Introduction to Poisson processes with R

5 ModIA

2024/2025

1 - Homogeneous Poisson processes observed on a fixed window

First, we consider the case of a fixed observation window (and thus a random number of events).

1.1 - Simulation

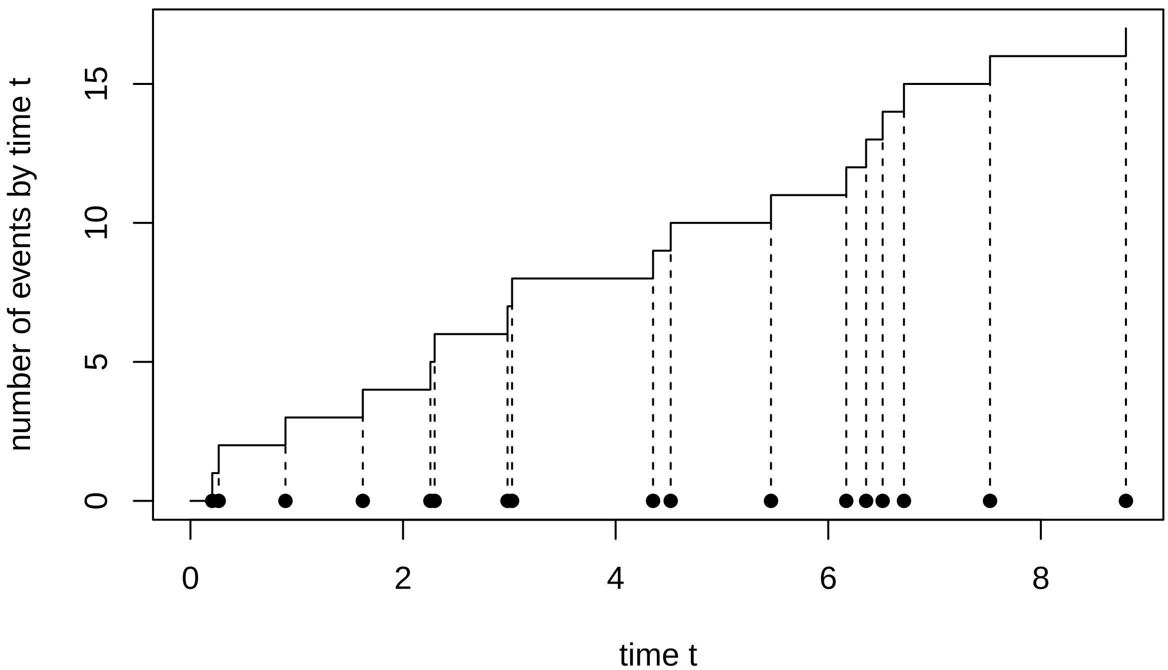
Simulation of a homogeneous Poisson process with intensity `lambda` on the window [0,`Tmax`].

```
simulPPh1 <- function(lambda,Tmax)
{
  N_Tmax <- rpois(1,lambda*Tmax)
  return(sort(runif(N_Tmax,0,Tmax)))
}
```

Let us define a plot function for a counting process `PP`.

```
plot.PP<- function(PP)
{
  # plot the counting process (with jumps of size 1 (starting at point (0,0))):
  plot(c(0,PP),0:length(PP),type="s",xlab="time t",ylab="number of events by time t")
  # add the arrival times on the horizontal axis:
  points(PP,0*PP,type="p",pch=16)
  # link arrival times with the counts:
  lines(PP,0:(length(PP)-1),type="h",lty=2)
}
```

```
# Simulation and plot of a homogeneous Poisson process:
PPh1 = simulPPh1(lambda=2,Tmax=10)

# plot:
plot.PP(PPh1)
```



1.2 - Maximum Likelihood Estimator (MLE)

Maximum Likelihood estimator of a homogeneous Poisson process `PPh` observed on [0,`Tmax`]:

```
MLE1 <- function(PPh,Tmax)
{
  N_Tmax <- length(PPh)
  return(N_Tmax/Tmax)
}
```

```
# Application on an example:
MLE1(PPh1,10)
```

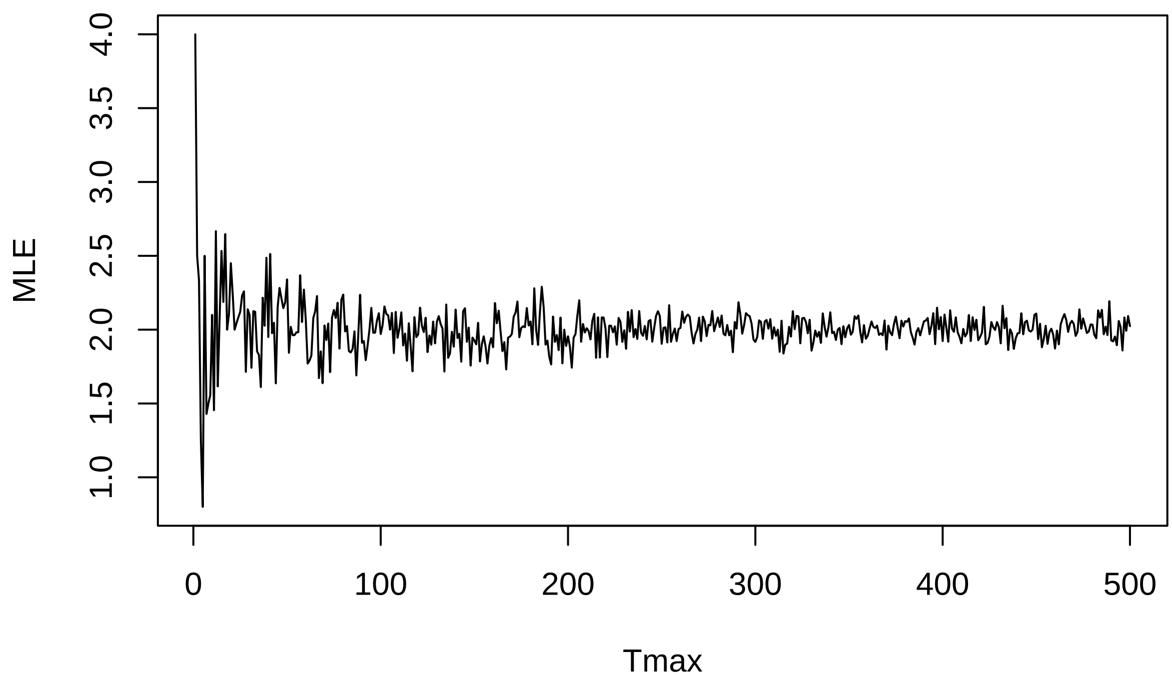
```
## [1] 1.7
```

Comment: On trouve $\widehat{\lambda}_T = \frac{\widehat{N}_T}{T} = 1.7 \approx 2$.

1.3 - Asymptotic behavior of the MLE

1.3.1 - LLN-type result

```r
lambda = 2
Tillustr = 1:500

MLE_Tmax = list()
for(Tmax in Tillustr)
{
  MLE_Tmax = c(MLE_Tmax,MLE1(simulPPh1(lambda,Tmax),Tmax))
}

# plot:
plot(Tillustr,MLE_Tmax,type="l",xlab="Tmax",ylab="MLE")
```

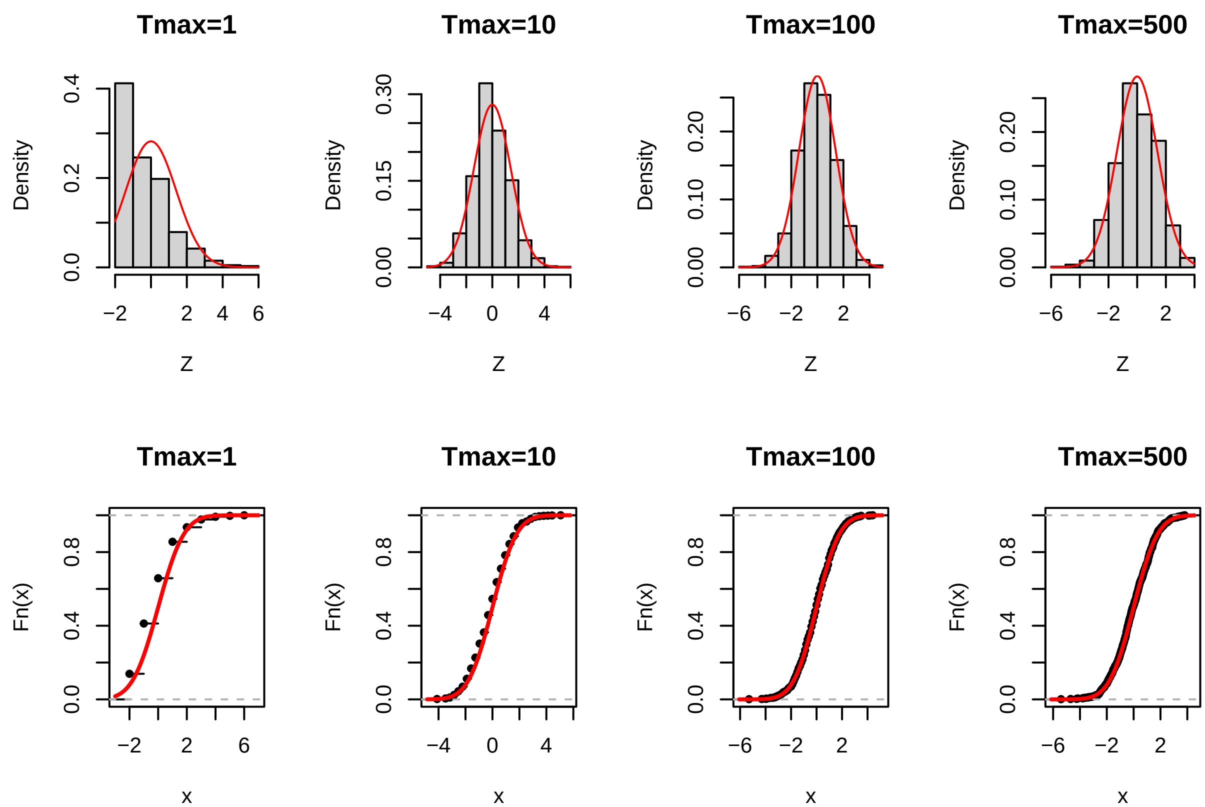


Comment: On observe que $\widehat{\lambda}_T \xrightarrow[T\to+\infty]{a.s} \lambda$.

1.3.2 - CLT-type result

```r
lambda = 2
Tillustr = c(1,10,100,500) # possible values of Tmax
K = 1000 # number of simulations of Z for each value of Tmax


Z_Tmax = list()
```

```r
# simulation of Z for each value of Tmax:
for(Tmax in Tillustr)
{
  Z=rep(0,K)
  for(k in 1:K)
  {
    pph=simulPPh1(lambda,Tmax)
    mle=MLE1(pph,Tmax)
    Z[k]=sqrt(Tmax)*(mle-lambda)
  }
  Z_Tmax[[as.character(Tmax)]] = Z
}

# plot:
par(mfrow=c(2,4))
# plot with density
for(Tmax in Tillustr)
{
  Z=Z_Tmax[[as.character(Tmax)]]
  hist(Z, freq=FALSE, main=paste("Tmax",Tmax, sep="="))
  curve(dnorm(x, mean=0, sd=sqrt(lambda)), col="red", add=TRUE)
}
# plot with cdf
for(Tmax in Tillustr)
{
  Z=Z_Tmax[[as.character(Tmax)]]
  plot(ecdf(Z), main=paste("Tmax",Tmax, sep="="))
  curve(pnorm(x, mean=0, sd=sqrt(lambda)), lwd=2, col="red", add=TRUE)
}
```



Comment: On observe que $\sqrt{T}(\widehat{\lambda}_T - \lambda) \xrightarrow[T\to+\infty]{\mathcal{L}} \mathcal{N}(0, \lambda)$.

1.4 - Statistical inference: hyptothesis testing

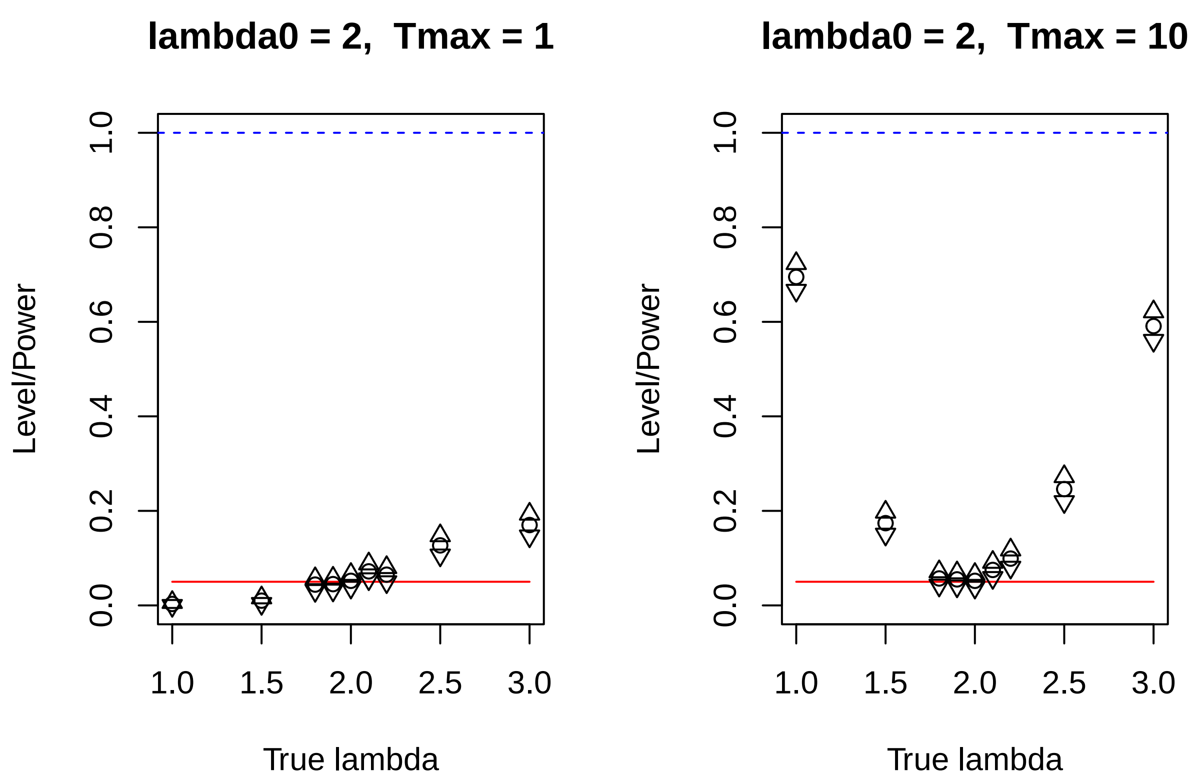
The `test1` function returns the p-value of the test of $\mathcal{H}_0 : \lambda =$ `lambda0` againts $\mathcal{H}_1 : \lambda \neq$ `lambda0` given the observation of a homogeneous Poisson process `PPh` on [0,`Tmax`].

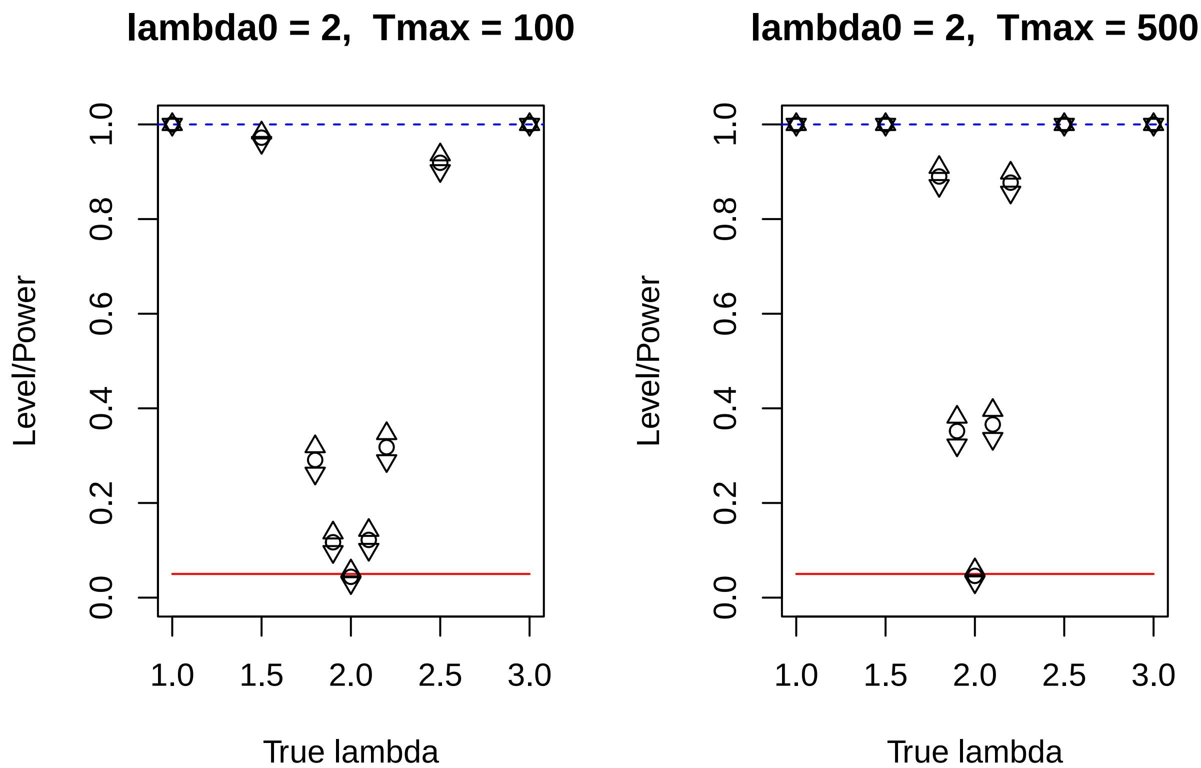
```
test1 <- function(PPh,Tmax,lambda0)
{
  MLE=MLE1(PPh,Tmax)
  Z=sqrt(Tmax/lambda0)*(MLE-lambda0)
  return(2*(1-pnorm(abs(Z))))
}
```

```
# Plot confidence intervals for the proportion of times alpha-level tests rejects the
# null hypothesis "lambda=lambda0" under each true distribution lambda in TrueLambda
plot.level.power1 <- function(Tmax,lambda0,TrueLambda,alpha,nsimu)
{
  plot(range(TrueLambda),c(alpha,alpha),ylim=c(0,1),xlab="True lambda",ylab="Level/Power",
       type="l",col="red",main=paste("lambda0 = ",lambda0,",  Tmax = ",Tmax,sep=""))
  abline(1,0,lty=2,col="blue")

  for(lambda in TrueLambda)
  {
    # estimating the proportion under lambda in TrueLambda
```

```
    propReject=0
    for(sim in 1:nsimu){
      propReject=propReject+(test1(simulPPh1(lambda,Tmax),Tmax,lambda0) <= alpha)/nsimu
    }
  # plot the confidence intervals
  points(lambda,propReject)
  points(lambda,propReject+sqrt(abs(propReject*(1-propReject))/nsimu)*qnorm(0.975),pch=2)
  points(lambda,propReject-sqrt(abs(propReject*(1-propReject))/nsimu)*qnorm(0.975),pch=6)
  }
}
```

```
# Application on an example:
alpha=0.05
nsimu=1000
lambda0=2
TrueLambda=c(1,1.5,1.8,1.9,2,2.1,2.2,2.5,3)

par(mfrow=c(1,2))
for(Tmax in c(1,10,100,500))
{
  plot.level.power1(Tmax,lambda0,TrueLambda,alpha,nsimu)
}
```





Comments: On observe que le niveau de test est bien α et que la puissance du test augmente avec $T_{\max}$.

2 - Inhomogeneous Poisson processes

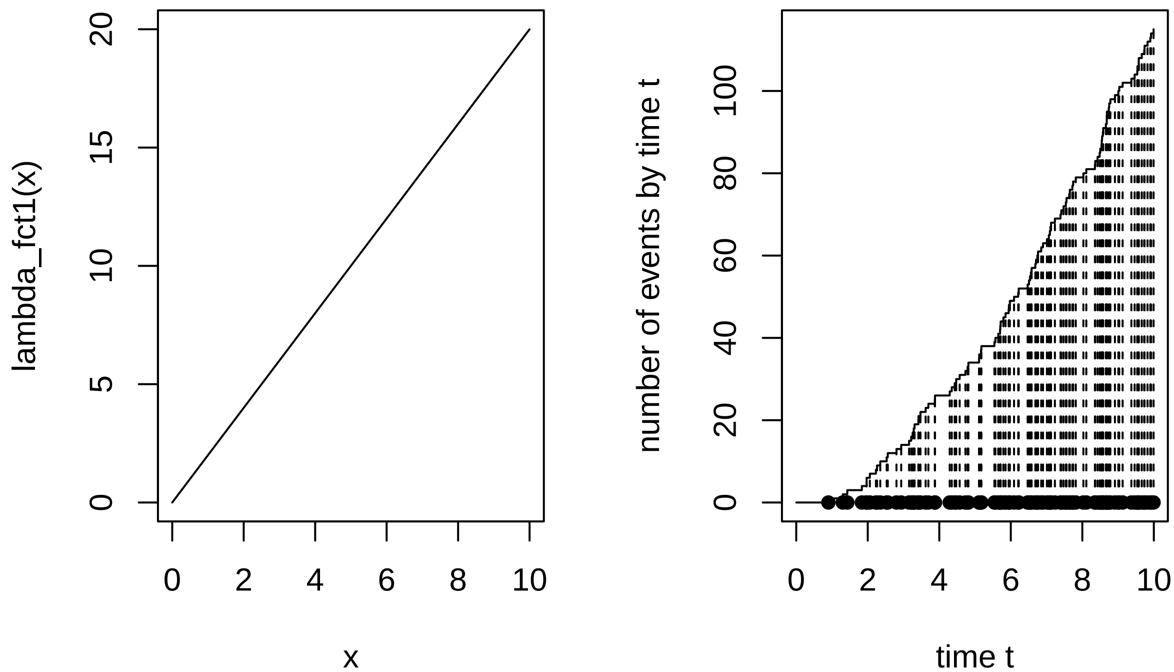
Simulation of an inhomogeneous Poisson processes with given intensity function `lambda_fct` on a fixed window [0,`Tmax`].

```
simulPPi = function(lambda_fct,Tmax,M)
{
  # simulation of a Poisson process with constant rate M
  hpp <- simulPPh1(M,Tmax)
  # simulation of the points:
  U <- runif(length(hpp),0,M)
  return(sort(hpp[U<=lambda_fct(hpp)]))
}
```

Application to $\lambda_1 : x \mapsto 2x$.

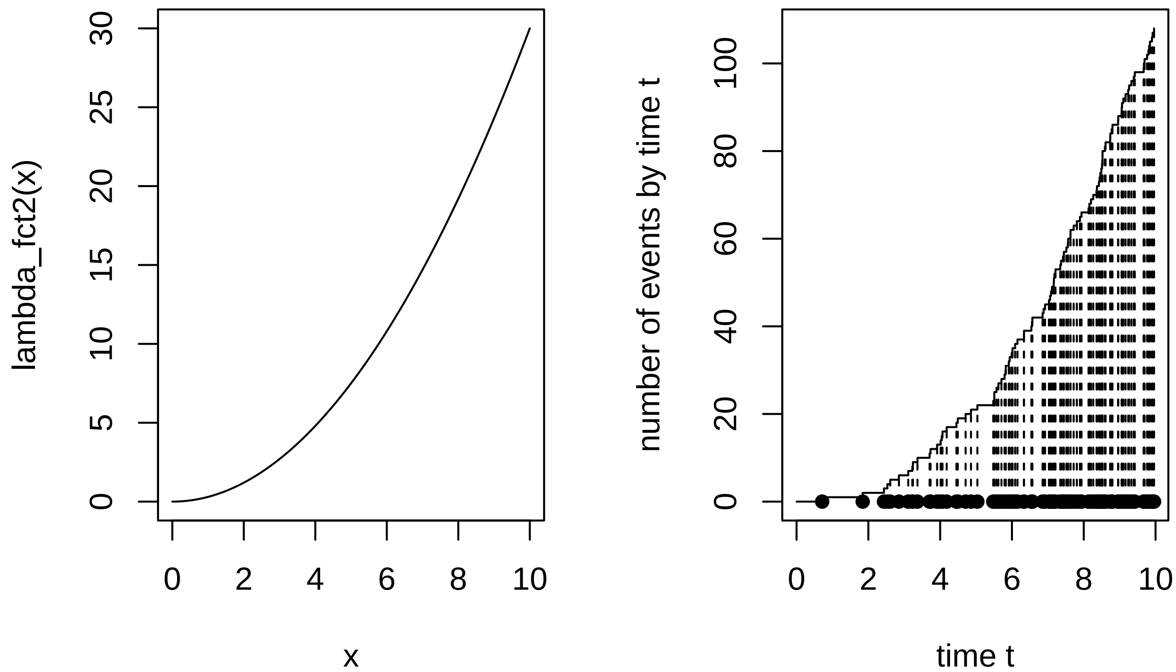
```
Tmax=10
lambda_fct1 <- function(x){return(2*x)}
M1=lambda_fct1(Tmax)
PPi1 = simulPPi(lambda_fct1,Tmax,M1)

# plot:
```

```
par(mfrow=c(1,2))
curve(lambda_fct1,from=0,to=Tmax,n=1000)
plot.PP(PPi1)
```



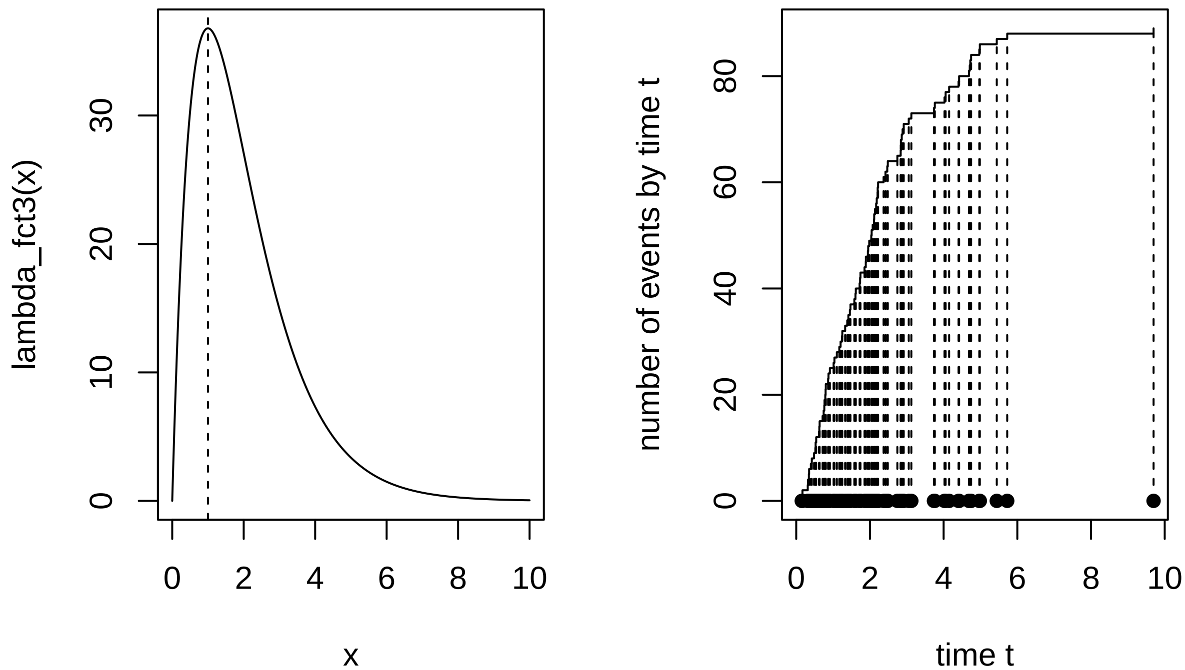
Application to $\lambda_2 : x \mapsto \frac{3}{10}x^2$.

```
Tmax=10
lambda_fct2 <- function(x){return(3/10*x^2)}
M2=lambda_fct2(Tmax)
PPi2 = simulPPi(lambda_fct2,Tmax,M2)

# plot:
par(mfrow=c(1,2))
curve(lambda_fct2,from=0,to=Tmax,n=1000)
plot.PP(PPi2)
```



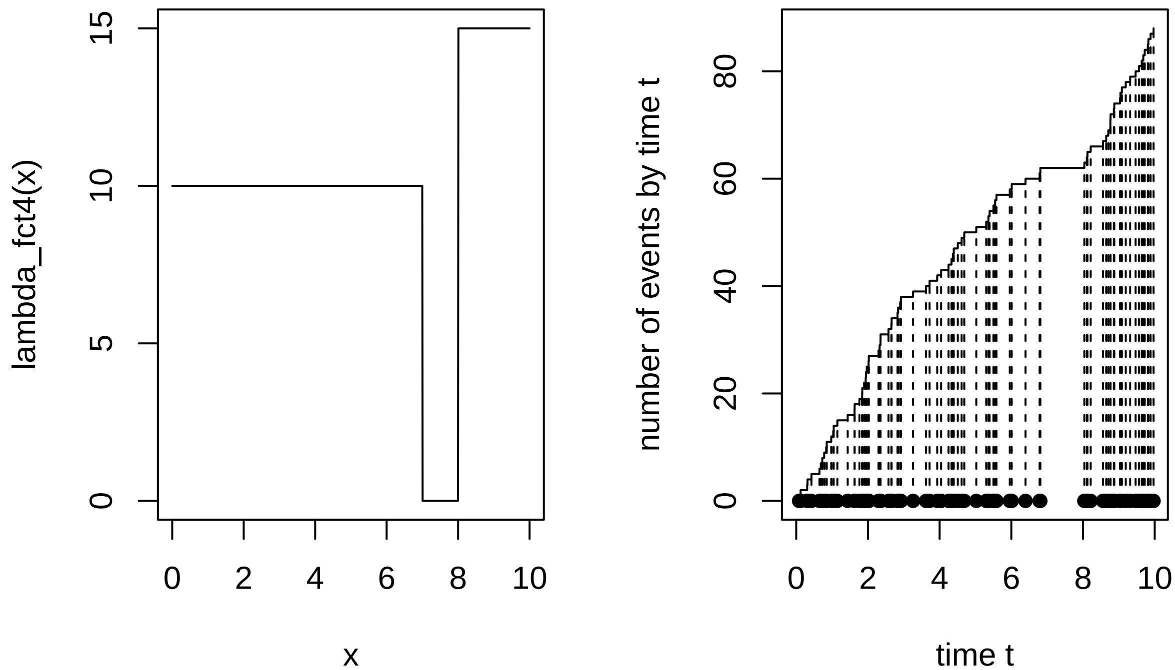
Application to $\lambda_3 : x \mapsto 100xe^{-x}$.

```
Tmax=10
lambda_fct3 <- function(x){return(100*x*exp(-x))}
M3=lambda_fct3(1)
PPi3 = simulPPi(lambda_fct3,Tmax,M3)


# plot:
par(mfrow=c(1,2))
curve(lambda_fct3,from=0,to=Tmax,n=1000);abline(v=1,lty=2)
plot.PP(PPi3)
```



Application to $\lambda_4 : x \mapsto 10 \times 1_{[0,7]}(x) + 15 \times 1_{]8,10]}(x)$.

```r
Tmax=10
lambda_fct4 <- function(x){return(10*(x<=7 & x>=0)+15*(x>8 & x<=10))}
M4=15
PPi4 = simulPPi(lambda_fct4,Tmax,M4)

# plot:
par(mfrow=c(1,2))
curve(lambda_fct4,from=0,to=Tmax,n=1000)
plot.PP(PPi4)
```



3 - Homogeneous Poisson processes with fixed number of points

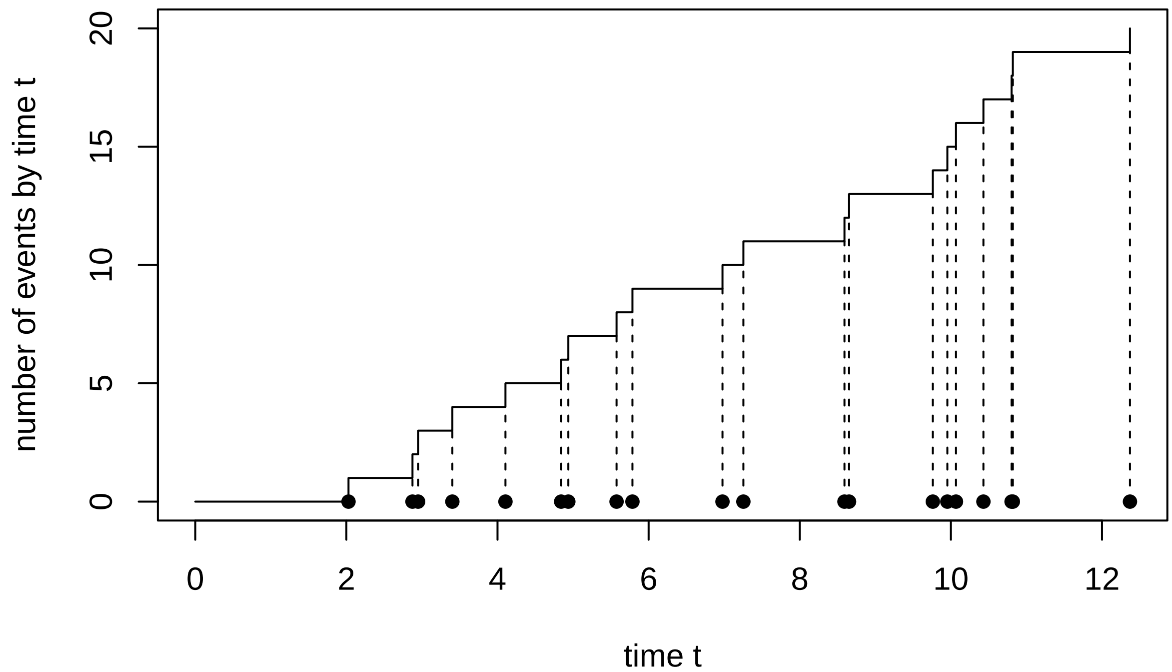
Now, we consider the case of a fixed number of points (and thus a random observation window).

3.1 - Simulation

Simulation of the `n` first arrival times of a homogeneous Poisson process with intensity `lambda`.

```
simulPPh2 <- function(lambda,n)
{
  return(cumsum((rexp(n,lambda))))
}
```

```
# Simulation and plot of a homogeneous Poisson process:
PPh2 = simulPPh2(lambda=2,n=20)

# plot:
plot.PP(PPh2)
```



3.2 - Maximum likelihood estimator

Maximum Likelihood estimator of a homogeneous Poisson process `PPh` with fixed number of points:

```
MLE2 <- function(PPh)
{
  return(length(PPh)/max(PPh))
}
```

```
# Application on an example:
MLE2(PPh2)
```

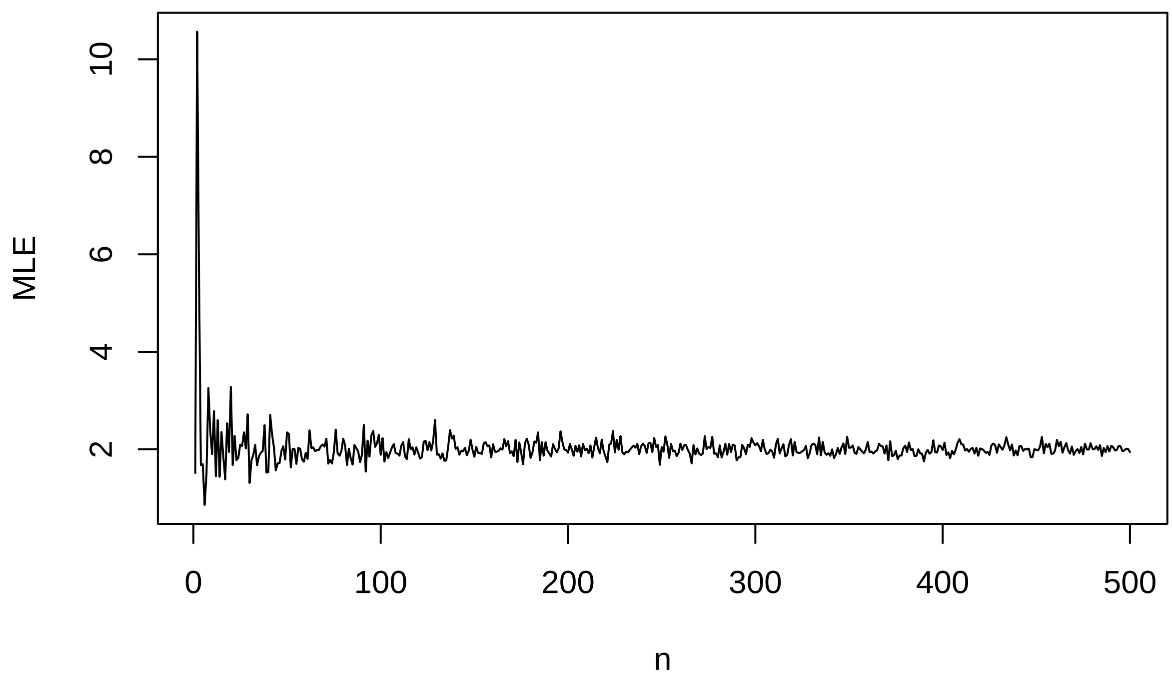
```
## [1] 1.616805
```

Comment: On trouve $\widehat{\lambda}_n = \frac{n}{\widehat{T}_n} = 1.6168046 \approx 2.$

3.3 Asymptotic behavior of the MLE

3.3.1 - LLN-type result

```r
lambda = 2
Nillustr = 1:500

MLE_n = list()
for(n in Nillustr)
{
  MLE_n = c(MLE_n,MLE2(simulPPh2(lambda,n)))
}

# plot:
plot(Nillustr,MLE_n,type="l",xlab="n",ylab="MLE")
```

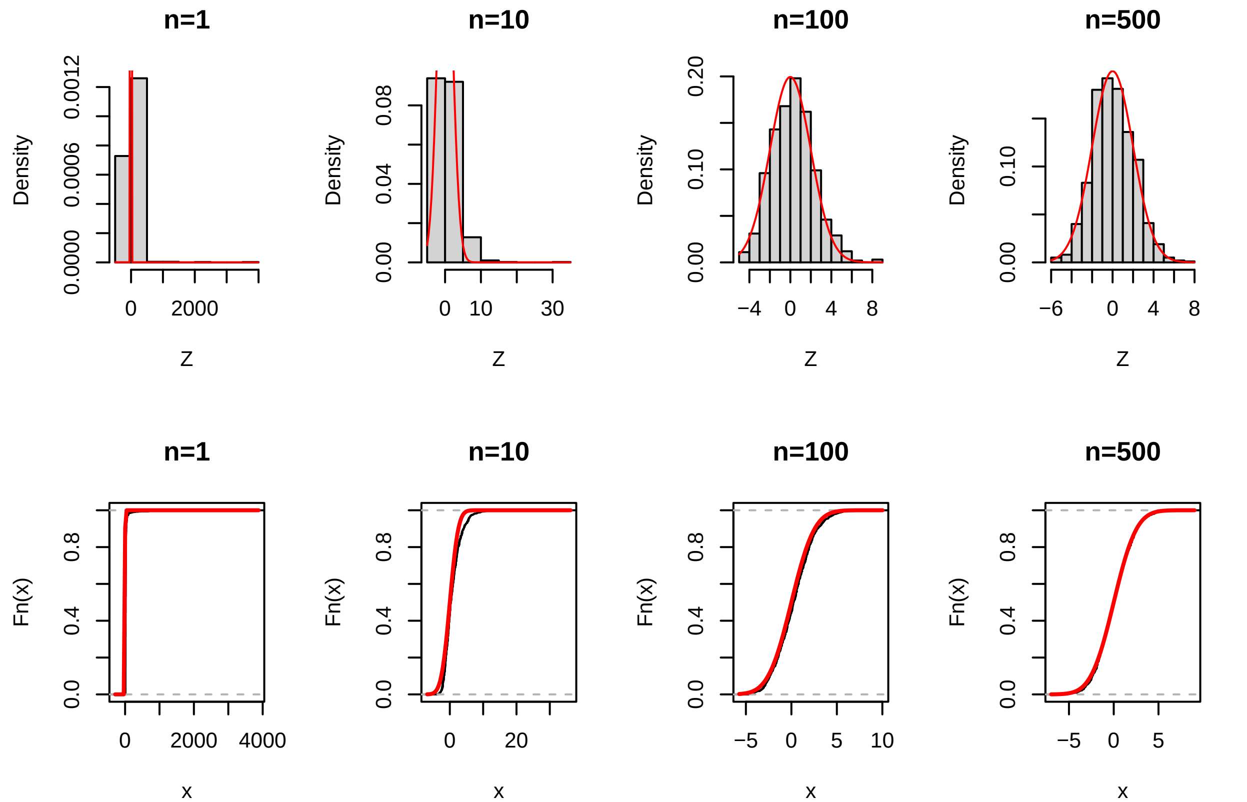


Comment: On observe que $\widehat{\lambda}_n \xrightarrow[n\to+\infty]{a.s} \lambda$.

3.3.2 - CLT-type result

```r
lambda = 2
Nillustr = c(1,10,100,500) # possible values of n
K = 1000 # number of simulations of Z for each value of n

Z_n = list()

# simulation of Z for each value of n:
for(n in Nillustr)
```

```
{
  Z=rep(0,K)
  for(k in 1:K)
  {
    pph=simulPPh2(lambda,n)
    mle=MLE2(pph)
    Z[k]=sqrt(n)*(mle-lambda)
  }
  Z_n[[as.character(n)]] = Z
}

# plot:
par(mfrow=c(2,4))
# plot with density
for(n in Nillustr)
{
  Z=Z_n[[as.character(n)]]
  hist(Z, freq=FALSE, main=paste("n",n, sep="="))
  curve(dnorm(x, mean=0, sd=lambda), col="red", add=TRUE)
}
# plot with cdf
for(n in Nillustr)
{
  Z=Z_n[[as.character(n)]]
  plot(ecdf(Z), main=paste("n",n, sep="="))
  curve(pnorm(x, mean=0, sd=lambda), lwd=2, col="red", add=TRUE)
}
```



Comment: On observe que $\sqrt{n}(\widehat{\lambda}_n - \lambda) \xrightarrow[n\to+\infty]{\mathcal{L}} \mathcal{N}(0,\lambda^2)$.

3.4 - Statistical inference : confidence intervals

Confidence interval for the unkown intensity of a homogeneous Poisson process `PPh` with confidence level 1-`alpha` that is `asymptotic` (by default) of not.

```
IC2 <- function(PPh,alpha=0.05,asymptotic=TRUE)
{
  if(asymptotic)
  {
    n=length(PPh)
    Tn=max(PPh)
    z_alpha=qnorm(1-alpha/2)
    return(c(n/(Tn*(1+z_alpha/sqrt(n))),n/(Tn*(1-z_alpha/sqrt(n)))))
  }
  else
  {
    n=length(PPh)
    Tn=max(PPh)
    x_alpha1=qchisq(alpha/2,2*n)
    x_alpha2=qchisq(1-alpha/2,2*n)
    return(c(x_alpha1/(2*Tn),x_alpha2/(2*Tn)))
  }
}
```

```r
# Application on an example:
PPh2 <- simulPPh2(lambda=10,n=100)
IC2(PPh2,alpha=0.05,asymptotic=TRUE)
```

```
## [1]  9.796898 14.573386
```

```r
IC2(PPh2,alpha=0.05,asymptotic=FALSE)
```

```
## [1]  9.533463 14.122442
```

Comment: On trouve l'intervalle de confiance asymptotique $\left[\frac{n}{T_n(1+\frac{z_{\alpha/2}}{\sqrt{n}})}, \frac{n}{T_n(1-\frac{z_{\alpha/2}}{\sqrt{n}})}\right]$ et l'intervalle de confiance non-asymptotique $\left[\frac{x_{2n,\alpha/2}}{2T_n}, \frac{x_{2n,1-\alpha/2}}{2T_n}\right]$.

```r
# Validation on simulated data
lambda=2 ; nsimu=1000 ; n=10 # or n=100
alpha=0.05
ICasymptotic=matrix(0,nsimu,2)
ICnonasymptotic=matrix(0,nsimu,2)
for(sim in 1:nsimu)
{
  PPh2 <- simulPPh2(lambda,n)
  ICasymptotic[sim,] = IC2(PPh2,alpha=alpha,asymptotic=TRUE)
  ICnonasymptotic[sim,] = IC2(PPh2,alpha=alpha,asymptotic=FALSE)
}
```

Comment: