



**Escuela de Gobierno y
Transformación Pública
Tecnológico de Monterrey**

TAREA # 1, PARTE 1

Ciencia de datos

Entrega: 5 de febrero del 2021

Ejercicios resueltos en colaboración con Nagib Gobera Mac Farland, Elva Deyanira
Martínez González, Álvaro Isaac Vázquez Aguilar y Elvia Daniela Flores Resendez

Guillermo Alberto García Candanosa
A01034958@itesm.mx

1. Ejercicios de R

1.1 Análisis de texto y caracteres

Considera el siguiente texto:

Todos los Estados, todas las dominaciones que han ejercido y ejercen soberanía sobre los hombres, han sido y son repúblicas o principados. Los principados son, o hereditarios, cuando una misma familia ha reinado en ellos largo tiempo, o nuevos. Los nuevos, o lo son del todo, como lo fue Milán bajo Francisco Sforza, o son como miembros agregados al Estado hereditario del príncipe que los adquiere, como es el reino de Nápoles para el rey de España. Los dominios así adquiridos están acostumbrados a vivir bajo un príncipe o a ser libres; y se adquieren por las armas propias o por las ajenas, por la suerte o por la virtud.

Construye un iterador que recorra el texto anterior y ejecute las siguientes instrucciones:

- a) Asigna las palabras a un vector que las incluya todas (elimina comas y puntos).

```
texto = "tarea1_1.txt"
vec_texto = readLines(texto)
lista_texto <- Corpus (VectorSource(vec_texto))

lista_texto <- tm_map(lista_texto, stripWhitespace)
lista_texto <- tm_map(lista_texto, removePunctuation)
lista_texto <- tm_map(lista_texto, tolower)
vec2_texto <- unlist(lista_texto)
list_palabras <- strsplit(vec2_texto, " ")
vec_palabras <- unlist(list_palabras)
```

- b) Itera en el vector de palabras y elimina preposiciones y artículos. Salvar como un nuevo vector

```
vec_preposicionesyarticulos <- c("a", "ante", "bajo", "cabe", "con", "contra", "de",  
  "desde", "durante", "en", "entre", "hacia", "hasta",  
  "mediante", "para", "por", "según", "sin", "so", "sobre",  
  "tras", "versus", "vía", "el", "la", "los", "las", "uno",  
  "una", "unos", "unas", "del")  
  
i <- is.character("")  
j <- is.character("")  
for (i in vec_palabras){  
  for (j in vec_preposicionesyarticulos){  
    if ( i == j){  
      pos <- match(i,vec_palabras)  
      vec_palabras <- vec_palabras[-pos]  
    }  
  }  
}  
  
vec_palabras_nuevo <- vec_palabras  
vec_palabras <- unlist(list_palabras)
```

- c) Emplea el vector resultante del paso anterior y clasifica tus palabras en una lista que contenga los siguientes tres elementos: un vector con las palabras que empiezan en vocal y terminan en consonante, un vector con las palabras que empiezan en vocal y terminan en vocal y un vector con el resto de las palabras que no aplican al caso anterior.
- d) El resultado final es una lista con los tres vectores de palabras.

2. Simulaciones en R

2.1 Juego de cartas en R

Considera las siguientes líneas de código para generar una baraja de póker:

```
num <- c(1,2,3,4,5,6,7,8,9,10,11,12,13)  
palos <- c(" trebol "," espadas "," corazones "," picas ")  
baraja <- c()  
  
for (i in num ){  
  for (j in palos ){  
    card <- ( paste (i, j, sep ="-"))  
    baraja <- c( baraja , card )  
  }  
}
```

Toma el código anterior como punto de partida y utiliza R para resolver las siguientes tareas:

- a) Construye una función que tome una carta del mazo al azar.

```
alAzar <- function(input_baraja,input_numero_cartas){  
  carta = sample(input_baraja,input_numero_cartas)  
  return(carta)  
}  
  
alAzar(baraja,1)
```

R: Se generó la función `alAzar(input_baraja,input_numero_cartas)`, a la cual se le alimenta el vector de la baraja a utilizar y el número de cartas que se escogerán al azar de la baraja.

- b) ¿Cuál es la probabilidad de sacar un as de espadas? Calcula la probabilidad de forma analítica y presenta tu resultado. Calcula la probabilidad simulando que sacas 10,000 cartas al azar y verificando cuántas de esas 10,000 cartas son un as de espadas y presenta tu código.

```
prob_analitica <- 1/52 #probabilidad analítica de sacar un as de espadas  
  
# Declaro mis variables iniciales  
as_espadas <- is.numeric(0)  
no_as_espadas <- is.numeric(0)  
k <- is.numeric(0)  
  
# Inicio ciclo de sacar 10,000 cartas al azar.  
for (k in 1 : 10000){  
  carta = alAzar(baraja,1)  
  if(carta == "1- espadas"){  
    as_espadas = as_espadas + 1  
  }  
  else{  
    no_as_espadas = no_as_espadas + 1  
  }  
}  
  
# Cálculo empírico de probabilidad de sacar un as de espadas con 10,000 experimentos  
prob_as_espadas <- as_espadas/(as_espadas + no_as_espadas)
```

R: Se determinó que la probabilidad analítica de obtener un as de espadas, resultado de dividir $1/52$ en el objeto `prob_analitica`, es de 0.0192. Al determinar esta misma probabilidad de manera empírica, se determinó en el objeto `prob_as_espadas` una probabilidad de 0.0189.

- c) Revuelve tus 52 cartas al aleatoriamente y retira cuatro cartas al azar. Corre una simulación para calcular la probabilidad de que obtengas un J, Q, R del mismo palo. Presenta el resultado y el código para generarlo.

```
# Declaro mis variables iniciales
l <- is.numeric(0)
jqr <- is.numeric(0)
njqr <- is.numeric(0)
counter <- is.numeric(0)

for (l in 1 : 10000){
  baraja_revuelta <- sample(baraja) # aquí revuelvo mi baraja
  cuatroCartas <- alAzar(baraja_revuelta,4)
  cuatroCartas_list <- strsplit(cuatroCartas, "-")
  cuatroCartas_vec <- unlist(cuatroCartas_list)
  numero_vec <- as.numeric(c(cuatroCartas_vec [1], cuatroCartas_vec [3],
                             cuatroCartas_vec [5], cuatroCartas_vec [7]))
  palos_vec <- as.character(c(cuatroCartas_vec [2], cuatroCartas_vec [4],
                              cuatroCartas_vec [6], cuatroCartas_vec [8]))

  if ((palos_vec[1] == palos_vec[2]) & (palos_vec[1] == palos_vec[3]) & (palos_vec[1] ==
palos_vec[4])){
    for (m in numero_vec){
      if(m > 10){
        counter = counter + 1
      }
    }
    if (counter == 3){
      jqr = jqr + 1
      counter = 0
    } else {
      njqr = njqr + 1
      counter = 0
    }
  } else if ((palos_vec[1] == palos_vec[2]) & (palos_vec[1] == palos_vec[3])){
    if (sum(numero_vec[1],numero_vec[2],numero_vec[3]) == 36){
      jqr = jqr + 1
    } else {
      njqr = njqr + 1
    }
  } else if ((palos_vec[1] == palos_vec[2]) & (palos_vec[1] == palos_vec[4])){
    if (sum(numero_vec[1],numero_vec[2],numero_vec[4]) == 36){
      jqr = jqr + 1
    } else {
      njqr = njqr + 1
    }
  }
}
```

```

} else if ((palos_vec[1] == palos_vec[3]) & (palos_vec[1] == palos_vec[4])){
if (sum(numero_vec[1],numero_vec[3],numero_vec[4]) == 36){
  jqr = jqr + 1
} else {
  njqr = njqr + 1
}
} else if ((palos_vec[2] == palos_vec[3]) & (palos_vec[2] == palos_vec[4])){
if (sum(numero_vec[2],numero_vec[3],numero_vec[4]) == 36){
  jqr = jqr + 1
} else {
  njqr = njqr + 1
}
} else {
  njqr = njqr + 1
}
}

prob_jqr <- jqr/(jqr + njqr)

```

R: Se determinó empíricamente que la probabilidad de obtener J, Q y R del mismo palo al tomar 4 cartas de una baraja revuelta es de 0.00059988.

2.2 Romeo y Julieta

Romeo y Julieta tienen una cita a una hora determinada, cada uno llegará a la cita con un retraso de entre 0 y 1 hora, con cada par de demoras siendo igualmente probables. El primero en llegar esperará por 15 minutos y se irá si el otro no llega en ese lapso. ¿Cuál es la probabilidad de que Romeo y Julieta se encuentren? Simúlalo en R.

```
se_encuentran <- is.numeric(0)
no_se_encuentran <- is.numeric(0)
counter <- is.numeric(0)

for (romeo in 0:59){
  for (julieta in 0:59){
    if (abs(romeo-julieta) <= 15){
      se_encuentran = se_encuentran + 1
    }
    else {
      no_se_encuentran = no_se_encuentran + 1
    }
    counter = counter + 1
  }
}

prob_se_encuentran = se_encuentran/(se_encuentran + no_se_encuentran)
```

R: Se determinó que la probabilidad de que se encuentren, representada por el objeto `prob_se_encuentran`, es de 0.450.