

Diagnóstico del modelo - Contrastes de bondad de ajuste

Grado en Ingeniería Informática

2022/23

1. Introducción

El objetivo de esta práctica es asignar un modelo de probabilidad a un conjunto de datos de una muestra de tal manera que el modelo elegido pueda representar a la población de la que se tomaron los datos. La tarea de buscar el modelo adecuado se denomina **ajuste de distribución**. Para seleccionar un buen modelo de probabilidad para un conjunto de datos dado, es necesario realizar pruebas estadísticas. La tarea de ejecutar estas pruebas se llama **diagnóstico del modelo**. Por lo tanto, diremos que un modelo **se ajusta bien** a los datos si nuestra muestra de datos pasará positivamente las pruebas del **diagnóstico**.

La forma habitual de realizar el ajuste de distribución es la siguiente. Partimos de una muestra de datos y comparamos su distribución empírica con la de los modelos conocidos (Normal, Poisson, Exponencial, etc.). Para evaluar la bondad de ajuste de un modelo utilizaremos el contraste Chi-cuadrado.

A continuación utilizaremos los datos contenidos en el archivo `TiempoaccesoWeb.xlsx`. Comenzamos analizando 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 una computadora 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.

2. Ajuste del Modelo. Variable `Ordenador_Uni`

2.1 Análisis descriptivo de los datos

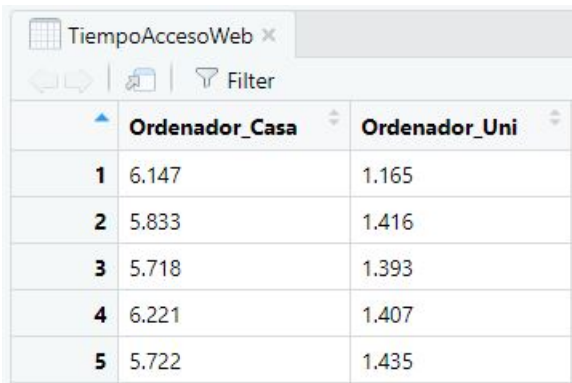
Lo primero que se debe hacer es el análisis descriptivo de los datos (calcular las medidas características e inspeccionar el histograma). De esta manera, podríamos tener una primera idea de qué modelo usar.

Primero leemos y vemos el archivo de datos. La figura muestra las primeras cinco observaciones de este archivo de datos. Tenga en cuenta que la línea `View(TiempoAccesoWeb)` aparece como un comentario, para ejecutarla, simplemente elimine el símbolo `#`.

```
library(readxl)
```

```
## Warning: package 'readxl' was built under R version 4.2.2
```

```
TiempoAccesoWeb <- read_excel("TiempoAccesoWeb.xlsx")
#View(TiempoAccesoWeb)
```

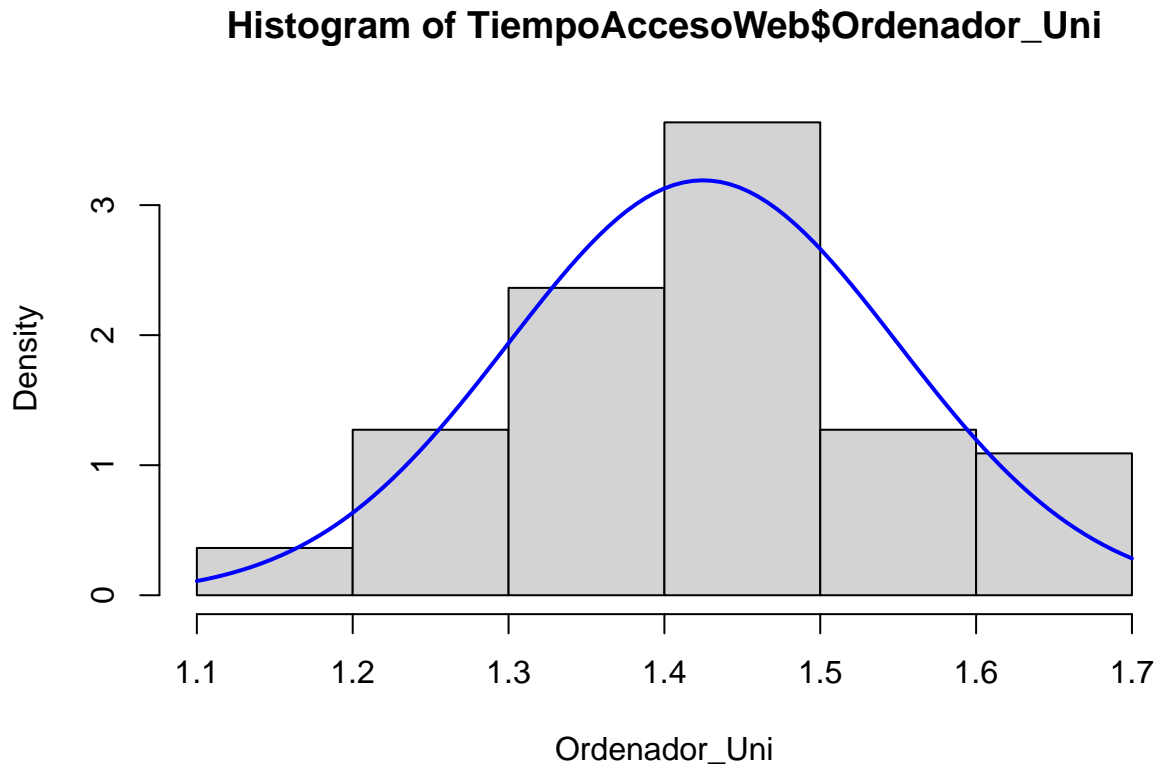


	Ordenador_Casa	Ordenador_Uni
1	6.147	1.165
2	5.833	1.416
3	5.718	1.393
4	6.221	1.407
5	5.722	1.435

```
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
```

```
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.

2.2 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 $\hat{\mu} = 1.42481818$ y $\hat{\sigma} = 0.12389484$ 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, $X\text{-cuadrado} = \sum_{i=1}^k \frac{(o_i - e_i)^2}{e_i}$, 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.

2.3 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)
```

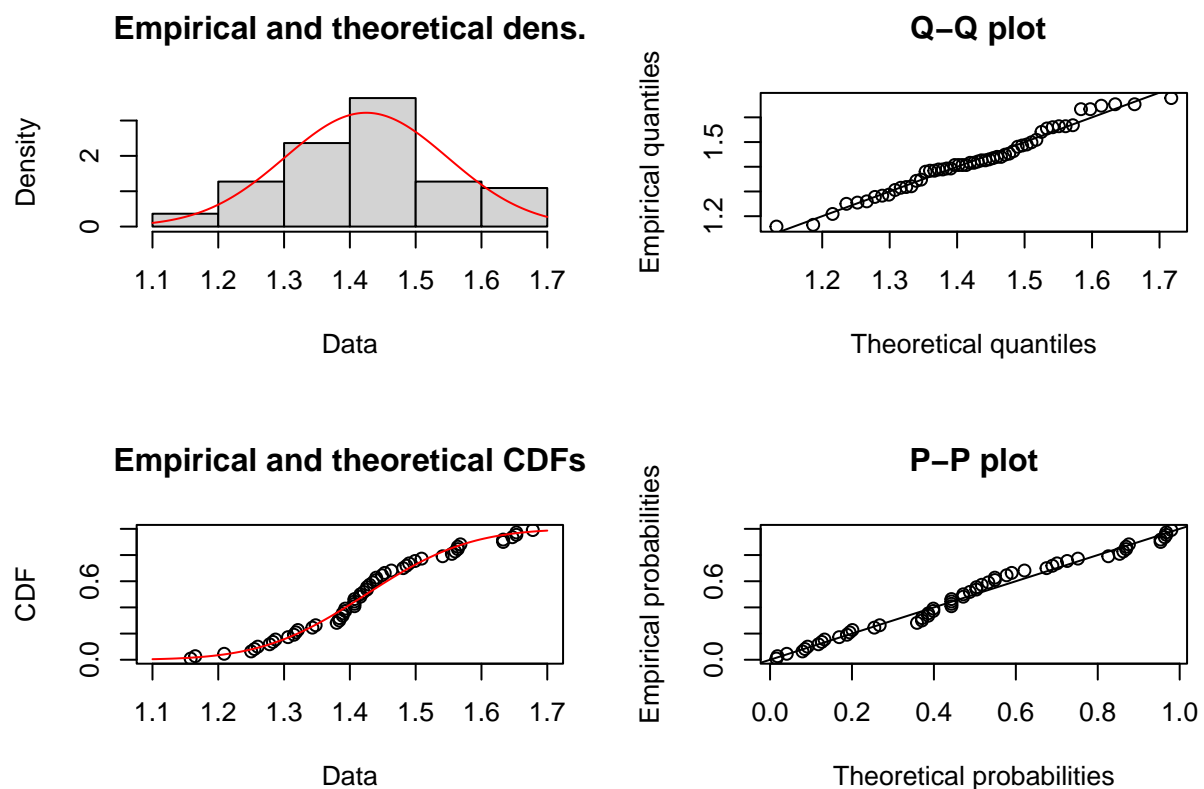
```
##
## Pearson chi-square normality test
##
## data:  TiempoAccesoWeb$Ordenador_Uni
## P = 5.9091, p-value = 0.5504
```

```
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. Ajuste del modelo para la 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.

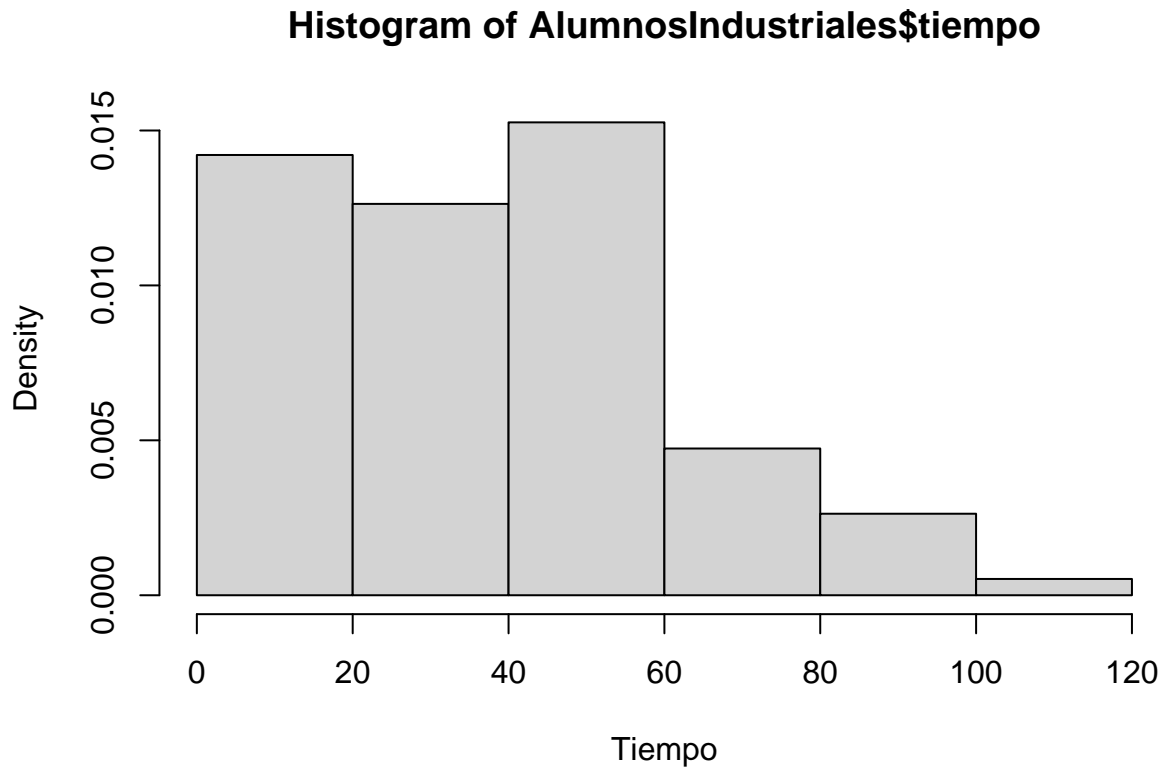
3.1 Análisis descriptivo de datos

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

```
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
```

```
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 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.

```
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 = 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
```

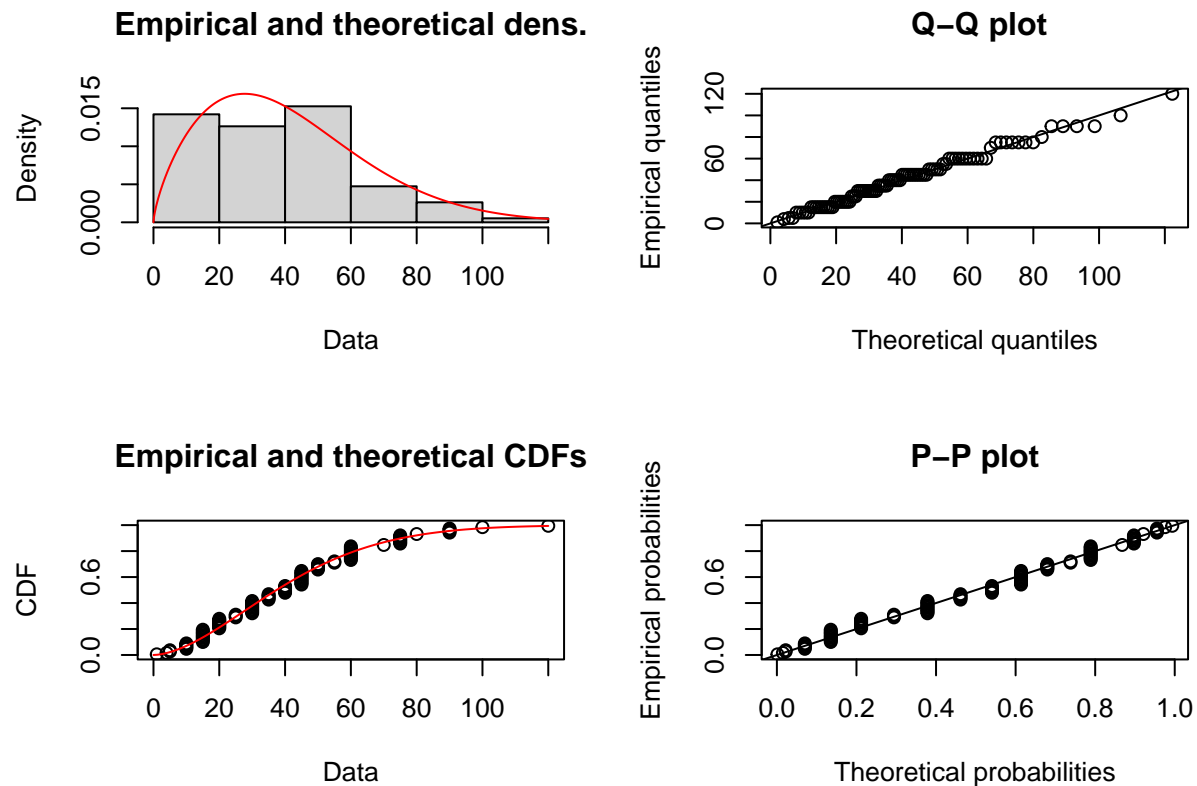
```
##
## Chi-squared test for given probabilities
##
## data: Partition$counts
## X-squared = 7.0387, df = 5, p-value = 0.2178
```

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

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

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

```
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.

3.3 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
```

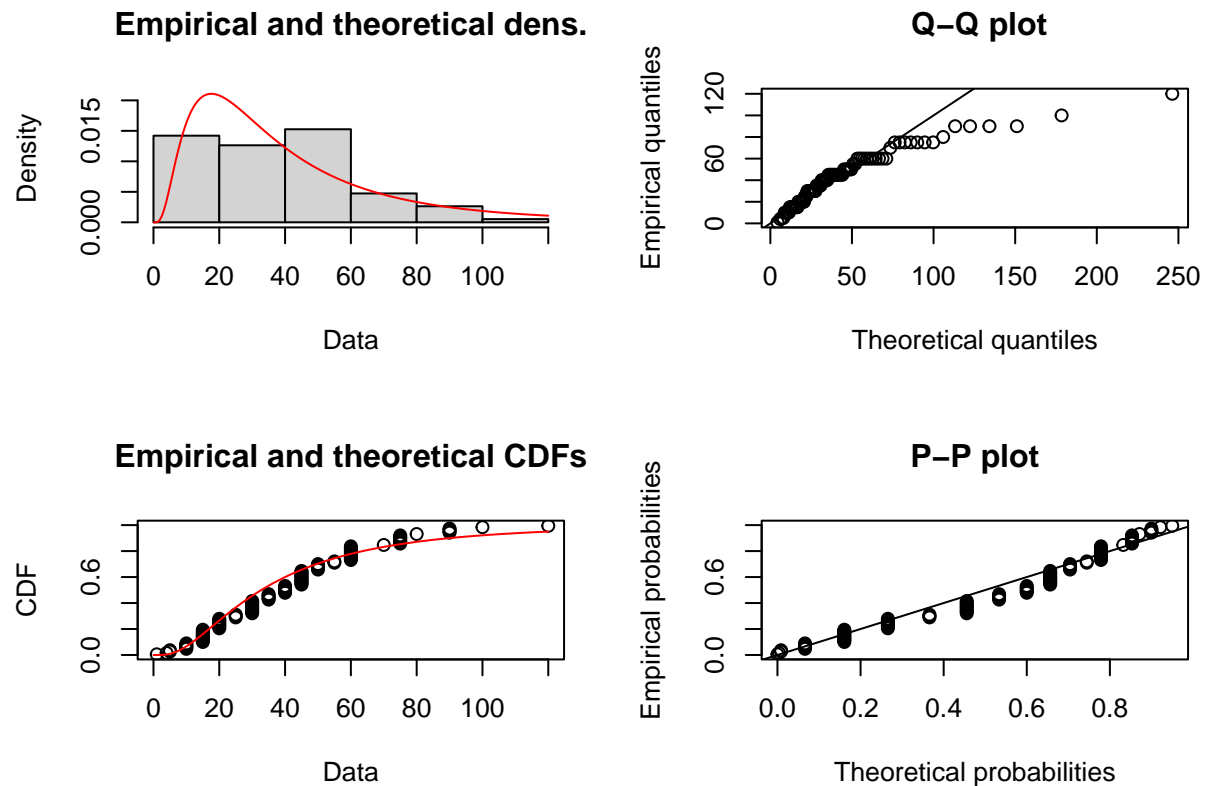
```
##
## $counts
## [1] 27 24 29 9 5 1
##
## $density
## [1] 0.0142105263 0.0126315789 0.0152631579 0.0047368421 0.0026315789
## [6] 0.0005263158
##
## $mids
## [1] 10 30 50 70 90 110
##
## $xname
## [1] "AlumnosIndustriales$tiempo"
##
## $equidist
## [1] TRUE
##
## attr("class")
## [1] "histogram"

CumulativeProbabilities = plnorm(c(Partition$breaks[-7], Inf),
                                lognormalfit$estimate[1], lognormalfit$estimate[2])
Probabilities = diff(CumulativeProbabilities)
Expected = length(AlumnosIndustriales$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.

3.4 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
```

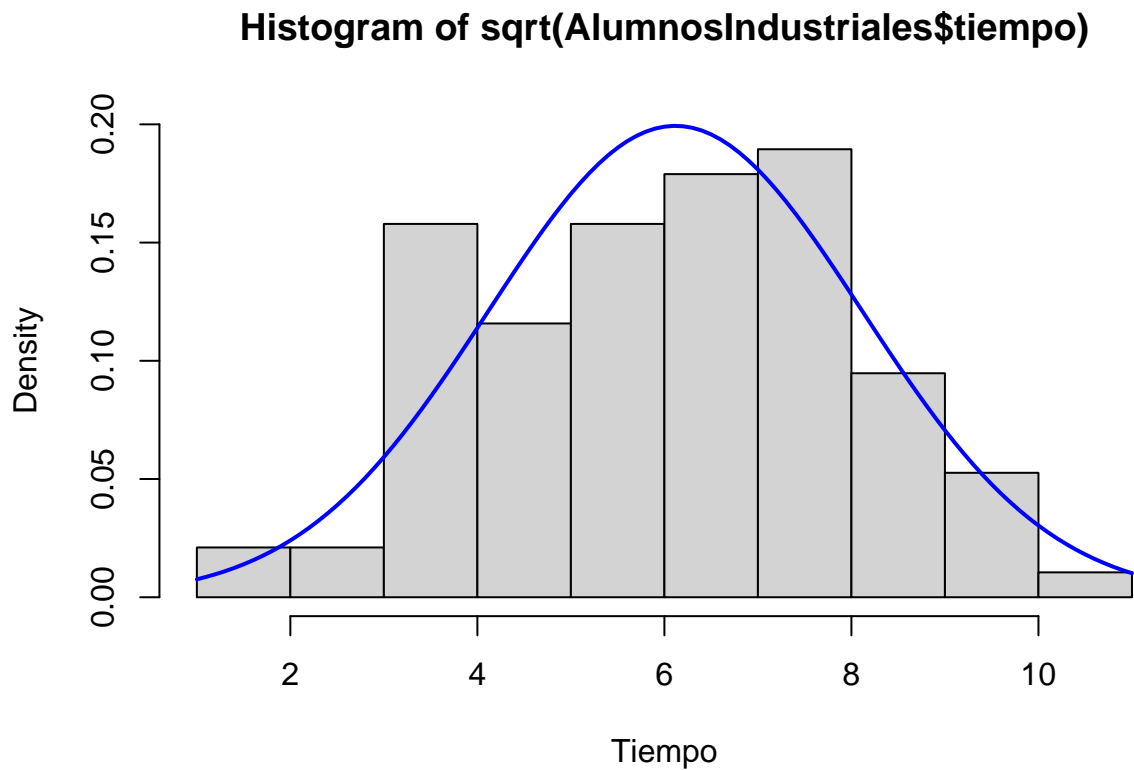
El p-valor teniendo en cuenta que se estimaron dos parámetros es

```
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 2.3:

```
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
```

4. Ejemplo de una aplicación del contraste de bondad de ajuste

Es muy útil tener un buen modelo que represente la población de la que podemos haber obtenido una muestra de datos. Permite, entre otras cosas, calcular las probabilidades de eventos de una manera mucho más precisa que usar las frecuencias relativas observadas del conjunto de datos de la muestra.

En este ejemplo, calculamos la probabilidad de que un estudiante viva a una distancia de más de una hora de la Universidad. Podemos hacer esto usando el modelo de Weibull, así como también usando el modelo Normal aplicado a la raíz cuadrada de la variable `tiempo`. Estos dos modelos nos darán dos resultados diferentes, sin embargo, esperamos que estén muy cerca el uno del otro.

4.1 Cálculo usando el modelo Weibull

Como hemos visto anteriormente, el modelo Weibull que mejor se ajusta a nuestros datos tiene los siguientes parámetros: `shape = 1.7088275` y `scale = 46.3508101`. Luego, podemos calcular la probabilidad requerida, $\Pr(Tiempo > 60)$, por

```
pweibull(60, shape = 1.7088275, scale = 46.3508101, lower.tail = FALSE)
```

```
## [1] 0.2113264
```

Podemos concluir que la probabilidad de que un estudiante viva a una distancia de más de una hora de la Universidad es aproximadamente igual a 0.211.

4.2 Cálculo utilizando el modelo Normal aplicado a la raíz cuadrada de la variable

Como se ve arriba, la raíz cuadrada puede ajustarse bien a una distribución Normal. Para calcular la probabilidad de que el estudiante tarde más de 60 minutos en llegar a la Universidad, es equivalente a calcular la probabilidad de que la raíz cuadrada del tiempo empleado sea mayor que $\sqrt{60} = 7.745967$ (medido como raíz cuadrada de minutos). La distribución normal que mejor se ajusta a nuestros datos tiene los siguientes parámetros estimados: `mean = 6.1169314` y `sd = 2.0010506`.

Entonces podemos calcular la probabilidad requerida para esta distribución por

```
pnorm(sqrt(60), mean = 6.1169314, sd = 2.0010506, lower.tail = FALSE)
```

```
## [1] 0.2077967
```

Por lo tanto, al usar este modelo, la probabilidad de que un estudiante viva a una distancia de más de una hora de la Universidad es aproximadamente igual a 0.208, y está muy cerca de la calculada usando el modelo de Weibull.

El siguiente gráfico proporciona una comparación de la función de distribución estimada utilizando el modelo de Weibull (en rojo) y el modelo Normal (en azul). Está claro que ambos modelos son muy similares, y se ajustan razonablemente a la función de distribución empírica.

```
plot(ecdf(AlumnosIndustriales$tiempo))  
lines(0:130, pweibull(0:130, shape = 1.7088275, scale = 46.3508101), col="red")  
lines(0:130, pnorm(sqrt(0:130), mean = 6.1169314, sd = 2.0010506), col="blue")
```

