Leonardo Daniel Rivera Valdez

Matrices

Los vectores solamente corren en una dimensión, sin embargo las matrices lo hacen en 2 dimensiones. Como lo muestra la siguiente imagen

Como se ve, arriba tenemos un vector que corre hacia la derecha con valores de 1,2,3,4,5. Sin embargo nuestra matriz lo hace también hacia abajo, generando así un plano, o una tabla de valores.

Luego, si uno quiere acceder a un valor de la matriz uno se topa con un problema, con el hecho de que hay tanto filas como columnas, y por ende uno no puede indicar el valor deseado como se hacía con un vector. Si uno pone A[3], entonces es ambigua la instrucción ya que hay N valores en 3 como columna o 3 como fila; dependiendo de las dimensiones de nuestra matriz. Esto se ve en la siguiente imagen:

Si uno quiere seleccionar un valor se debe dar dos posiciones, una relativa a las columnas, y otra a las filas. En el caso de arriba se ve que para indicar el cero uno debe dar las coordenadas A[3, 4]. El primero indica la fila, y el segundo la columna, esto se llama indexación.

La manera en que R acomoda esto es, como se ve en la imagen de arriba, tal que las filas se expresan como [x,] y las columnas como [,y]. Si uno solamente indica [x,] o [,y] uno se refiere a todos los valores de la fila o la columna, un vector en esencia. Una última cosa es

que las matrices solamente aceptan datos de un solo tipo, ya sea double, integer, character, etc.

Construyendo tu primer matriz

Hay varias formas de generar matrices, una de estas es mediante el uso de la función **matrix()**. Lo que hace es tomar un vector y acomodarlo en sentido de una matriz, lo cual puede no ser muy bueno porque se pierde el orden de nuestros datos. Si tenemos un vector de 1:12, entonces:

1	4	7	10
2	5	8	11
3	6	9	12

Otra manera es mediante la función **rbind()**, r es de row, esta une los vectores de tal manera que un vector se toma como fila y se van apilando formando la matriz. Si fuese más corto un vector que otro se aplica la ley del reciclado de vectores.

Fila 1	Fila 1	Fila 1	Fila 1
Fila 2	Fila 2	Fila 2	Fila 2
Fila 3	Fila 3	Fila 3	Fila 3

Una última es mediante **cbind()**, c es de column, y lo que hace es que une los vectores tomándolos como columnas. Igualmente aplica la ley del reciclado.

Columna 1	Columna 2	Columna 3	Columna 4
Columna 1	Columna 2	Columna 3	Columna 4
Columna 1	Columna 2	Columna 3	Columna 4

matriz usando **matrix()**, te pide nrow y ncol

```
mis.datos <- 1:25
mis.datos

#Ojo, es importante que el número de datos sea igual a nrow * ncol
M <- matrix(mis.datos, nrow = 5, ncol = 5)
M

#Si quisiéramos un valor de esta matriz usamos lo siguiente
M[1,3]
M[2,4]
M[1.1]
M[3,4]
```

#matriz usando **rbind()**

```
v1 <- c(1,2,3,4,5)
v2 <- c(10,11,12,13,14)
v3 <- c(20,21,22,23,24)

M2 <- rbind(v1,v2, v3)
M2
```

#matriz con **cbind()**

```
M3 <- cbind(v1,v2,v3)
M3
```

Nombrando las dimensiones

Uno puede ponerle nombres a las columnas. Supongamos que hacemos lo siguiente

	“A”	“B”	“C”	“D”
“Dinosaurio”	Dino 1	Dino 2	Dino 3	Dino 4
“Ave”	Ave 1	Ave 2	Ave 3	Ave 4
“Reptil”	Reptil 1	Reptil 2	Reptil 3	Reptil 4
“Mamífero”	Mamífero 1	Mamífero 2	Mamífero 3	Mamífero 4

Hemos nombrado las filas y columnas, si uno quisiera obtener el mamífero 3 entonces uno pondría M[“Mamífero”, C]. Igual que como se hacía cuando nuestras filas y columnas eran números y un ponía la coordenada entre [X, Y] para obtener x o y elemento de la matriz. Es importante mencionar que si nuestra matriz ya tenía previamente otros nombres o símbolos asignados a sus columnas y filas, entonces se pueden seguir usando. Incluso se pueden combinar con los nuevos, uno podría poner M[“Mamífero”, 3] y sería correcto si previamente teníamos número de en lugar de nuestras letras.

También se puede aplicar a los vectores como el siguiente

“A”	“B”	“C”	“D”
Dino 1	Dino 2	Dino 3	Dino 4

Aquí uno pondría V[“B”] si quisiera acceder al elemento Dino 3

Colnames() y rownames()

```
#Primero hacemos un vector
Pau <- 1:5
Pau

#Aquí se hace que el vector Pau siempre sea acompañado por el vector names()
names(Pau) <- c("a", "b", "c", "d", "e")

#Ahora cuando se corre Pau se obtiene que el vector original de 1:5 esté acompañado por c("a", "b", "c", "d", "e")

Pau

#También podemos quitar los nombres acompañantes mediante NULL
names(Pau) <- NULL
Pau

#Ahora a nombrar matrices
#Se almacena primero un vector simple en un objeto
V1 <- rep(c("a", "b", "c", "d", "e"), each = 3)

#Luego se vuelve este una matriz, se especifican las dimensiones y se almacena como otro objeto
Dennett <- matrix(V1, 3, 3)
Dennett

#Posteriormente le ponemos los nombres a las filas y columnas por las funciones rownames() y colnames() respectivamente
rownames(Dennett) <- c("Dinosaurio", "Mamífero", "Reptil")
colnames(Dennett) <- c("A", "B", "C")
Dennett
```

```
#Aplicando el principio que se mencionó antes podemos tomar cosas de la matriz
aprovechando todos los nombres que han tenido las filas y columnas
Dennett["Reptil", "A"]
Dennett[3,1]

#Como lo hicimos anteriormente, igualmente se pueden quitar los nombres por medio de
NULL
rownames(Dennett) <- NULL
colnames(Dennett) <- NULL
```

Operaciones con matrices

#En el anexo vienen los datos que se usan en el curso. No son míos y todos los derechos son de www.superdatascience.com.

#Las operaciones en R son muy simples, como con los vectores solamente necesitamos el símbolo

```
tiros_anotados/juegos
round(tiros_anotados /juegos, 1)
```

#Esto solamente lo hice para ir analizando los datos de la matriz más a fondo

```
mean(tiros_anotados)
max(tiros_anotados)
min(tiros_anotados)
```

```
minutos_jugados
round(minutos_jugados/juegos, 1)
```

```
mean(minutos_jugados)
max(minutos_jugados)
min(minutos_jugados)
```

#También, aunque no tiene mucho sentido para estos datos, es posible hacer otro tipo de operaciones

```
minutos_jugados * tiros_anotados
sqrt(minutos_jugados)
minutos_jugados + tiros_anotados
```

Visualizando con matplot()

1. Solo tiros anotados

t() nos permite transponer matrices, que no es otra cosa que poner lo que está como filas como columnas y viceversa

x <- t(tiros_anotados)

#Una vez con la matriz transpuesta podemos usar matplot() que la necesita así

#El **type=** indica que vamos a usar puntos y líneas

#**pch=** indica el tipo puntos a usar

#**col=** finalmente nos dicta los colores, donde 1:4 indica el rango de los colores a usar (son cíclicos por eso se especifica)

matplot(x, type="b", pch=15:18, col=c(1:4, 6))

#Luego, como nuestros colores no indican el jugador necesitamos una leyenda que lo marque

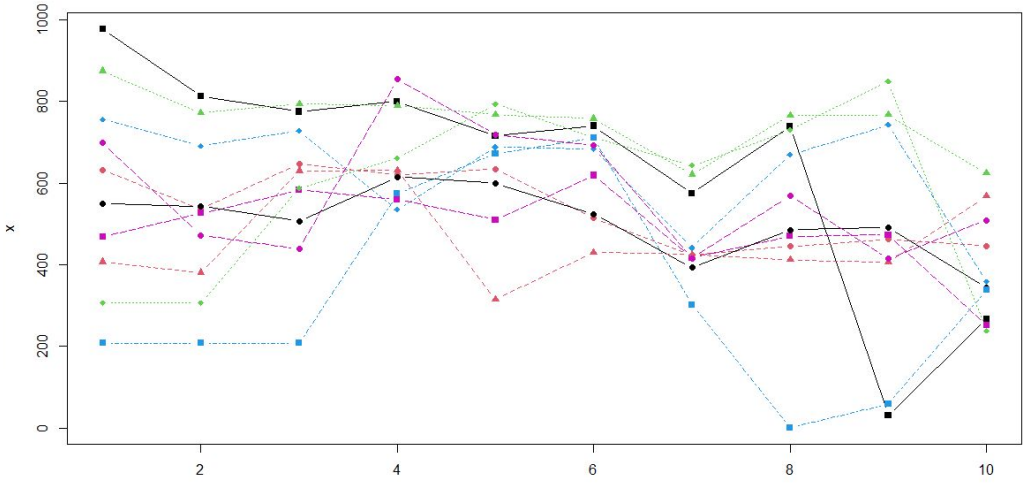
#Primero se pone la posición

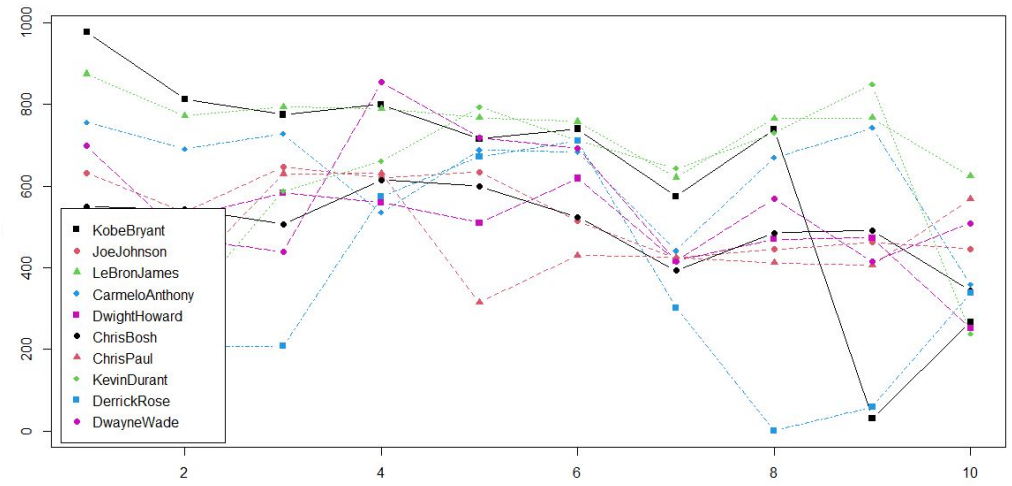
#Posteriormente se pone **inset=** para indicar la distancia de la posición inicial indicada previamente y lo que se quiere que indique la leyenda

#Luego ponemos los colores de manera idéntica al **matplot()**, además del pch

#Finalmente solamente se indica la posición

legend("bottomleft", inset = .01, legend = jugadores, col=c(1:4, 6),pch=15:18, horiz = F)



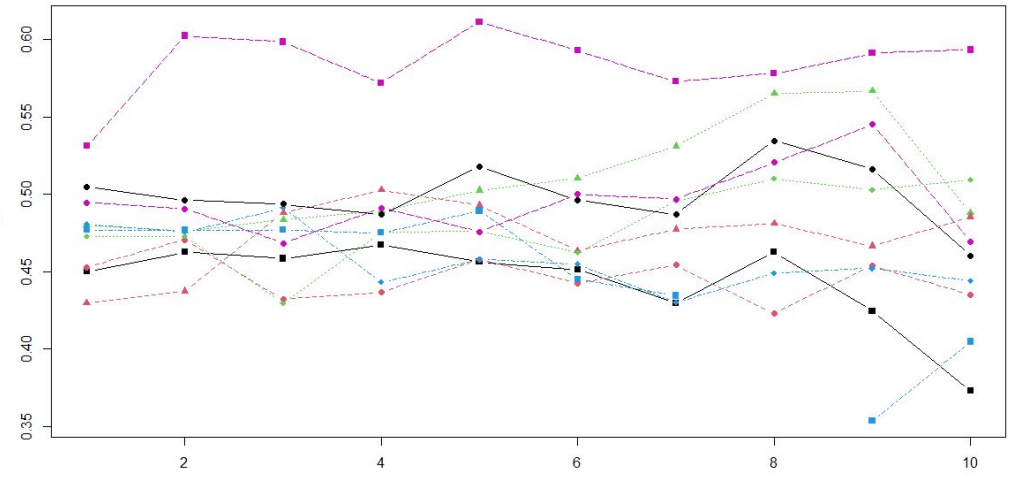


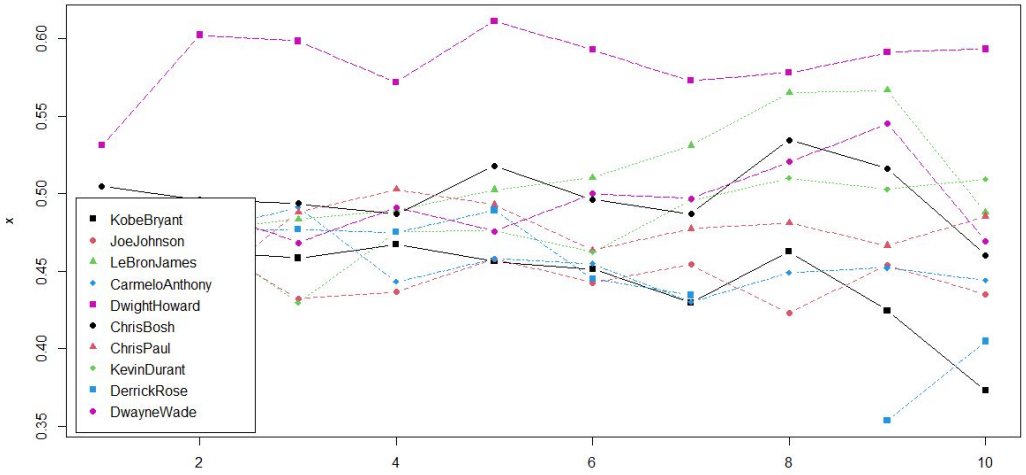
2. Tiros anotados/tiros intentados

```
# t() nos permite transponer matrices, que no es otra cosa que poner lo que está como filas como columnas y viceversa
x <- t(tiros_anotados/tiros_intentados)

#Una vez con la matriz transpuesta podemos usar matplot() que la necesita así
#El type= indica que vamos a usar puntos y líneas
#pch= indica el tipo puntos a usar
#col= finalmente nos dicta los colores, donde 1:4 indica el rango de los colores a usar (son cíclicos por eso se especifica)
matplot(x, type="b", pch=15:18, col=c(1:4, 6))

#Luego, como nuestros colores no indican el jugador necesitamos una leyenda que lo marque
#Primero se pone la posición
#Posteriormente se pone inset= para indicar la distancia de la posición inicial indicada previamente y lo que se quiere que indique la leyenda
#Luego ponemos los colores de manera idéntica al matplot(), además del pch
#Finalmente solamente se indica la posición
legend("bottomleft", inset = .01, legend = jugadores, col=c(1:4, 6),pch=15:18, horiz = F)
```



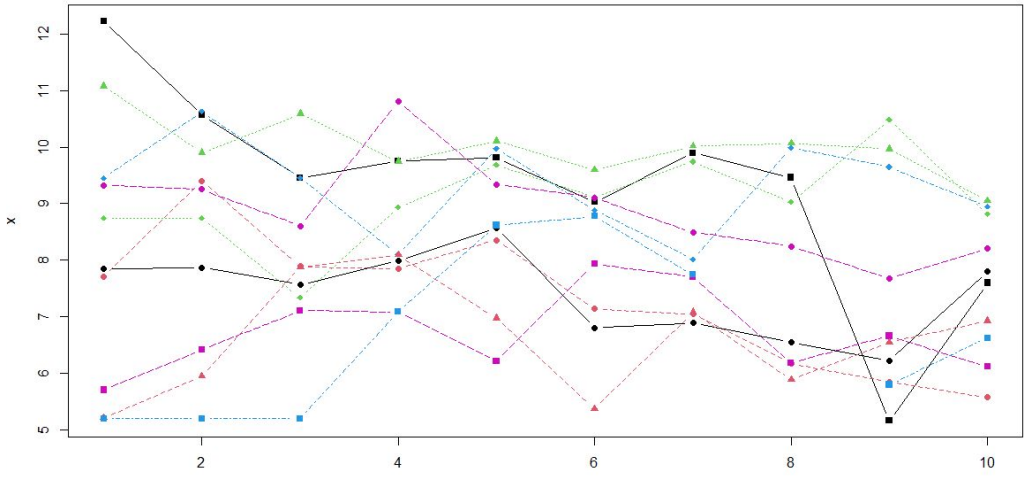


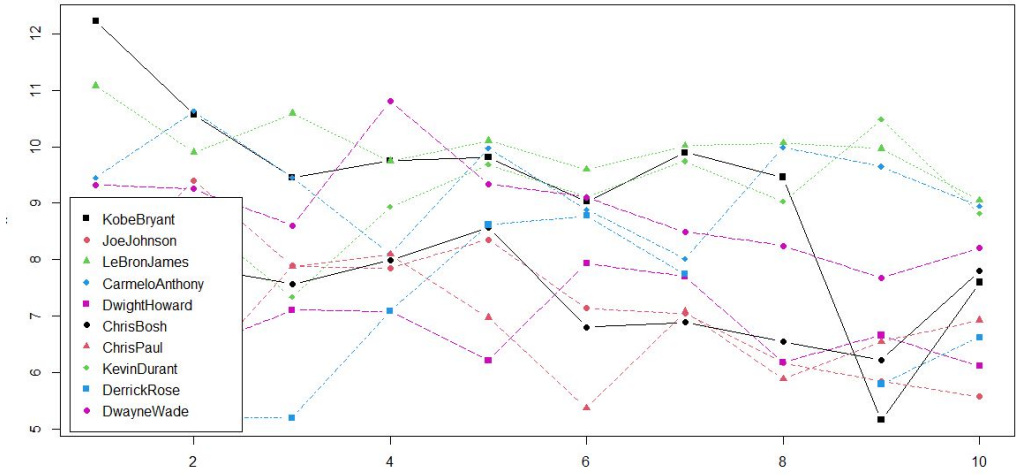
3. Tiros anotados/juegos

```
# t() nos permite transponer matrices, que no es otra cosa que poner lo que está como filas como columnas y viceversa
x <- t(tiros_anotados/juegos)

#Una vez con la matriz transpuesta podemos usar matplot() que la necesita así
#El type= indica que vamos a usar puntos y líneas
#pch= indica el tipo puntos a usar
#col= finalmente nos dicta los colores, donde 1:4 indica el rango de los colores a usar (son cíclicos por eso se especifica)
matplot(x, type="b", pch=15:18, col=c(1:4, 6))
```

#Luego, como nuestros colores no indican el jugador necesitamos una leyenda que lo marque
#Primero se pone la posición
#Posteriormente se pone inset= para indicar la distancia de la posición inicial indicada previamente y lo que se quiere que indique la leyenda
#Luego ponemos los colores de manera idéntica al matplot(), además del pch
#Finalmente solamente se indica la posición
legend("bottomleft", inset = .01, legend = jugadores, col=c(1:4, 6),pch=15:18, horiz = F)





Subconjuntos de datos

#Subconjuntos (subsettings), literalmente son conjuntos formados de un conjunto de datos, ya sea un vector o una matriz

```
vector <- c(1,2,3,4,5,6,7,8,9,10)
vector

#Por medio de nombre del vector y las posiciones uno puede generar un subconjunto del vector
vector[c(1,4)]
vector[c(2,3,5,7)]
vector[1]

#Mismo principio aplica para las matrices, pero hay que especificar las filas y columnas
juegos
juegos[1:3,5:10]
juegos[1:2, 1:2]
juegos[1:5, 6:8]
juegos[, c("2007", "2010")]
juegos[,c("2005", "2007")]
juegos[c("KobeBryant", "LeBronJames"),]
juegos[c("ChrisPaul", "DwayneWade", "CarmeloAnthony"),]

#Pero si uno hace lo siguiente uno transforma su subconjunto en un vector a partir de una matriz. Todos los anteriores tomaban o la matriz o el vector y el subconjunto era del mismo tipo

juegos[1,]

is.matrix(juegos[1,])
is.matrix(juegos[1,1])
```

#Esto es debido a que damos solamente una dimensión

#Para evitar esto se hace lo siguiente y nos dará una matriz nuevamente

juegos[1,,**drop=FALSE**]

is.matrix(juegos[1,,drop=FALSE])

juegos[1,1,,drop=F]

is.matrix(juegos[1,1,drop=FALSE])

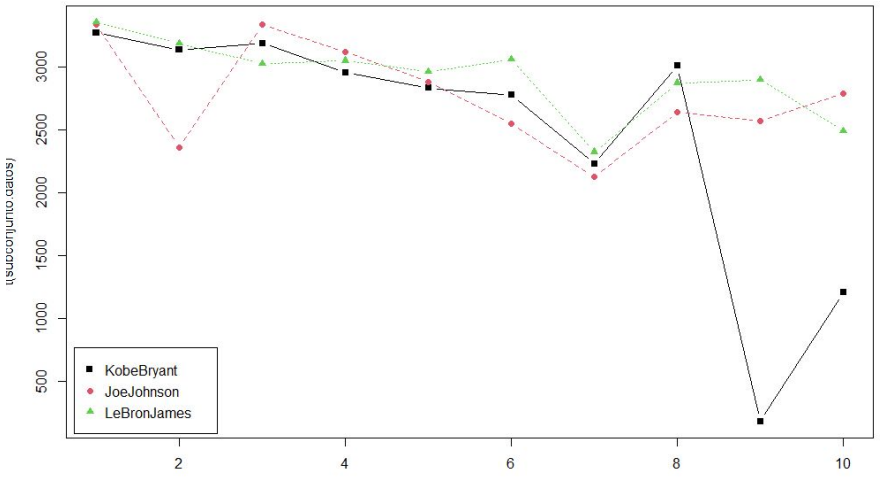
Visualizando los subconjuntos

#Visualizar los subconjunos

subconjunto.datos <- minutos_jugados[1:3,]

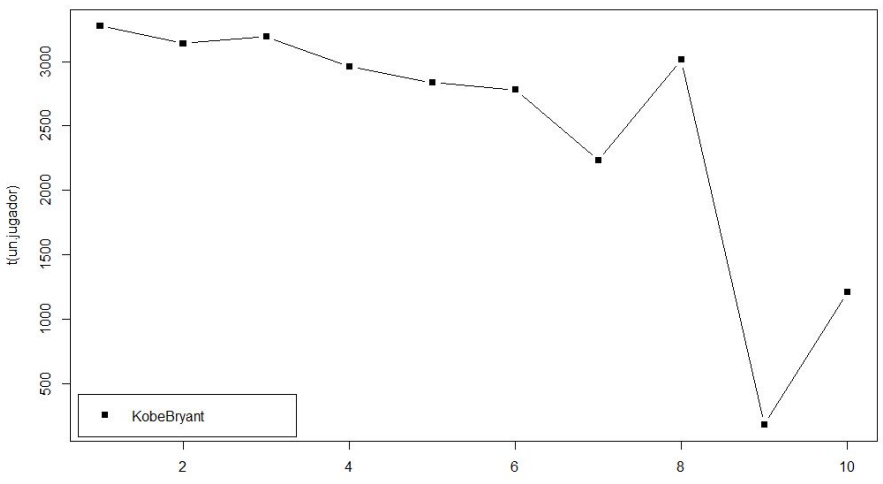
matplot(t(subconjunto.datos), type = "b", pch = 15:18, col = c(1:4, 6))

legend("bottomleft", inset = .01, legend = jugadores[1:3], pch = 15:18, col = c(1:4, 6))



#Si quisiéramos solamente ver al primer jugador. Es importante aquí indicar por medio de drop que no es un vector sino una matriz, esto debido a que matplot() necesita matrices para funcionar

```
un.jugador <- minutos_jugados[1,,drop=FALSE]
matplot(t(un.jugador), type = "b", pch = 15:18, col = c(1:4, 6) )
legend("bottomleft",inset = .01, legend = jugadores[1], pch = 15:18, col = c(1:4, 6))
```



Creando funciones

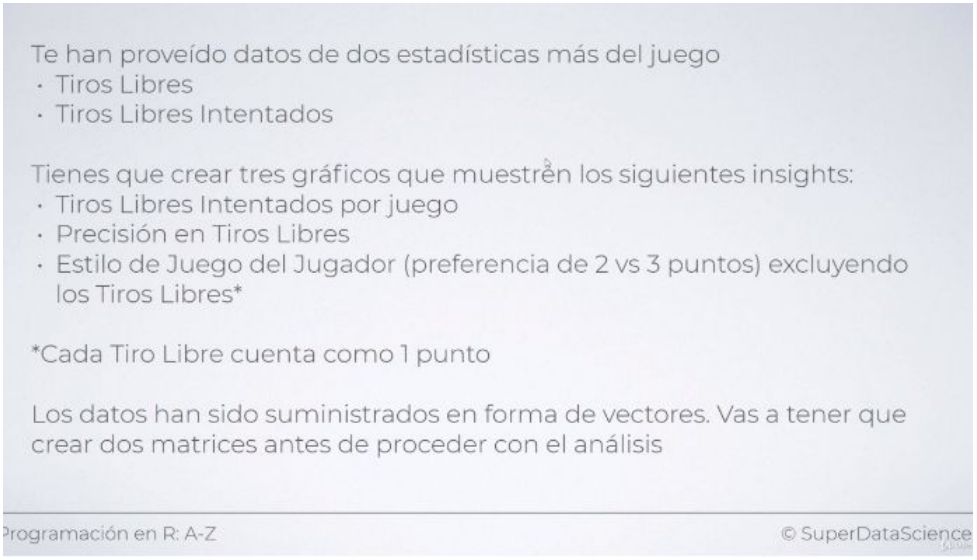
#Una función te permite almacenar todo un código en una frase en que podemos escribir fácilmente.
#Aquí se hace uso de **function(){}** para generar la función
#Se agregan **parámetros en ()** que permiten flexibilidad a la función. De tal modo que uno puede pedir los datos, y las filas a usar. Una vez que las da estas se acoplan al código dado.
#En nuestro caso se toma data y este se almacena en datos y estos luego son transpuestos para poder ser usados por matplot()
#Así mismo se piden filas para que esas sean las tomadas del data y usadas en el código. Además se pone un valor default tal que se toman automáticamente las fila 1:10 si solamente se da el data

```
mi_función <-function(data, filas=1:10){
  datos <- data[filas,,drop=FALSE]
  matplot(t(datos), type = "b", pch = 15:18, col = c(1:4, 6) )
  legend("bottomleft",inset = .01, legend = jugadores[filas], pch = 15:18, col = c(1:4, 6))
}

mi_función(sueldos/juegos, 1)
```

#Esta nos permite hacer cualquiera de las gráficas que hemos hecho antes de manera rápida y sencilla

Práctica basketball de tiros libres



#Primero debemos generar las matrices de tiros libres y tiros libres intentados

#Matriz tiros libres
tiros_libres <- rbind(KobeBryant_TL, JoeJohnson_TL, LeBronJames_TL,
CarmeloAnthony_TL, DwightHoward_TL,
ChrisBosh_TL, ChrisPaul_TL, KevinDurant_TL, DerrickRose_TL,
DwayneWade_TL)
rm (KobeBryant_TL, JoeJohnson_TL, LeBronJames_TL, CarmeloAnthony_TL,
DwightHoward_TL,
ChrisBosh_TL, ChrisPaul_TL, KevinDurant_TL, DerrickRose_TL, DwayneWade_TL)
colnames(tiros_libres) <- temporadas
rownames(tiros_libres) <- jugadores
tiros_libres

#Matriz tiros libres intentados
tiros_libres_intentados <- rbind(KobeBryant_TLI, JoeJohnson_TLI, LeBronJames_TLI,
CarmeloAnthony_TLI, DwightHoward_TLI, ChrisBosh_TLI,
ChrisPaul_TLI, KevinDurant_TLI,
DerrickRose_TLI, DwayneWade_TLI)
rm(KobeBryant_TLI, JoeJohnson_TLI, LeBronJames_TLI,

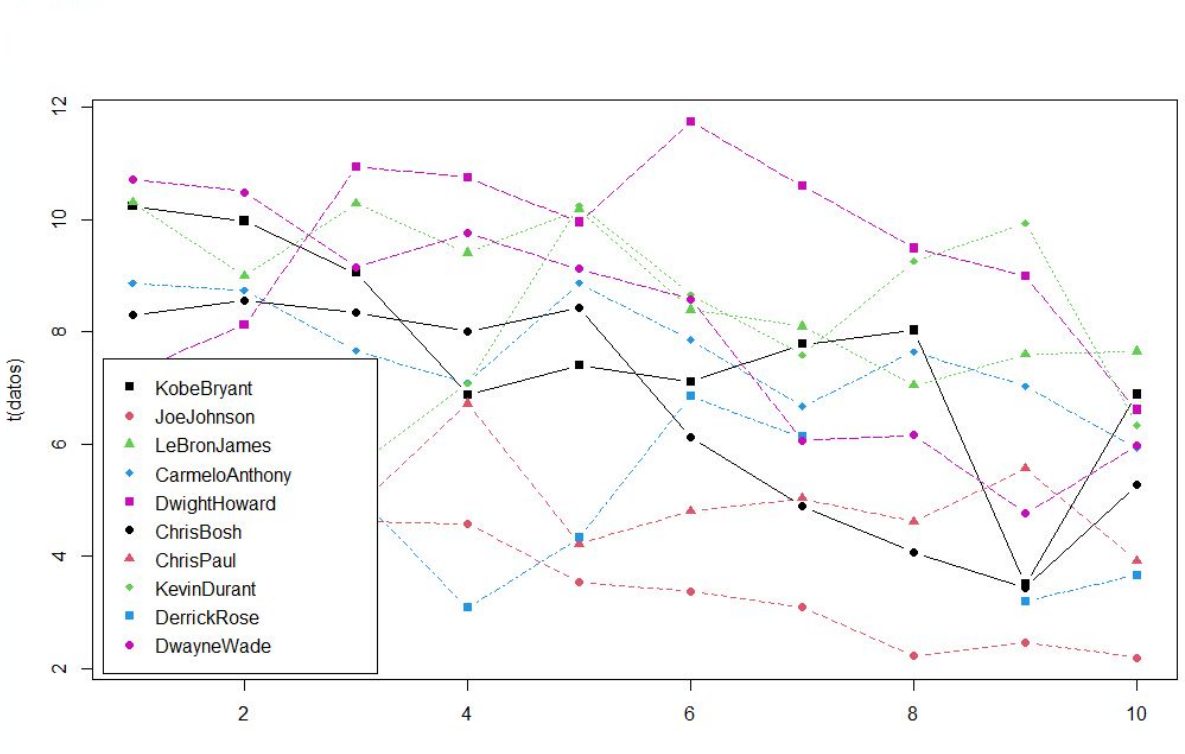
CarmeloAnthony_TLI, DwightHoward_TLI, ChrisBosh_TLI, ChrisPaul_TLI, KevinDurant_TLI,

DerrickRose_TLI, DwayneWade_TLI)

```
colnames(tiros_libres_intentados) <- temporadas
rownames(tiros_libres_intentados) <- jugadores
tiros_libres_intentados
```

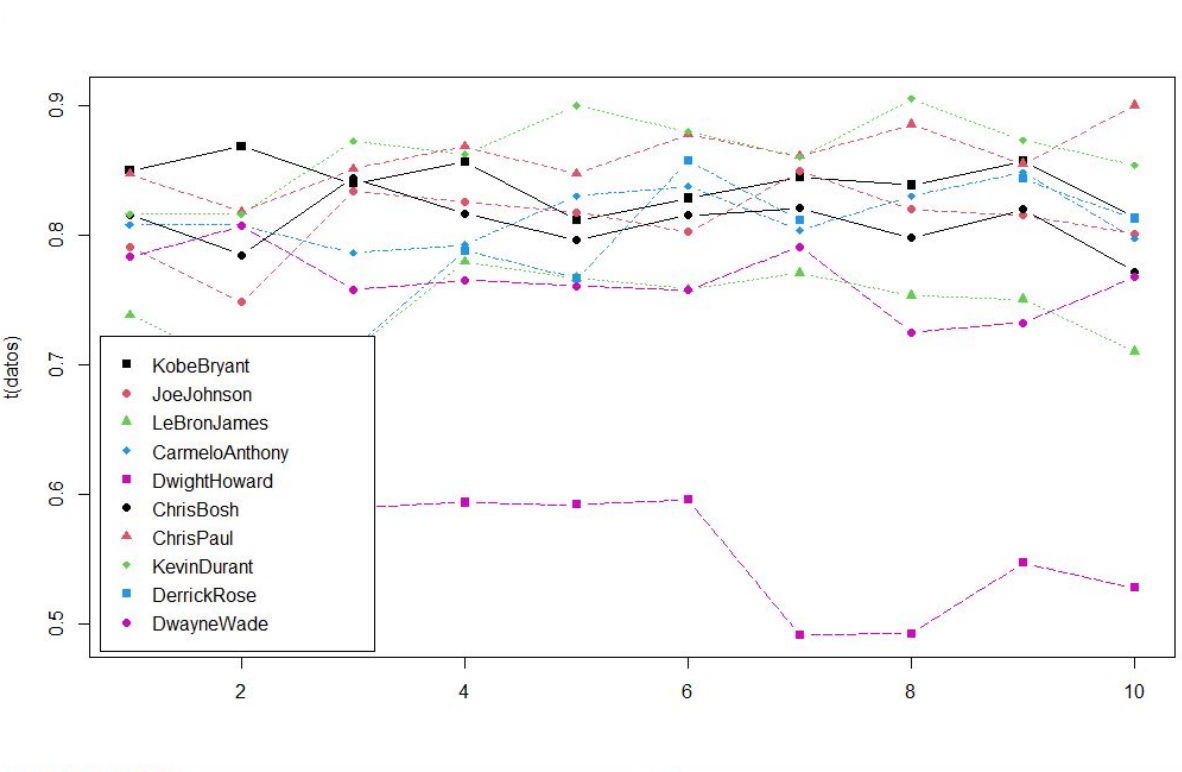
#Tiros libres intentados por juego

```
mi_función(tiros_libres_intentados/juegos)
```



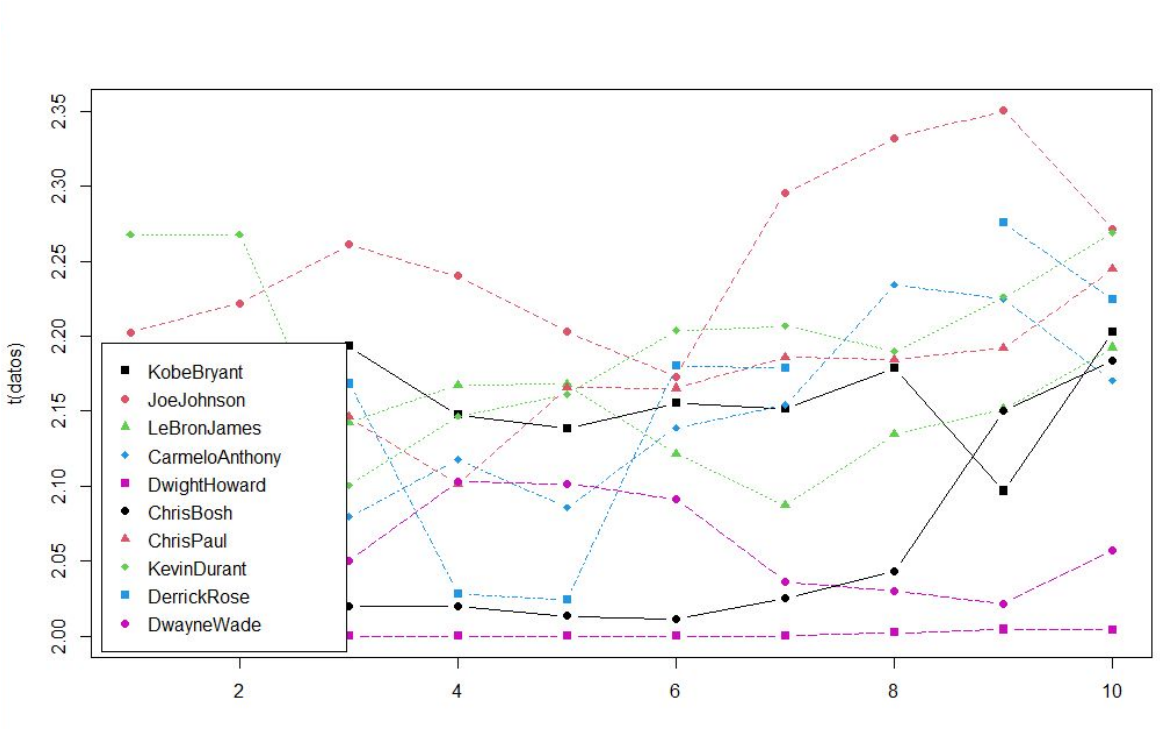
#Precisión en tiros libre

```
mi_función(tiros_libres/tiros_libres_intentados)
```



#Estilo del jugador excluyendo tiros libres

mi_función((puntos - tiros_libres)/tiros_anotados)



Anexos

Esta base de datos no es mía y los scripts pertenecen a www.superdatascience.com. Solamente se deben correr para poder hacer las actividades de las secciones por si alguien quisiera hacerlo. De todos modos es muy recomendable tomar el curso en Udemy (o en la página de http://www.superdatascience.com), es excelente, y no solamente revisar estas notas.

```
#Comentarios:
#Las temporadas están etiquetadas con el primer año de la temporada
#Ejemplo: la temporada 2012-2013 es presentada como 2012
#
# Notas adicionales:
#Kevin Durant: 2006 - Datos de temporada universitaria
#Kevin Durant: 2005 - Creada con datos  del 2006
#Derrick Rose: 2012 - No jugó
#Derrick Rose: 2007 - Datos de temporada universitaria
#Derrick Rose: 2006 - Creada con datos de temporada 2007
#Derrick Rose: 2005 - Creada con datos de temporada 2007



#Temporadas
temporadas <- c("2005","2006","2007","2008","2009","2010","2011","2012","2013","2014")

#Jugadores
jugadores <-
c("KobeBryant","JoeJohnson","LeBronJames","CarmeloAnthony","DwightHoward","ChrisB
osh","ChrisPaul","KevinDurant","DerrickRose","DwayneWade")

#Sueldo
KobeBryant_sueldos <-
c(15946875,17718750,19490625,21262500,23034375,24806250,25244493,27849149,30453
805,23500000)
```

```
JoeJohnson_sueldos <-
c(12000000,12744189,13488377,14232567,14976754,16324500,18038573,19752645,21466
718,23180790)
LeBronJames_sueldos <-
c(4621800,5828090,13041250,14410581,15779912,14500000,16022500,17545000,1906750
0,20644400)
CarmeloAnthony_sueldos <-
c(3713640,4694041,13041250,14410581,15779912,17149243,18518574,19450000,2240747
4,22458000)
DwightHoward_sueldos <-
c(4493160,4806720,6061274,13758000,15202590,16647180,18091770,19536360,20513178,
21436271)
ChrisBosh_sueldos <-
c(3348000,4235220,12455000,14410581,15779912,14500000,16022500,17545000,1906750
0,20644400)
ChrisPaul_sueldos <-
c(3144240,3380160,3615960,4574189,13520500,14940153,16359805,17779458,18668431,2
0068563)
KevinDurant_sueldos <-
c(0,0,4171200,4484040,4796880,6053663,15506632,16669630,17832627,18995624)
DerrickRose_sueldos <-
c(0,0,0,4822800,5184480,5546160,6993708,16402500,17632688,18862875)
DwayneWade_sueldos <-
c(3031920,3841443,13041250,14410581,15779912,14200000,15691000,17182000,1867300
0,15000000)
#Matriz
sueldos <- rbind(KobeBryant_sueldos, JoeJohnson_sueldos, LeBronJames_sueldos,
CarmeloAnthony_sueldos, DwightHoward_sueldos, ChrisBosh_sueldos, ChrisPaul_sueldos,
KevinDurant_sueldos, DerrickRose_sueldos, DwayneWade_sueldos)
rm(KobeBryant_sueldos, JoeJohnson_sueldos, CarmeloAnthony_sueldos,
DwightHoward_sueldos, ChrisBosh_sueldos, LeBronJames_sueldos, ChrisPaul_sueldos,
DerrickRose_sueldos, DwayneWade_sueldos, KevinDurant_sueldos)
colnames(sueldos) <- temporadas
rownames(sueldos) <- jugadores
```

```
#Juegos
KobeBryant_j <- c(80,77,82,82,73,82,58,78,6,35)
JoeJohnson_j <- c(82,57,82,79,76,72,60,72,79,80)
LeBronJames_j <- c(79,78,75,81,76,79,62,76,77,69)
CarmeloAnthony_j <- c(80,65,77,66,69,77,55,67,77,40)
DwightHoward_j <- c(82,82,82,79,82,78,54,76,71,41)
ChrisBosh_j <- c(70,69,67,77,70,77,57,74,79,44)
ChrisPaul_j <- c(78,64,80,78,45,80,60,70,62,82)
KevinDurant_j <- c(35,35,80,74,82,78,66,81,81,27)
DerrickRose_j <- c(40,40,40,81,78,81,39,0,10,51)
DwayneWade_j <- c(75,51,51,79,77,76,49,69,54,62)
#Matriz
juegos = rbind(KobeBryant_j, JoeJohnson_j, LeBronJames_j, CarmeloAnthony_j,
DwightHoward_j, ChrisBosh_j, ChrisPaul_j, KevinDurant_j, DerrickRose_j,
DwayneWade_j)
rm(KobeBryant_j, JoeJohnson_j, CarmeloAnthony_j, DwightHoward_j, ChrisBosh_j,
LeBronJames_j, ChrisPaul_j, DerrickRose_j, DwayneWade_j, KevinDurant_j)
colnames(juegos) <- temporadas
rownames(juegos) <- jugadores

#Minutos Jugados
KobeBryant_mj <- c(3277,3140,3192,2960,2835,2779,2232,3013,177,1207)
JoeJohnson_mj <- c(3340,2359,3343,3124,2886,2554,2127,2642,2575,2791)
LeBronJames_mj <- c(3361,3190,3027,3054,2966,3063,2326,2877,2902,2493)
CarmeloAnthony_mj <- c(2941,2486,2806,2277,2634,2751,1876,2482,2982,1428)
DwightHoward_mj <- c(3021,3023,3088,2821,2843,2935,2070,2722,2396,1223)
ChrisBosh_mj <- c(2751,2658,2425,2928,2526,2795,2007,2454,2531,1556)
ChrisPaul_mj <- c(2808,2353,3006,3002,1712,2880,2181,2335,2171,2857)
KevinDurant_mj <- c(1255,1255,2768,2885,3239,3038,2546,3119,3122,913)
DerrickRose_mj <- c(1168,1168,1168,3000,2871,3026,1375,0,311,1530)
DwayneWade_mj <- c(2892,1931,1954,3048,2792,2823,1625,2391,1775,1971)
#Matriz
```

```
minutos_jugados <- rbind(KobeBryant_mj, JoeJohnson_mj, LeBronJames_mj,
CarmeloAnthony_mj, DwightHoward_mj, ChrisBosh_mj, ChrisPaul_mj, KevinDurant_mj,
DerrickRose_mj, DwayneWade_mj)
rm(KobeBryant_mj, JoeJohnson_mj, CarmeloAnthony_mj, DwightHoward_mj,
ChrisBosh_mj, LeBronJames_mj, ChrisPaul_mj, DerrickRose_mj, DwayneWade_mj,
KevinDurant_mj)
colnames(minutos_jugados) <- temporadas
rownames(minutos_jugados) <- jugadores


#Tiros Anotados
KobeBryant_ta <- c(978,813,775,800,716,740,574,738,31,266)
JoeJohnson_ta <- c(632,536,647,620,635,514,423,445,462,446)
LeBronJames_ta <- c(875,772,794,789,768,758,621,765,767,624)
CarmeloAnthony_ta <- c(756,691,728,535,688,684,441,669,743,358)
DwightHoward_ta <- c(468,526,583,560,510,619,416,470,473,251)
ChrisBosh_ta <- c(549,543,507,615,600,524,393,485,492,343)
ChrisPaul_ta <- c(407,381,630,631,314,430,425,412,406,568)
KevinDurant_ta <- c(306,306,587,661,794,711,643,731,849,238)
DerrickRose_ta <- c(208,208,208,574,672,711,302,0,58,338)
DwayneWade_ta <- c(699,472,439,854,719,692,416,569,415,509)
#Matriz
tiros_anotados <- rbind(KobeBryant_ta, JoeJohnson_ta, LeBronJames_ta,
CarmeloAnthony_ta, DwightHoward_ta, ChrisBosh_ta, ChrisPaul_ta, KevinDurant_ta,
DerrickRose_ta, DwayneWade_ta)
rm(KobeBryant_ta, JoeJohnson_ta, CarmeloAnthony_ta, DwightHoward_ta, ChrisBosh_ta,
LeBronJames_ta, ChrisPaul_ta, DerrickRose_ta, DwayneWade_ta, KevinDurant_ta)
colnames(tiros_anotados) <- temporadas
rownames(tiros_anotados) <- jugadores


#Tiros Intentados
KobeBryant_ti <- c(2173,1757,1690,1712,1569,1639,1336,1595,73,713)
JoeJohnson_ti <- c(1395,1139,1497,1420,1386,1161,931,1052,1018,1025)
LeBronJames_ti <- c(1823,1621,1642,1613,1528,1485,1169,1354,1353,1279)
```

```
CarmeloAnthony_ti <- c(1572,1453,1481,1207,1502,1503,1025,1489,1643,806)
DwightHoward_ti <- c(881,873,974,979,834,1044,726,813,800,423)
ChrisBosh_ti <- c(1087,1094,1027,1263,1158,1056,807,907,953,745)
ChrisPaul_ti <- c(947,871,1291,1255,637,928,890,856,870,1170)
KevinDurant_ti <- c(647,647,1366,1390,1668,1538,1297,1433,1688,467)
DerrickRose_ti <- c(436,436,436,1208,1373,1597,695,0,164,835)
DwayneWade_ti <- c(1413,962,937,1739,1511,1384,837,1093,761,1084)
#Matriz
tiros_intentados <- rbind(KobeBryant_ti, JoeJohnson_ti, LeBronJames_ti,
CarmeloAnthony_ti, DwightHoward_ti, ChrisBosh_ti, ChrisPaul_ti, KevinDurant_ti,
DerrickRose_ti, DwayneWade_ti)
rm(KobeBryant_ti, JoeJohnson_ti, LeBronJames_ti, CarmeloAnthony_ti, DwightHoward_ti,
ChrisBosh_ti, ChrisPaul_ti, KevinDurant_ti, DerrickRose_ti, DwayneWade_ti)
colnames(tiros_intentados) <- temporadas
rownames(tiros_intentados) <- jugadores


#Puntos
KobeBryant_puntos <- c(2832,2430,2323,2201,1970,2078,1616,2133,83,782)
JoeJohnson_puntos <- c(1653,1426,1779,1688,1619,1312,1129,1170,1245,1154)
LeBronJames_puntos <- c(2478,2132,2250,2304,2258,2111,1683,2036,2089,1743)
CarmeloAnthony_puntos <- c(2122,1881,1978,1504,1943,1970,1245,1920,2112,966)
DwightHoward_puntos <- c(1292,1443,1695,1624,1503,1784,1113,1296,1297,646)
ChrisBosh_puntos <- c(1572,1561,1496,1746,1678,1438,1025,1232,1281,928)
ChrisPaul_puntos <- c(1258,1104,1684,1781,841,1268,1189,1186,1185,1564)
KevinDurant_puntos <- c(903,903,1624,1871,2472,2161,1850,2280,2593,686)
DerrickRose_puntos <- c(597,597,597,1361,1619,2026,852,0,159,904)
DwayneWade_puntos <- c(2040,1397,1254,2386,2045,1941,1082,1463,1028,1331)
#Matriz
puntos <- rbind(KobeBryant_puntos, JoeJohnson_puntos, LeBronJames_puntos,
CarmeloAnthony_puntos, DwightHoward_puntos, ChrisBosh_puntos, ChrisPaul_puntos,
KevinDurant_puntos, DerrickRose_puntos, DwayneWade_puntos)
```

```
rm(KobeBryant_puntos, JoeJohnson_puntos, LeBronJames_puntos,
CarmeloAnthony_puntos, DwightHoward_puntos, ChrisBosh_puntos, ChrisPaul_puntos,
KevinDurant_puntos, DerrickRose_puntos, DwayneWade_puntos)
colnames(puntos) <- temporadas
rownames(puntos) <- jugadores
```