estaciones.okgr <- raster::mask(raster(estaciones.ordkg), cat)

# Visualització
plot(estaciones.okgr)
```



Obtenim un resultat millorat. I tal com podem veure, les temperatures més baixes corresponen al Pirineu, en canvi la diagonal cap a la costa les temperatures mínimes són més suaus, tal com mostrava el mapa de l'exercici 1.