Regression Models (Part VII: Regression to the Mean)

A. Statistical linear regression models

Up to this point, we've only considered estimation. Estimation is useful, but we also need to know how to extend our estimates to a population. This is the process of statistical inference. Our approach to statistical inference will be through a statistical model. At the bare minimum, we need a few distributional assumptions on the errors. However, we'll focus on full model assumptions under Gaussianity.

B. Basic regression model with additive Gaussian errors.

Consider developing a probabilistic model for linear regression. Our starting point will assume a systematic component via a line and then independent and identically distributed Gaussian errors. We can write the model out as:

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

Here, the ϵ_i are assumed to be independent and identically distributed as $N(0, \sigma^2)$. Under this model,

$$E[Y_i \mid X_i = x_i] = \mu_i = \beta_0 + \beta_1 x_i$$

and

$$Var(Y_i \mid X_i = x_i) = \sigma^2.$$

This model implies that the Y_i are independent and normally distributed with means $\beta_0 + \beta_1 x_i$ and variance σ^2. We could write this more compactly as

$$Y_i \mid X_i = x_i \sim N(\beta_0 + \beta_1 x_i, \sigma^2).$$

While this specification of the model is a perhaps better for advanced purposes, specifying the model as linear with additive error terms is generally more useful. With that specification, we can hypothesize and discuss the nature of the errors. In fact, we'll even cover ways to estimate them to investigate our model assumption.

Remember that our least squares estimates of β_0 and β_1 are:

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

It is convenient that under our Gaussian additive error model that the maximum likelihood estimates of β_0 and β_1 are the least squares estimates.

C. Interpreting regression coefficients, the intercept

Our model allows us to attach statistical interpretations to our parameters. Let's start with the intercept; β_0 represents the expected value of the response when the predictor is 0. We can show this as:

$$E[Y|X = 0] = \beta_0 + \beta_1 \times 0 = \beta_0.$$

Note, the intercept isn't always of interest. For example, when $X = 0$ is impossible or far outside of the range of data. Take as a specific instance, when X is blood pressure, no one is interested in studying blood pressure's impact on anything for values near 0.

There is a way to make your intercept more interpretable. Consider that:

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

Therefore, shifting your X values by value a changes the intercept, but not the slope. Often a is set to $\bar{X}$, so that the intercept is interpreted as the expected response at the average X value.

D. Interpreting regression coefficients, the slope

Now that we understand how to interpret the intercept, let's try interpreting the slope. Our slope, β_1, is the expected change in response for a 1 unit change in the predictor. We can show that as follows:

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

Notice that the interpretation of β_1 is tied to the units of the X variable. Let's consider the impact of changing the units.

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

Therefore, multiplication of X by a factor a results in dividing the coefficient by a factor of a.

As an example, suppose that X is height in meters (m) and Y is weight in kilograms (kg). Then β_1 is kg/m. Converting X to centimeters implies multiplying X by 100 cm/m. To get β_1 in the right units if we had fit the model in meters, we have to divide by 100 cm/m. Or, we can write out the notation as:

$$Xm \times \frac{100cm}{m} = (100X)cm \;\text{ and }\; \beta_1 \frac{kg}{m} \times \frac{1m}{100cm} = \left(\frac{\beta_1}{100}\right)\frac{kg}{cm}$$

E. Using regression for prediction

Regression is quite useful for prediction. If we would like to guess the outcome at a particular value of the predictor, say X, the regression model guesses:

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

In other words, just find the Y value on the line with the corresponding X value. Regression, especially linear regression, often doesn't produce the best prediction algorithms. However, it produces parsimonious and interpretable models along with the predictions. Also, as we'll see later we'll be able to get easily described estimates of uncertainty associated with our predictions.

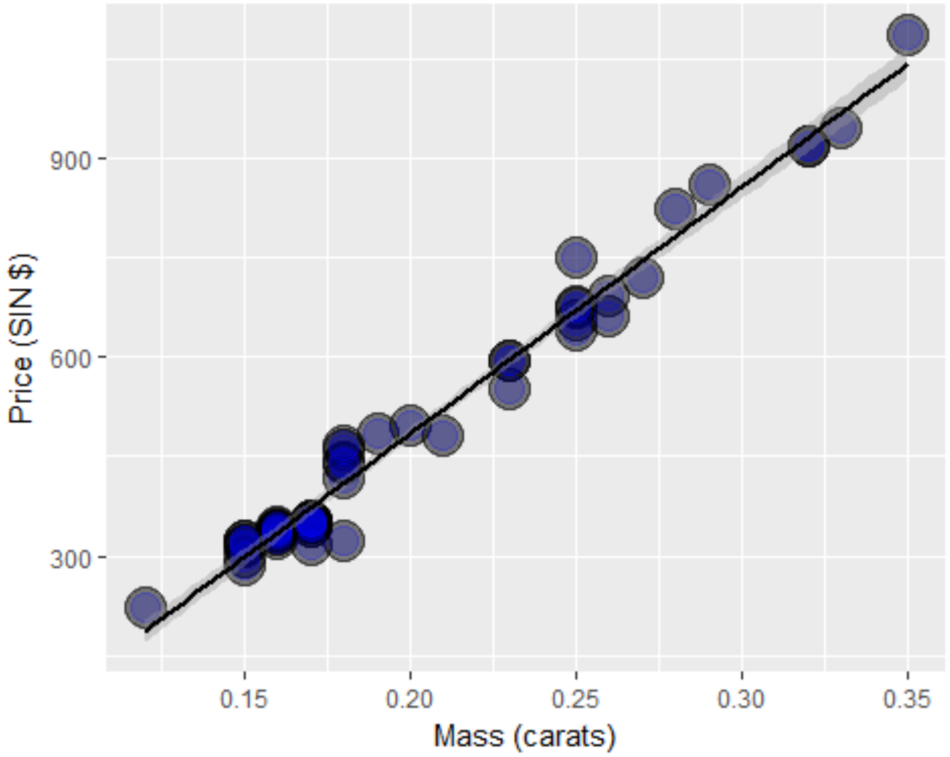
F. Example

Let's analyze the `diamond` data set from the `UsingR` package. The data is diamond prices (in Singapore dollars) and diamond weight in carats. Carats are a standard measure of diamond mass, 0.2 grams. To get the data use `library(UsingR); data(diamond)`

First let's plot the data. Here is the plot:

```r
library(UsingR)
data(diamond)
library(ggplot2)
g = ggplot(diamond, aes(x = carat, y = price))
g = g + xlab("Mass (carats)")
g = g + ylab("Price (SIN $)")
g = g + geom_point(size = 7, colour = "black", alpha=0.5)
g = g + geom_point(size = 5, colour = "blue", alpha=0.2)
```

```
g = g + geom_smooth(method = "lm", colour = "black")
g
```



First, let's fit the linear regression model. This is done with the `lm` function in R (`lm` stands for linear model). The syntax is `lm(Y ~ X)` where `Y`is the response and `X` is the predictor.

```
fit <- lm(price ~ carat, data = diamond)
coef(fit)
## (Intercept)       carat
##   -259.6259   3721.0249
```

The function `coef` grabs the fitted coefficients and conveniently names them for you. Therefore, we estimate an expected 3721.02 (SIN) dollar increase in price for every carat increase in mass of diamond. The intercept -259.63 is the expected price of a 0 carat diamond.

We're not interested in 0 carat diamonds (it's hard to get a good price for them ;-). Let's fit the model with a more interpretable intercept by centering our *X* variable.

```
fit2 <- lm(price ~ I(carat - mean(carat)), data = diamond)
coef(fit2)
##            (Intercept) I(carat - mean(carat))
##               500.0833              3721.0249
```

Thus the new intercept, 500.1, is the expected price for the average sized diamond of the data (0.2042 carats). Notice the estimated slope didn't change at all.

Now let's try changing the scale. This is useful since a one carat increase in a diamond is pretty big. What about changing units to 1/10th of a carat? We can just do this by just dividing the coefficient by 10, no need to refit the model.

Thus, we expect a 372.102 (SIN) dollar change in price for every 1/10th of a carat increase in mass of diamond.

Let's show via R that this is the same as rescaling our *X* variable and refitting. To go from 1 carat to 1/10 of a carat units, we need to multiply our data by 10.

```
fit3 <- lm(price ~ I(carat * 10), data = diamond)
coef(fit3)
##   (Intercept) I(carat * 10)
##     -259.6259      372.1025
```

Now, let's predicting the price of a diamond. This should be as easy as just evaluating the fitted line at the price we want to

```
newx <- c(0.16, 0.27, 0.34)
coef(fit)[1] + coef(fit)[2] * newx
## [1]  335.7381  745.0508 1005.5225
```

Therefore, we predict the price to be 335.7, 745.1 and 1005.5 for a 0.16, 0.26 and 0.34 carat diamonds. Of course, our prediction models will get more elaborate and R has a generic function, predict, to put our *X* values into the model for us. This is generally preferable and less The data has to go into the model as a data frame with the same named *X* variables.

```r
predict(fit, newdata = data.frame(carat = newx))
##         1         2         3
##  335.7381  745.0508 1005.5225
```