Modelos de Probabilidad y Contrastes de Bondad de Ajuste

Grado en Ingeniería Informática

2025/26

1. Objetivos

En esta práctica tenemos dos grupos de objetivos:

- Modelos de probabilidad:
  - Representar las funciones de probabilidad/densidad y distribución de diferentes modelos de variables aleatorias continuas/discretas.
  - Calcular probabilidades utilizando diferentes distribuciones.
  - Interpretar y comparar gráficos de distribución.
- Contrastes de Bondad de Ajuste
  - Contraste Chi-cuadrado.
  - Contrastes de normalidad.

2. Modelos de probabilidad

El paquete `stats` de R implementa los modelos de probabilidad más utilizados. La siguiente lista no es exhaustiva, pero contiene todos los modelos que se utilizarán durante este curso.

- Distribución Beta, `dbeta`.
- Binomial (incluye Bernoulli), `dbinom`.
- Distribución Cauchy, `dcauchy`.
- Distribución Chi-squared, `dchisq`.
- Distribución Exponencial, `dexp`.
- Distribución F de Fisher, `df`.
- Distribución Gamma, `dgamma`.
- Distribución Geométrica, `dgeom`.
- Distribución Hipergeométrica, `dhyper`.
- Distribución Log-normal, `dlnorm`.
- Distribución Multinomial, `dmultinom`.
- Distribución binomial negative, `dnbinom`.
- Distribución Normal, `dnorm`.
- Distribución Poisson, `dpois`.
- Distribución t de Student, `dt`.
- Distribución Uniforme, `dunif`.
- Distribución Weibull, `dweibull`.

Como se puede deducir de la lista, el nombre de las funciones está formado por la letra `d` más el `name` o una abreviatura del nombre de la distribución. Esta regla es común a todas las distribuciones y las letras `d`, `p`, `q` y `r` se utilizan para denotar la densidad o la función de masa, la función de distribución acumulada, la función cuantílica y la generación de números aleatorios, respectivamente.

Primero, ilustraremos su uso con el modelo uniforme, que es uno de los más simples. Luego, en las siguientes secciones, estudiaremos en detalle aquellos modelos que se usan con mayor frecuencia para resolver fenómenos encontrados en Ingeniería.

La distribución uniforme, $U(a, b)$, tiene la siguiente función de densidad

$$f(x) = \begin{cases} \frac{1}{b-a} & \text{si} \quad x \in [a, b] \\ 0 & \text{si} \quad x \notin [a, b] \end{cases}$$

Supongamos que $a = -1$ y $b = 1$, luego $\frac{1}{b-a} = 1/2$ si $x \in [-1, 1]$ y cero en caso contrario. Esto se puede verificar con el siguiente código

```
x_below_the_interval = seq(-2,-1.1,.1)
dunif(x_below_the_interval, min = -1, max = 1)
```

```
##  [1] 0 0 0 0 0 0 0 0 0 0
```

```
x_in_the_interval = seq(-1,1,.1)
dunif(x_in_the_interval, min = -1, max = 1)
```

```
##  [1] 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5
## [20] 0.5 0.5
```

```
x_above_the_interval = seq(1.1,2,.1)
dunif(x_above_the_interval, min = -1, max = 1)
```

```
##  [1] 0 0 0 0 0 0 0 0 0 0
```

Por supuesto, lo anterior es solo una ilustración, no una prueba formal.

La función de distribución de una variable aleatoria uniforme, $U(a, b)$, viene dada por

$$F(x) = \begin{cases} 0 & \text{si} \quad x < a \\ \frac{x-a}{b-a} & \text{si} \quad x \in [a, b) \\ 1 & \text{si} \quad x \geq b \end{cases},$$

que se puede ilustrar, para $a = -1$ y $b = 1$, con el siguiente código

```
punif(x_below_the_interval, min = -1, max = 1)
```

```
##  [1] 0 0 0 0 0 0 0 0 0 0
```

```
punif(x_in_the_interval, min = -1, max = 1)
```

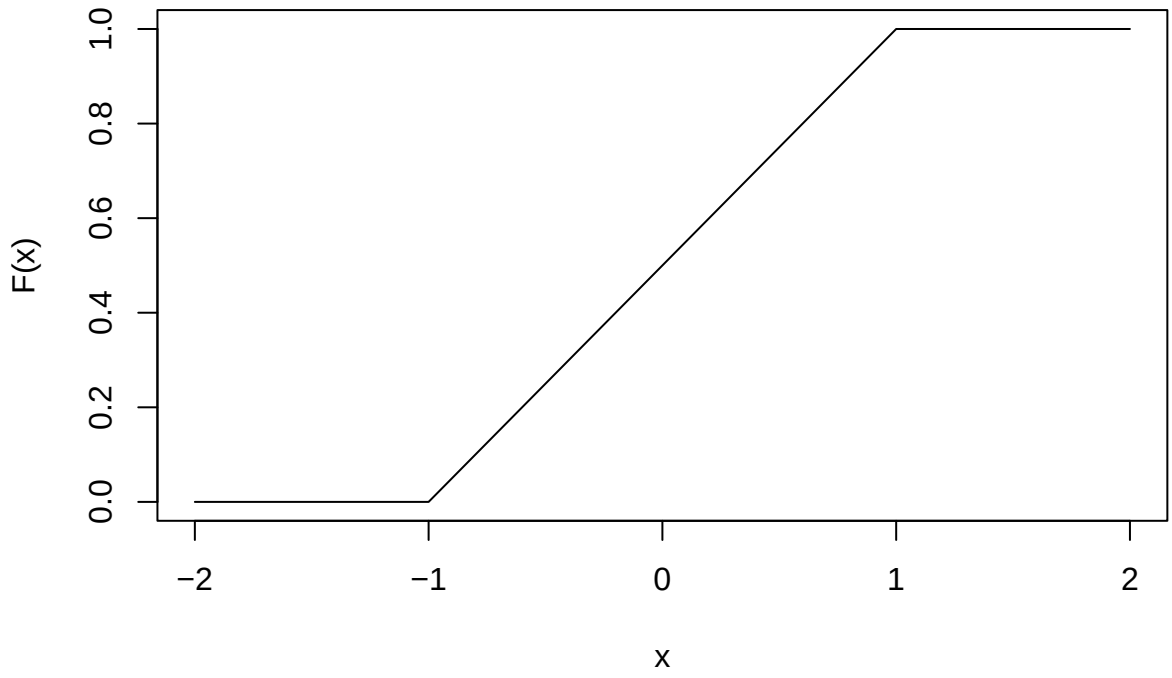
```
##  [1] 0.00 0.05 0.10 0.15 0.20 0.25 0.30 0.35 0.40 0.45 0.50 0.55 0.60 0.65 0.70
## [16] 0.75 0.80 0.85 0.90 0.95 1.00
```

```r
punif(x_above_the_interval, min = -1, max = 1)
```

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

Además, podemos obtener la representación de esta función de distribución mediante

```r
x = c(x_below_the_interval, x_in_the_interval, x_above_the_interval)
Fx = punif(x, min = -1, max = 1)
plot(x, Fx, type = "l", ylab = "F(x)")
```



La función cuantílica de una variable aleatoria uniforme, $U(a, b)$, se define por

$$Q(p) = a + p(b - a) \text{ si } p \in [0, 1].$$

Por ejemplo, si queremos calcular los cuartiles de una $U(a, b)$ debemos evaluar la expresión anterior en los puntos $p = 0.25$, 0.5 y 0.75 para el primer, segundo (mediana) y tercer cuartiles, respectivamente. Veamos el código para $a = -1$ and $b = 1$

```r
p = c(0.25, 0.5, 0.75)
qunif(p, min = -1, max = 1)
```

```
## [1] -0.5  0.0  0.5
```

Los cuartiles de una $U(-1,1)$ son -0.5, 0 y 0.5.

Por supuesto, la función cuantílica no está definida fuera del intervalo $[0,1]$ y R devuelve `NaN`, *Not a Number*.

```
p = c(-0.25, 1.25)
qunif(p, min = -1, max = 1)
```

```
## Warning in qunif(p, min = -1, max = 1): NaNs produced
```

```
## [1] NaN NaN
```

2.1. Distribuciones discretas: Binomial y Poisson.

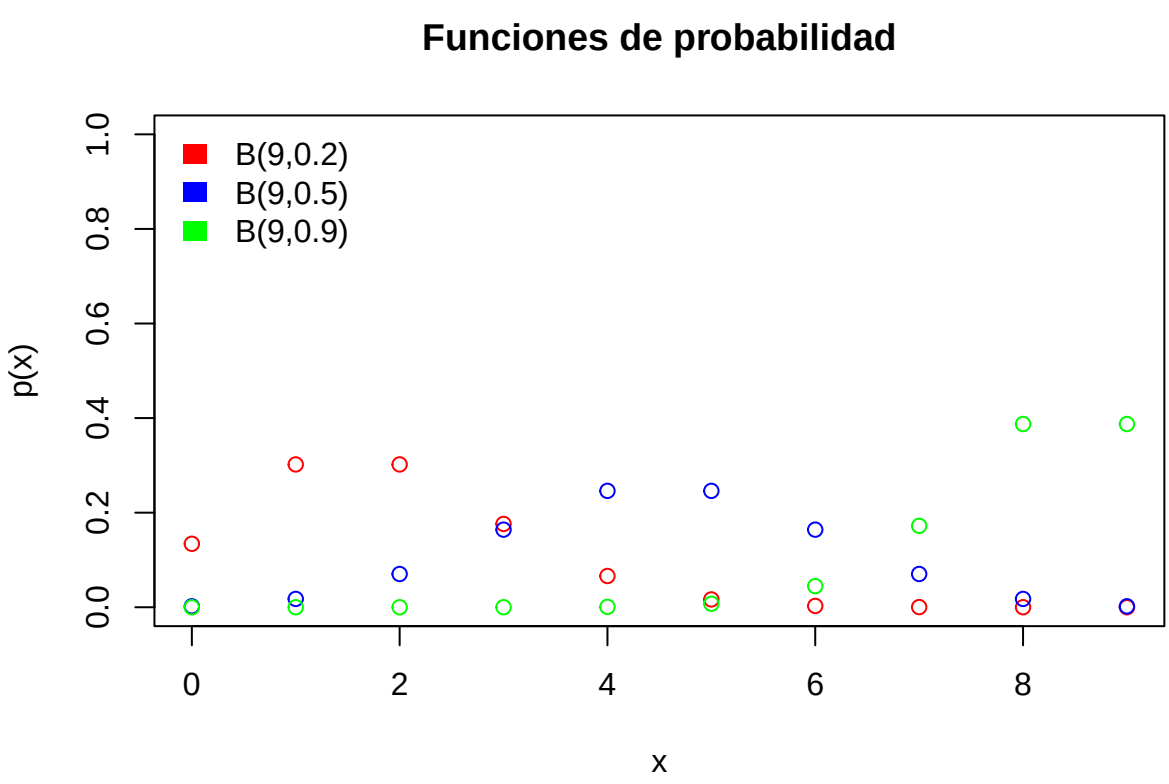
2.1.1. Distribución binomial, $X \sim B(n,p)$

Recordamos que una distribución binomial con parámetros n y p representa una variable aleatoria que cuenta el número de éxitos que podríamos obtener repitiendo un experimento de Bernoulli, es decir, un experimento cuyos únicos resultados son 1 (éxito) y 0 (fracaso). n es el número de veces que repetimos el experimento de Bernoulli (número de ensayos) y p es la probabilidad de éxito (probabilidad de evento) y se supone que los experimentos son independientes.

Representación gráfica de las funciones de probabilidad y distribución:

En el siguiente ejemplo usamos tres distribuciones binomiales diferentes B(9,0.2), B(9,0.5) and B(9,0.9).

```
n = 9
p1 = 0.2
p2 = 0.5
p3 = 0.9
x = 0:n
Px1 = dbinom(x, n, prob = p1)
Px2 = dbinom(x, n, prob = p2)
Px3 = dbinom(x, n, prob = p3)
plot(x, Px1, xlim = c(0,9), ylim = c(0,1), col = "red",
     main = "Funciones de probabilidad", ylab = "p(x)")
points(x, Px2, xlim = c(0,9), ylim = c(0,1), col = "blue")
points(x, Px3, xlim = c(0,9), ylim = c(0,1), col = "green")
legend('topleft',c('B(9,0.2)', 'B(9,0.5)', 'B(9,0.9)'),
       fill = c("red", "blue", "green"), bty = 'n', border = NA)
```

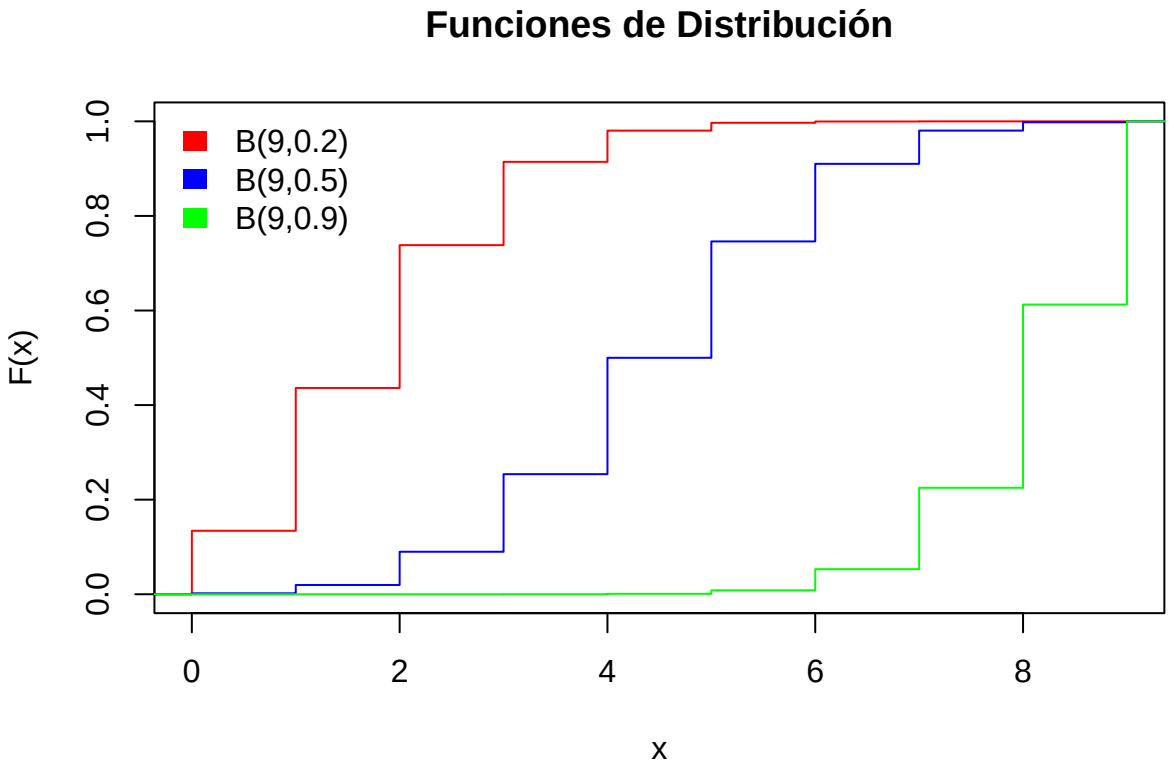


Podemos notar que:

- cuando $p = 0.5$ (en la figura anterior se muestra en azul) la distribución es simétrica,
- cuando $p < 0.5$ (en la figura anterior se muestra en rojo, correspondiente a $p = 0.2$), la distribución es asimétrica hacia la derecha, significa que la variable aleatoria es asimétrica positiva,
- cuando $p > 0.5$ (en la figura anterior se muestra en verde, correspondiente a $p = 0.9$), la distribución es asimétrica hacia la izquierda, significa que la variable aleatoria es asimétrica negativa.

Si queremos mostrar las funciones de distribución, solo necesitamos cambiar `dbinom` por `pbinom` y como sabemos que es una función escalonada usamos `type = "s"`:

```
x = c(-1, x, 10)
Fx1 = pbinom(x, n, prob = p1)
Fx2 = pbinom(x, n, prob = p2)
Fx3 = pbinom(x, n, prob = p3)
plot(x, Fx1, xlim = c(0,9), ylim = c(0,1), col = "red", main = "Funciones de Distribución",
     type = "s", ylab = "F(x)")
lines(x, Fx2, xlim = c(0,9), ylim = c(0,1), col = "blue", type = "s")
lines(x, Fx3, xlim = c(0,9), ylim = c(0,1), col = "green", type = "s")
legend('topleft',c('B(9,0.2)', 'B(9,0.5)', 'B(9,0.9)'), fill = c("red", "blue", "green"),
       bty = 'n', border = NA)
```



Cabe señalar que agregamos dos valores adicionales en `x = c(-1, x, 10)`, -1 y 10, que están fuera del conjunto $\{0, 1, ..., 9\}$ donde $B(9, p)$ toma valores. La razón es obtener un gráfico completo de la función de distribución. Observe también que usamos `lines` en lugar de `points`.

Cálculo de Probabilidades

Supongamos que tenemos una variable aleatoria $X \sim B(12, 0.4)$ y queremos calcular las siguientes probabilidades:

- $\Pr(X = 7)$
- $\Pr(X > 3)$
- $\Pr(X \leq 8)$
- $\Pr(X < 5)$

Las solutiones, en R, son

```
dbinom(7, 12, prob = 0.4)
```

```
## [1] 0.1009024
```

```
1-pbinom(3, 12, prob = 0.4) # puesto que Pr(X > 3) = 1 - Pr(X =< 3)
```

```
## [1] 0.7746627
```

```r
pbinom(8, 12, prob = 0.4)
```

```
## [1] 0.9847327
```

```r
pbinom(4, 12, prob = 0.4) # puesto que Pr(X < 5) = Pr(X =< 4)
```

```
## [1] 0.4381782
```

Cálculo de los percentiles de la distribución

Supongamos que queremos calcular los percentiles de una variable aleatoria dada X. Dado un porcentaje p, el percentil correspondiente es un número tal que el $p\%$ de los individuos de una población tienen valores menores o iguales. En la práctica, seleccionamos un valor p y la función "`q + name`" devolverá el valor, a, tal que $\Pr(X \leq a) = p$.

Por ejemplo, suponga que $X \sim B(4, 0.5)$. Queremos calcular el valor, a, tal que $\Pr(X \leq a) = 0.3125$.

```r
qbinom(0.3125, 4, prob = 0.5)
```

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

Podemos verificar el resultado anterior calculando

```r
pbinom(0:4, 4, prob = 0.5)
```

```
## [1] 0.0625 0.3125 0.6875 0.9375 1.0000
```

donde podemos ver que $\Pr(X \leq 1) = 0.3125$.

Cabe señalar que para una variable aleatoria discreta la ecuación $\Pr(X \leq a) = p$ puede no tener solución. Por ejemplo, en el resultado anterior, vemos que no hay ningún valor que satisfaga $\Pr(X \leq a) = 0.1$. Sin embargo, la función `qbinom` devuelve un valor

```r
qbinom(0.1, 4, prob = 0.5)
```

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

que es el primer valor de a que satisface la siguiente desigualdad $\Pr(X \leq a) \geq p$.

Ejemplo práctico:

Un viajero del metro va todas las mañanas a la misma hora a la plataforma del metro. El 18% de las veces el tren ya está allí y el resto de las veces tiene que esperar el tren.

a) Considerando siete días consecutivos, ¿cuál es la probabilidad de que sólo uno de los siete días no tenga que esperar al tren?

b) Considerando quince días consecutivos, ¿cuál es la probabilidad de que a lo sumo tres días no tenga que esperar al tren?

c) Considerando dieciocho días consecutivos, ¿cuál es la probabilidad de que no tenga que esperar al tren por más de cinco días?

Defina X como el número de días que el viajero no tiene que esperar el tren. Entonces $X \sim B(n, p)$, es decir, se distribuye como una binomial con probabilidad de evento $p = 0.18$. Obtenemos que

a) $X \sim B(7, 0.18)$; $\Pr(X = 1) = 0.3830484$ usando `dbinom(1, 7, 018)`.

b) $X \sim B(15, 0.18)$; $\Pr(X \leq 3) = 0.7218051$ usando `pbinom(3, 15, 0.18)`.

c) $X \sim B(18, 0.18)$; $\Pr(X > 5) = 0.08893546$ usando `1-pbinom(5, 18, 0.18)` o `pbinom(5, 18, 0.18, lower.tail = FALSE)`.

2.1.2. Distribución de Poisson, $X \sim Poisson(\lambda)$

Una variable aleatoria de Poisson, $X \sim Poisson(\lambda)$, puede representar el número de eventos independientes que pueden ocurrir en una unidad de tiempo cuando el proceso subyacente es un proceso de Poisson con parámetro constante λ. El único parámetro λ representa el número promedio de eventos en la unidad de tiempo (que también puede ser longitud, superficie, volumen o lo que sea la unidad de medida continua elegida).

Para el modelo de Poisson, podemos hacer los mismos cálculos que para el modelo binomial, por lo que iremos directamente a un ejemplo práctico.

Ejemplo práctico:

El número de usuarios que acceden a un servidor de red es, en promedio, 3000 por hora. La red puede prestar servicios con un rendimiento óptimo de hasta 100 accesos por minuto. Suponiendo que los accesos se producen de manera independiente y a velocidad constante, queremos calcular la probabilidad de que en un minuto dado haya

a) exactamente 40 usuarios que acceden,

b) entre 40 y 50 usuarios que acceden,

c) más de 100 accesos, y por lo tanto haya retrasos en las comunicaciones de la red.

Sea $X =$ número de accesos por minuto, eso significa $X \sim Poisson(\lambda = 50)$. Las probabilidades requeridas están dadas por

a) $\Pr(X = 40) = 0.02149963$ usando `dpois(40, lambda = 50)`.

b) $\Pr(40 \leq X \leq 50) = \Pr(X \leq 50) - \Pr(X \leq 39) = 0.4729463$ usando `ppois(50, lambda = 50) - ppois(39, lambda = 50)`.

c) $\Pr(X > 100) = 1.569746e-10$ usando `ppois(100, lambda = 50, lower.tail = FALSE)`. Por lo tanto, la red está bien dimensionada para la cantidad real de tráfico.

2.2. Distribuciones continuas: Normal y Exponencial

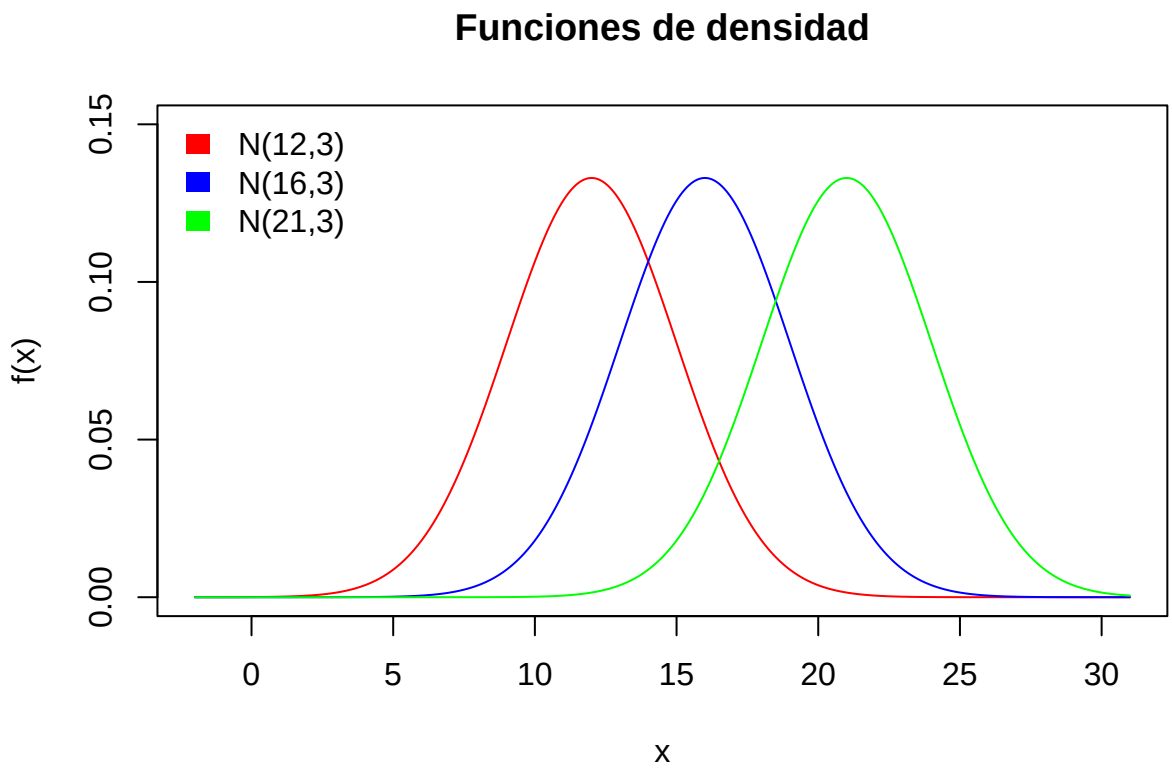
2.2.1. Distribución normal, $X \sim \mathcal{N}(\mu, \sigma)$

La distribución normal es simétrica. La media, μ, coincide con la moda y la mediana. La función de densidad tiene forma de campana y suele llamarse "campana de Gauss".

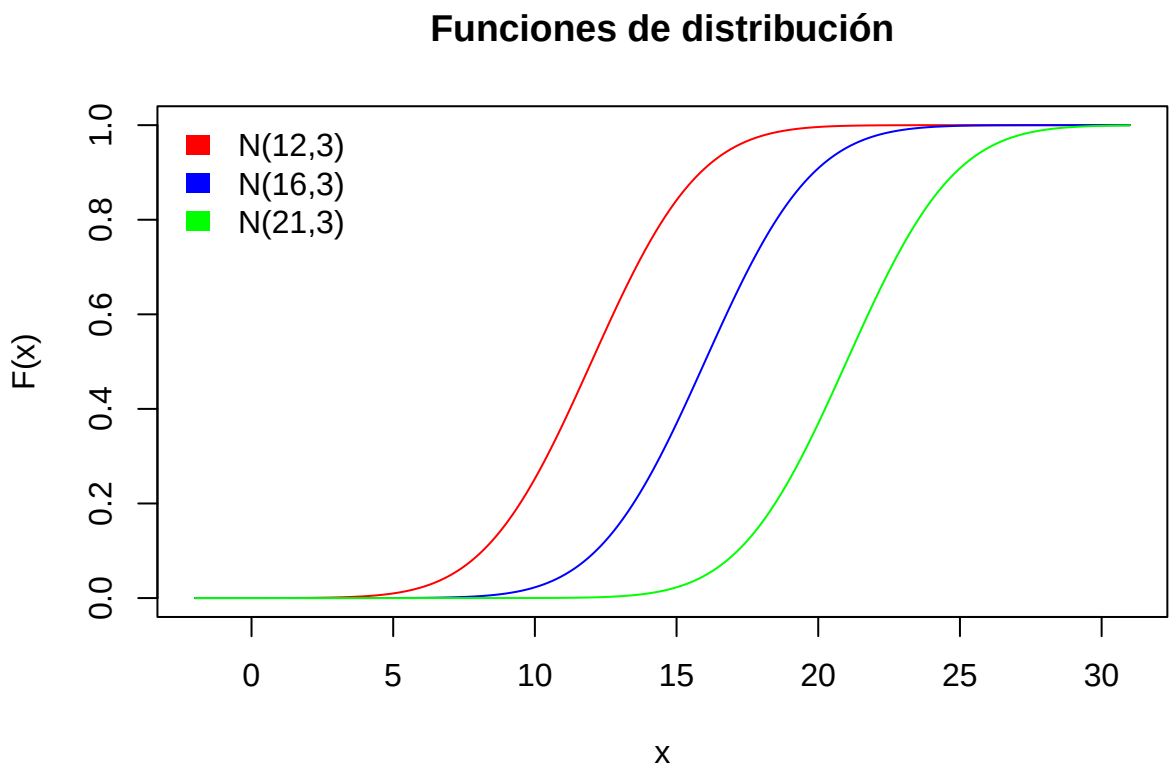
Comparación de la densidad y las funciones de distribución

Dibujamos la densidad y las funciones de distribución de tres variables aleatorias normales diferentes, con igual varianza $\sigma^2 = 9$ y diferentes μ: $\mathcal{N}(12, 3)$, $\mathcal{N}(16, 3)$ y $\mathcal{N}(21, 3)$. Podemos ver cómo la campana mueve su centro a lo largo del eje real sin cambiar su aspecto.

```
x = seq(-2, 31, .01)
gx1 = dnorm(x, mean = 12, sd = 3)
gx2 = dnorm(x, mean = 16, sd = 3)
gx3 = dnorm(x, mean = 21, sd = 3)
plot(x, gx1, xlim = c(-2,31), ylim = c(0,.15), col = "red", main = "Funciones de densidad",
     type = "l", ylab = "f(x)")
lines(x, gx2, xlim = c(-2,31), ylim = c(0,.15), col = "blue")
lines(x, gx3, xlim = c(-2,31), ylim = c(0,.15), col = "green")
legend('topleft',c('N(12,3)', 'N(16,3)', 'N(21,3)'), fill = c("red", "blue", "green"),
       bty = 'n', border = NA)
```

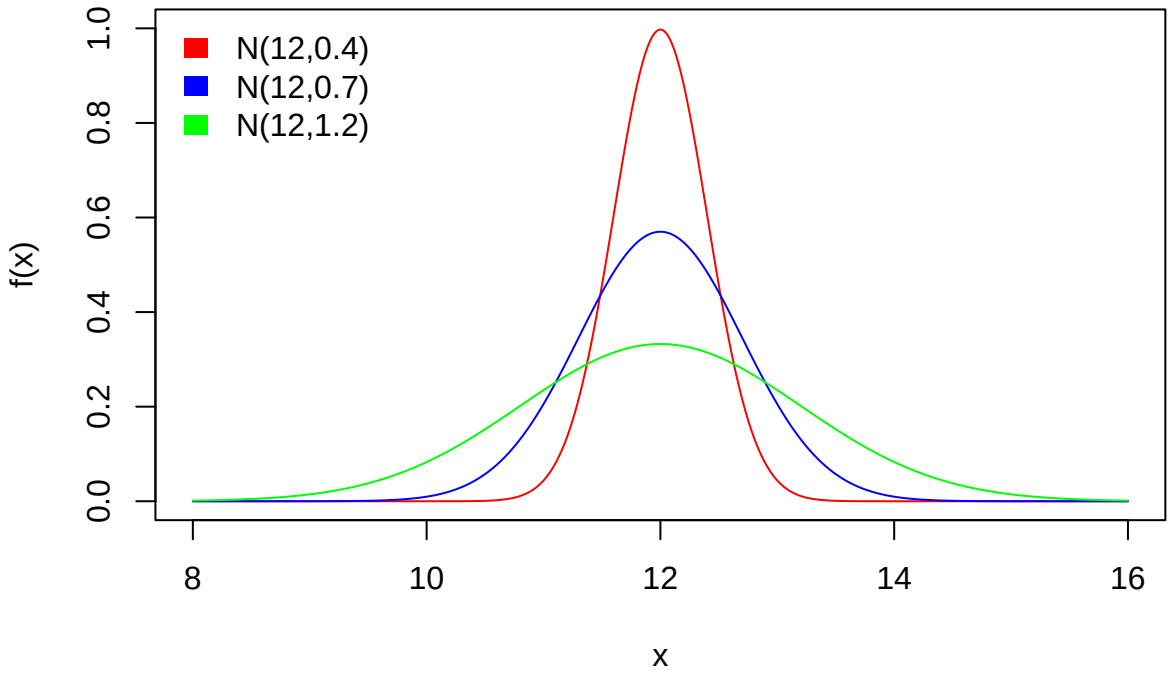


```r
Gx1 = pnorm(x, mean = 12, sd = 3)
Gx2 = pnorm(x, mean = 16, sd = 3)
Gx3 = pnorm(x, mean = 21, sd = 3)
plot(x, Gx1, xlim = c(-2,31), ylim = c(0,1), col = "red", main = "Funciones de distribución",
     type = "l", ylab = "F(x)")
lines(x, Gx2, xlim = c(-2,31), ylim = c(0,1), col = "blue")
lines(x, Gx3, xlim = c(-2,31), ylim = c(0,1), col = "green")
legend('topleft',c('N(12,3)', 'N(16,3)', 'N(21,3)'), fill = c("red", "blue", "green"),
       bty = 'n', border = NA)
```

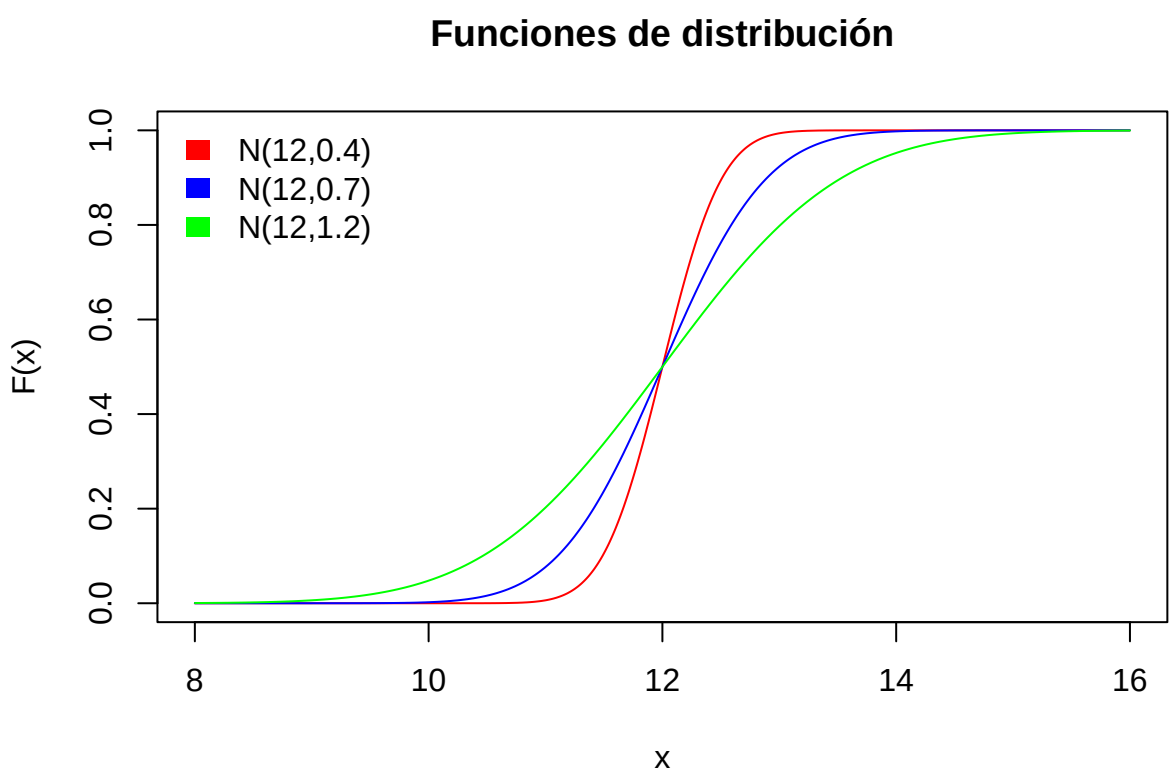


Ahora dibujamos la densidad y las funciones de distribución de tres variables aleatorias normales con media igual $\mu = 12$ y diferentes desviaciones estándar σ: $\mathcal{N}(12, 0.4)$, $\mathcal{N}(12, 0.7)$ y $\mathcal{N}(12, 1.2)$. Ahora podemos ver que las campanas tienen el mismo centro pero diferentes tamaños (la dispersión de los valores cambia).

```r
x = seq(8, 16, .01)
gx1 = dnorm(x, mean = 12, sd = 0.4)
gx2 = dnorm(x, mean = 12, sd = 0.7)
gx3 = dnorm(x, mean = 12, sd = 1.2)
plot(x, gx1, xlim = c(8,16), ylim = c(0,1), col = "red", main = "Funciones de densidad",
     type = "l", ylab = "f(x)")
lines(x, gx2, xlim = c(8,16), ylim = c(0,1), col = "blue")
lines(x, gx3, xlim = c(8,16), ylim = c(0,1), col = "green")
legend('topleft',c('N(12,0.4)', 'N(12,0.7)', 'N(12,1.2)'),
       fill = c("red", "blue", "green"), bty = 'n', border = NA)
```




```r
Gx1 = pnorm(x, mean = 12, sd = 0.4)
Gx2 = pnorm(x, mean = 12, sd = 0.7)
Gx3 = pnorm(x, mean = 12, sd = 1.2)
plot(x, Gx1, xlim = c(8,16), ylim = c(0,1), col = "red", main = "Funciones de distribución",
     type = "l", ylab = "F(x)")
lines(x, Gx2, xlim = c(8,16), ylim = c(0,1), col = "blue")
lines(x, Gx3, xlim = c(8,16), ylim = c(0,1), col = "green")
legend('topleft',c('N(12,0.4)', 'N(12,0.7)', 'N(12,1.2)'),
       fill = c("red", "blue", "green"), bty = 'n', border = NA)
```



Cálculo de Probabilidades

Según la definición de función de densidad, calcular una probabilidad dada es equivalente a calcular el valor de una integral. De hecho, la probabilidad de que una variable aleatoria X tome valores en un intervalo dado es igual al valor de la integral de la función de densidad en ese intervalo.[1]

Ejemplo: Dada $X \sim \mathcal{N}(8, 2.6)$ calcule $\Pr(X > 11.3)$, $\Pr(X < 7.9)$, $\Pr(-1 < X < 4)$ y $\Pr(X \geq 18)$.

a) $\Pr(X > 11.3) = 0.1021794$ usando `pnorm(11.3, 8, 2.6, lower.tail = FALSE)`.

b) $\Pr(X < 7.9) = 0.4846598$ usando `pnorm(7.9, 8, 2.6)`.

c) $\Pr(-1 < X < 4) = \Pr(X < 4) - \Pr(X < -1) = 0.06169935$ usando `pnorm(4, 8, 2.6) - pnorm(-1, 8, 2.6)`,

d) $\Pr(X \geq 18) = 5.999322e-05$ usando `pnorm(18, 8, 2.6, lower.tail = FALSE)`.

Cálculo de percentiles

Para calcular los percentiles, debemos seguir los mismos pasos que para el caso discreto. Por ejemplo, suponga que queremos calcular los percentiles 90%, 95%, 97.5% y 99% de la distribución normal estándar, $Z \sim \mathcal{N}(0, 1)$

[1] Puesto que X es una variable aleatoria continua, $\Pr(X = x) = 0$ para todo x, por tanto $\Pr(X \leq x) = \Pr(X < x)$.

```r
p = c(0.9, 0.95, 0.975, 0.99)
qnorm(p, 0, 1)
```

```
## [1] 1.281552 1.644854 1.959964 2.326348
```

Estos valores generalmente se denotan por $z_{0.1} = 1.281552$, $z_{0.05} = 1.644854$, $z_{0.025} = 1.959964$ y $z_{0.01} = 2.326348$.

Además, en el caso específico de la variable aleatoria normal, es interesante saber cuál es la probabilidad de que la variable aleatoria $X \sim \mathcal{N}(\mu, \sigma)$ tome valores que disten 1, 2 ó 3 desviaciones estándar σ del centro μ. En otras palabras, podríamos estar interesados en las probabilidades de que X caiga en el intervalo $(\mu - k\sigma, \mu + k\sigma)$, donde $k = 1, 2$ o 3.

$$\Pr(\mu - k\sigma < X < \mu + k\sigma) = \Pr\left(-k < \frac{X - \mu}{\sigma} < k\right) = \Pr(-k < Z < k),$$

donde Z es la notación habitual para la distribución normal estándar, $\mathcal{N}(0, 1)$. Las probabilidades anteriores se pueden calcular mediante

```r
pnorm(3,0,1) - pnorm(-3,0,1)
```

```
## [1] 0.9973002
```

```r
pnorm(2,0,1) - pnorm(-2,0,1)
```

```
## [1] 0.9544997
```

```r
pnorm(1,0,1) - pnorm(-1,0,1)
```

```
## [1] 0.6826895
```

Por tanto,

- $\Pr(X \in (\mu - \sigma, \mu + \sigma)) = 0.6826895$,
- $\Pr(X \in (\mu - 2\sigma, \mu + 2\sigma)) = 0.9544997$,
- $\Pr(X \in (\mu - 3\sigma, \mu + 3\sigma)) = 0.9973002$.

2.2.2. Distribución exponencial

La distribución exponencial es interesante por su aplicación a diferentes fenómenos:

- El tiempo de espera hasta el primer evento en un proceso de Poisson (incluso podría ser una llegada, una llamada telefónica o cualquier evento sobre un soporte continuo). Si contamos el número de estos eventos en la unidad de tiempo, tendríamos una Poisson(λ), donde λ es el número medio de eventos en la unidad de tiempo. La suposición es que los eventos llegan de manera independiente y a un ritmo constante.
- El tiempo transcurrido desde un instante dado hasta la próxima aparición de un evento.

Es importante recordar que la variable aleatoria exponencial tiene la propiedad de ausencia de memoria.

Ejemplo práctico:

El número de usuarios que acceden a un servidor de red es, en promedio, 3000 por hora. La red puede prestar servicios con un rendimiento óptimo de hasta 100 accesos por minuto. Suponiendo que los accesos se producen de manera independiente y a una velocidad constante, queremos calcular la probabilidad de que el tiempo que pasa entre dos accesos sea de al menos 5 segundos.

Solución: El supuesto de independencia y la tasa constante implican que el número de accesos por unidad de tiempo se distribuye de acuerdo con una variable aleatoria de Poisson y, por lo tanto, el tiempo entre dos accesos sucesivos es una variable aleatoria exponencial, $T \sim Exp(\lambda = 3000/3600\ accesos/s)$. La media de T es $1/\lambda = 3600/3000 = 1.2\ s$. La probabilidad requerida está dada por

$$\Pr(T > 5) = 0.01550385$$

que se obtiene mediante `pexp(5, rate = 3000/3600, lower.tail = FALSE)`.

3. Contrastes de bondad de ajuste

En esta sección, utilizaremos la variable `Ordenador_Uni` en el archivo`TiempoAccesoWeb.xlsx`. Esta variable contiene 55 mediciones de tiempos, medidos en segundos, que son los tiempos necesarios para acceder a la página web de la Universidad UC3M desde un ordenador de su biblioteca. A partir de este conjunto de datos, queremos encontrar un modelo de probabilidad que describa bien la población de los tiempos de acceso necesarios para acceder desde una computadora de la biblioteca a la página web de la Universidad UC3M. Posteriormente analizamos la variable `tiempo` del archivo `AlumnosIndustriales.xlsx` que contiene mediciones del tiempo que un grupo de estudiantes invierte para llegar a la Universidad.

3.1 Ajuste del Modelo. Variable `Ordenador_Uni`

3.1.1 Análisis descriptivo de los datos

Lo primero que se debe hacer es el análisis descriptivo de los datos (calcular las medidas características e inspeccionar el histograma). De esta manera, podríamos tener una primera idea de qué modelo usar.

Primero leemos y vemos el archivo de datos. La figura muestra las primeras cinco observaciones de este archivo de datos.

```
library(readxl)
```

```
## Warning: package 'readxl' was built under R version 4.2.3
```

```
TiempoAccesoWeb <- read_excel("TiempoAccesoWeb.xlsx")
head(TiempoAccesoWeb,5)
```

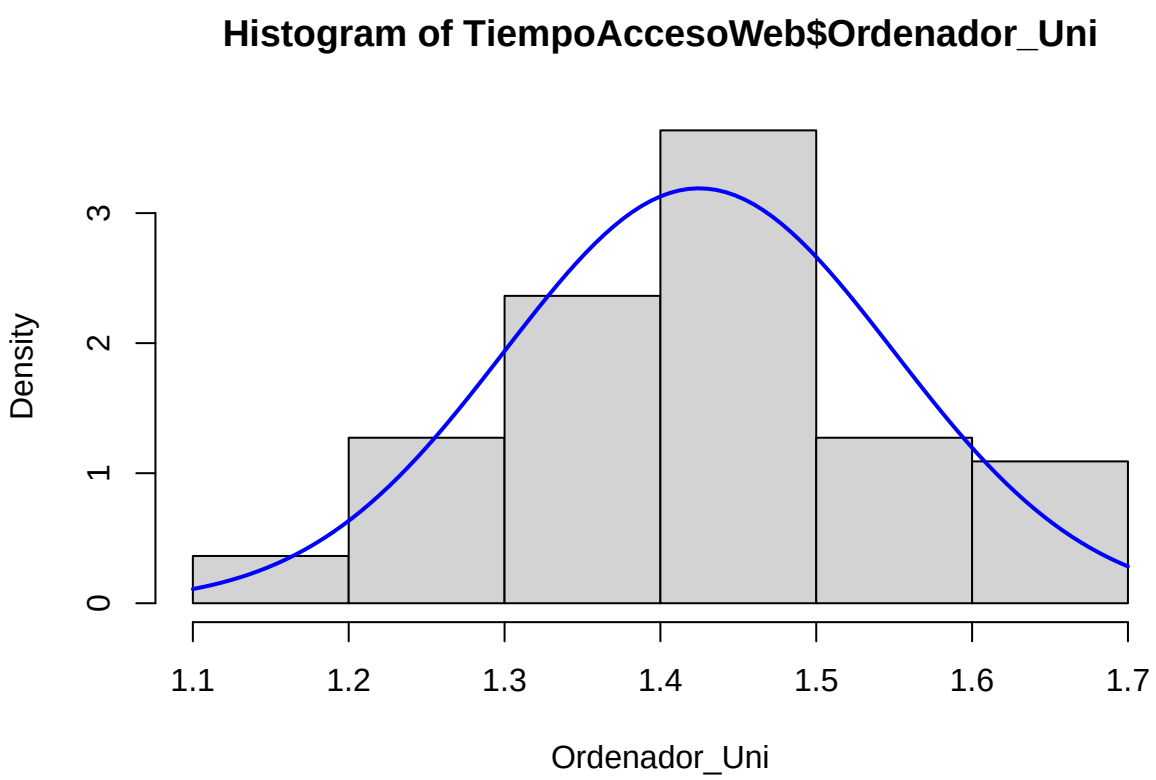
```
## # A tibble: 5 x 2
##   Ordenador_Casa Ordenador_Uni
##            <dbl>         <dbl>
## 1           6.15          1.16
```

```
## 2           5.83          1.42
## 3           5.72          1.39
## 4           6.22          1.41
## 5           5.72          1.44
```

```r
suppressWarnings(library(summarytools))
descr(TiempoAccesoWeb$Ordenador_Uni)
```

```
## Descriptive Statistics
## TiempoAccesoWeb$Ordenador_Uni
## N: 55
##
##                      Ordenador_Uni
## ----------------- ---------------
##              Mean            1.42
##           Std.Dev            0.13
##               Min            1.16
##                Q1            1.34
##            Median            1.42
##                Q3            1.50
##               Max            1.68
##               MAD            0.11
##               IQR            0.15
##                CV            0.09
##          Skewness            0.08
##       SE.Skewness            0.32
##          Kurtosis           -0.47
##           N.Valid           55.00
##         Pct.Valid          100.00
```

```r
hist(TiempoAccesoWeb$Ordenador_Uni,
     probability = TRUE, # histograma tiene area = 1
     xlab = "Ordenador_Uni")
curve(dnorm(x, mean(TiempoAccesoWeb$Ordenador_Uni), sd(TiempoAccesoWeb$Ordenador_Uni)),
      col="blue", lwd=2, add=TRUE, yaxt="n")
```



Podemos apreciar que el histograma se parece a la función de densidad Normal. De hecho, es unimodal y bastante simétrico (`Skewness = 0.08`) aunque su campana no es exactamente como la de Gauss (`Kurtosis = -0.29`). De esto podemos deducir que una distribución normal podría ajustarse bien a nuestros datos y, por lo tanto, podría ser un buen modelo para la población que estamos estudiando.

3.1.2 Diagnóstico del modelo elegido

Para evaluar la bondad del modelo ajustado podemos usar el contraste Chi-cuadrado. Debemos recordar que el estadístico del contraste Chi-cuadrado es una medida de discrepancia entre el número de observaciones observadas y esperadas en una partición dada

$$\sum_{i=1}^{k} \frac{(O_i - E_i)^2}{E_i},$$

donde k es el número de intervalos o celdas en la partición, O_i es el número de observaciones que se encuentran en la celda i-ésima y E_i es el número esperado de observaciones en la misma celda.

Primero, debemos construir una partición de $\mathbb{R}$ y contar cuántos valores de `Ordenador_Uni` caen en cada intervalo de la partición. Una manera fácil es usar la partición obtenida por la función `hist`

```
Partition <- hist(TiempoAccesoWeb$Ordenador_Uni, plot = FALSE)
Partition
```

```
## $breaks
## [1] 1.1 1.2 1.3 1.4 1.5 1.6 1.7
```

```
##
## $counts
## [1]  2  7 13 20  7  6
##
## $density
## [1] 0.3636364 1.2727273 2.3636364 3.6363636 1.2727273 1.0909091
##
## $mids
## [1] 1.15 1.25 1.35 1.45 1.55 1.65
##
## $xname
## [1] "TiempoAccesoWeb$Ordenador_Uni"
##
## $equidist
## [1] TRUE
##
## attr(,"class")
## [1] "histogram"
```

El componente `breaks` de `Partition` da los puntos que definen los intervalos en el histograma. Es decir, los seis intervalos en la partición son $(1.1, 1.2]$, $(1.2, 1.3]$, $(1.3, 1.4]$, $(1.4, 1.5]$, $(1.5, 1.6]$ and $(1, 6, 1.7]$. El componente `counts` da el número de observaciones dentro de cada intervalo o celda. Estos son los **observados**, O_i.

Cabe señalar que la partición anterior no cubre todo $\mathbb{R}$ ya que los intervalos $(-\infty, 1.1]$ y $(1.7, +\infty)$ no se consideran. Asumiremos que el primer intervalo de la partición es $(-\infty, 1.2]$ y el último intervalo es $(1.6, +\infty)$.

A continuación, ajustamos el modelo normal a `Ordenador_Uni`

```
library(fitdistrplus)
normalfit <- fitdist(TiempoAccesoWeb$Ordenador_Uni, "norm")
normalfit
```

```
## Fitting of the distribution ' norm ' by maximum likelihood
## Parameters:
##       estimate Std. Error
## mean 1.4248182 0.01670598
## sd   0.1238948 0.01180945
```

Los parámetros estimados para la variable aleatoria Normal son en nuestro caso $\widehat{\mu} = 1.42481818$ y $\widehat{\sigma} = 0.12389484$ que son iguales a los valores correspondientes mostrados en el análisis descriptivo de la variable. Por lo tanto, el modelo ajustado es

$$X \sim \mathcal{N}(1.42481818, 0.12389484).$$

Finalmente, realizamos una prueba de diagnóstico para apreciar la bondad de nuestro ajuste. Debemos calcular el número esperado de observaciones bajo la distribución normal *ajustada*

```
CummulativeProbabilities = pnorm(c(-Inf, Partition$breaks[c(-1,-7)], Inf),
                      normalfit$estimate[1], normalfit$estimate[2])
Probabilities = diff(CummulativeProbabilities)
Expected = length(TiempoAccesoWeb$Ordenador_Uni)*Probabilities
chisq.test(Partition$counts, p = Probabilities)
```

```
## Warning in chisq.test(Partition$counts, p = Probabilities): Chi-squared
## approximation may be incorrect
```

```
##
##  Chi-squared test for given probabilities
##
## data:  Partition$counts
## X-squared = 2.625, df = 5, p-value = 0.7576
```

El resultado de la prueba de Chi-cuadrado se puede resumir en las siguientes tres cantidades

- El estadístico de contraste calculado, `X-cuadrado` $= \sum_{i=1}^{k} \frac{(o_i - e_i)^2}{e_i}$, donde o_i es el número de observaciones en la muestra que está en la celda i-ésima y e_i es el número esperado de observaciones en la misma celda.

  Este estadístico resume la relación entre el histograma y la curva continua de la función de densidad. Cuanto mayor es su valor, peor es la bondad del ajuste del modelo teórico elegido.

- `df` (grados de libertad), representa el parámetro de la distribución Chi-cuadrado seleccionada y se utiliza como punto de referencia para apreciar la calidad del ajuste.

  - Los grados de libertad en la función `chisq.test` se calculan como`df` $= k-1$ ya que no tiene en cuenta el número de parámetros estimados.

  - Los grados de libertad deben calcularse como `df` $= k - p - 1$, donde p es el número de parámetros desconocidos del modelo que se estiman utilizando la muestra de datos, en este caso es igual a 2 (la media y la varianza).

- `p-value` (p-valor) es la probabilidad de que el estadístico del contraste tome un valor mayor que `X-squared`. En este caso está dado por el valor del área de la cola derecha a partir de 2.625 calculada con la función de densidad de una distribución Chi-cuadrado con grados de libertad `df`.

  - Observe que `df = 5` corresponde al número de celdas menos uno, $k-1$, pero estimamos dos parámetros, por lo que debemos usar una distribución χ^2 con`df = 3`, $k - p - 1$.

```
pchisq(2.625, 3, lower.tail = FALSE)
```

```
## [1] 0.4531236
```

Es decir, el `p-value` correcto $= 0.4531236$.

Si el p-valor es menor que 0.05, suponemos que es bastante improbable obtener el valor resultante del estadístico del contraste si el modelo fuera bueno. Por lo tanto, concluimos que la prueba no es satisfactoria. Por otro lado, si el p-valor es mayor que 0.05, concluimos que el ajuste es relativamente bueno y que el modelo elegido puede considerarse razonable para representar a la población.

En nuestro caso, el p-valor es igual a 0.4637294 y, por lo tanto, concluimos que el modelo normal es un modelo razonable para representar a nuestra población.

3.1.3 Otros contrastes de bondad de ajuste de normalidad

El contraste chi-cuadrado generalmente no se recomienda para probar la hipótesis de la normalidad debido a que tiene una potencia inferior en comparación con otros contrastes. Hay muchas funciones en R para hacer diferentes contrastes de bondad de ajuste. Todos ellos pueden interpretarse mirando los p-valores de la misma manera que lo hicimos mirando el contraste Chi-cuadrado. En particular, el paquete `nortest` incluye los siguientes:

- `ad.test`: Contraste de Anderson-Darling
- `cvm.test`: Contraste de Cramer-von Mises
- `lillie.test`: Contraste de Kolmogorov-Smirnov-Lilliefors
- `pearson.test`: Contraste de chi-cuadrado de Pearson para normalidad
- `sf.test`: Contraste de Shapiro-Francia

Por ejemplo, es posible verificar que los valores p correspondientes a estas pruebas también sean mayores que 0.05, corroborando así nuestra selección del modelo Normal.

```
library(nortest)
ad.test(TiempoAccesoWeb$Ordenador_Uni)
```

```
##
##  Anderson-Darling normality test
##
## data:  TiempoAccesoWeb$Ordenador_Uni
## A = 0.4312, p-value = 0.2958
```

```
cvm.test(TiempoAccesoWeb$Ordenador_Uni)
```

```
##
##  Cramer-von Mises normality test
##
## data:  TiempoAccesoWeb$Ordenador_Uni
## W = 0.073781, p-value = 0.2447
```

```
lillie.test(TiempoAccesoWeb$Ordenador_Uni)
```

```
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  TiempoAccesoWeb$Ordenador_Uni
## D = 0.088043, p-value = 0.3582
```

```
pearson.test(TiempoAccesoWeb$Ordenador_Uni)
```

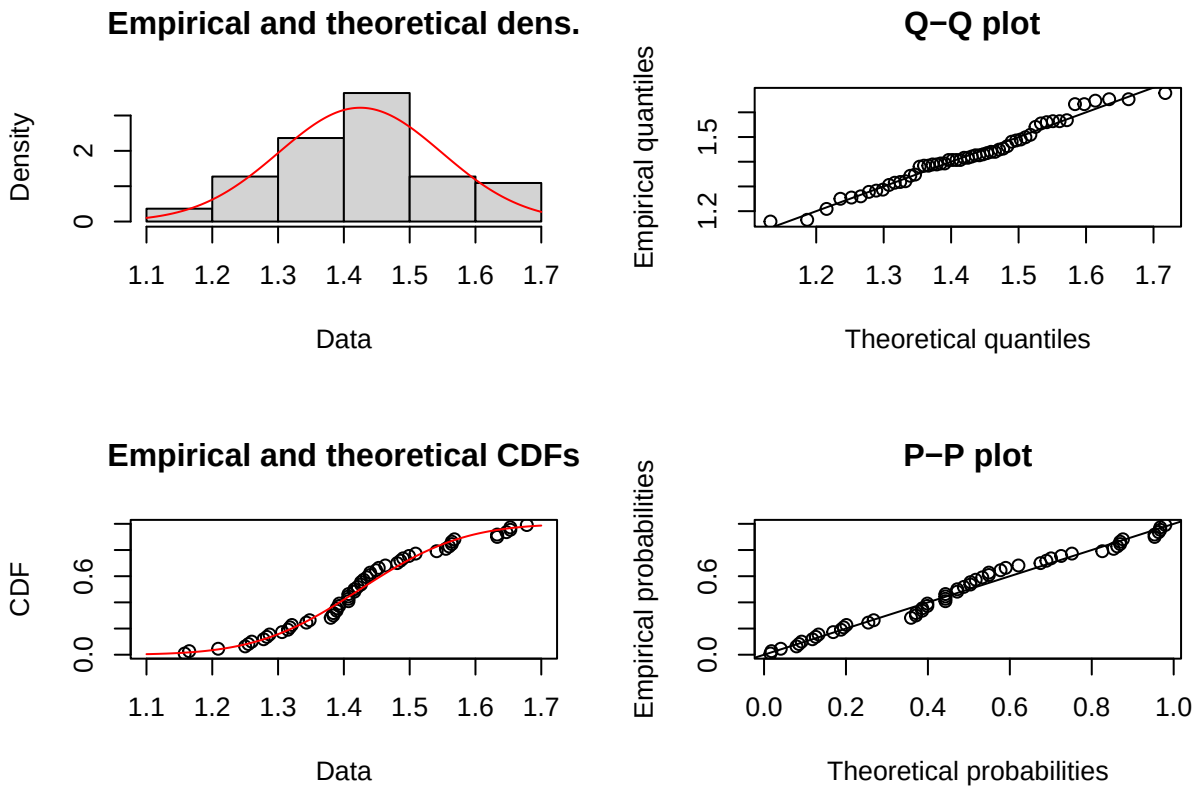
```
##
##  Pearson chi-square normality test
##
## data:  TiempoAccesoWeb$Ordenador_Uni
## P = 5.9091, p-value = 0.5504
```

```
sf.test(TiempoAccesoWeb$Ordenador_Uni)
```

```
##
##  Shapiro-Francia normality test
##
## data:  TiempoAccesoWeb$Ordenador_Uni
## W = 0.98159, p-value = 0.4749
```

Además, es posible obtener una representación gráfica del ajuste mediante

```
plot(normalfit)
```



3.2 Ajuste del Modelo. Variable `tiempo`

En esta sección repetimos el análisis anterior para la variable `tiempo` en el archivo `AlumnosIndustriales.xlsx`. Esta variable contiene mediciones del tiempo que invierten un grupo de estudiantes para llegar a la Universidad. El tamaño de la muestra es igual a 95.

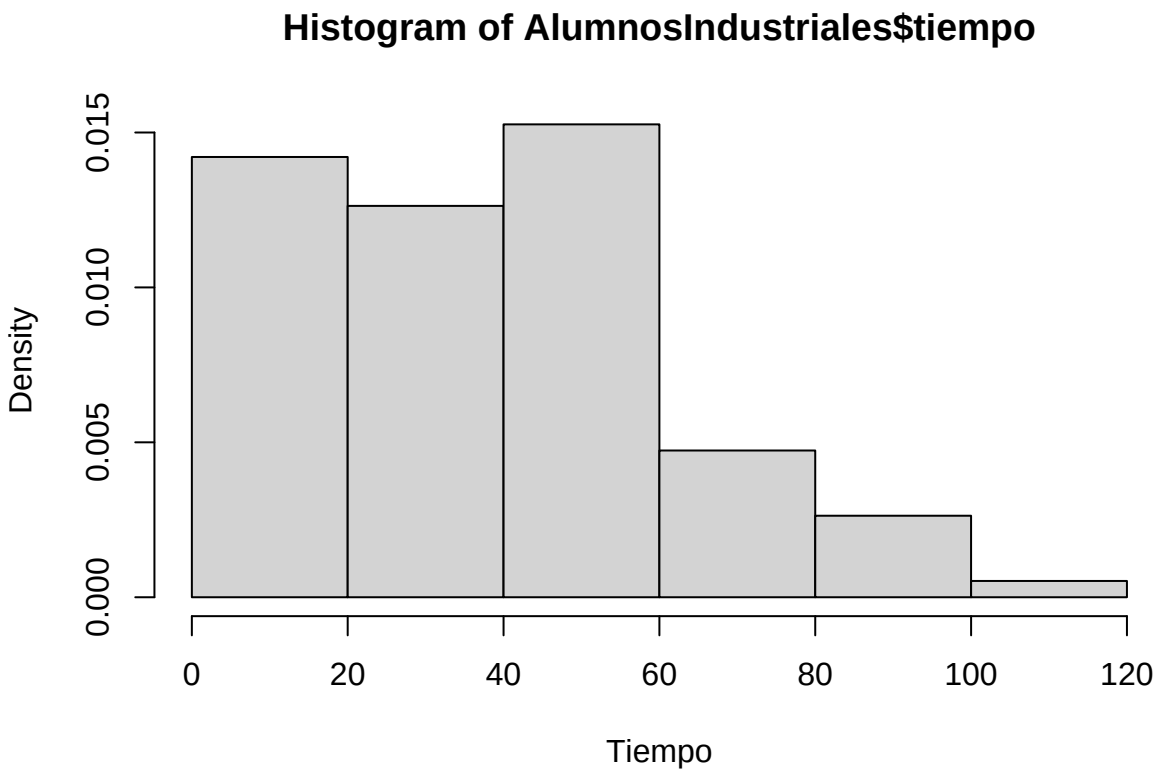
3.2.1 Análisis descriptivo de datos

Después de cargar el archivo `AlumnosIndustriales.xlsx`, realizamos el análisis descriptivo de la variable `tiempo` (calculando las medidas características e inspeccionando el histograma).

```r
suppressWarnings(library(summarytools))
descr(AlumnosIndustriales$tiempo)
```

```
## Descriptive Statistics
## AlumnosIndustriales$tiempo
## N: 95
##
##                      tiempo
## ----------------- --------
##              Mean    41.42
##           Std.Dev    24.74
##               Min     1.00
##                Q1    20.00
##            Median    40.00
##                Q3    60.00
##               Max   120.00
##               MAD    29.65
##               IQR    40.00
##                CV     0.60
##          Skewness     0.63
##       SE.Skewness     0.25
##          Kurtosis    -0.04
##           N.Valid    95.00
##         Pct.Valid   100.00
```

```r
hist(AlumnosIndustriales$tiempo,
     probability = TRUE, # histograma tiene area = 1
     xlab = "Tiempo")
```



Los datos parecen unimodales y con asimetría positiva. Tenemos dos opciones para ajustar un modelo a estos datos. Primero intentamos ajustar un modelo que tenga asimetría positiva, como por ejemplo la distribución de Weibull o la distribución Lognormal. A continuación, intentaremos realizar una transformación de los datos para corregir la asimetría e intentar ajustar una distribución Normal. Por ejemplo, podríamos intentar aplicar la operación de raíz cuadrada (tenga en cuenta que ajustar una Normal al logaritmo de una variable es lo mismo que ajustar una distribución Lognormal a la variable sin transformación).

3.2.2 Ajuste de una distribución Weibull

Como en el ejemplo anterior, ajustamos el modelo

```
library(fitdistrplus)
weibullfit <- fitdist(AlumnosIndustriales$tiempo, "weibull")
weibullfit
```

```
## Fitting of the distribution ' weibull ' by maximum likelihood
## Parameters:
##        estimate Std. Error
## shape  1.708639  0.1393375
## scale 46.341096  2.9242445
```

Ahora, obtendremos el número observado y esperado de observaciones en los intervalos definidos por el histograma predeterminado.

```r
Partition <- hist(AlumnosIndustriales$tiempo, plot = FALSE)
Partition
```

```
## $breaks
## [1]   0  20  40  60  80 100 120
##
## $counts
## [1] 27 24 29  9  5  1
##
## $density
## [1] 0.0142105263 0.0126315789 0.0152631579 0.0047368421 0.0026315789
## [6] 0.0005263158
##
## $mids
## [1]  10  30  50  70  90 110
##
## $xname
## [1] "AlumnosIndustriales$tiempo"
##
## $equidist
## [1] TRUE
##
## attr(,"class")
## [1] "histogram"
```

```r
CummulativeProbabilities = pweibull(c(Partition$breaks[-7], Inf),
                      weibullfit$estimate[1], weibullfit$estimate[2])
Probabilities = diff(CummulativeProbabilities)
Expected = length(AlumnosIndustriales$tiempo)*Probabilities
chisq.test(Partition$counts, p = Probabilities)
```

```
## Warning in chisq.test(Partition$counts, p = Probabilities): Chi-squared
## approximation may be incorrect
```

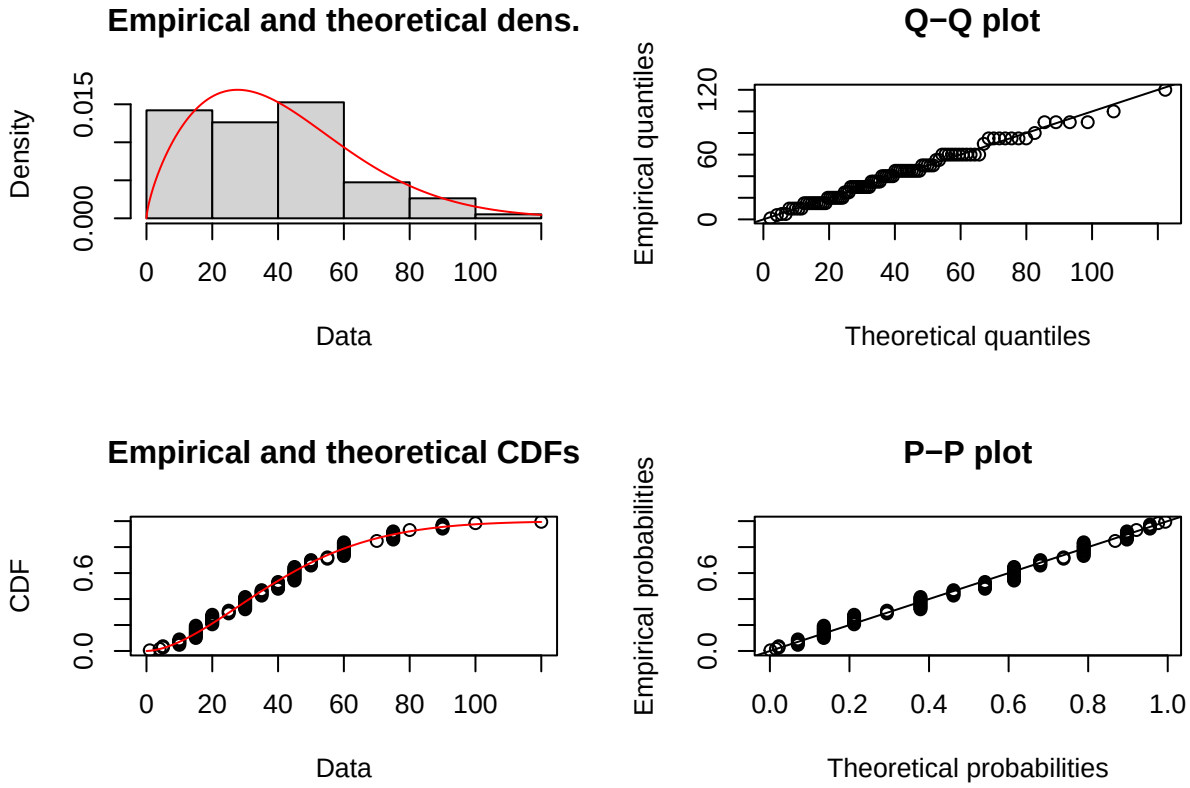
```
##
##  Chi-squared test for given probabilities
##
## data:  Partition$counts
## X-squared = 7.0387, df = 5, p-value = 0.2178
```

Aquí, nuevamente, debemos volver a calcular el p-valor ya que estimamos los dos parámetros de la distribución de Weibull.

```r
pchisq(7.0387, 3, lower.tail = FALSE)
```

```
## [1] 0.07067445
```

```r
plot(weibullfit)
```



Al comparar el histograma con la función de densidad de Weibull y al observar el p-valor, nos damos cuenta de que el ajuste es satisfactorio. Esto significa que podríamos usar el modelo de probabilidad de Weibull para describir el tiempo que los estudiantes invierten para llegar a la Universidad.

3.2.3 Ajuste de una distribución Lognormal

Procedemos como antes: (i) ajuste del modelo; (ii) cálculo del número observado y esperado de observaciones en cada intervalo del histograma y (iii) contraste Chi-cuadrado.

```
library(fitdistrplus)
lognormalfit <- fitdist(AlumnosIndustriales$tiempo, "lnorm")
lognormalfit
```

```
## Fitting of the distribution ' lnorm ' by maximum likelihood
## Parameters:
##          estimate Std. Error
## meanlog 3.4891976 0.08090337
## sdlog   0.7885485 0.05720691
```

```
Partition <- hist(AlumnosIndustriales$tiempo, plot = FALSE)
Partition
```

```
## $breaks
## [1]   0  20  40  60  80 100 120
```

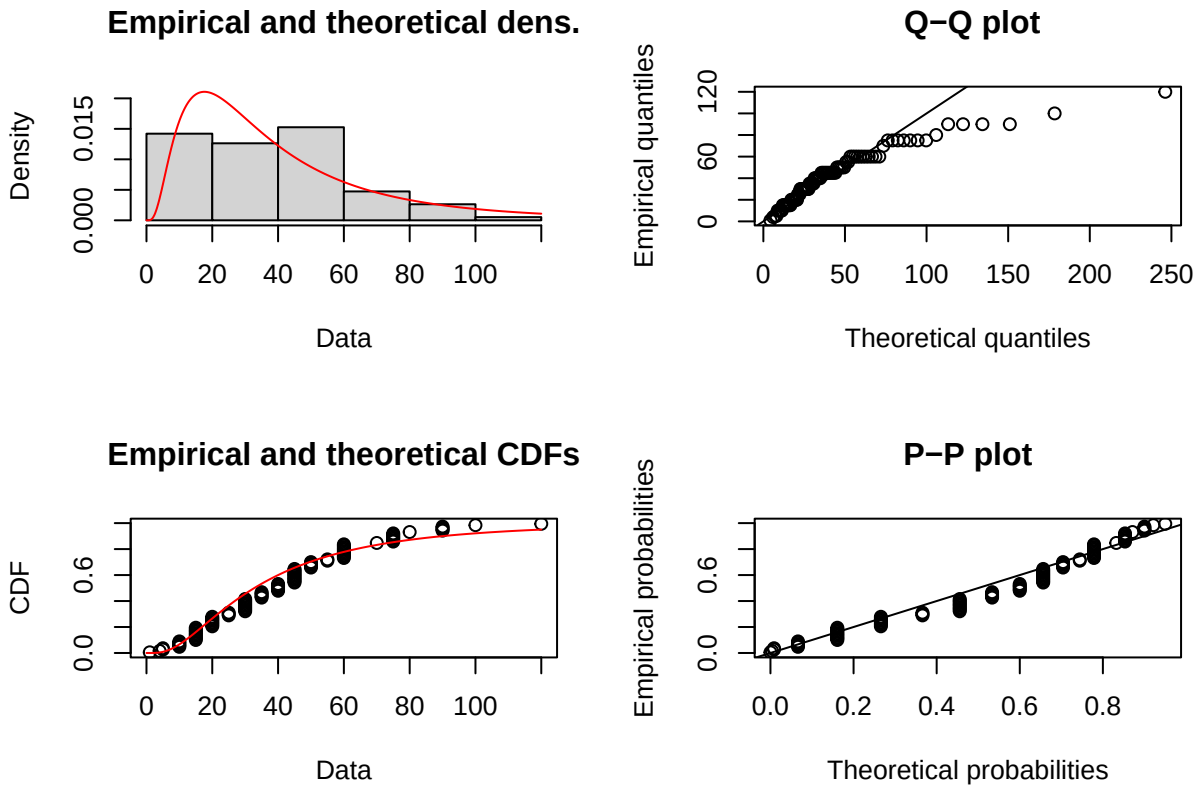
```
##
## $counts
## [1] 27 24 29  9  5  1
##
## $density
## [1] 0.0142105263 0.0126315789 0.0152631579 0.0047368421 0.0026315789
## [6] 0.0005263158
##
## $mids
## [1]  10  30  50  70  90 110
##
## $xname
## [1] "AlumnosIndustriales$tiempo"
##
## $equidist
## [1] TRUE
##
## attr(,"class")
## [1] "histogram"
```

```
CummulativeProbabilities = plnorm(c(Partition$breaks[-7], Inf),
                      lognormalfit$estimate[1], lognormalfit$estimate[2])
Probabilities = diff(CummulativeProbabilities)
Expected = length(AlumnosIndustriales$tiempo)*Probabilities
chisq.test(Partition$counts, p = Probabilities)
```

```
##
##  Chi-squared test for given probabilities
##
## data:  Partition$counts
## X-squared = 16.15, df = 5, p-value = 0.00643
```

Parece claro que este ajuste no es tan bueno como el anterior. El p-valor obtenido por el contraste Chi-cuadrado es muy bajo. De hecho, el `p-value` es más pequeño ya que deberíamos usar `pchisq (16.15, 3, lower.tail = FALSE)`.

```
plot(lognormalfit)
```



El histograma nos da la razón del mal ajuste; de hecho, la distribución Lognormal tiene una curtosis más alta que el conjunto de datos. En conclusión, el modelo Lognormal no es adecuado para representar nuestros datos.

3.2.4 Ajuste de una distribución normal a una transformación del conjunto de datos

La variable `tiempo` es asimétrica positiva, sin embargo, su raíz cuadrada parece bastante simétrica. Si ajustamos una distribución Normal a la raíz cuadrada de los datos, obtendremos los siguientes resultados:

```
library(fitdistrplus)
normalfit <- fitdistr(sqrt(AlumnosIndustriales$tiempo), "normal")
normalfit
```

```
##      mean         sd
##   6.1169314   2.0010506
##  (0.2053035) (0.1451715)
```

```
Partition <- hist(sqrt(AlumnosIndustriales$tiempo), plot = FALSE)
Partition
```

```
## $breaks
##  [1]  1  2  3  4  5  6  7  8  9 10 11
##
```

```
## $counts
##  [1]  2  2 15 11 15 17 18  9  5  1
##
## $density
##  [1] 0.02105263 0.02105263 0.15789474 0.11578947 0.15789474 0.17894737
##  [7] 0.18947368 0.09473684 0.05263158 0.01052632
##
## $mids
##  [1]  1.5  2.5  3.5  4.5  5.5  6.5  7.5  8.5  9.5 10.5
##
## $xname
## [1] "sqrt(AlumnosIndustriales$tiempo)"
##
## $equidist
## [1] TRUE
##
## attr(,"class")
## [1] "histogram"
```

```
CummulativeProbabilities = pnorm(c(-Inf, Partition$breaks[c(-1,-11)], Inf),
                      normalfit$estimate[1], normalfit$estimate[2])
Probabilities = diff(CummulativeProbabilities)
Expected = length(AlumnosIndustriales$tiempo)*Probabilities
chisq.test(Partition$counts, p = Probabilities)
```

```
##
##  Chi-squared test for given probabilities
##
## data:  Partition$counts
## X-squared = 9.3823, df = 9, p-value = 0.4028
```

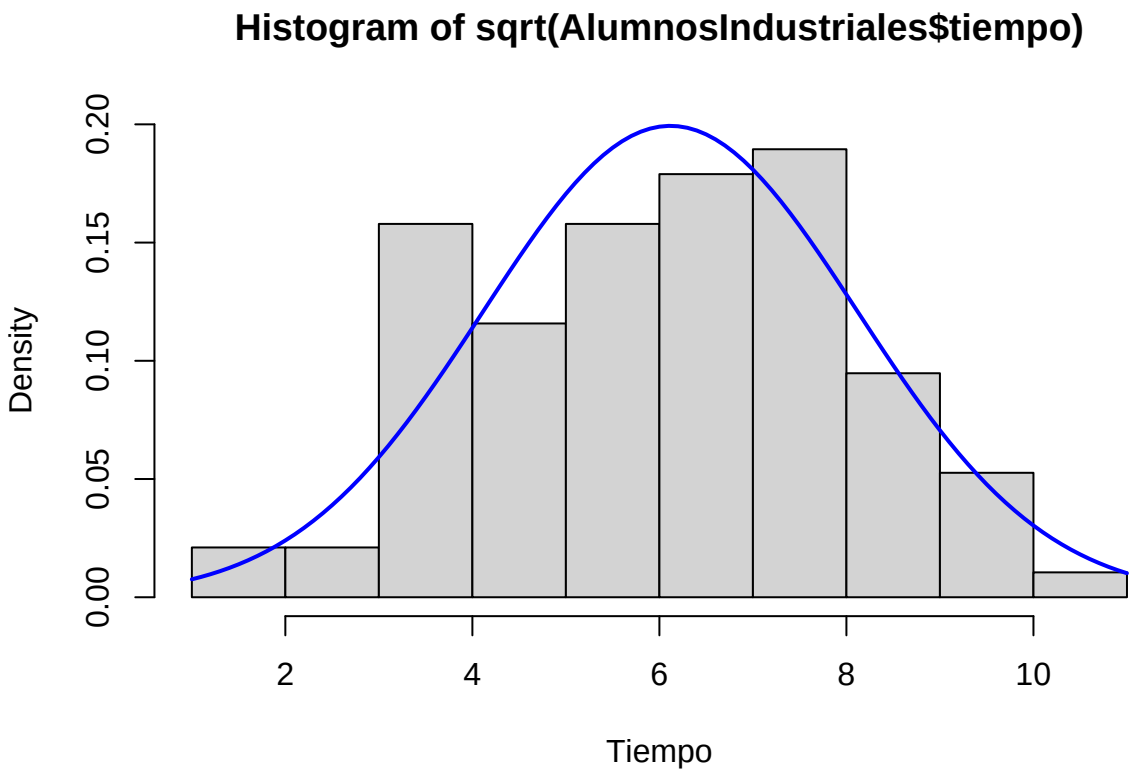
El `p-valor` teniendo en cuenta que se estimaron dos parámetros es

```
pchisq(9.3823, 7, lower.tail = FALSE)
```

```
## [1] 0.226361
```

que es mayor que 0.05.

```
hist(sqrt(AlumnosIndustriales$tiempo),
          probability = TRUE, # histograma tiene area = 1
     xlab = "Tiempo", ylim = c(0,0.2))
curve(dnorm(x, normalfit$estimate[1], normalfit$estimate[2]),
      col="blue", lwd=2, add=TRUE, yaxt="n")
```



El ajuste se ve casi tan bueno como el que se hace usando la distribución Weibull.

Podemos verificar los resultados anteriores mediante los contrastes de normalidad mencionados en la sección 3.1.3:

```
library(nortest)
ad.test(sqrt(AlumnosIndustriales$tiempo))
```

```
##
##  Anderson-Darling normality test
##
## data:  sqrt(AlumnosIndustriales$tiempo)
## A = 0.52436, p-value = 0.1773
```

```
cvm.test(sqrt(AlumnosIndustriales$tiempo))
```

```
##
##  Cramer-von Mises normality test
##
## data:  sqrt(AlumnosIndustriales$tiempo)
## W = 0.086902, p-value = 0.1664
```

```
lillie.test(sqrt(AlumnosIndustriales$tiempo))
```

```
##
```

```
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  sqrt(AlumnosIndustriales$tiempo)
## D = 0.078749, p-value = 0.1562
```

```
pearson.test(sqrt(AlumnosIndustriales$tiempo), n.classes = 10)
```

```
##
##  Pearson chi-square normality test
##
## data:  sqrt(AlumnosIndustriales$tiempo)
## P = 10.368, p-value = 0.1686
```

```
sf.test(sqrt(AlumnosIndustriales$tiempo))
```

```
##
##  Shapiro-Francia normality test
##
## data:  sqrt(AlumnosIndustriales$tiempo)
## W = 0.98791, p-value = 0.458
```