CIMAT

Centro de Investigación en Matemáticas, A.C.
Inferencia Estadística

Ayudantías de Modelos Estadísticos

José Miguel Saavedra Aguilar

---

1. Introducción a Inferencia Estadística

1.1. Ejemplo 1

Para una v.a. $X \sim \exp$, graficamos la función de densidad para distintas tasas $\lambda = 5, 2, 0.5$.

```
# Creamos un vector de 100 puntos en el soporte.
# En nuestro caso, tomamos de 0 a 10.
x <- seq(0, 10, length = 100)
```

```
# y es la función de densidad elegida evaluada en los puntos de x
# Para obtener la función de densidad, se toma "d"+nombre de la distribución,
# por ejemplo dnorm o dexp
y1 <- dexp(x, rate = 5)
y2 <- dexp(x, rate = 0.5)
y3 <- dexp(x, rate = 2)
```

Ahora, graficamos las respectivas funciones de distribución de X.

```
# f es la función de distribución asociada evaluada en los puntos de x
# Para obtener la función de densidad, se toma "p"+nombre de la distribución,
# por ejemplo pnorm o pexp.
f1 <- pexp(x, rate = 5)
f2 <- pexp(x, rate = 0.5)
f3 <- pexp(x, rate = 2)
```

Para $\lambda = 0.5$, simulamos una muestra aleatoria de tamaño $n = 100$ datos de X.

```
# Tomamos n=100 y simulamos n datos pseudoaleatorios de la distribución elegida
# Para obtener los datos aleatorios, se toma "r"+nombre de la distribución,
# por ejemplo rnorm o rexp
n <- 100
z <- rexp(n, rate = 0.5)
```

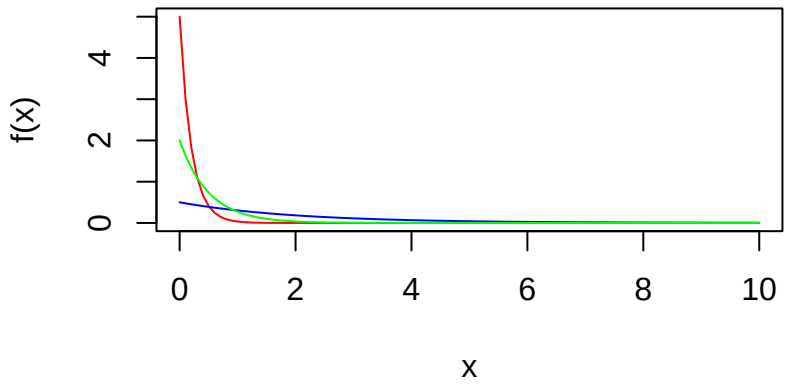

Figura 1: Densidad de una v.a. exponencial para distintas medias

Ordenamos los puntos simulados en $x_{(1)}, x_{(2)}, \ldots, x_{(n)}$. Les asociamos k_i definido por

$$k_i = \frac{i}{n+1} \tag{1}$$

```
#Ahora vamos a ordenar los datos como indica la tarea.
z2 <- sort(z)
# Inicializamos k =i/n+1
k <- numeric()

for (i in 1:n) {
  k[i] <- i / (n + 1)
}
```

Para conocer más sobre la función de distribución empírica, pueden consultar Wikipedia.

2. Ejemplos de Knitr

Se ejemplifica el uso de knitr. Es recomendable consultar el libro de Yihui Xie [2] para mayor información.

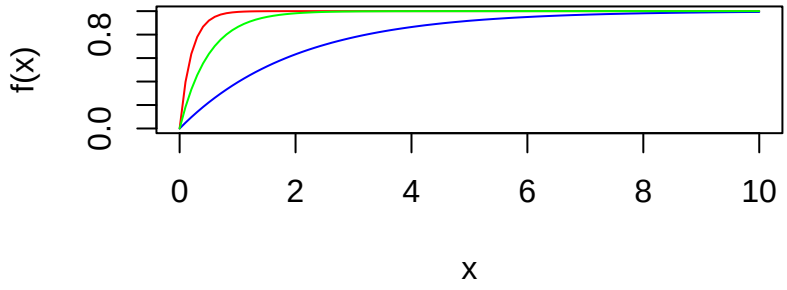

Figura 2: Distribución exponencial para distintas medias

2.1. Ejemplo 2

Una variable aleatoria discreta X tiene función de masa de probabilidad:

x	0	1	2	3	4
$p(x)$	0.1	0.2	0.2	0.2	0.3

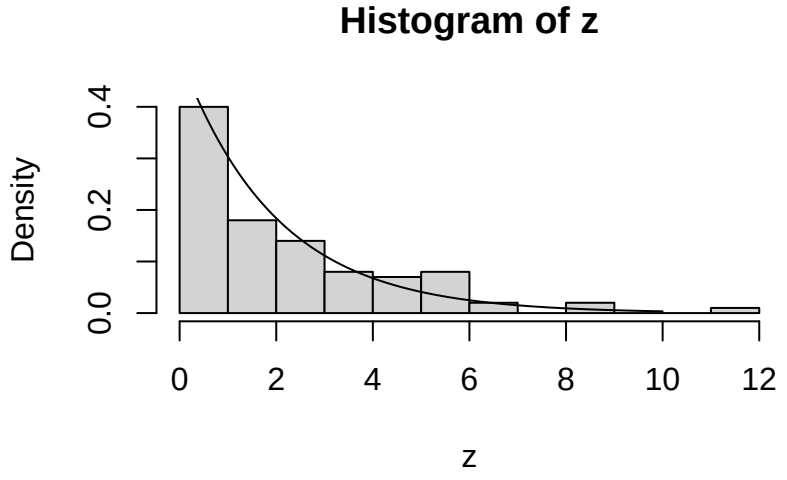

Figura 3: Histograma de una m.a. Exponencial con $\lambda = 0.5$

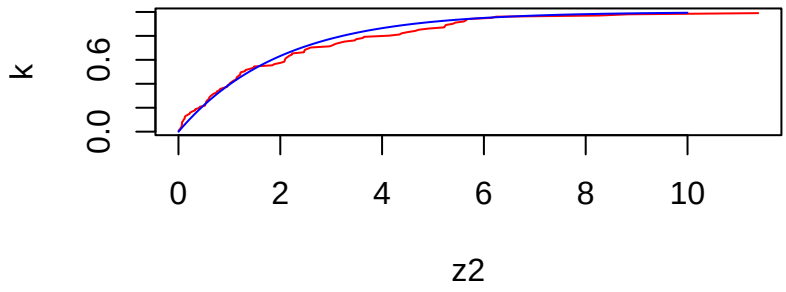

Figura 4: Comparación entre la función de densidad empírica y la teórica a partir de la muestra

Utilicen el teorema de la transformación inversa para generar una muestra aleatoria de tamaño 1000 de la distribución de X. Construyan una tabla de frecuencias relativas y comparen las probabilidades empíricas con las teóricas.
Repitan considerando la función de R sample.

```
#Ejemplo 2
set.seed(157017)
prob <- c(.1, 0.3, 0.5, 0.7, 1)
frec <- findInterval(runif(1000), prob)
table(frec) / 1000


## frec
##     0     1     2     3     4
## 0.125 0.168 0.224 0.185 0.298


set.seed(157017)
frecSample <- sample(x = 0:4, size = 1000, replace = TRUE,
                     prob = c(0.1, 0.2, 0.2, 0.2, 0.3))
table(frecSample) / 1000


## frecSample
##     0     1     2     3     4
## 0.103 0.195 0.224 0.185 0.293
```

2.2. Ejemplo 3

Obtengan una muestra de $10,000$ números de la siguiente distribución discreta:

$$p(x) = \frac{2x}{k(k+1)}, x = 1, 2, \ldots, k$$

para $k = 100$

```
#Ejemplo 3
p <- function(n) {
  values <- sample(1:100, n, replace = TRUE)
  prob <- (2 * values) / (100 * 101)
  return(prob)
}
head(p(10000), n = 30)
```

```
##  [1] 0.0091089109 0.0172277228 0.0106930693 0.0011881188 0.0112871287
##  [6] 0.0164356436 0.0001980198 0.0065346535 0.0196039604 0.0061386139
## [11] 0.0116831683 0.0108910891 0.0106930693 0.0194059406 0.0077227723
## [16] 0.0108910891 0.0186138614 0.0186138614 0.0182178218 0.0003960396
## [21] 0.0091089109 0.0142574257 0.0166336634 0.0172277228 0.0081188119
## [26] 0.0037623762 0.0011881188 0.0196039604 0.0061386139 0.0150495050
```

2.3. Ejemplo 4

Una compañía de seguros tiene 1000 asegurados, cada uno de los cuales presentará de manera independiente una reclamación en el siguiente mes con probabilidad $p = 0.09245$. Suponiendo que las cantidades de los reclamos hechos son variables aleatorias Gamma(7000,1), hagan simulación para estimar la probabilidad de que la suma de los reclamos exceda $\$500,000$.

```
#Ejemplo 4
mayor <- 0
for (i in 1:10000){
  reclamaciones <- sum(rbinom(1000, 1, 0.09245))
  montos <- sum(rgamma(reclamaciones, 7000, 1))
  if (montos > 500000) {
    mayor <- mayor + 1
  }
}
mayor / 10000
```

```
## [1] 0.9897
```

3. Tarea 3

3.1. Ejercicio 1

Simula las siguientes muestras Poisson, todas con $\lambda = 3$, pero de distintos tamaños, $n = 10, 20, 40, 80, 200$. Para cada muestra de estas tres calcula los tres estimadores de momentos dados en las notas en la pág. 3, λ_1, λ_2 y λ_3 .

```
est1 <- function(x) {
  mean(x)
}

est2 <- function(x) {
  n <- length(x)
  sum((x - mean(x))^2) / n
}

est3 <- function(x) {
  n <- length(x)
  -0.5 + sqrt(.25 + sum(x^2) / n)
}
```

```
## [1] "  n |  est1,  est2,  est3"
## [1] " 10 | 2.900, 2.690, 2.869"
## [1] " 20 | 2.900, 3.690, 3.014"
## [1] " 40 | 3.025, 2.774, 2.989"
## [1] " 80 | 3.188, 3.152, 3.183"
## [1] "200 | 3.140, 3.770, 3.226"
```

3.2. Ejercicio 2

Simula una muestra de $n = 15$ variables aleatorias independientes $X_1, ..., X_n$, idénticamente distribuidas como normales con media $\mu = 60$ y parámetro de escala$\sigma = 5$.

```
n1 <- 15
x1 <- rnorm(n1, mean = 60, sd = 5)
k1 <- 1:n1 / (n1)
```

Calcula los estimadores de momentos de μ y σ basados en ecuaciones de los primeros dos momentos, los primeros no centrados y los segundos momentos centrados. Denota a estos estimadores como $\hat{\mu}$y $\hat{\sigma}$.

```r
estMu <- function(x) {
  mean(x)
}

estSigma <- function(x) {
  n <- length(x)
  sqrt(sum((x - mean(x))^2) / n)
}

mu1 <- estMu(x1)
sigma1 <- estSigma(x1)
k1 <- 1 : n1 / (n1)
```

En una misma figura, grafica la función de distribución teórica con línea continua, la distribución estimada con guiones y la función de distribución empírica, graficando puntos de las siguientes coordenadas

$$\left(x_i, \frac{i}{n+1}\right)$$

para $i = 1, ..., n$.

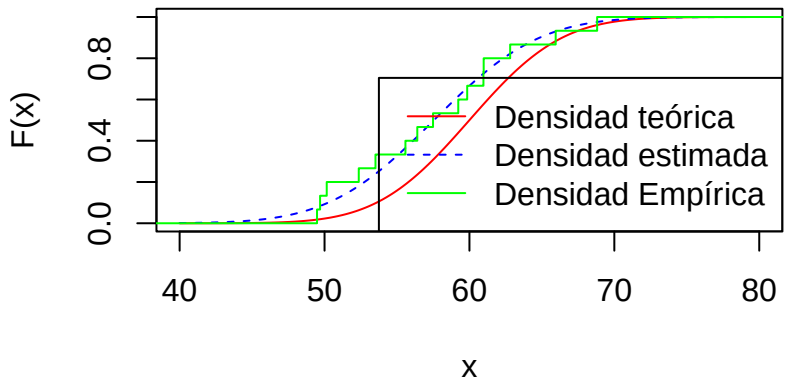

Figura 5: Funciones de distribución teórica, empírica y estimada, $n = 15$

Repite lo mismo pero ahora para $n = 30$ y luego para $n = 100$.

```r
mu2 <- estMu(x2)
sigma2 <- estSigma(x2)
mu3 <- estMu(x3)
```

```
sigma3 <- estSigma(x3)
```

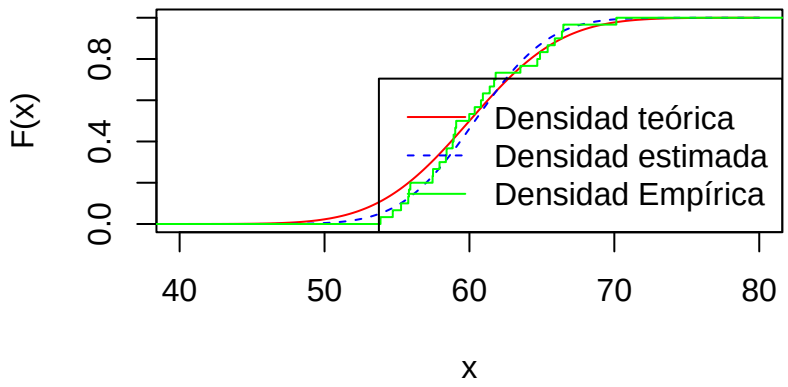

Figura 6: Funciones de distribución teórica, empírica y estimada, $n = 30$

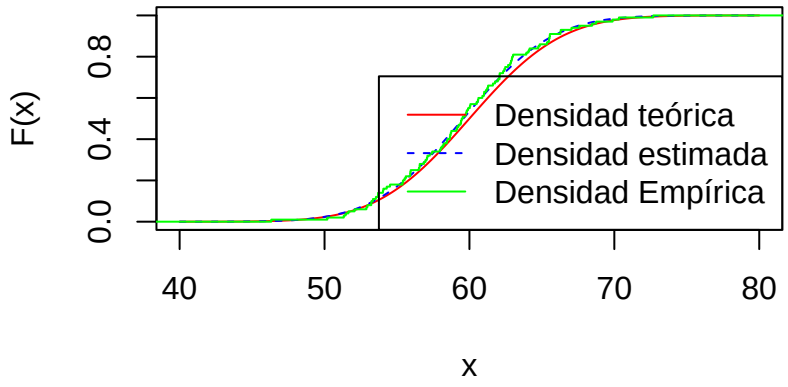

Figura 7: Funciones de distribución teórica, empírica y estimada, $n = 100$

4. Bootstrap no paramétrico

Para esta ayudantía, nos basamos en el libro [1]. Haremos bootstrap no paramétrico para estimar los cuantiles 0.025 y 0.975 de una v.a. exponencial. Sea $X \sim \exp(\lambda = 5)$. Iniciamos con una muestra aleatoria de X con $n = 300$.

```
n <- 300
lambda <- 5
x <- rexp(300, rate = lambda)
```

Para el bootstrap, tomaremos $K = 100$ muestras de tamaño $m = 30$, de las cuales obtendremos el cuantil empírico 0.025 q_1^k y 0.975 q_2^k.

```
r <- mean(x)
k <- 100
m <- 30
q1 <- 0.025
q2 <- 0.975

q1Est <- numeric(k)
q2Est <- numeric(k)

for (i in 1:k){
  z <- sample(x,size = m, replace = FALSE)
  q1Est[i] <- quantile(z,q1)
  q2Est[i] <- quantile(z,q2)
}
```

Finalmente, mostramos la estimación de la media y la desviación estándar de los cuantiles $\overline{q_1}$, $\overline{q_2}$, $\mathrm{sd}\,(q_1)$, $\mathrm{sd}\,(q_2)$ obtenida por bootstrap no paramétrico.

```
Q1 <- mean(q1Est)
Q2 <- mean(q2Est)

S1 <- sd(q1Est)
S2 <- sd(q2Est)

print(sprintf("La media del cuantil %1.3f es %2.3f", q1, Q1))


## [1] "La media del cuantil 0.025 es 0.014"


print(sprintf("La media del cuantil %1.3f es %2.3f", q2, Q2))


## [1] "La media del cuantil 0.975 es 0.679"
```

```r
print(sprintf("La desviación estándar del cuantil %1.3f es %2.3f", q1, S1))
```

```
## [1] "La desviación estándar del cuantil 0.025 es 0.009"
```

```r
print(sprintf("La desviación estándar del cuantil %1.3f es %2.3f", q2, S2))
```

```
## [1] "La desviación estándar del cuantil 0.975 es 0.176"
```

5. Bootstrap paramétrico

Haremos bootstrap paramétrico para estimar el cuantil 0.025 y 0.975 de una muestra exponencial. Sea $X \sim \exp(\lambda = 5)$. Recordemos

$$\mathbb{E}[X] = \frac{1}{\lambda}$$

Por lo que el estimador de λ por el método de momentos es:

$$\hat{\lambda} = \frac{1}{\mathbb{E}[X]} \tag{2}$$

Iniciamos con una muestra aleatoria de X con $n = 300$.

```r
n <- 300
lambda <- 5
x <- rexp(300, rate = lambda)
```

Para el bootstrap, tomaremos $K = 100$ muestras de tamaño $m = 30$, de las cuales obtendremos el estimador por el método de momentos de cada muestra $\hat{\lambda}_k$, para posteriormente encontrar los cuantiles teórico 0.025 q_1^k y 0.975 q_2^k..

```r
r <- mean(x)
k <- 100
m <- 30
q1 <- 0.025
q2 <- 0.975

q1Est <- numeric(k)
q2Est <- numeric(k)

for (i in 1:k){
  z <- rexp(m, rate = 1 / r)
  rBoot <- mean(z)
```

```
  q1Est[i] <- qexp(q1, rate = 1 / rBoot)
  q2Est[i] <- qexp(q2, rate = 1 / rBoot)
}
```

Finalmente, mostramos la estimación de la media y la desviación estándar de los cuantiles $\overline{q_1}$, $\overline{q_2}$, $\mathrm{sd}\,(q_1)$, $\mathrm{sd}\,(q_2)$ obtenida por bootstrap paramétrico.

```
Q1 <- mean(q1Est)
Q2 <- mean(q2Est)

S1 <- sd(q1Est)
S2 <- sd(q2Est)

print(sprintf("La media del cuantil %1.3f es %2.3f", q1, Q1))
```

```
## [1] "La media del cuantil 0.025 es 0.005"
```

```
print(sprintf("La media del cuantil %1.3f es %2.3f", q2, Q2))
```

```
## [1] "La media del cuantil 0.975 es 0.753"
```

```
print(sprintf("La desviación estándar del cuantil %1.3f es %2.3f", q1, S1))
```

```
## [1] "La desviación estándar del cuantil 0.025 es 0.001"
```

```
print(sprintf("La desviación estándar del cuantil %1.3f es %2.3f", q2, S2))
```

```
## [1] "La desviación estándar del cuantil 0.975 es 0.133"
```

6. Intervalos de confianza a partir de bootstrap paramétrico

Haremos bootstrap paramétrico para estimar intervalos del 95 % de confianza de los cuantiles 0.025 y 0.975 de una muestra exponencial. Sea $X \sim \exp(\lambda = 5)$. Recordemos

$$\mathbb{E}\,[X] = \frac{1}{\lambda}$$

Por lo que el estimador de λ por el método de momentos es:

$$\hat{\lambda} = \frac{1}{\mathbb{E}[X]} \tag{3}$$

Iniciamos con una muestra aleatoria de X con $n = 300$.

```
n <- 300
lambda <- 5
x <- rexp(300, rate = lambda)
```

Para el bootstrap, tomaremos $K = 100$ muestras de tamaño $m = 30$, de las cuales obtendremos el estimador por el método de momentos de cada muestra $\hat{\lambda}_k$, para posteriormente encontrar los cuantiles teórico 0.025 q_1^k y 0.975 q_2^k..

```
r <- mean(x)
k <- 100
m <- 30
q1 <- 0.025
q2 <- 0.975

q1Est <- numeric(k)
q2Est <- numeric(k)

for (i in 1:k){
  z <- rexp(m, rate = 1 / r)
  rBoot <- mean(z)
  q1Est[i] <- qexp(q1, rate = 1 / rBoot)
  q2Est[i] <- qexp(q2, rate = 1 / rBoot)
}
```

Ahora, mostramos los intervalos de confianza por el método normal de los cuantiles $\overline{q_1}$, $\overline{q_2}$, $\mathrm{sd}(q_1)$, $\mathrm{sd}(q_2)$ obtenidos por bootstrap paramétrico.

```
Q1 <- mean(q1Est)
Q2 <- mean(q2Est)

S1 <- sd(q1Est)  # R uses ddof=1 by default for unbiased standard deviation
S2 <- sd(q2Est)

T1L <- Q1 - S1 * qnorm(0.975)
T1R <- Q1 + S1 * qnorm(0.975)

T2L <- Q2 - S2 * qnorm(0.975)
T2R <- Q2 + S2 * qnorm(0.975)

q1_real <- qexp(q1, rate = lambda)
```

```r
q2_real <- qexp(q2, rate = lambda)

print("Por el método normal:")
```

```
## [1] "Por el método normal:"
```

```r
print(sprintf("El intervalo de confianza para el cuantil %.3f es (%.5f, %.5f)", q1, T1L, T1R))
```

```
## [1] "El intervalo de confianza para el cuantil 0.025 es (0.00303, 0.00637)"
```

```r
if (T1L < q1_real && q1_real < T1R) {
  print(sprintf("El cuantil %.3f está en el intervalo de confianza", q1))
} else {
  print(sprintf("El cuantil %.3f no está en el intervalo de confianza", q1))
}
```

```
## [1] "El cuantil 0.025 está en el intervalo de confianza"
```

```r
print(sprintf("El intervalo de confianza para el cuantil %.3f es (%.5f, %.5f)", q2, T2L, T2R))
```

```
## [1] "El intervalo de confianza para el cuantil 0.975 es (0.44117, 0.92873)"
```

```r
if (T2L < q2_real && q2_real < T2R) {
  print(sprintf("El cuantil %.3f está en el intervalo de confianza", q2))
} else {
  print(sprintf("El cuantil %.3f no está en el intervalo de confianza", q2))
}
```

```
## [1] "El cuantil 0.975 está en el intervalo de confianza"
```

Finalmente, mostramos los intervalos de confianza por el método pivotal de los cuantiles $\overline{q_1}$, $\overline{q_2}$, $\mathrm{sd}\,(q_1)$, $\mathrm{sd}\,(q_2)$ obtenidos por bootstrap paramétrico.

```r
q1_N <- qexp(q1, rate = 1 / r)
q2_N <- qexp(q2, rate = 1 / r)

q1_R <- quantile(q1Est, 0.025)
q1_L <- quantile(q1Est, 0.975)
q2_R <- quantile(q2Est, 0.025)
q2_L <- quantile(q2Est, 0.975)
```

```r
P1L <- 2 * q1_N - q1_L
P1R <- 2 * q1_N - q1_R
P2L <- 2 * q2_N - q2_L
P2R <- 2 * q2_N - q2_R

print("Por el método pivotal:")
```

```
## [1] "Por el método pivotal:"
```

```r
print(sprintf("El intervalo de confianza para el cuantil %.3f es (%.5f, %.5f)", q1, P1L, P1R))
```

```
## [1] "El intervalo de confianza para el cuantil 0.025 es (0.00273, 0.00601)"
```

```r
if (P1L < q1_real && q1_real < P1R) {
  print(sprintf("El cuantil %.3f está en el intervalo de confianza", q1))
} else {
  print(sprintf("El cuantil %.3f no está en el intervalo de confianza", q1))
}
```

```
## [1] "El cuantil 0.025 está en el intervalo de confianza"
```

```r
print(sprintf("El intervalo de confianza para el cuantil %.3f es (%.5f, %.5f)", q2, P2L, P2R))
```

```
## [1] "El intervalo de confianza para el cuantil 0.975 es (0.39764, 0.87631)"
```

```r
if (P2L < q2_real && q2_real < P2R) {
  print(sprintf("El cuantil %.3f está en el intervalo de confianza", q2))
} else {
  print(sprintf("El cuantil %.3f no está en el intervalo de confianza", q2))
}
```

```
## [1] "El cuantil 0.975 está en el intervalo de confianza"
```

Referencias

[1] L. Wasserman, *All of Statistics*. Springer New York, 2004. [Online]. Available: https://doi.org/10.1007/978-0-387-21736-9

[2] Y. Xie, *Dynamic Documents with R and knitr, Second Edition*, 2nd ed., ser. Chapman & Hall/CRC The R Series. Philadelphia, PA: Chapman & Hall/CRC, Jun. 2015.