Guillermo Alberto García Candanosa

A01034958@itesm.mx

Entrega: 17 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

Tarea # 1, Parte 2

Ciencia de datos

# Escuela de Gobierno y Transformación Pública | Tec de Monterrey

# Generales

## 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:

1. 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)*

1. 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)*

1. 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.
2. 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:

1. 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.

1. ¿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.

1. 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.