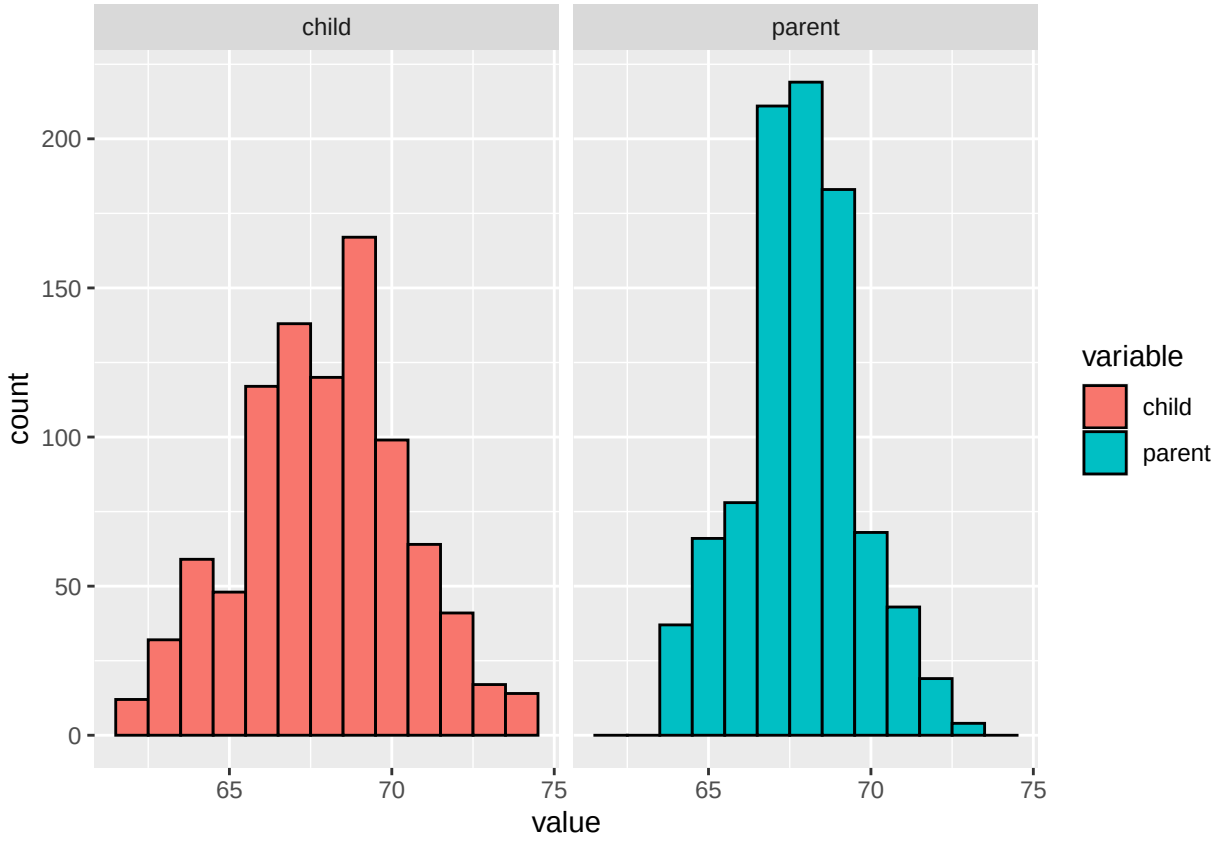
Introduction

videos

Distribuciones de padres e hijos

```
library(UsingR); data(galton); library(reshape); long <- melt(galton)
g <- ggplot(long, aes(x = value, fill = variable))
g <- g + geom_histogram(colour = "black", binwidth=1)
g <- g + facet_grid(. ~ variable)
g
```



Puede definirse la media μ como el valor que minimiza el error cuadratico, el cual se define que para cada y_i, $i = 1, 2...n$

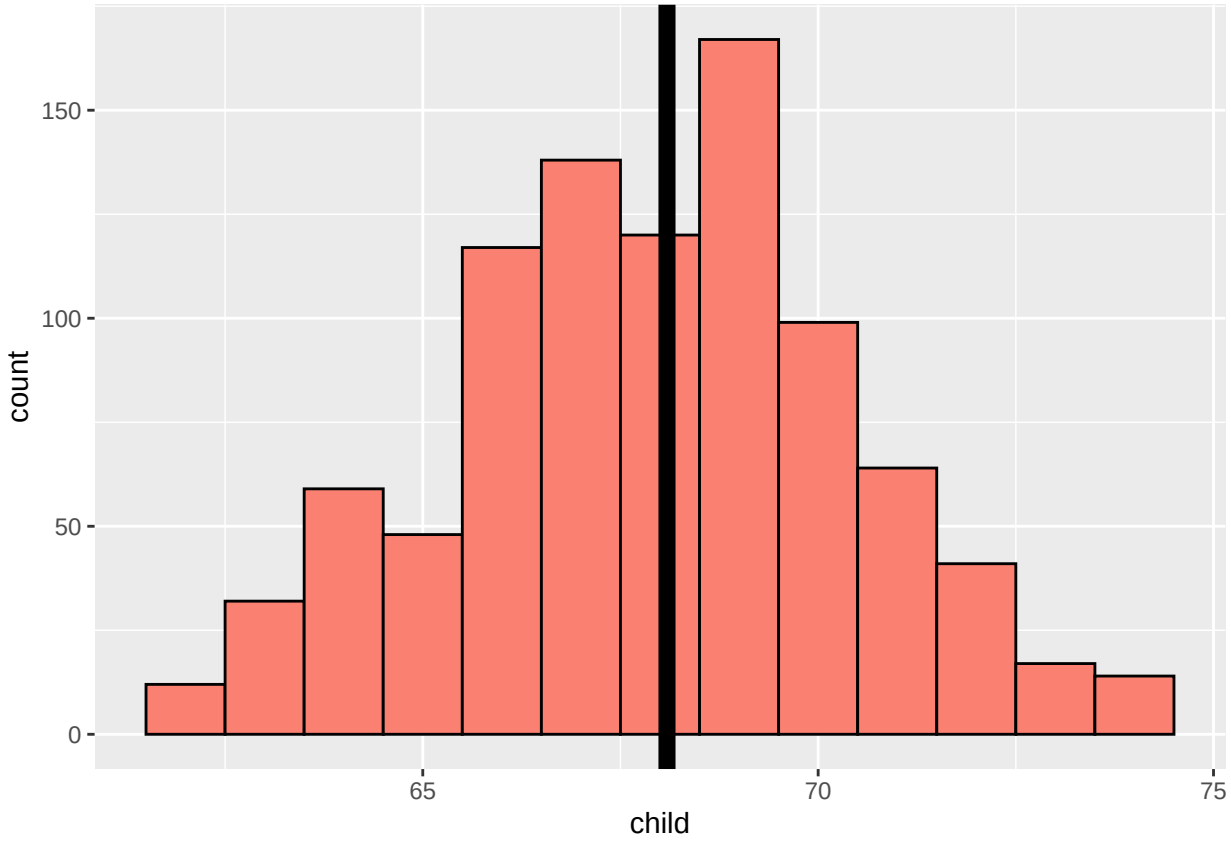
$$sum_{i=1}^{N}(y_i - \mu)^2$$

ya sabemos que un estimador para μ es $\bar{x}$

```
library(manipulate)
myHist <- function(mu){
    mse <- mean((galton$child - mu)^2)
    g <- ggplot(galton, aes(x = child)) + geom_histogram(fill = "salmon", colour = "black", binwidth=1)
    g <- g + geom_vline(xintercept = mu, size = 3)
    g <- g + ggtitle(paste("mu = ", mu, ", MSE = ", round(mse, 2), sep = ""))
    g
}
#manipulate(myHist(mu), mu = slider(62, 74, step = 0.5))
```

el grafico con el error cuadratico minimo es

```
g <- ggplot(galton, aes(x = child)) + geom_histogram(fill = "salmon", colour = "black", binwidth=1)
g <- g + geom_vline(xintercept = mean(galton$child), size = 3)
g
```

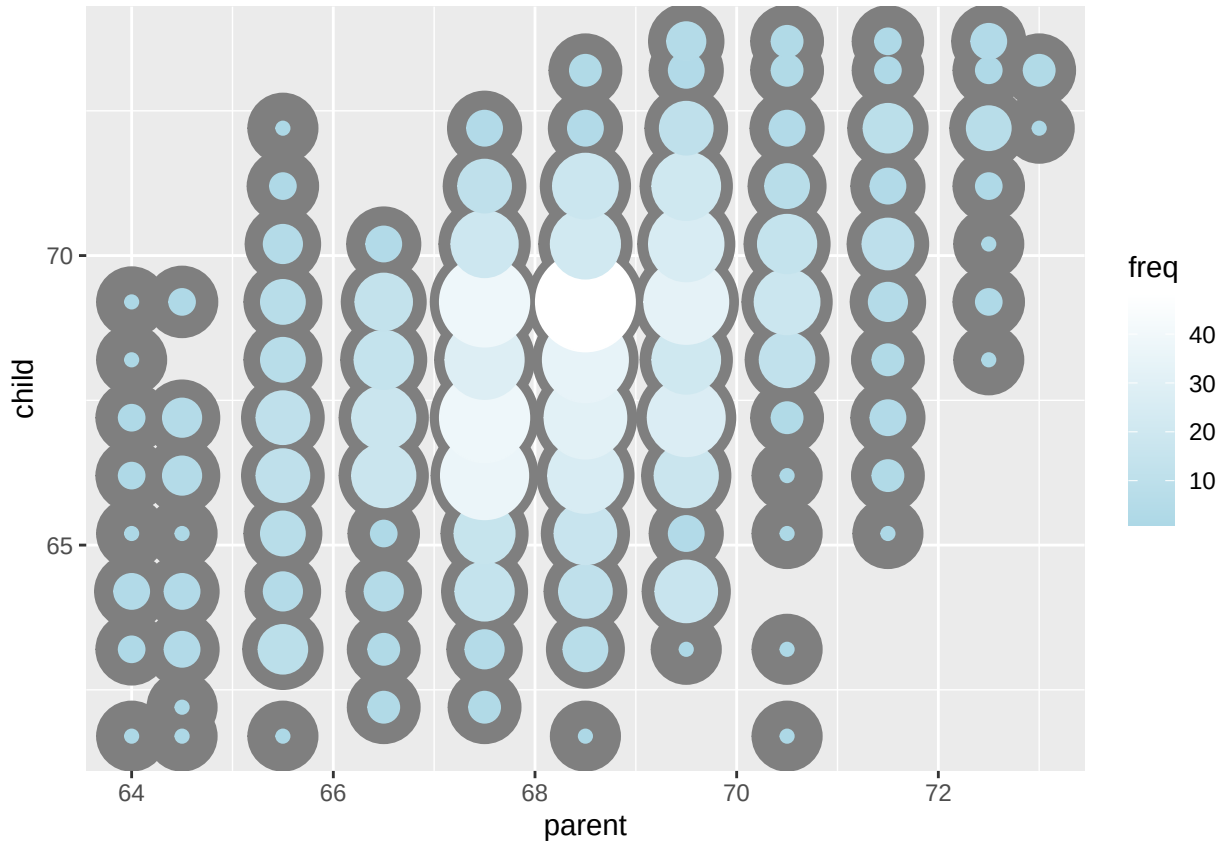


resulta que $\bar{x}$ es la estimacion minima para el error cuadratico

$$
\begin{aligned}
\sum_{i=1}^{n} (Y_i - \mu)^2 &= \sum_{i=1}^{n} \left(Y_i - \bar{Y} + \bar{Y} - \mu\right)^2 \\
&= \sum_{i=1}^{n} \left(Y_i - \bar{Y}\right)^2 + 2\sum_{i=1}^{n} \left(Y_i - \bar{Y}\right)\left(\bar{Y} - \mu\right) + \sum_{i=1}^{n} \left(\bar{Y} - \mu\right)^2 \\
&= \sum_{i=1}^{n} \left(Y_i - \bar{Y}\right)^2 + 2\left(\bar{Y} - \mu\right)\sum_{i=1}^{n} \left(Y_i - \bar{Y}\right) + \sum_{i=1}^{n} \left(\bar{Y} - \mu\right)^2 \\
&= \sum_{i=1}^{n} \left(Y_i - \bar{Y}\right)^2 + 2\left(\bar{Y} - \mu\right)\left(\left(\sum_{i=1}^{n} Y_i\right) - n\bar{Y}\right) + \sum_{i=1}^{n} \left(\bar{Y} - \mu\right)^2 \\
&= \sum_{i=1}^{n} \left(Y_i - \bar{Y}\right)^2 + \sum_{i=1}^{n} \left(\bar{Y} - \mu\right)^2 \\
&\geq \sum_{i=1}^{n} \left(Y_i - \bar{Y}\right)^2
\end{aligned}
$$

grafica que muestra las frecuencias correctas:

```
library(dplyr)
freqData <- as.data.frame(table(galton$child, galton$parent))
names(freqData) <- c("child", "parent", "freq")
freqData$child <- as.numeric(as.character(freqData$child))
freqData$parent <- as.numeric(as.character(freqData$parent))
g <- ggplot(filter(freqData, freq > 0), aes(x = parent, y = child))
g <- g  + scale_size(range = c(2, 20), guide = "none" )
g <- g + geom_point(colour="grey50", aes(size = freq+20, show_guide = FALSE))
g <- g + geom_point(aes(colour=freq, size = freq))
g <- g + scale_colour_gradient(low = "lightblue", high="white")
g
```



ahora podemos forzar la interseccion de la recta restando la media a cada subconjunto de datos, de esta forma solo tenemos que encontrar el valor optimo de β tal que

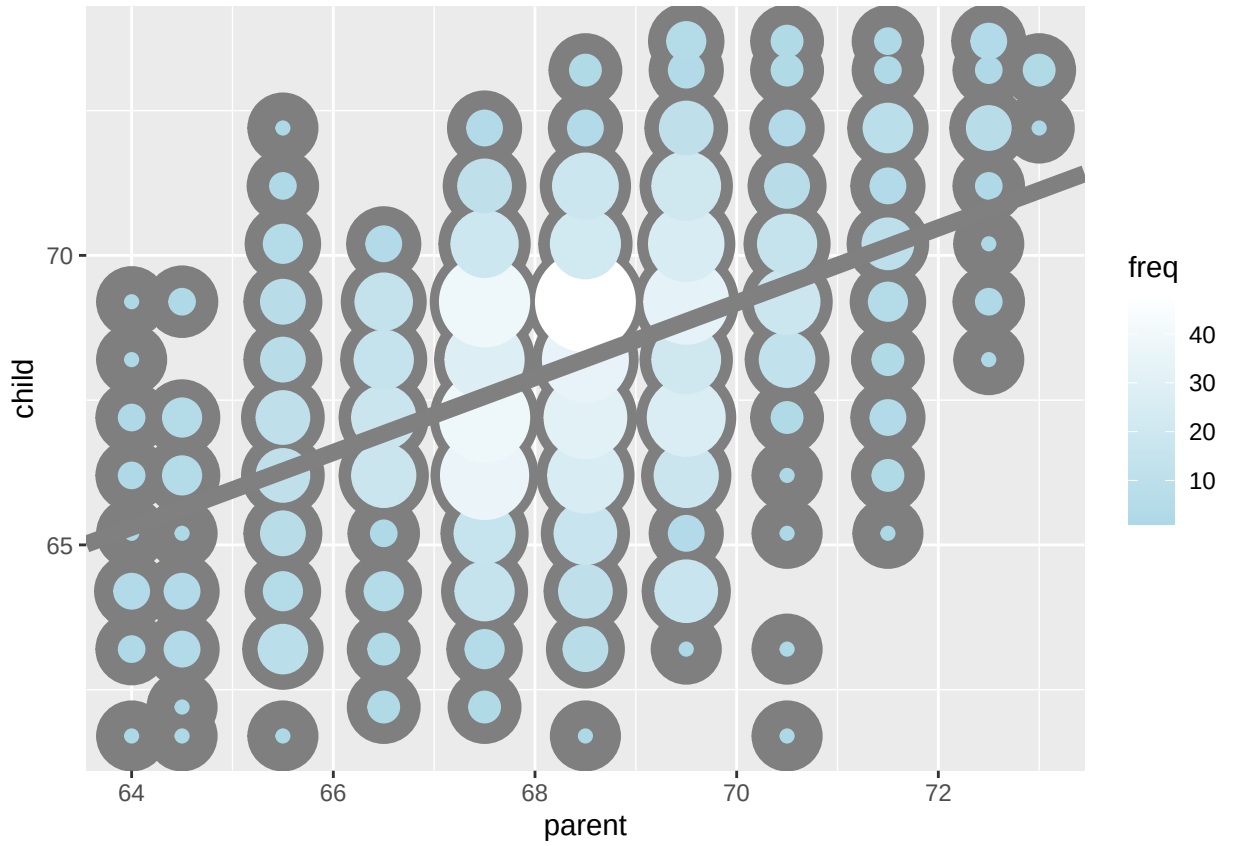
$$\sum_{i=1}^{n}(Y_i - \beta x_i)^2$$

este es el codigo de simulacion de distintos valores de β

```
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")
freqData$child <- as.numeric(as.character(freqData$child))
freqData$parent <- as.numeric(as.character(freqData$parent))
myPlot <- function(beta){
    g <- ggplot(filter(freqData, freq > 0), aes(x = parent, y = child))
    g <- g  + scale_size(range = c(2, 20), guide = "none" )
    g <- g + geom_point(colour="grey50", aes(size = freq+20, show_guide = FALSE))
    g <- g + geom_point(aes(colour=freq, size = freq))
    g <- g + scale_colour_gradient(low = "lightblue", high="white")
    g <- g + geom_abline(intercept = 0, slope = beta, size = 3)
    mse <- mean( (y - beta * x) ^2 )
    g <- g + ggtitle(paste("beta = ", beta, "mse = ", round(mse, 3)))
    g
}
#manipulate(myPlot(beta), beta = slider(0.6, 1.2, step = 0.02))
```

este es grafico con la recta correcta:

```
freqData <- as.data.frame(table(galton$child, galton$parent))
names(freqData) <- c("child", "parent", "freq")
freqData$child <- as.numeric(as.character(freqData$child))
freqData$parent <- as.numeric(as.character(freqData$parent))
g <- ggplot(filter(freqData, freq > 0), aes(x = parent, y = child))
g <- g  + scale_size(range = c(2, 20), guide = "none" )
g <- g + geom_point(colour="grey50", aes(size = freq+20, show_guide = FALSE))
g <- g + geom_point(aes(colour=freq, size = freq))
g <- g + scale_colour_gradient(low = "lightblue", high="white")
lm1 <- lm(galton$child ~ galton$parent)
g <- g + geom_abline(intercept = coef(lm1)[1], slope = coef(lm1)[2], size = 3, colour = grey(.5))
g
```



consideremos el grafico de dispercion, la mejor "linea" que se ajusta es tal que minimiza la suma:

$$\sum_{i=1}^{n}\{Y_i - (\beta_0 + \beta_1 X_i)\}^2$$

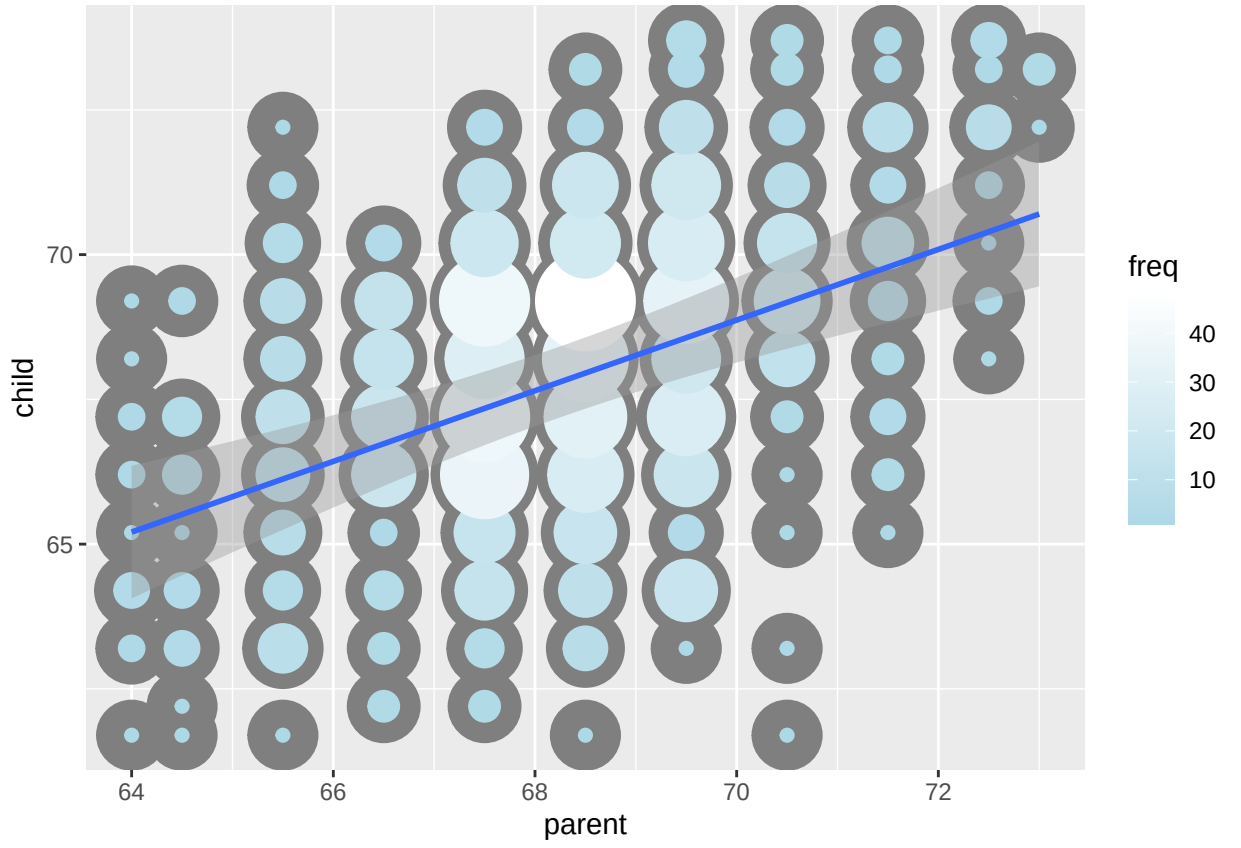
los estimadores para encontrar β_0 y β_1 son

$$\hat{\beta}_1 = Cor(Y,X)\frac{Sd(Y)}{Sd(X)} \text{ y } \hat{\beta}_0 = \bar{Y} - \hat{\beta}_1\bar{X}$$

si se centran los datos entonces $\hat{\beta}_1 = Cor(Y,X)$ y $\hat{\beta}_0 = 0$

grafica con intervalos de confianza

```
g <- ggplot(filter(freqData, freq > 0), aes(x = parent, y = child))
g <- g  + scale_size(range = c(2, 20), guide = "none" )
g <- g + geom_point(colour="grey50", aes(size = freq+20, show_guide = FALSE))
g <- g + geom_point(aes(colour=freq, size = freq))
g <- g + scale_colour_gradient(low = "lightblue", high="white")
g <- g + geom_smooth(method="lm", formula=y~x)
g
```



swir

Esta es la primera lección sobre modelos de regresión. Comenzaremos con el concepto de "regresión hacia la media" y lo ilustraremos con un trabajo pionero del padre de la ciencia forense, Sir Francis Galton.

Sir Francis estudió la relación entre la altura de los padres y sus hijos. Su trabajo mostró que los padres que eran más altos que el promedio tenían hijos que también eran altos pero más cercanos a la estatura promedio. De manera similar, los padres que eran más bajos que el promedio tenían hijos que también eran más bajos que el promedio, pero menos que mamá y papá. Es decir, estaban más cerca de la altura media. De una generación a la siguiente, las alturas se acercaron a la media o retrocedieron hacia la media.

Para esta lección usaremos los datos de altura de padre / hijo de Sir Francis que nos hemos tomado la libertad de cargar para usted como la variable, galton. (Los datos son del sitio web de John Verzani, http://wiener.math.csi.cuny.edu/UsingR/.) ¡Así que comencemos!

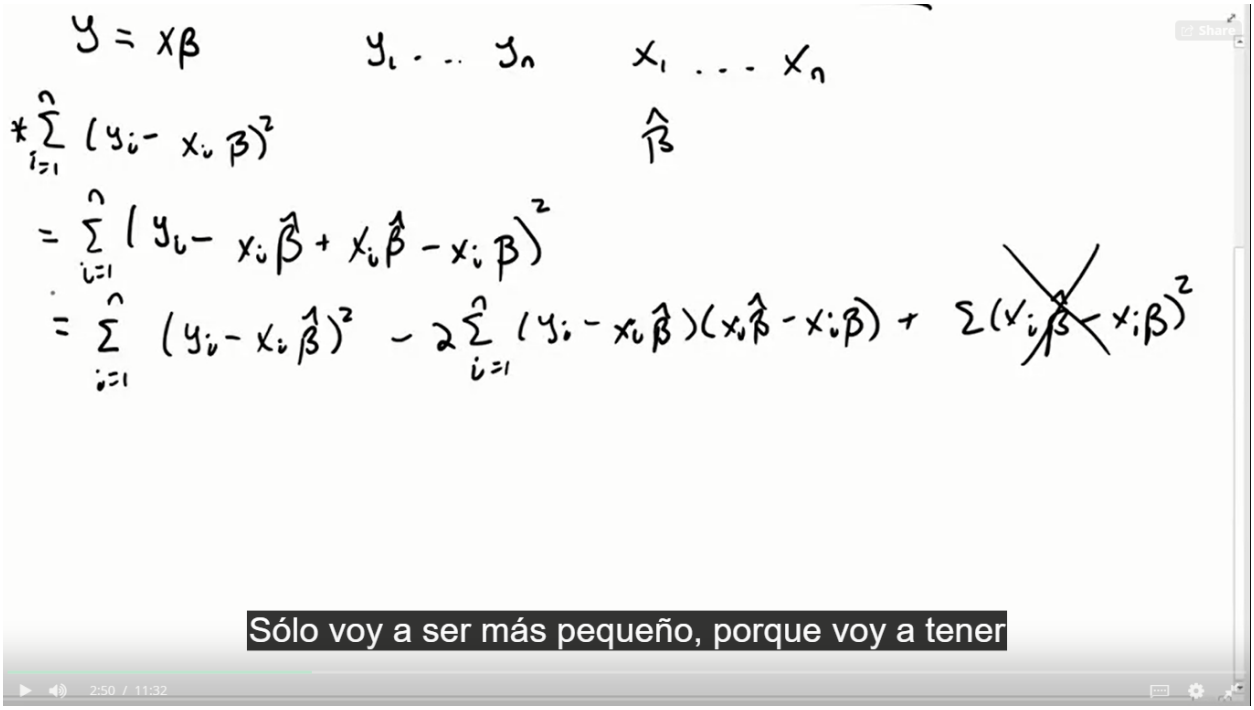


Figure 1: A caption

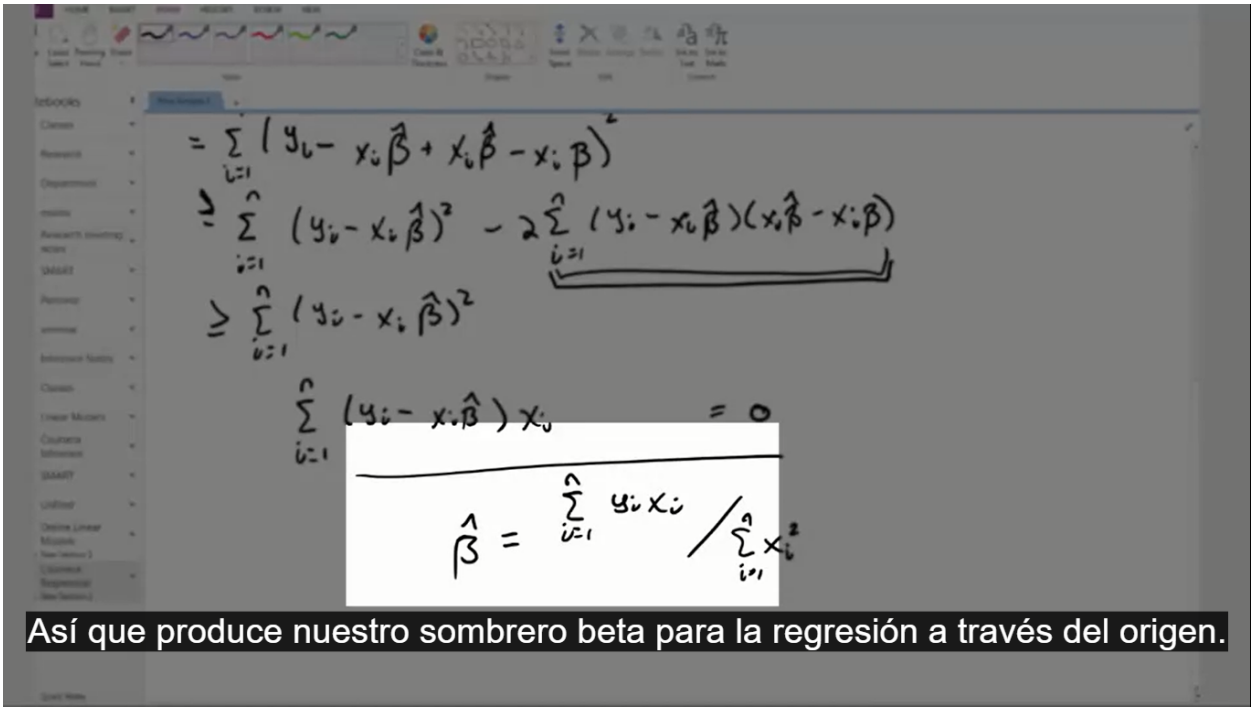


Figure 2: A caption

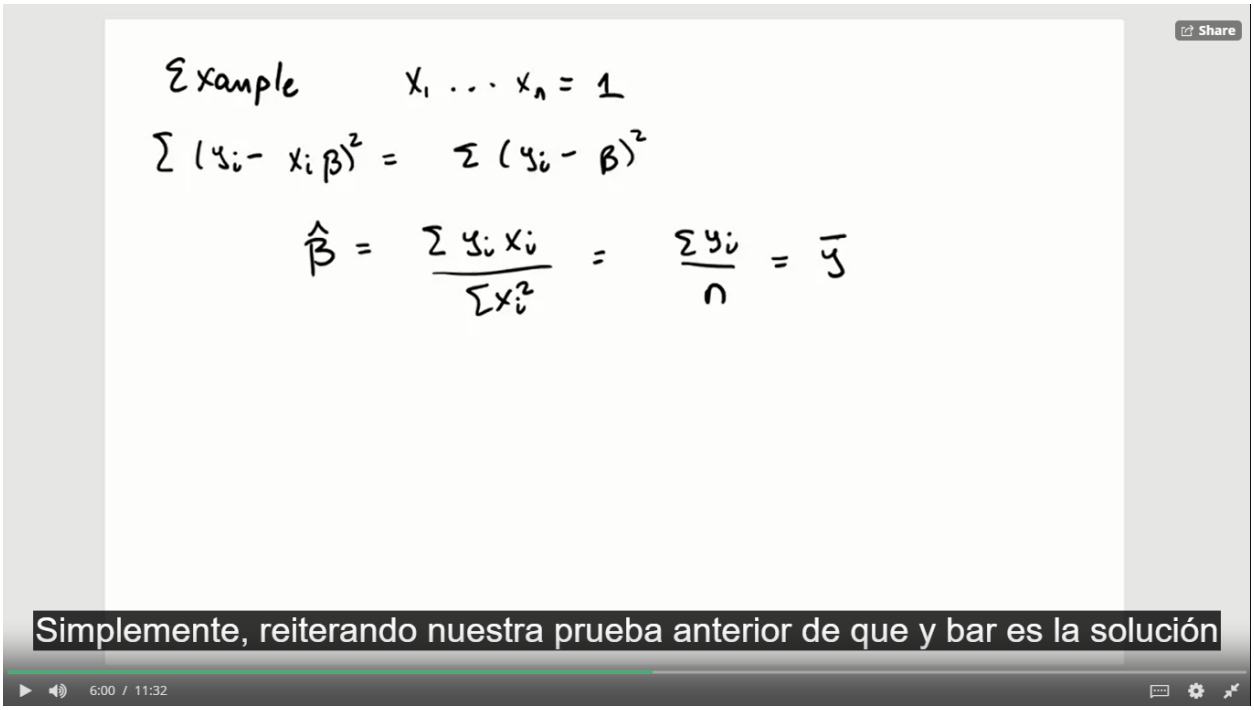


Figure 3: A caption

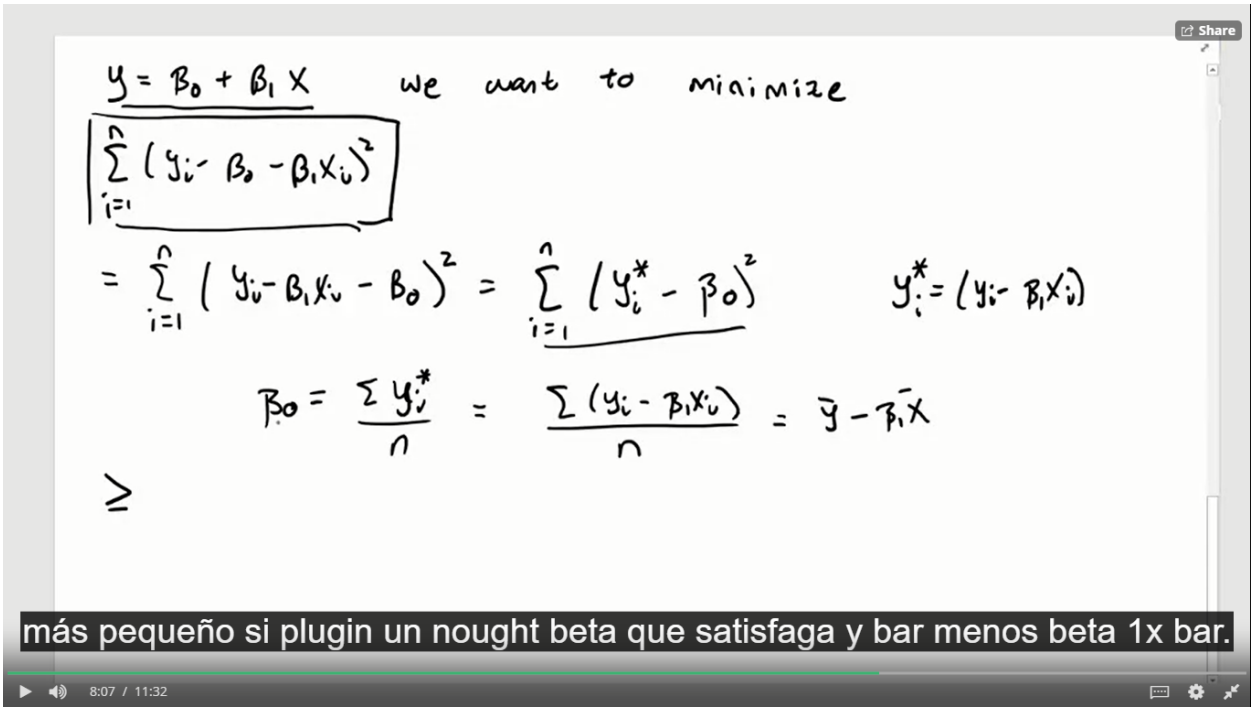


Figure 4: A caption

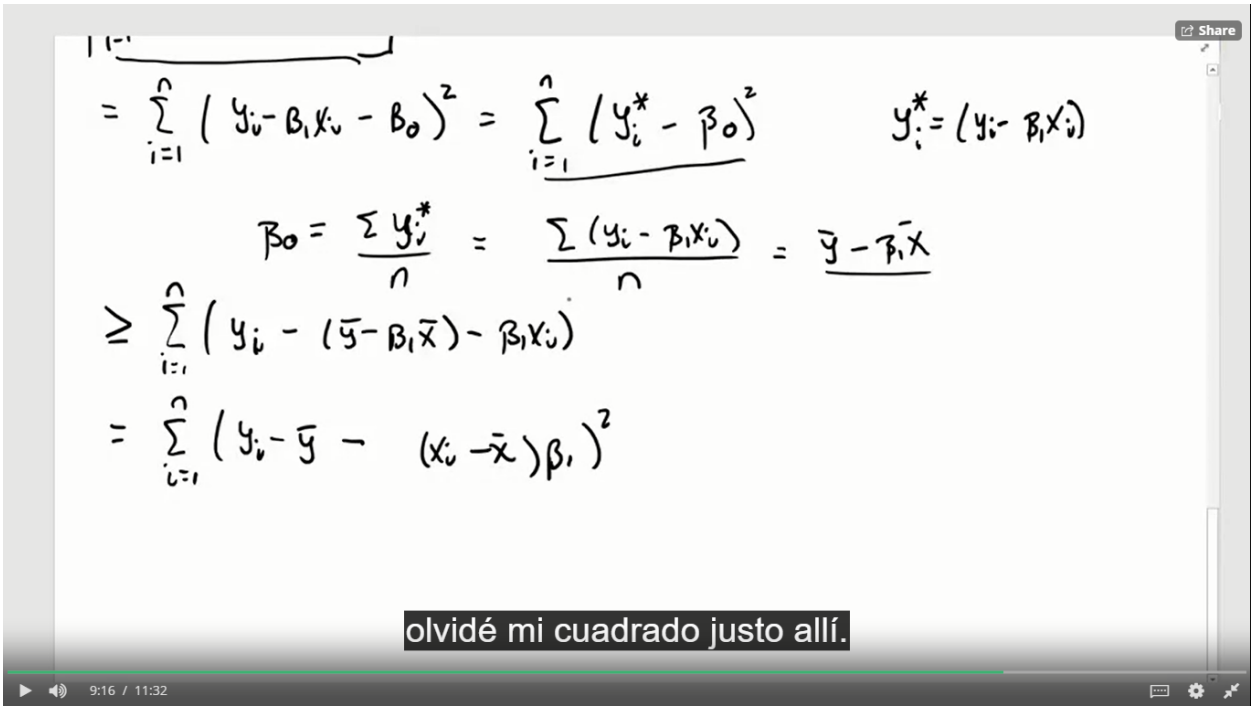


Figure 5: A caption

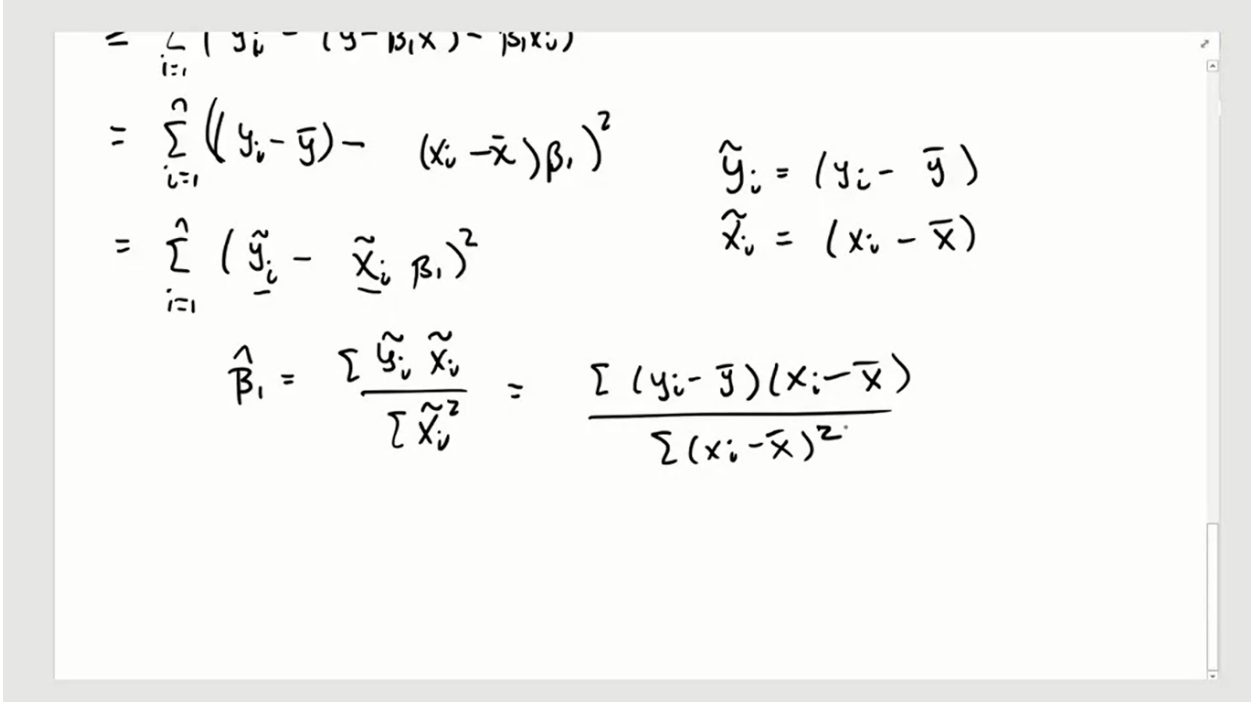


Figure 6: A caption

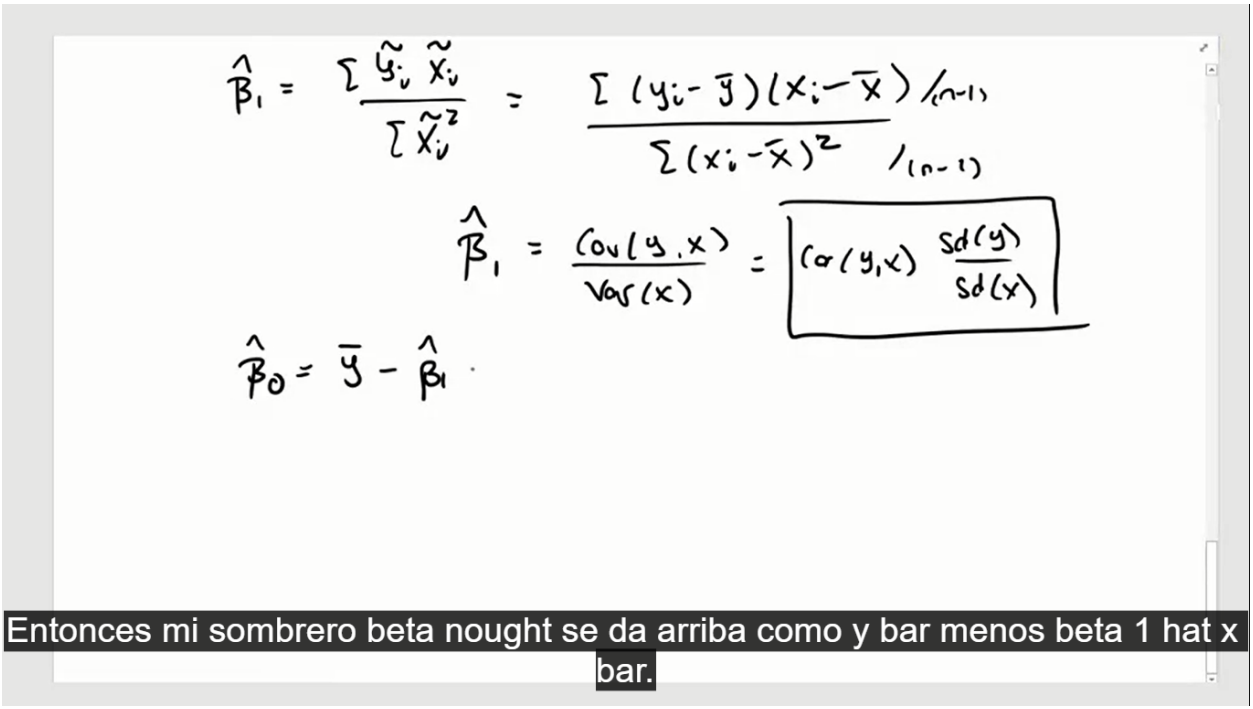


Figure 7: A caption

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

Aquí hay una gráfica de los datos de Galton, un conjunto de 928 pares de estatura de padres e hijos. Las alturas de las mamás y los papás se promediaron juntas (después de ajustar las alturas de las mamás en un factor de 1.08). En nuestro gráfico usamos la función R "jitter" en las alturas de los niños para resaltar las alturas que ocurrieron con mayor frecuencia. Las manchas oscuras en cada columna se elevan de izquierda a derecha, lo que sugiere que la altura de los niños depende de la de sus padres. Los padres altos tienen hijos altos y los padres bajos tienen hijos bajos.

Aquí agregamos una línea roja (45 grados) de pendiente 1 e interceptamos 0 a la gráfica. Si los niños tendieran a tener la misma altura que sus padres, esperaríamos que los datos varíen uniformemente sobre esta línea. Vemos que este no es el caso. En la mitad izquierda del gráfico vemos una concentración de alturas por encima de la línea, y en la mitad derecha vemos la concentración por debajo de la línea.

Ahora hemos agregado una línea de regresión azul al gráfico. Esta es la línea que tiene la mínima variación de los datos a su alrededor. (Para ver la teoría, vea las diapositivas). Su pendiente es mayor que cero, lo que indica que la altura de los padres afecta a sus hijos. La pendiente también es menor que 1, como habría sido el caso si los niños tendieran a tener la misma altura que sus padres. vamos a crea la grafica

```
plot(child ~ parent, galton)
```

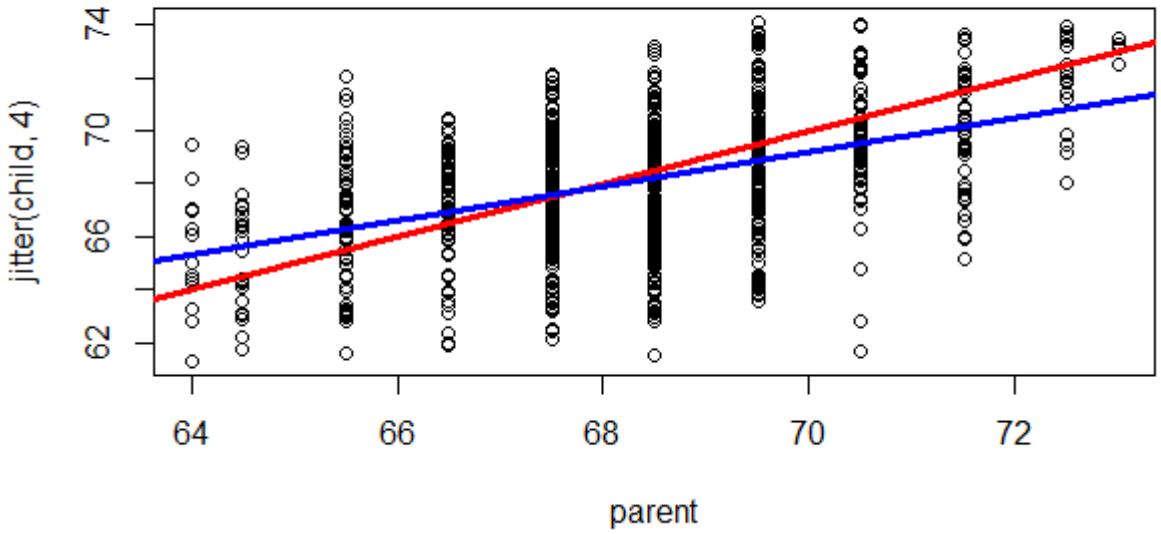
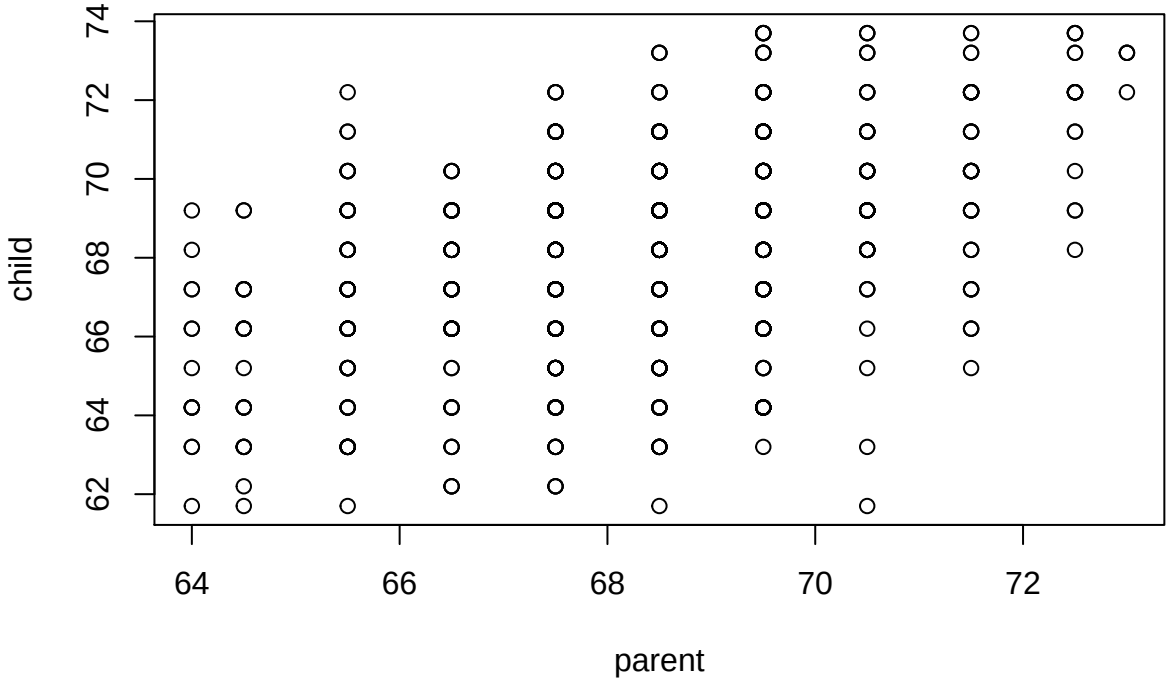
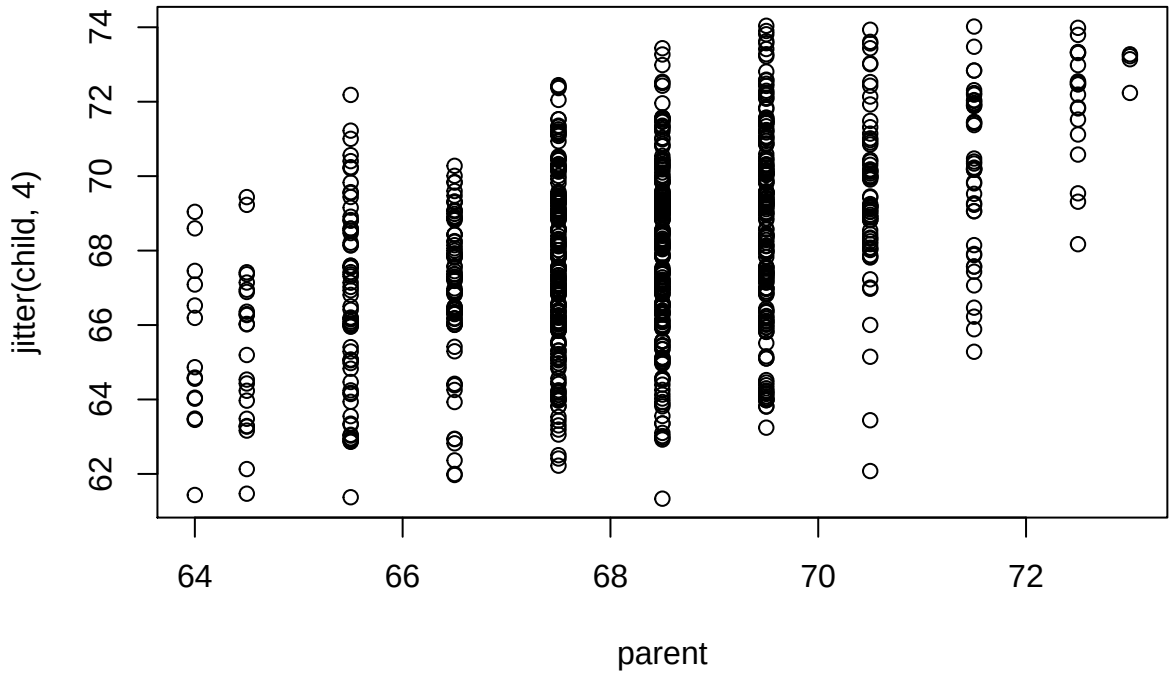


Figure 8: A caption



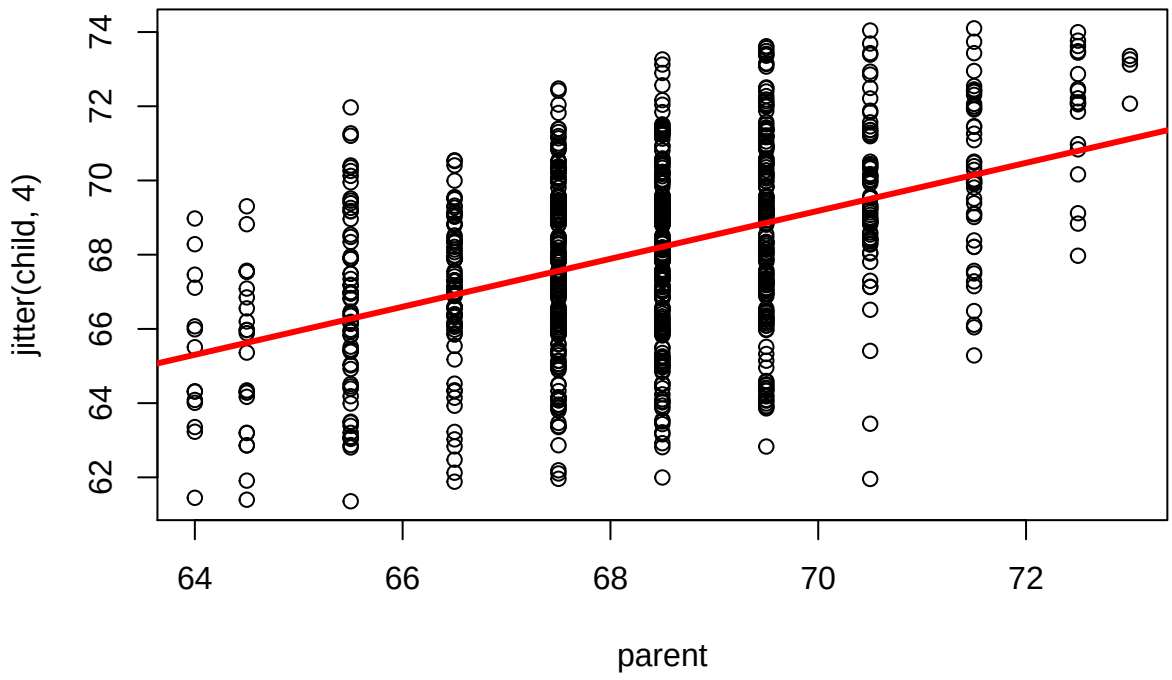
Notarás que esta trama se ve muy diferente a la original que mostramos. ¿Por qué? Muchas personas tienen la misma altura dentro del error de medición, por lo que los puntos caen uno encima del otro. Puede ver que algunos círculos aparecen más oscuros que otros. Sin embargo, al usar la función "jitter" de R en las alturas de los niños, podemos distribuir los datos para simular los errores de medición y hacer que las alturas de alta frecuencia sean más visibles.

```
plot(jitter(child,4) ~ parent,galton)
```



Ahora para la línea de regresión. Esto es bastante fácil en R. La función lm (modelo lineal) necesita una "fórmula" y un conjunto de datos. Puede escribir "? Fórmula" para obtener más información, pero, en términos simples, solo necesitamos especificar la variable dependiente (altura de los niños) ~ la variable independiente (altura de los padres)

```
plot(jitter(child,4) ~ parent,galton)
regrline <- lm(child ~ parent, galton)
abline(regrline, lwd=3, col='red')
```



La línea de regresión tendrá una pendiente y una intersección que se estiman a partir de los datos. Las estimaciones no son exactas. Su precisión se mide mediante técnicas teóricas y se expresa en términos de "error estándar". Puede utilizar "resumen (regrline)" para examinar la línea de regresión de Galton. Hacerlo ahora.

```
summary(regrline)
```

```
##
## Call:
## lm(formula = child ~ parent, data = galton)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.8050 -1.3661  0.0487  1.6339  5.9264
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 23.94153    2.81088   8.517   <2e-16 ***
## parent       0.64629    0.04114  15.711   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.239 on 926 degrees of freedom
## Multiple R-squared:  0.2105, Adjusted R-squared:  0.2096
## F-statistic: 246.8 on 1 and 926 DF,  p-value: < 2.2e-16
```

La pendiente de la línea es la estimación del coeficiente, o multiplicador, de “padre”, la variable independiente de nuestros datos (en este caso, las alturas de los padres). y es .64629,el error estándar de la pendiente es .04114 Un coeficiente estará dentro de 2 errores estándar de su estimación aproximadamente el 95% del tiempo. Esto significa que la pendiente de nuestra regresión es significativamente diferente de 0 o 1, ya que (.64629) +/- (2 * .04114) no está cerca de 0 ni de 1.

Ahora estamos agregando dos líneas azules para indicar las medias de las alturas de los niños (horizontal) y las de los padres (vertical). Tenga en cuenta que estas líneas y la línea de regresión se cruzan en un punto. Bastante bien, ¿eh? Hablaremos más sobre esto en una lección posterior. (Algo que puede esperar).

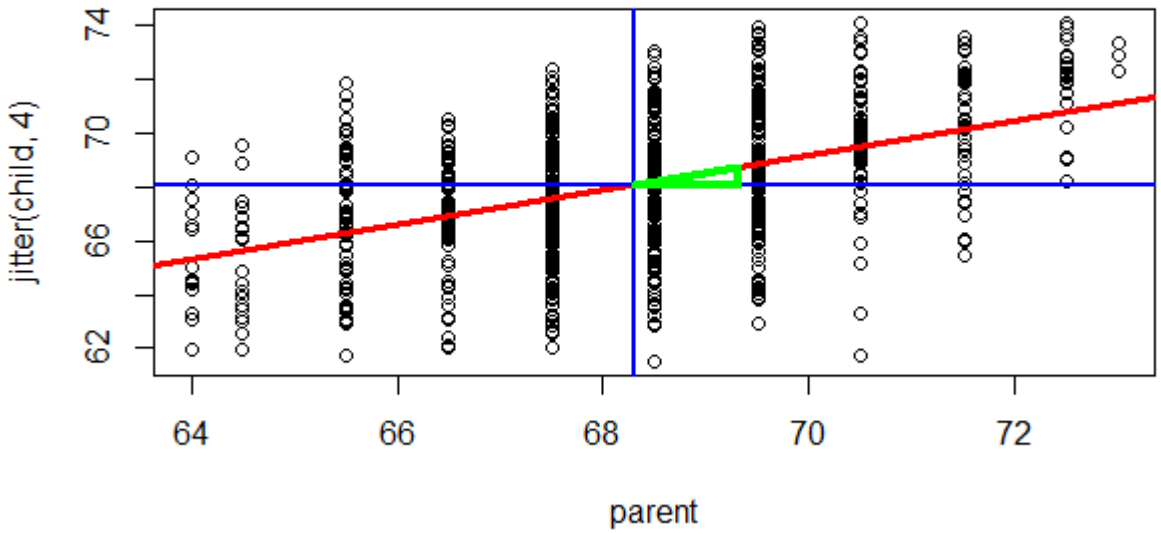


Figure 9: A caption

La pendiente de una línea muestra cuánto cambio en la dirección vertical se produce por un cambio en la dirección horizontal. Por lo tanto, los padres “1 pulgada” por encima de la altura media tienden a tener hijos que están sólo .65 pulgadas por encima de la media. El triángulo verde ilustra este punto. A partir de la media, moverse una “distancia de 1 pulgada” horizontalmente hacia la derecha (aumentando la altura de los padres) produce un aumento de “0,65 pulgadas” en la dirección vertical (altura de los niños).

De manera similar, los padres que tienen una altura de 1 pulgada por debajo del promedio tienen hijos que están solo .65 pulgadas por debajo de la altura promedio. El triángulo púrpura ilustra esto. A partir de la media, moverse una “distancia de 1 pulgada” horizontalmente hacia la izquierda (disminuyendo la altura de los padres) produce una disminución de “.65 pulgadas” en la dirección vertical (altura de los niños).

residuos

Esta lección se enfocará en los residuos, las distancias entre las alturas reales de los niños y las estimaciones dadas por la línea de regresión. Dado que todas las líneas se caracterizan por dos parámetros, una pendiente y una intersección, usaremos el criterio de mínimos cuadrados para proporcionar dos ecuaciones en dos incógnitas para poder resolver estos parámetros, la pendiente y la intersección.

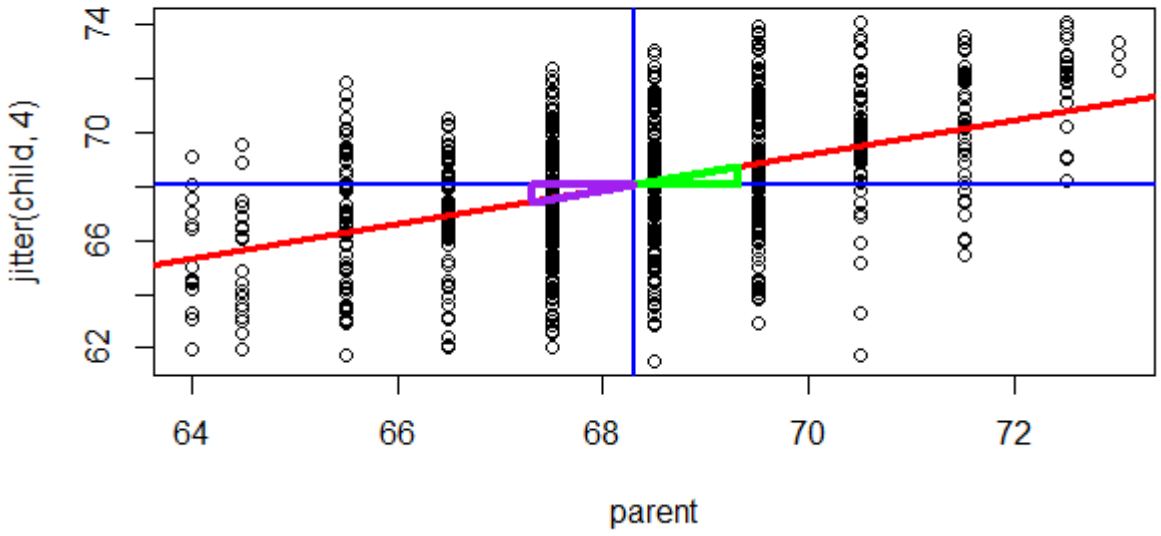


Figure 10: A caption

La primera ecuación dice que los "errores" en nuestras estimaciones, los residuos, tienen una media de cero. En otras palabras, los residuos están "equilibrados" entre los puntos de datos; es tan probable que sean positivos como negativos. La segunda ecuación dice que nuestros residuos deben no estar correlacionados con nuestros predictores, la altura de los padres. Esto tiene sentido: si los residuos y los predictores estuvieran correlacionados, podría realizar una mejor predicción y reducir las distancias (residuos) entre los resultados reales y las predicciones.

Demostraremos estos conceptos ahora. Primero regenere la línea de regresión y llámela ajuste. Utilice la función R lm. Recuerde que, por defecto, su primer argumento es una fórmula como "hijo ~ padre" y su segundo es el conjunto de datos, en este caso galton.

```
library(UsingR)
data(galton)
fit <- lm(child ~ parent, galton)
```

Ahora examinaremos el ajuste para ver su pendiente e intersección. Los residuos que nos interesan se almacenan en el vector de 928 longitudes fit$residuals. Si escribe fit$residuals, verá muchos números desplazarse, lo que no es muy útil; sin embargo, si escribe "summary(fit))", verá una visualización más concisa de los datos de regresión. Hacerlo ahora.

```
summary(fit)
```

```
##
## Call:
## lm(formula = child ~ parent, data = galton)
##
## Residuals:
```

```
##     Min      1Q  Median      3Q     Max
## -7.8050 -1.3661  0.0487  1.6339  5.9264
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 23.94153    2.81088   8.517   <2e-16 ***
## parent       0.64629    0.04114  15.711   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.239 on 926 degrees of freedom
## Multiple R-squared:  0.2105, Adjusted R-squared:  0.2096
## F-statistic: 246.8 on 1 and 926 DF,  p-value: < 2.2e-16
```

Primero verifique la media de fit$residuales para ver si está cerca de 0.

```
mean(fit$residuals)
```

```
## [1] -2.359884e-15
```

Ahora verifique la correlación entre los residuos y los predictores.

```
cov(fit$residuals, galton$parent)
```

```
## [1] -1.790153e-13
```

Como se muestra algebraicamente en las diapositivas, las ecuaciones para la intersección y la pendiente se encuentran suponiendo que se realiza un cambio en la intersección y la pendiente. Al cuadrar las expresiones resultantes se obtienen tres sumas. La primera suma es el término original al cuadrado, antes de que se cambiaran la pendiente y la intersección. La tercera suma suma los cambios al cuadrado ellos mismos. Por ejemplo, si hubiéramos cambiado la intersección del ajuste agregando 2, la tercera suma sería el total de 928 4. Se garantiza que la suma media sea cero precisamente cuando se satisfacen las dos ecuaciones (las condiciones de los residuos).

Verificaremos estas afirmaciones ahora. Hemos definido para usted dos funciones R, est y sqe. Ambos toman dos entradas, una pendiente y una intersección. La función est calcula la altura de un niño (coordenada y) utilizando la línea definida por los dos parámetros (pendiente e intersección) y las alturas de los padres en los datos de Galton como coordenadas x.

Sea "mch" la media de las alturas de los niños galton y "mph" la media de las alturas de los padres galton. Deje que "ic" y "slope" representen la intersección y la pendiente de la línea de regresión, respectivamente. Como se muestra en las diapositivas y lecciones anteriores, el punto (mph, mch) se encuentra en la línea de regresión. Esto significa que mch = ic + slope*mph

La función sqe calcula la suma de los residuos al cuadrado, las diferencias entre las alturas reales de los niños y las alturas estimadas especificadas por la línea definida por los parámetros dados (pendiente e intersección). R proporciona la desviación de la función para hacer exactamente esto usando un modelo ajustado (por ejemplo, ajuste) como argumento. Sin embargo, proporcionamos sqe porque lo usaremos para probar líneas de regresión diferentes de fit.

Veremos que cuando variamos o modificamos los valores de pendiente e intersección de la línea de regresión que se almacenan en $ fit $ coef, los residuos cuadrados resultantes son aproximadamente iguales a la suma de dos sumas de cuadrados, la de los residuos de regresión originales. y el de los propios retoques. Más precisamente, hasta el error numérico,

sqe(ols.slope+sl,ols.intercept+ic) == deviance(fit) + sum(est(sl,ic)^2)

Equivalentemente, sqe(ols.slope+sl,ols.intercept+ic) == sqe(ols.slope, ols.intercept) + sum(est(sl,ic)^2)

El lado izquierdo de la ecuación representa los residuos al cuadrado de una nueva línea, la línea de regresión "modificada". Los términos "sl" e "ic" representan las variaciones en la pendiente y la intersección, respectivamente. El lado derecho tiene dos términos. El primero representa los residuos al cuadrado de la línea de regresión original y el segundo es la suma de los cuadrados de las variaciones mismas.

Lo demostraremos ahora.

```
ols.ic <- fit$coef[1]
ols.slope <- fit$coef[2]
```

Ahora le mostraremos un código R que genera los lados izquierdo y derecho de esta ecuación. Tómate un momento para revisarlo. Hemos formado dos vectores de variaciones de 6 largos, uno para la pendiente y otro para la intersección. Luego tenemos dos bucles "for" para generar los dos lados de la ecuación.

```
est<-function(slope, intercept)intercept + slope*galton$parent
sqe<-function(slope, intercept)sum( (est(slope, intercept)-galton$child)^2)
```

```
#Here are the vectors of variations or tweaks
sltweak <- c(.01, .02, .03, -.01, -.02, -.03) #one for the slope
ictweak <- c(.1, .2, .3, -.1, -.2, -.3)  #one for the intercept
lhs <- numeric()
rhs <- numeric()
#left side of eqn is the sum of squares of residuals of the tweaked regression line
for (n in 1:6) lhs[n] <- sqe(ols.slope+sltweak[n],ols.ic+ictweak[n])
#right side of eqn is the sum of squares of original residuals + sum of squares of two tweaks
for (n in 1:6) rhs[n] <- sqe(ols.slope,ols.ic) + sum(est(sltweak[n],ictweak[n])^2)
```

Reste el lado derecho, el vector rhs, del lado izquierdo, el vector lhs, para ver la relación entre ellos. Debería obtener un vector de números muy pequeños, casi 0. También puede usar la función R all.equal con lhs y rhs como argumentos para probar la igualdad.

```
lhs-rhs
```

```
## [1]  1.264198e-09  2.527486e-09  3.801688e-09 -1.261469e-09 -2.522938e-09
## [6] -3.767127e-09
```

```
all.equal(lhs,rhs)
```

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

Ahora mostraremos que la varianza en las alturas de los niños es la suma de la varianza en las estimaciones de OLS y la varianza en los residuos de OLS. Primero use la función R var para calcular la varianza en las alturas de los niños y almacénela en la variable varChild.

```
varChild <- var(galton$child)
```

Utilice la función R var para calcular la varianza en estos residuos ahora y almacénela en la variable varRes.

```
varRes <- var(fit$residuals)
```

Recuerde que la función “est” calcula las estimaciones (coordenadas y) de los valores a lo largo de la línea de regresión definida por las variables “ols.slope” y “ols.ic”. Calcule la varianza en las estimaciones y almacénela en la variable varEst.

```
varEst <- var(est(ols.slope, ols.ic))
```

ahora vamos a verificar

```
all.equal(varChild,varEst+varRes)
```

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

Dado que las varianzas son sumas de cuadrados (y por lo tanto siempre positivas), esta ecuación que acabamos de demostrar, var (datos) = var (estimación) + var (residuos), muestra que la varianza de la estimación es SIEMPRE menor que la varianza de los datos.

```
data(attenu)
```

Las dos propiedades de los residuos que hemos enfatizado aquí se pueden aplicar a conjuntos de datos que tienen múltiples predictores. En esta lección, hemos cargado el conjunto de datos atenu que proporciona datos para 23 terremotos en California. Las aceleraciones se estiman en base a dos predictores, distancia y magnitud.

```
efit <- lm(accel ~ mag+dist, attenu)
```

verifiqumos la media de los residuos y las covarianzas entre residuos

```
mean(efit$residuals)
```

```
## [1] -1.785061e-18
```

```
cov(efit$residuals, attenu$mag)
```

```
## [1] 5.338694e-17
```

```
cov(efit$residuals, attenu$dist)
```

```
## [1] 5.253433e-16
```

estimacion de minimos cuadrados

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.

```r
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.

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

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

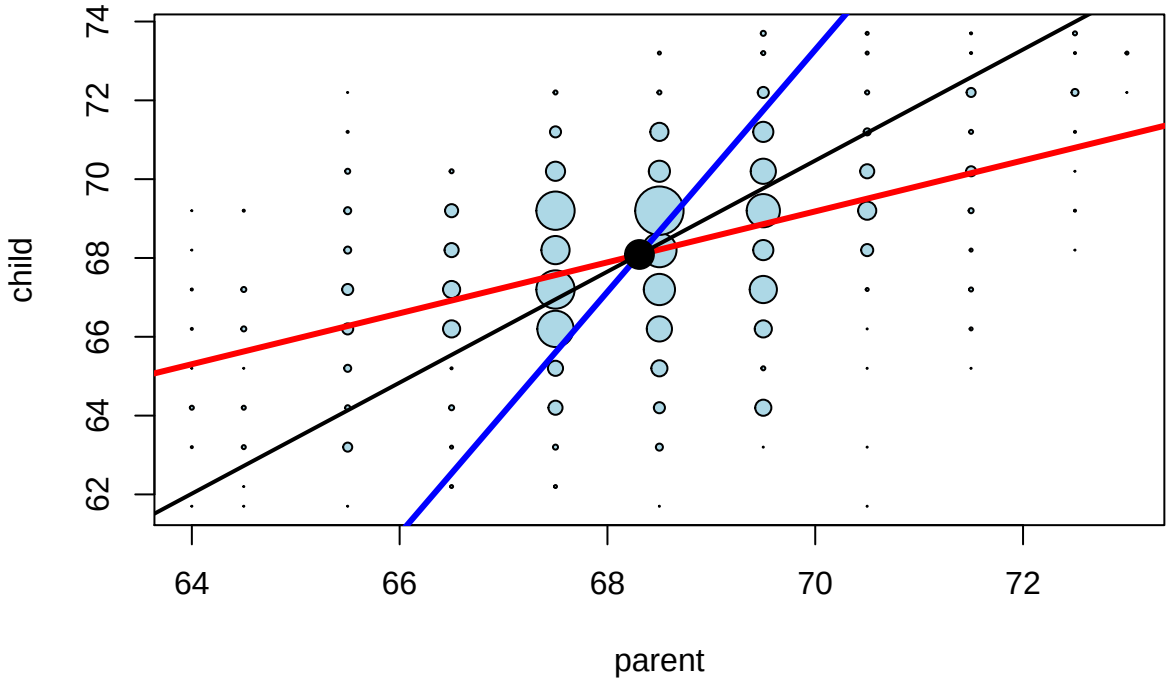
Si intercambiaste el resultado (Y) y el predictor (X) de tus datos originales (no normalizados) (por ejemplo, usaste la altura de los niños para predecir a sus padres), ¿cuál sería la pendiente de la nueva línea de regresión? correlation(X,Y) * sd(X)/sd(Y) Cerraremos con una visualización final del código 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.

```r
#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")

#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
```


child
parent
62
64
66
68
70
72
74
64
66
68
70
72