Simulated Linear Regression

Now we'll be performing simulated linear regression on a simulated model to identify how good least squares is.

Let's use the model

$$Y = 5 - 2x + \epsilon$$

So, in our model,

$$\beta_1 = 5, \beta_2 = -2$$

and to produce a stochiastic dependant variable, we'll distribute

$$\epsilon \sim N(\mu = 0, \sigma^2 = 9)$$

. Aha! Let's have a look why the assumptions in the CLRM are important, by introducing some correlation and stuff and seeing how this changes our model!

```
num_obs = 21
beta_0 = 5
beta_1 = -2
sigma = 3
```

Now, let's generate some simulated values of epsilon.

```
set.seed(1)
epsilon = rnorm(n = num_obs, mean = 0, sd = sigma)
```

Using our CLRM model, the assumption is

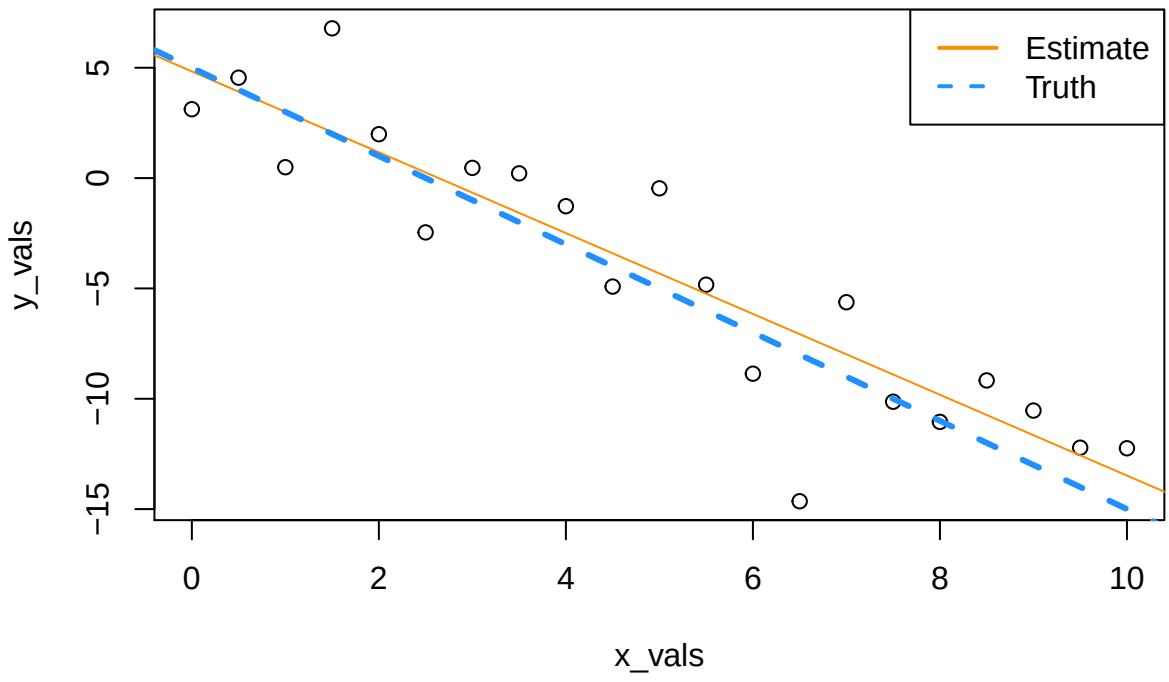
$$E(Y_i|X_i) = \beta_0 + \beta_1 * X_i$$

, i.e the expected values are fixed, so we can generate the values.

```
x_vals = seq(from = 0, to = 10, length.out = num_obs)
y_vals = beta_0 + beta_1 * x_vals + epsilon
```

Now, let's use SLRM to calculate the values.

```
S_xx = sum((x_vals - mean(x_vals))^2)
S_xy = sum((x_vals - mean(x_vals))*(y_vals - mean(y_vals)))
S_yy = sum((y_vals - mean(y_vals))^2)
beta_1_hat = sum((x_vals - mean(x_vals))*(y_vals - mean(y_vals)))/ sum((x_vals - mean(x_vals))^2)
beta_0_hat = mean(y_vals) - beta_1_hat * mean(x_vals)
plot(y_vals ~ x_vals)
abline(beta_0_hat, beta_1_hat, col="darkorange")
abline(beta_0, beta_1, lwd = 3, lty = 2, col="dodgerblue")
legend("topright", c("Estimate", "Truth"), lty = c(1,2), lwd=2, col=c("darkorange", "dodgerblue"))
```



Just to reduce repetition, lets' create a function to do CLRM quickly.

```
least_squares <- function(x_vals, y_vals) {
#  S_xx = sum((x_vals - mean(x_vals))^2)
#  S_xy = sum((x_vals - mean(x_vals))*(y_vals - mean(y_vals)))
#  S_yy = sum((y_vals - mean(y_vals))^2)
  beta_1_hat = sum((x_vals - mean(x_vals))*(y_vals - mean(y_vals)))/ sum((x_vals - mean(x_vals))^2)
  beta_0_hat = mean(y_vals) - beta_1_hat * mean(x_vals)
  return(c(beta_0_hat, beta_1_hat))
}
```

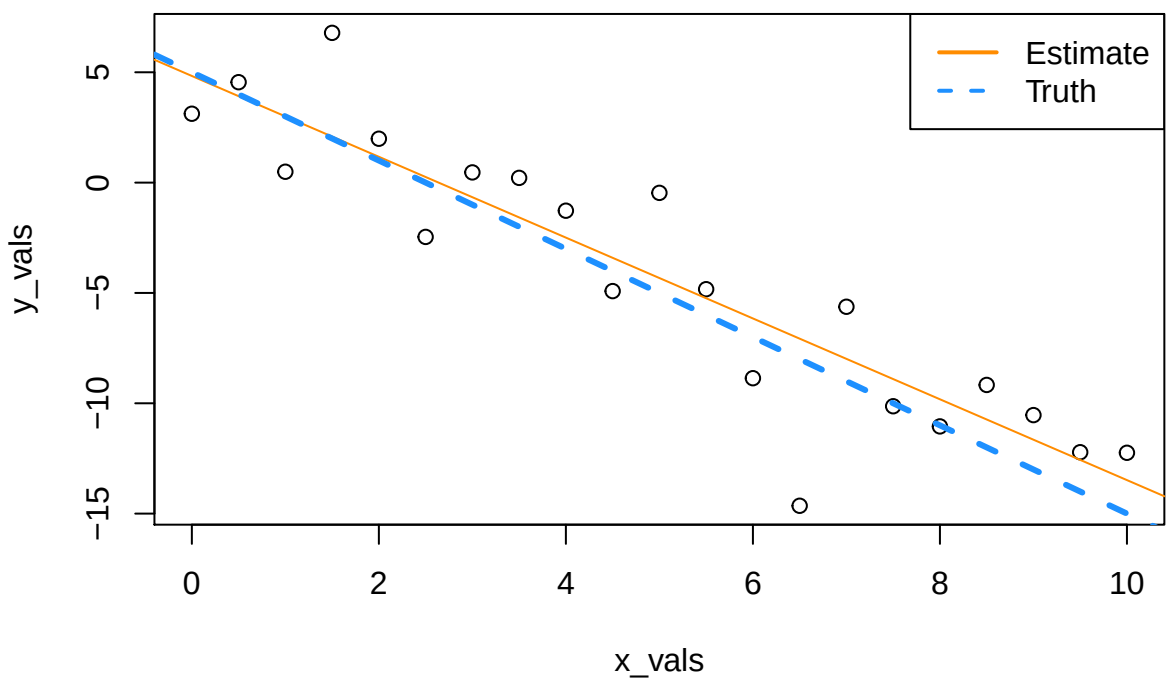
And, lets also make a function to plot the data as well:

```
plot_comparison <- function(x_vals, y_vals, beta_0, beta_1, beta_0_hat, beta_1_hat) {
  plot(y_vals ~ x_vals)
  abline(beta_0_hat, beta_1_hat, col="darkorange")
  abline(beta_0, beta_1, lwd = 3, lty = 2, col="dodgerblue")
  legend("topright", c("Estimate", "Truth"), lty = c(1,2), lwd=2, col=c("darkorange", "dodgerblue"))
}
```

So, our prior work can be reproduced as follows:

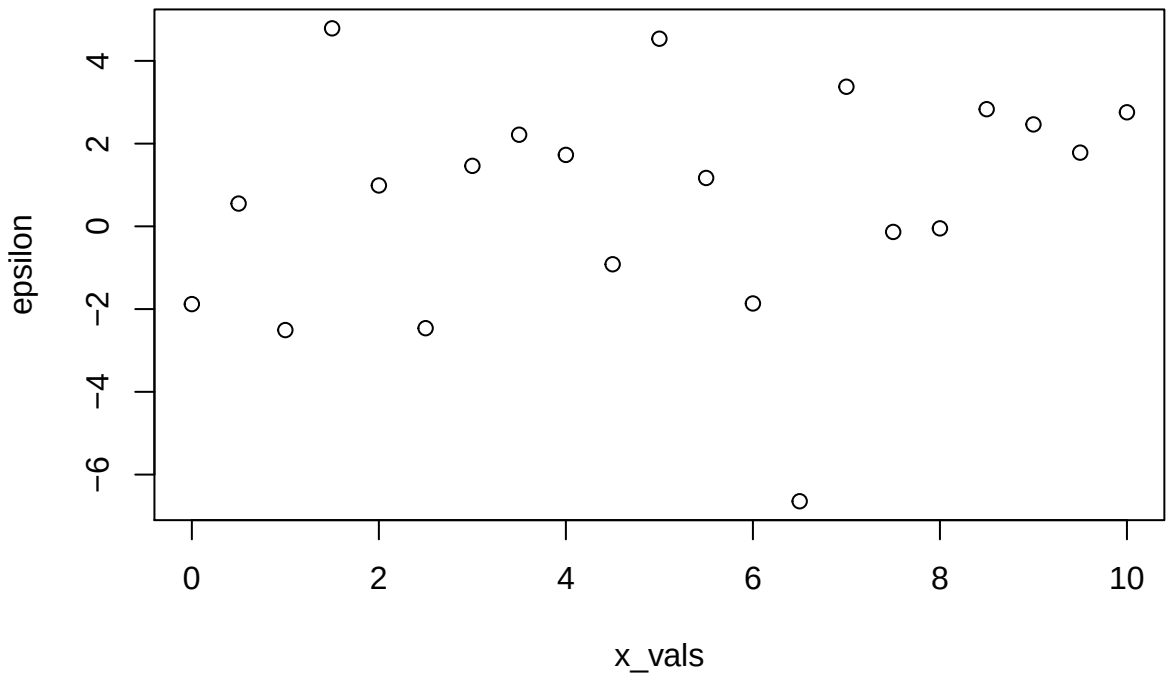
```
complete_method <- function(x_vals, y_vals, beta_0, beta_1) {
  result = least_squares(x_vals, y_vals)
  beta_0_hat = result[1]
  beta_1_hat = result[2]
  plot_comparison(x_vals, y_vals, beta_0, beta_1, beta_0_hat, beta_1_hat)
}
```

```
complete_method(x_vals, y_vals, beta_0, beta_1)
```



Alright! Sweet, it looks like we've go some fancy stuff. Let's try screwing it up! Let's try correlating the epsilon with the variance. First, let's plot x_vals against epsilon.

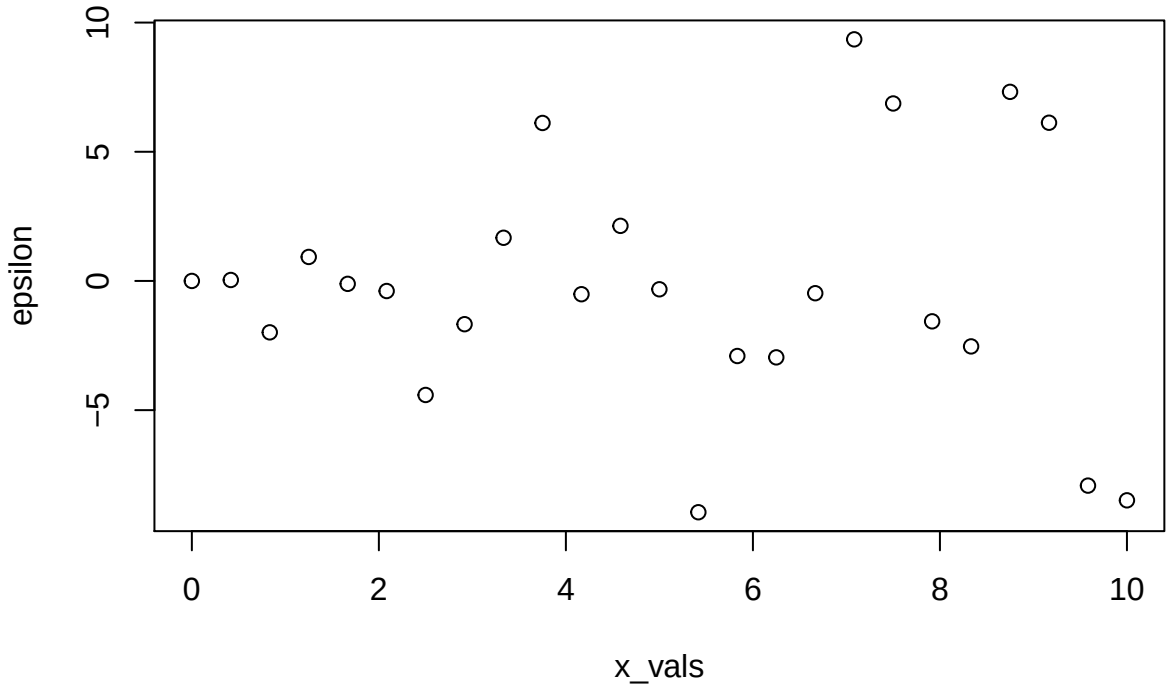
```
plot(epsilon ~ x_vals)
```



Looks like theres no real corellation. Let's try changing that

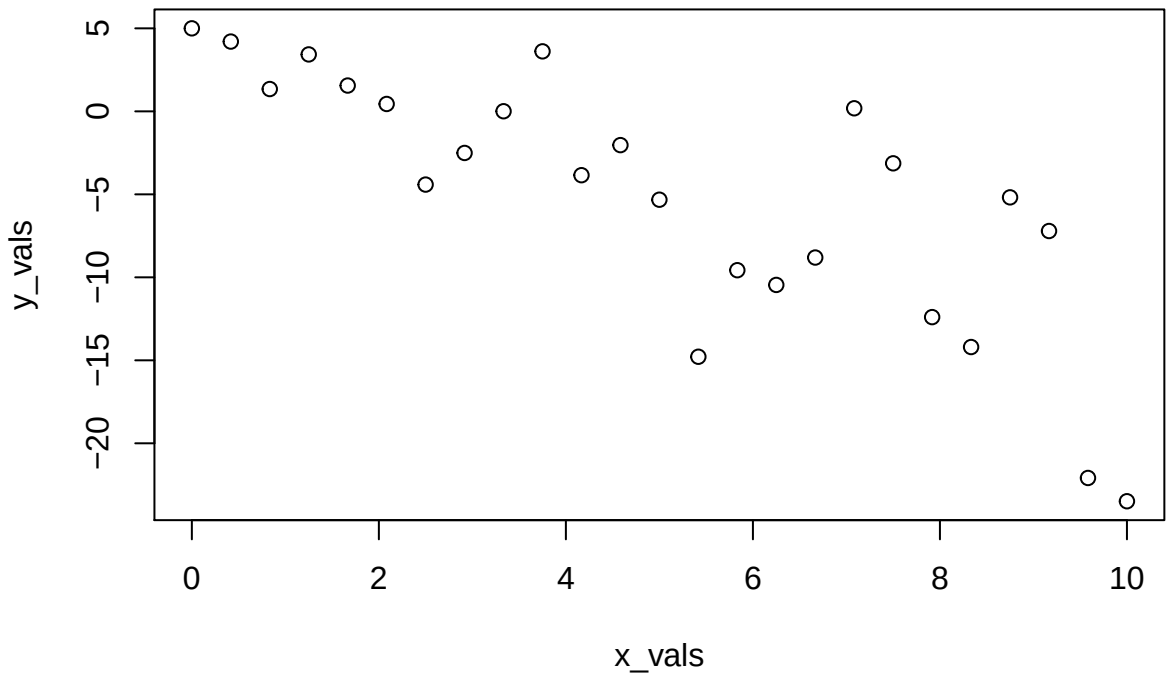
```
num_obs = 25
x_vals = seq(from = 0, to = 10, length.out = num_obs)
epsilon = rnorm(n = num_obs, mean = 0, sd = sigma)
epsilon = (epsilon * x_vals * 2)/mean(x_vals)
y_vals = beta_0 + beta_1 * x_vals + epsilon

plot(epsilon ~ x_vals)
```



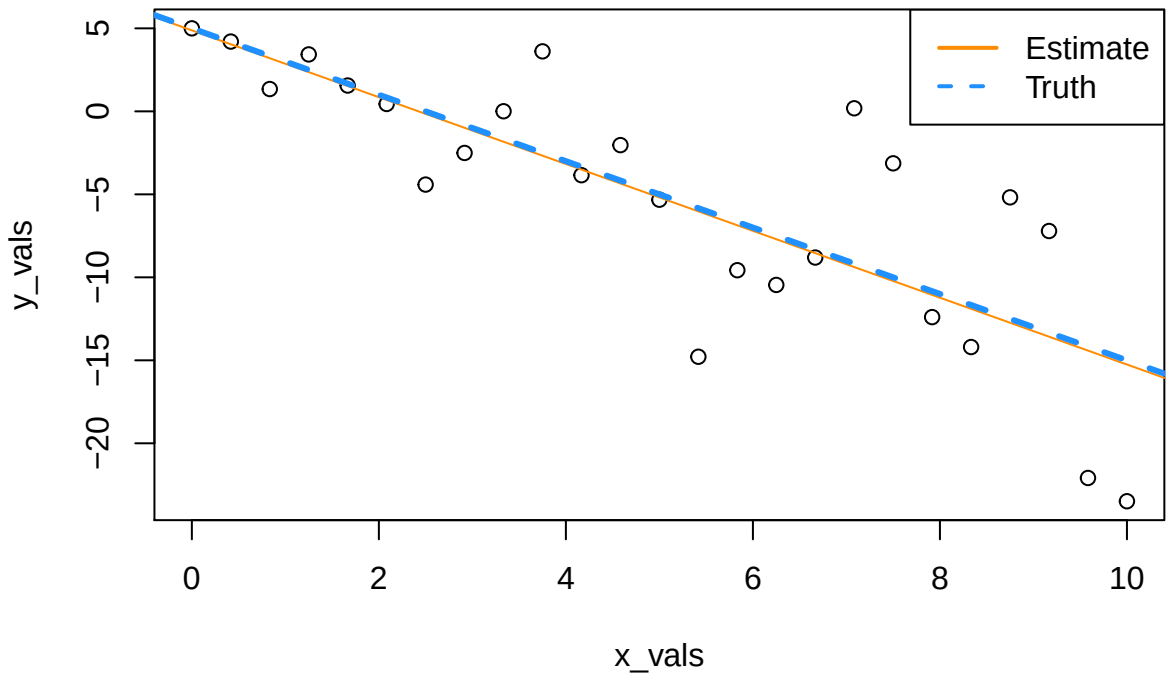
Oh oh. That looks gnarly dude! Let's see if CLRM works well in this condition? Wait, what does the plot look like?

```
plot(y_vals ~ x_vals)
```



Alright CLRM - do your best!

```
complete_method(x_vals, y_vals, beta_0, beta_1)
```



Can't say it looks much worse, but I guess the guys who developed the technique must have used the assumptions of heteroscedascticity somewhere in their derivation.

Note: Ah! Note, the estimators for

$$\beta_1$$

and

$$\beta_0$$

are normally distrubuted as they are generated from a linear combination of

$$y_i$$

and as

$$y_i$$

is normally distributed- AHHHHH! Just realised again!!!!! Because we assumes x is non-stochastic - consider the formula for

$$S_{xy}/S_{xx}$$

:

$$\frac{S_{xy}}{S_{xx}} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})y_i}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

Which, as we have assumed x is non-stochastic, this is simply a linear combination of a set of random variables (drawn from normals with the same distribution) - thus,

$$\beta_1$$

and

$$\beta_2$$

are normally distributed.