

Multiple Regression

James H. Steiger

Department of Psychology and Human Development
Vanderbilt University

P310, 2011

Psychology 310 — Course Goals and Strategy

- 1 Fitting Regression Models
- 2 The Multiple Regression Model
- 3 Multiple Regression with Interactions

Fitting Regression Models

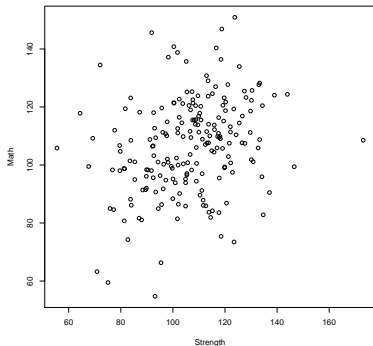
- In our previous lectures, we saw how to compute the slope and intercept of the best-fitting linear regression line relating two variables Y and X .
- Let's load in some artificial data and review those calculations.
- The data can be downloaded from a file called *regression.data.txt*.
- There are 3 variables in the file, which represents scores on *Math* and *Strength* for 100 sixth grade boys and 100 eighth grade boys. The *Grade* variable is coded 0 for sixth graders and 1 for eighth graders.

```
> data <- read.table("regression.data.txt",  
+ header=T, sep=",")  
> attach(data)
```

Fitting Regression Models

- If we plot all the data together, we get a scatterplot like this:

```
> plot(Strength, Math)
```



Fitting Regression Models

- In an earlier lecture, we saw that the regression slope and intercept for the best fitting straight line $\hat{Y} = \beta_1 X + \beta_0$ can be written as

$$\begin{aligned}\beta_1 &= r_{yx} s_y / s_x \\ \beta_0 &= \bar{Y}_\bullet - \beta_1 \bar{x}_\bullet\end{aligned}$$

- We can compute the values easily for predicting *Math* from *Strength* as

```
> beta.1 <-  
+   cor(Math,Strength) * sd(Math) / sd(Strength)  
> beta.0 <- mean(Math) - beta.1 * mean(Strength)  
> beta.1  
[1] 0.2567642  
> beta.0  
[1] 79.25515
```

Fitting Regression Models

- R can fit any linear regression model effortlessly with its `lm` command.
- R is an “object-oriented” language. Objects can contain data and functions, and are also able to respond in specific ways when functions operate on them. When fitting linear models in R, it is wise to save the results in a linear model object. Here is an example:

```
> model.1 <- lm(Math ~ 1 + Strength)
```

- The call to `lm` included a model specification. The “1” stands for the intercept term. All variable names are included as predictors. So, the above function call fits the linear model

$$Math = \beta_0 + \beta_1 Strength.$$

Fitting Regression Models

- To see the numerical results of the model fit, you can use the function `summary` on the model fit object.

```
> summary(model.1)
```

Call:

```
lm(formula = Math ~ 1 + Strength)
```

Residuals:

```
      Min       1Q   Median       3Q      Max
-48.441 -10.599   0.066   9.791  42.772
```

Coefficients:

```
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  79.25515     7.01015   11.31 < 2e-16 ***
Strength      0.25676     0.06501    3.95 0.000109 ***
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Residual standard error: 15.74 on 198 degrees of freedom

Multiple R-squared: 0.07303, Adjusted R-squared: 0.06835

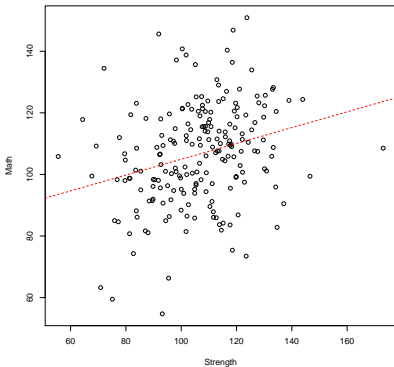
F-statistic: 15.6 on 1 and 198 DF, p-value: 0.0001088

- Notice that, in this case, both β_0 (the intercept) and β_1 (the coefficient of *Strength*) are statistically significant, having *p*-values less than .001.

Fitting Regression Models

- To plot the regression line, you can use the `abline` function on the linear model object. I chose to plot a dotted red line.

```
> plot(Strength,Math)  
> abline(model.1,col="red",lty=2)
```



The Multiple Regression Model

- The multiple regression model includes additional terms besides the single predictor in the linear regression model.
- As a simple example, consider the model

$$\hat{Y} = b_0 + b_1X_1 + b_2X_2 + e \quad (1)$$

- It is easy to fit this model using the `lm` function.
- Below, we fit the model predicting *Math* from *Strength* and *Grade*.
- Multiple R^2 is the correlation between the predicted scores and the criterion. With only one predictor, it is equal to the correlation between the predictor and the criterion.
- Note that, with *Grade* in the equation, the R^2 value increased to .20, while coefficient for *Strength* is no longer significant. On the other hand, the coefficient for *Grade* is highly significant.
- How should we interpret these results?

```
> model.2 <- lm(Math ~ 1 + Strength + Grade)
> summary(model.2)
```

Call:

```
lm(formula = Math ~ 1 + Strength + Grade)
```

Residuals:

Min	1Q	Median	3Q	Max
-44.235	-10.156	0.261	9.999	37.250

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	91.21817	6.86970	13.278	< 2e-16 ***
Strength	0.08319	0.06803	1.223	0.223
Grade	13.03284	2.32957	5.595	7.33e-08 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 14.66 on 197 degrees of freedom

Multiple R-squared: 0.2001, Adjusted R-squared: 0.192

F-statistic: 24.64 on 2 and 197 DF, p-value: 2.807e-10

The Multiple Regression Model

- In this case, one of our predictors, *Strength*, is continuous, while the other, *Grade*, is categorical (binary) and is scored 0-1.
- Because *Grade* is categorical 0-1, for 6th graders, the model becomes

$$Y = \beta_0 + \beta_1 \textit{Strength} + e \quad (2)$$

- For 8th graders, the equation becomes

$$Y = (\beta_0 + \beta_2) + \beta_1 \textit{Strength} + e \quad (3)$$

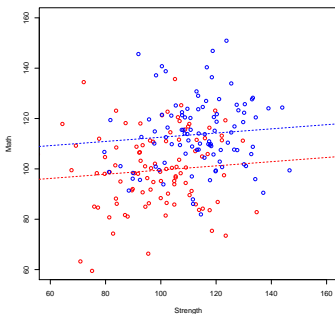
- In other words, this model, in effect, simultaneously fits two regression models, with different intercepts but the same slope, to the *Strength-Math* data. The 6th graders have an intercept of β_0 and a slope of β_1 , while the 8th graders have an intercept of $\beta_0 + \beta_2$, and a slope of β_1 . So, a test that $\beta_2 = 0$ is also a test of equal intercepts (given equal slopes).
- Most textbooks begin the discussion of multiple regression with two continuous predictors. I started our discussion of multiple regression with this somewhat unusual example with a categorical predictor and a continuous predictor in order to emphasize, from the beginning, that multiple linear regression modeling offers “more than meets the eye” in analyzing data.

The Multiple Regression Model

Separate Intercepts, Same Slopes

- Here is a picture of the data with the two separate regression lines

```
> plot(Strength[1:100],Math[1:100],col="red",  
+ xlim=c(60,160),ylim=c(60,160),xlab="Strength",ylab="Math")  
> points(Strength[101:200],Math[101:200],col="blue")  
> beta <- coef(model.2)  
> abline(beta[1],beta[2],col="red",lty=2)  
> abline(beta[1]+ beta[3],beta[2],col="blue",lty=2)
```



The Multiple Regression Model

- Notice that the preceding model assumed, implicitly, that there is no difference in the slopes of the regression lines for 6th and 8th graders.
- Can we fit a model that allows different slopes *and* different intercepts for the two grades?

The Multiple Regression Model with Interactions

- Suppose we fit the following model to our *Strength-Math* data:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_1 X_2 + e \quad (4)$$

- For the special case where X_2 is a binary variable coded 0-1, 6th graders have $X_2 = 0$, and so the model becomes

$$Y = \beta_0 + \beta_1 X_1 + e \quad (5)$$

- For 8th graders with $X_2 = 1$, we get

$$\begin{aligned} Y &= \beta_0 + \beta_1 X_1 + \beta_2 + \beta_3 X_1 + e \\ &= (\beta_0 + \beta_2) + (\beta_1 + \beta_3) X_1 + e \end{aligned}$$

- Note that this is a model that specifies different slopes and intercepts for 6th and 8th graders. The 6th graders have a slope of β_1 and an intercept of β_0 , while the 8th graders have a slope of $\beta_1 + \beta_3$ and an intercept of $\beta_0 + \beta_2$. A test that $\beta_2 = 0$ corresponds to a test of equal intercepts, while a test that $\beta_3 = 0$ corresponds to a test of equal slopes.

The Multiple Regression Model with Interactions

- Here is how you specify this model in R

```
> model.3 <- lm(Math ~ Strength + Grade + Strength:Grade)
> summary(model.3)
```

Call:

```
lm(formula = Math ~ Strength + Grade + Strength:Grade)
```

Residuals:

Min	1Q	Median	3Q	Max
-44.143	-9.927	0.265	10.197	37.696

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	89.56084	9.55565	9.373	<2e-16 ***
Strength	0.09999	0.09571	1.045	0.297
Grade	16.66965	14.72432	1.132	0.259
Strength:Grade	-0.03412	0.13640	-0.250	0.803

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 14.69 on 196 degrees of freedom

Multiple R-squared: 0.2004, Adjusted R-squared: 0.1881

F-statistic: 16.37 on 3 and 196 DF, p-value: 1.557e-09

The Multiple Regression Model

Separate Slopes and Intercepts

- Here is a picture of the data with the two separate regression lines

```
> plot(Strength[1:100],Math[1:100],col="red",  
+ xlim=c(60,160),ylim=c(60,160),xlab="Strength",ylab="Math")  
> points(Strength[101:200],Math[101:200],col="blue")  
> beta <- coef(model.3)  
> abline(beta[1],beta[2],col="red",lty=2)  
> abline(beta[1]+ beta[3],beta[2]+beta[4],col="blue",lty=2)
```

