Estimacion de minimos cuadrados

luis manuel

10/4/2021

En esta lección, si está utilizando RStudio, podrá jugar con parte del código que aparece en las diapositivas. Si no está usando RStudio, puede mirar el código pero no podrá experimentar con la función "manipular". Le proporcionamos el código para que pueda examinarlo sin tener que escribirlo todo. En RStudio, cuando la ventana de edición muestra el código, asegúrese de que el cursor parpadeante esté nuevamente en la ventana de la consola antes de presionar "Enter" o cualquier botón del teclado, de lo contrario, podría alterar accidentalmente el código. Si modifica el archivo, en RStudio, puede presionar Ctrl z en el editor hasta que desaparezcan todos los cambios no deseados. En otros editores, tendrá que usar cualquier combinación de teclas que realice "deshacer" para eliminar todos los cambios no deseados.

Aquí están los datos de Galton y la línea de regresión que se ven en la Introducción. La línea de regresión resume la relación entre la altura de los padres (los predictores) y la de sus hijos (los resultados).

```
library(UsingR)
data(galton)
```

```
library(manipulate)
myPlot <- function(beta){
  y <- galton$child - mean(galton$child)
  x <- galton$parent - mean(galton$parent)
  freqData <- as.data.frame(table(x, y))
  names(freqData) <- c("child", "parent", "freq")
  plot(
    as.numeric(as.vector(freqData$parent)),
    as.numeric(as.vector(freqData$child)),
    pch = 21, col = "black", bg = "lightblue",
    cex = .15 * freqData$freq,
    xlab = "parent",
    ylab = "child"
  )
  abline(0, beta, lwd = 3)
  points(0, 0, cex = 2, pch = 19)
  mse <- mean( (y - beta * x)^2 )
  title(paste("beta = ", beta, "mse = ", round(mse, 3)))
}
#manipulate(myPlot(beta), beta = manipulate::slider(0.4, .8, step = 0.02))
```

Aprendimos en la última lección que la línea de regresión es la línea a través de los datos que tiene el "error" mínimo (mínimo) cuadrado, la distancia vertical entre las 928 alturas reales de los niños y las alturas predichas por la línea. Al cuadrar las distancias, se asegura que los puntos de datos por encima y por debajo de la línea se traten de la misma manera. Este método para elegir la "mejor" línea de regresión (o "ajustar" una línea a los datos) se conoce como mínimos cuadrados ordinarios.

Como se muestra en las diapositivas, la línea de regresión contiene el punto que representa las medias de los dos conjuntos de alturas. Estos se muestran mediante las delgadas líneas horizontales y verticales. El punto de intersección se muestra mediante el triángulo en el gráfico. Su coordenada x es la media de las alturas de los padres y la coordenada y es la media de las alturas de los niños.

Como se muestra en las diapositivas, la pendiente de la línea de regresión es la correlación entre los dos conjuntos de alturas multiplicada por la relación de las desviaciones estándar (de los niños a los padres o de los resultados a los predictores).

Aquí mostramos un código que demuestra cómo el cambio de la pendiente de la línea de regresión afecta el error cuadrático medio entre los valores reales y predichos. Míralo para ver lo sencillo que es.

Recuerde que normaliza los datos restando su media y dividiendo por su desviación estándar. Hemos hecho esto para los datos de padres e hijos de Galton por usted. Hemos almacenado estos valores normalizados en dos vectores, gpa_nor y gch_nor, los datos padre e hijo normalizados de Galton.

```
gpa_nor<-scale(galton$parent)
gch_nor<-scale(galton$child)
cor(gpa_nor,gch_nor)
```

```
##           [,1]
## [1,] 0.4587624
```

¿Cómo se relaciona esta correlación con la correlación de los datos no normalizados?, es la misma

Utilice la función "lm" de R para generar la línea de regresión utilizando estos datos normalizados. Guárdelo en una variable llamada l_nor. Utilice la altura de los padres como predictores (variable independiente) y la de los niños como predicción (dependiente). Recuerde, 'lm' necesita una fórmula de la forma dependiente ~ independiente. Dado que hemos creado los vectores de datos para usted, no es necesario que proporcione un segundo argumento de "datos" como lo hizo anteriormente.

```
l_nor <- lm(gch_nor ~ gpa_nor)
l_nor$coefficients
```

```
##  (Intercept)      gpa_nor
## 2.982917e-15 4.587624e-01
```

Si intercambiaste el resultado (Y) y el predictor (X) de tus datos originales (no normalizados) (por ejemplo, usaste la altura de los niños para predecir a sus padres), ¿cuál sería la pendiente de la nueva línea de regresión? correlation(X,Y) * sd(X)/sd(Y) Cerraremos con una visualización final del código fuente de las diapositivas. Traza los datos de Galton con tres líneas de regresión, la original en rojo con los niños como resultado, una nueva línea azul con los padres como resultado y los niños como predictor, y una línea negra con la pendiente escalada para que sea igual a la razón. de las desviaciones estándar.

```
#plot the original Galton data points with larger dots for more freq pts
y <- galton$child
x <- galton$parent
freqData <- as.data.frame(table(galton$child, galton$parent))
names(freqData) <- c("child", "parent", "freq")
plot(as.numeric(as.vector(freqData$parent)),
     as.numeric(as.vector(freqData$child)),
     pch = 21, col = "black", bg = "lightblue",
     cex = .07 * freqData$freq, xlab = "parent", ylab = "child")
```

```r
#original regression line, children as outcome, parents as predictor
abline(mean(y) - mean(x) * cor(y, x) * sd(y) / sd(x), #intercept
       sd(y) / sd(x) * cor(y, x),  #slope
       lwd = 3, col = "red")

#new regression line, parents as outcome, children as predictor
abline(mean(y) - mean(x) * sd(y) / sd(x) / cor(y, x), #intercept
       sd(y) / cor(y, x) / sd(x), #slope
       lwd = 3, col = "blue")

#assume correlation is 1 so slope is ratio of std deviations
abline(mean(y) - mean(x) * sd(y) / sd(x), #intercept
       sd(y) / sd(x),  #slope
       lwd = 2)
points(mean(x), mean(y), cex = 2, pch = 19) #big point of intersection
```