Section 3.1: Multiple Linear Regression

Jared S. Murray
The University of Texas at Austin
McCombs School of Business

The Multiple Regression Model

Many problems involve more than one independent variable or factor which affects the dependent or response variable.

- More than size to predict house price!
- Demand for a product given prices of competing brands, advertising,house hold attributes, etc.

In SLR, the conditional mean of Y depends on X. The Multiple Linear Regression (MLR) model extends this idea to include more than one independent variable.

The MLR Model

Same as always, but with more covariates.

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p + \epsilon$$

Recall the key assumptions of our linear regression model:

(i) The conditional mean of Y is linear in the X_j variables.

(ii) The error term (deviations from line)

- are normally distributed
- independent from each other
- identically distributed (i.e., they have constant variance)

$$(Y|X_1 \ldots X_p) \sim N(\beta_0 + \beta_1 X_1 \ldots + \beta_p X_p, \sigma^2)$$

The MLR Model

Our interpretation of regression coefficients can be extended from the simple single covariate regression case:

$$\beta_j = \frac{\partial E[Y|X_1, \ldots, X_p]}{\partial X_j}$$

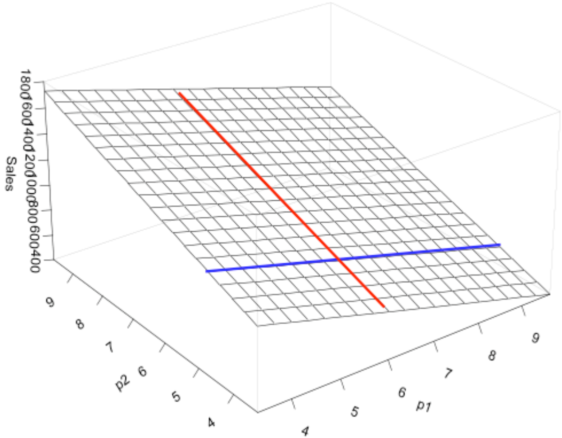
Holding all other variables constant, β_j is the average change in Y per unit change in X_j.

The MLR Model

If $p = 2$, we can plot the regression surface in 3D.

Consider sales of a product as predicted by price of this product (P1) and the price of a competing product (P2).

$$Sales = \beta_0 + \beta_1 P1 + \beta_2 P2 + \epsilon$$



Parameter Estimation

$$Y = \beta_0 + \beta_1 X_1 \ldots + \beta_p X_p + \varepsilon, \quad \varepsilon \sim N(0, \sigma^2)$$

How do we estimate the MLR model parameters?

The principle of Least Squares is exactly the same as before:

- Define the fitted values
- Find the best fitting plane by minimizing the sum of squared residuals.

Then we can use the least squares estimates to find s...

Least Squares

Just as before, each b_i is our estimate of β_i

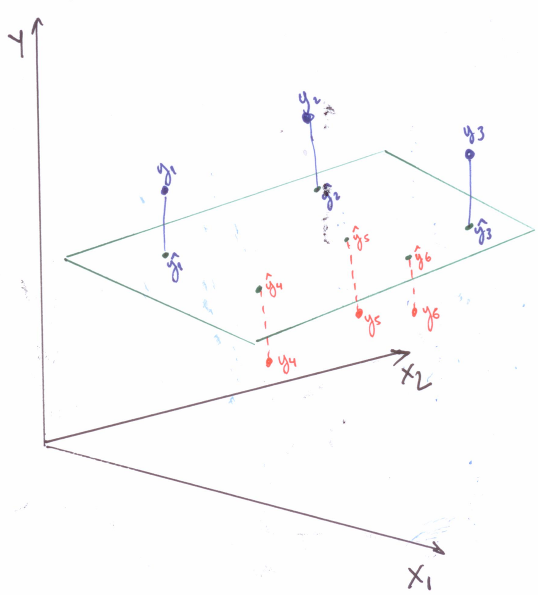
Fitted Values: $\hat{Y}_i = b_0 + b_1 X_{1i} + b_2 X_{2i} \ldots + b_p X_p.$

Residuals: $e_i = Y_i - \hat{Y}_i.$

Least Squares: Find $b_0, b_1, b_2, \ldots, b_p$ to minimize $\sum_{i=1}^{n} e_i^2.$

In MLR the formulas for the b_j's are too complicated so we won't talk about them...

Least Squares



Residual Standard Error

The calculation for s^2 is exactly the same:

$$s^2 = \frac{\sum_{i=1}^{n} e_i^2}{n - p - 1} = \frac{\sum_{i=1}^{n} (Y_i - \hat{Y}_i)^2}{n - p - 1}$$

- $\hat{Y}_i = b_0 + b_1 X_{1i} + \cdots + b_p X_{pi}$
- The residual "standard error" is the estimate for the standard deviation of ϵ,i.e,

$$\hat{\sigma} = s = \sqrt{s^2}.$$

Example: Price/Sales Data

The data...

p1	p2	Sales
5.1356702	5.2041860	144.48788
3.4954600	8.0597324	637.24524
7.2753406	11.6759787	620.78693
4.6628156	8.3644209	549.00714
3.5845370	2.1502922	20.42542
5.1679168	10.1530371	713.00665
3.3840914	4.9465690	346.70679
4.2930636	7.7605691	595.77625
4.3690944	7.4288974	457.64694
7.2266002	10.7113247	591.45483
...	...	...

Example: Price/Sales Data

Model: $Sales_i = \beta_0 + \beta_1 P1_i + \beta_2 P2_i + \epsilon_i$, $\epsilon \sim N(0, \sigma^2)$

```
fit = lm(Sales~p1+p2, data=price_sales)
print(fit)

##
## Call:
## lm(formula = Sales ~ p1 + p2, data = price_sales)
##
## Coefficients:
## (Intercept)           p1           p2
##      115.72       -97.66       108.80
```

$b_0 = \hat{\beta}_0 = 115.72$, $b_1 = \hat{\beta}_1 = -97.66$, $b_2 = \hat{\beta}_2 = 108.80$.

```
print(sigma(fit)) # sigma(fit) extracts s from an lm fit

## [1] 28.41801
```

$s = \hat{\sigma} = 28.42$

Prediction in MLR: Plug-in method

Suppose that by using advanced corporate espionage tactics, I discover that my competitor will charge \$10 the next quarter. After some marketing analysis I decided to charge \$8. How much will I sell?

Our model is

$$Sales = \beta_0 + \beta_1 P1 + \beta_2 P2 + \epsilon$$

with $\epsilon \sim N(0, \sigma^2)$

Our estimates are $b_0 = 115$, $b_1 = -97$, $b_2 = 109$ and $s = 28$ which leads to

$$Sales = 115 + -97 * P1 + 109 * P2 + \epsilon$$

with $\epsilon \sim N(0, 28^2)$

Plug-in Prediction in MLR

By plugging-in the numbers,

$$\begin{aligned} Sales &= 115.72 + -97.66 * 8 + 108.8 * 10 + \epsilon \\ &\approx 422 + \epsilon \end{aligned}$$

$$Sales | P1 = 8, P2 = 10 \sim N(422.44, 28^2)$$

and the 95% Prediction Interval is $(422 \pm 2 * 28)$

$$366 < Sales < 478$$

Better Prediction Intervals in R

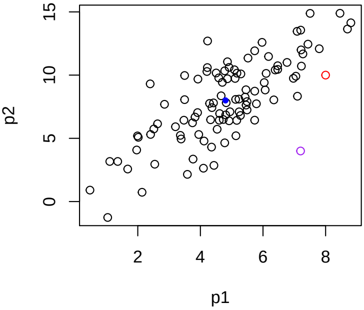
```
new_data = data.frame(p1=8, p2=10)
predict(fit, newdata = new_data,
        interval="prediction", level=0.95)

##        fit      lwr      upr
## 1 422.4573 364.2966 480.6181
```

Pretty similar to (366,478), right?

Like in SLR, the difference gets larger the "farther" our new point (here $P1 = 8,\ P2 = 10$) gets from the observed data

Still be careful extrapolating!

In SLR "farther" is measured as distance from $\bar{X}$; in MLR the idea of extrapolation is a little more complicated.



Blue: (P1=$\bar{P}1, P2 = \bar{P}2$), red: (P1=8, P2=10), purple: (P1=7.2, P2=4). Red looks "consistent" with the data; purple not so much.

Residuals in MLR

As in the SLR model, the residuals in multiple regression are purged of any linear relationship to the independent variables. Once again, they are on average zero.

Because the fitted values are an exact linear combination of the X's they are not correlated with the residuals.

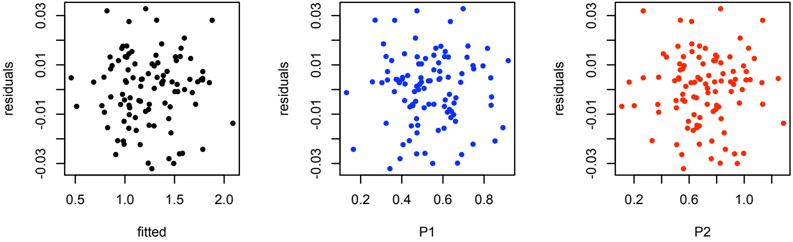
We decompose Y into the part predicted by X and the part due to idiosyncratic error.

$$Y = \hat{Y} + e$$

$$\bar{e} = 0; \quad \text{corr}(X_j, e) = 0; \quad \text{corr}(\hat{Y}, e) = 0$$

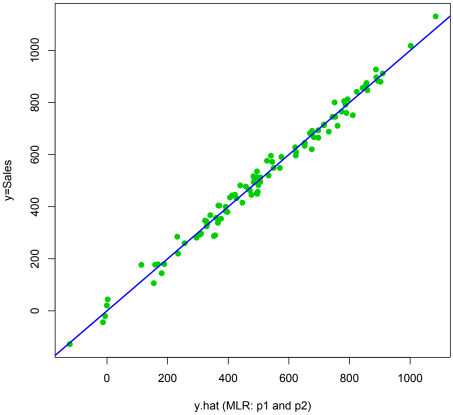
Residuals in MLR

Consider the residuals from the Sales data:



Fitted Values in MLR

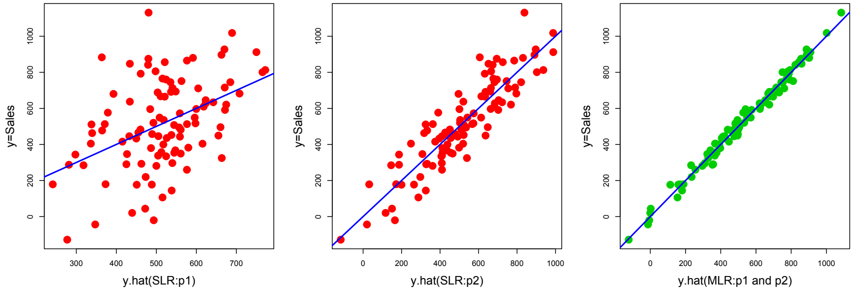
Another great plot for MLR problems is to look at Y (true values) against $\hat{Y}$ (fitted values).



If things are working, these values should form a nice straight line. Can you guess the slope of the blue line?

Fitted Values in MLR

Now, with $P1$ and $P2$...



- First plot: *Sales* regressed on $P1$ alone...
- Second plot: *Sales* regressed on $P2$ alone...
- Third plot: Sales regressed on $P1$ and $P2$

R-squared

- We still have our old variance decomposition identity...

$$SST = SSR + SSE$$

- ... and R^2 is once again defined as

$$R^2 = \frac{SSR}{SST} = 1 - \frac{SSE}{SST} = 1 - \frac{\text{var}(e)}{\text{var}(y)}$$

  telling us the percentage of variation in Y explained by the X's. Again, $R^2 = \text{corr}(Y, \hat{Y})^2$.

- In R, R^2 is found in the same place...

Back to Baseball

$$R/G = \beta_0 + \beta_1 OBP + \beta_2 SLG + \epsilon$$

```
both_fit = lm(RPG ~ OBP + SLG, data=baseball)
print(both_fit)

##
## Call:
## lm(formula = RPG ~ OBP + SLG, data = baseball)
##
## Coefficients:
## (Intercept)          OBP          SLG
##      -7.014       27.593        6.031
```

Back to Baseball

```
summary(both_fit)
```

```
## ...
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -7.0143     0.8199  -8.555 3.61e-09 ***
## OBP          27.5929     4.0032   6.893 2.09e-07 ***
## SLG           6.0311     2.0215   2.983  0.00598 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1486 on 27 degrees of freedom
## Multiple R-squared:  0.9134,Adjusted R-squared:  0.9069
## F-statistic: 142.3 on 2 and 27 DF,  p-value: 4.563e-15
```

Remember, our highest R^2 from SLR was 0.88 using OBP.

Back to Baseball

$$R/G = \beta_0 + \beta_1 OBP + \beta_2 SLG + \epsilon$$

```
both_fit = lm(RPG ~ OBP + SLG, data=baseball); coef(both_fit)

## (Intercept)         OBP         SLG
##   -7.014316   27.592869    6.031124
```

Compare to individual SLR models:

```
obp_fit = lm(RPG ~ OBP, data=baseball); coef(obp_fit)

## (Intercept)         OBP
##   -7.781631   37.459254
```

```
slg_fit = lm(RPG ~ SLG, data=baseball); coef(slg_fit)

## (Intercept)         SLG
##   -2.527758   17.541932
```

Back to Baseball: Some questions

Why are the b_j's smaller in the SLG+OBP model?

Remember, in MLR β_j gives you the average change in Y for a 1 unit change in X_j **given (i.e. holding constant) the other X's in the model**.

Here, OBP is less informative once we know SLG, and vice-versa. In general, coefficients can stay about the same, go up, go down and even change sign as we add variables. (To be continued!)

Why did R^2 go up? Does this mean we have a better model with OBP+SLG? Not necessarily...