Part 4 Simualtion and stochastic process

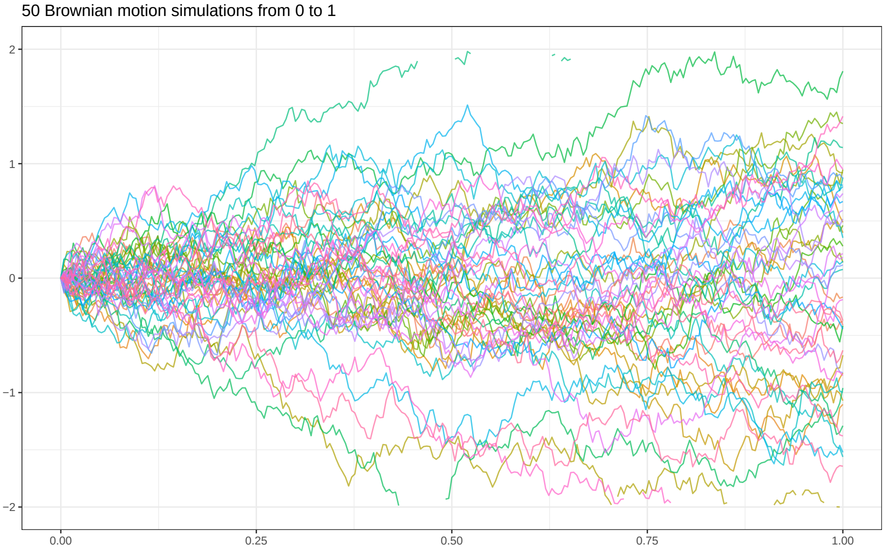
Dr. Nguyen Quang Huy

Dec 08, 2020

Brownian motion

- Brownian motion: named after a Scottish botanist and paleobotanist Robert Brown (1773-1858)
- Brownian motion is also called Weiner process (named after a mathematician Norbert Weiner(1894-1964))
- In mathematics, 1 dimensional-Brownian motion denoted by $(B_t)_{t \geq 0}$ is a continuous-time stochastic process characterized by three facts
  1. $B_0 = 0$
  2. $B_{t_2} - B_{t_1} \perp\!\!\!\perp B_{s_2} - B_{s_1}$ where $t_2 \geq t_1 \geq s_2 \geq s_1$
  3. $B_t - B_s \sim \mathcal{N}(0, t - s)$ where $t \geq s$

Brownian motion



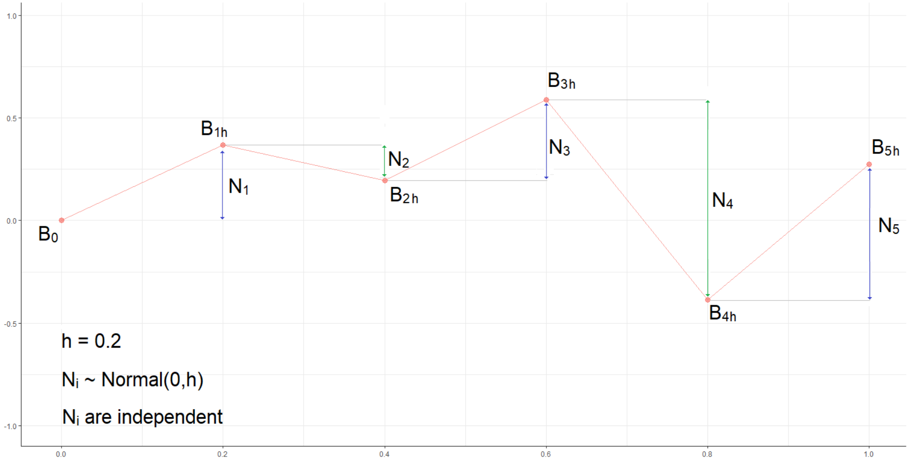
Brownian motion

To simulate a stochastic process (in general) $(X)_{t \leq s \leq T}$ from t to T

- Let $\Delta = \dfrac{T-t}{n}$
- For $i = 1, 2, \cdots, n$
  - Determine the conditional distribution of $X_{i\Delta}|X_{(i-1)\Delta}$ (or $\left(X_{i\Delta} - X_{(i-1)\Delta}\right)|X_{(i-1)\Delta}$)
  - Simulate $X_{i\Delta}$ based on the conditional distribution of $X_{i\Delta}|X_{(i-1)\Delta}$

If X_t is a Brownian motion (B_t), what is conditional distribution of $B_{i\Delta}|B_{(i-1)\Delta}$?

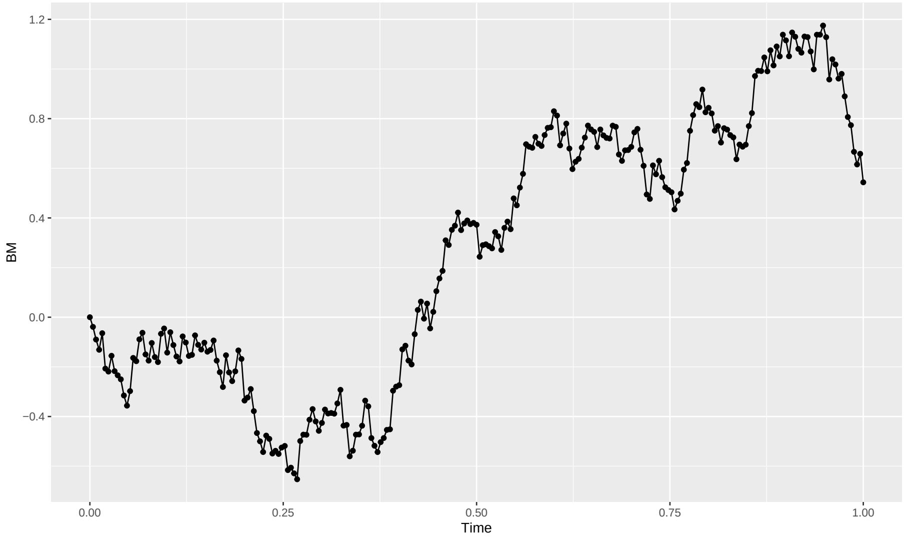
Brownian motion



Brownian motion

```
T<-1
n<-250
delta<-T/n
time<-seq(0,T,length=(n+1))
# Vector BM contains simulated value of a BM
BM<-rep(0,(n+1))
for (i in 2:(n+1)){
  # (BM[i] - BM[i-1]) is a normal r.v (0,delta)
  Ni<-rnorm(1,0,1)[1]*sqrt(delta)
  BM[i]<-BM[i-1]+Ni
}
dat<-data.frame("Time"=time,"BM"=BM)
dat%>%ggplot(aes(Time,BM))+geom_point()+geom_line()
```

Brownian motion



Brownian motion

- Write a code to simulate 100 Brownian motion from time 0 to time 5?

Application of brownian motion

Given $(X_t)_{t \geq 0}$ where $X_0 = 0$ and

$$X_{t+\Delta} = X_t + \mu \times \Delta + \sigma \times (B_{t+\Delta} - B_t)$$

or we can write (in a stochastic differential equation)

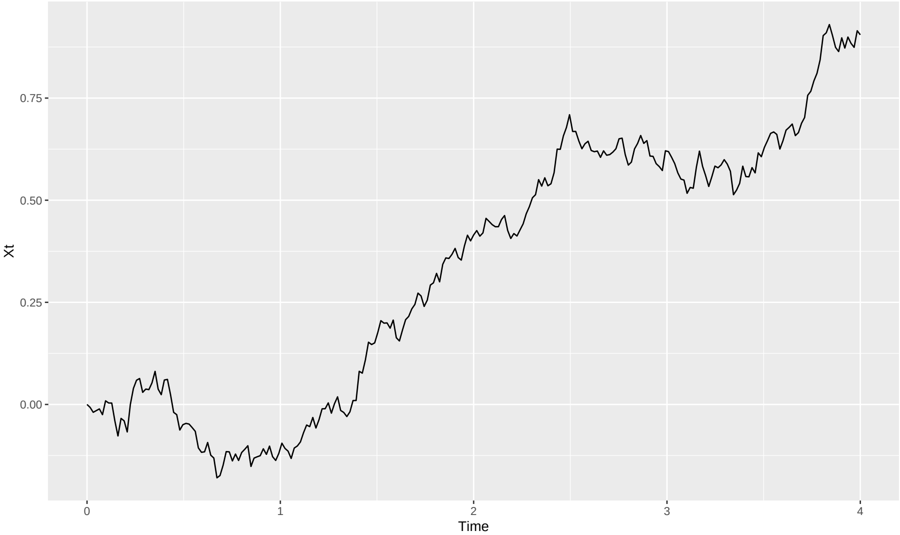
$$dX_t = \mu dt + \sigma dB_t$$

where

$$\begin{cases} \Delta & \rightarrow dt \\ X_{t+\Delta} - X_t & \rightarrow dX_t \\ B_{t+\Delta} - B_t & \rightarrow dB_t \end{cases}$$

What is the distribution function of $dX_t | X_t$?

Application of brownian motion



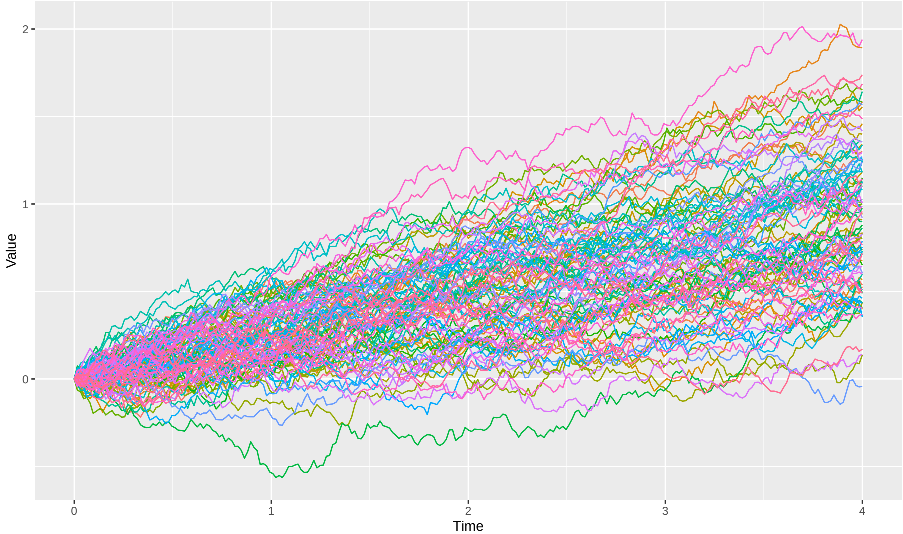
Application of brownian motion

Given $(X_t)_{t \geq 0}$ where $X_0 = 0$ and

$$X_{t+\Delta} = X_t + \mu \times \Delta + \sigma \times (B_{t+\Delta} - B_t)$$

- Prove that X_T has a normal distribution with mean μT and variance $\sigma^2 T$.
- Simulate m = 100 processes X_t from 0 with T = 4, n = 1000, $\mu = 0.25$ and $\sigma = 0.2$

Application of brownian motion



Application of brownian motion

- A well-known assumption of Black-Scholes model: stock price return satisfies the following SDE

$$\frac{dS_t}{S_t} = \mu dt + \sigma dB_t$$

  where μ is the expected return of stock price under **the real world probability measure**. σ is the standard deviation of stock price return. (σ is called stock price return's volatility).

- Given S_t, what is the distribution function of dS_t?

Application of brownian motion

- A well-known assumption of Black-Scholes model: stock price return satisfies the following SDE

$$\frac{dS_t}{S_t} = \mu dt + \sigma dB_t$$

where μ is the expected return of stock price under **the real world probability measure**. σ is the standard deviation of stock price return. (σ is called stock price return's volatility).

- Given S_t, what is the distribution function of dS_t?

$$\begin{aligned} dS_t &= S_t \mu dt + S_t \sigma dB_t \\ dS_t &\sim \mathcal{N}\left(dt(S_t\mu), dt(S_t\sigma)^2\right) \end{aligned}$$

Application of brownian motion

Write a function of parameters (S_0, T, n, μ, σ) to simulate a stock price from t to T satisfying the SDE

$$\frac{dS_t}{S_t} = \mu dt + \sigma dB_t$$

Application of brownian motion

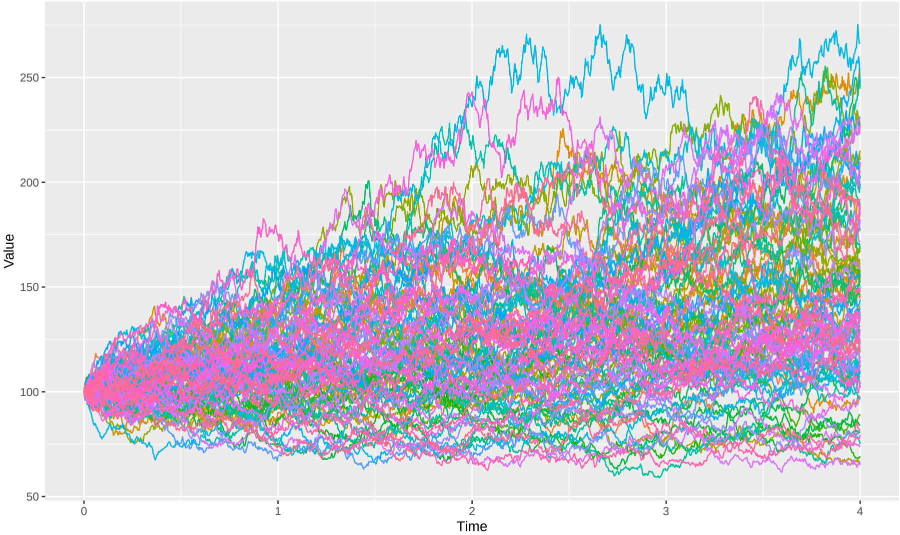
- Write a code to simulate 10 stock price from time t = 0 to time T = 4 with $S_0 = 100$, $\mu = 0.1$, $\sigma = 0.2$, and $n = 1000$

Application of brownian motion

- Write a code to simulate 10 stock price from time $t = 0$ to time $T = 4$ with $S_0 = 100$, $\mu = 0.1$, $\sigma = 0.2$, and $n = 1000$

```
T<-4
mu<-0.1
sigma<-0.2
S0<-100
time<-seq(0,T,length=(n+1))
n<-1000
dat<-data.frame("Time"=time)
for (i in 1:100){
  St<-StockPriceMC(S0,T,n,mu,sigma)
  dat<-mutate(dat,St)
  names(dat)[(i+1)]<-paste0("S",i)
}
dat%>%gather("StockPrice",Value,"S1":"S100")%>%
  ggplot(aes(Time,Value,group=StockPrice,color=StockPrice))+ge
```

Application of brownian motion



Option pricing and MC simulation

If $f(S_T)$ is the pay-off of a derivative product based on an underlying asset S_T then the price (fair price) of this product can be calculated by

$$f_0 = e^{-rT}\ \mathbb{E}^{\mathbb{Q}}(f(S_T))$$

where f_0 is the fair price of the derivative at time 0, r is the continuous risk-free interest rate, and $\mathbb{Q}$ is the **risk neutral** probability measure.

In the case that S_t is a stock price, S_t satisfies the following SDE

$$\frac{dS_t}{S_t} = rdt + \sigma dB_t$$

where B_t is a Brownian motion under $\mathbb{Q}$.

Option pricing and MC simulation

- When $f(S_T) = max(S_T - K, 0) = (S_T - K)^+$ we have an european Call option on stock S with strike price K.
- When $f(S_T) = max(K - S_T, 0) = (K - S_T)^+$ we have an european Put option on stock S with strike price K.

The fair price of Call and Put option can be written as follows:

$$c_0 = e^{-rT}\ \mathbb{E}^{\mathbb{Q}}\left((S_T - K)^+\right)$$
$$p_0 = e^{-rT}\ \mathbb{E}^{\mathbb{Q}}\left((K - S_T)^+\right)$$

where S_t satisfies the SDE

$$\frac{dS_t}{S_t} = rdt + \sigma dB_t$$

Option pricing and MC simulation

1. Simulate $(S_t)_{t\geq 0}$ using the following SDE to obtain S_T.

$$\frac{dS_t}{S_t} = rdt + \sigma dB_t$$

2. Calculate the option payoff

$$C_T = ((S_T - K)^+)$$
$$P_T = ((K - S_T)^+)$$

3. Repeat step 1. and 2. (N times) to obtain a vector of Call option payoff V_{C_T} and a vector of Put option payoff V_{P_T}

4. Return

$$c_0 = exp(-rT) \times mean\,(V_{C_T})$$
$$p_0 = exp(-rT) \times mean\,(V_{P_T})$$

Option pricing and MC simulation

Using $N = 10^4$ simulation, calculate the price of an european call option with $S_0 = 100$, $T = 1$, $r = 2\%$, $\sigma = 20\%$, $K = 100$.

```
T<-1
r<-0.02
n<-52
sigma<-0.2
S0<-100
K<-100
N<-10^5
CT<-rep(0,N)
for (i in 1:N){
  ST<-StockPriceMC(S0,T,n,r,sigma)[n+1]
  CT[i]<-max(ST-K,0)
}
mean(CT)*exp(-r*T)
```

Option pricing and MC simulation

Using $N = 10^4$ simulation, calculate the price of an european put option with $S_0 = 100$, $T = 1$, $r = 2\%$, $\sigma = 20\%$, $K = 100$.

```
T<-1
r<-0.02
n<-52
sigma<-0.2
S0<-100
K<-100
N<-10^5
PT<-rep(0,N)
for (i in 1:N){
  ST<-StockPriceMC(S0,T,n,r,sigma)[n+1]
  PT[i]<-max(K-ST,0)
}
mean(PT)*exp(-r*T)
```

Option pricing and MC simulation

There are analytic formulas for european Call and european Put option as follow:

$$
\begin{aligned}
c_0 &= S_0\ \Phi(d_1) - e^{-rT} K\ \Phi(d_2) \\
p_0 &= e^{-rT} K\ \Phi(-d_2) - S_0\ \Phi(-d_1)
\end{aligned}
$$

where Φ is the distribution function of standard normal random variable and

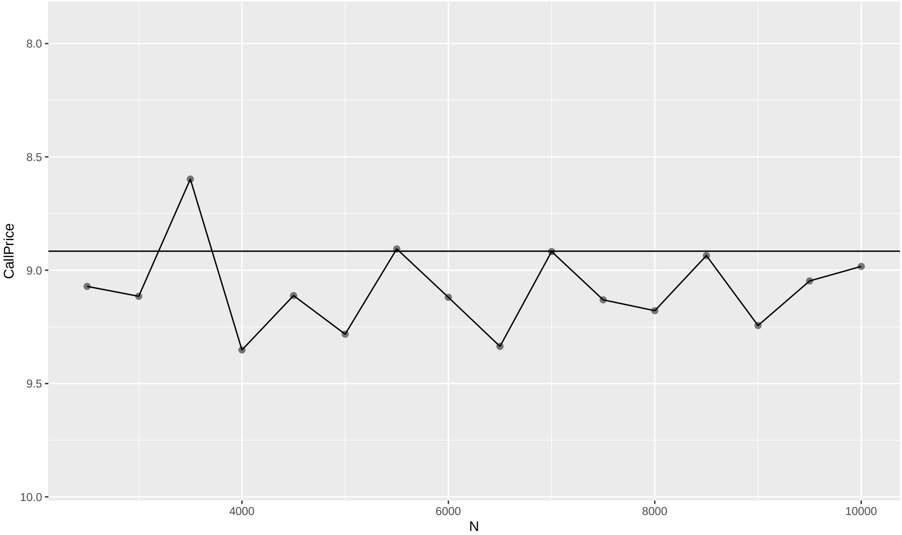
$$
\begin{aligned}
d_1 &= \frac{log(S_0/K) + (r + \sigma^2/2)\ T}{\sigma\sqrt{T}} \\
d_2 &= d_1 - \sigma\sqrt{T}
\end{aligned}
$$

Write functions to calculate the fair price of european call and european put option using these folmulas. Compare the results with MC simulation.

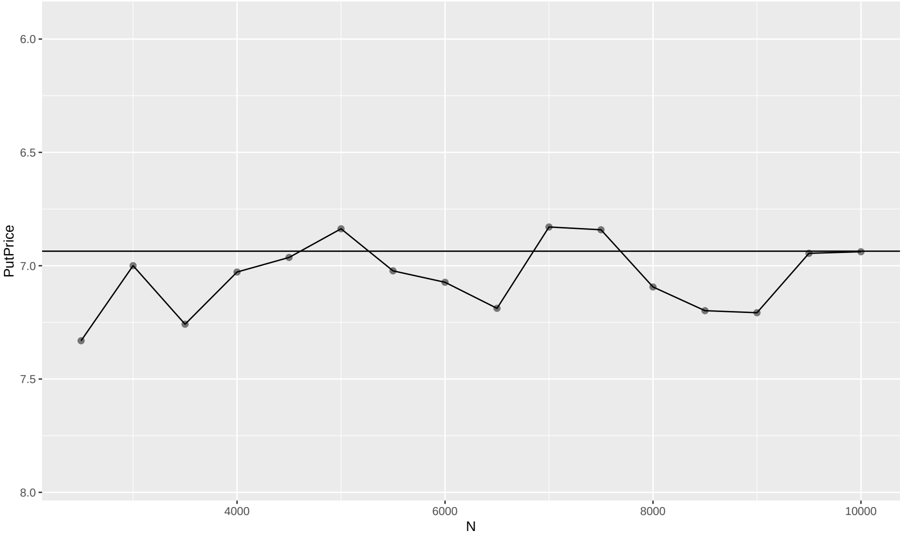
Option pricing and MC simulation

```r
EuCallPrice<-function(S0,K,T,r,sigma){
  d1<-(log(S0/K)+(r+sigma^2/2)*T)/(sigma*sqrt(T))
  d2<-d1-sigma*sqrt(T)
  EuCallPrice<-S0 * pnorm(d1) - exp(-r*T)*K*pnorm(d2)
}
c0<-EuCallPrice(S0,K,T,r,sigma)
EuPutPrice<-function(S0,K,T,r,sigma){
  d1<-(log(S0/K)+(r+sigma^2/2)*T)/(sigma*sqrt(T))
  d2<-d1-sigma*sqrt(T)
  EuPutPrice<- - S0 * pnorm(-d1) + exp(-r*T)*K*pnorm(-d2)
}
p0<-EuPutPrice(S0,K,T,r,sigma)
```

Option pricing and MC simulation



Option pricing and MC simulation



Option pricing and MC simulation

1. **Barrier call option:** Tính giá của một barrier call option trên cổ phiếu S với $S_0 = 100$, $\sigma = 20\%$, $r = 2\%$, $T = 1$ và $K = 100$ biết rằng payoff của barrier call option là $(S_T - K)^+$ tuy nhiên người sở hữu quyền chọn bán chỉ được chi trả với điều kiện S_T lớn hơn một mức giá barrier $(P_B = 105)$

2. **Asian call option:** Khác với quyền chọn kiểu châu Âu sử dụng giá cổ phiếu tại thời điểm đáo hạn T, quyền chọn kiểu châu Á sử dụng giá trị trung bình của cổ phiếu S trong khoảng thời gian từ 0 đến T. Mức giá trung bình của cổ phiếu được tính bằng trung bình cộng của giá cổ phiếu tại các thời điểm được xác định trước. Tính giá của quyền chọn kiểu châu Á với $S_0 = 100$, $\sigma = 20\%$, $r = 2\%$, $T = 1$ và $K = 100$ với payoff $(\bar{S}_T - K)^+$ trong đó

$$\bar{S}_T = \frac{S_{T/4} + S_{2T/4} + S_{3T/4} + S_T}{4}$$

Option pricing and MC simulation

```r
# BARRIER CALL OPTION
T<-1
r<-0.02
n<-52
sigma<-0.2
S0<-100
K<-100
PB<-105
N<-10^4
CT<-rep(0,N)
for (i in 1:N){
  ST<-StockPriceMC(S0,T,n,r,sigma)[n+1]
  CT[i]<-ifelse(ST>105,max(ST-K,0),0)
}
mean(CT)*exp(-r*T)
```

```
## [1] 8.076387
```

Option pricing and MC simulation

```
# ASIAN CALL OPTION
T<-1
r<-0.02
n<-52
sigma<-0.2
S0<-100
K<-100
N<-10^4
CT<-rep(0,N)
for (i in 1:N){
  St<-StockPriceMC(S0,T,n,r,sigma)
  CT[i]<-max((St[n/4+1]+St[2*n/4+1]+St[3*n/4+1]+St[4*n/4+1])/4
}
mean(CT)*exp(-r*T)
```

```
## [1] 5.857523
```

Poisson process

The **homogeneous** Poisson process is a stochastic process $N(t)$,$(t \geq 0)$ has the following three properties:

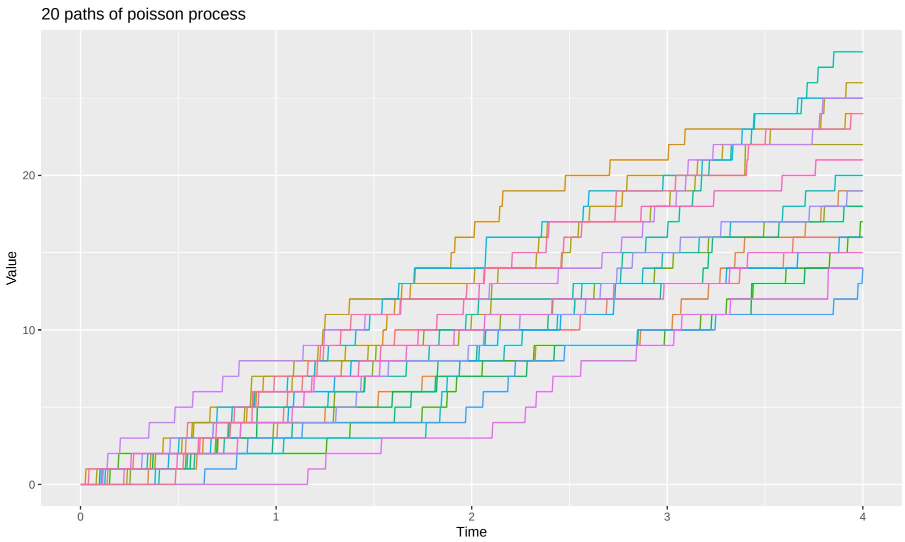
1. $N(0) = 0$
2. $N(t_2) - N(t_1)$ and $N(s_2) - N(s_1)$ are independent with $t_2 \geq t_1 \geq s_2 \geq s_1$
3. $N(t + \Delta) - N(t)$ is a Poisson r.v with mean $\lambda\Delta$

Write R code to simulate a Poisson process from 0 to $T = 4$ with $\lambda = 5$ using $n = 1000$

Poisson process

```
k<-20 # number of Nt
T<-4
lambda<-5
n<-1000
delta<-T/n
Nt<-matrix(0,k,(n+1))
for (i in 2:(n+1)){
  dNt<-rpois(k,lambda*delta)
  Nt[,i]<-Nt[,(i-1)]+dNt
}
dat<-as.data.frame(t(Nt))
names(dat)<-paste0("Nt",1:k)
dat<-dat%>%mutate(Time=seq(0,T,T/n))%>%gather(Nt,Value,Nt1:pas
dat%>%ggplot(aes(Time,Value,group=Nt,col=Nt))+geom_line()+
  theme(legend.position = "none")+ggtitle(paste0(k, " paths of
```

Poisson process



Compound Poisson process

- A compound Poisson process is a process $(Y(t))$ given by

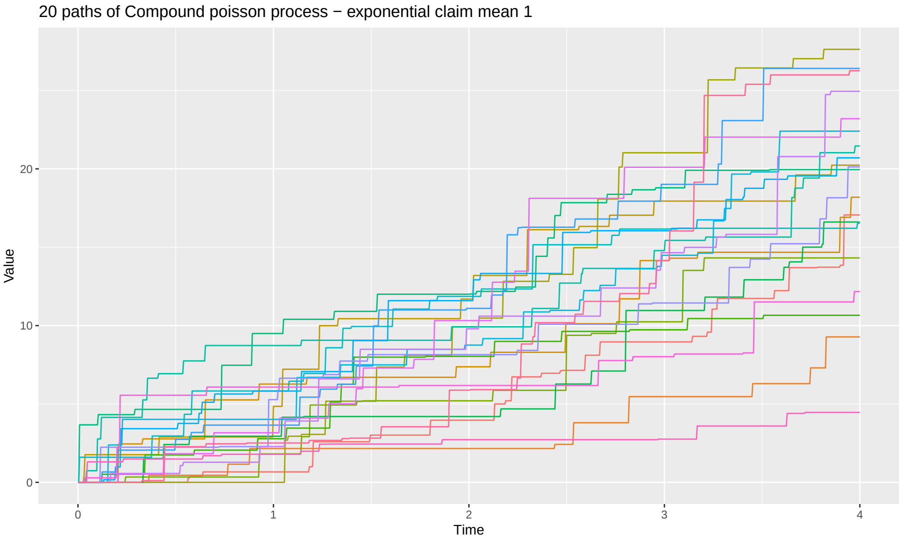
$$Y(t) = \begin{cases} 0 & \text{if N(t) = 0} \\ \sum_{i=1}^{N(t)} X_i & \text{, otherwise} \end{cases}$$

where X_i are i.i.d random variables and $N(t)$ is an homogeneous Poisson process.

$$dY(t) = \begin{cases} 0 & \text{if dN(t) = 0} \\ \sum_{i=1}^{dN(t)} X_i & \text{, otherwise} \end{cases}$$

- Write R code to simulate a Compound poisson process from 0 to $T = 4$ with $\lambda = 5$, and $X_i \sim exp(1)$ using $n = 1000$

Compound poisson process



Application of Poisson process

An insurance company has initial capital u_0. Suppose that the premium is paid at a constant rate c i.e. from time t to time $(t + \Delta)$, the premium collected is $c\Delta$. Number of claim from time t to time $(t + \Delta)$ is a Poisson r.v with mean $\lambda\Delta$ and claim amount is an exponential random variable with mean μ. The capital process of the insurance company can be write as follow:

$$u_t = u_0 + ct - Y(t)$$

where $Y(t)$ is a compound poisson process.

With $u_0 = 3$, $c = 2$, $\lambda = 10$ and $\mu = 0.1$, simulate a path of u_t from 0 to $T = 4$