Tarea 1

Pablo Gracia Galeana, Cesar Gonzalez Macedo, Miguel Ángel Fuentes Borboa, Roberto Antonio Yglesias Galeana

Problema 1

Por demostrar:$Z_i \equiv \left[a^i Z_0 + c\left(\frac{a^i-1}{a-1}\right)\right] \bmod m$
Para n = 1 tenemos: $Z_1 \equiv (aZ_0 + c) \bmod m = \left[a^1 Z_0 + c\left(\frac{a^1-1}{a-1}\right)\right] \bmod m$.
Como se cumple para $n = 1$, queda por demostrar que si es cierto para cualquier n, entonces es cierto para $n+1$.
Sabemos que: $Z_n = \left[a^n Z_0 + c\left(\frac{a^n-1}{a-1}\right)\right] \bmod m$

$$\Rightarrow \exists k \in \mathbb{Z} \text{ tal que } km + Z_n = a^n Z_0 + c\left(\frac{a^n-1}{a-1}\right) Z_{n+1} \equiv (aZ_n + c) \bmod m$$

$$\Rightarrow Z_{n+1} \equiv \left[a\left(a^n Z_0 + c\left(\frac{a^n-1}{a-1}\right) - km\right) + c\right] \bmod m$$

$$\Rightarrow Z_{n+1} \equiv \left[a^{n+1} Z_0 + c\left(\frac{a^{n+1}-a}{a-1}\right) - akm + c\right] \bmod m$$

$$\Rightarrow Z_{n+1} \equiv \left[a^{n+1} Z_0 + c\left(\frac{a^{n+1}-1}{a-1}\right) - akm\right] \bmod m \Rightarrow Z_{n+1} \equiv \left[a^{n+1} Z_0 + c\left(\frac{a^{n+1}-1}{a-1}\right)\right] \bmod m$$

$$\Rightarrow n+1 \text{ Satisface}$$

$$\therefore \forall n \in \mathbb{N}\, Z_n \equiv \left[a^n Z_0 + c\left(\frac{a^n-1}{a-1}\right)\right] \bmod m$$

Problema 2

```
z<-NULL
seed <- 120395
a<-5
c<-3
m<-31
z[1]<-((a*seed+c) %% m)
for(i in 2:10){
z[i]<-((a*z[i-1]+c) %% m)
}
z
```

```
##  [1] 20 10 22 20 10 22 20 10 22 20
```

El periodo del GLC es 3

Problema 3

Hay que comprobar que cada GLC cumpla con las condiciones del teorema de periodo completo.

- $Z_i \equiv [13Z_{i-1} + 13] \bmod 16$. Con $a = 13$, $c = 13$, $m = 16$
  1. $c = 13, m = 16$ son primos relativos.
  2. El único primo que divide a 16 es 2, que también divide a $13 - 1 = 12$
  3. 4 divide a 16 y a 12
     $\therefore$ tiene periodo completo.
- $Z_i \equiv [12Z_{i-1} + 13] \bmod 16$. Con $a = 12$, $c = 13$, $m = 16$
  1. $c = 13, m = 16$ son primos relativos.
  2. 2 es un primo que divide a 16 pero no divide a 12-1=11.
     $\therefore$ No tiene periodo completo.
- $Z_i \equiv [25437Z_{i-1} + 35421] \bmod 2^{10}$
  1. El máximo común divisor de $c = 35421$ y $m = 2^{10}$ es 1. Entonces son primos relativos.
  2. El único número primo que divide a 2^{10} es 2, y este divide a 25437-1=25436
  3. 4 divide a 2^{10} y a 25437-1=25436
     $\therefore$ Tiene periodo completo.
- $Z_i \equiv [Z_{i-1} + 12] \bmod 13$
  1. $c = 12, m = 13$ El único primo que divide a 13 es 13, que también divide a 0
  2. 4 no divide a 13.
     $\therefore$ No tiene periodo completo.
- GLC con $a = 2,814,749,767,109, c = 59,482,661,568,307, m = 2^{48}$. Se tiene que $m = 2^{48} > 4$ y c es impar y $a \equiv 1 \bmod m$. $\therefore$ El GLC tiene periodo completo.

Problema 4

Por demostrar el promedio de las U_i de un ciclo completo de un GLC de periodo completo es: $\frac{1}{2} - \frac{1}{2m}$.

$$\sum_{i=1}^{m} \frac{U_i}{m} = \frac{1}{m} \sum_{i=1}^{m} U_i = \frac{1}{m} \sum_{i=0}^{m-1} \frac{Z_i}{m} = \frac{1}{m^2} \sum_{i=0}^{m} Z_i = \frac{1}{m^2} \frac{m(m-1)}{2} = \frac{1}{m^2} \frac{m^2 - m)}{2} = \frac{1}{2} - \frac{1}{2m}$$

Problema 5

Función de Generadores lineales congruenciales

```
GLC <- function(a, c, m, s){

z[1]<-((a*s+c) %% m)
for(i in 2:50){
z[i]<-((a*z[i-1]+c) %% m)
}
```

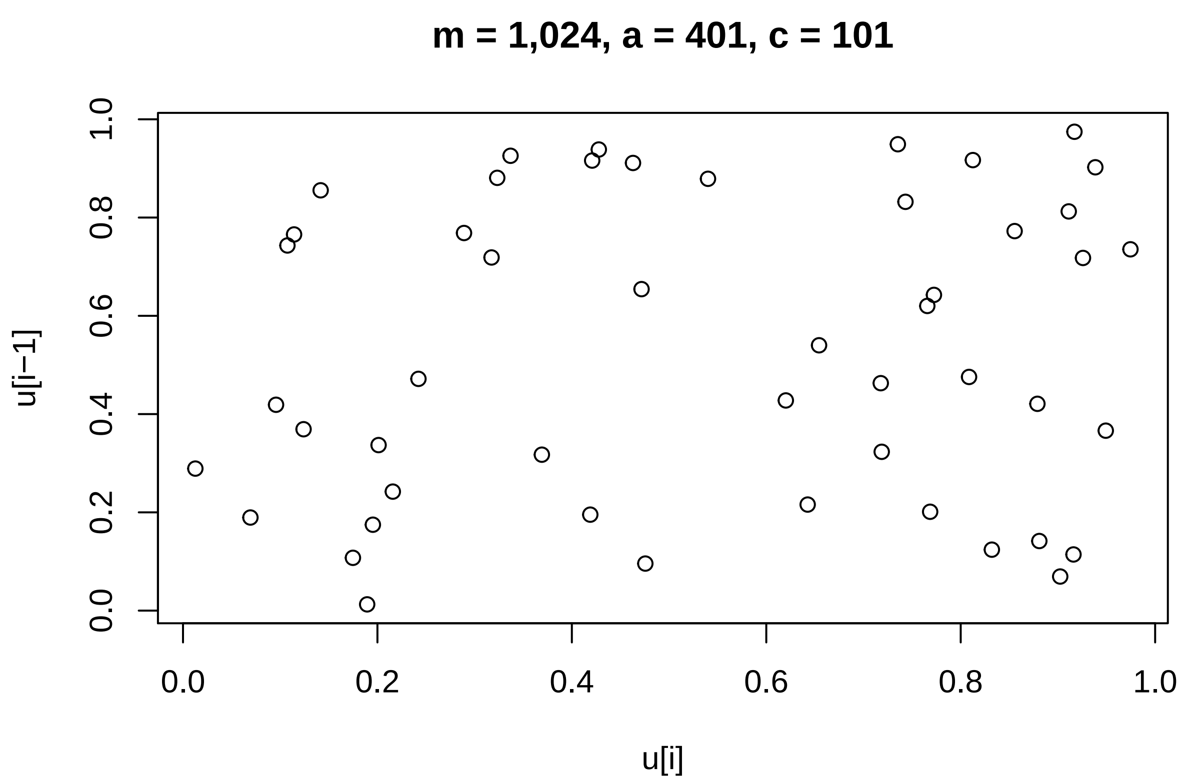
```
return(z/m)
}
```

La muestra con los parametros $m = 1,024, a = 401, c = 101$ es

```
z<-GLC(a=401, c=101, m=1024, s=280306 )
z
```

```
##  [1] 0.36621094 0.94921875 0.73535156 0.97460938 0.91699219 0.81250000
##  [7] 0.91113281 0.46289062 0.71777344 0.92578125 0.33691406 0.20117188
## [13] 0.76855469 0.28906250 0.01269531 0.18945312 0.06933594 0.90234375
## [19] 0.93847656 0.42773438 0.62011719 0.76562500 0.11425781 0.91601562
## [25] 0.42089844 0.87890625 0.54003906 0.65429688 0.47167969 0.24218750
## [31] 0.21582031 0.64257812 0.77246094 0.85546875 0.14160156 0.88085938
## [37] 0.32324219 0.71875000 0.31738281 0.36914062 0.12402344 0.83203125
## [43] 0.74316406 0.10742188 0.17480469 0.19531250 0.41894531 0.09570312
## [49] 0.47558594 0.80859375
```

```
plot(z[2:50],z[1:49],xlab="u[i]",ylab="u[i-1]", main =
"m = 1,024, a = 401, c = 101")
```

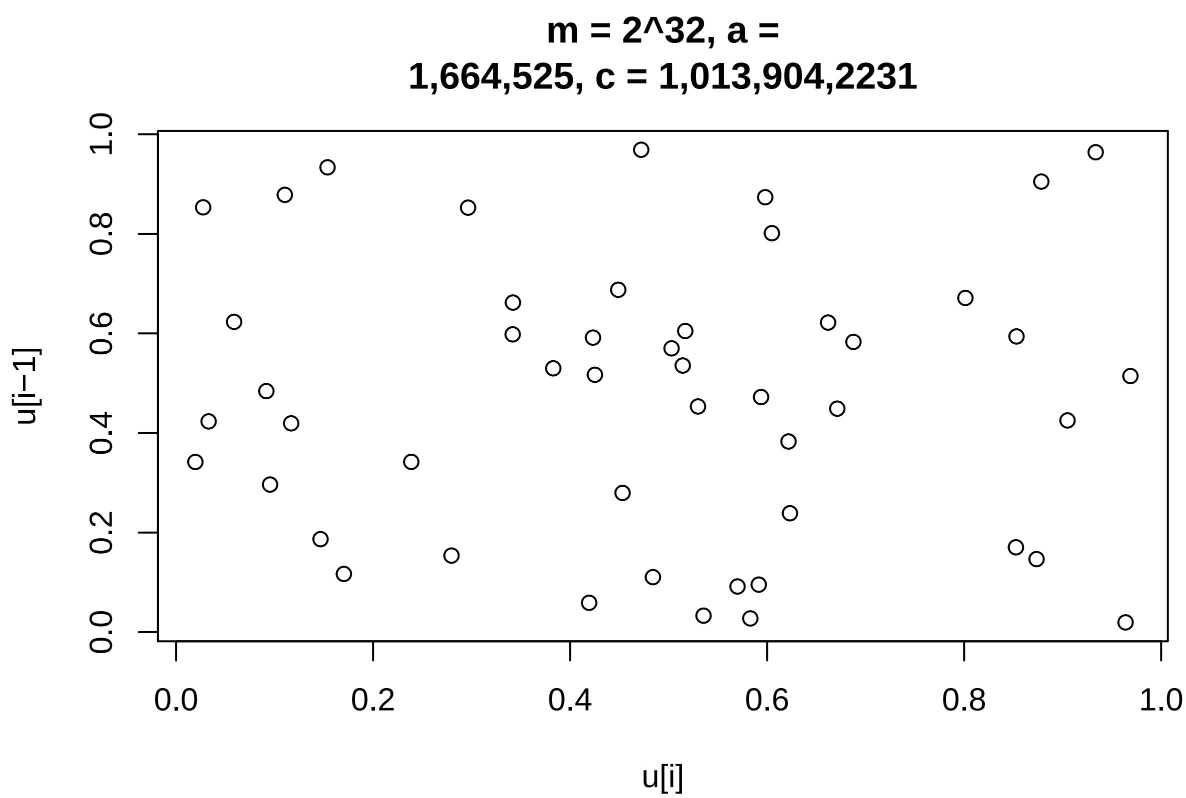


La muestra con los parametros $m = 2^{32}, a = 1,664,525, c = 1,013,904,223$ es

```
z<-GLC(a=41664525, c=1664525, m=2^32, s=280306 )
z
```

```
##  [1] 0.18671419 0.14663949 0.87352837 0.59806239 0.34172693 0.01955877
##  [7] 0.96388011 0.93355038 0.15375441 0.27959033 0.45325342 0.52982803
## [13] 0.38290298 0.62174197 0.66189053 0.34196389 0.23876486 0.62319064
## [19] 0.05889714 0.41930967 0.11683376 0.17034950 0.85258054 0.29642849
## [25] 0.09541296 0.59152505 0.42326075 0.03305072 0.53545462 0.51434864
## [31] 0.96884132 0.47215579 0.59385712 0.85315952 0.02749643 0.58293876
## [37] 0.68761171 0.44886132 0.67121703 0.80128221 0.60486144 0.51690656
## [43] 0.42518996 0.90493229 0.87833785 0.11042993 0.48407232 0.09162172
## [49] 0.56994896 0.50300095
```

```
plot(z[2:50],z[1:49],xlab="u[i]",ylab="u[i-1]", main =
"m = 2^32, a =
1,664,525, c = 1,013,904,2231")
```



Problema 6

Por demostrar: la parte fraccional de $U_1 + U_2 + \ldots + U_k$, donde cada $U_i \sim U(0,1)$ con $i = 1, \ldots, k$ se distribuye tambien uniforme en $(0,1)$.

Demostración por inducción

Para $n = 2$
Sean $U_1, U_2 \sim U(0,1)$
$S_2 = U_1 + U_2$

$$f_{S_2}(s) = \begin{cases} v & 0 \le s \le 2 \\ 2-s & 1 \le s \le 2 \end{cases}$$

Sea $S_2' = S_1 - \lfloor S_2 \rfloor$. Tenemos dos casos:

- Si $0 \le S_2 \le 1 \Rightarrow \{S_2\} = S_2$
  $\therefore P(S_2' \le s) = P(S_2 \le s)$
- Si $1 \le S_2 \le 2 \Rightarrow \{S_2\} = S_2 - 1$
  $\therefore P(S_2' \le 2) = P(S_2 - 1 \le s) = P(S_2 \le s + 1)$
  $\therefore F_{S_2'}(s) = P(\{S_2\} \le s) = \int_0^s f_{S_2}(u)du + \int_1^{s+1} f_{S_2}(u)du = S$

Supongamos que es cierto para cualquier k. Por demostrar que si es cierto para cualquier k, entonces es cierto para $k+1$.

$$U_1 + \ldots + U_k = (U_1 + \ldots + U_{k-1}) + U_k$$

$\therefore$ la parte fraccional de $U_1 + U_2 + \ldots + U_k$ se distribuye uniforme en $(0,1)$.

Problema 7

Un generador de Fibonacci obtiene X_{n+1} a partir de X_n y X_{n-1} de la siguiente forma:

$$X_{i+1} \equiv (X_i + X_{i-1}) \bmod m$$

donde X_0 y X_1 están especificados. Supongan que $m = 5$. Solo dos ciclos son posibles. Encontrarlos, así como su respectivo periodo.

Solución. Para el valor $m = 5$ hay $\binom{52}{25}$ posibles valores iniciales. Obteniendo la secuencia para cada posible valor inicial, podemos construir la siguiente matriz que tiene por columnas cada una de las combinaciones de valores iniciales.

```
fibonacci <- function (m=5,x0,x1){(x0+x1) %% m}
A <- as.matrix(expand.grid(0:4,0:4)); names(A) <- NULL
M <- NULL
for(j in 1:25){
  x <- as.vector(A[j,])
  for(i in 3:100) {
    x[i] <- fibonacci(x0=x[i-2],x1=x[i-1])
  }
  M <- cbind(M,x)
}
head(M, n=50)
```

```
##       x x x x x x x x x x x x x x x x x x x x x x x x x
##  [1,] 0 1 2 3 4 0 1 2 3 4 0 1 2 3 4 0 1 2 3 4 0 1 2 3 4
##  [2,] 0 0 0 0 0 1 1 1 1 1 2 2 2 2 2 3 3 3 3 3 4 4 4 4 4
##  [3,] 0 1 2 3 4 1 2 3 4 0 2 3 4 0 1 3 4 0 1 2 4 0 1 2 3
##  [4,] 0 1 2 3 4 2 3 4 0 1 4 0 1 2 3 1 2 3 4 0 3 4 0 1 2
##  [5,] 0 2 4 1 3 3 0 2 4 1 1 3 0 2 4 4 1 3 0 2 2 4 1 3 0
##  [6,] 0 3 1 4 2 0 3 1 4 2 0 3 1 4 2 0 3 1 4 2 0 3 1 4 2
```

```
##  [7,] 0 0 0 0 0 3 3 3 3 3 1 1 1 1 1 4 4 4 4 4 2 2 2 2 2
##  [8,] 0 3 1 4 2 3 1 4 2 0 1 4 2 0 3 4 2 0 3 1 2 0 3 1 4
##  [9,] 0 3 1 4 2 1 4 2 0 3 2 0 3 1 4 3 1 4 2 0 4 2 0 3 1
## [10,] 0 1 2 3 4 4 0 1 2 3 3 4 0 1 2 2 3 4 0 1 1 2 3 4 0
## [11,] 0 4 3 2 1 0 4 3 2 1 0 4 3 2 1 0 4 3 2 1 0 4 3 2 1
## [12,] 0 0 0 0 0 4 4 4 4 4 3 3 3 3 3 2 2 2 2 2 1 1 1 1 1
## [13,] 0 4 3 2 1 4 3 2 1 0 3 2 1 0 4 2 1 0 4 3 1 0 4 3 2
## [14,] 0 4 3 2 1 3 2 1 0 4 1 0 4 3 2 4 3 2 1 0 2 1 0 4 3
## [15,] 0 3 1 4 2 2 0 3 1 4 4 2 0 3 1 1 4 2 0 3 3 1 4 2 0
## [16,] 0 2 4 1 3 0 2 4 1 3 0 2 4 1 3 0 2 4 1 3 0 2 4 1 3
## [17,] 0 0 0 0 0 2 2 2 2 2 4 4 4 4 4 1 1 1 1 1 3 3 3 3 3
## [18,] 0 2 4 1 3 2 4 1 3 0 4 1 3 0 2 1 3 0 2 4 3 0 2 4 1
## [19,] 0 2 4 1 3 4 1 3 0 2 3 0 2 4 1 2 4 1 3 0 1 3 0 2 4
## [20,] 0 4 3 2 1 1 0 4 3 2 2 1 0 4 3 3 2 1 0 4 4 3 2 1 0
## [21,] 0 1 2 3 4 0 1 2 3 4 0 1 2 3 4 0 1 2 3 4 0 1 2 3 4
## [22,] 0 0 0 0 0 1 1 1 1 1 2 2 2 2 2 3 3 3 3 3 4 4 4 4 4
## [23,] 0 1 2 3 4 1 2 3 4 0 2 3 4 0 1 3 4 0 1 2 4 0 1 2 3
## [24,] 0 1 2 3 4 2 3 4 0 1 4 0 1 2 3 1 2 3 4 0 3 4 0 1 2
## [25,] 0 2 4 1 3 3 0 2 4 1 1 3 0 2 4 4 1 3 0 2 2 4 1 3 0
## [26,] 0 3 1 4 2 0 3 1 4 2 0 3 1 4 2 0 3 1 4 2 0 3 1 4 2
## [27,] 0 0 0 0 0 3 3 3 3 3 1 1 1 1 1 4 4 4 4 4 2 2 2 2 2
## [28,] 0 3 1 4 2 3 1 4 2 0 1 4 2 0 3 4 2 0 3 1 2 0 3 1 4
## [29,] 0 3 1 4 2 1 4 2 0 3 2 0 3 1 4 3 1 4 2 0 4 2 0 3 1
## [30,] 0 1 2 3 4 4 0 1 2 3 3 4 0 1 2 2 3 4 0 1 1 2 3 4 0
## [31,] 0 4 3 2 1 0 4 3 2 1 0 4 3 2 1 0 4 3 2 1 0 4 3 2 1
## [32,] 0 0 0 0 0 4 4 4 4 4 3 3 3 3 3 2 2 2 2 2 1 1 1 1 1
## [33,] 0 4 3 2 1 4 3 2 1 0 3 2 1 0 4 2 1 0 4 3 1 0 4 3 2
## [34,] 0 4 3 2 1 3 2 1 0 4 1 0 4 3 2 4 3 2 1 0 2 1 0 4 3
## [35,] 0 3 1 4 2 2 0 3 1 4 4 2 0 3 1 1 4 2 0 3 3 1 4 2 0
## [36,] 0 2 4 1 3 0 2 4 1 3 0 2 4 1 3 0 2 4 1 3 0 2 4 1 3
## [37,] 0 0 0 0 0 2 2 2 2 2 4 4 4 4 4 1 1 1 1 1 3 3 3 3 3
## [38,] 0 2 4 1 3 2 4 1 3 0 4 1 3 0 2 1 3 0 2 4 3 0 2 4 1
## [39,] 0 2 4 1 3 4 1 3 0 2 3 0 2 4 1 2 4 1 3 0 1 3 0 2 4
## [40,] 0 4 3 2 1 1 0 4 3 2 2 1 0 4 3 3 2 1 0 4 4 3 2 1 0
## [41,] 0 1 2 3 4 0 1 2 3 4 0 1 2 3 4 0 1 2 3 4 0 1 2 3 4
## [42,] 0 0 0 0 0 1 1 1 1 1 2 2 2 2 2 3 3 3 3 3 4 4 4 4 4
## [43,] 0 1 2 3 4 1 2 3 4 0 2 3 4 0 1 3 4 0 1 2 4 0 1 2 3
## [44,] 0 1 2 3 4 2 3 4 0 1 4 0 1 2 3 1 2 3 4 0 3 4 0 1 2
## [45,] 0 2 4 1 3 3 0 2 4 1 1 3 0 2 4 4 1 3 0 2 2 4 1 3 0
## [46,] 0 3 1 4 2 0 3 1 4 2 0 3 1 4 2 0 3 1 4 2 0 3 1 4 2
## [47,] 0 0 0 0 0 3 3 3 3 3 1 1 1 1 1 4 4 4 4 4 2 2 2 2 2
## [48,] 0 3 1 4 2 3 1 4 2 0 1 4 2 0 3 4 2 0 3 1 2 0 3 1 4
## [49,] 0 3 1 4 2 1 4 2 0 3 2 0 3 1 4 3 1 4 2 0 4 2 0 3 1
## [50,] 0 1 2 3 4 4 0 1 2 3 3 4 0 1 2 2 3 4 0 1 1 2 3 4 0
```

Problema 8

Generador lineal congruencial de numeros aleatorios

```
lgc_10m <- function(m = 2^31-1, a = 7^5,z0 = 1){
  z <- z0
  i <- 1
  repeat{
```

```
    i <- i+1
    z[i] <- (a*z[i-1]) %% m
    if (i>9999) break
  }
  return(z/m)
}
```

Creo el vector de 10,000 numeros aleatorios

```
z <- lgc_10m()
head(z, n = 50)
```

```
##  [1] 4.656613e-10 7.826369e-06 1.315378e-01 7.556053e-01 4.586501e-01
##  [6] 5.327672e-01 2.189592e-01 4.704462e-02 6.788647e-01 6.792964e-01
## [11] 9.346929e-01 3.835021e-01 5.194164e-01 8.309653e-01 3.457211e-02
## [16] 5.346164e-02 5.297002e-01 6.711494e-01 7.698186e-03 3.834157e-01
## [21] 6.684224e-02 4.174860e-01 6.867727e-01 5.889766e-01 9.304365e-01
## [26] 8.461669e-01 5.269288e-01 9.196489e-02 6.539190e-01 4.159994e-01
## [31] 7.011906e-01 9.103208e-01 7.621980e-01 2.624530e-01 4.746451e-02
## [36] 7.360819e-01 3.282342e-01 6.326386e-01 7.564105e-01 9.910374e-01
## [41] 3.653387e-01 2.470389e-01 9.825503e-01 7.226604e-01 7.533558e-01
## [46] 6.515186e-01 7.268588e-02 6.316347e-01 8.847071e-01 2.727100e-01
```

Prueba Chi-sq

```
prueba.chisq.uniforme <- function(x,k=ceiling(length(x)/5)){
  n <- length(x)
  part <- seq(0,1,length=k+1) #particion
  z <- hist(x,breaks = part, plot = F)$counts
  ch <- (k/n)*sum((z-n/k)^2) #estadistica chi
  pval <- pchisq(ch,k-1,lower.tail = F)
  return(list(part=part,freqs = z, estadistica = ch, pval = pval))
}
```

Aplico la prueba con 10 celdas

```
prueba.chisq.uniforme(z,k=10)
```

```
## $part
##  [1] 0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0
##
## $freqs
##  [1]  994 1007  998  958 1000 1049  989  963 1026 1016
##
## $estadistica
## [1] 6.676
##
## $pval
## [1] 0.6708111
```

Si pasa la prueba ya que el p-value es de 0.67.

Prueba de Rachas

Cuenta los cambios de signo

```
nrachas <- function(x){
  n <- length(x)
  signo <- x[-1] - x[-n]
  s <- ifelse(signo<0,-1,1)
  R <- 1 + sum(s[-1] != s[-(n-1)])
  return(R)
}
nrachas(z)
```

```
## [1] 6609
```

```
y <- (nrachas(z)-(2*length(z)-1)/3)/sqrt((16*nrachas(z)-29)/90);
y
```

```
## [1] -1.672862
```

```
pnorm(y,0.025)
```

```
## [1] 0.0447669
```

No pasa la prueba ya que el p-value es 0.045.

Problema 9

Metodo de cuadrado de medio de Von Neumann

Ejemplo función cuadrado medio con semilla = 93

```
##  [1] 64  9  8  6  3  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
```

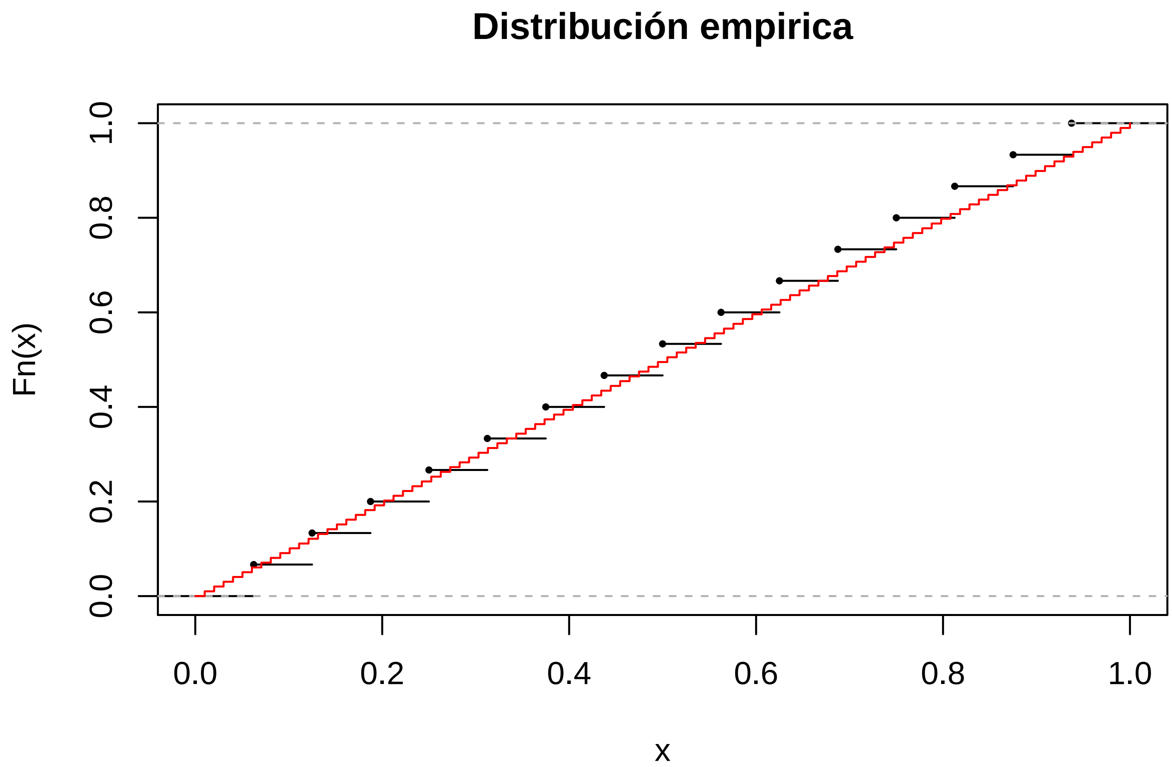
Ultimos 6 ciclos

```
##       [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13]
## [94,]   83   88   74   47   20   40   60   60   60    60    60    60    60
## [95,]    2    0    0    0    0    0    0    0    0     0     0     0     0
## [96,]   21   44   93   64    9    8    6    3    0     0     0     0     0
## [97,]   40   60   60   60   60   60   60   60   60    60    60    60    60
## [98,]   60   60   60   60   60   60   60   60   60    60    60    60    60
## [99,]   80   40   60   60   60   60   60   60   60    60    60    60    60
##       [,14] [,15] [,16] [,17] [,18] [,19] [,20]
## [94,]    60    60    60    60    60    60    60
## [95,]     0     0     0     0     0     0     0
## [96,]     0     0     0     0     0     0     0
## [97,]    60    60    60    60    60    60    60
## [98,]    60    60    60    60    60    60    60
## [99,]    60    60    60    60    60    60    60
```

Problema 10

Generador por congruencias

```
## Empirical CDF:     15 unique values with summary
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0625  0.2812  0.5000  0.5000  0.7188  0.9375
```




```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  gen
## D = 0.0625, p-value = 1
## alternative hypothesis: two-sided
```

Problema 11

Obtencion de los datos

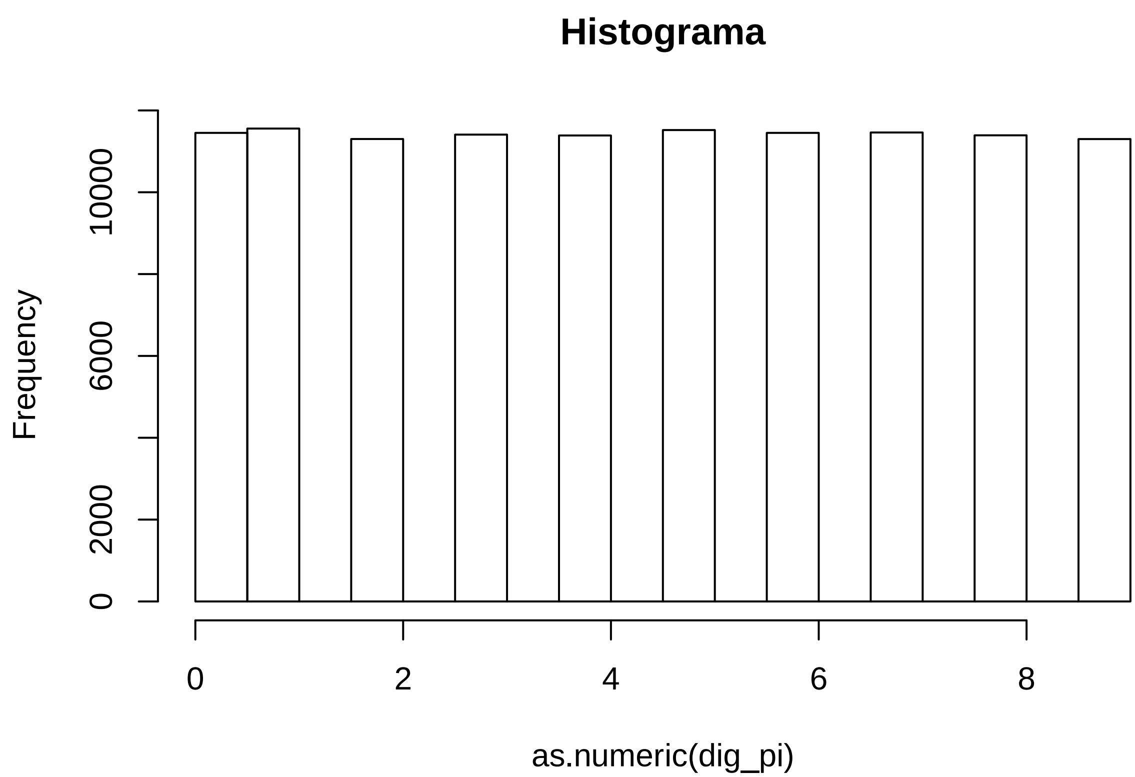
```
dig_pi
```

```
## [1] "14159265358979323846264338327950288419716939937510582097494459230781640628620899862803482534211
```

Creacion del vector de digitos de pi

```r
dig_pi <- as.numeric(unlist(strsplit(dig_pi, ""))[-(1:2)])
head(dig_pi, n=100)
```

```
##   [1] 1 5 9 2 6 5 3 5 8 9 7 9 3 2 3 8 4 6 2 6 4 3 3 8 3 2 7 9 5 0 2 8 8 4 1
##  [36] 9 7 1 6 9 3 9 9 3 7 5 1 0 5 8 2 0 9 7 4 9 4 4 5 9 2 3 0 7 8 1 6 4 0 6
##  [71] 2 8 6 2 0 8 9 9 8 6 2 8 0 3 4 8 2 5 3 4 2 1 1 7 0 6 7 9 8 2
```



Prueba de gaps

Calcula todos las longitudes de gaps de la serie x

```r
gaps <- function(x){
  lgaps <- NULL
  for (i in 0:9){
    pos <- which(x==i)
    l <- diff(pos)
    lgaps <- c(lgaps,l-1)
  }
  L <- table(lgaps)
  return(L)
}
l <- gaps(dig_pi)
cumsum(l/sum(l))
```

```
##          0          1          2          3          4          5
## 0.09932586 0.18976537 0.27073192 0.34580634 0.41216074 0.47033794
```

```
##          6          7          8          9         10         11
## 0.52382245 0.57180879 0.61433199 0.65206619 0.68681492 0.71892838
##         12         13         14         15         16         17
## 0.74653301 0.77197514 0.79490457 0.81477850 0.83352303 0.85014884
##         18         19         20         21         22         23
## 0.86497111 0.87828752 0.89028191 0.90098932 0.91106636 0.92002276
##         24         25         26         27         28         29
## 0.92791980 0.93504640 0.94154264 0.94723341 0.95227631 0.95718788
##         30         31         32         33         34         35
## 0.96114516 0.96490107 0.96846437 0.97165995 0.97439153 0.97685169
##         36         37         38         39         40         41
## 0.97885659 0.98099282 0.98285764 0.98452985 0.98599195 0.98754159
##         42         43         44         45         46         47
## 0.98882858 0.98995798 0.99082472 0.99162143 0.99247943 0.99327613
##         48         49         50         51         52         53
## 0.99398529 0.99457188 0.99504465 0.99557871 0.99592891 0.99627911
##         54         55         56         57         58         59
## 0.99669935 0.99704080 0.99731220 0.99765365 0.99789879 0.99817020
##         60         61         62         63         64         65
## 0.99833654 0.99852040 0.99866048 0.99878305 0.99891438 0.99902819
##         66         67         68         69         70         71
## 0.99915952 0.99922956 0.99932586 0.99940466 0.99952723 0.99956225
##         72         73         74         75         76         77
## 0.99962353 0.99965855 0.99968482 0.99973735 0.99976361 0.99977237
##         78         79         80         81         82         83
## 0.99979863 0.99983365 0.99985116 0.99987743 0.99989494 0.99990369
##         85         86         89         90         92         93
## 0.99991245 0.99992120 0.99992996 0.99993871 0.99994747 0.99996498
##         96        105        106        117
## 0.99997373 0.99998249 0.99999124 1.00000000
```

Cálculo de las frecuencias teóricas a través del uso de una distribucion geometrica

```r
pgeom(as.numeric(names(l)),prob=0.1)
```

```
##  [1] 0.1000000 0.1900000 0.2710000 0.3439000 0.4095100 0.4685590 0.5217031
##  [8] 0.5695328 0.6125795 0.6513216 0.6861894 0.7175705 0.7458134 0.7712321
## [15] 0.7941089 0.8146980 0.8332282 0.8499054 0.8649148 0.8784233 0.8905810
## [22] 0.9015229 0.9113706 0.9202336 0.9282102 0.9353892 0.9418503 0.9476652
## [29] 0.9528987 0.9576088 0.9618480 0.9656632 0.9690968 0.9721872 0.9749684
## [36] 0.9774716 0.9797244 0.9817520 0.9835768 0.9852191 0.9866972 0.9880275
## [43] 0.9892247 0.9903023 0.9912720 0.9921448 0.9929303 0.9936373 0.9942736
## [50] 0.9948462 0.9953616 0.9958254 0.9962429 0.9966186 0.9969567 0.9972611
## [57] 0.9975350 0.9977815 0.9980033 0.9982030 0.9983827 0.9985444 0.9986900
## [64] 0.9988210 0.9989389 0.9990450 0.9991405 0.9992264 0.9993038 0.9993734
## [71] 0.9994361 0.9994925 0.9995432 0.9995889 0.9996300 0.9996670 0.9997003
## [78] 0.9997303 0.9997573 0.9997815 0.9998034 0.9998230 0.9998407 0.9998567
## [85] 0.9998839 0.9998955 0.9999238 0.9999314 0.9999445 0.9999500 0.9999636
## [92] 0.9999859 0.9999873 0.9999960
```

```r
D <- max(abs(cumsum(l/sum(l))-pgeom(as.numeric(names(l)),prob=0.1)))
pval <- 2*exp(-2*sum(l)*D^2)
pval
```

```
## [1] 0.4017313
```

Pasa la prueba ya que el p-value es de 0.67

Problema 12

Probabilidad Conjunta

.	1 Cargado	2	3	4	5	6
1	0.0408163	0.0204082	0.0204082	0.0204082	0.0204082	0.0204082
2	0.0408163	0.0204082	0.0204082	0.0204082	0.0204082	0.0204082
3	0.0408163	0.0204082	0.0204082	0.0204082	0.0204082	0.0204082
4	0.0408163	0.0204082	0.0204082	0.0204082	0.0204082	0.0204082
5	0.0408163	0.0204082	0.0204082	0.0204082	0.0204082	0.0204082
6 Cargado	0.0816327	0.0408163	0.0408163	0.0408163	0.0408163	0.0408163

Se obtiene la probabilidad P_s donde s es la suma de los dados.

s	P_s
2	0.0408163
3	0.0612245
4	0.0816327
5	0.1020408
6	0.1224490
7	0.1836735
8	0.1224490
9	0.1020408
10	0.0816327
11	0.0612245
12	0.0408163

Problema 13

```r
x<-c(2,6,10,16,18,32,20,13,16,9,2)
p<-c(1/36,2/36,3/36,4/36,5/36,6/36,5/36,4/36,3/36,2/36,1/36)
chisq.test(x=x, p =p  )
```

```
##
##  Chi-squared test for given probabilities
##
## data:  x
## X-squared = 7.7208, df = 10, p-value = 0.6561
```

La prueba nos dice que aceptemos la hipótesis de que son dados honestos. Una posible explicación es que a pesar de que los dados están cargados, las frecuencias observadas si siguen una distribución triangular.