videos

$$Y_i = \beta_0 + \beta_1 X_i + \epsilon_i$$

* Here the ϵ_i are assumed iid $N(0, \sigma^2)$. * Note, $E[Y_i \mid X_i = x_i] = \mu_i = \beta_0 + \beta_1 x_i$ * Note, $Var(Y_i \mid X_i = x_i) = \sigma^2$.

donde σ^2 es la varianza poblacional

Consider that

$$Y_i = \beta_0 + \beta_1 X_i + \epsilon_i = \beta_0 + a\beta_1 + \beta_1(X_i - a) + \epsilon_i = \tilde{\beta}_0 + \beta_1(X_i - a) + \epsilon_i$$

entonces cando modificamos los datos por ejemplo al centrar lo que cambia es el itercepto

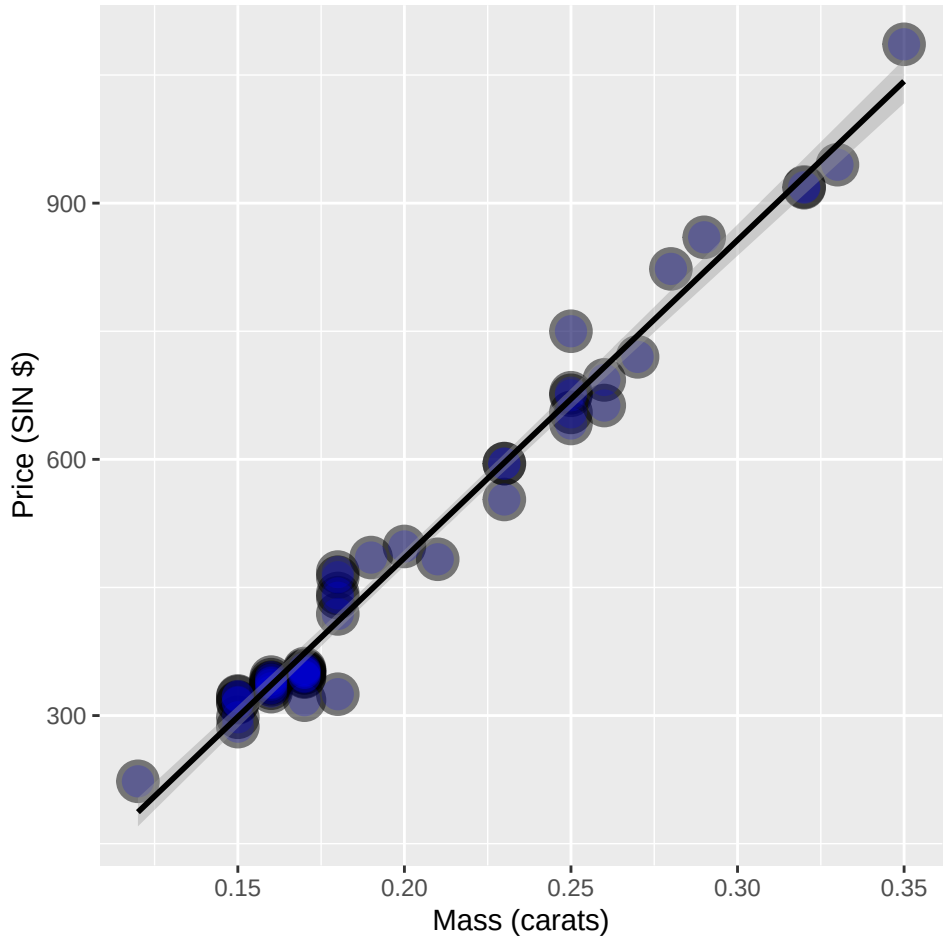
β_1 es el cambio esperado en respuesta a un cambio de 1 unidad en el predictor

$$E[Y \mid X = x + 1] - E[Y \mid X = x] = \beta_0 + \beta_1(x + 1) - (\beta_0 + \beta_1 x) = \beta_1$$

Considere el impacto de cambiar las unidades de X.

$$Y_i = \beta_0 + \beta_1 X_i + \epsilon_i = \beta_0 + \frac{\beta_1}{a}(X_i a) + \epsilon_i = \beta_0 + \tilde{\beta}_1(X_i a) + \epsilon_i$$

Por lo tanto, la multiplicación de X por un factor a da como resultado la división del coeficiente por un factor de a.



```r
fit <- lm(price ~ carat, data = diamond)
coef(fit)
```

```
## (Intercept)       carat
##   -259.6259   3721.0249
```

```r
fit2 <- lm(price ~ I(carat - mean(carat)), data = diamond)
coef(fit2)
```

```
##            (Intercept) I(carat - mean(carat))
##               500.0833             3721.0249
```

la pendiente sigue igual pero la intercepcion ha cambiado

```r
fit3 <- lm(price ~ I(carat * 10), data = diamond)
coef(fit3)
```

```
##   (Intercept) I(carat * 10)
##     -259.6259      372.1025
```

ahora lo que cambia es la pendiente

hay dos formas para predecir, si omites "newdata" devuelve las predicciones de el vector original

```r
newx <- c(0.16, 0.27, 0.34)
coef(fit)[1] + coef(fit)[2] * newx
```

```
## [1]  335.7381  745.0508 1005.5225
```

```r
predict(fit, newdata = data.frame(carat = newx))
```

```
##         1         2         3
##  335.7381  745.0508 1005.5225
```

Una forma de pensar en los residuos e_i es como una estimacion de ϵ_i

- $E[e_i] = 0$.

```r
library(UsingR)
data(diamond)
y <- diamond$price; x <- diamond$carat; n <- length(y)
fit <- lm(y ~ x)
e <- resid(fit)
yhat <- predict(fit)
max(abs(e -(y - yhat)))
```

```
## [1] 9.485746e-13
```

```
max(abs(e - (y - coef(fit)[1] - coef(fit)[2] * x)))
```

```
## [1] 9.485746e-13
```

- Si se incluye una intersección, $\sum_{i=1}^{n} e_i = 0$

```
sum(e)
```

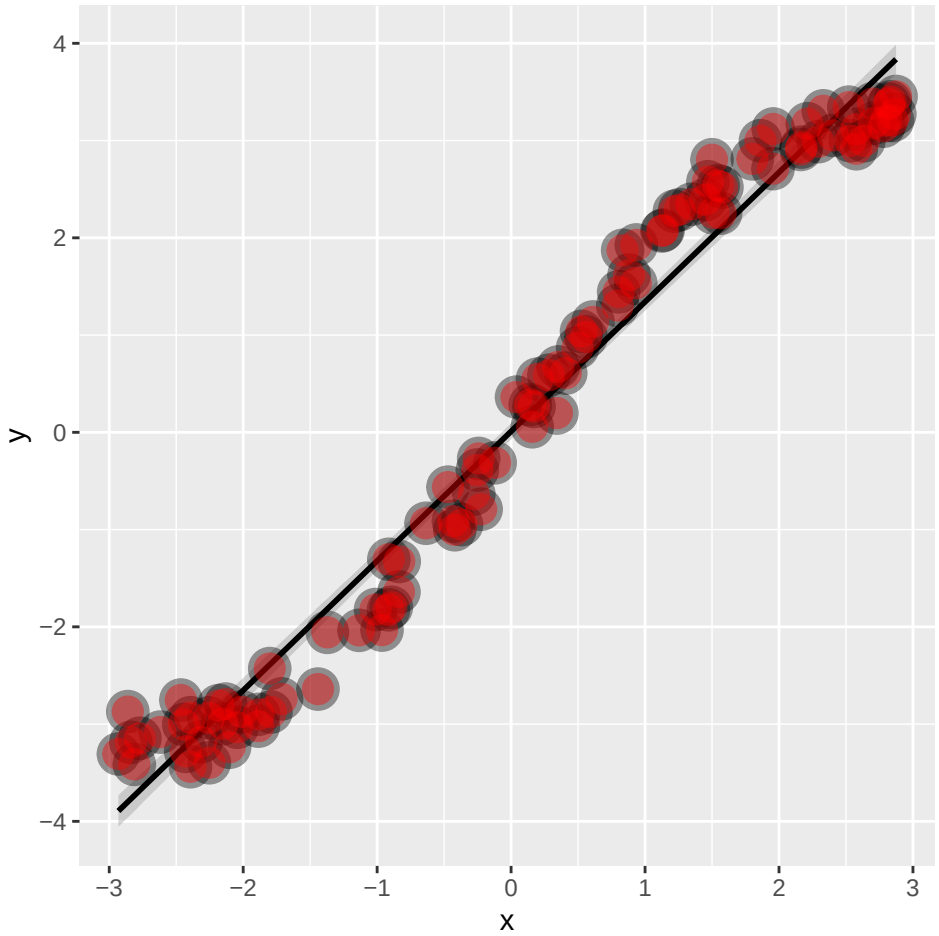
```
## [1] -1.865175e-14
```

- Si es una variable regresora, X_i, es incluida en el modelo $\sum_{i=1}^{n} e_i X_i = 0$.

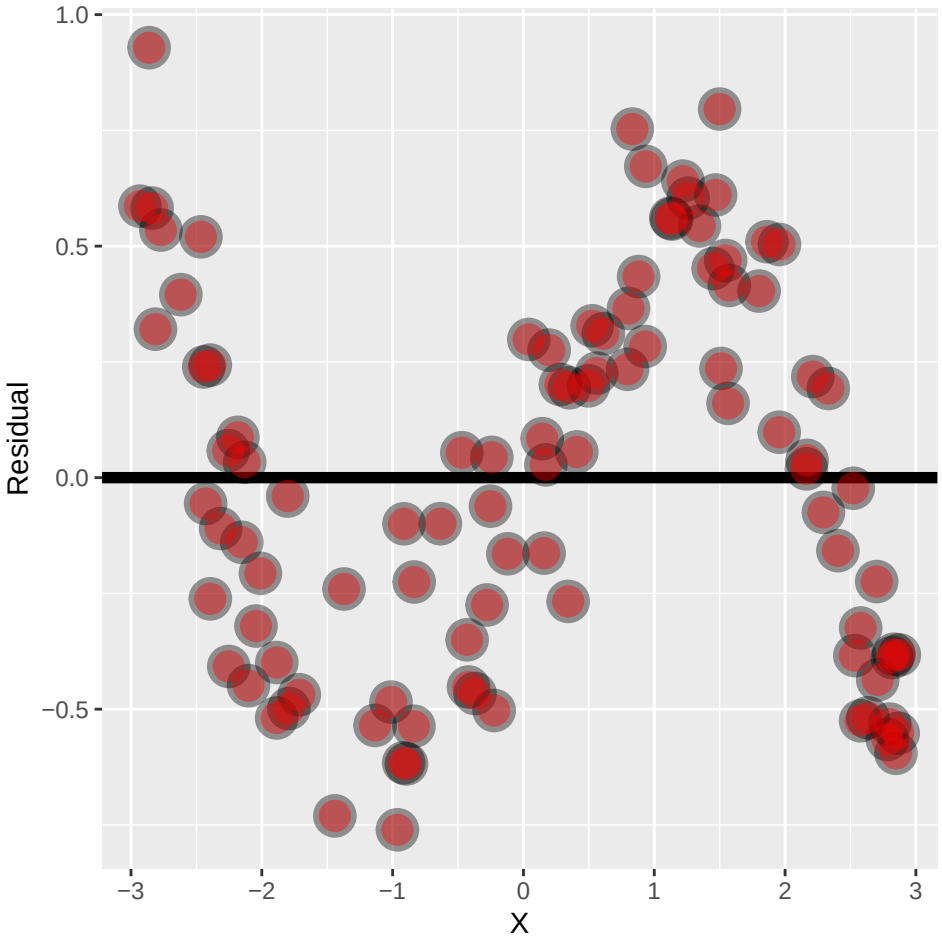
```
sum(e*x)
```

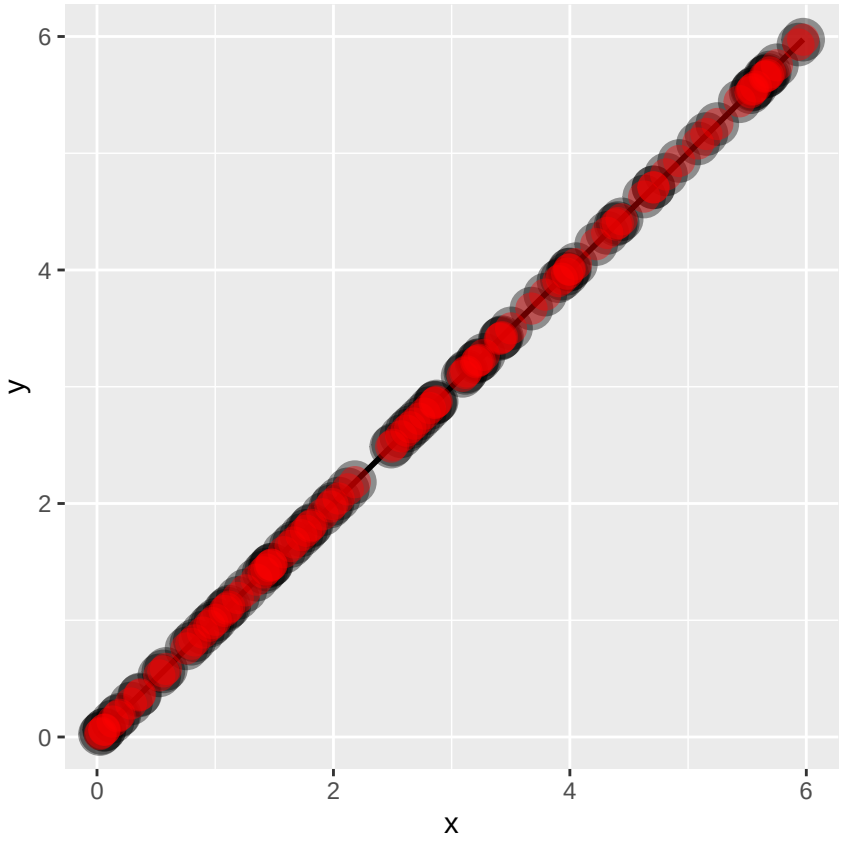
```
## [1] 6.959711e-15
```

- Los residuos son útiles para investigar un ajuste deficiente del modelo.
- Los residuos positivos están por encima de la línea, los residuos negativos están por debajo.
- Se puede pensar en los residuos como el resultado (Y) con la asociación lineal del predictor (X) removida
- Se diferencia la variación residual (variación después de eliminar el predictor) de la variación sistemática (variación explicada por el modelo de regresión).
- Los gráficos de residuos resaltan un ajuste deficiente del modelo.

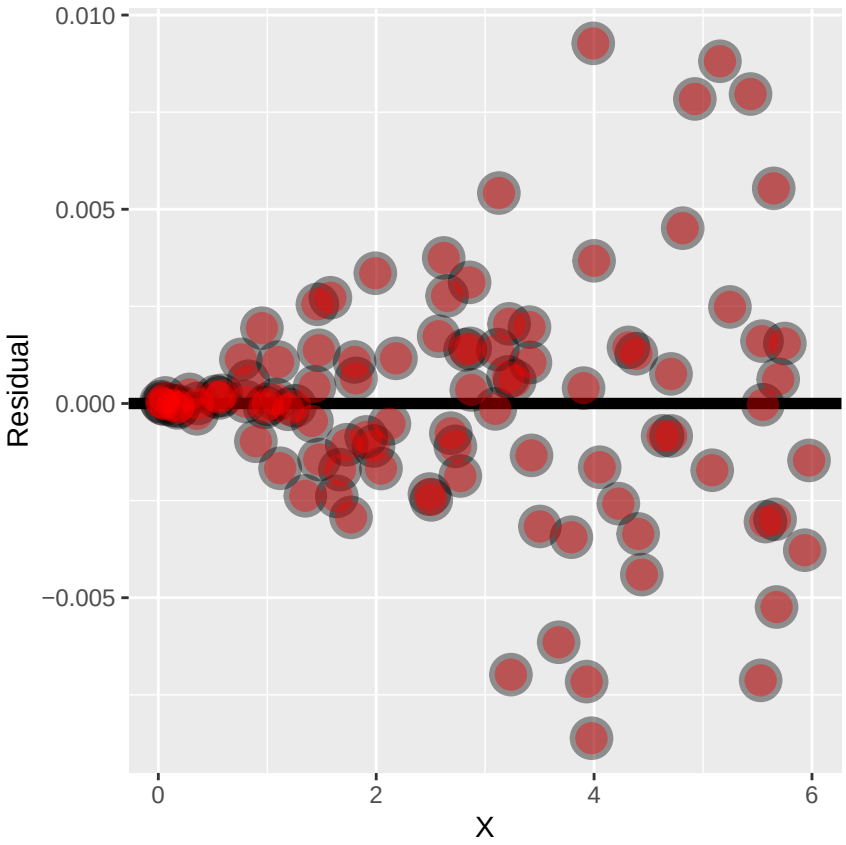


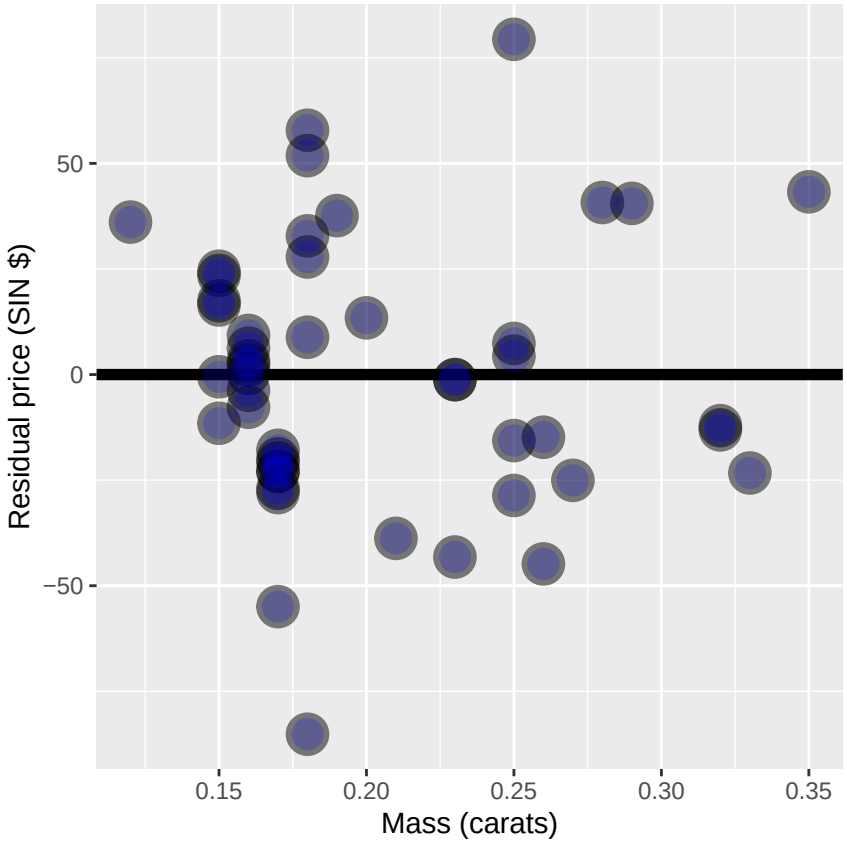
graficacion de los residuos


1.0
0.5
0.0
−0.5
Residual
−3
−2
−1
0
1
2
3
X



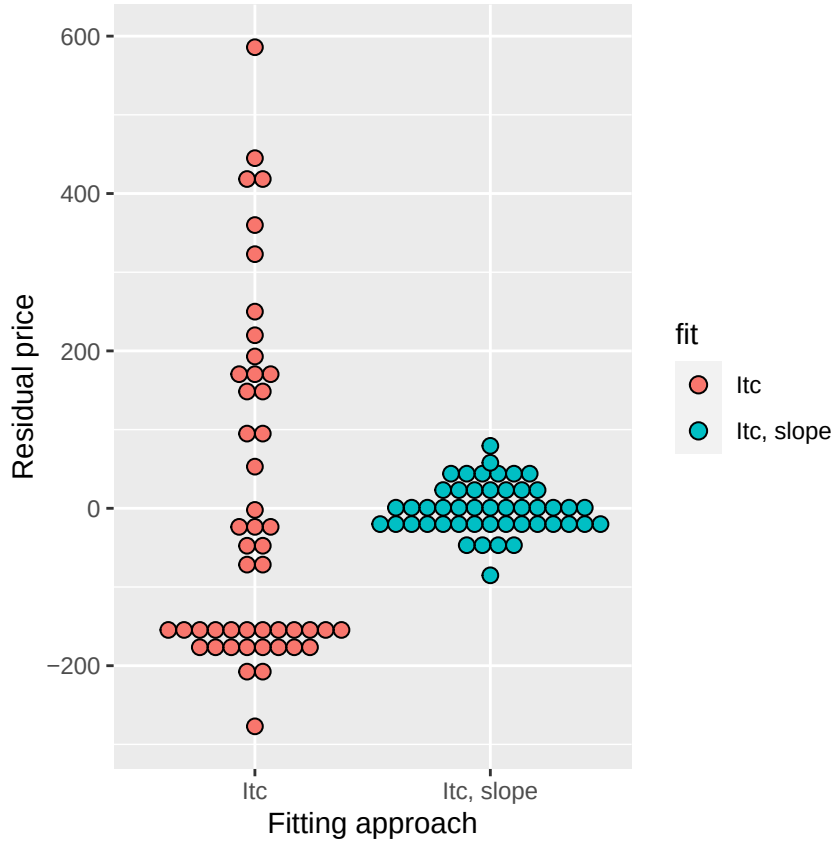
Heteroskedasticity


0.010
0.005
0.000
−0.005
Residual
0
2
4
6
X



diamantes

parece no haber ningun patron esntonces el modelo lineal es bastante bueno



una estimacion de σ^2 es $\frac{1}{n}\sum_{i=1}^{n} e_i^2$, para evitar sesgo muchas personas usan

$$\hat{\sigma}^2 = \frac{1}{n-2}\sum_{i=1}^{n} e_i^2.$$

```
y <- diamond$price; x <- diamond$carat; n <- length(y)
fit <- lm(y ~ x)
summary(fit)$sigma
```

```
## [1] 31.84052
```

```
sqrt(sum(resid(fit)^2) / (n - 2))
```

```
## [1] 31.84052
```

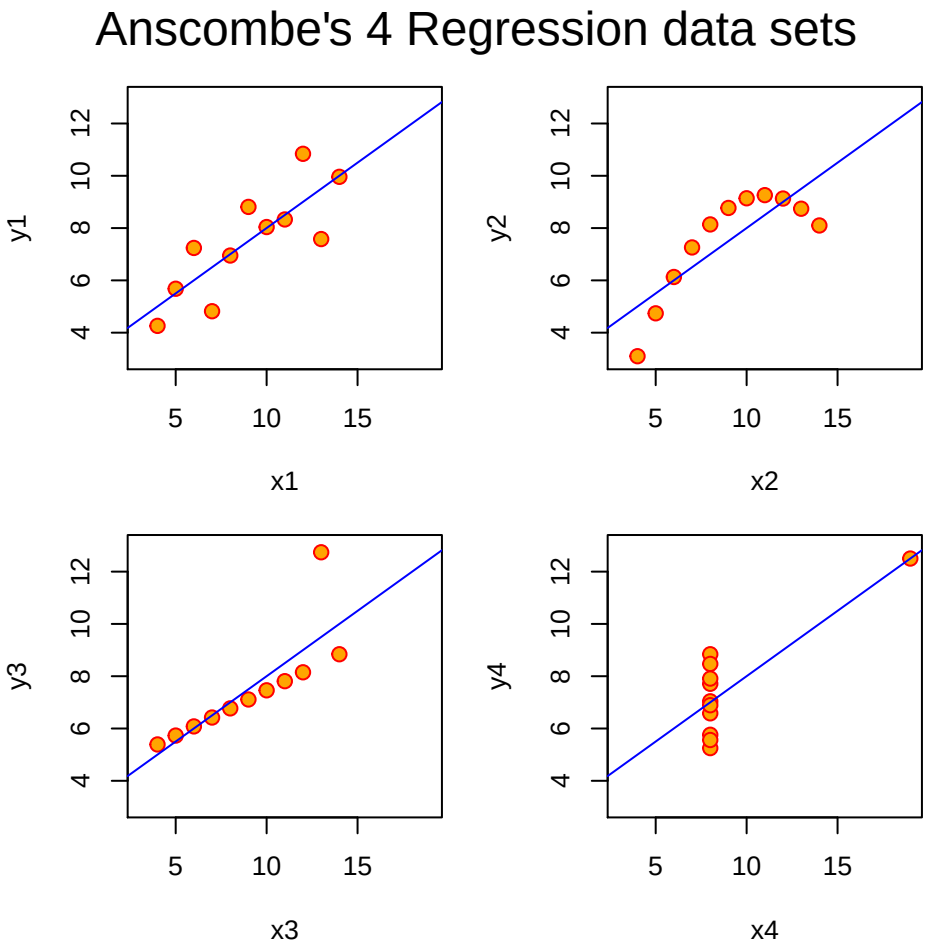
- La variabilidad total en nuestra respuesta es la variabilidad alrededor de una intersección (piense en la regresión de la media solamente) $\sum_{i=1}^{n}(Y_i - \bar{Y})^2$
- La variabilidad de regresión es la variabilidad que se explica al agregar el predictor $\sum_{i=1}^{n}(\hat{Y}_i - \bar{Y})^2$
- La variabilidad del error es lo que queda alrededor de la línea de regresión. $\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$
- Hecho genial

$$\sum_{i=1}^{n}(Y_i - \bar{Y})^2 = \sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2 + \sum_{i=1}^{n}(\hat{Y}_i - \bar{Y})^2$$

- R cuadrado es el porcentaje de la variabilidad total que se explica por la relación lineal con el predictor

$$R^2 = \frac{\sum_{i=1}^{n}(\hat{Y}_i - \bar{Y})^2}{\sum_{i=1}^{n}(Y_i - \bar{Y})^2}$$

- R^2 es el porcentaje de variación explicado por el modelo de regresión.
- $0 \leq R^2 \leq 1$
- R^2 es la correlación de la muestra al cuadrado.
- R^2 puede ser un resumen engañoso del ajuste del modelo.
  - Eliminar datos puede inflar R^2.
  - (Para más adelante). Agregar términos a un modelo de regresión siempre aumenta R^2.



- Básicamente la misma media y varianza de X e Y.
- Correlaciones idénticas (por lo tanto, mismas R^2)
- Misma relación de regresión lineal.
- $\sigma^2_{\hat{\beta}_1} = Var(\hat{\beta}_1) = \sigma^2 / \sum_{i=1}^{n}(X_i - \bar{X})^2$
- $\sigma^2_{\hat{\beta}_0} = Var(\hat{\beta}_0) = \left(\frac{1}{n} + \frac{\bar{X}^2}{\sum_{i=1}^{n}(X_i - \bar{X})^2}\right)\sigma^2$ *En la práctica, σ se reemplaza por su estimación.

- Probablemente no sea sorprendente que bajo iid errores gaussianos

$$\frac{\hat{\beta}_j - \beta_j}{\hat{\sigma}_{\hat{\beta}_j}}$$

  sigue una distribución t con $n-2$ grados de libertad y una distribución normal para n grandes.
- Esto se puede utilizar para crear intervalos de confianza y realizar pruebas de hipótesis.

podemos usar el test $H_0 : \theta = \theta_0$ contra $H_a : \theta >, <, \neq \theta_0$

```
library(UsingR); data(diamond)
y <- diamond$price; x <- diamond$carat; n <- length(y)
beta1 <- cor(y, x) * sd(y) / sd(x)
beta0 <- mean(y) - beta1 * mean(x)
e <- y - beta0 - beta1 * x
sigma <- sqrt(sum(e^2) / (n-2))
ssx <- sum((x - mean(x))^2)
seBeta0 <- (1 / n + mean(x) ^ 2 / ssx) ^ .5 * sigma
seBeta1 <- sigma / sqrt(ssx)
tBeta0 <- beta0 / seBeta0; tBeta1 <- beta1 / seBeta1
pBeta0 <- 2 * pt(abs(tBeta0), df = n - 2, lower.tail = FALSE)
pBeta1 <- 2 * pt(abs(tBeta1), df = n - 2, lower.tail = FALSE)
coefTable <- rbind(c(beta0, seBeta0, tBeta0, pBeta0), c(beta1, seBeta1, tBeta1, pBeta1))
colnames(coefTable) <- c("Estimate", "Std. Error", "t value", "P(>|t|)")
rownames(coefTable) <- c("(Intercept)", "x")
coefTable
```

```
##              Estimate Std. Error   t value      P(>|t|)
## (Intercept) -259.6259   17.31886 -14.99094 2.523271e-19
## x           3721.0249   81.78588  45.49715 6.751260e-40
```

```
fit <- lm(y ~ x);
summary(fit)$coefficients
```

```
##              Estimate Std. Error   t value     Pr(>|t|)
## (Intercept) -259.6259   17.31886 -14.99094 2.523271e-19
## x           3721.0249   81.78588  45.49715 6.751260e-40
```

```
sumCoef <- summary(fit)$coefficients
sumCoef[1,1] + c(-1, 1) * qt(.975, df = fit$df) * sumCoef[1, 2]
```

```
## [1] -294.4870 -224.7649
```

```
(sumCoef[2,1] + c(-1, 1) * qt(.975, df = fit$df) * sumCoef[2, 2]) / 10
```

```
## [1] 355.6398 388.5651
```

```
confint(fit)[1, ]
```

```
##     2.5 %    97.5 %
## -294.4870 -224.7649
```

```r
confint(fit)[2, ]/10
```

```
##    2.5 %   97.5 %
## 355.6398 388.5651
```

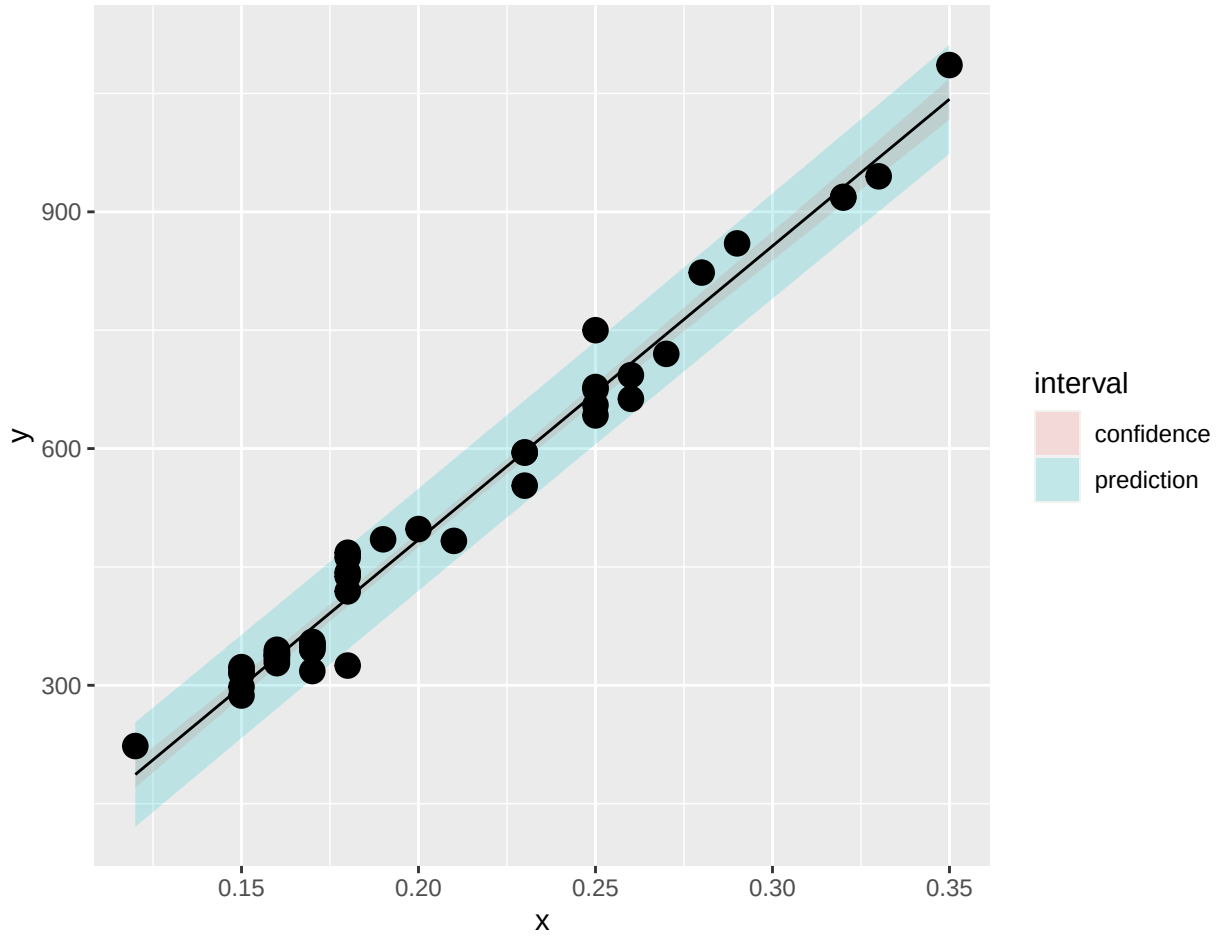
Así que, está diciendo, con 95% de confianza, estimamos que un aumento de 0.1 quilates en el tamaño del diamante va a resultar en un aumento de 356 a 389 en el precio, en dólares de Singapur, algo similar pasa con la interseccion

Considere predecir Y a un valor de X Predecir el precio de un diamante dado el quilate * Predecir la altura de un niño dada la altura de los padres * La estimación obvia para la predicción en el punto x_0 es

$$\hat{\beta}_0 + \hat{\beta}_1 x_0$$

* Se necesita un error estándar para crear un intervalo de predicción. * Hay una distinción entre los intervalos para la línea de regresión en el punto x_0 y la predicción de lo que sería y en el punto x_0.

- Línea en x_0 se, $\hat{\sigma}\sqrt{\frac{1}{n} + \frac{(x_0 - \bar{X})^2}{\sum_{i=1}^{n}(X_i - \bar{X})^2}}$
- Intervalo de predicción se en x_0, $\hat{\sigma}\sqrt{1 + \frac{1}{n} + \frac{(x_0 - \bar{X})^2}{\sum_{i=1}^{n}(X_i - \bar{X})^2}}$



confidence es en intervalo para la linea, y prediccion es un intervalo para la prediccion

variacion residual

Como se muestra en las diapositivas, los residuos son útiles para indicar qué tan bien los puntos de datos se ajustan a un modelo estadístico. "Pueden considerarse como el resultado (Y) con la asociación lineal del predictor (X) eliminada. Se diferencia la variación residual (variación después de eliminar el predictor) de la variación sistemática (variación explicada por el modelo de regresión)".

También se puede demostrar que, dado un modelo, la estimación de máxima verosimilitud de la varianza del error aleatorio es el residuo cuadrático medio. Sin embargo, dado que nuestro modelo lineal con un predictor requiere dos parámetros, solo tenemos (n-2) grados de libertad. Por lo tanto, para calcular un residuo cuadrado "promedio" para estimar la varianza, usamos la fórmula 1 / (n-2) * (la suma de los residuos cuadrados). Si dividimos la suma de los residuos al cuadrado por n, en lugar de n-2, el resultado daría una estimación sesgada.

Para ver esto, usaremos nuestros datos de altura favoritos de Galton. Primero regenere la línea de regresión y llámela ajuste. Utilice la función R lm y recuerde que, por defecto, su primer argumento es una fórmula como "hijo ~ padre" y el segundo es el conjunto de datos, en este caso galton.

```
library(UsingR)
data(galton)
fit <- lm(child ~ parent, galton)
n<-928
```

Primero, usaremos los residuos (fit$residual) de nuestro modelo para estimar la desviación estándar (sigma) del error. Ya hemos definido n para usted como el número de puntos en el conjunto de datos de Galton (928).

```
sqrt(sum(fit$residuals^2) / (n - 2))
```

```
## [1] 2.238547
```

ahora vamos a comparar con el sigma dado

```
summary(fit)$sigma
```

```
## [1] 2.238547
```

otra forma es

```
sqrt(deviance(fit)/(n-2))
```

```
## [1] 2.238547
```

Otro dato útil mostrado en las diapositivas fue

Variación total = Variación residual + Variación de regresión

El término R ^ 2 representa el porcentaje de variación total descrito por el modelo, la variación de regresión (el término sobre el que no preguntamos en las preguntas de opción múltiple anteriores). Además, dado que es un porcentaje, necesitamos una razón o fracción de sumas de cuadrados. Hagamos esto ahora para nuestros datos de Galton.

Comenzaremos con pasos sencillos. Calcula la media de las alturas de los niños y guárdala en una variable llamada mu. Recuerde que hacemos referencia a las alturas de los niños con la expresión 'galton$child' y las alturas de los padres con la expresión 'galton$parent'.

```
mu <- mean(galton$child)
```

Recuerde que centrar los datos significa restar la media de cada punto de datos. Ahora calcule la suma de los cuadrados de las alturas de los niños centrados y almacene el resultado en una variable llamada sTot. Esto representa la variación total de los datos.

```
sTot <- sum((galton$child-mu)^2)
```

Ahora cree la variable sRes. Utilice la desviación de la función R para calcular la suma de los cuadrados de los residuos. Estas son las distancias entre la altura de los niños y la línea de regresión. Esto representa la variación residual.

```
sRes <- deviance(fit)
```

Finalmente, la razón sRes / sTot representa el porcentaje de la variación total aportada por los residuos. Para encontrar el porcentaje contribuido por el modelo, es decir, la variación de regresión, reste la fracción sRes / sTot de 1. Este es el valor R ^ 2.

```
1-sRes/sTot
```

```
## [1] 0.2104629
```

Para divertirse, puede comparar su resultado con los valores que se muestran en el resumen (ajuste) $ r.squared para ver si le resulta familiar. Hacer esto ahora.

```
summary(fit)$r.squared
```

```
## [1] 0.2104629
```

Para ver algo de magia real, calcule el cuadrado de la correlación de los datos de Galton, los niños y los padres. Utilice la función R cor.

```
cor(galton$parent,galton$child)^2
```

```
## [1] 0.2104629
```

Ahora resumiremos datos útiles sobre R ^ 2. Es el porcentaje de variación explicado por el modelo de regresión. Como porcentaje está entre 0 y 1. También es igual a la correlación de la muestra al cuadrado. Sin embargo, R ^ 2 no cuenta toda la historia.

regresión multivariable

En esta lección ilustraremos que la regresión en muchas variables equivale a una serie de regresiones en una. Usando la regresión en una variable, mostraremos cómo eliminar cualquier regresor elegido, reduciendo así una regresión en N variables, a una regresión en N-1. Por lo tanto, si sabemos cómo hacer una regresión en 1 variable, podemos hacer una regresión en 2. Una vez que sabemos cómo hacer una regresión en 2 variables, podemos hacer una regresión en 3, y así sucesivamente. Comenzamos con los datos de Galton y una revisión de cómo eliminar la intersección restando las medias.

Cuando realizamos una regresión en una variable, como lm (hijo ~ padre, galton), obtenemos dos coeficientes, una pendiente y una intersección. La intersección es realmente el coeficiente de un regresor especial que tiene el mismo valor, 1, en cada muestra. La función, lm, incluye este regresor por defecto.

Lo demostraremos sustituyendo un regresor propio de todos unos. Este regresor debe tener el mismo número de muestras que galton (928). Cree tal objeto y asígnele el nombre ones, usando ones <- rep (1, nrow (galton)), o alguna expresión equivalente.

```
ones <- rep (1, nrow (galton))
```

Los datos de Galton ya se han cargado. La intersección predeterminada se puede excluir utilizando -1 en la fórmula. Realice una regresión que sustituya nuestro regresor, unos, por el predeterminado usando lm (hijo ~ unos + padre -1, galton). Como queremos que se imprima el resultado, no lo asigne a una variable.

```
lm(child ~ ones + parent -1, galton)
```

```
##
## Call:
## lm(formula = child ~ ones + parent - 1, data = galton)
##
## Coefficients:
##    ones   parent
## 23.9415   0.6463
```

El coeficiente de unos es 23,9415. Ahora use el valor predeterminado, lm (hijo ~ padre, galton), para mostrar que la intersección tiene el mismo valor. Esta vez, NO suprima la intersección con -1.

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

```
##
## Call:
## lm(formula = child ~ parent, data = galton)
##
## Coefficients:
## (Intercept)       parent
##     23.9415       0.6463
```

En lecciones anteriores demostramos que la línea de regresión dada por lm (hijo ~ padre, galton) pasa por el punto x = media (padre), y = media (hijo). También mostramos que si restamos la media de cada variable, la línea de regresión pasa por el origen, x = 0, y = 0, por lo que su intersección es cero. Por lo tanto, al restar las medias, eliminamos uno de los dos regresores, la constante, dejando solo uno, el padre. El coeficiente del regresor restante es la pendiente.

Restar los medios para eliminar la intersección es un caso especial de una técnica general que a veces se denomina Eliminación Gaussiana. Como se aplica aquí, la técnica general es elegir un regresor y reemplazar todas las demás variables por los residuos de sus regresiones contra ese.

Suponga, como se afirma, que restar la media de una variable es un caso especial de reemplazar la variable con un residual. En este caso especial, sería el residuo de una regresión contra la constante 1

La media de una variable es el coeficiente de su regresión contra la constante, 1. Por lo tanto, restar la media es equivalente a reemplazar una variable por el residual de su regresión contra 1. En una fórmula R, el regresor constante se puede representar por un 1 en el lado derecho. Por lo tanto, la expresión lm (niño ~ 1, galton), retrocede niño contra la constante 1. Recuerde que en los datos de galton, la estatura media de un niño era 68.09 pulgadas. Utilice lm (hijo ~ 1, galton) para comparar el coeficiente resultante (la intersección) y la altura media de 68,09. Como queremos que se imprima el resultado, no le asigne un nombre.

```
lm(child ~ 1, galton)
```

```
##
## Call:
## lm(formula = child ~ 1, data = galton)
##
## Coefficients:
## (Intercept)
##       68.09
```

Para ilustrar el caso general, usaremos los datos de árboles del paquete de conjuntos de datos. La idea es predecir el volumen de madera que podría producir un árbol a partir de las medidas de su altura y circunferencia. Para evitar tratar la intersección como un caso especial, hemos agregado una columna de unos a los datos que usaremos en su lugar. Tómese un momento para inspeccionar los datos usando View (árboles) o head (árboles)

```
head(trees)
```

```
##   Girth Height Volume
## 1   8.3     70   10.3
## 2   8.6     65   10.3
## 3   8.8     63   10.2
## 4  10.5     72   16.4
## 5  10.7     81   18.8
## 6  10.8     83   19.7
```

Se ha copiado y obtenido un archivo de código relevante en su directorio de trabajo. El archivo elimination.R debería haber aparecido en su editor. Si no es así, ábralo manualmente.

```
# Regress the given variable on the given predictor,
# suppressing the intercept, and return the residual.
regressOneOnOne <- function(predictor, other, dataframe){
  # Point A. Create a formula such as Girth ~ Height -1
  formula <- paste0(other, " ~ ", predictor, " - 1")
  # Use the formula in a regression and return the residual.
  resid(lm(formula, dataframe))
}

# Eliminate the specified predictor from the dataframe by
# regressing all other variables on that predictor
# and returning a data frame containing the residuals
# of those regressions.
eliminate <- function(predictor, dataframe){
  # Find the names of all columns except the predictor.
  others <- setdiff(names(dataframe), predictor)
  # Calculate the residuals of each when regressed against the given predictor
  temp <- sapply(others, function(other)regressOneOnOne(predictor, other, dataframe))
  # sapply returns a matrix of residuals; convert to a data frame and return.
  as.data.frame(temp)
}
```

La técnica general consiste en elegir un predictor y reemplazar todas las demás variables por los residuos de sus regresiones contra ese. La función, regressOneOnOne, en eliminar.R realiza el primer paso de este

proceso. Dado el nombre de un predictor y otra variable, otro, devuelve el residuo de otro cuando se realiza una regresión contra el predictor. En su primera línea, denominada Punto A, crea una fórmula. Supongamos que los predictores fueran 'Circunferencia' y otros fueran 'Volumen'. ¿Qué fórmula crearía?

La función restante, eliminar, aplica regressOneOnOne a todas las variables excepto a un predictor dado y recopila los residuos en un marco de datos. Primero mostraremos que cuando eliminamos un regresor de los datos, una regresión del resto producirá sus coeficientes correctos. (Por supuesto, faltará el coeficiente del regresor eliminado, pero más sobre eso más adelante).

Como referencia, cree un modelo llamado ajuste, basado en los tres regresores, Circunferencia, Altura y Constante, y asigne el resultado a una variable llamada ajuste. Utilice una expresión como ajuste <- lm (Volumen ~ Circunferencia + Altura + Constante -1, árboles). No olvide el -1 y asegúrese de nombrar el modelo adecuado para su uso posterior

Ahora eliminemos Girth del conjunto de datos. Llame al conjunto de datos reducido trees2 para indicar que solo tiene 2 regresores. Utilice la expresión árboles2 <- eliminar ("Circunferencia", árboles).

```
trees2 <- eliminate("Girth", trees)
head(trees2)
```

```
##     Height     Volume
## 1 24.38809  -9.793826
## 2 17.73947 -10.520109
## 3 14.64038 -11.104298
## 4 14.29818  -9.019900
## 5 22.19910  -7.104089
## 6 23.64956  -6.446183
```

Ahora cree un modelo, llamado fit2, usando el conjunto de datos reducido. Utilice una expresión como fit2 <- lm (Volumen ~ Altura + Constante -1, árboles2). No olvides usar -1 en la fórmula.

```
Constant<-rep (1, nrow (galton),length(trees2$Height))
fit2 <- lm(Volume ~ Height + Constant -1, trees2)
```

ahora comparemos los datos

```
lapply(list(fit, fit2), coef)
```

```
## [[1]]
## (Intercept)      parent
##  23.9415302   0.6462906
##
## [[2]]
##     Height   Constant
## -0.4717493 -0.3955385
```

Por supuesto, falta el coeficiente de la variable eliminada. Una forma de obtenerlo sería volver a los datos originales, los árboles, eliminar un regresor diferente, como Altura, y hacer otra regesión de 2 variables, como se indicó anteriormente. Hay formas mucho más eficientes, pero la eficiencia no es el objetivo de esta demostración. Hemos mostrado cómo reducir una regresión en 3 variables a una regresión en 2. Podemos ir más allá y eliminar otra variable, reduciendo una regresión en 2 variables a una regresión en 1.

Aquí está el paso final. Hemos utilizado eliminar ("Altura", árboles2) para reducir los datos al resultado, Volumen y al regresor constante. Hemos hecho una regresión de Volumen en Constante e impreso el coeficiente como se muestra en el comando sobre la respuesta. Como puede ver, el coeficiente de constante concuerda con los valores anteriores.

```
lm(formula = Volume ~ Constant - 1, data = eliminate("Height",
    trees2))
```

```
##
## Call:
## lm(formula = Volume ~ Constant - 1, data = eliminate("Height",
##     trees2))
##
## Coefficients:
## Constant
##  -0.3777
```

Hemos ilustrado que la regresión en muchas variables equivale a una serie de regresiones en una. Los algoritmos reales utilizados por funciones como lm son más eficientes, pero son computacionalmente equivalentes a lo que hemos hecho. Es decir, los algoritmos usan pasos equivalentes pero los combinan de manera más eficiente y abstracta. Esto completa la lección.

ejemplo multiples variables

En esta lección, veremos algunos ejemplos de modelos de regresión con más de una variable. Comenzaremos con los datos suizos que nos hemos tomado la libertad de cargar para usted. Estos datos son parte del paquete de conjuntos de datos de R. Se recopiló en 1888, una época de cambio demográfico en Suiza, y midió seis cantidades en 47 provincias de habla francesa de Suiza. Usamos el código de las diapositivas (los pares de funciones R) para mostrar aquí una matriz de diagramas de dispersión de 6 por 6 que muestran relaciones por pares entre las variables. Todas las variables, excepto la fecundidad, son proporciones de población. Por ejemplo, "Examen" muestra el porcentaje de reclutas que recibieron la calificación más alta en un examen del ejército y "Educación" el porcentaje de reclutas con educación más allá de la escuela primaria.

```
require(datasets); data(swiss)
```

Primero, use la función R lm para generar el modelo lineal "todos" en el que la fertilidad es la variable dependiente de todas las demás. Utilice la abreviatura R "." para representar las cinco variables independientes en la fórmula pasada a lm. Recuerde que los datos son "suizos"

```
all <- lm(Fertility ~ ., swiss)
summary(all)
```

```
##
## Call:
## lm(formula = Fertility ~ ., data = swiss)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -15.2743  -5.2617   0.5032   4.1198  15.3213
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)      66.91518   10.70604   6.250 1.91e-07 ***
## Agriculture      -0.17211    0.07030  -2.448  0.01873 *
## Examination      -0.25801    0.25388  -1.016  0.31546
```

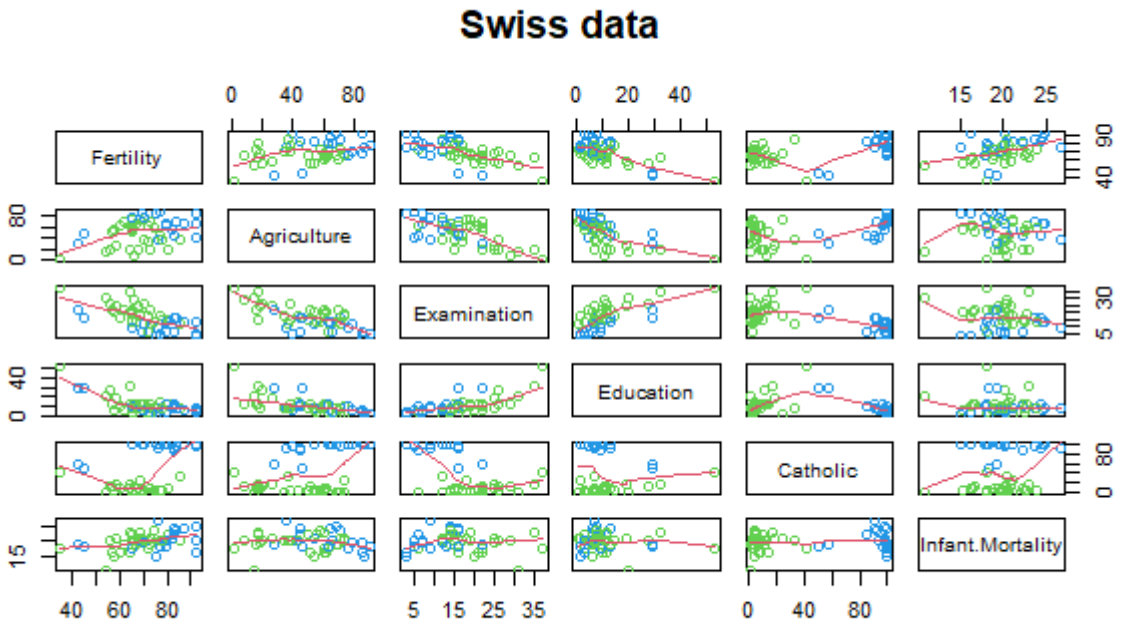


Figure 1: A caption

```
## Education        -0.87094    0.18303  -4.758 2.43e-05 ***
## Catholic          0.10412    0.03526   2.953  0.00519 **
## Infant.Mortality  1.07705    0.38172   2.822  0.00734 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.165 on 41 degrees of freedom
## Multiple R-squared:  0.7067, Adjusted R-squared:  0.671
## F-statistic: 19.76 on 5 and 41 DF,  p-value: 5.594e-10
```

Recuerde que las Estimaciones son los coeficientes de las variables independientes del modelo lineal (todas las cuales son porcentajes) y reflejan un cambio estimado en la variable dependiente (fecundidad) cuando cambia la correspondiente variable independiente. Entonces, por cada aumento del 1% en el porcentaje de hombres involucrados en la agricultura como ocupación, esperamos una disminución de .17 en la fertilidad, manteniendo todas las demás variables constantes; por cada aumento del 1% en el catolicismo, esperamos un aumento de .10 en la fertilidad, manteniendo constantes todas las demás variables.

El "*" al final de la fila indica que la influencia de la agricultura en la fertilidad es significativa. ¿A qué nivel alfa es significativa la prueba t de Agricultura? r=0.05

ahora genere el resumen de otro modelo lineal (no lo almacene en una nueva variable) en el que la fertilidad depende solo de la agricultura.

```
summary(lm(Fertility ~ Agriculture, swiss))
```

```
##
## Call:
## lm(formula = Fertility ~ Agriculture, data = swiss)
```

```
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -25.5374  -7.8685  -0.6362   9.0464  24.4858
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 60.30438    4.25126  14.185   <2e-16 ***
## Agriculture  0.19420    0.07671   2.532   0.0149 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11.82 on 45 degrees of freedom
## Multiple R-squared:  0.1247, Adjusted R-squared:  0.1052
## F-statistic: 6.409 on 1 and 45 DF,  p-value: 0.01492
```

Lo interesante es que el signo del coeficiente de Agricultura cambió de negativo (cuando todas las variables estaban incluidas en el modelo) a positivo (cuando el modelo solo consideraba Agricultura). Evidentemente, la presencia de los otros factores incide en la influencia que tiene la agricultura sobre la fertilidad.

Consideremos la relación entre algunos de los factores. ¿Cómo espera que se relacionen el nivel de educación y el rendimiento en un examen? r=They would be correlated

```
cor(swiss$Examination,swiss$Education)
```

```
## [1] 0.6984153
```

ahora encuentra la correlacion entre agricultura y educacion

```
cor(swiss$Agriculture,swiss$Education)
```

```
## [1] -0.6395225
```

La correlación negativa (-.6395) entre agricultura y educación podría estar afectando la influencia de la agricultura en la fertilidad. Cargué y obtuve el archivo swissLMs.R en su directorio de trabajo. En ella hay una función makelms () que genera una secuencia de cinco modelos lineales. Cada modelo tiene una variable independiente más que el modelo anterior, por lo que el primero tiene solo una variable independiente, Agricultura, y el último tiene las 5. Intenté cargar el código fuente en su editor. Si no lo he hecho, abra el archivo manualmente para que pueda ver el código.

```
makelms <- function(){
  # Store the coefficient of linear models with different independent variables
  cf <- c(coef(lm(Fertility ~ Agriculture, swiss))[2],
          coef(lm(Fertility ~ Agriculture + Catholic,swiss))[2],
          coef(lm(Fertility ~ Agriculture + Catholic + Education,swiss))[2],
          coef(lm(Fertility ~ Agriculture + Catholic + Education + Examination,swiss))[2],
          coef(lm(Fertility ~ Agriculture + Catholic + Education + Examination +Infant.Mortality, swiss
  print(cf)
}

# Regressor generation process 1.
rgp1 <- function(){
```

```
  print("Processing. Please wait.")
  # number of samples per simulation
  n <- 100
  # number of simulations
  nosim <- 1000
  # set seed for reproducability
  set.seed(4321)
  # Point A:
  x1 <- rnorm(n)
  x2 <- rnorm(n)
  x3 <- rnorm(n)
  # Point B:
  betas <- sapply(1 : nosim, function(i)makelms(x1, x2, x3))
  round(apply(betas, 1, var), 5)
}

# Regressor generation process 2.
rgp2 <- function(){
  print("Processing. Please wait.")
  # number of samples per simulation
  n <- 100
  # number of simulations
  nosim <- 1000
  # set seed for reproducability
  set.seed(4321)
  # Point C:
  x1 <- rnorm(n)
  x2 <- x1/sqrt(2) + rnorm(n) /sqrt(2)
  x3 <- x1 * 0.95 + rnorm(n) * sqrt(1 - 0.95^2)
  # Point D:
  betas <- sapply(1 : nosim, function(i)makelms(x1, x2, x3))
  round(apply(betas, 1, var), 5)
}
```

Ahora ejecute la función makelms () para ver cómo la adición de variables afecta el coeficiente de Agricultura en los modelos.

```
makelms ()
```

```
## Agriculture Agriculture Agriculture Agriculture Agriculture
##   0.1942017   0.1095281  -0.2030377  -0.2206455  -0.1721140
```

la adicion de educacion cambio

Ahora mostraremos lo que sucede cuando agregamos una variable que no proporciona nueva información lineal a un modelo. Cree una variable ec que sea la suma de Swiss $ Examination y Swiss $ Catholic.

```
ec <- swiss$Examination+swiss$Catholic
```

Ahora genere un nuevo modelo de efit con Fertilidad como variable dependiente y las 5 restantes de las variables originales Y ec como variables independientes. Utilice la abreviatura R “. + Ec” para el lado derecho de la fórmula

```
efit <- lm(Fertility ~ . + ec, swiss)
efit
```

```
##
## Call:
## lm(formula = Fertility ~ . + ec, data = swiss)
##
## Coefficients:
##      (Intercept)       Agriculture       Examination         Education
##          66.9152           -0.1721           -0.2580           -0.8709
##         Catholic  Infant.Mortality                ec
##           0.1041            1.0770                NA
```

reste ahora los coeficientes

```
all$coefficients-efit$coefficients
```

```
##      (Intercept)      Agriculture      Examination        Education
##                0                0                0                0
##         Catholic Infant.Mortality               ec
##                0                0               NA
```

agregar la variable no afecta al modelo