Multivariate OLS Estimation of GPA Data

Introduction

In this demonstration, we will revisit the GPA dataset to practice estimating a multiple regression model.

Previously, we fit a bivariate linear model, predicting GPA as a function of ACT score

$$colGPA = \beta_0 + \beta_1 ACT + u$$

Here's a quick recap of the steps we took.

Bivariate Model Estimation

First, we perform our basic setup and loading of data.

```
#setwd("~/Desktop/data_w203")
#getwd()

# We use the stargazer package to display nice regression tables.
library(stargazer)
```

```
##
## Please cite as:
##  Hlavac, Marek (2015). stargazer: Well-Formatted Regression and Summary Statistics Tables.
##  R package version 5.2. http://CRAN.R-project.org/package=stargazer
```

```
load("GPA1.rdata")
head(data)
```

We examined the colGPA and ACT variables individually, which we omit here.

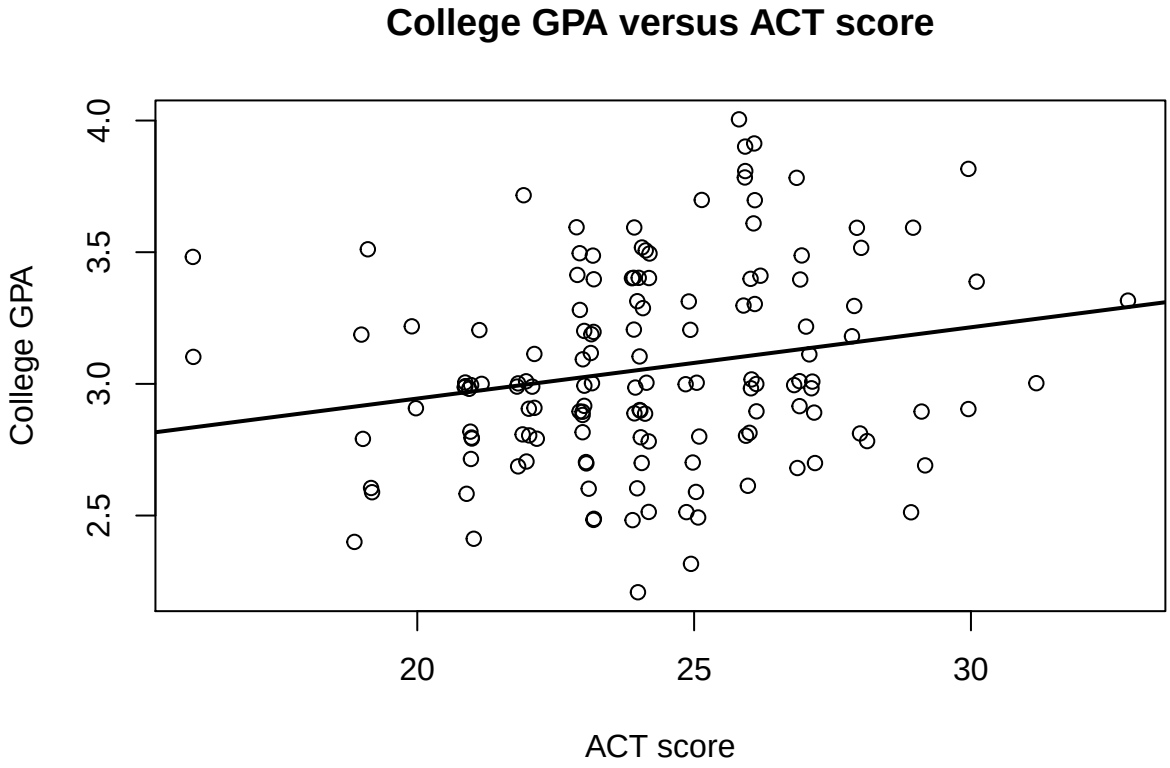
We then created a scatterplot of the two variables and fit a linear model.

```
# create the scatterplot
plot(jitter(data$ACT), jitter(data$colGPA), xlab = "ACT score", ylab = "College GPA", main = "College G

# fit the linear model
(model1 = lm(colGPA ~ ACT, data = data))
```

```
##
## Call:
## lm(formula = colGPA ~ ACT, data = data)
##
## Coefficients:
## (Intercept)          ACT
##     2.40298      0.02706
```

```
# Add regression line to scatterplot
abline(model1, lwd=2)
```

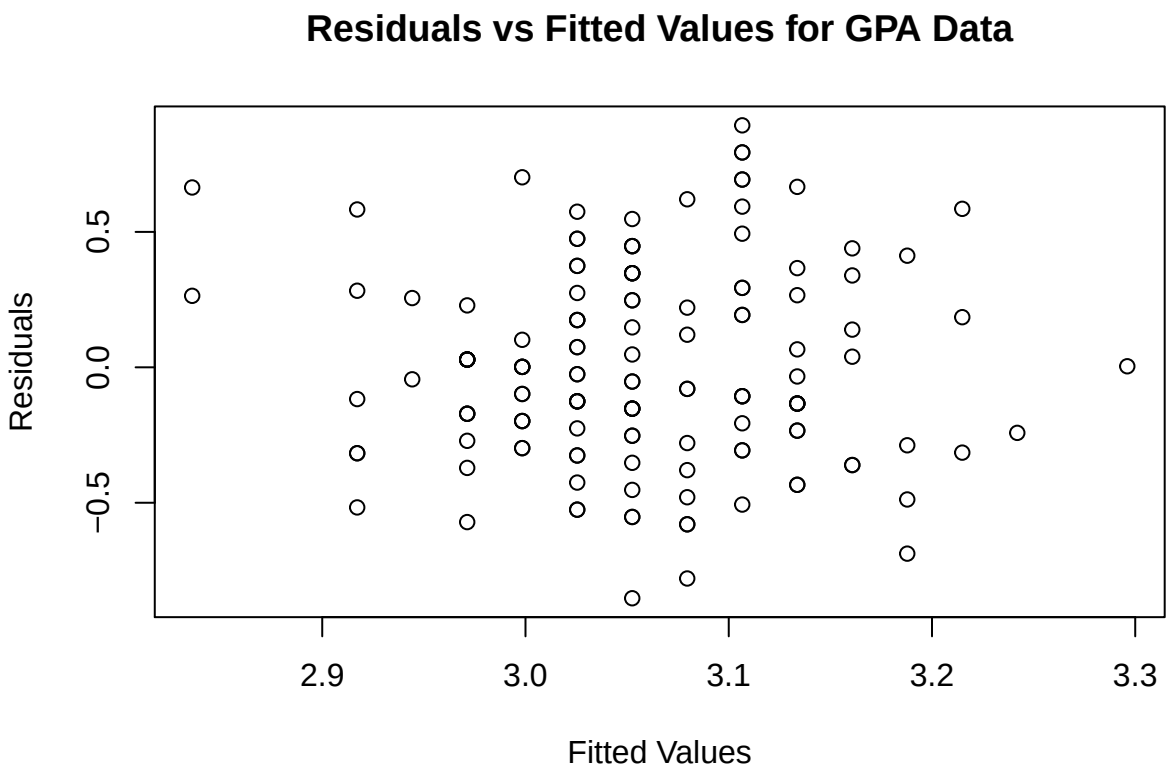


Check the assumptions for unbiasedness.

1. Linear population model - Since we haven't constrained the error, nothing to check yet.
2. Random Sampling - We need to understand where the data comes from. Wooldrige says "Christopher Lemmon, a former MSU undergraduate, collected these data from a survey he took of MSU students in Fall 1994." We don't know the details of the survey, so we can't assess this assumptions perfectly. We notice that there are no survey weights included, and there is the possibility that the participants are not representative of the population distribution. There is nothing we can do to correct this, so we note this as a potential weakness and proceed with the analysis.
3. Multicolinearity - Since we have just one predictor variable, we only need to observe that the variable isn't constant.
4. Zero-conditional mean - Here, we examine the residuals versus fitted values plot.

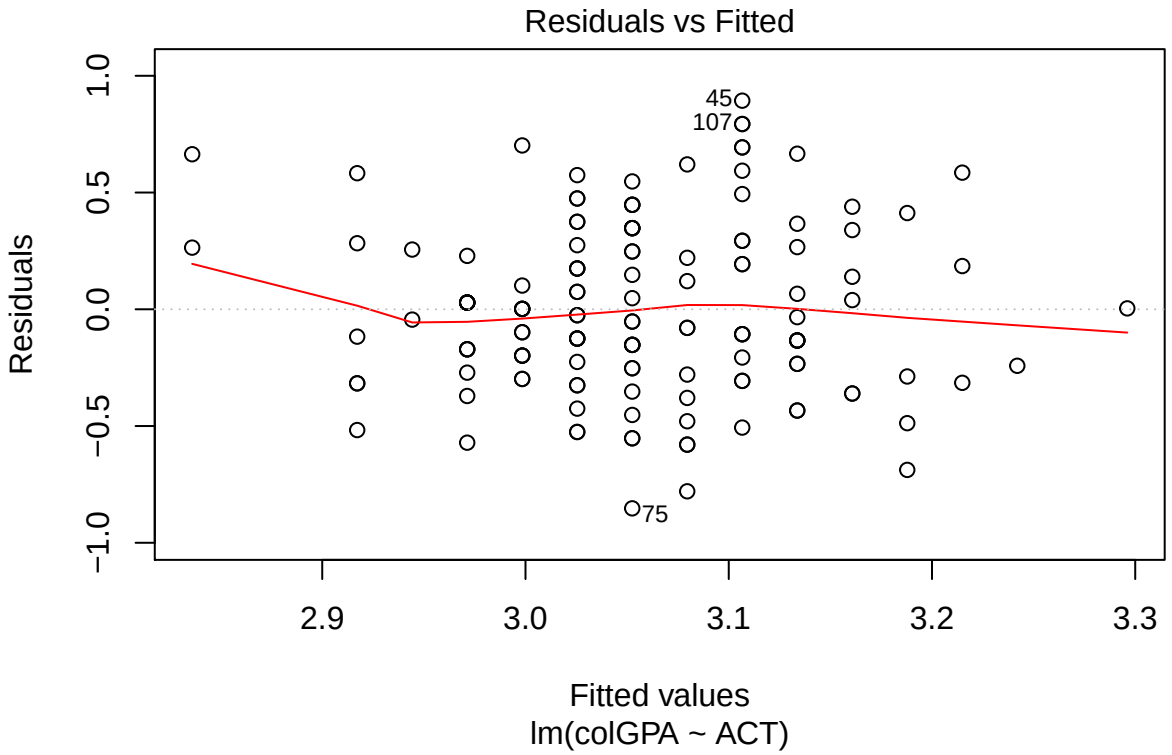
We can construct the residuals versus fitted values plot manually as follows:

```
plot(model1$fitted.values, model1$residuals,
     main = "Residuals vs Fitted Values for GPA Data",
     xlab = "Fitted Values", ylab = "Residuals")
```



We can also use the plot command directly on our linear model object. This command gives us a selection of the most common diagnostic plots. The residuals-vs-fitted values plot is the very first one.

```
plot(model1, which = 1)
```



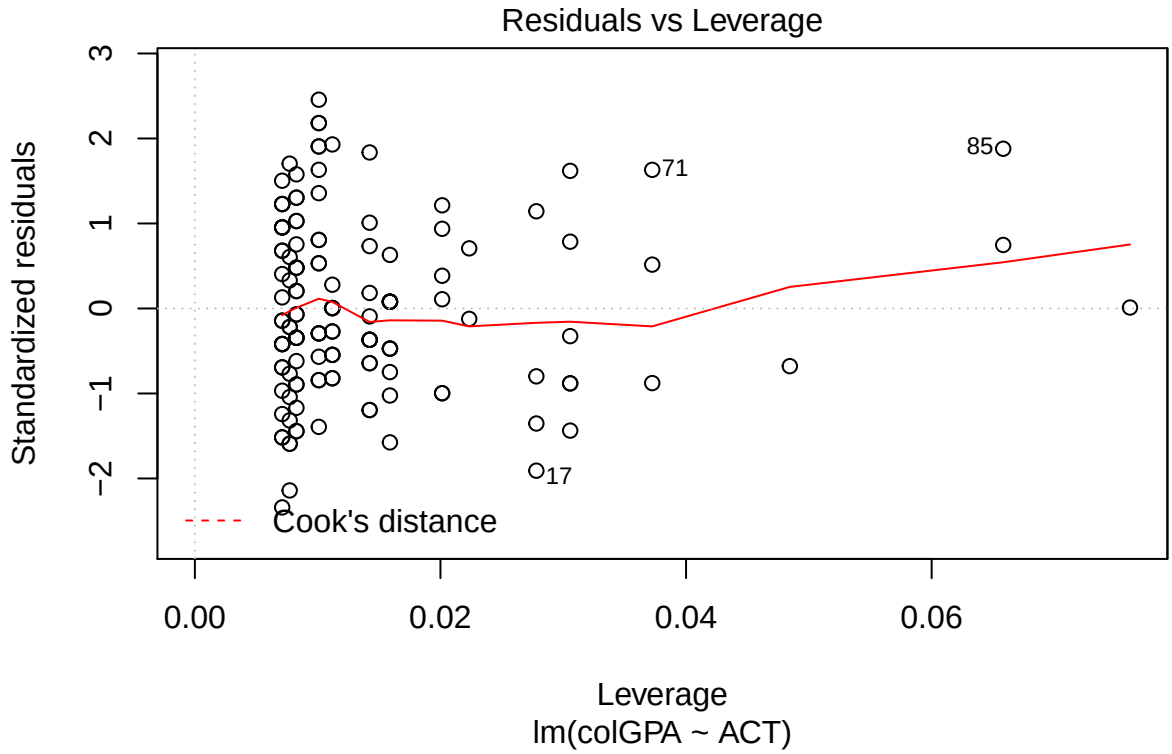
Notice that there is a bit of an uptick on the left of the graph, but this may also reflect the fact that there

are only two data points there. On the whole, there's not much to worry about here.

Even if we did detect a violation of zero-conditional mean, we have a large sample, so we could rely on asymptotics. Since we're just fitting an associative model, we can assume exogeneity and we know that our coefficients will be consistent.

Next, we will want to examine our data to check for any unusually influential cases. We can use a residuals vs. leverage plot for this purpose. This plot shows how much leverage each case has on the x-axis. It would also indicate any points with Cook's distance greater than 1 using a dotted red line.

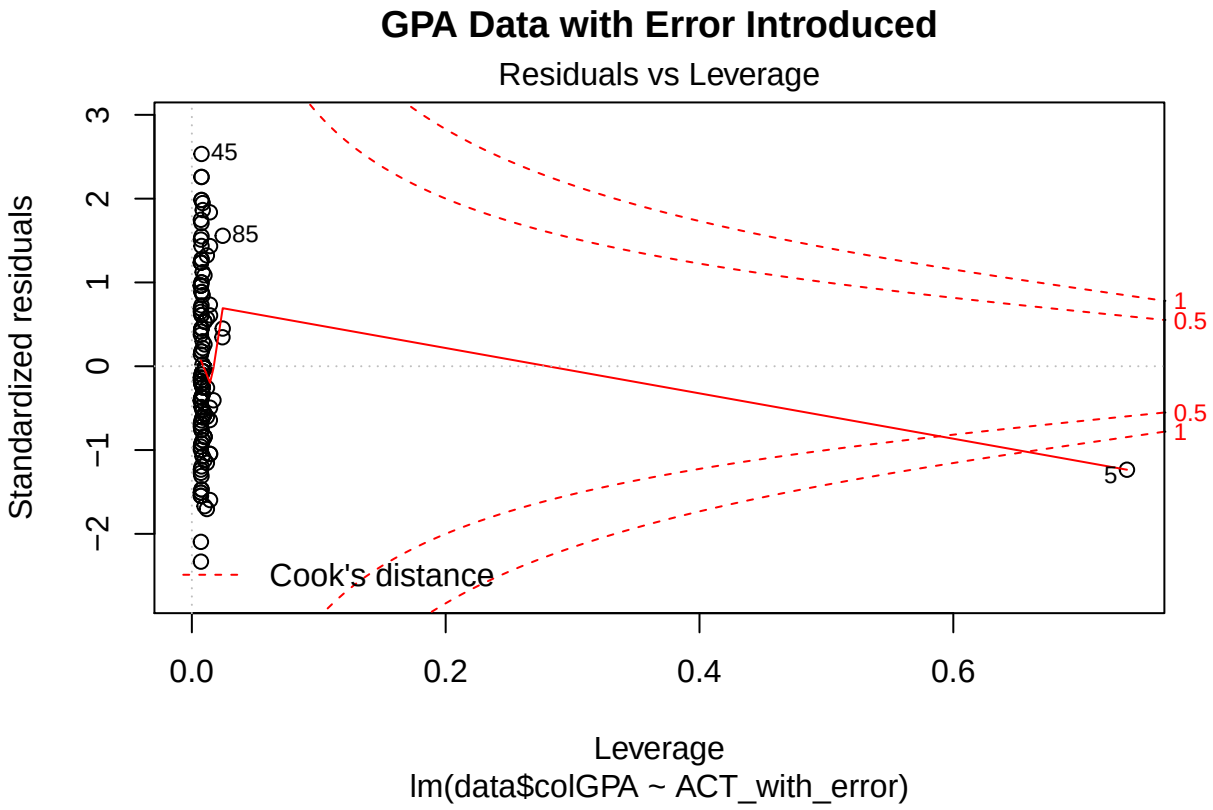
```
plot(model1, which = 5)
```



In this case, no point has Cook's distance close to 1, so the dotted red line is absent.

The following code shows what would happen if we introduced an error into the data set, resulting in a point with high influence.

```
ACT_with_error = data$ACT
ACT_with_error[5] = 80
model1_with_error = lm(data$colGPA ~ ACT_with_error)
plot(model1_with_error, which=5, main = "GPA Data with Error Introduced")
```



Notice that the point now stands out as having Cook's distance greater than 1.

Warning: when we find an influential case, we never automatically remove it from the data set. Cook's distance is only a tool to draw our attention to influential cases. We then have to assess whether the case represents an error, or if the value is meaningful. If there are reasons to be suspicious but not enough to remove the case, we could discuss how the results would change with the case removed.

We've argued that our coefficients are unbiased and there are no problematic cases that should be removed from the data set. When we move on to statistical inference, there will be more assumptions for us to test. We'll learn more about regression diagnostics in the next unit.

Multivariate Linear Model Estimation

Here, we recreate the regression from the lecture and from Woodridge chapter 3. We predict colGPA from both ACT and high school GPA (hsGPA). Our second model looks like this.

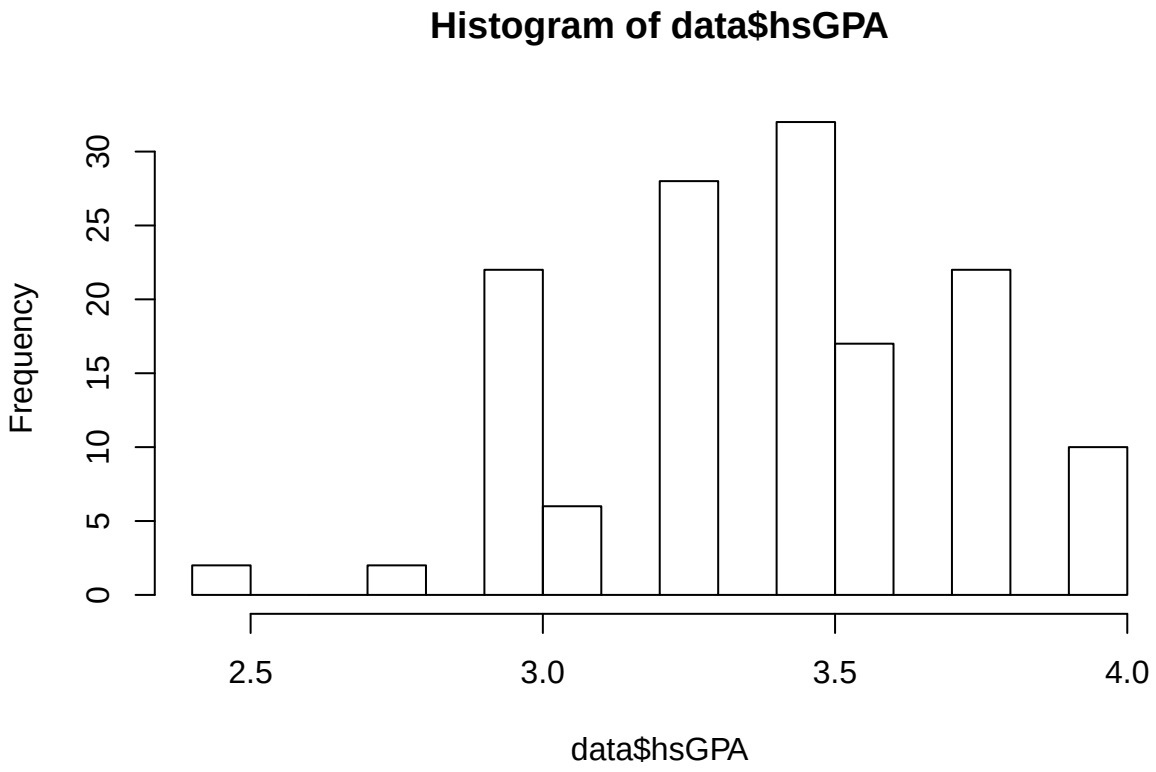
$$colGPA = \beta_0 + \beta_1 ACT + \beta_2 hsGPA + u$$

We first examine the high school GPA variable.

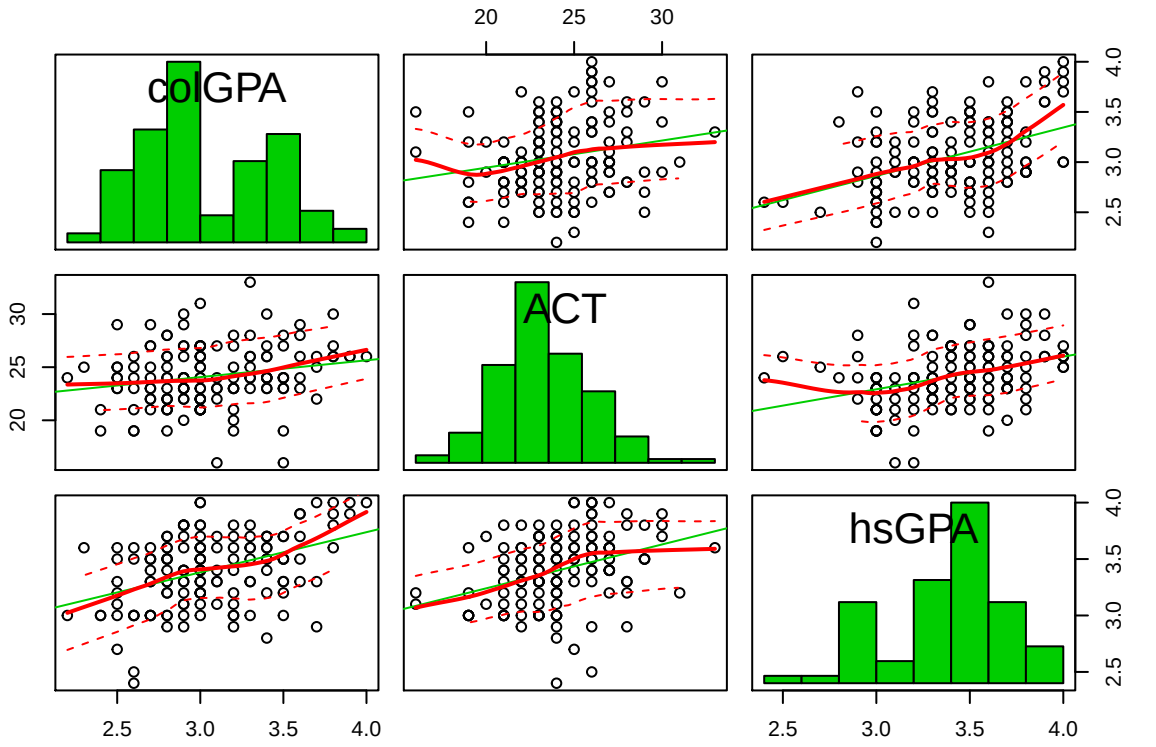
```
summary(data$hsGPA)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   2.400   3.200   3.400   3.402   3.600   4.000
```

```
hist(data$hsGPA, breaks = 20)
library(car)
```




```
scatterplotMatrix(data[,c("colGPA", "ACT", "hsGPA")], diagonal = "histogram")
```



We noticed that there is some negative skew. There are no unusual peaks and all values are in the correct range from 0 to 4. Even though the variable is attenuated between 0 and 4, there is no large build-up of data at the ends.

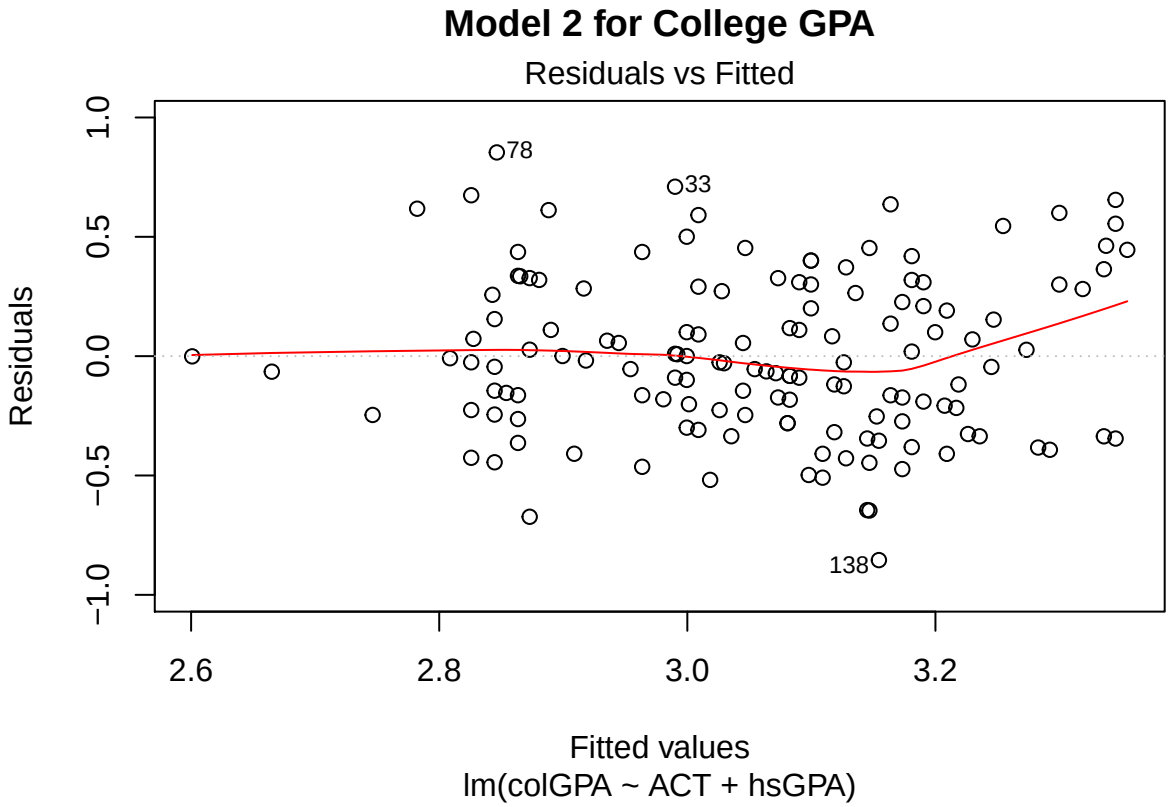
Next, we fit the linear model.

```
(model2 = lm(colGPA ~ ACT + hsGPA, data = data))
```

```
##
## Call:
## lm(formula = colGPA ~ ACT + hsGPA, data = data)
##
## Coefficients:
## (Intercept)          ACT        hsGPA
##    1.286328     0.009426     0.453456
```

Once again, let's look at the residuals versus fitted values plot

```
plot(model2, which = 1, main = "Model 2 for College GPA")
```



There's a small uptick on the right, which might indicate a violation of zero-conditional mean. However, we have a large sample and we're building an associative model, so we still know that our coefficients are consistent.

Recall that the coefficient of ACT in the bivariate model was .027 Notice that it has now fallen by about a factor of 3, to .0094.
The coefficient for hsGPA is .453 - almost 1/2. This means that a student with a letter grade higher average in high school but the same ACT score is expected to have a half-letter grade higher average in college.

What's happening is that ACT score in the simple regression was probably picking up some of the effect of high school GPA, since students with a higher ACT score likely have a higher GPA as well. This regression suggests that high school GPA is a better predictor of college GPA than ACT score is.

Let's compare the R-squares for our two models.

```
summary(model1)$r.square
```

```
## [1] 0.04274732
```

```
summary(model2)$r.square
```

```
## [1] 0.1764216
```

In our simple regression, R-square was 0.043. Notice that it has now increased to 0.176. This is a notable increase, but remember that R-squared can only go up when adding new variables.

For an assessment of model fit that penalizes extra variables, we can use the Akaike Information Criterion (AIC) or the Bayesian Information Criterion (BIC)

```
AIC(model1)
```

```
## [1] 120.3534
```

```
AIC(model2)
```

```
## [1] 101.1457
```

Notice that the AIC has gone down, indicating a better model fit. This isn't too surprising, since hsGPA seems to be such a good predictor of college GPA.

Presenting Regression Output

For a lot of reasons, we usually want to display the results of more than one linear model.

1. There can be good arguments for different specifications
2. We want to show that an effect is robust across models
3. We want to show that we're not cherry-picking a model that supports our argument

Because of these reasons, we often want to present the results of multiple models in a *regression table*. The stargazer package is a great way to create these tables. It takes a little time to learn stargazer, but it's well worth the effort.

```
stargazer(model1, model2, type = "latex", report = "vc",
          header = FALSE,
          title = "Linear Models Predicting College GPA",
          keep.stat = c("aic", "rsq", "n"),
          omit.table.layout = "n")
```

Table 1: Linear Models Predicting College GPA

	Dependent variable:	
	colGPA	
	(1)	(2)
ACT	0.027	0.009
hsGPA		0.453
Constant	2.403	1.286
Observations	141	141
R^2	0.043	0.176

We could also output an html table by setting type = “html” and including the argument out = “GPA_table.htm”.

This table invites the reader to compare the models and reason about their differences. It is not unusual to see a regression table with 5, or even 10 different model specifications. Often, you’ll see one model with just the key variables of interest, then a series of models that include different blocks of covariates that make sense together. This process will make more sense as you learn more about model specification.

The Stargazer Cheatsheet, by Jake Russ, is a great place to get started with stargazer. http://jakeruss.com/cheatsheets/stargazer.html