

TAREA # 4

Diagramas de Voronoi

Natalia Berenice Pérez López

21 de septiembre de 2021

1. Objetivo

El objetivo de esta práctica es examinar el efecto del número de semillas k en la penetración de las grietas que se forman, en términos de la mayor distancia Manhattan entre la grieta y el borde de la pieza, todo esto manteniendo constante el tamaño n de la zona, visualizando los resultados con diagramas caja-bigote o similar sobre las réplicas y aplicando métodos estadísticos para establecer el efecto de k en ello.

2. Desarrollo

Para generar el código objetivo de esta práctica se realizaron diversos códigos prueba, los cuales se encuentran en [mi repositorio](#) en GitHub. Se inició tomando como base el código revisado en clase [2] para generar una matriz de tamaño n que contiene cierta cantidad de semillas k y que es fracturada por una grieta de longitud y dirección aleatoria. Los valores que se eligieron para las variables del experimento se muestran en el cuadro 1.

Cuadro 1: Variables del experimento.

VARIABLE	VALOR
Semillas (k)	6, 12, 24
Réplicas	15 para cada k
Tamaño de la matriz (n)	18

Las modificaciones que se le realizaron al código de RStudio revisado en clase fueron; agregar un ciclo `for` para variar k , cambiar el número de réplicas y definir el tamaño constante de la matriz. Se intentó modificar el código para obtener automáticamente la distancia Manhattan de la penetración de la fractura al borde de la matriz, pero esto no resultó con éxito. También se realizó una videollamada con mi compañera Claudia Hernández para discutir ideas sobre como realizar el objetivo de la práctica, sin embargo tampoco tuvimos éxito, por lo cual óptamos por la idea de calcular la distancia Manhattan en `excel` utilizando las posiciones de los 0 de la matriz, la cual se puede obtener fácilmente en RStudio. Con las matrices exportadas a `excel` se realizó el cálculo de las distancias de cada posición a los bordes de la matriz, después se obtuvieron la distancias mínimas y de éstas se obtuvo la distancia máxima que hace referencia a la distancia Manhattan de los bordes a la posición más profunda que tiene la grieta.

A continuación se muestra un fragmento del código utilizado para variar el número de semillas k .

```
1 n <- 18
2 numsem <- c(6, 12, 24)
3
4 for (k in numsem) {
5   zona <- matrix(rep(0, n * n), nrow = n, ncol = n)
6   x <- rep(0, k) # ocupamos almacenar las coordenadas x de las semillas
7   y <- rep(0, k) # igual como las coordenadas y de las semillas
8
9   for (semilla in 1:k) {
10    while (TRUE) { #hasta que hallamos una posicion vacia para la semilla
11      fila <- sample(1:n, 1)
12      columna <- sample(1:n, 1)
13      if (zona[fila, columna] == 0){
14        zona[fila, columna] = semilla
15        x[semilla] <- columna
16        y[semilla] <- fila
17      }
18    }
19  }
```

```

17     break
18   }
19 }
20 }
21
22 celda <- function(pos){
23   fila <- floor((pos - 1) / n) + 1
24   columna <- ((pos - 1) %% n) + 1
25   if (zona[fila, columna] > 0){ #es una semilla
26     return(zona[fila, columna])
27   } else{
28     cercano <- NULL #sin valor por el momento
29     menor <- n * sqrt(2) #mayor posible para comenzar la busqueda
30     for (semilla in 1:k){
31       dx <- columna - x[semilla]
32       dy <- fila - y[semilla]
33       dist <- sqrt(dx^2 + dy^2)
34       if (dist < menor){
35         cercano <- semilla
36         menor <- dist
37       }
38     }
39     return(cercano)
40   }
41 }
42
43 celdas <- sapply(1:(n * n), function(p) celda(p))
44 voronoi <- matrix(celdas, nrow = n, ncol = n, byrow=TRUE)
45 rotate <- function(x) t(apply(x, 2, rev))
46 sem=paste("p4s", "-", k, ".png", sep="")
47 png(sem)
48 par(mar = c(0,0,0,0))
49 image(rotate(zona), col=rainbow(k+1), xaxt='n', yaxt='n')
50 graphics.off()
51 cel=paste("p4c", "-", k, ".png", sep="")
52 png(cel)
53 par(mar = c(0,0,0,0))
54 image(rotate(voronoi), col=rainbow(k+1), xaxt='n', yaxt='n')
55 graphics.off()
56 }

```

Listing 1: Código para variar la cantidad de semillas k .

Enseguida se muestra el fragmento del código que permite exportar a excel la matriz que contiene la fractura.

```

1   if (elegido > 0) { # si se va a propagar
2     vecino <- vp[elegido,]
3     xg <- xg + vecino$dx
4     yg <- yg + vecino$dy
5   } else {
6     break # ya no se propaga
7   }
8 }
9
10 if (largo >= limite) {
11   png(paste("p4g_", replica, ".png", sep=""))
12   par(mar = c(0,0,0,0))
13   image(rotate(grieta), col=rainbow(k+1), xaxt='n', yaxt='n')
14   graphics.off()
15 }
16 return(grieta)
17 }
18
19 suppressMessages(library(doParallel))
20 registerDoParallel(makeCluster(detectCores() - 1))
21 lagrieta <- foreach(r = 1:1, .combine=c) %dopar% propaga(r)
22 stopImplicitCluster()
23 lagrieta
24 datos = rbind(datos, lagrieta)
25 write_xlsx(datos, "misdatos.xlsx")

```

Listing 2: Código para exportar la matriz de la fractura a excel.

Para cada 0 que tenía la matriz se obtuvo su coordenada x, y y se calcularon las distancias hacia el borde izquierdo, derecho, superior e inferior de la siguiente manera:

Cuadro 2: Fórmulas para calcular las distancias.

$di = x$	distancia al borde izquierdo
$dd = n - x$	distancia al borde derecho
$ds = n - y$	distancia al borde superior
$din = y$	distancia al borde inferior
$menores = \min(di, dd, ds, din)$	distancia mínima
$max(menores)$	distancia Manhattan

Las distancias Manhattan que se obtuvieron para cada réplica se muestran en el cuadro 3.

Cuadro 3: Distancias Manhattan de acuerdo a número de semillas para cada réplica.

Semillas	Distancias Manhattan
6	6, 4, 5, 5, 9, 4, 6, 7, 9, 4, 9, 7, 6, 5, 7
12	9, 6, 4, 9, 9, 4, 5, 5, 5, 8, 9, 5, 6, 6, 4
24	9, 8, 5, 9, 9, 4, 6, 5, 9, 9, 9, 9, 8, 9, 5

Las distancias del cuadro 3 se importaron de excel a RStudio para realizar el diagrama caja-bigote mostrado en la figura 1.

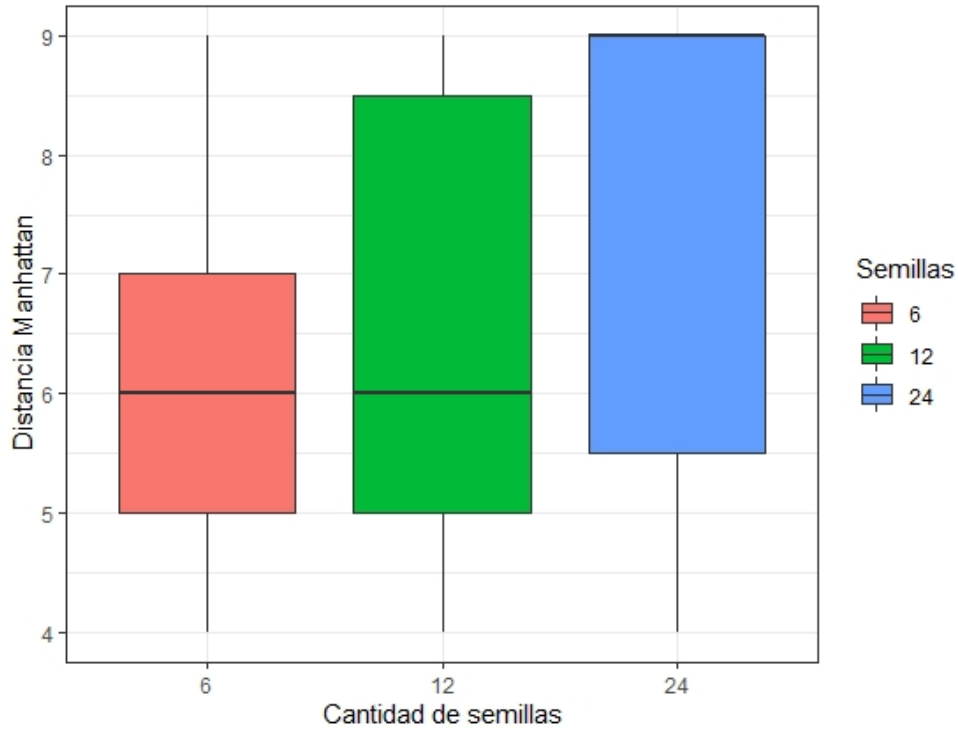


Figura 1: Distancia Manhattan de acuerdo al número de semillas en una matriz de $n = 18$.

En la figura 1 podemos observar que con un número pequeño de semillas se obtienen distancias más cortas para la penetración de la fractura, y al aumentar el número de semillas la distancia de penetración de la fractura se puede hacer más larga.

Para analizar si existe una relación entre la variación del número de semillas y la distancia de penetración de la fractura, primero se eligió utilizar la prueba estadística ANOVA de una vía, pero debido a los resultados obtenidos al revisar la normalidad de los datos, se eligió realizar la prueba estadística **Kruskal Wallis** [1].

En el cuadro 4 se resumen los resultados de la revisión de los supuestos para poder aplicar la prueba estadística ANOVA. El supuesto outliers se refiere a la cantidad de valores atípicos que existen en los grupos, la normalidad por grupos se obtuvo con la prueba de **Shapiro Wilk**, y la homogeneidad de varianza se obtuvo con la prueba de **Levene**.

Cuadro 4: Resultados del los supuestos para aplicar la prueba estadística ANOVA.

Outliers	0
Normalidad por grupo	6 semillas: $p = 0,0749$ 12 semillas: $p = 0,0100$ 24 semillas: $p = 0,0006$
Homogeneidad de varianza	$p = 0,926$

En los resultados se observa que los datos de los grupos de 12 y 24 semillas no tienen normalidad. Por lo tanto se debe realizar la prueba estadística **Kruskal Wallis** ya que ésta es útil para cuando no se tiene normalidad en los grupos de datos.

Al realizar la prueba **Kruskal Wallis** se obtienen los resultados mostrados en el cuadro 5.

Cuadro 5: Resultados al aplicar la prueba estadística **Kruskal Wallis**.

Chi cuadrada	Valor de p
3,8803	0,1437

Hipótesis nula : Las medias son iguales en todos los grupos

Debido a que $p > 0,05$ se acepta la hipótesis nula, no existen diferencias significativas entre las medias de los grupos. Se entiende entonces que la variación de las semillas k no tiene un efecto significativo en la distancia de la penetración de la fractura.

A continuación se muestra el código para realizar las pruebas estadísticas:

```

1 #PRUEBA ESTADISTICA
2 library(readxl)
3 library(ggplot2)
4 library(tidyverse)
5 library(ggpubr)
6 library(car)
7
8 ruta_excel = "C:\\Users\\beren\\OneDrive\\Escritorio\\Maestr a\\2do semestre\\5. Simulaci n
  computacional de nanomateriales\\Tareas\\Tarea_4\\datos.xlsx"
9 datos = read_excel(ruta_excel)
10 datos
11
12 datos$Semillas= as.factor(datos$Semillas) #crear vector a partir del dataframe
13
14 diagrama = ggplot(data = datos, aes(x = Semillas, y = Distancia, fill = Semillas)) +
15   geom_boxplot() + theme_bw() + labs(x = "Cantidad de semillas", y = "Distancia Manhattan")
16
17 diagrama
18
19 #Estadísticas descriptivas
20 datos = datos %>%
21   rstatix::reorder_levels(Semillas, order = c("6", "12", "24"))
22
23 datos %>%
24   group_by(Semillas) %>%
25   get_summary_stats(Distancia, type = "mean_sd")

```

```

26
27 #SUPUESTOS PARA ANOVA
28 #1:Outliers
29 datos %>%
30   group_by(Semillas) %>%
31   identify_outliers(Distancia)
32
33 #2:Normalidad con residuales
34 normalidad = lm(Distancia ~ Semillas, data = datos)
35 head(datos)
36 head(normalidad$fitted.values)
37 head(normalidad$residuals)
38
39 grafica1 = ggqqplot(residuals(normalidad))+
40   labs(title = 'Normalidad con residuales')
41 grafica1
42
43 shapiro_test(residuals(normalidad))
44
45 datos %>%
46   group_by(Semillas) %>%
47   shapiro_test(Distancia)
48
49 #3:Homogeneidad de varianza con prueba Levene
50 datos %>%
51   levene_test(Distancia~Semillas)
52
53
54 #PRUEBA ESTADISTICA KRUSKAL WALLIS
55 kruskal.test(Distancia ~ Semillas, data = datos)
56
57 pairwise.wilcox.test(datos$Distancia, datos$Semillas)
58
59 grafica2 = ggline(data = datos, x = "Semillas", y = "Distancia", add = c("mean_se", "jitter"))
60 grafica2

```

Listing 3: Código para hacer la prueba estadística Kruskal Wallis.

3. Conclusión

Con base en los resultados obtenidos de la prueba estadística **Kruskal Wallis** puedo concluir que el hecho de variar la cantidad de semillas en la matriz no influye tanto en la distancia de la penetración de la grieta, sin embargo en el diagrama caja-bigote se puede observar que cuando existen más semillas la distancia puede tener valores más grandes pero no deja de presentar valores pequeños al igual que con pocas semillas.

En general ésta práctica me permitió darme cuenta que me falta mejorar en el entendimiento de los códigos para ser capaz de modificarlos de manera adecuada, también me permitió aplicar otras pruebas estadísticas y analizar su utilidad.

Referencias

- [1] José Antonio: Estadística Aplicada. Kruskal-wallis en rstudio, 2020. URL https://www.youtube.com/watch?v=WEjudFpbCcE&t=996s&ab_channel=Jos%C3%A9Antonio%3AEstad%C3%ADsticaAplicada.
- [2] Elisa Schaeffer. Voronoi diagrams, 2021. URL <https://github.com/satuelisa/Simulation/blob/master/VoronoiDiagrams/fracture.R>.