

Data Visualization

Lecture 5 Mathamatics Visualization

Phuc Loi Luu, PhD
p.luu@garvan.org.au
luu.p.loi@googlemail.com

Content

Plot a mathematics graph

- Introduction to Probability and Statistics
- Create a ShinyApp

Some important discrete distribution

1. Binomial distribution

Notation: $X \sim \text{Binomial}(n, p)$.

Description: number of successes in n independent trials, each with probability p of success.

Probability function:

$$f_X(x) = \mathbb{P}(X = x) = \binom{n}{x} p^x (1-p)^{n-x} \quad \text{for } x = 0, 1, \dots, n.$$

Mean: $\mathbb{E}(X) = np$.

Variance: $\text{Var}(X) = np(1-p) = npq$, where $q = 1 - p$.

Sum: If $X \sim \text{Binomial}(n, p)$, $Y \sim \text{Binomial}(m, p)$, and X and Y are independent, then

$$X + Y \sim \text{Bin}(n + m, p).$$

2. Poisson distribution

Notation: $X \sim \text{Poisson}(\lambda)$.

Description: arises out of the Poisson process as the number of events in a fixed time or space, when events occur at a constant average rate. Also used in many other situations.

Probability function: $f_X(x) = \mathbb{P}(X = x) = \frac{\lambda^x}{x!} e^{-\lambda}$ for $x = 0, 1, 2, \dots$

Mean: $\mathbb{E}(X) = \lambda$.

Variance: $\text{Var}(X) = \lambda$.

Sum: If $X \sim \text{Poisson}(\lambda)$, $Y \sim \text{Poisson}(\mu)$, and X and Y are independent, then

$$X + Y \sim \text{Poisson}(\lambda + \mu).$$

3. Geometric distribution

Notation: $X \sim \text{Geometric}(p)$.

Description: number of failures before the first success in a sequence of independent trials, each with $\mathbb{P}(\text{success}) = p$.

Probability function: $f_X(x) = \mathbb{P}(X = x) = (1-p)^x p \quad \text{for } x = 0, 1, 2, \dots$

Mean: $\mathbb{E}(X) = \frac{1-p}{p} = \frac{q}{p}$, where $q = 1 - p$.

Variance: $\text{Var}(X) = \frac{1-p}{p^2} = \frac{q}{p^2}$, where $q = 1 - p$.

Sum: if X_1, \dots, X_k are independent, and each $X_i \sim \text{Geometric}(p)$, then
 $X_1 + \dots + X_k \sim \text{Negative Binomial}(k, p)$.

4. Negative Binomial distribution

Notation: $X \sim \text{NegBin}(k, p)$.

Description: number of failures before the kth success in a sequence of independent trials, each with $\mathbb{P}(\text{success}) = p$.

Probability function:

$$f_X(x) = \mathbb{P}(X = x) = \binom{k+x-1}{x} p^k (1-p)^x \quad \text{for } x = 0, 1, 2, \dots$$

Mean: $\mathbb{E}(X) = \frac{k(1-p)}{p} = \frac{kq}{p}$, where $q = 1 - p$.

Variance: $\text{Var}(X) = \frac{k(1-p)}{p^2} = \frac{kq}{p^2}$, where $q = 1 - p$.

Sum: If $X \sim \text{NegBin}(k, p)$, $Y \sim \text{NegBin}(m, p)$, and X and Y are independent, then

$$X + Y \sim \text{NegBin}(k+m, p).$$

Some important discrete distribution

5. Hypergeometric distribution

Notation: $X \sim \text{Hypergeometric}(N, M, n)$.

Description: Sampling without replacement from a finite population. Given N objects, of which M are ‘special’. Draw n objects without replacement. X is the number of the n objects that are ‘special’.

Probability function:

$$f_X(x) = \mathbb{P}(X = x) = \frac{\binom{M}{x} \binom{N-M}{n-x}}{\binom{N}{n}} \quad \text{for } \begin{cases} x = \max(0, n + M - N) \\ \text{to } x = \min(n, M). \end{cases}$$

Mean: $\mathbb{E}(X) = np$, where $p = \frac{M}{N}$.

Variance: $\text{Var}(X) = np(1-p)\left(\frac{N-n}{N-1}\right)$, where $p = \frac{M}{N}$.

6. Multinomial distribution

Notation: $\mathbf{X} = (X_1, \dots, X_k) \sim \text{Multinomial}(n; p_1, p_2, \dots, p_k)$.

Description: there are n independent trials, each with k possible outcomes. Let $p_i = \mathbb{P}(\text{outcome } i)$ for $i = 1, \dots, k$. Then $\mathbf{X} = (X_1, \dots, X_k)$, where X_i is the number of trials with outcome i , for $i = 1, \dots, k$.

Probability function:

$$f_{\mathbf{X}}(\mathbf{x}) = \mathbb{P}(X_1 = x_1, \dots, X_k = x_k) = \frac{n!}{x_1! \dots x_k!} p_1^{x_1} p_2^{x_2} \dots p_k^{x_k}$$

for $x_i \in \{0, \dots, n\} \forall_i$ with $\sum_{i=1}^k x_i = n$, and where $p_i \geq 0 \forall_i$, $\sum_{i=1}^k p_i = 1$.

Marginal distributions: $X_i \sim \text{Binomial}(n, p_i)$ for $i = 1, \dots, k$.

Mean: $\mathbb{E}(X_i) = np_i$ for $i = 1, \dots, k$.

Variance: $\text{Var}(X_i) = np_i(1-p_i)$, for $i = 1, \dots, k$.

Covariance: $\text{cov}(X_i, X_j) = -np_ip_j$, for all $i \neq j$.

Revisit: Hypothesis testing

Example: Weird Coin?

I toss a coin 10 times and get 9 heads. How weird is that?



What is ‘weird’?

- Getting 9 heads out of 10 tosses: we’ll call this *weird*.
- Getting 10 heads out of 10 tosses: *even more weird!*
- Getting 8 heads out of 10 tosses: *less weird*.
- Getting 1 head out of 10 tosses: *same as getting 9 tails out of 10 tosses: just as weird as 9 heads if the coin is fair.*
- Getting 0 heads out of 10 tosses: *same as getting 10 tails: more weird than 9 heads if the coin is fair.*

Set of weird outcomes

If our coin is fair, the outcomes that are *as weird or weirder* than 9 heads are:

9 heads, 10 heads, 1 head, 0 heads.

So how weird is 9 heads or worse, if the coin is fair?

Define $X = \#\text{heads out of 10 tosses}$.

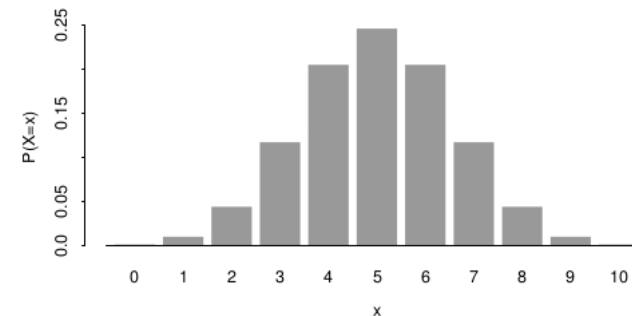
Distribution of X , if the coin is fair: $X \sim \text{Binomial}(n = 10, p = 0.5)$.

Probability of observing something at least as weird as 9 heads, if the coin is fair:

We can add the probabilities of all the outcomes that are *at least as weird* as 9 heads out of 10 tosses, assuming that the coin is fair.

$$\mathbb{P}(X = 9) + \mathbb{P}(X = 10) + \mathbb{P}(X = 1) + \mathbb{P}(X = 0) \quad \text{where } X \sim \text{Binomial}(10, 0.5).$$

Probabilities for $\text{Binomial}(n = 10, p = 0.5)$



For $X \sim \text{Binomial}(10, 0.5)$, we have:

$$\mathbb{P}(X = 9) + \mathbb{P}(X = 10) + \mathbb{P}(X = 1) + \mathbb{P}(X = 0) =$$

$$\begin{aligned} & \binom{10}{9}(0.5)^9(0.5)^1 + \binom{10}{10}(0.5)^{10}(0.5)^0 + \\ & \binom{10}{1}(0.5)^1(0.5)^9 + \binom{10}{0}(0.5)^0(0.5)^{10} \end{aligned}$$

$$= 0.00977 + 0.00098 + 0.00977 + 0.00098$$

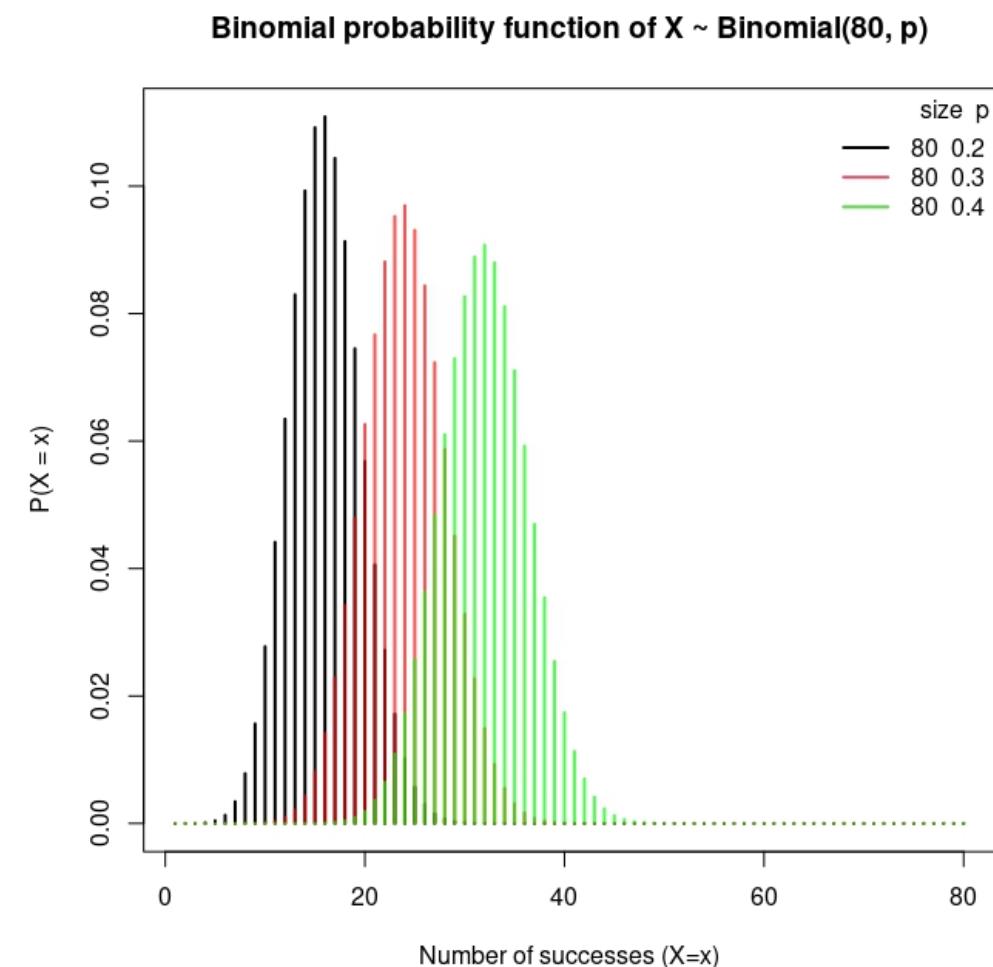
$$= 0.021.$$

The binomial distribution

- The **probability mass function** (PMF) is $P(X = x) = \binom{n}{x} p^x q^{n-x}$ if $x = 0, 1, 2, \dots, n$.
- The cumulative **distribution function** (CDF) is $F(x) = I_q(1 - x, n - x)$.
- The **quantile function** is $Q(p) = F^{-1}(p)$.
- The **expected mean and variance** of X are $E(X) = np$ and $Var(X) = npq$, respectively.

The functions of the previous lists can be computed in R for a set of values with the `dbinom` (probability), `pbinom` (distribution) and `qbinom` (quantile) functions. In addition, the `rbinom` function allows drawing n random samples from a binomial distribution in R. The following table describes briefly these R functions.

Function	Description
<code>dbinom</code>	Binomial probability mass function (Probability function)
<code>pbinom</code>	Binomial distribution (Cumulative distribution function)
<code>qbinom</code>	Binomial quantile function
<code>rbinom</code>	Binomial pseudorandom number generation



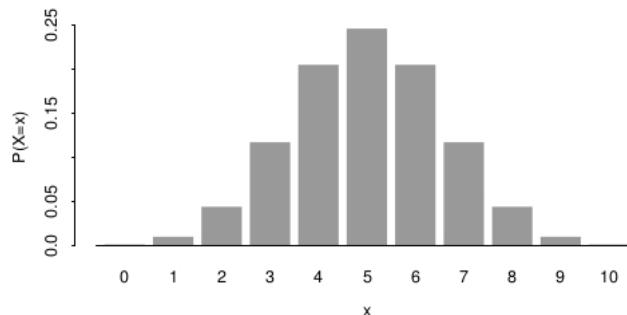
The binomial distribution

Probability of observing something at least as weird as 9 heads, if the coin is fair:

We can add the probabilities of all the outcomes that are *at least as weird* as 9 heads out of 10 tosses, assuming that the coin is fair.

$$\mathbb{P}(X = 9) + \mathbb{P}(X = 10) + \mathbb{P}(X = 1) + \mathbb{P}(X = 0) \quad \text{where } X \sim \text{Binomial}(10, 0.5).$$

Probabilities for Binomial($n = 10, p = 0.5$)



For $X \sim \text{Binomial}(10, 0.5)$, we have:

$$\begin{aligned}\mathbb{P}(X = 9) + \mathbb{P}(X = 10) + \mathbb{P}(X = 1) + \mathbb{P}(X = 0) &= \\ \binom{10}{9}(0.5)^9(0.5)^1 + \binom{10}{10}(0.5)^{10}(0.5)^0 + \\ \binom{10}{1}(0.5)^1(0.5)^9 + \binom{10}{0}(0.5)^0(0.5)^{10} &= \\ 0.00977 + 0.00098 + 0.00977 + 0.00098 &= \\ 0.021. &\end{aligned}$$

The R command, **dbinom** function calculate the binomial probability function for a set of values $X=x$, a number of trials n and a probability of success p :

`dbinom(x=0, size=10, prob=0.5)=P(X=0)=0.000977
where X ~ Binomial(10, 0.5)`

```
dbinom(x=0, size=10, prob=0.5)
# [1] 0.0009765625
dbinom(x=1, size=10, prob=0.5)
# [1] 0.009765625
dbinom(x=9, size=10, prob=0.5)
# [1] 0.009765625
dbinom(x=10, size=10, prob=0.5)
# [1] 0.0009765625
x <- c(0, 1, 9, 10)
sum(sapply(x, function(x) dbinom(x=x, size=10,
prob=0.5)))
# [1] 0.02148438
```

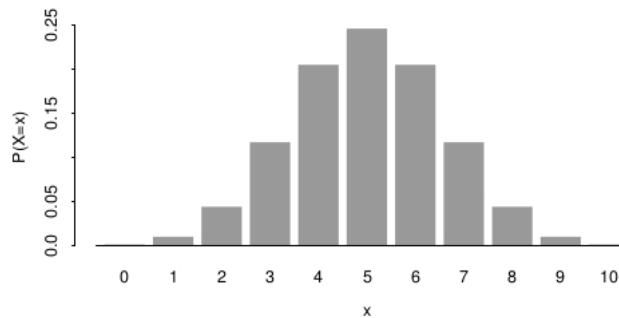
The binomial probability function

Probability of observing something at least as weird as 9 heads,
if the coin is fair:

We can add the probabilities of all the outcomes that are *at least as weird* as 9 heads out of 10 tosses, assuming that the coin is fair.

$$\mathbb{P}(X = 9) + \mathbb{P}(X = 10) + \mathbb{P}(X = 1) + \mathbb{P}(X = 0) \quad \text{where } X \sim \text{Binomial}(10, 0.5).$$

Probabilities for Binomial($n = 10, p = 0.5$)



For $X \sim \text{Binomial}(10, 0.5)$, we have:

$$\begin{aligned} \mathbb{P}(X = 9) + \mathbb{P}(X = 10) + \mathbb{P}(X = 1) + \mathbb{P}(X = 0) &= \\ \binom{10}{9}(0.5)^9(0.5)^1 + \binom{10}{10}(0.5)^{10}(0.5)^0 + \\ \binom{10}{1}(0.5)^1(0.5)^9 + \binom{10}{0}(0.5)^0(0.5)^{10} &= \\ 0.00977 + 0.00098 + 0.00977 + 0.00098 &= \\ 0.021. \end{aligned}$$

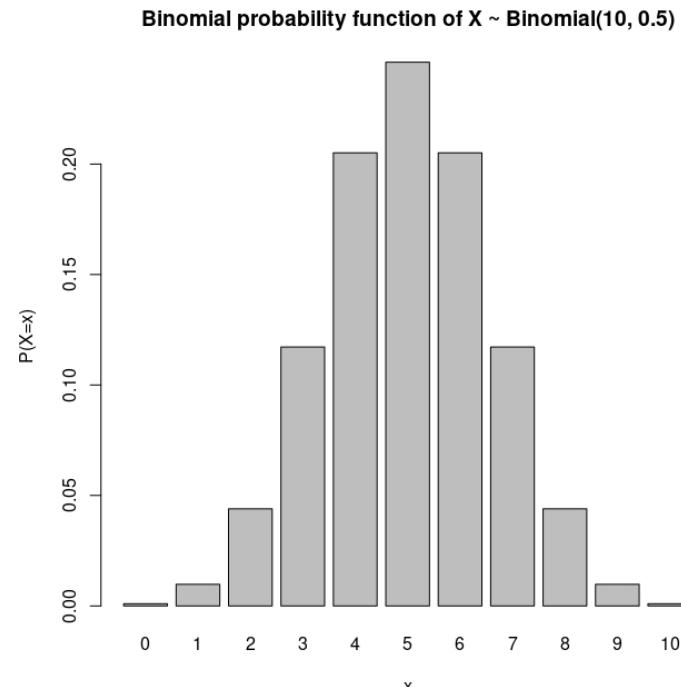
The R command, **dbinom** function calculate the binomial probability function for a set of values $X=x$, a number of trials n and a probability of success p :

`x <- seq(0, 10, 1)`

`d <- dbinom(x=x, size=10, prob=0.5)`

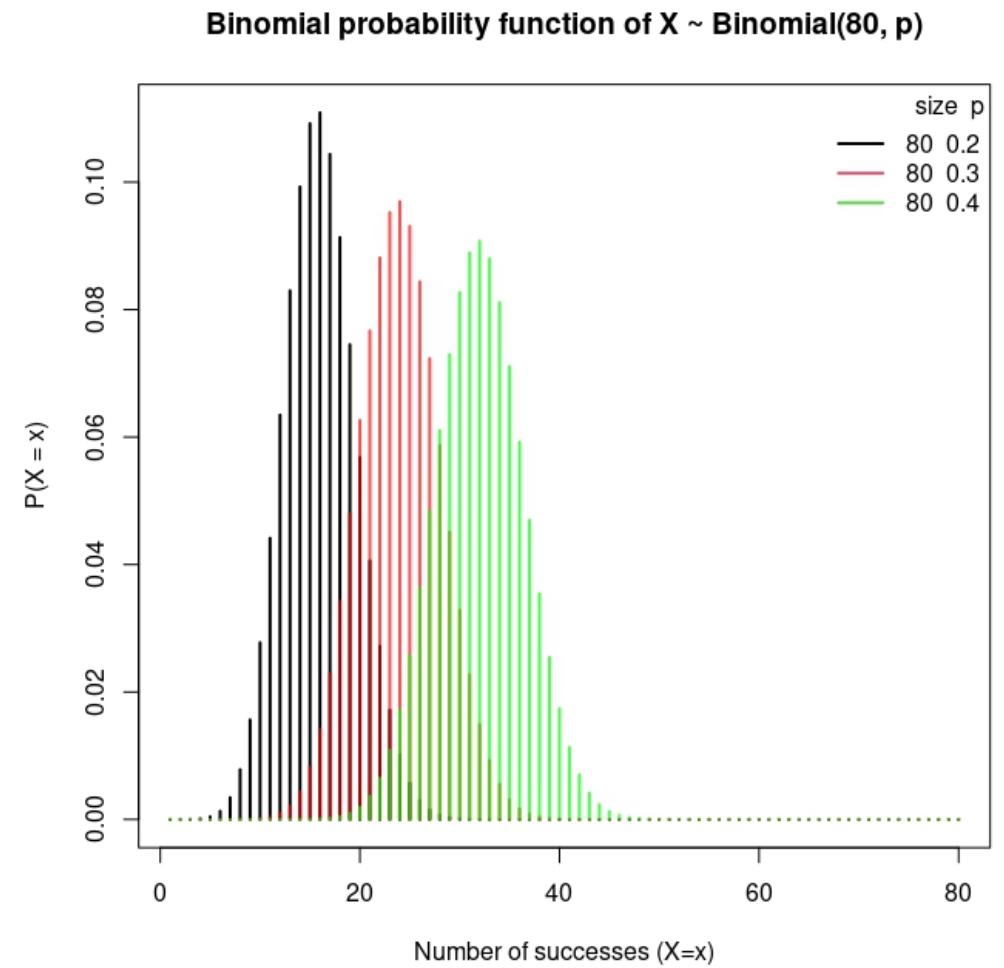
`names(d) <- x`

`barplot(d, main="Binomial probability function of X ~ Binomial(10, 0.5)", ylab="P(X=x)", xlab="x")`



The binomial probability function

```
# The R command, plot Binomial probability function  
# Grid of X-axis values  
x <- 1:80  
  
# size=80, prob=0.2  
plot(dbinom(x, size=80, prob=0.2), type="h", lwd=2,  
      main="Binomial probability function of  $X \sim \text{Binomial}(80, p)$ ",  
      ylab="P( $X=x$ )", xlab="Number of successes ( $X=x$ )")  
  
# size=80, prob=0.3  
lines(dbinom(x, size=80, prob=0.3), type="h",  
      lwd=2, col=rgb(1,0,0, 0.7))  
  
# size=80, prob=0.4  
lines(dbinom(x, size=80, prob=0.4), type="h",  
      lwd=2, col=rgb(0, 1, 0, 0.7))  
  
# Add a legend  
legend("topright", legend=c("80 0.2", "80 0.3", "80 0.4"),  
      title="size p", title.adj=0.95,  
      lty=1, col=1:3, lwd=2, box.lty=0)
```



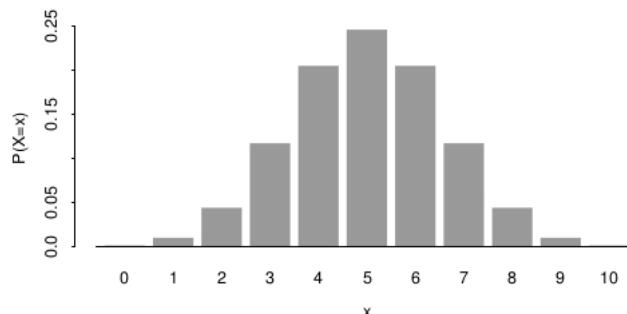
Hypothesis testing with R

Probability of observing something at least as weird as 9 heads, if the coin is fair:

We can add the probabilities of all the outcomes that are *at least as weird* as 9 heads out of 10 tosses, assuming that the coin is fair.

$$\mathbb{P}(X = 9) + \mathbb{P}(X = 10) + \mathbb{P}(X = 1) + \mathbb{P}(X = 0) \quad \text{where } X \sim \text{Binomial}(10, 0.5).$$

Probabilities for Binomial($n = 10, p = 0.5$)



For $X \sim \text{Binomial}(10, 0.5)$, we have:

$$\begin{aligned} \mathbb{P}(X = 9) + \mathbb{P}(X = 10) + \mathbb{P}(X = 1) + \mathbb{P}(X = 0) &= \\ \binom{10}{9}(0.5)^9(0.5)^1 + \binom{10}{10}(0.5)^{10}(0.5)^0 + \\ \binom{10}{1}(0.5)^1(0.5)^9 + \binom{10}{0}(0.5)^0(0.5)^{10} &= \\ 0.00977 + 0.00098 + 0.00977 + 0.00098 &= \\ 0.021. \end{aligned}$$

- The R command, **pbinom** calculate the probability of a variable X following a binomial distribution taking values lower than or equal to x
- The **pbinom** is equivalent to the **cumulative distribution function** for the Binomial distribution:

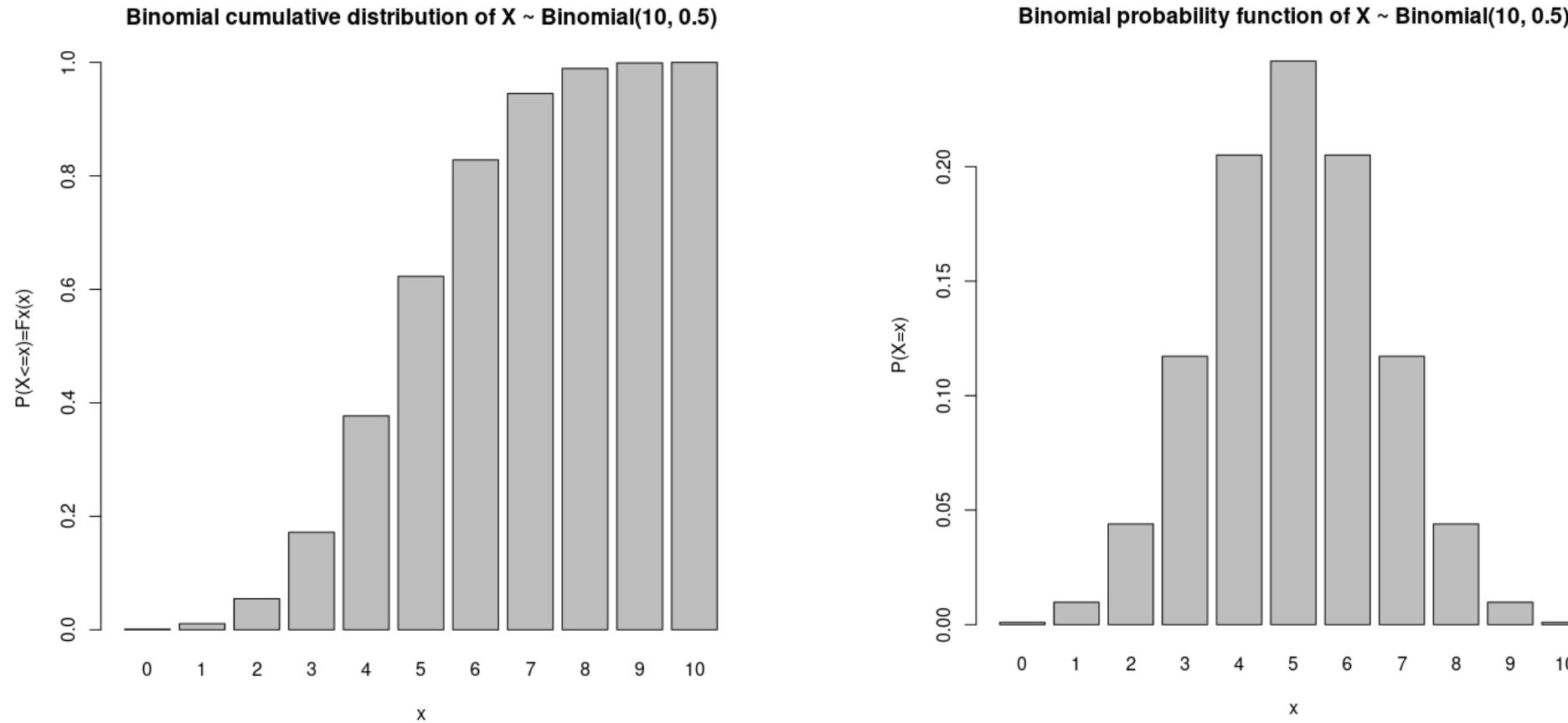
- $\text{pbinom}(q=0, \text{size}=10, \text{prob}=0.5) = \mathbb{P}(X \leq 0) = F_X(0)$
where $X \sim \text{Binomial}(10, 0.5)$

```
pb $\_$ inom(q=0, size=10, prob=0.5)  
# [1] 0.0009765625
```

- For $X \sim \text{Binomial}(10, 0.5)$. The overall p-value in this example is $2 * F_X(1)$.

```
p $\_$ value <- 2*pb $\_$ inom(q=1, size=10, 0.5)  
# [1] 0.02148438
```

The binomial cumulative function



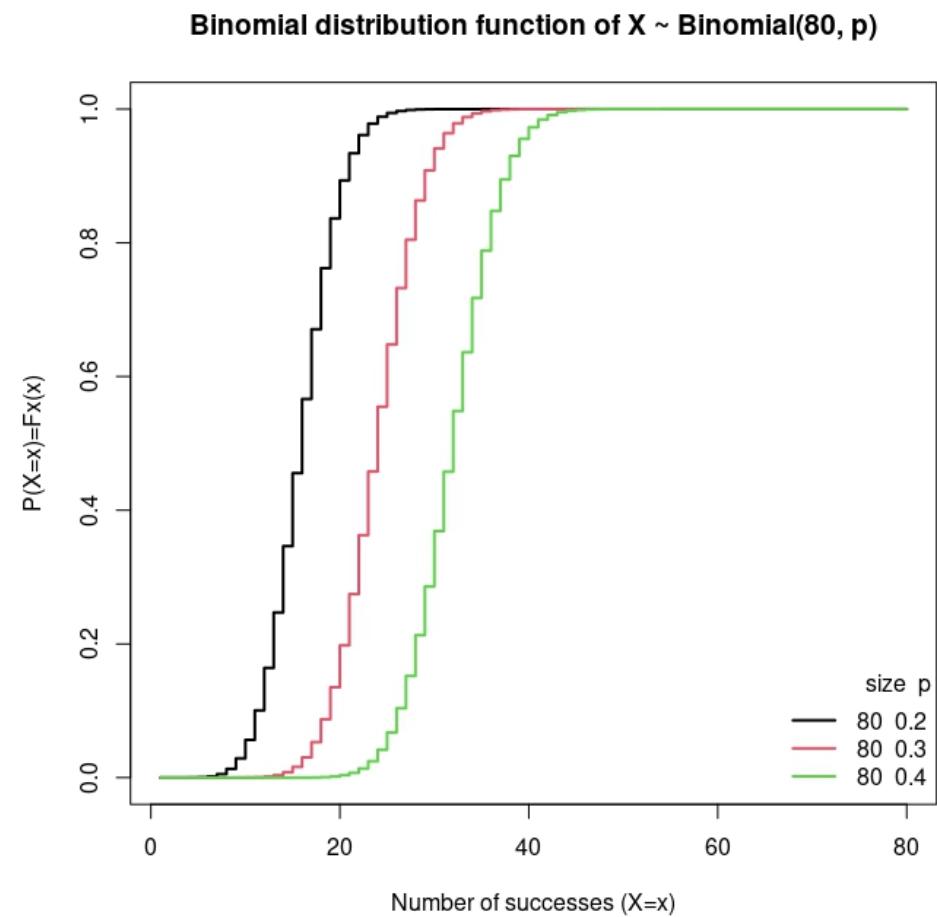
- Binomial cumulative distribution of $X \sim \text{Binomial}(10, 0.5)$

```
x <- seq(0, 10, 1)
acc <- pbinom(q=x, size=10, prob=0.5)
names(acc) <- x
barplot(acc, main="Binomial cumulative distribution of X ~ Binomial(10, 0.5)", ylab="P(X<=x)=Fx(x)", xlab="x")
```

The binomial cumulative function

- Binomial cumulative distribution of $X \sim \text{Binomial}(80, p)$

```
# Grid of X-axis values  
x <- 1:80  
  
# size=80, prob=0.2  
plot(pbinom(x, size=80, prob=0.2), type="s", lwd=2,  
     main="Binomial distribution function of X ~ Binomial(80, p)",  
     xlab="Number of successes (X=x)", ylab="P(X=x)=Fx(x)")  
  
# size=80, prob=0.3  
lines(pbinom(x, size=80, prob=0.3), type="s", lwd=2, col=2)  
  
# size=80, prob=0.4  
lines(pbinom(x, size=80, prob=0.4), type="s", lwd=2, col=3)  
  
# Add a legend  
legend("bottomright", legend=c("80 0.2", "80 0.3", "80 0.4"),  
      title="size p", title.adj=0.95, lty=1, col=1:3, lwd=2, box.lty=0)
```



The binomial cumulative function

```
# size: number of trials (n >=0)
# prob: probability of success on each trial
# lb: lower bound of the sum
# ub: upper bound of the sum
# col: color
# lwd: line width
binom_sum <- function(size, prob, lb, ub, col=4, lwd=1, ...) {
  x <- 0:size

  if (missing(lb)) {
    lb <- min(x)
  }
  if (missing(ub)) {
    ub <- max(x)
  }

  plot(dbinom(x, size=size, prob=prob), type="h", lwd=lwd, ...)

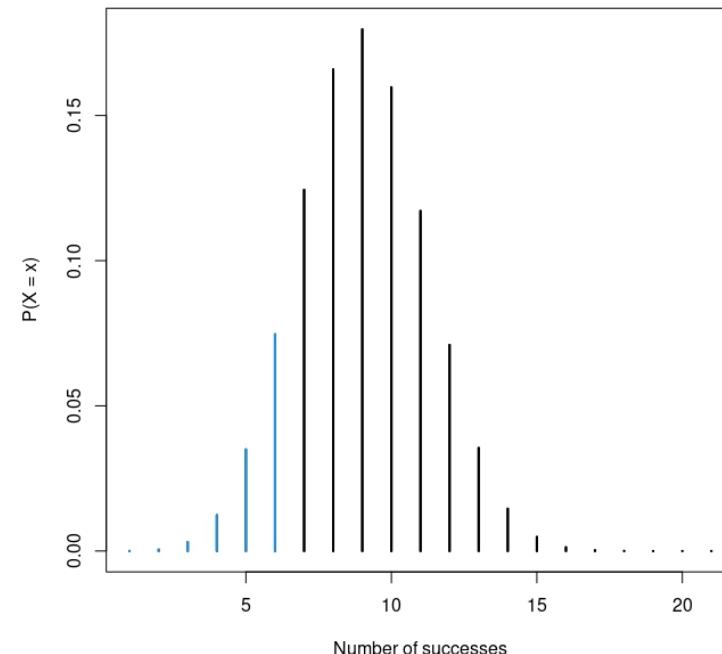
  if(lb == min(x) & ub == max(x)) {
    color <- col
  } else {
    color <- rep(1, length(x))
    color[(lb + 1):ub ] <- col
  }

  lines(dbinom(x, size=size, prob=prob), type="h",
        col= color, lwd=lwd, ...)
}
```

$$P(X \leq 6) = F_X(X=6) = F(6)$$

pbinom(q=6, size=20, prob=0.4)

*binom_sum(size=20, prob=0.4, ub=6, lwd=2,
ylab="P(X=x)", xlab="Number of successes")*



The binomial cumulative function

```
# size: number of trials (n >=0)
# prob: probability of success on each trial
# lb: lower bound of the sum
# ub: upper bound of the sum
# col: color
# lwd: line width
binom_sum <- function(size, prob, lb, ub, col=4, lwd=1, ...) {
  x <- 0:size

  if (missing(lb)) {
    lb <- min(x)
  }
  if (missing(ub)) {
    ub <- max(x)
  }

  plot(dbinom(x, size=size, prob=prob), type="h", lwd=lwd, ...)

  if(lb == min(x) & ub == max(x)) {
    color <- col
  } else {
    color <- rep(1, length(x))
    color[(lb + 1):ub ] <- col
  }

  lines(dbinom(x, size=size, prob=prob), type="h",
        col= color, lwd=lwd, ...)
}
```

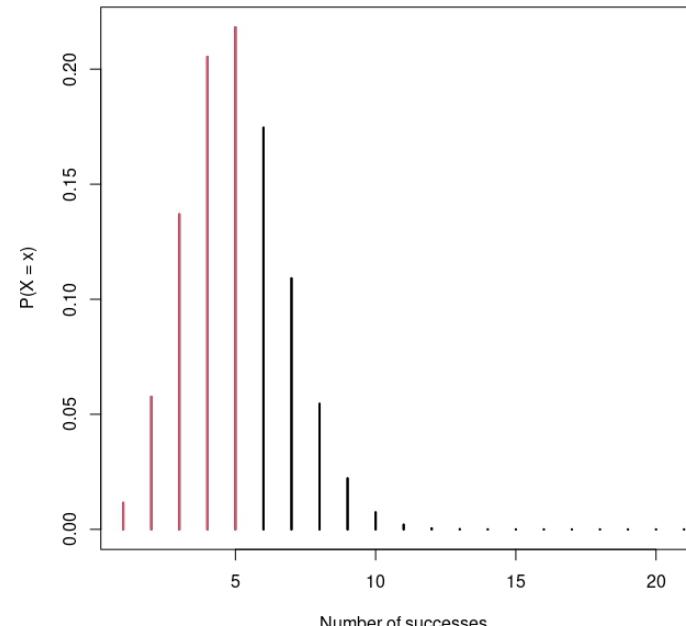
$$P(X < 6) = P(X \leq 5) = F_x(X=5) = F(5)$$

$$pbinom(q=5, size=20, prob=0.4)$$

$$1 - pbinom(q=5, size=20, prob=0.4, lower.tail=FALSE)$$
$$sum(dbinom(0:5, size=20, prob=0.4))$$

$$binom_sum(size=20, prob=0.2, lwd=2, col=2, ub=5,$$

$$ylab="P(X=x)", xlab="Number of successes")$$



The binomial cumulative function

```
# size: number of trials (n >=0)
# prob: probability of success on each trial
# lb: lower bound of the sum
# ub: upper bound of the sum
# col: color
# lwd: line width
binom_sum <- function(size, prob, lb, ub, col=4, lwd=1, ...) {
  x <- 0:size

  if (missing(lb)) {
    lb <- min(x)
  }
  if (missing(ub)) {
    ub <- max(x)
  }

  plot(dbinom(x, size=size, prob=prob), type="h", lwd=lwd, ...)

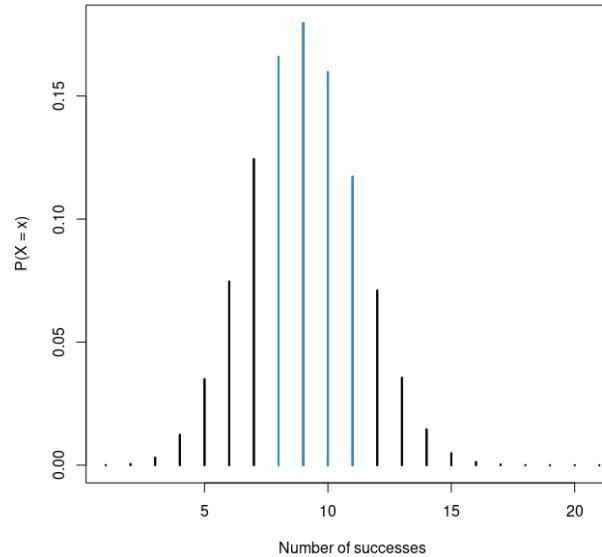
  if(lb == min(x) & ub == max(x)) {
    color <- col
  } else {
    color <- rep(1, length(x))
    color[(lb + 1):ub ] <- col
  }

  lines(dbinom(x, size=size, prob=prob), type="h",
        col= color, lwd=lwd, ...)
}
```

$$P(7 \leq X \leq 11) = ?$$

$pbinom(q=11, size=20, prob=0.4) - pbinom(q=7, size=20, prob=0.4)$
 $sum(dbinom(7:11, size=20, prob=0.4))$

$binom_sum(size=20, prob=0.4, lb=7, ub=11, lwd=2,$
 $ylab="P(X=x)", xlab="Number of successes")$



The binomial cumulative function, exercise 1

```
# size: number of trials (n >=0)
# prob: probability of success on each trial
# lb: lower bound of the sum
# ub: upper bound of the sum
# col: color
# lwd: line width
binom_sum <- function(size, prob, lb, ub, col=4, lwd=1, ...) {
  x <- 0:size

  if (missing(lb)) {
    lb <- min(x)
  }
  if (missing(ub)) {
    ub <- max(x)
  }

  plot(dbinom(x, size=size, prob=prob), type="h", lwd=lwd, ...)

  if(lb == min(x) & ub == max(x)) {
    color <- col
  } else {
    color <- rep(1, length(x))
    color[(lb + 1):ub ] <- col
  }

  lines(dbinom(x, size=size, prob=prob), type="h",
        col= color, lwd=lwd, ...)
}
```

For $X \sim \text{Binomial}(10, 0.5)$. The overall p-value in this example is $2 * F_X(1)$.

```
2*pb $\text{inom}(q=1, size=10, prob=0.5)$ 
# [1] 0.02148438
```

```
binom_sum(size=10, prob=0.5, lwd=2, col=2, lb=?,
ub=?, ylab="P(X=x)", xlab="Number of successes")
```

Exercise 1: rewrite binom_sum function to visualize $P(X \leq 1)$ and $P(X \geq 9)$ in the same plot

Binomial Distribution: generate random

The `rbinom` function allows you to draw `nn` random observations from a binomial distribution.

Example, generate 15 random observations from a binomial distribution if the number of trials is 30 and the probability of success on each trial is 0.1

```
set.seed(2)
rbinom(n=15, size=30, prob=0.1)
[1] 2 4 3 1 6 6 1 5 3 3 3 2 4 1 2

library(ggplot2)
library(reshape2)

# Data
set.seed(5)
x <- cbind(N="", rnorm(10), n100=rnorm(100), n10=rnorm(1000))
df <- data.frame(x)

# Histogram with kernel density
ggplot(df, aes(x = x)) +
  geom_histogram(aes(y = ..density..),
                 colour = 1, fill = "white") +
  geom_density(lwd = 1, colour = 4,
               fill = 4, alpha = 0.25)
```

Examples of Conditional Expectation and Variance

1. Swimming with dolphins

Fraser runs a dolphin-watch business.

Every day, he is unable to run the trip due to bad weather with probability p , independently of all other days. Fraser works every day except the bad-weather days, which he takes as holiday.



Let Y be the number of consecutive days Fraser has to work between bad-weather days. Let X be the total number of customers who go on Fraser's trip in this period of Y days. Conditional on Y , the distribution of X is

$$(X | Y) \sim \text{Poisson}(\mu Y).$$

- Name the distribution of Y , and state $\mathbb{E}(Y)$ and $\text{Var}(Y)$.
- Find the expectation and the variance of the number of customers Fraser sees between bad-weather days, $\mathbb{E}(X)$ and $\text{Var}(X)$.

Examples of Conditional Expectation and Variance

(a) Let ‘success’ be ‘bad-weather day’ and ‘failure’ be ‘work-day’.

Then $\mathbb{P}(\text{success}) = \mathbb{P}(\text{bad-weather}) = p$.

Y is the number of failures before the first success.

So

$$Y \sim \text{Geometric}(p).$$

Thus

$$\mathbb{E}(Y) = \frac{1-p}{p},$$

$$\text{Var}(Y) = \frac{1-p}{p^2}.$$

(b) We know $(X | Y) \sim \text{Poisson}(\mu Y)$: so

$$\mathbb{E}(X | Y) = \text{Var}(X | Y) = \mu Y.$$

Examples of Conditional Expectation and Variance

By the Law of Total Expectation:

$$\begin{aligned}\mathbb{E}(X) &= \mathbb{E}_Y\{\mathbb{E}(X | Y)\} \\ &= \mathbb{E}_Y(\mu Y) \\ &= \mu \mathbb{E}_Y(Y)\end{aligned}$$

$$\therefore \mathbb{E}(X) = \frac{\mu(1-p)}{p}.$$

By the Law of Total Variance:

$$\begin{aligned}\text{Var}(X) &= \mathbb{E}_Y\left(\text{Var}(X | Y)\right) + \text{Var}_Y\left(\mathbb{E}(X | Y)\right) \\ &= \mathbb{E}_Y(\mu Y) + \text{Var}_Y(\mu Y) \\ &= \mu \mathbb{E}_Y(Y) + \mu^2 \text{Var}_Y(Y) \\ &= \mu \left(\frac{1-p}{p}\right) + \mu^2 \left(\frac{1-p}{p^2}\right) \\ &= \frac{\mu(1-p)(p+\mu)}{p^2}.\end{aligned}$$

Examples of Conditional Expectation and Variance

Checking your answer in R:

If you know how to use a statistical package like *R*, you can check your answer to the question above as follows.

```
> # Pick a value for p, e.g. p = 0.2.  
> # Pick a value for mu, e.g. mu = 25  
>  
> # Generate 10,000 random values of Y ~ Geometric(p = 0.2):  
> y <- rgeom(10000, prob=0.2)  
>  
> # Generate 10,000 random values of X conditional on Y:  
> # use (X | Y) ~ Poisson(mu * Y) ~ Poisson(25 * Y)  
> x <- rpois(10000, lambda = 25*y)  
> # Find the sample mean of X (should be close to E(X)):  
> mean(x)  
[1] 100.6606  
>  
> # Find the sample variance of X (should be close to var(X)):  
> var(x)  
[1] 12624.47  
>  
> # Check the formula for E(X):  
> 25 * (1 - 0.2) / 0.2  
[1] 100  
>  
> # Check the formula for var(X):  
> 25 * (1 - 0.2) * (0.2 + 25) / 0.2^2  
[1] 12600
```

The formulas we obtained by working give $\mathbb{E}(X) = 100$ and $\text{Var}(X) = 12600$. The sample mean was $\bar{x} = 100.6606$ (close to 100), and the sample variance was 12624.47 (close to 12600). Thus our working seems to have been correct.

Continuous Random Variables

1. Uniform distribution

Notation: $X \sim \text{Uniform}(a, b)$.

Probability density function (pdf): $f_X(x) = \frac{1}{b-a}$ for $a < x < b$.

Cumulative distribution function:

$$F_X(x) = \mathbb{P}(X \leq x) = \frac{x-a}{b-a} \quad \text{for } a < x < b.$$

$F_X(x) = 0$ for $x \leq a$, and $F_X(x) = 1$ for $x \geq b$.

Mean: $\mathbb{E}(X) = \frac{a+b}{2}$.

Variance: $\text{Var}(X) = \frac{(b-a)^2}{12}$.

2. Exponential distribution

Notation: $X \sim \text{Exponential}(\lambda)$.

Probability density function (pdf): $f_X(x) = \lambda e^{-\lambda x}$ for $0 < x < \infty$.

Cumulative distribution function:

$$\begin{aligned} F_X(x) &= \mathbb{P}(X \leq x) = 1 - e^{-\lambda x} \quad \text{for } 0 < x < \infty. \\ F_X(x) &= 0 \quad \text{for } x \leq 0. \end{aligned}$$

Mean: $\mathbb{E}(X) = \frac{1}{\lambda}$.

Variance: $\text{Var}(X) = \frac{1}{\lambda^2}$.

Sum: if X_1, \dots, X_k are independent, and each $X_i \sim \text{Exponential}(\lambda)$, then

$$X_1 + \dots + X_k \sim \text{Gamma}(k, \lambda).$$

3. Gamma distribution

Notation: $X \sim \text{Gamma}(k, \lambda)$.

Probability density function (pdf):

$$f_X(x) = \frac{\lambda^k}{\Gamma(k)} x^{k-1} e^{-\lambda x} \quad \text{for } 0 < x < \infty,$$

where $\Gamma(k) = \int_0^\infty y^{k-1} e^{-y} dy$ (the Gamma function).

Cumulative distribution function: no closed form.

Mean: $\mathbb{E}(X) = \frac{k}{\lambda}$.

Variance: $\text{Var}(X) = \frac{k}{\lambda^2}$.

Sum: if X_1, \dots, X_n are independent, and $X_i \sim \text{Gamma}(k_i, \lambda)$, then

$$X_1 + \dots + X_n \sim \text{Gamma}(k_1 + \dots + k_n, \lambda).$$

4. Normal distribution

Notation: $X \sim \text{Normal}(\mu, \sigma^2)$.

Probability density function (pdf):

$$f_X(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad \text{for } -\infty < x < \infty.$$

Cumulative distribution function: no closed form.

Mean: $\mathbb{E}(X) = \mu$.

Variance: $\text{Var}(X) = \sigma^2$.

Sum: if X_1, \dots, X_n are independent, and $X_i \sim \text{Normal}(\mu_i, \sigma_i^2)$, then

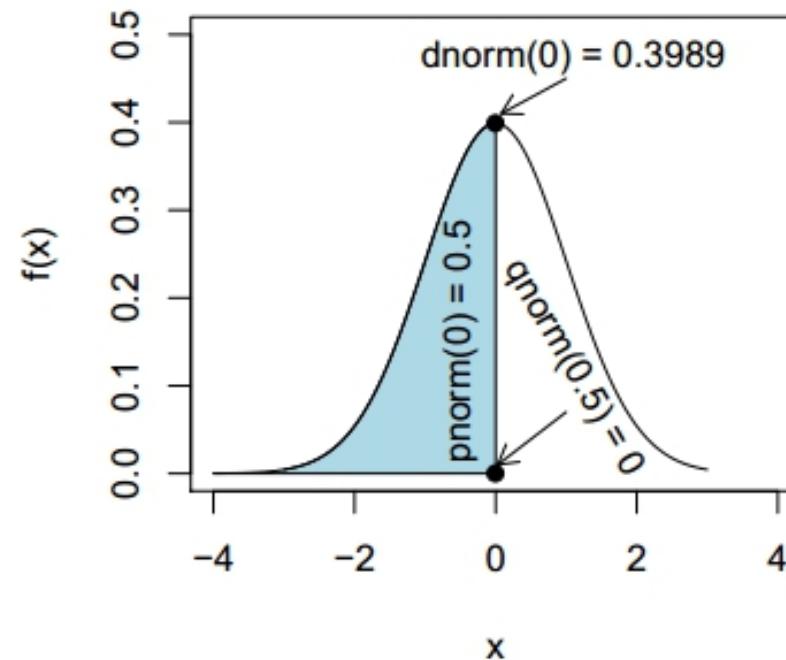
$$X_1 + \dots + X_n \sim \text{Normal}(\mu_1 + \dots + \mu_n, \sigma_1^2 + \dots + \sigma_n^2).$$

Normal Distribution $X \sim N(\text{mean}, \text{variance})$

- The **probability density function** (PDF), also known as Bell curve, of x is $f(x) = \frac{1}{\sqrt{2\pi}\sigma^2} e^{-\frac{1}{2}(\frac{x-\mu}{\sigma})^2}$.
- The **cumulative distribution function** (CDF) is $F(x) = P(X \leq x)$.
- The **quantile function** is $Q(p) = F^{-1}(p)$.
- The **expected mean and variance** are $E(X) = \mu$ and $\text{Var}(X) = \sigma^2$, respectively.

In R there exist the `dnorm`, `pnorm` and `qnorm` functions, which allows calculating the normal density, distribution and quantile function for a set of values. In addition, the `rnorm` function allows obtaining random observations that follow a normal distribution. The following table summarizes the functions related to the normal distribution:

Function	Description
<code>dnorm</code>	Normal density (Probability Density Function)
<code>pnorm</code>	Normal distribution (Cumulative Distribution Function)
<code>qnorm</code>	Quantile function of the Normal distribution
<code>rnorm</code>	Normal random number generation



Normal Distribution: probability function

```
par(mfrow=c(1, 2))

# Grid of X-axis values
x <- seq(-4, 8, 0.1)

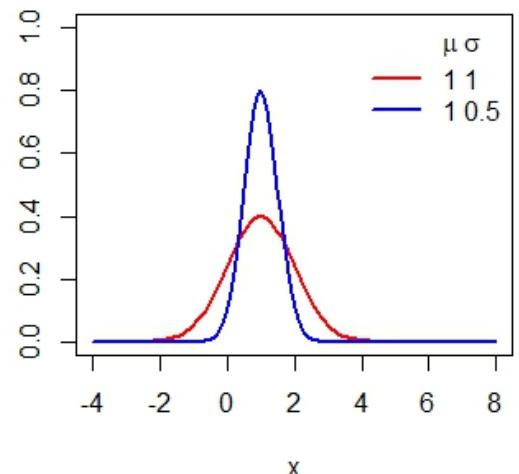
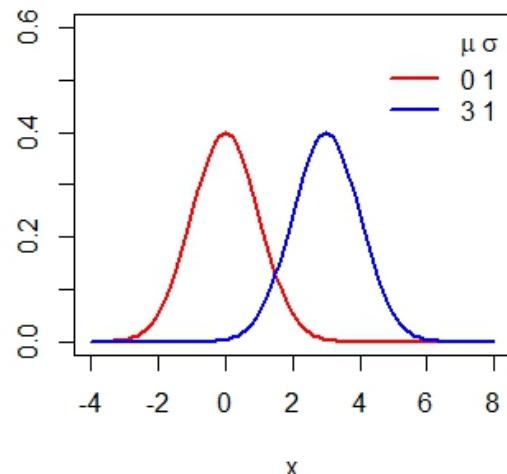
#-----
# Same standard deviation, different mean
#-----
# Mean 0, sd 1
plot(x, dnorm(x, mean=0, sd=1), type="l",
      ylim=c(0, 0.6), ylab="", lwd=2, col="red")
# Mean 3, sd 1
lines(x, dnorm(x, mean=3, sd=1), col="blue", lty=1, lwd=2)

# Adding a legend
legend("topright", legend=c("0 1", "3 1"), col=c("red", "blue"),
       title=expression(paste(mu, " ", sigma)),
       title.adj=0.9, lty=1, lwd=2, box.lty=0)

#-----
# Same mean, different standard deviation
#-----
# Mean 1, sd 1
plot(x, dnorm(x, mean=1, sd=1), type="l",
      ylim=c(0, 1), ylab="", lwd=2, col="red")
# Mean 1, sd 0.5
lines(x, dnorm(x, mean=1, sd=0.5), col="blue", lty=1, lwd=2)

# Adding a legend
legend("topright", legend=c("1 1", "1 0.5"), col=c("red", "blue"),
       title=expression(paste(mu, " ", sigma)),
       title.adj=0.75, lty=1, lwd=2, box.lty=0)

par(mfrow=c(1, 1))
```



Normal Distribution

```
# Normal probability distribution plot
Mean <- 1000
Sd <- 10

# X grid for non-standard normal distribution
x <- seq(-3, 3, length=100) * Sd + Mean

# Density function
f <- dnorm(x, Mean, Sd)

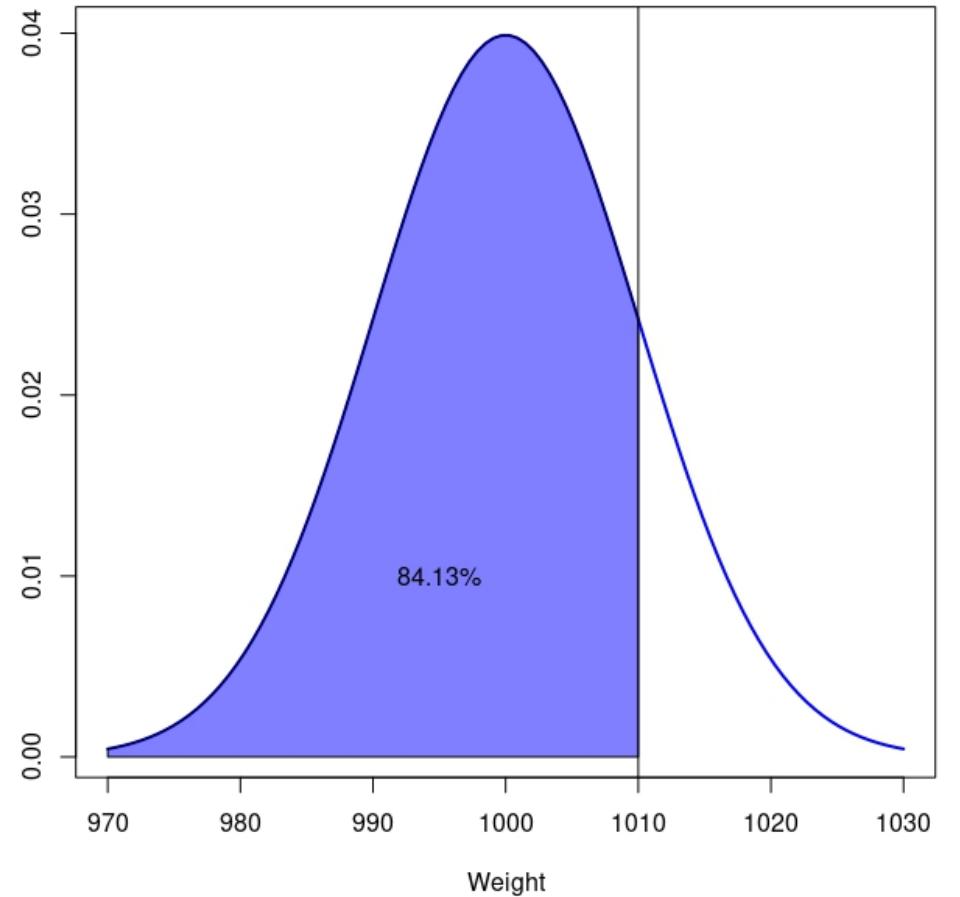
plot(x, f, type="l", lwd=2, col="blue", ylab="", xlab="Weight")
abline(v=Mean) # Vertical line on the mean

lb <- min(x) # Lower bound
ub <- 1010 # Upper bound

x2 <- seq(min(x), ub, length=100) # New Grid
y <- dnorm(x2, Mean, Sd) # Density

plot(x, f, type="l", lwd=2, col="blue", ylab="", xlab="Weight")
abline(v=ub)

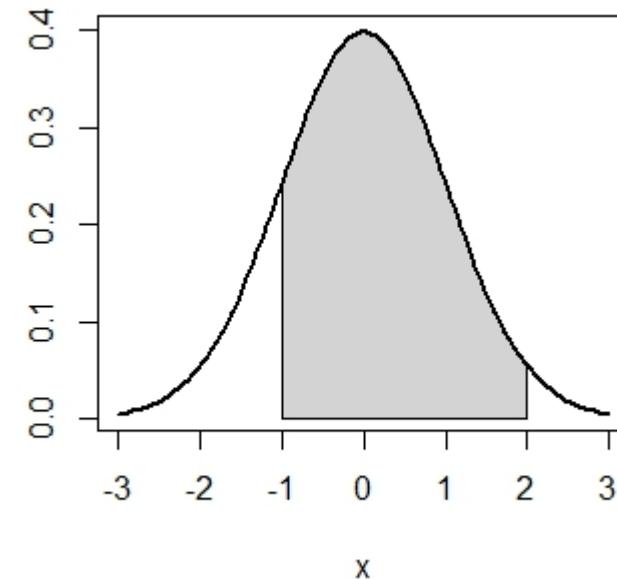
polygon(c(lb, x2, ub), c(0, y, 0), col=rgb(0, 0, 1, alpha=0.5))
text(995, 0.01, "84.13%")
```



Normal Distribution

```
# mean: mean of the Normal variable  
# sd: standard deviation of the Normal variable  
# lb: lower bound of the area  
# ub: upper bound of the area  
# acolor: color of the area  
# ....: additional arguments to be passed to lines function  
  
normal_area <- function(mean=0, sd=1, lb, ub, acolor="lightgray", ...) {  
  x <- seq(mean - 3 * sd, mean + 3 * sd, length=100)  
  
  if (missing(lb)) {  
    lb <- min(x)  
  }  
  if (missing(ub)) {  
    ub <- max(x)  
  }  
  
  x2 <- seq(lb, ub, length=100)  
  plot(x, dnorm(x, mean, sd), type="n", ylab="")  
  
  y <- dnorm(x2, mean, sd)  
  polygon(c(lb, x2, ub), c(0, y, 0), col=acolor)  
  lines(x, dnorm(x, mean, sd), type="l", ...)  
}
```

```
normal_area(mean=0, sd=1, lb=-1, ub=2, lwd=2)
```



Normal Distribution

* Probability of a box weighing more than 980 grams
 $P(X > 980) = P(X \geq 980)$

```
pnorm(980, Mean, Sd, lower.tail=FALSE) # 0.9772499 or 97.72%
```

```
1 - pnorm(980, Mean, Sd) # Equivalent
```

```
pnorm(1020, Mean, Sd) # Equivalent by symmetry
```

```
normal_area(mean=Mean, sd=Sd, lb=980, acolor=rgb(0, 0, 1, alpha=0.5))  
text(1000, 0.01, "97.72%")
```



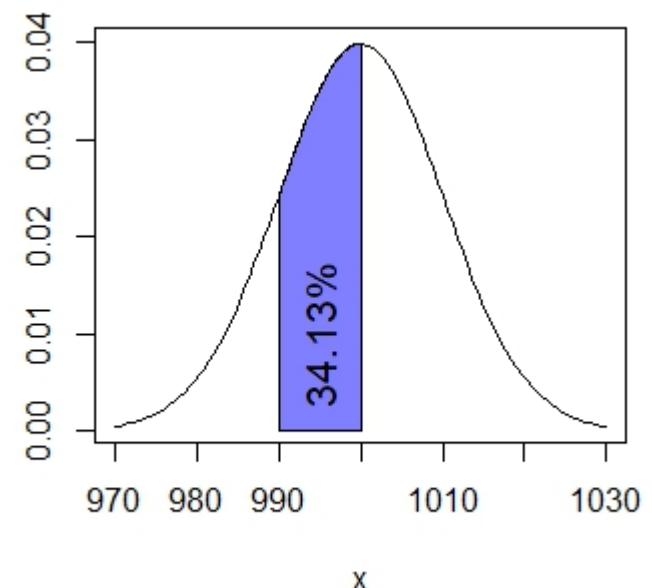
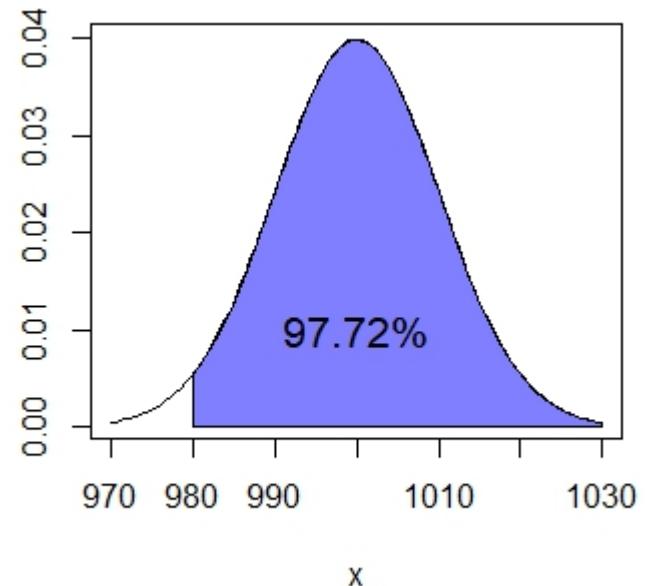
For continuous variables, as $P(X = x) = 0$, $P(X \geq x) = P(X > x)$
and $P(X \leq x) = P(X < x)$.

* Probability of a box weighing more than 990 grams and less than 1000 you have
to calculate ?

$P(990 < X < 1000)$

```
pnorm(1000, Mean, Sd) - pnorm(990, Mean, Sd) # 0.3413447 or 34.13%
```

```
normal_area(mean=Mean, sd=Sd, lb=990, ub=1000, acolor=rgb(0, 0, 1, alpha=0.5))  
text(995, 0.01, "34.13%", srt=90)
```



Normal Distribution: Plot cumulative distribution

```
par(mfrow=c(1, 2))
# Grid of X-axis values
x <- seq(-4, 8, 0.1)

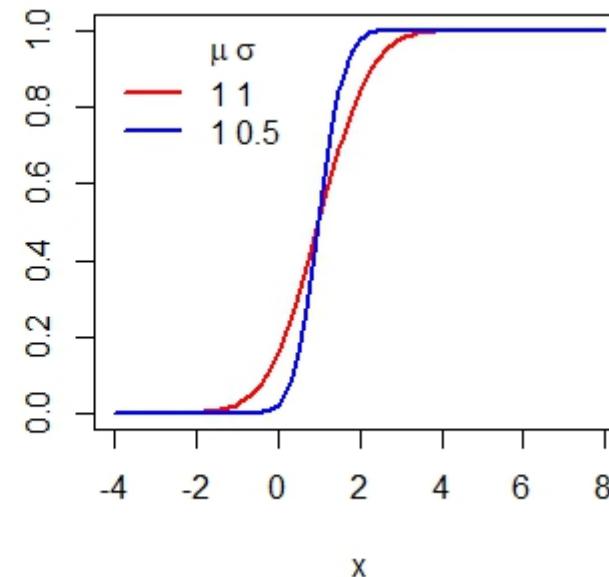
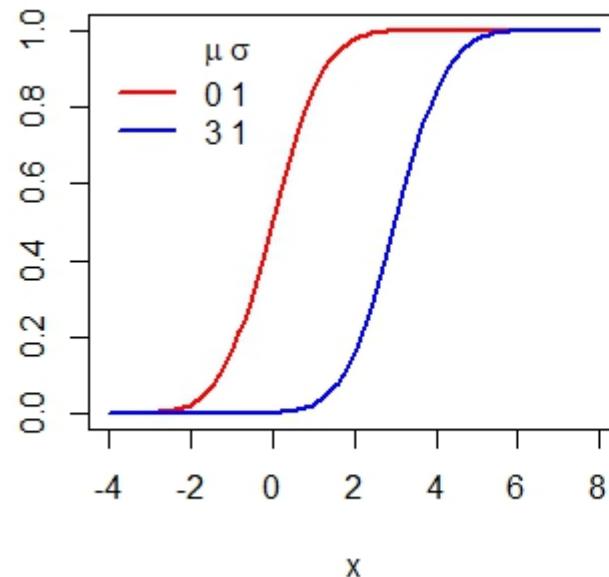
#-----
# Same standard deviation, different mean
#-----
# Mean 0, sd 1
plot(x, pnorm(x, mean=0, sd=1), type="l", ylim=c(0, 1), ylab="", lwd=2, col="red")
# Mean 3, sd 1
lines(x, pnorm(x, mean=3, sd=1), col="blue", lty=1, lwd=2)

# Legend
legend("topleft", legend=c("0 1", "3 1"), col=c("red", "blue"),
       title=expression(paste(mu, " ", sigma)),
       title.adj=0.9, lty=1, lwd=2, box.lty=0)

#-----
# Same mean, different standard deviation
#-----
# Mean 1, sd 1
plot(x, pnorm(x, mean=1, sd=1), type="l",
      ylim=c(0, 1), ylab="", lwd=2, col="red")
# Mean 1, sd 0.5
lines(x, pnorm(x, mean=1, sd=0.5), col="blue", lty=1, lwd=2)

# Legend
legend("topleft", legend=c("1 1", "1 0.5"), col=c("red", "blue"),
       title=expression(paste(mu, " ", sigma)),
       title.adj=0.75, lty=1, lwd=2, box.lty=0)

par(mfrow=c(1, 1))
```

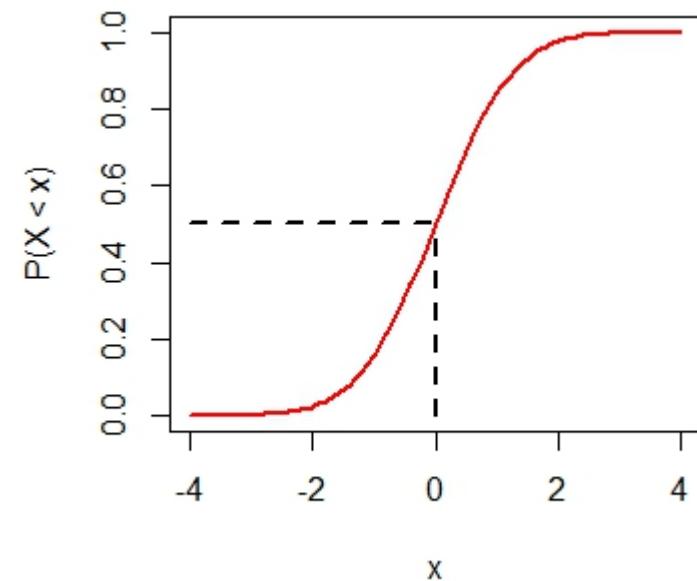


Normal Distribution: Plot cumulative distribution

$P(X < 0) = 0.5$ for a standard Normal distribution

```
x <- seq(-4, 4, 0.1)

plot(x, pnorm(x, mean=0, sd=1), type="l", ylim=c(0, 1), ylab="P(X < x)", lwd=2, col="red")
segments(0, 0, 0, 0.5, lwd=2, lty=2)
segments(-4, 0.5, 0, 0.5, lwd=2, lty=2)
```



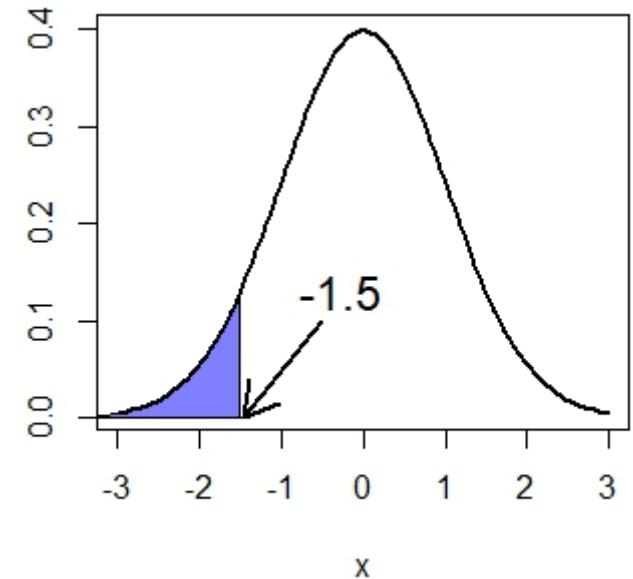
Normal Distribution: quantile for probability

- The qnorm function allows you to find the quantile (percentile) Q for any probability p .
- The qnorm function is the inverse of the pnorm function
- The quantile for probability 0.5 ($Q(0.5)$) on a symmetric distribution is equal to the mean

```
qnorm(0.5, mean=0, sd=1) # 0
```

```
x <- pnorm(q=-1.5, mean=0, sd=1) # 0.0668072  
qnorm(x, mean=0, sd=1) # -1.5
```

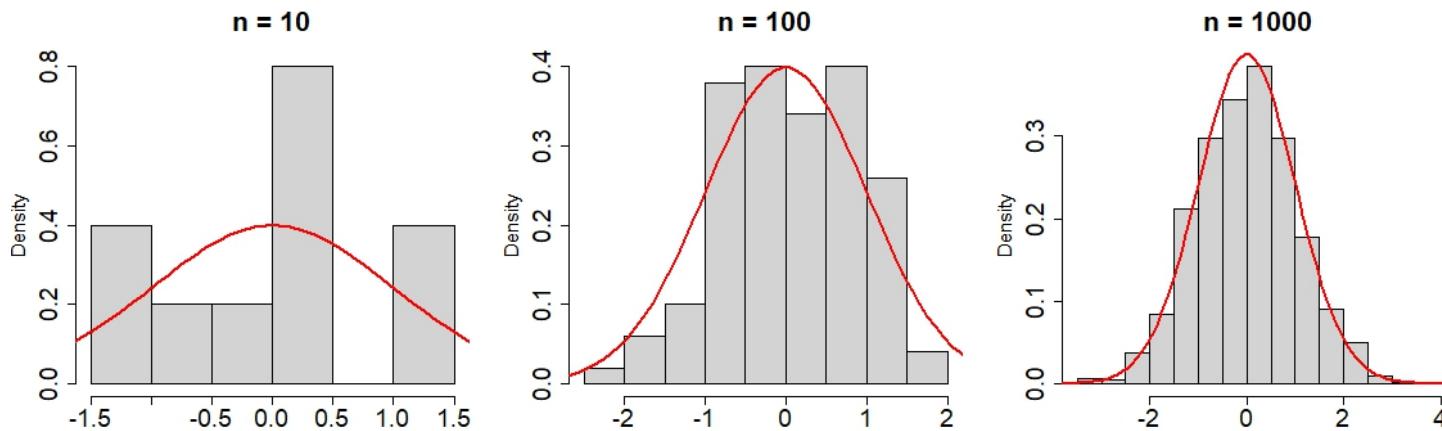
```
normal_area(mean=0, sd=1, ub=-1.5, lwd=2, acolor=rgb(0, 0,  
1, alpha=0.5))  
arrows(-0.5, 0.1, -1.45, 0, lwd=2, length=0.2)  
text(-0.25, 0.13, "-1.5", cex=1.5)
```



Normal Distribution: generate random

- The **rnorm** function generates n observations from the Normal distribution with mean μ and standard deviation σ
- Generate 10 observations of a standard Normal distribution

```
set.seed(1)
rnorm(10)
[1] -0.19250276 1.25912983 -0.53794513 -0.76502724
0.23342112 -1.47363447
[7] 0.04490047 0.99764774 -1.95028133 -0.68274247
```



```
par(mfrow=c(1, 3))
x <- seq(-10, 10, length=200)
set.seed(3)

# n=10
hist(rnorm(10, mean=0, sd=1), main="n=10",
      xlab="", prob=TRUE)
lines(x, dnorm(x), col="red", lwd=2)

# n=100
hist(rnorm(100, mean=0, sd=1), main="n=100",
      xlab="", prob=TRUE)
lines(x, dnorm(x), col="red", lwd=2)

# n=1000
hist(rnorm(1000, mean=0, sd=1), main="n=1000",
      xlab="", prob=TRUE)
lines(x, dnorm(x), col="red", lwd=2)

par(mfrow=c(1, 1))
```

Normal Distribution: One Sample Z-Test in R

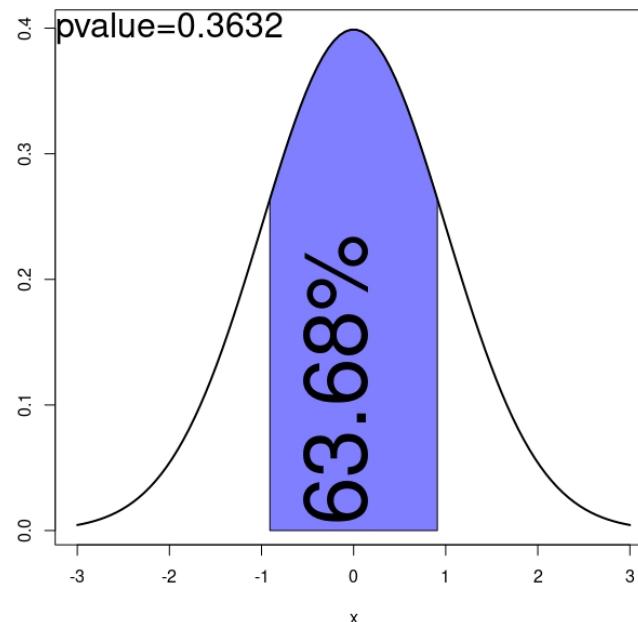
- Suppose the IQ in a certain population is normally distributed with a mean of $\mu = 100$ and standard deviation of $\sigma = 15$.
- A scientist wants to know if a new medication affects IQ levels, so she recruits 20 patients to use it for one month and records their IQ levels at the end of the month.

Z-TEST

Formula to find the value of Z (z-test) is:

$$Z = \frac{\bar{x} - \mu_0}{\sigma / \sqrt{n}}$$

- \bar{x} = mean of sample
- μ_0 = mean of population
- σ = standard deviation of population
- n = no. of observations



```
# install.packages("BSDA")
# Z test for mean of one sample
library(BSDA)

mu <- 100
sigma <- 15
# enter IQ levels for 20 patients
data <- c(88, 92, 94, 94, 96, 97, 97, 97, 99, 99,
        105, 109, 109, 109, 110, 112, 112, 113, 114, 115)

# perform one sample z-test using z.test in BSDA package
z.test(data, mu=mu, sigma.x=sigma)
# One-sample z-Test

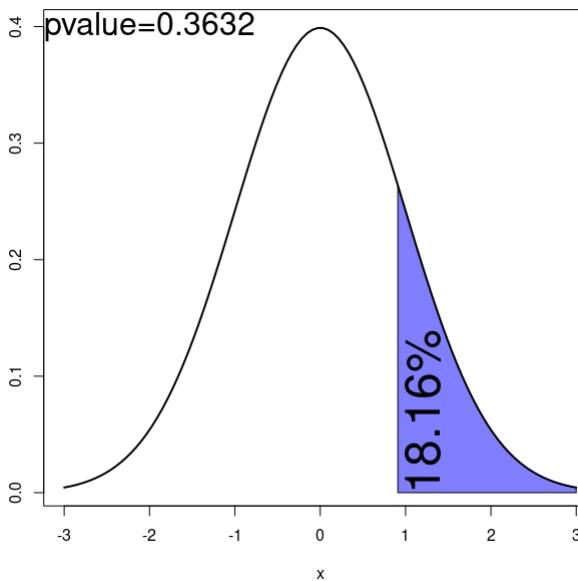
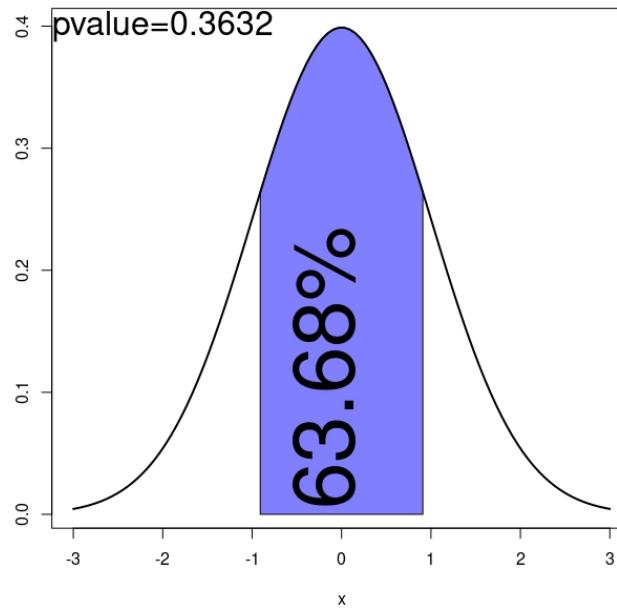
# data: data
# z = 0.90933, p-value = 0.3632
# alternative hypothesis: true mean is not equal to 100
# 95 percent confidence interval:
#  96.47608 109.62392
# sample estimates:
# mean of x
# 103.05

# perform one sample z-test manually
Zst <- (mean(data)-mu)/(sigma/sqrt(length(data)))
dtp <- 1 - 2*(1-pnorm(Zst, mean=0, sd=1))
pvalue <- 2*(1-pnorm(Zst, mean=0, sd=1))
print(paste0("Z = ", Zst))
print(paste0("p value = ", pvalue))
# [1] "Z = 0.909334310849914"
# [1] "p value = 0.363173684651059"

# plot
normal_area(mean=0, sd=1, lb=-Zst, ub=Zst, lwd=2,
            acolor=rgb(0, 0, 1, alpha=0.5))
text(-0.2, 0.12, paste0(100*round(dtp,4), "%"), srt = 90, cex=5)
text(-2, 0.4, paste0("pvalue=", round(pvalue,4)), cex=2)
```

Normal Distribution: One Sample Z-Test in R

- Suppose the IQ in a certain population is normally distributed with a mean of $\mu = 100$ and standard deviation of $\sigma = 15$.
- A scientist wants to know if a new medication affects IQ levels, so she recruits 20 patients to use it for one month and records their IQ levels at the end of the month.



```
# install.packages("BSDA")
# Z test for mean of one sample
library(BSDA)

mu <- 100
sigma <- 15
# enter IQ levels for 20 patients
data <- c(88, 92, 94, 94, 96, 97, 97, 97, 99, 99,
        105, 109, 109, 109, 110, 112, 112, 113, 114, 115)

# perform one sample z-test using z.test in BSDA package
z.test(data, mu=mu, sigma.x=sigma)

# perform one sample z-test manually
Zst <- (mean(data)-mu)/(sigma/sqrt(length(data)))
dtp <- 1 - 2*(1-pnorm(Zst, mean=0, sd=1))
pvalue <- 2*(1-pnorm(Zst, mean=0, sd=1))
print(paste0("Z = ", Zst))
print(paste0("p value = ", pvalue))
# [1] "Z = 0.909334310849914"
# [1] "p value = 0.363173684651059"

# plot
normal_area(mean=0, sd=1, lb=-Zst, ub=Zst, lwd=2,
            acolor=rgb(0, 0, 1, alpha=0.5))
text(-0.2, 0.12, paste0(100*round(dtp,4), "%"), srt = 90, cex=5)
text(-2, 0.4, paste0("pvalue=", round(pvalue,4)), cex=2)

# plot
normal_area(mean=0, sd=1, lb=Zst, lwd=2, acolor=rgb(0, 0, 1,
            alpha=0.5))
text(1.2, 0.07, paste0(100*round(pvalue/2,4), "%"), srt = 90,
     cex=3)
text(-2, 0.4, paste0("pvalue=", round(pvalue,4)), cex=2)
```

Normal Distribution: Two Sample Z-Test in R, exercise 2

- Suppose the IQ levels among individuals in two different cities are known to be normally distributed each with population standard deviations of 15.
- A scientist wants to know if the mean IQ level between individuals in city A and city B are different, so she selects a simple random sample of 20 individuals from each city and records their IQ levels.

$$z = \frac{(\bar{x}_1 - \bar{x}_2) - (\mu_1 - \mu_2)}{\sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}}}$$

```
library(BSDA)

#enter IQ levels for 20 individuals from each city
cityA = c(82, 84, 85, 89, 91, 91, 92, 94, 99, 99,
         105, 109, 109, 109, 110, 112, 112, 113, 114, 114)

cityB = c(90, 91, 91, 91, 95, 95, 99, 99, 108, 109,
         109, 114, 115, 116, 117, 117, 128, 129, 130, 133)

#perform two sample z-test
z.test(x=cityA, y=cityB, mu=0, sigma.x=15, sigma.y=15)

Two-sample z-Test

data: cityA and cityB
z = -1.7182, p-value = 0.08577
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
-17.446925 1.146925
sample estimates:
mean of x mean of y
100.65 108.80

Exercise 2: manually calculate z test for 2 samples using R
```

<https://www.statology.org/z-test-in-r/>

Some Distribution Functions in R

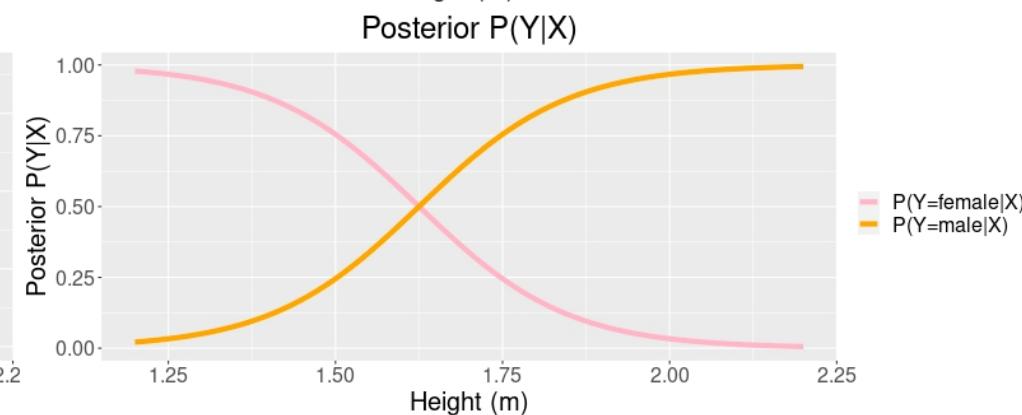
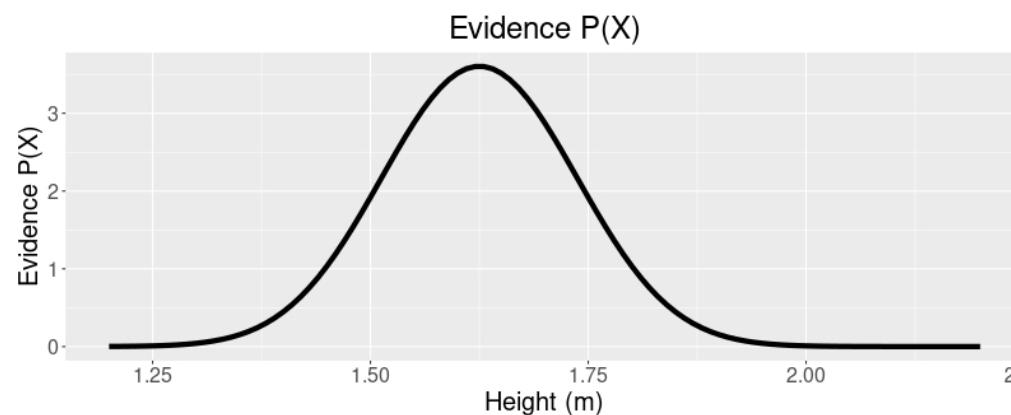
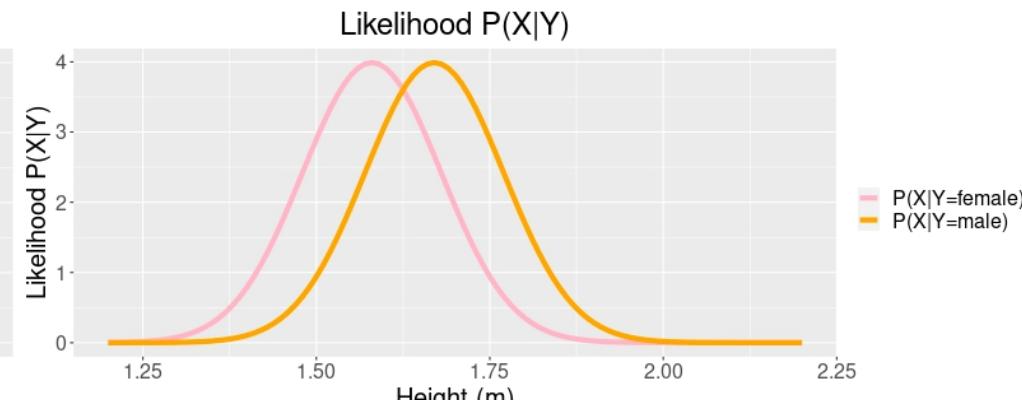
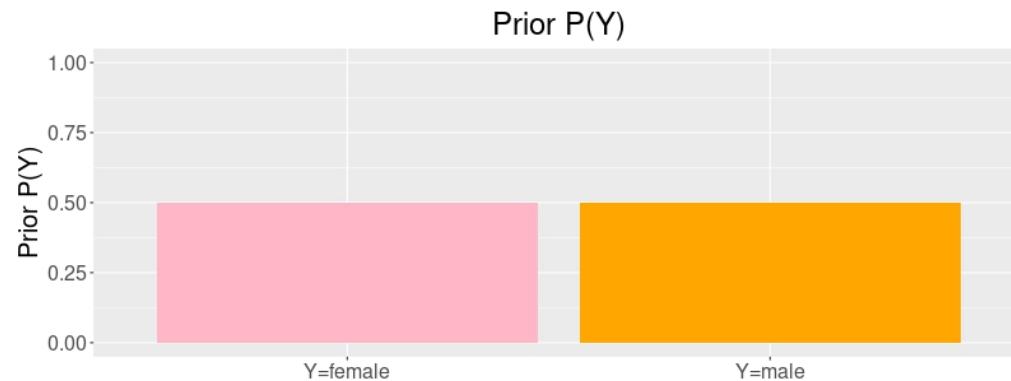
Distribution	Functions			
Beta	pbeta	qbeta	dbeta	rbeta
Binomial	pbinom	qbinom	dbinom	rbinom
Cauchy	pcauchy	qcauchy	dcauchy	rcauchy
Chi-Square	pchisq	qchisq	dchisq	rchisq
Exponential	pexp	qexp	dexp	rexp
F	pf	qf	df	rf
Gamma	pgamma	qgamma	dgamma	rgamma
Geometric	pgeom	qgeom	dgeom	rgeom
Hypergeometric	phyper	qhyper	dhyper	rhyper
Logistic	plogis	qlogis	dlogis	rlogis
Log_Normal	plnorm	qlnorm	dlnorm	rlnorm
Negative Binomial	pnbino	qnbinom	dnbinom	rnbinom
Normal	pnorm	qnorm	dnorm	rnorm
Poisson	ppois	qpois	dpois	rpois
Student t	pt	qt	dt	rt
Studentized Range	ptukey	qtukey	dtukey	rtukey
Uniform	punif	qunif	dunif	runif
Weibull	pweibull	qweibull	dweibull	rweibull
Wilcoxon Rank Sum Statistic	pwilcox	qwilcox	dwilcox	rwilcox
Wilcoxon Signed Rank Statistic	psignrank	qsignrank	dsignrank	rsignrank

Bayesian Classifier

Bayesian Decision Theory: Gender Classification using height

$$P(Y = \text{female}|X) = \frac{P(X|Y = \text{female})P(Y = \text{female})}{P(X)} = \frac{P(X|Y = \text{female})P(Y = \text{female})}{P(X|Y = \text{female})P(Y = \text{female}) + P(X|Y = \text{male})P(Y = \text{male})} = \frac{\text{Likelihood} * \text{Prior}}{\text{Evidence}} = \text{Posterior}$$

Set Prior distribution of female $P(Y=\text{female})$



Two steps to create a ShinyApp

1) Write R script to do the task and plot with basic, ggplot2 or plotly

2) Create a ShinyApp by modifying the template of App.R

2.1) The interface part

```
4 library(shiny) # Required to run any Shiny app
5 library(ggplot2) # For creating pretty plots
6 library(gridExtra)
7 library(reshape2)
8
9 # example for classification of male vs female using height
10 # ui.R ----
11 ui <- fluidPage(
12   titlePanel(
13     h1("Bayesian Decision Theory: Gender Classification using height", align = "center")
14   ),
15   # https://shiny.rstudio.com/gallery/mathjax.html
16   withMathJax(
17     helpText('$$P(Y=female|X)=\frac{P(X|Y=female)P(Y=female)}{\{P(X|Y=female)P(Y=female)+P(X|Y=males)P(Y=males)\}}=\frac{Likelihood*Prior}{Evidence}=Posterior$$')
18   ),
19   mainPanel(
20     sliderInput(inputId="priorf", label="Set Prior distribution of female P(Y=female)", value=0.50, min=0.00, max=1.00),
21     plotOutput(outputId="barplot", width="100%")
22   )
23 )
```

2.2) The Server part

```
25 # server.R ----
26 server <- function(input, output) {
27   output$barplot <- renderPlot({
28     # Initialization
29     height <- seq(1.2, 2.2, by = .01)
30     ### Prior Distribution or Class probability P(Y)
31     # P(Y=female)
32     Yf <- input$priorf
33     # P(Y=males)
34     Ym <- 1 - Yf
35     # Put them to the dataframe to be plotted
36     prior <- data.frame(class=c("Y=males", "Y=female"), PY=c(Ym, Yf), check.names=F)
37
38     ### Class conditional or Likelihood P(X|Y)
39     # P(X|Y=female)
40     XYf <- dnorm(height, mean=1.58, sd=0.1)
41     # P(X|Y=males)
42     XYm <- dnorm(height, mean=1.67, sd=0.1)
43     # Put them to the dataframe to be plotted
44     df <- data.frame(Height=height, 'P(X|Y=female)'=XYf, 'P(X|Y=males)'=XYm,
45       check.names=F)
45     likelihood <- melt(df, id.vars="Height")
```

Exercise: Modify Bayesian Decision theory App

Ex3: Adding the slider to interactive change the means of $P(X | Y=\text{female})$ and $P(X | Y=\text{male})$. Explain the observation for Posteriors

Ex4: Adding the slider to interactive change the variances of $P(X | Y=\text{female})$ and $P(X | Y=\text{male})$. Explain the observation for Posteriors

Exercise: create a ShinyApp for COVID-19 severity using Bayesian Classifier

Ex5: Predict severity and mild of COVID-19 using Methylation Risk Score of COGRAM

1) Write R script to do the task and plot all of Probability distribution below

X = Methylation Risk Score of COGRAM

Y = Severity or Mild

$P(X)$ = estimate from data

$P(X | Y=\text{Severity})$ = estimate from data

$P(X | Y=\text{Mild})$ = estimate from data

$P(Y=\text{Severity}) = 0.03$

$P(Y=\text{Severity} | X) = ?$

$P(Y=\text{Mild} | X) = ?$

2) Create a ShinyApp by modifying the template of App.R

2.1) The interface part

2.2) The Server part

Ex6: Predict ICUD and non-ICUD of COVID-19 using Methylation Risk Score of ICUD

Same as Ex1 but using the ICUD data

Exercise: create a ShinyApp for optimization of rate and epoch in Gradient Descent

Ex7: Minimize the function $f(x) = x^2 + 4x - 1$

7.1) Add slider for rate = 0.01, 0.1, 0.5, 1, 2, 10, 100 and keep n = 10. Plot the trajectory on the f(x) for the rate and explain the observation.

7.2) Add slider for rate = 0.01, 0.1, 0.5, 1, 2, 10, 100 and n = 10, 20, 50, 100, 500, 1000. Plot the trajectory on f(x) for rate and n. Then explain the important observations.

Ex8: Minimize the function $f(x) = x^2 + 4x + y^2 - 1$

7.1) Add slider for rate = 0.01, 0.1, 0.5, 1, 2, 10, 100 and keep n = 10. Plot the trajectory for the rate on the contour of f(x) and explain the observation.

7.2) Add slider for rate = 0.01, 0.1, 0.5, 1, 2, 10, 100 and n = 10, 20, 50, 100, 500, 1000. Plot the trajectory on the contour of f(x) for rate and n. Then explain the important observations.