Lecture 3: Linear Regression and Prediction

Introduction to Time Series, Fall 2023

Ryan Tibshirani

Related reading: Chapter 2 of Shumway and Stoffer (SS); Chapters 5.8, 5.10, and 7 of Hyndman and Athanasopoulos (HA).

1 Simple regression

1.1 Population version

- We'll start off by learning the very basics of linear regression, assuming you have not seen it before. A lot of what we'll learn here is not necessarily specific to the time series setting, though of course (especially as the lecture goes on) we'll emphasize the time series angle as appropriate
- A *simple linear regression* model for a response variable y and predictor (or covariate, or feature) variable x is one in which we seek *coefficients* (or parameters) β_0 and β_1, such that, informally,

  $$y \approx \beta_0 + \beta_1 x$$

  To be clear, here x, y are all real-valued (rather than multivariate) random variables
- If we had access to the full distributions of x, y, which is what we call the "population version" of regression, then we could ask: what is the best choice of parameters β_0, β_1 with respect to expected squared error?
- Mathematically, we are looking to solve

  $$\min_{\beta_0, \beta_1} \mathbb{E}\big[(y - \beta_0 - \beta_1 x)^2\big] \tag{1}$$

  or in other words, we are asking for the "line of best fit" at the population level. You'll often also hear this referred to as the "least squares" problem
- We can find the answer by differentiating the loss $Q = \mathbb{E}[(y - \beta_0 - \beta_1 x)^2]$ in (1) with respect to each parameter and setting it equal to zero. Differentiating inside the expectation gives:

  $$\frac{\partial Q}{\partial \beta_0} = \mathbb{E}\big[2(\beta_0 + \beta_1 x - y)\big] = 0$$
  $$\frac{\partial Q}{\partial \beta_1} = \mathbb{E}\big[2x(\beta_0 + \beta_1 x - y)\big] = 0$$

- As you'll show on the homework, solving this pair of equations gives the *population regression coefficients*:

  $$\beta_1^* = \frac{\mathrm{Cov}(x, y)}{\mathrm{Var}(x)}, \quad \beta_0^* = \mathbb{E}(y) - \beta_1^* \mathbb{E}(x) \tag{2}$$

- Recalling that $\mathrm{Cor}(x, y) = \mathrm{Cov}(x, y)/\sqrt{\mathrm{Var}(x)\,\mathrm{Var}(y)}$, we may rewrite the slope as

  $$\beta_1^* = \mathrm{Cor}(x, y)\sqrt{\frac{\mathrm{Var}(y)}{\mathrm{Var}(x)}},$$

  which shows that it treats x, y *asymmetrically*. This is important to remember. In general, when y is the response and x is the predictor, we speak this relationship as the "regression of y on x"

1.2 Sample version

- For the "sample version" of linear regression, we seek β_0, β_1 such that for given samples x_i, y_i (covariate and response pairs), $i = 1, \ldots, n$,

$$y_i \approx \beta_0 + \beta_1 x_i, \quad i = 1, \ldots, n$$

but without access to the full distributions of each x_i and y_i, just given these samples

- We can imagine two ways to proceed:
  1. Start from the population-level formula (2), and use plug-in estimates for the covariance and variance
  2. Start from the population-least least squares problem (1), write down a sample version, then solve it
- Somewhat remarkably, these two strategies end up at the same answer (which need not be the case)
- For strategy 1, we use the sample covariance and sample variance,

$$\widehat{\mathrm{Cov}}(x, y) = \frac{1}{n}\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y}), \quad \widehat{\mathrm{Var}}(x) = \frac{1}{n}\sum_{i=1}^n (x_i - \bar{x})^2$$

where $\bar{x} = \frac{1}{n}\sum_{t=1}^n x_t$ and $\bar{y} = \frac{1}{n}\sum_{i=1}^n y_i$ are the sample means, and plug these into (2) to get:

$$\hat{\beta}_1 = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^n (x_i - \bar{x})^2}, \quad \hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x} \tag{3}$$

which we call the *sample regression coefficients*

- For strategy 2, we write down the sample least squares problem

$$\min_{\beta_0, \beta_1} \sum_{i=1}^n (y_i - \beta_0 - \beta_1 x_i)^2 \tag{4}$$

- Similar to before, denote the loss in (1) by $Q = \sum_{i=1}^n (y_i - \beta_0 - \beta_1 x_i)^2$ and differentiate with respect to each parameter and set it equal to zero:

$$\frac{\partial Q}{\partial \beta_0} = \sum_{i=1}^n 2(\beta_0 + \beta_1 x_i - y_i) = 0$$
$$\frac{\partial Q}{\partial \beta_1} = \sum_{i=1}^n 2x_i(\beta_0 + \beta_1 x_i - y_i) = 0$$

- You'll show on the homework that solving this pair of equations leads you right back to (3)
- The `lm()` function in R performs linear regression. The notation you use is `lm(y ~ x)`, where `y ~ x` is called a "formula". This can be read as an instruction: "regress y on x"
- Figure 1 gives an example where we regress chicken prices—our response, y, on time—our predictor x. Just to give you a clear sense, the data are

$$y_1 = 65.58,\ y_2 = 66.48,\ y_3 = 65.70, \ldots$$
$$x_1 = 2001.583,\ x_2 = 2001.667,\ x_3 = 2001.750, \ldots$$

where we interpret each value of x as a given year plus a fraction, representing the month of the year

- After running `lm()`, the resulting object is a (special) list, with a lot of useful components. Calling `coef()` on the object gives the regression coefficients

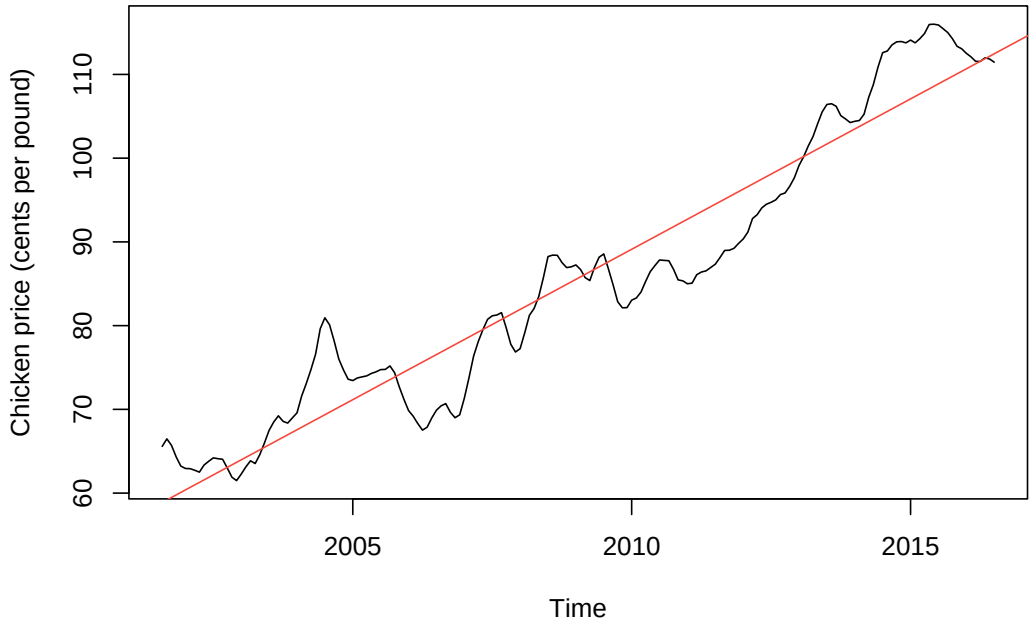

Figure 1: *Linear regression of chicken prices and on time (from SS).*

- Figure 2 gives another example, of a different flavor. Now the response y is itself one time series: cardiovascular mortality in Los Angeles over a certain time period, and the covariate x is itself another time series: particulate levels in Los Angeles over the same time period. The top panel in Figure 2 plots them individually as time series, and the bottom panel plots them together, as a scatter plot, toether with the fitted line from linear regression. We can imagine, in a future period (beyond the end date of these time series), using the estimated regression coefficients to predict mortality from particulate levels

1.3 Prediction: ex-ante and ex-post

- We can use the estimated coefficients $\hat{\beta}_0, \hat{\beta}_1$ in (3) from linear regression estimates to make *predictions* about the response given a new predictor value x_{new}. This prediction is

$$\hat{y}_{\text{new}} = \hat{\beta}_0 + \hat{\beta}_1 x_{\text{new}},$$

  where the "hat" notation on the left-hand side emphasizes that it is an not observed, but an estimated (predicted) value of the response

- In time series context, we often will use the term *forecasting* synomously with *prediction*. In the time series, there is an interesting distinction between two types of forecasts, based on *whether or not the predictor value x_{new} needed to make forecasts is available in advance*. Specifically:
  - An *ex-ante forecast* is a "true" forecast, using only information that is available at the time the forecast was issued. So the predictor values need to either be available, or themselves be forecasted. For instance, we can make ex-ante forecasts in the chicken regression example
  - An *ex-post forecast* is made using later information on the predictors. So we wait until x_{new} is observed, then issue our forecast. For instance, we can make ex-post forecasts in the mortality

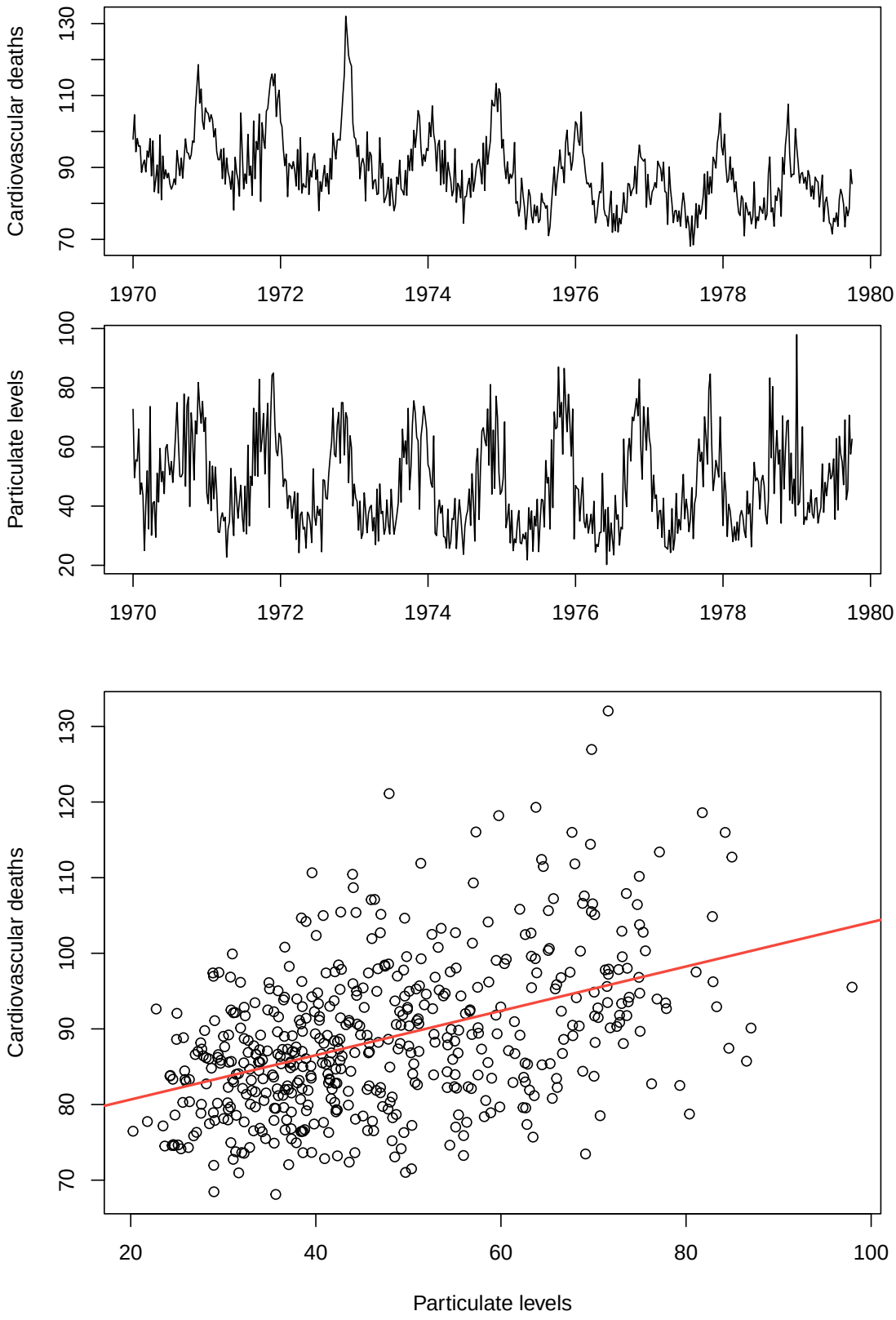

Figure 2: *Linear regression of cardiovascular mortality on particulate levels in Los Angeles (from SS).*

regression example (but cannot easily make ex-ante forecasts unless we somehow could forecast particulate levels into the future)

- One way around the circumvent the potential difficulty of making ex-ante forecasts is to use *lagged predictors*; we'll discuss this and other forecasting issues in more detail later

1.4 A note on assumptions and philosophy

- What have we assumed above? *Nothing.* That is, *we do not need to assume that the true relationship between y and x is linear in order to perform linear regression as in* (3)
- Thus, in general, there need not be any "true regression coefficients" that we're actually tracking ... but, we can always think of the sample estimates $\hat{\beta}_0, \hat{\beta}_1$ in (3) as tracking the population quantities β_0^*, β_1^* in (2). The latter are basically also always well-defined, regardless of linearity. Recall, they are the viewed as the best linear approximation at the population level
- So, to be clear, we can always fit sample coefficients $\hat{\beta}_0, \hat{\beta}_1$ and use them to make predictions (forecasting, in time series). Sometimes we call this our "working model": to use a linear working model is a modeling decision, not an assumption
- If this predicts well (has good accuracy), then our working model was a good decision, and depending on our use case, we may not even care about whether the true model is linear (or related assumptions in classical linear modeling)
- Meanwhile, for use cases would that require *inference*, we require lots of assumptions. More, later

2 Multiple regression

2.1 Population version

- What about the case where we have more than out covariate? This is called *multiple linear regression.* Now let $x = (x_1, \ldots, x_p) \in \mathbb{R}^p$ be a random vector of covariates, with each entry x_j being an individual covariate of interest
- We seek β_0, which is an intercept term as before, and also a whole coefficient vector $\beta = (x_1, \ldots, x_p) \in \mathbb{R}^p$, such that

$$y \approx \beta_0 + \underbrace{\beta_1 x_1 + \cdots + \beta_p x_p}_{\beta^\mathsf{T} x}$$

- Our convention throughout this class will be to *treat all vectors as column vectors.* Thus a^T, which is the transpose of a vector a, is a row vector, and for vectors $a, b \in \mathbb{R}^d$, we can use $a^\mathsf{T} b = a_1 b_1 + \cdots a_d b_d$ to denote their inner product.
- (Note that of course $a^\mathsf{T} b = b^\mathsf{T} a$, so it doesn't matter whether we write $\beta^\mathsf{T} x$ or $x^\mathsf{T} \beta$ in our model)
- As in (1), we define the population-level regression coefficients by minimizing expected squared error,

$$\min_{\beta_0, \beta} \mathbb{E}\big[(y - \beta_0 - x^\mathsf{T}\beta)^2\big] \tag{5}$$

- The solution, which you can think of as generalizing (2), is

$$\beta^* = \mathrm{Cov}(x)^{-1} \mathrm{Cov}(x, y), \quad \beta_0^* = \mathbb{E}(y) - \mathbb{E}(x)^\mathsf{T} \beta^* \tag{6}$$

- Let's check that the dimensions make sense: here $\mathrm{Cov}(x) \in \mathbb{R}^{p \times p}$, a $p \times p$ matrix of real values; the element in its i^{th} row and j^{th} column is

$$[\mathrm{Cov}(x)]_{ij} = \mathrm{Cov}(x_i, x_j)$$

Also $\text{Cov}(x, y) \in \mathbb{R}^p$, a p-dimensional vector, with i^{th} entry

$$[\text{Cov}(x, y)]_i = \text{Cov}(x_i, y)$$

So $\text{Cov}(x)^{-1}\,\text{Cov}(x, y) \in \mathbb{R}^p$, itself a p-dimensional vector, which is what we need for β^* in (6). Similarly, you can check that the dimensions make sense for β_0^*

- To derive (6) as the minimizer in (5), we can again differentiate with respect to each β_j and set the result to zero, but the calculation is a little more difficult (maybe it'll be a bonus on the homework)

2.2 Sample version

- The sample version of multiple linear regression falls out of the population version entirely analogously, as it did in the simple linear regression case. We seek $\beta_0 \in \mathbb{R}$ and $\beta_1 \in \mathbb{R}^p$ such that for given samples x_i, y_i (covariate and response pairs), $i = 1, \ldots, n$,

$$y_i \approx \beta_0 + \underbrace{\beta_1 x_{i1} + \cdots + \beta_p x_{ip}}_{x_i^\mathsf{T}\beta}, \quad i = 1, \ldots, n$$

- We can do this either by plug-in estimates from (6), or by writing down a sample version of the least squares problem (5) and solving it, and again they will both lead to the same answer
- Let's pursue the latter. It will be convenient to adopt the following convention, which alleviates us from keeping track of an explicit intercept term β_0, without any loss of generality. We simply write

$$y_i \approx x_i^\mathsf{T}\beta, \quad i = 1, \ldots, n$$

without intercept. Then all results that we will derive can be translated to the model with intercept via the following trick: we redefine each vector x_i so that it has a 1 prepended to it:

$$x = (1, x_1, \ldots, x_p) \tag{7}$$

Then we read of results in this new parametrization: post-transformation, the first entry of β serves as the intercept, and the rest serve as the coefficients multiplying each x_j

- (There is another way to get rid of the intercept as well, which we'll learn when we connect multiple to simple linear regression a bit later, which may be more intuitive to some of you)
- The sample least squares problem for multiple regression is now as follows:

$$\min_{\beta \in \mathbb{R}^p} \sum_{i=1}^n (y_i - x_i^\mathsf{T}\beta)^2 \tag{8}$$

- The solution, obtained again by taking derivatives and setting equal to zero, is

$$\hat{\beta} = \left(\sum_{i=1}^n x_i x_i^\mathsf{T} \right)^{-1} \sum_{i=1}^n x_i y_i \tag{9}$$

- You can check that the dimensions all make sense here (that the right-hand side in (9) produces a p-dimensional vector)

2.3 Matrix notation

- It is more convenient, once you've sufficiently familiarized yourself with matrix notation, to recast regression in terms of matrices and vectors. Let $y = (y_1, \ldots, y_n) \in \mathbb{R}^n$ be the vector of our response values, and $X \in \mathbb{R}^{n \times p}$ the matrix of our predictor vectors, whose i^{th} row is x_i^T

- In other words,

$$y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}, \quad X = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & & & \\ x_{n1} & x_{n2} & \cdots & x_{np} \end{bmatrix}$$

- Recall, for a coefficient vector $\beta \in \mathbb{R}^p$, the matrix-vector product $X\beta \in \mathbb{R}^n$, which to emphasize is an n-dimensional vector, is

$$X\beta = \begin{bmatrix} x_1^\mathsf{T}\beta \\ x_2^\mathsf{T}\beta \\ \vdots \\ x_n^\mathsf{T}\beta \end{bmatrix}$$

  This means that we can write our sample working model compactly as

$$y \approx X\beta$$

- Recall, the Euclidean norm $\|\cdot\|$ of a vector $a \in \mathbb{R}^d$ is defined as $\|a\|^2 = \sum_{i=1}^d a_i^2$. This means we can write our sample least squares problem (8) compactly as

$$\min_{\beta \in \mathbb{R}^p} \|y - X\beta\|^2 \tag{10}$$

- Finally, the sample least squares estimates (9) can be written as

$$\hat{\beta} = (X^\mathsf{T}X)^{-1}X^\mathsf{T}y \tag{11}$$

  This form (11) is by far the more commonly-used (and easily-remembered) form for the least squares coefficient estimates, compared to (9). You'll prove the equivalence between the two forms (9), (11) on the homework

- Important technical note: above in (11) (also in (9)), we have implicitly assumed that the features—the columns of X—are *linearly independent*. This *can only happen if* $p \leq n$, i.e., if we have no more features than samples. Otherwise, $X^\mathsf{T}X$ will not have an inverse, strictly speaking. This is not the end of the world and there are many interesting things to talk about (along the lines of regularization) when $p > n$, but it's beyond our scope right now

2.4 Multiple vs simple: a connection

- There is quite an interesting connection between multiple regression of y on x and simple regression of y on each x_j, the latter often referred to as a *marginal linear regression*

  Warning

  There will be a notational clash between x_i as used above and x_j as will be used here. Previously, we used $x_i \in \mathbb{R}^p$ to refer to vector containing all feature values for the i^{th} sample. Here, we are going to use $x_j \in \mathbb{R}^n$ to refer to the vector containing the j^{th} feature values measured over all samples.

  For example, suppose we have two features: apples and bananas, and we measure the quantity of each across $n = 100$ households. Then the previous subsection used $x_i \in \mathbb{R}^2$ as the number of apples and bananas in the i^{th} household. But here we'll use $x_j \in \mathbb{R}^{100}$ as the number of apples (if $j = 1$) in the 100 households, or bananas (if $j = 2$) in the households. Get it?

  Since the beginning of ~~time~~ regression, scholars have run up against this problem and have racked their brains for notational solutions. But there is no real good notational solution to this. (Yes there are lots of options but all of them have some ugliness to them.)

We'll just make to use $x_i \in \mathbb{R}^p$ to always *refer to all features for the i^{th} sample, and $x_j \in \mathbb{R}^n$ to* always *refer to the the j^{th} feature for all samples (and try to never mix indices). In other words, the i^{th} row and j^{th} column of the feature matrix X, respectively. So you just have to remember that i indexes samples (rows), and j indexes features (columns).*

End warning

- Now that we've gotten that important notational piece out of the way, we will define more quantities in order to describe the connection between multiple and marginal regression
- Fix any j (arbitrary). Given a single feature $x_j = (x_{1j}, \ldots, x_{nj}) \in \mathbb{R}^n$, and response vector $y = (y_1, \ldots, y_n) \in \mathbb{R}^n$, the marginal (or simple) regression of y on x_j, *without intercept*, is a very similar formula to what you saw in (3):

$$\tilde{\beta}_j = \frac{\sum_{i=1}^n x_{ij} y_i}{\sum_{i=1}^n x_{ij}^2}$$

  (Note: the effect of the intercept is only that it centers the values of x_{ij} around their sample average, which you'll revisit on the homework)
- We can rewrite this succinctly in inner product notation as:

$$\tilde{\beta}_j = \frac{x_j^\mathsf{T} y}{x_j^\mathsf{T} x_j} \tag{12}$$

- Meanwhile, let's consider the j^{th} estimated regression coefficient $\hat{\beta}_j$ from the multiple regression of y on X, as in (11). Is this the same? That is, is the j^{th} component of (11) the same as (12)?
- The answer is generally no! Putting in other features into the linear regression will generally affect the estimated coefficient for x_j, i.e., will alter its "predictive influence" on y
- However, there is a precise connection between multiple regression and marginal regression. Let's write $X_{-j} \in \mathbb{R}^{n \times p}$ for the feature matrix but after dropping $x_j \in \mathbb{R}^n$, the j^{th} column. Suppose we:
  - Regress y on X_{-j} (a regression of y on all but the j^{th} feature), yielding an estimated coefficient vector $\hat{\alpha} \in \mathbb{R}^{p-1}$, and residual

$$\hat{y}^{-j} = y - X_{-j}\,\hat{\alpha} \tag{13}$$

  - Regress x_j on X_{-j} (a regression of x_j on all of the other features), yielding an estimated coefficient vector $\hat{\theta} \in \mathbb{R}^{p-1}$, and residual

$$\hat{x}_j^{-j} = x_j - X_{-j}\,\hat{\theta} \tag{14}$$

- In these residuals (13), (14), we have "regressed out the influence" of all other features on each of y and x_j. Then, after removing such infuence, if we perform a marginal regression of $\hat{y}^{-j}$ on $\hat{x}_j^{-j}$, we get precisely the j^{th} multiple regression coefficient:

$$\hat{\beta}_j = \frac{(\hat{x}_j^{-j})^\mathsf{T} \hat{y}^{-j}}{(\hat{x}_j^{-j})^\mathsf{T} \hat{x}_j^{-j}} \tag{15}$$

- In other words, (15) connects multiple regression to marginal regression: it shows that the j^{th} coefficient in a multiple regression (11) is equivalent to a marginal regression of y on x_j, but only after we have accounted for the effects of all the other predictors, by "regressing them out"
- The best way to understand the relationship between (15) and (11) is geometrically (which is also a great way to view linear regression in general) but that perspecive requires a bit more advanced linear algebra, which we won't cover. For now, you can just think of the following: the bigger the *correlations* between x_j and columns of X_{-j}, the bigger the effect will be in (14), where we regress out X_{-j} from x_j, and this will make the multiple (15) and marginal (12) coefficients quite different from each other

- On the other hand, when x_j is uncorrelated with each column of X_{-j}, then the residual $\hat{x}_j^{-j}$ in (14) is no different from x_j, and in fact one can show that the multiple regression coefficient in (15) and the marginal one in (12) are exactly the same
- Figure 3 revisits the cardiovascular mortality example from earlier. Now we perform the regression of cardiovascular mortality on two features: particulate levels and temperature. (In R we can regress `y` on two features `x1` and `x2` by running `lm(y ~ x1 + x2)`.) For each feature, the coefficients from multiple regression aren't too different from the marginal regression coefficients (as is seen in the bottom panel by the slopes of the lines), though the intercept changes noticeably. The relatively small change in the coefficients on particulate levels and temperature is due to the fact that these two are not very correlated (as is seen by looking at their time series, which are a bit "out of phase" with each other)

2.5 Interlude: best linear unbiased estimator

- Briefly, we discuss an important optimality property here of least squares estimates, before moving on to classical inferential results next. We are going to need to introduce an assumption for this part: we assume that the response vector $y \in \mathbb{R}^n$ is related to the feature matrix $X \in \mathbb{R}^{n \times p}$ by

$$y = X\beta + \epsilon, \quad \text{where } \epsilon \sim \mathrm{WN}(0, \sigma^2 I) \tag{16}$$

  for some unknown coefficient vector $\beta \in \mathbb{R}^p$, and a white noise vector $\epsilon \in \mathbb{R}^n$
- In (16), $\epsilon \sim WN(0, \sigma^2 I)$ is our notation for a white noise vector: the first argument specifies that the mean is zero, $\mathbb{E}(\epsilon) = 0$, and the second argument specifies that the covariance satisfies $\mathrm{Cov}(\epsilon) = \sigma^2 I$, where I is the $n \times n$ identity matrix. That is, the components satisfy $\mathrm{Var}(\epsilon_i) = \sigma^2$ and $\mathrm{Cov}(\epsilon_i, \epsilon_j) = 0$ whenever $i \neq j$
- Importantly, in (16), the feature matrix X is assumed to be *fixed* (not random). Therefore, we can also write (16) even more compactly as

$$y \sim \mathrm{WN}(X\beta, \sigma^2 I)$$

- Assuming that X is fixed in the context of the above model is a fairly strong condition. We'll go into more why in the next section (when we go even further and assume normality of the error distribution), but for now we'll just emphasize that we are assuming that the mean of the response is truly linear in the features, and that the errors are homoskedastic—they have equal variance regardless of the feature values
- Ok! So under the model (16), what can we say about the least squares estimates in (11)? First, note that these are *unbiased* for the true coefficients β:

$$\begin{aligned} \mathbb{E}(\hat{\beta}) &= \mathbb{E}\big[(X^\mathsf{T}X)^{-1}X^\mathsf{T}y\big] \\ &= (X^\mathsf{T}X)^{-1}X^\mathsf{T}\mathbb{E}(y) \\ &= (X^\mathsf{T}X)^{-1}X^\mathsf{T}X\beta \\ &= \beta \end{aligned}$$

- This implies it is unbiased for any contrast of β. This is the term we sometimes give to an estimand of the form $a^\mathsf{T}\beta$, for an arbitrary vector $a \in \mathbb{R}^d$. Observe,

$$\begin{aligned} \mathbb{E}(a^\mathsf{T}\hat{\beta}) &= a^\mathsf{T}\mathbb{E}(\hat{\beta})\big[(X^\mathsf{T}X)^{-1}X^\mathsf{T}y\big] \\ &= (X^\mathsf{T}X)^{-1}X^\mathsf{T}\mathbb{E}(y) \\ &= (X^\mathsf{T}X)^{-1}X^\mathsf{T}X\beta \\ &= \beta \end{aligned}$$

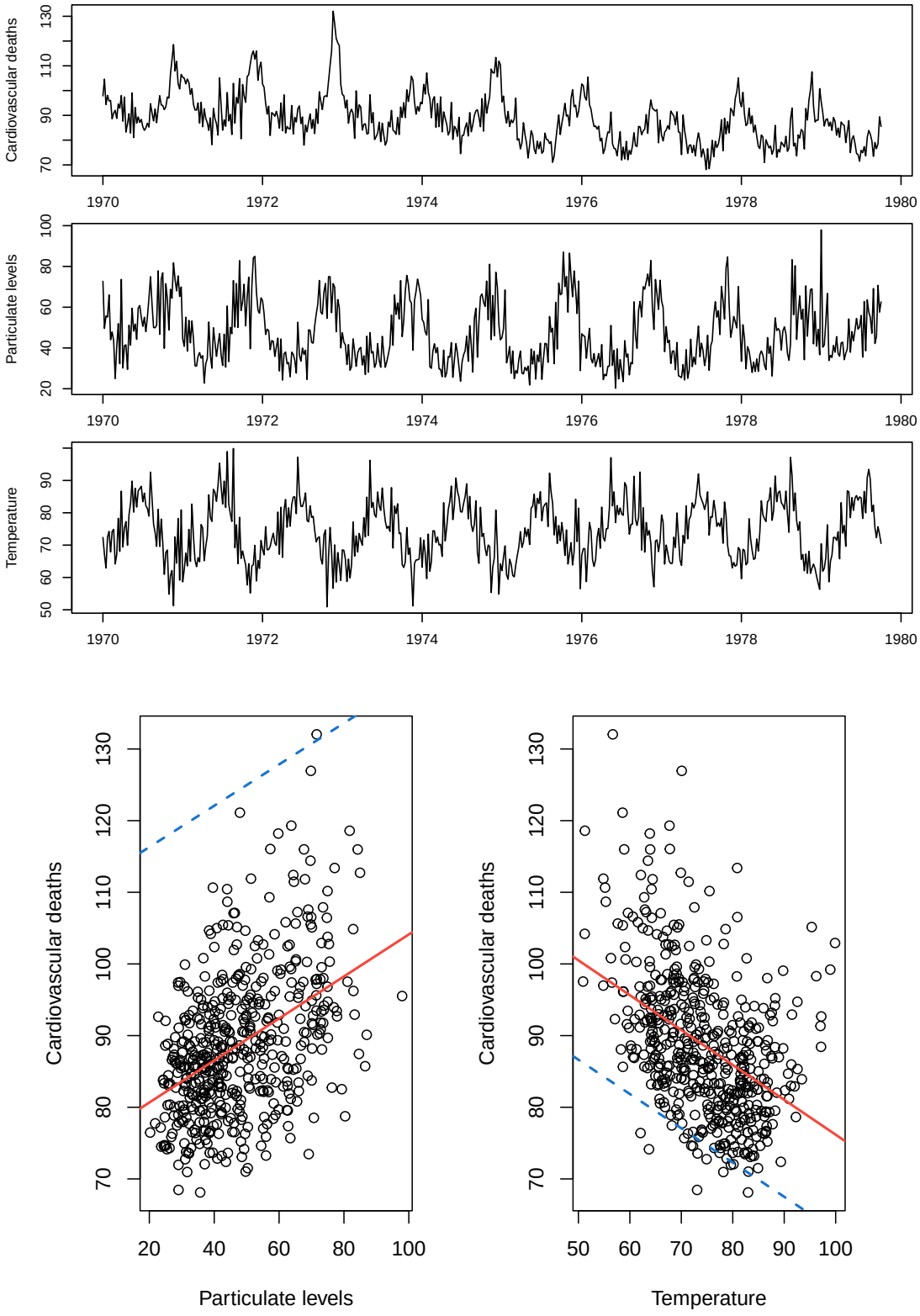

Figure 3: *Linear regression of cardiovascular mortality on particulate levels and temperature (this is multiple regression, with two features) in Los Angeles (from SS). The solid red lines denote the estimates from marginal regression; the dashed blue from multiple regression.*

- One common way to measure the quality of an estimator is its *mean squared error* (MSE), which for the least squares estimator $a^\mathsf{T}\hat{\beta}$ of the contrast $a^\mathsf{T}\beta$, is

$$\mathrm{MSE}(a^\mathsf{T}\hat{\beta}) = \mathbb{E}\big[(a^\mathsf{T}\hat{\beta} - a^\mathsf{T}\beta)\big]^2$$

  To be clear, the expectation here is being taken with respect to data from the model (16)

- With respect to MSE, how good is the least squares estimator? It is, in a certain precise sense, the *best*. It is both unbiased for $a^\mathsf{T}\beta$, as proved above, and also a *linear* estimator as a function of y: we can write

$$a^\mathsf{T}\hat{\beta} = \underbrace{(a^\mathsf{T}X^\mathsf{T}X)^{-1}X^\mathsf{T}}_{b^\mathsf{T}}y$$

  That is, $a^\mathsf{T}\hat{\beta} = b^\mathsf{T}y$, where $b = X(X^\mathsf{T}X)^{-1}a$

- Note: linearity of the estimator has nothing to do with linearity of the true regression model! The former (linearity of the estimator) is a statement about linearity in y, the response; the latter (linearity of the true regression model) is a statement about linearity in X, the features. There is a notational collision, but linearity is really referring to different things here

- Now for the main event: the *Gauss-Markov theorem* tells us that the least squares estimator is the best linear unbiased estimator (BLUE) of $a^\mathsf{T}\beta$. In other words, for any other linear estimator $c^\mathsf{T}y$, such that $\mathbb{E}(c^\mathsf{T}y) = a^\mathsf{T}\beta$ (unbiasedness), we have

$$\mathrm{MSE}(a^\mathsf{T}\hat{\beta}) \leq \mathrm{MSE}(c^\mathsf{T}y)$$

- A proof of this fact follows from the geometric perspective on least squares, which we won't cover, but you can ask about it in office hours if you are curious

- (An interesting side note! Econometricians have been recently arguing about whether or not we can drop the "L" from BLUE: is least squares the BUE? That is, is least squares the best unbiased estimator, period—best among all unbiased estimators, not just linear ones? The answer is ... yes, in a sense, but it depends on how you set up the problem, and in certain problem settings the only unbiased estimators are linear in y anyway.[1])

3 Classical inference

3.1 Here comes the assumptions

- Now we're going to cover some classical statistical inference for linear regression estimates. For this part, we're going to need to assume:

$$y = X\beta + \epsilon, \quad \text{where } \epsilon \sim N(0, \sigma^2 I) \tag{17}$$

  In comparison to (16), note that we have additionally assumed that the errors are multivariate Gaussian (this implies they are independent across observations as well)

- As before, the feature matrix X is assumed to be *fixed* (not random). Thus we can write (17) more compactly as

$$y \sim N(X\beta, \sigma^2 I)$$

[1] See Hansen (2022), "A modern Gauss-Markov theorem", and then Pötscher and Preinerstorfer (2022), "A modern Gauss-Markov theorem? Really?", and then Portnoy (2022), "Linearity of unbiased linear model estimators". A nice and friendly overview is given by Allison (2022), "Is OLS BLUE or BUE?", https://statisticalhorizons.com/is-ols-blue-or-bue/ (this is a blog post). A masterful, but much more mathematical treatment is given in Lei and Wooldrige (2022), who also weave in important historical results that seem to have been overlooked, and prove new ones as well.

- Taking X to be fixed here is a strong assumption. To see this, let's write this out as

$$y_i = x_i^\mathsf{T}\beta + \epsilon_i, \quad \text{where } \epsilon_i \overset{\text{iid}}{\sim} N(0, \sigma^2), \quad i = 1, \ldots, n$$

  If each x_i were indeed random (which is often the case in practice) then we would need *condition on* x_i in order to treat it as fixed. So then, what the above model really says is that each $\epsilon_i | x_i$ is normal with mean zero and variance σ^2. And for this to be true across all observations, we need the *errors* ϵ_i *and feature vectors* x_i *to be independent*

- Now we have gotten to the heart of why this is such a strong assumption. For the errors and features to be independent, we cannot have *heteroskedasticity* (error variance depending on the features); we also cannot really have any *omitted variables*, because if they are correlated with x_i, then they would appear in the effective error term ϵ_i, and break independence. Can you really make the argument that you have measured *all* of the relevant predictor variables in any given practical application of linear regression?

3.2 t-test for individual coefficients

- Under (17), we can define a *t-test*, based on the least squares estimate (11), for testing whether or not an individual coefficient is zero at the population-level: that is, for testing the hypothesis

$$H_0 : \beta_j = 0 \tag{18}$$

- To set us up to discuss this, we first define an estimate of the noise variance σ^2 in (17):

$$\begin{aligned} \hat{\sigma}^2 &= \frac{1}{n-p}\|y - X\hat{\beta}\|^2 \\ &= \frac{1}{n-p}\sum_{i=1}^n (y_i - x_i^\mathsf{T}\hat{\beta})^2 \end{aligned}$$

  This is the residual sum of squares from linear regression, divided by $n - p$

- We also define the matrix $C = (X^\mathsf{T}X)^{-1}$. Why is this important? Because:

$$\begin{aligned} \text{Cov}(\hat{\beta}) &= \text{Cov}\left((X^\mathsf{T}X)^{-1}X^\mathsf{T}y\right) \\ &= (X^\mathsf{T}X)^{-1}X^\mathsf{T}\,\text{Cov}(y)X(X^\mathsf{T}X)^{-1} \\ &= (X^\mathsf{T}X)^{-1}X^\mathsf{T}\sigma^2 I X(X^\mathsf{T}X)^{-1} \\ &= \sigma^2 X^\mathsf{T}X \\ &= \sigma^2 C \end{aligned}$$

  Note: in the second line we the property of covariance: for a random vector z and matrix A of appropriate dimension, $\text{Cov}(Az) = A\,\text{Cov}(z)A^\mathsf{T}$. We'll revisit this and related properties on the homework

- The above result implies that $\text{Var}(\hat{\beta}_j) = \sigma^2 C_{jj}$, where C_{jj} is the j^{th} diagonal element of C

- Recall that we also know that $\mathbb{E}[\hat{\beta}_j] = \beta_j$, since we already proved unbiasedness, above. Combining two these facts (about the mean and variance of $\hat{\beta}_j$) leads us to define the *t-statistic*

$$t_j = \frac{\hat{\beta}_j - \beta_j}{\hat{\sigma}\sqrt{C_{jj}}} \tag{19}$$

  Under (17), this is has a *t-distribution* with $n - p$ degrees of freedom. This can be used to test (18), or form a confidence interval for β_j. This follows the same general recipe that you have learned (or will learn) about hypothesis testing and confidence intervals in your concepts of statistics class

- When you call `lm()` in R, and then call `summary()` on its output, you are presented with these t-statistics and associated p-values (for the test that the coefficient is zero, as in (18))

- We will not cover this further, in any real detail, because (a) you'll likely spend a good amount of time on this when you learn regression in your concepts of statistics course, and (b) we won't really rely on classical inferential tools such as this t-test very much. After all, they rest on the assumption that the model (17) is correct, which (as we've discussed already) can be a dubious one in practice

3.3 F-test for subgroups of coefficients

- Even more briefly, we can form an *F-test* for the hypothesis that an entire *group* of coefficients is zero, which (with a loss of generality, just by relabeling the features), we can write as

$$H_0 : \beta_{k+1} \cdots = \beta_p = 0 \tag{20}$$

- We define

$$\text{SSE} = \sum_{i=1}^n (y_i - x_i^T \hat{\beta})^2, \quad \text{and} \quad \text{SSE}(k) = \sum_{i=1}^n \left(y_i - x_i^T \hat{\beta}(k)\right)^2$$

  where $\hat{\beta}(k) \in \mathbb{R}^k$ denotes the estimated regression coefficients when only the first k features are present

- We then define the *F-statistic*

$$F_k = \frac{(\text{SSE}(k) - \text{SSE})/(p-k)}{\text{SSE}/(n-p)} \tag{21}$$

  Under (17) and (20), this has a *central F-distribution* with $p-k$ and $n-p$ degrees of freedom. We can use this to test (20), the hypothesis that all but the first k true coefficients are zero

- An important special case: when $k = p-1$, we are testing whether $\beta_p = 0$, and it can be shown that the F-statistic F_p in (21) is equivalent to the t-statistic t_p in (19) for testing $\beta_p = 0$, from the previous subsection

3.4 AIC, BIC, AICc, R^2, adjusted R^2, ...

- We will not cover these. These are classical tools for variable/model selection. You can read about them in SS (Chapter 2.1) and/or HA (Chapter 7.5) books if you are interested. We'll take a more predictive angle, and focus instead on tools like cross-validation

3.5 Diagnostics, transformations

- There are a variety of diagnostic (statistical) tools for evaluating the output of a regression model. One of the most relevant tools to us, in the time series setting, is to examine the auto-correlation function of the residuals

$$\hat{\epsilon}_i = y_i - x_i^T \hat{\beta}, \quad i = 1, \ldots, n$$

- Recalling (16), one of the assumptions underlying the theory of linear regression (the Gauss-Markov theorem, t-tests, F-tests, etc.) is that the errors $\epsilon_i = y_i - x_i^T \beta$, $i = 1, \ldots, n$ form a white noise sequence

- Thus inspecting the auto-correlation function of residuals gives us a sense of whether this white noise assumption is true or not

- Figure 4 shows the result for the mortality regression example. We see that there are some very clear auto-correlations persisting through about lag 15. This should have us call into question any classical inferential tools we may want to apply. However, more constructively, it means that we may be able to improve the regression by taking into account the fact that there is structure/information left in the residuals. We will revisit this when we discuss ARIMA models, later

- Beyond looking at the auto-correlation of the residuals, there are a variety of classical diagnostics for regression. You can read about these in many sources, including the R help file for `plot.lm()`, and the HA book (Chapter 7.3). We will not be able to cover them, for the sake of time, however they

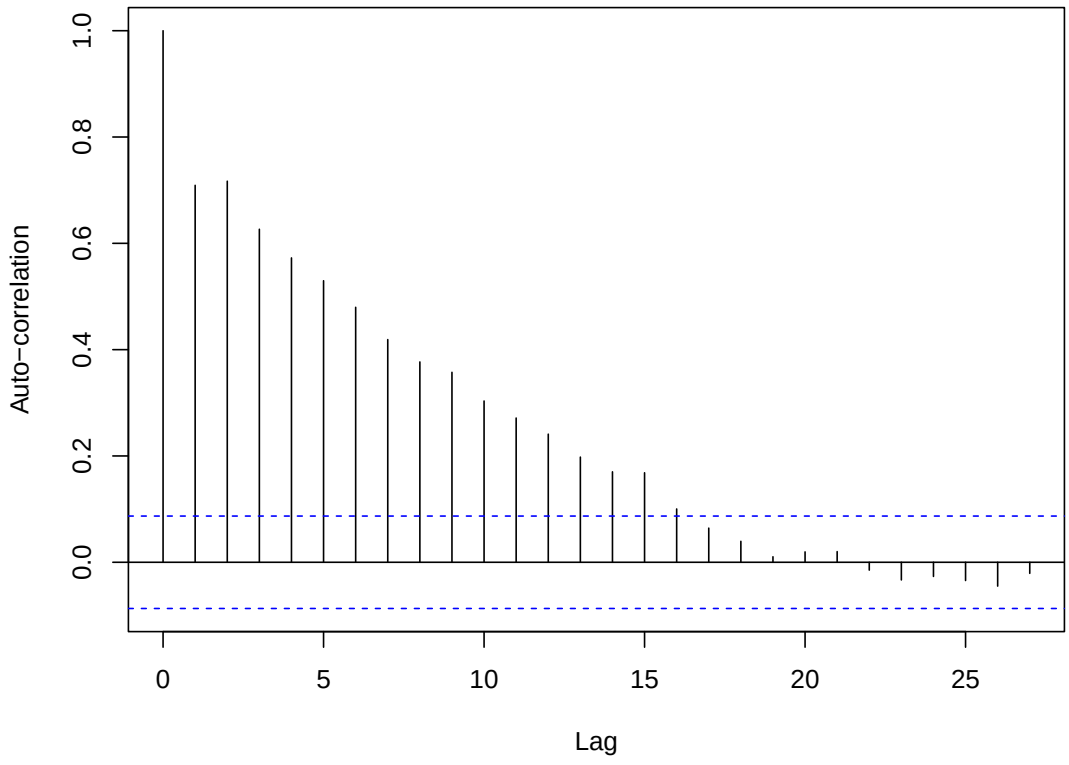

Figure 4: Auto-correlation function of the residuals for the linear regression of cardiovascular mortality on particulate levels.

are worth being generally aware of, and so worth reading up on separately (and/or hopefully you will see them in more detail in another course)

- Finally, closely related to diagnostics, and another classic topic, are transformations— made either to the response or features—which can improve the regression model. We won't cover them for the sake of time, but you can look at the HA book for details (Chapters 3.1, 5.6, and 7.4)

3.6 Correlation vs causation?

- *Correlation is not causation.* Hopefully that's very clear to you in the abstract, and you've been warned of it several times already in previous courses
- However, in the context of regression, it gets a bit murky. Estimation in regression is, at its core, about correlation (recall, e.g., (15)). But causality lurks in the background in the interpretation of regression coefficients (which we have purposefully avoided), and inferential tools like the t-test for individual coefficients as in (18)
- You have to be very careful to remember that these are predicated on the correctness of the model (17) (with fixed X). *It is this model that allows us to translate statements about correlations to ones that look causal.* If this model is wrong (likely!) then these translations break down
- As already mentioned several times, we are taking a predictive angle and will be mostly focused on evaluating models for their utility in prediction, or forecasting in time series
- Fundamentally, *correlations* between variables can be useful for forecasting, whether or not they are causal. Everybody would probably agree that they'd prefer a simple causal relationship if they could find one; but that is a much, much harder problem. Focusing on correlations and predictions (and

eschewing much of classical inference and interpretation) can be much more tractable, in a sense

- But at the bottom of it all, *"does the model predict well?"* is simply a different question. And it may not be the question that you want to answer in every application, so having inference tools in your toolkit, understanding when they are valid, and learning about causality, will make you a more well-rounded statistician

4 Prediction

4.1 Lagged predictors

- In certain situations, as discussed above, ex-ante forecasts may be difficult to obtain. Recall that in the cardiovascular mortality regression example, where y_t is the number of cardiovascular deaths at time t and x_t is the level of particulate matter at time t, a forecast of mortality $\hat{y}_{t+k}$ at time $t+k$ would only be possible given the particulate level x_{t+k} at time $t+k$, which is not available in advance
- One way around this is use *lagged predictors* in the regression, i.e., instead of regressing y_t on x_t, we regress y_t on x_{t-k}, i.e., we regress deaths each week on particulate levels k weeks earlier:

$$y_t \approx \beta_0 + \beta_1 x_{t-k}, \quad t = 1, 2, 3, \ldots$$

- In doing so, we can make predictions up to k weeks into the future. To be clear, if we've only observed data up through time t, then we can still make forecasts

$$\hat{y}_{t+i} = \hat{\beta}_0 + \hat{\beta}_1 x_{t+i-k}, \quad i = 1, \ldots, k$$

- Figure 5 gives an example where we do so with $k = 4$. We are therefore able to make true, ex-ante forecasts 4 weeks past the end of the time series. However, we are not able to validate these, since don't actually have data past the summer of 1979 in this example
- As validation, we can refit the lagged regression on the first half of the time series, and use the second half to 4-week ahead make forecasts and compare them to the observed series. This is done in Figure 6. It looks OK in terms of its *dynamics*: the trend (rise and fall) in the predictions matches the observed trend fairly well, but pretty soon it appears to be biased upwards, particularly at the trough of each yearly cycle
- (A word of warning: pulling this off correctly in R is actually more tricky than you might expect, so study the code in the R notebook carefully ...)
- Of course, we can also use more than one lag as a predictor, and use as our working model:

$$y_t \approx \beta_0 + \sum_{j=1}^{m} \beta_j x_{t-k_j}, \quad t = 1, 2, 3, \ldots$$

  for lags $k_1 < \cdots < k_m$. Note that this model would allow us to make forecasts k_1 time steps into the future (a lagged predictor model is limited by its smallest lag)
- The more lags—the more features, in general—that we include in the working model, the more *expressive* it is, meaning, it is able to capture and propogate more intricate trends. But also, the more *volatile* it can become, meaning, it can make more wild predictions
- Thus there is a tradeoff here, often referred to as the *bias-variance tradeoff*. More expressive = lower bias, more volatile = higher variance. This tradeoff generally expresses itself differently in different problems, and is not something we can anticipate precisely in advance. However, fortunately, we can still use generic tools like cross-validation to estimate prediction error (which measures bias and variance in aggregate). We will cover this next
- Another important general tool to mention is regularization, which can often tilt the bias-variance tradeoff in our favor. We will also cover this shortly

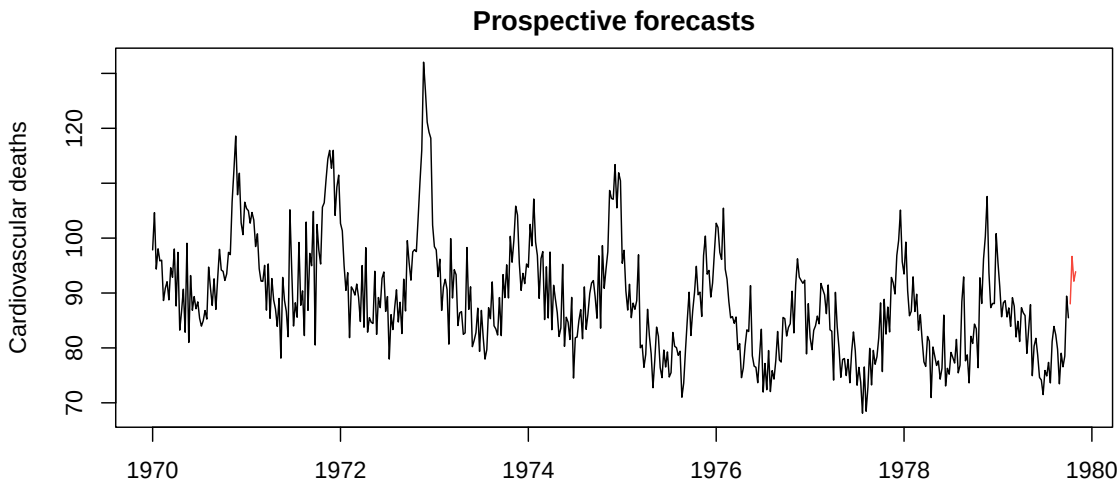

Figure 5: Forecasts of cardiovascular mortality made at a 4-week ahead horizon, using particulate level as a lagged predictor. These are prospective forecasts, made at the end of the time series.

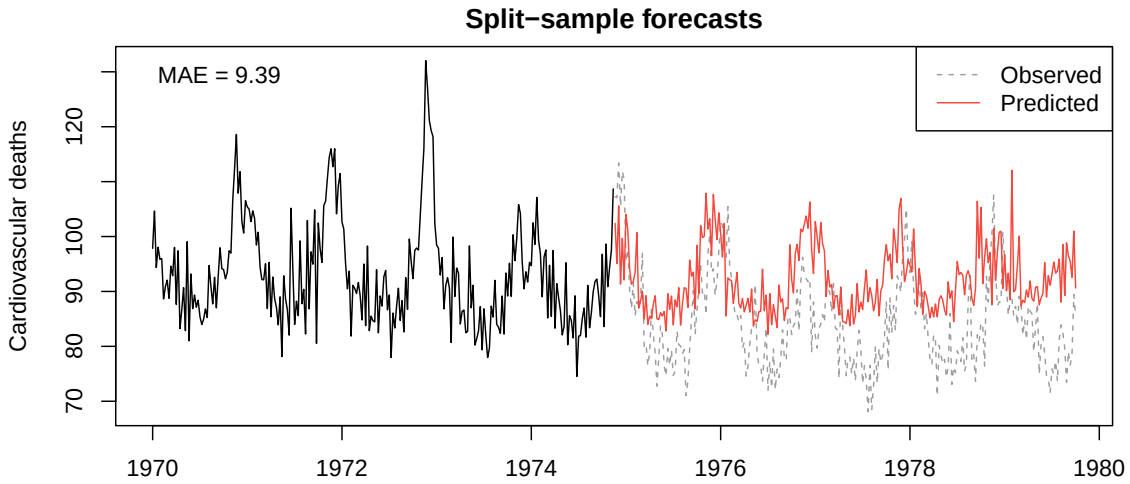

Figure 6: As in the above figure, but pseudo-prospective forecasts on the second half, from a regression model fit only to the first half.

4.2 Error metrics

- To evaluate regression models for their predictive accuracy, we'll first have to talk about error metrics: the precise formulae we will be using to measure this. The most common one is ean squared error (MSE) between predictions $\hat{y}_{\text{new},t}$ and unseen observations $y_{\text{new},t}$, over test times $t = 1, \ldots, N$:

$$\text{MSE} = \frac{1}{N} \sum_{t=1}^{N} \left(y_{\text{new},t} - \hat{y}_{\text{new},t} \right)^2$$

- Another common one is mean absolute error (MAE), which tends to focus less on extreme errors:

$$\text{MAE} = \frac{1}{N} \sum_{t=1}^{N} \left| y_{\text{new},t} - \hat{y}_{\text{new},t} \right|$$

- Note that MSE and MAE are scale-dependent: they depend on the scale of the response and hence

are not universally interpretable. This leads some to prefer mean absolute percentage error (MAPE):

$$\text{MAPE} = 100 \times \frac{1}{N} \sum_{t=1}^{N} \frac{|y_{\text{new},t} - \hat{y}_{\text{new},t}|}{y_{\text{new},t}}$$

- While intuitively appealing, MAPE can also have undesirable and erratic behavior if the observation $y_{\text{new},t}$ is close to zero. (Another problem that is often overlooked is that it assumes the unit of measurement actually has a meaningful zero—e.g., are we using Farenheit or Celsius for temperature forecasts? This would give potentially very different answers in terms of MAPE)
- A nice alternative for forecasting applications, which maintains the scale-free aspect but avoids the zero-pitfall, is mean absolute scaled error (MASE):

$$\text{MASE} = 100 \times \frac{\frac{1}{N} \sum_{t=1}^{N} |\hat{y}_{\text{new},t} - y_{\text{new},t}|}{\frac{1}{N-1} \sum_{t=2}^{N} |y_{\text{new},t} - y_{\text{new},t-1}|}$$

  In words, we are normalizing the error of our forecasts by that of a naive method which always predicts the last observation
- This is *not* just a thought exercise. Error metrics matter in practice! They really, really do. By this, we mean that different metrics might lead you to prefer different forecasting models. We'll return to this in more detail later in the course when we talk about forecast scoring rules

4.3 Optimism of training error

- The cheapest, easiest estimate of prediction error, whether measured by MSE, MAE, MAPE, etc., is to look at *training error*, which is the name we give to the average error we made on the training set that was used to fit the model. For example, for the mean squared error metric, the training error would be

$$\text{TrainErr} = \frac{1}{n} \sum_{t=1}^{n} (\hat{y}_t - y_t)^2$$

  where y_t, $t = 1, \ldots, n$ are the responses that were used to fit the model that produced $\hat{y}_t$, $t = 1, \ldots, n$
- This, in general, is *not* a good estimate of prediction whatsoever! It can be wrong both in absolute and relative terms
- That is, for a given model, it is generally *too optimistic* as an estimate of prediction error. Equally, if not more more problematic: it is *more optimistic the more complex the model*. We'll examine these issues on the homework
- (Classical optimism theory in regression, which we won't cover, is actually very beautiful; ask about it in office hours if you are curious)

4.4 Time series cross-validation

- A better estimate of prediction error is given by *cross-validation* or related techniques
- Actually, even simpler is called a *split-sample* (also called a *hold-out*) estimate of prediction error: this is what we did in Figure 6, where we only fit the model to the first half of the time series, and used the second half to make predictions. Formally, if we just measured MSE (or MAE, or whatever error metric we like) on the second half of the data, starting at time $t_0 + 1$, then these would be our hold-out estimates of prediction error:

$$\text{SplitErr} = \frac{1}{n - t_0} \sum_{t=t_0+1}^{n} (\hat{y}_t - y_t)^2$$

  Critically, the responses y_t, $t > t_0$ were *not* used to fit the model that produced the predictions $\hat{y}_t$, $t > t_0$. Rather, this model was only fit on data y_t, $t \leq t_0$

- The top left corner of Figure 6 reports the split-sample MAE, computed precisely in this way
- In forecasting, split-sample estimates of prediction error mimic a situation where we would *never refit the model* in the future. However, if we would indeed refit the model in the future (given access to more data), then they are not really appropriate, and are generally pessimistic
- Enter *time series cross-validation*, where we walk forward in time, refit the model given all data that we would have available at the given forecast date, make a forecast, record the error, and continue. For 1-step ahead forecasts, the cross-validation (CV) error estimate is

$$\text{CVErr} = \frac{1}{n-t_0} \sum_{t=t_0+1}^{n} \big(\hat{y}_{t|t-1} - y_t\big)^2$$

  where $\hat{y}_{t|t-1}$ indicates that this prediction came from a model that was fit on data up through time $t-1$: that is, y_s, $s \leq t-1$
- For k-step ahead forecasts, the CV error estimate is

$$\text{CVErr} = \frac{1}{n-t_0} \sum_{t=t_0+1}^{n} \big(\hat{y}_{t|t-k} - y_t\big)^2$$

  where similarly $\hat{y}_{t|t-k}$ indicates that this came from a model that was fit on data y_s, $s \leq t-k$
- Note that in either of the above two displays, we typically do not set $t_0 = 0$, but allow ourselves to fit the initial model (in the first step of time series CV) on some nontrivial amount of data $y_1, \ldots, y_{t_0}$ that we is sometimes call the *burn-in set*
- CV will be our main tool for *model selection*. Unsure how many lags to include (or whether to use regularization, or what regularization strength to use, and so on)? Compute CV error estimates for each candidate model, and select the one that performs best according to the error metric at hand
- Figure 7 visualizes the split-sample and cross-validation schemes for time series
- Figure 8 then displays the walk-forward predictions from time series cross-validation on the cardiovascular mortality example. (Recall, these are 4-step ahead forecasts, so the CV scheme is as in the bottom panel of Figure 7.) We can see that the MAE has improved a bit from the split-sample forecasts in Figure 6 (from about 9.39 to 7.95). However, the forecasts still look biased upwards

4.5 Trailing windows

- It turns out that we can further improve the forecasts in the cardiovascular mortality example by training on a *trailing window*, rather than on all past. In other words, to fit the regression model we use to make a forecast at time t, instead of solving

$$\min_{\beta_0,\beta_1} \sum_{s=1}^{t-1} (y_s - \beta_0 - \beta_1 x_{s-k})^2$$

  we solve

$$\min_{\beta_0,\beta_1} \sum_{s=t-w}^{t-1} (y_s - \beta_0 - \beta_1 x_{s-k})^2$$

  for a choice of window length w
- Figure 9 shows such forecasts with a window length $w = 10$ (i.e., 10 weeks). We see that it does significantly better, both qualitatively and in terms of MAE (from about 7.95 to 5.26)
- Why does this happen? Because the relationship between cardiovascular mortality and particulate level is changing over time. In the language of the last lecture, these two series, cardiovascular mortality and particulate level, are *not jointly stationary*

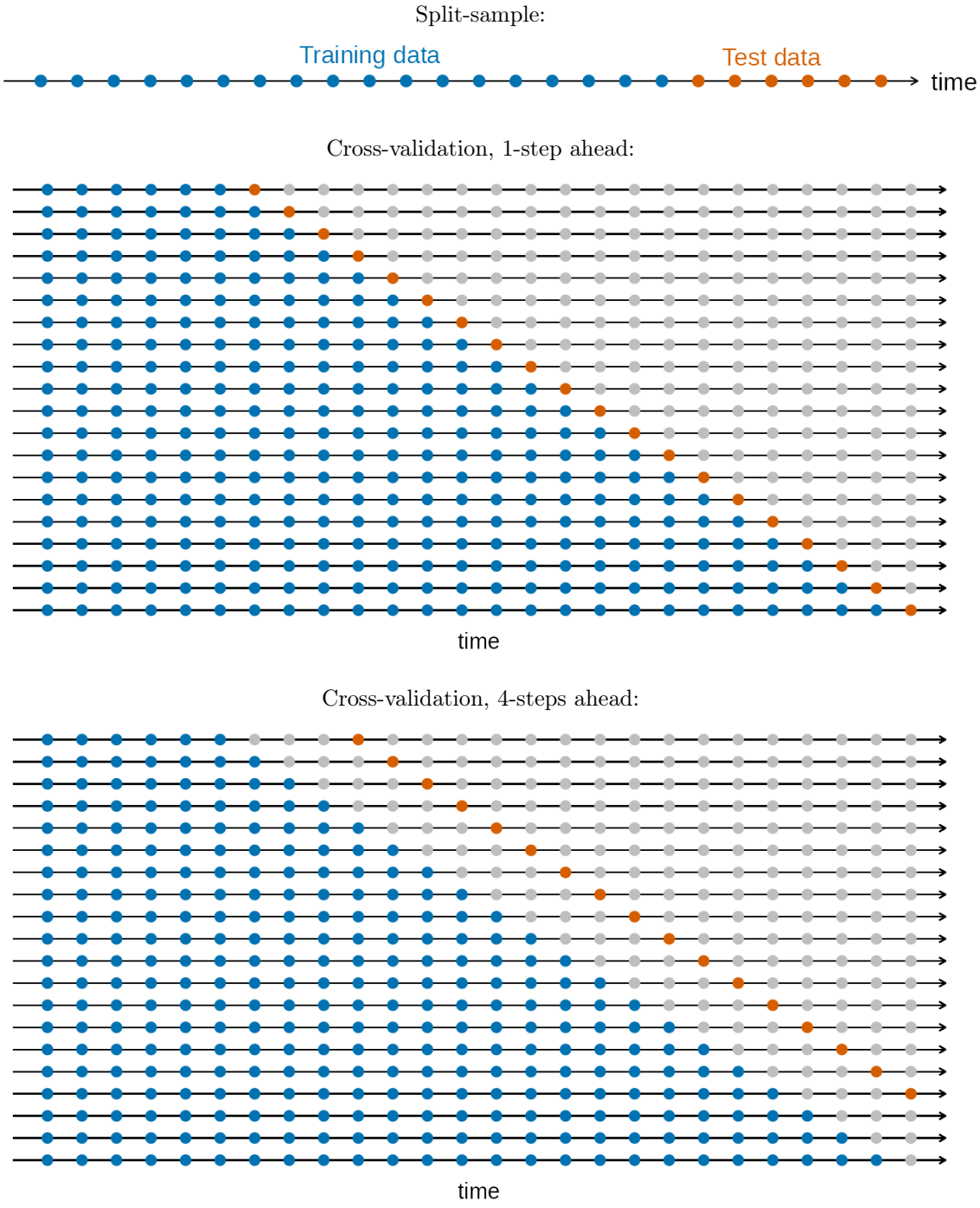

Figure 7: Visualization of split-sample and time series cross-validation schemes (from HA).

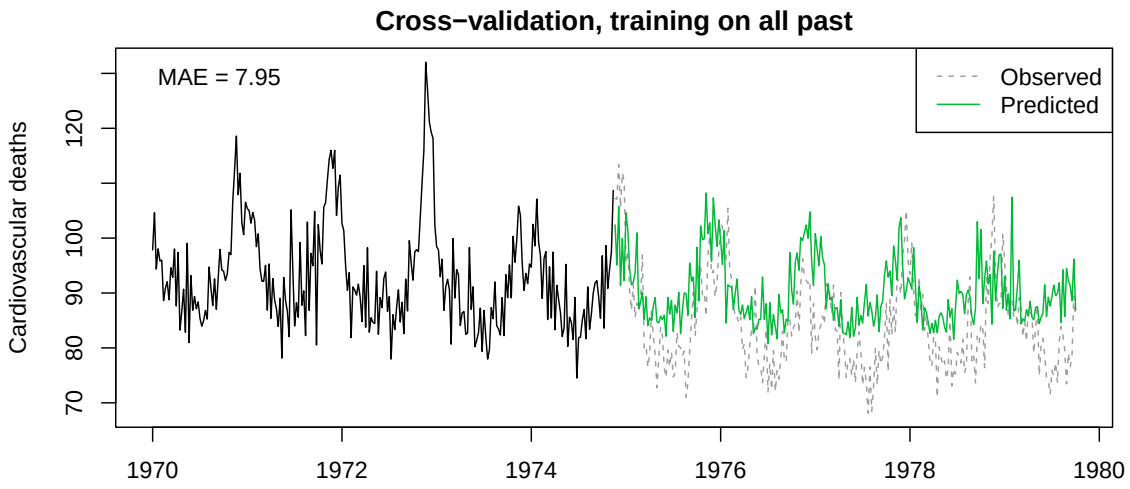

Figure 8: Walk-forward forecasts of cardiovascular mortality, made at a 4-week ahead horizon, using particulate level as a lagged predictor. The regression model is trained on all past.

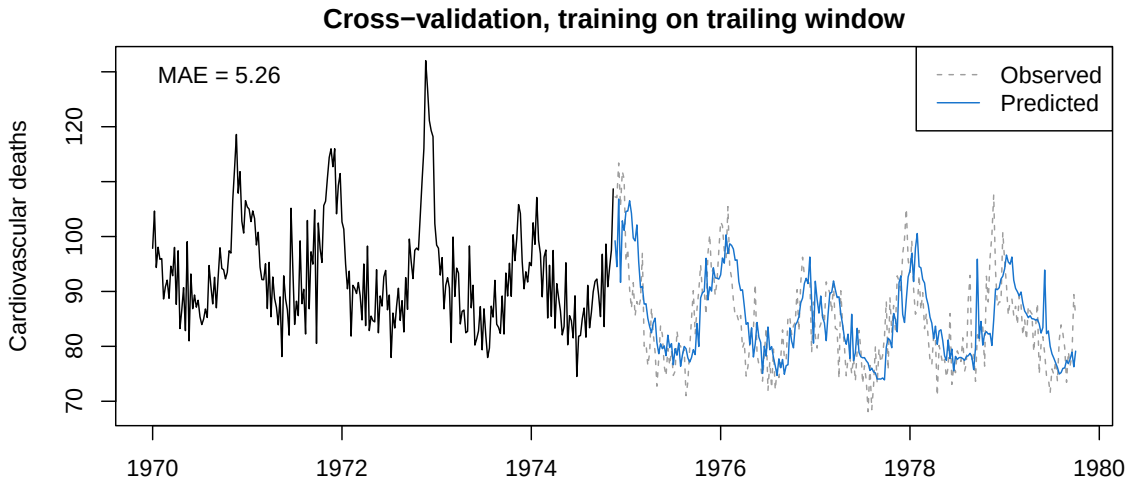

Figure 9: As in the above figure, but with the regression model trained on a trailing window of 10 weeks.

- Training on a trailing window helps to hone in on the most relevant (recent) data for prediction. If the window is too long, then the fitted model cannot adapt to nonstationarity as well; if the window is too short, then the fitted model may be too volatile (trained on too little data)
- The choice of window w should be rigorously examined, just like the choice and number of lags. To do so, we can use CV, once again—and to reiterate, CV is our main tool to select *tuning parameters* of the working model (the number of lags and the length of the trailing window being two examples)