05 Discrete Random Variables

Introduction

本章節對應 LN.CH5，以 R 實作介紹離散隨機變數 (discrete random variable)，以及其 pmf/cdf/qf/rf。並說明 discrete random variable (rv) 的 transformation 和 expectation。末了，介紹一些常見的 discrete rv 的 distribution，並以課堂上提到的範例作演練。

Discrete random variables

隨機變數 (random variable) 是從原本的樣本空間經過某種轉換 (transformation) 到實數線上，呈現出來的是一個值，但背後有一種機率在控制著。呈現出來的值若是 discrete 的，那這隨機變數就是離散的。若是 continuous 的，那就是連續的隨機變數。

以丟銅板三次為範例。因為每個銅板只會有正 (h) 反 (t) 兩面的結果，所以丟三次銅板可能出現的值只會有 8 種 (outcome)，這 8 種稱為樣本空間 (sample space)，在此我們存放在 omega 中。(LNp.5-3~LNp.5-11)

這 8 種 outcome 有各自的機率值，假設銅板出現正反面的機率是均等的，那這 8 種 outcome 的機率值會是各 1/8。

```
# Generate all possible outcomes
omega <- data.frame(trial1=c(”h”,”h”,”h”, ”t”, ”h”, ”t”, ”t”, ”t”),
                    trial2=c(”h”,”h”,”t”,”h”,”t”,”h”,”t”,”t”),
                    trial3=c(”h”,”t”,”h”,”h”,”t”,”t”,”h”,”t”))
omega
```

```
##   trial1 trial2 trial3
## 1      h      h      h
## 2      h      h      t
## 3      h      t      h
## 4      t      h      h
## 5      h      t      t
## 6      t      h      t
## 7      t      t      h
## 8      t      t      t
```

再來，我們以 3 種方式將這樣本空間 transform 到實數線上，這就形成 3 個 rv。例如，本範例的第一個 rv (x1) 是銅板丟擲三次其中正面的次數、第二個 rv (x2) 是第一次是正面的次數、第三個 rv (x3) 是正面次數減去反面次數。

```
library(plyr) # For ddply
# Generate the outcome of the three random variables x1, x2, x3
omega <- ddply(omega, .(trial1, trial2, trial3), transform,
               x1=(trial1==”h”)+(trial2==”h”)+(trial3==”h”),
               x2=sum(trial1==”h”),
               x3=(trial1==”h”)+(trial2==”h”)+(trial3==”h”)-(trial1==”t”)-(trial2==”t”)-(t
omega
```

```
##   trial1 trial2 trial3 x1 x2 x3
## 1      h      h      h  3  1  3
## 2      h      h      t  2  1  1
## 3      h      t      h  2  1  1
## 4      h      t      t  1  1 -1
## 5      t      h      h  2  0  1
```

```
## 6       t       h       t  1   0 -1
## 7       t       t       h  1   0 -1
## 8       t       t       t  0   0 -3
```

當 transformation 發生時，rv 的機率值會從原本的樣本空間 (omega) 被帶過來．舉例而言，銅板丟擲三次其中正面的次數只會有 1 和 3 兩種 outcome，而其機率值由原本樣本空間那 8 種 outcome 的機率值所構成．

以下的 x1, x2, x3 就是這三個 rv．freq 是其對應到原本 sample space 的發生次數，prob 則是機率值．

事實上，在機率論中我們常會談論到某某分佈 (distribution)，這 distribution 實際上指的就是 prob 的分佈．Distribution 名稱的由來，就是把 1 的總機率值分佈到各個可能的值的想法．

```
# Calculate the probability and cumulative probability for x1, x2, x3
x1 <- ddply(omega, .(x1), summarize, freq=length(x1), prob=freq/nrow(omega))
x1
```

```
##   x1 freq  prob
## 1  0    1 0.125
## 2  1    3 0.375
## 3  2    3 0.375
## 4  3    1 0.125
```

```
x2 <- ddply(omega, .(x2), summarize, freq=length(x2), prob=freq/nrow(omega))
x2
```

```
##   x2 freq prob
## 1  0    4  0.5
## 2  1    4  0.5
```

```
x3 <- ddply(omega, .(x3), summarize, freq=length(x3), prob=freq/nrow(omega))
x3
```

```
##   x3 freq  prob
## 1 -3    1 0.125
## 2 -1    3 0.375
## 3  1    3 0.375
## 4  3    1 0.125
```

在這我順便也將 x1~x3 的累進 (cumulative) 機率值算出來，存在 acc_prob 中．累進機率值是以 x 由小至大的機率值逐漸加起來，因機率最小值為 0、總和為 1，故累進機率值會由 0 遞增到 1．

在機率論中，為了理論發展和實務應用，發展出了幾個有用的函式 (function):

函式	全名	數學表示式
pmf	Probability Mass Function	$f_X(x)$
cdf	Cumulative Density Function	$F_X(X)$
mgf	Moment Generating Function	$M_X(t)$
qf	Quantile Function	
rf	Random Generating Function	

套用到這邊的例子，pmf 就是接受 x 值，回傳 prob．cdf 就是接受 x 值，回傳 acc_prob．quantile 就是接受 acc_prob，回傳 x 值．

(qf 和 rf 是我自己命名的縮寫．)

(mgf 在課堂上大部分是用來證明，實務上我還沒看到拿來當 function 用，所以這邊就沒有著墨，等將來有用到再說．)

```
x1$acc_prob <- cumsum(x1$prob)
x1
```

```
##   x1 freq  prob acc_prob
## 1  0    1 0.125    0.125
## 2  1    3 0.375    0.500
## 3  2    3 0.375    0.875
## 4  3    1 0.125    1.000
```

```
x2$acc_prob <- cumsum(x2$prob)
x2
```

```
##   x2 freq prob acc_prob
## 1  0    4  0.5      0.5
## 2  1    4  0.5      1.0
```

```
x3$acc_prob <- cumsum(x3$prob)
x3
```

```
##   x3 freq  prob acc_prob
## 1 -3    1 0.125    0.125
## 2 -1    3 0.375    0.500
## 3  1    3 0.375    0.875
## 4  3    1 0.125    1.000
```

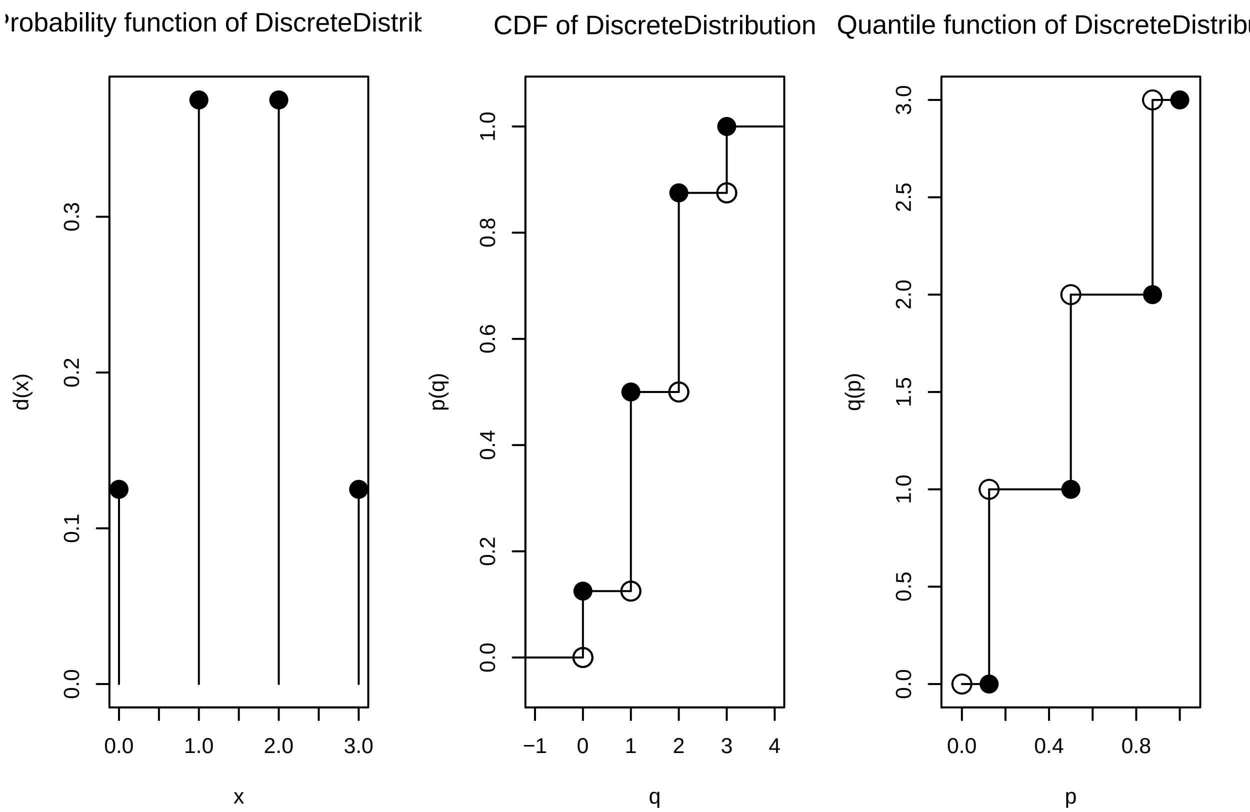
為了得到 x1, x2, x3 的 pmf/cdf/qf，我們可以自己根據 x1, x2, x3 的資料設計 function，最簡單的方式就是先把值都存在表中，然後查表就是了．

但是 R 有提供更方便的函式 DiscreteDistribution．我們只要把值和機率值填入，它就會自動產生出 pmf/cdf/qf，甚至連遵從這 distribution 的亂數產生器都能產生出來．以下以 x1 為例，其 pmf 為 x1.pmf、cdf 為 x1.cdf、qf 為 x1.quantile、rf 為 x1.rand．

```
library(distr)     # For DiscreteDistribution
x1.dist <- DiscreteDistribution (supp = x1$x1, prob = x1$prob)
x1.pmf <- d(x1.dist)   # pmf
x1.cdf <- p(x1.dist)   # cdf
x1.quantile <- q(x1.dist)   # qf
x1.rand <- r(x1.dist) # rf
```

我們可以將這些 pmf/cdf/qf 畫出來看看．

```
plot(x1.dist)
```



pmf/cdf/qf/rf

我們接著利用 x1 和其產生出來的 pmf/cdf/qf/rf 探討這些 function 的一些特性。

rf

首先，我認為 rf 是最能體現 rv 這詞的概念。當呼叫 rf 時，它都只會吐出一個值，這個值你無法預測，但背後控制的機率卻是確定的。這就是隨機變數。

以下面為例，要求 x1 吐出 10 個值，雖然無法預測會出現哪十個值，但我們知道出現 1 和 2 的比例會是最高的。

```
x1.rand(10)
```

```
##  [1] 2 2 0 3 1 1 1 0 2 2
```

pmf

pmf 若要成立，需要符合以下 3 個條件。第一是機率值必須要大於等於 0，第二是在 x 不存在的地方機率值為 0，第三是總機率值必須等於 1。

$$f_X(x) \geq 0, \text{ for all } x \in \mathbb{R}$$
$$f_X(x) = 0, \text{ for } x \notin X$$
$$\sum_{x \in X} f_X(x) = 1$$

以 x1 為例，pmf 的回傳值一定大於 0。

而因為其只會有 [0, 3] 四種 outcome，所以在這四個值上才會有機率。舉例而言，x=2.5 是不會有機率值的。可參考以下 x1.pmf 的實例。這其實就是 discrete rv 的 discrete 特性，x 值都是 countable 的。

再者，我們把所有機率值加總，和為 1。

```r
x1.pmf(c(0:3))
```

```
## [1] 0.125 0.375 0.375 0.125
```

```r
x1.pmf(2)
```

```
## [1] 0.375
```

```r
x1.pmf(3)
```

```
## [1] 0.125
```

```r
x1.pmf(2.5)  # No probability exists at 2.5
```

```
## [1] 0
```

```r
sum(x1.pmf(c(0:3)))  # Total probability is 1
```

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

cdf

cdf 若要成立，需要符合以下 3 個條件。第一是累進機率值必須不遞減，第二是累進機率值必須能由右至左逼近，第三是累進機率值必須由 0 增加到 1。

$$F_X(x) \text{ is nondecreasing}$$
$$\text{For any } x \in \mathbb{R}, F_X(x) \text{ is continuous from the right}$$
$$\lim_{x \to \infty} F_X(x) = 1 \text{ and } \lim_{x \to -\infty} F_X(x) = 0$$

cdf 因為是累進的機率值，就算值不剛好落在 x1 上，只要是在區間內，還是會有機率值。以下方為例，x1.cdf(2.5) 是會有機率值的，而其值其實剛好就是 x1=0, x1=1, x1=2 的機率值總和。

這裡需要特別注意的一點是，cdf 的機率值累加是在 x 有出現機率值時，所以當 x1 在 2 有機率值時，x1.cdf(2) 其實和 x1.cdf(2.5) 是一樣的。但 x1.cdf(1.9) 就不同了。

```r
x1.cdf(2.5)  # probability exists at 2.5 due to cdf natural
```

```
## [1] 0.875
```

```
x1.pmf(0) + x1.pmf(1) + x1.pmf(2)
```

```
## [1] 0.875
```

```
x1.cdf(2)
```

```
## [1] 0.875
```

```
x1.cdf(1.9)
```

```
## [1] 0.5
```

cdf 在觀念上，可以想成要求 X1 小於等於某 x1 的機率值總和．而因著總機率值為 **1**，所以如果要求 X1 大於某 x1 的機率值總合，只要用 **1** 減去就是了．如下面範例所示．

```
x1.pmf(2) + x1.pmf(3) == 1 - x1.cdf(1.9)
```

```
## [1] TRUE
```

qf

qf 正好就是 cdf 的反函式，cdf 是接收 x 值，回傳機率值．qf 則是接收機率值，回傳 x 值．Quantile 在之後的範圍估計 (interval estimation) 及假設檢定 (hypothesis testing) 很常用到，用來查詢某機率值假設之下的 x 值為何．

```
x1.cdf(2)
```

```
## [1] 0.875
```

```
x1.quantile(0.875)
```

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

Transformation

因為 rv 是個值，所以可以對其做運算 $(+ - */log...)$，這個運算稱為 Transformation．舉例而言，車輛銷售數目若是一個隨機變數 car，那新車車輪數目 wheel 就是 car*4．若是對 X 做 transformation 成為 Y，在數學上表示法為：(LNp.5-12)

$$Y = g(X)$$

先前提到 rv 的產生是稱為由原本樣本空間做 transformation 而來，這邊的 transformation 跟那時的概念是一樣的．可以理解為由某一個 rv 的樣本空間 transformation 到另一個 rv 的樣本空間．

對值做運算很簡單，但運算過後的新 rv，要計算其背後控制的機率分佈會變成什麼，就不那麼直覺．

Method of pmf

要計算 transformation 過後的變數的機率，根本的觀念在於，要把原本 sample space 上對應的集合的機率值搬過來．舉例而言，若 y1 是 x1 做平方的 transformation，那麼要知道 y1=4 的機率值，只要去找 x1=-2 和 x1=2 的機率值總和就是了．用數學公式寫出來是這樣子：

$$f_Y(y) = \sum_{x \in X, g(x)=y} f_X(x)$$

在這我們以先前的 x1 為例，對其做 transformation 創造 $y1 = g(x) = x1^2$．我們示範兩種計算 y1 的方式．

第一種方式是每一種 x1 都計算出 y1，然後用 y1 的值和原本對應的 x1 的機率值，透過 DiscreteDistribution() 函式產生出 pmf/cdf/...．

```
# Assume a transformation Y1=g(X1): Y1=X1^2
# Approach 1: map Y1 to the original sample space to get its pmf fy(Y=y)
y1 <- ddply(x1, .(x1), transform, y1=x1^2)
y1
```

```
##   x1 freq  prob acc_prob y1
## 1  0    1 0.125    0.125  0
## 2  1    3 0.375    0.500  1
## 3  2    3 0.375    0.875  4
## 4  3    1 0.125    1.000  9
```

```
y1.dist <- DiscreteDistribution (supp = y1$y1, prob = y1$prob)
y1.pmf <- d(y1.dist)
```

第二種方式是由數學推導上著手，用 transformation 的反函式做 (詳情見課堂說明)．雖然不像第一種方式那麼直接，但概念上其實還是去原本 sample space 搬機率值就是了．

由結果來看，第一種方法算出的 y1.pmf 和第二種方法算出的 y1.pmf2 是一樣的．

```
# Approach 2: replace fx(X=x) with fx(X=g'(Y1))
y1.pmf2 <- function(y) {
  x1.pmf(sqrt(y)) + x1.pmf(-sqrt(y))
}
y1.pmf2(c(0:4))
```

```
## [1] 0.250 0.375 0.000 0.000 0.375
```

```
y1.pmf(c(0:4))
```

```
## [1] 0.125 0.375 0.000 0.000 0.375
```

Expection

Expection 是對資料做加權平均 $E(X) = \sum_{x \in X} x f_X(x)$，以 $E()$ 符號代表．如果是對原值做加權平均 $E(X)$，結果就是平均 μ_X．如果是對 $(X - \mu_X)^2$ 做加權平均 $E((X - \mu_X)^2)$，結果就是變異數 (variance)σ_X^2．根據公式，計算 variance 也可以用 $E(x^2) - (E(x))^2$．(LNp.5-17)

如果有原始資料的話，R 的函式 mean() 就可以直接算平均了．但 R 的函式 var() 則不行，因為這算的是 sample variance，會跟這邊定義的 variance 有些微差異．

計算範例如下：

```r
x <- data.frame(x=c(0,1,2,3,4), fx=c(5/210, 50/210, 100/210, 50/210, 5/210))
# calculate mean: x * f(x)
(mean = sum(x$x * x$fx))
```

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

```r
# calculate variance: (x-u)^2 * f(x)
(var = sum( (x$x - mean)^2 * x$fx ))
```

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

```r
# calculate variance: E(x^2) - (E(x))^2
sum((x$x)^2 * x$fx) - (mean^2)
```

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

在 mgf 中會提到的第 k 個 moment，其實就是 $\mu_k = E(X^k)$。而第 k 個 central moment，則是 $\mu'_k = E((X - \mu_X)^k)$。下方範例是計算 x 的第二個 moment。

Moment 在統計中佔有很根本的角色，因為它能顯出 rv 的特徵值。舉例而言，first moment(μ_1) 就是 mean，顯出資料的重心。second central moment(μ'_2) 就是 variance，顯出資料變動範圍。μ'_3/σ^3 是 skewness，顯出 distribution 的左右偏移。μ'_4/σ^4 是 kurtosis，顯出 distribution 的尖緩。

```r
# calculate moment: x^n * f(x)
# here, n is set to 2: x^2 * f(x)
n <- 2
(mean_of_sqrt = sum((x$x)^n * x$fx))
```

```
## [1] 4.666667
```

提到 Expectation 就不免要順帶提到 Mean Square Error (MSE)。MSE 是用來評估數據相對於某定值 c 的距離，其評估方式是計算 x 距離 c 的平方的加權平均 $E((X - c)^2)$。如果把 c 換成 mean，就成了 x 的 second central moment 了。

以下以 c=3 為範例計算 MSE。有兩種計算方法，第一種就是用 $E((X - c)^2)$ 來算。第二種則是可以用 $variance + bias^2$ 來算，其實這就是根據先前所提到 variance 的第二種計算公式而來 $E(X^2) - (E(X))^2$。

第二種計算公式中的 variance 和 bias 也可稱為 accuracy 和 precision。accuracy 指的是資料集中度 (用尺的精準度來想像)，就是 variance 的意義。而 precision 指的是離目標 c 的距離，也就是 bias 的意義。

MSE 在許多地方都廣泛應用，在統計學裡用來評估估計式 (estimator) 的效率 (efficiency)。在 deep learning 裡當做 cost function 來評估學習效果。

```r
# Calculate MSE, given c=3
c <- 3
# Approach 1:
mse1 <- sum((x$x - c)^2 * x$fx)
mse1
```

```
## [1] 1.666667
```

```r
# Approach 2: bias and variance
bias <- c-mean
mse2 <- (var + bias^2)
mse2
```

```
## [1] 1.666667
```

Expectation of transformation

數學上，若要求 transformation 後的 expectation，不需要先求出 transformation 後的 distribution 再求 expection，而是可以直接用原本的 expectation 就可以算出來。

在這我們假設 transformation 是 $Y = g(X)$。要求 Y 的 expectation，公式如下。

$$E(Y) = \sum_{x \in X} g(x) f_X(x)$$

這定理被稱為 LOTUS: Law of the Unconscious Statistician (意識不清的統計學家定理)。這名稱的由來大概是因為大家都被複雜的 transformation 弄到頭昏腦脹了，連算 transformation 的 expection 都覺得這一定要很困難的機率搬移，而事實上，只是對原值做運算而已，對於頭腦不清楚的統計學家實在是簡單到不可思議。

如果假設 $Y = g(X) = aX + b$ 這樣的線性轉換的話，應用到 μ_X 和 σ_X 的 transformation 會是這樣：

$$\text{For } a, b \in \mathbb{R}, E(aX + b) = aE(X) + b$$

$$\text{For } a, b \in \mathbb{R}, Var(aX + b) = a^2 Var(X)$$

以 x1 為例，若我們做 $Y = 2X + 3$ 的 transformation，以下範例顯示，對 x 做 transformation 再算 μ_Y 和 σ_Y，跟直接用 μ_X 和 σ_X 計算，結果是一樣的。

藉由此範例，我們可以來討論 mean 和 variance 的 transformation 意義。下圖畫出 x1(藍) 和 y(紅) 的 distribution。

mean 直觀上代表的是 distribution 的重心。當線性轉換發生時，重心就跟著轉換。倍數 a 和偏移量 b 都會對重心影響。原本 x1 的重心在 2，經過 $Y = 2X + 3$ 轉換後到了 7。

variance 直觀上代表的是 distribution 以重心為基準的分散程度。當線性轉換發生時，分散程度只跟倍數 a 有關係，偏移量 b 是不影響分散程度的。而分散程度影響的幅度是 a^2 倍。

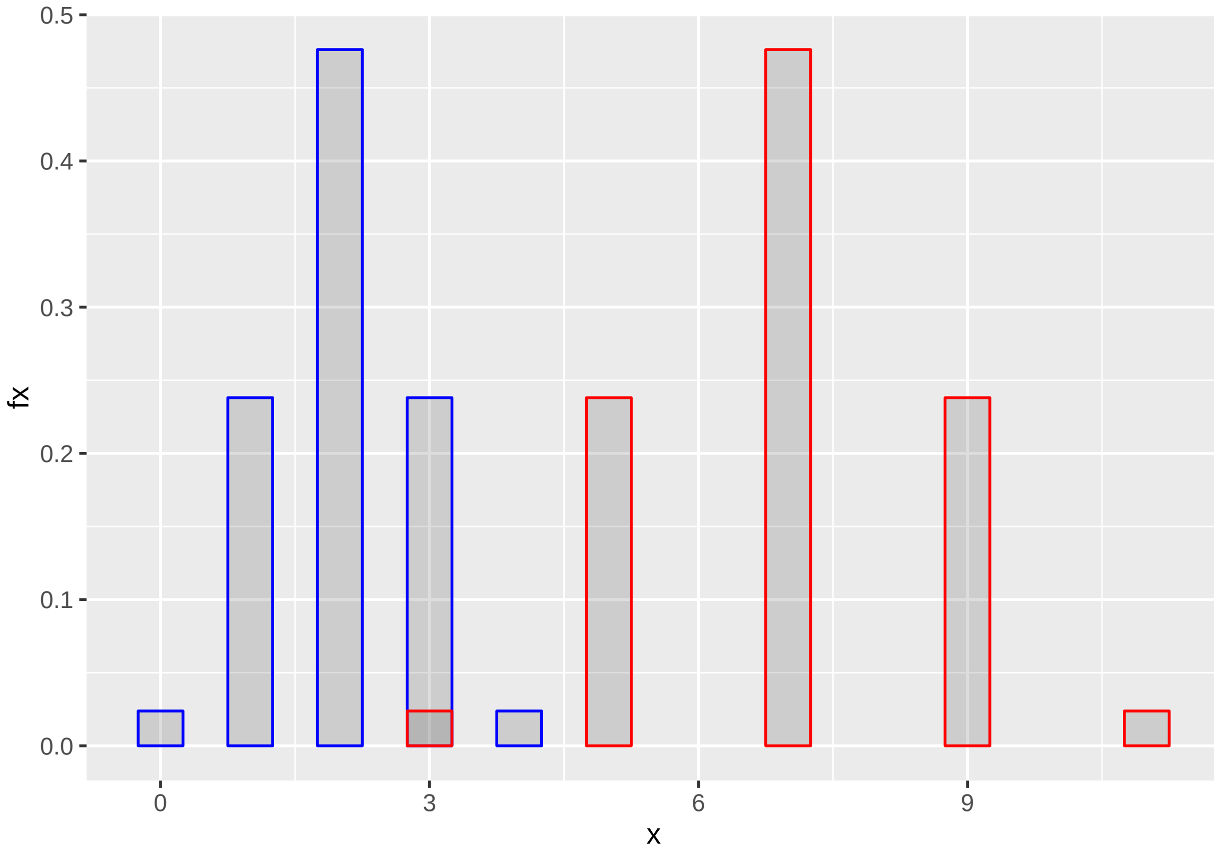
```r
g <- function(x) {2*x+3}
x <- ddply(x, .(x), transform, y=g(x))
sum(x$y*x$fx) == g(mean)
```

```
## [1] TRUE
```

```r
sum((x$y-mean(x$y))^2*x$fx) == 2^2 * var
```

```
## [1] TRUE
```

```r
library(ggplot2)
ggplot(x, aes(x=x, y=fx)) + geom_bar(stat="identity", color="blue", alpha=0.2, width=0.5)
  geom_bar(aes(x=y, y=fx), stat="identity", color="red", alpha=0.2, width=0.5)
```



Common Distribution Model

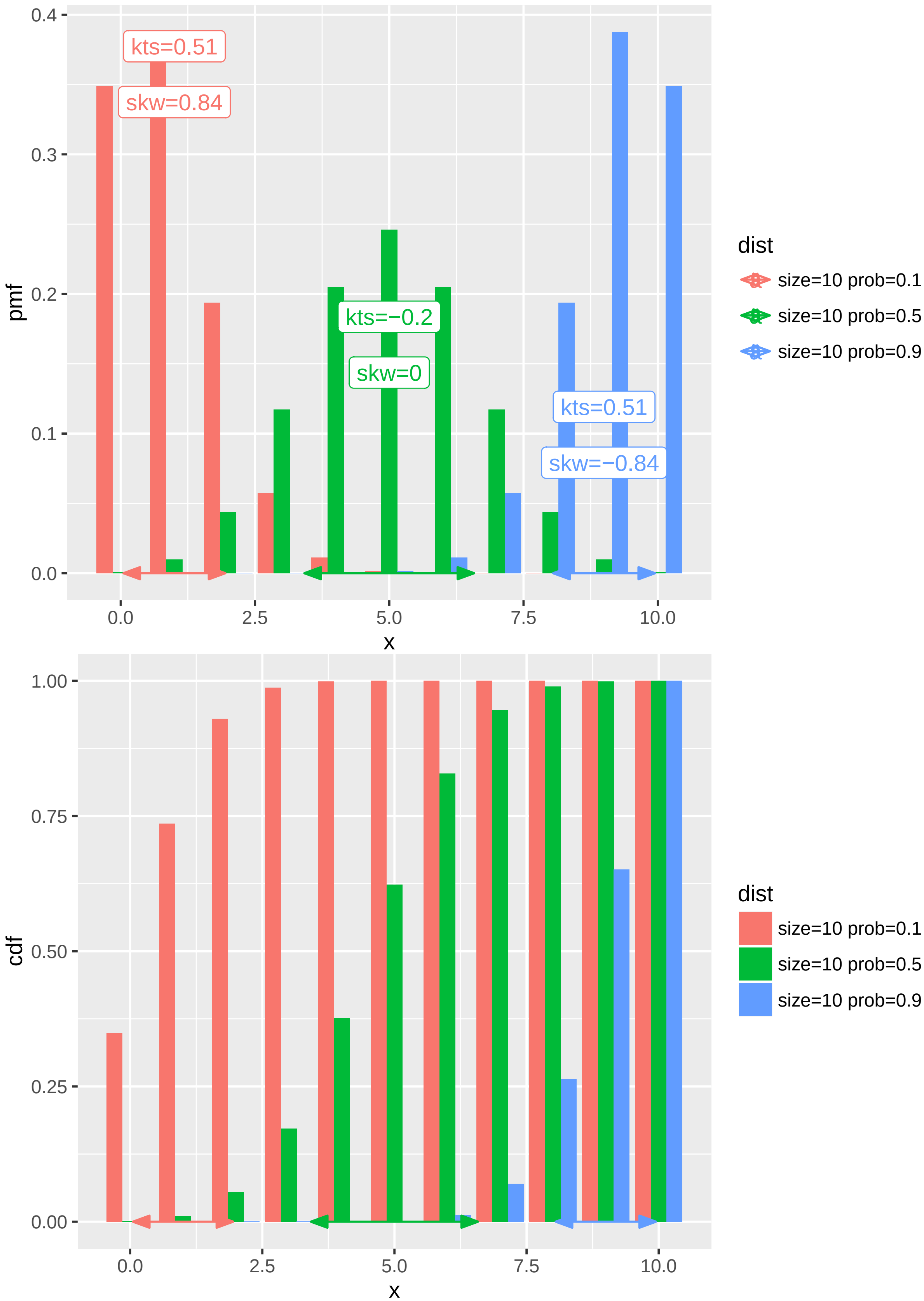
以下列出常見的 discrete distribution model，因為部份 R 的 distribution function 的參數和課堂上所假設的參數有些微差異，而日後實務上都是以 R 操作，所以所有討論描述都以 R 的 distribution function prototype 為主。

Binomial distribution

項目	內容
描述	每次嘗試的成功機率為 p，在 n 次嘗試中，成功幾次 若 n 為 1，則 Binomial 同等於 Bernoulli
參數對應	定義的 n= 函式的 size 定義的 p= 函式的 prob
Notation	$X \sim Binomial(n, p)$
Range	$x = \{0, 1, 2, ...n\}$
pmf	$f_X(x) = \binom{n}{x} p^x (1-p)^{n-x}, \forall x \in X$
Parameters	$n \in \{1, 2, 3, ...\} and\, 0 \le p \le 1$
Mean	np
Variance	$np(1-p)$
Skewness	$\frac{1-2p}{\sqrt{np(1-p)}}$

項目	內容
Kurtosis	$\frac{6p^2-6p+1}{np(1-p)}$
mgf	$M_X(t) = (1-p+pe^t)^n, \forall t < -\log(1-p)$
pmf	dbinom(x, size, prob, log = FALSE)
cdf	pbinom(q, size, prob, lower.tail = TRUE, log.p = FALSE)
qf	qbinom(p, size, prob, lower.tail = TRUE, log.p = FALSE)
rf	rbinom(n, size, prob)

以下為 Binomial distribution 的長相。若 prob 偏低，則機率高的會偏向成功次數低的；若 prob 偏高，則機率高的會偏向成功次數高的；若機率均等 (0.5)，則會呈現左右對稱。



Binomial 的應用練習．玩 9 次撲克牌，其中南方至少 5 次沒拿到 Ace 的機率為何? (LNp.5-20)

```r
# What is the probability that South gets no Aces on at least k=5 of n=9 hands?
prob <- 0.3038
sum(dbinom(c(5:9), 9, prob))
```

```
## [1] 0.103532
```

利用 qf 也可以反著問．玩 9 次撲克牌，若機率為 0.9，至少會有幾次沒拿到 Ace?

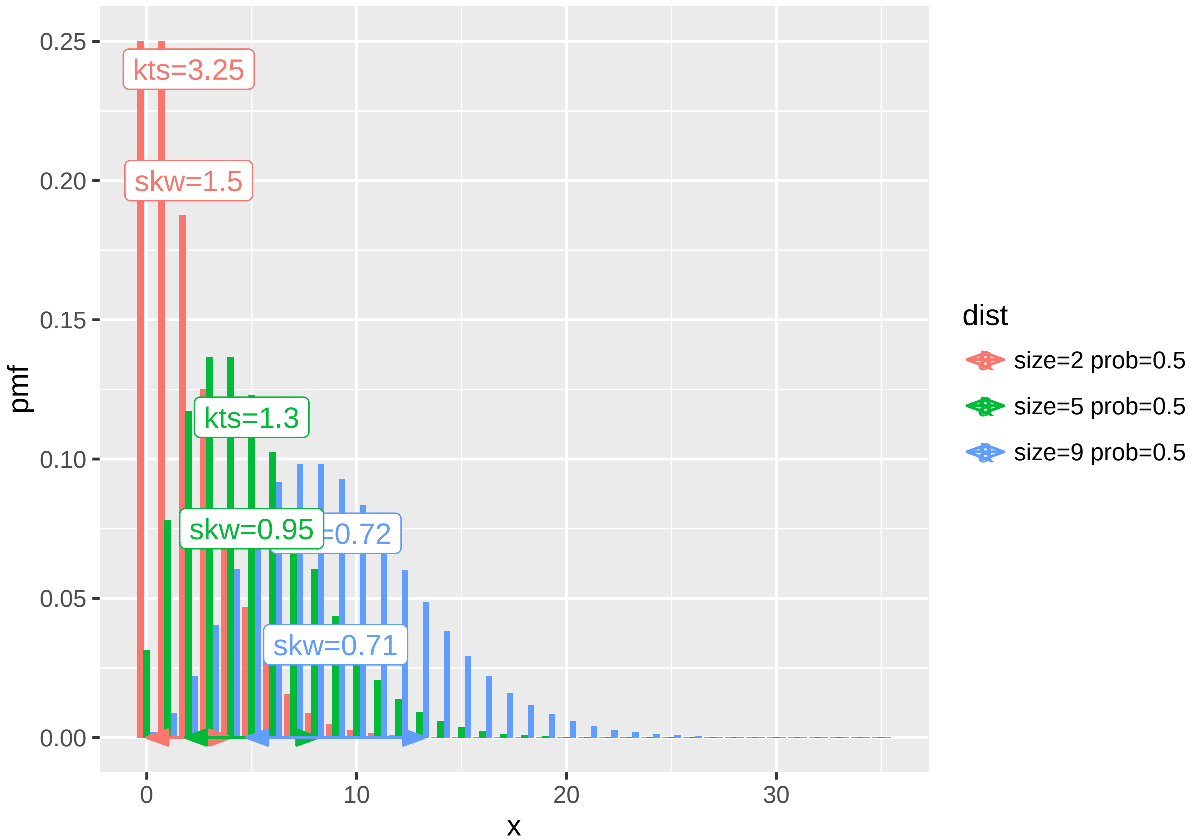
```r
# Given probability = 0.9, how many trails expected to have Aces at n=9 hands?
qbinom(0.9, 9, prob)
```

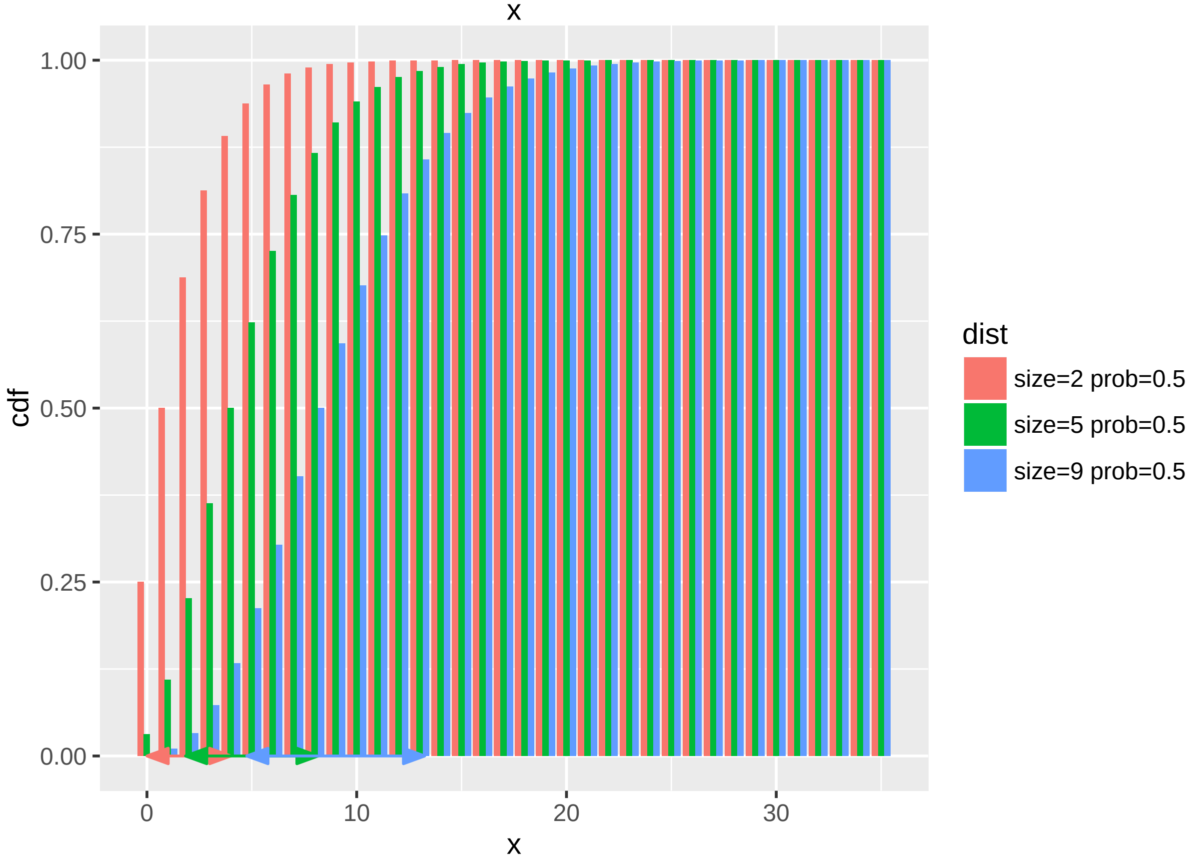
```
## [1] 5
```

Negative binomial distribution

項目	內容
描述	每次成功機率為 p，要求 r 次成功，總共需要失敗幾次 每次嘗試都視為獨立事件 若 r 為 1，則 Negative Binomial 同等於 Geometric
參數對應	定義的 r= 函式的 size 定義的 p= 函式的 prob
Notation	$X \sim NegativeBinomial(r, p)$
Range	$x = \{0, 1, 2, ...\}$
pmf	$f_X(x) = \binom{x+r-1}{r-1} p^r (1-p)^x, \forall x \in X$
Parameters	$r \in \{1, 2, 3, ...\} and\, 0 \leq p \leq 1$
Mean	$r/p - r$
Variance	$r(1-p)/p^2$
Skewness	$\frac{2-p}{\sqrt{r(1-p)}}$
Kurtosis	$\frac{6-p(6-p)}{r(1-p)}$
mgf	$M_X(t) = [\frac{pe^t}{1-(1-p)e^t}]^r, \forall t < -\log(1-p)$
pmf	dnbinom(x, size, prob, mu, log = FALSE)
cdf	pnbinom(q, size, prob, mu, lower.tail = TRUE, log.p = FALSE)
qf	qnbinom(p, size, prob, mu, lower.tail = TRUE, log.p = FALSE)
rf	rnbinom(n, size, prob, mu)
備註	課堂上的 X 是總共需要幾次，而 R 的版本是總共失敗幾次．在這都以 R 的版本調整所有公式．

下圖我們固定成功機率值，觀察要求成功次數不同對 distribution 的影響．我們可發現如果只要求 2 次成功，那麼機率值就會集中在 x 值比較小的部分．隨著要求成功的次數增加，所需要的總失敗次數也跟著增加．





Negative binomial 和 binomial 是有相關的，因著 Negative binomial 是第 size 次成功所需要嘗試的次數 (x)，可以想成在 x-1 次嘗試中

只要成功 size-1 次 (Binomial 問題)，然後再下一次一定成功的機率。

舉例來說，每次面試有 1/3 的機率錄取，如果要錄取 3 個人，總共剛好需要 10 次面試 (失敗 3 次) 的機率值是 0.078。這機率值同等於在 9 次面試中錄取 2 個人的機率，再乘上下一次成功錄取的機率 (1/3)。(LNp.5-27) 如以下所示：

```r
dnbinom(7, 3, 1/3)
```

```
## [1] 0.07803688
```

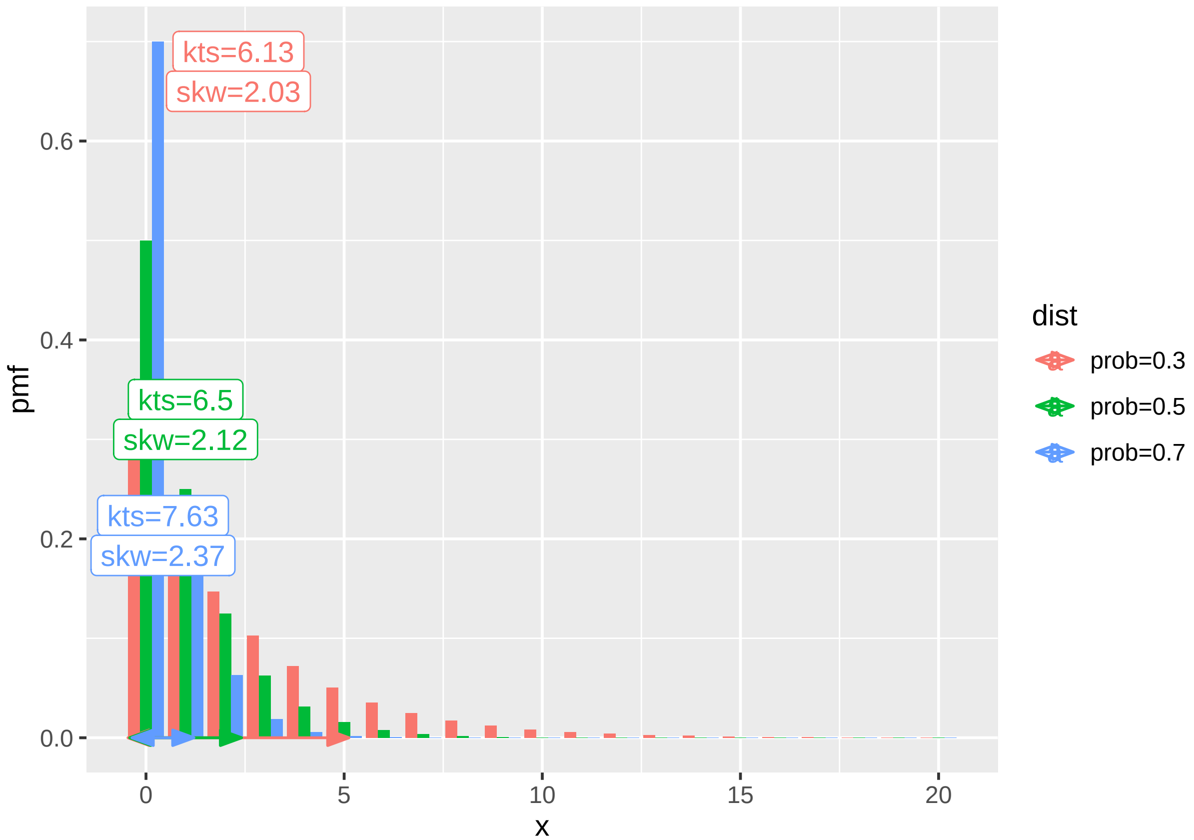
```r
dbinom(2, 9, 1/3) * (1/3)
```

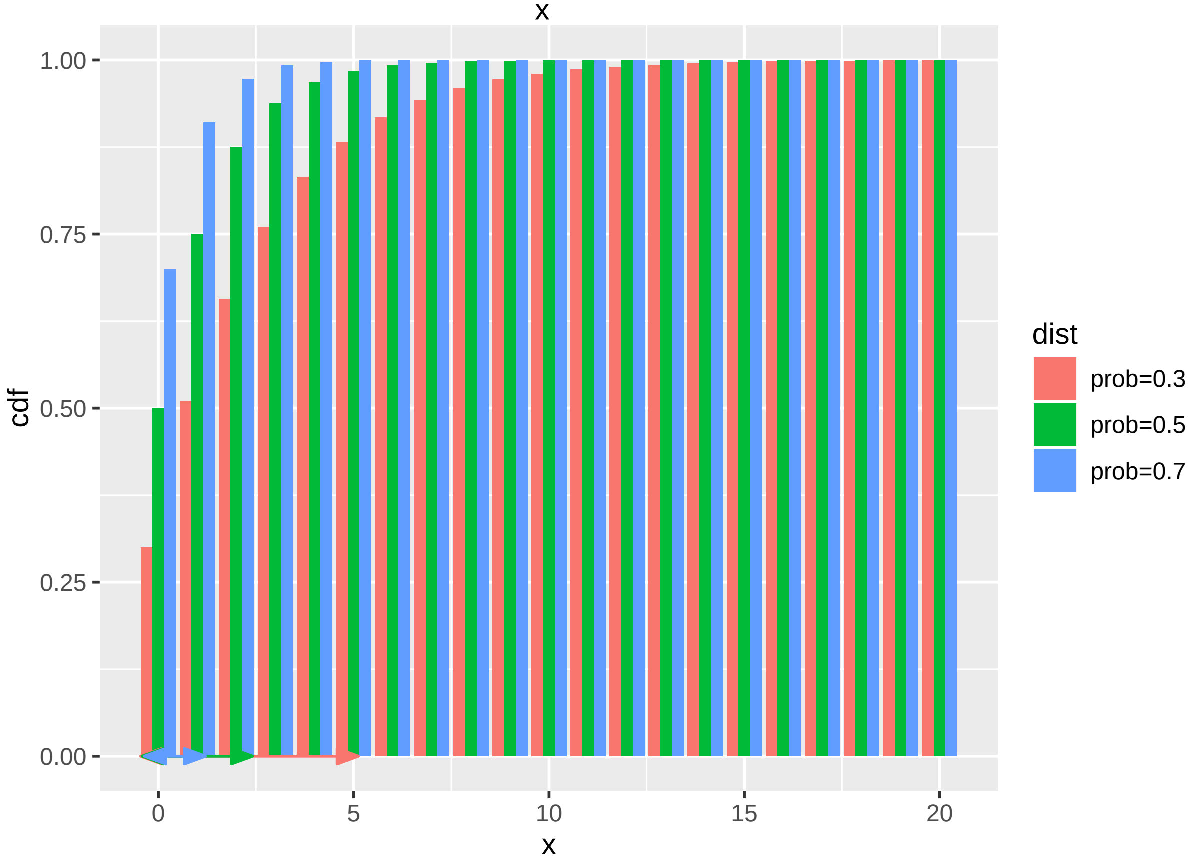
```
## [1] 0.07803688
```

Geometric distribution:

項目	內容
描述	每次成功機率為 p，要求 1 次成功，總共需要失敗幾次 每次嘗試都視為獨立事件，有 memoryless 特性 Negative Binomial 的 r 為 1 時的特例
參數對應	定義的 p= 函式的 prob
Notation	$Geometry(p)$
Range	$x = \{0, 1, 2, ...\}$
pmf	$f_X(x) = p(1-p)^{x-1}, \forall x \in X$
Parameters	$0 \leq p \leq 1$
Mean	$1/p - 1$
Variance	$(1-p)/p^2$
Skewness	$\frac{2-p}{\sqrt{1-p}}$
Kurtosis	$5 - p + \frac{1}{1-p}$
mgf	$M_X(t) = \frac{pe^t}{1-(1-p)e^t}, \forall t < -\log(1-p)$
pmf	dgeom(x, prob, log = FALSE)
cdf	pgeom(q, prob, lower.tail = TRUE, log.p = FALSE)
qf	qgeom(p, prob, lower.tail = TRUE, log.p = FALSE)
rf	rgeom(n, prob)
備註	課堂上的 X 是總共需要幾次，而 R 的版本是總共失敗幾次。在這都以 R 的版本調整所有公式。

下圖我們觀察成功機率值對 distribution 的影響。我們可發現如果成功機率偏高，那麼機率值就會集中在 x 值比較小的部分。如果成功機率減低，較大的 x 值就會出現，distribution 往右的尾巴會拉長。





Geometric 是第一次成功時所需要的嘗試次數，這其實就是當成功次數為 1 時的 Negative binomial。(LNp.5-29) 如下範例所示：

```r
dnbinom(9, 1, 1/3)
```

```
## [1] 0.008670765
```

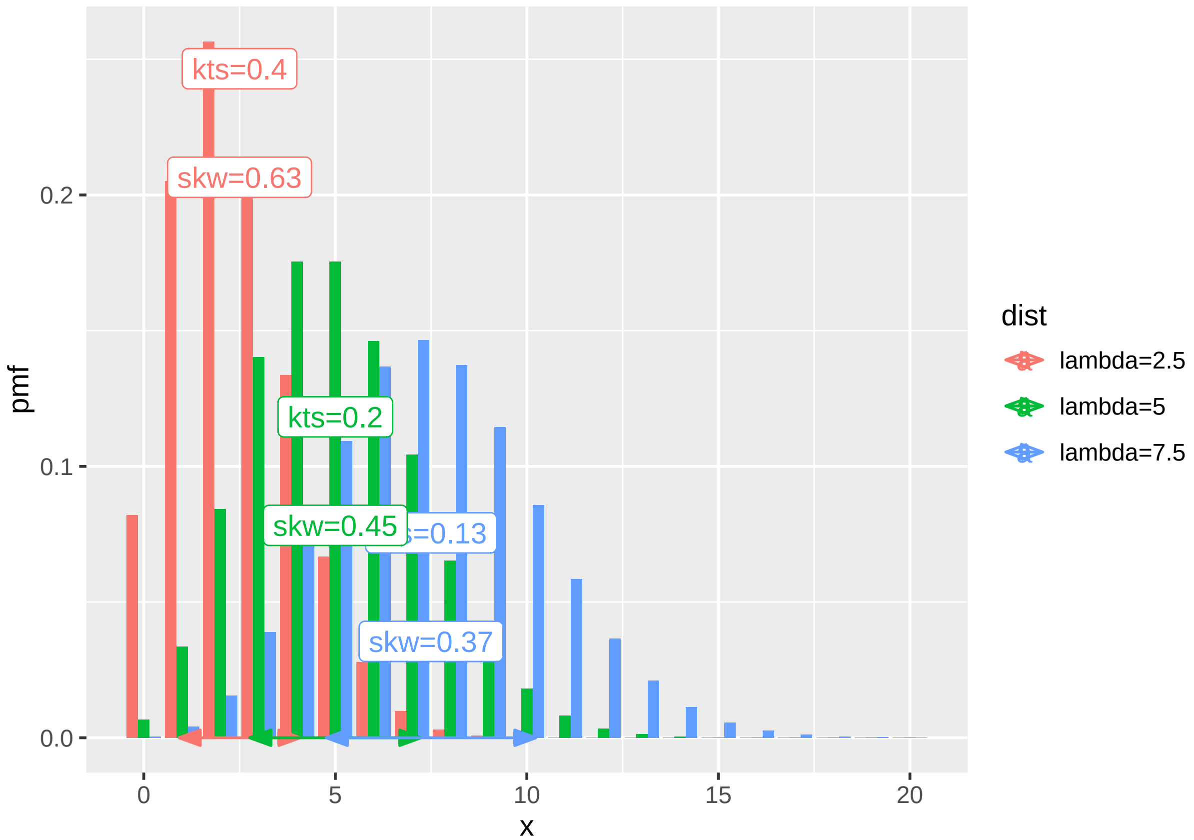
```r
dgeom(9, 1/3)
```

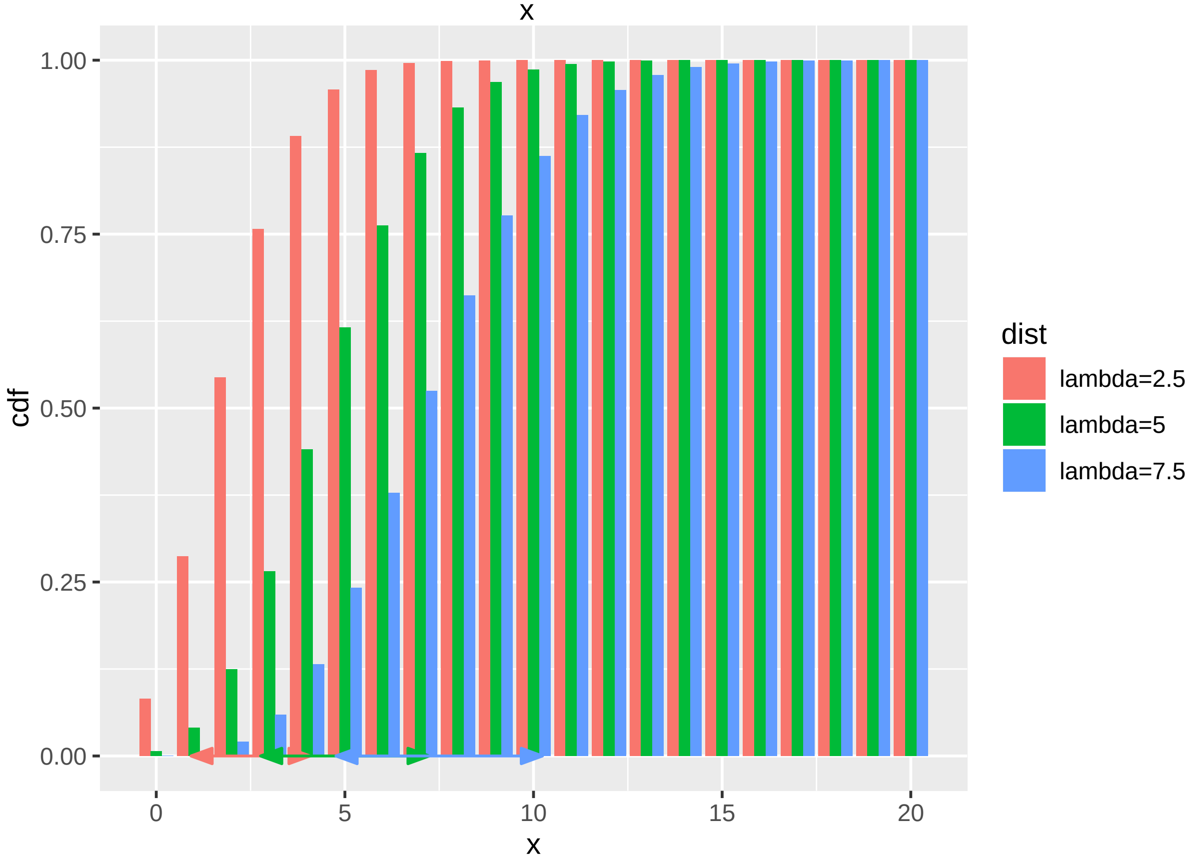
```
## [1] 0.008670765
```

Poisson distribution:

項目	內容
描述	每次嘗試的成功機率為 p，在 n 次嘗試中，成功幾次 必須條件: n 很大, p 很小 lambda = n x p 每次嘗試都視為獨立事件
Notation	$Poisson(\lambda)$
Range	$x = \{1, 2, 3, ...\}$
pmf	$f_X(x) = \lambda^x e^{-\lambda}/x!, \forall x \in X$
Parameters	$0 < \lambda < \infty$
Mean	λ
Variance	λ
Skewness	$\lambda^{\frac{-1}{2}}$
Kurtosis	$\frac{1}{\lambda}$
mgf	$M_X(t) = e^{\lambda(e^\lambda - 1)}$
pmf	dpois(x, lambda, log = FALSE)
cdf	ppois(q, lambda, lower.tail = TRUE, log.p = FALSE)
qf	qpois(p, lambda, lower.tail = TRUE, log.p = FALSE)
rf	rpois(n, lambda)
備註	課堂上的參數和 R 有些不同，在這都以 R 的版本調整所有公式。

由下圖看見，隨著成功率提升，成功次數也提升。


kts=0.4
skw=0.63
kts=0.2
skw=0.45
=0.13
skw=0.37
pmf
0.2
0.1
0.0
0
5
10
15
20
x
dist
lambda=2.5
lambda=5
lambda=7.5


cdf
1.00
0.75
0.50
0.25
0.00
0
5
10
15
20
x
dist
lambda=2.5
lambda=5
lambda=7.5

```r
# For verify the equation
theta
```

```
##   lambda mean       sd  skewness  kurtosis     sd_fr     sd_to       dist
## 1    2.5  2.5 1.581139 0.6324555 0.4000000 0.9188612  4.081139 lambda=2.5
## 2    5.0  5.0 2.236068 0.4472136 0.2000000 2.7639320  7.236068   lambda=5
## 3    7.5  7.5 2.738613 0.3651484 0.1333333 4.7613872 10.238613 lambda=7.5
```

```r
ddply(plot_data, .(dist), summarize, mean=sum(x*pmf), sd=sqrt(sum((x-mean)^2*pmf)), skewn
```

```
##         dist     mean       sd  skewness  kurtosis
## 1 lambda=2.5 2.500000 1.581139 0.6324555 0.4000000
## 2   lambda=5 4.999998 2.236063 0.4471873 0.1997988
## 3 lambda=7.5 7.499169 2.737227 0.3613991 0.1131608
```

A professor hits the wrong key with probability p=0.001 each time he types a letter. Assume independence for the occurrence of errors between different letter typings. What's the probability that 5 or more errors in n=2500 letters. (LNp.5-33)

```r
1-sum( dpois(c(0:4), 2500 * 0.001) )
```

```
## [1] 0.108822
```

Traffic accident occurs according to a Poisson process at a rate of 5.5 per month. What is the probability of 3 or more accidents occur in a 2 month periods?

```r
1-sum(dpois(c(0:2), lambda = 5.5 * 2))
```

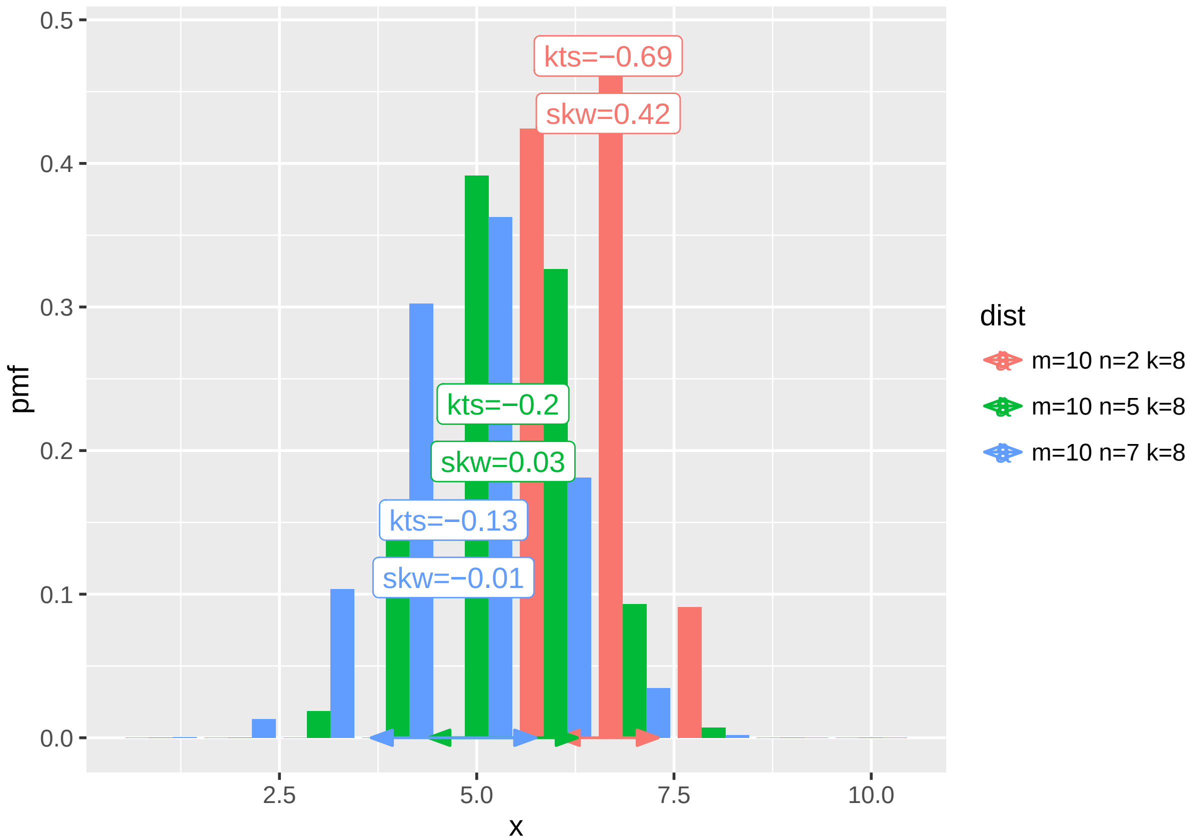
```
## [1] 0.9987891
```

Hypergeometric distribution:

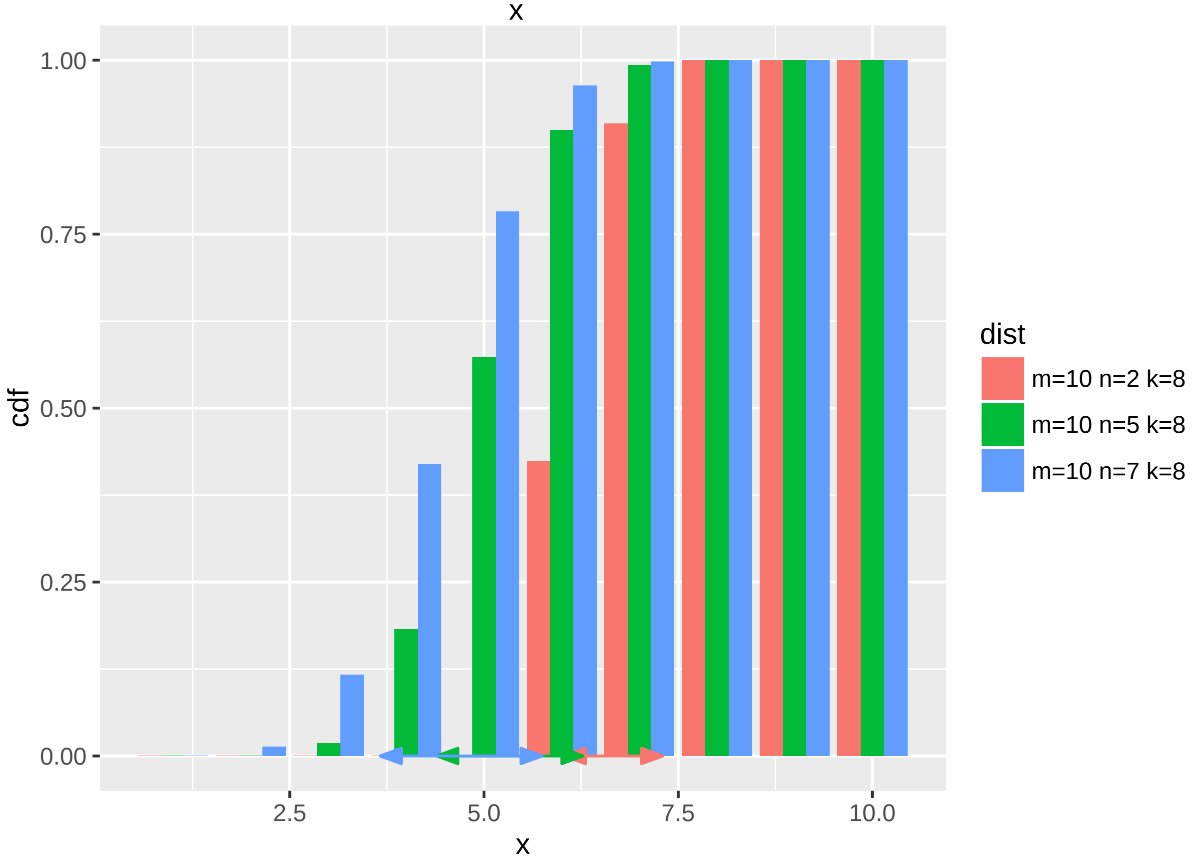
項目	內容
描述	盒子中共有 m 個白球、n 個黑球，從盒子中要抽 k 球 (抽球後不放回)，其中幾個白球 抽球後不放回，每次嘗試的成功機率不是獨立事件 若抽球有放回，則是 Binomial 運用在工業上，用來了解產品瑕疵率 m: 壞掉的產品 n: 好的產品 k: 抽驗產品數
Notation	$Hypergeometric(m, n, k)$
Range	$x = \{0, 1, 2, ..., n\}$
pmf	$f_X(x) = \binom{m}{x}\binom{n}{k-x}/\binom{m+n}{k}, \forall x \in X$
Parameters	$m, n, k \in \{1, 2, 3, ...\} and\ k \leq m + n$
Mean	$km/(m + n)$
Variance	$\frac{kmn(m+n-k)}{(m+n)^2(m+n-1)}$
Skewness	$\frac{-(m-n)(m+n-2k)}{m+n-2}\sqrt{\frac{m+n-1}{mnk(m+n-k)}}$

項目	內容
Kurtosis	$\frac{1}{kmn(m+n-k)(m+n-2)(m+n-3)}[(m+n-1)(m+n)^2((m+n)(m+n+1)-6mn-6k(m+n-k))+6kmn(m+n-k)(5(m+n)-6)]$
mgf	
pmf	dhyper(x, m, n, k, log = FALSE)
cdf	phyper(q, m, n, k, lower.tail = TRUE, log.p = FALSE)
qf	qhyper(p, m, n, k, lower.tail = TRUE, log.p = FALSE)
rf	rhyper(nn, m, n, k)

以下我們固定白球和抽球次數，但逐漸增加黑球數目來觀察 distribution 變化．隨著黑球數目增加時，抽到白球的機率逐漸下降，機率值往 x 小的地方偏移．



kts=−0.69
skw=0.42
kts=−0.2
skw=0.03
kts=−0.13
skw=−0.01
pmf
0.5
0.4
0.3
0.2
0.1
0.0
2.5
5.0
7.5
10.0
x
dist
m=10 n=2 k=8
m=10 n=5 k=8
m=10 n=7 k=8



cdf
1.00
0.75
0.50
0.25
0.00
2.5
5.0
7.5
10.0
x
dist
m=10 n=2 k=8
m=10 n=5 k=8
m=10 n=7 k=8