Introduction

To the Reader

This book began as the notes for 36-402, Advanced Data Analysis, at Carnegie Mellon University. This is the methodological capstone of the core statistics sequence taken by our undergraduate majors (usually in their third year), and by undergraduate and graduate students from a range of other departments. The pre-requisite for that course is our class in modern linear regression, which in turn requires students to have taken classes in introductory statistics and data analysis, probability theory, mathematical statistics, linear algebra, and multivariable calculus. This book does not presume that you once learned but have forgotten that material; it presumes that you *know* those subjects and are ready to go further (see p. 15, at the end of this introduction). The book also presumes that you can read and write simple functions in R. If you are lacking in any of these areas, this book is not really for you, at least not now.

ADA is a class in *statistical methodology*: its aim is to get students to understand something of the range of modern[1] methods of data analysis, and of the considerations which go into choosing the right method for the job at hand (rather than distorting the problem to fit the methods you happen to know). Statistical *theory* is kept to a minimum, and largely introduced as needed. Since ADA is also a class in *data analysis*, there are a lot of assignments in which large, real data sets are analyzed with the new methods.

There is no way to cover every important topic for data analysis in just a semester. Much of what's not here — sampling theory and survey methods, experimental design, advanced multivariate methods, hierarchical models, the intricacies of categorical data, graphics, data mining, spatial and spatio-temporal statistics — gets covered by our other undergraduate classes. Other important areas, like networks, inverse problems, advanced model selection or robust estimation, have to wait for graduate school[2].

The mathematical level of these notes is deliberately low; nothing should be beyond a competent third-year undergraduate. But every subject covered here can be profitably studied using vastly more sophisticated techniques; that's why

[1] Just as an undergraduate "modern physics" course aims to bring the student up to about 1930 (more specifically, to 1926), this class aims to bring the student up to about 1990–1995, maybe 2000.

[2] Early drafts of this book, circulated online, included sketches of chapters covering spatial statistics, networks, and experiments. These were all sacrificed to length, and to actually finishing.

this is advanced data analysis from an *elementary* point of view. If reading these pages inspires anyone to study the same material from an *advanced* point of view, I will consider my troubles to have been amply repaid.

A final word. At this stage in your statistical education, you have gained two kinds of knowledge — a few general statistical principles, and many more specific procedures, tests, recipes, etc. Typical students are much more comfortable with the specifics than the generalities. But the truth is that while none of your recipes are *wrong*, they are tied to assumptions which hardly ever hold[3]. Learning more flexible and powerful methods, which have a much better hope of being reliable, will demand a lot of hard thinking and hard work. Those of you who succeed, however, will have done something you can be proud of.

Organization of the Book

supervised

Part I is about regression and its generalizations. The focus is on nonparametric regression, especially smoothing methods. (Chapter 2 motivates this by dispelling some myths and misconceptions about linear regression.) The ideas of cross-validation, of simulation, and of the bootstrap all arise naturally in trying to come to grips with regression. This part also covers classification and specification-testing.

unsupervised

Part II is about learning distributions, especially multivariate distributions, rather than doing regression. It is possible to learn essentially arbitrary distributions from data, including conditional distributions, but the number of observations needed is often prohibitive when the data is high-dimensional. This motivates looking for models of special, simple structure lurking behind the high-dimensional chaos, including various forms of linear and non-linear dimension reduction, and mixture or cluster models. All this builds towards the general idea of using graphical models to represent dependencies between variables.

no

Part III is about causal inference. This is done entirely within the graphical-model formalism, which makes it easy to understand the difference between causal prediction and the more ordinary "actuarial" prediction we are used to as statisticians. It also greatly simplifies figuring out when causal effects are, or are not, identifiable from our data. (Among other things, this gives us a sound way to decide what we ought to control for.) Actual estimation of causal effects is done as far as possible non-parametrically. This part ends by considering procedures for discovering causal structure from observational data.

some

Part IV moves away from independent observations, more or less tacitly as-

[3] "Econometric theory is like an exquisitely balanced French recipe, spelling out precisely with how many turns to mix the sauce, how many carats of spice to add, and for how many milliseconds to bake the mixture at exactly 474 degrees of temperature. But when the statistical cook turns to raw materials, he finds that hearts of cactus fruit are unavailable, so he substitutes chunks of cantaloupe; where the recipe calls for vermicelli he uses shredded wheat; and he substitutes green garment dye for curry, ping-pong balls for turtle's eggs and, for Chalifougnac vintage 1883, a can of turpentine." — Stefan Valavanis, quoted in Roger Koenker, "Dictionary of Received Ideas of Statistics" (`http://www.econ.uiuc.edu/~roger/dict.html`), s.v. "Econometrics".

sumed earlier, to dependent data. It specifically considers models of time series, and time series data analysis, and simulation-based inference for complex or analytically-intractable models.

Parts III and IV are mostly independent of each other, but both rely on Parts I and II.

The online appendices contain a number of optional topics omitted from the main text in the interest of length, some mathematical reminders, and advice on writing R code for data analysis.

R Examples

The book is full of worked computational examples in R. In most cases, the code used to make figures, tables, etc., is given in full in the text. (The code is deliberately omitted for a few examples for pedagogical reasons.) To save space, comments are generally omitted from the text, but comments are vital to good programming (§J.9.1), so fully-commented versions of the code for each chapter are available from the book's website.

Problems

There are two kinds of problems included here. Mathematical and computational exercises go at the end of chapters, since they are mostly connected to those pieces of content. (Many of them are complements to, or fill in details of, material in the chapters.) There are also data-centric assignments, consisting of extended problem sets, in the companion document. Most of these draw on material from multiple chapters, and many of them are based on specific papers.

Solutions will be available to teachers from the publisher; giving them out to those using the book for self-study is, sadly, not feasible.

To Teachers

?

The usual one-semester course for this class has contained Chapters 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 15, 16, 17, 18, 19, 20, 21, 22 and 23, and Appendix and J (the latter quite early on). Other chapters and appendices have rotated in and out from year to year. One of the problem sets from Appendix 24.3 (or a similar one) was due every week, either as homework or as a take-home exam.

Corrections and Updates

The page for this book is `http://www.stat.cmu.edu/~cshalizi/ADAfaEPoV/`. The latest version will live there. The book will eventually be published by Cambridge University Press, at which point there will still be a free next-to-final draft at that URL, and errata. While the book is still in a draft, the PDF contains notes to myself for revisions, [[like so]]; you can ignore them.

[[Also marginal notes-to-self]]

Concepts You Should Know

If more than a few of these are unfamiliar, it's unlikely you're ready for this book.

LINEAR ALGEBRA: Vectors; arithmetic with vectors; inner or dot product of vectors, orthogonality; linear independence; basis vectors. Linear subspaces. Matrices, matrix arithmetic, multiplying vectors and matrices; geometric meaning of matrix multiplication. Eigenvalues and eigenvectors of matrices. Projection.

CALCULUS: Derivative, integral; fundamental theorem of calculus. Multivariable extensions: gradient, Hessian matrix, multidimensional integrals. Finding minima and maxima with derivatives. Taylor approximations (App. B).

PROBABILITY: Random variable; distribution, population, sample. Cumulative distribution function, probability mass function, probability density function. Specific distributions: Bernoulli, binomial, Poisson, geometric, Gaussian, exponential, t, Gamma. Expectation value. Variance, standard deviation.

Joint distribution functions. Conditional distributions; conditional expectations and variances. Statistical independence and dependence. Covariance and correlation; why dependence is not the same thing as correlation. Rules for arithmetic with expectations, variances and covariances. Laws of total probability, total expectation, total variation. Sequences of random variables. Stochastic process. Law of large numbers. Central limit theorem.

STATISTICS: Sample mean, sample variance. Median, mode. Quartile, percentile, quantile. Inter-quartile range. Histograms. Contingency tables; odds ratio, log odds ratio.

Parameters; estimator functions and point estimates. Sampling distribution. Bias of an estimator. Standard error of an estimate; standard error of the mean; how and why the standard error of the mean differs from the standard deviation. Consistency of estimators. Confidence intervals and interval estimates.

Hypothesis tests. Tests for differences in means and in proportions; Z and t tests; degrees of freedom. Size, significance, power. Relation between hypothesis tests and confidence intervals. χ^2 test of independence for contingency tables; degrees of freedom. KS test for goodness-of-fit to distributions.

Likelihood. Likelihood functions. Maximum likelihood estimates. Relation between confidence intervals and the likelihood function. Likelihood ratio test.

REGRESSION: What a linear model is; distinction between the regressors and the regressand. Predictions/fitted values and residuals of a regression. Interpretation of regression coefficients. Least-squares estimate of coefficients. Relation between maximum likelihood, least squares, and Gaussian distributions. Matrix formula for estimating the coefficients; the hat matrix for finding fitted values. R^2; why adding more predictor variables never reduces R^2. The t-test for the significance of individual coefficients given other coefficients. The F-test and partial F-test for the significance of groups of coefficients. Degrees of freedom for residuals. Diagnostic examination of residuals. Confidence intervals for parameters. Confidence intervals for fitted values. Prediction intervals. (Most of this material is reviewed at `http://www.stat.cmu.edu/~cshalizi/TALR/`.)

Part I

Regression and Its Generalizations

1

Regression: Predicting and Relating Quantitative Features

1.1 Statistics, Data Analysis, Regression

Statistics is the branch of mathematical engineering which designs and analyses methods for drawing reliable inferences from imperfect data.

The subject of most sciences is some aspect of the world around us, or within us. Psychology studies minds; geology studies the Earth's composition and form; economics studies production, distribution and exchange; mycology studies mushrooms. Statistics does not study the world, but some of the ways we try to understand the world — some of the intellectual tools of the other sciences. Its utility comes indirectly, through helping those other sciences.

This utility is very great, because all the sciences have to deal with imperfect data. Data may be imperfect because we can only observe and record a small fraction of what is relevant; or because we can only observe indirect signs of what is truly relevant; or because, no matter how carefully we try, our data always contain an element of noise. Over the last two centuries, statistics has come to handle all such imperfections by modeling them as random processes, and probability has become so central to statistics that we introduce random events deliberately (as in sample surveys).[1]

Statistics, then, uses probability to model inference from data. We try to mathematically understand the properties of different procedures for drawing inferences: Under what conditions are they reliable? What sorts of errors do they make, and how often? What can they tell us when they work? What are signs that something has gone wrong? Like other branches of engineering, statistics aims not just at understanding but also at improvement: we want to analyze data *better*: more reliably, with fewer and smaller errors, under broader conditions, faster, and with less mental effort. Sometimes some of these goals conflict — a fast, simple method might be very error-prone, or only reliable under a narrow range of circumstances.

One of the things that people most often want to know about the world is how different variables are related to each other, and one of the central tools statistics has for learning about relationships is regression.[2] In your linear regression class,

[1] Two excellent, but very different, histories of how statistics came to this understanding are Hacking (1990) and Porter (1986).

[2] The origin of the name is instructive (Stigler, 1986). It comes from 19th century investigations into the relationship between the attributes of parents and their children. People who are taller (heavier, faster, . . .) than average tend to have children who are also taller than average, but not *quite* as tall.

you learned about how it could be used in data analysis, and learned about its properties. In this book, we will build on that foundation, extending beyond basic linear regression in many directions, to answer many questions about how variables are related to each other.

This is intimately related to prediction. Being able to make predictions isn't the *only* reason we want to understand relations between variables — we also want to answer "what if?" questions — but prediction *tests* our knowledge of relations. (If we misunderstand, we might still be able to predict, but it's hard to see how we could understand and *not* be able to predict.) So before we go beyond linear regression, we will first look at prediction, and how to predict one variable from nothing at all. Then we will look at predictive relationships between variables, and see how linear regression is just one member of a big family of smoothing methods, all of which are available to us.

1.2 Guessing the Value of a Random Variable

We have a quantitative, numerical variable, which we'll imaginatively call Y. We'll suppose that it's a random variable, and try to predict it by guessing a single value for it. (Other kinds of predictions are possible — we might guess whether Y will fall within certain limits, or the probability that it does so, or even the whole probability distribution of Y. But some lessons we'll learn here will apply to these other kinds of predictions as well.) What is the best value to guess? More formally, what is the **optimal point forecast** for Y?

To answer this question, we need to pick a function to be optimized, which should measure how good our guesses are — or equivalently how bad they are, i.e., how big an error we're making. A reasonable, traditional starting point is the **mean squared error**:

$$\mathrm{MSE}(m) \equiv \mathbb{E}\left[(Y - m)^2\right] \tag{1.1}$$

So we'd like to find the value μ where $\boxed{\mathrm{MSE}(m)}$ is smallest. Start by re-writing the MSE as a (squared) bias plus a variance:

$$\begin{aligned}
\mathrm{MSE}(m) &= \mathbb{E}\left[(Y - m)^2\right] && (1.2)\\
&= (\mathbb{E}[Y - m])^2 + \mathbb{V}[Y - m] && (1.3)\\
&= (\mathbb{E}[Y - m])^2 + \mathbb{V}[Y] && (1.4)\\
&= (\mathbb{E}[Y] - m)^2 + \mathbb{V}[Y] && (1.5)
\end{aligned}$$

Notice that only the first, bias-squared term depends on our prediction m. We want to find the derivative of the MSE with respect to our prediction m, and

Likewise, the children of unusually short parents also tend to be closer to the average, and similarly for other traits. This came to be called "regression towards the mean," or even "regression towards mediocrity"; hence the line relating the average height (or whatever) of children to that of their parents was "the regression line," and the word stuck.

confusing
notation.
$\mu = \hat{m}$

then set that to zero at the optimal prediction
$TrueRegFunc$:

trivial

$\mathbb{R} \to \mathbb{R}$

$$\frac{d\text{MSE}}{dm} = -2\left(\mathbb{E}[Y] - m\right) + 0 \tag{1.6}$$

$$\left.\frac{d\text{MSE}}{dm}\right|_{m=\mu} = 0 \tag{1.7}$$

$$2(\mathbb{E}[Y] - \mu) = 0 \tag{1.8}$$

$$\mu = \mathbb{E}[Y] \tag{1.9}$$

So, if we gauge the quality of our prediction by mean-squared error, the best prediction to make is the expected value.

1.2.1 Estimating the Expected Value

Of course, to make the prediction $\mathbb{E}[Y]$ we would have to know the expected value of Y. Typically, we do not. However, if we have sampled values, $y_1, y_2, \ldots y_n$, we can estimate the expectation from the sample mean:

$$\widehat{\mu} \equiv \frac{1}{n}\sum_{i=1}^{n} y_i \tag{1.10}$$

If the samples are independent and identically distributed (IID), then the law of large numbers tells us that

$$\widehat{\mu} \to \mathbb{E}[Y] = \mu \tag{1.11}$$

and algebra with variances (Exercise 1.1) tells us something about how fast the convergence is, namely that the squared error will typically be $\mathbb{V}[Y]/n$.

Of course the assumption that the y_i come from IID samples is a strong one, but we can assert pretty much the same thing if they're just uncorrelated with a common expected value. Even if they are correlated, but the correlations decay fast enough, all that changes is the rate of convergence (§23.2.2.1). So "sit, wait, and average" is a pretty reliable way of estimating the expectation value.

1.3 The Regression Function

Of course, it's not very useful to predict *just one number* for a variable. Typically, we have lots of variables in our data, and we believe they are related *somehow*. For example, suppose that we have data on two variables, X and Y, which might look like Figure 1.1.[3] The feature Y is what we are trying to predict, a.k.a. the **dependent variable** or **output** or **response** or **regressand**, and X is the **predictor** or **independent variable** or **covariate** or **input** or **regressor**. Y might be something like the profitability of a customer and X their credit rating, or, if you want a less mercenary example, Y could be some measure of

[3] Problem set 27 features data that looks rather like these made-up values.

improvement in blood cholesterol and X the dose taken of a drug. Typically we won't have just one input feature X but rather many of them, but that gets harder to draw and doesn't change the points of principle.

Figure 1.2 shows the same data as Figure 1.1, only with the sample mean added on. This clearly tells us something about the data, but also it seems like we should be able to do better — to reduce the average error — by using X, rather than by ignoring it.

non-trivial
$\mathbb{R} \to \mathbb{R}$
(one-to-one?)

Let's say that the we want our prediction to be a *function* of X, namely $f(X)$. What should that function be, if we still use mean squared error? We can work this out by using the law of total expectation, i.e., the fact that $\mathbb{E}[U] = \mathbb{E}[\mathbb{E}[U|V]]$ for any random variables U and V.

$$\begin{aligned} \mathrm{MSE}(f) &= \mathbb{E}\left[(Y - f(X))^2\right] & (1.12) \\ &= \mathbb{E}\left[\mathbb{E}\left[(Y - f(X))^2 | X\right]\right] & (1.13) \\ &= \mathbb{E}\left[\mathbb{V}\left[Y - f(X)|X\right] + \left(\mathbb{E}\left[Y - f(X)|X\right]\right)^2\right] & (1.14) \\ &= \mathbb{E}\left[\mathbb{V}\left[Y|X\right] + \left(\mathbb{E}\left[Y - f(X)|X\right]\right)^2\right] & (1.15) \end{aligned}$$

When we want to minimize this, the first term inside the expectation doesn't depend on our prediction, and the second term looks just like our previous optimization only with all expectations conditional on X, so for our optimal function $\mu(x)$ we get

$$\mu(x) = \mathbb{E}[Y|X = x] \tag{1.16}$$

In other words, the (mean-squared) optimal *conditional* prediction is just the conditional expected value. The function $\mu(x)$ is called the **regression function**. This is what we would like to know when we want to predict Y.

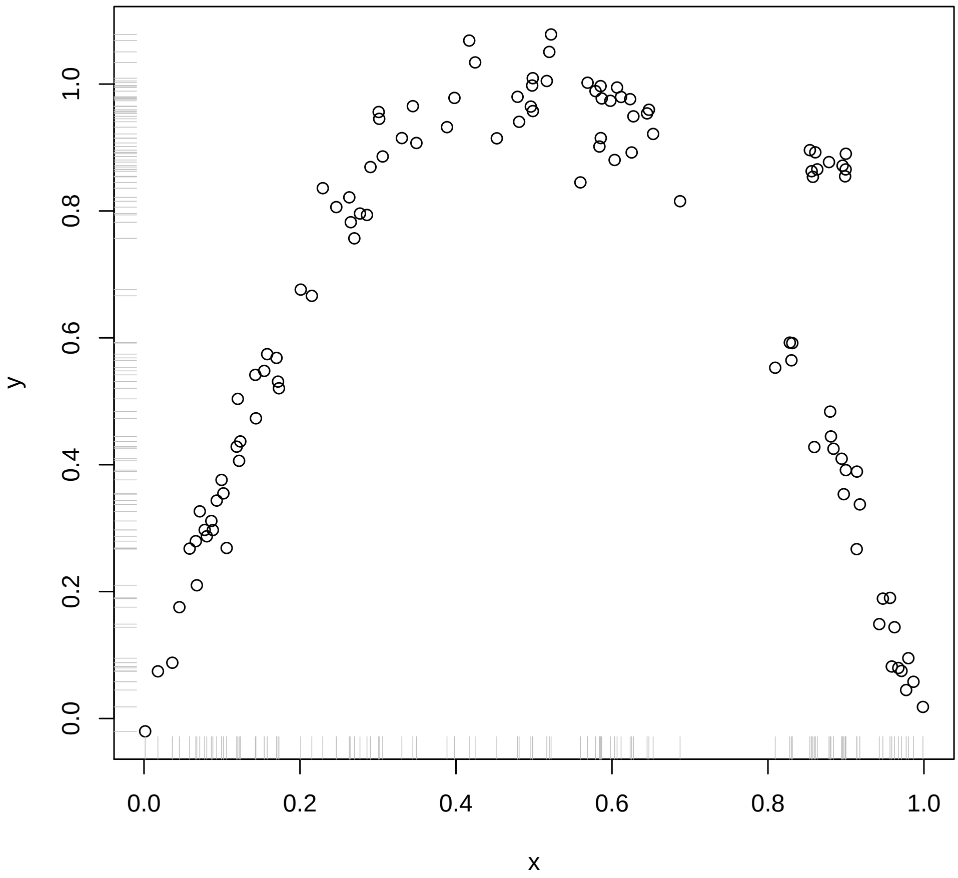
Some Disclaimers

It's important to be clear on what is and is not being assumed here. Talking about X as the "independent variable" and Y as the "dependent" one suggests a causal model, which we might write

$$Y \leftarrow \mu(X) + \epsilon \tag{1.17}$$

(const variance)

where the direction of the arrow, $\leftarrow$, indicates the flow from causes to effects, and ϵ is some noise variable. If the gods of inference are very kind, then ϵ would have a fixed distribution, independent of X, and we could without loss of generality take it to have mean zero. ("Without loss of generality" because if it has a non-zero mean, we can incorporate that into $\mu(X)$ as an additive constant.) However, *no* such assumption is required to get Eq. 1.16. It works when predicting effects from causes, or the other way around when predicting (or "retrodicting") causes from effects, or indeed when there is no causal relationship whatsoever between X and



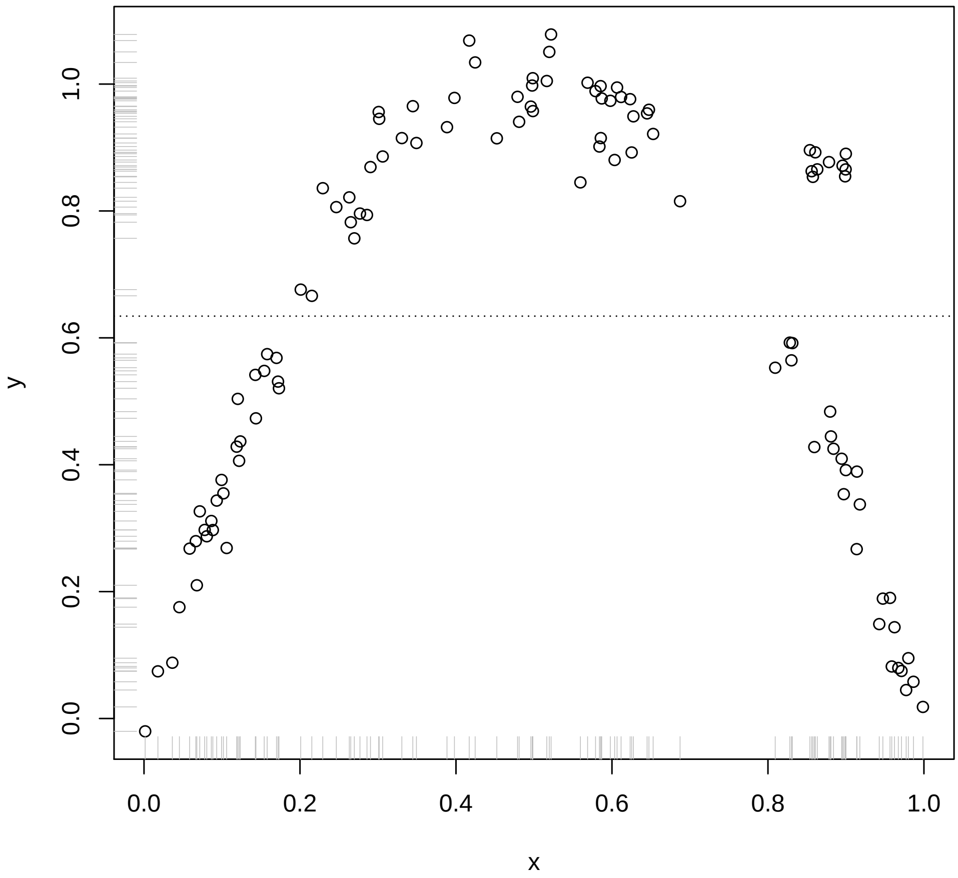

```
plot(all.x, all.y, xlab = "x", ylab = "y")
rug(all.x, side = 1, col = "grey")
rug(all.y, side = 2, col = "grey")
```

Figure 1.1 Scatterplot of the (made up) running example data. `rug()` adds horizontal and vertical ticks to the axes to mark the location of the data; this isn't necessary but is often helpful. The data are in the `basics-examples.Rda` file.

Y[4]. It *is* always true that

$$Y|X = \mu(X) + \epsilon(X) \tag{1.18}$$

[4] We will cover causal inference in detail in Part III.




```
plot(all.x, all.y, xlab = "x", ylab = "y")
rug(all.x, side = 1, col = "grey")
rug(all.y, side = 2, col = "grey")
abline(h = mean(all.y), lty = "dotted")
```

Figure 1.2 Data from Figure 1.1, with a horizontal line at $\overline{y}$.

where $\epsilon(X)$ is a random variable with expected value 0, $\mathbb{E}[\epsilon|X = x] = 0$, but as the notation indicates the distribution of this variable generally depends on X.

It's also important to be clear that if we find the regression function is a constant, $\mu(x) = \mu_0$ for all x, that this does not mean that X and Y are statistically independent. If they are independent, then the regression function is a constant, but turning this around is the logical fallacy of "affirming the consequent"[5].

1.4 Estimating the Regression Function

We want the regression function $\mu(x) = \mathbb{E}[Y|X = x]$, but what we have is a pile of training examples, of pairs $(x_1, y_1), (x_2, y_2), \ldots (x_n, y_n)$. What should we do?

If X takes on only a finite set of values, then a simple strategy is to use the conditional sample means:

$$\widehat{\mu}(x) = \frac{1}{\#\{i : x_i = x\}} \sum_{i:x_i=x} y_i \tag{1.19}$$

esp in high dims

Reasoning with the law of large numbers as before, we can be confident that $\widehat{\mu}(x) \to \mathbb{E}[Y|X = x]$.

Unfortunately, this *only* works when X takes values in a finite set. If X is continuous, then in general the probability of our getting a sample at any *particular* value is zero, as is the probability of getting *multiple* samples at *exactly* the same value of x. This is a basic issue with estimating any kind of function from data — the function will always be **undersampled**, and we need to fill in between the values we see. We also need to somehow take into account the fact that each y_i is a *sample* from the conditional distribution of $Y|X = x_i$, and generally not equal to $\mathbb{E}[Y|X = x_i]$. So any kind of function estimation is going to involve interpolation, extrapolation, and de-noising or smoothing.

Different methods of estimating the regression function — different regression methods, for short — involve different choices about how we interpolate, extrapolate and smooth. These are choices about how to approximate $\mu(x)$ with a limited class of functions which we know (or at least hope) we can estimate. There is no guarantee that our choice leads to a *good* approximation in the case at hand, though it is sometimes possible to say that the approximation error will shrink as we get more and more data. This is an extremely important topic and deserves an extended discussion, coming next.

1.4.1 The Bias-Variance Trade-off

Suppose that the true regression function is $\mu(x)$, but we use the function $\widehat{\mu}$ to make our predictions. Let's look at the mean squared error at $X = x$ in a slightly different way than before, which will make it clearer what happens when we can't use μ to make predictions. We'll begin by expanding $\boxed{(Y - \widehat{\mu}(x))^2}$, since the MSE at x is just the expectation of this.

$$(Y - \widehat{\mu}(x))^2 \tag{1.20}$$
$$= (Y - \mu(x) + \mu(x) - \widehat{\mu}(x))^2$$
$$= (Y - \mu(x))^2 + 2(Y - \mu(x))(\mu(x) - \widehat{\mu}(x)) + (\mu(x) - \widehat{\mu}(x))^2 \tag{1.21}$$

[5] As in combining the fact that all human beings are featherless bipeds, and the observation that a cooked turkey is a featherless biped, to conclude that cooked turkeys are human beings.

Eq. 1.18 tells us that $Y - \mu(X) = \epsilon$, a random variable which has expectation zero (and is uncorrelated with X). Taking the expectation of Eq. 1.21, nothing happens to the last term (since it doesn't involve any random quantities); the middle term goes to zero (because $\mathbb{E}[Y - \mu(X)] = \mathbb{E}[\epsilon] = 0$), and the first term becomes the variance of ϵ, call it $\sigma^2(x)$:

$$\mathrm{MSE}(\widehat{\mu}(x)) = \sigma^2(x) + (\mu(x) - \widehat{\mu}(x))^2 \tag{1.22}$$

The $\sigma^2(x)$ term doesn't depend on our prediction function, just on how hard it is, intrinsically, to predict Y at $X = x$. The second term, though, is the extra error we get from not knowing μ. (Unsurprisingly, ignorance of μ cannot *improve* our predictions.) This is our first **bias-variance decomposition**: the total MSE at x is decomposed into a (squared) bias $\mu(x) - \widehat{\mu}(x)$, the amount by which our predictions are *systematically* off, and a variance $\sigma^2(x)$, the unpredictable, "statistical" fluctuation around even the best prediction.

All this presumes that $\widehat{\mu}$ is a single fixed function. Really, of course, $\widehat{\mu}$ is something we estimate from earlier data. But if those data are random, the regression function we get is random too; let's call this random function $\widehat{M}_n$, where the subscript reminds us of the finite amount of data we used to estimate it. What we have analyzed is really $\mathrm{MSE}(\widehat{M}_n(x)|\widehat{M}_n = \widehat{\mu})$, the mean squared error *conditional on* a particular estimated regression function. What can we say about the prediction error of the *method*, averaging over all the possible training data sets?

$$\begin{aligned}
\mathrm{MSE}(\widehat{M}_n(x)) &= \mathbb{E}\left[(Y - \widehat{M}_n(X))^2 | X = x\right] && (1.23)\\
&= \mathbb{E}\left[\mathbb{E}\left[(Y - \widehat{M}_n(X))^2 | X = x, \widehat{M}_n = \widehat{\mu}\right] | X = x\right] && (1.24)\\
&= \mathbb{E}\left[\sigma^2(x) + (\mu(x) - \widehat{M}_n(x))^2 | X = x\right] && (1.25)\\
&= \sigma^2(x) + \mathbb{E}\left[(\mu(x) - \widehat{M}_n(x))^2 | X = x\right] && (1.26)\\
&= \sigma^2(x) + \mathbb{E}\left[(\mu(x) - \mathbb{E}\left[\widehat{M}_n(x)\right] + \mathbb{E}\left[\widehat{M}_n(x)\right] - \widehat{M}_n(x))^2\right] && (1.27)\\
&= \sigma^2(x) + \left(\mu(x) - \mathbb{E}\left[\widehat{M}_n(x)\right]\right)^2 + \mathbb{V}\left[\widehat{M}_n(x)\right] && (1.28)
\end{aligned}$$

This is our second bias-variance decomposition — I pulled the same trick as before, adding and subtracting a mean inside the square. The first term is just the variance of the process; we've seen that before and it isn't, for the moment, of any concern. The second term is the bias in using $\widehat{M}_n$ to estimate μ — the **approximation bias or approximation error**. The third term, though, is the variance in our *estimate* of the regression function. Even if we have an unbiased *method* ($\mu(x) = \mathbb{E}\left[\widehat{M}_n(x)\right]$), if there is a lot of variance in our estimates, we can expect to make large errors.

The approximation bias depends on the true regression function. For example, if $\mathbb{E}\left[\widehat{M}_n(x)\right] = 42 + 37x$, the error of approximation will be zero at all x if $\mu(x) = 42+37x$, but it will be larger and x-dependent if $\mu(x) = 0$. However, there are flexible methods of estimation which will have small approximation biases for *all* μ in a broad range of regression functions. The catch is that, at least past a certain point, decreasing the approximation bias can only come through increasing the estimation variance. This is the **bias-variance trade-off**. However, nothing says that the trade-off has to be one-for-one. Sometimes we can lower the total error by *introducing* some bias, since it gets rid of more variance than it adds approximation error. The next section gives an example.

In general, both the approximation bias and the estimation variance depend on n. A method is **consistent**[6] when both of these go to zero as $n \to \infty$ — that is, if we recover the true regression function as we get more and more data.[7] Again, consistency depends not just on the method, but also on how well the method matches the data-generating process, and, again, there is a bias-variance trade-off. There can be multiple consistent methods for the same problem, and their biases and variances don't have to go to zero at the same *rates*.

EFFICIENCY

1.4.2 The Bias-Variance Trade-Off in Action

Let's take an extreme example: we could decide to approximate $\mu(x)$ by a constant μ_0. The implicit smoothing here is very strong, but sometimes appropriate. For instance, it's appropriate when $\mu(x)$ really is a constant! Then trying to estimate any additional structure in the regression function is just wasted effort. Alternately, if $\mu(x)$ is *nearly* constant, we may still be better off approximating it as one. For instance, suppose the true $\mu(x) = \mu_0 + a \sin(\nu x)$, where $a \ll 1$ and $\nu \gg 1$ (Figure 1.3 shows an example). With limited data, we can actually get better predictions by estimating a constant regression function than one with the correct functional form.

cf Ludwig et al. / asymptotic approx

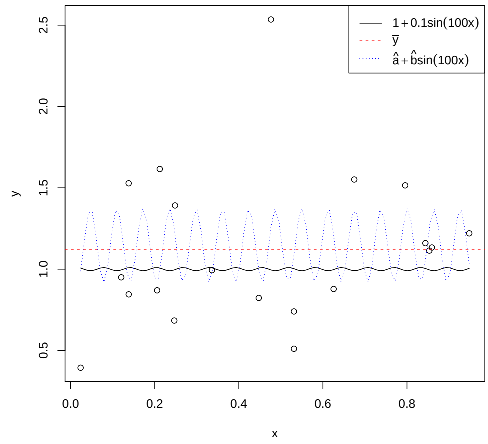
1.4.3 Ordinary Least Squares Linear Regression as Smoothing

Let's revisit ordinary least-squares linear regression from this point of view. We'll assume that the predictor variable X is one-dimensional, just to simplify the book-keeping.

We *choose* to approximate $\mu(x)$ by $b_0 + b_1 x$, and ask for the best values β_0, β_1 of

[6] To be precise, **consistent for** μ, or **consistent for conditional expectations**. More generally, an estimator of any property of the data, or of the whole distribution, is consistent if it converges on the truth.

[7] You might worry about this claim, especially if you've taken more probability theory — aren't we just saying something about average performance of the $\widehat{M}_n$, rather than any *particular* estimated regression function? But notice that if the estimation variance goes to zero, then by Chebyshev's inequality, $\Pr(|X - \mathbb{E}[X]| \geq a) \leq \mathbb{V}[X]/a^2$, each $\widehat{M}_n(x)$ comes arbitrarily close to $\mathbb{E}\left[\widehat{M}_n(x)\right]$ with arbitrarily high probability. If the approximation bias goes to zero, therefore, the estimated regression functions converge *in probability* on the true regression function, not just *in mean.*




```
ugly.func <- function(x) {
    1 + 0.01 * sin(100 * x)
}
x <- runif(20)
y <- ugly.func(x) + rnorm(length(x), 0, 0.5)
plot(x, y, xlab = "x", ylab = "y")
curve(ugly.func, add = TRUE)
abline(h = mean(y), col = "red", lty = "dashed")
sine.fit = lm(y ~ 1 + sin(100 * x))
curve(sine.fit$coefficients[1] + sine.fit$coefficients[2] * sin(100 * x), col = "blue",
    add = TRUE, lty = "dotted")
legend("topright", legend = c(expression(1 + 0.1 * sin(100 * x)), expression(bar(y)),
    expression(hat(a) + hat(b) * sin(100 * x))), lty = c("solid", "dashed", "dotted"),
    col = c("black", "red", "blue"))
```

Figure 1.3 When we try to estimate a rapidly-varying but small-amplitude regression function (solid black line, $\mu = 1 + 0.01 \sin 100x + \epsilon$, with mean-zero Gaussian noise of standard deviation 0.5), we can do better to use a constant function (red dashed line at the sample mean) than to estimate a more complicated model of the *correct* functional form $\hat{a} + \hat{b} \sin 100x$ (dotted blue line). With just 20 observations, the mean predicts slightly better on new data (square-root MSE, RMSE, of 0.52) than does the estimate sine function (RMSE of 0.55). The bias of using the wrong functional form is less than the extra variance of estimation, so using the true model form hurts us.

those constants. These will be the ones which minimize the mean-squared error.

$$\begin{aligned}
\mathrm{MSE}(a, b) &= \mathbb{E}\left[(Y - b_0 - b_1 X)^2\right] && (1.29)\\
&= \mathbb{E}\left[\mathbb{E}\left[(Y - b_0 - b_1 X)^2 | X\right]\right] && (1.30)\\
&= \mathbb{E}\left[\mathbb{V}\left[Y|X\right] + \left(\mathbb{E}\left[Y - b_0 - b_1 X | X\right]\right)^2\right] && (1.31)\\
&= \mathbb{E}\left[\mathbb{V}\left[Y|X\right]\right] + \mathbb{E}\left[\left(\mathbb{E}\left[Y - b_0 - b_1 X | X\right]\right)^2\right] && (1.32)
\end{aligned}$$

The first term doesn't depend on b_0 or b_1, so we can drop it for purposes of optimization. Taking derivatives, and then bringing them inside the expectations,

equiv to $Y - X\beta$ form

$$\frac{\partial \mathrm{MSE}}{\partial b_0} = \mathbb{E}\left[2(Y - b_0 - b_1 X)(-1)\right] \tag{1.33}$$

$$0 = \mathbb{E}\left[Y - \beta_0 - \beta_1 X\right] \tag{1.34}$$

$$\beta_0 = \mathbb{E}\left[Y\right] - \beta_1 \mathbb{E}\left[X\right] \tag{1.35}$$

So we need to get β_1:

$$\frac{\partial \mathrm{MSE}}{\partial b_1} = \mathbb{E}\left[2(Y - b_0 - b_1 X)(-X)\right] \tag{1.36}$$

$$0 = \mathbb{E}\left[XY\right] - \beta_1 \mathbb{E}\left[X^2\right] + (\mathbb{E}\left[Y\right] - \beta_1 \mathbb{E}\left[X\right])\mathbb{E}\left[X\right] \tag{1.37}$$

$$= \mathbb{E}\left[XY\right] - \mathbb{E}\left[X\right]\mathbb{E}\left[Y\right] - \beta_1(\mathbb{E}\left[X^2\right] - \mathbb{E}\left[X\right]^2) \tag{1.38}$$

$$\beta_1 = \frac{\mathrm{Cov}\left[X, Y\right]}{\mathbb{V}\left[X\right]} \tag{1.39}$$

using our equation for β_0. That is, the mean-squared optimal *linear* prediction is

$$\mu(x) = \mathbb{E}\left[Y\right] + \frac{\mathrm{Cov}\left[X, Y\right]}{\mathbb{V}\left[X\right]}(x - \mathbb{E}\left[X\right]) \tag{1.40}$$

Now, if we try to estimate this from data, there are (at least) two approaches. One is to replace the true, population values of the covariance and the variance with their sample values, respectively

$$\frac{1}{n}\sum_i (y_i - \overline{y})(x_i - \overline{x}) \tag{1.41}$$

and

$$\frac{1}{n}\sum_i (x_i - \overline{x})^2 \equiv \widehat{\mathbb{V}}\left[X\right]. \tag{1.42}$$

$\frac{1}{n-1}$? bias vs ...

The other is to minimize the **in-sample** or **empirical** mean squared error,

$$\frac{1}{n}\sum_i (y_i - b_0 - b_1 x_i)^2 \tag{1.43}$$

You may or may not find it surprising that both approaches lead to the same answer:

$$\widehat{\beta_1} = \frac{\frac{1}{n}\sum_i (y_i - \overline{y})(x_i - \overline{x})}{\widehat{\mathbb{V}}\left[X\right]} \tag{1.44}$$

$$\widehat{\beta_0} = \overline{y} - \widehat{\beta_1}\overline{x} \tag{1.45}$$

$$\tag{1.46}$$

Provided that $\mathbb{V}\left[X\right] > 0$, these will converge with IID samples, so we have a consistent estimator.

We are now in a position to see how the least-squares linear regression model

need finite second moments???

is really a weighted averaging of the data. Let's write the estimated regression function explicitly in terms of the training data points.

✱

$$
\begin{aligned}
\widehat{\mu}(x) &= \widehat{\beta_0} + \widehat{\beta_1} x & (1.47)\\
&= \overline{y} + \widehat{\beta_1}(x - \overline{x}) & (1.48)\\
&= \frac{1}{n}\sum_{i=1}^{n} y_i + \left(\frac{\frac{1}{n}\sum_i (y_i - \overline{y})(x_i - \overline{x})}{\frac{1}{n}\sum_i (x_i - \overline{x})^2}\right)(x - \overline{x}) & (1.49)\\
&= \frac{1}{n}\sum_{i=1}^{n} y_i + \frac{(x - \overline{x})}{n\hat{\sigma}^2_X}\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y}) & (1.50)\\
&= \frac{1}{n}\sum_{i=1}^{n} y_i + \frac{(x - \overline{x})}{n\hat{\sigma}^2_X}\sum_{i=1}^{n}(x_i - \overline{x})y_i - \frac{(x - \overline{x})}{n\hat{\sigma}^2_X}(n\overline{x} - n\overline{x})\overline{y} & (1.51)\\
&= \sum_{i=1}^{n}\frac{1}{n}\left(1 + \frac{(x - \overline{x})(x_i - \overline{x})}{\hat{\sigma}^2_X}\right) y_i & (1.52)
\end{aligned}
$$

~ leverage

In words, our prediction is a weighted average of the observed values y_i of the regressand, where the weights are proportional to how far x_i and x both are from the center of the data (relative to the variance of X). If x_i is on the same side of the center as x, it gets a positive weight, and if it's on the opposite side it gets a negative weight.

Figure 1.4 adds the least-squares regression line to Figure 1.1. As you can see, this is only barely slightly different from the constant regression function (the slope is X is 0.014). Visually, the problem is that there should be a positive slope in the left-hand half of the data, and a negative slope in the right, but the slopes and the densities are balanced so that the best *single* slope is near zero.[8]

Mathematically, the problem arises from the peculiar way in which least-squares linear regression smoothes the data. As I said, the weight of a data point depends on how far it is from the *center* of the data, not how far it is from the *point at which we are trying to predict*. This works when $\mu(x)$ really is a straight line, but otherwise — e.g., here — it's a recipe for trouble. However, it does suggest that if we could somehow just tweak the way we smooth the data, we could do better than linear regression.

smooths ?

1.5 Linear Smoothers

The sample mean and the least-squares line are both special cases of **linear smoothers**, which estimates the regression function with a weighted average:

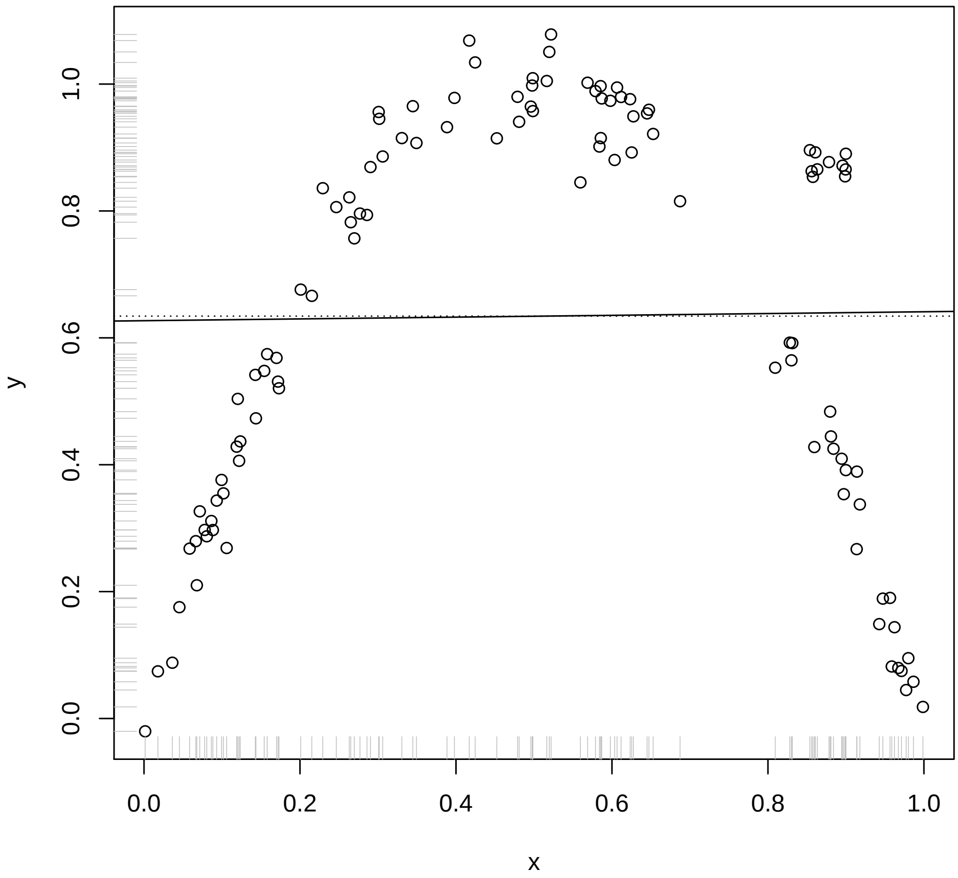
$$\widehat{\mu}(x) = \sum_i y_i \widehat{w}(x_i, x) \tag{1.53}$$

These are called linear smoothers because the predictions are linear in the *responses* y_i; as functions of x they can be and generally are nonlinear.

[8] The standard test of whether this coefficient is zero is about as far from rejecting the null hypothesis as you will ever see, $p = 0.89$. Remember this the next time you look at linear regression output.

CAREFUL! don't take magnitude of p as strength of effect

($p \sim U(0,1)$ if NH is true)




```
plot(all.x, all.y, xlab = "x", ylab = "y")
rug(all.x, side = 1, col = "grey")
rug(all.y, side = 2, col = "grey")
abline(h = mean(all.y), lty = "dotted")
fit.all = lm(all.y ~ all.x)
abline(fit.all)
```

Figure 1.4 Data from Figure 1.1, with a horizontal line at the mean (dotted) and the ordinary least squares regression line (solid).

As I just said, the sample mean is a special case; see Exercise 1.7. Ordinary linear regression is another special case, where $\widehat{w}(x_i, x)$ is given by Eq. 1.52. Both of these, as remarked earlier, ignore how far x_i is from x. Let us look at some linear smoothers which are not so silly.

1.5.1 k-Nearest-Neighbors Regression

At the other extreme from ignoring the distance between x_i and x, we could do **nearest-neighbor regression**:

$$\widehat{w}(x_i, x) = \begin{cases} 1 & x_i \text{ nearest neighbor of } x \\ 0 & \text{otherwise} \end{cases} \tag{1.54}$$

This is very sensitive to the distance between x_i and x. If $\mu(x)$ does not change too rapidly, and X is pretty thoroughly sampled, then the nearest neighbor of x among the x_i is probably close to x, so that $\mu(x_i)$ is probably close to $\mu(x)$. However, $y_i = \mu(x_i) + \text{noise}$, so nearest-neighbor regression will include the noise into its prediction. We might instead do k-nearest-neighbors regression,

$$\widehat{w}(x_i, x) = \begin{cases} 1/k & x_i \text{ one of the } k \text{ nearest neighbors of } x \\ 0 & \text{otherwise} \end{cases} \tag{1.55}$$

Again, with enough samples all the k nearest neighbors of x are probably close to x, so their regression functions there are going to be close to the regression function at x. But because we average their values of y_i, the noise terms should tend to cancel each other out. As we increase k, we get smoother functions — in the limit $k = n$ and we just get back the constant. Figure 1.5 illustrates this for our running example data.[9] To use k-nearest-neighbors regression, we need to pick k somehow. This means we need to decide *how much* smoothing to do, and this is not trivial. We will return to this point in Chapter 3.

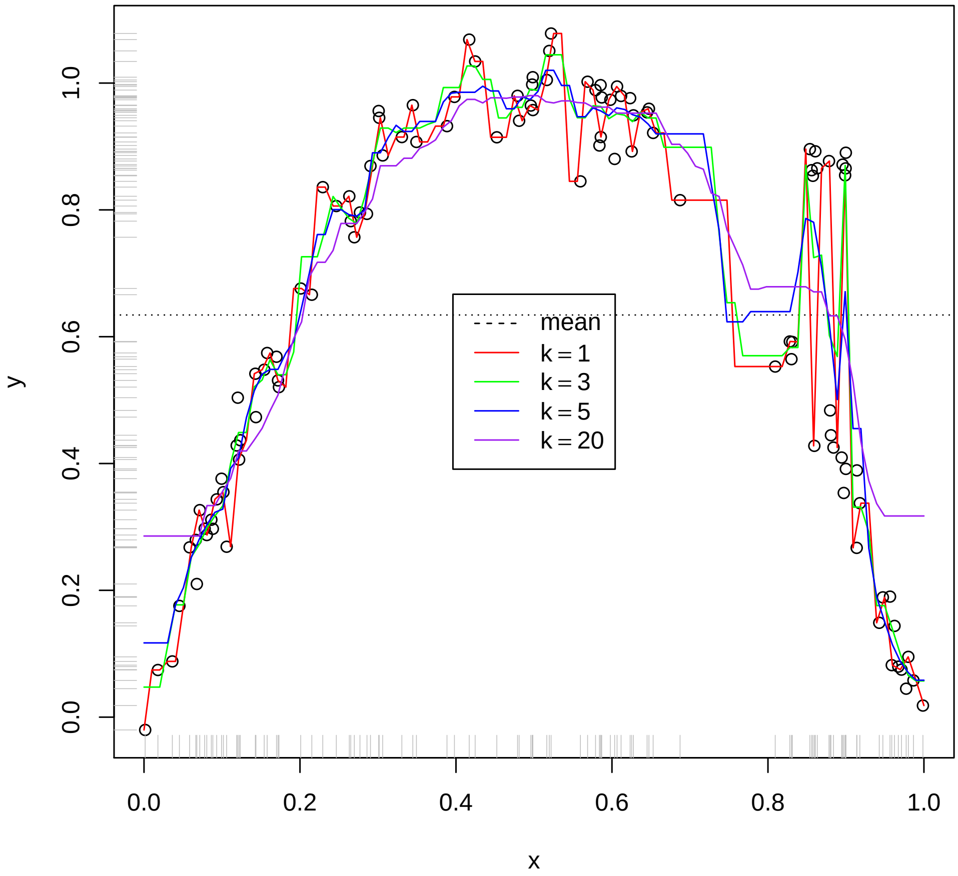
Because k-nearest-neighbors averages over only a fixed number of neighbors, each of which is a noisy sample, it always has some noise in its prediction, and is generally not consistent. This may not matter very much with moderately-large data (especially once we have a good way of picking k). If we want consistency, we need to let k grow with n, but not too fast; it's enough that as $n \to \infty$, $k \to \infty$ and $k/n \to 0$ (Györfi *et al.*, 2002, Thm. 6.1, p. 88).

1.5.2 Kernel Smoothers

Changing k in a k-nearest-neighbors regression lets us change how much smoothing we're doing on our data, but it's a bit awkward to express this in terms of a number of data points. It feels like it would be more natural to talk about a range in the independent variable over which we smooth or average. Another problem with k-NN regression is that each testing point is predicted using information from only a few of the training data points, unlike linear regression or the sample mean, which always uses all the training data. It'd be nice if we could somehow use all the training data, but in a location-sensitive way.

There are several ways to do this, as we'll see, but a particularly useful one is

[9] The code uses the k-nearest neighbor function provided by the package `FNN` (Beygelzimer *et al.*, 2013). This requires one to give both a set of training points (used to learn the model) and a set of test points (at which the model is to make predictions), and returns a list where the actual predictions are in the `pred` element — see `help(knn.reg)` for more, including examples.




```
library(FNN)
plot.seq <- matrix(seq(from = 0, to = 1, length.out = 100), byrow = TRUE)
lines(plot.seq, knn.reg(train = all.x, test = plot.seq, y = all.y, k = 1)$pred, col = "red")
lines(plot.seq, knn.reg(train = all.x, test = plot.seq, y = all.y, k = 3)$pred, col = "green")
lines(plot.seq, knn.reg(train = all.x, test = plot.seq, y = all.y, k = 5)$pred, col = "blue")
lines(plot.seq, knn.reg(train = all.x, test = plot.seq, y = all.y, k = 20)$pred,
    col = "purple")
legend("center", legend = c("mean", expression(k == 1), expression(k == 3), expression(k ==
    5), expression(k == 20)), lty = c("dashed", rep("solid", 4)), col = c("black",
    "red", "green", "blue", "purple"))
```

Figure 1.5 Points from Figure 1.1 with horizontal dashed line at the mean and the k-nearest-neighbors regression curves for various k. Increasing k smooths out the regression curve, pulling it towards the mean. — The code is repetitive; can you write a function to simplify it?

kernel smoothing, a.k.a. **kernel regression** or **Nadaraya-Watson regression**. To begin with, we need to pick a **kernel function**[10] $K(x_i, x)$ which satisfies the following properties:

1. $K(x_i, x) \geq 0$;
2. $K(x_i, x)$ depends only on the distance $x_i - x$, not the individual arguments;
3. $\int xK(0, x)dx = 0$; and
4. $0 < \int x^2 K(0, x)dx < \infty$.

These conditions together (especially the last one) imply that $K(x_i, x) \to 0$ as $|x_i - x| \to \infty$. Two examples of such functions are the density of the $\mathrm{Unif}(-h/2, h/2)$ distribution, and the density of the standard Gaussian $\mathcal{N}(0, \sqrt{h})$ distribution. Here h can be any positive number, and is called the **bandwidth**. Because $K(x_i, x) = K(0, x_i - x)$, we will often write K as a one-argument function, $K(x_i - x)$. Because we often want to consider similar kernels which differ only by bandwidth, we'll either write $K(\frac{x_i - x}{h})$, or $K_h(x_i - x)$.

Gaussian prob has some nice properties

The Nadaraya-Watson estimate of the regression function is

$$\widehat{\mu}(x) = \sum_i y_i \frac{K(x_i, x)}{\sum_j K(x_j, x)} \tag{1.56}$$

i.e., in terms of Eq. 1.53,

$$\widehat{w}(x_i, x) = \frac{K(x_i, x)}{\sum_j K(x_j, x)} \tag{1.57}$$

(Notice that here, as in k-NN regression, the sum of the weights is always 1. Why?)[11]

What does this achieve? Well, $K(x_i, x)$ is large if x_i is close to x, so this will place a lot of weight on the training data points close to the point where we are trying to predict. More distant training points will have smaller weights, falling off towards zero. If we try to predict at a point x which is very far from any of the training data points, the value of $K(x_i, x)$ will be small for all x_i, but it will typically be *much, much smaller* for all the x_i which are not the nearest neighbor of x, so $\widehat{w}(x_i, x) \approx 1$ for the nearest neighbor and ≈ 0 for all the others.[12] That is, far from the training data, our predictions will tend towards nearest neighbors, rather than going off to $\pm\infty$, as linear regression's predictions do. Whether this

[10] There are many other mathematical objects which are *also* called "kernels". Some of these meanings are related, but not all of them. (Cf. "normal".)

[11] What do we do if $K(x_i, x)$ is zero for some x_i? Nothing; they just get zero weight in the average. What do we do if *all* the $K(x_i, x)$ are zero? Different people adopt different conventions; popular ones are to return the global, unweighted mean of the y_i, to do some sort of interpolation from regions where the weights are defined, and to throw up our hands and refuse to make any predictions (computationally, return `NA`).

[12] Take a Gaussian kernel in one dimension, for instance, so $K(x_i, x) \propto e^{-(x_i - x)^2/2h^2}$. Say x_i is the nearest neighbor, and $|x_i - x| = L$, with $L \gg h$. So $K(x_i, x) \propto e^{-L^2/2h^2}$, a small number. But now for any other x_j, $K(x_i, x) \propto e^{-L^2/2h^2} e^{-(x_j - x_i)L/2h^2} e^{-(x_j - x_i)^2/2h^2} \ll e^{-L^2/2h^2}$. — This assumes that we're using a kernel like the Gaussian, which never quite goes to zero, unlike the box kernel.

R/Julia imp?

SOS package

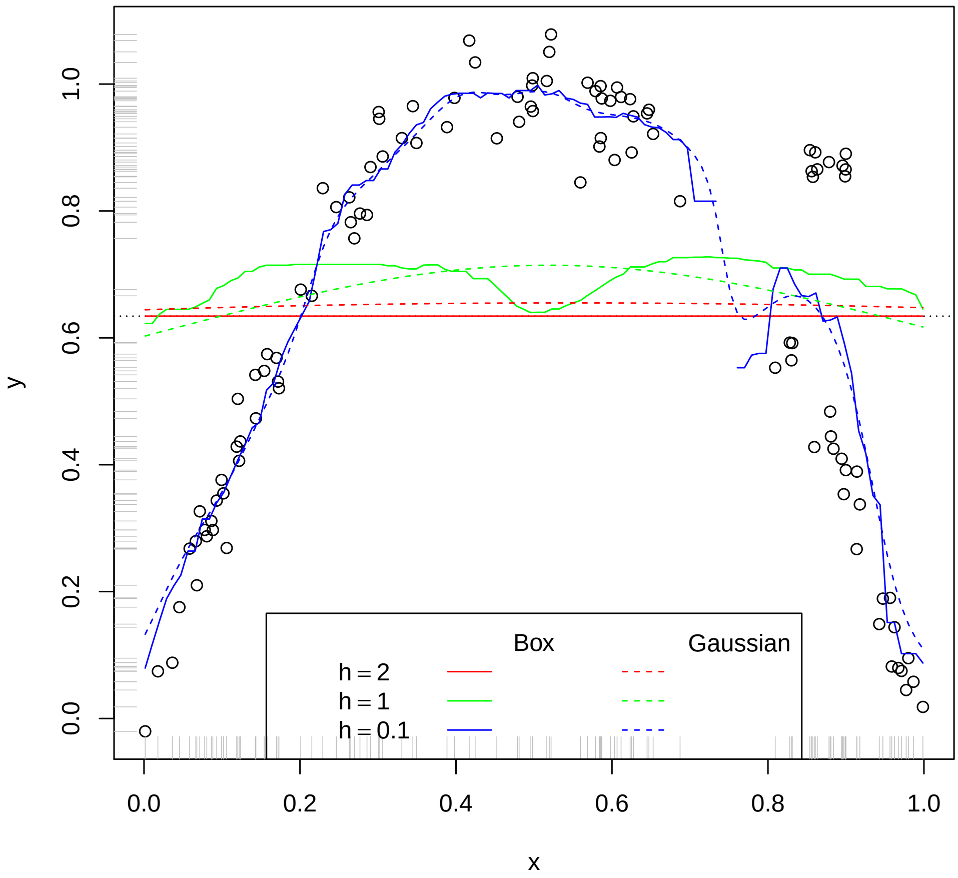
is good or bad of course depends on the true $\mu(x)$ — and how often we have to predict what will happen very far from the training data.

Figure 1.6 shows our running example data, together with kernel regression estimates formed by combining the uniform-density, or **box**, and Gaussian kernels with different bandwidths. The box kernel simply takes a region of width h around the point x and averages the training data points it finds there. The Gaussian kernel gives reasonably large weights to points within h of x, smaller ones to points within $2h$, tiny ones to points within $3h$, and so on, shrinking like $e^{-(x-x_i)^2/2h}$. As promised, the bandwidth h controls the degree of smoothing. As $h \to \infty$, we revert to taking the global mean. As $h \to 0$, we tend to get spikier functions — with the Gaussian kernel at least it tends towards the nearest-neighbor regression.

If we want to use kernel regression, we need to choose both which kernel to use, and the bandwidth to use with it. Experience, like Figure 1.6, suggests that the bandwidth usually matters a lot more than the kernel. This puts us back to roughly where we were with k-NN regression, needing to control the degree of smoothing, without knowing how smooth $\mu(x)$ really is. Similarly again, with a fixed bandwidth h, kernel regression is generally not consistent. However, if $h \to 0$ as $n \to \infty$, but doesn't shrink *too* fast, then we can get consistency.

slightly more surprising than inconsistency of K-NN.
(consistent for 'flat enough' kernel?)

preference for smoothness, 'full' support




```
lines(ksmooth(all.x, all.y, "box", bandwidth = 2), col = "red")
lines(ksmooth(all.x, all.y, "box", bandwidth = 1), col = "green")
lines(ksmooth(all.x, all.y, "box", bandwidth = 0.1), col = "blue")
lines(ksmooth(all.x, all.y, "normal", bandwidth = 2), col = "red", lty = "dashed")
lines(ksmooth(all.x, all.y, "normal", bandwidth = 1), col = "green", lty = "dashed")
lines(ksmooth(all.x, all.y, "normal", bandwidth = 0.1), col = "blue", lty = "dashed")
legend("bottom", ncol = 3, legend = c("", expression(h == 2), expression(h == 1),
    expression(h == 0.1), "Box", "", "", "", "Gaussian", "", "", ""), lty = c("blank",
    "blank", "blank", "blank", "blank", "solid", "solid", "solid", "blank", "dashed",
    "dashed", "dashed"), col = c("black", "black", "black", "black", "black", "red",
    "green", "blue", "black", "red", "green", "blue"), pch = NA)
```

Figure 1.6 Data from Figure 1.1 together with kernel regression lines, for various combinations of kernel (box/uniform or Gaussian) and bandwidth. Note the abrupt jump around $x = 0.75$ in the $h = 0.1$ box-kernel (solid blue) line — with a small bandwidth the box kernel is unable to interpolate smoothly across the break in the training data, while the Gaussian kernel (dashed blue) can.

1.5.3 Some General Theory for Linear Smoothers

Some key parts of the theory you are familiar with for linear regression models carries over more generally to linear smoothers. They are not quite so important any more, but they do have their uses, and they can serve as security objects during the transition to non-parametric regression.

Throughout this sub-section, we will temporarily assume that $Y = \mu(X) + \epsilon$, with the noise terms ϵ having constant variance σ^2, no correlation with the noise at other observations. Also, we will define the **smoothing**, **influence** or **hat matrix** $\hat{\mathbf{w}}$ by $\hat{w}_{ij} = \hat{w}(x_i, x_j)$. This records how much influence observation y_j had on the smoother's fitted value for $\mu(x_i)$, which (remember) is $\widehat{\mu}(x_i)$ or $\widehat{\mu}_i$ for short[13], hence the name "hat matrix" for $\hat{w}$.

cf hat matrix from lin reg.

1.5.3.1 Standard error of predicted mean values

It is easy to get the standard error of any predicted mean value $\widehat{\mu}(x)$, by first working out its variance:

$$\mathbb{V}\left[\widehat{\mu}(x)\right] = \mathbb{V}\left[\sum_{j=1}^{n} w(x_j, x) Y_j\right] \tag{1.58}$$

$$= \sum_{j=1}^{n} \mathbb{V}\left[w(x_j, x) Y_j\right] \tag{1.59}$$

$$= \sum_{j=1}^{n} w^2(x_j, x) \mathbb{V}\left[Y_j\right] \tag{1.60}$$

$$= \sigma^2 \sum_{j=1}^{n} w^2(x_j, x) \tag{1.61}$$

The second line uses the assumption that the noise is uncorrelated, and the last the assumption that the noise variance is constant. In particular, for a point x_i which appeared in the training data, $\mathbb{V}\left[\widehat{\mu}(x_i)\right] = \sigma^2 \sum_j w_{ij}^2$.

Notice that this is the variance in the predicted *mean* value, $\widehat{\mu}(x)$. It is not an estimate of $\mathbb{V}\left[Y \mid X = x\right]$, though we will see how conditional variances can be estimated using nonparametric regression in Chapter 10.

Notice also that we have *not* had to assume that the noise is Gaussian. If we did add that assumption, this formula would also give us a confidence interval for the fitted value (though we would still have to worry about estimating σ).

1.5.3.2 (Effective) Degrees of Freedom

For linear regression models, you will recall that the number of "degrees of freedom" was just the number of coefficients (including the intercept). While degrees of freedom are less important for other sorts of regression than for linear models, they're still worth knowing about, so I'll explain here how they are defined and

[13] This is often written as $\hat{y}_i$, but that's not very logical notation; the quantity is a *function* of y_i, not an estimate of it; it's an *estimate* of $\mu(x_i)$.

linreg as lin alg

calculated. In general, we can't use the number of parameters to define degrees of freedom, since most linear smoothers don't *have* parameters. Instead, we have to go back to the reasons *why* the number of parameters actually matters in ordinary linear models. (Linear algebra follows.)

We'll start with an $n \times p$ data matrix of predictor variables $\mathbf{x}$ (possibly including an all-1 column for an intercept), and an $n \times 1$ column matrix of response values $\mathbf{y}$. The ordinary least squares estimate of the p-dimensional coefficient vector β is

$$\hat{\beta} = \left(\mathbf{x}^T\mathbf{x}\right)^{-1}\mathbf{x}^T\mathbf{y} \tag{1.62}$$

This lets us write the fitted values in terms of $\mathbf{x}$ and $\mathbf{y}$:

$$\widehat{\mu} = \mathbf{x}\hat{\beta} \tag{1.63}$$

$$= \left(\mathbf{x}\left(\mathbf{x}^T\mathbf{x}\right)^{-1}\mathbf{x}^T\right)\mathbf{y} \tag{1.64}$$

$$= \mathbf{w}\mathbf{y} \tag{1.65}$$

where $\mathbf{w}$ is the $n \times n$ matrix, with w_{ij} saying how much of each observed y_j contributes to each fitted $\widehat{\mu}_i$. This is what, a little while ago, I called the influence or hat matrix, in the special case of ordinary least squares.

Notice that $\mathbf{w}$ depends *only* on the predictor variables in $\mathbf{x}$; the observed response values in $\mathbf{y}$ don't matter. If we change around $\mathbf{y}$, the fitted values $\widehat{\mu}$ will also change, but only within the limits allowed by $\mathbf{w}$. There are n independent coordinates along which $\mathbf{y}$ can change, so we say the data have n degrees of freedom. Once $\mathbf{x}$ (and thus $\mathbf{w}$) are fixed, however, $\widehat{\mu}$ has to lie in a p-dimensional linear subspace in this n-dimensional space, and the residuals have to lie in the $(n-p)$-dimensional space orthogonal to it.

Geometrically, the dimension of the space in which $\widehat{\mu} = \mathbf{w}\mathbf{y}$ is confined is the rank of the matrix $\mathbf{w}$. Since $\mathbf{w}$ is an idempotent matrix (Exercise 1.5), its rank equals its trace. And that trace is, exactly, p:

cf ESL derivation

$$\operatorname{tr}\mathbf{w} = \operatorname{tr}\left(\mathbf{x}\left(\mathbf{x}^T\mathbf{x}\right)^{-1}\mathbf{x}^T\right) \tag{1.66}$$

$$= \operatorname{tr}\left(\mathbf{x}^T\mathbf{x}\left(\mathbf{x}^T\mathbf{x}\right)^{-1}\right) \tag{1.67}$$

$$= \operatorname{tr}\mathbf{I}_p = p \tag{1.68}$$

since for any matrices $\mathbf{a}, \mathbf{b}$, $\operatorname{tr}(\mathbf{ab}) = \operatorname{tr}(\mathbf{ba})$, and $\mathbf{x}^T\mathbf{x}$ is a $p \times p$ matrix[14].

For more general linear smoothers, we can still write Eq. 1.53 in matrix form,

$$\widehat{\mu} = \mathbf{w}\mathbf{y} \tag{1.69}$$

We now *define* the degrees of freedom[15] to be the trace of $\mathbf{w}$:

$$df(\widehat{\mu}) \equiv \operatorname{tr}\mathbf{w} \tag{1.70}$$

*This may not be an integer.

[14] This all assumes that $\mathbf{x}^T\mathbf{x}$ has an inverse. Can you work out what happens when it does not?

[15] Some authors prefer to say "*effective* degrees of freedom", to emphasize that we're not just counting parameters.

cf effective N for correlated data

Covariance of Observations and Fits

Eq. 1.70 defines the number of degrees of freedom for linear smoothers. A yet more general definition includes nonlinear methods, assuming that $Y_i = \mu(x_i) + \epsilon_i$, and the ϵ_i consist of uncorrelated noise of constant[16] variance σ^2. This is

$$df(\widehat{\mu}) \equiv \frac{1}{\sigma^2} \sum_{i=1}^{n} \text{Cov}\left[Y_i, \widehat{\mu}(x_i)\right] \tag{1.71}$$

In words, this is the normalized covariance between each observed response Y_i and the corresponding predicted value, $\widehat{\mu}(x_i)$. This is a very natural way of measuring how flexible or stable the regression model is, by seeing how much it shifts with the data.

cf cor?

If we do have a linear smoother, Eq. 1.71 reduces to Eq. 1.70.

$$\text{Cov}\left[Y_i, \widehat{\mu}(x_i)\right] = \text{Cov}\left[Y_i, \sum_{j=1}^{n} w_{ij} Y_j\right] \tag{1.72}$$

$$= \sum_{j=1}^{n} w_{ij} \text{Cov}\left[Y_i, Y_j\right] \tag{1.73}$$

$$= w_{ii} \mathbb{V}\left[Y_i\right] = \sigma^2 w_{ii} \tag{1.74}$$

Here the first line uses the fact that we're dealing with a linear smoother, and the last line the assumption that ϵ_i is uncorrelated and has constant variance. Therefore

$$df(\widehat{\mu}) = \frac{1}{\sigma^2} \sum_{i=1}^{n} \sigma^2 w_{ii} = \text{tr}\, \mathbf{w} \tag{1.75}$$

as promised.

1.5.3.3 Prediction Errors

Bias

Because linear smoothers are linear in the response variable, it's easy to work out (theoretically) the expected value of their fits:

$$\mathbb{E}\left[\widehat{\mu}_i\right] = \sum_{j=1}^{n} w_{ij} \mathbb{E}\left[Y_j\right] \tag{1.76}$$

In matrix form,

$$\mathbb{E}\left[\widehat{\mu}\right] = \mathbf{w} \mathbb{E}\left[\mathbf{Y}\right] \tag{1.77}$$

This means the smoother is unbiased if, and only if, $\mathbf{w}\mathbb{E}[\mathbf{Y}] = \mathbb{E}[\mathbf{Y}]$, that is, if $\mathbb{E}[\mathbf{Y}]$ is an eigenvector of $\mathbf{w}$. Turned around, the condition for the smoother to be unbiased is

$$(\mathbf{I}_n - \mathbf{w})\mathbb{E}\left[\mathbf{Y}\right] = \mathbf{0} \tag{1.78}$$

with e'value of 1?

[16] But see Exercise 1.10.

In general, $(\mathbf{I}_n - \mathbf{w})\mathbb{E}[\mathbf{Y}] \neq \mathbf{0}$, so linear smoothers are more or less biased. Different smoothers are, however, unbiased for different families of regression functions. Ordinary linear regression, for example, is unbiased if and only if the regression function really is linear.

In-sample mean squared error

When you studied linear regression, you learned that the expected mean-squared error on the data used to fit the model is $\sigma^2(n-p)/n$. This formula generalizes to other linear smoothers. Let's first write the residuals in matrix form.

$$\mathbf{y} - \widehat{\mu} = \mathbf{y} - \mathbf{w}\mathbf{y} \tag{1.79}$$
$$= \mathbf{I}_n\mathbf{y} - \mathbf{w}\mathbf{y} \tag{1.80}$$
$$= (\mathbf{I}_n - \mathbf{w})\mathbf{y} \tag{1.81}$$

The in-sample mean squared error is $n^{-1}\left\|\mathbf{y} - \widehat{\mu}\right\|^2$, so

$$\frac{1}{n}\left\|\mathbf{y} - \widehat{\mu}\right\|^2 = \frac{1}{n}\left\|(\mathbf{I}_n - \mathbf{w})\mathbf{y}\right\|^2 \tag{1.82}$$
$$= \frac{1}{n}\mathbf{y}^T(\mathbf{I}_n - \mathbf{w}^T)(\mathbf{I}_n - \mathbf{w})\mathbf{y} \tag{1.83}$$

Taking expectations[17],

$$\mathbb{E}\left[\frac{1}{n}\left\|\mathbf{y} - \widehat{\mu}\right\|^2\right] = \frac{\sigma^2}{n}\operatorname{tr}\left((\mathbf{I}_n - \mathbf{w}^T)(\mathbf{I}_n - \mathbf{w})\right) + \frac{1}{n}\left\|(\mathbf{I}_n - \mathbf{w})\mathbb{E}[\mathbf{y}]\right\|^2 \tag{1.84}$$
$$= \frac{\sigma^2}{n}\left(\operatorname{tr}\mathbf{I}_n - 2\operatorname{tr}\mathbf{w} + \operatorname{tr}(\mathbf{w}^T\mathbf{w})\right) + \frac{1}{n}\left\|(\mathbf{I}_n - \mathbf{w})\mathbb{E}[\mathbf{y}]\right\|^2 \tag{1.85}$$
$$= \frac{\sigma^2}{n}\left(n - 2\operatorname{tr}\mathbf{w} + \operatorname{tr}(\mathbf{w}^T\mathbf{w})\right) + \frac{1}{n}\left\|(\mathbf{I}_n - \mathbf{w})\mathbb{E}[\mathbf{y}]\right\|^2 \tag{1.86}$$

The last term, $n^{-1}\left\|(\mathbf{I}_n - \mathbf{w})\mathbb{E}[\mathbf{y}]\right\|^2$, comes from the bias: it indicates the distortion that the smoother would impose on the regression function, even without noise. The first term, proportional to σ^2, reflects the variance. Notice that it involves not only what we've called the degrees of freedom, $\operatorname{tr}\mathbf{w}$, but also a second-order term, $\operatorname{tr}\mathbf{w}^T\mathbf{w}$. For ordinary linear regression, you can show (Exercise 1.9) that $\operatorname{tr}(\mathbf{w}^T\mathbf{w}) = p$, so $2\operatorname{tr}\mathbf{w} - \operatorname{tr}(\mathbf{w}^T\mathbf{w})$ would also equal p. For this reason, some people prefer either $\operatorname{tr}(\mathbf{w}^T\mathbf{w})$ or $2\operatorname{tr}\mathbf{w} - \operatorname{tr}(\mathbf{w}^T\mathbf{w})$ as the definition of degrees of freedom for linear smoothers, so be careful.

1.5.3.4 Inferential Statistics

Many of the formulas underlying things like the F test for whether a regression predicts significantly better than the global mean carry over from linear regression to linear smoothers, if one uses the right definitions of degrees of freedom, *and* one believes that the noise is always IID and Gaussian. However, we will see ways of

[17] By using the general result that $\mathbb{E}\left[\vec{X} \cdot \mathbf{a}\vec{X}\right] = \operatorname{tr}\left(\mathbf{a}\mathbb{V}\left[\vec{X}\right]\right) + \mathbb{E}\left[\vec{X}\right] \cdot \mathbf{a}\mathbb{E}\left[\vec{X}\right]$ for any random vector $\vec{X}$ and non-random square matrix $\mathbf{a}$.

cf mixed models

doing inference on regression models which don't rely on Gaussian assumptions at all (Ch. 6), so I won't go over these results.

1.6 Further Reading

In Chapter 2, we'll look more at the limits of linear regression and some extensions; Chapter 3 will cover some key aspects of evaluating statistical models, including regression models; and then Chapter 4 will come back to kernel regression, and more powerful tools than `ksmooth`. Chapters 10–8 and 13 all introduce further regression methods, while Chapters 11–12 pursue extensions.

Good treatments of regression, emphasizing linear smoothers but *not* limited to linear regression, can be found in Wasserman (2003, 2006), Simonoff (1996), Faraway (2006) and Györfi *et al.* (2002). The last of these in particular provides a very thorough theoretical treatment of non-parametric regression methods.

On generalizations of degrees of freedom to non-linear models, see Buja *et al.* (1989, §2.7.3), and Ye (1998).

Historical notes

All the forms of nonparametric regression covered in this chapter are actually quite old. Kernel regression was introduced independently by Nadaraya (1964) and Watson (1964). The origin of nearest neighbor methods is less clear, and indeed they may have been independently invented multiple times — Cover and Hart (1967) collects some of the relevant early citations, as well as providing a pioneering theoretical analysis, extended to regression problems in Cover (1968a,b).

Exercises

1.1 Suppose $Y_1, Y_2, \ldots Y_n$ are random variables with the same mean μ and standard deviation σ, and that they are all uncorrelated with each other, but not necessarily independent[18] or identically distributed. Show the following:

1. $\mathbb{V}\left[\sum_{i=1}^{n} Y_i\right] = n\sigma^2$.
2. $\mathbb{V}\left[n^{-1}\sum_{i=1}^{n} Y_i\right] = \sigma^2/n$.
3. The standard deviation of $n^{-1}\sum_{i=1}^{n} Y_i$ is $\sigma/\sqrt{n}$.
4. The standard deviation of $\mu - n^{-1}\sum_{i=1}^{n} Y_i$ is $\sigma/\sqrt{n}$.

Can you state the analogous results when the Y_i share mean μ but each has its own standard deviation σ_i? When each Y_i has a distinct mean μ_i? (Assume in both cases that the Y_i remain uncorrelated.)

1.2 Suppose we use the **mean absolute error** instead of the mean squared error:

$$\mathrm{MAE}(m) = \mathbb{E}\left[|Y - m|\right] \tag{1.87}$$

Is this also minimized by taking $m = \mathbb{E}[Y]$? If not, what value $\tilde{\mu}$ minimizes the MAE? Should we use MSE or MAE to measure error?

1.3 Derive Eqs. 1.45 and 1.44 by minimizing Eq. 1.43.

[18] See Appendix **??** for a refresher on the difference between "uncorrelated" and "independent".