

Tp_probabi_informe

Klimkowski Victoria (1390/21) y De Erausquin Carla (126/18)

2024-04-05

```
set.seed(18)
```

Primera simplificación

Suponga que las figuritas se compran individualmente.

1) Simular el número de una figurita elegida al azar si el álbum es de 6 figuritas.

Tomamos 1 muestra al azar entre el 1 y el 6 (incluidos).

```
elegir_figurita <- function() {  
  figurita <- sample(1:6, 1, replace=TRUE)  
  return(figurita)  
}  
  
print(paste("tu figurita es:", elegir_figurita()))  
  
## [1] "tu figurita es: 3"
```

2) Simular el llenado de un álbum de 6 figuritas e indicar cuántas figuritas se debieron comprar para completarlo.

El vector `album_completo` son las figuritas del 1 al 6. Luego, hacemos un `while` que va eligiendo al azar una figurita con la función anterior y las vamos guardando en el vector `tus_figuritas` hasta que todos los elementos de éste vector estén en `album_completo` o, análogamente, mientras que haya elementos en `album_completo` que no estén en `tus_figuritas`. Esto lo hacemos mientras la longitud del vector de la diferencia sea positivo. La función devuelve las figuritas que fueron saliendo y contamos cuántas son.

```
llenar_album <- function() {  
  album_completo <- c(1:6)  
  tus_figuritas <- c()  
  
  while (length(setdiff(album_completo, tus_figuritas)) > 0) {  
    figurita_actual <- elegir_figurita()  
    tus_figuritas <- c(tus_figuritas, figurita_actual)  
  }  
  
  return(tus_figuritas)  
}  
  
print(paste("Completaste tu álbum en :", length(llenar_album()), "intentos"))  
  
## [1] "Completaste tu álbum en : 9 intentos"
```

3) Implementar una función `cuantasFigs(figusTotal)` que, dado el tamaño del álbum (`figusTotal`), simule su llenado y devuelva la cantidad de figuritas que se debieron adquirir para completarlo

Análogo a la función anterior pero donde estaba el "6" se reemplaza por una variable dada por parámetro.

```
cuantasFigs <- function(figusTotal) {  
  album_completo <- c(1:figusTotal)  
  tus_figuritas <- c()  
  
  while (length(setdiff(album_completo, tus_figuritas)) > 0) {  
    figurita_actual <- sample(1:figusTotal, 1, replace=TRUE)  
    tus_figuritas <- c(tus_figuritas, figurita_actual)  
  }  
  
  return(length(tus_figuritas))  
}  
  
print(paste("Completaste tu álbum en :", cuantasFigs(6), "intentos"))  
  
## [1] "Completaste tu álbum en : 10 intentos"
```

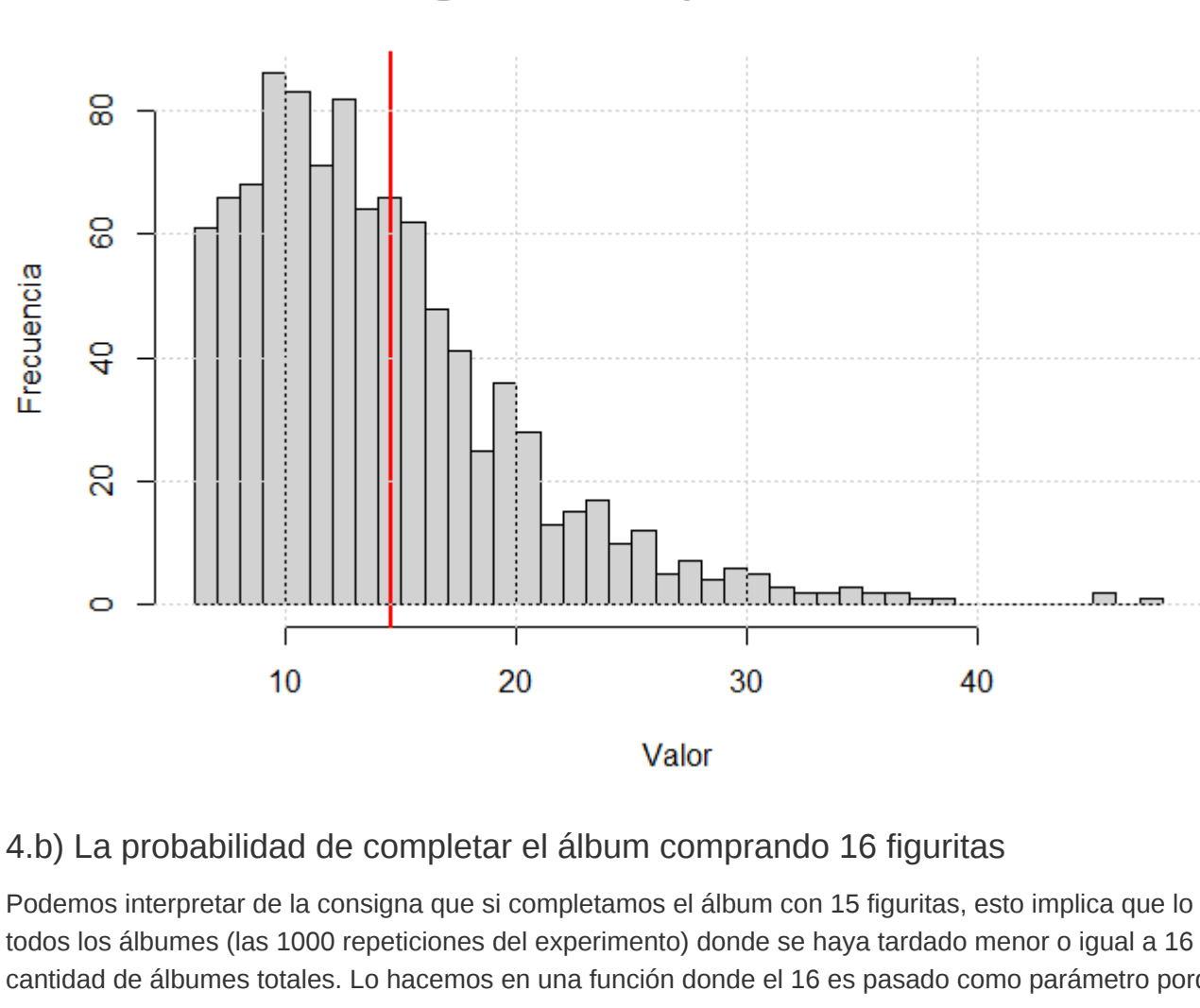
4) Calcular $Nrep=1000$ veces la función anterior utilizando `figusTotal=6` y guarde los resultados obtenidos en cada repetición. Con los resultados obtenidos para un álbum de seis figuritas, estimar:

Hace el experimento anterior con `figusTotal` cantidad de figuritas pero con `repeticiones` cantidad de veces. Ambas son pasadas por parámetro. Devolvemos un vector de longitud `repeticiones` que nos dirá la cantidad de figuritas necesarias según cada experimento.

```
repetirAlbum <- function(figusTotal, repeticiones) {  
  repeticion <- 0  
  albumes <- c()  
  
  while (repeticiones > repeticion) {  
    cantidad_compras_album <- cuantasFigs(figusTotal)  
    albumes <- c(albumes, cantidad_compras_album)  
    repeticion <- 1 + repeticion  
  }  
  
  return(albumes)  
}  
  
albumes_experimentacion <- repetirAlbum(6, 1000)  
longitud_promedio_album <- ceiling(mean(albumes_experimentacion))  
print(paste("En promedio, tuviste que realizar:", longitud_promedio_album, "compras de figuritas"))  
  
## [1] "En promedio, tuviste que realizar: 15 compras de figuritas"
```

Más aún, lo podemos ver gráficamente con un histograma marcando donde está el promedio

```
histogramaAlbumes <- function(albumes){  
  hist(albumes, breaks = 30, main = "Histograma de completación de álbum",  
    xlab = "Valor", ylab = "Frecuencia")  
  abline(v = mean(albumes), col = "red", lwd = 2)  
  grid()  
}  
  
histogramaAlbumes(albumes_experimentacion)
```



4.b) La probabilidad de completar el álbum comprando 16 figuritas

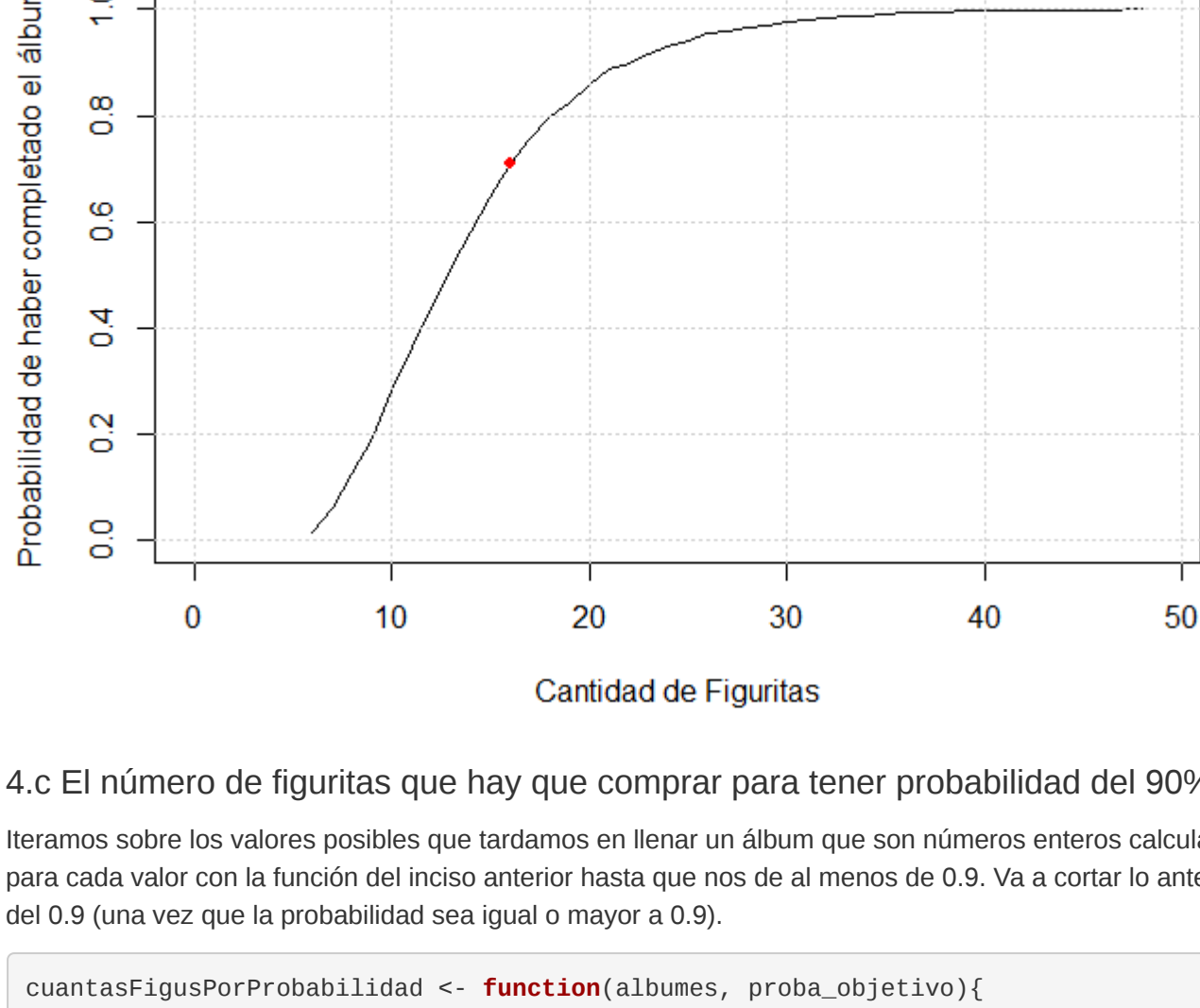
Podemos interpretar de la consigna que si completamos el álbum con 15 figuritas, esto implica que lo vamos a poder completar con 16. Tomamos todos los álbumes (las 1000 repeticiones del experimento) donde se haya tardado menor o igual a 16 figuritas en completarlo y lo dividimos por la cantidad de álbumes totales. Lo hacemos en una función donde el 16 es pasado como parámetro porque nos va a ser útil también más adelante.

```
probabilidadCompletarAlbum <- function(albumes, n) {  
  p <- sum(albumes <= n) / length(albumes)  
  return(p)  
}  
  
print(paste("La probabilidad de llenar el album de 6 figuritas en a lo sumo 16 intentos es:", probabilidadCompletarAlbum(albumes_experimentacion, 16)))  
  
## [1] "La probabilidad de llenar el album de 6 figuritas en a lo sumo 16 intentos es: 0.789"
```

Más aún, lo podemos ver gráficamente para todos los valores posibles donde se completa el álbum y marcamos con rojo a la probabilidad para las 16 figuritas. Es el gráfico de la probabilidad acumulada. Miramos todos los valores que puede llegar a tomar completar un álbum para tomarlos como referencia al mirar la probabilidad acumulada hasta ese punto

```
plotProbabilidadAcumulada <- function(albumes, n, tipo = "figus"){  
  valores_unicos <- sort(unique(albumes))  
  prob_acumulada <- numeric(length(valores_unicos))  
  
  for (i in seq_along(valores_unicos)) {  
    prob_acumulada[i] <- probabilidadCompletarAlbum(albumes, valores_unicos[i])  
  }  
  
  if (tipo != "figus") {  
    xlab_text <- "Cantidad de Paquetes"  
  } else {  
    xlab_text <- "Cantidad de Figuritas"  
  }  
  
  plot(valores_unicos, prob_acumulada, type = "n",  
    main = "Función de Probabilidad Acumulada",  
    xlab = xlab_text, ylab = "Probabilidad de haber completado el álbum",  
    xlim = c(0, max(valores_unicos) + 1), ylim = c(0, 1))  
  lines(valores_unicos, prob_acumulada, col = "black")  
  if (n <= max(valores_unicos)){  
    points(n, probabilidadCompletarAlbum(albumes, n), col = "red", pch = 19, cex = 0.9)  
  }  
  grid()  
}  
  
plotProbabilidadAcumulada(albumes_experimentacion, 16)
```

Función de Probabilidad Acumulada



4.c) El número de figuritas que hay que comprar para tener probabilidad del 90% de completar el álbum.

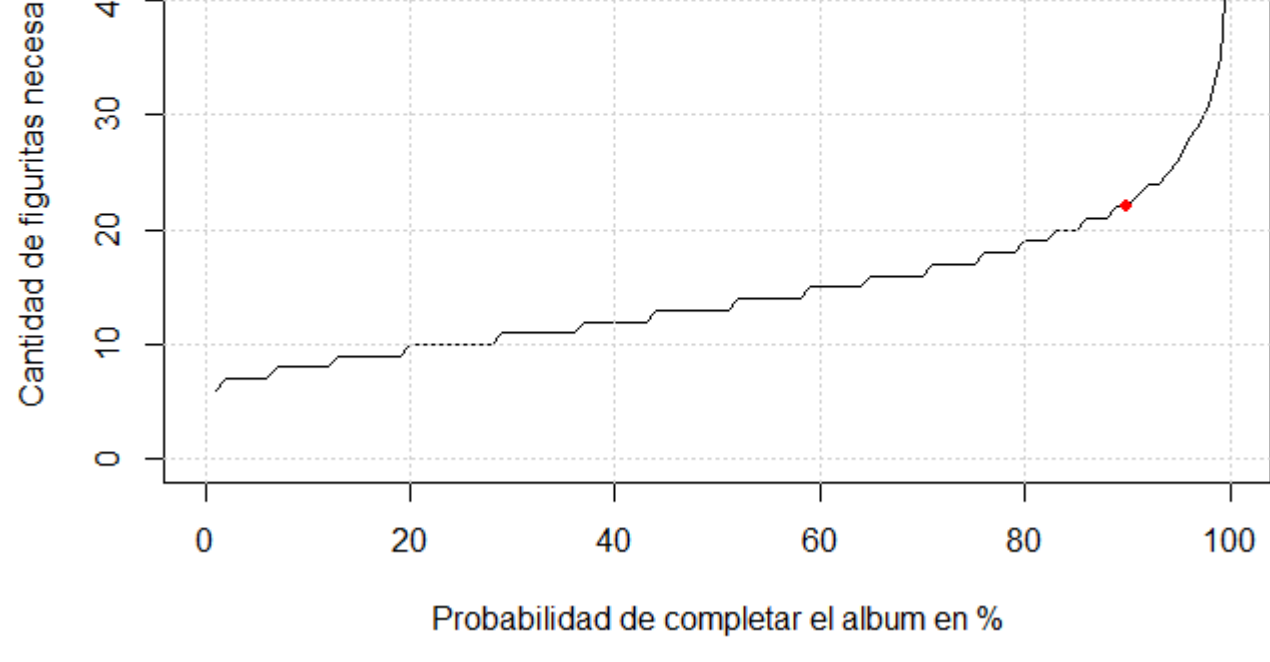
Iteramos sobre los valores posibles que tardamos en llenar un álbum que son números enteros calculando la probabilidad de completar el álbum para cada valor con la función del inciso anterior hasta que nos de al menos de 0.9. Va a cortar lo antes posible, es decir lo más cerca que esté del 0.9 (una vez que la probabilidad sea igual o mayor a 0.9).

```
cuantasFigsPorProbabilidad <- function(albumes, proba_objetivo){  
  min_figuritas <- min(albumes)  
  max_figuritas <- max(albumes)  
  
  for (n in min_figuritas:max_figuritas) {  
    if (probabilidadCompletarAlbum(albumes, n) >= proba_objetivo) {  
      return(n)  
    }  
  }  
  
  figus_por_proba <- cuantasFigsPorProbabilidad(albumes_experimentacion, 0.9)  
  print(paste("Para tener una probabilidad de llenar el album del 90% deberías comprar:", figus_por_proba))  
  
  ## [1] "Para tener una probabilidad de llenar el album del 90% deberías comprar: 22"
```

Más aún, lo podemos ver gráficamente para cada porcentaje y marcamos con rojo el correspondiente al porcentaje de 90%

```
plotProbabilidadCompletarAlbum <- function(albumes, proba, tipo= "figus"){  
  porcentajes <- seq(0.01, 1, by = 0.01)  
  figus_por_porcentajes <- numeric(length(porcentajes))  
  
  for (i in seq_along(porcentajes)) {  
    figus_por_porcentajes[i] <- cuantasFigsPorProbabilidad(albumes, porcentajes[i])  
  }  
  
  if (tipo != "figus") {  
    ylab_text <- "Cantidad de paquetes necesarios"  
  } else {  
    ylab_text <- "Cantidad de figuritas necesarias"  
  }  
  
  plot(porcentajes * 100, figus_por_porcentajes, type = "n",  
    xlab = "Probabilidad de completar el album en %",  
    ylab = ylab_text,  
    xlim = c(0, 100), ylim = c(0, max(albumes) + 1),  
    main = "Para completar el álbum según probabilidad solicitada")  
  
  lines(porcentajes * 100, figus_por_porcentajes, col = "black")  
  points(proba * 100, figus_por_porcentajes[porcentajes == proba], col = "red", pch = 19, cex = 0.9)  
  grid()  
}  
  
plotProbabilidadCompletarAlbum(albumes_experimentacion, 0.9)
```

Para completar el álbum según probabilidad solicitada



Caso intermedio: consideremos ahora que el álbum es de `figusTotal=640` figuritas y que las figuritas se siguen comprando individualmente.

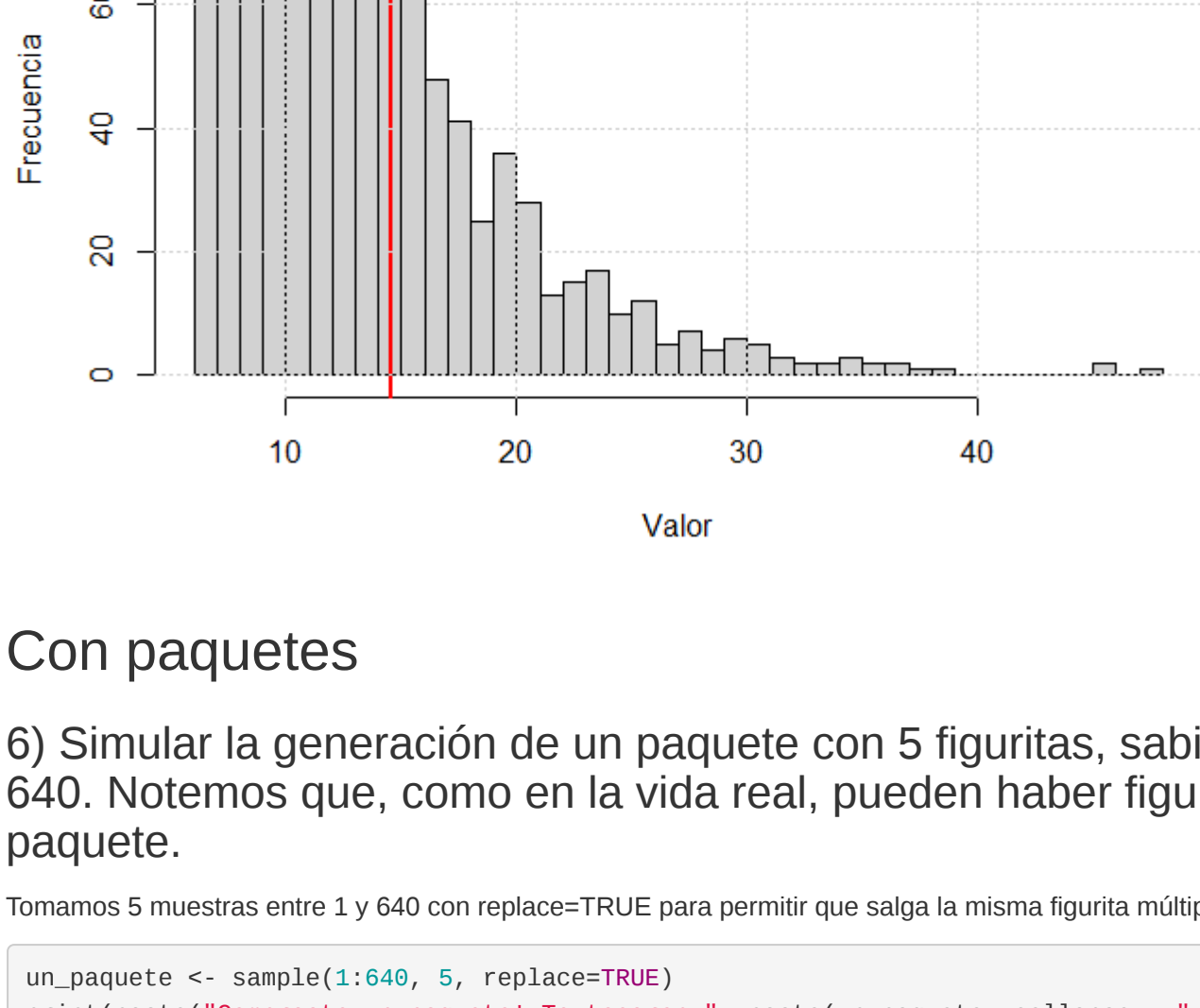
5) Calcular $Nrep=100$ veces la función `cuantasFigs(figusTotal=640)` utilizando `figusTotal=640` y guardar los resultados obtenidos en cada repetición en una lista. Con los resultados obtenidos estimar cuántas figuritas hay que comprar, en promedio, para completar el álbum (de 640 figuritas).

Tomamos de vuelta la función anterior de `repetirAlbum()` pero la llamamos para 640 cantidad de figuritas y 100 repeticiones. Calculamos el promedio de la cantidad de figuritas que se tarda en completar un álbum y redondeamos para arriba para tener un número natural de figuritas.

```
albumes_experimentacion_grande <- repetirAlbum(640, 100)  
longitud_promedio_album <- ceiling(mean(albumes_experimentacion_grande))  
print(paste("En promedio tuviste que realizar:", longitud_promedio_album, "compras de figuritas"))  
  
## [1] "En promedio tuviste que realizar: 4583 compras de figuritas"
```

Más aún, lo podemos ver gráficamente con el histograma del ejercicio 4.a

```
histogramaAlbumes(albumes_experimentacion)
```



Con paquetes

6) Simular la generación de un paquete con 5 figuritas, sabiendo que el álbum es de 640. Notemos que, como en la vida real, pueden haber figuritas repetidas en un paquete.

Tomamos 5 muestras entre 1 y 640 con `replace=TRUE` para permitir que salga la misma figurita múltiples veces.

```
un_paquete <- sample(1:640, 5, replace=TRUE)  
print(paste("Compraste un paquete! Te tocaron:", paste(un_paquete, collapse = ", ")))  
  
## [1] "Compraste un paquete! Te tocaron: 599, 48, 555, 214, 225"
```

7) Implementar una función `genPaquete(figusTotal, figusPaquete)` que dado el tamaño del álbum (`figusTotal`) y la cantidad de figuritas por paquete (`figusPaquete`) genere un paquete de figuritas al azar. Notemos que, como en la vida real, pueden haber figuritas repetidas en un paquete.

Hacemos lo mismo que en el punto anterior pero en una función que vaya hasta `figusTotal` en vez de 640 y tome `figusPaquete` en vez de 5 muestras. Las pasamos por parámetro.

```
genPaquete <- function(figusTotal, figusPaquete) {  
  un_paquete <- sample(1:figusTotal, figusPaquete, replace=TRUE)  
  return(un_paquete)  
}  
  
print(paste("Compraste un paquete! Te tocaron:", paste(genPaquete(640, 5), collapse = ", ")))  
  
## [1] "Compraste un paquete! Te tocaron: 333, 86, 10, 134, 4"
```

8) Implementar una función `cuantosPaquetes(figusTotal, figusPaquete)` que dado el tamaño del álbum (`figusTotal`) y la cantidad de figuritas por paquete (`figusPaquete`) simule el llenado del álbum y devuelva cuántos paquetes se debieron adquirir para completarlo.

Es análogo a la función `cuantasFigs()` solo que en vez de tomar una sola figurita tomamos la cantidad que vengan en un paquete y devolvemos la cantidad de paquetes.

```
cuantosPaquetes <- function(figusTotal, figusPaquete) {  
  album_completo <- c(1:figusTotal)  
  tus_figuritas <- c()  
  figuritas_paquetes <- list()  
  cantidad_paquetes <- 0  
  
  while (length(setdiff(album_completo, tus_figuritas)) > 0) {  
    paquete_actual <- genPaquete(figusTotal, figusPaquete)  
    cantidad_paquetes <- cantidad_paquetes + 1  
    tus_figuritas <- c(tus_figuritas, paquete_actual)  
    figuritas_paquetes[[cantidad_paquetes]] <- paquete_actual  
  }  
  
  return(cantidad_paquetes)  
}  
  
print(paste("Completaste tu álbum comprando", (cuantosPaquetes(640,5)), "paquetes"))  
  
## [1] "Completaste tu álbum comprando 1091 paquetes"
```

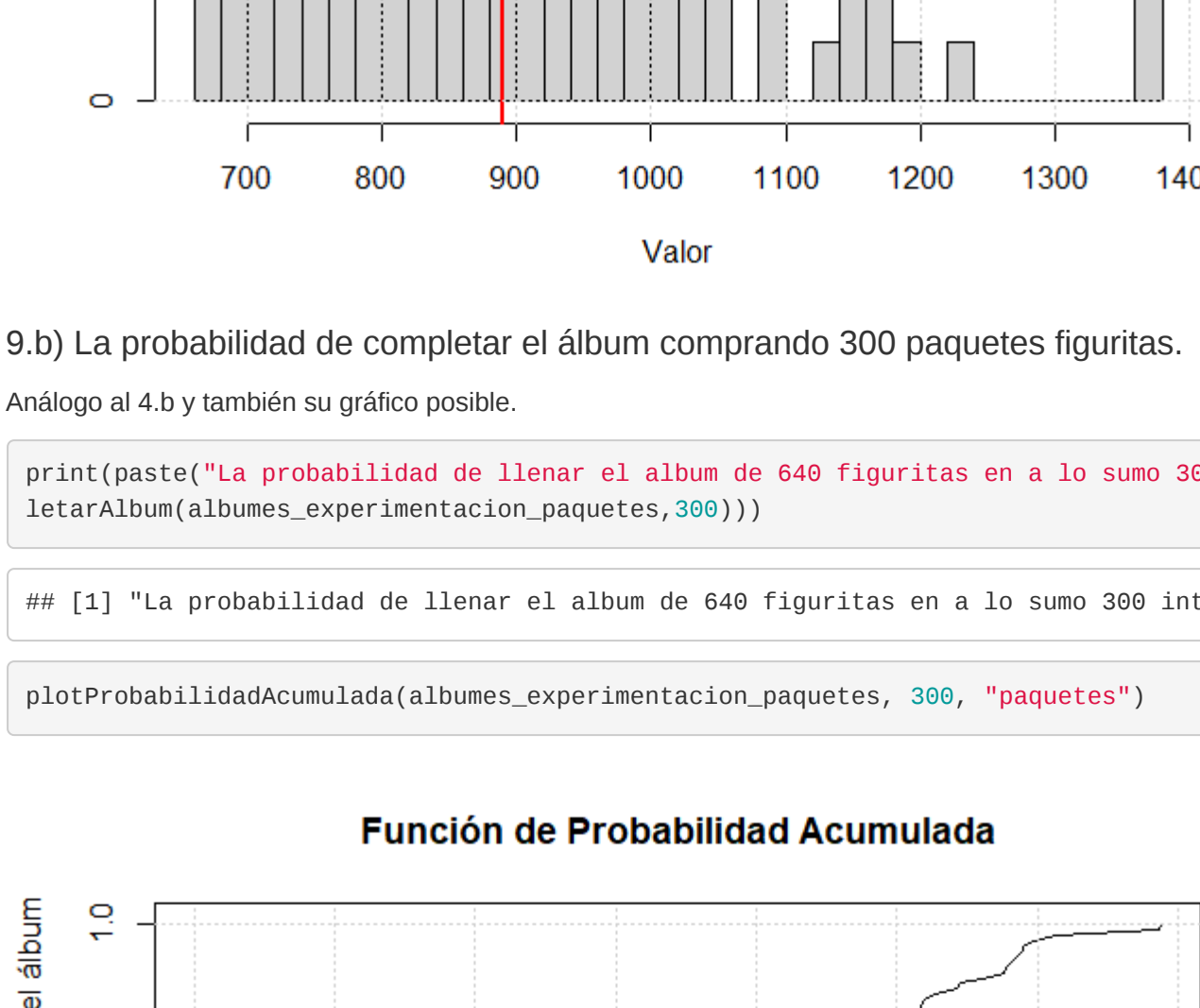
9) Calcular $Nrep=100$ veces la función `cuantosPaquetes`, utilizando `figusTotal=640`, `figusPaquete=5` y guarde los resultados obtenidos en una lista. Calcular

Corremos la función anterior `repeticiones` cantidad de veces para guardar las longitudes obtenidas en el vector `cantidad_compras_album`.

```
repetirAlbumConPaquetes <- function(figusTotal, figusPaquete, repeticiones) {  
  repeticion <- 0  
  albumes <- c()  
  
  while (repeticiones > repeticion) {  
    cantidad_compras_album <- cuantosPaquetes(figusTotal, figusPaquete)  
    albumes <- c(albumes, cantidad_compras_album)  
    repeticion <- repeticion + 1  
  }  
  
  return(albumes)  
}  
  
albumes_experimentacion_paquetes <- repetirAlbumConPaquetes(640, 5, 100)  
paquetes_promedio_album <- ceiling(mean(albumes_experimentacion_paquetes))  
print(paste("En promedio, tuviste que realizar:", paquetes_promedio_album, "compras de paquetes, es decir:", paquetes_promedio_album * 5, "cantidad de figuritas"))  
  
## [1] "En promedio, tuviste que realizar: 890 compras de paquetes, es decir: 4450 cantidad de figuritas"
```

Más aún, lo podemos ver gráficamente con el histograma del ejercicio 4.a. Estamos viendo la cantidad de paquetes que compramos.

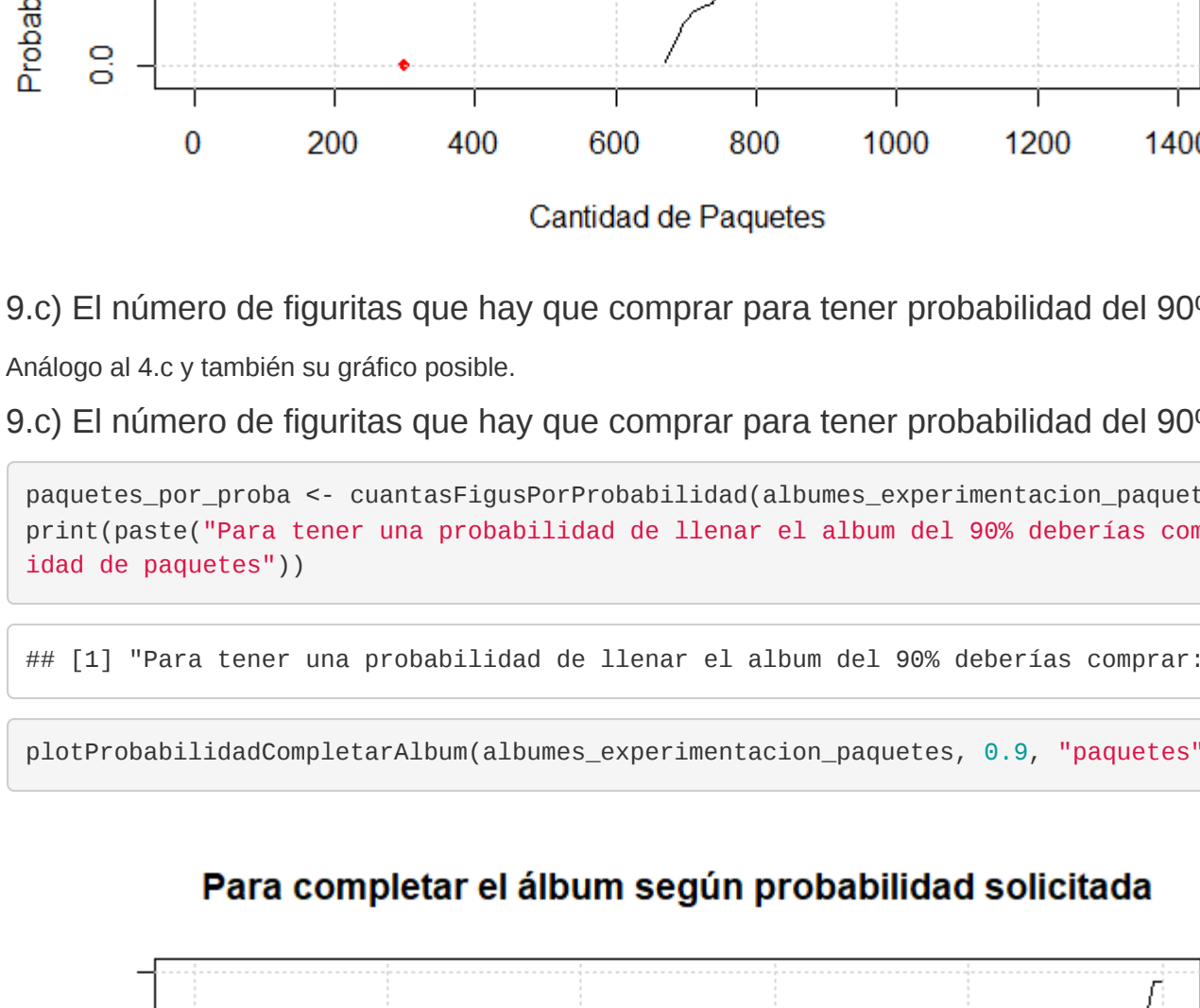
```
histogramaAlbumes(albumes_experimentacion_paquetes)
```



9.b) La probabilidad de completar el álbum comprando 300 paquetes figuritas.

Análogo al 4.b y también su gráfico posible.

```
print(paste("La probabilidad de llenar el album de 640 figuritas en a lo sumo 300 intentos es:", probabilidadCompletarAlbum(albumes_experimentacion_paquetes, 300)))  
  
## [1] "La probabilidad de llenar el album de 640 figuritas en a lo sumo 300 intentos es: 0"
```



9.c) El número de figuritas que hay que comprar para tener probabilidad del 90% de completar el álbum.

Análogo al 4.c y también su gráfico posible.

9.c) El número de figuritas que hay que comprar para tener probabilidad del 90% de completar el álbum

```
paquetes_por_proba <- cuantasFigsPorProbabilidad(albumes_experimentacion_paquetes, 0.9)  
print(paste("Para tener una probabilidad de llenar el album del 90% deberías comprar:", paquetes_por_proba, "cantidad de paquetes"))  
  
## [1] "Para tener una probabilidad de llenar el album del 90% deberías comprar: 1123 cantidad de paquetes"
```

Para completar el álbum según probabilidad solicitada

