Regression Models

Sébastien Plat

Contents

Regression Models

Definition

Regression Models are used to predict outcomes based on existing data.

Notations

We commonly use:

- $X_1, X_2, \ldots, X_n$ to describe n data points
- Greek letters for things we don't know, eg. μ for a mean we'd like to estimate
- $\bar{X}$ for the empirical mean
- $\tilde{X}_i = X_i - \bar{X}$ for data with mean 0 ("centering" the random variables)
- S^2 the empirical variance
- X_i/S for data with variance 1 ("scaling" the random variables)
- $Z_i = \tilde{X}_i/S$ for data with mean 0 and variance 1 ("normalizing" the ramdom variable)
- $\hat{X}$ for the estimate of X using data

We will also use ML estimates for **Maximum Likelihood** estimates.

Reminder:

$$\bar{X} = \frac{1}{n}\sum_{i=1}^{n} X_i$$

$$S^2 = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \bar{X})^2 = \frac{1}{n-1}\left(\sum_{i=1}^{n} X_i^2 - n\bar{X}^2\right)$$

Covariance and Correlation

The Covariance $Cov(X, Y)$ is a measure of **how much two random variables change together**, ie. the degree to which two random variables tend to deviate from their expected values in similar ways.

$$Cov(X, Y) = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y}) = \frac{1}{n-1}\left(\sum_{i=1}^{n} X_i Y_i - n\bar{X}\bar{Y}\right)$$

The Correlation coefficient $Cor(X, Y)$ measures the **strength of the linear relationship** between X Y. The closer it is from -1 or 1, the stronger the relationship *(0 means no relationship)*.

It is calculated by **normalizing the Covariance**:

$$Cor(X, Y) = \frac{Cov(X, Y)}{S_x S_y}$$

where S_x and S_y are the standard deviations estimates for X_i and Y_i.

Linear Least Square Errors

Definition

Let's try to describe the relation between two variables X_i and Y_i as linear: $Y_i \simeq \beta_0 + \beta_1 \times X_i$. We want to find the **line of best fit**, ie. the line that is the best approximation of the given set of data.

The best approximation is the line that **minimizes the square errors**, ie. the **difference between the predictions and actual outcomes**:

$$Min(\sum_{i=1}^{n}(Y_i - (\beta_0 + \beta_1 X_i))^2 \ \ when \ \ \beta_1 = Cor(X,Y) \times \frac{Sd(Y)}{Sd(X)} \ \ and \ \ \beta_0 = \bar{Y} - \hat{\beta}_1 \bar{X}$$

Note: if we choose $\beta_1 = 0$, then $\beta_0 = \bar{Y}$.

A few additional points:

- The line passes through the point $(\bar{X}, \bar{Y})$
- The slope of the regression line with X as the outcome and Y as the predictor is $Cor(Y,X)Sd(X)/Sd(Y)$
- The slope is unchanged for centered data $(X_i - \bar{X}, Y_i - \bar{Y})$
- For centered data, the regression goes through the origin (as both means equal zero)
- For normalized data $\{\frac{X_i - \bar{X}}{Sd(X)}, \frac{Y_i - \bar{Y}}{Sd(Y)}\}$, the slope is $Cor(Y,X)$

Interpreting regression coefficients

ITC

The **intercept** (ITC) β_0 is the **expected value of the response when the predictor is 0**.

It is not always meaningful to the study, so we can use $\tilde{X}_i = X_i - \bar{X}$ instead: the ITC is the expected response at the average X value.

Slope

The **slope** β_1 is the **expected change in response for a 1 unit change in the predictor**.

If we change the unit of X, we must adjust β_1 to keep the same slope: $oldX = \alpha \, newX \implies new\beta_1 = old\beta_1/\alpha$.

Regression to the mean

According to Wikipedia:

> Regression toward (or to) the mean: if a variable is extreme on its first measurement, it will tend to be closer to the average on its second measurement. If it is extreme on its second measurement, it will tend to have been closer to the average on its first.

Regression Model with additive Gaussian errors

Definition

The Linear Least Square Errors is an estimation tool. We can develop a **probabilistic model** for Linear Regression by adding **Gaussian errors** to the linear fit:

$$Y_i = \beta_0 + \beta_1 X_i + \epsilon_i \quad where\ \epsilon_i\ are\ iid\ N(0, \sigma^2)$$

- $E[Y_i \mid X_i = x_i] = \beta_0 + \beta_1 x_i$
- $Var(Y_i \mid X_i = x_i) = \sigma^2$

Residuals

Residuals are the difference between the observed outcome Y_i and prediction $\hat{Y}_i$. They are estimates of ϵ_i.

$$e_i = Y_i - \hat{Y}_i$$

- $E[e_i] = 0$: the residuals are as likely to be positive as negative
- If an intercept is included, the model goes through $(\bar{X}, \bar{Y})$ so: $\sum_{i=1}^{n} e_i = 0$
- If a regressor variable X_i is included in the model: $\sum_{i=1}^{n} e_i X_i = 0$
- Residuals are the parts of outcome (Y) not explained by its linear association with predictor X
- So they can highlight poor model fit

Residuals Variance

We assume that $\epsilon_i \sim N(0, \sigma^2)$. As $mean(e_i) = 0$, The ML estimate of σ^2 is $\frac{1}{n}\sum_{i=1}^{n} e_i^2$, the average squared residual.

To take actual degrees of freedom (intercept & covariance bring two constraints), we most commonly use:

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

The $n-2$ instead of n is so that $E[\hat{\sigma}^2] = \sigma^2$.

The residuals variance can easily be calculated in R:

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

Variability of Data

Definition

The total variability of our outcome is the variability around its mean: $\sum_{i=1}^{n}(Y_i - \bar{Y})^2 = nVar(data)$.

We can split this variability in two:

- **Regression Variability**: variability explained by the Linear Model $\sum_{i=1}^{n}(\hat{Y}_i - \bar{Y})^2 = nVar(est)$
- **Error/Residual Variability**: what's leftover around the regression line $\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2 = nVar(res)$

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

Total Variability = Residual Variability + Regression Variability

Note: as variances > 0, the variance of residuals is always less than the variance of data

Correlation Coefficient

The percentage of the total variability explained by the Linear Model is:

$$R^2 = \frac{Regression\ Variability}{Total\ Variability} = 1 - \frac{Residual\ Variability}{Total\ Variability}$$

As $(\hat{Y}_i - \bar{Y}) = \beta_1(X_i - \bar{X})$, we can prove that:

$$R^2 = Cor(X, Y)^2$$

Inference

Definition

We can compute a Confidence Interval for β_0 and β_1 estimates under iid Gaussian errors $\epsilon \sim N(0, \sigma^2)$:

$$\sigma^2_{\hat{\beta}_1} = Var(\hat{\beta}_1) = \frac{\sigma^2}{\sum_{i=1}^{n}(X_i - \bar{X})^2}$$

$$\sigma^2_{\hat{\beta}_0} = Var(\hat{\beta}_0) = \left(\frac{1}{n} + \frac{\bar{X}^2}{\sum_{i=1}^{n}(X_i - \bar{X})^2}\right)\sigma^2$$

In practice, σ is replaced by its estimate.

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

follows a t distribution with $n-2$ degrees of freedom and a normal distribution for large n. This can be used to create confidence intervals and perform hypothesis tests.

Prediction Intervals

A **standard error** is needed to create a prediction interval. There's a distinction between:

- intervals for the **regression line** at point x_0
- the **prediction** of what a y would be at point x_0

The standard error for predictions is bigger than for the regression line:

- Line at x_0: $SE = \hat{\sigma}\sqrt{\frac{1}{n} + \frac{(x_0 - \bar{X})^2}{\sum_{i=1}^{n}(X_i - \bar{X})^2}}$
- Prediction at x_0: $SE = \hat{\sigma}\sqrt{1 + \frac{1}{n} + \frac{(x_0 - \bar{X})^2}{\sum_{i=1}^{n}(X_i - \bar{X})^2}}$

It means that **the Confidence Interval for predictions is wider than for the regression line**. Also:

- Both intervals have varying widths
- Least width at $X = \bar{X}$
- We are quite confident in the regression line, so the interval is very narrow
- If we knew β_0 and β_1, this interval would have zero width
- The prediction interval must incorporate the variabilibity of data around the prediciton line
- Even if we knew β_0 and β_1, this interval would still have width

Example: Diamonds data

Introduction

The library 'UsingR' includes a set of data called 'diamonds'. It gives the weight and price of 48 diamonds, in carats and Singapore dollars respectively.

Price as F(carat)

We can see how the price of diamonds relates to their weight by a linear model:

```
#calculate estimates
data(diamond)
fit <- lm(price ~ carat, data = diamond)
diamEst <- diamond %>% mutate (price = predict(fit)) # predictions of Y = f(X)
diamRes <- diamond %>% mutate (price = resid(fit)) # residuals of Y = f(X)
```

Let's calculate meaningful ITC and slope (the factor 10 gives the slope for 1/10th of a carat):

```
# ITC and slope
fit <- lm(price ~ I(carat*10 - mean(carat*10)), data = diamond)
round(coef(fit),2)
```

```
                    (Intercept) I(carat * 10 - mean(carat * 10))
                         500.08                            372.10
```

```
# calculate 95% CI for beta0 and beta1
sumCoef <- summary(fit)$coefficients
round(sumCoef[1,1] + c(-1, 1) * qt(.975, df = fit$df) * sumCoef[1, 2],2) # beta0
```

```
[1] 490.83 509.33
```

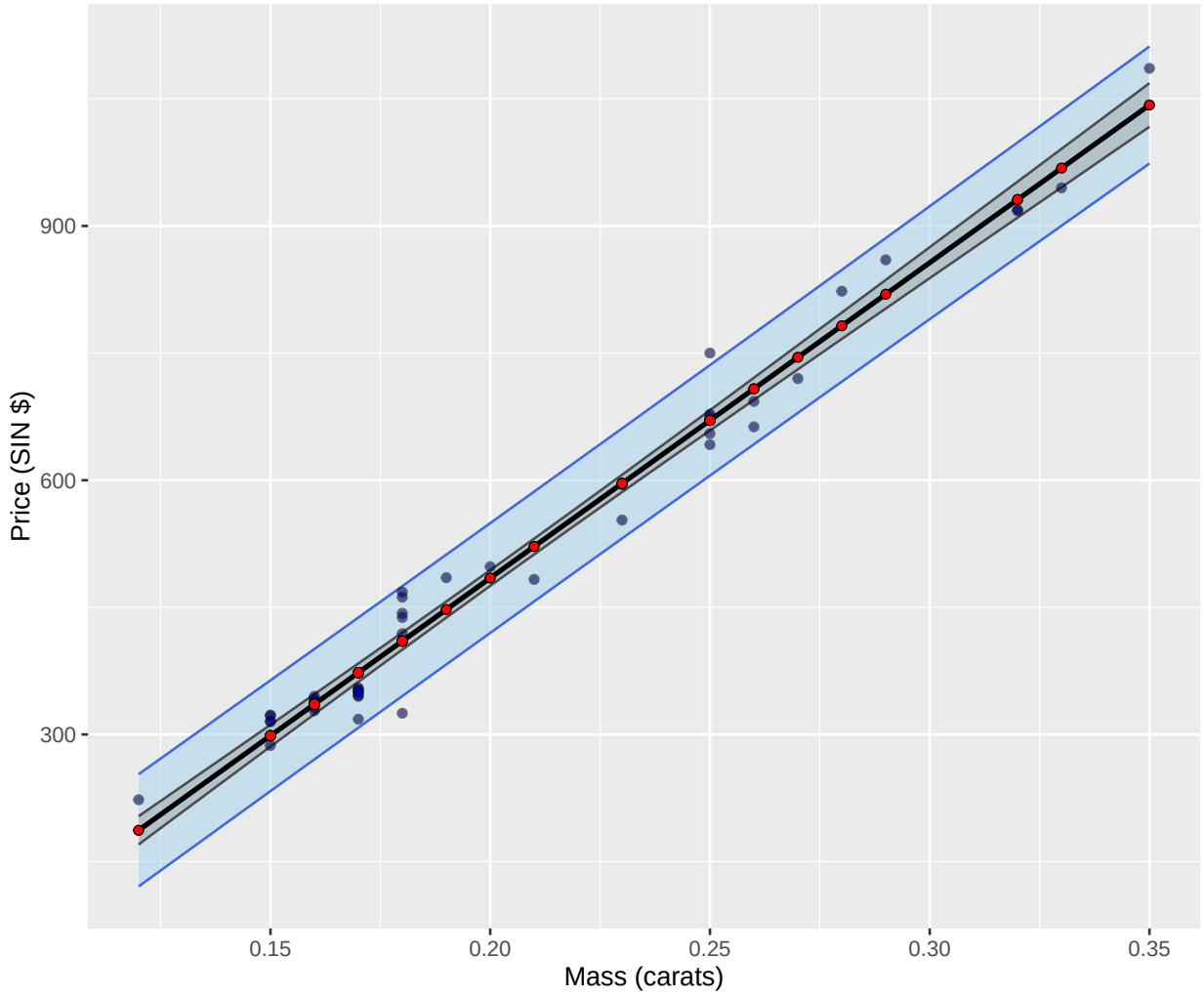
```
round(sumCoef[2,1] + c(-1, 1) * qt(.975, df = fit$df) * sumCoef[2, 2],2) # beta1
```

```
[1] 355.64 388.57
```

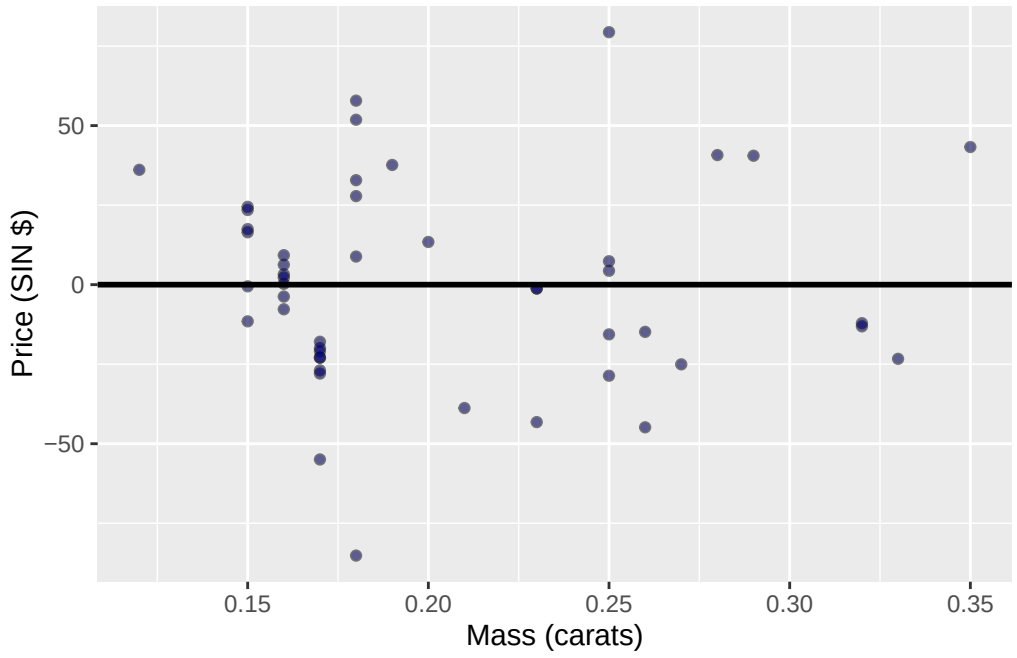
- **$500.08** is the expected price for the average sized diamond (0.20 carats)
- **$372.10** is the expected change in price for every 1/10th of a carat increase in mass
- With 95% confidence, we estimate that a 0.1 carat increase results in a **$355.60 to $388.60 increase** in price

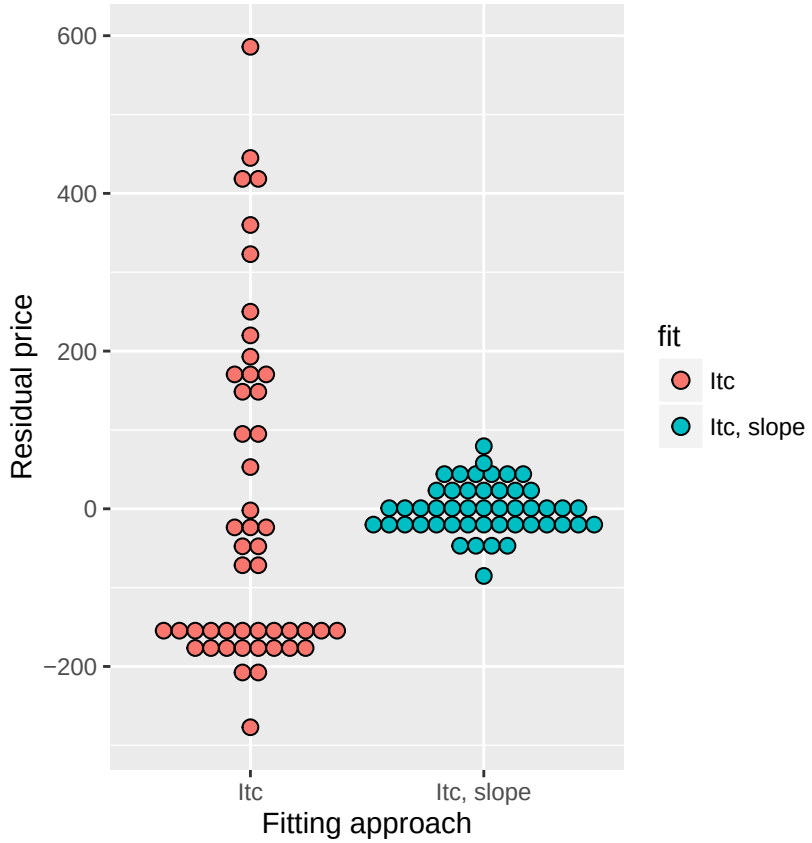
Linear Regression Model

```
#creating a vector of x values, from Xmin to Xmax
xVals <- seq(min(diamond$carat), max(diamond$carat), by = .01)
newdata <- data.frame(carat = xVals)
fit <- lm(price ~ carat, data = diamond)

#computing predictions for the regression line ("confidence")
p1 <- cbind(newdata, predict(fit, newdata, interval = ("confidence"))) # x, fit, lwr, uwr

#computing predictions for price estimations ("predictions")
p2 <- cbind(newdata, predict(fit, newdata, interval = ("prediction"))) # x, fit, lwr, uwr
```



Residuals


Price (SIN $)
50
0
−50
0.15
0.20
0.25
0.30
0.35
Mass (carats)


Residual price
600
400
200
0
−200
ltc
ltc, slope
Fitting approach
fit
ltc
ltc, slope

Multivariable Regression Analysis

Introduction

We can generalize Simple Linear Regression (SLR) to incorporate lots of regressors for the purpose of prediction.

What are the consequences of adding lots of regressors?

- Surely there must be consequences to throwing variables in that aren't related to Y?
- Surely there must be consequences to omitting variables that are?

Linear Model

When we perform a regression in one variable, we get two coefficients, a slope and an intercept. The intercept is really the coefficient of a special regressor which has the same value, 1, at every sample. The R function lm includes this regressor by default.

Note: `lm(var ~ 1, data)` *regresses child against the constant, 1. It returns mean(var).*

The general linear model extends simple linear regression (SLR) by adding terms linearly into the model. Using a vector representation (usually $X_{1i} = 1$, so that an intercept is included):

$$X_i = \begin{bmatrix} X_{1i} = 1 \\ \vdots \\ X_{pi} \end{bmatrix} \quad and \quad \beta = \begin{bmatrix} \beta_1 \\ \vdots \\ \beta_p \end{bmatrix} \quad then \quad E[Y_i] = \mu_i = X_i^T \beta$$

$$Y_i = \beta_1 X_{1i} + \beta_2 X_{2i} + \ldots + \beta_p X_{pi} + \epsilon_i = \sum_{k=1}^{p} X_{ik}\beta_j + \epsilon_i = X_i^T \beta + \epsilon_i \quad where\ \epsilon_i \sim N(0, \sigma^2)$$

Least squares (and hence ML estimates under iid Gaussianity of the errors) minimizes:

$$\sum_{i=1}^{n} \left(Y_i - X_i^T \beta\right)^2$$

Note, the important linearity is linearity in the coefficients.

$$Y_i = \beta_1 X_{1i}^2 + \beta_2 X_{2i}^2 + \ldots + \beta_p X_{pi}^2 + \epsilon_i$$

is still a linear model. (We've just squared the elements of the predictor variables.)

Estimates

Multivariate regression estimates are exactly those of a SLR through the origin, having removed the linear relationship of the other variables from both the regressor and response. In this sense, multivariate regression "adjusts" a coefficient for the linear impact of the other variables.

- Fitted responses: $\hat{Y}_i = X_i^T \hat{\beta}$
- Residuals: $e_i = Y_i - \hat{Y}_i$
- Variance estimate: $\hat{\sigma}^2 = \frac{1}{n-p} \sum_{i=1}^{n} e_i^2$
- Coefficients have standard errors, $\hat{\sigma}_{\hat{\beta}_k}$, and $\frac{\hat{\beta}_k - \beta_k}{\hat{\sigma}_{\hat{\beta}_k}}$ follows a T distribution with $n-p$ degrees of freedom
- Predicted responses have standard errors and we can calculate predicted and expected response intervals

Interpretation of the coefficient

The interpretation of a multivariate regression coefficient is the **expected change in the response per unit change in the regressor**, holding **all of the other regressors fixed**.

Dummy variables are smart

We can apply linear models to compare k+1 groups by using binary variables (Treated versus not in a clinical trial, for example).

$$Y_i = \beta_0 + X_{i,1}\beta_1 + \ldots + X_{i,k}\beta_k \epsilon_i$$

where each $X_{i,j}$ is binary + 1 if measurement X_i is in group j + 0 otherwise

It means that: + for measurements in group 0, $E[Y_i] = \beta_0$ + for all other measurememtns, $E[Y_i] = \beta_0 + \beta_j$

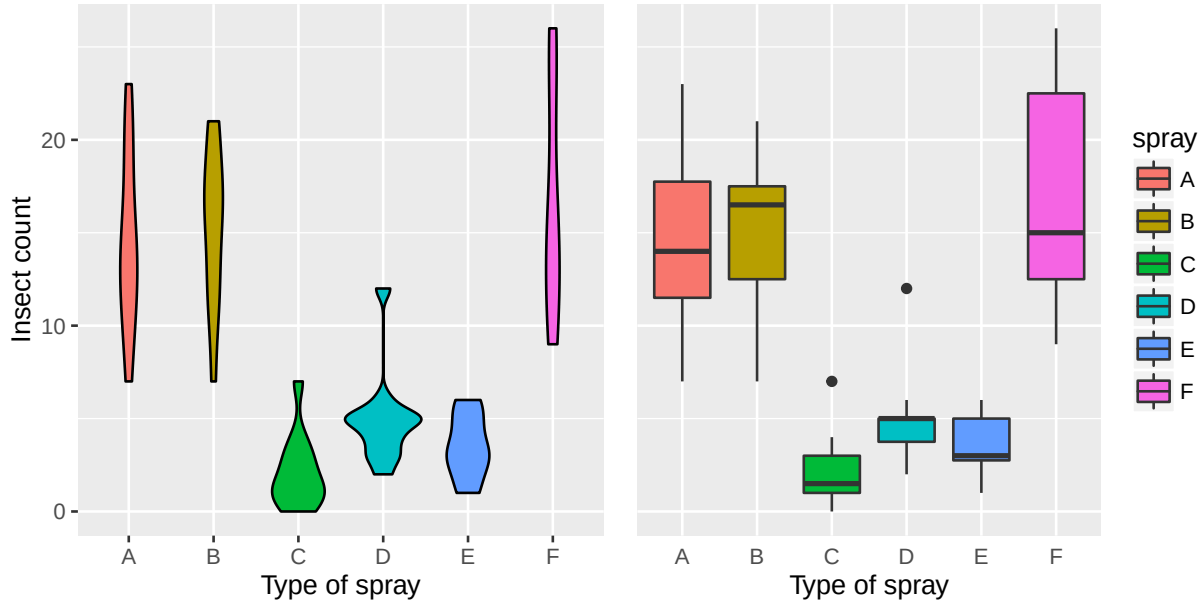
So β_j is interpretted as the increase or decrease in the mean comparing group j to group 0, called the **reference group**.

Note: using a model without an intercept will return the means of each group, without comparizon.

We can easily perform a linear model in R using `lm`.

- the reference group is the one alphabetically first one
- t-tests are for comparizons of means vs reference group
- t-test H_0: difference in mean is 0
- comparizon with another group can be done with `relevel`

Example: Insect Sprays data




```
# means of each group
summary(lm(count ~ spray - 1, data = InsectSprays))$coef
```

```
        Estimate Std. Error   t value     Pr(>|t|)
sprayA 14.500000   1.132156 12.807428 1.470512e-19
sprayB 15.333333   1.132156 13.543487 1.001994e-20
sprayC  2.083333   1.132156  1.840148 7.024334e-02
sprayD  4.916667   1.132156  4.342749 4.953047e-05
sprayE  3.500000   1.132156  3.091448 2.916794e-03
sprayF 16.666667   1.132156 14.721181 1.573471e-22
```

```
# variations in means compared to group C (dafault: vs group A)
spray2 <- relevel(InsectSprays$spray, "C")
summary(lm(count ~ spray2, data = InsectSprays))$coef
```

```
             Estimate Std. Error  t value     Pr(>|t|)
(Intercept)  2.083333   1.132156 1.840148 7.024334e-02
spray2A     12.416667   1.601110 7.755038 7.266893e-11
spray2B     13.250000   1.601110 8.275511 8.509776e-12
spray2D      2.833333   1.601110 1.769606 8.141205e-02
spray2E      1.416667   1.601110 0.884803 3.794750e-01
spray2F     14.583333   1.601110 9.108266 2.794343e-13
```

Notes:

- Counts are bounded from below by 0, violates the assumption of normality of the errors
- Variance does not appear to be constant
- Poisson GLM are more adapted for fitting count data

Example: Swiss data

We want to compare the relationship between Agriculture and Fertility in Catholic vs Protestant provinces. We can do it by using binary variables with an interaction term:

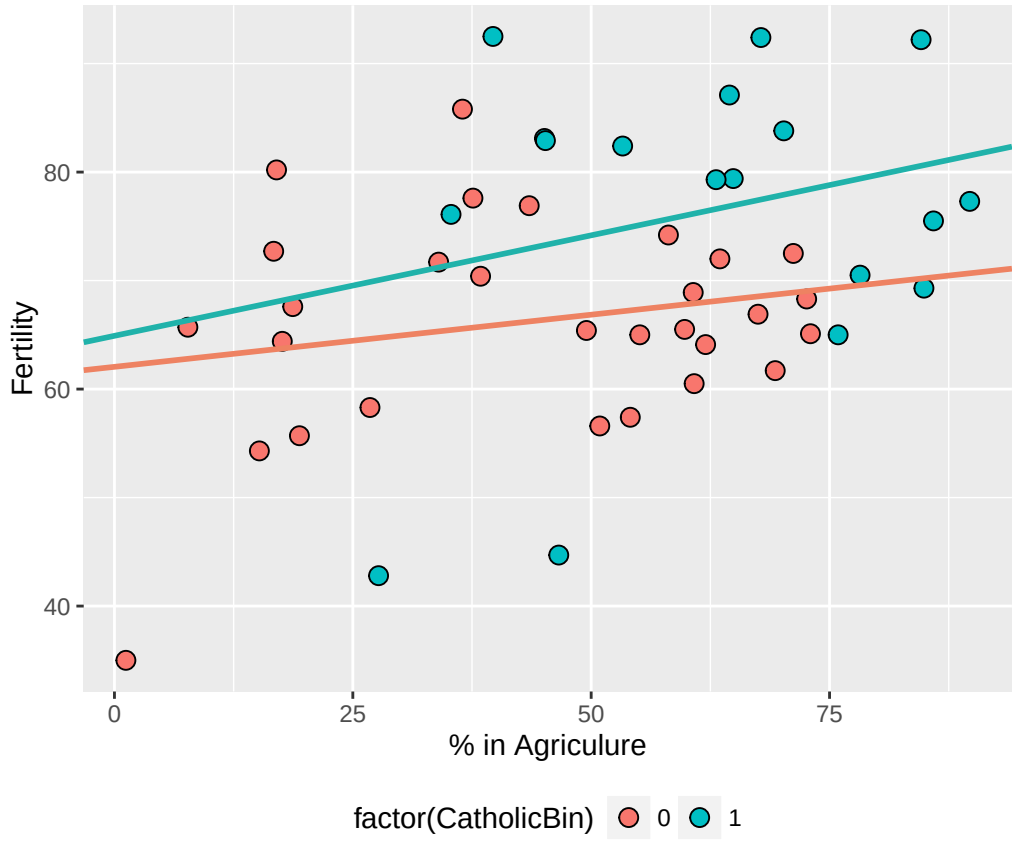
$$E[Y|X_1, X_2] = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_1 X_2$$

where:

- X_1 is the Agriculture percentage
- X_2 is the Catholic binary variable

This model gives us two ITC and two slopes.

```
data(swiss)

# Provinces are either majority Catholic or majority Protestant
# hist(swiss$Catholic)
# So we can create a group to identify them clearly
swiss <- mutate (swiss, CatholicBin = 1 * (Catholic > 50))

# Then we apply our model with an interaction term
fit <- lm (Fertility ~ Agriculture * factor (CatholicBin), data=swiss)
```



Outliers, Influence and Leverage

Definition

An **outlier** is a data point **far from the mean of X and/or Y**.

An outlier is said to be **influential** when it **significantly affect the fit**, ie. when not conforming to the regresison relationship of the other points.

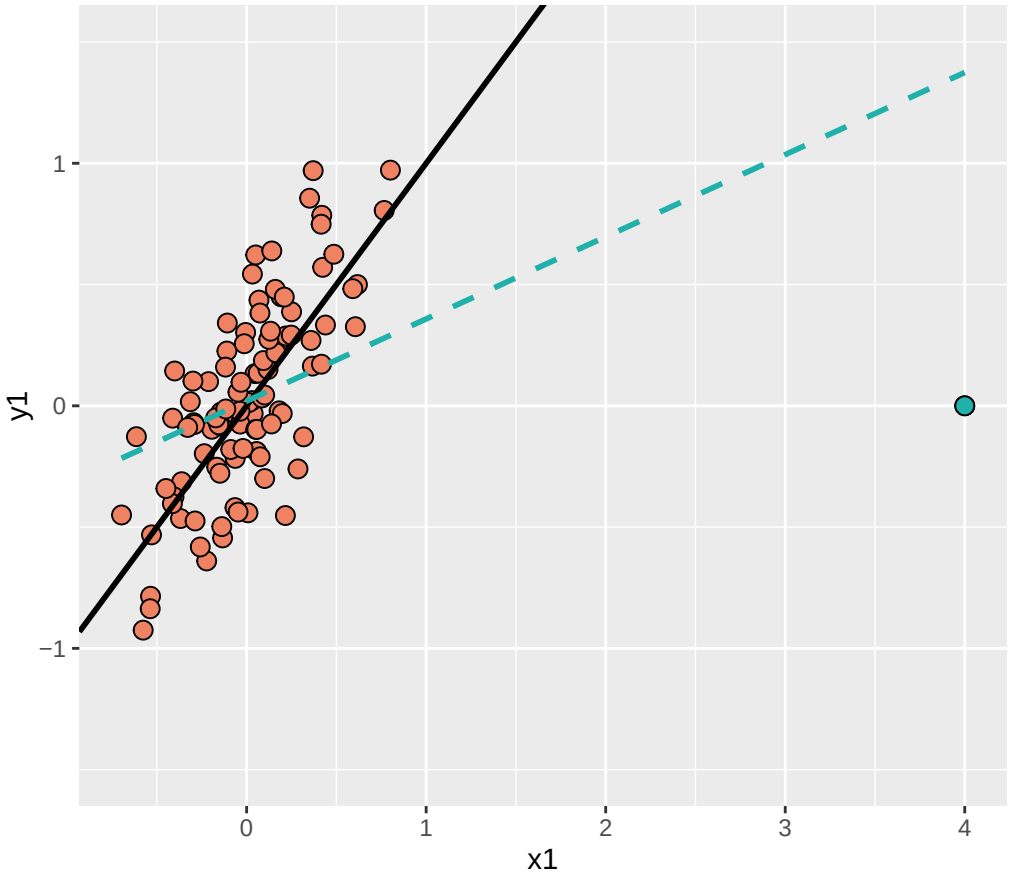
An outlier is said to have **leverage** when far from **mean of X**.

Outliers may or may not belong in the data. They may represent real events or they may be spurious. In any case, they should be examined. The basic technique is to examine the effects of leaving one sample out.

Example: single outlier

Let's look at a random set of points where a clear outlier c(4, 0) has created an artificial but strong regression relationship where there shouldn't be one.

```
set.seed(1000)
n <- 100
x0 <- rnorm(n, sd = .3); y0 <- x0 + rnorm(n, sd = .3)
x1 <- c(4, x0); y1 <- c(0, y0)
fit <- lm (y1 ~ x1)
```

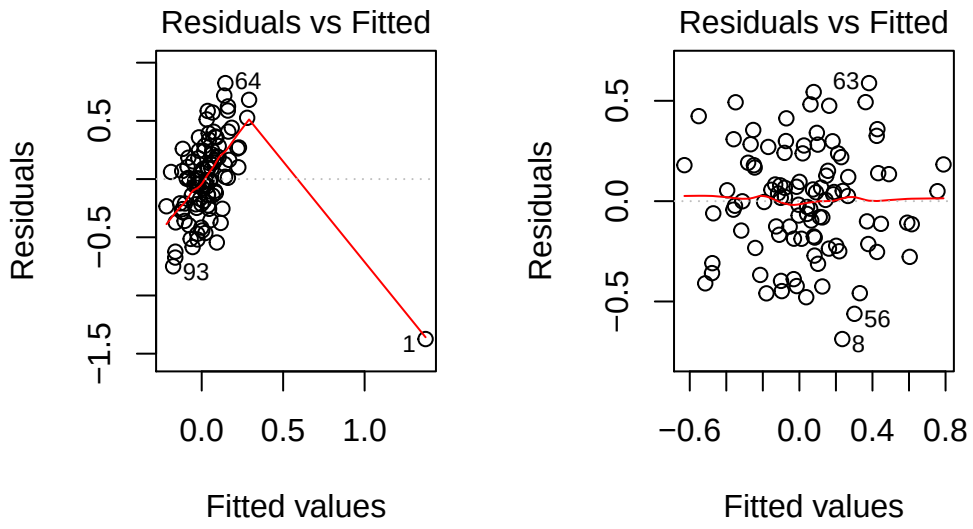


Residuals versus fitted values

The simplest diagnostic plot displays residuals versus fitted values. Residuals should be:

- uncorrelated with the fit
- independent
- (almost) identically disributed with mean zero

```
fit <- lm (y1 ~ x1)
y2 <- y1[-1]; x2 <- x1[-1]
fitno <- lm (y2 ~ x2)
```



On the left, there is a linear pattern involving all but one residual and the fit. The Residuals vs Fitted plot labels certain points with their row names or numbers, numbers in our case. The influential outlier is row N°1.

Without the outlier, on the right, the plot has none of the patterned appearance: residuals are independently and (almost) identically distributed with zero mean, and are uncorrelated with the fit.

The change in coefficients induced by including/excluding a sample is a simple measure of its influence:

```
mcoef <- rbind(c(0,1), coef(fitno), coef(fit)-coef(fitno), coef(fit))
rownames (mcoef) <- c("fitnoExact", "fitno", "coefVar", "fit")
print(mcoef)
```

```
           (Intercept)         x2
fitnoExact  0.00000000  1.0000000
fitno       0.03038857  0.9468564
coefVar    -0.01075490 -0.6081808
fit         0.01963367  0.3386756
```

dbeta

The function **dfbeta** does the equivalent calculation for every sample in the data.

- The first row of dfbeta(fit) matches the difference we've just calculated.
- The first sample has a much larger effect (x100) on the slope than the others.
- Its effect on the intercept is not very distinctive essentially because its y coordinate is 0, the mean of the other samples.

```
head(dfbeta(fit), 5)
```

```
   (Intercept)            x1
1 -0.010754902 -6.081808e-01
2 -0.005363172  3.766103e-03
3 -0.002269186  3.492842e-03
4 -0.000457082  5.939344e-05
5  0.003557280  2.190419e-03
```

Influence measure: hatvalues

When a sample is **included** in a model, it **pulls the regression line closer to itself**. So for influential samples, their residual (actual y value minus regression line value) will be much smaller when it is included.

It means that **1 minus the ratio of the two residuals**, included vs excluded, measures the **sample's influence**: near 0 for points which are not influential, and near 1 for points which are.

This measure is sometimes called influence, sometimes leverage, and sometimes **hat value**.

```
res <- resid(fit)[1] # included
resno <- y1[1] - predict(fitno, newdata=data.frame(x2=x1[1])) # not included
cbind(1 - res / resno,
      hatvalues(fit)[1]) # hat value
```

```
       [,1]      [,2]
1 0.6400202 0.6400202
```

Variance impact - Standardized and Studentized residuals

Residuals of individual samples are sometimes treated as having the same variance, which is estimated as the variance of the entire set of residuals.

Theoretically, however, residuals of individual samples have different variances and these differences can become large in the presence of outliers.

Standardized and Studentized residuals attempt to compensate for this effect in two slightly different ways. Both use hat values.

Standardized residuals

We assume residuals are **Gaussian iid** of mean 0 and sd σ:

```
# residuals standard deviation sigma
sigma <- sqrt(sum(resid(fit)^2)/fit$df.residual)
cbind(sigma,
      summary(fit)$sigma)
```

```
         sigma
[1,] 0.3552361 0.3552361
```

Ordinarily we would just divide fit's residual (which has mean 0) by sigma to get a standard normal distribution.

But to account for the different variances, we will use the following formula:

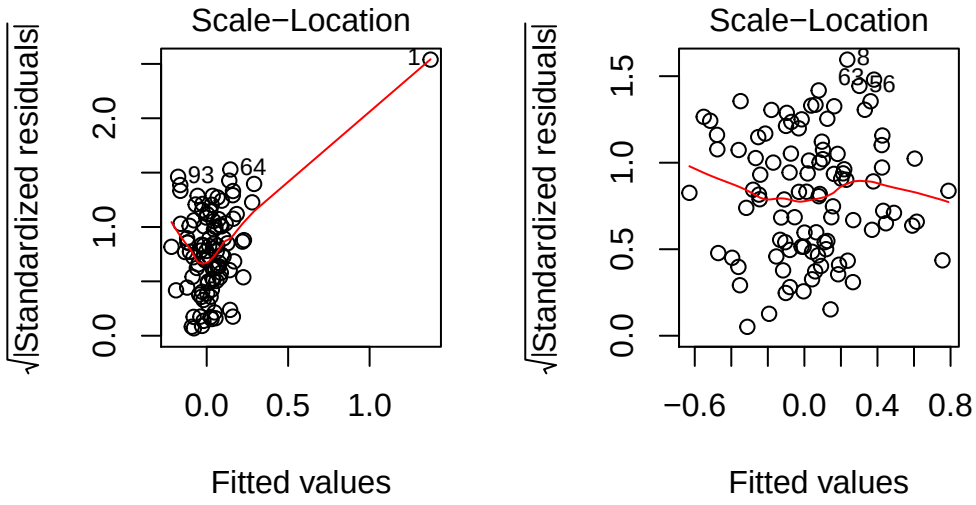
$$\epsilon_i/(\sigma * \sqrt{1 - \hat{fit}_i)) \;\; where \; \hat{fit}_i \; is \; the \; hat \; value \; of \; sample \; i$$

The result is called the standardized residuals, ie residuals with the same variance 1.

```
# standardized residuals
rstd <- resid(fit) / (sigma*sqrt(1-hatvalues(fit)))
cbind(head(rstd),
      head(rstandard(fit)))
```

```
        [,1]       [,2]
1 -6.4481738 -6.4481738
2 -1.4689489 -1.4689489
3 -0.5960158 -0.5960158
4 -0.1285617 -0.1285617
5  1.0334794  1.0334794
6 -0.3884428 -0.3884428
```

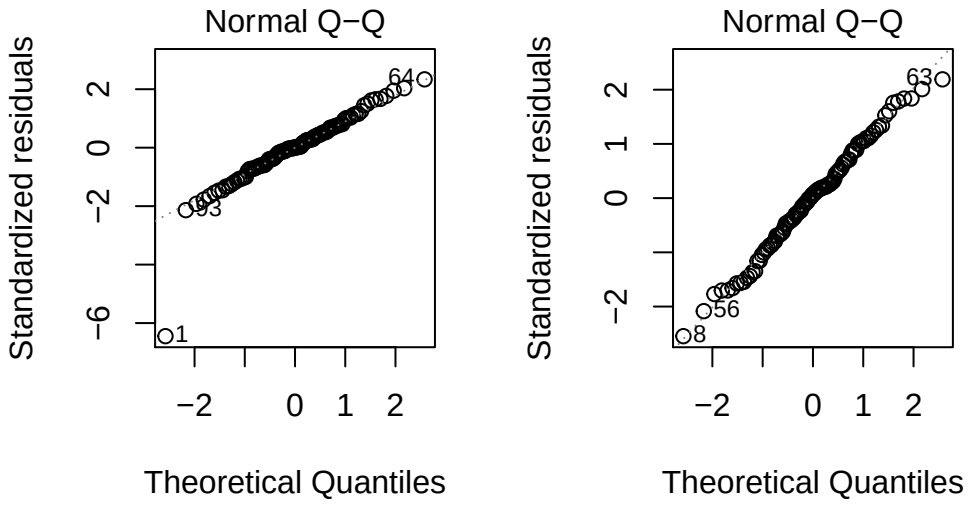
A Scale-Location plot shows the square root of standardized residuals against fitted values.



QQPlot

Most of the diagnostic statistics under discussion were developed because of perceived shortcomings of other diagnostics and because their distributions under a null hypothesis could be characterized.

The assumption that residuals are approximately normal is implicit in such characterizations. Since standardized residuals adjust for individual residual variances, a QQ plot of standardized residuals against normal with constant variance is of interest. On the left, the outlier is about -7 standard deviations from the mean.



Studentized residuals

Studentized residuals, (sometimes called externally Studentized residuals,) estimate the standard deviations of individual residuals using, in addition to individual hat values, the deviance of a model which leaves the associated sample out.

We'll illustrate using the outlier. Recalling that the model we called fitno omits the outlier sample, calculate the sample standard deviation of fitno's residual by dividing its deviance, by its residual degrees of freedom and taking the square root.

Store the result in a variable called sigma1.

```
# residuals standard deviation sigma
sigma1 <- sqrt(sum(resid(fitno)^2)/fitno$df.residual)
cbind(sigma1,
      summary(fitno)$sigma)
```

```
        sigma1
[1,] 0.2719191 0.2719191
```

```
# studentized residuals
rstd1 <- resid(fit) / (sigma1*sqrt(1-hatvalues(fit)))
cbind(head(rstd1),
      head(rstudent(fit)))
```

```
        [,1]       [,2]
1 -8.4239192 -8.4239192
2 -1.9190406 -1.4777040
3 -0.7786373 -0.5940648
4 -0.1679536 -0.1279215
5  1.3501415  1.0338385
6 -0.5074632 -0.3867708
```

Cook's distance

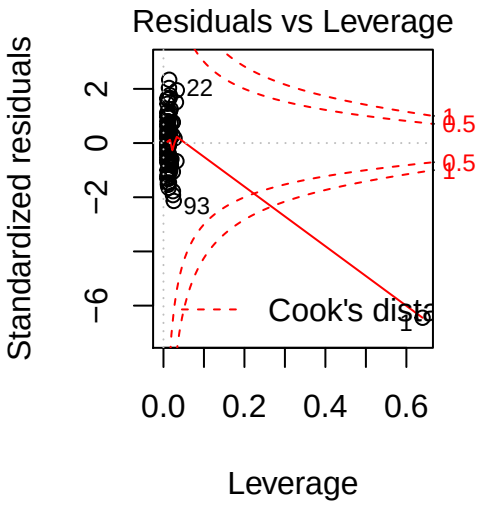
Cook's distance is essentially the sum of squared differences between values fitted with and without a particular sample.

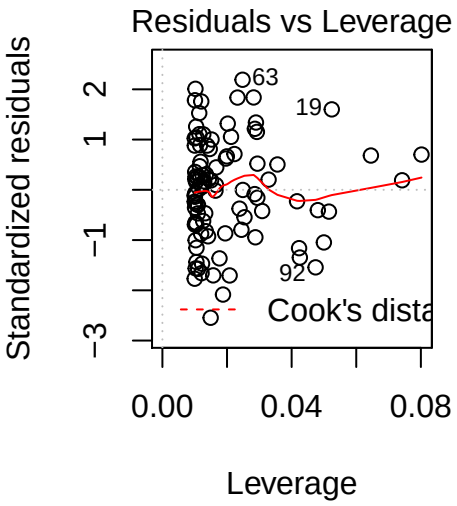
It is normalized (divided by) residual sample variance times the number of predictors which is 2 in our case (the intercept and x.)

It essentially tells how much a given sample changes a model.

```
dy <- predict(fit) -
      predict(fitno, newdata=data.frame(x2=x1))

cbind(sum(dy^2)/(2*sigma^2), # Cook's distance
      cooks.distance(fit)[1])
```

```
     [,1]    [,2]
1 36.9623 36.9623
```


Residuals vs Leverage
Standardized residuals
Leverage
Cook's dista
22
93
16
0.5
0.5
1
1
0.0
0.2
0.4
0.6
2
0
−2
−6


Residuals vs Leverage
Standardized residuals
Leverage
Cook's dista
63
19
92
0.00
0.04
0.08
2
1
−1
−3

Annex: Galton's Data

How can we:

- use parents' heights to predict childrens' heights
- find a parsimonious, easily described mean relationship between parent and children's heights
- investigate the variation in childrens' heights that appears unrelated to parents' heights (residual variation)
- quantify what impact genotype information has beyond parental height in explaining child height
- figure out how/whether and what assumptions are needed to generalize findings beyond the data in question
- explain why do children of very tall parents tend to be tall, but a little shorter than their parents
- explain why children of very short parents tend to be short, but a little taller than their parents

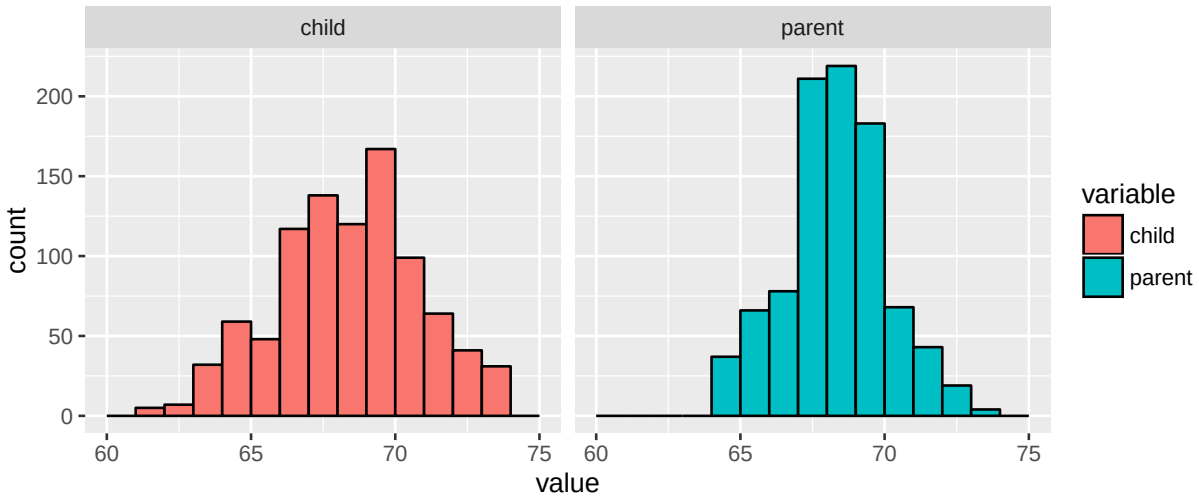
Note: the last two points are called 'Regression to the mean'.

Parents vs Children Height

Let's look at the marginal (parents disregarding children and children disregarding parents) distributions first.

- Parent distribution is all heterosexual couples.
- Correction for gender via multiplying female heights by 1.08.
- Overplotting is an issue from discretization.

```
data(galton); 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
```



Comparing childrens' heights and their parents' heights

We can easily calculate our linear model fit in R:

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