

Análisis de granulometría de sedimentos de playa Palenque

Ana Valera, Carolain Pérez, José Martínez (tali)

2023-02-16

Paquetes y funciones

```
# remotes::install_github("bceaton/GSDtools")
library(GSDtools)
library(kableExtra)
library(tidyverse)

## -- Attaching packages ----- tidyverse 1.3.1 --
## v ggplot2 3.3.5      v purrr  0.3.4
## v tibble  3.1.7      v dplyr  1.0.10
## v tidyr   1.2.1      v stringr 1.4.0
## v readr   2.1.3      v forcats 0.5.1

## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter()      masks stats::filter()
## x dplyr::group_rows() masks kableExtra::group_rows()
## x dplyr::lag()         masks stats::lag()

source('R/funciones.R')
```

Datos

```
datos_orig <- leer_tabla('pesos_granulometria_playa_palenque.ods')
```

Obtener dimensiones *phi* a partir de mm

```

pesos_tamanos <- datos_orig[-7,] #Quitando última fila
(pesos_tamanos$micrones <- as.numeric(pesos_tamanos$micrones)) # Convertida a numérica

## [1] 4000 2000 500 250 125 63

(pesos_tamanos$mm <- pesos_tamanos$micrones / 1000) # Convertida a numérica

## [1] 4.000 2.000 0.500 0.250 0.125 0.063

(pesos_tamanos$phi <- round(obtener_phi(pesos_tamanos$mm), 1)) #Crear columna de diámetro

## [1] -2 -1 1 2 3 4

#Conservando columnas de interés
pesos_tamanos_mm_phi <- pesos_tamanos[, c('mm', 'phi', grep('^A', colnames(pesos_tamanos), value = T))]
# Imprimir con kable
pesos_tamanos_mm_phi %>%
  kable(booktabs=T) %>%
  kable_styling(latex_options = c("HOLD_position", "scale_down")) %>%
  gsub(' NA ', '', .)

```

mm	phi	AVA001	AVA002	AVA003	AVA004	AVA005	AVA006
4.000	-2	4.0	114.0	6.3	0.2	42.0	62.2
2.000	-1	0.3	0.0	0.6	0.8	0.8	66.0
0.500	1	25.3	51.2	28.8	12.3	23.8	53.9
0.250	2	197.1	246.6	270.4	2513.0	345.5	8.9
0.125	3	133.0	139.1	243.5	390.5	346.0	7.0
0.063	4	3.7	5.7	16.2	28.2	12.6	0.7

Generar frecuencias acumuladas

```
# Copiando objeto
pesos_tamanos_mm_phi_rel <- pesos_tamanos_mm_phi
# Columnas de pesos relativos
pesos_tamanos_mm_phi_rel[, grep('^A', colnames(pesos_tamanos_mm_phi_rel))] <- sapply(
  pesos_tamanos_mm_phi_rel[, grep('^A', colnames(pesos_tamanos_mm_phi_rel))],
  function(x) x/sum(x)*100)
# Columnas de pesos relativos acumulados
pesos_tamanos_mm_phi_rel_acum <- sapply(
  pesos_tamanos_mm_phi_rel[, grep('^A', colnames(pesos_tamanos_mm_phi_rel))],
  function(x) cumsum(x))
# Añadiendo "_acum" a nombre de columnas correspondientes
colnames(pesos_tamanos_mm_phi_rel_acum) <- paste0(colnames(pesos_tamanos_mm_phi_rel_acum), '_acum')
# cbind para unir tabla de valores relativos con acumulados
pesos_tamanos_mm_phi_rel_acum_unido <- cbind(pesos_tamanos_mm_phi_rel, pesos_tamanos_mm_phi_rel_acum)
# Imprimiendo con kable
pesos_tamanos_mm_phi_rel_acum_unido %>%
  kable(booktabs=T, digits = 3) %>%
  kable_styling(latex_options = c("HOLD_position", "scale_down")) %>%
  gsub(' NA ', '', .)
```

mm	phi	AVA001	AVA002	AVA003	AVA004	AVA005	AVA006	AVA001_acum	AVA002_acum	AVA003_acum	AVA004_acum	AVA005_acum	AVA006_acum
4.000	-2	1.101	20.481	1.113	0.007	5.450	31.303	1.101	20.481	1.113	0.007	5.450	31.303
2.000	-1	0.083	0.000	0.106	0.027	0.104	33.216	1.183	20.481	1.220	0.034	5.553	64.519
0.500	1	6.962	9.199	5.090	0.418	3.088	27.126	8.145	29.680	6.310	0.452	8.641	91.646
0.250	2	54.238	44.305	47.791	85.331	44.829	4.479	62.383	73.985	54.100	85.783	53.471	96.125
0.125	3	36.599	24.991	43.036	13.260	44.894	3.523	98.982	98.976	97.137	99.042	98.365	99.648
0.063	4	1.018	1.024	2.863	0.958	1.635	0.352	100.000	100.000	100.000	100.000	100.000	100.000

```
pesos_tamanos_mm_phi_rel_acum_unido <- rbind(
  c(8, obtener_phi(8), rep(0,12)),
  pesos_tamanos_mm_phi_rel_acum_unido)
# write_csv(pesos_tamanos_mm_phi_rel_acum_unido, 'pesos_tamanos_mm_phi_rel_acum_unido.csv')
```

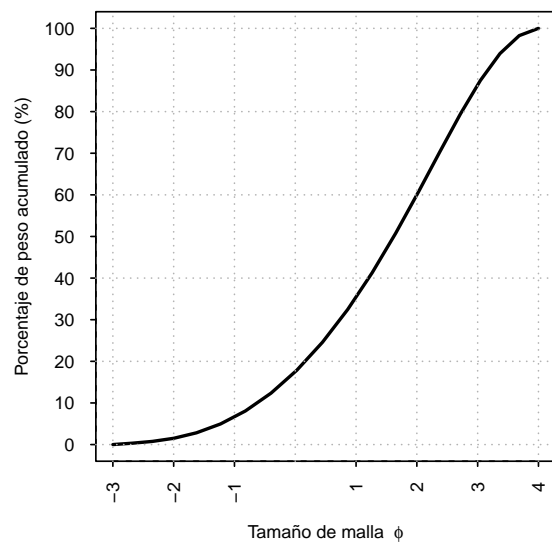
Generar curvas de tamaño de grano de frecuencias acumuladas suavizadas

Adaptado de: <https://rpubs.com/manchulu/706871>

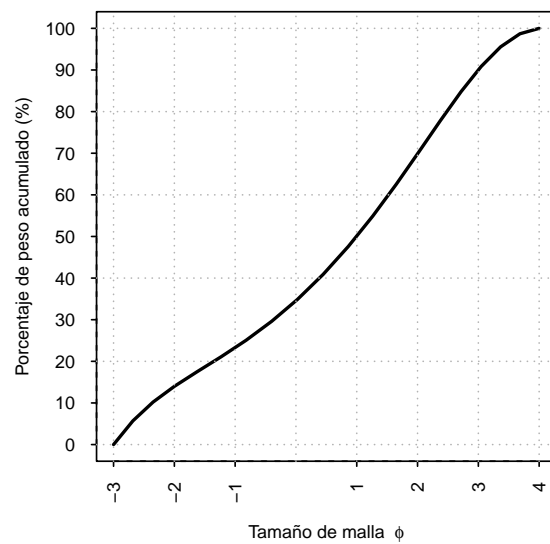
- Usando phi

```
if(interactive()) dev.new()
par(mfrow = c(2, 3))
lapply(
  grep('AVA[0-9]{3,}$', colnames(pesos_tamamos_mm_phi_rel_acum_unido), value = T),
  function(x) crear_grafico(x))
```

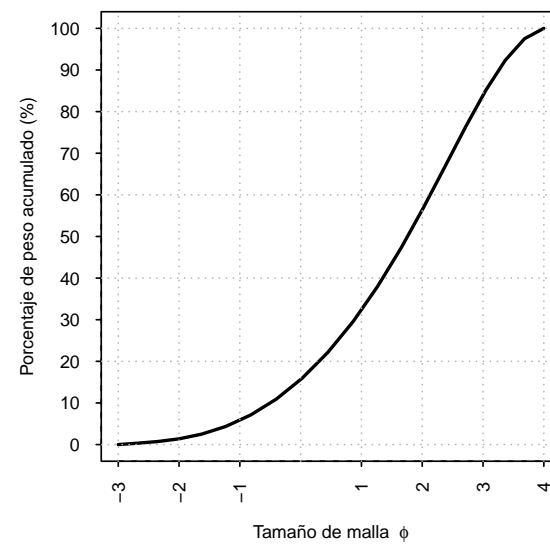
Curva de frecuencia acumulada AVA001



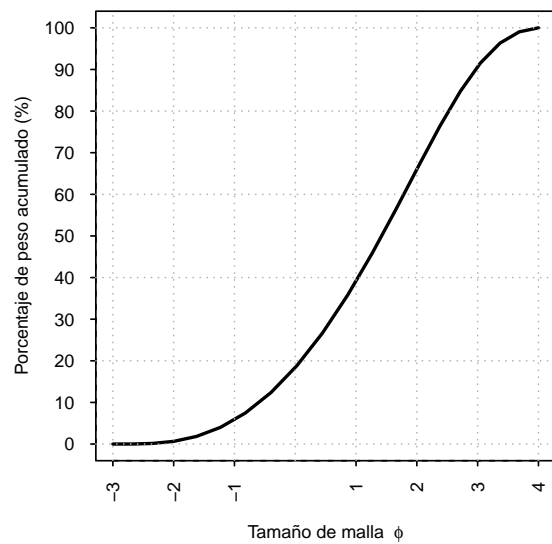
Curva de frecuencia acumulada AVA002



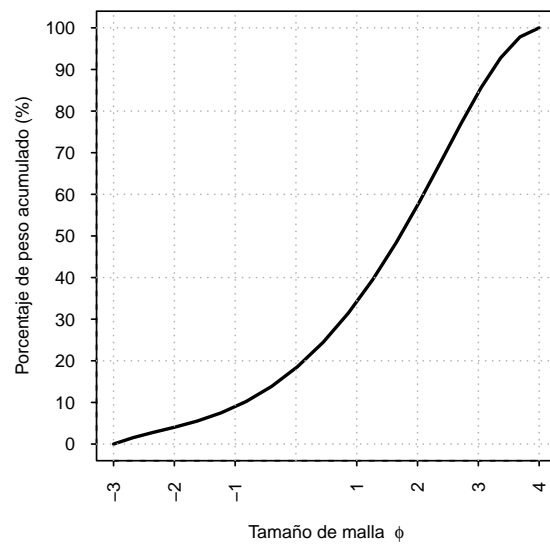
Curva de frecuencia acumulada AVA003



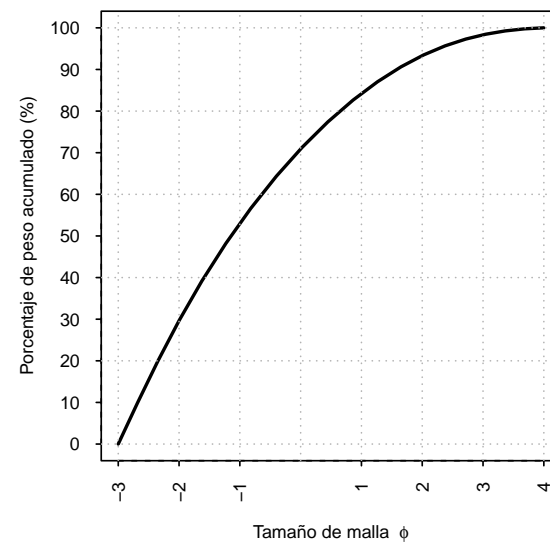
Curva de frecuencia acumulada AVA004



Curva de frecuencia acumulada AVA005



Curva de frecuencia acumulada AVA006

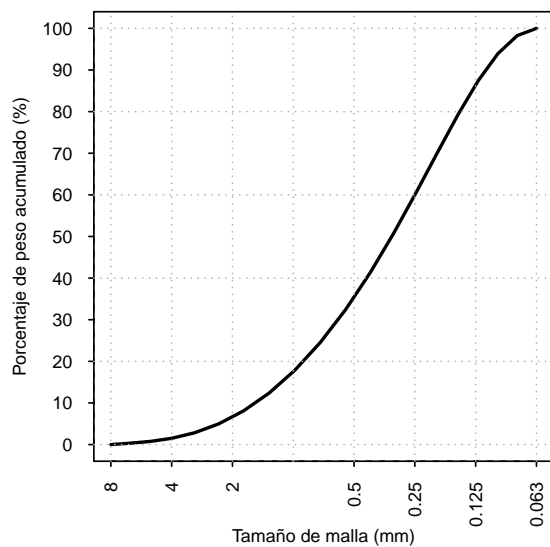


```
## [[1]]
## NULL
##
## [[2]]
## NULL
##
## [[3]]
## NULL
##
## [[4]]
## NULL
##
## [[5]]
## NULL
##
## [[6]]
## NULL
```

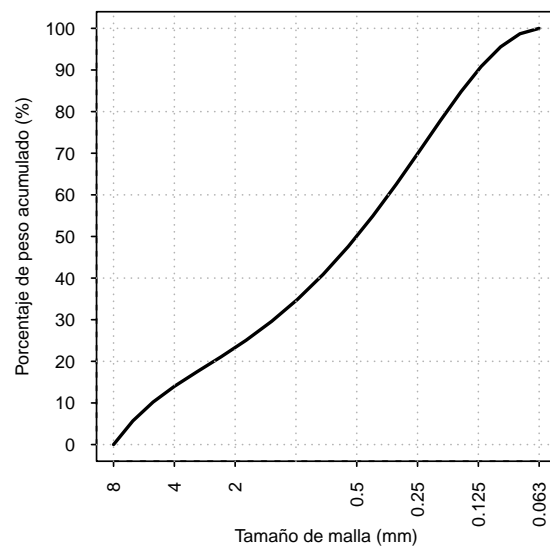
- Usando mm

```
if(interactive()) dev.new()
par(mfrow = c(2, 3))
lapply(
  grep('AVA[0-9]{3,}$', colnames(pesos_tamanos_mm_phi_rel_acum_unido), value = T),
  function(x) crear_grafico(x, phi=F))
```

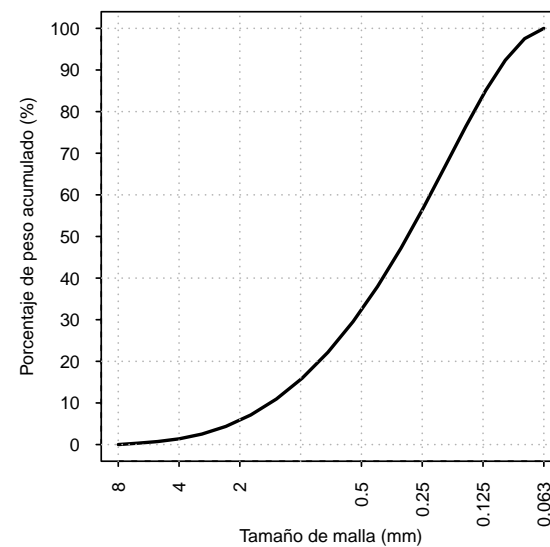
Curva de frecuencia acumulada AVA001



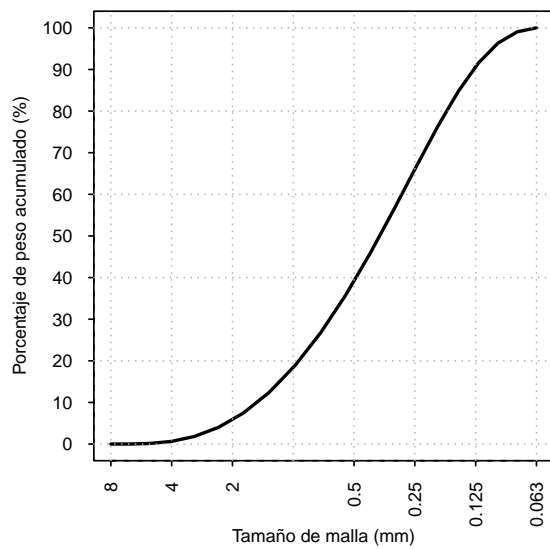
Curva de frecuencia acumulada AVA002



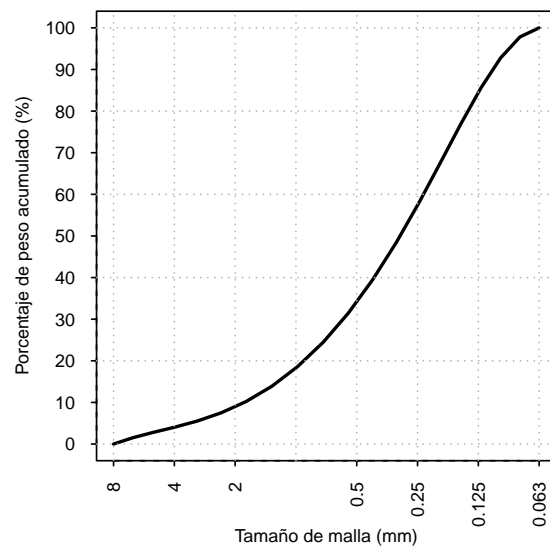
Curva de frecuencia acumulada AVA003



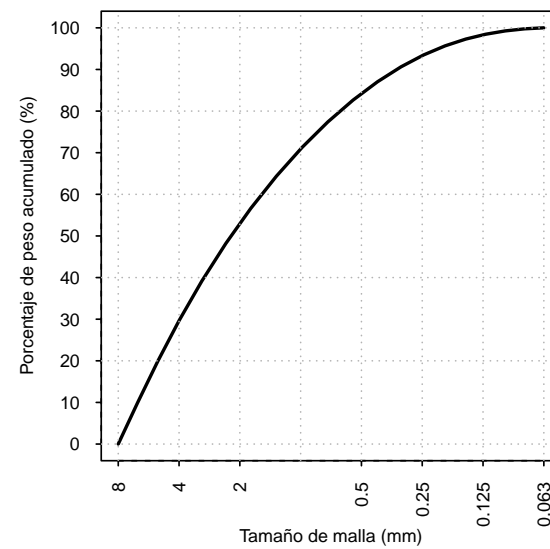
Curva de frecuencia acumulada AVA004



Curva de frecuencia acumulada AVA005



Curva de frecuencia acumulada AVA006



```
## [[1]]  
## NULL  
##  
## [[2]]  
## NULL  
##  
## [[3]]  
## NULL  
##  
## [[4]]  
## NULL  
##  
## [[5]]  
## NULL  
##  
## [[6]]  
## NULL
```