

# Distribution and random numbers



**Feng Li**

**feng.li@cufe.edu.cn**

**School of Statistics and Mathematics  
Central University of Finance and Economics**

# Today we are going to learn...

- 1 Basic concepts of random numbers
- 2 Continuous random variables
- 3 Likelihood Function
- 4 Walking APP example
- 5 Discrete random variables

# Preliminary

- **Pseudo random numbers**

- an algorithm for generating a sequence of numbers that approximates the properties of random numbers.
- The sequence is not truly random in that it is completely determined by a relatively small set of initial values, called the PRNG's state.
- Pseudo random numbers are important in practice for their **speed** in number generation and their **reproducibility**.

- **Random seed**

A random seed (or seed state, or just seed) is a number (or vector) used to initialize a pseudo random number generator.

- The most important random numbers are from uniform distributed numbers.  
`> runif(n,a,b)`
- Numbers selected from a non-uniform probability distribution can be generated using a uniform distribution PRNG and a function that relates the two distributions.
- Assume you have uniformly distributed random numbers from  $[0, 1]$ , how do you extend it to  $[a, b]$ ?

# Normal Distribution

- The normal density function

$$f(x, \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

> dnorm(x,mu,sigma)

> dnorm(x,mu,sigma, log=TRUE)

- In theory, `dnorm(x,mu,sigma, log=TRUE)==log(dnorm(x,mu,sigma))` but `dnorm(x,mu,sigma, log=TRUE)` but is more stable for very large values.

Why?

- We love logs.

- The CDF (cumulate desity function)

$$\Phi(x) = \int_{-\infty}^x f(t, \mu, \sigma) dt$$

> pnorm(q,mu,sigma)

- The quantile (Given CDF, what is x?), i.e.  $\Phi^{-1}(p)$

> qnorm(p,mu,sigma)

- Random numbers from normal distribution

> rnorm(n,mu,sigma)

# Likelihood function

- Given that  $x_i \sim N(\mu, \sigma)$  for  $i = 1, \dots, n$ , the **likelihood function** is

$$\prod_{i=1}^n f(x_i, \mu, \sigma)$$

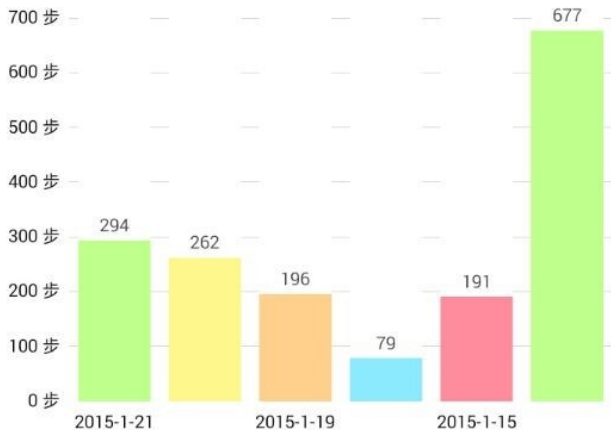
- However the **log likelihood function** is more often used

$$\sum_{i=1}^n \log f(x_i, \mu, \sigma)$$

- Do you know why?

## How long do you walk every day?

- Here is a list about my past six days walking statistics. Can you estimate how long do I walk everyday? and what is the variation?



# The likelihood function

- We assume everyday's walking steps ( $x_i$ ) are independent, and  $x_i$  follows standard normal distribution  $\sim N(\mu, \sigma)$ , the corresponding likelihood function is

$$\prod_{i=1}^n f(x_i, \mu, \sigma)$$

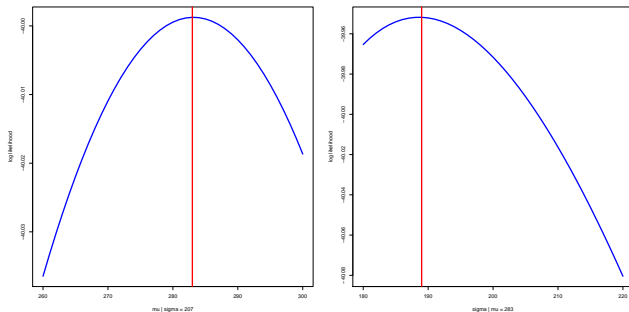
which can be easily written in R as

```
logNormLike <- function(mu, sigma, data)
{
  out = sum(dnorm(x = data, mean = mu, sd = sigma, log = TRUE))
  return(out)
}
```

- **The scope** Find a proper combination of  $\mu$  and  $\sigma$  that maximizes the loglikelihood function.

# Conditional likelihood function I

- Fix other parameters

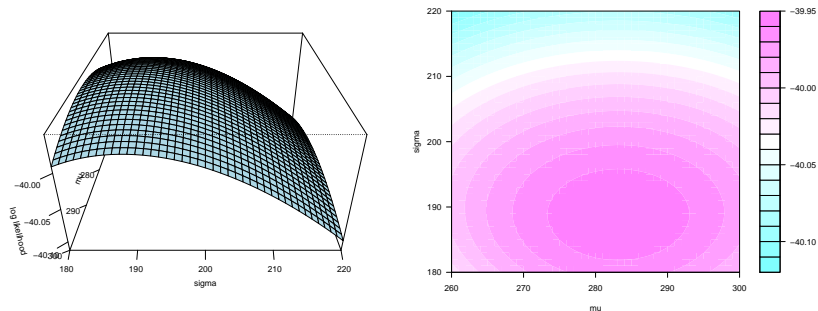


**Figure:** Left: fix variance to allow  $\mu$  to change with likelihood function. Right: fix mean to allow  $\sigma$  to change with likelihood function.

- Are  $\mu$  and  $\sigma$  we obtained the best combination?



# Conditional likelihood function II



**Figure:** 2D and 3D loglikelihood function

# Likelihood function for linear regression

- Assume you want to make a regression model

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i$$

where  $\epsilon_i \sim N(0, \sigma^2)$

- What is the (log) likelihood function?
- What are the unknown parameters?
- How do we estimate the parameters?
  - Write down a likelihood function with respect to the unknown parameters.
  - Use an optimization algorithm to find the estimates of the unknown parameters.

# The Likelihood function

```
logNormLikelihood <- function(par, y, x)
{
  beta0 <- par[1]
  beta1 <- par[2]
  sigma <- par[3]

  mean <- beta0 + x*beta1

  logDens <- dnorm(x = y, mean = mean,
                  sd = sigma, log = TRUE)
  loglikelihood <- sum(logDens)

  return(loglikelihood)
}
```





## Suggested reading

- Jones (2009): **Chapter 14, 15, 16**