1 Introduction

1.0.0.1. **Statistical learning** refers to a vast set of tools for understanding data.

1.0.0.2. **Supervised learning** builds a statistical model for predicting or estimating an output based on one or more inputs. Examples include regression, classification, neural networks. Think of it as modeling a function from data.

1.0.0.3. **Unsupervised learning** has inputs, but no supervising output (no labeled training data). These tools look for relationships and patterns within input data. Examples: clustering, principal component analysis.

1.0.0.4. **Continuous or quantitative** refers to numerical values. Modeling continuous data is referred to as **regression**.

1.0.0.5. **Categorical or qualitative** refers to non-numerical data. Modeling categorical data is referred to as **classification**.

1.0.0.6. Grouping data based upon observed characteristics is called **clustering**. This is an example of unsupervised learning.

2 Statistical Learning

2.1 What is Statistical Learning?

2.1.0.1. **Input variables** also known as **predictors**, **independent variables**, **features**, or sometimes just **variables**, are the features which we typically know, and use to make predictions about an output variable. Input variables are typically denoted by X, possibly with a subscript.

2.1.0.2. The **output variable**, also sometimes known as the **response** or **dependent variable** is the quantity we are trying to model and is typically denoted Y.

2.1.0.3. Suppose we observe a quantitative response Y and p different predictors $X_1, X_2, ..., X_p$. We assume there is a relationship between Y and $X = (X_1, ..., X_p)$ of the form:

$$Y = f(X) + \epsilon$$

where f is a fixed but unknown function and ϵ is an **error term**, independent of X with mean 0.

We say that f represents the **systematic information** that X provides about Y.

Statistical learning refers to a set of approaches to estimating f.

2.1.1 Why Estimate f?

2.1.1.1. We want to estimate f for two main reasons: **prediction** and **inference**.

2.1.1.2. Using the above formulation, since the error term ϵ averages to 0, we can predict Y using:

$$\hat{Y} = \hat{f}(X)$$

where $\hat{f}$ is the estimate for f and $\hat{Y}$ is our subsequent prediction for Y.

2.1.1.3. $\hat{f}$ is treated like a **black box**, in the sense that we don't particularly care about the details of $\hat{f}$ as long as it does a good job of predicting Y.

2.1.1.4. The accuracy of $\hat{Y}$ as a prediction for Y depends on **reducible error** and **irreducible error**. **Reducible error** is error that can potentially be reduced by improving the estimate $\hat{f}$.

2.1.1.5. Even if reducible error were reduced completely, so that we have a perfect estimate of f, there would still be **irreducible error** since Y is presumed to depend on ϵ which cannot possibly be predicted using X. This is because ϵ may contain unmeasured variables which affect Y. Or it may be that ϵ contained unmeasurable variation.

2.1.1.6. We can see reducible and irreducible error numerically. Fix $\hat{f}$ and predictors X so that the only variability comes in Y from ϵ. Then:

$$\begin{aligned}
\mathbb{E}(Y-\hat{Y})^2 &= \mathbb{E}[f(X)+\epsilon-\hat{f}(X)]^2 \\
&= \mathbb{E}[(f(X)-\hat{f}(X))+\epsilon]^2 \\
&= \mathbb{E}[(f(X)-\hat{f}(X))^2+2\epsilon(f(X)-\hat{f}(X))+\epsilon^2] \\
&= \mathbb{E}(f(X)-\hat{f}(X))^2+\mathbb{E}(2\epsilon(f(X)-\hat{f}(X)))+\mathbb{E}(\epsilon^2) && \text{(linearity of expectation)} \\
&= [f(X)-\hat{f}(X)]^2+2(f(X)-\hat{f}(X))\mathbb{E}(\epsilon)+\mathrm{Var}(\epsilon) && (X \text{ is fixed}) \\
&= \underbrace{[f(X)-\hat{f}(X)]^2}_{\text{Reducible}}+\underbrace{\mathrm{Var}(\epsilon)}_{\text{Irreducible}} && (\mathbb{E}(\epsilon)=0)
\end{aligned}$$

2.1.1.7. Irreducible error will always provide an upper bound on the accuracy of our prediction for Y. This bound is almost always unknown in practice.

2.1.1.8. Often, the goal is to understand the relationship between Y and $X_1, ...X_p$. And so we try to estimate f. But now, when prediction isn't the goal, we do not treat $\hat{f}$ as a black box.

2.1.1.9. In inference, we might be interested in questions like:

- Which predictors are associated with the response?
- What is the relationship between the response and the predictor?
- Can the relationship between Y and each predictor be summarized using a linear equation, or is the relationship more complicated?

2.1.1.10. Sometimes the goal can be a combination of prediction and inference. The type of model used might be determined by the nature of the goal. Linear modelling is relatively easy to interpret, so lends itself well to inference, whereas more complicated models may do a better job at prediction, but be more difficult to interpret.

2.1.2 How Do We Estimate f?

2.1.2.1. **Notation:** n is the number of data points in our **training data** – the observations we use to train or teach our estimate $\hat{f}$. Let x_{ij} be the value of the jth predictor in the ith observation. And let y_i be the value of the response variable in the ith observation. Therefore, the training data is:

$$\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\} \quad \text{where} \quad x_i = (x_{i1}, x_{i2}, \ldots, x_{ip})^T$$

2.1.2.2. Broadly speaking, statistical learning methods fall into one of two types: **parametric** and **non-parametric**.

2.1.2.3. A rough characterization of the difference might be that parametric models are “form then data” whereas non-parametric models are “data then form.”

2.1.2.4. **Parametric models** involve a two-step approach:

- First, make an assumption about the form of f. One simple example is that f is linear in X:

  $$f(X) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p$$

  This is a **linear model**. With the assumption that f is linear, the problem of estimating f becomes the problem of estimating the **parameters** $\beta_0, \ldots, \beta_p$.

- After making the assumption about the model (model selection), **train** or **fit** the model. One approach to fitting the model is called **(ordinary) least squares**, but there are many.

2.1.2.5. Examples of parametric methods: regression.

2.1.2.6. The disadvantage of a parametric approach is that $\hat{f}$ will usually not match the true, unknown f.

2.1.2.7. We can try to address this problem by choosing **flexible** models that can fit many f well. In general, a more flexible model involves a greater number of parameters. This can lead to **overfitting** the data which means that the **noise** in the data is followed too closely.

2.1.2.8. **Non-parametric methods** do not make explicit assumptions about the functional form of f. Instead, they seek an estimate of f that gets as close to the data points as possible without being too rough or wiggly.

2.1.2.9. The advantage of non-parametric models is that they have the potential to accurately fit a wider range of possible shapes for f.

2.1.2.10. The disadvantage is that whereas parametric methods reduce the problem of estimating f to estimating a few parameters, estimating f using non-parametric methods do not admit this simplification and, consequently, requires a large number of observations to obtain an accurate $\hat{f}$.

2.1.2.11. Examples of non-parametric methods:

- **Kernel methods** like kernel regression or support vector machines.
- **Splines**
- **Decision Trees**
- **Nearest neighbor methods** like KNN.

2.1.2.12. We can judge these models by their **smoothness**. This is a somewhat under-defined quality of a model – it speaks to the dichotomy between "smooth" and "bumpy". Lower levels of smoothness may be indicative of overfitting. Choosing the correct amount of smoothness will be discussed later.

2.1.3 The Trade-Off Between Prediction Accuracy and Model Interpretability

2.1.3.1. As flexibility of a model increases, interpretability generally goes down. *

2.1.3.2. In order, from least flexible to most flexible:

- Subset Selection Lasso
- Least Squares, linear regression
- Generalized Additive Models (GAMs), Trees
- Bagging, Boosting
- Support Vector Machines
- Deep Learning

2.1.3.3. If the goal is interpretability, a less flexible model may be preferable. Whereas if prediction is the goal, it turns out that sometimes less flexible models make better predictions – this is because of overfitting.

2.1.4 Supervised vs Unsupervised Learning

2.1.4.1. In **supervised learning**, the value of the response variable Y is known. In **unsupervised learning**, the vector of measurements of $X_1, ..., X_p$ may be known for our observations, but there is no associated response.

2.1.4.2. Linear regression is supervised learning.

2.1.4.3. **Cluster analysis**, a method to ascertain on the basis of $x_1, ..., x_n$ whether the observations in the data fall into relatively distinct groups.

2.1.4.4. **Semi-supervised learning** is when the value of the response variable is available for only a strict subset of the observations – this may be because collecting the data on the response variable is difficult or expensive. This topic is not covered in this book.

2.1.5 Regression versus Classification Problems

2.1.5.1. Variables can be either **quantitative** or **qualitative** (also known as **categorical**).

2.1.5.2. Quantitative variables take on numerical values, whereas qualitative variables take on values in one of K different **classes**, or categories.

2.1.5.3. Problems with a quantitative response are referred to as **regression problems**, whereas those involving a qualitative response are **classification problems**.

2.1.5.4. Sometimes, the distinction is not always crisp. Logistic regression is typically used with a two-class or binary response variable and thus is a classification problem. However, since it estimates probabilities of being in a class, it can also be thought of as a regression problem.

2.1.5.5. Whether predictors, the X's are qualitative or quantitative is generally considered to be less important. "Most of the statistical learning methods discussed in this book can be applied regardless of the predictor variable type, provided that any qualitative predictors are properly coded before the analysis."

2.2 Assessing Model Accuracy

2.2.0.1. It is necessary to introduce many different statistical learning approaches, because no one method dominates all others over all data sets.

2.2.0.2. Selecting the best approach can be one of the most challenging parts of statistical learning in practice.

2.2.1 Measuring the Quality of Fit

2.2.1.1. To quantify the extent to which the predicted response value for a given observation is close to the true response value for that observation, in the regression setting, the most commonly-used measure is the **mean squared error** (MSE):

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{f}(x_i))^2$$

2.2.1.2. Because this MSE is computed using training data, it is also sometimes called **training MSE**.

2.2.1.3. If prediction is our task, then predicting training data well is not really what we care about.Rather, we are interested in the accuracy of predictions when we apply our method to previously unseen **test data**.

2.2.1.4. We want to minimize **test MSE**:

$$\text{Ave}(y_0 - \hat{f}(x_0))^2$$

over test observations (x_0, y_0).

2.2.1.5. How to select a method which minimizes test MSE? We can reserve a set of observations that were not used to train the statistical learning method called the **test data**.

2.2.1.6. Or we might just take the method which minimizes training MSE, believing that this should also minimize the test MSE. The problem here is that we can minimize training MSE by computing a spline which perfectly interpolates all data points, but which massively overfits the data, and so performs poorly on test MSE.

2.2.1.7. Training MSE decreases monotonically with flexibility (greater degrees of freedom). Test MSE looks like a U-shape. One important method which will be discussed in the future for finding the lowest point of the U-curve is called **cross-validation**.

2.2.1.8. The irreducible error $\text{Var}(\epsilon)$ is the lowest achievable test MSE.

2.2.2 The Bias-Variance Trade-Off

2.2.2.1. For a fixed observation x_0, the **expected test MSE at** x_0 is the average value of $(y_0 - \hat{f}(x_0))^2$, where we average over the different possible realizations of training data that generated $\hat{f}$. Consider a particular training data set $\mathcal{T}$ coming from the set of all possible samples $\mathcal{S}$, presumably of the same size. Then $\hat{f}$, really should be denoted $\hat{f}_{\mathcal{T}}$ since $\hat{f}$ is obtained by training on $\mathcal{T}$. Then the bias is:

$$\text{Expected Test MSE at } x_0 = \mathbb{E}_{\mathcal{T}\in\mathcal{S}}(y_0 - \hat{f}_{\mathcal{T}}(x_0))^2$$

2.2.2.2. **Bias** of a statistical learning method is the difference of the average value of prediction over different realizations of training data and the actual value:

$$\text{Bias}(\hat{f}(x_0)) = \mathbb{E}[\hat{f}(x_0) - f(x_0)]$$

Generally speaking, perhaps as a rule of thumb, as flexibility increases, bias decreases.

2.2.2.3. **Variance** of a statistical learning method is the variance in $\hat{f}(x_0)$ over different realizations of the training set.

$$\text{Variance}(\hat{f}(x_0)) = \mathbb{E}\Big[(\hat{f}(x_0) - \mathbb{E}[\hat{f}(x_0)])^2\Big]$$

High variance means that small changes in the training data can result in large changes in $\hat{f}$. Generally, more flexible statistical methods have higher variance.

2.2.2.4. One can show that the expected test MSE at x_0 can be decomposed into the sum of the variance, the squared bias of, and the variance of the error:

$$\mathbb{E}[(y_0 - \hat{f}(x_0))^2] = \text{Var}(\hat{f}(x_0)) + [\text{Bias}(\hat{f}(x_0))]^2 + \text{Var}(\epsilon)$$

Proof:

$$\begin{aligned}
\mathbb{E}[(y_0 - \hat{f}(x_0))^2] &= \mathbb{E}[((f(x_0) + \epsilon) - \hat{f}(x_0))^2] \\
&= \mathbb{E}[(f(x_0) - \hat{f}(x_0) + \epsilon)^2] \\
&= \mathbb{E}[(f(x_0) - \hat{f}(x_0))^2 - 2\epsilon(f(x_0) - \hat{f}(x_0)) + \epsilon^2] \\
&= \mathbb{E}[(f(x_0) - \hat{f}(x_0))^2] - 2\mathbb{E}[\epsilon(f(x_0) - \hat{f}(x_0))] + \mathbb{E}[\epsilon^2] && \text{(linearity of expectation)} \\
&= \mathbb{E}[(f(x_0) - \hat{f}(x_0))^2] - 2\mathbb{E}[\epsilon]\mathbb{E}[(f(x_0) - \hat{f}(x_0))] + \mathbb{E}[\epsilon^2] && \text{(independence of } \epsilon \text{ and } \hat{f}(x_0)\text{)} \\
&= \mathbb{E}[(f(x_0) - \hat{f}(x_0))^2] + \text{Var}(\epsilon) && (\mathbb{E}[\epsilon] = 0) \\
&= \mathbb{E}\Big[(f(x_0) - \mathbb{E}[\hat{f}(x_0)] + \mathbb{E}[\hat{f}(x_0)] - \hat{f}(x_0))^2\Big] + \text{Var}(\epsilon) \\
&= \mathbb{E}\Big[(f(x_0) - \mathbb{E}[\hat{f}(x_0)])^2 + (\mathbb{E}[\hat{f}(x_0)] - \hat{f}(x_0))^2 + 2(f(x_0) \\
&\qquad - \mathbb{E}[\hat{f}(x_0)])(\mathbb{E}[\hat{f}(x_0)] - \hat{f}(x_0))\Big] + \text{Var}(\epsilon) \\
&= \underbrace{\mathbb{E}[(f(x_0) - \mathbb{E}[\hat{f}(x_0)])^2]}_{\text{Bias squared}} + \underbrace{\mathbb{E}[(\mathbb{E}[\hat{f}(x_0)] - \hat{f}(x_0))^2]}_{\text{Variance}} + \\
&\qquad + \mathbb{E}[2(f(x_0) - \mathbb{E}[\hat{f}(x_0)])(\mathbb{E}[\hat{f}(x_0)] - \hat{f}(x_0))] + \text{Var}(\epsilon) \\
&= [\text{Bias}(\hat{f}(x_0))]^2 + \text{Var}(\hat{f}(x_0)) + 2\Big(f(x_0) - \mathbb{E}[\hat{f}(x_0)]\Big)\underbrace{\mathbb{E}\Big[\mathbb{E}[\hat{f}(x_0)] - \hat{f}(x_0)\Big]}_{=0} + \text{Var}(\epsilon) \\
&= [\text{Bias}(\hat{f}(x_0))]^2 + \text{Var}(\hat{f}(x_0)) + \text{Var}(\epsilon)
\end{aligned}$$

□

2.2.2.5. For any point estimator $\hat{\theta}$ of an estimand θ, we can define the mean squared error (MSE) which is

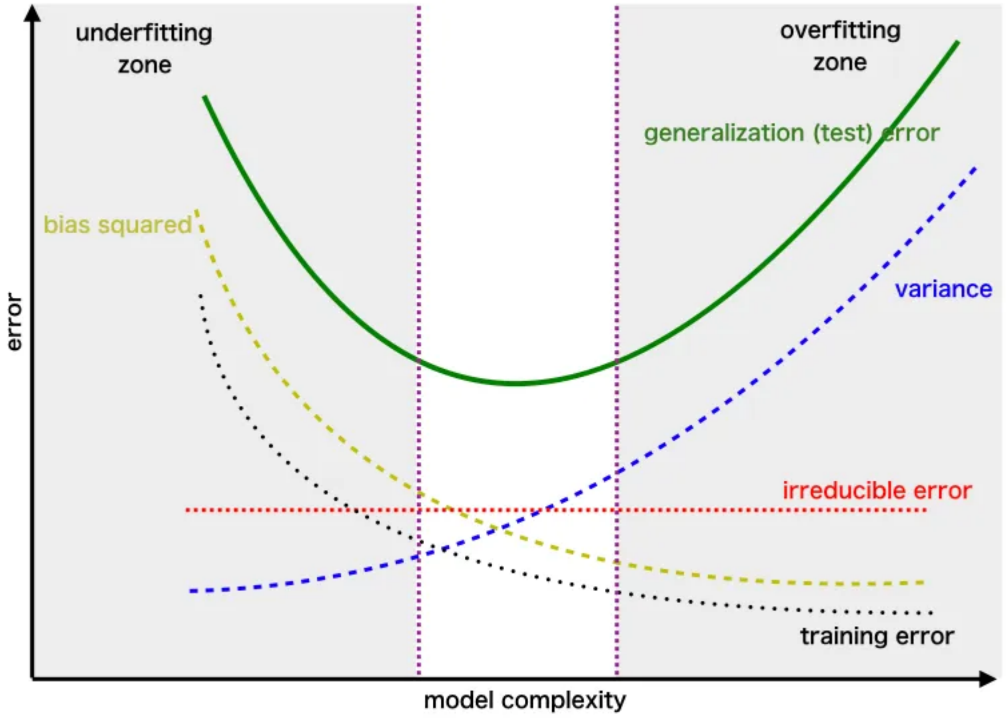
$$\text{MSE}(\hat{\theta}) = \mathbb{E}[(\theta - \hat{\theta})^2]$$

There is a similar decomposition of MSE into the sum of variance and bias squared.

2.2.2.6. As a general rule, as flexibility increases in model selection, variance increases and bias decreases. But the overall effect on test MSE depends on the relative rates of change of these quantities. For a very flexible model, a small increase in flexibility gives a small decrease in bias, but a large increase in variance, thus we see an increase in test MSE.

2.2.2.7. Since variance concerns variation in the model if we were to use different training sets, the smaller the training set is, the more variance we should expect to see. Consequently, if we have a large training set, variance should be less of a concern. In this case, we can afford to use more complicated models.

2.2.2.8. Here is a typical image representing the state of affairs:



2.2.2.9. There is a "trade-off" since, at the two-extremes of an empty model where we always predict the grand mean of the response variable, and an overfitting model which perfectly interpolates the data, represent the extremes in variance and bias. In the first case, we have very low variance (in fact, none) but high bias, and in the second case we see high variance and very low bias.

2.2.2.10. In real life, f is typically unknown. So it is generally not possible to compute test MSE, bias or variance.

2.2.3 The Classification Setting

2.2.3.1. Suppose now that the response variable is qualitative, rather than quantitative. We say we are in the classification setting. We still want to estimate f. How do quantify the accuracy of $\hat{f}$ in the classification setting? We use the **error rate**, also called the **training error rate**:

$$\text{error rate} = \frac{1}{n}\sum_{i=1}^{n} I(y_i \neq \hat{y}_i)$$

where I is an **indicator variable**:

$$I(\text{proposition}) = \begin{cases} 1 & \text{if proposition is TRUE} \\ 0 & \text{if proposition is FALSE} \end{cases}$$

so that:

$$I(y_i \neq \hat{y}_i) = \begin{cases} 1 & \text{if } y_i \neq \hat{y}_i \\ 0 & \text{if } y_i = \hat{y}_i \end{cases}$$

Therefore, $\sum I(y_i \neq \hat{y}_i)$ counts up the number of misclassifications.

2.2.3.2. The **test error rate** is the error rate over test observations. A good classifier is one for which the test error rate is smallest.

2.2.3.3. The optimal classifier is the one which assigns each observation to the most likely class, given its predictor values. That is, we predict that for an observation x_0 that Y is class j if:

$$P(Y = j|X = x_0)$$

is largest. We call this the **Bayes classifier**.

2.2.3.4. The Bayes classifer produces the lowest possible error rate.

2.2.3.5. Upon input $X = x_0$, the Bayes classifier will correctly classify with probability precisely

$$\max_j P(Y = j|X = x_0).$$

Therefore, the error rate upon input $X = x_0$ of the Bayes classifier will be:

$$1 - \max_j P(Y = j|X = x_0)$$

The overall Bayes error rate is

$$1 - \mathbb{E}\left[\max_j P(Y = j|X = x_0)\right]$$

2.2.3.6. The overall Bayes error rate is analogous to irreducible error.

2.2.3.7. If we have a two-class or binary classification problem, then the Bayes classifier selects the class for which the conditional probability is greater than 0.5.

2.2.3.8. The values of X which make the conditional probability equal to 0.5 is called the **Bayes decision boundary**.

2.2.3.9. Computing the Bayes classifier is impossible since we do not know the conditional distribution of Y given X. The Bayes classifier is the theoretically optimal model. Many approaches estimate the conditional distribution of Y given X, then classify an observation using the highest estimated probability.

2.2.3.10. One such method is the **K-nearest neighbors** or **KNN classifer**. Given a positive integer K and a test observation x_0, the KNN-classifier finds the K points in the training data which are nearest to x_0 represented by $\mathcal{N}_0$. It then estimates the conditional probability as the fraction of the points in $\mathcal{N}_0$ whose response values equal j:

$$P(Y = j|X = x_0) = \frac{1}{K} \sum_{i \in \mathcal{N}_0} I(y_i = j)$$

The KNN classifier classifies the test observation to be in the class j with the highest probability.

2.2.3.11. In binary classification problems, there is a KNN decision boundary. The goal is to find K for which the KNN decision boundary is very close to the Bayes decision boundary.

2.2.3.12. Small values of K give a very flexible model, while very large values of K are not flexible. And so we see the bias-variance tradeoff at work in this family of models.

3 Linear Regression

3.0.0.1. **Linear regression** is a very simple approach to supervised learning.

3.0.0.2. It is a good jumping-off point for newer approaches.

3.0.0.3. Some questions that linear regression can answer:

- Is there a relationship between a predictor and response variable?
- How strong is the relationship?

- Which subset of predictors are associated with response variable?
- How large is the association?
- How accurately can we predict response variable?
- Is the relationship linear?
- Is there **synergy** among predictors?

3.1 Simple Linear Regression

3.1.0.1. This is for predicting a quantitative response variable Y using a single predictor X.

$$Y \approx \beta_0 + \beta_1 X$$

3.1.0.2. We can read "$\approx$" as "is approximately modeled as."

3.1.0.3. We sometimes say that we are **regressing** Y **onto** X.

3.1.0.4. We call β_0 the **intercept** and β_1 the **slope**. These are the model **coefficients** or **parameters.**

3.1.0.5. We use training data to produce estimates $\hat{\beta}_0$ and $\hat{\beta}_1$ We use these to produce an prediction for the value of Y:

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x$$

3.1.1 Estimating the Coefficients

3.1.1.1. Let $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$ represent n observation pairs.

3.1.1.2. Using this data, we want to obtain coefficient estimates $\hat{\beta}_0$ and $\hat{\beta}_1$ so that $y_i \approx \hat{\beta}_0 + \hat{\beta}_1 x_i$. Here we are using "$\approx$" in the numerical sense.

3.1.1.3. There are a number of ways of measuring **closeness**, but the most common approach is to minimize the **least squares** criterion. Other approaches will be discussed later.

3.1.1.4. Let $\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i$ be the prediction for Y upon the ith value of X. Then $e_i = y_i - \hat{y}_i$ is the ith **residual**.

3.1.1.5. The **residual sum of squares** (RSS) is:

$$\text{RSS} = e_1^2 + e_2^2 + \cdots + e_n^2$$

which is equivalent to:

$$\text{RSS} = (y_1 - \hat{\beta}_0 - \hat{\beta}_1 x_1)^2 + (y_2 - \hat{\beta}_0 - \hat{\beta}_1 x_2)^2 + \cdots + (y_n - \hat{\beta}_0 - \hat{\beta}_1 x_n)^2$$

The least squares approach chooses $\hat{\beta}_0$ and $\hat{\beta}_1$ to minimize RSS.

3.1.1.6. Let's minimize RSS:

$$\begin{aligned} 0 = \frac{\partial}{\partial \hat{\beta}_0}\text{RSS} &= \frac{\partial}{\partial \hat{\beta}_0} \sum_{i=1}^{n} (y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i)^2 \\ &= \sum_{i=1}^{n} \frac{\partial}{\partial \hat{\beta}_0} (y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i)^2 \\ &= \sum_{i=1}^{n} -2(y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i) \end{aligned}$$

which implies that $\hat{\beta}_0 = \overline{y} - \hat{\beta}_1\overline{x}$.

$$\begin{aligned}
0 = \frac{\partial}{\partial \hat{\beta}_1}\text{RSS} &= \frac{\partial}{\partial \hat{\beta}_1}\sum_{i=1}^{n}(y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i)^2 \\
&= \sum_{i=1}^{n}\frac{\partial}{\partial \hat{\beta}_1}(y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i)^2 \\
&= \sum_{i=1}^{n} -2x_i(y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i) \\
&= \sum_{i=1}^{n} -2x_i(y_i - (\overline{y} - \hat{\beta}_1\overline{x}) - \hat{\beta}_1 x_i) \\
&= \sum_{i=1}^{n} -2x_i(y_i - \overline{y} - \hat{\beta}_1(x_i - \overline{x}))
\end{aligned}$$

This is equivalent to:

$$\begin{aligned}
0 &= \sum_{i=1}^{n} x_i(y_i - \overline{y} - \hat{\beta}_1(x_i - \overline{x})) \\
0 &= \sum_{i=1}^{n}(x_i - \overline{x} + \overline{x})(y_i - \overline{y} - \hat{\beta}_1(x_i - \overline{x})) \\
0 &= \sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y}) - \hat{\beta}_1(x_i - \overline{x})^2 + \overline{x}(y_i - \overline{y}) - \hat{\beta}_1\overline{x}(x_i - \overline{x}) \\
\sum_{i=1}^{n}\hat{\beta}_1(x_i - \overline{x})^2 &= \sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y}) + \overline{x}\underbrace{\left(\sum_{i=1}^{n} y_i - \overline{y}\right)}_{=0} - \hat{\beta}_1\overline{x}\underbrace{\left(\sum_{i=1}^{n} x_i - \overline{x}\right)}_{=0} \\
\hat{\beta}_1 &= \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sum_{i=1}^{n}(x_i - \overline{x})^2}
\end{aligned}$$

Thus, to minimize RSS, we have:

Least squares regression coefficient estimates:

$$\begin{aligned}
\hat{\beta}_1 &= \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sum_{i=1}^{n}(x_i - \overline{x})^2} \\
\hat{\beta}_0 &= \overline{y} - \hat{\beta}_1\overline{x}
\end{aligned}$$

3.1.2 Assessing the Accuracy of the Coefficient Estimates

3.1.2.1. Recall that our underlying assumption is that the true relationship between X and Y is of the form

$$Y = f(X) + \epsilon$$

If f is to be approximated by a linear function, then we can write this relationship as:

$$Y = \beta_0 + \beta_1 X + \epsilon$$

This defines the **population regression line**.

3.1.2.2. Here, β_0 is the expected value of Y when $X = 0$ (the intercept), and β_1 is the slope – the average increase in Y associated with a one-unit increase in X. Here, the error term ϵ catches everything that is missed by this simple model.

3.1.2.3. The **least squares line**, the one characterized by the above formulas for $\hat{\beta}_0$ and $\hat{\beta}_1$, is derived from a sample of data arising from the population. Thus, there can be many different least squares lines.

3.1.2.4. Recall that the sample mean is an **unbiased estimator** of the population mean. This means that if you take the average over all possible samples of the sample mean, you will exactly obtain the population mean. In the same way, our estimates for β_0 and β_1 are unbiased estimators.

3.1.2.5. Continuing the analogy, we can ask how accurate a sample mean $\hat{\mu}$ as an estimate of μ. We answer this question by computing the **standard error** of $\hat{\mu}$:

$$\mathrm{Var}(\hat{\mu}) = \mathrm{SE}(\hat{\mu})^2 = \frac{\sigma^2}{n}$$

where σ is the standard deviation of the population. This roughly tells us the average amount that $\hat{\mu}$ differs from μ.

The standard errors associated with our estimates for β_0 and β_1 are:

Standard errors of least squares regression coefficient estimates:

$$\mathrm{SE}(\hat{\beta}_0)^2 = \sigma^2 \left[\frac{1}{n} + \frac{\overline{x}}{\sum_{i=1}^{n}(x_i - \overline{x})^2} \right]$$

$$\mathrm{SE}(\hat{\beta}_1)^2 = \frac{\sigma^2}{\sum_{i=1}^{n}(x_i - \overline{x})^2}$$

where $\sigma^2 = \mathrm{Var}(\epsilon)$.

3.1.2.6. The derivation of these formulas will come later, when we study multiple linear regression. (see `https://stats.stackexchange.com/questions/44838/how-are-the-standard-errors-of-coefficients-calculated-in-a-regression/44841#44841`)

3.1.2.7. Notice that $\mathrm{SE}(\hat{\beta}_0)$ is equal to $\mathrm{SE}(\hat{\mu})$ if $\overline{x} = 0$ (in which case, $\hat{\beta}_0 = \overline{y}$).

3.1.2.8. Notice that $\mathrm{SE}(\hat{\beta}_1)$ is gets smaller the more spread out the x_i gets.

3.1.2.9. We do not know σ^2. We can estimate it from the data. The estimate is called **residual standard error**:

$$\mathrm{RSE} = \sqrt{\frac{\mathrm{RSS}}{n-2}}$$

When we use this estimate, the notation should be $\widehat{\mathrm{SE}}(\hat{\beta}_1)$ to indicate that an estimate has been made, but this hat will be dropped.

3.1.2.10. Standard errors can be used to compute **confidence intervals**. "A 95% confidence interval is defined as a range of values such that with 95% probability, the range will contain the true unknown value of the parameter." This statement seems wrong. According to frequentist statistics, we cannot make a statement like this. We do not treat β_1 as a random variable – it is a fixed quantity so there is no probability distribution that we can use to describe β_1.

3.1.2.11. The 95% confidence interval has the following property. If we take different random samples, and construct the confidence interval for each sample, 95% of the confidence intervals will contain the true, unknown value of the parameter.

3.1.2.12. For linear regression, the 95% confidence interval for β_1 is of the form:

$$\hat{\beta}_1 \pm 2 \cdot \mathrm{SE}(\hat{\beta}_1)$$

Similarly for β_0.

3.1.2.13. We can also use standard errors to do **hypothesis tests** on the coefficients. We test the **null hypothesis**:

$$H_0 : \text{There is no relationship between } X \text{ and } Y.$$

versus the **alternative hypothesis**.

$$H_1 : \text{There is some relationship between } X \text{ and } Y.$$

3.1.2.14. We can rewrite these hypotheses mathematically:

$$H_0 : \beta_1 = 0$$

$$H_1 : \beta_1 \neq 0$$

3.1.2.15. To test the null hypothesis, we need to determine whether $\hat{\beta}_1$ is sufficiently far from 0 that we can be confident β_1 is non-zero. "Sufficiently far" is determined by $\text{SE}(\hat{\beta}_1)$ – if $\text{SE}(\hat{\beta}_1)$ is small, then even small values of $\hat{\beta}_1$ may suggest that $\beta_1 \neq 0$.

3.1.2.16. We compute a t**-statistic**, given by:

$$t = \frac{\hat{\beta}_1 - 0}{\text{SE}(\hat{\beta}_1)}$$

If we assume the null hypothesis, then we expect the distribution of t to be a t**-distribution** with $n-2$ degrees of freedom. We can then compute the probability of observing a number equal to $|t|$ or larger in absolute value, assuming $\beta_1 = 0$. We call this the p**-value**.

3.1.2.17. If the p-value is small, then we **reject the null hypothesis**, and thus conclude that there is a relationship between X and Y.

3.1.3 Assessing the Accuracy of the Model

3.1.3.1. If we have rejected the null hypothesis, thus concluding that there is a relationship between X and Y, then we want to know the extent to which the model fits the data. For this, we use the residual standard error (RSE) and the R^2 statistic.

3.1.3.2. Recall the RSS, the residual sum of squares:

$$RSS = \sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

3.1.3.3. Recall also that from our regression model, associated with each observation is an error term ϵ. The **residual standard error** or RSE is an estimate of the standard deviation of ϵ. Roughly, it is the average amount the response will deviate from the true regression line. In the case of simple regression, RSE is computed as:

$$\text{RSE} = \sqrt{\frac{1}{n-2}\text{RSS}} = \sqrt{\frac{1}{n-2}\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}$$

There is a more general formula in the case of multiple linear regression.

3.1.3.4. The RSE is considered a measure of the **lack of fit** of the model to the data.

3.1.3.5. The R^2 statistic is also a measure of fit, but it is not measured in units of Y and so it is more easily interpretable.

3.1.3.6. R^2 measures the proportion of variance explained. It is always between 0 and 1. We define the **total sum of squares** (TSS) to be:

$$\text{TSS} = \sum_{i=1}^{n} (y_i - \overline{y})^2$$

(this is the error of the empty model). In this case, the amount of error explained by the model is $TSS - RSS$.

3.1.3.7. R^2 is defined as the proportion of TSS explained by the model:

$$R^2 = \frac{\text{TSS} - \text{RSS}}{\text{TSS}} = 1 - \frac{\text{RSS}}{\text{TSS}}$$

3.1.3.8. We can also see R^2 as the amount of variance in Y explained by X.

3.1.3.9. It's hard to know what a "good" R^2 is. Obviously, the bigger the better, but different fields have different standards. In fields like biology, psychology, marketing, an R^2 at 0.1 or below might be good.

3.1.3.10. **Sample Correlation**, defined as:

$$r = \text{Cor}(X, Y) = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \overline{y})^2}}$$

is also a measure of the linear relationship between X and Y. How does this work?

- The denominator is something like the product of standard deviations.
- The numerator is proportional to the **covariance** between X and Y. Notice that if it tends to be the case that y_i is large when x_i is large (take large here to mean "bigger than the mean value"), then the covariance is large and positive, and so we see a strong correlation.
- Notice that the units on covariance are $[X][Y]$, and correlation is dimensionless.
- One can think of correlation as being a normalization of covariance.
- Regarding goodness of fit, it is easy to show that if $y = mx + b$, then the correlation is sgn(m).
- It's also easy to show that if X and Y are independent, that is $\mathbb{E}[XY] = \mathbb{E}[X]\mathbb{E}[Y]$, then $\text{Cor(X, Y)} = 0$.
- From the Cauchy-Schwartz inequality $|\langle u, v\rangle| \leq ||u|| \cdot ||v||$, we can see that r is between 0 and 1.
- There is also **population correlation**. This is given by:

$$\rho_{X,Y} = \frac{\text{Cov}(X, Y)}{\sigma_X \sigma_Y}, \qquad \text{where} \qquad \text{Cov}(X, Y) = \mathbb{E}[X - \mu_X] \cdot \mathbb{E}[Y - \mu_Y]$$

3.1.3.11. In the context of simple linear regression, it turns out that $R^2 = r^2$. This is not quite the case to multiple linear regression.

3.2 Multiple Linear Regression

3.2.0.1. If you have multiple predictors, one might think of running multiple simple regressions. However, this is not entirely satisfactory as:

- it is unclear how to predict the outcome variable using multiple simple regressions.
- If there is correlations between predictors, this can lead to poor estimates of the outcome.

3.2.0.2. Instead, we generalize simple linear regression to accommodate p predictors in a linear model. This is called **multiple linear regression**. The model takes the form:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p + \epsilon$$

We interpret β_j as the average effect of Y of a one unit increase in X_j holding all other predictors fixed.

3.2.1 Estimating the Regression Coefficients

3.2.1.1. Just as before, we estimate $\beta_0, \dots \beta_p$, and make predictions using:

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \cdots + \hat{\beta}_p x_p$$

3.2.1.2. We again use the least squares approach and choose $\hat{\beta}_0, \dots \hat{\beta}_p$ to minimize the sum of squared residuals: TYPO in text page 72

$$\text{RSS} = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 = \sum_{i=1}^{n} (y_i - \hat{\beta}_0 + \hat{\beta}_1 x_{i1} - \cdots - \hat{\beta}_p x_{ip})^2$$

3.2.1.3. The text does not give formulas or derivations for the estimates. Their formulas are best presented using matrix algebra. Computers can calculate these estimates easily in `R`. The text also glazes over the computation of standard errors of estimates, t-statistics, and p-values.

3.2.1.4. Situations such as the following can sometimes arise:

- A simple regression can show a statistically significant relationship between Y and X_1.
- However, when multiple regression between Y and $X_1, ...X_p$ is run, it shows that there is no significant relationship between X_1 and Y. This might happen because there is a correlation between X_1 and some other predictor, X_2 say. And we may see that X_2 does have a significant relationship with Y.
- In this case, in the simple regression, X_1 is getting credit for X_2's effect on Y.
- The classic example of this phenomenon is that ice cream sales are correlated with shark attacks at the beach. These are correlated,

3.2.2 Some Important Questions

3.2.2.1. **1. Is there a relationship between the response and predictors?** This is a hypothesis testing question:

$$H_0 : \beta_1 = \beta_2 = \cdots = \beta_p = 0$$

$$H_a : \text{at least one of } \beta_j \text{ is non-zero.}$$

3.2.2.2. We use the F**-statistic** to answer this question:

$$F = \frac{(\text{TSS} - \text{RSS})/p}{\text{RSS}/(n-p-1)}$$

If the linear model assumptions are correct, then one can show that the expected value of the denominator is σ^2. And if H_0 is true, then the expected value of the numerator is σ^2. Therefore, if the null hypothesis is true, we should expect $F = 1$. And if H_a is true, then we expect $\mathbb{E}[(\text{TSS} - \text{RSS})/p] > \sigma^2$, so we expect F to be greater than 1.

3.2.2.3. I would like to see a justification for the two statements above.

- Maybe this is unrelated, but I have seen in several places that

  $$\text{TSS} = \text{MSS} + \text{RSS}$$

  where:

  $$\text{TSS} = \sum_{i=1}^{n} (y_i - \overline{y})^2, \quad \text{MSS} = \sum_{i=1}^{n} (\hat{y}_i - \overline{y})^2, \quad \text{RSS} = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

  This is not obvious! Let's prove it for simple linear regression, though I believe it holds for multiple linear regression as well.

  Proof:

  $$y_i - \overline{y} = (y_i - \hat{y}_i) + (\hat{y}_i - \overline{y})$$

Now square both sides and sum:

$$\text{TSS} = \text{MSS} + \text{RSS} + \sum_{i=1}^{n} 2(y_i - \hat{y}_i)(\hat{y}_i - \overline{y})$$

We want to show the last sum is 0. By simple linear regression, we know that:

$$\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i$$

where

$$\hat{\beta}_0 = \overline{y} - \hat{\beta}_1 \overline{x}, \qquad \text{and} \qquad \hat{\beta}_1 = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sum_{i=1}^{n}(x_i - \overline{x})^2}$$

Therefore:

$$\begin{aligned}
\sum_{i=1}^{n}(y_i - \hat{y}_i)(\hat{y}_i - \overline{y}) &= \sum_{i=1}^{n}(y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i)(\hat{\beta}_0 + \hat{\beta}_1 x_i - \overline{y}) \\
&= \sum_{i=1}^{n}(y_i - \overline{y} + \hat{\beta}_1 \overline{x} - \hat{\beta}_1 x_i)(\overline{y} - \hat{\beta}_1 \overline{x} + \hat{\beta}_1 x_i - \overline{y}) \\
&= \sum_{i=1}^{n}\Big((y_i - \overline{y}) - \hat{\beta}_1(x_i - \overline{x})\Big)\Big(\hat{\beta}_1(x_i - \overline{x})\Big) \\
&= \sum_{i=1}^{n}\hat{\beta}_1(x_i - \overline{x})(y_i - \overline{y}) - \hat{\beta}_1^2(x_i - \overline{x})^2 \\
&= \hat{\beta}_1\left[\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y}) - \hat{\beta}_1\sum_{i=1}^{n}(x_i - \overline{x})^2\right] \\
&= \hat{\beta}_1\left[\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y}) - \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sum_{i=1}^{n}(x_i - \overline{x})^2}\sum_{i=1}^{n}(x_i - \overline{x})^2\right] = 0
\end{aligned}$$

□

- What are the linear model assumptions? This is that the errors ϵ are independently and identically distributed with mean 0 and variance σ^2.
- The numerator of F is referred to as the **MSM, mean square model**. Notice that:

  $$\text{MSM} = \frac{\text{MSS}}{p}$$

  we should view p as the degrees of freedom in the model. We can see this arise if we examine the degrees of freedom in the alternate realization of MSS as TSS − RSS. The degrees of freedom of TSS is $n-1$, and the degrees of freedom remaining after incorporating p predictors is $n-p-1$. The difference of these is p.
- The denominator is referred to as the **MSE, or mean squared error**. Notice that:

  $$\text{MSE} = \frac{\text{RSS}}{n-p-1}$$

  which is in line with what we have above – sum of squared deviation divided by degrees of freedom.
- As an aside to this aside, the reason we divide by these degrees of freedoms is to get unbiased estimators, meaning, in this case, that the expected value of MSM and MSE should both be σ^2.
- I'm starting to realize that this unbiased estimator thing is going to require a tremendous amount of work. For this reason, I will build up a separate document. I think I also should understand multiple linear regression's exposition in terms of matrix algebra before really attempting to understand why MSS/p and RSS/$(n-p-1)$ are unbiased – I might be able to prove it for $p=1$ though.

3.2.2.4. How large does the F-statistic need to be before we reject H_0? This depends on n and p. When n is large, F can be just a bit larger than 1 to obtain a sufficiently small p-value.

3.2.2.5. When H_0 is true and the errors ϵ_i are normally distributed, the F-statistic follows an F-distribution.

3.2.2.6. There is no discussion about the F-distribution, or details of computing the p-value of a F-statistic in the text. The text does say that `R` can do it.

3.2.2.7. Suppose we want to show that a subset of the coefficients are 0. This corresponds to a null hypothesis:

$$H_0 : \beta_{p-q+1} = \beta_{p-q+2} = \cdots = \beta_p = 0$$

A second model is fit which uses all the predictors except these q, and a residual sum of squares RSS_0 is computed. This quantity RSS_0 is used in place of TSS in the computation of the F-statistic (when $q = p$, this second model is the empty model). That is:

$$F = \frac{(\mathrm{RSS}_0 - \mathrm{RSS})/q}{\mathrm{RSS}/(n-p-1)}$$

3.2.2.8. One can use this to test the significance of each individual predictor while holding the others fixed. In fact, the F-statistic so computed is the square of a t-statistic associated to the predictor in the multiple linear regression.

3.2.2.9. Here is an example relating three predictors to a response variable called `sales`.

	Coefficient	Std. error	t-statistic	p-value
`Intercept`	2.939	0.3119	9.42	< 0.0001
`TV`	0.046	0.0014	32.81	< 0.0001
`radio`	0.189	0.0086	21.89	< 0.0001
`newspaper`	-0.001	0.0059	-0.18	0.8599

If we were to compute the F-statistic where $q = 1$ and we omit `newspaper`, then the resulting value would be $(-0.18)^2$.

3.2.2.10. It is important in hypothesis testing to use the F-statistic instead of relying on the p-values in the above table for each individual predictor, particularly when the number of predictors p is large. This is because we can achieve a small enough p-value (< 0.05) simply by chance alone with a large number of predictors. In light of this, it would be foolish to reject the null hypothesis with this sort of method.

3.2.2.11. If $p > n$, then there are more coefficients to estimate than observations from which to estimate them. In this case, multiple linear regression using least squares does not make sense. This situation is referred to as a **high-dimensional** setting, as is discussed later in the text.

3.2.2.12. **2. Which variables are important?** If we decide to reject the null hypothesis, it is natural to wonder which of the predictors is related to the response. The task of determining which predictors are associated with the response in order to fit a single model involving only those predictors is referred to as **variable selection.**

3.2.2.13. As we discussed, it would be foolish to look at individual p-values for the predictors, especially when the number of predictors is large.

3.2.2.14. We might try constructing the 2^p possible models using one of the 2^p subsets of the predictors. But how do we determine which model is best? There are various statistics which can be used to judge the quality of a model – these will be discussed in Chapter 6. However, 2^p grows quickly with p, thus rendering this strategy infeasible.

3.2.2.15. There are three classical approaches to select models to consider:

- **Forward selection:** Start with the **null model** (aka empty model). Perform the p simple linear regressions and add to the null model the variable with the lowest RSS. Then do $p-1$ two-variable regressions where one of the other variables is added. Keep going until some stopping rule is satisfied.
- **Backward selection:** Start with all variables, then remove the variable with the largest p-value. Fit the new $(p-1)$-variable model, then remove the largest p-value variable. Continue until a stopping rule is reached – perhaps stop when all variables have a p-value below a threshold.

- **Mixed selection:** Start with no variables. Add the variable of best fit. Continue to add variables, and if the p-value for any of the variables rises above a threshold, remove it. Continue until all variables in the model have low p-value, and all other variables would have a give large p-values if added.

3.2.2.16. **3. Model Fit** The two most common numerical measures of model fit are RSE and R^2.

3.2.2.17. We saw that in simple regression of Y onto X, R^2 turns out to be the square of the correlation between X and Y. In multiple linear regression, it turns out that

$$R^2 = \mathrm{Cor}(Y, \hat{Y})^2$$

Theorem: $\mathrm{Cor}(Y, \hat{Y}) = |\mathrm{Cor}(X, Y)|$ in the case of simple regression $\hat{Y} = \hat{\beta}_0 + \hat{\beta}_1 X$.

Proof:

$$\begin{aligned}
\mathrm{Cor}(Y, \hat{Y}) = \mathrm{Cor}(Y, \hat{\beta}_0 + \hat{\beta}_1 X) &= \frac{\sum_{i=1}^{n} (y_i - \overline{y})(\hat{\beta}_0 + \hat{\beta}_1 x_i - (\overline{\hat{\beta}_0 + \hat{\beta}_1 x_i})}{\sqrt{\sum_{i=1}^{n} (y_i - \overline{y})^2} \sqrt{\sum_{i=1}^{n} (\hat{\beta}_0 + \hat{\beta}_1 x_i - (\overline{\hat{\beta}_0 + \hat{\beta}_1 x_i})^2}} \\
&= \frac{\sum_{i=1}^{n} (y_i - \overline{y})(\hat{\beta}_0 + \hat{\beta}_1 x_i - \hat{\beta}_0 - \hat{\beta}_1 \overline{x})}{\sqrt{\sum_{i=1}^{n} (y_i - \overline{y})^2} \sqrt{\sum_{i=1}^{n} (\hat{\beta}_0 + \hat{\beta}_1 x_i - \hat{\beta}_0 - \hat{\beta}_1 \overline{x})^2}} \\
&= \frac{\sum_{i=1}^{n} (y_i - \overline{y})(\hat{\beta}_1 (x_i - \overline{x}))}{\sqrt{\sum_{i=1}^{n} (y_i - \overline{y})^2} \sqrt{\sum_{i=1}^{n} \hat{\beta}_1^2 (x_i - \overline{x})^2}} \\
&= \frac{\hat{\beta}_1}{|\hat{\beta}_1|} \cdot \frac{\sum_{i=1}^{n} (y_i - \overline{y})(x_i - \overline{x})}{\sqrt{\sum_{i=1}^{n} (y_i - \overline{y})^2} \sqrt{\sum_{i=1}^{n} (x_i - \overline{x})^2}} = \mathrm{sgn}(\hat{\beta}_1) \cdot \mathrm{Cor}(X, Y)
\end{aligned}$$

□

3.2.2.18. Every additional variable will decrease R^2. However, if adding a variable leads only to a small increase in R^2, then we run into the problem of potential overfitting – the model may be picking up on noise.

3.2.2.19. Recall that the RSE, the **residual standard error** is an estimate of the standard deviation of the residual. In general, RSE is defined as:

$$\mathrm{RSE} = \sqrt{\frac{1}{n - p - 1} \mathrm{RSS}}$$

3.2.2.20. Adding a variable to a model will always cause RSS to decrease. However, RSE may nonetheless increase since the denominator gets smaller.

3.2.2.21. **4. Prediction** The multiple regression model allows us to make predictions on the response variable using a values for the predictors $X_1, ...X_p$. Where does the uncertainty in our prediction $\hat{Y}$ lie?

- The coefficient estimates is a form of reducible error. We compute a confidence interval to account for this error.
- Assuming a linear model is an additional source of potentially reducible error called **model bias**.
- There is also the irreducible error ϵ. To account for this, we use a **prediction interval**. This is an interval which incorporates both the reducible error in our coefficient estimates, as well as the irreducible error.

- The confidence interval is used to quantify the uncertainty surrounding the average value of $f(X)$ – if we collect a large number of data sets with observations in X, and construct confdience intervals for each, then 95% of these will contain the true value of the average of $f(X)$. The prediction interval is an estimate of the uncertainty around a single predicted value. It provides a range that we expect a new observation to fall within given the values of the predictors. It not only captures uncertainty in the parameters (like the confidence interval), but also random error in the observation.

3.2.2.22. The text discusses confidence intervals and prediction intervals for a particular set of inputs. It gives no indication of how these were constructed.

3.3 Other Considerations in the Regression Model

3.3.1 Qualitative Predictors

3.3.1.1. Another term for a qualitative predictor is **factor.** A factor can have **levels**, or possible values.

3.3.1.2. If a factor has two levels, (for example FALSE and TRUE)then we can create an **indicator** or **dummy variable** which is either 0 or 1 depending on the value of the factor.

3.3.1.3. We then use this in a regression model:

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i = \begin{cases} \beta_0 + \beta_1 + \epsilon_i & \text{if } x_i = \text{TRUE} \\ \beta_0 + \epsilon_i & \text{if } x_i = \text{FALSE} \end{cases}$$

3.3.1.4. Here, β_0 is the average value of Y among those observations for which X is FALSE.

3.3.1.5. And β_1 is the average difference in values in Y between those observations for which X is TRUE and X is FALSE.

3.3.1.6. The encoding of the dummy variable is arbitrary. Changing the values of x_i to be -1 and 1, say, will change the interpretation of the resulting regression coefficients, but will not change the predictions.

3.3.1.7. If a factor has more than two levels, then we use additional dummy variables. For example, if X can take values {For (F), Neutral (N), Against (A)}, then you need two dummy variables. Perhaps:

$$x_{i1} = \begin{cases} 1 & \text{if } X = \text{F} \\ 0 & \text{if } X \neq \text{F} \end{cases}$$

and

$$x_{i2} = \begin{cases} 1 & \text{if } X = \text{A} \\ 0 & \text{if } X \neq \text{A} \end{cases}$$

Then our regression model:

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 xi2 + \epsilon_i = \begin{cases} \beta_0 + \beta_1 + \epsilon_i & \text{if } X = \text{F} \\ \beta_0 + \beta_2 + \epsilon_i & \text{if } X = \text{A} \\ \beta_0 + \epsilon_i & \text{if } X = \text{N} \end{cases}$$

3.3.1.8. There will always be one fewer dummy variable than the number of levels.

3.3.1.9. The level for which all dummy variables are 0 is known as the **baseline**.

3.3.1.10. There is no issue with mixing quantitative and qualitative variables in a regression model.

3.3.2 Extensions of the Linear Model

3.3.2.1. Linear regression assumes that the relationship between predictors and response is both **additive** and **linear**. Additive means that the relationship between a predictor X_j and response Y does not depend on any of values of the other predictors. Linear means that the change in the response Y associated with a one-unit change in X_j is constant, regardless of the value of X_j.

3.3.2.2. Suppose there is a **synergy** (aka **interaction**) between predictors X_1 and X_2, meaning that the value of, say, X_2 can have an effect on the relationship of X_1 on Y.

3.3.2.3. For example, suppose you have a fixed $100,000 budget to spend on advertising. You fit the response `sales` with predictors `tv` and `radio`. A regression model would imply that to maximize `sales`, one should plunge the full budget into the media (tv or radio) with the higher coefficient – let's say that's tv. However, it could be the case that spending half on radio and half on tv increases sales more than giving the entire amount to tv. This is an interaction effect.

3.3.2.4. One way to extend the linear model to include interactions is to include **interaction terms** constructed by computing the product of X_1 and X_2:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_1 X_2 + \epsilon$$

In this model, changing the value of X_2 will change the association between X_1 and Y.

3.3.2.5. In regression, non-interaction terms are called **main effects**.

3.3.2.6. It can be the case that an interaction term has a small p-value, but the main effects do not. However, the **hierarchical principle** states that if an interaction term is included in a model, then the model should also include the main effects, even if the p-values associated with their coefficients are not significant. I don't fully understand the rationale for why this principle should be adopted and I don't find the following to be very convincing. The rationale for this principle is that if $X_1 X_2$ is related to the response, then whether or not the coefficients of the main effects are zero is of little interest.

3.3.2.7. Interaction terms can involve qualitative variables as well – this is straightforward.

3.3.2.8. Non-linear relationships can be fit using **polynomial regression**. For example, we can fit a model:

$$Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \epsilon$$

This is still a linear model.

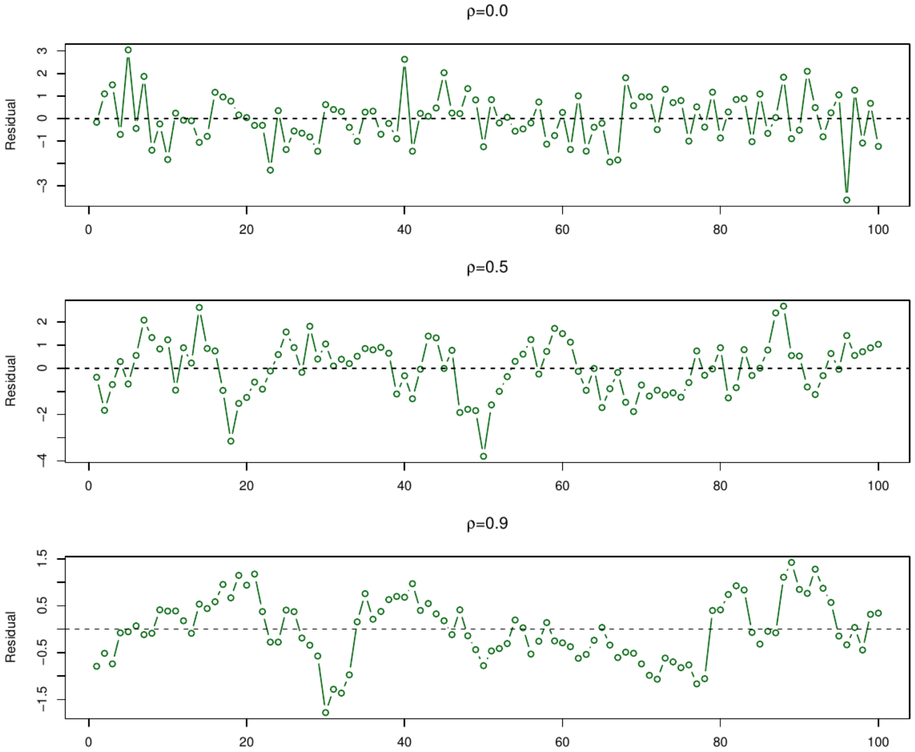
3.3.3 Potential Problems

3.3.3.1. Many problems may occur when fitting a linear regression model.

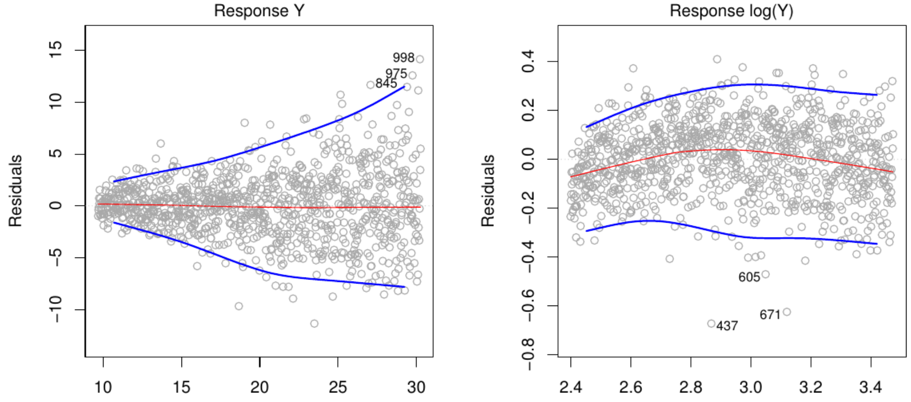
- **Non-linearity of the response-predictor relationships.** A good way of identifying non-linearity are **residual plots** which plot $e_i = y_i - \hat{y}_i$ versus the predictor x_i. The presence of a pattern in a residual plot may indicate a problem with a linear assumption.
- **Correlation of error terms**. If there is correlation among error terms, then estimated standard errors will tend to underestimate true standard errors, which leads to a narrowing of confidence and prediction intervals. This also leads to a smaller p-value. A good example of this is if we were to double all observations increasing the sample size to $2n$. Estimated parameters would not change, but confidence intervals would be narrower by a factor of $\sqrt{2}$. I don't really understand the problem with this. If the data is doubled, and we had two observations for each data point, isn't it sensible that standard error is reduced? ChatGPT agrees with me for what that's worth.

  Correlations between error terms frequently occur in **time series** data, which consists of observations at discrete points in time. Correlations can be made evident by plotting residuals against time. Correlation may appear as adjacent observations have correlated error terms – this is called **tracking**.

  Pictured below is an example of correlated error terms at different levels of corrleation.



- **Non-constant variance of error terms** Also known as **heteroscedasticity**. Can be detected in a funnel shape in the residual plot. When possible solution is to transform the response using a concave function: $\log(Y) or \sqrt{Y}$ for instance – this shrinks larger values in response which leads to a reduction in heteroscedasticity.



Sometimes, variance in response is well understood. For instance, if Y_i is the average of n_i observations, each of which has known variance σ^2, then Y_i is known to have variance $\frac{\sigma^2}{n_i}$. Then a simple remedy is to model by **weighted least squares** – in this example, make weights $w_i = n_i$ so so to make the error constant.

- **Outliers** An **outlier** is a point for which y_i is far from $\hat{y}_i$. Again, residual plots make outliers identifiable though in practice it can be difficult to decide how large a residual can be before it is considered an outlier. Thus, we can instead plot **studentized residuals** which divides each residual e_i by its estimated standard error. We then use the rule that responses which are greater than 3 in absolute standardized residual are outliers.

  Outliers may indicate a missing predictor.

- **High-leverage points** are observations with an unusual value for predictors. Note that even if an observation has values for predictors which fall within the ranges of predictors, the conjunction of values for the predictors can cause an observation to be a high leverage point. We can quantify an observations leverage using the **leverage statistic**. For simple linear regression, this is:

$$h_i = \frac{1}{n} + \frac{(x_i - \overline{x})^2}{\sum_{j=1}^{n}(x_j - \overline{x})^2}$$

The average value of this statistic is $2/p$, so if an observation has a leverage much greater than $2/p$, we may suspect a high-leverage point.

High leverage points have a large effect on the regression line.

- **Collinearity** refers to when predictor variables are highly correlated. This is a problem for regression since it can be difficult to tease apart the association between correlated variables and the response.

  Imagine that X_1 and X_2 are highly correlated. Then for a fixed value of RSS, there is a large range of possible values of $\hat{\beta}_1$ and $\hat{\beta}_2$ which can give the value of RSS. Another way to say this is that there is a large standard error on these coefficient estimates. This causes the t-statistic to decrease, which causes the p-value to grow and confidence intervals to widen. This in turn makes it harder to reject the null hypothesis in regression – that is, the **power** of the hypothesis test which is the probability of correctly detecting a non-zero coefficient is reduced by collinearity.

  Collinearity can be detected by looking at the correlation matrix of predictors. However, it is possible for collinearity to exist between three or more variables even if no pair of variables has a particularly high correlation. This is called **multicollinearity**.

  To assess multicollinearity, we use the **variance inflation factor** (VIF). This is the ratio of the variance of $\hat{\beta}_j$ when fitting the full model divided by the variance of $\hat{\beta}_j$ if fit on its own. The smallest possible value for VIF is 1, which is the complete absence of collinearity.

  This implies that the variance of $\hat{\beta}_j$ is larger when fitting the full model than when fitting the simple model. Why is this so? What is the variance of $\hat{\beta}_j$ anyway? ChatGPT says that the variance of $\hat{\beta}_j$ is:

  $$\mathrm{Var}(\hat{\beta}_j) = \frac{\sigma^2}{\sum x_j^2}$$

  I'm really getting the feeling that I need to learn multiple linear regression.

  The VIF for each variable can be computed using the formula:

  $$\mathrm{VIF}(\hat{\beta}_j) = \frac{1}{1 - R^2_{X_j|X_{-j}}}$$

  where $R^2_{X_j|X_{-j}}$ is the R^2 from a regression of X_j onto all the other predictors. If $R^2_{X_j|X_{-j}}$ is close to one, then collinearity is present, so the VIF will be large.

  When collinearity is present, some solutions might be to drop one of the variables, or to create a new variable which is a mixture of collinear variables.

3.4 The Marketing Plan

3.5 Comparison of Linear Regression with K-Nearest Neighbors

3.5.0.1. In **K-nearest neighbors regression** (KNN regression), given a value for $K \in \mathbb{N}$ and a prediction point x_0, KNN regression considers the K training observations that are closest to x_0 – denote this set by $\mathcal{N}_0$. Then:

$$\hat{f}(x_0) = \frac{1}{K}\sum_{x_i \in \mathcal{N}_0} y_i$$

3.5.0.2. KNN regression is a non-parametric method.

3.5.0.3. Notice that if $K = 1$, then KNN regression gives perfect interpolation. As K increases, we get a smoother and less variable fit since the prediction is an average of several points. However, this smoothing may cause bias by masking some of the structure in $f(X)$.

3.5.0.4. The parametric approach will outperform the non-parametric approach if the parametric form that has been selected is close to the true form of f. In this case, the non-parametric approach incurs a cost in variance which is not offset by a reduction in bias.

3.5.0.5. The more non-linear the relationship between predictors and output is, the better KNN will perform. On the other hand, linear regression may still win out when the number of predictors is large. This is because of the **curse of dimensionality.** When data is distributed in a high-dimensional space, distances between data points tend to equalize.

3.5.0.6. One way to think about this is to examine the ratio of the volume of a spherical ball D^n of radius r to the volume of an n-cube I^n with edgelength $2r$. We have:

$$\mathrm{Vol}(D^n) = \frac{2r^n \pi^{n/2}}{n\Gamma(n/2)}$$

And so:

$$\lim_{n\to\infty} \frac{\mathrm{Vol}(D^n)}{\mathrm{Vol}(I^n)} = \lim_{n\to\infty} \frac{\pi^{n/2}}{n2^{n-1}\Gamma(n/2)} = 0$$

So if we choose data randomly sampled from I^n, when n is large, these tend to fall outside the spherical ball.

But they also tend to stay away from the corners! In fact, the volume of the cube is concentrated around a sphere of radius $\sqrt{n}/3$. To show this, let's consider the average value of x_i^2 where x_i is a coordinate of a point in the cube of edgelength 2:

$$\begin{aligned}\mathbb{E}[x_i^2] &= \frac{1}{2}\int_{-1}^{1} x^2\,dx \\ &= \frac{1}{2}\left[\frac{1}{3}x^3\Big|_{-1}^{1}\right] \\ &= \frac{1}{3}\end{aligned}$$

and the variance:

$$\begin{aligned}\mathrm{Var}(x_i^2) &= \frac{1}{2}\int_{-1}^{1} x^4\,dx - (\mathbb{E}[x_i^2])^2 \\ &= \frac{1}{2}\cdot\frac{2}{5} - \frac{1}{9} = \frac{4}{45}\end{aligned}$$

Therefore, the squared distance of a point from the origin $r^2 = \sum_i^n x_i^2$ has average value $n/3$ and variance $4n/45$. We can apply the CLT to the average r^2/n (which is analogous to the sample mean) to conclude that this is close to a normal distribtuion with mean $1/3$ and standard deviation $2/\sqrt{45n}$. Thus, most of the volume of the cube is concentrated near the sphere of radius $\sqrt{n/3}$.

4 Classification

4.0.0.1. **Classification** refers to approaches for predicting qualitative or categorical responses. A particular classfication technique is called a **classifier**.

4.1 An Overview of Classification

4.1.0.1. Classification problems occur often, maybe even more so than regression problems.

4.1.0.2. Some examples:

- A person visits the doctor exhibiting a set of symptoms which might point to one of three different diagnoses. Which of the three conditions does the individual have?
- An online banking service must determine if a transaction is fraudulent on the basis of a user's IP address, past transaction history, and so forth.
- Which DNA mutations are deleterious (disease-causing), and which are not.
- Will a credit card holder default `Default` on their payments? We will base our prediction on annual income `income` and card balance `balance`. This is the problem that will be highlighted in this chapter.

4.2 Why Not Linear Regression?

4.2.0.1. Linear regression is not appropriate in the case of a qualitative response. Two main reasons:

- Regression methods cannot accommodate a qualitative response with more than two classes since these classes must be coded and this coding can be arbitrary.
- A regression method will not provide meaningful probability estimates when Y is a variable with two classes.

4.3 Logistic Regression

4.3.0.1. **Logistic regression** models the probability that Y – a two valued response variable – takes on a particular value.

4.3.0.2. In the example of `default`, logistic regression models the probability of default given a value of `balance`. We will abbreviate $P(\texttt{default}|\texttt{balance})$ by $p(\texttt{balance})$.

4.3.0.3. One might predict $\texttt{default} = \texttt{Yes}$ for any individual for whom $p(\texttt{balance}) > \texttt{0.5}$. Though one could be more conservative and put the threshold at 0.1 instead.

4.3.1 The Logistic Model

4.3.1.1. In general, in logistic regression, we want to find a relationship between X and $p(X) = P(Y = 1|X)$. Here we are using a generic $0/1$ coding for the values of Y.

4.3.1.2. If we were to use linear regression:

$$p(x) = \beta_0 + \beta_1 X$$

we may very well obtain values outside the range of $[0, 1]$ that we want for probabilities, which is a problem.

4.3.1.3. In logistic regression, we use the **logistic function** to model probabilities:

$$p(X) = \frac{e^{\beta_0+\beta_1 X}}{1 + e^{\beta_0+\beta_1 X}}$$

As we let X range, this produces an S-shaped curve ranging between 0 and 1.

4.3.1.4. We can manipulate the logistic function to obtain:

$$\frac{p(X)}{1 - p(X)} = e^{\beta_0+\beta_1 X}$$

The expression on the left is called the **odds**.

4.3.1.5. If we take the log of both sides, we obtain:

$$\log\left(\frac{p(X)}{1 - p(X)}\right) = \beta_0 + \beta_1 X$$

The left hand side is called the **log odds** or **logit**.

4.3.1.6. We conclude that the logistic regression model has a logit which is linear in X.

4.3.1.7. Increasing X by 1 unit changes the log odds by β_1. This corresponds to multiplying the odds by e^{β_1}.

4.3.2 Estimating the Regression Coefficients

4.3.2.1. Using training data, we will estimate β_0 and β_1 in the logistic regression model.

4.3.2.2. To do this, we do not use least squares (though we could). We instead use a method known as **maximum likelihood** since it has better statistical properties.

4.3.2.3. Maximum likelihood estimation is a huge subject used all over statistics whenever a parameter estimate is required. The general idea is to maximize a likeihood function so that, under the assumed statistical model, the observed data is most probable.

The mathematical details are suppressed in the text. I have looked for texts which might shed some light – Casella and Berger *Statistical Inference* comes highly receommended.

4.3.2.4. We can use `R` to compute MLE for β_0 and β_1 in logistic regression.

4.3.2.5. We typically don't care too much about the significance of β_0 since it's main purpose is to adjust the average fitted probabilities to the proportion of ones in the data.

4.3.2.6. The significance of β_1 is of interest. We can use a z-statistic and associated p-value to do hypothesis testing on whether $\beta_1 = 0$. Of course, if the null hypothesis is rejected, this suggests there is a relationship between X and $p(X)$.

4.3.3 Making Predictions

4.3.3.1. Predictions are done in the obvious way once estimates $\hat{\beta}_0$ and $\hat{\beta}_1$ are computed – just plug in the desired value for X.

4.3.3.2. Qualitative predictors can be used in logistic regression. For example, if we wanted to know the probability of a student defaulting vs a non-student defaulting, we could do the usual dummy variable trick of coding a student as 1 and a non-student as 0 and computing the logistic regression model as usual.

4.3.4 Multiple Logistic Regression

4.3.4.1. To predict a binary response using multiple predictors, we generalize:

$$\log\left(\frac{p(x)}{1-p(X)}\right) = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X$$

As before, we use MLE to estimate the β's.

4.3.4.2. In the running example, we can fit logistic regression to `default` on `balance`, `income`, and `student` status. A funny thing happens here.

- When you do logistic regression using `student` only, you get a positive coefficient which implies that students default at a higher rate than non-students.
- However, in this multiple logistic regression, we obtain a *negative* coefficient for `student`. This implies that for a fixed value of `balance` and `income`, the model predicts students default at rates lower than non-students.
- Running simple logistic regression of `default` against `balance` shows that the higher the balance, the higher the default rate.
- It turns out that `student` is positively correlated with `balance`. In other words, students tend to have higher balances. Thus, a higher proportion of students default because they tend to have higher balances.
- This sort of phenomenon is called **confounding.**

4.3.5 Multinomial Logistic Regression

4.3.5.1. **Multinomial logistic regression** extends logistic regression to response variables with $K > 2$ classes.

4.3.5.2. We first select a class to serve as the **baseline**. WLOG, this will be the Kth class.

4.3.5.3. The model is:

$$\begin{aligned} P(Y = k|X = x) &= \frac{e^{\beta_{k0}+\beta_{k1}x_1+\cdots+\beta_{kp}x_p}}{1+\sum_{l=1}^{K-1} e^{\beta_{l0}+\beta_{l1}x_1+\cdots+\beta_{lp}x_p}}, \qquad k = 1, \ldots, K-1 \\ P(Y = K|X = x) &= \frac{1}{1+\sum_{l=1}^{K-1} e^{\beta_{l0}+\beta_{l1}x_1+\cdots+\beta_{lp}x_p}} \end{aligned}$$

This is clearly a probability distribution on $Y = 1, ..., K$.

4.3.5.4. This model has $(p + 1)(K - 1)$ parameters.

4.3.5.5. It is easy to show that the log-odds between any pair of classes is linear in the features. For instance:

$$\log\left(\frac{P(Y = k|X = x)}{P(Y = K|X = x)}\right) = \beta_{k0} + \beta_{k1}x_1 + \cdots + \beta_{kp}x_p, \qquad k = 1, 2, \ldots, K-1$$

and for $k \neq K$ and $j \neq K$:

$$\begin{aligned} \log\left(\frac{P(Y = k|X = x)}{P(Y = j|X = x)}\right) &= \log\left(\frac{\exp(\beta_{k0} + \beta_{k1}x_1 + \cdots + \beta_{kp}x_p)}{\exp(\beta_{j0} + \beta_{j1}x_1 + \cdots + \beta_{jp}x_p)}\right) \\ &= (\beta_{k0} - \beta_{j0}) + (\beta_{k1} - \beta_{j1})x_1 + \cdots + (\beta_{kp} - \beta_{jp})x_p \end{aligned}$$

4.3.5.6. In multinomial regression, the choice of the baseline predictor is unimportant. However, the interpretation of the multinomial logistic regression model is tied to the choice of baseline. For instance, β_{k0} is the log odds of $Y = k$ versus $Y = K$ when all predictors are 0.

4.3.5.7. **Softmax coding**, used extensively in ML, is an alternative scheme for coding multinomial logistic regression, but which is equivalent in the sense that all fitted values, log odds between any pair of classes, and other key model outputs will remain the same. Rather than picking a baseline, all classes are symmetrically treated:

$$P(Y = k|X = x) = \frac{\exp(\beta_{k0} + \beta_{k1}x_1 + \cdots + \beta_{kp}x_p)}{\sum_{i=1}^{K} \exp(\beta_{i0} + \beta_{i1}x_1 + \cdots + \beta_{ip}x_p)}$$

Here, we have the log odds between class k and j (as before, but now including K):

$$\begin{aligned} \log\left(\frac{P(Y = k|X = x)}{P(Y = j|X = x)}\right) &= \log\left(\frac{\exp(\beta_{k0} + \beta_{k1}x_1 + \cdots + \beta_{kp}x_p)}{\exp(\beta_{j0} + \beta_{j1}x_1 + \cdots + \beta_{jp}x_p)}\right) \\ &= (\beta_{k0} - \beta_{j0}) + (\beta_{k1} - \beta_{j1})x_1 + \cdots + (\beta_{kp} - \beta_{jp})x_p \end{aligned}$$

4.3.5.8. So why would one ever NOT use softmax? The text doesn't go into it so I asked ChatGPT:

1. Ordinal Outcomes: Softmax coding treats each category as separate from all the others. If your outcome variable is ordinal, meaning the categories have a specific order (like "low", "medium", "high"), then it could make more sense to use an ordinal logistic regression model, which respects this order and can sometimes provide better predictions.

2. Computational Concerns: In very large datasets or models with a high number of classes, softmax regression can be computationally expensive because it essentially requires estimating a separate model for each category. In these cases, other approaches like decision trees or random forests might be more efficient.

3. Extreme Class Imbalance: In situations with extreme class imbalance (where one or a few categories have many more observations than the others), one-vs-all coding could lead to models that are biased towards the majority class. There are techniques to handle class imbalance, but in some cases, alternative strategies like one-vs-one coding could be more appropriate.

4. Multilabel Problems: In problems where each observation can belong to more than one category, softmax coding may not be applicable because it assumes that each observation belongs to exactly one category. In these cases, other methods such as binary relevance (training a separate model for each category) might be more appropriate.

4.4 Generative Models for Classification

4.4.0.1. An alternative, and less direct approach to modeling the conditional distribution of the variable Y given X is to model the distribution of X for each of the response classes. Then use Bayes' theorem to flip these around to obtain estimates for the conditional distribution.

4.4.0.2. When the distribution of X within each class is normal, the model is similar to logistic regression (no proof in text).

4.4.0.3. Some potential advantages over logistic regression:

- When there is substantial separation between two classes, parameter estimates in logistic regression can be suprisingly unstable. The methods of this subsection do not suffer from this problem.
- If distribution of X given Y is approximately normal and the sample size is small, then the methods of this subsection may be more accurate.

4.4.0.4. Suppose Y can take on one of $K \geq 2$ values. Let π_k denote the **prior** probability of a randomly chosen observation comes from the kth class. Let $f_k(X) = P(X|Y = k)$ denote the **density function of** X. **Bayes' theorem** states:

$$P(Y = k|X = x) = \frac{\pi_k f_k(x)}{\sum_{i=1}^{K} \pi_i f_i(x)}$$

We will use $p_k(x) = P(Y = k|X = x)$ for the **posterior**.

4.4.0.5. Estimating the prior π_k is easy if we have a random sample of observations from the population – we just compute the fraction of training observations that belong to the kth class.

4.4.0.6. Estimating the density $f_k(x)$ is much more challenging. We will typically have to make some simplifying assumptions.

4.4.0.7. We will discuss three classifiers that use different estimates of $f_k(x)$ to estimate the Bayes classifer (the theoretically optimal classifier from training data): linear discriminant analysis, quadratic discriminant analysis, and naive Bayes.

4.4.1 Linear Discriminant Analysis for $p = 1$

4.4.1.1. Assume $p = 1$. We assume that $f_k(x)$ is **normal** or **Gaussian**:

$$f_k(x) = \frac{1}{\sqrt{2\pi}\sigma_k} \exp\left(-\frac{1}{2\sigma_k^2}(x - \mu_k)^2\right)$$

where μ_k and σ_k^2 are the mean and variance of X for the kth class.

4.4.1.2. We will further assume that $\sigma_1^2 = \cdots = \sigma_K^2 = \sigma^2$.

4.4.1.3. We then plug this into Bayes' theorem to obtain:

$$p_k(x) = \frac{\pi_k \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2\sigma^2}(x - \mu_k)^2\right)}{\sum\limits_{i=1}^{K} \pi_i \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2\sigma^2}(x - \mu_i)^2\right)}$$

4.4.1.4. Under the above assumptions, the Bayes' classifer by computing for which value of k the above posterior is largest.

4.4.1.5. We can find $\max\limits_{k} p_k(x)$ by finding the maximum of the log of the numerator and only keeping track of terms that involve k. Working out the basic calculation gives the observation that the class which maximimizes

$$\delta_k(x) = x \cdot \frac{\mu_k}{\sigma^2} - \frac{\mu_k^2}{2\sigma^2} + \log(\pi_k)$$

will maximize $p_k(x)$.

4.4.1.6. Suppose $K = 2$ and $\pi_1 = \pi_2$. Then the Bayes classifer assigns an observation x to class 1 if

$$\begin{aligned} \delta_1(x) - \delta_2(x) &> 0 \\ x \cdot \frac{\mu_1}{\sigma^2} - \frac{\mu_1^2}{2\sigma^2} - x \cdot \frac{\mu_2}{\sigma^2} + \frac{\mu_2^2}{2\sigma^2} &> 0 \\ 2x(\mu_1 - \mu_2) &> \mu_1^2 - \mu_2^2 \end{aligned}$$

4.4.1.7. The Bayes decision boundary is where $\delta_1(x) = \delta_2(x)$, which is equivalent to:

$$x = \frac{\mu_1 + \mu_2}{2}$$

4.4.1.8. In practice, we have to estimate parameters $\mu_1, \dots, \mu_K, \pi_1, \dots, \pi_K$, and σ^2 to implement our classifier. **Linear discriminant analysis** (LDA) uses the following estimates:

$$\begin{aligned} \hat{\mu}_k &= \frac{1}{n_k} \sum_{i:y_i=k} x_i \\ \hat{\pi}_k &= \frac{n_k}{n} \\ \hat{\sigma}^2 &= \frac{1}{n-K} \sum_{k=1}^{K} \sum_{i:y_i=k} (x_i - \hat{\mu}_k)^2 \end{aligned}$$

where n is the total number of training observations and n_k is the number of observations in the kth class.

4.4.1.9. Note that $\hat{\sigma}^2$ is a weighted average of sample variances.

4.4.1.10. The term $1/(n - K)$ is a normalization factor that ensures we have an unbiased estimate of σ^2 (details?)

4.4.1.11. We then plug all of this into δ to obtain:

$$\hat{\delta}_k(x) = x \cdot \frac{\hat{\mu}_k}{\hat{\sigma}^2} - \frac{\hat{\mu}_k^2}{2\hat{\sigma}^2} + \log(\hat{\pi}_k)$$

This function is called the **discriminant function**. The "linear" in LDA comes from the fact that the discriminant function $\hat{\delta}(x)$ is linear.

4.4.2 Linear Discriminant Analysis for $p > 1$

4.4.2.1. Now let's extend LDA classifier to the case of multiple predictors.

4.4.2.2. We will assume that $X = (X_1, X_2, \dots, X_p)$ is drawn from a **multivariate Gaussian** distribution.

4.4.2.3. This distribution assumes that each individual predictor follows a 1-d normal distribution, with some correlation between each pair of predictors.

4.4.2.4. If $\mathrm{Var}(X_1) = \mathrm{Var}(X_2)$ and $\mathrm{Cor}(X_1, X_2) = 0$, then one obtains a perfectly circular bell shape. This bell shape is distored if the predictors are correlated or have unequal variances, in which case the bell will be elliptical.

4.4.2.5. If a p-dimensional random variabe X has a multivariate Gaussian distribution, then we write $X \sim N(\mu, \Sigma)$ where $\mu = \mathbb{E}[X]$ and $\Sigma = \mathrm{Cov}(X)$ is the $p \times p$ covariance matrix of X.

4.4.2.6. The multivariate Gaussian density is defined as:

$$f(x) = \frac{1}{(2\pi)^{p/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x - \mu)^T \Sigma^{-1} (x - \mu)\right)$$

4.4.2.7. The LDA classifier assumes that the observations in the kth class are drawn from a multivariate Gaussian distribution $N(\mu_k, \Sigma)$ where μ_k is a class-specific vector, and Σ is a covariance matrix common to all classes.

4.4.2.8. Plugging in μ_k and Σ into the density function, we obtain the density for the kth class $f_k(x)$. This can then be plugged into the Bayes theorem. After some computation, we learn that the Bayes classifier assigns an observation $X = x$ to the class for which:

$$\delta_k(x) = x^T \Sigma^{-1} \mu_k - \frac{1}{2} \mu_k^T \Sigma^{-1} \mu_k + \log \pi_k$$

Setting these equal to eachother divides the space from which X is drawn into K regions bounded by hyperplanes. The planes form the Bayes decision boundary.

4.4.2.9. From the data, we estimate the parameters $\mu_1, \ldots, \mu_K, \pi_1, \ldots, \pi_K$, and σ^2 much as before. LDA plugs these estimates to obtain discriminants $\hat{\delta}_k(x)$. As before, we obtain LDA decision boundaries.

4.4.2.10. Sometimes error rates can look low. For instance, applying the LDA model to the `Default` model, using balance and student status as predictors yields a training error rate of 2.75%.

- We should be skeptical that this really is a low rate since the test error rate – the rate we really care about – is typically higher than the training error rate.
- Also, in the training sample, only 3.33% defaulted. Thus, a null classifier that classifies everyone as not defaulting will have an error rate that is only a bit higher than the LDA training error rate.

4.4.2.11. There are two types of errors: false positives and false negatives. A **confusion matrix** is a conventient way to display this information. For our running problem, the confusion matrix is:

$$\begin{bmatrix} 9644 & 252 \\ 23 & 81 \end{bmatrix}$$

4.4.2.12. There are a total of 333 out of the sample of 10000 who defaulted. 252 of these were missed by the LDA. So the overall error rate, 2.75% may look low, the error rate among those who defaulted is $252/333 = 75.7\%$.

4.4.2.13. **Sensitivity** is the percentage of defaulters correctly identified by the model – 24.3%. The **specificity** is the percentage of non-defaulters that are correctly identified – $9644/9667 = 99.8\%$.

4.4.2.14. If we would like to increase the sensitivity of our model (perhaps being able to correctly identify defaulters is a priority) we might deviate from the Bayes classifer by changing the threshold posterior probability for assigning default from 50% to 20%. The new confusion matrix is:

$$\begin{bmatrix} 9432 & 138 \\ 235 & 195 \end{bmatrix}$$

Notice that the overall error rate has increased to $(235 + 138)/10000 = 3.73\%$. But the class specific error rate among defaulters is now $138/333 = 41.4\%$ which is a large improvement over the 75.7% error rate seen in the unmodified LDA classifier.

4.4.2.15. Here is a good summary of all the synonyms for things like false positives (type 1 errors) and false negatives (type 2 errors).

		True class		
		− or Null	+ or Non-null	Total
Predicted class	− or Null	True Neg. (TN)	False Neg. (FN)	N^*
	+ or Non-null	False Pos. (FP)	True Pos. (TP)	P^*
	Total	N	P	

TABLE 4.6. *Possible results when applying a classifier or diagnostic test to a population.*

Name	Definition	Synonyms
False Pos. rate	FP/N	Type I error, 1−Specificity
True Pos. rate	TP/P	1−Type II error, power, sensitivity, recall
Pos. Pred. value	TP/P^*	Precision, 1−false discovery proportion
Neg. Pred. value	TN/N^*	

TABLE 4.7. *Important measures for classification and diagnostic testing, derived from quantities in Table 4.6.*

4.4.3 Quadratic Discriminant Analysis

4.4.3.1. **Quadratic discriminant analysis** (QDA) again assumes that the observations are drawn from a Gaussian distribution, but it assumes that each class has its own covariance matrix. That is, for observations in the kth class, we have $X \sim N(\mu_k, \Sigma_k)$.

4.4.3.2. The Bayes classifer assigns an observation $X = x$ to the class for which:

$$\delta_k(x) = -\frac{1}{2}x^T\Sigma_k^{-1}x + x^T\Sigma_k^{-1}\mu_k - \frac{1}{2}\mu_k^T\Sigma_k^{-1}\mu_k - \frac{1}{2}\log|\Sigma_k| + \log\pi_k$$

is largest. Here, the discriminant is quadratic in x.

4.4.3.3. Why would one prefer LDA to QDA or vice versa? Note that if we have p predictors, then estimating a covariance matrix requires $p(p+1)/2$ parameters. With QDA, with a separate covariance matrix for every class, this comes to a total of $Kp(p+1)/2$ parameters. Thus, we see LDA is less flexible than QDA and so has lower variance. But LDA can be highly biased if the assumption of having a common covariance matrix is badly off.

4.4.3.4. LDA is preferred if there are relatively few training observations and so reducing variance is crucial. QDA is preferred if the training set is large so that the variance of the classifier is not a major concern. (refer to the section on bias-variance tradeoff for more information).

4.4.4 Naive Bayes

4.4.4.1. Instead of assuming that $f_k(x)$ belongs to any particular family of distributions, **naive Bayes** makes only the assumption that within the kth class, the p predictors are independent. This means:

$$f_k(x) = f_{k1}(x_1) \times f_{k2}(x_2) \times \cdots \times f_{kp}(x_p)$$

4.4.4.2. The reason this assumption is so powerful is that now we do not need to consider the **joint distribution** of the predictors – only the **marginal distribution** needs to be considered. Joint distributions are relatively easy to obtain for multivariate normal distributions – they are summarized by the off-diagonal elements of the covariance matrix. However, for general multivariate distributions, joint distributions can be hard to estimate.

4.4.4.3. The posterior probability in naive Bayes:

$$P(Y = k|X = x) = \frac{\pi_k \times f_{k1}(x_1) \times f_{k2}(x_2) \times \cdots \times f_{kp}(x_p)}{\sum_{i=1}^{K} \pi_i \times f_{i1}(x_1) \times f_{i2}(x_2) \times \cdots \times f_{ip}(x_p)}$$

4.4.4.4. To estimate the one-dimensional density functions f_{kj} using training data $x_{1j}, \dots, x_{nj}$, we have options:

- If the predictor is quantitative, we can make the assumption that within each class, the jth predictor is drawn from a univariate normal distribution. This is basically doing QDA with an additional assumption that the class-specific covariance matrix is diagonal.
- If the predictor is quantitative, we can use a non-parametric estimate of f_{kj}. We could do this by making a histogram which sorts the n_k training observations which belong to class k into bins, then model f_{kj} using the histogram. We can also use a **kernel density estimator** to get a smoothed version of the historgram.
- If X_j is qualitative – simply count the proportion of training observations for the jth predictor.

4.4.4.5. Naive Bayes does not necessarily outperform LDA, but is useful in situations in which p is large or n is small, as reducing variance is important.

4.5 A Comparison of Classification Methods

4.5.1 An Analytical Comparison

4.5.1.1. We will compare analytically LDA, QDA, naive Bayes, and logistic regression on the K class classification problem.

4.5.1.2. For each approach, we will assign an observation to the class that maximizes $P(Y = k|X = x)$. This is equivalent to setting a baseline class (we'll use K as the baseline), and assign an observation to the class that maximizes the log odds of an observation being in class k vs. class K.

$$\log\left(\frac{P(Y = k|X = x)}{P(Y = K|X = x)}\right)$$

4.5.1.3. For LDA, (within class densities of X normal, class specific means, shared covariance matrix), we have the following log odds:

$$\log\left(\frac{P(Y = k|X = x)}{P(Y = K|X = x)}\right) = a_k + \sum_{j=1}^{p} b_k j x_j$$

where

$$a_k = \log\left(\frac{\pi_k}{\pi_K}\right) - \frac{1}{2}(\mu_k + \mu_K)^T \Sigma^{-1}(\mu_k - \mu_K)$$
$$b_k = j\text{th component of } \Sigma^{-1}(\mu_k - \mu_K)$$

We see that LDA, like logistic regression, has log odds of the posterior probabilities which are linear in x.

4.5.1.4. For QDA (within class densities of X normal, class specific means, class specific covariance matrices), we have log odds:

$$\log\left(\frac{P(Y = k|X = x)}{P(Y = K|X = x)}\right) = a_k + \sum_{j=1}^{p} b_k j x_j + \sum_{j=1}^{p}\sum_{l=1}^{p} c_{kjl} x_j x_l$$

where a_k, b_{kj}, c_{kjl} are functions of $\pi_k, \pi_K, \mu_k, \mu_K, \Sigma_k$ and Σ_K. We see that in QDA, the log odds of the posterior probabilities are quadratic in x.

4.5.1.5. For naive Bayes where $f_k(x) = \prod_{j=1}^{K} f_{kj}(x_j)$

$$\log\left(\frac{P(Y = k|X = x)}{P(Y = K|X = x)}\right) = a_k + \sum_{j=1}^{p} \log\left(\frac{f_{kj}(x_j)}{f_{Kj}(x_j)}\right)$$

This is not a linear model. Denote by $g_{kj}(x_j) = \log\left(\frac{f_{kj}(x_j)}{f_{Kj}(x_j)}\right)$. But it does take the form of a **generalized additive model** which will be further discussed later.

4.5.1.6. Some observations:

- We know LDA is a special case of QDA where $\Sigma_i = \Sigma$ for all i. In fact, we see this in the posterior log odds – just make $c_kjl = 0$.
- Any classifier with a linear decision boundary is a special case of naive Bayes where $g_{kj}(x_j) = b_{kj}(x_j)$. I'm trying to find some justification for this statement, and I just can't. Neither the internet nor ChatGPT seem to strongly support this statment.
- In the case where where we model $f_{kj}(x_j)$ as 1-d Gaussians, then it happens that naive Bayes looks like LDA where Σ is diagonal.
- Neither QDA nor naive Bayes can be thought of as a special case of the other. Naive Bayes is a purely additive fit, whereas QDA involves multiplication of the x_is.

4.5.1.7. Logistic regression give log odds which are linear in the predictors. Of course, it makes no assumption about the distributions of the predictors which LDA does. As a result, we expect LDA to perform better when the normality assumption approximately holds, and we expect logistic regression to perform better when it does not.

4.5.1.8. Here are some observations about how KNN classification fits into the picture:

- KNN is non-parametric. We expect this to dominate the more structured models like LDA and logistic regression when the decision boundary is highly non-linear, provided that n is very large and p is small.
- In terms of the bias-variance tradeoff, KNN – being so flexible – is very unbiased and has high variance. We want n to be much larger than p to effectively use KNN.
- If the decision boundary is non-linear, and if n is modest, or p is large, QDA may be preferred since its parametric form may reduce variance.
- Unlike logistic regression, KNN cannot tell us which predictors are important.

4.5.2 An Empirical Comparison

4.5.2.1. The result of empirical investigations shows that when true decision boundaries are linear, then LDA and logistic regression perform well. When boundaries are moderately non-linear, QDA or naive Bayes may give better results. For complicated decision boundaries, KNN can be superior.

4.5.2.2. We can achieve results somewhere between LDA and QDA by including higher powers of the predictors (X^2, X^3, etc) in logistic regression.

4.6 Generalized Linear Models

4.6.0.1. What if Y is neither quantitative nor qualitative? For example, if Y takes on non-negative integer values, we call these values **counts** and consider them to be neither quantitative nor qualitative.

4.6.0.2. We wish to model `bikers`, the number of hourly users of a bike sharing program in Washington DC. We are going to use the following covariates:

- `mnth` – month of the year
- `hr` – hour of the day, from 0 to 23
- `workingday` – 1 if it is neither a weekend or holiday
- `temp` – normalized temperature in Celsius
- `weathersit` – clear, misty/cloudy, light rain/snow, heavy rain/snow

4.6.0.3. All of these predictors except `temp` will be treated as qualitative variables.

4.6.1 Linear Regression on the Bikeshare Data

4.6.1.1. The problem with applying linear regression is that we sometimes get nonsenisical predictions such as negative values for `bikers`.

4.6.1.2. Another problem is heteroscedasticity – variance is a non-constant function of the covariates. However, linear regression assumes that the variance is constant.

4.6.1.3. And of course, `bikers` is integer-valued, but linear regression models Y has a continuous variable.

4.6.2 Poisson Regression on the Bikeshare Data

4.6.2.1. The **Poisson distribution** for a random variable which takes non-negative integer values is:

$$P(Y = k) = \frac{e^{-\lambda}\lambda^k}{k!}, \qquad \text{for } k = 0, 1, 2, \ldots$$

4.6.2.2. $\mathbb{E}[Y] = \lambda$

4.6.2.3. $\text{Var}[Y] = \lambda$

4.6.2.4. Let Y denote the number of bike sharing program during a particular hour of the the day, under a particular set of weather conditions, and during a particular month of the year. We could model Y as a Poisson distribution with some mean λ.

4.6.2.5. In reality, we expect the mean number of users of the bike sharing program to vary as a function of the hour of the day, month of the year, weather conditions, and so forth.

4.6.2.6. So we model the mean as a function of the predictors: $\lambda(X_1, \ldots, X_p)$. In particular, in Poisson regression, we model the log of λ as linear in the predictors:

$$\log(\lambda(X_1, \ldots, X_p)) = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p$$

4.6.2.7. To estimate the coefficients, we use maximum likelihood estimation. Given n independent observations from the Poisson regression model, the likelihood is:

$$l(\beta_0, \beta_1, \ldots, \beta_p) = \prod_{i=1}^{n} \frac{e^{-\lambda(x_i)}\lambda(x_i)^{y_i}}{y_i!}$$

where $\lambda(x_i) = \exp(\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip})$.

4.6.2.8. Interpretation of the Poisson coefficients is that a one unit increase in X_j changes λ by a factor of $\exp(\beta_j)$.

4.6.2.9. Because in Poisson, the mean is equal to the variance, we do a better job of modeling heteroscedasticity. In linear regression, variance is assumed to be constant. For our Bikeshare model, we in fact have **overdispersion**, a situation in which the variance is much higher than is expected for a chosen statistical model.

4.6.3 Generalized Linear Models in Greater Generality

4.6.3.1. We have discussed three types of regression models: linear, logistic, and Poisson. All have the following common characteristics:

- We assume that, conditional on the predictors, Y belongs to a certain family of distributions. For linear: normal. For logistic: Bernoulli. For Poisson: Poisson.
- Each approach models the mean of Y as a function of the predictors:

$$\begin{aligned}
\text{linear:} \quad & \mathbb{E}(Y|X_1, \ldots, X_p) = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p \\
\text{logistic:} \quad & \mathbb{E}(Y|X_1, \ldots, X_p) = P(Y = 1|X_1, \ldots, X_p) \\
& \qquad = \frac{\exp(\beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p)}{1 + \exp(\beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p)} \\
\text{Poisson:} \quad & \mathbb{E}(Y|X_1, \ldots, X_p) = \lambda(X_1, \ldots, X_p) = \exp(\beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p)
\end{aligned}$$

- Each of these examples can be expressed using a **link function** η which when applied to the mean gives a linear function in the predictors:

$$\eta(\mathbb{E}(Y|X_1, \ldots, X_p)) = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p$$

  For linear: $\eta(\mu) = \mu$. For logistic: $\eta(\mu) = \log\left(\frac{\mu}{1-\mu}\right)$. For Poisson: $\eta(\mu) = \log(\mu)$.

4.6.3.2. The Gaussian, Bernoulli, and Poisson distributions are all members of a wider class of distributions known as the **exponential family**. We can perform a regression by modeling the response Y as coming from a distribution in the exponential family, and then transforming the mean of the response so that the transformed mean is a linear function of the predictors. Any regression approach that follows these steps is known as a **generalized linear model** (GLM).

5 Resampling methods

5.0.0.1. **Resampling methods** involve repeatedly drawing samples from a training set and refitting a model of interest on each sample in order to obtain additional information about the fitted model.

5.0.0.2. For example, we can repeatedly draw different samples from training data, fit a linear regression to each sample, and then examine to what extent the resulting fits differ.

5.0.0.3. Resampling can be computationally expensive, but with recent advances in computing power, these costs are not prohibitive.

5.0.0.4. **Cross-validation** is a resampling mthod which can be used to estimate test error associated with a learning method in order to evaluate its performance – aka **model assessment**.

5.0.0.5. **Bootstrap** measures accuracy of a parameter estimate of a given statistical learning method.

5.1 Cross-Validation

5.1.0.1. Recall the distinction between the test error rate and the training error rate. The test error rate is the average error rate that results when a fitted model is used on a new observation. We judge the quality of a model by its test error rate. We can obtain an estimate of the test error rate by using a designated test set of observations.

5.1.0.2. The training error rate is easy to compute. However, it is generally a poor estimate of the test error rate – it can dramatically underestimate the test error rate.

5.1.0.3. There are techniques to estimate the test error rate using training data. In this section, we consider a class of methods that estimate the test error rate by holding out a subset of the training observations from the fitting process, and then applying the statistical learning method to those held out observations.

5.1.1 The Validation Set Approach

5.1.1.1. The **validation set approach** randomly divides the available set of observations into a training set and a validation set or hold-out set. The model is fit on the training set, and the fitted model is used to predict the responses on the validation set. The resulting validation set error rate, typically assessed using MSE in the case of a quantiative response, provides an estimate of the test error rate.

5.1.1.2. We can apply the validation set approach to the problem of predicting `mpg` using polynomial functions of `horsepower`. And we can do this repeatedly by divvying the available data into training and validation sets in different ways. We obtain:

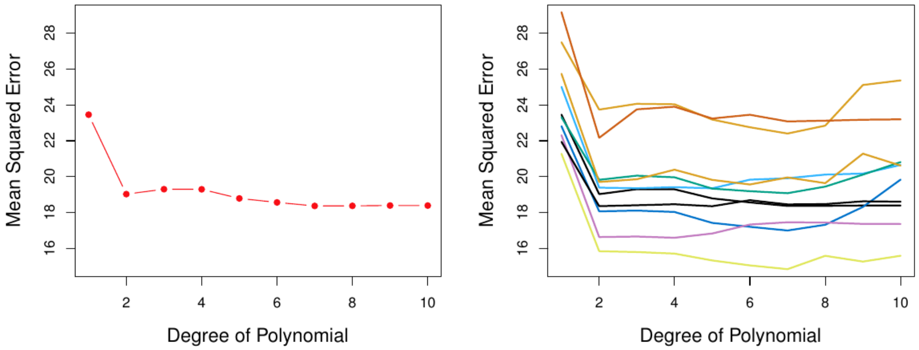


FIGURE 5.2. *The validation set approach was used on the* `Auto` *data set in order to estimate the test error that results from predicting* `mpg` *using polynomial functions of* `horsepower`*.* Left: *Validation error estimates for a single split into training and validation data sets.* Right: *The validation method was repeated ten times, each time using a different random split of the observations into a training set and a validation set. This illustrates the variability in the estimated test MSE that results from this approach.*

5.1.1.3. We can see that the validation estimate of the test error rate can be highly variable depending on the validation set.

5.1.1.4. We can surmise that since statistical methods tend to perform worse with fewer observations, and since we use up a subset of the data for the validation set, the validation set error rate may tend to overestimate the test error rate.

5.1.2 Leave-One-Out Cross-Validation

5.1.2.1. LOOCV uses the validation set approach, but uses a single observation for the validation set. We measure the test error rate by computing the MSE of the prediction of our single observation. We then repeat n times, excluding each observation (x_i, y_i), and measuring $\text{MSE}_i = (y_i - \hat{y}_i)^2$.

5.1.2.2. THe LOOCV estimate for the test MSE is the average:

$$\text{CV}_{(n)} = \frac{1}{n}\sum_{i=1}^{n}\text{MSE}_i$$

5.1.2.3. Because we are fitting the model with $n-1$ observations – nearly the entire data set – we do not expect to overestimate the test error rate as in the usual validation set method.

5.1.2.4. LOOCV is not variable. The estimate produced will be the same no matter what since there is no randomness in the training/validation split.

5.1.2.5. LOOCV appears to be expensive to implement. However, with least squares linear or polynomial regression, an amazing shortcut makes the cost of LOOCV the same as a single model fit:

$$\text{CV}_{(n)} = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_i - \hat{y}_i}{1-h_i}\right)^2$$

where h_i is the observation's leverage statistic. Recall that an observation for which the predictor is outside the normal range of observations is referred to as having high leverage. The leverage statistic is:

$$h_i = \frac{1}{n} + \frac{(x_i - \overline{x})^2}{\sum_{j=1}^{n}(x_j - \overline{x})^2}$$

I need to learn more linear regression – particularly the matrix presentation – before I can really understand

5.1.3 k-Fold Cross-Validation

5.1.3.1. **k-fold CV** randomly divides the set of observations into k groups, or **folds**, of approximately equal sold. Initially, the first fold is a validation set, and the method is fit on the remaining data. The MSE is computed. The process is repeated k times using each fold as the validation set.

5.1.3.2. The k-fold CV estimate is computed by averaging the MSE so generated

$$\text{CV}_{(k)} = \frac{1}{k}\sum_{i=1}^{k}\text{MSE}_i$$

5.1.3.3. LOOCV is a special case of k-fold CV where $k = n$.

5.1.3.4. The obvious advantage of doing k-fold CV over LOOCV is computational cost.

5.1.3.5. Here is the result of k-fold CV on simulated data. Smoothing splines were applied to the simulated data at various levels of flexibility:

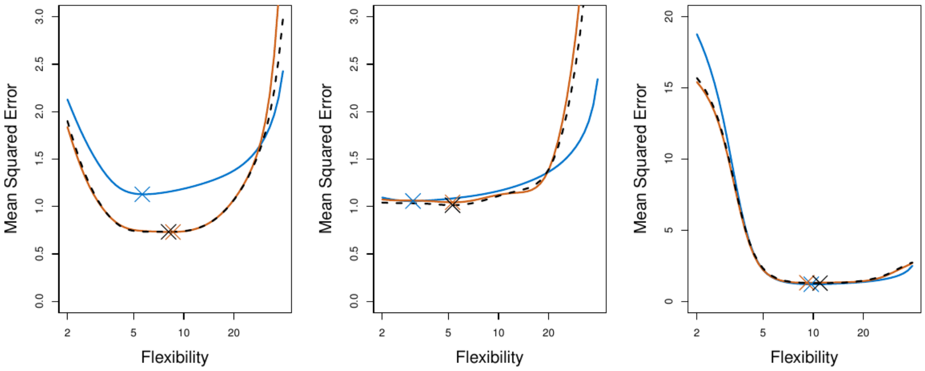


FIGURE 5.6. *True and estimated test MSE for the simulated data sets in Figures 2.9 (*left*), 2.10 (*center*), and 2.11 (*right*). The true test MSE is shown in blue, the LOOCV estimate is shown as a black dashed line, and the 10-fold CV estimate is shown in orange. The crosses indicate the minimum of each of the MSE curves.*

5.1.3.6. We observe that LOOCV estimates and 10-fold CV perform similarly.

5.1.3.7. Though the estimates may not always get the true test MSE with high accuracy, we see that the minimum MSE can be reasonably inferred from CV. This is an important tool for model selection.

5.1.4 Bias-Variance Trade-Off for k-Fold Cross Validation

5.1.4.1. Comparing k-fold CV to LOOCV, we should expect that since the fit of the model uses a greater proportion of the data available, LOOCV will produce less biased estimates of the test error rate.

5.1.4.2. However, it turns out that LOOCV has higher variance than k-fold CV. This comes from the fact that the mean of many highly correlated (highly correlated means positively correlated) quantities has higher variance than does the mean of many quantities that are not highly correlated. One way to understand this is the formula:

$$\mathrm{Var}\left(\sum_{i=1}^{n} X_i\right) = \sum_{i=1}^{n} \mathrm{Var}(X_i) + 2\sum_{i<j} \mathrm{Cov}(X_i, X_j)$$

If X and Y are positively correlated, then the covariance sum positively contributes to the variance, whereas, in the extreme where the variables are independent, the covariance terms are all 0.

5.1.4.3. So we see there is a bias-variance tradeoff associated with the choice of k in k-fold cross-validation. When k is large, we expect less bias and greater variance. And when k is small, we expect more bias and less variance.

5.1.5 Cross-Validation on Classification Problems

5.1.5.1. Cross-validation works on classification problems in much the same was as with regression.

5.1.5.2. Rather than use MSE to quantify test error, we use the number of misclassified observations.

5.1.5.3. So the LOOCV error rate is:

$$\mathrm{CV}_{(n)} = \frac{1}{n}\sum_{i=1}^{n} \mathrm{Err}_i$$

5.2 The Bootstrap

5.2.0.1. The **bootstrap** is a widely applicable and powerful statistical tool to quantify the uncertainty associated with an estimator or statistical learning method.

5.2.0.2. It can be applied to a wide range of statistical learning methods, including some for which getting a measure of variability is difficult (even by statistical software).

5.2.0.3. Consider the following problem:

- Suppose we wish to invest a fixed sum of money in two financial assets that yeild returns of X and Y. We will invest a fraction α of our money in X, and the remainder $1-\alpha$ in Y. We wish to choose α to minimize the total risk, or variance, of our investment. So we want to minimize $\mathrm{Var}(\alpha X + (1-\alpha)Y)$.
- One can show that the value which minimizes risk is:

$$\alpha = \frac{\sigma_Y^2 - \sigma_{XY}}{\sigma_X^2 + \sigma_Y^2 - 2\sigma_{XY}}$$

  This follows fairly easily from $\mathrm{Var}(X+Y) = \mathrm{Var}(X) + \mathrm{Var}(Y) + 2\mathrm{Cov}(X,Y)$, $\mathrm{Var}(\alpha X) = \alpha^2$, and $\mathrm{Cov}(\alpha X, Y) = \alpha \mathrm{Cov}(X,Y)$.
- Using estimates for these quantities $\hat{\sigma}_X^2, \hat{\sigma}_Y^2, \hat{\sigma}_{XY}$ coming from data, we can obtain an estimate:

$$\hat{\alpha} = \frac{\hat{\sigma}_Y^2 - \hat{\sigma}_{XY}}{\hat{\sigma}_X^2 + \hat{\sigma}_Y^2 - 2\hat{\sigma}_{XY}}$$

- Simulating 100 pairs of returns repeatedly, we can creates estimates and obtain values for $\hat{\alpha}$. By doing this enough times, we can approximate the distribution of $\hat{\alpha}$, including the mean $\overline{\alpha}$ and the standard deviation.
- However, in most problems, simulations such as these are impossible since we do not know the true values of the quantities. We also cannot repeatedly sample from the popultion.
- The bootstrap is to repeatedly sample with replacement from the original data set. Repeating this B times, we obtain B bootstrapped samples $Z^{*1}, Z^{*2}, \ldots, Z^{*B}$.
- Each of these bootstrapped samples can be used to create B estimates of α $\hat{\alpha}^{*1}, \ldots, \hat{\alpha}^{*B}$.
- We can then compute the standard error of these bootstrap estimates using the formula:

$$\mathrm{SE}_B(\hat{\alpha}) = \sqrt{\frac{1}{B-1} \sum_{r=1}^{B} \left(\hat{\alpha}^{*r} - \frac{1}{B} \sum_{s=1}^{r} \hat{\alpha}^{*s} \right)}^{2}$$

- In this example, bootstrapping gives a distribution of $\hat{\alpha}$ very close to the "true" distribution coming from the simulations.

5.2.0.4. The bootstrap can estimate other quantities. In the lab section of this chapter, there is the implmentation of the bootstrap to estimate the standard error of linear regression coefficients. In section 3.1.2, given are formulas for the standard errors of linear regression coefficient estimates. However, these formulas assume that errors for each observation have common variance σ^2 and are uncorrelated. If we perform linear regression in on data generated by a non-linear system, then we should not expect these formulas to yield accurate estimates. The bootstrap does not rely on these assumptions, and so give a better estimate of standard error.

6 Linear Model Selection and Regularization

6.0.0.1. The standard linear model

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

is typically fit using least squares. In this chapter, we will discuss ways in which the linear can be improved with some alternative fitting procedures.

6.0.0.2. Why would we want to do this?

- **Prediction Accuracy:** If the true relationship is approximately linear, then least squares will have low bias. If $n \gg p$, then the least squares will have low variance. However, if n is not so much larger, there will be lots of variance. And if $p > n$, there is no longer a unique least squares estimate – the variance is infinite so the method cannot be used at all. By **constraining** or **shrinking** the estimated coefficients, we can often reduce the varaince at the cost of a small increase in bias.

- **Model Interpretability:** Some predictors are not associated with the response. Keeping these irrelevant variables around leads to unnecessary complexity. We will see approaches for performing **feature selection** or **variable selection**.

6.0.0.3. Three important classes of methods which are alternatives to least squares:

- **Subset selection** Identify a subset of the predictors and fit on these only.
- **Shrinkage** Shrink estimated coefficients toward zero. This has the effect of reducing variance. Some methods may shrink coefficients to zero, which has the effect of variable selection.
- **Dimension Reduction:** Project the p predictors onto an M-dimensional subspace where $M < p$, and fit using M projections.

6.1 Subset Selection

6.1.1 Best Subset Selection

6.1.1.1. There are 2^p subsets of the p predictors. We can fit all 2^p models and select the one that is best.

6.1.1.2. Break up the task into stages:

- Let $\mathcal{M}_0$ be the **null model**, which contains no predictors. This model predicts the sample mean for each observation.
- For $k = 1, \dots, p$, fit all $\binom{n}{k}$ models with k predictors. Let $\mathcal{M}_k$ be the one with lowest RSS, or, equivalently, largest R^2.
- Select the best model among $\mathcal{M}_0, \dots \mathcal{M}_p$ using cross-validated prediction error, C_p(AIC), BIC, or adjusted R^2.

6.1.1.3. The reason our metric changes from RSS to something different is because naturally, the more complicated model will have a lower RSS. However, the more complicated model will not necessarily perform well on a test set.

6.1.1.4. RSS is not appropriate for all models. In the case of logistic regression, instead of ordering by RSS, we use **deviance**. Deviance is a measure of goodness of fit for generalized linear models. It's not really explained in the text, but for a model $\mathcal{M}$, the deviance of $\mathcal{M}$ is:

$$D_{\mathcal{M}} = -2 \log \left(\frac{L_{\mathcal{M}}}{L_{\mathcal{S}}} \right) = -2(\log L_{\mathcal{M}} - \log L_{\mathcal{S}})$$

where $\mathcal{S}$ denotes the saturated model, and L_{model} is the likelihood of the model. (Note that the log likelihood of the saturated model is not necessarily 0).

But we can think of the deviance of $\mathcal{M}$ as being -2 times the log likelihood of $\mathcal{M}$ plus a constant (which is the log likelihood of the saturated model). This generalizes RSS for linear regression. Try examining:

`https://statisticaloddsandends.wordpress.com/2019/03/27/what-is-deviance/`

If we are comparing two models $\mathcal{M}_0$ and $\mathcal{M}_1$ with $\mathcal{M}_0 \subset \mathcal{M}_1$ (meaning that every predictor in $\mathcal{M}_0$ is a predictor in $\mathcal{M}_1$), then the test statistic is

$$\begin{aligned} D &= D_{\mathcal{M}_0} - D_{\mathcal{M}_1} \\ &= -2(\log L_{\mathcal{M}_0} - \log L_{\mathcal{M}_1}) \end{aligned}$$

If $\mathcal{M}_0$ and $\mathcal{M}_1$ have p_0 and p_1 parameters, then by Wilk's theorem, under the null hypothesis that $\mathcal{M}_1$ is not a better fit, as the sample size goes to ∞, then $D \sim \chi^2_{p_1 - p_0}$.

6.1.1.5. Of course, best subset selection suffers from high computational cost.

6.1.1.6. Also, the larger p is, the higher chance of running into overfit models.

6.1.2 Stepwise Selection

6.1.2.1. **Forward stepwise selection** begins with the null model, then adds predictors one-at-a-time, until all of the predictors are in the model. At each stpe, the predictor that gives the greatest additional improvement is added.

6.1.2.2. The number of models fit: 1 for the null model, and $p - k$ in the kth step. Thus, the total number is:

$$\sum_{k=0}^{p-1}(p-k) = 1 + \frac{p(p+1)}{2}$$

6.1.2.3. When $p = 20$, best subset requires 1,048,576 fits. Forward stepwise selection requires only 211.

6.1.2.4. Here's the algorithm for forward stepwise selection:

- Find $\mathcal{M}_0$.
- For $k = 0, \ldots p-1$, fit the $p-k$ models that augment the predictors in $\mathcal{M}_k$ with one additional predictor. Let $\mathcal{M}_{k+1}$ be the model with smallest RSS.
- Select the best model from $\mathcal{M}_0, \ldots, \mathcal{M}_p$ using cross-validated prediction error, C_p (AIC), BIC, or adjusted R^2.

6.1.2.5. Forward stepwise selection is not guaranteed to find the best model out of the 2^p possible models.

6.1.2.6. This method can be used in high-dimensional settings where $n < p$. However, since each model is fit using least squares, there will not be a unique model obtained when $p \geq n$.

6.1.2.7. **Backward stepwise selection**

- Fit the **full model** $\mathcal{M}_p$ – the model with all p predictors.
- For $k = p, \ldots, 1$, consider all k models that contain all but one of the predictors in $\mathcal{M}_k$. Let $\mathcal{M}_{k-1}$ be the model with smallest RSS.
- Select the best model from $\mathcal{M}_0, \ldots, \mathcal{M}_p$ using cross-validated prediction error, C_p (AIC), BIC, or adjusted R^2.

6.1.2.8. Backward selection requires $n \geq p$ so that the full model can be fit.

6.1.2.9. Hybrid methods combining best subset, forward stepwise and backward stepwise are also used.

6.1.3 Choosing the Optimal Model

6.1.3.1. How to determine which model among a collection of models is best? Here best means low test error. We know that training error can be a poor estimate of test error, so RSS and R^2 are not suitable metrics for determining the best model.

6.1.3.2. There are two common approaches to estimate test error:

- Indirectly estimate test error by making an adjustment to the training error to account for bias due to overfitting.
- Directly estimate test error using a validation set approach or a cross-validation approach (discussed in Chapter 5).

6.1.3.3. We learned in Chapter 2 that the training MSE = RSE/n is generally an underestimate of the test MSE. This is because when fitting a model to the training data using least squares, we seek to minimize training RSS. We know also that the training error always descreases as more variables are added to the model, but test error could go down or up.

6.1.3.4. Four approaches for adjusting the training error to obtain better estimates of test error:

- C_p: for a fitted least squares model containing d predictors, the C_p estimate of test MSE is

  $$C_p = \frac{1}{n}(\mathrm{RSS} + 2d\hat{\sigma}^2)$$

  where $\hat{\sigma}^2$ is an estimate of the variance of the error ϵ. One can show that C_p is an unbiased estimate of test MSE. So when determining which set of models is best, we choose a model with lowest C_p.
- **AIC – Akaike information criterion:** This is defined for a large class of models fit by maximum likelihood. For linear models with Gaussian errors, maximum likelihood and least squares are the same thing. So, in this case:

  $$\mathrm{AIC} = \frac{1}{n}(\mathrm{RSS} + 2d\hat{\sigma}^2)$$

  where some constants have been removed. So, in this case, AIC and C_p are proportional to one another.
- **BIC – Bayesian information criterion:** is similar to AIC, except the penalty for an additional predictor is $\log(n)\hat{\sigma}^2$. For least squares, up to irrelevant constants:

  $$\mathrm{BIC} = \frac{1}{n}(\mathrm{RSS} + \log(n)d\hat{\sigma}^2)$$

  BIC generally places a heavier penalty on models with many variables, compared to AIC.
- **Adjusted R^2:** Recall that $R^2 = 1 - \mathrm{RSS}/\mathrm{TSS}$. It is the propotion of variance explained by the model. However, this makes no penalty for using more variables. For a least squares model with d variables:

  $$\text{Adjusted } R^2 = 1 - \frac{\mathrm{RSS}/(n-d-1)}{\mathrm{TSS}/(n-1)}$$

  If a variable is added to a model which does not decrease RSS much, then the adjusted R^2 will tend to get smaller, whereas R^2 will increase. Notice that maximizing adjusted R^2 is the same as minimizing $\mathrm{RSS}/(n-d-1)$.

6.1.3.5. With the advent of abundant compute, directly estimating test error using cross-validation are hardly ever an issue, and so are attractive for model selection.

6.1.3.6. When validation and cross-validation shows test error rates for models to be very similar, then we use the **one-standard-error rule**. We calculated the stadnard error the estimated test MSE for each model size, and then select the smallest model for which the estimated test error is within one stadnard error of the lowest point on the curve. This fixes issues of variance in the estimate of test error.

6.2 Shrinkage Methods

6.2.0.1. As an alternative to least squares, we can fit a model containing all p predictors using a technique that **constrains** or **regularizes** the ceofficient estimates. This is equivalent to **shrinking** coefficient estimates toward zero.

6.2.0.2. Shrinking can significantly reduce variance.

6.2.1 Ridge Regression

6.2.1.1. Lease squares finds $\beta_0, \ldots, \beta_p$ which minimizes:

$$\mathrm{RSS} = \sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j X_j\right)^2$$

6.2.1.2. **Ridge regression** finds $\beta_0, \ldots, \beta_p$ which minimizes:

$$\mathrm{RSS} + \lambda\sum_{j=1}^{p}\beta_j^2$$

where $\lambda \geq 0$ is a **tuning parameter**. The second term $\lambda\sum_{j=1}^{p}\beta_j^2$ is called a **shrinkage penalty**.

6.2.1.3. The shrinking penalty will be small when the β's are close to 0.

6.2.1.4. As $\lambda \to \infty$, β's go to 0.

6.2.1.5. Note the shrinking penalty does not apply to the intercept β_0.

6.2.1.6. It can happen that as λ increases, a coefficient estimate first increases, then decreases.

6.2.1.7. Recall that least squares coefficient estimates are **scale equivariant**: multiplying X_j by a constant c will scale the correpsonding least square coefficient estimate by $1/c$. In contrast, ridge regression coefficent estimates can change substantially. To avoid this scaling issue, it is best to apply ridge regression after **standardizing the predictors** using the formula:

$$\tilde{x}_{ij} = \frac{x_{ij}}{\sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_{ij} - \overline{x}_j)^2}}$$

The final fit will not depend on the scale on which the predictors are measured.

6.2.1.8. As λ increases, the flexibility of ridge regression decreases leading to decreased variance, but increased bias.

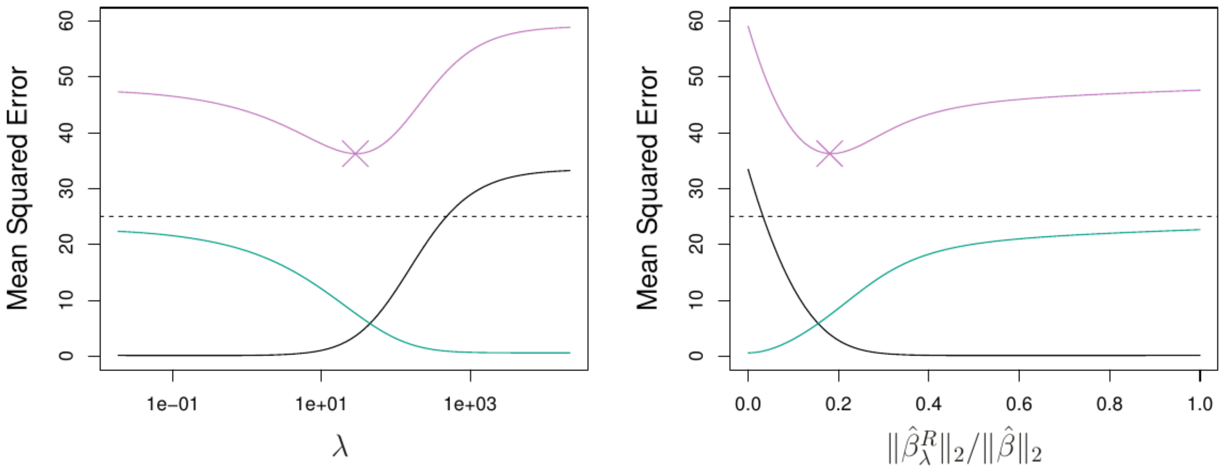


FIGURE 6.5. *Squared bias (black), variance (green), and test mean squared error (purple) for the ridge regression predictions on a simulated data set, as a function of* λ *and* $\|\hat{\beta}^R_\lambda\|_2/\|\hat{\beta}\|_2$*. The horizontal dashed lines indicate the minimum possible MSE. The purple crosses indicate the ridge regression models for which the MSE is smallest.*

6.2.1.9. In situations where the true data generating process is close to linear, then least squares will have low bias and high variance. As the number of variables grows, the variance also grows. Hence, ridge regression can perform well by exchanging a small increase in bias for a large decrease in variance.

6.2.1.10. Ridge regression is computationally inexpensive. In fact, the computations to solve the ridge regression minimzation problem can be done simultaneously for all values of λ with computations almost identical to fitting a model using least squares.

6.2.2 The Lasso

6.2.2.1. Ridge regression uses all p predictors. This makes model interpretation a challenge.

6.2.2.2. The **lasso** overcomes this. The lasso coefficients $\hat{\beta}^L_\lambda$ minimize

$$\text{RSS} + \lambda \sum_{j=1}^{p} |\beta_j|$$

6.2.2.3. Lasso uses an ℓ_1 norm, while ridge regression uses an ℓ_2 norm on the coefficient vector β (excluding β_0.)

6.2.2.4. The ℓ_1 penalty has the effect of forcing some of the coefficient estimates to be exactly equal to zero when λ is sufficiently large. Thus, the lasso performs variable selection.

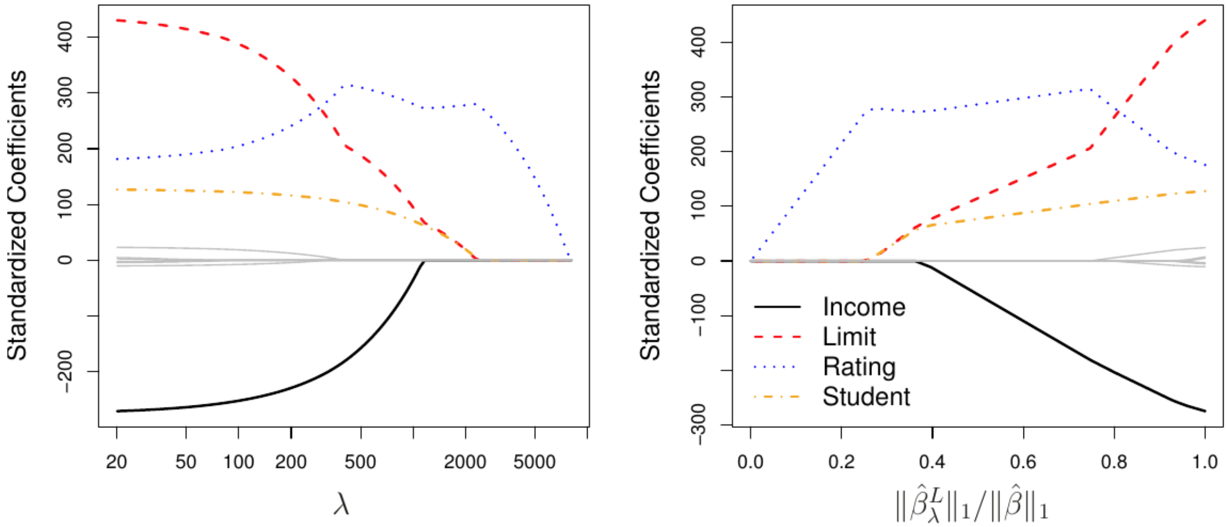


FIGURE 6.6. *The standardized lasso coefficients on the* `Credit` *data set are shown as a function of* λ *and* $\|\hat\beta^L_\lambda\|_1/\|\hat\beta\|_1$.

6.2.2.5. One can show that the lasso and ridge regression solve the following minimization problems:

$$\min_\beta \left\{ \sum_{i=1}^n \left(y_i - \beta_0 - \sum_{j=1}^p \beta_j x_{ij} \right)^2 \right\} \quad \text{subject to} \quad \sum_{j=1}^p |\beta_j| \le s$$

$$\min_\beta \left\{ \sum_{i=1}^n \left(y_i - \beta_0 - \sum_{j=1}^p \beta_j x_{ij} \right)^2 \right\} \quad \text{subject to} \quad \sum_{j=1}^p \beta_j^2 \le s$$

That is, there exists an s such that solving the above gives the lasso or ridge regression coefficients for some given value of λ. These are convex optimization problems. Convex optimization is an huge enterprise, but I imagine the mathematical details here are not so bad.

6.2.2.6. Let's try to understand how we get from the constrained problem to the unconstrained problem.

Recall the idea behind optimization via Lagrange multipliers. Our goal is to find a point x^* at which $f(x)$ is minimized, subject to a constraint $g(x) = 0$. Such a point must have the property that the gradient of f is orthogonal to the codimension-1 level set $g(x) = 0$, otherwise it would be possible to nudge the point a little bit along the constraint to obtain a smaller value. Since ∇g is orthogonal to the constraint level set, we must have that ∇f is a scalar multiple of ∇g. We build the Lagrangian function:

$$L(x, \lambda) = f(x) + \lambda g(x)$$

To find possible extrema, we set ∇L equal to the zero vector and solve.

Now suppose that the constraint is an inequuality $g(x) \le 0$. This defines a constraint region in which x^* must lie. Then either x^* lies on the interior $g(x) < 0$ (the constraint is said to be inactive) or on the boundary $g(x) = 0$ (the constraint is said to be active). The reason for this terminology is that if x^* is in the interior, then $g(x)$ plays no role in minimizing $f(x)$ – we can treat the problem as an unconstrained problem on $f(x)$.

If x^* lies on the boundary $g(x) = 0$, then notice that ∇g must point out toward the exterior. However, ∇f must point into the interior – otherwise we could get a smaller value for f by moving x into the interior. Therefore, we require

$$\nabla f = -\lambda \nabla g$$

where $\lambda > 0$.

Notice that we can capture both of these cases (x^* in the interior or on the boundary) with the condition $\lambda g(x) = 0$. This implies that either $\lambda = 0$ (in which case we are performing an unconstrained optimization of $f(x)$) or $g(x) = 0$ (in which case, we are doing a standard Lagrange optimization.)

So, we can reformulate our minimization with inequality constraint problem as optimizing the Lagrange function

$$L(x, \lambda) = f(x) + \lambda g(x)$$

subject to the conditions:

$$\begin{aligned} g(x) &\leq 0 \\ \lambda &\geq 0 \\ \lambda g(x) &= 0 \end{aligned}$$

These are known as Karush-Kuhn-Tucker (KKT) conditions.

From here, it is straightforward to see how we get from the constrained version of ridge and lasso, to the unconstrained version.

Most of this material came from Appendix E of *Pattern Recognition and Machine Learning* by CM Bishop.

6.2.2.7. Best subset selection can be fit into this optimization with constraint paradigm:

$$\min_\beta \left\{ \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 \right\} \quad \text{subject to} \quad \sum_{j=1}^{p} I(\beta_j \neq 0) \leq s$$

6.2.2.8. Increasing λ causes the coefficients in each case to get smaller. Increasing s causes the coefficients in each case to get larger.

6.2.2.9. Note that the constraints in each case correspond to β lying in a polytope in the lasso case, and in a spherical ball in the ridge case. Note that the vertices of the lasso polytope lie precisely on the axes where one of the β's is equal to 0.

6.2.2.10. The contours of the RSS term common to both problems look like ellipses with $\hat{\beta}$, the least squares estimate at the center. If $\hat{\beta}$ lies in the constraint region, then the lasso or ridge regression coefficient estimates are the same as the least squares estimate.

6.2.2.11. The reason that lasso tends to result in coefficient estimates exactly equal to 0 is that the level curves of RSS radiate away from $\hat{\beta}$ in ellipses, with larger ellipses corresponding to larger values of RSS. The optimum value for β will occur at the smallest ellipse which touches the constraint region. In lasso, the constraint region has corners at the axes where one of the βs is 0. It is likely tht the level curves of RSS will first touch at the corners thus giving rise to the variable selection property of the lasso.

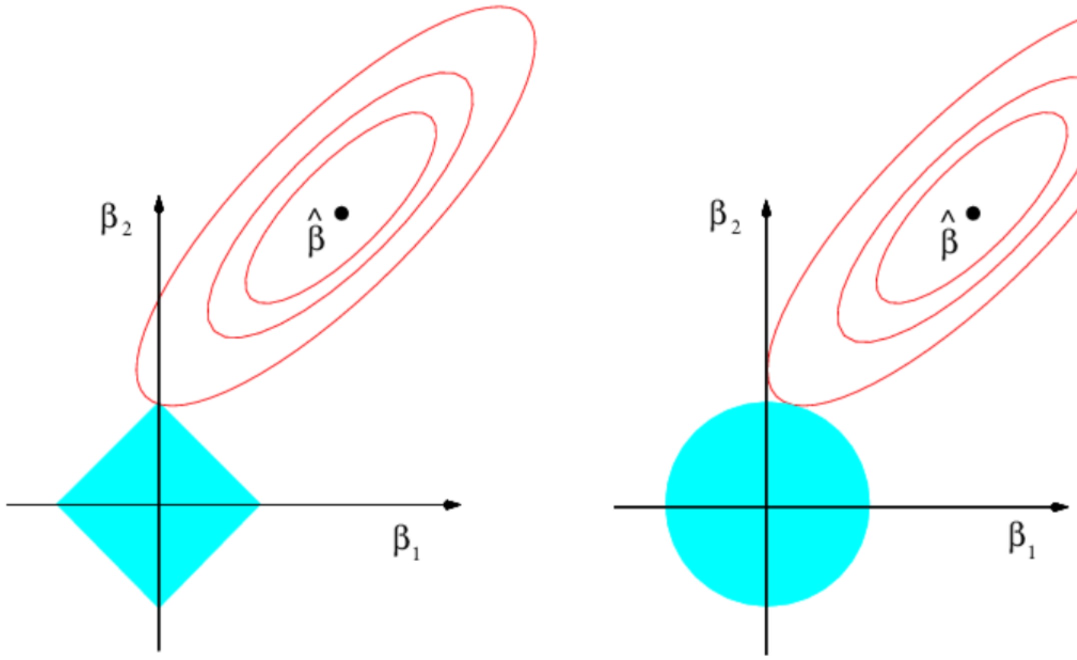


FIGURE 6.7. *Contours of the error and constraint functions for the lasso* (left) *and ridge regression* (right). *The solid blue areas are the constraint regions,* $|\beta_1| + |\beta_2| \leq s$ *and* $\beta_1^2 + \beta_2^2 \leq s$, *while the red ellipses are the contours of the RSS.*

6.2.2.12. Clearly, lasso is superior to ridge w.r.t. interpretability.

6.2.2.13. In general, one expects lasso to perform better when a small number of predictors have substantial coefficients with the rest of the coefficients small or zero. Ridge will perform better with many predictors all of whose coefficients are roughly the same size.

6.2.2.14. Here is a simple special case of ridge and lasso. Let's consider $n = p$, and the design matrix $X = [x_{ij}]$ to be the identity matrix. To further simplify, assume that we are performing regression without an intercept.

- The usual least squares problem in this case is to minimize:

$$\sum_{j=1}^{p} (y_j - \beta_j)^2$$

  The solution here is clearly $\hat{\beta}_j = y_j$.

- Ridge regression minimizes:

$$\sum_{j=1}^{p} (y_j - \beta_j)^2 + \lambda \sum_{j=1}^{p} \beta_j^2$$

  Denote the above expression $f(\beta)$.

$$\frac{\partial f}{\partial \beta_j} = -2(y_j - \beta_j) + 2\lambda\beta_j$$

$$\frac{\partial^2 f}{\partial \beta_j^2} = 2 + 2\lambda > 0$$

$$\frac{\partial^2 f}{\partial \beta_j \partial \beta_k} = 0, \qquad j \neq k$$

Set the first derivative equal to 0 and solve for β_j:

$$\begin{aligned} 0 &= -2(y_j - \beta_j) + 2\lambda\beta_j \\ 0 &= -y_j + \beta_j(1+\lambda) \end{aligned}$$

Therefore, the ridge regression coefficients are

$$\beta_j^R = \frac{y_j}{1+\lambda}$$

- Lasso regression minimizes:

$$\sum_{j=1}^{p}(y_j - \beta_j)^2 + \lambda\sum_{j=1}^{p}|\beta_j|$$

Denote the above expression $f(\beta)$:

$$\begin{aligned} \frac{\partial f}{\partial \beta_j} &= -2(y_j - \beta_j) + \lambda \mathrm{sgn(\beta_j)} \\ \frac{\partial^2 f}{\partial \beta_j^2} &= 2 > 0 \\ \frac{\partial^2 f}{\partial \beta_j \partial \beta_k} &= 0, \qquad j \neq k \end{aligned}$$

Set the first derivative equal to 0 and solve for β_j. We do this in cases:
If $\beta_j > 0$, then $\beta_j = y_j - \frac{\lambda}{2}$. Notice that if $y_j < \frac{\lambda}{2}$, then the right hand side is negative. But since we assumed $\beta_j > 0$, we see that $y_j < \frac{\lambda}{2}$ implies the equation has no solutions.
If $\beta_j < 0$, then $\beta_j = y_j + \frac{\lambda}{2}$. Notice that if $y_j > -\frac{\lambda}{2}$, then the right hand side is positive. But since we assumed $\beta_j < 0$, we see that $y_j < -\frac{\lambda}{2}$ implies the equation has no solutions.
In the cases where there are no solutions, the only critical point occurs at $\beta = \overrightarrow{0}$, where the derivative is undefined.
The lasso estimates are:

$$\hat{\beta}_j = \begin{cases} y_j - \lambda/2 & \text{if } y_j > \lambda/2 \\ y_j + \lambda/2 & \text{if } y_j < -\lambda/2 \\ 0 & \text{if } |y_j| \leq \lambda/2 \end{cases}$$

We see lasso estimtes exhibit **soft-thresholding**, in which least squares coefficients which are sufficiently small get shrunk all the way to 0.

- For more complicated design matrices X, the story is more complicated. However, ridge regression more or less shrinks least squares estimates by the same proportion, and lasso shrinks all coefficients toward zero by a constant amount, with soft-thresholding taking sufficiently small coefficients to 0.

6.2.2.15. We can give a Bayesian interpretation for both ridge and lasso. Suppose the coefficient vector β has a prior $p(\beta)$. Denote the likelihood $f(Y|X,\beta)$. Then the posterior distribution is proportional to the product of the likelihood and the prior:

$$p(\beta|X,Y) \propto f(Y|X,\beta)p(\beta)$$

Assume the linear model

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

supposing the errors are independent and drawn from a normal distribution. Further, assume that

$$p(\beta) = \prod_{j=1}^{p} g(\beta_j)$$

for some density function g. Then

- If g is a Gaussian with mean zero and standard deviation a function of λ, then the **posterior mode** for β – the most likely value of β given the data – is given by the ridge regression solution.
- If g is a double-exponential (Laplace) distribution with mean zero and scale parameter a function of λ, then it follows that the posterior mode for β is the lass solution.

6.2.2.16. Becuase the Laplace distribution is steeply peaked at 0, the lasso expects that, a priori, many of the coefficients are exactly 0.

6.2.2.17. According to ChatGPT, ordinary least squares can be fit into this Bayesian paradigm with a uniform or flat prior.

6.2.3 Selecting the Tuning Parameter

6.2.3.1. To select a value for the tuning parameter λ, we choose a grid of values for λ, and compute the cross-validation error for each value of λ. We choose the tuning parameter for which the cross-validation error is smallest. Finally, the model is re-fit using all data and the chosen λ.

6.3 Dimension Reduction Methods

6.3.0.1. **Dimension reduction methods** attempt to reduce variance by fitting a least squares model using transformed predictors.

6.3.0.2. Let $Z_1, ..., Z_M$ be $M < p$ linear combinations of predictors:

$$Z_m = \sum_{j=1}^{p} \phi_{jm} X_j$$

We then fit the linear regression model:

$$y_i = \theta_0 + \sum_{m=1}^{M} \theta_m z_{im} + \epsilon_i, \qquad i = 1, \ldots, n$$

using least squares.

6.3.0.3. Notice that

$$\sum_{m=1}^{M} \theta_m z_{im} = \sum_{m=1}^{M} \theta_m \sum_{j=1}^{p} \phi_{jm} x_{ij} = \sum_{j=1}^{p} \sum_{m=1}^{M} \theta_m \phi_{jm} x_{ij} = \sum_{j=1}^{p} \beta_j x_{ij}$$

where

$$\beta_j = \sum_{m=1}^{M} \theta_m \phi_{jm}$$

So dimension reduction is a special case of fitting the linear model, but now with additional constraints on the regression coefficients.

6.3.0.4. Note that if $M = p$ and the Z_m are linearly independent, then the constraints imposed on β_j are vacuous, and so no dimension reduction occurs. Fitting using the new variables $Z_1, \ldots, Z_p$ is equivalent to performing least squares on the original predictors.

6.3.1 Principle Components Regression

6.3.1.1. **Principal Components Analysis** (PCA) is a technique for reducing the dimension of an $n \times p$ data matrix X.

6.3.1.2. The **first principal component** direction of the data is that along which the observations vary the most. If we project the data onto a line with this direction, then we will see the most variability/spread. The first principal component is the eigenvector of the sample covariance matrix X^tX with largest eigenvalue (note that the sample covariance matrix is symmetric and positive semidefinite so it has non-negative eigenvalues.

To find the next principal component, the previous principal component is subtracted from X. I'll write down all the details at some point. In particular, there is a presentation of the PCA using the singular value decomposition of X. This avoids having to compute X^tX which can be very expensive.)

6.3.1.3. PCR works well when the first few principal components suffice to capture most of the variation in the predictors as well as the relationship with the response. This is because fitting with $M << p$ variables will mitigate overfitting.

6.3.1.4. Note that PCR is not a feature selection model since it uses linear combinations of the original features.

6.3.1.5. Cross-validation is typically used to select the number of principal components M used in PCR.

6.3.1.6. Standardizing predictors is generally recommended prior to using PCR. If the variables all have the same units, this may be skipped.

6.3.1.7. Note that PCR is unsupervised – the value of Y has no bearing on the principal components. Consequently, there is no guarantee that the directions that best explain the predictors will also be the best for predicting the response.

6.3.2 Partial Least Squares

6.3.2.1. **Partial Least Squares** (PLS) is a supervised alternative to PCR. It uses Y to identify new features $Z_1, \ldots, Z_M$ that not only approximate old features well, but are also related to the response.

6.3.2.2. After standardizing the p predictors, PLS computes the first direction Z_1 by setting each ϕ_{j1} in the equation:

$$Z_1 = \sum_{j=1}^{p} \phi_{j1} X_j$$

equal to the coefficient of simple linear regression of Y onto X_j. This coefficient is proportional to $\mathrm{Cor}(Y, X_j)$. Recall from 3.1.1., the formula:

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sum_{i=1}^{n}(x_i - \overline{x})^2}$$

Notice that since we have standardized the predictors, that the denominator is $n-1$. Therefore, in computing Z_1, PLS places the highest weight on variables that are most strongly related to the response.

6.3.2.3. To obtain Z_2, we adjust each of the variables for Z_1 by regressing each variable on Z_1 and taking residuals. These residuals can be interpreted as the remaining information that has not been explained by the first PLS direction. I rather enjoy the conflation here of "information" and "variation" – at least, that's what I suspect is going on. We then compute Z_2 using this **orthogonalized** data in the same way that Z_1 was computed. This term is used since the vector of residuals is orthogonal to the original data: $(Y - \hat{Y})^t\hat{Y} = 0$.

6.3.2.4. The number of partial least squares directions M is chosen by cross-validation.

6.3.2.5. In practice, PLS performs no better than ridge regression or PCR.

6.3.2.6. Supervised dimension reduction can reduce bias, but it also can increase variance making the overall benefit of PLS relative to PCR a wash.

6.4 Considerations in High Dimensions

6.4.1 High-Dimensional Data

6.4.1.1. Most traditional techniques for regression and classification are intended for **low-dimensional** data sets in which $n \gg p$.

6.4.1.2. The **high-dimensional setting** is when $p \gg n$.

6.4.1.3. Some examples of problems: If you have a set of 200 patients you wish to study for some sort of condition, you may include in your set of predictors the nearly 500,000 single nucleotide polymorphisms (SNPs) – individual DNA mutations that are relatively common in the popultion.

6.4.1.4. Connecting online shopping patterns with search terms is another example where $p \gg n$.

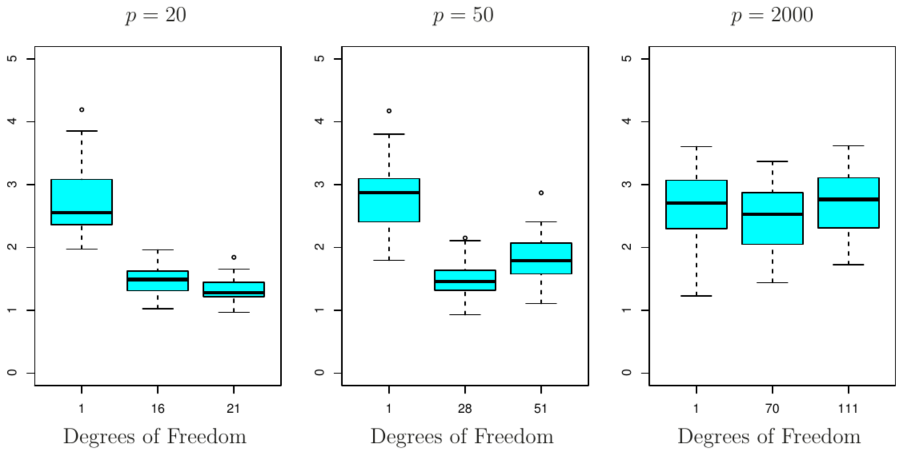
6.4.2 What Goes Wrong in High Dimensions?

6.4.2.1. As p gets larger relative to a fixed n, we see the model becoming more and more flexible. This allows for better and better fits at the training stage, but worse performance at validation because of overfitting.

6.4.2.2. Notice that if $p = n$, then a tried-and-true technique like least squares will give a perfect fit to the data since there are as many variables as there are constraints.

6.4.3 Regression in High Dimensions

6.4.3.1. Less flexible least squares models like forward stepwise selection, ridge regression, the lasso and PCR are useful for high dimensions since we avoid overfitting.



$p = 20$ $p = 50$ $p = 2000$

FIGURE 6.24. *The lasso was performed with* $n = 100$ *observations and three values of* p*, the number of features. Of the* p *features, 20 were associated with the response. The boxplots show the test MSEs that result using three different values of the tuning parameter* λ *in (6.7). For ease of interpretation, rather than reporting* λ*, the* degrees of freedom *are reported; for the lasso this turns out to be simply the number of estimated non-zero coefficients. When* $p = 20$*, the lowest test MSE was obtained with the smallest amount of regularization. When* $p = 50$*, the lowest test MSE was achieved when there is a substantial amount of regularization. When* $p = 2{,}000$ *the lasso performed poorly regardless of the amount of regularization, due to the fact that only 20 of the 2,000 features truly are associated with the outcome.*

6.4.3.2. Notice a few things from the above figure:

- regularization/shrinkage is important in high-dimensional settings.
- tuning parameter selection is crucial.
- test error tends to increase as the dimensionality increases unless the additional features are truly associated with the response.

6.4.3.3. On the third point above, note that adding signal features does improve the fitted model. But adding noise features will not since the risk of overfitting goes up as the dimensionality increases.

6.4.3.4. Therefore, technologies that allow for massive collection of measurements for thousands or millions of features are double-edged – they will lead to improved predictive models if these features are in fact relevant to the problem at hand, and will lead to worse results if not. And, even if they are relevant, the variance incurred by adding the variable may outweigh the reduction in bias they bring.

6.4.4 Interpreting Results in High Dimensions

6.4.4.1. Multicollinearity, correlation among predictors, is an extreme problem in high dimensions. Any variable can be written as a linear combination of all the other variables in the model, and so we can never know exactly which variables (if any) are truly predictive of the outcome.

6.4.4.2. For example, if one were trying to predict high blood pressure using SNPs, and forward stepwise selection indicates 17 SNPs which lead to a good predictive model on the training data, it would be incorrect to conclude that these SNPs predict bp more effectively than some other set of 17 SNPs. It is important for such a model to be further validated on independent data sets.

6.4.4.3. Reporting errors and measures of fit cannot be done in the high-dimensional setting with squared errors, p-values, R^2, etc. This is because we know that $p > n$ can give useless models with zero residuals.

6.4.4.4. However, the values of these quantities on an independent test set is valid.