Bondad de Ajuste

Luis Gerardo Guzmán Rojas

30/11/2020

Contraste K-S en R

La función básica para realizar el test K-S es `ks.test`. Su sintaxis básica para una muestra es

```
ks.test(x, y, parámetros)
```

donde:

- `x` es la muestra de una variable continua.
- `y` puede ser un segundo vector, y entonces se contrasta si ambos vectores han sido generados por la misma distribución continua, o el nombre de la función de distribución (empezando con `p`) que queremos contrastar, entre comillas; por ejemplo `"pnorm"` para la distribución normal.
- Los `parámetros` de la función de distribución si se ha especificado una; por ejemplo `mean=0, sd=1` para una distribución normal estándar.

Ejemplo en R

Para realizar el contraste K-S para nuestro ejemplo, tenemos que hacer lo siguiente:

```
muestra=iris$Sepal.Width
ks.test(muestra,"pnorm",mean=3,sd=1.5)
```

```
## Warning in ks.test(muestra, "pnorm", mean = 3, sd = 1.5): ties should not be
## present for the Kolmogorov-Smirnov test
```

```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  muestra
## D = 0.29611, p-value = 7.54e-12
## alternative hypothesis: two-sided
```

Observamos que nos da el mismo valor de **discrepancia máxima** que hemos obtenido anteriormente con un p-valor altísimo, hecho que corrobora la conclusión que hemos escrito.

Test de normalidad

Test de Kolmogorov-Smirnov-Lilliefors (K-S-L)

Para contrastar si una muestra proviene de una distribución normal con los parámetros μ y σ desconocidos, el **test K-S** nos "obliga" a darles un valor.

Los valores "óptimos" para dichos parámetros serían las estimaciones dadas por los **estimadores de máxima verosimilitud**: la **media muestral** para μ y la **desviación típica muestral** para sigma.

La **prueba de Kolmogorov-Smirnov-Lilliefors** consiste en estimar dichos parámetros, calcular la **discrepancia máxima** tal como hemos explicado pero a la hora de calcular el p-valor, se usa otra distribución, llamada **distribución de Lilliefors** en lugar de usar la **distribución de Kolmogorov** ya que con la **distribución de Lilliefors**, el contraste es más robusto.

Test K-S-L en `R`

Vamos a aplicar el **test K-S-L** a nuestra muestra anterior para ver si se distribuye según la ley normal.

El **test K-S-L** test en `R` se aplica usando la función `lillie.test` del paquete `nortest`:

```
library(nortest)
lillie.test(muestra)
```

```
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  muestra
## D = 0.10566, p-value = 0.0003142
```

Vemos que el p-valor no es tan grande como en el caso anterior pero aún así, es suficientemente grande para concluir que no tenemos indicios suficientes para rechazar que nuestra muestra siga la distribución normal.

Test K-S-L

El test K-S-L tiene un inconveniente: aunque es muy sensible a las diferencias entre la muestra y la distribución teórica alrededor de sus valores medios, le cuesta detectar diferencias prominentes en un extremo u otro de la distribución.

Su potencia se ve afectada por dicho inconveniente.

Test K-S-L

Veamos un ejemplo de este hecho intentando ver si una muestra de una distribución t de Student nos acepta que es normal o no:

```
set.seed(100)
x=rt(50,3)
lillie.test(x)
```

```
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  x
## D = 0.10332, p-value = 0.2013
```

Test K-S-L

Nos dice que no podemos rechazar que la muestra `x` sea normal.

Esto es debido a que la función de densidad de la distribución t de Student es algo más aplanada que la distribución normal, donde en los dos extremos está por encima de la de la normal.

Como el test K-S-L no detecta las diferencias en los extremos, acepta que `x` es normal.

Test de normalidad de Anderson-Darling (A-D)

El **test de normalidad de Anderson-Darling** resuelve el inconveniente del **test de K-S-L**.

Este test está implementado en la función `ad.test` del paquete `nortest`.

Si ahora, aplicamos el **test A-D** a la muestra anterior de la distribución t de Student, la normalidad queda rechazada:

```
ad.test(x)
```

```
##
##  Anderson-Darling normality test
##
## data:  x
## A = 1.1657, p-value = 0.004334
```

Test de normalidad de Anderson-Darling (A-D)

Un inconveniente común a los **tests K-S-L y A-D** es que, si bien pueden usarse con muestras pequeñas (pongamos de más de 5 elementos), se comportan mal con muestras grandes, de varios miles de elementos.

En muestras de este tamaño, cualquier pequeña divergencia de la normalidad se magnifica y en estos dos tests aumenta la probabilidad de errores de tipo I.

Test de Shapiro-Wilks (S-W)

Un test que resuelve este problema es el de **Shapiro-Wilk** (**S-W**), implementado en la función `shapiro.test` de la instalación básica de `R`.

Apliquemos el **test S-W** a las dos muestras anteriores: la muestra del ejemplo y la muestra de la distribución t de Student:

Test de Shapiro-Wilks (S-W)

```r
shapiro.test(muestra)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  muestra
## W = 0.98492, p-value = 0.1012
```

```r
shapiro.test(x)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  x
## W = 0.89494, p-value = 0.0003285
```

Test de Shapiro-Wilks (S-W)

Vemos que acepta que la muestra del ejemplo anterior sea normal y rechaza la normalidad en el caso de la muestra de la t de Student.

Si nuestra muestra de valores tiene empates, los p-valores de los contrastes calculados a partir de las distribuciones de los estadísticos usados en los tests **K-S-L**, **A-D** y **S-W** se pueden ver afectados hasta el punto de que, si hay muchos empates, su significado no tenga ningún sentido.

Hay que decir que el menos afectado por los empates es el test de **S-W**.

Test omnibus de D'Agostino-Pearson

Un test que no es sensible a los empates es el test de normalidad de **D'Agostino-Pearson**.

Este test se encuentra implementado en la función `dagoTest` del paquete **fBasics**, y lo que hace es cuantificar lo diferentes que son la asimetría y la curtosis de la muestra (dos parámetros estadísticos relacionados con la forma de la gráfica de la función de densidad muestral) respecto de los esperados en una distribución normal, y resume esta discrepancia en un p-valor con el significado usual.

Para poder aplicar dicho test, el tamaño de la muestra debe ser 20 como mínimo.

Por tanto, sólo podemos aplicar dicho test a la muestra de datos correspondiente a la distribución t de Student:

Test omnibus de D'Agostino-Pearson

```r
library(fBasics)
dagoTest(x)
```

```
##
## Title:
##  D'Agostino Normality Test
##
```

```
## Test Results:
##   STATISTIC:
##     Chi2 | Omnibus: 21.8125
##     Z3   | Skewness: 2.8069
##     Z4   | Kurtosis: 3.7328
##   P VALUE:
##     Omnibus  Test: 1.834e-05
##     Skewness Test: 0.005001
##     Kurtosis Test: 0.0001894
##
## Description:
##  Mon Nov 30 14:24:56 2020 by user: MI00780
```

Test omnibus de D'Agostino-Pearson

Si nos fijamos en el resultado, el test calcula tres estadísticos de contraste: el test **Omnibus** basado en la distribución χ^2, el test de asimetría y el test de curtosis con sus correspondientes p-valores. Para más información, id a https://en.wikipedia.org/wiki/D%27Agostino%27s_K-squared_test

Vemos que según el test de **D'Agostino-Pearson**, la muestra `x` correspondiente a la distribución t de Student no sigue la ley normal.

Guía rápida

- `qqPlot` del paquete **car**, sirve para dibujar un Q-Q-plot de una muestra contra una distribución teórica. Sus parámetros principales son:
  - `distribution`: el nombre de la familia de distribuciones, entre comillas.
  - Los parámetros de la distribución: `mean` para la media, `sd` para la desviación típica, `df` para los grados de libertad, etc.
  - Los parámetros usuales de `plot`.

Guía rápida

- `chisq.test` sirve para realizar tests χ^2 de bondad de ajuste. Sus parámetros principales son:
  - `p`: el vector de probabilidades teóricas.
  - `rescale.p`: igualado a `TRUE`, indica que los valores de `p` no son probabilidades, sino sólo proporcionales a las probabilidades.
  - `simulate.p.value`: igualado a `TRUE`, R calcula el p-valor mediante simulaciones.
  - `B`: en este último caso, permite especificar el número de simulaciones.

Guía rápida

- `ks.test` realiza el test de Kolmogorov-Smirnov. Tiene dos tipos de uso:
  - `ks.test(x,y)`: contrasta si los vectores `x` e `y` han sido generados por la misma distribución continua.
  - `ks.test(x, "distribución", parámetros)`: contrasta si el vector `x` ha sido generado por la distribución especificada, que se ha de indicar con el nombre de la función de distribución de R (la que empieza con p).

- `lillie.test` del paquete **nortest**, realiza el test de normalidad de Kolmogorov-Smirnov-Lilliefors.
- `ad.test` del paquete **nortest**, realiza el test de normalidad de Anderson-Darling.
- `shapiro.test`, realiza el test de normalidad de Shapiro-Wilk.
- `dagoTest` del paquete **fBasics**, realiza el test ómnibus de D'Agostino-Pearson.