23:46 Saturday 24th October, 2015
See updates and corrections at http://www.stat.cmu.edu/~cshalizi/mreg/

Lecture 16: Polynomial and Categorical Regression

36-401, Fall 2015, Section B

22 October 2015

Contents

1 Essentials of Multiple Linear Regression

We predict a scalar random variable Y as a linear function of p different predictor variables $X_1, \ldots X_p$, plus noise:

$$Y = \beta_0 + \beta_1 X_1 + \ldots \beta_p X_p + \epsilon$$

and assume that $\mathbb{E}[\epsilon|X] = 0$, $\mathrm{Var}[\epsilon|X] = \sigma^2$, with ϵ being uncorrelated across observations. In matrix form,

$$\mathbf{Y} = \mathbf{X}\beta + \epsilon$$

the design matrix $\mathbf{X}$ including an extra column of 1s to handle the intercept, and $\mathbb{E}[\epsilon|\mathbf{X}] = \mathbf{0}$, $\mathrm{Var}[\epsilon|\mathbf{X}] = \sigma^2\mathbf{I}$.

If we add the Gaussian noise assumption, $\epsilon \sim MVN(\mathbf{0}, \sigma^2\mathbf{I})$, independent of all the predictor variables.

The least squares estimate of the coefficient vector, which is also the maximum likelihood estimate if the noise is Gaussian, is

$$\widehat{\beta} = (\mathbf{x}^T\mathbf{x})^{-1}\mathbf{x}^T\mathbf{y}$$

These are unbiased, with variance $\sigma^2(\mathbf{x}^T\mathbf{x})^{-1}$. Under the Gaussian noise assumption, $\widehat{\beta}$ itself has a Gaussian distribution. The standard error $\widehat{\mathrm{se}}\left[\hat{\beta}_i\right] = \sigma\sqrt{(\mathbf{x}^T\mathbf{x})^{-1}_{ii}}$. Fitted values are given by $\mathbf{x}\widehat{\beta} = \mathbf{H}\mathbf{y}$, and residuals by $\mathbf{e} = (\mathbf{I}-\mathbf{H})\mathbf{y}$. Fitted values $\widehat{\mathbf{m}}$ and residuals $\mathbf{e}$ are also unbiased and have Gaussian distributions, with variance matrices $\sigma^2\mathbf{H}$ and $\sigma^2(\mathbf{I}-\mathbf{H})$, respectively.

When (as is usually the case) σ^2 is unknown, the maximum likelihood estimator is the in-sample mean-squared error, $n^{-1}(\mathbf{e}^T\mathbf{e})$ is a negatively biased estimator of σ^2: $\mathbb{E}\left[\hat{\sigma}^2\right] = \sigma^2\frac{n-p-1}{n}$. Under the Gaussian noise assumption, $n\hat{\sigma}^2/\sigma^2 \sim \chi^2_{n-p-1}$. Also under the Gaussian noise assumption, the Gaussian sampling distribution of any particular coefficient or conditional mean can be converted into a t distribution, with $n-p-1$ degrees of freedom, by using the appropriate standard error, obtained by plugging in the de-biased estimate of σ^2.

None of these results require any assumptions on the predictor variables X_i, except that they take real numerical values, and that they are linearly independent.

2 Adding Curvature: Polynomial Regression

Because the predictor variables are almost totally arbitrary, there is no harm in making one predictor variable a function of another, so long as it isn't a linear function. In particular, there is nothing wrong with a model like

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_1^2 + \ldots \beta_d X_1^d + \beta_{d+1} X_2 + \ldots \beta_{p+d-1} X_p + \epsilon$$

where instead of Y being linearly related to X_1, it's polynomially related, with the order of the polynomial being d. We just add $d-1$ columns to the design matrix $\mathbf{x}$, containing $x_1^2, x_1^3, \ldots x_1^d$, and treat them just as we would any other predictor variables. With this expanded design matrix, it's still true that $\widehat{\mathbf{x}} = (\mathbf{x}^T\mathbf{x})^{-1}\mathbf{x}^T\mathbf{y}$, that fitted values are $\mathbf{H}\mathbf{y}$ (using the expanded $\mathbf{x}$ to get $\mathbf{H}$), etc. The number of degrees of freedom for the residuals will be $n-(p+1+(d-1))$.

Nor is there principled reason why every predictor variable can't have its own polynomial, each with (potentially) a different degree d_i. In that case, numbering the βs sequentially gets tricky, and better notation would be something like

$$Y = \beta_0 + \sum_{i=1}^{p}\sum_{j=1}^{d_i}\beta_{i,j}X_i^j + \epsilon$$

though then we'd have to remember to "stack" the $\beta_{i,j}$s into a vector of length $1 + \sum_{i=1}^{p} d_i$ for estimation.

Mathematically, we are treating X_i and X_i^2 (and X_i^3, etc.) as distinct predictor variables, but that's fine, since they won't be linearly dependent on each other[1], or linearly dependent on other predictors[2]. Again, we just expand the design matrix with extra columns for all the desired powers of each predictor variable. The number of degrees of freedom for the residuals will be $n - (1 + \sum_i d_i)$.

There are a bunch of mathematical and statistical points to make about polynomial regression, but let's take a look at how we'd actually estimate one of these models in R first.

2.1 R Practicalities

There are a couple of ways of doing polynomial regression in R.

The most basic is to manually add columns to the data frame with the desired powers, and then include those extra columns in the regression formula:

```
df$x.sq <- df$x^2
lm(y~x+x.sq, data=df)
```

I do not recommend using this form, since it means that you need to do a lot of repetitive, boring, error-prone work, and get it exactly right. (For example, to do predictions with `predict`, you'd need to specify the values for all the powers of all the predictors.)

A somewhat more elegant alternative is to tell R to use various powers in the formula itself:

```
lm(y ~ x + I(x^2), data=df)
```

Here `I()` is the **identity function**, which tells R "leave this alone". We use it here because the usual symbol for raising to a power, `^`, has a special meaning in linear-model formulas, relating to interactions. (We'll cover this in Lecture 19, or, if you're impatient, see `help(formula.lm)`.) When you do this, `lm` will create the relevant columns in the matrix it uses internally to calculate the estimates, but it leaves `df` alone. When it comes time to make a prediction, however, R will take care of the transformations on the new data.

Finally, since it can grow tedious to write out all the powers one wants, there is the convenience function `poly`, which will create all the necessary columns for a polynomial of a specified degree:

[1] Well, hardly ever: if X_i was only ever, say, 0 or 1, then it would be each to X_i^2. Such awkward cases happen with probability 0 for continuous variables.

[2] Again, you can contrive awkward cases where this is not true, if you really want to. For instance, if X_1 and X_2 are horizontal and vertical coordinates of points laid out on a circle, they are linearly independent of each other and of their own squares, but X_1^2 and X_2^2 are linearly dependent. (Why?) The linear dependence would be broken if the points were laid out in an ellipse or oval, however. (Why?)

```
lm(y ~ poly(x,2), data=df)
```

Here the second argument, `degree`, tells `poly` what order of polynomial to use. R remembers how this works when the estimated model is used in `predict`. My *advice* is to use `poly`, but the other forms aren't wrong.

Small demo Here is a small demo of polynomial regression, using the data from the first data analysis project.

```
# Load the data
mobility <- read.csv("http://www.stat.cmu.edu/~cshalizi/mreg/15/dap/1/mobility.csv")
mob.quad <- lm(Mobility ~ Commute + poly(Latitude,2)+Longitude, data=mobility)
```

This fits a quadratic in the `Latitude` variable, but linear terms for the other two predictors. You will notice that `summary` does nothing strange here:

```
summary(mob.quad)
```

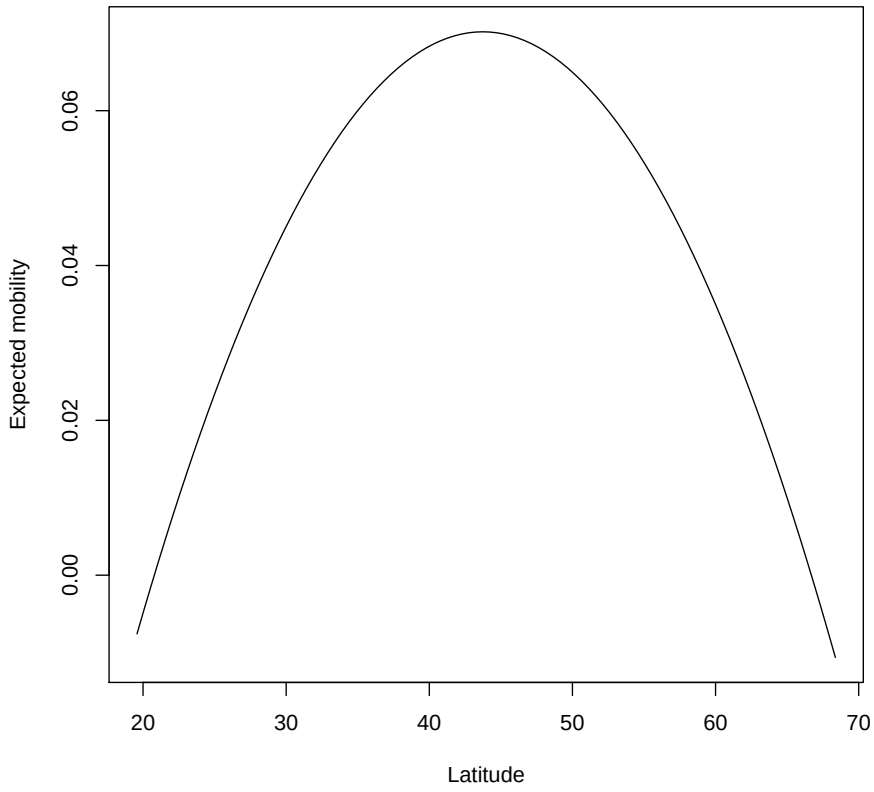
```
##
## Call:
## lm(formula = Mobility ~ Commute + poly(Latitude, 2) + Longitude,
##     data = mobility)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.12828 -0.02384 -0.00691  0.01722  0.32190
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)        -0.0261223  0.0121233  -2.155   0.0315
## Commute             0.1898429  0.0137167  13.840  < 2e-16
## poly(Latitude, 2)1  0.1209235  0.0475524   2.543   0.0112
## poly(Latitude, 2)2 -0.2596006  0.0484131  -5.362 1.11e-07
## Longitude          -0.0004245  0.0001394  -3.046   0.0024
##
## Residual standard error: 0.04148 on 724 degrees of freedom
## Multiple R-squared:  0.3828,Adjusted R-squared:  0.3794
## F-statistic: 112.3 on 4 and 724 DF,  p-value: < 2.2e-16
```

and we can use `predict` as usual:

```
predict(mob.quad, newdata=data.frame(Commute=0.298,
                      Latitude=40.57, Longitude=-79.58))
```

```
##          1
## 0.07079416
```

See also Figure 1 for an illustration that this really is giving us behavior which is non-linear in the `Latitude` variable.




```
hypothetical.pghs <- data.frame(Commute=0.287,
                                Latitude=seq(from=min(mobility$Latitude),
                                  to=max(mobility$Latitude), length.out=100),
                                Longitude=-79.92)
plot(hypothetical.pghs$Latitude, predict(mob.quad, newdata=hypothetical.pghs),
     xlab="Latitude", ylab="Expected mobility", type="l")
```

FIGURE 1: *Predicted rates of economic mobility for hypothetical communities at the same longitude as Pittsburgh, and with the same proportion of workers with short commutes, but different latitudes.*

2.2 Properties, Issues, and Caveats

Diagnostic plots The appropriate diagnostic plot is of residuals against the predictor. There is no need to make separate plots of residuals against each power of the predictor.

Smoothness Polynomial functions vary continuously in all their arguments. In fact, they are "smooth" in the sense in which mathematicians use that word, meaning that all their derivatives exist and are continuous, too. This is desirable if you think the real regression function you're trying to model is smooth, but not if you think there are sharp thresholds or jumps. Polynomials *can* approximate thresholds arbitrarily closely, but you end up needing a very high order polynomial.

Interpretation In a linear model, we were able to offer simple interpretations of the coefficients, in terms of slopes of the regression surface.

In the multiple linear regression model, we could say

$$\beta_i = \mathbb{E}\left[Y|X_i = x_i + 1, X_{-i} = x_{-i}\right] - \mathbb{E}\left[Y|X_i = x_i, X_{-i} = x_{-i}\right]$$

("β_i is the difference in the expected response when X_i is increased by one unit, all other predictor variables being equal"), or

$$\beta_i = \frac{\mathbb{E}\left[Y|X_i = x_i + h, X_{-i} = x_{-i}\right] - \mathbb{E}\left[Y|X_i = x_i, X_{-i} = x_{-i}\right]}{h}$$

("β_i is the slope of the expected response as X_i is varied, all other predictor variables being equal"), or

$$\beta_i = \frac{\partial \mathbb{E}\left[Y|X = x\right]}{\partial x_i}$$

("β_i is the rate of change in the expected response as X_i varies"). None of these statements is true any more in a polynomial regression.

Take them in reverse order. The rate of change in $\mathbb{E}\left[Y|X\right]$ when we vary X_i is now

$$\frac{\partial \mathbb{E}\left[Y|X = x\right]}{\partial x_i} = \sum_{j=1}^{d} j\beta_{i,j} x_i^{j-1}$$

This not only involves all the coefficients for all the powers of X_i, but also has a different answer at different points x_i. The linear coefficient on X_i, $\beta_{i,1}$, is the rate of change when $X_i = 0$, but not otherwise. There just is no one answer to "what's the rate of change?".

Similarly, if we ask for the slope,

$$\frac{\mathbb{E}\left[Y|X_i = x_i + h, X_{-i} = x_{-i}\right] - \mathbb{E}\left[Y|X_i = x_i, X_{-i} = x_{-i}\right]}{h}$$

that isn't given by one single number either; it depends on the starting value x_i and the size of the change h. If h is very close to very, the slope will be approximately $h\sum_{j=1}^{d} j\beta_{i,j}x_i^{j-1}$, but not, generally, otherwise. If you really want to know, you have to actually plug in to the polynomial.

Finally, the change associated with a one-unit change in X_i is just a special case of the slope when $h = 1$, and so not equal to any of the coefficients either. It will definitely change as the starting point x_i changes.

Rather than trying to give one single rate of change (or slope or response-associated-to-a-one-unit-change) when none exists, a more honest procedure is to make a plot, either of the polynomial itself, or of the derivative. (See the example in the model report for the first DAP.)

Interpreting the polynomial as a transformation of X_i If you really wanted to, you could try to complete the square (cube, other polynomial) to re-write the polynomial

$$\beta_1 X_1 + \beta_2 X_1^2 + \ldots \beta_d X_1^d = k + \beta_d \prod_{j=1}^{d} (X_1 - c_j)$$

You could then say that β_d was the change in the response for a one-unit change in $\prod_{j=1}^{d}(X_1 - c_j)$, etc., etc. The zeroes or roots of the polynomial, c_j, will be functions of the coefficients on the lower powers of X_1, but their sampling distributions, unlike those of the β_j, would be very tricky, and so, consequently, would their confidence sets. Moreover, it is not very common for the transformed predictor $\prod_{j=1}^{d}(X_1 - c_j)$ to itself be a particularly interpretable variable, so this is often a considerable amount of work for little gain.

"Testing for nonlinearity" It is not uncommon to see people claiming to test whether the relationship between Y and X_i is linear by adding a quadratic term in X_i and testing whether the coefficient on it significantly different from zero. This would work fine if you knew that the only possible sort of nonlinearity was quadratic — that if the relationship wasn't a straight line, it was a parabola. Since it is perfectly possible to have a very nonlinear relationship where the coefficient on X_i^2 is zero, this is not a very powerful test.

Over-fitting and wiggliness A polynomial of degree d can exactly fit any d points. (Any two points lie on a line, any three on a parabola, etc.) Using a high-order polynomial, or even summing a large number of low-order polynomials, can therefore lead to curves which come very close to the data we used to estimate them, but predict very badly. In particular, high-order polynomials can display very wild oscillations in between the data points. Plotting the function in between the data points (using `predict`) is a good way of noting this. We will also look at more formal checks when we cover cross-validation later in the course.

Picking the polynomial order The best way to pick the polynomial order is on the basis of some actual scientific theory which says that the relationship between Y and X_i should, indeed, by a polynomial of order d_i. Failing that, carefully examining the diagnostic plots is your next best bet. Finally, the methods we'll talk about for variable and model selection in forthcoming lectures can also be applied to picking the order of a polynomial, though as we will see, you need to be very careful about what those methods actually do, and whether that's really what you want.

2.3 Orthogonal Polynomials

I have written out polynomial regression above in its most readily-comprehended way, but that is not always the most best way to estimate it. We know, from our previous examination of multiple linear regression, that we'll get smaller standard errors when our predictor variables are uncorrelated. While X_i and its higher powers are linearly independent, they are generally (for most distributions) somewhat correlated. An alternative to regressing on the powers of X_i is to regress on linear function of X_i, a quadratic function of X_i, a cubic, etc., which are chosen so that they are *un*-correlated on the data. These functions, being uncorrelated, are called **orthogonal**. Any polynomial could also be expressed as a linear combination of these **basis functions**, which are thus called **orthogonal polynomials**. The advantage, again, is that the estimates of coefficients on these basis functions have less variance than using the powers of X_i.

In fact, this is what the `poly` function does by default; to force it to use the powers of X_i, we need to set the `raw` option to `TRUE`.

To be concrete, let's start with the linear function. We'll arrange it so that it has mean zero (and therefore doesn't contribute to the intercept):

$$\sum_{i=1}^{n} \alpha_{i10} + \alpha_{i11} x_{i1} = 0$$

Here I am using α_{ijk} to indicate the coefficient on X_i^k in the j^{th} order basis function for X_i. This is one equation with two unknowns, so we need another equation to be able to solve the system. What `poly` does is to impose a constraint on the sample variance:

$$\sum_{i=1}^{n} (\alpha_{i10} + \alpha_{i11} x_{i1})^2 = 1$$

(Why is this a constraint on the variance?) The quadratic function is found by requiring that it have mean zero,

$$\sum_{i=1}^{n} \alpha_{i20} + \alpha_{i21} x_{i1} + \alpha_{i22} x_{i1}^2 = 0 \ ,$$

that it be uncorrelated with the linear function,

$$\sum_{i=1}^{n} (\alpha_{i10} + \alpha_{i11} x_{i1}) \left(\alpha_{i20} + \alpha_{i21} x_{i1} + \alpha_{i22} x_{i1}^2\right) = 0 \ ,$$

and that it have the same variance as the linear function:

$$\sum_{i=1}^{n} \left(\alpha_{i20} + \alpha_{i21} x_{i1} + \alpha_{i22} x_{i1}^2\right)^2 = 1$$

To get the j^{th} basis function, we need all the $j-1$ basis functions that came before it, so we can make sure it has mean 0, that it's uncorrelated with all of the others, and that it has the same variance. All of the coefficients I've written α are encoded in the attributes of the output of `poly`, though not always in an especially humanly-readable way. (For details, see `help(poly)`, and the references it cites.)

Notice that changing the sample values of X_i will change the basis functions; one reason to use the powers of X_i instead would be to make it easier to compare coefficients across data sets. If the distribution of X_i is known, one can work out systems of orthogonal polynomials in advance, for instance, the Legendre polynomials which are orthogonal when the predictor variable has a uniform distribution[3].

2.4 Non-Polynomial Function Bases

There are basically three reasons to want to use polynomials. First, many scientific theories claim that there are polynomial relationships between variables in the real world. Second, they're things we've all been familiar with since basic algebra, so we understand them very well, we find them un-intimidating, and very little math is required to use them. Third, they have the nice property that any well-behaved function can be approximated arbitrarily closely by a polynomial of sufficiently high degree[4].

If we don't have strong scientific reasons to want to use polynomials, and are willing to go beyond basic algebra, there are many other systems of functions which also have the universal approximation property. If we're just doing curve fitting, it can be just as good, and sometimes much better, to use one of these other function bases. For instance, we might use sines and cosines at multiples of a basic frequency ω,

$$\sum_{j=1}^{d} \gamma_{i1j} \sin(j\omega X_i) + \gamma_{i2j} \cos(j\omega X_i)$$

Such a basis would be especially appropriate for variables which are really angles, or when there is a periodicity in the system. Exactly matching a sum of sines

[3] See, for instance, Wikipedia, s.v. "Legendre polynomials".
[4] See further reading, below, for details.

and cosines like the above would require an infinite-order polynomial; conversely, matching a linear function with a sum of sines and cosines would require letting $d \to \infty$.

As this suggests, there is a bit of an art to picking a suitable function basis; as it also suggests, it's an area where knowledge of more advanced mathematics (specifically, functional analysis) can be really useful to actually doing statistics.

3 Categorical Predictors

We often have variables which we think are related to Y which are not real numbers, but are qualitative rather than quantitative — answers to "what kind?" rather than to "how much?". For people, these might be things like sex, gender, race, caste, religious affiliation, education attainment, occupation, whether they've had chicken pox, whether they have previously defaulted on a loan, or their country of citizenship. For geographic communities (as in the data analysis project), state was a categorical variable, though not one we used because we didn't know how.

Some of these are purely qualitative, coming in distinct types, but with no sort of order or ranking implied; these are often specifically called "categorical", and the distinct values "categories". (The values are also called "levels", though that's not a good metaphor without an order.) Other have distinct levels which can be put in a sensible order, but there is no real sense that the *distance* between one level and the next is the same — they are **ordinal** but not **metric**. When it is necessary to distinguish non-ordinal categorical variables, they are often called **nominal**, to indicate that their values have names but no order.

In R, categorical variables are represented by a special data type called `factor`, which has as a sub-type for ordinal variables the data type `ordered`.

In this section, we'll see how to include both categorical and ordinal variables in multiple linear regression models, by **coding** them as numerical variables, which we know how to handle.

3.1 Binary Categories

The simplest case is that of a binary variable B, one which comes in two qualitatively different types. To represent this in a format which fits with the regression model, we pick one of the two levels or categories as the "reference" or "baseline" category. We then add a column X_B to the design matrix $\mathbf{x}$ which indicates, for each data point, whether it belongs to the reference category ($X_B = 0$) or to the other category ($X_B = 1$). This is called an **indicator variable** or **dummy variable**. That is, we **code** the qualitative categories as 0 and 1.

We then regress on the indicator variable, along with all of the others, getting the model

$$Y = \beta_0 + \beta_B X_b + \beta_1 X_1 + \ldots \beta_p X_p + \epsilon$$

The coefficient β_b is the expected difference in Y between two units which are identical, except that one of them has $X_b = 0$ and the other has $X_b = 1$. That is, it's the expected difference in the response between members of the reference category and members of the other category, all else being equal. For this reason, β_B is often called the **contrast** between the two classes.

Geometrically, if we plot the expected value of Y against $X_1, \ldots X_p$, we will now get *two* regression surfaces: they will be parallel to each other, and offset by β_B. We thus have a model where each category gets its own intercept: β_0 for the reference class, $\beta_0 + \beta_B$ for the other class. You should, at this point,

convince yourself that if we had switched which class was the reference class, we'd get exactly the same slopes, only with the over-all intercept being $\beta_0 + \beta_B$ and the contrast being $-\beta_B$ (Exercise 1).

In R If a data frame has a column which is a two-valued factor already, and it's included in the right-hand side of the regression formula, `lm` handles creating the column of indicator variables internally.

Here, for instance, we use a classic data set to regress the weight of a cat's heart on its body weight and its sex. (If it worked, such a model would be useful in gauging doses of veterinary heart medicines.)

```
library(MASS)
data(cats)
Hwt.lm <- lm(Hwt ~ Sex+Bwt, data=cats)
summary(Hwt.lm)
```

```
##
## Call:
## lm(formula = Hwt ~ Sex + Bwt, data = cats)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.5833 -0.9700 -0.0948  1.0432  5.1016
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.4149     0.7273  -0.571    0.569
## SexM         -0.0821     0.3040  -0.270    0.788
## Bwt           4.0758     0.2948  13.826   <2e-16
##
## Residual standard error: 1.457 on 141 degrees of freedom
## Multiple R-squared:  0.6468,Adjusted R-squared:  0.6418
## F-statistic: 129.1 on 2 and 141 DF,  p-value: < 2.2e-16
```

`Sex` is coded as `F` and `M`, and R's output indicates that it chose `F` as the reference category.

Diagnostics The mean of the residuals within each category is guaranteed to be zero (Exercise 2), but they should also have the same variance and otherwise the same distribution, so there is still some point in plotting residuals against X_B. Sometimes a little jitter on the horizontal axis helps, or making a box-plot.

Inference There is absolutely nothing special about the inferential statistics for the estimated contrast $\hat{\beta}_B$. It works just like inference for any other regression coefficient.

Why not just split the data? If we want to give each class its own intercept, why not just split the data and estimate two models, one for each class? The answer is that sometimes we'll do just this, especially if there's a lot of data for each class. However, if the regression surfaces for the two categories really are parallel to each other, by splitting the data we're losing some precision in our estimate of the common slopes, without gaining anything. In fact, if the two surfaces are *nearly* parallel, for moderate sample sizes the small bias that comes from pretending the slopes are all equal can be overwhelmed by the reduction in variance.

Why not two columns? It's natural to wonder why we have to pick out one level as the reference, and estimate a contrast. Why not add *two* columns to $\mathbf{x}$, one indicating each class? The problem is that then those two columns will be linearly dependent (they'll always add up to one), so the data would be collinear and the model in-estimable.

Why not two slopes? The model we've specified has two parallel regression surfaces, with the same slopes but different intercepts. We could also have a model with the same intercept across categories, but different slopes for each variable. Geometrically, this would mean that the regression surfaces weren't parallel, but would meet at the origin (and elsewhere). We'll see how to make that work when we deal with interactions in a few lectures. If we wanted different slopes and intercepts, we might as well just split the data.

Contrasts need contrasts Just as we can't estimate β_i if $\mathrm{Var}[X_i] = 0$, we can't estimate any categorical contrasts if all the data points belong to the same category.

3.1.1 "Adjusted effect of a category"

As I said, β_B is the expected difference in Y between two individuals which have the same value for all of the variables *except* the category. This is generally *not* the same as the difference in expectations between the two categories:

$$\beta_B \neq \mathbb{E}[Y|X_B = 1] - \mathbb{E}[Y|X_B = 0]$$

One of the few situations where $\beta_B = \mathbb{E}[Y|X_B = 1] - \mathbb{E}[Y|X_B = 0]$ is when the distribution of all the *other* variables is the same between the categories. (Said another way, the categories are statistically independent of the other predictors.) Another is when there are no other predictors.

Because of this, it's very natural to want to interpret β_B as the difference in the response between the two groups, *adjusting for* all of the other variables. It's even common to talk about β_B as "the adjusted effect" of the category. As you might imagine, such interpretations come up all the time in disputes about discrimination.

Even leaving aside the emotional charge of such arguments, it is wise to be cautious about such interpretations, for several reasons.

1. The regression is only properly adjusting for all of the other variables if it's well-specified. If it's not, the contrast between the categories will also pick up some of the average difference in bias (due to getting the model wrong), which is not relevant.

2. As usual, finding that the contrast coefficient isn't significant doesn't necessarily mean there is no contrast! It means that the contrast, if there is one, can't be reliably distinguished from 0, which could be because it's very small or because we can't estimate it well. Again as usual, a confidence interval is called for.

3. It's not clear that we always *do* want to adjust for other variables, even when we can measure them. For instance, if economists in Lilliput found no effect on income between those who broke their eggs at the big end and those at the little end, after adjusting for education and occupational prestige (Swift, 1726), that wouldn't necessarily settle the question of whether big-endians were discriminated against. After all, it might be that they have less access to education and high-paid jobs *because* they were big-endians. And this could be true even if Lilliputians were initially randomly assigned between big- and little- end-breaking. The same goes for finding that there *is* an "adjusted effect".

The last point brings us close to topics of causal inference, which we won't get to until 402. For now, a good rule of thumb is not to adjust for variables which might themselves be effects of the variable we're interested in.

3.2 Categorical Variables with More than Two Levels

Suppose our categorical variable C has more than two levels, say k of them. We can handle it in almost exactly the same way as the binary case. We pick one level — it really doesn't matter which — as the reference level. We then introduce $k-1$ columns into the design matrix $\mathbf{x}$, which are indicators for the other categories. If, for instance, $k = 3$ and the classes are `North`, `South`, `West`, we pick one level, say `North`, as the reference, and then add a column $X_{\texttt{South}}$ which is 1 for data points in class `South` and 0 otherwise, and another column $X_{\texttt{West}}$ which is 1 for data points in that class and 0 otherwise.

Having added these columns to the design matrix, we regress as usual, and get $k-1$ contrasts. The over-all β_0 is really the intercept for the reference class; the contrasts are added to β_0 to get the intercept for each class. Geometrically, we now have k parallel regression surfaces, one for each level of the variable.

Interpretation $\beta_{C=c}$ is the expected difference between two individuals who are otherwise identical, except that one is in the reference category and the other is in class c. The expected difference between two otherwise-identical individuals in two different categories, say c and d, is therefore the difference in their contrasts, $\beta_{C=d} - \beta_{C=c}$.

Diagnostics and inference Work just the same as in the binary case.

Why not k columns? Because, just like in the binary case, that would make all those columns for that variable sum to 1, causing problems with collinearity.

Contrasts need contrast If we know there are k categories, but some of them don't appear in our data, we can't estimate their contrasts.

Category-specific slopes and splitting the data The same remarks apply as under binary predictor variables.

3.3 Two, Three, Many Categorical Predictors

Nothing in what we did above requires that there be only one categorical predictor; the other variables in the model could be indicator variables for other categorical predictors. Nor do all the categorical predictors have to have the same number of categories. The only wrinkle with having multiple categories is that β_0, the over-all intercept, is now the intercept for individuals where *all* categorical variables are in their respective reference levels. Each combination of categories gets its own regression surface, still parallel to each other.

With multiple categories, it is natural to want to look at interactions — to let their be an intercept for left-handed little-endian plumber, rather than just adding up contrasts for being left-handed and being a little-endian and being a plumber. We'll look at that when we deal with interactions.

3.4 Analysis of Variance: Only Categorical Predictors

A model in which there are *only* categorical predictors is, for historical reasons, often called an **analysis of variance** model. Estimating such a model presents absolutely no special features beyond what we have already covered, but it's worth a paragraph or two on the interpretation and the origins of such models.

Suppose, for simplicity, that there are two categorical predictors, B and C, and the reference level for each is written $\emptyset$. The conditional expectation of Y will be pinned down by giving a level for each, say b and c, respectively. Then

$$\mathbb{E}[Y|B=b, C=c] = \beta_0 + \beta_b \delta_{b\emptyset} + \beta_c \delta_{c\emptyset}$$

That is, we add the appropriate contrast for each categorical variable, and nothing else. (This presumes no interactions, a limitation which we'll lift next week.) Conversely, if we knew $\mathbb{E}[Y|B=b, C=c]$ for every category, we could work out the contrasts without having to ever (explicitly) compute $(\mathbf{x}^T\mathbf{x})^{-1}\mathbf{x}^T\mathbf{y}$, which was a very real consideration before computation became so cheap[5]. Obviously, however, it is not much of an issue now.

[5] To see how, notice that $\hat{\beta}_0$ can be estimated by the sample mean of all cases where $B = \emptyset$, $C = \emptyset$. Then to get, say, β_b, we average the difference in means between cases where $B = b, C = c$ and $B = \emptyset, C = c$ for each level c of the other variable. (This averaging of differences eliminates the contribution from β_c.)

As for the name, it arises from the basic probability fact sometimes called the "law of total variance":

$$\mathrm{Var}[Y] = \mathrm{Var}[\mathbb{E}[Y|X]] + \mathbb{E}[\mathrm{Var}[Y|X]]$$

If X is our complete set of categorical variables, each of which defines a group, this says "The total variance of the response is the variance in average responses across groups, plus the average variance within a group". Thus, after estimating the contrasts, we have decomposed or analyzed the variance in Y into between-group and across-group variance. This was extremely useful in the early days of agricultural and industrial experimentation, but has frankly become a bit of a fossil, if not a fetish.

An "analysis of covariance" model is just a regression with both qualitative and quantitative predictors.

3.5 Ordinal Variables

An ordinal variable, as I said, is one where the qualitatively-distinct levels can be put in a sensible order, but there's no implication that the distance from one level to the next is constant. At our present level of sophistication, we have basically two ways to handle them:

1. Ignoring the ordering and treat them like nominal categorical variables.
2. Ignoring the fact that they're only ordinal and not metric, assign them numerical codes (say 1, 2, 3, ...) and treat them like ordinary numerical variables.

The first procedure is unbiased, but can end up dealing with a lot of distinct coefficients. It also has the drawback that if the relationship between Y and the categorical variable is monotone, that may not be respected by the coefficients we estimate. The second procedure is very easy, but usually without any substantive or logical basis. It implies that each step up in the ordinal variable will predict exactly the *same* difference in Y, and why should that be the case? If, after treating an ordinal variable like a nominal one, we get contrasts which are all (approximately) equally spaced, we might then try the second approach.

Other procedures for ordinal variables which are, perhaps, more conceptually satisfying need much more math than we're presuming here; see the further reading.

3.6 Detailed R Example

The data set for the first data analysis project included a categorical variable, `State`, which we did not use. Let's try adding it to the model.

First, let's do some basic counting and examination:

```
# How many levels does State have?
nlevels(mobility$State)
```

```
## [1] 51
```

```
# What are they?
levels(mobility$State)
```

```
##  [1] "AK" "AL" "AR" "AZ" "CA" "CO" "CT" "DC" "DE" "FL" "GA" "HI" "IA" "ID"
## [15] "IL" "IN" "KS" "KY" "LA" "MA" "MD" "ME" "MI" "MN" "MO" "MS" "MT" "NC"
## [29] "ND" "NE" "NH" "NJ" "NM" "NV" "NY" "OH" "OK" "OR" "PA" "RI" "SC" "SD"
## [43] "TN" "TX" "UT" "VA" "VT" "WA" "WI" "WV" "WY"
```

There are 51 levels for `State`, as there should be, corresponding to the 50 states and the District of Columbia. We see that these are given by the two-letter postal codes, in alphabetical order.

Running a model with `State` and `Commute` as the predictors, we therefore expect to get 52 coefficients (1 intercept, 1 slope, and 51-1 = 50 contrasts). R will calculate contrasts from the first level, which here is `AK`, or Alaska.

```
mob.state <- lm(Mobility ~ Commute + State, data=mobility)
signif(coefficients(mob.state),3)
```

```
## (Intercept)     Commute     StateAL     StateAR     StateAZ     StateCA
##    0.018400    0.126000   -0.005600    0.001840    0.007290    0.031500
##     StateCO     StateCT     StateDC     StateDE     StateFL     StateGA
##    0.044100    0.021500    0.071300    0.007460    0.004160   -0.022000
##     StateHI     StateIA     StateID     StateIL     StateIN     StateKS
##    0.029100    0.052200    0.029700    0.013200    0.010000    0.042400
##     StateKY     StateLA     StateMA     StateMD     StateME     StateMI
##    0.011900    0.021100    0.001230    0.018700    0.004710    0.005230
##     StateMN     StateMO     StateMS     StateMT     StateNC     StateND
##    0.055300    0.011900   -0.018700    0.045200   -0.011400    0.146000
##     StateNE     StateNH     StateNJ     StateNM     StateNV     StateNY
##    0.060400    0.032200    0.062700    0.006670    0.045400    0.022300
##     StateOH     StateOK     StateOR     StatePA     StateRI     StateSC
##   -0.000559    0.036000    0.013800    0.035000    0.022400   -0.019300
##     StateSD     StateTN     StateTX     StateUT     StateVA     StateVT
##    0.042300    0.000761    0.032200    0.060500    0.014100    0.017300
##     StateWA     StateWI     StateWV     StateWY
##    0.025800    0.031700    0.057800    0.061200
```

In the interest of space, I won't run `summary` on this, but you can. You will find that quite a few of the contrasts are statistically significant. We'd expect about $50 \times 0.05 = 2.5$ to be significant at the 5% level, even if all the true contrasts were zero, but many more them are than this baseline. As usual, of course, it doesn't mean the model is right; it just means that if we were going to put in an intercept, a slope for `Commute`, and a contrast for every other state, we should really put in contrasts for those states as well.

```
# Set up a function to make maps
  # "Terrain" color levels set based on quantiles of the variable being plotted
# Inputs: vector to be mapped over the data frame; number of levels to
  # use for colors; other plotting arguments
# Outputs: invisibly, list giving cut-points and the level each observation
  # was assigned
mapper <- function(z, levels, ...) {
    # Which quantiles do we need?
    probs <- seq(from=0, to=1, length.out=(levels+1))
    # What are those quantiles?
    z.quantiles <- quantile(z, probs)
    # Assign each observation to its quantile
    z.categories <- cut(z, z.quantiles, include.lowest=TRUE)
    # Make up a color scale
    shades <- terrain.colors(levels)
    plot(x=mobility$Longitude, y=mobility$Latitude,
         col=shades[z.categories], ...)
    invisible(list(quantiles=z.quantiles, categories=z.categories))
}
```

FIGURE 2: *Function for making maps, from the model DAP 1.*

One issue with the simple linear regression from the DAP was that its residuals were very strongly correlated spatially. We might hope that adding all these state-by-state contrasts has gotten rid of some of that correlation.

When we have a large number of categories, it's often tempting to try compressing them to a smaller number, by grouping together some of the levels. If we do this right, we reduce the variance in our estimates of the coefficients, while introducing little (if any) bias.

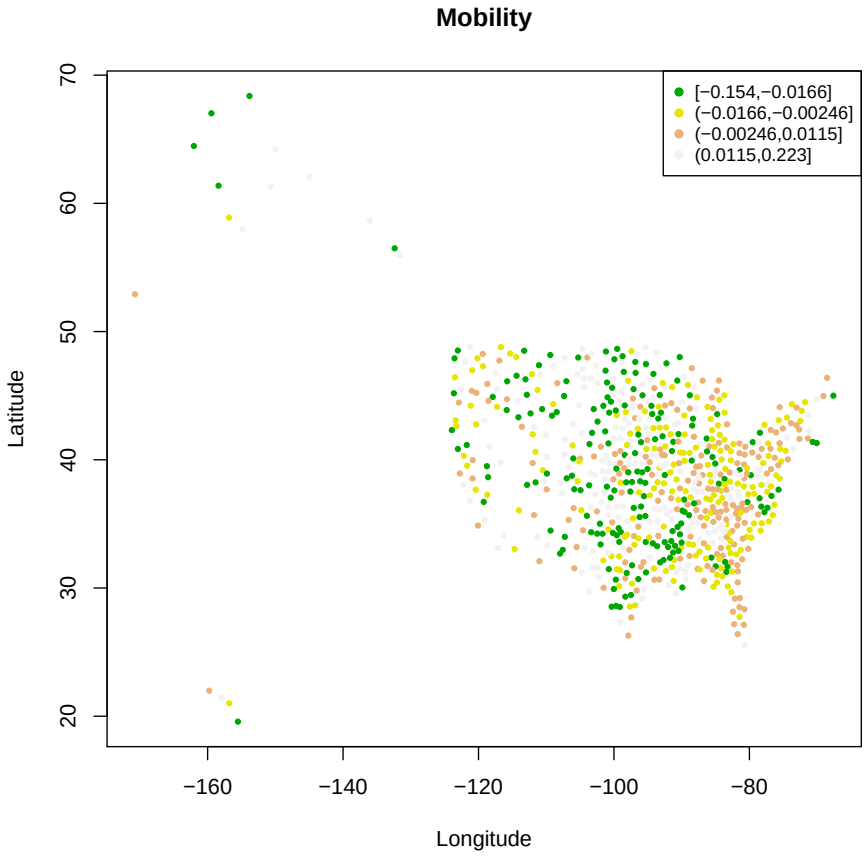
To illustrate this, let's try boiling down the 51 states (and DC) into two categories: the South versus the rest of the country. The South has long been quite distinct from the rest of the country culturally, politically and economically, in ways which are, plausibly, very relevant to economic mobility. More relevantly, when we looked at the residuals in the DAP, there was a big cluster of negative residuals in the southeastern part of the map. To make this concrete, I'll define the South as consisting of those states which joined the Confederacy during the Civil War (Alabama, Arkansas, Florida, Georgia, Louisiana, Mississippi, North Carolina, South Carolina, Tennessee, Texas and Virginia).

Let's start by adding the relevant column to the data frame:

```
# The states of the Confederacy
Confederacy <- c("AR", "AL", "FL", "GA", "LA", "MS", "NC", "SC", "TN", "TX", "VA")
mobility$Dixie <- mobility$State %in% Confederacy
```

The new `Dixie` column of `mobility` will contain the values `TRUE`, for each community located in one of those states, and `FALSE`, for the rest. R will in such circumstances treat `FALSE` as the reference category.

FIGURE 3: *Map of residuals from a basline linear regression of the rate of economic mobility on the fraction of workers with short commutes.*



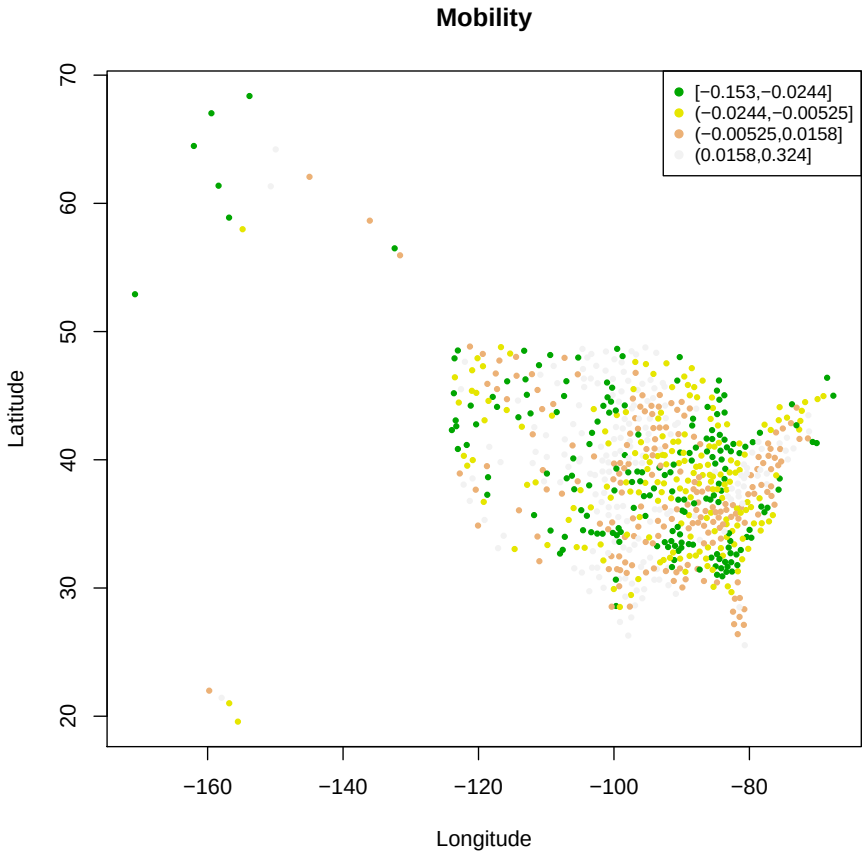

```
residuals.state.map <- mapper(residuals(mob.state), levels=4, pch=19,
                              cex=0.5, xlab="Longitude", ylab="Latitude",
                              main="Mobility")
legend("topright", legend=levels(residuals.state.map$categories), pch=19,
       col=terrain.colors(4), cex=0.8)
```

FIGURE 4: *Map of the residuals for the model with state-level contrasts. (See Figure 2 for the* ***mapper*** *function.)*

```
mob.dixie <- lm(Mobility ~ Commute + Dixie, data=mobility)
signif(coefficients(summary(mob.dixie)),3)
```

```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.0190    0.00607    3.13 1.84e-03
## Commute       0.1950    0.01180   16.50 2.94e-52
## DixieTRUE    -0.0217    0.00354   -6.14 1.37e-09
```

The contrast for the old Confederacy versus the rest of the country is negative, meaning those states have lower levels of economic mobility, and highly statistically significant. Of course, the model could still be wrong. The residuals, while better than a model with no geographic contrasts, don't look as random as in the one with contrasts for each state.




```
residuals.dixie.map <- mapper(residuals(mob.dixie), levels=4, pch=19,
                              cex=0.5, xlab="Longitude", ylab="Latitude",
                              main="Mobility")
legend("topright", legend=levels(residuals.dixie.map$categories), pch=19,
       col=terrain.colors(4), cex=0.8)
```

FIGURE 5: *Map of the residuals from the model based on* `Commute`*, and a categorical contrast between the old Confederacy and the rest of the country.*

4 Further Reading

Polynomial regression and categorical predictors are both ancient topics; I don't know who first introduced either.

The above discussion has assumed that when we use a polynomial, we use the *same* polynomial for all values of X_i. An alternative is to use different, low-order polynomials in different regions. If these piecewise polynomial functions are required to be continuous, they are called **splines**, and regression with splines will occupy us for much of 402, because it gives us ways to tackle lots of the issues with polynomials, like over-fitting (Shalizi, forthcoming, chs. 8 and 9). Personally, I have found splines to almost always be a better tool than polynomial regression, but they do demand a bit more math.

The matter of "adjusted effects" and causal inference will occupy us for about the last quarter of 36-402.

Tutz (2012) is a thorough and modern survey of regression with categorical *response* variables. We will go over this in some detail in 402, but his book covers many topics we won't have time for.

Winship and Mare (1984) proposes some interesting techniques for dealing with ordinal variables, under the (strong) assumption that they arise from taking continuous variables and breaking them into discrete categories. This seems to require rather strong assumptions about the measurement process. Another direction we could go would be to estimate a separate contrast for each level of an ordinal variable (except the lowest), but require these to be either all increasing or all decreasing, so the response to the ordinal variable was monotone. This would mean solving a *constrained* least squares (or maximum likelihood) problem to get the estimates, not an unconstrained on. Worse, the constraints would be a somewhat awkward set of inequalities. Still, it's do-able in principle, though I don't know of a straightforward R implementation.

Analysis of variance models were introduced by R. A. Fisher, probably the greatest statistician who ever lived, in connection with problems in genetics and in designing and interpreting experiments. They have given rise to a huge literature and an elaborate system of notation and terminology, much of which boils down to short-cuts for computing regression estimates when the design matrix $\mathbf{x}$ has very special structure. As I said, there were many decades when such short-cuts were vital, but I am frankly skeptical how much value these techniques retain in the present day. In the interest of balance, see Gelman (2005) for a contrary view.

I mentioned that one reason to use polynomials is that any well-behaved function can be approximated arbitrarily closely by polynomials of sufficiently high degree. Obviously "well-behaved" needs a proper definition, as (perhaps less obviously) does "approximated arbitrarily closely". What I had in mind was the Stone-Weierstrass theorem, which states that you can pick any continuous function f, interval $[a, b]$, and tolerance $\epsilon > 0$ you like, and I can find some

polynomial which is within ϵ of f everywhere on the interval,

$$\max_{a \leq x \leq b} \left| f(x) - \sum_{j=1}^{d} \gamma_j x^j \right| \leq \epsilon$$

provided there is no limit on the order d or the magnitude of the coefficients γ_j. This is a standard result of real analysis, which will be found in almost textbook on that subject, or on functional analysis or approximation theory. There are parallel results for other function bases.

5 Exercises

To think through or practice on, not to hand in.

1. Consider regressing Y on a binary categorical variable B, plus some other predictors. Suppose we switch which level is the reference category and which one is contrasted with it. Show that this produces the following changes to the parameters, and leaves all the others unchanged:

$$\beta_0 \rightarrow \beta_0 + \beta_B \quad (1)$$
$$\beta_B \rightarrow -\beta_B \quad (2)$$

   Hint: Show that the change to the indicator variable is $X_B \rightarrow 1 - X_B$.

2. Consider again regressing Y on a binary variable B, plus some other predictors, and estimating all coefficients by least squares. Show that the average of all residuals where $X_B = 1$ must be exactly 0, as must the average of all residuals where $X_B = 0$. *Hint:* Use the estimating equations to show $\sum_i e_i = 0$, $\sum_i e_i x_{Bi} = 0$, and algebra to show $\sum_i e_i(1 - x_{Bi}) = 0$.

References

Gelman, Andrew (2005). "Analysis of Variance — Why It Is More Important than Ever." *Annals of Statistics*, **33**: 1–53. URL `http://projecteuclid.org/euclid.aos/1112967698`. doi:10.1214/009053604000001048.

Shalizi, Cosma Rohilla (forthcoming). *Advanced Data Analysis from an Elemntary Point of View*. Cambridge, England: Cambridge University Press. URL `http://www.stat.cmu/~cshalizi/ADAfaEPoV`.

Swift, Jonathan (1726). *Gulliver's Travels*. London: Benjamin Motte. URL `http://www.gutenberg.org/ebooks/829`. Originally published as *Travels into Several Remote Nations of the World. In Four Parts. By Lemuel Gulliver, First a Surgeon, and then a Captain of several Ships.*

Tutz, Gerhard (2012). *Regression for Categorical Data*. Cambridge, England: Cambridge University Press.

Winship, Christopher and Robert D. Mare (1984). "Regression Models with Ordinal Variables." *American Sociological Review*, **49**: 512–525. URL `http://scholar.harvard.edu/files/cwinship/files/asr_1984.pdf`.