Tarea1

Abraham Nieto 51556

20 de enero de 2017

Parte 1

1.-Investigar y ejemplificar como se representa cada tipo de Dato en R.

Utilizamos la fución “class” para revisar Tipo de datos básicos: 1 Reales-datos numéricos de tipo float.

```
class(2.6)
```

```
## [1] "numeric"
```

```
class(c(2.4,4.6,7.8))
```

```
## [1] "numeric"
```

2. Reales enteros Numéricos de tipo integer

```
class(4L)
```

```
## [1] "integer"
```

```
class(c(3L,4L,5L))
```

```
## [1] "integer"
```

3.-Booleanos indican la validez de una condición (TRU/FALSE)

```
class(TRUE)
```

```
## [1] "logical"
```

```
class(c(TRUE,FALSE,FALSE))
```

```
## [1] "logical"
```

4.-números complejos

```
class(2+3i)
```

```
## [1] "complex"
```

5.-Datos categóricos(nominales)

```
class('2.6')
```

```
## [1] "character"
```

```
class(c('a','b','c'))
```

```
## [1] "character"
```

```
class(factor(c('a','b','c')))
```

```
## [1] "factor"
```

6.-Datos Ordinales son tipo nominal o numérico pero con un orden De forma simple no se identifica este tipo de dato, por tanto la forma de caracterizarlo es utilizando la función factor que funciona con datos numéricos o de tipo categórico:

```
x<-c('b','c','a')
class(x)
```

```
## [1] "character"
```

```
factor(x)
```

```
## [1] b c a
## Levels: a b c
```

primero x es un vector de tipo categórico y al usar ordered define un orden de este vector de tal manera que tenemos un dato ordinal

```
xx<-ordered(x)
class(x)
```

```
## [1] "character"
```

```
class(xx)
```

```
## [1] "ordered" "factor"
```

```
xx
```

```
## [1] b c a
## Levels: a < b < c
```

al final el nuevo vector xx aparece de tipo ‘factor’ es un vector ordinal, por tanto usando esta función definimos el dato de tipo ordinal.

Parte 2

2.-¿Cómo se miden las distancias entre vectores de un mismo tipo de dato

booleanos o binarios:

Cuando las variables son binarias(valores 0,1), supongamos que tenemos p variables binarias

$$X_1, \ldots, Xp$$

para cada par de individuos

$$(i, j)$$

sean a,b,c,d las frecuencias de (1,1), (1,0), (0,1) y (0,0) respectivamente con

$$p = a + b + c + d$$

un coeficiente de similaridad debería ser función de

$$a, b, c, d$$

por ejemplo

$$s_{ij} = (a + d)/p(Sokal - Michener)$$

$$s_{ij} = (a)/(a + b + c)(Jaccard)$$

$$s_{ij} = a/(a + 2(b + c))\ (Sokal - Sneath)$$

$$s_{ij} = 2a/(a + b)(a + c)\ (Sokal - Michener)$$

Podemos transformar la similaridad en distancia aplicando:

$$d_{ij}^2 = (1 - s_{ij})$$

además es una distancia euclidea.

```
b1<-as.logical(c(1,0,1,0))
b2<-as.logical(c(0,0,1,1))
J=1/(1+1+1)
1-J
```

```
## [1] 0.6666667
```

Categóricos o nominales: índice de similitud simple suponagmos 2 vectores nominales

$$v_1, v_2$$

tal que la distancia entre ellos está definida de la siguiente forma:

$$d(i,j) = 1 - mm/p$$

donde mm es el numero de variables donde los individuos (i,j) coinciden en las categorías y p es el número de variables por tanto:

```
simnom<-function(v1,v2){
  mm<-0
  for(i in 1:3){
    if (v1[i]==v2[i]){
      mm<-mm+1
    }
  }
  return(1-(mm/length(v1)))
}

v1<-c('a','b','c')
v2<-c('b','b','c')
simnom(v1,v2)
```

```
## [1] 0.3333333
```

existe la funcion daisy para calcular esta distancia por ejemplo:

```
library(cluster)
similitud<-daisy(rbind(factor(v1),factor(v2)),metric = 'gower')
1-similitud
```

```
## Dissimilarities :
##           1
## 2 0.3333333
##
## Metric :  mixed ;  Types = I, I, I
## Number of objects : 2
```

Supongamos que las variables pueden ser clasificadas en k-categorías excluyentes

$$\$A_1, \ldots, A_k$$

con probabilidades

$$p = (p_1, \ldots, p_{k)}$$

donde

$$\sum_{i=1}^{n} p_i = 1$$

podemos definir distancia si 2 individuos

$$(i,j)$$

tiene categorías

$$A_h$$

y

$$A_h{}^{'}$$

respectivamente una distancia al cuadrado entre

$$i,j$$

es:

$$d_{i,j}^2 = 0 \; si \; h = h^{'} \; o \; d_{i,j}^2 = p_h^- 1 + p_{h^{'}}^- 1 \; si \; h \neq h'$$

si hay varios conjuntos de variables nominales con un total de K categorías o estados un coeficiente de similaridad es

$$\alpha/K$$

(‘matching coefficient’) donde

$$\alpha$$

es el número de coicidencias.

Categoricos y Booleanos: Para poder comparar 2 vectores de datos categóricos o booleanos podemos utilizar métricas de similardad, por ejemplo la distancia de Hamming cuya característica es que se aplica a vectores del mismo tamaño y cuando tenemos vectores de distintos tamaños podemos rellenar el vector de menor dimensión con datos vacíos hacia la derecha para compensar. La distancia de Hamming se define como el número de entradas en las mismas posiciones que son distintas, entre más cerca del cero más parecidos son los vectores:

```
library(e1071)
a<-c('a','b','c')
b<-c('c','b','a')
print('distancia es')
```

```
## [1] "distancia es"
```

```
hamming.distance(a,b)
```

```
## [1] 2
```

Podemos observar que la distancia entre los vectores a y b es 2 ya que las entradas 1 y 3 son distintas. existen otras métricas sobretodo para el caso de datos categóricos como la distancia de Levenshtein, Gower.

Numéricos de escala de intervalo y Ordinales: Para el caso de las variables númericas de intevalo establecen un orden o jerarquía entre categorías y las distancias entre cada intervalo son iguales entonces podemos tratar ambos tipos de variables (ordinal y de intervalo) de la misma forma en tema de distancias y de acuerdo con la

definición de las distancias de Gower podemos definir la distancia entre 2 observaciones de una misma variable como el valor absoluto de la diferencia entre ellas entre el rango de la variable:

```
library(cluster)
cc<-ordered(c(1,9,2),levels=c(1:10))
cc
```

```
## [1] 1 9 2
## Levels: 1 < 2 < 3 < 4 < 5 < 6 < 7 < 8 < 9 < 10
```

```
daisy(cbind(cc),metric="gower")
```

```
## Dissimilarities :
##       1     2
## 2 1.000
## 3 0.125 0.875
##
## Metric :  mixed ;  Types = I
## Number of objects : 3
```

utilizamos la funcion daisy que está en la librería cluster para mostrar como se dan estas distancias, ejemplo la distancia de la observación 1 a la 3 es 0.125 y se calculó haciendo |1-2|/8 ya que 8 es el rango de la columna.

ahora si vemos los datos como vector…

```
aa<-ordered(c(2,4,5),levels=c(1:10))
bb<-ordered(c(5,7,2),levels=c(1:8))
zw<-cbind(aa,bb,cc)
zw
```

```
##      aa bb cc
## [1,]  2  5  1
## [2,]  4  7  9
## [3,]  5  2  2
```

```
daisy(zw,metric="gower")
```

```
## Dissimilarities :
##           1         2
## 2 0.6888889
## 3 0.5750000 0.7361111
##
## Metric :  mixed ;  Types = I, I, I
## Number of objects : 3
```

entonces para el cálculo de la distancia entre el primer y tercer individuo se hicieron las diferencias en cada variable por tanto |2-5|/3=1, |5-2|/5=0.6,|1-2|/8=0.125, y promediando…tenemos que la distancia es 0.575 como lo muestra la función.