TP1 Análisis Exploratorio de Datos

Gonzalo Barrera Borla y Octavio M. Duarte

Abril 11, 2018

Función que Simula la Colección del Álbum

La función elegida tiene más parámetros de los estrictamente necesarios porque nos interesó continuar el experimento y simular algunas cuestiones fuera de consigna como el efecto de un *mercado altruista* de intercambio de figuritas repetidas o de un subconjunto de *figuritas difíciles*.

En esencia es un bucle **while** que en cada ciclo aprovecha **sample** para generar un sobre de cinco figuritas. El album fue modelado como un conjunto de **figuritas-album** números que se inicializa en 0 para cada componente y suma 1 a cada una de ellas, cada vez que un sobre contiene su número respectivo.

Esto permite llevar la cuenta de las repetidas en caso de que se quiera simular un mayor número de colecciones en las condiciones indicadas antes.

Cuando todos los elementos del vector tienen valor mayor o igual a la cantidad de álbumes especificados, significa que se completaron las colecciones. En particular cuando todos son mayores o iguales a 1 hay un álbum lleno.

```
llenarAlbum <- function (
  figuritas_album = 500, figuritas_sobre = 5, n_albumes = 1, sobres_con_repetidas = F,
  pesos = rep(1, figuritas_album)) {
  # figuritas_album:= cantidad de figuritas distintas del album
  # figuritas_sobre:= cantidad de figuritas por sobre
  # n_albumes:= cantidad de albumes que se pretende completar
  #   Un unico coleccionista que rellena n albumes es una buena aproximacion a
  #   mercado de n coleccionistas "altruistas", que intercambian toda figurita repetida.
  # sobres_con_repetidas:= ¿Pueden los sobres contener figuritas repetidas?
  # pesos:= (opcional) Pesos para la "dificultad relativa" de cada figurita

  if (figuritas_album < 1) {
    print(paste("Pruebe con una cantidad positiva de figuritas por album."))
    return()
  }

  album <- rep(0, figuritas_album)
  n_sobres <- 0

  while(!all(album >= n_albumes)) {
    sobre <- sample(
      1:figuritas_album,
      figuritas_sobre,
      replace = sobres_con_repetidas,
      prob = pesos
    )

    album[sobre] <- album[sobre] + 1
    n_sobres <- n_sobres + 1
  }
```

```
  return(n_sobres)
}
```

Simulación y Tabulación de los Resultados

Entendimos que este procedimiento quedaba fuera de la consigna, así que optamos por el medio que nos pareció más conveniente: generamos un *dataframe* que va tomando sus propias columnas como parámetros para alimentar la función llenarAlbum, de tal forma que la información está organizada desde el momento de su generación.

La columna `k` indica el valor del subíndice a_k, que determina la cantidad de simulaciones a realizar. Para cada $k = 5, 10, 50, 100$, realizamos $n = 1000$ ensayos de k simulaciones cada uno, y guardamos el vector con los k resultados en la columna `obs`. Finalmente, "mutamos" los datos de la columna `obs` aplicándoles la función `mean`, ya que su promedio nos provee una buena estimación del valor esperado de a_k.

Guardamos el *dataframe* completo en disco para simplificar la creación de este informe, pero siéntase libre de ejecutar este fragmento de código y recrearlos.

```
# Usamos 1000 en lugar de 100 para mejorar la "definición" de los gráficos
n <- 1000
k <- rep(c(5,10,50,100), n)
obs <- map(k, ~replicate(., llenarAlbum()))
data_frame(k = k,
           obs = obs) %>%
  mutate(a_k = map_dbl(obs, mean)) -> datos

save(datos, file='TP1.RData')
```

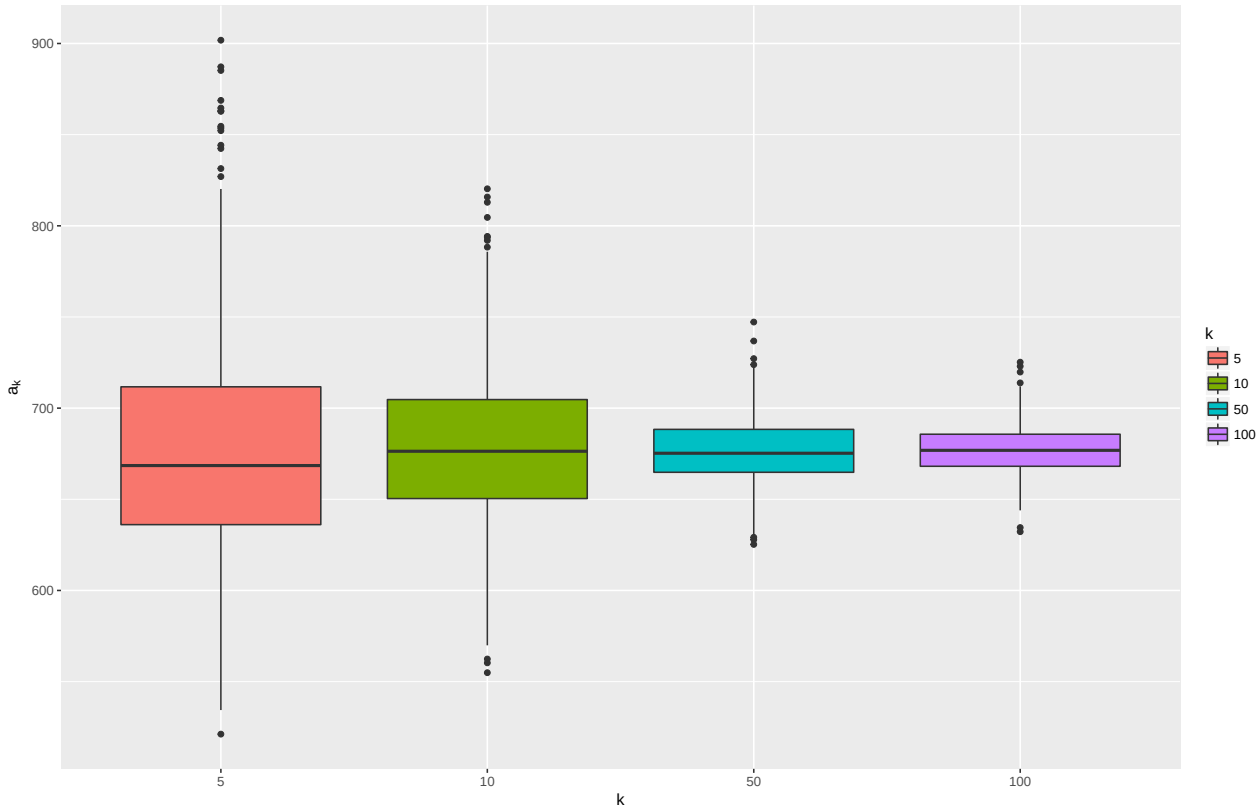
Gráficos de Caja

```
load('TP1.RData')

ggplot(data = datos, mapping = aes(x = factor(k), y = a_k, fill = factor(k))) +
  geom_boxplot() +
  labs(title = TeX("Fig. 1: Boxplot para $a_k$, agrupados según k"),
       x = "k", y = TeX("$a_k$"), fill = "k")-> fig1_boxplot

ggsave("fig1_boxplot.png", fig1_boxplot, width = 12, height = 8)
fig1_boxplot
```

Fig. 1: Boxplot para a_k, agrupados según k

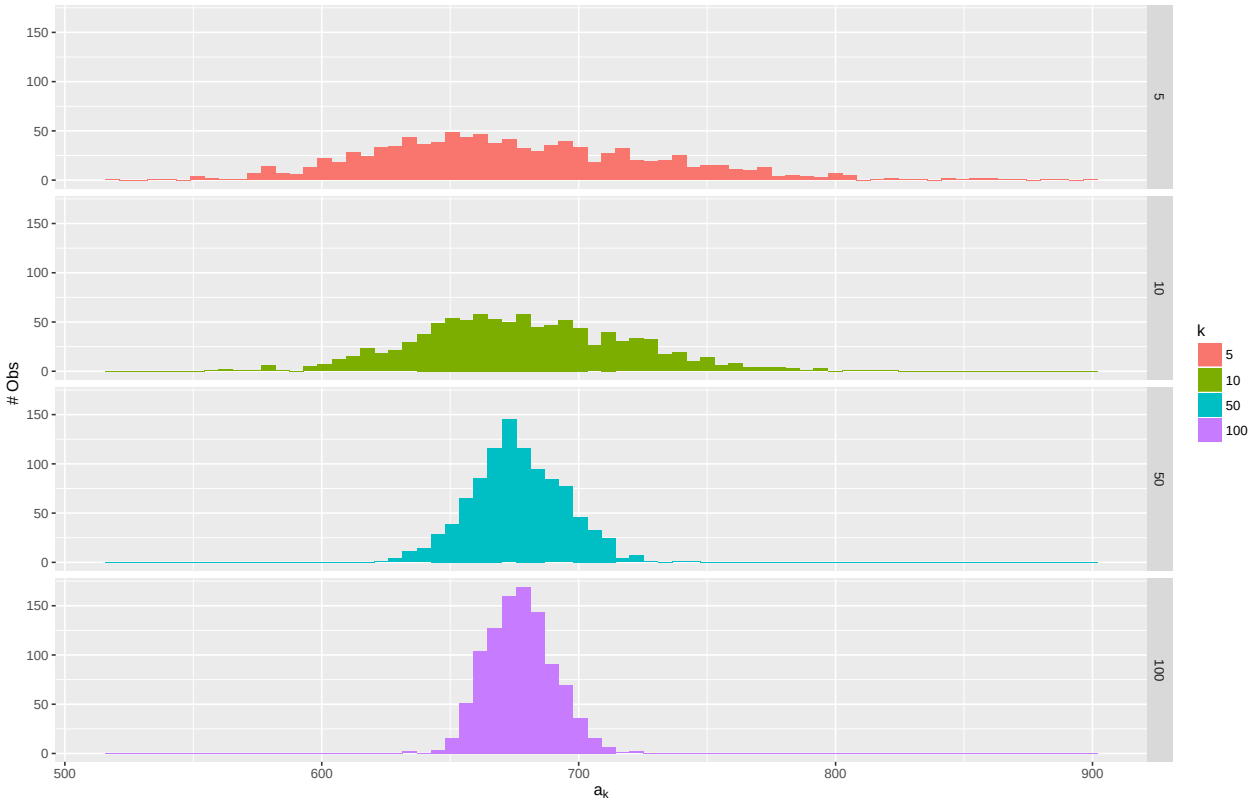


Los gráficos de caja no revelan ninguna tendencia inusual. Para una baja cantidad de repeticiones, la variable estudiada presenta una gran dispersión pero esta se reduce notablemente con el incremento de la variable k hasta el punto de que el diagrama de a_{100} y sus datos atípicos prácticamente caben dentro de la caja del diagrama de a_5

```
ggplot(datos, aes(x = a_k, group = factor(k), fill = factor(k))) +
  geom_histogram(bins = 70) +
  facet_grid(k ~ .) +
  labs(title = TeX("Fig. 2: Histogramas para $a_k$, agrupados según k (n = 1000)"),
       x = TeX("$a_k$"), y = "# Obs", fill = "k")-> fig2_hist

ggsave("fig2_hist.png", fig2_hist, width = 12, height = 8)
fig2_hist
```

Fig. 2: Histogramas para a_k, agrupados según k (n = 1000)



El histograma parece coincidir con lo exhibido por el gráfico de caja: a medida que se incrementa la variable k de datos usados para calcular la esperanza esta se estabiliza y se concentra.

```
datos %>%
  group_by(k) %>%
  summarise(
    media_ak = mean(a_k),
    mediana_ak = median(a_k),
    media_ak_01 = mean(a_k, trim = 0.1),
    var_ak = var(a_k),
    iqr_ak = IQR(a_k),
    mad_ak = mad(a_k)
  ) -> estadisticos_resumen
kable(estadisticos_resumen)
```

k	media_ak	mediana_ak	media_ak_01	var_ak	iqr_ak	mad_ak
5	675.8886	668.500	672.9402	3218.7267	75.600	54.11490
10	678.3066	676.350	677.5124	1670.2556	54.300	39.80781
50	676.2989	675.260	676.1632	321.9196	23.545	17.74672
100	677.2378	676.855	676.9644	168.2273	17.585	13.08394