Finanças Quantitativas: Lista 1

Lucas Machado Moschen

March 3, 2020

Problema 1.1

Assuma que $F_1(x)$ *e* $F_2(x)$ *são cdfs que satisfazem* $F_1(x) \leq F_2(x)$ *para todo* x.

1. *Qual das duas distribuições tem cauda inferior mais pesada? Explique.*

Resposta: Suponha que ambas as distribuições se extendem para $-\infty$. Se $X_i \sim F_i(x), i = 1, 2$, sabemos que $P[X_2 \leq \pi_p^{(2)}] = p = P[X_1 \leq \pi_p^{(1)}] \leq P[X_2 \leq \pi_p^{(1)}] \implies \pi_p^{(1)} \geq \pi_p^{(2)}$, onde $\pi_p^{(i)}$ é função quantil das distribuições $F_i(x), i = 1, 2$. Desta maneira, o quantil da primeira distribuição vai mais devagar para $-\infty$ e, portanto, a segunda curva tem cauda inferior mais pesada.

2. *Qual das duas distribuições tem cauda superior mais pesada? Explique.*

Resposta: Suponha que ambas as distribuições se extendem para ∞. Pelo mesmo argumento utilizado na primeira pergunta, podemos observar que quando $p \to 1$, o quantil da primeira curva cresce mais rapidamente e, consequentemente, a primeira curva terá causa superior mais pesada.

3. *Se estas duas distribuições são propostas para modelar o retorno de um dado portifolio no próximo mês e se você é perguntado para computar* $VaR_{0.01}$ *para este portifolio neste período, qual dessas duas distribuições resultaria o maior "value at risk".*

Resposta: Observe que se $P_{t+1}^i - P_t^i, i = 1, 2$ indica o retorno modeladado pela distribuição 1 e 2, respectivamente. Assim

$P[P_{t+1}^2 - P_t^2 < -VaR_p^2] = p = P[P_{t+1}^1 - P_t^1 < -VaR_p^1] \leq P[P_{t+1}^2 - P_t^2 < -VaR_p^1]$

Desta forma, como já observamos, $-VaR_p^2 \leq -VaR_p^1 \implies VaR_p^2 \geq VaR_p^1$ e obtemos que se calcularmos VaR_p utilizando a segunda distribuição, ele terá um resultado maior.

Problema 1.2

1. *Em R, gere uma* $N = 1024$ *amostras da distribuição exponencial com taxa* $r = 0.2$ *. Chame* X *o vetor com as amostras.* **Resposta:**

```
# Gerando variável aleatória X:
U = runif(n = 1024, min = 0, max = 1)
# Usamos a universalidade da Uniforme e a distribuição F da exponencial
r = 0.2
```

```r
F <- function(x, r){
  return <-  1 - exp(-r*x)
}
invF <- function(x,r){
  return <- -1/r*log(1 - x)
}
X = invF(U,r)
print(X[0:10])
```

```
##  [1]  0.6374925  1.0299589  3.3463657 19.3593488 25.4603870  1.7264858
##  [7]  7.0846877  0.4913075  9.2456724  6.2128635
```

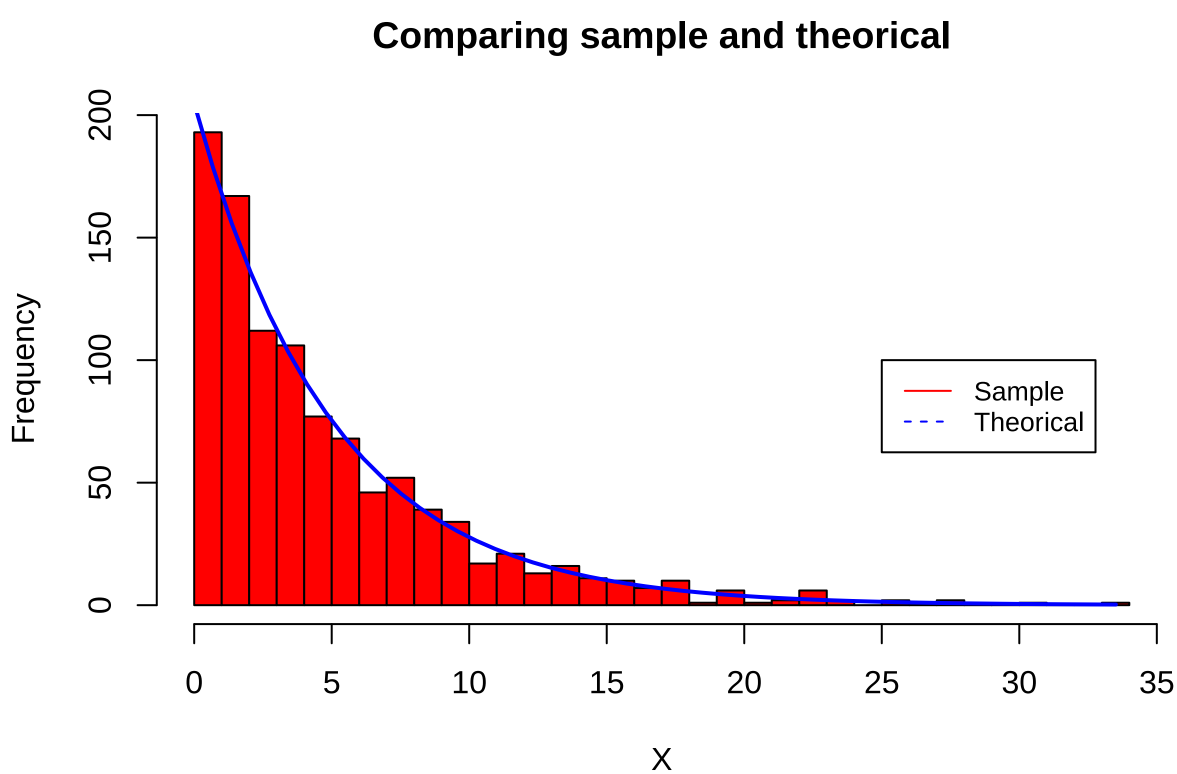
2. *Plote no mesmo gráfico a densidade exata de X e o histograma de X.*

Resposta:

```r
# Calculando o histograma
h <- hist(X, breaks = 30, col = 'red', main = 'Comparing sample and theorical')

xfit <- seq(0, max(X),length= 50)
# Calculando a distribuição exponencial
yfit <- dexp(xfit,rate = 0.2 )
yfit <- yfit*diff(h$mids[1:2])*length(X)
lines(xfit, yfit, col="blue", lwd = 2)

legend(25, 100, legend=c("Sample", "Theorical"),
       col=c("red", "blue"), lty=1:2, cex=0.8)
```

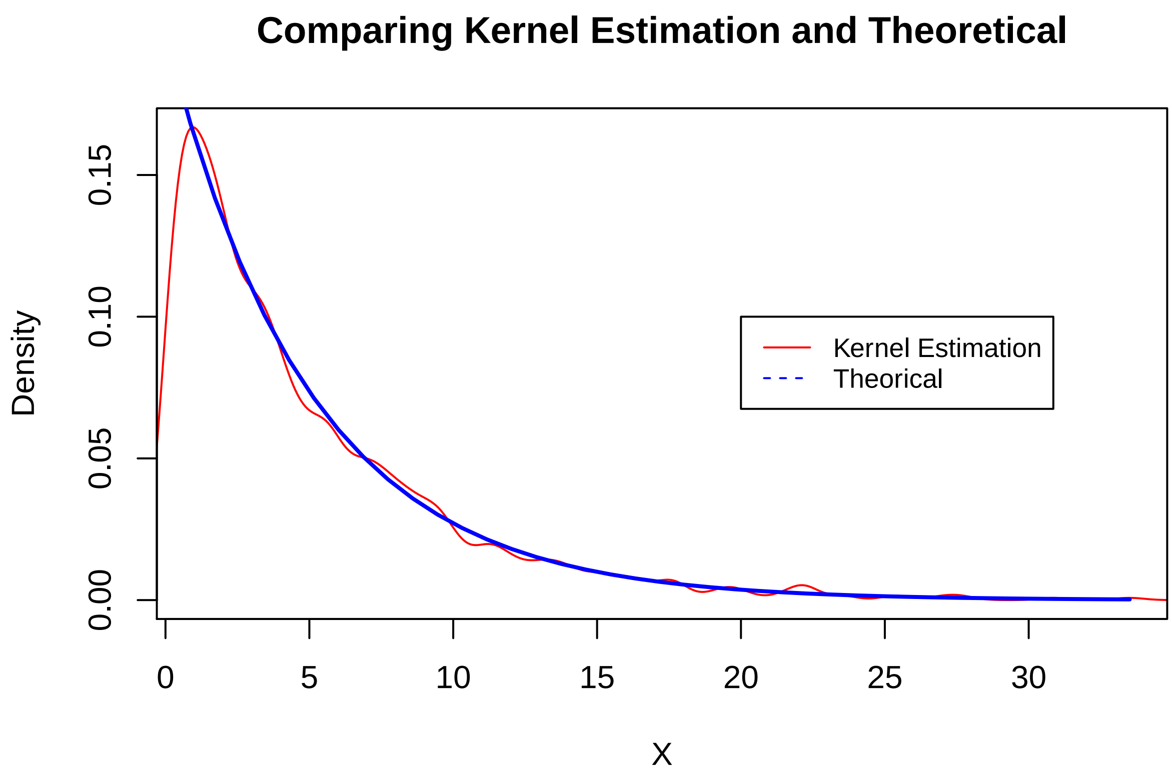


3. *Plote no mesmo gráfico a densidade exata de X e uma estimativa da densidade kernel de X.*

```
xfit <- seq(0, max(X),length= 40)

# A distribuição exponencial
yfit <- dexp(xfit,rate = 0.2)
# A estimação da densidade kernel
yden <- density(X, bw = 0.5)
#Plotes
plot(yden$x, yden$y, col='red', xlim = c(1,max(X)), type = 'l', main = "Comparing Kernel Estima
     xlab = "X", ylab = "Density")
lines(xfit, yfit, col="blue", lwd = 2)

legend(20, 0.10, legend=c("Kernel Estimation", "Theorical"),
       col=c("red", "blue"), lty=1:2, cex=0.8)
```



4. *Compare os dois plotes e explique as razões das diferenças. Diga qual estimativa você prefere, e explique porquê.*

Resposta: Observe que tanto o histograma quanto a estimação kernel da densidade se utilizam dos N dados. O histograma pode ser escrito em forma de uma função kernel, só que uma descontínua, já que tem um valor caso x, o valor da função, e x_i, uma amostra, estão no mesmo bin, e 0 caso não estejam. No caso da estimação kernel, essa diferença é suave, bastando escolher uma função suave. Os saltos da aproximação no caso do histograma é determinado pelo número de bins, enquanto no caso da estimação kernel é o parâmetro "bandwidth", logo podemos ter uma função tão suave quanto queremos. Desta maneira, é preferível essa estimação kernel, pois suavizamos a função e podemos obter resultados mais precisos com a densidade almejada.

Problem 1.3

```r
#again, we can input our R code here.
```