Lecture 14: Multiple Linear Regression

36-401, Section B, Fall 2015

15 October 2015

Contents

1 Recap on Simple Linear Regression in Matrix Form

Let's start with a brief summary of re-doing simple linear regression with matrices. We collect all our observations of the response variable into a vector, which we write as an $n \times 1$ matrix $\mathbf{y}$, one row per data point. We group the two coefficients into a 2×1 matrix β. We create an $n \times 2$ matrix $\mathbf{x}$, where the first column

consists entirely of 1s, and the second column contains all our observations of the predictor variable, again, one row per data point. Our point predictions are then given by $\mathbf{x}\beta$, and the mean squared error by $n^{-1}(\mathbf{y} - \mathbf{x}\beta)^T(\mathbf{y} - \mathbf{x}\beta)$.

The derivative of the MSE with respect to β is

$$\frac{2}{n}(-\mathbf{x}^T\mathbf{y} + \mathbf{x}^T\mathbf{x}\beta) \tag{1}$$

Setting this to zero at the optimum coefficient vector $\widehat{\beta}$ gives the (matrix) estimating equation

$$-\mathbf{x}^T\mathbf{y} + \mathbf{x}^T\mathbf{x}\widehat{\beta} = 0 \tag{2}$$

whose solution is of course

$$\widehat{\beta} = (\mathbf{x}^T\mathbf{x})^{-1}\mathbf{x}^T\mathbf{y} \tag{3}$$

We verified last time that $\widehat{\beta}$ does, in fact, coincide with what we already know the least squares solutions to be. Before, we had two estimating equations for two unknowns ($\hat{\beta}_0$ and $\hat{\beta}_1$), and we had to keep track of how they related to each other and how to solve either one. The matrix inversion and multiplication in Eq. 3 encapsulates all of that book-keeping.

We also saw that the fitted values at the data points used to estimate the model are linear combinations of the observed responses, with weights given by the **hat** or **influence** matrix:

$$\widehat{\mathbf{m}} = \mathbf{x}(\mathbf{x}^T\mathbf{x})^{-1}\mathbf{x}^T\mathbf{y} = \mathbf{H}\mathbf{y} \tag{4}$$

Geometrically, this means that we find the fitted values by taking the vector of observed responses $\mathbf{y}$ and projecting it on to a certain plane, which is entirely defined by the values in $\mathbf{x}$.

2 Multiple Linear Regression

We are now ready to go from the *simple* linear regression model, with one predictor variable, to em multiple linear regression models, with more than one predictor variable[1]. Let's start by presenting the statistical model, and get to estimating it in just a moment.

2.1 The Statistical Model, without Assuming Gaussian Noise

In the basic form of the **multiple linear regression model**,

[1] You might wonder why the jargon here contrasts "simple" with "multiple", rather than with "complex". The reason is that the older sense of "simple" is "having only one part" or "made from just one ingredient".

1. There are p quantitative predictor variables, $X_1, X_2, \ldots X_p$. We make no assumptions about their distribution; in particular, they may or may not be dependent. X without a subscript will refer to the vector of all of these taken together.
2. There is a single response variable Y.
3. $Y = \beta_0 + \sum_{i=1}^{p} \beta_i X_i + \epsilon$, for some constants (coefficients) $\beta_0, \beta_1, \ldots \beta_p$.
4. The noise variable ϵ has $\mathbb{E}[\epsilon|X = x] = 0$ (mean zero), $\mathrm{Var}[\epsilon|X = x] = \sigma^2$ (constant variance), and is uncorrelated across observations.

In matrix form, when we have n observations,

$$\mathbf{Y} = \mathbf{X}\beta + \epsilon \tag{5}$$

where $\mathbf{X}$ is a $n \times (p+1)$ matrix of random variables (including an all-and-always 1 first column), and ϵ is an $n \times 1$ matrix of noise variables. By the modeling assumptions, $\mathbb{E}[\epsilon|\mathbf{X}] = 0$ while $\mathrm{Var}[\epsilon|\mathbf{X}] = \sigma^2\mathbf{I}$.

2.2 The Statistical Model, Assuming Gaussian Noise

In the **multiple linear regression model with Gaussian noise**,

1. There are p quantitative predictor variables, $X_1, X_2, \ldots X_p$. We make no assumptions about their distribution; in particular, they may or may not be dependent. X without a subscript will refer to the vector of all of these taken together.
2. There is a single response variable Y.
3. The variables are related through

$$Y = \beta_0 + \sum_{i=1}^{p} \beta_i X_i + \epsilon \ , \tag{6}$$

   for some constants (coefficients) $\beta_0, \beta_1, \ldots \beta_p$.
4. The noise variables ϵ have a jointly-Gaussian $MVN(\mathbf{0}, \sigma^2\mathbf{I})$ distribution, independent of $\mathbf{X}$.

From these assumptions, it follows that, conditional on $\mathbf{X}$, $\mathbf{Y}$ has a multivariate Gaussian distribution,

$$\mathbf{Y}|\mathbf{X} \sim MVN(\mathbf{X}\beta, \sigma^2\mathbf{I}) \tag{7}$$

2.3 Parameter Interpretation

β_0 is the expected value of Y are the origin:

$$\beta_0 = \mathbb{E}\left[Y|X_1 = 0, X_2 = 0, \ldots X_p = 0\right] \tag{8}$$

The multiple linear regression model assumes that each predictor variable makes a separate contribution to the expected response, that these contributions add up without any interaction, and that each predictor's contribution is linear[2]. Thus β_i is the rate at which $\mathbb{E}[Y]$ changes as X_i, and *only* X_i, changes, regardless of where X_i starts (linearity in X_i), and regardless of what any of the other variables might be (additivity across variables).

3 Derivation of the Least Squares Estimator

We now wish to estimate the model by least squares. Fortunately, we did essentially all of the necessary work last time.

This is because the formula we derived for the mean squared error,

$$\frac{1}{n}(\mathbf{y} - \mathbf{x}\beta)^T(\mathbf{y} - \mathbf{x}\beta) \tag{9}$$

did not actually care whether $\mathbf{x}$ was $n \times 2$ or $n \times (p+1)$ for any larger p, so long as β was $(p+1) \times 1$. Neither did any of the matrix calculus we did, so it remains true that

$$\nabla_\beta MSE(\beta) = \frac{2}{n}\left(-\mathbf{x}^T\mathbf{y} + \mathbf{x}^T\mathbf{x}\beta\right) \ ; \tag{10}$$

that the estimating equation is

$$-\mathbf{x}^T\mathbf{y} + \mathbf{x}^T\mathbf{x}\widehat{\beta} = 0 \tag{11}$$

and that the solution, the **ordinary least squares** (OLS) estimator, is

$$\widehat{\beta} = (\mathbf{x}^T\mathbf{x})^{-1}\mathbf{x}^T\mathbf{y} \tag{12}$$

Eq. 12 is going to keep coming up again and again; whether you memorize it deliberately or through sheer exposure is up to you.

(We didn't *have* to use matrix notation to arrive at this point. In principle, we could have written out the MSE as an explicit sum over data points, and then taken $p+1$ partial derivatives with respect to the $p+1$ coefficients. This would have led to a system of $p+1$ linear equations in $p+1$ unknowns, which we could then try to solve. But all of this machinery is conveniently assembled into the linear algebra, which makes it *much* easier to handle.)

[2] We will see some ways of allowing predictor variables to interact later in this class. The topic will be explored more fully in 402, along with additive but nonlinear models.

3.1 Slightly Alternate Derivation

To appreciate what's going on in Eq. 12, it may help to look at a *slightly* different derivation, which explicitly separates the intercept from the other coefficients. So, in this subsection, *and this sub-section only*, β_0 will be the scalar intercept, β will be a $p \times 1$ vector of slope coefficients (not $(p+1) \times 1$!), and $\mathbf{x}$ will be an $n \times p$ matrix of observations of the predictors (i.e., no column of 1s).

The mean squared error will be

$$\frac{1}{n}(\mathbf{y} - \beta_0 \mathbf{1} - \mathbf{x}\beta)^T(\mathbf{y} - \beta_0 \mathbf{1} - \mathbf{x}\beta) \tag{13}$$

where $\mathbf{1}$ is the $n \times 1$ matrix of all 1s. The relevant derivatives are

$$\frac{\partial MSE}{\partial \beta_0} = -\frac{2}{n}\mathbf{1}^T(\mathbf{y} - \beta_0 \mathbf{1} - \mathbf{x}\beta) \tag{14}$$

and

$$\nabla_\beta MSE = \frac{2}{n}(\beta_0 \mathbf{x}^T\mathbf{1} - \mathbf{x}^T\mathbf{y} + \mathbf{x}^T\mathbf{x}\beta) \tag{15}$$

Setting both derivatives to zero at the optimum, we get

$$\hat{\beta}_0 = \frac{1}{n}\mathbf{1}^T\mathbf{y} - \frac{1}{n}\mathbf{1}^T\mathbf{x}\widehat{\beta} \tag{16}$$

Notice that $n^{-1}\mathbf{1}^T\mathbf{y}$ is just our old friend $\overline{y}$. Similarly, $\frac{1}{n}\mathbf{1}^T\mathbf{x}$ is the $1 \times p$ matrix giving the sample means for each coordinate of x; lets call this $\overline{\mathbf{x}}$. Thus

$$\hat{\beta}_0 = \overline{y} - \overline{\mathbf{x}}\widehat{\beta} \tag{17}$$

and the intercept will make sure the regression surface goes through the mean of the data.

Turning to the equation for $\widehat{\beta}$,

$$0 = \hat{\beta}_0 \frac{1}{n}\mathbf{x}^T\mathbf{1} - \frac{1}{n}\mathbf{x}^T\mathbf{y} + \frac{1}{n}\mathbf{x}^T\mathbf{x}\widehat{\beta} \tag{18}$$

At this point, let's make two moves which will simplify things later. First, notice that $\hat{\beta}_0$ is a scalar, so we can move it all the way to the right of the first term we're adding, getting

$$0 = \frac{1}{n}\mathbf{x}^T\mathbf{1}\hat{\beta}_0 - \frac{1}{n}\mathbf{x}^T\mathbf{y} + \frac{1}{n}\mathbf{x}^T\mathbf{x}\widehat{\beta} \tag{19}$$

Second, notice that $n^{-1}\mathbf{x}^T\mathbf{1} = \overline{\mathbf{x}}^T$. Thus

$$0 = \overline{\mathbf{x}}^T\hat{\beta}_0 - \frac{1}{n}\mathbf{x}^T\mathbf{y} + \frac{1}{n}\mathbf{x}^T\mathbf{x}\widehat{\beta} \tag{20}$$

Now substitute in Eq. 17 for $\hat{\beta}_0$:

$$\begin{aligned} 0 &= \overline{\mathbf{x}}^T(\overline{y} - \overline{\mathbf{x}}\widehat{\beta}) - \frac{1}{n}\mathbf{x}^T\mathbf{y} + \frac{1}{n}\mathbf{x}^T\mathbf{x}\widehat{\beta} && (21)\\ 0 &= \overline{\mathbf{x}}^T\overline{y} - \overline{\mathbf{x}}^T\overline{\mathbf{x}}\widehat{\beta} - \frac{1}{n}\mathbf{x}^T\mathbf{y} + \frac{1}{n}\mathbf{x}^T\mathbf{x}\widehat{\beta} && (22)\\ \frac{1}{n}\mathbf{x}^T\mathbf{x}\widehat{\beta} - \overline{\mathbf{x}}^T\overline{\mathbf{x}}\widehat{\beta} &= -\overline{\mathbf{x}}^T\overline{y} + +\frac{1}{n}\mathbf{x}^T\mathbf{y} && (23)\\ \left(\frac{1}{n}\mathbf{x}^T\mathbf{x} - \overline{\mathbf{x}}^T\overline{\mathbf{x}}\right)\widehat{\beta} &= \frac{1}{n}\mathbf{x}^T\mathbf{y} - \overline{\mathbf{x}}^T\overline{y} && (24) \end{aligned}$$

It is straight-forward to check that (Exercise 1)

$$\frac{1}{n}\mathbf{x}^T\mathbf{y} - \overline{\mathbf{x}}^T\overline{y} \tag{25}$$

is the $p \times 1$ matrix whose i^{th} entry is the sample covariance between X_i and Y. Similarly (Exercise 2),

$$\frac{1}{n}\mathbf{x}^T\mathbf{x} - \overline{\mathbf{x}}^T\overline{\mathbf{x}} \tag{26}$$

is the $p \times p$ sample variance-covariance matrix of the X_i's. (This is why I left in the seeming-redundant factors of $1/n$.)

Let us call these two matrices, respectively, $\mathbf{c}_{X,Y}$ and $\mathbf{v}_X$. Then our equation for the vector of slopes is

$$\mathbf{v}_X\widehat{\beta} = \mathbf{c}_{X,Y} \tag{27}$$

which of course has the solution

$$\widehat{\beta} = \mathbf{v}_X^{-1}\mathbf{c}_{X,Y} \tag{28}$$

In words: we find the slopes by first finding the covariance between each predictor and the response, and then multiplying by the inverse of the predictor's covariance matrices. The intercept is just a fudge-factor to make sure the regression surface goes through the mean of the data.

Taking the $n \to \infty$ limit As the sample size grows, the law of large numbers tells us $\mathbf{v}_X \to \mathrm{Var}[X]$, the true $p \times p$ variance-covariance matrix of the predictors. Similarly, $\mathbf{c}_{\mathbf{X},Y} \to \mathrm{Cov}[X, Y]$, the $p \times 1$ matrix of covariances between the predictors and the response. Hence (by continuity)

$$\widehat{\beta} \to \mathrm{Var}[X]^{-1}\,\mathrm{Cov}[X, Y] \tag{29}$$

I leave it as an exercise (3) to show that, first, under the model assumptions, the true vector of slopes β is indeed equal to $\mathrm{Var}[X]^{-1}\,\mathrm{Cov}[X, Y]$, and, second, that this vector of slopes would minimize the *expected* squared error (not the in-sample mean squared error).

3.2 Why Multiple Regression Isn't Just a Bunch of Simple Regressions

When we do multiple regression, the slopes we get for each variable aren't the same as the ones we'd get if we just did p separate simple regressions. Why not?

The book-keeping answer In §3.1, we saw that the slopes are determined by $\mathbf{v}_X^{-1}\mathbf{c}_{X,Y}$. If $\mathbf{v}_X^{-1}$ is diagonal, then our multiple regression *will* give the same slopes as many simple regressions. In turn, $\mathbf{v}_X^{-1}$ is diagonal if and only if $\mathbf{v}_X$ is diagonal, which means that none of the predictor variables can have any (sample) correlation with any of the others. Otherwise, minimizing the mean squared error means shifting the slopes away from what they'd be in simple regressions.

(Since $\mathbf{x}$ is called the **design matrix**, a data set where $\mathbf{v}_X$ is diagonal is said to have an **orthogonal design**. As the word suggests, this is much more common in deliberately planned experiments than in observational studies.)

The predictive answer Suppose the real model is $Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \epsilon$. (Nothing turns on $p = 2$, it just keeps things short.) What would happen if we did a simple regression of Y on just X_1? We know (Lecture 1) that the optimal (population) slope on X_1 should be

$$\frac{\text{Cov}\left[X_1, Y\right]}{\text{Var}\left[X_1\right]} \tag{30}$$

Let's substitute in the model equation for Y:

$$\begin{aligned}
\frac{\text{Cov}\left[X_1, Y\right]}{\text{Var}\left[X_1\right]} &= \frac{\text{Cov}\left[X_1, \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \epsilon\right]}{\text{Var}\left[X_1\right]} && (31)\\
&= \frac{\beta_1 \text{Var}\left[X_1\right] + \beta_2 \text{Cov}\left[X_1, X_2\right] + \text{Cov}\left[X_1, \epsilon\right]}{\text{Var}\left[X_1\right]} && (32)\\
&= \beta_1 + \frac{\beta_2 \text{Cov}\left[X_1, X_2\right] + 0}{\text{Var}\left[X_1\right]} && (33)\\
&= \beta_1 + \beta_2 \frac{\text{Cov}\left[X_1, X_2\right]}{\text{Var}\left[X_1\right]} && (34)
\end{aligned}$$

The total covariance between X_1 and Y includes X_1's direct contribution to Y, plus the indirect contribution through correlation with X_2, and X_2's contribution to Y. (All of this applies, with subscripts swapped, to regressing Y on X_2 as well.)

Said slightly differently, when there's correlation between X_1 and X_2, we can predict (a bit of) X_2 from X_1 and vice versa. When we do simple regression, we don't care — adding up the direct and indirect relationships of Y and X_1 is fine. When we do multiple regression, we don't want to "double count" that contribution to Y, so the slopes should just reflect the relationship the response

and the part of each predictor variable we couldn't have already guessed from knowing the others.

(If you're wondering, "Wait, what if there's really an X_3 but we only regressed on X_1 and X_2, wouldn't we have the same sort of problem?", congratulations — you've just discovered **omitted variable bias**.)

The geometric answer Refer again to §3.1. The optimal slopes are given by

$$\mathrm{Var}\left[X\right]^{-1}\mathrm{Cov}\left[X,Y\right] \tag{35}$$

which means that the optimal predictions are given by

$$X^T\mathrm{Var}\left[X\right]^{-1}\mathrm{Cov}\left[X,Y\right] \tag{36}$$

(The transpose on X is because I chose to write vectors as column matrices, and we need to make this come out a scalar.)

Now, $\mathrm{Var}\left[X\right]$ is a square, symmetric $p \times p$ matrix, so it makes sense to talk about its square root[3], i.e., a symmetric $p \times p$ matrix $\mathrm{Var}\left[X\right]^{1/2}$ such that $\mathrm{Var}\left[X\right] = \mathrm{Var}\left[X\right]^{1/2}\mathrm{Var}\left[X\right]^{1/2}$. It follows that $\mathrm{Var}\left[X\right]^{-1}$ also has a square root, $\mathrm{Var}\left[X\right]^{-1/2}$, given by $\left(\mathrm{Var}\left[X\right]^{1/2}\right)^{-1}$. Thus we can say that the optimal predictions are given by

$$\begin{aligned} X^T\mathrm{Var}\left[X\right]^{-1}\mathrm{Cov}\left[X,Y\right] &= X^T\mathrm{Var}\left[X\right]^{-1/2}\mathrm{Var}\left[X\right]^{-1/2}\mathrm{Cov}\left[X,Y\right] & (37)\\ &= (\mathrm{Var}\left[X\right]^{-1/2}X)^T\mathrm{Cov}\left[\mathrm{Var}\left[X\right]^{-1/2}X, Y\right] & (38)\end{aligned}$$

By the rules for algebra with variances,

$$\begin{aligned} \mathrm{Var}\left[\mathrm{Var}\left[X\right]^{-1/2}X\right] &= \mathrm{Var}\left[X\right]^{-1/2}\mathrm{Var}\left[X\right]\mathrm{Var}\left[X\right]^{-1/2} & (39)\\ &= \mathrm{Var}\left[X\right]^{-1/2}\mathrm{Var}\left[X\right]^{1/2}\mathrm{Var}\left[X\right]^{1/2}\mathrm{Var}\left[X\right]^{-1/2} = \mathbf{I} & (40)\end{aligned}$$

Multiplying a vector by a matrix rotates and stretches the coordinate system for the vector. Multiplying X by $\mathrm{Var}\left[X\right]^{-1/2}$ rotates and stretches the coordinates so that all the components of X are uncorrelated with each other, and they all have variance 1. The point of the $\mathrm{Var}\left[X\right]^{-1}$ in the formula for the regression slopes is that it, implicitly, finds the new coordinate system where the predictors are uncorrelated, and then does a bunch of simple regressions.

3.3 Point Predictions and Fitted Values

Just as with simple regression, the vector of fitted values $\widehat{\mathbf{m}}$ is linear in $\mathbf{y}$, and given by the hat matrix:

[3]For instance, we know from the "spectral" or "eigendecomposition" theorem in linear algebra that such a matrix can be written as $U\Lambda U^T$, where U is the $p \times p$ matrix whose columns are the eigenvectors, and Λ is the diagonal matrix of eigenvalues.

$$\begin{aligned}
\widehat{\mathbf{m}} &= \mathbf{x}\widehat{\beta} && (41)\\
&= \mathbf{x}(\mathbf{x}^T\mathbf{x})^{-1}\mathbf{x}^T\mathbf{y} && (42)\\
&= \mathbf{H}\mathbf{y} && (43)
\end{aligned}$$

All of the interpretations given of the hat matrix in the previous lecture still apply.

4 Properties of the Estimates

We will only look at the most basic properties of bias and variance here, deferring the full sampling distribution, and confidence sets, to next time.

The fundamental observation is the following. Let's hold $\mathbf{x}$ fixed, and let $\mathbf{Y}$ vary randomly. Since

$$\widehat{\beta} = (\mathbf{x}^T\mathbf{x})^{-1}\mathbf{x}^T\mathbf{Y} \tag{44}$$

and

$$\mathbf{Y} = \mathbf{x}\beta + \epsilon \tag{45}$$

we have

$$\widehat{\beta} = (\mathbf{x}^T\mathbf{x})^{-1}\mathbf{x}^T\mathbf{x}\beta + (\mathbf{x}^T\mathbf{x})^{-1}\mathbf{x}^T\epsilon = \beta + (\mathbf{x}^T\mathbf{x})^{-1}\mathbf{x}^T\epsilon \tag{46}$$

4.1 Bias

This is straight-forward:

$$\begin{aligned}
\mathbb{E}\left[\widehat{\beta}|\mathbf{x}\right] &= \mathbb{E}\left[\beta + (\mathbf{x}^T\mathbf{x})^{-1}\mathbf{x}^T\epsilon|\mathbf{x}\right] && (47)\\
&= \beta + (\mathbf{x}^T\mathbf{x})^{-1}\mathbf{x}^T\mathbb{E}\left[\epsilon|\mathbf{x}\right] && (48)\\
&= \beta && (49)
\end{aligned}$$

Thus, the least squares estimate of the general linear model's coefficients is conditionally unbiased, no matter what p is.

Notice that we needed to use one of the modeling assumptions to get this: if the true regression function wasn't linear, we couldn't say $\mathbb{E}\left[\epsilon|\mathbf{x}\right] = 0$.

4.2 Variance and Standard Errors

This needs a little more work.

$$\begin{aligned}
\mathrm{Var}\left[\widehat{\beta}|\mathbf{x}\right] &= \mathrm{Var}\left[\beta + (\mathbf{x}^T\mathbf{x})^{-1}\mathbf{x}^T\epsilon|\mathbf{x}\right] && (50)\\
&= \mathrm{Var}\left[(\mathbf{x}^T\mathbf{x})^{-1}\mathbf{x}^T\epsilon|\mathbf{x}\right] && (51)\\
&= (\mathbf{x}^T\mathbf{x})^{-1}\mathbf{x}^T\mathrm{Var}\left[\epsilon|\mathbf{x}\right]\mathbf{x}(\mathbf{x}^T\mathbf{x})^{-1} && (52)\\
&= (\mathbf{x}^T\mathbf{x})^{-1}\mathbf{x}^T\sigma^2\mathbf{I}\mathbf{x}(\mathbf{x}^T\mathbf{x})^{-1} && (53)\\
&= \sigma^2(\mathbf{x}^T\mathbf{x})^{-1}\mathbf{x}^T\mathbf{x}(\mathbf{x}^T\mathbf{x})^{-1} && (54)\\
&= \sigma^2(\mathbf{x}^T\mathbf{x})^{-1} && (55)
\end{aligned}$$

Again, this is true whatever p might be.

To understand this a little better, let's re-write it slightly:

$$\text{Var}\left[\widehat{\beta}|\mathbf{x}\right] = \frac{\sigma^2}{n}(n^{-1}\mathbf{x}^T\mathbf{x})^{-1} \tag{56}$$

The first term, σ^2/n, is what we're familiar with from the simple linear model. As n grows, we expect the entries in $\mathbf{x}^T\mathbf{x}$ to be increasing in magnitude, since they're sums over all n data points; dividing all entries in the matrix by n compensates for this. If the sample covariances between all the predictor variables were 0, when we took the inverse we'd get $1/s^2_{X_i}$ down the diagonal (except for the top of the diagonal), just as we got $1/s^2_X$ in the simple linear model.

5 Collinearity

I have been silently assuming that $(\mathbf{x}^T\mathbf{x})^{-1}$ exists, that $\mathbf{x}^T\mathbf{x}$ is "invertible" or "non-singular". There are a number of equivalent conditions for a matrix to be invertible:

1. Its determinant is non-zero.
2. It is of "full column rank", meaning all of its columns are linearly independent[4].
3. It is of "full row rank", meaning all of its rows are linearly independent.

The equivalence of these conditions are mathematical facts, proved in linear algebra; I will not re-prove them here.

What does this amount to in terms of our data? It means (Exercise 5) that the variables must be linearly independent *in our sample*. That is, there must not be any set of constants $a_0, a_1, \ldots a_p$ where, for *all* rows i,

$$a_0 + \sum_{j=1}^{p} a_j x_{ij} = 0 \tag{57}$$

This, in other words, means that $\mathbf{x}$ must be of full column rank.

To understand why linearly dependence among variables is a problem, take an easy case, where two predictors, say X_1 and X_2, are exactly equal to each other. It's then not surprising that we don't have any way of estimating their coefficients. If we get one set of predictions with coefficients β_1, β_2, we'd get exactly the same predictions from $\beta_1 + \gamma$, $\beta_2 - \gamma$, no matter what γ might be. If there are other exact linear relations among two variables, we can similarly trade off their coefficients against each other, without any change in anything we can observe. If there are exact linear relationships among more than two variables, all of their coefficients become ill-defined.

[4]Recall that a set of vectors is linearly independent if no linear combination of them is exactly zero.

We will come back in a few lectures to what to do when faced with collinearity. For now, we'll just mention a few clear situations:

- If $n < p + 1$, the data are collinear.
- If one of the predictor variables is constant, the data are collinear.
- If two of the predictor variables are proportional to each other, the data are collinear.
- If two of the predictor variables are otherwise linearly related, the data are collinear.

While it's important to double-check for these, especially
for right now, we'll hope it doesn't happen. That does mean, however, that we need to look and see whether it *is* happening.

6 R Practicalities

6.1 `lm`

`lm` works in almost the same way as for simple linear models. Let's look at the model from the last data analysis project:

```
mobility <- read.csv("http://www.stat.cmu.edu/~cshalizi/mreg/15/dap/1/mobility.csv")
```

The only real change is that we need to tell `lm`, through the formula, what all the predictor variables are; we do this with `+` signs:

```
# Fit a model with three predictors
mob.lm <- lm( Mobility ~ Commute + Latitude + Longitude, data=mobility)
```

The order of the predictor variables only matters for the order in which the coefficients will be listed. All of the utility functions we already know still work, in exactly the same way:

```
# Basic print-out:
print(mob.lm)
```

```
##
## Call:
## lm(formula = Mobility ~ Commute + Latitude + Longitude, data = mobility)
##
## Coefficients:
## (Intercept)      Commute     Latitude    Longitude
##  -3.136e-02    2.010e-01    9.383e-04   -4.305e-05
```

```
# Coefficients:
coefficients(mob.lm)
```

```
##   (Intercept)       Commute      Latitude     Longitude
## -3.136000e-02  2.009679e-01  9.383055e-04 -4.304546e-05
```

```
# Confidence intervals for parameters:
confint(mob.lm)
```

```
##                     2.5 %        97.5 %
## (Intercept) -0.0563094963 -0.0064104992
## Commute      0.1738437953  0.2280920687
## Latitude     0.0003580771  0.0015185339
## Longitude   -0.0002827799  0.0001966889
```

```
# Fitted values:
head(fitted(mob.lm))
```

```
##          1          2          3          4          5          6
## 0.07172344 0.06156703 0.07867982 0.06006085 0.06464329 0.06943562
```

```
# Residuals:
head(residuals(mob.lm))
```

```
##            1            2            3            4            5
## -0.009524631 -0.007915094 -0.006044680 -0.003779634 -0.019842494
##            6
## -0.017599771
```

`summary` is also basically the same, but slightly more elaborate.

```
summary(mob.lm)
```

```
##
## Call:
## lm(formula = Mobility ~ Commute + Latitude + Longitude, data = mobility)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.17583 -0.02222 -0.00586  0.01758  0.32290
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3.136e-02  1.271e-02  -2.468  0.01383 *
## Commute      2.010e-01  1.382e-02  14.546  < 2e-16 ***
## Latitude     9.383e-04  2.956e-04   3.175  0.00156 **
## Longitude   -4.305e-05  1.221e-04  -0.353  0.72456
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.04227 on 725 degrees of freedom
## Multiple R-squared:  0.3583,Adjusted R-squared:  0.3557
## F-statistic: 134.9 on 3 and 725 DF,  p-value: < 2.2e-16
```

This lists t-tests for every coefficient; we will go exactly how to interpret those next time.

As usual, it is *much better* to use a formula with just column names and a `data` argument than to hard-code in particular vectors.

6.2 predict

`predict` also works in exactly the same way, only we need to give a data frame with columns for each of the predictor variables:

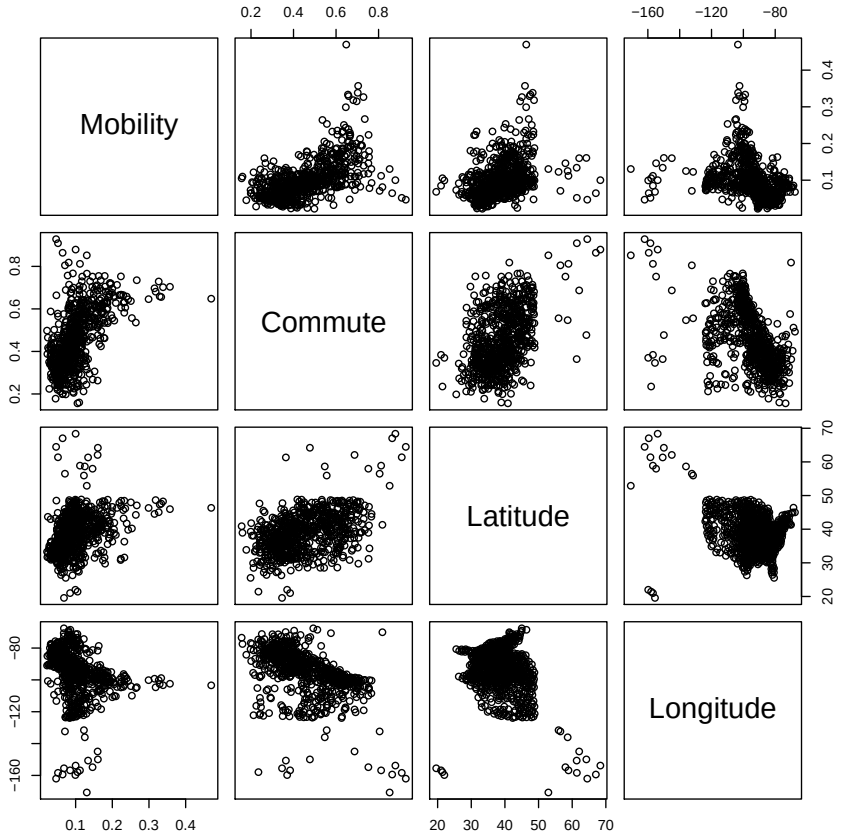
```
predict(mob.lm,
        newdata=data.frame(Commute=0.5,
                           Latitude=40.35,
                           Longitude=-79.92)) # Where is that?
```

```
##         1
## 0.1104248
```

Confidence intervals for conditional means, and prediction intervals, work in just the same way as before.

6.3 Exploratory Plots

While we will go over the diagnostic plots next time, some exploratory plots are needed at this point. The simplest thing to do is a bivariate scatter-plot for every pair of variables. You *could* do this by writing `plot` umpteen times, but this is such a common task that there's a useful R function to make all possible scatterplots, called `pairs` (Figure 1).




```
pairs(~ Mobility + Commute + Latitude + Longitude, data=mobility)
```

FIGURE 1: *Example of using* `pairs`*: the formula has an empty left-hand side (because there isn't really a distinguished response variable), and all the variables we want to plot on the right-hand side. If we left out the formula, we'd get plots of all variables against all others: why isn't that sensible here? What would happen if we used the formula* `Mobility ~ Commute + Latitude + Longitude`*?*

7 Exercises

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

1. Show that
$$\frac{1}{n}\mathbf{x}^T\mathbf{y} - \overline{\mathbf{x}}^T\overline{y} \tag{58}$$
is the $p \times 1$ matrix whose i^{th} entry is the sample covariance between X_i and Y.

2. Show that
$$\frac{1}{n}\mathbf{x}^T\mathbf{x} - \overline{\mathbf{x}}^T\overline{\mathbf{x}} \tag{59}$$
is the $p \times p$ matrix whose i, j entry is the sample covariance between X_i and X_j.

3.) Show the following:
   (a) That in the multiple-regression model, the true vector of slopes β equals $\mathrm{Var}[X]^{-1}\mathrm{Cov}[X, Y]$.
   (b) That this vector of slopes minimizes the *expected* squared error.

4. Assume $p = 2$. Work out $n^{-1}\mathbf{x}^T\mathbf{x}$ and $(n^{-1}\mathbf{x}^T\mathbf{x})^{-1}$ in terms of $\overline{x_1}$, $\overline{x_2}$, $\overline{x_1 x_2}$, $\overline{x_1^2}$ and $\overline{x_2^2}$.

5. (a) Show if $\mathbf{x}$ is of full column rank, than $\mathbf{x}^T\mathbf{x}$ is also of full rank.
   (b) Show that if $\mathbf{x}^T\mathbf{x}$ is not of full rank, then $\mathbf{x}$ must be of less than full column rank.