

introduce of R

Table of Contents

一、R 語言基本介紹	2
1.資料型態.....	2
向量	2
串列	3
矩陣與陣列	4
運算.....	5
基本四則運算	5
統計相關基本運算.....	5
矩陣運算.....	6
二、作圖.....	7
1.high level plots.....	8
bar charts 長條圖	8
dot chart 點圖.....	10
pie charts 圓餅圖	11
histogram 直方圖.....	12
box plot 盒形圖.....	13
scatterplots 點陣圖	15
QQplot	15
low level plots	18

一、R 語言基本介紹

1.資料型態

向量

向量 `vector` 可使用兩種簡單的方式建構

```
x=c(5,7,9)
x
## [1] 5 7 9
```

```
x=1:5
x
## [1] 1 2 3 4 5
```

同時也可以複合數用

```
x=c(c(5,7,9),1:5)
x
## [1] 5 7 9 1 2 3 4 5
```

此外一也可以使用序列(Sequences)`seq` 來創建一個向量，其函數可指定特定間隔

```
x=seq(1,20,0.5)
x
## [1] 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0 5.5 6.0 6.5 7.0 7.5
## [15] 8.0 8.5 9.0 9.5 10.0 10.5 11.0 11.5 12.0 12.5 13.0 13.5 14.0 14.5
## [29] 15.0 15.5 16.0 16.5 17.0 17.5 18.0 18.5 19.0 19.5 20.0
```

若對向量做運算，將會使其中所有元素套用該運算

```
x=seq(1,10)
x
## [1] 1 2 3 4 5 6 7 8 9 10
x+2
## [1] 3 4 5 6 7 8 9 10 11 12
x*5
## [1] 5 10 15 20 25 30 35 40 45 50
```

若對兩個向量進行運算，將會將各自對應的元素做相對應之運算

```

x=seq(1,10)
y=seq(11:20)
x+y

## [1]  2  4  6  8 10 12 14 16 18 20

x*y

## [1]  1  4  9 16 25 36 49 64 81 100

```

要注意的是，不需要一樣，若不一樣 R 會自動將短的向量重複直到做完所有運算

```

x=seq(1,10)
y=1:5
x+y

## [1]  2  4  6  8 10  7  9 11 13 15

```

串列

串列 `list`，串列可以儲存各種資料，像是數字、浮點數、字串

```

x = 1:10
y = 3:10
z = list(x,y)
z

## [[1]]
## [1]  1  2  3  4  5  6  7  8  9 10
##
## [[2]]
## [1]  3  4  5  6  7  8  9 10

z[1]

## [[1]]
## [1]  1  2  3  4  5  6  7  8  9 10

x = "你好"
y = "hello"
z = list(x,y)
z

## [[1]]
## [1] "你好"
##
## [[2]]
## [1] "hello"

z[1]

```

```
## [[1]]  
## [1] "你好"
```

矩陣與陣列

`matrix` 為矩陣，其計算方式有獨立出來後面將會另外做介紹，基本設定方法為：

```
matrix(1:6, nrow = 2, ncol = 3)
```

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

`byrow` 可以元素排列指定方向

```
matrix(1:6, nrow = 2, ncol = 3, byrow = TRUE)
```

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

`dimnames` 則可以為行列命名

```
matrix(1:6, nrow = 2, ncol = 3,  
      dimnames = list(c("row1", "row2"),  
                      c("C.1", "C.2", "C.3")))
```

```
##      C.1 C.2 C.3  
## row1    1    3    5  
## row2    2    4    6
```

若是想建立 3 維以上的陣列，則需使用 `array`，其函數與 `matrix` 完全一樣，差別在於個維度元素改用 `dim` 來指定

```
array(1:24, dim = c(4, 3, 2), dimnames = list(  
  X = c("A1", "A2", "A3", "A4"),  
  Y = c("B1", "B2", "B3"), Z = c("C1", "C2")))
```

```
## , , Z = C1  
##  
##      Y  
## X    B1 B2 B3  
## A1    1  5  9  
## A2    2  6 10  
## A3    3  7 11  
## A4    4  8 12  
##  
## , , Z = C2  
##  
##      Y  
## X    B1 B2 B3  
## A1   13 17 21
```

```
## A2 14 18 22
## A3 15 19 23
## A4 16 20 24
```

運算

基本四則運算

`31+7`

```
## [1] 38
```

`31-7`

```
## [1] 24
```

`31*7`

```
## [1] 217
```

`31/7`

```
## [1] 4.428571
```

次方

`5^3`

```
## [1] 125
```

商數

`31%%7`

```
## [1] 3
```

餘數

`31%/%7`

```
## [1] 4
```

統計相關基本運算

`x=1:11`

`x`

```
## [1] 1 2 3 4 5 6 7 8 9 10 11
```

總和

`sum(x)`

```
## [1] 66
```

平均數

```
mean(x)
```

```
## [1] 6
```

標準差

```
var(x)
```

```
## [1] 11
```

矩陣運算

基本的加減乘除都可以運用

```
x=matrix(1:9, nrow = 3, ncol = 3)
```

```
y=matrix(1:9, nrow = 3, ncol = 3)
```

x

```
##      [,1] [,2] [,3]
```

```
## [1,]    1    4    7
```

```
## [2,]    2    5    8
```

```
## [3,]    3    6    9
```

y

```
##      [,1] [,2] [,3]
```

```
## [1,]    1    4    7
```

```
## [2,]    2    5    8
```

```
## [3,]    3    6    9
```

x+y

```
##      [,1] [,2] [,3]
```

```
## [1,]    2    8   14
```

```
## [2,]    4   10   16
```

```
## [3,]    6   12   18
```

x*y

```
##      [,1] [,2] [,3]
```

```
## [1,]    1   16   49
```

```
## [2,]    4   25   64
```

```
## [3,]    9   36   81
```

將矩陣轉置可使用 t

```
t(x)
```

```
##      [,1] [,2] [,3]
```

```
## [1,]    1    2    3
```

```
## [2,]    4    5    6
## [3,]    7    8    9
```

內積`%*`或外積`%o`，其中外積也可以使用函數 `outer`

```
x%*%y
##      [,1] [,2] [,3]
## [1,]   30   66  102
## [2,]   36   81  126
## [3,]   42   96  150
```

```
c(1,2,3)%o%c(4,5,6)
```

```
##      [,1] [,2] [,3]
## [1,]    4    5    6
## [2,]    8   10   12
## [3,]   12   15   18
```

```
outer(1:3,4:6)
```

```
##      [,1] [,2] [,3]
## [1,]    4    5    6
## [2,]    8   10   12
## [3,]   12   15   18
```

若想解反矩陣，則需使用函數 `solve`

```
m = matrix(c(1, 0, 1, 5, -3, 1, 2, 4, 7), nrow = 3)
m
```

```
##      [,1] [,2] [,3]
## [1,]    1    5    2
## [2,]    0   -3    4
## [3,]    1    1    7
```

```
solve(m)
```

```
##      [,1] [,2] [,3]
## [1,] -25  -33   26
## [2,]    4    5   -4
## [3,]    3    4   -3
```

二、作圖

r 語言強大的地方在於作圖能力，其內部有非常多的內建函數可創建各式表格，其中大致可將其分類為 `high level` 及 `low level`

1.high level plots

high level plot 在繪製時優點在於系統會自動將其繪製完整，但缺點也很明顯:無法做出細部的改變，以下將介紹幾個常用的函數

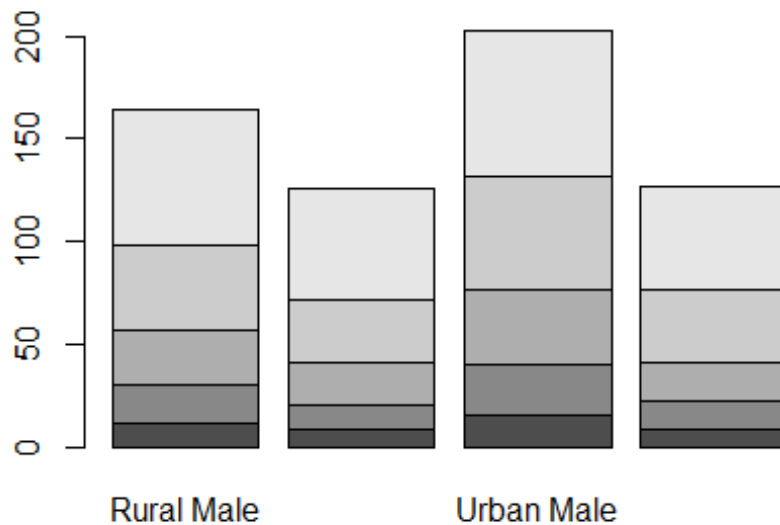
bar charts 長條圖

我們將使用 VADeaths 這個 r 語言內部設立之資料進行示範

VADeaths

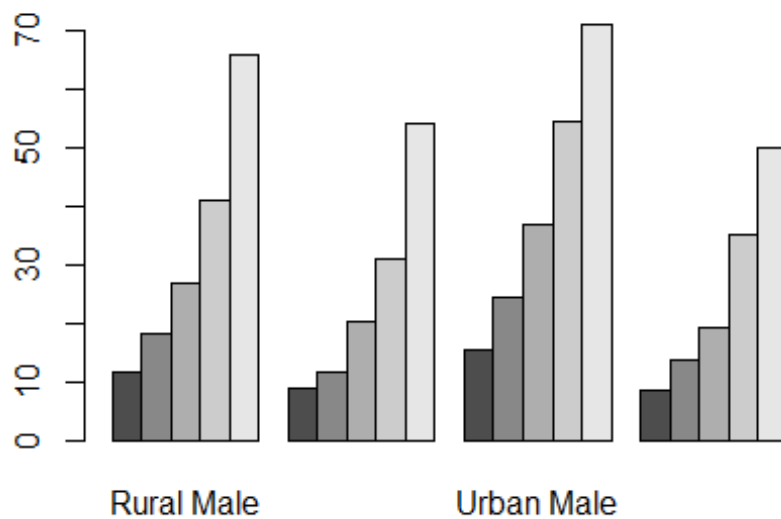
```
##      Rural Male Rural Female Urban Male Urban Female
## 50-54      11.7      8.7      15.4      8.4
## 55-59      18.1     11.7     24.3     13.6
## 60-64      26.9     20.3     37.0     19.3
## 65-69      41.0     30.9     54.6     35.1
## 70-74      66.0     54.3     71.1     50.0
```

```
barplot((VADeaths))
```



可以使用 `beside` 來調整(預設 `False`)

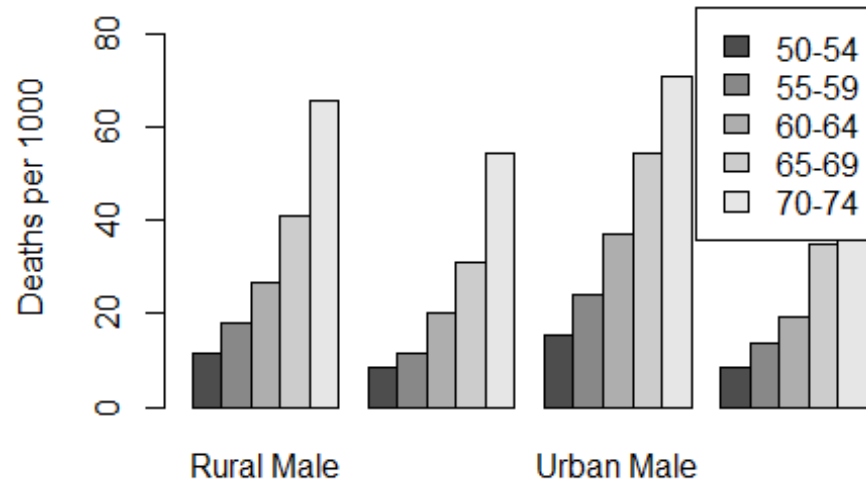
```
barplot(VADeaths,beside=TRUE)
```

另外有些函數是控制標題、座標軸上下限，這些因子在各種繪圖函數都可以使用，這裡簡單介紹幾個，`legend` 可決定是否放置圖例 `ylab` 可設定 y 軸名稱，同樣有針對 x 軸設定的函數 `xlab` `ylim` 可設定 y 軸上下限，對應之 x 軸設定為 `xlim` `main` 可設定圖名

```
barplot(VADeaths,beside=TRUE,legend=TRUE,ylim=c(0,90),ylab="Deaths per 1000",main="Death rates in virginia")
```

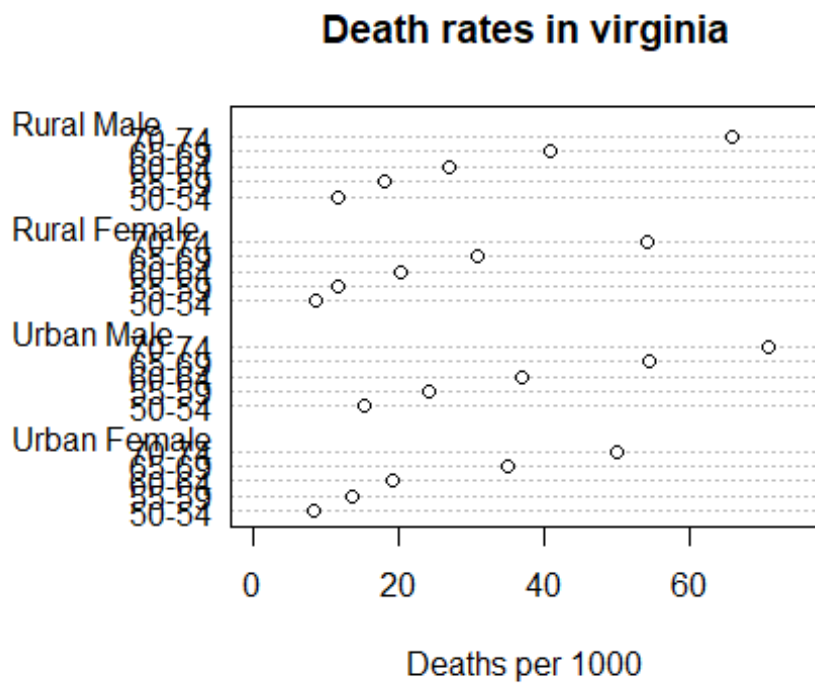
Death rates in virginia



dot chart 點圖

點圖則可以使用 `dotchart` 來創建，函數與 `barplot` 大致雷同，不再一一贅述

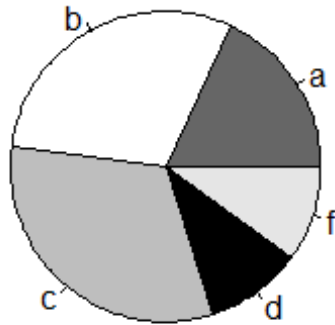
```
dotchart(VADeaths,xlim=c(0,75),xlab="Deaths per 1000",main="Death rates  
in virginia")
```



pie charts 圓餅圖

使用 `pie` 即可創建一個圓餅圖，其中 `col` 可指定區塊顏色

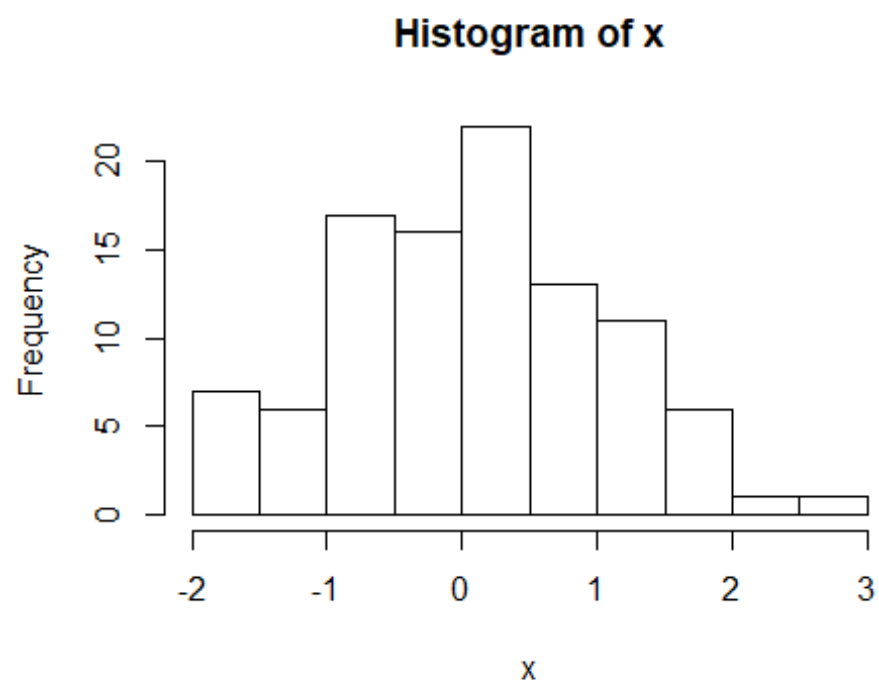
```
x = c(18,30,32,10,10)
y = c("a","b","c","d","f")
z = c("grey40","white","grey","black","grey90")
pie(x,y,col=z)
```



histogram 直方圖

可使用 `hist` 來畫出直方圖，這邊我們將使用 `rnorm` 這個函數來生成一個標準常態分配

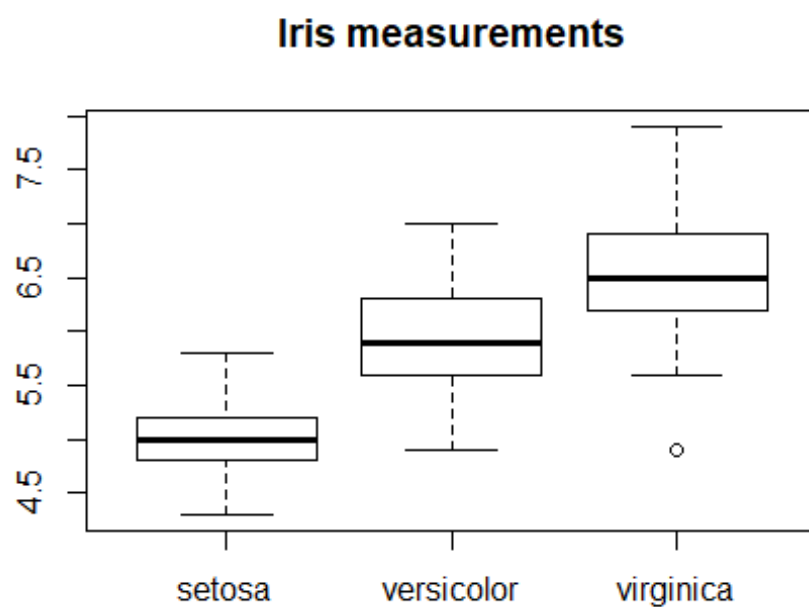
```
x = rnorm(100)
hist(x)
```



box plot 盒形圖

使用 `boxplot` 可創建盒形圖，在這邊我們將使用 `r` 內建資料 `iris`

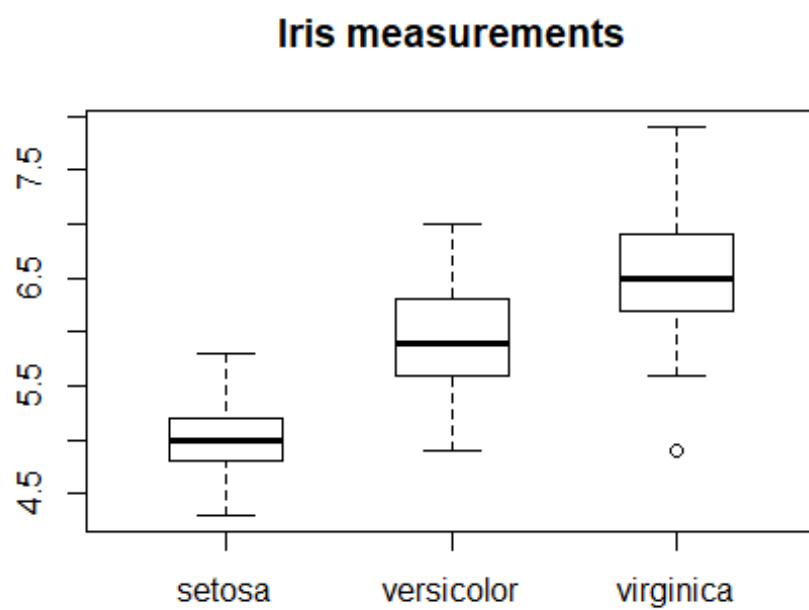
```
boxplot(Sepal.Length ~ Species, data = iris, ylab="Sepal length(cm)", main="Iris measurements")
```



若加入

`boxwex` 則可調整每個盒子的寬度

```
boxplot(Sepal.Length ~ Species, data = iris, ylab="Sepal length(cm)", main="Iris measurements", boxwex=0.5)
```



scatterplots 點陣圖

我們一樣使用 `rnorm` 來創建一個標準常帶分配，另外我們使用 `rplis` 來創建一個普瓦松分配

```
x = rnorm(100)
y = rpois(100,30)
```

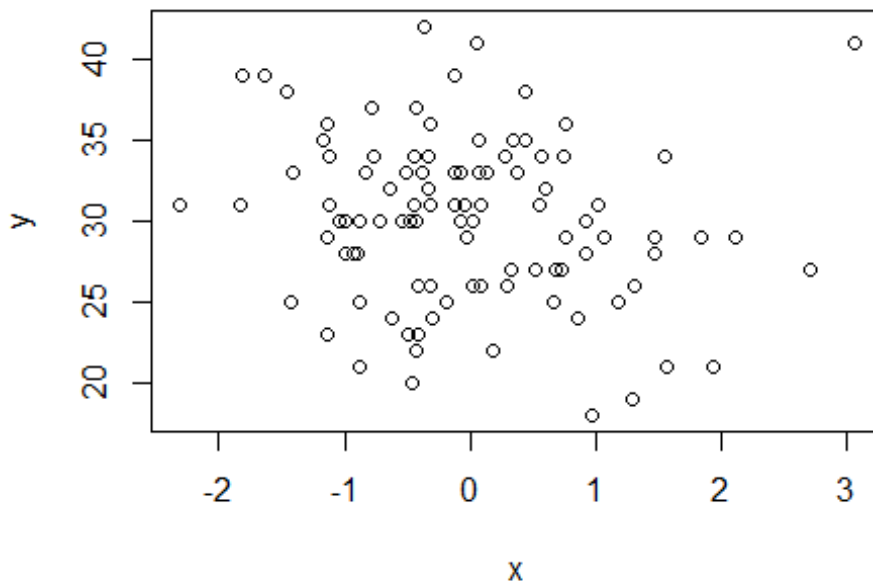
`rpois` 中的 30 代表其平均數為 30(會有些微誤差)

```
mean(y)
```

```
## [1] 30.08
```

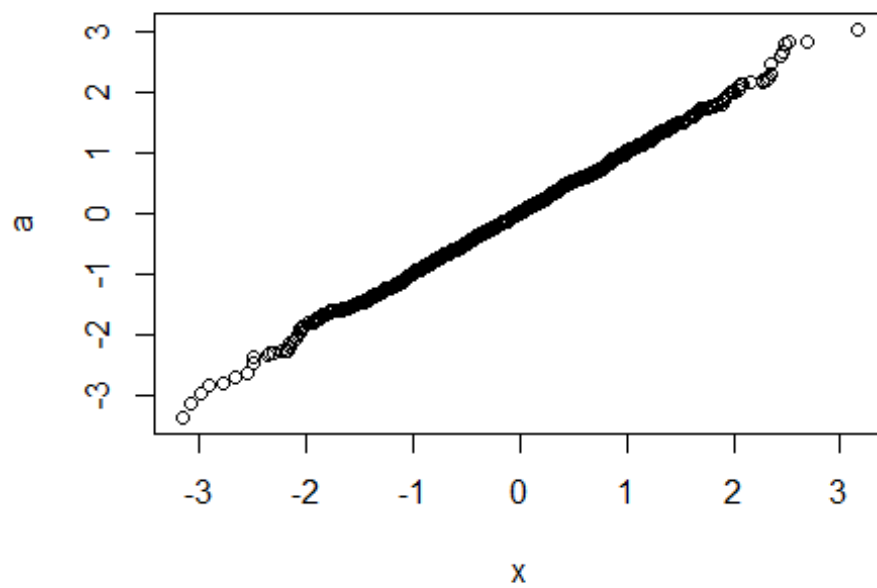
我們使用的函數為 `plot`

```
plot(x,y)
```

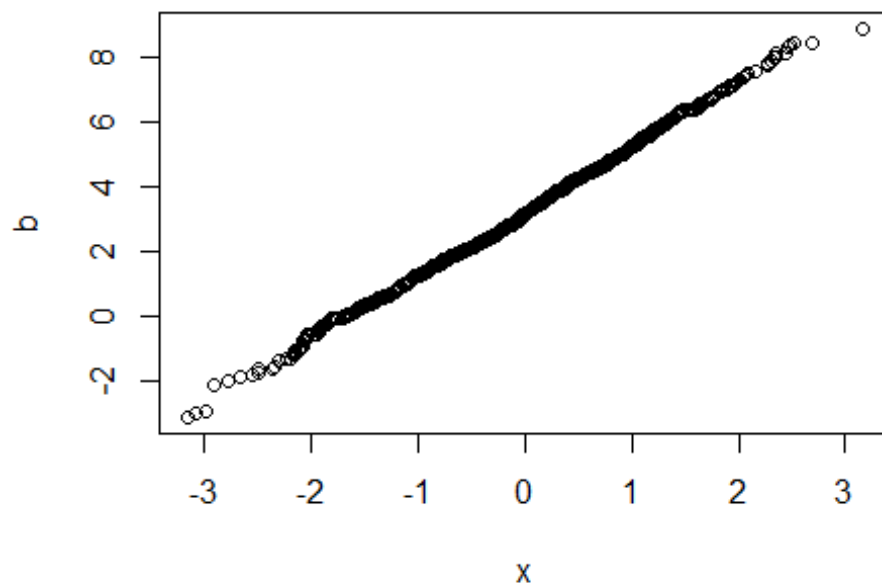


QQplot

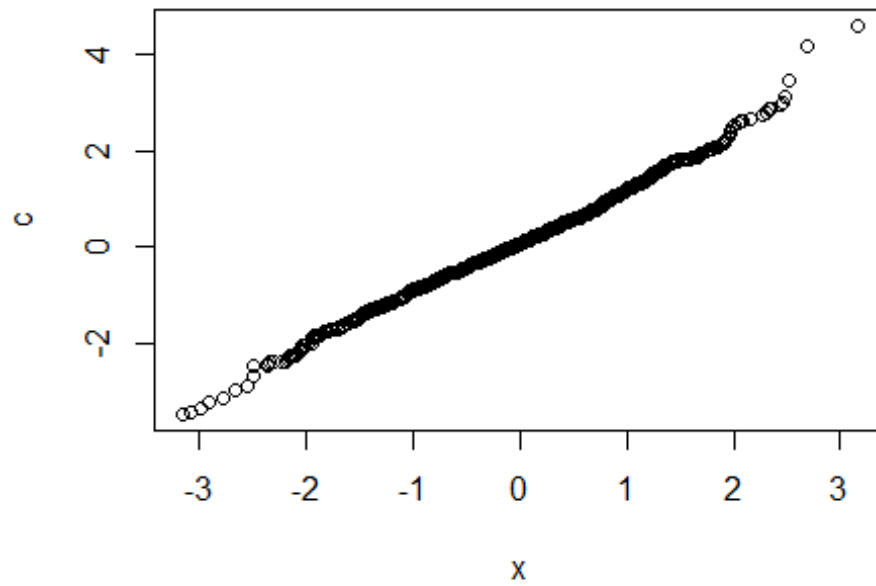
```
x = rnorm(1000)
a = rnorm(1000)
qqplot(x,a)
```



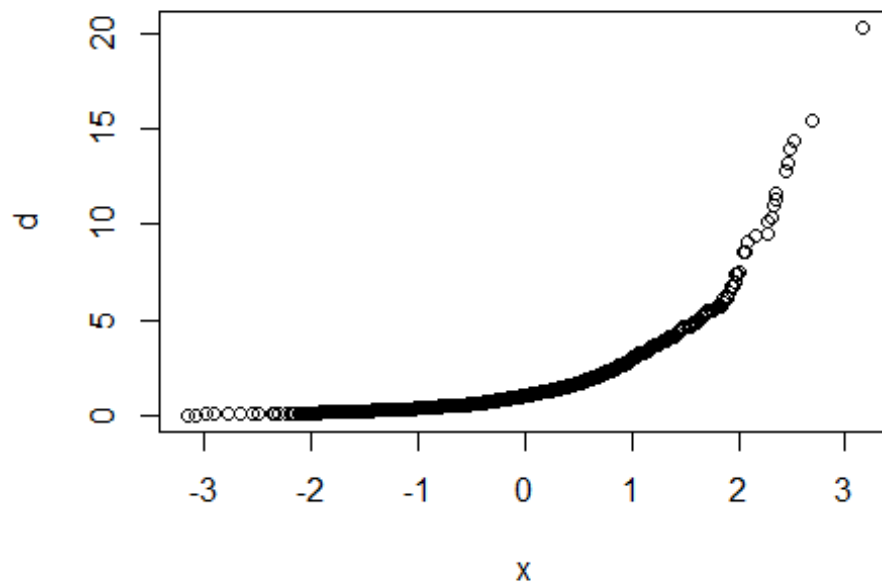
```
b = rnorm(1000,mean=3,sd=2)
qqplot(x,b)
```




```
c = rt(1000,df=20)  
qqplot(x,c)
```



```
d = exp(rnorm(1000))  
qqplot(x,d)
```



low level plots

low level 則是許多細部的畫圖參數，其特色就是可以一個一個的加入使用者所需要的函數以下將舉幾個例子 **point** 可添加多個點到圖形中 **lines** 添加多條線段到圖形中 **abline** 添加一條直線到圖形中 **legend** 添加圖例