Chapter 4 - Continuous Probability Distribution

Dr. Alessandro Ruggieri

Contents

Exponential distribution

The probability density function of an exponential distribution is equal to $f(x) = \lambda \exp(-\lambda x) \quad \forall x \geq 0$, while the cumulative distribution function is given by $F(x) = 1 - \exp(-\lambda x) \quad \forall x \geq 0$. It is not defined for $x < 0$.

Let's first assign the scale parameter:

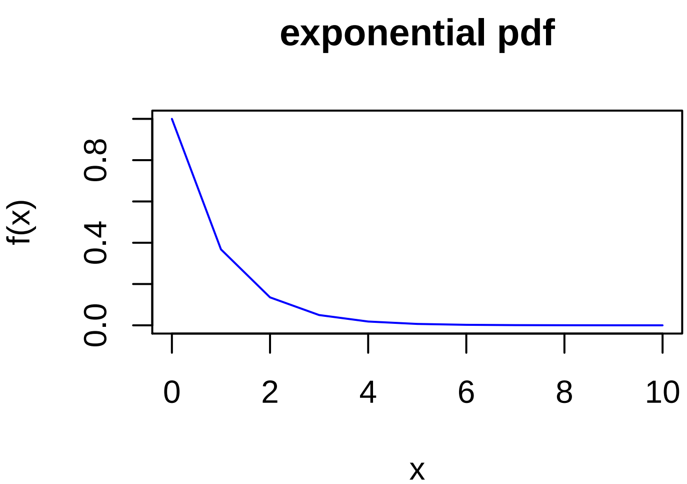
```r
# Scale parameter
lambda<-1
```

Then we assign the support of the distribution and compute pdf and cdf for any point of the support:

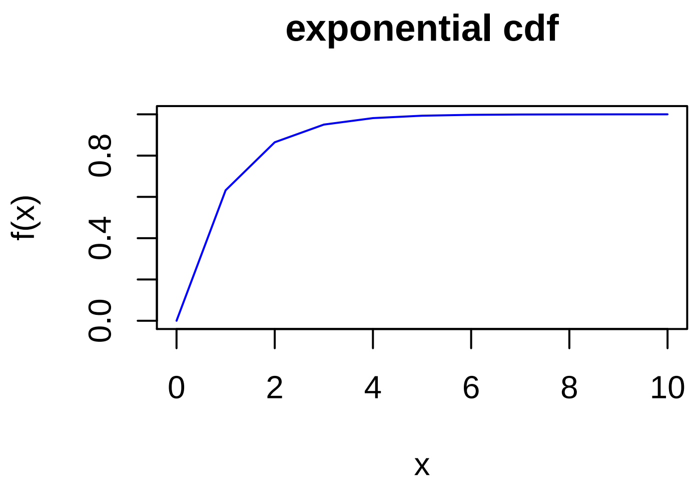
```r
# Support
x<-c(0:10)
# PDF
pdf<-lambda*exp(-lambda*x)
# CDF
cdf<-1-exp(-lambda*x)
```

Finally, we can plot the pdf and cdf

```r
# plot functions
par(mfrow=c(1,1))
plot(x,pdf,xlab="x", ylab="f(x)", main="exponential pdf", type="l", col="blue")
```



```r
plot(x,cdf,xlab="x", ylab="f(x)", main="exponential cdf", type="l", col="blue")
```



We can evaluate the exponential pdf and the cdf at a certain value of the support using the function dexp and p(exp), i.e.

```r
# value of the support
x<-2.5
# evaluate pdf at x
dexp(x,lambda)
```

```
## [1] 0.082085
```

```r
# evaluate cdf at x
pexp(x,lambda)
```

```
## [1] 0.917915
```

To check if the outcomes are correct, we can compare them to the values compute using the pdf and the cdf function above, i.e.

```r
# compare pdf at x with analytical value
dexp(x,lambda) == lambda*exp(-lambda*x)
```

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

```r
# compare cdf at x with analytical value
pexp(x,lambda) == 1-exp(-lambda*x)
```

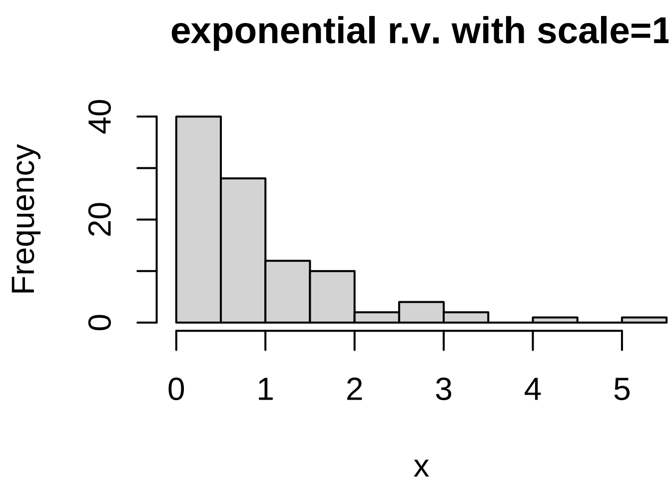
```
## [1] TRUE
```

We can also generate numbers from an exponential distribution using the function rexp( ) as follows

```r
# numbers to extracts
n<-100
# generate n numbers from exponential with scale lambda
v<-rexp(n, lambda)
```

and plot them using an histogram

```r
hist(v,main="exponential r.v. with scale=1", xlab="x")
```



To compute the probability that an exponential random variables lie between two scalar a and b, we can evaluate the function pexp( ) at those scalar and subtract them:

```
# lower bound
a<- 2
# upper bound
b<- 5
# evaluate cdf at a and b using R function
pa<-pexp(a,lambda)
pb<-pexp(b,lambda)
# compute probability
p<-pb-pa
print(p)
```

```
## [1] 0.1285973
```

Normal distribution

We can evaluate the PDF of a normal distribution at a point x using the function dnorm( ).

We first construct the support to be between -15 and 15, with steps of length equal to 0.05.

```
# Support
x <- seq(-15, 15, by = .05)
```

We then use the function dnorm( ) by specifying mean and standard deviation as options. If not specified, dnorm( ) uses a standard normal:

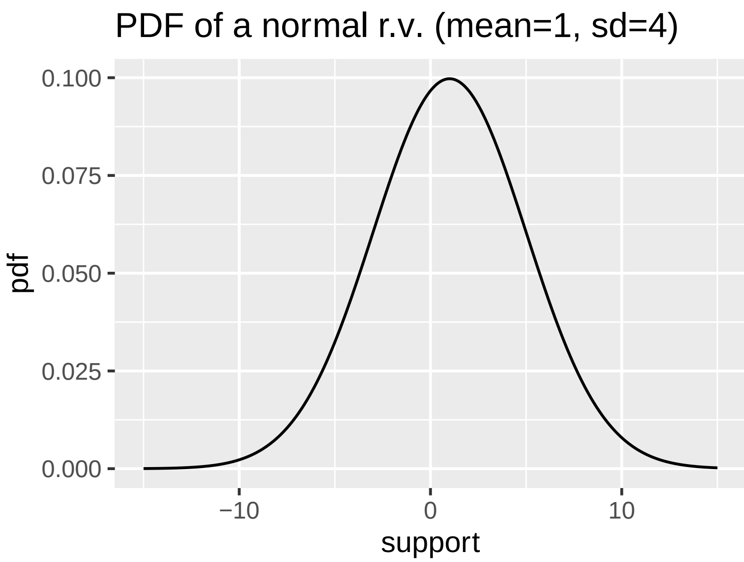
```
# PDF
y <-dnorm(x = x, mean = 1, sd = 4)
```

To plot the PDF, we define a data frame and use the function ggplot( )

```
# Plot PDF
norm_pdf <- data.frame(x,y)

library("ggplot2")
ggplot(norm_pdf) +
  geom_line(aes(x, y))  +
  ggtitle("PDF of a normal r.v. (mean=1, sd=4)")  + # for the main title
```

```
  xlab("support") + # for the x axis label
  ylab("pdf") # for the y axis label
```



We can also generate numbers from a normal distribution using the function rnorm( ) as follows

```
# numbers to extracts
n<-100
# generate n numbers from exponential with scale lambda
v<-rnorm(n, mean=1, sd=4)
```

and plot them using an histogram

```
hist(v,main="normal r.v. (mean=1, sd=4)", xlab="x")
```