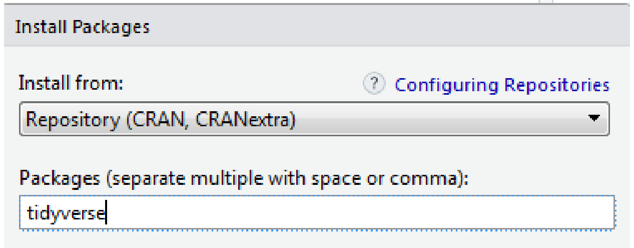
Chapter 10

BASIC DATA PROCESSING (2)

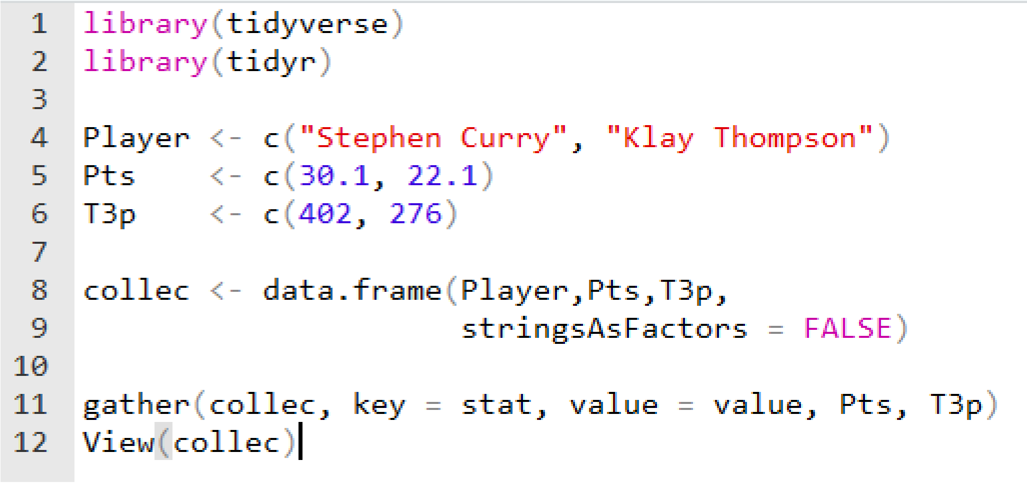
長寬表格轉換 (tydyverse)



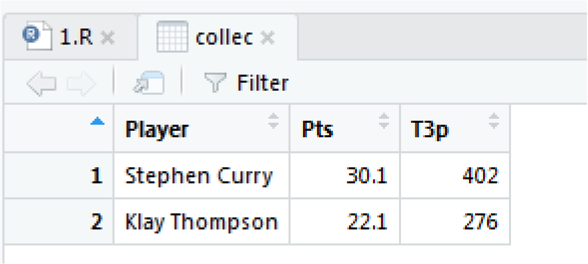
Install Packages

Install from: Repository (CRAN, CRANextra)

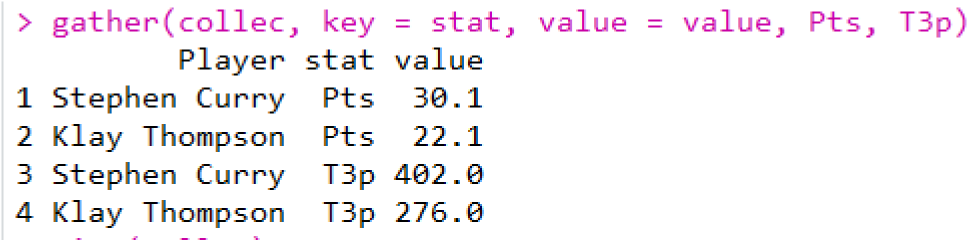
Packages (separate multiple with space or comma): tidyverse



```r
library(tidyverse)
library(tidyr)

Player <- c("Stephen Curry", "Klay Thompson")
Pts    <- c(30.1, 22.1)
T3p    <- c(402, 276)

collec <- data.frame(Player,Pts,T3p,
                     stringsAsFactors = FALSE)

gather(collec, key = stat, value = value, Pts, T3p)
View(collec)
```



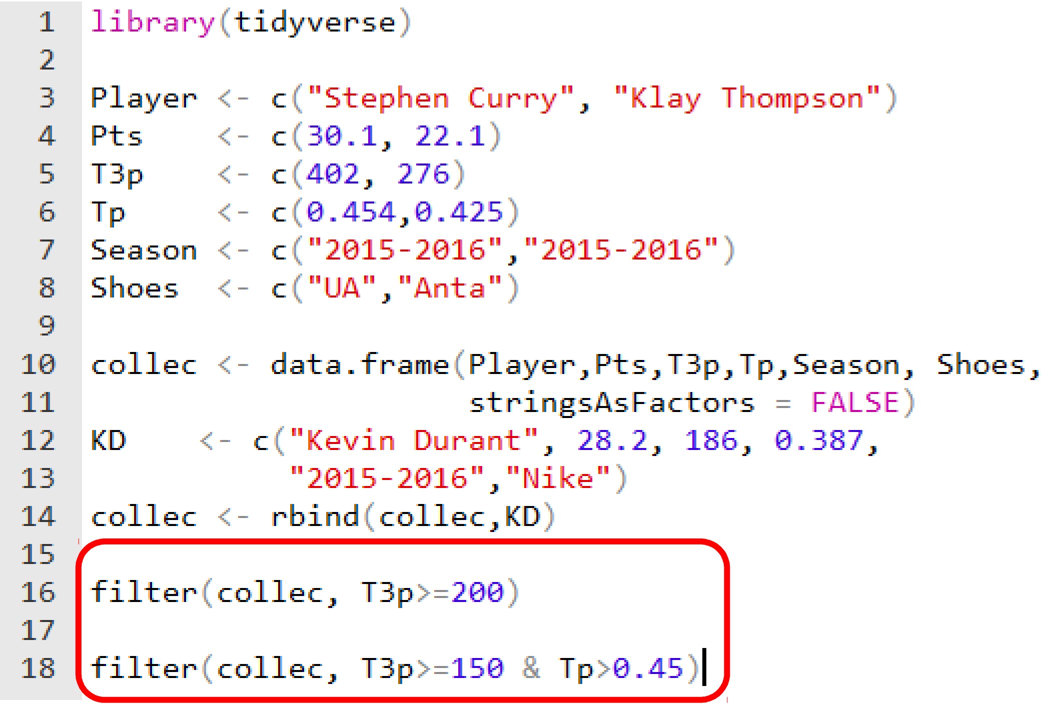
	Player	Pts	T3p
1	Stephen Curry	30.1	402
2	Klay Thompson	22.1	276



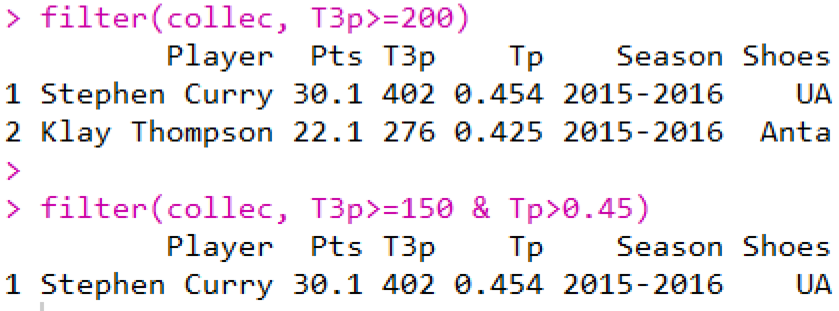
```
> gather(collec, key = stat, value = value, Pts, T3p)
         Player stat value
1 Stephen Curry  Pts  30.1
2 Klay Thompson  Pts  22.1
3 Stephen Curry  T3p 402.0
4 Klay Thompson  T3p 276.0
```

結構化查詢 (tydyverse)

Function	Meaning
filter()	篩選（過濾）
select()	選擇
mutate()	新增
arrange()	排序
summarise()	聚合函數
group_by()	分組



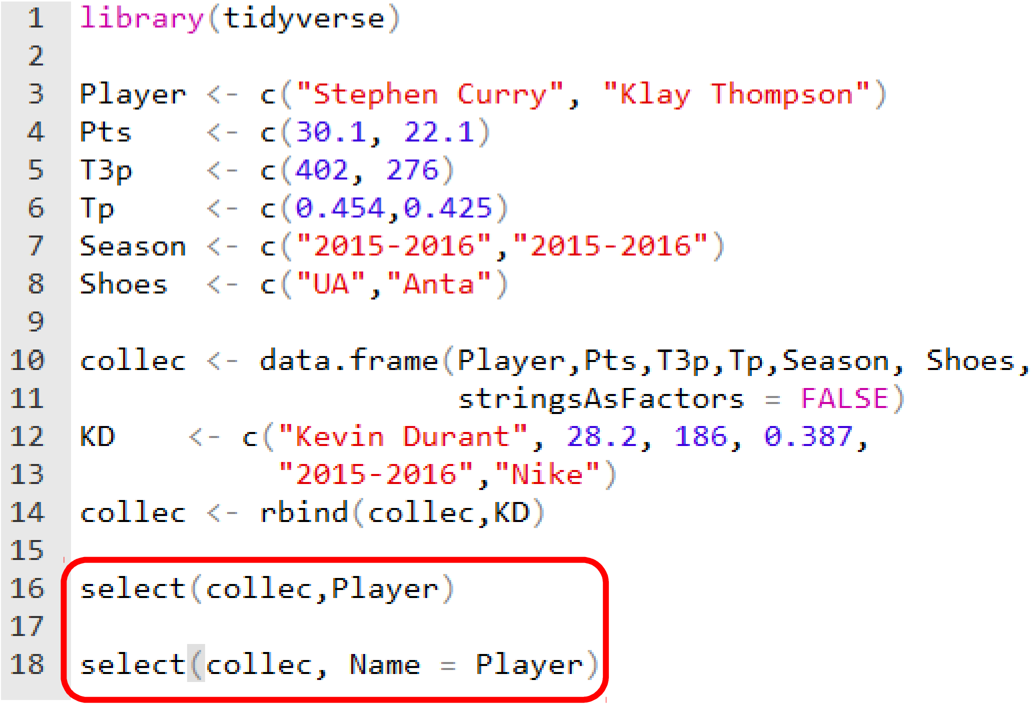
```r
library(tidyverse)

Player <- c("Stephen Curry", "Klay Thompson")
Pts    <- c(30.1, 22.1)
T3p    <- c(402, 276)
Tp     <- c(0.454,0.425)
Season <- c("2015-2016","2015-2016")
Shoes  <- c("UA","Anta")

collec <- data.frame(Player,Pts,T3p,Tp,Season, Shoes,
                     stringsAsFactors = FALSE)
KD    <- c("Kevin Durant", 28.2, 186, 0.387,
           "2015-2016","Nike")
collec <- rbind(collec,KD)

filter(collec, T3p>=200)

filter(collec, T3p>=150 & Tp>0.45)
```



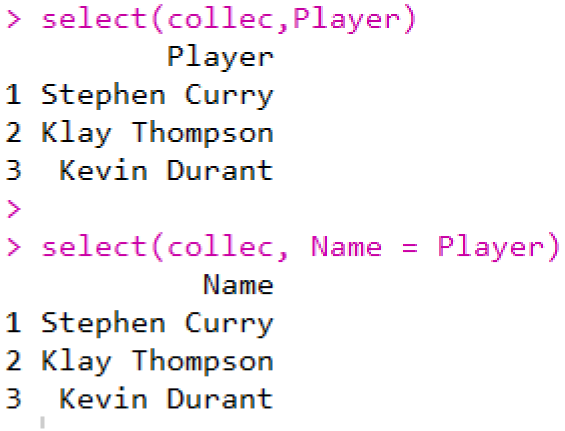
```
> filter(collec, T3p>=200)
         Player  Pts T3p    Tp    Season Shoes
1 Stephen Curry 30.1 402 0.454 2015-2016    UA
2 Klay Thompson 22.1 276 0.425 2015-2016  Anta
> 
> filter(collec, T3p>=150 & Tp>0.45)
         Player  Pts T3p    Tp    Season Shoes
1 Stephen Curry 30.1 402 0.454 2015-2016    UA
```

結構化查詢 (tydyverse)

Function	Meaning
filter()	篩選 (過濾)
select()	選擇
mutate()	新增
arrange()	排序
summarise()	聚合函數
group_by()	分組



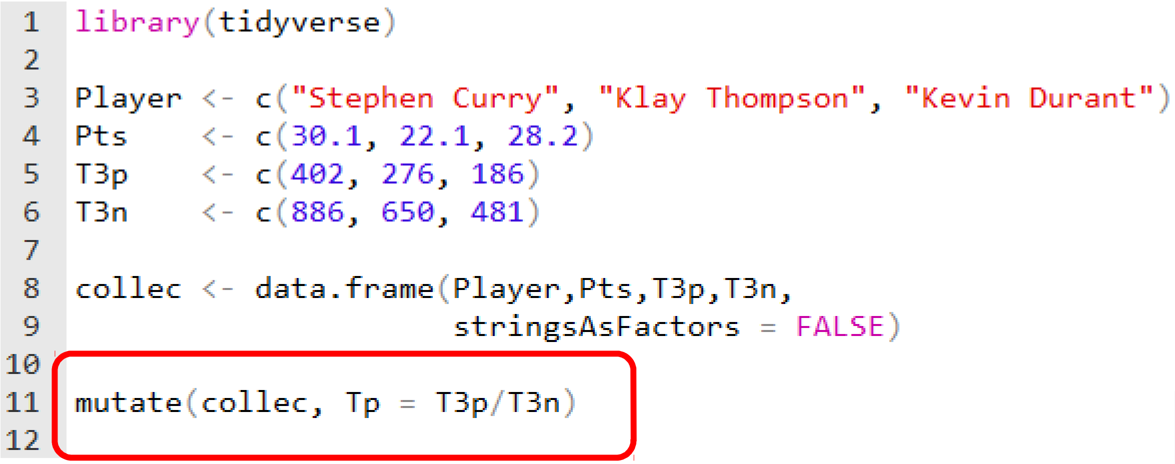
```r
library(tidyverse)

Player <- c("Stephen Curry", "Klay Thompson")
Pts    <- c(30.1, 22.1)
T3p    <- c(402, 276)
Tp     <- c(0.454,0.425)
Season <- c("2015-2016","2015-2016")
Shoes  <- c("UA","Anta")

collec <- data.frame(Player,Pts,T3p,Tp,Season, Shoes,
                     stringsAsFactors = FALSE)
KD    <- c("Kevin Durant", 28.2, 186, 0.387,
           "2015-2016","Nike")
collec <- rbind(collec,KD)

select(collec,Player)

select(collec, Name = Player)
```



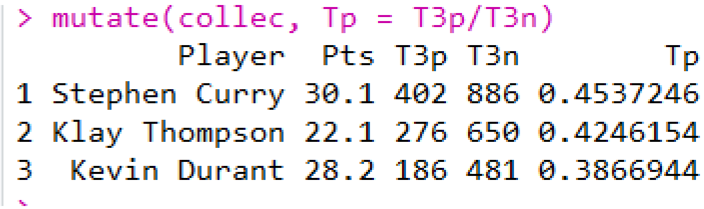
```
> select(collec,Player)
         Player
1 Stephen Curry
2 Klay Thompson
3  Kevin Durant
> 
> select(collec, Name = Player)
          Name
1 Stephen Curry
2 Klay Thompson
3  Kevin Durant
```

結構化查詢 (tydyverse)

Function	Meaning
filter()	篩選 (過濾)
select()	選擇
mutate()	新增
arrange()	排序
summarise()	聚合函數
group_by()	分組

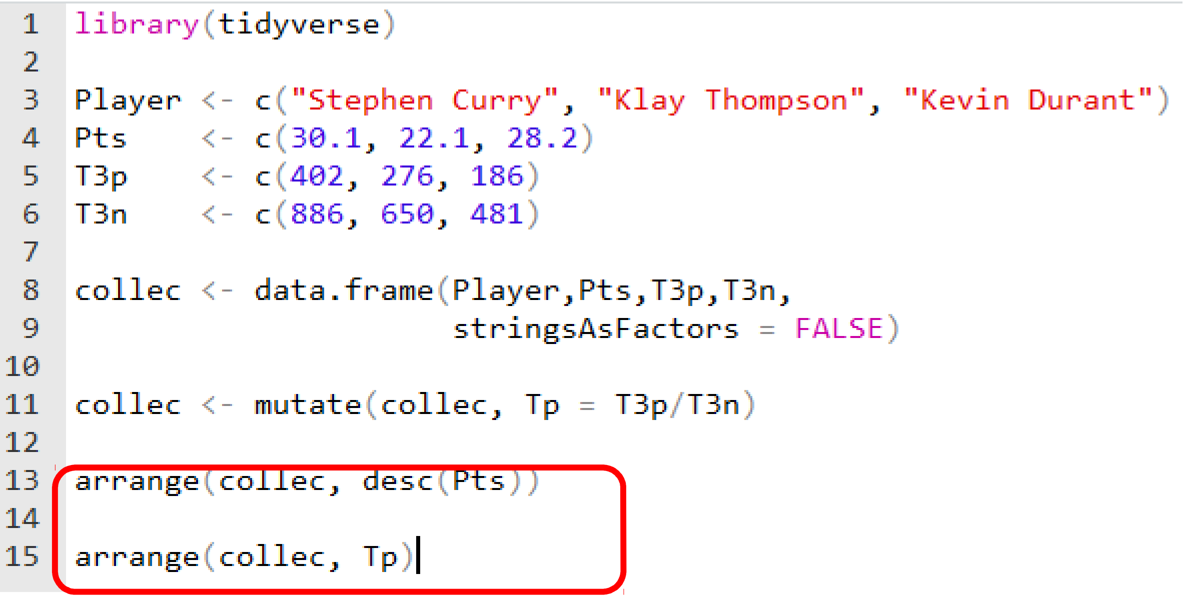


```r
library(tidyverse)

Player <- c("Stephen Curry", "Klay Thompson", "Kevin Durant")
Pts    <- c(30.1, 22.1, 28.2)
T3p    <- c(402, 276, 186)
T3n    <- c(886, 650, 481)

collec <- data.frame(Player,Pts,T3p,T3n,
                     stringsAsFactors = FALSE)

mutate(collec, Tp = T3p/T3n)

```



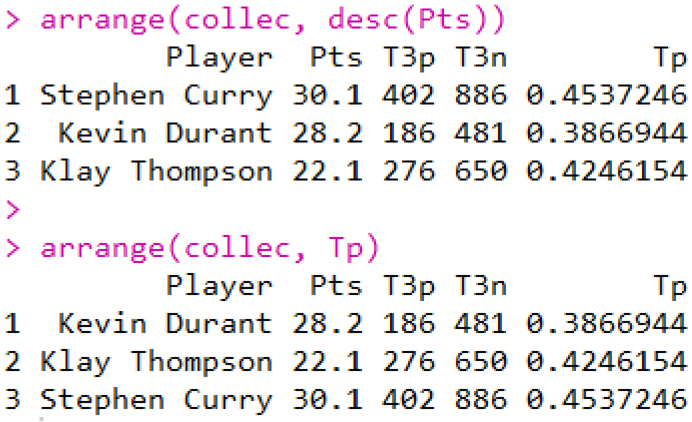
```
> mutate(collec, Tp = T3p/T3n)
         Player  Pts T3p T3n        Tp
1 Stephen Curry 30.1 402 886 0.4537246
2 Klay Thompson 22.1 276 650 0.4246154
3  Kevin Durant 28.2 186 481 0.3866944
```

結構化查詢 (tydyverse)



```r
library(tidyverse)

Player <- c("Stephen Curry", "Klay Thompson", "Kevin Durant")
Pts    <- c(30.1, 22.1, 28.2)
T3p    <- c(402, 276, 186)
T3n    <- c(886, 650, 481)

collec <- data.frame(Player,Pts,T3p,T3n,
                     stringsAsFactors = FALSE)

collec <- mutate(collec, Tp = T3p/T3n)

arrange(collec, desc(Pts))

arrange(collec, Tp)
```

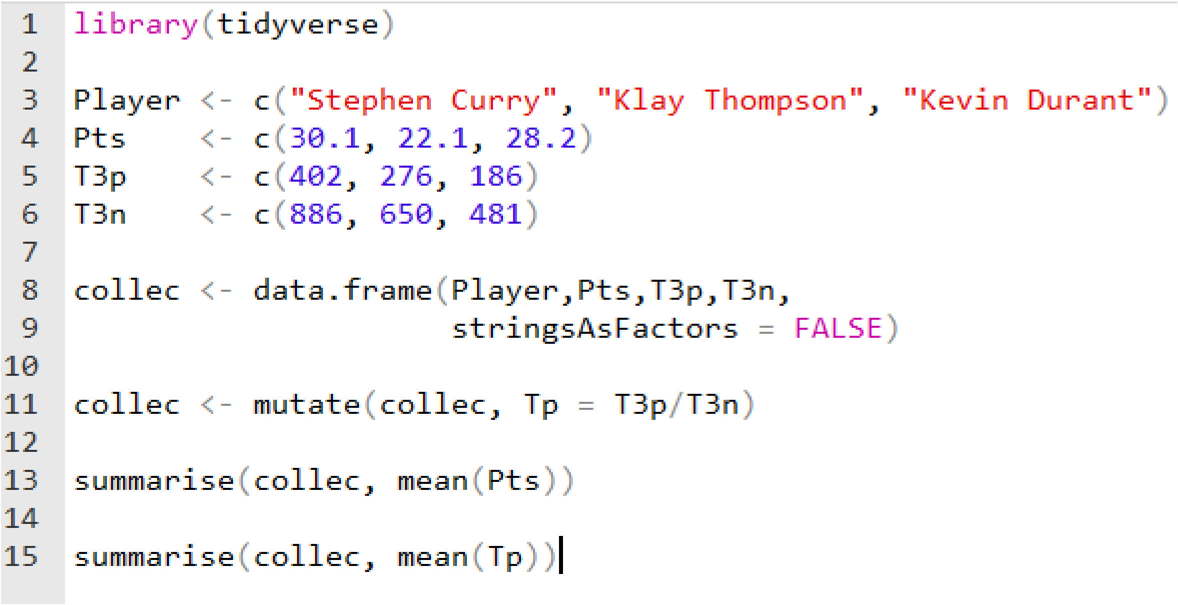
Function	Meaning
filter()	篩選 (過濾)
select()	選擇
mutate()	新增
arrange()	排序
summarise()	聚合函數
group_by()	分組



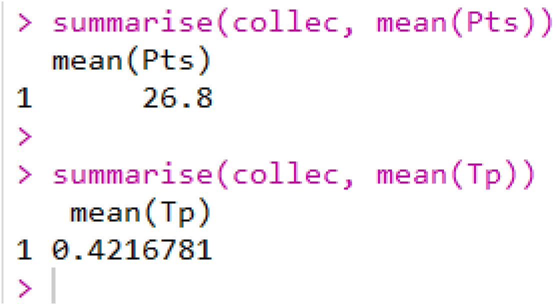
```
> arrange(collec, desc(Pts))
         Player  Pts T3p T3n        Tp
1 Stephen Curry 30.1 402 886 0.4537246
2  Kevin Durant 28.2 186 481 0.3866944
3 Klay Thompson 22.1 276 650 0.4246154
> 
> arrange(collec, Tp)
         Player  Pts T3p T3n        Tp
1  Kevin Durant 28.2 186 481 0.3866944
2 Klay Thompson 22.1 276 650 0.4246154
3 Stephen Curry 30.1 402 886 0.4537246
```

結構化查詢 (tydyverse)

Function	Meaning
filter()	篩選 (過濾)
select()	選擇
mutate()	新增
arrange()	排序
summarise()	聚合函數
group_by()	分組



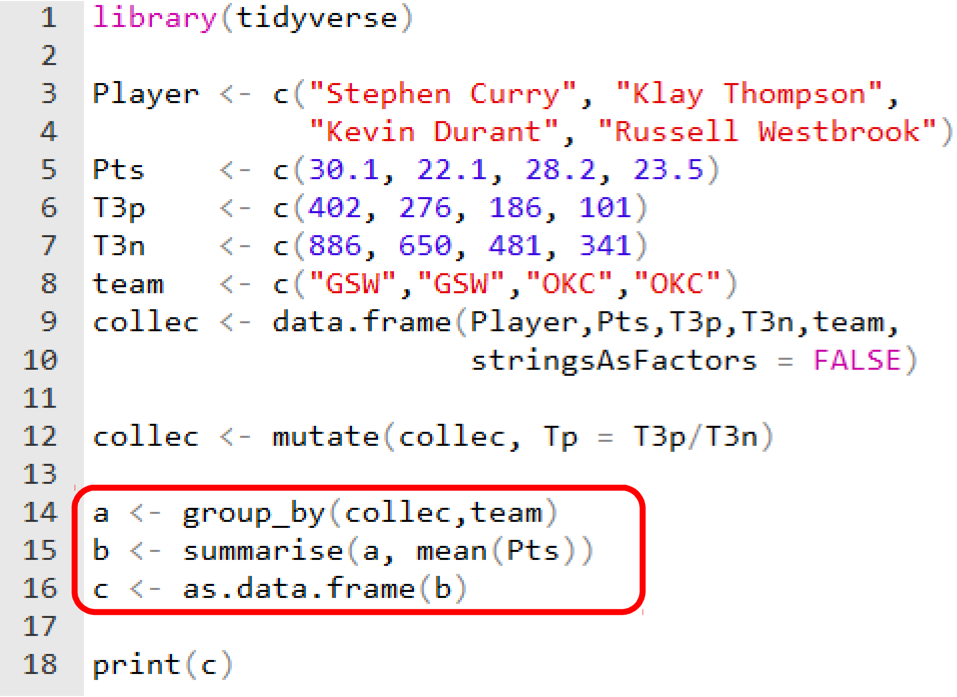
```
library(tidyverse)

Player <- c("Stephen Curry", "Klay Thompson", "Kevin Durant")
Pts    <- c(30.1, 22.1, 28.2)
T3p    <- c(402, 276, 186)
T3n    <- c(886, 650, 481)

collec <- data.frame(Player,Pts,T3p,T3n,
                     stringsAsFactors = FALSE)

collec <- mutate(collec, Tp = T3p/T3n)

summarise(collec, mean(Pts))

summarise(collec, mean(Tp))
```



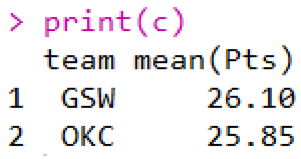
```
> summarise(collec, mean(Pts))
  mean(Pts)
1      26.8
> 
> summarise(collec, mean(Tp))
   mean(Tp)
1 0.4216781
> 
```

結構化查詢 (tydyverse)

Function	Meaning
filter()	篩選 (過濾)
select()	選擇
mutate()	新增
arrange()	排序
summarise()	聚合函數
group_by()	分組



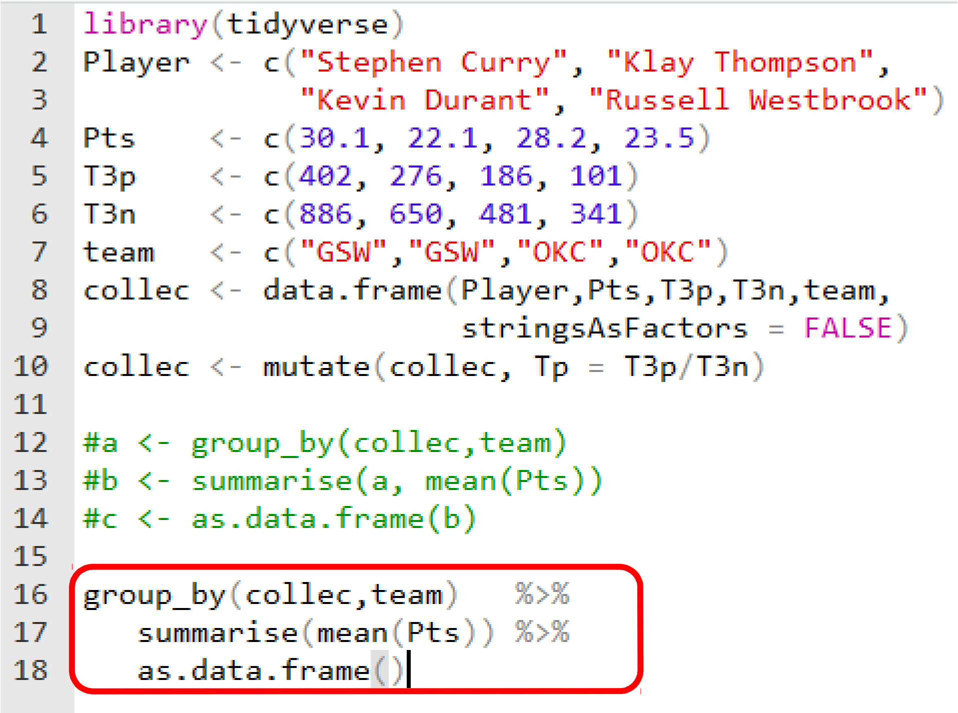
```r
library(tidyverse)

Player <- c("Stephen Curry", "Klay Thompson",
            "Kevin Durant", "Russell Westbrook")
Pts    <- c(30.1, 22.1, 28.2, 23.5)
T3p    <- c(402, 276, 186, 101)
T3n    <- c(886, 650, 481, 341)
team   <- c("GSW","GSW","OKC","OKC")
collec <- data.frame(Player,Pts,T3p,T3n,team,
                      stringsAsFactors = FALSE)

collec <- mutate(collec, Tp = T3p/T3n)

a <- group_by(collec,team)
b <- summarise(a, mean(Pts))
c <- as.data.frame(b)

print(c)
```



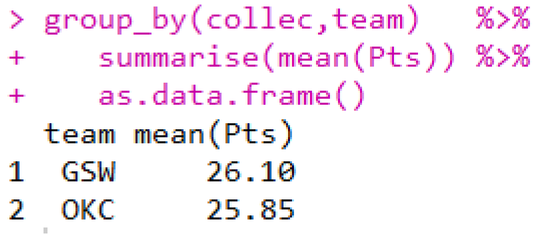
```
> print(c)
  team mean(Pts)
1  GSW     26.10
2  OKC     25.85
```

結構化查詢 (%>%)

Function	Meaning
filter()	篩選 (過濾)
select()	選擇
mutate()	新增
arrange()	排序
summarise()	聚合函數
group_by()	分組

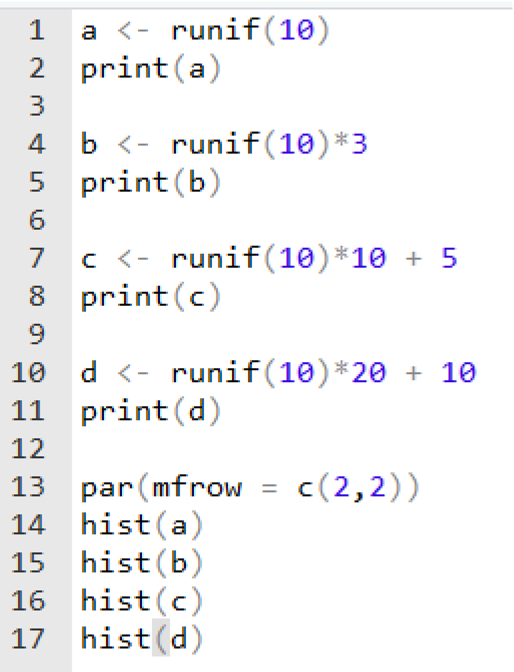


```r
library(tidyverse)
Player <- c("Stephen Curry", "Klay Thompson",
            "Kevin Durant", "Russell Westbrook")
Pts    <- c(30.1, 22.1, 28.2, 23.5)
T3p    <- c(402, 276, 186, 101)
T3n    <- c(886, 650, 481, 341)
team   <- c("GSW","GSW","OKC","OKC")
collec <- data.frame(Player,Pts,T3p,T3n,team,
                     stringsAsFactors = FALSE)
collec <- mutate(collec, Tp = T3p/T3n)

#a <- group_by(collec,team)
#b <- summarise(a, mean(Pts))
#c <- as.data.frame(b)

group_by(collec,team)   %>%
   summarise(mean(Pts)) %>%
   as.data.frame()
```

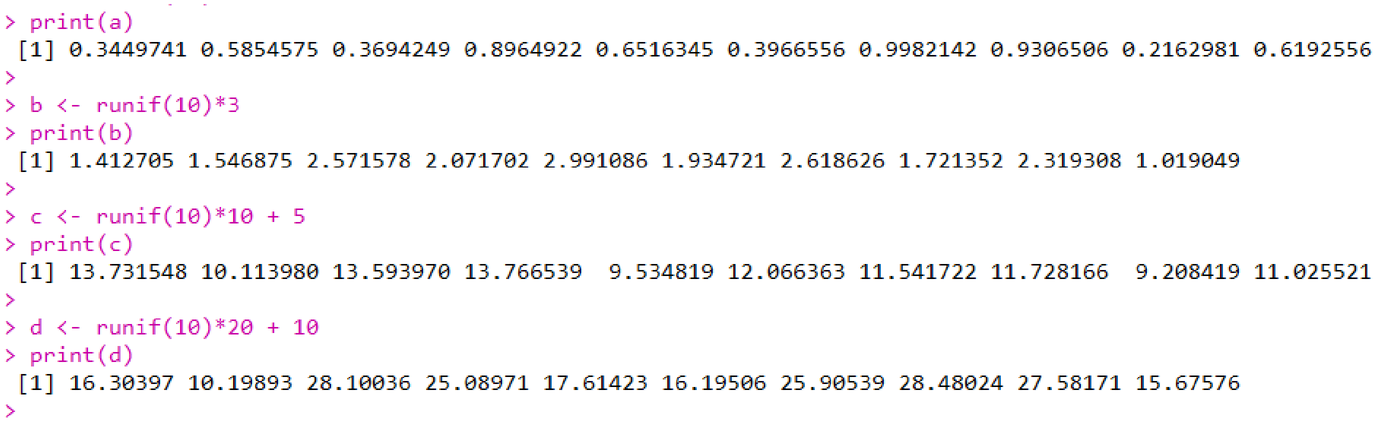


```
> group_by(collec,team)   %>%
+    summarise(mean(Pts)) %>%
+    as.data.frame()
  team mean(Pts)
1  GSW     26.10
2  OKC     25.85
```

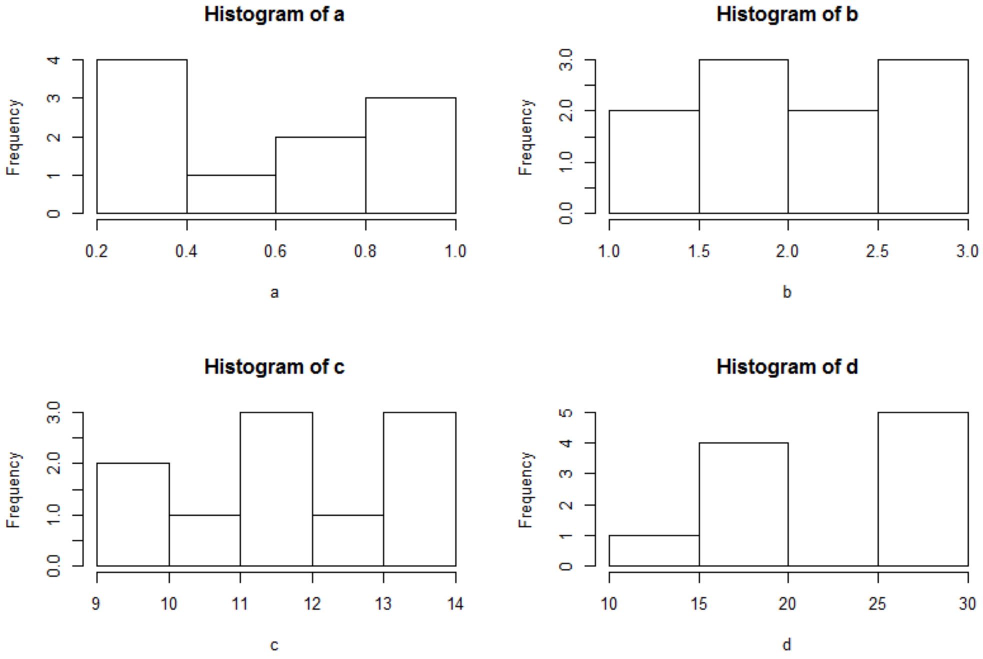
隨機分佈亂數 (runif)



```
a <- runif(10)
print(a)

b <- runif(10)*3
print(b)

c <- runif(10)*10 + 5
print(c)

d <- runif(10)*20 + 10
print(d)

par(mfrow = c(2,2))
hist(a)
hist(b)
hist(c)
hist(d)
```

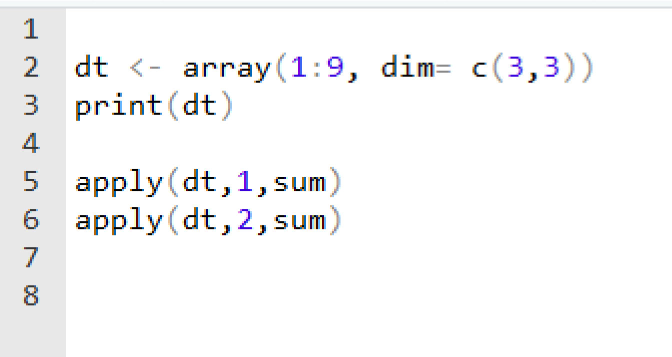


```
> print(a)
 [1] 0.3449741 0.5854575 0.3694249 0.8964922 0.6516345 0.3966556 0.9982142 0.9306506 0.2162981 0.6192556
> 
> b <- runif(10)*3
> print(b)
 [1] 1.412705 1.546875 2.571578 2.071702 2.991086 1.934721 2.618626 1.721352 2.319308 1.019049
> 
> c <- runif(10)*10 + 5
> print(c)
 [1] 13.731548 10.113980 13.593970 13.766539  9.534819 12.066363 11.541722 11.728166  9.208419 11.025521
> 
> d <- runif(10)*20 + 10
> print(d)
 [1] 16.30397 10.19893 28.10036 25.08971 17.61423 16.19506 25.90539 28.48024 27.58171 15.67576
> 
```

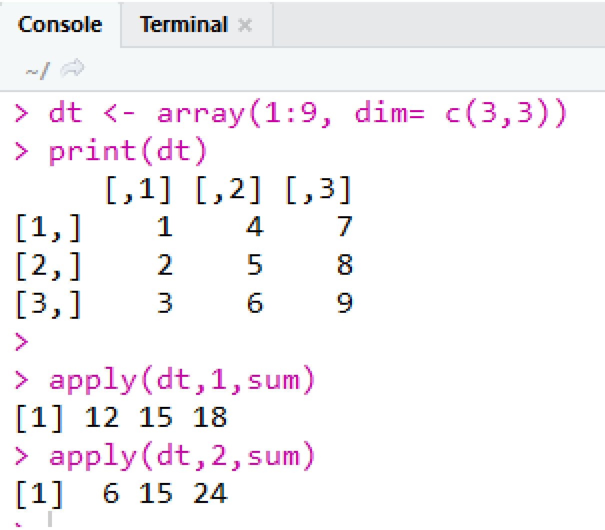


利用 apply() 取代迴圈

apply(data, MARGIN, FUN)	
MARGIN	1: 列 2: 行
FUN	內建 / 自訂函數

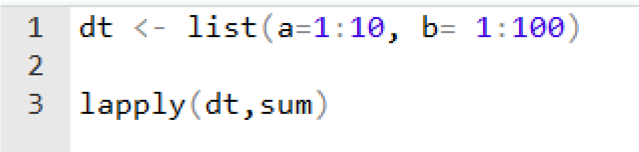


```

dt <- array(1:9, dim= c(3,3))
print(dt)

apply(dt,1,sum)
apply(dt,2,sum)


```

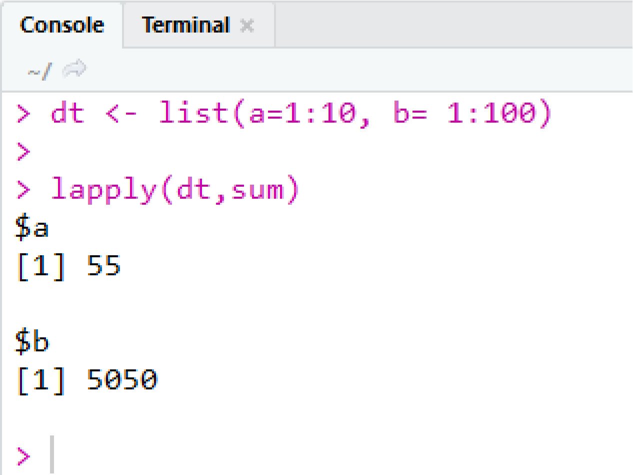


```
> dt <- array(1:9, dim= c(3,3))
> print(dt)
     [,1] [,2] [,3]
[1,]    1    4    7
[2,]    2    5    8
[3,]    3    6    9
>
> apply(dt,1,sum)
[1] 12 15 18
> apply(dt,2,sum)
[1]  6 15 24
```

利用 lapply() 取代迴圈

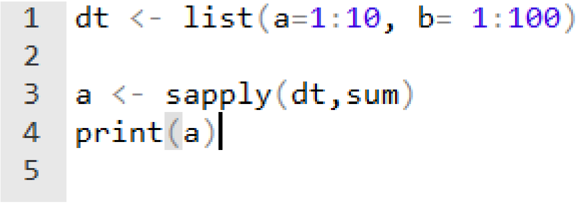


```r
dt <- list(a=1:10, b= 1:100)

lapply(dt,sum)
```

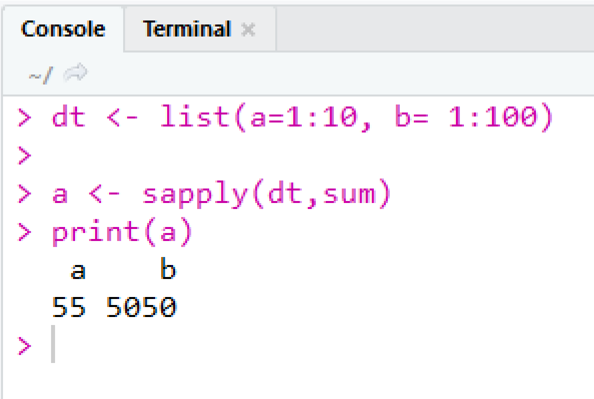


```
> dt <- list(a=1:10, b= 1:100)
>
> lapply(dt,sum)
$a
[1] 55

$b
[1] 5050

> 
```

利用 sapply() 取代迴圈



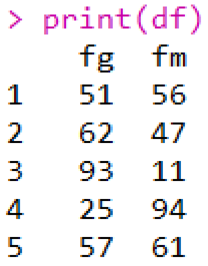
```
dt <- list(a=1:10, b= 1:100)

a <- sapply(dt,sum)
print(a)

```



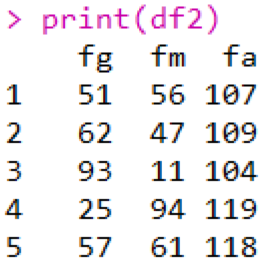
```
> dt <- list(a=1:10, b= 1:100)
> 
> a <- sapply(dt,sum)
> print(a)
  a    b 
 55 5050 
> 
```

隨堂練習 1

1. 隨機產生 50 個人的 3 分球投進與沒投進的次數，並加總投籃次數

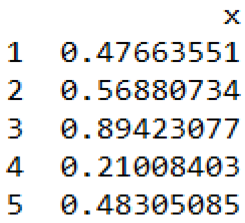


```
> print(df)
    fg  fm
1   51  56
2   62  47
3   93  11
4   25  94
5   57  61
```



```
> print(df2)
    fg  fm  fa
1   51  56 107
2   62  47 109
3   93  11 104
4   25  94 119
5   57  61 118
```

2. 使用 apply() / mapply() 來計算命中率



```
            x
1  0.47663551
2  0.56880734
3  0.89423077
4  0.21008403
5  0.48305085
```

Any Questions !?