

Tarea 7

Eduardo Navarro

Octubre 2021

1. Introducción

En esta práctica se realizó un estudio sobre el movimiento hacia los máximos dentro de una función dada y acotada.

2. Desarrollo

Con las instrucciones de la tarea [4] y lo visto en clase [5] se le hicieron modificaciones al código para obtener los máximos añadiendo información al `for` que limita la función, a diversos pasos y sus respectivas réplicas con un `for` para el análisis estadístico de una función dada. Así mismo se creó un gif que muestra el movimiento de los puntos.

Listing 1: Código para la obtención de máximos.

```
library(lattice)
library(sp)
library(viridisLite)
library(reshape2)

datos<- data.frame()

g <- function(x,y) { # modificamos para que sea interesante
  return((10* cos(x^2) - 10 * x + 12 * cos(y^2) -5* y))
}
x <- seq(-1, 6, 0.4)
y <- x
z <- outer(x, y, g)

low <- -1
high <- 6
j <- seq(0.25, 4, 0.25)
repe<- (1:30)
replicas <- 50 #puntitos

for (step in j) {
  for (rep in repe) {

replica <- function(t) {
  curr <- c(runif(1, low, high), runif(1, low, high))
  best <- curr
  for (tiempo in 1:t) {

    delta <- runif(1, 0, step)

    left <- curr +c(-delta,0)
```

```

right <- curr + c(delta,0)
up <- curr + c(0,-delta)
down <- curr + c(0,delta)
puntos<- c(left ,right ,up,down)

for(p in 1:8){
  if(puntos[p] < (-1)){
    puntos[p] <- puntos[p]+6
  }
  if(puntos[p] > 6){
    puntos[p] <- puntos[p]-1
  }
}

ux<-c()
uy<-c()

for(q in 1:8){
  if(q %%2 == 0){
    uy <- c(uy,puntos[q])
  }else{
    ux <- c(ux,puntos[q])
  }
}

ug<- c()

for(r in 1:4){
  ug <- c(ug, g(ux[r], uy[r]) )
}
ppmax <- which.max(ug)
curr <- c(ux[ppmax], uy[ppmax])
if(g(curr[1],curr[2]) > g(best[1],best[2])){

  best <- curr
}
}
return(best)
}

```

Listing 2: Código para la obtención de imágenes.

```

for (pot in 1:40) { #numero de imagenes a repeticiones dadas
  tmax <- pot
  resultados <- foreach(i = 1:replicas, .combine=c) %lopar% replica(tmax)

ux<- c()
uy<- c()

sal<-(2*replicas)

for(q in 1:sal){

```

```

  if(q %%2 == 0){
    uy <- c(uy, resultados[q])
  } else {
    ux <- c(ux, resultados[q])
  }
}

val <- c()
for(r in 1:replicas){
  val <- c(val, g(ux[r], uy[r]))
}

maximo <- which.max(val)
x <- seq(-1, 6, 0.4)
y <- x
z <- outer(x, y, g)

dimnames(z) <- list(x, y)
d <- melt(z)
names(d) <- c("x", "y", "z")

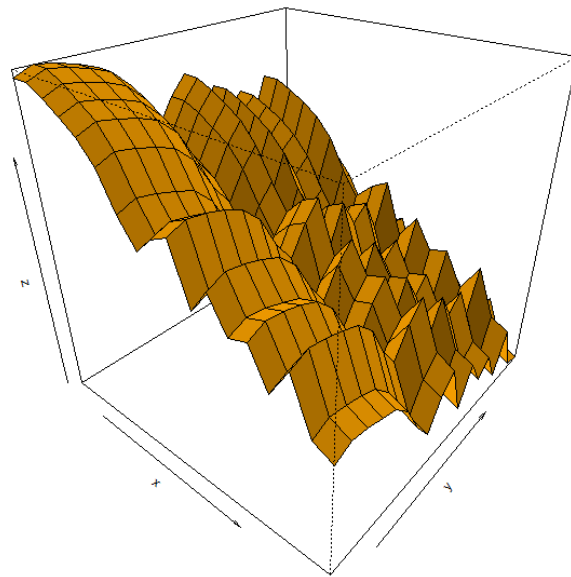
png(paste0("t7_", tmax, ".png", sep=""), width=500, height=500)
plot(levelplot(z ~ x * y, data = d))
trellis.focus("panel", 1, 1, highlight=FALSE)
lpoints(ux, uy, pch=1, col="red", cex=2)
trellis.unfocus()
trellis.focus("panel"[1], 1, 1, highlight=FALSE)
lpoints(ux[maximo], uy[maximo], pch=20, col="green", cex=3)
trellis.unfocus()

graphics.off()
}
ultimo<-min(val)
datos<- rbind(datos, c(step, ultimo))

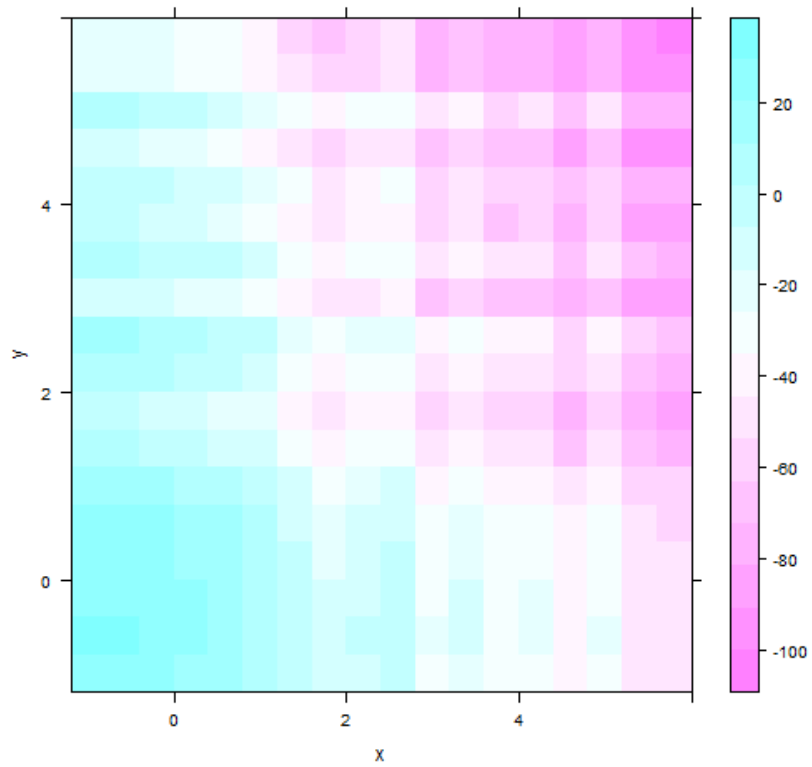
```

Se trabajó sobre el gráfico con una proyección 2D del plano sobre el cual observaremos el movimiento gracias a un gif [2].

Gráfica 1: Gráfica 3D de la función.



Gráfica 2: Gráfica 2D de la función.

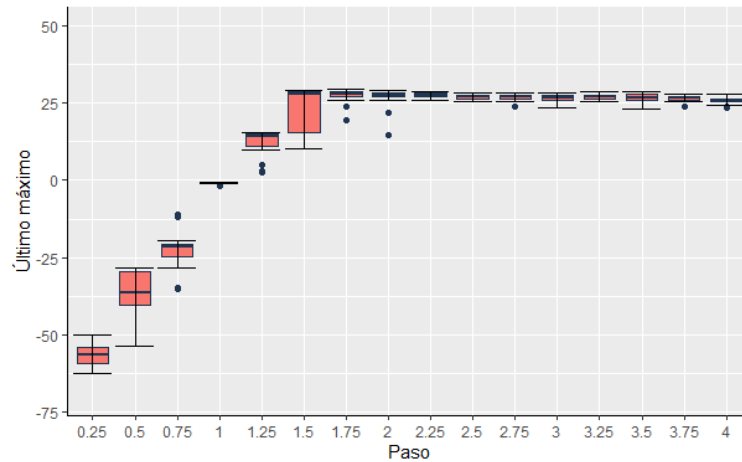


Con esto se generaron los datos de la tabla 1 y se le hicieron las correspondientes pruebas estadísticas.

Tabla 1: Ejemplo de datos obtenidos.

Paso	Distancia
0.25	-56.1624408
0.25	-56.2513706
0.25	-55.7223354
0.25	-62.1704963
0.25	-53.4912933
0.25	-62.2414052
0.25	-56.2131058
0.25	-55.7417835

Gráfica 3: Último Máximo obtenido a partir del tamaño de paso.



Listing 3: Código para la obtención de la gráfica y pruebas estadísticas.

```
library(ggplot2)
library(tidyverse)

names(datos)<-c("paso", "distancia")
datos$paso = as.factor(datos$paso)

ggplot(datos, aes(x=paso , y= distancia , fill= rep)) + # fill=name allow to
  automatically dedicate a color for each group
  geom_boxplot(fill = "#F8766D", colour = "#1F3552")+
  stat_boxplot(geom = "errorbar", width = 0.9)+
  theme(axis.line = element_line(colour = "black", size = 0.25))+
  coord_cartesian(ylim = c(-70, 50))+
  labs(x="Paso", y= "Ultimo_maximo")

datos %>%
  group_by(paso) %>%
  summarise(

    promedio = mean(distancia , na.rm = TRUE),
    desviacion_std = sd(distancia , na.rm = TRUE),
    varianza = sd(distancia , na.rm = TRUE)^2,
    mediana = median(distancia , na.rm = TRUE),
    rango_intercuartil = IQR(distancia , na.rm = TRUE)
```

```
)

tapply(datos$distancia , datos$paso , shapiro.test)

kruskal.test(distancia ~ paso , data=datos)

pairwise.wilcox.test(datos$distancia , datos$paso)
```

De la gráfica 3 podemos observar un máximo efectivo en cuanto al paso del cual después no hay una diferencia notable. Esto se comprueba con las pruebas estadísticas de Shapiro Wilk [6] y Kruskal Wallis [1] además de que con la prueba Wilcox [3] se pueden apreciar las diferencias entre grupos.

Tabla 2: Estadísticas obtenidas de la tabla 1.

Paso	Promedio	Desviación std	Varianza	Mediana	Rango intercuartil
0.25	-56.8	3.5	12.2	-56.2	5.11
0.5	-36.3	6.43	41.3	-36.3	10.8
0.75	-21.2	6.21	38.6	-21.5	3.9
1	-0.966	0.281	0.0792	-0.899	0.221
1.25	12.6	3.6	12.9	14.4	3.95
1.5	23.1	6.96	48.4	27.9	13.2
1.75	27.5	1.89	3.57	28	1.44
2	27	2.7	7.3	27.6	1.54
2.25	27.6	0.659	0.434	27.6	0.992
2.5	26.9	0.793	0.629	27	1.21
2.75	26.7	0.913	0.834	26.8	1.14
3	26.6	1.25	1.56	26.8	1.68
3.25	26.9	0.777	0.604	27	1.31
3.5	26.6	1.41	1.99	26.8	1.86
3.75	26.5	0.875	0.766	26.6	1.06
4	25.9	0.986	0.973	26	0.977

Tabla 3: Resultados de la prueba Shapiro–Wilk para la tabla 1.

Paso	W	P
0.25	0.91671	0.02206
0.5	0.92926	0.04694
0.75	0.88267	0.003249
1	0.73805	$5,87 \times 10^{-6}$
1.25	0.73931	$6,15 \times 10^{-6}$
1.5	0.74202	$6,78 \times 10^{-6}$
1.75	0.70755	$2,01 \times 10^{-6}$
2	0.54997	$1,89 \times 10^{-8}$
2.25	0.94227	0.1047
2.5	0.96461	0.404
2.75	0.94005	0.09125
3	0.9346	0.06513
3.25	0.95696	0.2586
3.5	0.92479	0.03577
3.75	0.95538	0.2351
4	0.95128	0.183

Tabla 4: Resultados de la prueba Kruskal-Wallis para la tabla 1.

H(15)	p-value
349.49	$2,2 \times 10^{-16}$

Tabla 5: Resultados de la prueba por parejas de Wilcoxon para la tabla 1.

	0.25	0.5	0.75	1	1.25	1.5	1.75	2
0.5	$6,7 \times 10^{-14}$	-	-	-	-	-	-	-
0.75	2×10^{-15}	$5,6 \times 10^{-12}$	-	-	-	-	-	-
1	2×10^{-15}	2×10^{-15}	2×10^{-15}	-	-	-	-	-
1.25	2×10^{-15}	2×10^{-15}	2×10^{-15}	2×10^{-15}	-	-	-	-
1.5	2×10^{-15}	2×10^{-15}	2×10^{-15}	2×10^{-15}	$3,2 \times 10^{-7}$	-	-	-
1.75	2×10^{-15}	2×10^{-15}	2×10^{-15}	2×10^{-15}	2×10^{-15}	1	-	-
2	2×10^{-15}	2×10^{-15}	2×10^{-15}	2×10^{-15}	$1,9 \times 10^{-13}$	1	1	-
2.25	2×10^{-15}	2×10^{-15}	2×10^{-15}	2×10^{-15}	2×10^{-15}	1	1	1
2.5	2×10^{-15}	2×10^{-15}	2×10^{-15}	2×10^{-15}	2×10^{-15}	1	0.0533	0.3255
2.75	2×10^{-15}	2×10^{-15}	2×10^{-15}	2×10^{-15}	2×10^{-15}	1	0.0221	0.0759
3	2×10^{-15}	2×10^{-15}	2×10^{-15}	2×10^{-15}	2×10^{-15}	1	0.0259	0.2118
3.25	2×10^{-15}	2×10^{-15}	2×10^{-15}	2×10^{-15}	2×10^{-15}	1	0.0501	0.2643
3.5	2×10^{-15}	2×10^{-15}	2×10^{-15}	2×10^{-15}	2×10^{-15}	1	0.0579	0.2808
3.75	2×10^{-15}	2×10^{-15}	2×10^{-15}	2×10^{-15}	2×10^{-15}	1	0.0021	0.0077
4	2×10^{-15}	2×10^{-15}	2×10^{-15}	2×10^{-15}	2×10^{-15}	1	$1,1 \times 10^{-5}$	$4,3 \times 10^{-5}$

Tabla 6: Resultados de la prueba por parejas de Wilcoxon para la tabla 1 continuación.

	2.25	2.5	2.75	3	3.25	3.5	3.75
0.5	-	-	-	-	-	-	-
0.75	-	-	-	-	-	-	-
1	-	-	-	-	-	-	-
1.25	-	-	-	-	-	-	-
1.5	-	-	-	-	-	-	-
1.75	-	-	-	-	-	-	-
2	-	-	-	-	-	-	-
2.25	-	-	-	-	-	-	-
2.5	0.0517	-	-	-	-	-	-
2.75	0.0031	1	-	-	-	-	-
3	0.0269	1	1	-	-	-	-
3.25	0.0409	1	1	1	-	-	-
3.5	0.0629	1	1	1	1	-	-
3.75	$2,7 \times 10^{-5}$	1	1	1	1	1	-
4	$2,7 \times 10^{-8}$	0.0033	0.0333	0.2369	0.0043	0.3958	0.3922

3. Conclusiones

Con los resultados obtenidos podemos concluir que el paso influye en el máximo a obtener debido a que a mayor paso más distancia se recorre y esto le permite alcanzar sin tanta dificultad el punto mas alto. Con el gif se pudo observar el movimiento de los puntos alrededor de la superficie de la gráfica. En base a las pruebas estadísticas podemos concluir que a partir del paso 1.75 se obtiene la mejor velocidad para alcanzar el máximo ya que no hubo mejoras notables después de este paso.

Referencias

- [1] José Antonio: Estadística Aplicada. Kruskal-wallis en RStudio, 2020. URL <https://www.youtube.com/watch?v=WEjudFpbCcE>.
- [2] eduartilon. `graficagif.gif`, 2021. URL <https://github.com/eduartilon/Simulacion/blob/main/tarea7/graficagif.gif>.
- [3] Thomas Pernet. 9 Non Parametric tests, 2020. URL https://bookdown.org/thomas_pernet/Tuto/non-parametric-tests.html.
- [4] Elisa Schaeffer. Práctica 7: búsqueda local. <https://elisa.dyndns-web.com/teaching/comp/par/p7.html/>, 2021. [Online; accessed 11-October-2021].
- [5] Elisa Schaeffer. Simulación: búsqueda local (p7), 2021. URL <https://www.twitch.tv/videos/1169298413>.
- [6] El Tío Estadístico. Cómo hacer la Prueba de Normalidad en R, 2020. URL <https://www.youtube.com/watch?v=LAzSb6jCFbs>.