

clase_1

peter

27-12-2020

El tidyverse

Estructura de datos en R

Vectores: Un conjunto lineal de datos. Matrix: Una tabla con solo numeros. Data Frame: Una tabla donde cada columna tiene un tipo de dato. List: Aqui puede contener las anteriores categorias.

Vector

Secuencia lineal de datos, puede ser de muchos tipos (numéricos, caracteres, lógico, etc.) Ejemplo:

```
data(uspop)
uspop
```

```
## Time Series:
## Start = 1790
## End = 1970
## Frequency = 0.1
## [1] 3.93 5.31 7.24 9.64 12.90 17.10 23.20 31.40 39.80 50.20
## [11] 62.90 76.00 92.00 105.70 122.80 131.70 151.30 179.30 203.20
```

El comando “data()” carga base de datos que ya se encuentran en R.

Otro ejemplo de vector se consigue así:

```
# crear un vector x
x <- c(2,3,4,5,3,4,6,3,7,6,4)
x
```

```
## [1] 2 3 4 5 3 4 6 3 7 6 4
```

A este vector x creado, se le pueden aplicar varias funciones

```
mean(x) # media
```

```
## [1] 4.272727
```

```
sd(x) # desviación estandar
```

```
## [1] 1.55505
```

```
x * 4 # multiplica cada valor de x por 4
```

```
## [1] 8 12 16 20 12 16 24 12 28 24 16
```

El sub-seteo de los vectores se consigue de varias maneras, una de las mas simples se muestran a continuación:

```
uspop[5] # obtiene el 5to elemento
```

```
## [1] 12.9
```

```
uspop[c(5,8)] # obtiene el 5to y 8vo elemento
```

```
## [1] 12.9 31.4
```

```
uspop[5:8] # obtiene los valores del 5to al 8vo elemento
```

```
## [1] 12.9 17.1 23.2 31.4
```

Ademas, se puede crear otros vectores a partir del sub-seteo de un vector.

```
# creación de un vector y a partir del vector x
```

```
y <- x[c(5,8)] # crea el vector y con 2 elementos de vector x.
```

Data Frame

Una tabla, donde cada columna es un tipo de dato (numérico, lógico, etc.). Cada columna es un vector. Ejemplo:

```
data("iris")
```

```
View(iris) # abre iris en una ventana
```

Con un data.frame se puede trabajar seleccionando columnas mediante el signo "\$" luego del nombre del data, y escribiendo el nombre de la columna.

```
iris$Sepal.Length # selecciona la columna Sepal.Length y la muestra como vector.
```

```
## [1] 5.1 4.9 4.7 4.6 5.0 5.4 4.6 5.0 4.4 4.9 5.4 4.8 4.8 4.3 5.8 5.7 5.4 5.1
## [19] 5.7 5.1 5.4 5.1 4.6 5.1 4.8 5.0 5.0 5.2 5.2 4.7 4.8 5.4 5.2 5.5 4.9 5.0
## [37] 5.5 4.9 4.4 5.1 5.0 4.5 4.4 5.0 5.1 4.8 5.1 4.6 5.3 5.0 7.0 6.4 6.9 5.5
## [55] 6.5 5.7 6.3 4.9 6.6 5.2 5.0 5.9 6.0 6.1 5.6 6.7 5.6 5.8 6.2 5.6 5.9 6.1
## [73] 6.3 6.1 6.4 6.6 6.8 6.7 6.0 5.7 5.5 5.5 5.8 6.0 5.4 6.0 6.7 6.3 5.6 5.5
## [91] 5.5 6.1 5.8 5.0 5.6 5.7 5.7 6.2 5.1 5.7 6.3 5.8 7.1 6.3 6.5 7.6 4.9 7.3
## [109] 6.7 7.2 6.5 6.4 6.8 5.7 5.8 6.4 6.5 7.7 7.7 6.0 6.9 5.6 7.7 6.3 6.7 7.2
## [127] 6.2 6.1 6.4 7.2 7.4 7.9 6.4 6.3 6.1 7.7 6.3 6.4 6.0 6.9 6.7 6.9 5.8 6.8
## [145] 6.7 6.7 6.3 6.5 6.2 5.9
```

En los data.frame se puede seleccionar por filas y columnas mediante [], así luego del nombre del data.frame[filas,columnas]

```
iris[1,] # a la derecha de la coma indica las filas
```

```
## Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1 5.1 3.5 1.4 0.2 setosa
```

```
iris[c(1,4,6),] # selecciona las filas 1, 4 y 6
```

```
## Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1 5.1 3.5 1.4 0.2 setosa
## 4 4.6 3.1 1.5 0.2 setosa
## 6 5.4 3.9 1.7 0.4 setosa
```

```
iris[seq(5,150, by = 5),] # selecciona filas en secuencia de la 5 a la 150 de 5 en 5.
```

```
## Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 5 5.0 3.6 1.4 0.2 setosa
## 10 4.9 3.1 1.5 0.1 setosa
## 15 5.8 4.0 1.2 0.2 setosa
## 20 5.1 3.8 1.5 0.3 setosa
## 25 4.8 3.4 1.9 0.2 setosa
## 30 4.7 3.2 1.6 0.2 setosa
```

```
## 35      4.9      3.1      1.5      0.2      setosa
## 40      5.1      3.4      1.5      0.2      setosa
## 45      5.1      3.8      1.9      0.4      setosa
## 50      5.0      3.3      1.4      0.2      setosa
## 55      6.5      2.8      4.6      1.5      versicolor
## 60      5.2      2.7      3.9      1.4      versicolor
## 65      5.6      2.9      3.6      1.3      versicolor
## 70      5.6      2.5      3.9      1.1      versicolor
## 75      6.4      2.9      4.3      1.3      versicolor
## 80      5.7      2.6      3.5      1.0      versicolor
## 85      5.4      3.0      4.5      1.5      versicolor
## 90      5.5      2.5      4.0      1.3      versicolor
## 95      5.6      2.7      4.2      1.3      versicolor
## 100     5.7      2.8      4.1      1.3      versicolor
## 105     6.5      3.0      5.8      2.2      virginica
## 110     7.2      3.6      6.1      2.5      virginica
## 115     5.8      2.8      5.1      2.4      virginica
## 120     6.0      2.2      5.0      1.5      virginica
## 125     6.7      3.3      5.7      2.1      virginica
## 130     7.2      3.0      5.8      1.6      virginica
## 135     6.1      2.6      5.6      1.4      virginica
## 140     6.9      3.1      5.4      2.1      virginica
## 145     6.7      3.3      5.7      2.5      virginica
## 150     5.9      3.0      5.1      1.8      virginica
```

También se puede realizar una serie de tareas adicionales, en el siguiente bloque se creará un vector a partir de un data.frame, luego de ese vector se crearán 3 vectores más uno por cada especie, para luego obtener la media y desviación estándar de cada especie.

```
# Haré un vector con el largo de petalo
largo_petal <- iris$Petal.Length
largo_petal

##      [1] 1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 1.5 1.6 1.4 1.1 1.2 1.5 1.3 1.4
##     [19] 1.7 1.5 1.7 1.5 1.0 1.7 1.9 1.6 1.6 1.5 1.4 1.6 1.6 1.5 1.5 1.4 1.5 1.2
##     [37] 1.3 1.4 1.3 1.5 1.3 1.3 1.3 1.6 1.9 1.4 1.6 1.4 1.5 1.4 4.7 4.5 4.9 4.0
##     [55] 4.6 4.5 4.7 3.3 4.6 3.9 3.5 4.2 4.0 4.7 3.6 4.4 4.5 4.1 4.5 3.9 4.8 4.0
##     [73] 4.9 4.7 4.3 4.4 4.8 5.0 4.5 3.5 3.8 3.7 3.9 5.1 4.5 4.5 4.7 4.4 4.1 4.0
##     [91] 4.4 4.6 4.0 3.3 4.2 4.2 4.2 4.3 3.0 4.1 6.0 5.1 5.9 5.6 5.8 6.6 4.5 6.3
##    [109] 5.8 6.1 5.1 5.3 5.5 5.0 5.1 5.3 5.5 6.7 6.9 5.0 5.7 4.9 6.7 4.9 5.7 6.0
##   [127] 4.8 4.9 5.6 5.8 6.1 6.4 5.6 5.1 5.6 6.1 5.6 5.5 4.8 5.4 5.6 5.1 5.1 5.9
##   [145] 5.7 5.2 5.0 5.2 5.4 5.1

# con ese vector podemos separarlo según especies
largo_setosa <- largo_petal[1:50]
largo_versicolor <- largo_petal[51:100]
largo_virginica <- largo_petal[101:150]

# aplicando funciones a los vectores (media y desviación estándar)
mean(largo_setosa); sd(largo_setosa)

## [1] 1.462
## [1] 0.173664

mean(largo_versicolor); sd(largo_versicolor)
```

```
## [1] 4.26
```

```
## [1] 0.469911
```

```
mean(largo_virginica); sd(largo_virginica)
```

```
## [1] 5.552
```

```
## [1] 0.5518947
```