Ejercicios de R

Curso: Introducción a la Estadística y Probabilidades CM-274

Lecturas Importantes

1. Artículo de Karlijn Willems, sobre Jupyter and R Markdown: Notebooks con R
   `https://www.datacamp.com/community/blog/jupyter-notebook-r`.
2. Notas de John D. Cook
   `http://www.johndcook.com/blog/notes/` sobre algunos de los más importantes temas de Matemática, del Machine Learning y R.

Preguntas

1. (a) El própósito de la función `col_means()` definida de la siguiente manera

```
> col_means <- function(df) {
+   numeric <- sapply(df, is.numeric)
+   numeric_cols <- df[, numeric]

+   data.frame(lapply(numeric_cols, mean))
+ }
```

   Sin embargo, la función no funciona para entradas inusuales. Observa los siguientes resultados, decide cuáles son incorrectos y modifica `col_means ()` para que funcione correctamente

```
> col_means(mtcars)
> col_means(mtcars[, 0])
> col_means(mtcars[0, ])
> col_means(mtcars[, "mpg", drop = F])
> col_means(1:10)
> col_means(as.matrix(mtcars))
> col_means(as.list(mtcars))
>
> mtcars2 <- mtcars
> mtcars2[-1] <- lapply(mtcars2[-1], as.character)
> col_means(mtcars2)
```

   (b) La siguiente función `retrasa` un vector, devolviendo una versión de x que es n valores detrás del original. Mejora la función para que (1) devuelva un mensaje de error útil si n no es un vector, y (2) tenga un comportamiento razonable cuando n es 0 o es más grande que x.

2. Explica el siguiente código, en el contexto de lo siguiente

   Pedro y Pablo juegan un juego que implica lanzamientos repetidos de una moneda. En un lanzamiento dado, si se observan caras, Pedro gana \$1 ; de lo contrario, Pedro le da \$1 a Pablo. Si Pedro comienza con cero dolares y se han realizado 50 lanzamientos.

```
> options(width=60)
> sample(c(-1, 1), size=50, replace=TRUE)
>
>
> win = sample(c(-1, 1), size=50, replace=TRUE)
> cum.win = cumsum(win)
> cum.win
>
> par(mfrow=c(2, 2))
> for(j in 1:4){
+   win = sample(c(-1, 1), size=50, replace=TRUE)
+   plot(cumsum(win), type="l" ,ylim=c(-15, 15))
+   abline(h=0)
+ }
```

- ¿ Cuál es la probabilidad de que Pedro se retire incluso después de 50 lanzamientos?
- ¿ Cuál es el número probable de lanzamientos en los que Pedro estará ganando?
- ¿ Cuál será el valor de la mejor ganancia de Pedro durante el juego?

3. Considera la siguiente matriz circulante:

$$P = \begin{pmatrix} 0.1 & 0.2 & 0.3 & 0.4 \\ 0.4 & 0.1 & 0.2 & 0.3 \\ 0.3 & 0.4 & 0.1 & 0.2 \\ 0.2 & 0.3 & 0.4 & 0.1 \end{pmatrix}$$

   (a) P es un ejemplo de matriz `estocástica`. Usa la función `apply()` para verificar que la suma de filas es 1.

   (b) Calcula P^n para $n = 2, 3, 5, 10$. ¿Hay un patrón apareciendo?.

   (c) Encuentra un vector no negativo x cuyos elementos suman 1 y que satisface

$$(I - P^T)x = 0$$

   ¿ Existe alguna conexión entre P^{10} y $x = 0$?.

   (d) Usa un bucle, para generar una secuencia pseudoaleatorias de números y desde el conjunto $\{1, 2, 3, 4\}$, usando las siguientes condiciones

   - Sea $y_1 \leftarrow 1$
   - Para $j = 2, 3, \ldots, n$, sea $y_j = k$ con probabilidad $P_{y_{j-1},k}$.
     Por ejemplo, y_2 debería ser asignado al valor 1, con probabilidad 0.1; 2, con probabilidad 0.2; y así. Escoge un n de valor grande como 10000.
     El vector y resultante es un ejemplo de una cadena de Markov simulada.

   (e) Usa la función `table()` para determinar la distribución de frecuencias relativas de cuatro posibles valores en el vector y.

   (f) Repetir el ejercicio previo con la siguiente matriz

$$P = \begin{pmatrix} 0.1 & 0.2 & 0.3 & 0.4 & 0.0 & 0.0 & 0.0 \\ 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.4 \\ 0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 0.0 & 0.0 \\ 0.3 & 0.3 & 0.3 & 0.1 & 0.0 & 0.0 & 0.0 \\ 0.3 & 0.3 & 0.3 & 0.1 & 0.0 & 0.0 & 0.0 \\ 0.3 & 0.3 & 0.3 & 0.1 & 0.0 & 0.0 & 0.0 \\ 0.3 & 0.3 & 0.3 & 0.1 & 0.0 & 0.0 & 0.0 \end{pmatrix}$$

4. Explica los siguientes códigos de R

(a)
```
> infix <- mget(grep("<-",ls('package:base'), value = T), inherits = T)
> m <- sapply(infix, is.primitive)
>
> names(m[m == T])
```

(b)
```
> library(ggplot2)
> library(gridExtra)
> set.seed(10005)
>
> xvar <- c(rnorm(1500, mean = -1), rnorm(1500, mean = 1.5))
> yvar <- c(rnorm(1500, mean = 1), rnorm(1500, mean = 1.5))
> zvar <- as.factor(c(rep(1, 1500), rep(2, 1500)))
> xy <- data.frame(xvar, yvar, zvar)
>
> g1<-ggplot(xy, aes(xvar)) + geom_histogram()
> g2<-ggplot(xy, aes(xvar)) + geom_histogram(binwidth=1)
> g3<-ggplot(xy, aes(xvar)) + geom_histogram(fill=NA, color="black") + theme_bw()
>
> g4<-ggplot(xy, aes(x=xvar)) + geom_histogram(aes(y = ..density..), color="black", fill=NA)
> + theme_bw()
>
> grid.arrange(g1, g2, g3, g4, nrow=1)
>
>
> p1<-ggplot(xy, aes(xvar)) + geom_density()
>
> p2<-ggplot(xy, aes(x=xvar)) +
+   geom_histogram(aes(y = ..density..), color="black", fill=NA) +
+   geom_density(color="blue")
> p3<-ggplot(xy, aes(xvar, fill = zvar)) + geom_density(alpha = 0.2)
>
> grid.arrange(p1, p2, p3, nrow=1)
>
>
> b1<-ggplot(xy, aes(zvar, xvar)) +
+   geom_boxplot(aes(fill = zvar)) +
+   theme(legend.position = "none")
>
> b2<-ggplot(xy, aes(zvar, xvar)) +
+   geom_jitter(alpha=I(1/4), aes(color=zvar)) +
+   theme(legend.position = "none")
>
> b3<-ggplot(xy, aes(x = xvar)) +
+   stat_density(aes(ymax = ..density..,  ymin = -..density..,
+                fill = zvar, color = zvar),
+                geom = "ribbon", position = "identity") +
+   facet_grid(. ~ zvar) +
+   coord_flip() +
+   theme(legend.position = "none")
>
> grid.arrange(b1, b2, b3, nrow=1)
>
> ggplot(xy,aes(xvar,yvar))  + geom_point() + geom_rug(col="darkred",alpha=.1)
>
> scatter <- ggplot(xy,aes(xvar, yvar)) +
+   geom_point(aes(color=zvar)) +
```

```
+   scale_color_manual(values = c("orange", "purple")) +
+   theme(legend.position=c(1,1),legend.justification=c(1,1))
>
> plot_top <- ggplot(xy, aes(xvar, fill=zvar)) +
+   geom_density(alpha=.5) +
+   scale_fill_manual(values = c("orange", "purple")) +
+   theme(legend.position = "none")
>
>
> plot_right <- ggplot(xy, aes(yvar, fill=zvar)) +
+   geom_density(alpha=.5) +
+   coord_flip() +
+   scale_fill_manual(values = c("orange", "purple")) +
+   theme(legend.position = "none")
>
> grid.arrange(plot_top, empty, scatter, plot_right, ncol=2, nrow=2, widths=c(4, 1), heights=c(
```

5. En este ejercicio escribe un algoritmo, para la siguiente tarea geométrica

   Entrada Un conjunto de puntos en el plano $\{p_1 = (x_1, y_1), p_2 = (x_2, y_2) \ldots, p_n = (x_n, y_n)\}$

   Salida Un par de puntos, esto es $p_i \neq p_j$, cuya distancia entre p_i y p_j

   $$\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$$

   es mínima.

   Asume que n tiene potencia 2 y que todas las coordenadas x_i (x-coordenadas) son distintas como los son las coordenadas y_i (y-coordenadas).

   (a) Encuentra un valor x para el cúal exactamente la mitad de puntos tienen $x_i < x$ y la otra $x_i > x$. Sobre esta base, dividimos los puntos en dos grupos L y R.

   (b) Recursivamente encuentra los pares más cercanos en L y R. Sean los pares $p_L, q_L \in L$ y $p_R, q_R \in R$, cuyas distancias son respectivamente d_L y d_R. Sea d la menor de esas dos distancias.

   (c) Queda por ver si hay un punto en L y un punto en R que separados unos de otros están tiene una distancia menor que d . Para ello, quita todos los puntos con $x_i < x - d$ o $x_i > x + d$ y ordena los puntos que quedan (y-coordenadas).

   (d) A través de esta lista ordenada, y para cada punto, calcula la distancia a los puntos subsecuentes. Sea p_M y q_M los puntos más cercanos de esta forma.

   La respuesta es uno de los tres pares $\{p_L, q_L\}, \{p_R, q_R\}$ y $\{p_M, q_M\}$.