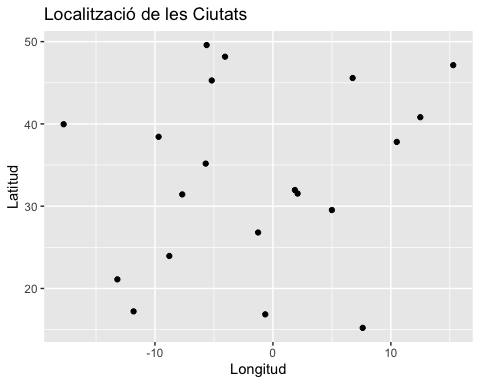
PAC3

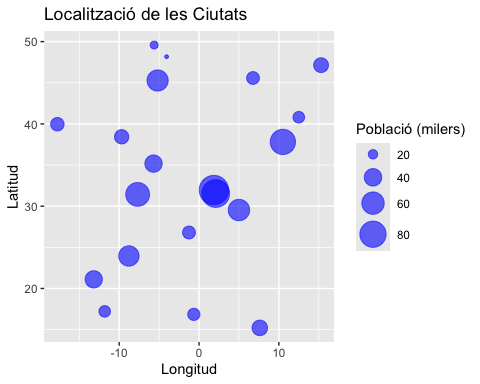
# Instal·lació i càrrega de paquets necessaris  
if (!require(ggplot2)) install.packages("ggplot2", dependencies = TRUE)  
if (!require(sf)) install.packages("sf", dependencies = TRUE)  
if (!require(raster)) install.packages("raster", dependencies = TRUE)  
if (!require(gstat)) install.packages("gstat", dependencies = TRUE)

## Exercici 1: Anàlisi i visualització de patrons de punts

# Carreguem paquet  
library(ggplot2)  
  
set.seed(100) # Establim una llavor para a la reproduibilitat  
n\_ciutats <- 20 # Definim el nombre de ciutats que es crearan  
ciutats <- data.frame(   
 nom = paste0("Ciutat ", LETTERS[1:n\_ciutats]), # Genera noms per a les ciutats   
 longitud = runif(n\_ciutats, -20, 20), # Genera longituds aleatòries   
 latitud = runif(n\_ciutats, 10, 50), # Genera latituds aleatòries  
 poblacio = sample(10000:100000, n\_ciutats, replace = TRUE) # Genera poblacions aleatòries  
)  
  
# Generem un plot bàsic de punts amb latitud i longitud  
ggplot(ciutats, aes(x = longitud, y = latitud)) +  
geom\_point() +  
labs(title = "Localització de les Ciutats", x = "Longitud", y = "Latitud")

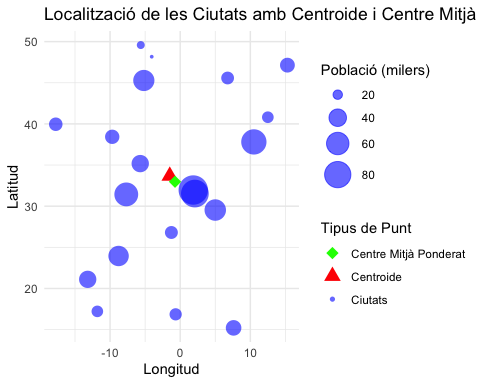


# Creem un plot de dispersió de longitud i latitud on la mida dels punts és proporcional a la població de cada ciutat  
ggplot(ciutats, aes(x = longitud, y = latitud, size = poblacio / 1e3)) +  
 geom\_point(color = "blue", alpha = 0.6) +  
 labs(title = "Localització de les Ciutats", x = "Longitud", y = "Latitud", size = "Població (milers)") +  
 scale\_size\_continuous(range = c(1, 10)) # Ajustem la mida dels punts per a una millor visualització



## Exercici 2: Càlcul de mesures centrogràfiques

#Utilitzant el conjunt de dades de ciutats creat anteriorment:  
# - Calculem el centroide de les ciutats  
centroides <- data.frame(  
 longitud = mean(ciutats$longitud),  
 latitud = mean(ciutats$latitud)  
)  
  
# - Calculem el centre mitjà ponderat, on les coordenades es ponderen per la població de cada ciutat  
centre\_ponderat <- data.frame(  
 longitud = sum(ciutats$longitud \* ciutats$poblacio) / sum(ciutats$poblacio),  
 latitud = sum(ciutats$latitud \* ciutats$poblacio) / sum(ciutats$poblacio)  
)  
  
# - Creem un gràfic de dispersió amb ggplot2 que mostri la localització de les ciutats i afegiu també punts per al centroide i el centre mitjà ponderat.  
# Gràfic de dispersió amb ggplot2  
ggplot() +  
 geom\_point(data = ciutats, aes(x = longitud, y = latitud, size = poblacio / 1e3, shape = "Ciutats"), color = "blue", alpha = 0.6) + # Punts per a les ciutats  
 geom\_point(data = centroides, aes(x = longitud, y = latitud, shape = "Centroide"), color = "red", size = 4) + # Punt per al centroide  
 geom\_point(data = centre\_ponderat, aes(x = longitud, y = latitud, shape = "Centre Mitjà Ponderat"), color = "green", size = 4) + # Punt per al centre mitjà ponderat  
 scale\_shape\_manual(  
 values = c("Ciutats" = 16, "Centroide" = 17, "Centre Mitjà Ponderat" = 18), # Assigna formes personalitzades  
 name = "Tipus de Punt" # Nom de la llegenda  
 ) +  
 labs(  
 title = "Localització de les Ciutats amb Centroide i Centre Mitjà Ponderat",  
 x = "Longitud",  
 y = "Latitud",  
 size = "Població (milers)"  
 ) +  
 scale\_size\_continuous(range = c(1, 10)) + # Ajusta la mida dels punts  
 theme\_minimal() # Tema minimalista per al gràfic



## Exercici 3: Descripció i aplicació de mètodes d’interpolació

#### Descripció dels Mètodes d’Interpolació

1. Polígons de Thiessen

* Principi bàsic: Aquest mètode divideix l’espai en regions (polígons) al voltant de cada punt de mostreig. Cada punt dins d’un polígon s’assigna al valor del punt de mostreig més proper. Els polígons es construeixen traçant línies perpendiculars als segments que uneixen els punts de mostreig.
* Avantatges: És senzill i no requereix supòsits sobre la distribució espacial de les dades.
* Limitacions: No té en compte la variació entre punts i pot produir resultats abruptes.

1. Distància Inversa Ponderada (IDW)

* Principi bàsic: Aquest mètode estima el valor d’un punt desconegut com una mitjana ponderada dels valors dels punts de mostreig propers. Els pesos són inversament proporcionals a la distància entre el punt desconegut i els punts de mostreig.
* Avantatges: És fàcil d’implementar i permet ajustar la influència dels punts propers mitjançant un exponent.
* Limitacions: Assumeix que els punts més propers tenen més influència, cosa que pot no ser certa en tots els casos.

1. Krigatge Ordinari

* Principi bàsic: És un mètode geoestadístic que utilitza la variació espacial (semivariograma) per estimar els valors en punts desconeguts. Té en compte tant la distància com la correlació espacial entre punts.
* Avantatges: Proporciona estimacions òptimes i inclou una mesura d’incertesa.
* Limitacions: És més complex i requereix un coneixement previ de la variació espacial de les dades.

#### Exemples d’Aplicació Pràctica

* Polígons de Thiessen: Assignació de zones de servei a hospitals o estacions de bombers.
* IDW: Creació de mapes de temperatura en una regió basada en dades de diverses estacions meteorològiques.
* Krigatge Ordinari: Estimació de reserves minerals en una mina o distribució de nutrients en camps agrícoles.

## Exercici 4:

library(sf)  
library(raster)  
library(gstat)  
  
# Generem un conjunt de dades simulades  
set.seed(100) # Per reproduïbilitat  
n\_pluviometres <- 100  
pluviometres <- data.frame(  
 id = 1:n\_pluviometres,  
 x = runif(n\_pluviometres, 0, 100), # Coordenades X dins d'un rectangle  
 y = runif(n\_pluviometres, 0, 100), # Coordenades Y dins d'un rectangle  
 pluja = runif(n\_pluviometres, 0, 50) # Valors de pluja entre 0 i 50  
)  
  
# Convertim a objecte espacial amb sf  
pluviometres\_sf <- st\_as\_sf(pluviometres, coords = c("x", "y"), crs = 4326)  
  
# Creem una malla per a la interpolació  
resolucio <- 1 # Resolució del raster  
x\_range <- seq(0, 100, by = resolucio)  
y\_range <- seq(0, 100, by = resolucio)  
grid <- expand.grid(x = x\_range, y = y\_range)  
grid\_sf <- st\_as\_sf(grid, coords = c("x", "y"), crs = 4326)  
  
# Interpolem amb la tècnica IDW corregida  
gstat\_model <- gstat::gstat(formula = pluja ~ 1, data = as(pluviometres\_sf, "Spatial"), nmax = 10, set = list(idp = 2))  
  
# Convertim la malla a un objecte Spatial  
grid\_sp <- as(grid\_sf, "Spatial")  
  
# Realitzem la interpolació amb predict  
idw\_result <- predict(gstat\_model, newdata = grid\_sp)

## [inverse distance weighted interpolation]

# Convertim el resultat de la interpolació a un data frame  
idw\_df <- as.data.frame(idw\_result)  
colnames(idw\_df)[1:3] <- c("x", "y", "var1.pred")  
  
# Convertim el resultat a un raster  
raster\_result <- rasterFromXYZ(idw\_df[, c("x", "y", "var1.pred")])  
  
# Visualitzem el raster resultant  
plot(raster\_result, main = "Interpolació IDW de Pluja")  
points(st\_coordinates(pluviometres\_sf), col = "red", pch = 20, cex = 0.7) # Pluviòmetres

