Primo blocco di esercizi TAD 2018/19

Table of Contents

# Esercizio sulle determinazioni simulate

1. Per ciascuna delle seguenti distribuzioni eseguire un numero più basso ed uno più elevato di simulazioni e confrontare i seguenti elementi empirici e teorici: valori attesi, varianze, funzioni di ripartizioni e di densità (eventualmente anche QQ plot, .). Nelle simulazioni usare gli algoritmi descritti a lezione e le funzioni disponibili in R.

Distribuzione uniforme;

Distribuzione normale;

Distribuzione log-normale;

Distribuzione esponenziale e gamma;

Distribuzione di Poisson;

Distribuzione Poisson composta, con metodo diretto e metodi approssimati.

## 1. Numeri generati da una Unif(0,1)

### Per n=1000 determinazioni simulate

# Indici sintetici  
n=1000  
det1<-runif(n,0,1)  
mean(det1)

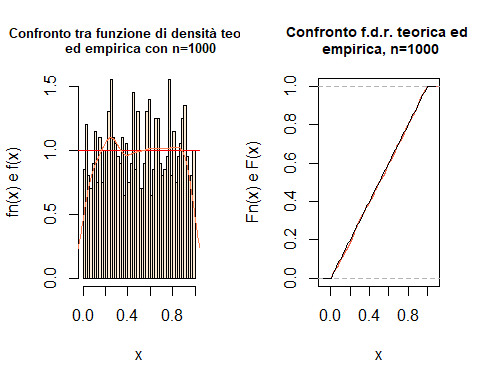
## [1] 0.5031767

var(det1)

## [1] 0.08185095

# I valori teorici sono rispettivamente 0.5 e (1-0)/12=0,083333

par(mfrow =c(1,2))  
# Funzione di densità   
hist(det1, freq=FALSE, main="Confronto tra funzione di densità teorica \n ed empirica con n=1000", ylim=c(0,1.5), breaks=50,col="antiquewhite",cex.main=0.8, xlab="x",ylab="fn(x) e f(x)")  
abline(1,0, col="RED")  
points(density(det1), xlim=c(min(det1), max(det1)), col="coral", type="l")  
# Funzione di ripartizione  
plot(ecdf(det1), col="coral2", main= "Confronto f.d.r. teorica ed \n empirica, n=1000", ylab="Fn(x) e F(x)", cex.main=0.9)  
curve(punif, col="black", add=TRUE)



### Per n=10000 determinazioni simulate

# Indici sintetici  
n=10000  
det2<-runif(n,0,1)  
mean(det2)

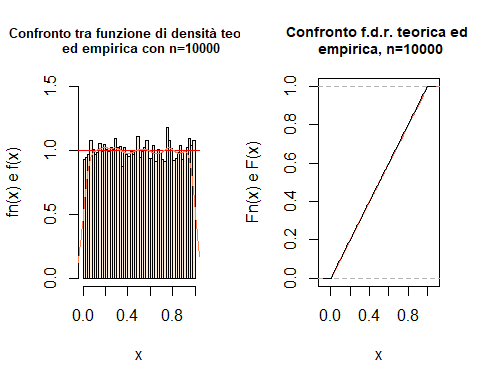
## [1] 0.5015728

var(det2)

## [1] 0.08377035

# I valori teorici sono rispettivamente 0.5 e (1-0)/12=0,083333

par(mfrow =c(1,2))  
# Funzione di densità   
hist(det2, freq=FALSE, main="Confronto tra funzione di densità teorica \n ed empirica con n=10000", ylim=c(0,1.5), breaks=50,col="antiquewhite",cex.main=0.8, xlab="x",ylab="fn(x) e f(x)")  
abline(1,0, col="RED")  
points(density(det2), xlim=c(min(det2), max(det2)), col="coral", type="l")  
# Funzione di ripartizione  
plot(ecdf(det2), col="coral2", main= "Confronto f.d.r. teorica ed \n empirica, n=10000", ylab="Fn(x) e F(x)", cex.main=0.9)  
curve(punif, col="black", add=TRUE)



### Confronto fra de1 e det2

mean(det1)

## [1] 0.5031767

mean(det2)

## [1] 0.5015728

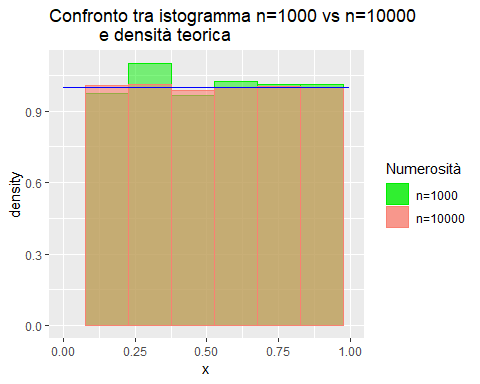
var(det1)

## [1] 0.08185095

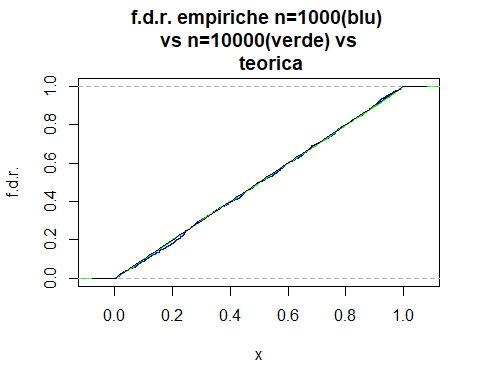
var(det2)

## [1] 0.08377035

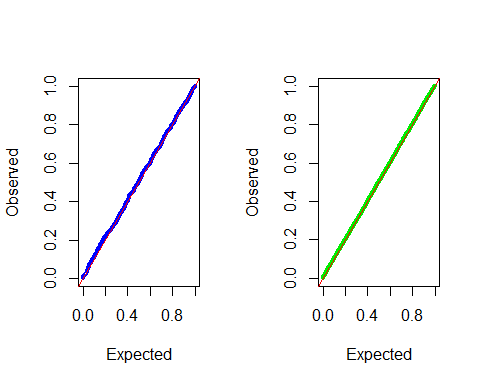
# Funzioni di densità  
data1 <- data.frame(det1)  
data2 <- data.frame(det2)  
ggplot(data.frame(x=c(-5,5)), aes(x)) +   
 geom\_histogram(aes(x=det1, y=..density.., fill="g", colour="g"), alpha=0.5, data=data1, binwidth = 0.15) +  
 geom\_histogram(aes(x=det2,y=..density.., fill="s", colour="s"), alpha=0.6, data=data2, binwidth = 0.15) +  
 scale\_colour\_manual(name="Numerosità", values=c("g" = "green2", "s"="salmon"), labels=c("s"="n=10000", "g"="n=1000")) +  
 scale\_fill\_manual(name="Numerosità", values=c("g" = "green2", "s"="salmon"),labels=c("s"="n=10000", "g"="n=1000")) +  
 xlim(c(min(det1),max(det1))) +  
 stat\_function(fun=dunif, color="blue")+   
ggtitle("Confronto tra istogramma n=1000 vs n=10000  
 e densità teorica")



# Funzioni di ripartizione  
plot(ecdf(det1), col="blue",   
main="f.d.r. empiriche n=1000(blu) \n vs n=10000(verde) vs   
 teorica", xlab="x", ylab="f.d.r.")  
plot(ecdf(det2), col="green2", add=T)  
curve(punif, col="black", add=TRUE)



# QQ-plot  
par(mfrow =c(1,2))  
qqunif(det1, type="unif",col="blue",cex = 0.5, logscale=FALSE, alpha=0.5)  
qqunif(det2, type="unif",cex = 0.5,col="green2",logscale=FALSE)



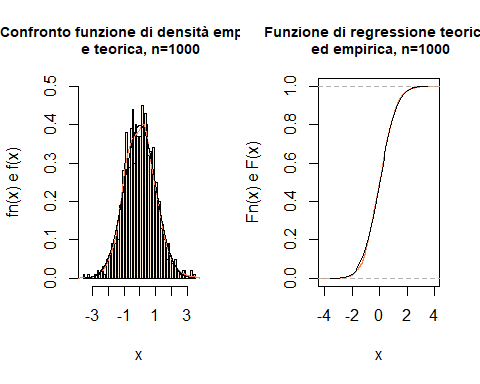
## 2. A) Normale standard con Box-Muller

### Per 1000 determinazioni simulate

## [1] -0.0002810408

## [1] 0.9627761

par(mfrow=c(1,2))  
# Confronto funzione di densità  
hist(det1BM, freq=FALSE, ylim=c(0,0.5), breaks=50, main="Confronto funzione di densità empirica \n e teorica, n=1000", xlab="x",ylab="fn(x) e f(x)", col="antiquewhite", cex.main=0.9)  
points(density(det1BM), col="coral", pch=3, type="l")  
curve(dnorm, xlim=c(min(det1BM), max(det1BM)), n=1000, col="black", pch=20, add=TRUE)  
  
# Confronto Funzione di regressione  
plot(ecdf(det1BM), main="Funzione di regressione teorica \n ed empirica, n=1000", ylab="Fn(x) e F(x)", col="coral", cex.main=0.9)  
curve(pnorm, xlim=c(min(det1BM),max(det1BM)), col="black", add=TRUE)



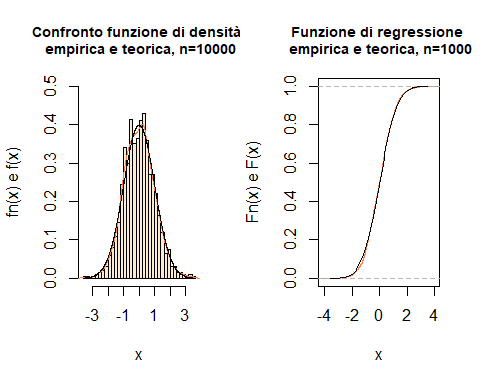
### Per 10000 determinazioni simulate

# Creazione dei numeri pseudo casuali   
detunif=runif(10000,min=0,max=1)  
det2BM=vector()  
for (i in seq(from=1, to=10000, by=2)){  
 det2BM[i]=sqrt(-2\*log(detunif[i]))\*cos(2\*pi\*detunif[i+1])  
 det2BM[i+1]=sqrt(-2\*log(detunif[i]))\*sin(2\*pi\*detunif[i+1])  
}  
# Media e Varianza empirica  
mean(det2BM)

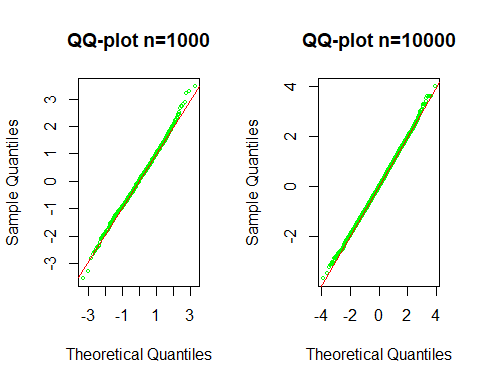
## [1] -0.01724034

# La media teorica è pari a 0 e la varianza è pari a 1

par(mfrow =c(1,2))  
# Funzione di densità   
hist(det1BM, freq=FALSE, ylim=c(0,0.5), breaks=30, main="Confronto funzione di densità \n empirica e teorica, n=10000", xlab="x",ylab="fn(x) e f(x)", col="antiquewhite", cex.main=0.9)  
points(density(det1BM), col="coral", pch=3, type="l")  
curve(dnorm, xlim=c(min(det1BM), max(det1BM)), n=1001, col="black", pch=20, add=TRUE)  
  
# Confronto Funzione di regressione  
plot(ecdf(det1BM), main="Funzione di regressione \n empirica e teorica, n=1000", ylab="Fn(x) e F(x)", col="coral", cex.main=0.9)  
curve(pnorm, xlim=c(min(det1BM),max(det1BM)), col="black", add=TRUE)



# QQ-plot cfr per n=1000 e n=10000  
par(mfrow=c(1,2))   
qqnorm(det1BM, main="QQ-plot n=1000",cex = 0.5, col="green")  
qqline(det1BM, col="red",cex = 0.5)  
  
qqnorm(det2BM,main="QQ-plot n=10000",cex = 0.5, col="green")  
qqline(det2BM, col="red",cex = 0.5)



## 2. B) Normale standard con rnorm

### Per 1000 determinazioni simulate

det1=rnorm(1000, mean=0, sd=1)  
# Media e Varianza empirica  
mean(det1)

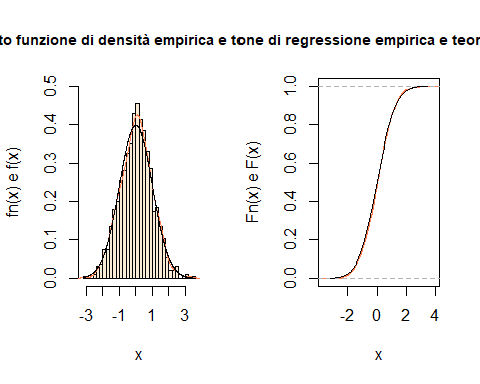
## [1] 0.03058485

var(det1)

## [1] 0.9318261

# Media teorica: 0  
# Varianza teorica: 1

par(mfrow=c(1,2))  
# Confronto funzione di densità  
hist(det1, freq=FALSE, ylim=c(0,0.5), breaks=30, main="Confronto funzione di densità empirica e teorica, n=1000", xlab="x",ylab="fn(x) e f(x)", col="antiquewhite", cex.main=0.9)  
points(density(det1), col="coral", pch=3, type="l")  
curve(dnorm, xlim=c(min(det1), max(det1)), n=1000, col="black", pch=20, add=TRUE)  
  
# Confronto Funzione di regressione  
plot(ecdf(det1), col="coral", main="Funzione di regressione empirica e teorica, n=1000", ylab="Fn(x) e F(x)", cex.main=0.9)  
curve(pnorm, xlim=c(min(det1),max(det1)), col="black", add=TRUE)



### Per 1000 determinazioni simulate

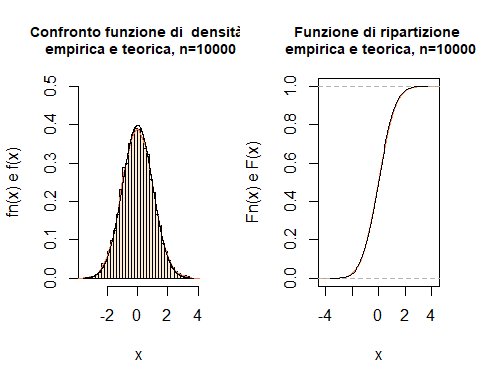
det2=rnorm(10000, mean=0, sd=1)  
  
# Media e Varianza empirica  
mean(det2)

## [1] -0.002888138

## [1] 0.00427139  
var(det2)

## [1] 1.036083

## [1] 0.990628  
# Media teorica: 0 e varianza teorica: 1  
  
par(mfrow=c(1,2))  
# Confronto funzione di densità  
hist(det2, freq=FALSE, ylim=c(0,0.5), breaks=30, main="Confronto funzione di densità \n empirica e teorica, n=10000", xlab="x",ylab="fn(x) e f(x)", col="antiquewhite", cex.main=0.9)  
points(density(det2), col="coral", pch=3, type="l")  
curve(dnorm, xlim=c(min(det2), max(det2)), n=1001, col="black", pch=20, add=TRUE)  
  
# Confronto Funzione di ripartizione  
plot(ecdf(det2), col="coral", main="Funzione di ripartizione \n empirica e teorica, n=10000", ylab="Fn(x) e F(x)", cex.main=0.9)  
curve(pnorm, xlim=c(min(det2),max(det2)), col="black", add=TRUE)



## Confronto tra det1BM, det2BM, det1 e det2

mean(det1)

## [1] 0.03058485

var(det1)

## [1] 0.9318261

mean(det2)

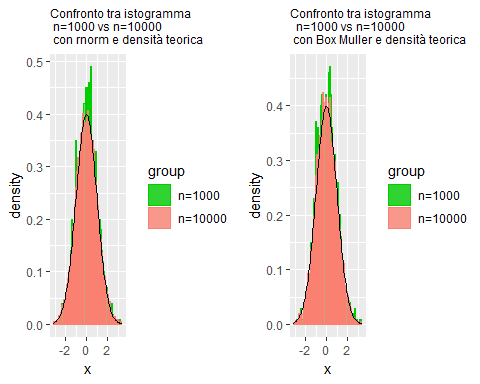
## [1] -0.002888138

var(det2)

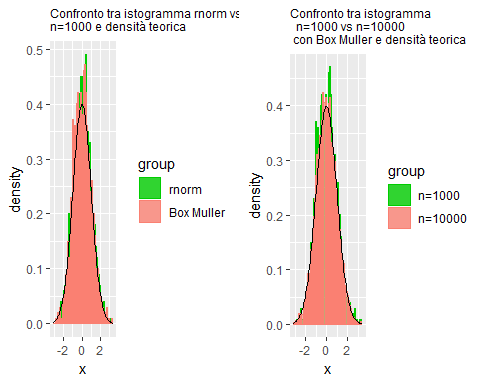
## [1] 1.036083

# Media teorica: 0 e varianza teorica: 1

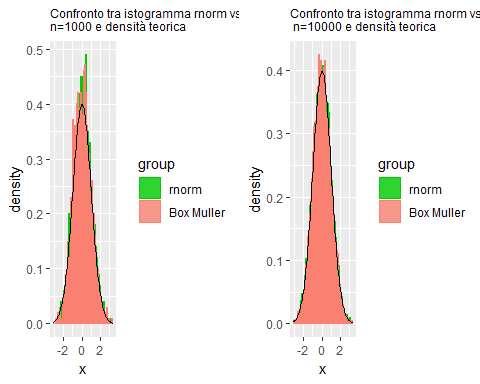
data1 <- data.frame(det1)  
data2 <- data.frame(det2)  
plot1=ggplot(data.frame(x=c(-5,5)),aes(x)) +   
 geom\_histogram(aes(x=det1, y=..density.., fill="g", colour="g"), alpha=0.5, data=data1,binwidth = 0.1) +  
 geom\_histogram(aes(x=det2,y=..density.., fill="s", colour="s"), alpha=0.6, data=data2, binwidth = 0.1) +  
 scale\_colour\_manual(name="group", values=c("g" = "green3", "s"="salmon"), labels=c("s"="n=10000", "g"="n=1000")) +  
 scale\_fill\_manual(name="group", values=c("g" = "green3", "s"="salmon"),labels=c("s"="n=10000", "g"="n=1000")) +  
 xlim(c(min(det1),max(det1))) +  
 stat\_function(fun = dnorm, color = "black")+  
 ggtitle("Confronto tra istogramma \n n=1000 vs n=10000 \n con rnorm e densità teorica")+  
 theme(plot.title = element\_text(size = 9))  
  
data1 <- data.frame(det1BM)  
data2 <- data.frame(det2BM)  
plot2=ggplot(data.frame(x=c(-5,5)),aes(x)) +   
 geom\_histogram(aes(x=det1BM, y=..density.., fill="g", colour="g"), alpha=0.5, data=data1,binwidth = 0.1) +  
 geom\_histogram(aes(x=det2BM,y=..density.., fill="s", colour="s"), alpha=0.6, data=data2,binwidth = 0.1) +  
 scale\_colour\_manual(name="group", values=c("g" = "green3", "s"="salmon"), labels=c("s"="n=10000", "g"="n=1000")) +  
 scale\_fill\_manual(name="group", values=c("g" = "green3", "s"="salmon"), labels=c("s"="n=10000", "g"="n=1000")) +   
 xlim(c(min(det1),max(det1))) +  
 stat\_function(fun = dnorm, color = "black")+  
 ggtitle("Confronto tra istogramma \n n=1000 vs n=10000 \n con Box Muller e densità teorica")+   
 theme(plot.title = element\_text(size = 9))  
ggarrange(plot1, plot2, ncol=2)



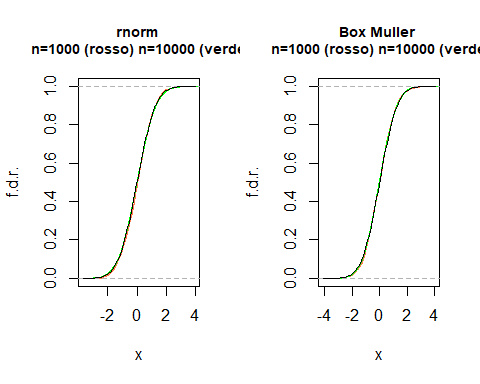
data1 <- data.frame(det1)  
data2 <- data.frame(det1BM)  
plot1=ggplot(data.frame(x=c(-5,5)),aes(x)) +   
 geom\_histogram(aes(x=det1, y=..density.., fill="g", colour="g"), alpha=0.5, data=data1,   
 binwidth = 0.1) +  
 geom\_histogram(aes(x=det1BM,y=..density.., fill="s", colour="s"), alpha=0.6, data=data2,   
 binwidth = 0.1) +  
 scale\_colour\_manual(name="group", values=c("g" = "green3", "s"="salmon"),   
 labels=c("s"="Box Muller", "g"="rnorm")) +  
 scale\_fill\_manual(name="group", values=c("g" = "green3", "s"="salmon"),   
 labels=c("s"="Box Muller", "g"="rnorm")) +  
 xlim(c(min(det1),max(det1))) +  
 stat\_function(fun = dnorm, color = "black")+  
 ggtitle("Confronto tra istogramma rnorm vs Box Muller, \nn=1000 e densità teorica")+  
 theme(plot.title = element\_text(size = 9))  
ggarrange(plot1, plot2, ncol=2)



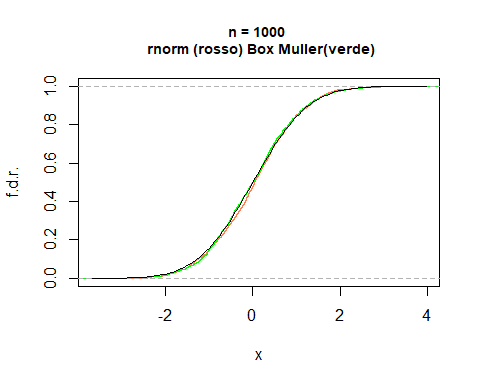
data1 <- data.frame(det2)  
data2 <- data.frame(det2BM)  
plot2=ggplot(data.frame(x=c(-5,5)),aes(x)) +   
 geom\_histogram(aes(x=det2, y=..density.., fill="g", colour="g"), alpha=0.5, data=data1,   
 binwidth = 0.1) +  
 geom\_histogram(aes(x=det2BM,y=..density.., fill="s", colour="s"), alpha=0.6, data=data2,   
 binwidth = 0.1) +  
 scale\_colour\_manual(name="group", values=c("g" = "green3", "s"="salmon"),   
 labels=c("s"="Box Muller", "g"="rnorm")) +  
 scale\_fill\_manual(name="group", values=c("g" = "green3", "s"="salmon"),   
 labels=c("s"="Box Muller", "g"="rnorm")) +  
 xlim(c(min(det1),max(det1))) +  
 stat\_function(fun = dnorm, color = "black")+  
 ggtitle("Confronto tra istogramma rnorm vs Box Muller, \n n=10000 e densità teorica")+  
 theme(plot.title = element\_text(size = 9))  
ggarrange(plot1, plot2, ncol=2)



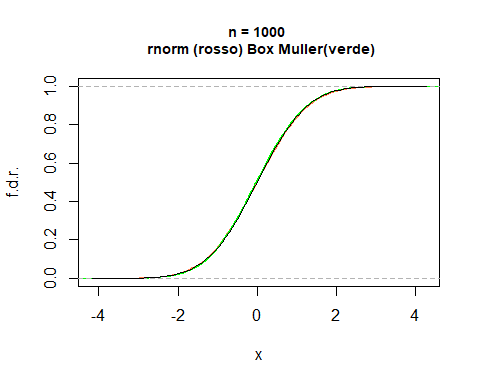
par(mfrow=c(1,2))  
# Confrontiamo ora le f.d.r. teoriche ed empiriche simulate con rnorm e con algoritmo Box Muller  
plot(ecdf(det1), col="coral", main="rnorm \n n=1000 (rosso) n=10000 (verde)",  
 ylab="f.d.r.", cex.main=0.9)  
plot(ecdf(det2), col="green", add=T)  
curve(pnorm, col="black", add=TRUE)  
  
plot(ecdf(det1BM), col="coral", main="Box Muller \n n=1000 (rosso) n=10000 (verde)",   
 ylab="f.d.r.", cex.main=0.9)  
plot(ecdf(det2BM), col="green", add=T)  
curve(pnorm, col="black", add=TRUE)



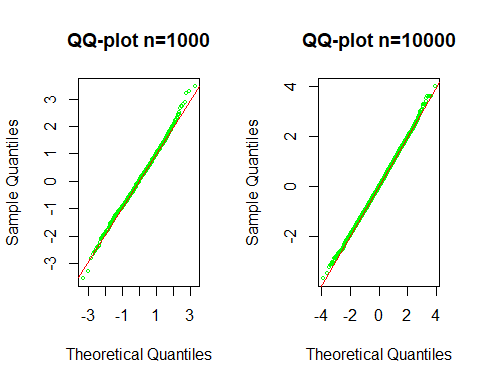
# Confrontiamo a parità di numero di determinazioni la f.d.r. generata con rnorm e Box Muller  
plot(ecdf(det1), col="coral", main="n = 1000 \n rnorm (rosso) Box Muller(verde)",   
 ylab="f.d.r.", cex.main=0.9)  
plot(ecdf(det1BM), col="green", add=T)  
curve(pnorm, col="black", add=TRUE)



plot(ecdf(det2), col="coral", main="n = 1000 \n rnorm (rosso) Box Muller(verde)",   
 ylab="f.d.r.", cex.main=0.9)  
plot(ecdf(det2BM), col="green", add=T)  
curve(pnorm, col="black", add=TRUE)



par(mfrow=c(1,2))   
qqnorm(det1BM, main="QQ-plot n=1000",cex = 0.5, col="green")  
qqline(det1BM, col="red",cex = 0.5)  
  
qqnorm(det2BM,main="QQ-plot n=10000",cex = 0.5, col="green")  
qqline(det2BM, col="red",cex = 0.5)



## 3. Numeri generati da una log-Normale

Premessa:

Il costo medio medio sinitri è di 4500 euro.

I momenti della log-normale sono:

da cui ricavo che

### Simulato con algoritmi di r

### 1000 determinazioni simulate

## Generazione dati  
det1=exp(rnorm(1000, mean=0, sd=1))  
## Si potrebbe usare direttamente il comando rlnorm  
  
##Indici sintetici  
mean(det1)

## [1] 1.570065

var(det1)

## [1] 4.402607

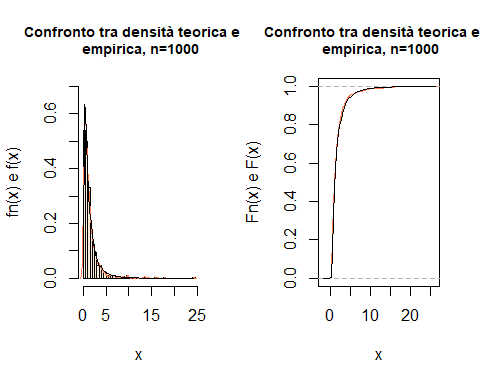
#I valori teorici sono rispettivamente   
exp(0+1/2)

## [1] 1.648721

exp(1)\*(exp(1)-1)

## [1] 4.670774

par(mfrow=c(1,2))  
## Funzione di densità  
hist(det1, freq=FALSE,ylim=c(0,0.7), breaks=50, col="antiquewhite", xlab="x",   
 ylab="fn(x) e f(x)", main = "Confronto tra densità teorica ed \n empirica, n=1000", cex.main=0.9)  
points(density(det1), xlim=c(min(det1), max(det1)), col="coral", type="l")  
curve(dlnorm, xlim=c(min(det1), max(det1)), col="black", pch=20, add=TRUE)  
  
## Funzione di ripartizione  
plot(ecdf(det1), col="coral", main="Confronto tra densità teorica ed \n empirica, n=1000", ylab="Fn(x) e F(x)", xlab="x", cex.main=0.9)  
curve(plnorm, col="black", add=TRUE)



### 10000 determinazioni simulate

## Generazione dati  
det2=exp(rnorm(10000, mean=0, sd=1))  
  
## Indici sintetici  
mean(det2)

## [1] 1.661668

var(det2)

## [1] 4.750256

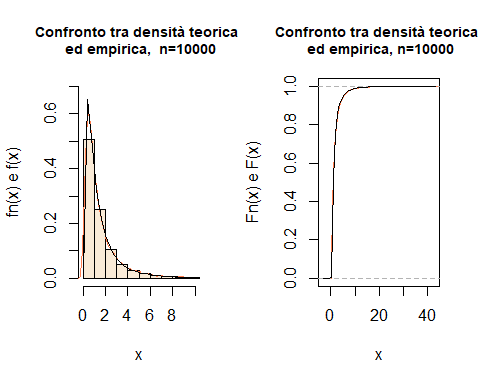
## Media e varianza empirica  
exp(0+1/2)

## [1] 1.648721

exp(1)\*(exp(1)-1)

## [1] 4.670774

par(mfrow=c(1,2))  
## Funzione di densità  
hist(det2, freq=FALSE,ylim=c(0,0.7), xlim=c(0,10), breaks=50, col="antiquewhite", xlab="x", ylab="fn(x) e f(x)", main = "Confronto tra densità teorica \n ed empirica, n=10000", cex.main=0.9)  
points(density(det2), xlim=c(min(det2), max(det2)), col="coral", type="l")  
curve(dlnorm, xlim=c(min(det2), max(det2)), col="black", pch=20, add=TRUE)  
  
## Funzione di ripartizione  
plot(ecdf(det2), col="coral", main="Confronto tra densità teorica \n ed empirica, n=10000", ylab="Fn(x) e F(x)", xlab="x", cex.main=0.9)  
curve(plnorm, col="black", add=TRUE)



### Confronto tra det1 e det2

mean(det1)

## [1] 1.570065

mean(det2)

## [1] 1.661668

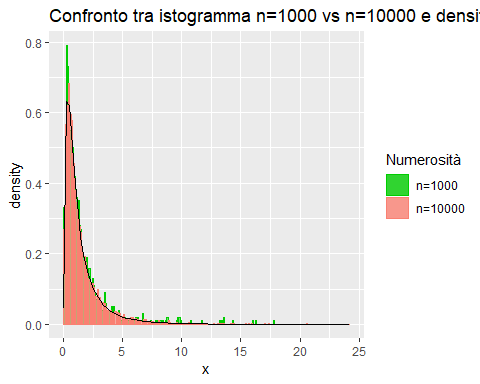
var(det1)

## [1] 4.402607

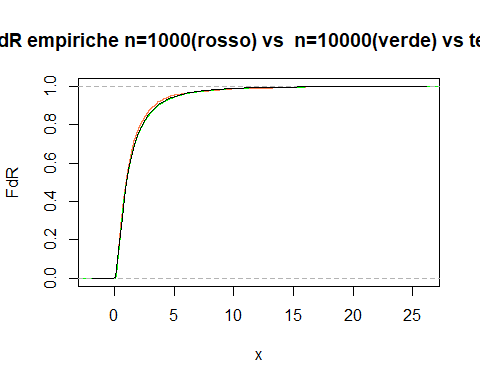
var(det2)

## [1] 4.750256

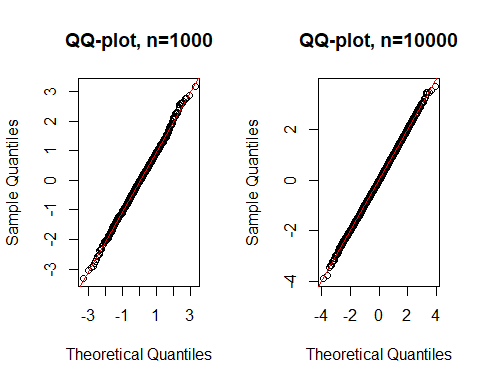
data1 <- data.frame(det1)  
data2 <- data.frame(det2)  
ggplot(data.frame(x=c(-5,5)),aes(x)) +   
 geom\_histogram(aes(x=det1, y=..density.., fill="g", colour="g"), alpha=0.5, data=data1, binwidth = 0.1) +  
 geom\_histogram(aes(x=det2,y=..density.., fill="s", colour="s"), alpha=0.6, data=data2, binwidth = 0.1) +  
 scale\_colour\_manual(name="Numerosità", values=c("g" = "green3", "s"="salmon"),  
 labels=c("s"="n=10000", "g"="n=1000")) +  
 scale\_fill\_manual(name="Numerosità", values=c("g" = "green3", "s"="salmon"),  
 labels=c("s"="n=10000", "g"="n=1000")) +  
 xlim(c(min(det1),max(det1))) +  
 stat\_function(fun = dlnorm, color = "black")+  
ggtitle("Confronto tra istogramma n=1000 vs n=10000 e densità teorica")



par(mfrow=c(1,1))  
plot(ecdf(det1), col="coral", main="FdR empiriche n=1000(rosso) vs n=10000(verde) vs teorica", xlab="x", ylab="FdR")  
plot(ecdf(det2), col="green", add=T)  
curve(plnorm, col="black", add=TRUE)



# QQ-plot (log-norm)  
par(mfrow =c(1,2))   
  
qqnorm(log(det1), main="QQ-plot, n=1000")  
qqline(log(det1), col= "red")  
  
qqnorm(log(det2), main="QQ-plot, n=10000")  
qqline(log(det2), col= "red")



### simulato con algoritmo (manca)

## 4. Numeri generati da una esponenziale (gamma)

Può essere usata per calcolare il numero atteso di sinistri.

### 1000 determinazioni simulate

## Ipotiziamo un parametro pari a 1  
det1=-log(runif(1000, 0, 1))  
## Si potrebbe usare direttamente il comando rexp.  
  
##Indici sintetici  
mean(det1)

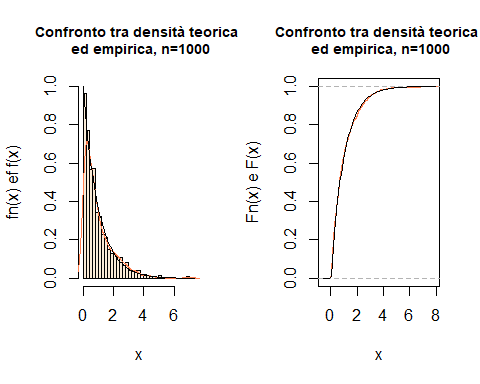
## [1] 0.9951924

var(det1)

## [1] 1.042555

## media teorica: 1 e varianza teorica: 1

par(mfrow=c(1,2))  
## Funzione di densità  
hist(det1, freq = FALSE, ylim=c(0,1), breaks= 30,col="antiquewhite", xlab="x",   
 ylab="fn(x) ef f(x)", main = "Confronto tra densità teorica \n ed empirica, n=1000", cex.main=0.9)  
points(density(det1), xlim=c(min(det1), max(det1)), col="coral", type="l")  
curve(dexp, xlim=c(min(det1), max(det1)), col="black", pch=20, add=TRUE)  
  
## Funzione di ripartizione  
plot(ecdf(det1), col="coral", main="Confronto tra densità teorica \n ed empirica, n=1000", cex.main=0.9, ylab="Fn(x) e F(x)", xlab="x")  
curve(pexp, col="black", add=TRUE)



### 10000 determinazioni simulate

# Generazione dati  
det2=-log(runif(10000, 0, 1))  
  
#Indici sintetici  
mean(det2)

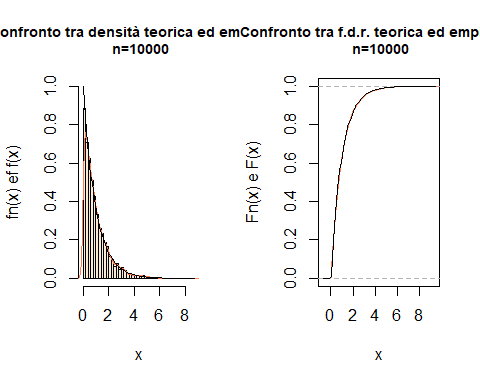
## [1] 1.011947

var(det2)

## [1] 1.0118

# media teorica: 1 e varianza teorica: 1

par(mfrow=c(1,2))  
## Funzione di densità  
hist(det2, freq = FALSE, ylim=c(0,1), breaks=50, col="antiquewhite", xlab="x", ylab="fn(x) ef f(x)", main = "Confronto tra densità teorica ed empirica, \n n=10000", cex.main=0.9)  
points(density(det2), xlim=c(min(det2), max(det2)), col="coral", type="l")  
curve(dexp, xlim=c(min(det2), max(det2)), col="black", pch=20, add=TRUE)  
  
## Funzione di ripartizione  
plot(ecdf(det2), col="coral", main="Confronto tra f.d.r. teorica ed empirica, \n n=10000", ylab="Fn(x) e F(x)", xlab="x", cex.main=0.9)  
curve(pexp, col="black", add=TRUE)



### Confronto tra det1 e det2

mean(det1)

## [1] 0.9951924

mean(det2)

## [1] 1.011947

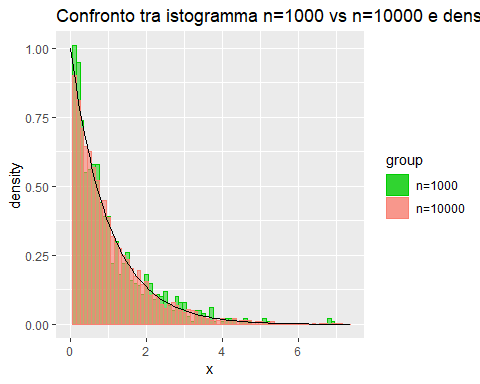
var(det1)

## [1] 1.042555

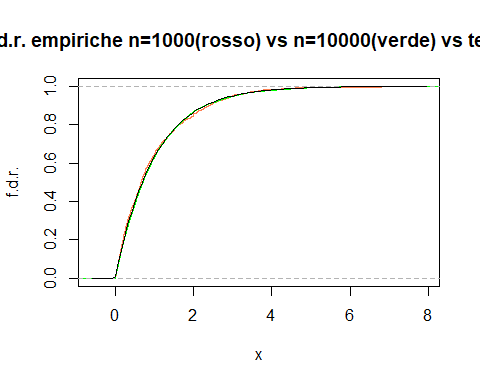
var(det2)

## [1] 1.0118

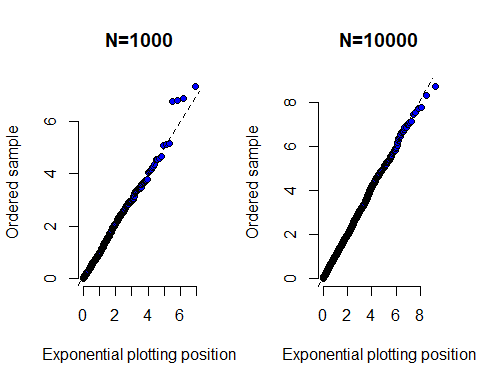
data1 <- data.frame(det1)  
data2 <- data.frame(det2)  
ggplot(data.frame(x=c(-5,5)),aes(x)) +  
 geom\_histogram(aes(x=det1, y=..density.., fill="g", colour="g"), alpha=0.5, data=data1,binwidth = 0.1) +  
 geom\_histogram(aes(x=det2,y=..density.., fill="s", colour="s"), alpha=0.6, data=data2, binwidth = 0.1) +  
 scale\_colour\_manual(name="group", values=c("g" = "green3", "s"="salmon"),  
 labels=c("s"="n=10000", "g"="n=1000")) +  
 scale\_fill\_manual(name="group", values=c("g" = "green3", "s"="salmon"),  
 labels=c("s"="n=10000", "g"="n=1000")) +  
 xlim(c(min(det1),max(det1))) +   
 stat\_function(fun = dexp, color = "black")+  
ggtitle("Confronto tra istogramma n=1000 vs n=10000 e densità teorica")



par(mfrow=c(1,1))  
plot(ecdf(det1), col="coral", main="f.d.r. empiriche n=1000(rosso) vs n=10000(verde) vs teorica", xlab="x", ylab="f.d.r.")  
plot(ecdf(det2), col="green", add=T)  
curve(pexp, col="black", add=TRUE)



par(mfrow =c(1,2))  
qqexp(det1,pch=21,bg="blue",bty="n", line=TRUE, main="N=1000")  
qqexp(det2,pch=21,bg="blue",bty="n", line=TRUE, main="N=10000")



## 5. Numeri generati da una erlangiana

### 1000 Determinazioni simulate

det1=rgamma(1000, 4.7,5.9)  
alfa= 4.7  
beta=5.9  
  
##Inidici sintetici  
mean(det1)

## [1] 0.7879152

var(det1)

## [1] 0.1511281

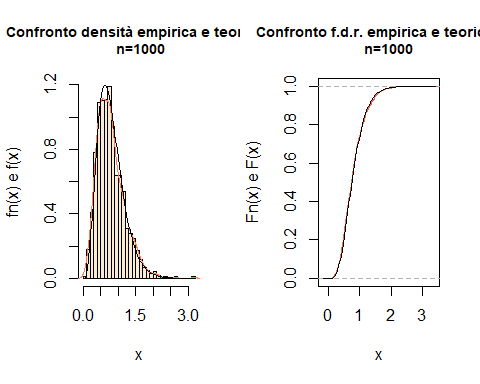
## Media e varianza teorica:  
alfa/beta

## [1] 0.7966102

alfa/beta^2

## [1] 0.1350187

par(mfrow=c(1,2))  
## Funzione di densità  
hist(det1, freq=FALSE, breaks=30,col="antiquewhite",   
main="Confronto densità empirica e teorica, \n n=1000", xlab="x", ylab="fn(x) e f(x)",  
cex.main=0.9)  
points(density(det1), xlim=c(min(det1), max(det1)), col="coral", type="l")  
curve(dgamma(x,4.7,5.9), xlim=c(min(det1), max(det1)), col="black", pch=20, add=TRUE)  
  
#Funzione di ripartizione  
plot(ecdf(det1), col="coral2", main="Confronto f.d.r. empirica e teorica,   
 n=1000",xlab="x", ylab="Fn(x) e F(x)", cex.main=0.9)  
curve(pgamma(x,alfa,beta), col="black", add=TRUE)



### 10000 Determinazioni simulate

det2=rgamma(10000, 4.7,5.9)  
alfa= 4.7  
beta=5.9  
  
## Indici sintetici  
mean(det2)

## [1] 0.79243

var(det2)

## [1] 0.1318591

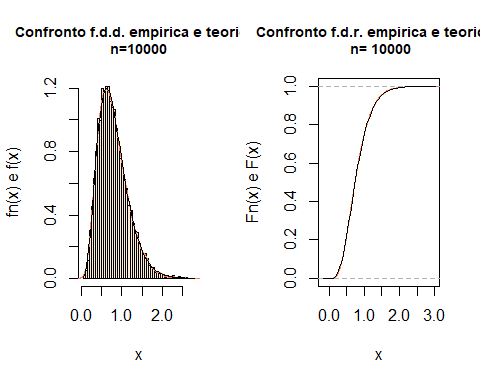
## Media e varianza teorica:   
alfa/beta

## [1] 0.7966102

alfa/beta^2

## [1] 0.1350187

par(mfrow=c(1,2))  
# Funzione di densità  
hist(det2, freq=FALSE,breaks=40, col="antiquewhite",   
main="Confronto f.d.d. empirica e teorica, \nn=10000", xlab="x", ylab="fn(x) e f(x)",  
cex.main=0.9)  
points(density(det2), xlim=c(min(det2), max(det2)), col="coral", type="l")  
curve(dgamma(x,4.7,5.9), xlim=c(min(det2), max(det2)), col="black", pch=20, add=TRUE)  
  
#Funzione di ripartizione  
plot(ecdf(det2), col="coral", main="Confronto f.d.r. empirica e teorica, \n n= 10000",  
 xlab="x", ylab="Fn(x) e F(x)", cex.main=0.9)  
curve(pgamma(x,alfa,beta), col="black", add=TRUE)



### Confronta tra det1 e det2

mean(det1)

## [1] 0.7879152

mean(det2)

## [1] 0.79243

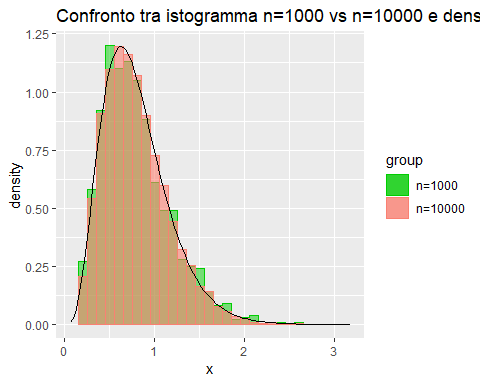
var(det1)

## [1] 0.1511281

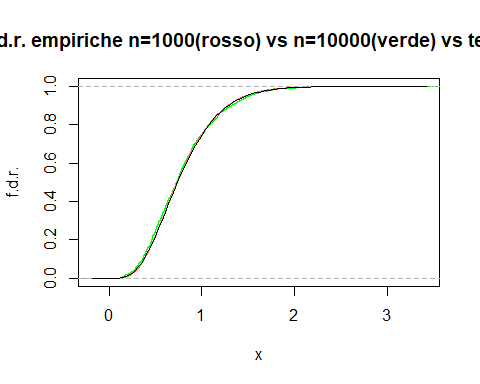
var(det2)

## [1] 0.1318591

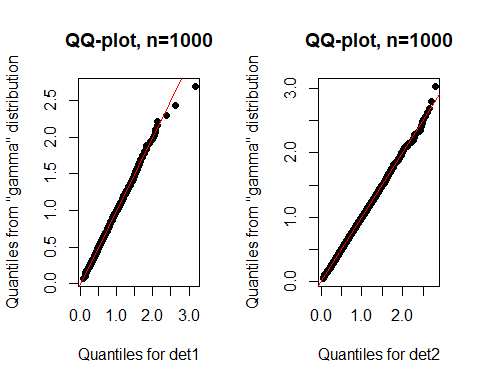
data1 <- data.frame(det1)  
data2 <- data.frame(det2)  
ggplot(data.frame(x=c(-5,5)),aes(x)) +   
 geom\_histogram(aes(x=det1, y=..density.., fill="g", colour="g"), alpha=0.5, data=data1, binwidth = 0.1) +  
 geom\_histogram(aes(x=det2,y=..density.., fill="s", colour="s"), alpha=0.6, data=data2, binwidth = 0.1) +  
 scale\_colour\_manual(name="group", values=c("g" = "green3", "s"="salmon"), labels=c("s"="n=10000", "g"="n=1000")) +  
 scale\_fill\_manual(name="group", values=c("g" = "green3", "s"="salmon"), labels=c("s"="n=10000", "g"="n=1000")) +   
 xlim(c(min(det1),max(det1))) +  
 stat\_function(fun = dgamma,args=list(alfa,beta), color = "black")+  
ggtitle("Confronto tra istogramma n=1000 vs n=10000 e densità teorica")



par(mfrow=c(1,1))  
plot(ecdf(det1), col="green", main="f.d.r. empiriche n=1000(rosso) vs n=10000(verde) vs teorica", xlab="x", ylab="f.d.r.")  
plot(ecdf(det2), col="coral2", add=T)  
curve(pgamma(x,alfa,beta), col="black", add=TRUE, lwd=0.8)



# Q-Q plot  
par(mfrow =c(1,2))  
  
qqPlot(det1, "gamma", plot.type = "Q-Q", confbounds=FALSE, main="QQ-plot, n=1000")  
qqPlot(det2, "gamma", plot.type = "Q-Q", confbounds=FALSE, main="QQ-plot, n=1000")



## 6. Numeri generati da una poisson

### A) Simulato con algoritmo derivante dalle proprietà del processo di poisson

### 1000 Determinazioni simulate

n=1000  
lambda=2.74  
det1PS=vector(mode="numeric")  
  
for (i in 1:n){  
 m=-1  
 a = 1  
 repeat {  
 m =m +1  
 b=a  
 u=runif(1)  
 a=a\*u  
 if (a<exp(-lambda) & exp(-lambda)<= b){  
 detvect=m  
 break  
 }  
 }  
 det1PS[i]=detvect  
}  
  
#Indici sintetici  
mean(det1PS)

## [1] 2.743

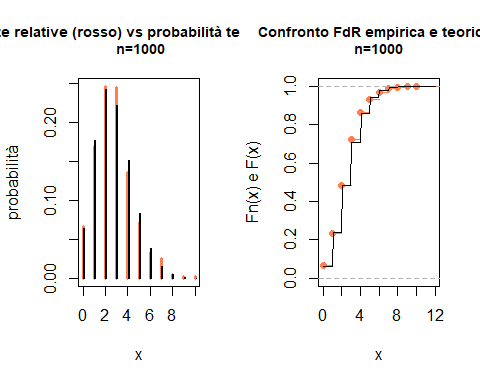
var(det1PS)

## [1] 2.873825

# Media e varianza teorica  
lambda

## [1] 2.74

par(mfrow=c(1,2))  
  
## Funzione di densità  
tab=table(det1PS)  
tabo=as.data.frame(tab)  
punti = as.numeric(levels(tabo$det1PS))  
freq = as.numeric(tab/n)  
plot(punti, freq, type="h", col="coral", ylab="probabilità", xlab="x",xlim=c(0,max(det1PS)), ylim=c(0,max(freq)), lwd=3,main="frequeze relative (rosso) vs probabilità teorica (nero), \n n=1000", cex.main=0.9)  
points(punti+0.1 ,dpois(punti,lambda),type="h",col="black", lwd=2)  
  
#Funzione di ripartizione  
plot(ecdf(det1PS),xlim=c(0,max(det1PS)+2), col="coral", cex.main=0.9, main="Confronto FdR empirica e teorica, \nn=1000",xlab="x", ylab="Fn(x) e F(x)")  
curve(ppois(x, lambda = lambda), col="black", add=TRUE)



### 10000 Determinazioni simulate

n=1000  
lambda=2.74  
det2PS=vector(mode="numeric")  
  
for (i in 1:n){  
 m=-1  
 a = 1  
 repeat {  
 m =m +1  
 b=a  
 u=runif(1)  
 a=a\*u  
 if (a<exp(-lambda) & exp(-lambda)<= b){  
 detvect=m  
 break  
 }  
 }  
 det2PS[i]=detvect  
}  
  
#Indici sintetici  
mean(det2PS)

## [1] 2.779

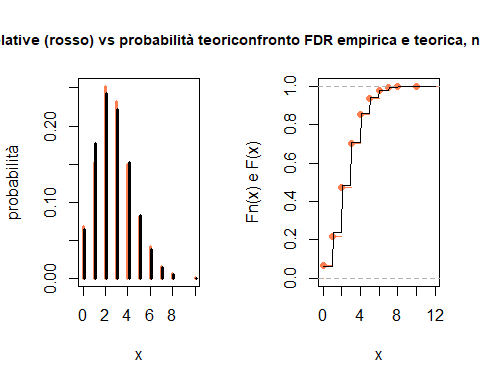
var(det2PS)

## [1] 2.732892

# Media e varianza teorica  
lambda

## [1] 2.74

par(mfrow=c(1,2))  
  
# Funzione di densità  
tab=table(det2PS)  
tabo=as.data.frame(tab)  
punti = as.numeric(levels(tabo$det2PS))  
freq = as.numeric(tab/n)  
plot(punti, freq, type="h", col="coral", ylab="probabilità", xlab="x",xlim=c(0,max(det2PS)), ylim=c(0,max(freq)), lwd=3,main="frequeze relative (rosso) vs probabilità teorica (nero), n=10000", cex.main=0.9)  
points(punti+0.1,dpois(punti,lambda),type="h",col="black", lwd=3)  
  
#Funzione di ripartizione  
plot(ecdf(det2PS),xlim=c(0,max(det2PS)+2), col="coral", cex.main=0.9, main="Confronto FDR empirica e teorica, n=10000",xlab="x", ylab="Fn(x) e F(x)")  
curve(ppois(x, lambda = lambda), col="black", add=TRUE)



### B) Simulato con algoritmi di r

n=1000  
det1=rpois(n, lambda)  
  
#Indici sintetici  
mean(det1)

## [1] 2.806

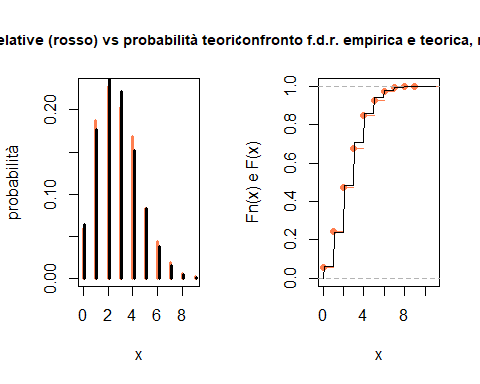
var(det1)

## [1] 2.895259

## Media e Varianza teorcia:  
lambda

## [1] 2.74

par(mfrow=c(1,2))  
# Funzione di densità  
tab=table(det1)  
tabo=as.data.frame(tab)  
punti = as.numeric(levels(tabo$det1))  
freq = as.numeric(tab/n)  
plot(punti, freq, type="h", col="coral", ylab="probabilità", xlab="x",xlim=c(0,max(det1)), ylim=c(0,max(freq)), lwd=3,main="frequeze relative (rosso) vs probabilità teorica (nero), n=1000", cex.main=0.9)  
points(punti+.1,dpois(punti,lambda),type="h",col="black", lwd=3)  
  
#Funzione di ripartizione  
plot(ecdf(det1),xlim=c(0,max(det1)+2), col="coral", cex.main=0.9,  
 main="Confronto f.d.r. empirica e teorica, n=1000",xlab="x", ylab="Fn(x) e F(x)")  
curve(ppois(x, lambda = lambda), col="black", add=TRUE)



### 10000 determinazioni simulate

n=10000  
det2=rpois(n, lambda)  
  
#Inidici sintetici  
mean(det2)

## [1] 2.7355

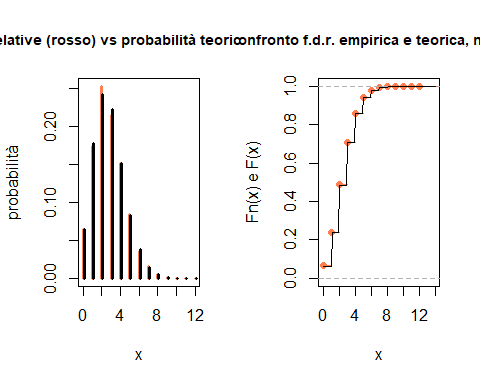
var(det2)

## [1] 2.741814

# Media e varianza teorica  
lambda

## [1] 2.74

par(mfrow=c(1,2))  
  
# Funzione di densità  
tab=table(det2)  
tabo=as.data.frame(tab)  
punti = as.numeric(levels(tabo$det2))  
freq = as.numeric(tab/n)  
plot(punti, freq, type="h", col="coral", ylab="probabilità", xlab="x",xlim=c(0,max(det2)), ylim=c(0,max(freq)), lwd=3,main="Frequeze relative (rosso) vs probabilità teorica (nero), n=10000", cex.main=0.9)  
points(punti+ 0.1,dpois(punti,lambda),type="h",col="black", lwd=3)  
  
#Funzione di ripartizione  
plot(ecdf(det2),xlim=c(0,max(det2)+2), col="coral", cex.main=0.9, main="Confronto f.d.r. empirica e teorica, n=10000",xlab="x", ylab="Fn(x) e F(x)")  
curve(ppois(x, lambda = lambda), col="black", add=TRUE)



### Confronta tra det1PS, det2PS, det1 e det2

mean(det1)

## [1] 2.806

mean(det2)

## [1] 2.7355

mean(det1PS)

## [1] 2.743

mean(det2PS)

## [1] 2.779

var(det1)

## [1] 2.895259

var(det2)

## [1] 2.741814

var(det1PS)

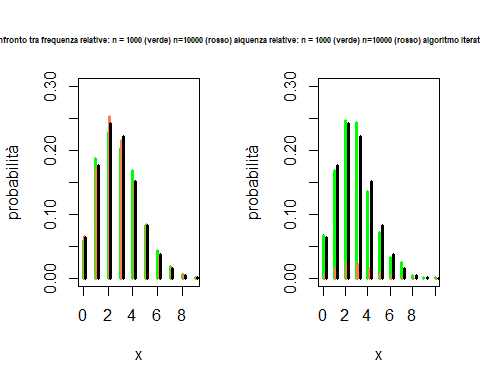
## [1] 2.873825

var(det2PS)

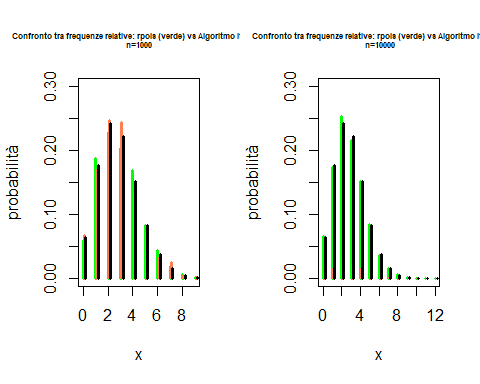
## [1] 2.732892

tab1=table(det1)  
tabo1=as.data.frame(tab1)  
punti1 = as.numeric(levels(tabo1$det1))  
freq1 = as.numeric(tab1/1000)  
  
tab2=table(det2)  
tabo2=as.data.frame(tab2)  
punti2 = as.numeric(levels(tabo2$det2))  
freq2 = as.numeric(tab2/10000)  
  
tab1PS=table(det1PS)  
tabo1PS=as.data.frame(tab1PS)  
punti1PS = as.numeric(levels(tabo1PS$det1PS))  
freq1PS = as.numeric(tab1PS/1000)  
  
tab2PS=table(det2PS)  
tabo2PS=as.data.frame(tab2PS)  
punti2PS = as.numeric(levels(tabo2PS$det2PS))  
freq2PS = as.numeric(tab2PS/10000)

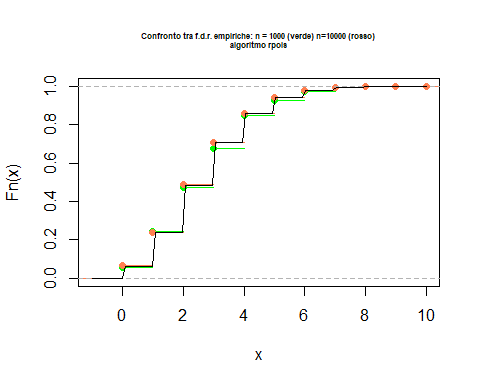
# Cfr Funzioni di probabilità  
par(mfrow=c(1,2))  
plot(punti1, freq1, type="h", col="green", ylab="probabilità", xlab="x",xlim=c(0,max(det1)), ylim=c(0,0.3), lwd=3, main="Confronto tra frequenza relative: n = 1000 (verde) n=10000 (rosso) algoritmo rpois", cex.main=0.5)  
points(punti2+.1, freq2, type="h", col="coral", lwd=3)  
points(punti1+.2,dpois(punti1,lambda),type="h",col="black", lwd=3)  
  
plot(punti1PS, freq1PS, type="h", col="green", ylab="probabilità", xlab="x", xlim=c(0,max(det1PS)), ylim=c(0,0.3), lwd=3, main="Confronto tra frequenza relative: n = 1000 (verde) n=10000 (rosso) algoritmo iterativo del processo stocastico", cex.main=0.5)  
points(punti2PS+ 0.1, freq2PS, type="h", col="coral", lwd=3)  
points(punti1PS+ 0.3,dpois(punti1PS,lambda),type="h",col="black", lwd=3)



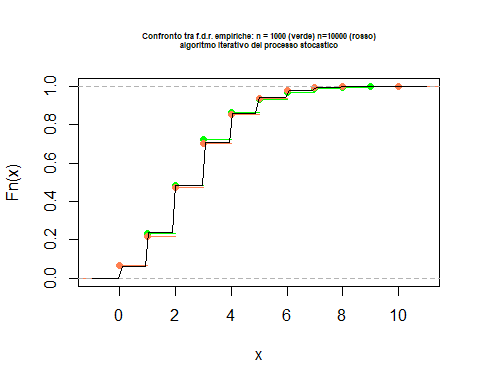
plot(punti1, freq1, type="h", col="green", ylab="probabilità", xlab="x",xlim=c(0,max(det1)), ylim=c(0,0.3), lwd=3, main="Confronto tra frequenze relative: rpois (verde) vs Algoritmo itertivo \nn=1000", cex.main=0.5)  
points(punti1PS+.1, freq1PS, type="h", col="coral", lwd=3)  
points(punti1PS+.2,dpois(punti1PS,lambda),type="h",col="black", lwd=3)  
  
plot(punti2, freq2, type="h", col="green", ylab="probabilità", xlab="x",xlim=c(0,max(det2)), ylim=c(0,0.3), lwd=3, main="Confronto tra frequenze relative: rpois (verde) vs Algoritmo itertivo \n n=10000", cex.main=0.5)  
points(punti2PS+.1, freq2PS, type="h", col="coral", lwd=3)  
points(punti2+.2,dpois(punti2,lambda),type="h",col="black", lwd=3)



# Cfr Funzioni di ripartizione  
plot(ecdf(det1), col="green",  
main="Confronto tra f.d.r. empiriche: n = 1000 (verde) n=10000 (rosso) \nalgoritmo rpois",  
cex.main=0.5)  
plot(ecdf(det2), col="coral", add=T)  
curve(ppois(x,lambda), col="black", add=TRUE)



plot(ecdf(det1PS), col="green",  
main="Confronto tra f.d.r. empiriche: n = 1000 (verde) n=10000 (rosso)  
algoritmo iterativo del processo stocastico", cex.main=0.5)  
plot(ecdf(det2PS), col="coral", add=T)  
curve(ppois(x,lambda), col="black", add=TRUE)



## 7. Disitribuzione Binomiale- Negativa

### Per n=10000 determinazioni simulate

# Indici sintetici  
n=1000  
alpha = 7.29   
p=0.91  
alpha/p

## [1] 8.010989

det1=rnbinom(n,alpha ,p)  
#Indici sintetici  
mean(det1)

## [1] 0.739

var(det1)

## [1] 0.8096887

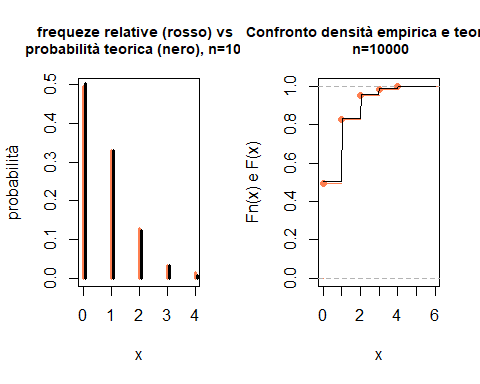
## Media e varianza teorica:  
alpha\*(1-p)/p #(vd details)

## [1] 0.720989

alpha\*(1-p)/p^2

## [1] 0.7922956

par(mfrow=c(1,2))  
  
## Funzione di densità  
tab=table(det1)  
tabo=as.data.frame(tab)  
punti = as.numeric(levels(tabo$det1))  
freq = as.numeric(tab/n)  
plot(punti, freq, type="h", col="coral", ylab="probabilità", xlab="x",xlim=c(0,max(det1)), ylim=c(0,max(freq)), lwd=3,main="frequeze relative (rosso) vs \n probabilità teorica (nero), n=1000", cex.main=0.9)  
points(punti+0.1,dnbinom(punti,alpha, p),type="h",col="black", lwd=3)  
  
#Funzione di ripartizione  
plot(ecdf(det1),xlim=c(0,max(det1)+2), col="coral", cex.main=0.9, main="Confronto densità empirica e teorica, \n n=10000",xlab="x", ylab="Fn(x) e F(x)")  
curve(pnbinom(x, alpha, p), col="black", add=TRUE)



### Per n=10000 determinazioni simulate

# Indici sintetici  
n=10000  
alpha = 7.29   
p=0.91  
alpha/p

## [1] 8.010989

det2=rnbinom(n,alpha ,p)  
#Inidici sintetici  
mean(det2)

## [1] 0.7245

var(det2)

## [1] 0.792679

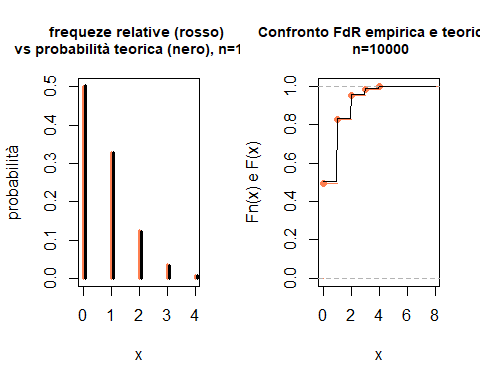
## Media e varianza teorica:  
alpha\*(1-p)/p #(vd details)

## [1] 0.720989

alpha\*(1-p)/p^2

## [1] 0.7922956

par(mfrow=c(1,2))  
  
## Funzione di densità  
tab=table(det2)  
tabo=as.data.frame(tab)  
punti = as.numeric(levels(tabo$det2))  
freq = as.numeric(tab/n)  
plot(punti, freq, type="h", col="coral", ylab="probabilità", xlab="x",xlim=c(0,max(det1)), ylim=c(0,max(freq)), lwd=3,main="frequeze relative (rosso) \n vs probabilità teorica (nero), n=1000", cex.main=0.9)  
points(punti+0.1,dnbinom(punti,alpha, p),type="h",col="black", lwd=3)  
  
#Funzione di ripartizione  
plot(ecdf(det1),xlim=c(0,max(det2)+2), col="coral", cex.main=0.9, main="Confronto FdR empirica e teorica, \n n=10000",xlab="x", ylab="Fn(x) e F(x)")  
curve(pnbinom(x, alpha, p), col="black", add=TRUE)



### Confronto fra det1 e det2

mean(det1)

## [1] 0.739

mean(det1)

## [1] 0.739

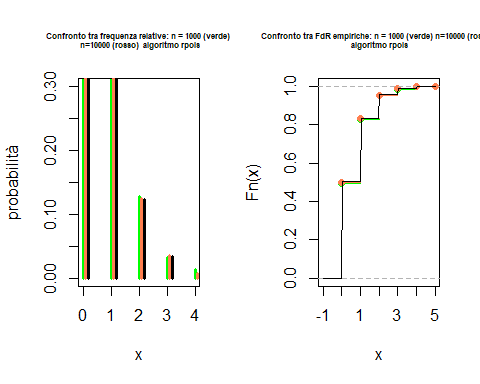
var(det1)

## [1] 0.8096887

var(det2)

## [1] 0.792679

tab1=table(det1)  
tabo1=as.data.frame(tab1)  
punti1 = as.numeric(levels(tabo1$det1))  
freq1 = as.numeric(tab1/1000)  
  
tab2=table(det2)  
tabo2=as.data.frame(tab2)  
punti2 = as.numeric(levels(tabo2$det2))  
freq2 = as.numeric(tab2/10000)  
  
# Confronto fra funzioni di densità  
par(mfrow=c(1,2))  
plot(punti1, freq1, type="h", col="green", ylab="probabilità", xlab="x",xlim=c(0,max(det1)), ylim=c(0,0.3), lwd=3, main="Confronto tra frequenza relative: n = 1000 (verde) \n n=10000 (rosso) algoritmo rpois", cex.main=0.5)  
points(punti2+.1, freq2, type="h", col="coral", lwd=3)  
points(punti1+.2,dnbinom(punti1,alpha, p),type="h",col="black", lwd=3)  
  
# Confronto fra funzioni di ripartizione  
plot(ecdf(det1), col="green", main="Confronto tra FdR empiriche: n = 1000 (verde) n=10000 (rosso) \n algoritmo rpois",  
cex.main=0.5)  
plot(ecdf(det2), col="coral", add=T)  
curve(pnbinom(x,alpha, p), col="black", add=TRUE)



## 8. Numeri generati da una poisson-composta

### 1000 determinazioni simulate con Wilson-Hilferty

n=1000  
alfa= 2000  
beta= 0.5  
lambda = 200  
mu = lambda\*alfa/beta  
m2= alfa\*(alfa+1)/(beta^2)  
m3 = alfa\*(alfa+1)\*(alfa+2)/(beta^3)  
sigma=sqrt(lambda\*m2)  
simm=m3/sqrt(lambda\*(m2^3))  
C1=simm/6-6/simm  
C2= 3\*((2/simm)^(2/3))  
C3=2/simm  
  
hinv= function(x){  
 result = ((x-C1)^3)/(C2^3)-C3  
 return (result)  
}

detnorm=rnorm(n, 0, 1)  
det1WH=vector(mode="numeric")  
det1WH = sigma\*hinv(detnorm)+mu  
  
## Indici sintetici  
mean(det1WH)

## [1] 801600

var(det1WH)

## [1] 3554529582

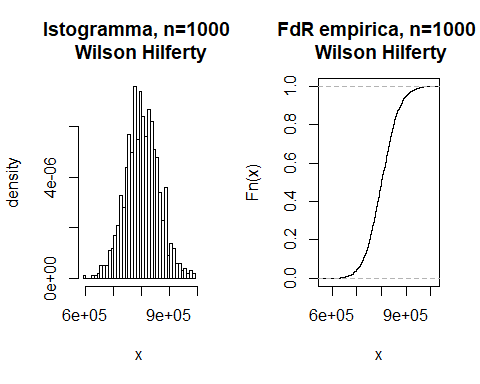
## Media e varianza teorica  
mu

## [1] 8e+05

sigma^2

## [1] 3201600000

par(mfrow=c(1,2))  
  
## Istogramma  
hist(det1WH, main="Istogramma, n=1000 \n Wilson Hilferty", xlab="x", ylab="density", prob=T, breaks=30)  
  
## Funzione di ripartizione empirica  
plot(ecdf(det1WH), main="FdR empirica, n=1000 \n Wilson Hilferty", xlab="x", ylab="Fn(x)")



### 10000 determinazioni simulate con Wilson Hilferty

n=10000  
alfa= 2000  
beta= 0.5  
lambda = 200  
mu = lambda\*alfa/beta  
m2= alfa\*(alfa+1)/(beta^2)  
m3 = alfa\*(alfa+1)\*(alfa+2)/(beta^3)  
sigma=sqrt(lambda\*m2)  
simm=m3/sqrt(lambda\*(m2^3))  
C1=simm/6-6/simm  
C2= 3\*((2/simm)^(2/3))  
C3=2/simm  
  
hinv= function(x){  
 result = ((x-C1)^3)/(C2^3)-C3  
 return (result)  
}  
  
detnorm=rnorm(n, 0, 1)  
det2WH=vector(mode="numeric")  
det2WH = sigma\*hinv(detnorm)+mu  
  
## Indici sintetici  
mean(det2WH)

## [1] 799448.6

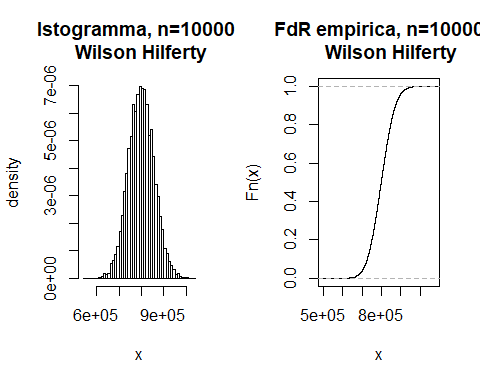
var(det2WH)

## [1] 3261745827

## Media e varianza teorica  
mu

## [1] 8e+05

par(mfrow=c(1,2))  
  
## Istogramma  
hist(det2WH, main="Istogramma, n=10000 \n Wilson Hilferty", xlab="x", ylab="density", prob=T, breaks=40)  
  
## Funzione di ripartizione empirica  
plot(ecdf(det2WH), main="FdR empirica, n=10000  
 Wilson Hilferty", xlab="x", ylab="Fn(x)")



### 1000 determinazioni simulate con Normal Power

n=10000  
alfa= 2000  
beta= 0.5  
lambda = 200  
mu = lambda\*alfa/beta  
m2= alfa\*(alfa+1)/(beta^2)  
m3 = alfa\*(alfa+1)\*(alfa+2)/(beta^3)  
sigma=sqrt(lambda\*m2)  
  
simm=m3/sqrt(lambda\*(m2^3))  
detnorm=rnorm(n, 0, 1)  
det1NP=vector(mode="numeric")  
det1NP = mu + sigma\*detnorm +(sigma\*simm)\*(detnorm^2-1)/6  
  
# Indici sintetici  
mean(det1NP)

## [1] 799038.4

var(det1NP)

## [1] 3308636227

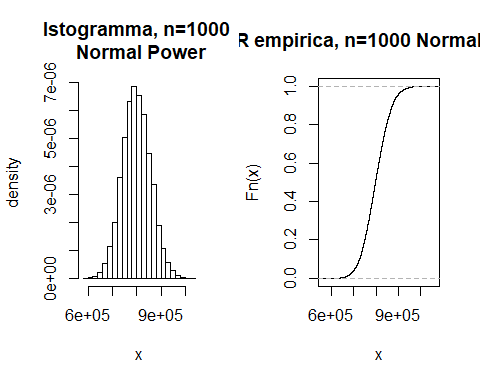
# Media e varianza teorica  
mu

## [1] 8e+05

sigma^2

## [1] 3201600000

par(mfrow=c(1,2))  
  
# Istogramma  
hist(det1NP, main="Istogramma, n=1000 \n Normal Power", xlab="x", ylab="density", prob=T, breaks=30)  
# Funzione di ripartizione empirica  
plot(ecdf(det1NP), main="FdR empirica, n=1000 Normal Power", xlab="x", ylab="Fn(x)")



### 10000 determinazioni simulate con Normal Power

n=10000  
alfa= 2000  
beta=0.5  
lambda = 200  
  
mu = lambda\*alfa/beta  
m2= alfa\*(alfa+1)/(beta^2)  
m3 = alfa\*(alfa+1)\*(alfa+2)/(beta^3)  
sigma=sqrt(lambda\*m2)  
simm=m3/sqrt(lambda\*(m2^3))  
detnorm=rnorm(n, 0, 1)  
det2NP=vector(mode="numeric")  
det2NP = mu + sigma\*detnorm +(sigma\*simm)\*(detnorm^2-1)/6  
  
## Indici sintetici  
mean(det2NP)

## [1] 800358.9

var(det2NP)

## [1] 3161186226

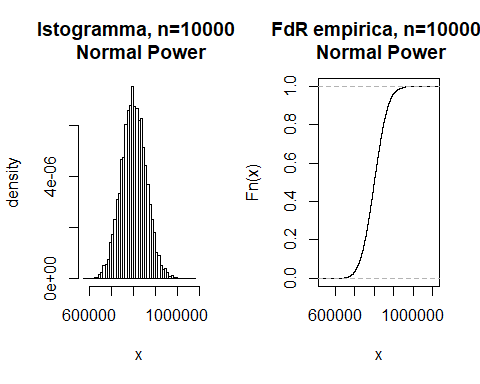
## Media e varianza teorica  
mu

## [1] 8e+05

sigma^2

## [1] 3201600000

par(mfrow=c(1,2))  
  
## Istogramma  
hist(det2NP, main="Istogramma, n=10000 \n Normal Power", xlab="x", ylab="density", prob=T, breaks=40)  
## Funzione di ripartizione empirica  
plot(ecdf(det2NP), main="FdR empirica, n=10000 \n Normal Power", xlab="x", ylab="Fn(x)")



### 1000 determinazioni simulate con procedimento diretto

n=1000  
alfa=2000  
beta=0.5  
lambda=200  
det1=vector()  
npois=vector()  
for (i in 1:n){  
 u = runif(1)  
 npois[i]=qpois(u,lambda)  
 if (npois[i]==0)  
 det1[i]=0  
 else{  
 v=runif(npois[i])  
 y=qgamma(v,alfa,beta)  
 det1[i]=sum(y)  
 }  
 }  
  
## Indici sintetici  
mean(det1)

## [1] 798113.3

var(det1)

## [1] 3185235873

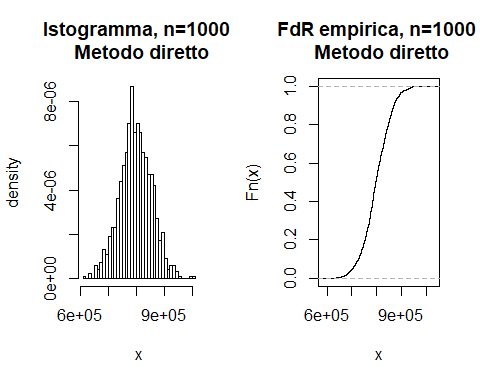
## Media e varianza teorica  
mu

## [1] 8e+05

sigma^2

## [1] 3201600000

par(mfrow=c(1,2))  
## Istogramma  
hist(det1, main="Istogramma, n=1000 \n Metodo diretto", xlab="x", ylab="density", prob=T, breaks=30)  
  
## Funzione di ripartizione empirica  
plot(ecdf(det1), main="FdR empirica, n=1000 \n Metodo diretto", xlab="x", ylab="Fn(x)")



### 10000 determinazioni simulate con procedimeto diretto

n=10000  
alfa=2000  
beta=0.5  
lambda=200  
det2=vector()  
npois=vector()  
  
for (i in 1:n){  
 u=runif(1)  
 npois[i]=qpois(u,lambda)  
 if (npois[i]==0)  
 det2[i]=0  
 else{  
 v=runif(npois[i])  
 y=qgamma(v,alfa,beta)  
 det2[i]=sum(y)  
 }  
}  
  
## Indici sintetici  
mean(det1)

## [1] 798113.3

var(det1)

## [1] 3185235873

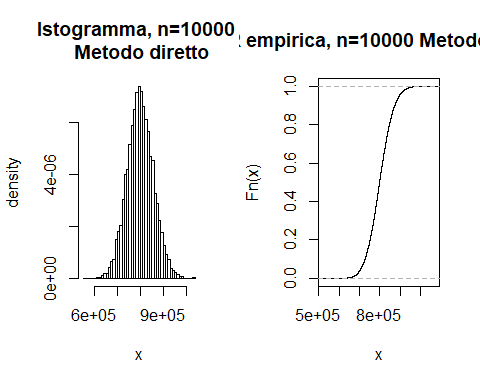
## Media e varianza teorica  
mu

## [1] 8e+05

sigma^2

## [1] 3201600000

par(mfrow=c(1,2))  
## Istogramma  
hist(det2, main="Istogramma, n=10000 \n Metodo diretto", xlab="x", ylab="density", prob=T, breaks=40)  
  
## Funzione di ripartizione empirica  
plot(ecdf(det2), main="FdR empirica, n=10000 Metodo diretto", xlab="x", ylab="Fn(x)")



### Confronti tra det1, det2, det1WH, det2WH, det1NP e det2NP

# Medie empiriche  
mean(det1)

## [1] 798113.3

mean(det2)

## [1] 799926.7

mean(det1WH)

## [1] 801600

mean(det2WH)

## [1] 799448.6

mean(det1NP)

## [1] 799038.4

mean(det2NP)

## [1] 800358.9

# Speranza matematica teorica:  
mu

## [1] 8e+05

# Varianza empirica  
var(det1)

## [1] 3185235873

var(det2)

## [1] 3249606112

var(det1WH)

## [1] 3554529582

var(det2WH)

## [1] 3261745827

var(det1NP)

## [1] 3308636227

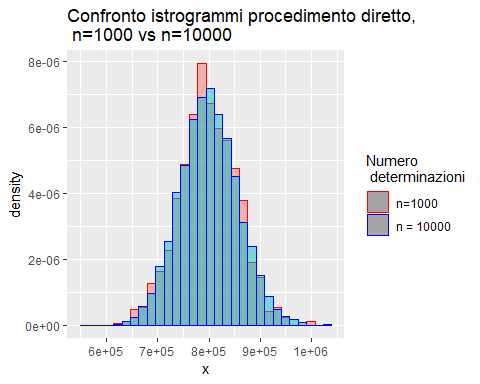
var(det2NP)

## [1] 3161186226

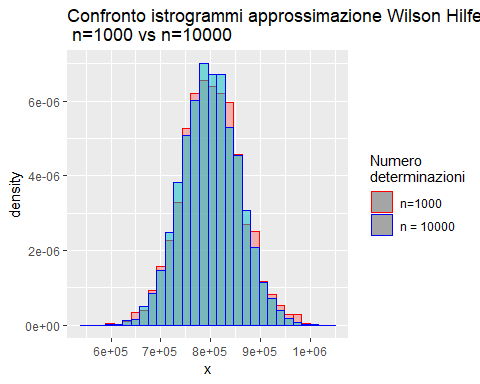
# Varianza teorica  
sigma^2

## [1] 3201600000

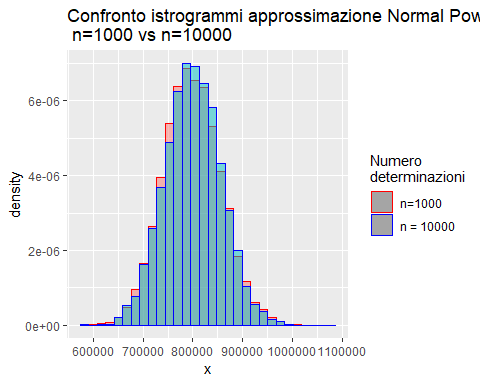
# Confronto istogrammi con determinazioni simulate mediante metodo diretto n=1000 vs n=10000  
data1=data.frame(x=det1)  
data2=data.frame(x=det2)  
data1$veg = 'x1'  
data2$veg = 'y1'  
pippo= rbind(data1,data2)  
ggplot(pippo,aes(x, color=veg)) +   
 geom\_histogram(alpha=0.5,aes(y=..density.., fill=veg), position = 'identity') + guides(fill=F)+ scale\_colour\_manual(name="Numero \n determinazioni",values=c("red","blue"), labels=c("x1"="n=1000", "y1"="n = 10000"))+   
 ggtitle("Confronto istrogrammi procedimento diretto, \n n=1000 vs n=10000")



# Confronto istogrammi con determinazioni simulate mediante approssimazione Wilson Hilferty, n=1000 vs n=10000  
data1WH=data.frame(x=det1WH)  
data2WH=data.frame(x=det2WH)  
data1WH$veg = 'x1'  
data2WH$veg = 'y1'  
ciao= rbind(data1WH,data2WH)  
ggplot(ciao,aes(x, color=veg)) +   
 geom\_histogram(alpha=0.5,aes(y=..density.., fill=veg), position = 'identity') +  
 guides(fill=F)+   
 scale\_colour\_manual(name="Numero\ndeterminazioni",values=c("red","blue"), labels=c("x1"="n=1000", "y1"="n = 10000"))+   
 ggtitle("Confronto istrogrammi approssimazione Wilson Hilferty, \n n=1000 vs n=10000")



### Confronto istogrammi con determinazioni simulate mediante approssimazione Wilson Hilferty, n=1000 vs n=10000  
data1NP=data.frame(x=det1NP)  
data2NP=data.frame(x=det2NP)  
data1NP$veg = 'x1'  
data2NP$veg = 'y1'  
ciao= rbind(data1NP,data2NP)  
ggplot(ciao,aes(x, color=veg)) +   
 geom\_histogram(alpha=0.5,aes(y=..density.., fill=veg), position = 'identity') +   
 guides(fill=F)+  
 scale\_colour\_manual(name="Numero\ndeterminazioni",values=c("red","blue"), labels=c("x1"="n=1000", "y1"="n = 10000"))+  
 ggtitle("Confronto istrogrammi approssimazione Normal Power, \n n=1000 vs n=10000")



# Esercizio sulla probabilità di rovina

Consegna: Valutare, mediante simulazione, la probabilità di rovina in un orizzonte limitato, nel modello della teoria collettiva del rischio (TCR).

probrov = function(R, tau, lambda, alfa, beta, teta, n){  
 mu=alfa/beta  
 c=(1+teta)\*lambda\*mu  
 det=vector()  
 for (i in 1:n){  
 ta=0  
 d=0  
 det[i]=0  
 while (!det[i]==1) {  
 u=runif(1)  
 w=-(1/lambda)\*log(u)  
 tb=ta+w  
  
 if (tb>tau)  
 {break}  
 else {  
 u=runif(1)  
 y=qgamma(u, alfa, beta)  
 d=d+c\*(tb-ta)-y  
 if (d<(-1)\*R)  
 det[i]=1  
 else  
 ta=tb  
 }  
 }  
 }  
 rovina=sum(det)/n  
 return(rovina)  
}  
  
par(mfrow=c(1,2))  
## Fissiamo tutti i valori e vediamo come varia la probabilità di rovina in funzione del capitale R  
varR=vector()  
r=vector()  
for (i in 1:50){  
 r[i]=500+1000\*i  
 varR[i]=probrov(r[i],20,20,2000,0.5,0.03, 500)  
}  
plot(r, varR, main="Andamento della probabilità di rovina \n in funzione del capitale R", ylab="Probabilità di Rovina", cex.main=0.9)  
## Fissiamo tutti i valori e vediamo come varia la probabilità di rovina in funzione del coefficiente di caricamento di sicurezza teta  
m=mean(r)  
varTeta=vector()  
Teta=vector()  
for (i in 1:50){  
 Teta[i]= 0.02\*i  
 varTeta[i]=probrov(m,20,20,2000,0.5, Teta[i], 500)  
}  
plot(Teta, varTeta, main="Andamento della probabilità di rovina in \n funzione del coefficiente di caricamento teta", ylab="Probabilità di Rovina", cex.main=0.9)

