Chapter 1

A Taxonomy of Problems

The term "data science" has been overused in recent years, and it has become something of a buzzword as a result[1]. However, I think it can best be described as:

> **data science**: Any endeavor in which statistics, machine learning, data analysis, computer science, and information science intersect with domain knowledge.

Data science is about using the machinery of statistics and computer science to solve real-world problems. In the clinical domain, that means incorporating methods from epidemiology, biostatistics, computer science, and machine learning with insights gained from the clinical research literature and the practical experiences of physicians, nurses, hospital administrators, operational teams, and biomedical researchers.

1.1 Project Examples

Whenever I teach, I ask students to provide some examples of projects for which they think data science could be useful. The following are real examples. They provide a broad representation of most of the types of problems clinicians and health system operations/population health teams are interested in.

[1] See also: "artificial intelligence", "machine learning", "deep learning".

1. *Unnecessary ER trips.* "Given a number of factors (types of admissions a person has had in the past, number of admissions/re-admissions, social determinants, etc.) can we predict who is going to show up at the emergency room unnecessarily"

2. *Good/poor candidates for program.* "determine if patients are good or poor candidates for one of our specialty care model bundle programs"

3. *Predicting unplanned admissions.* "predicting unplanned inpatient admissions based on many different variables (e.g. chronic conditions, engagement with primary care, etc.) and how these inputs interact with each other"

4. *Recommending an intervention.* "...stratification/prioritization of care management or other interventions or for clinical decision support...a tool would recommend an appropriate intervention based on the profile of the patient"

5. *Recommending a diagnosis.* "Based on unstructured chat conversations and also structured questions/forms/data...map out possible care pathways. For example, if someone says they have stomach pain, gives their zip code, insurance, pain tolerance and symptoms, and is logged in so we have past history, ask a few more questions and then we could determine they are 45% likely to have ulcer vs. constipation vs. food poisoning vs. appendicitis."

6. *Predicting the amount paid by patients.* "Patient bill estimates - learning from claims data typical amount paid by patients for appointment reasons/types (e.g. estimate of additional services/care administered, and associated cost, based on patient details such as age, gender, etc.)"

7. *Identifying patient subtypes.* "identify cohorts within a population with chronic conditions based on their differences in longitudinal care across the continuum of settings (inpatient, ambulatory, primary care, specialty care, etc.)"

8. *Which conversations are similar?* "using previous chat histories to train (a chatbot) and become more effective/efficient for different, future patient chat experiences"

9. *Predictors of COVID-19 outcomes.* "Get baseline diabetes control marker (HbA1C) and acute glycemic control (inpatient glucose values) and see if either is a stronger predictor of COVID-19 outcomes (ICU, intubation, death)."

10. *Factors influencing mortality in myelofibrosis.* "We see lots of patients who are ineligible for clinical trials based on comorbidities and underlying organ dysfunction. However it is unclear how these factors affect OS. I would like to extract comorbidity data and baseline laboratory factors in patients with myelofibrosis to see how these factors affect mortality, if controlled for such important factors such as treatment, age, sex, insurance, number of comorbidities, and clinical risk score (DIPSS)."

11. *Non-adherence and difficult-to-treat asthma.* "We want to see whether non-adherence to prescribed inhaled corticosteroids plays a major role in poorly controlled asthma. Difficult-to-treat asthma can be evaluated by the number of ED visits, hospitalizations, prescriptions of prednisone and prescriptions of biological therapies. Using EPIC [we] can obtain medicine reconciliation information, of prescriptions sent, what proportion of those prescriptions were dispensed by Pharmacy. Question is can we find associations between the percentage of prescriptions filled and difficult-to-treat asthma."

12. *Impact of diabetes and hyperglycemia on progression-free survival.* "Aim: Assess the impact of diabetes and hyperglycemia on first-line systemic therapy response (progression-free survival) in patients with advanced non-small cell lung cancer. Diabetes- defined by presence of diagnosis codes coding for diabetes. Hyperglycemia- random glucose >200 ng/dL. Covariates of interest- age, sex, other treatments (RT, surgery), malignancy characteristics (stage, histology), smoking history, ecog (performance status), comorbidities, medications (steroids, anti-hyperglycemics)"

13. *Effect of statin use on MACE.* "Retrospective cohort study in elderly patients with CAD taking statins…exposed group are patients on a high-intensity statin; control group are patients on a moderate- or low-intensity statin. Participants matched based on age, gender, LDL category, and Elixhauser index category…The primary efficacy outcome

would be the time-to-first-event of 3-point MACE[2].”

14. *Clustering patients with NAFLD.* “We wanted to understand non-alcoholic fatty liver disease (NAFLD) better, so we developed a cohort of NAFLD patients using EMR-based criteria and then clustered them based on co-morbidities, medications, vital signs, and lab values to identify NAFLD subtypes. We then characterized the phenotypes and outcomes of the different subtypes.”

1.2 Abstracting the Problem

All of these examples describe situations where we want to use data to answer questions of clinical or operational importance. While the details differ in each scenario, the important thing to notice here is that many of the tasks themselves are structurally similar.

For example, all of the items except $7-8$ and 14 describe situations where we want to associate information about a patient with a particular outcome or recommendation. Using information about a patient to estimate the size of a bill (#6) may appear to be a very different problem than uncovering factors influencing myelofibrosis mortality (#10), but the structure of the two problems is similar: the patient features are used as input, and the output is whatever quantity you care about (e.g. the cost to the patient in dollars or the probability of mortality by a certain timepoint).

Learning to see these types of similarities will give you a tremendous amount of power when attacking new problems in clinical data science. It will allow you to confidently deploy methods you used to solve one problem on a wide range of other problems. Each new method you learn then multiplies your capacity to solve problems, rather than adding to it.

Question 1.1

How are items $7-8$ and 14 different from the rest?

[2]MACE stands for “Major Adverse Cardiac Event”. The 3-point MACE is a composite of nonfatal stroke, nonfatal myocardial infarction, and cardiovascular death.

Question 1.2

How are items 1 − 6 similar to items 9 − 13 and how are they different?

Question 1.3

How do items 1 − 3 differ from items 4 − 5 and how are they similar?

Question 1.4

How do items 1 − 3 differ from item 6? How is item 6 different from all of the other items?

Question 1.5

How do items 9 − 11 differ from items 12 − 13?

1.3 Terms and Contrasts

The basic ways in which clinical data science problems vary can be characterized using a few broad conceptual distinctions. These draw from both traditional clinical disciplines, like epidemiology, as well as machine learning/statistics.

1.3.1 Guidance vs. Understanding

Before beginning any study, it is important to carefully consider the study's goal and how the findings from the study will be used. This will help guide you in choosing appropriate methods. For example, in some studies we care mainly about using data to provide **guidance** that will enable us to perform our jobs better in the future. We may want to predict whether a patient is likely to experience an adverse outcome, or we may want to learn the type of patient who is most likely to benefit from a particular treatment. In these cases, we want the data to guide us in making better choices.

Now, contrast this with a study whose primary goal is scientific **understanding**. In this case, we care more about using data to improve our understanding of a phenomenon than in operationalizing those findings. For example, we may be interested in whether a particular genetic variant affects a phenotype, or we may want to establish a causal link between a particular treatment and an outcome.

The distinction is fuzzy and often imperfect, and the same kinds of methods can often be used in both cases. Depending on the goal, however, one may be willing to make certain compromises. For example, complex, "black box" predictive models (e.g. deep learning models) may be appropriate when the goal is guidance, but offer little in the way of understanding. Conversely, regression models have become the de facto standard for clinical trials and causal inference, but may not lead to optimal predictive ability. In situations where the primary goal is a rigorous understanding of causal relationships, that may not matter as much.

1.3.2 Observational Study vs. Experiment

In **experimental studies**, the investigator manipulates some aspect of the subjects' experience and studies its effect on the outcome of interest. For example, here is the NIH's definition of a **clinical trial**:

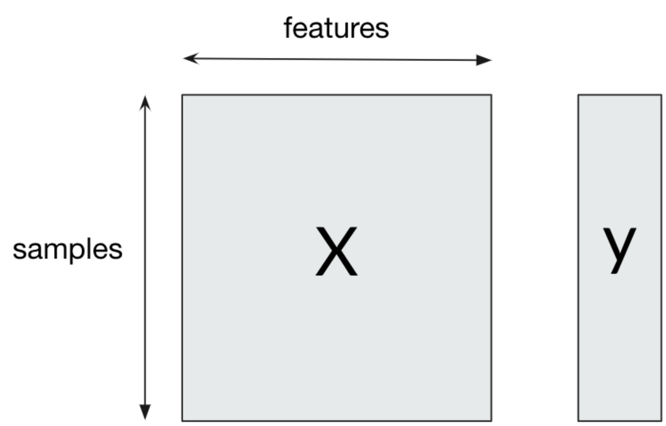
> A research study in which one or more human subjects are prospectively assigned to one or more interventions (which may include placebo or other control) to evaluate the effects of those interventions on health-related biomedical or behavioral outcomes.

A clinical trial, therefore, is an experiment, because we control the intervention and monitor the effect of that intervention on one or more outcomes. Usually experimental studies employ some type of **randomization** to ensure that comparisons between different intervention groups are fair.

An **observational study**, in contrast, makes no attempt to interfere with its subjects. Instead, these individuals are simply observed, and inferences are made about the associations between different parameters and the outcome(s). Observational study designs and analytic plans are carefully designed to minimize the effects of different sources of bias that can creep in due to lack of randomization. Although they're not usually referred to using this terminology, virtually all "big data" and machine learning oriented studies in healthcare are observational studies, because they use large datasets that were collected for other purposes.

1.3.3 Types of Machine Learning

This distinction, most often found in discussions of machine learning, refers to the way in which training data is applied to solve a problem. In **supervised learning**, the training data consist of pairs of input features and labels, and the algorithm learns to predict the value of the label from the input features. The general setup for supervised learning looks like this:



In **unsupervised learning**, only the input features are present (i.e. no y) and the algorithm learns to recognize patterns, clusters, or other structure in the inputs. Although they're almost never referred to using this terminology, clinical studies that examine the effect between one or more exposures and an outcome are examples of supervised learning. Studies that attempt to uncover groups, or clusters, of similar patients or samples are examples of unsupervised learning.

There are also two other types of machine learning. In **semi-supervised learning**, a small amount of labeled data is used to create a much larger, weakly-labeled set of training data that is then fed to a supervised learning algorithm. In **reinforcement learning**, an algorithm is trained with a reward system which provides feedback on the quality of the action the system performs in a given situation instead of (as in supervised learning) simply providing the "right answer".

Chapter 2

The Basics of Classification

Classification is a form of supervised learning in which our goal is to learn a mapping between some features, x, and an output, y. In classification, the output, y, is a category. In **binary classification** (by far the most common), there are only two categories: yes or no, usually represented as "0" (no) or "1" (yes). In **multi-class classification**, there are more than two categories.

To learn an appropriate mapping, we feed **training data** to a **learning algorithm**. Different algorithms learn different types of mappings.

2.1 Definitions

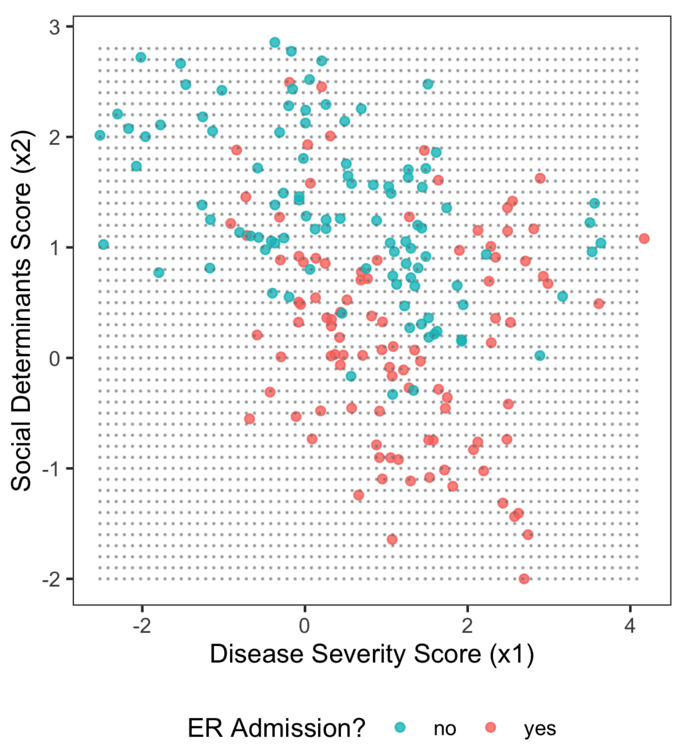
- **Training data:** The data used, along with an appropriate learning algorithm, to create the mapping between input and output. It is composed of **training examples**, a.k.a. **samples**, each consisting of one or more input features and a single output.
- **Test data:** An independent dataset, not used in model training, on which the performance of a trained supervised learning model is evaluated.
- **Feature:** Also known as a **predictor**, or **covariate**, one of the inputs to a supervised learning algorithm.
- **Output:** Also known as the **outcome**, or **label**, the thing you are trying to predict.

- **Feature space:** Envisioning each feature as having its own axis that is orthogonal to all of the other features' axes, the multidimensional space spanned by those axes (or rather: unit vectors in the directions of those axes)

- **Extrapolation:** Making predictions outside the region of the feature space occupied by the training data. This will often lead to errors.

2.2 Visualizing the Classification Problem

Imagine we want to predict whether a patient will be readmitted to the emergency room (ER) within 30 days of hospital discharge. We gather data on two predictors: a disease severity score (x_1), which characterizes the severity of illness, and a social determinants score (x_2), which characterizes the patient's socioeconomic status. We have data on 200 patients.

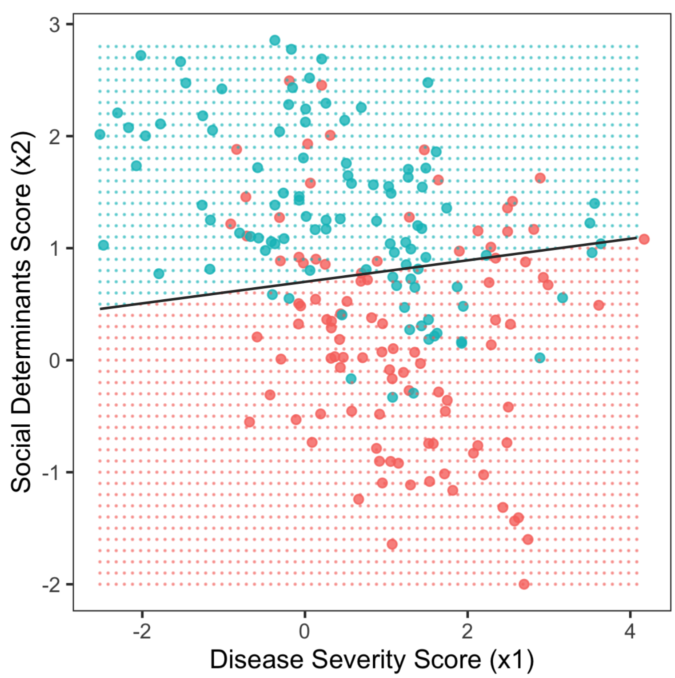


In this figure, the color refers to whether a patient was readmitted (blue = “no”, red = “yes”). The location of each point is governed by the patient’s disease severity score (x_1, horizontal axis) and social determinants score (x_2, vertical axis). Our goal in classification is to draw a **decision boundary** through this space, on one side of which we will predict that the patient is readmitted, and on the other side not.

2.3 Three Classification Algorithms

2.3.1 Logistic Regression

The simplest decision boundary is, arguably, a line. The logistic regression algorithm simply draws a line[1] through the feature space that divides the positive and negative training examples.



[1] In a higher-dimensional feature space, the decision boundary for logistic regression is a **hyperplane**.

The output of a fitted logistic regression model from R looks like this:

```
Call:
glm(formula = y ~ x1 + x2, family = "binomial", data = df)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-1.88232  -0.90614  -0.05965   0.86579   2.28489

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   0.9780     0.2945   3.321 0.000897 ***
x1            0.1344     0.1372   0.980 0.327272
x2           -1.3981     0.2316  -6.035 1.59e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 277.26  on 199  degrees of freedom
Residual deviance: 209.54  on 197  degrees of freedom
AIC: 215.54

Number of Fisher Scoring iterations: 4
```

The equation of the line (or, in higher dimensions, hyperplane) that forms the decision boundary in logistic regression can be obtained by setting the linear sum of coefficients of this model equal to zero.

$$\begin{aligned} 0.9780 + 0.1344x_1 - 1.3981x_2 &= 0 \\ \implies x_2 &= \frac{0.9780 + 0.1344x_1}{1.3981} \end{aligned}$$

At any point, (x_1, x_2), in the feature space, the model's predicted probability of a positive outcome (i.e. probability of an ER readmission) is related to the coefficients by this equation

$$\log \frac{P[Y=1]}{1 - P[Y=1]} = 0.9780 + 0.1344x_1 - 1.3981x_2$$

The decision boundary occurs when $P[Y=1] = 0.5$ (total uncertainty, e.g. a coin toss). Another way to write this equation is:

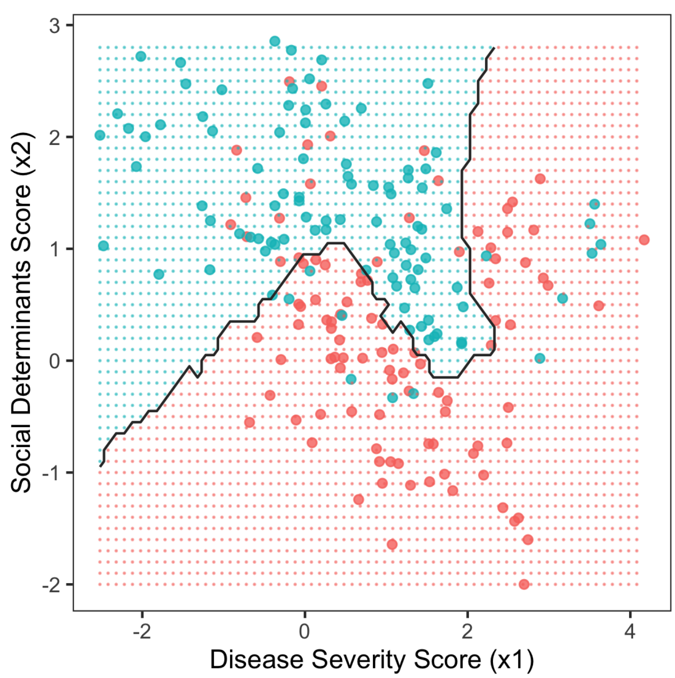
$$P[Y=1] = \frac{1}{1 + \exp(-(0.9780 + 0.1344x_1 - 1.3981x_2))}$$

The functional form on the right, $1/(1+\exp(-z))$, is called the **logistic function**; this is how logistic regression got its name. We will learn much more about the math behind logistic regression in subsequent chapters.

2.3.2 K Nearest Neighbors (KNN)

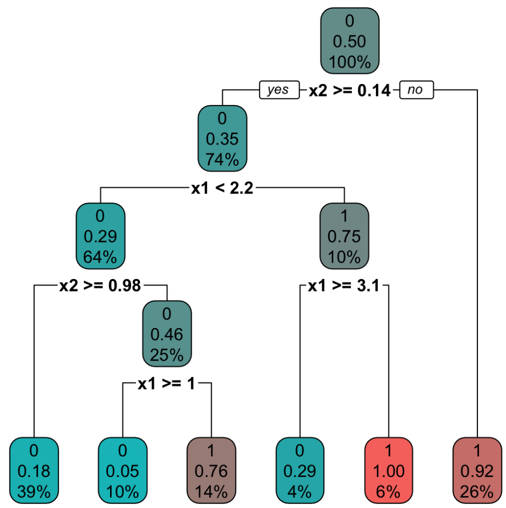
Another – completely different – approach to classification is to start with no assumptions about the shape of the decision boundary. To make a prediction about a new patient, we simply identify the K nearest neighbors to that patient from our training set and allow them to vote on whether or not the new patient will be readmitted. The parameter K must be set independently and is called a **hyperparameter**.

Here is the decision boundary for KNN with $K = 15$:

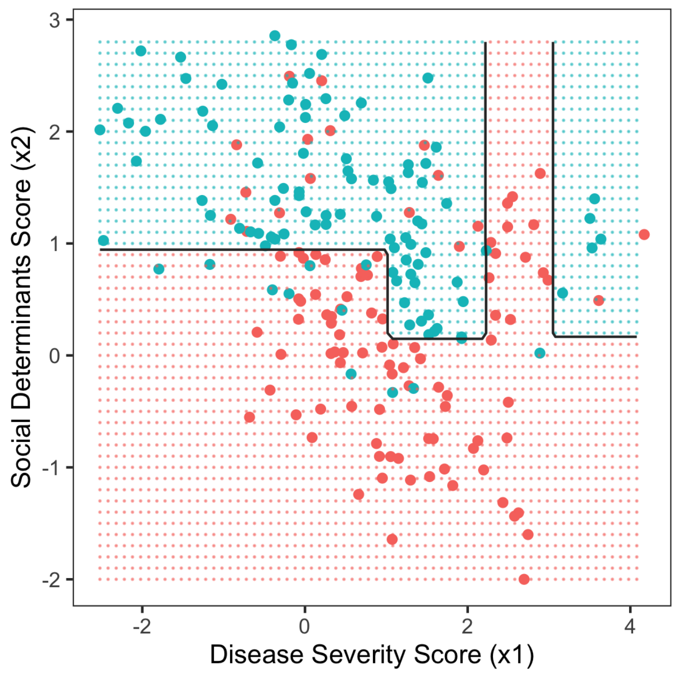


2.3.3 Decision Tree

Finally, we may choose to use our training data to build a decision tree, which will allow us to make predictions on new patients using a series of simple yes/no questions. There are different decision tree learning algorithms, but here is the tree produced by a famous one called CART:



And here is the decision boundary produced by this tree:



Question 2.1

How can you tell, just by looking at these images, which feature (x_1 or x_2) impacts the outcome the most? Which one is it?

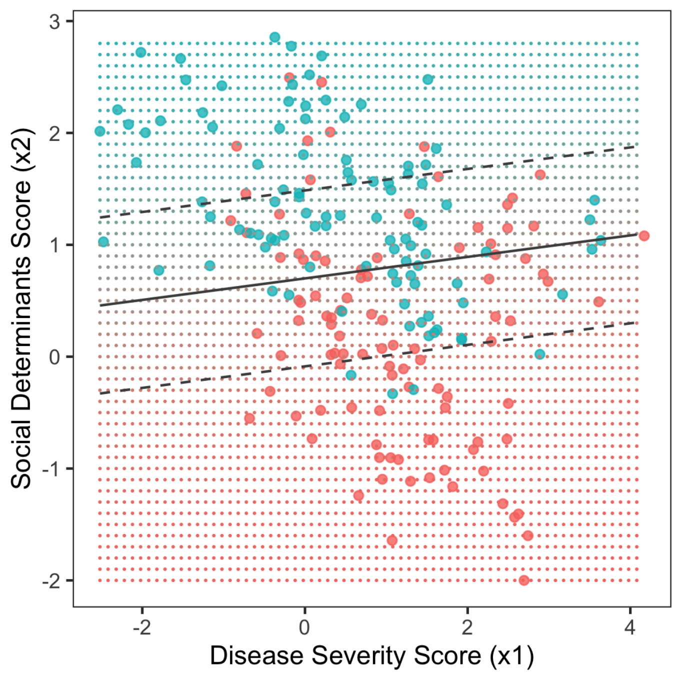
Question 2.2

There are six rectangular regions in the picture of the decision tree decision boundary. Each corresponds to one of the six leaves of the tree. Identify all six and which leaves they correspond to on the decision tree.

2.4 Classification with Probabilities

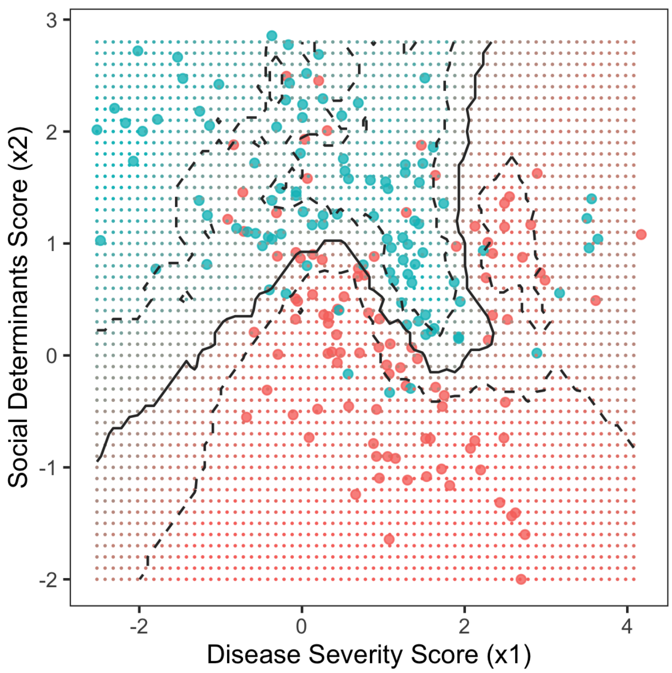
We can think of classification as simply drawing a decision boundary, but underlying each algorithm is a quantitative assessment of each point in the feature space. Each algorithm is, in its own way, able to provide a degree of certainty, or **probability**[2], that a point belongs to the positive outcome class.

For example, here is the feature space of the example we just saw, colored by the probability, according to logistic regression, that a sample at each point should be classified as positive (i.e. the patient will be readmitted to the ER):

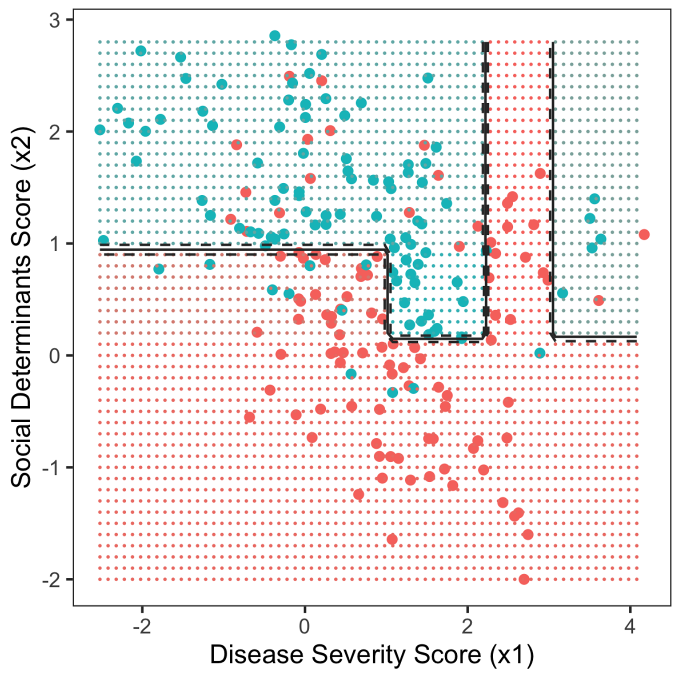


The solid line is the decision boundary, and the dashed lines indicate where the probability of a positive outcome (ER readmission) is 25% (top line) and 75% (bottom line). You can see that the color of the background gets purer red or purer blue the further you get from the decision boundary, but that near the decision boundary, the color is rather murky. That murkiness reflects the algorithm's uncertainty about the outcome. At the decision boundary, it is maximally uncertain. There the probability of a positive outcome is 50%: a coin toss. Here is a similar plot for KNN ($K = 15$):

[2]Pedantic footnote: this is a Bayesian definition of probability, as opposed to a frequentist definition. More on that later.



You can see that the shapes of the 25% and 75% probability lines have much more complex shapes than for logistic regression, but the story is the same: you have regions of pure blue or red, where the algorithm is certain, and you have a murky region near the decision boundary. Now, finally, here is the same plot for the decision tree:



The color of the background in the regions corresponding to the six leaves of the tree is the same throughout each region. That's because the probability in each rectangular region (corresponding to each leaf of the tree) is constant. It equals the number of red dots in that region divided by the total number of dots.

Question 2.3

What are the advantages and disadvantages of each algorithm?

1. Logistic regression?
2. KNN ($K = 15$)?
3. Decision tree?

Question 2.4

What makes a good classification algorithm? Consider issues of accuracy, generalizability, and speed (both to train the algorithm and to use it to make predictions on new samples).

Chapter 3

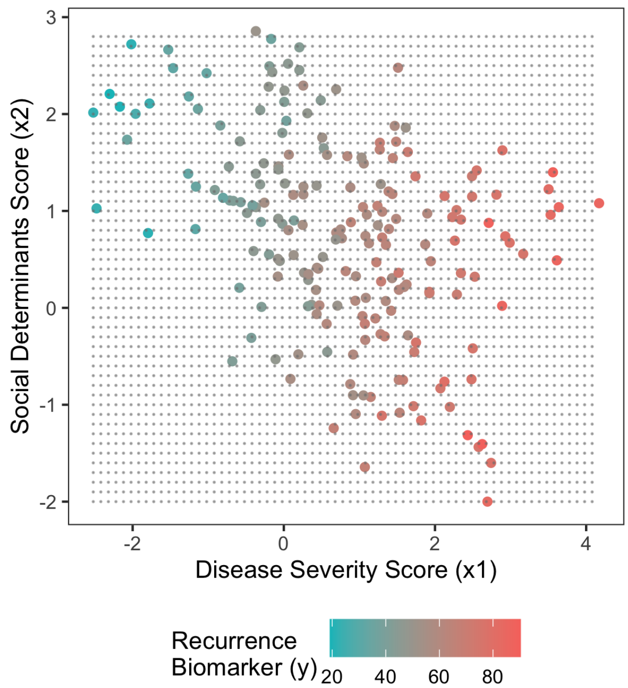
The Basics of Regression

Classification is a form of supervised learning in which the outcome is a category. **Regression** is another form of supervised learning in which the outcome is a numeric value. For example, it may be a lab value, physical characteristic (height, weight, etc.), or numeric measurement (e.g. oxygen saturation).

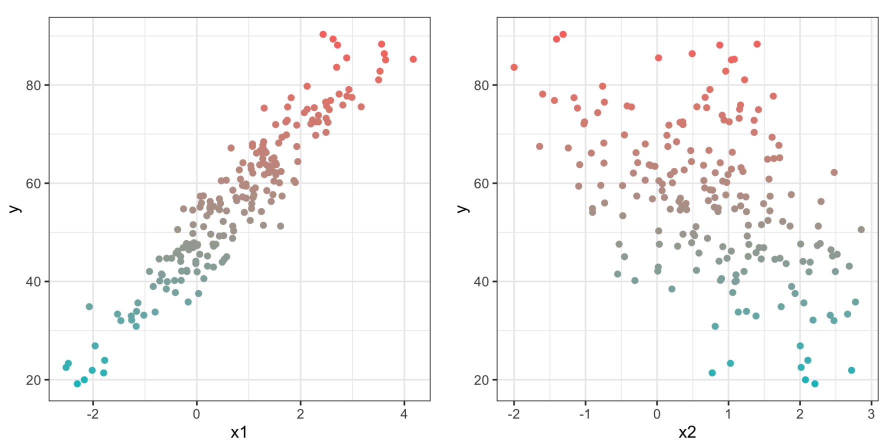
3.1 Visualizing the Regression Problem

Let's consider the same setup from Section 2.2 but this time with a quantitative outcome: a "recurrence biomarker" that indicates the likelihood of recurrence of disease.

Again, we have data on two predictors: a disease severity score (x_1), which characterizes the severity of the illness for which the patient was originally treated, and a social determinants score (x_2), which characterizes a patient's socioeconomic status. We have measurements of x_1 and x_2 on the same 200 patients as in Section 2.2.



This is a plot of the data in a single plane. The color represents the value of the recurrence biomarker – the height of the point above the plane. We want to design a model that will predict the value of the biomarker (y) based on the values of the two predictors, x_1 and x_2. These plots show the **univariate** relationship of each predictor with the outcome.



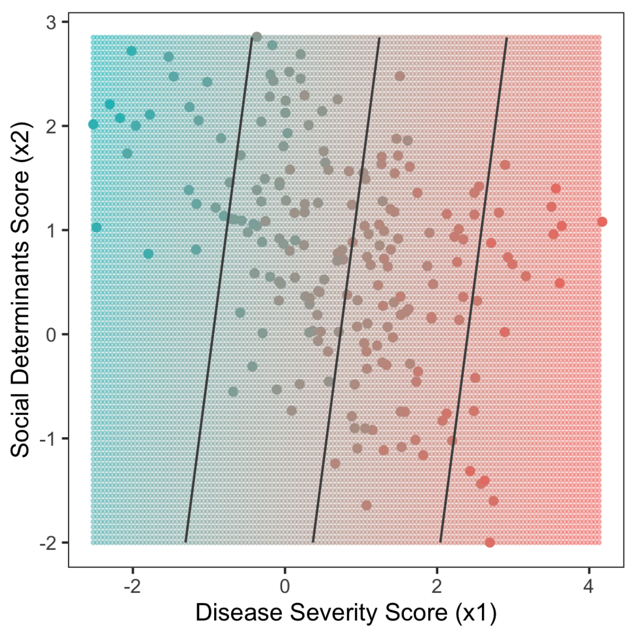
Question 3.1

Which of the two predictors, x_1 or x_2, appears to more strongly influence the value of the recurrence biomarker? Explain your reasoning using evidence from the preceding three plots.

3.2 Three Regression Algorithms

3.2.1 Linear Regression

The regression analogue of logistic regression is **linear regression**[1]. Linear regression creates a hyperplane that slices through the cloud of training datapoints such that it passes as close as possible, on average, to the data. This is, of course, easiest to see when the feature space is two-dimensional, as it is here:



[1]The terminology here is confusing. When we learn about generalized linear models in Chapter 12, you'll see why logistic regression has the word "regression" in its name even though it's a classification algorithm.

The three lines shown here sit on the hyperplane learned by the linear regression model. They are located at heights corresponding to the 25th, 50th, and 75th percentiles of the outcome, y (the biomarker value). The plane tilts downward toward the upper left corner of the $x_1 \times x_2$ grid and upward toward the bottom right corner. It may be helpful to visualize grabbing the $x_1 \times x_2$ plane and rotating/translating it so that it passes through the middle of the training data. Here is a summary of the trained linear regression model:

```
Call:
lm(formula = y ~ x1 + x2, data = df)

Residuals:
     Min       1Q   Median       3Q      Max
-11.9218  -3.1032   0.2891   2.8316  12.5813

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  49.8600     0.5370  92.844  < 2e-16 ***
x1           10.4372     0.2855  36.555  < 2e-16 ***
x2           -1.8824     0.3609  -5.215 4.63e-07 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4.769 on 197 degrees of freedom
Multiple R-squared:  0.9026,	Adjusted R-squared:  0.9016
F-statistic: 912.4 on 2 and 197 DF,  p-value: < 2.2e-16
```

At each point (x_1, x_2) in the feature space, the model's predicted value of the recurrence biomarker, $\hat{y}$, is

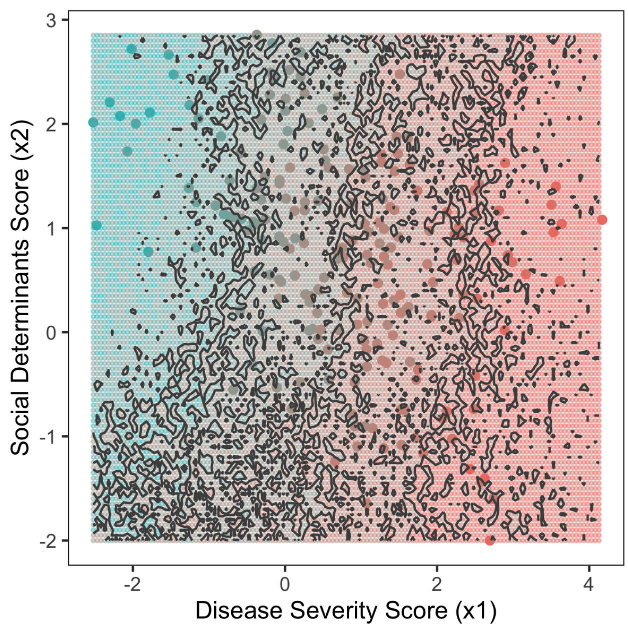
$$\hat{y} = 49.8600 + 10.4372x_1 - 1.8824x_2$$

Question 3.2

Compare and contrast the output from the linear regression model with the output from the logistic regression model in Chapter 2. What looks the same? What looks different? What is being predicted in each case?

3.2.2 K Nearest Neighbors (KNN)

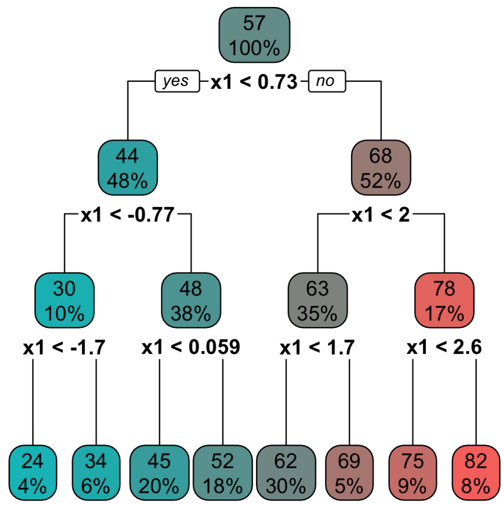
Regression using KNN works very similarly to KNN for classification. In classification, we allow the nearest K points to vote on the label of a new test point. In regression, we **interpolate** between the values of the surrounding points to come up with the value of y for a test point. Typically this is done just by averaging the y values of the nearest K points, but you can also do something more sophisticated, like weight their contributions by distance to the test point. Here is a contour plot of the regression surface produced by KNN ($K = 15$) for our example:



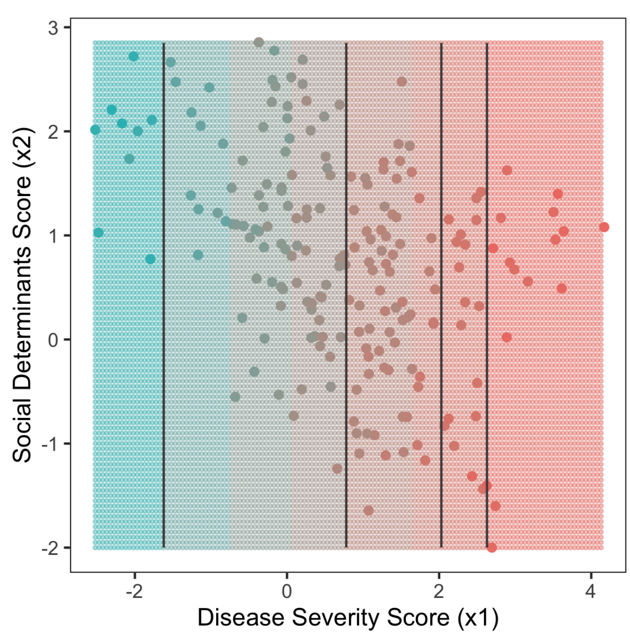
The contours are again drawn at the 25th, 50th, and 75th percentiles of the outcome, y. This looks like a bit of a mess compared to the linear regression plot, but at the same time, the KNN algorithm is able to capture arbitrarily complex relationships between x_1, x_2, and y that can be missed by other regression algorithms.

3.2.3 Decision Tree

Decision tree regression is similar to decision tree classification except that the output at each leaf is not a class label or the probability of membership in the positive training class (both of which are shown on the tree in Section 2.3.3), but a numeric value. That value corresponds to the mean outcome value for the points in that leaf.



The predicted biomarker values for a decision tree trained on this dataset (created using the `rpart` package in R with default parameters) are shown here:



You can see that the decision tree always chooses to split on x_1, the disease severity score, rather than x_2. Revisit Question 3.1 to remind yourself of why this is. The regression surface produced by the decision tree looks like a set of stairs climbing higher and higher as one moves from left to right across the $x_1 \times x_2$ plane. The predicted value of y, the recurrence biomarker, is constant within each stair.

Question 3.3

Compare this decision tree with the decision tree for the classification problem in Chapter 2. What is the same? What is different?

Question 3.4

This **regression tree** has eight leaves. What region of the feature space does each leaf correspond to?

Question 3.5

What are the advantages and disadvantages of each of these three regression algorithms (linear regression, KNN, regression tree)?

Chapter 4

Probability Distributions

Many of the methods we will examine in these workshops depend on basic concepts from probability theory. For example, linear and logistic regression are members of a class of supervised learning algorithms called **generalized linear models** (see Chapter 12) which make assumptions about the type of probability distribution followed by the outcome variable. Decision trees use a concept called **entropy** (see Chapter 7), whose mathematical formulation depends on the probability distribution underlying the outcome. Many **hypothesis tests** (see Chapter 6) likewise rely on probabilistic assumptions about the data. Probability is everywhere.

The following sections review some key probability concepts – in an extremely hand-wavey and non-rigorous way – and the properties of some of the most common probability distributions you will encounter in machine learning and statistics.

4.1 Definitions

A **probability distribution** is just a mathematical function that provides the relative likelihoods of various possible outcomes of an observation. We call the quantity that is being observed a **random variable**. Probability distributions can be discrete or continuous. The random variable involved can be a number, a vector of numbers, a category/class, etc. The **sample space** is the set of all possible outcomes. The integral (or sum) of the probability distribution over the entire sample space is 1.0. You will often hear probability distributions for continuous random variables referred to as **probability densities**.

Probability distributions are grouped into families that are characterized by their overall shapes. These families contain **parameters** that, when varied, produce different distributions. Specific probability distributions from within a single family can often look quite different.

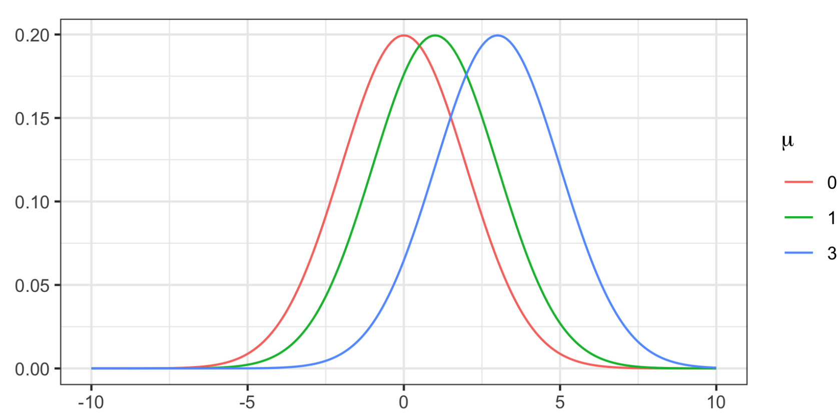
We use the notation $E[x|\theta]$ to refer to the **expected value**, or mean, of a distribution, given its parameter(s), θ. There can be more than one parameter, and it will not always be called θ; this is just an example. We use the notation $\text{var}(x|\theta)$ to refer to the **variance**, or spread, of a distribution around its mean.

4.2 Normal Distribution

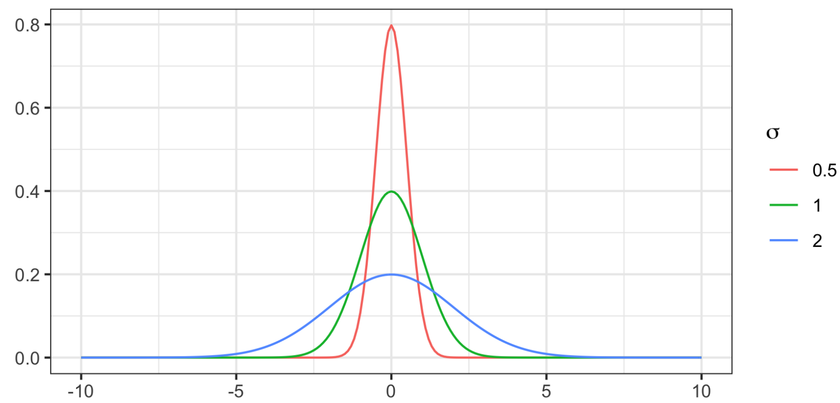
Also called the **Gaussian distribution**, the normal distribution is probably the most well-known continuous probability distribution. It has the following properties:

$$p(x|\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}}e^{-\frac{(x-\mu)^2}{2\sigma^2}} \qquad E[x|\mu,\sigma] = \mu \qquad \text{var}(x|\mu,\sigma) = \sigma^2$$

where $x \in \mathbb{R}$. We will abbreviate the normal distribution as $\mathcal{N}(\mu,\sigma)$. The value of μ changes the position of the center of the normal distribution.



The value of σ changes the width of the normal distribution.



Question 4.1

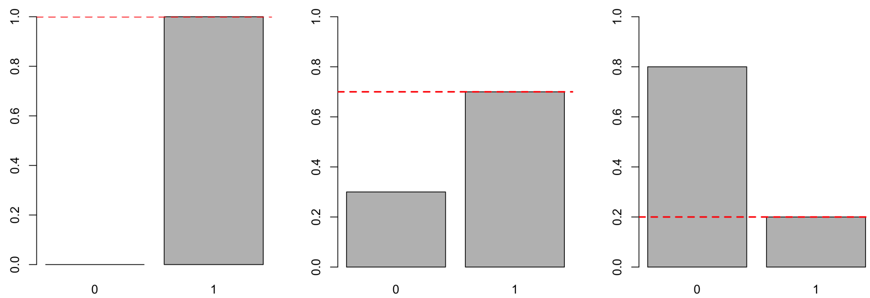
List 5 random variables from medicine or biology that should follow normal distributions.

4.3 Bernoulli Distribution

The **Bernoulli distribution** is a discrete probability distribution with the following properties:

$$p(x|\mu) = \mu^x(1-\mu)^{1-x} \qquad E[x|\mu] = \mu \qquad \text{var}(x|\mu) = \mu(1-\mu)$$

where $x \in \{0, 1\}$. It is used to model events where the outcome is yes/no. Think of it as a weighted coin, with μ the probability that the coin comes up "heads" on a single toss. Here are three Bernoulli distributions with (from left to right) $\mu = 1.0, 0.7, 0.2$. The number along the bottom is x, which can only be 0 or 1.



The **categorical distribution** is a generalization of the Bernoulli distribution to an outcome with more than two levels. The categorical distribution looks like this:

$$p(x|\phi_1, \ldots, \phi_K) = \phi_1^{\mathbb{I}(x=1)} \phi_2^{\mathbb{I}(x=2)} \cdots \phi_K^{\mathbb{I}(x=K)}$$

where $\sum_{k=1}^{K} \phi_k = 1$. The term $\mathbb{I}(x = j)$ is an **indicator**. It equals 1 if $x = j$ and 0 otherwise. For example, $\mathbb{I}(x = 2)$ is 1 if $x = 2$ and 0 otherwise.

Question 4.2

List 5 random variables from medicine or biology that should follow Bernoulli distributions.

4.4 Binomial Distribution

The **binomial distribution** models the number of positive outcomes, x, out of n independent[1] Bernoulli trials, each of which is positive with probability μ. This distribution has the following properties, with $x \in \{0, \ldots, n\}$:

$$p(x|n, \mu) = \binom{n}{x} \mu^x (1 - \mu)^{n-x} \qquad E[x|\mu] = n\mu \qquad \text{var}(x|\mu) = n\mu(1 - \mu)$$

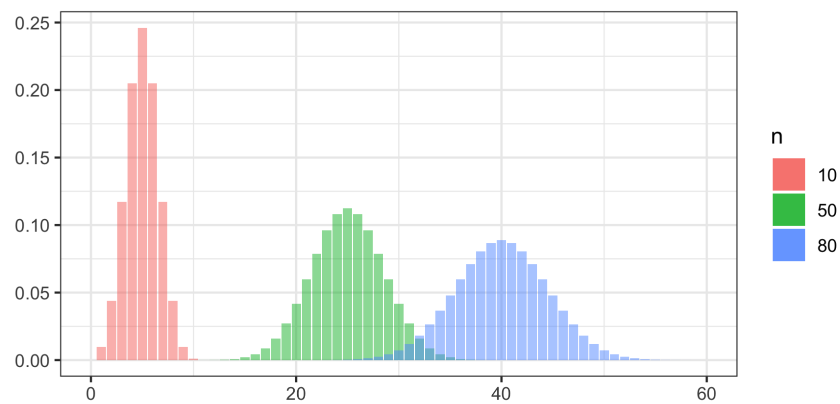
where the notation $\binom{n}{x}$ is defined as:

$$\binom{n}{x} = \frac{n!}{x!(n - x)!}.$$

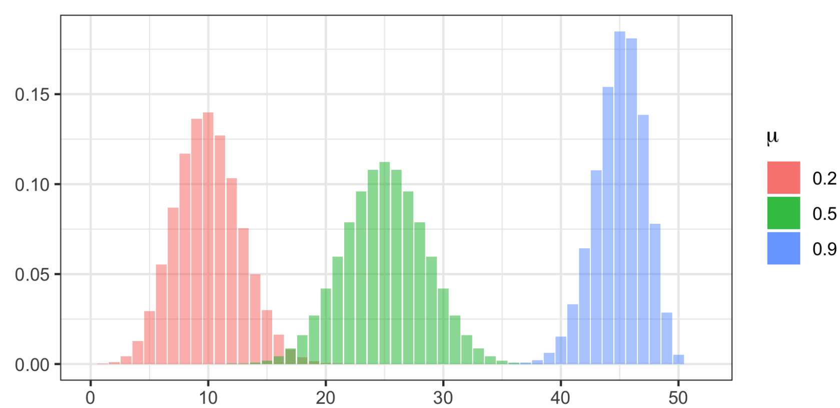
[1] The word **independent** just means that the outcome of one trial does not influence the outcome of any other trial.

This notation denotes the number of ways it is possible to choose x things out of a group of n things, where the ordering doesn't matter. The exclamation point denotes the **factorial function**: $x! = x(x-1)(x-2)\cdots(2)(1)$.

The shape of the binomial distribution is governed by the values of n and μ. Here, we vary n but keep μ constant at 0.5:



And here we vary μ but keep n constant at 50:



Question 4.3

List 5 random variables from medicine or biology that should follow binomial distributions.

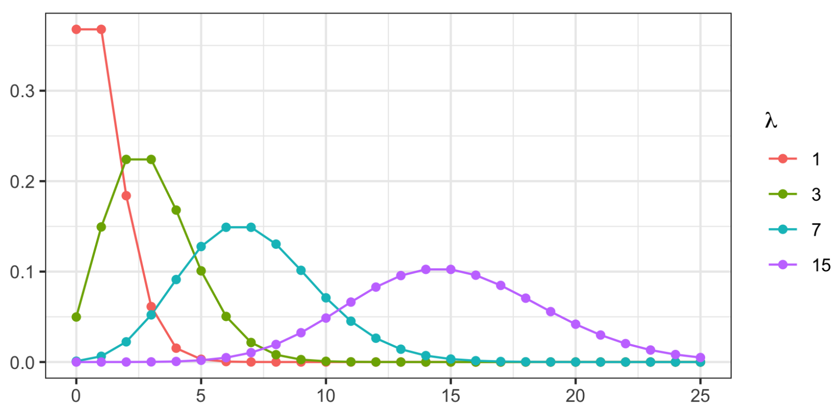
4.5 Poisson Distribution

The **Poisson distribution** is a probability distribution that is often used to model discrete quantitative data, such as counts. It has the following properties:

$$p(x|\lambda) = \frac{e^{-\lambda}\lambda^x}{x!} \qquad E[x|\lambda] = \lambda \qquad \text{var}(x|\lambda) = \lambda$$

where $x \in \{0, 1, 2, \ldots\}$. Below are four examples of Poisson distributions. If events of a particular type occur continuously and independently at a constant rate (**Poisson process**), the number of events within a time window of fixed width will be distributed according to the Poisson distribution, with rate parameter λ proportional to the width of the window.

Situations where the population size, n, is large, the probability of an individual event, p, is small, but the expected number of events, np, is moderate (say five or more) can generally be modeled using a Poisson distribution with $\lambda = np$.



Question 4.4

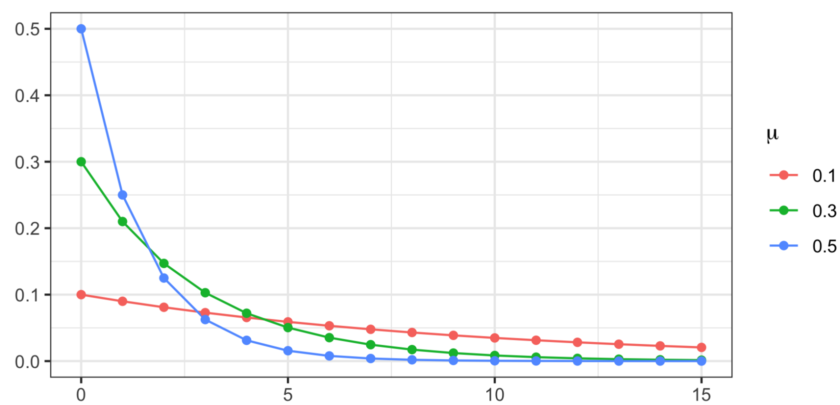
List 5 random variables from medicine or biology that should follow Poisson distributions.

4.6 Geometric

The **geometric distribution** models the number of failures in a sequence of Bernoulli trials before the first success. It has the following properties:

$$p(x|\mu) = (1-\mu)^x \mu \qquad E[x|\mu] = \frac{1-\mu}{\mu} \qquad \mathrm{var}(x|\mu) = \frac{1-\mu}{\mu^2}$$

for $x \in \{0, 1, 2, \ldots\}$, where μ refers to the probability (in the Bernoulli trial) that the trial is a success. Some examples of geometric distributions with different μ are shown below:



> **Question 4.5**
>
> List 5 random variables from medicine or biology that should follow geometric distributions.

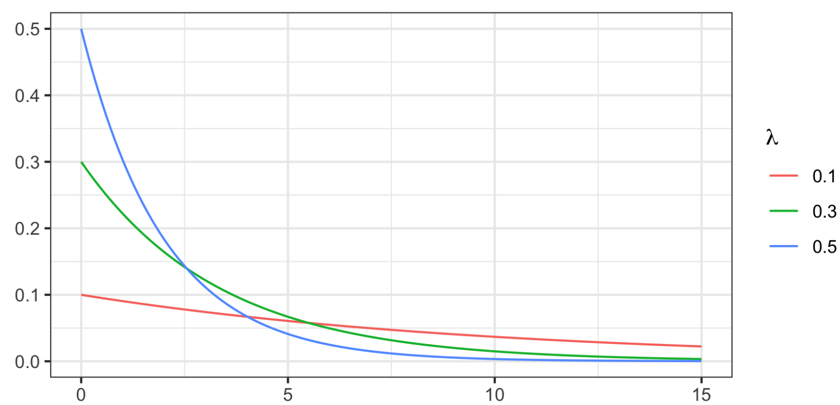
4.7 Exponential

The **exponential distribution** is a continuous probability distribution that models waiting times between events that happen independently and continuously at a constant rate (Poisson process), as well as many other random variables[2]. It has the following properties:

$$p(x|\lambda) = \lambda e^{-\lambda x} \qquad E[x|\lambda] = \frac{1}{\lambda} \qquad \text{var}(x|\lambda) = \frac{1}{\lambda^2}$$

where $x \in \mathbb{R}^+$ (x is a positive real number, or zero). The exponential distribution is the continuous analogue of the geometric distribution. It is memoryless, which means that the distribution of a waiting time until an event does not depend on how much time has elapsed already.

Here are some different exponential distributions. Compare them to the geometric distribution, above.



Question 4.6

List 5 random variables from medicine or biology that should follow exponential distributions.

4.8 Chi-Squared Distribution

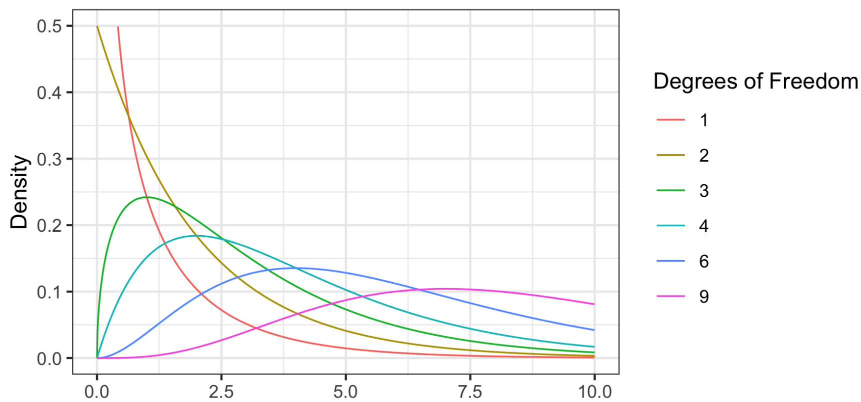
How this distribution arises:

[2] For example, in an epidemiologic model of an infectious process like COVID-19 community spread, exponential waiting times are often used to model transitions between the susceptible, exposed, infectious, and recovered compartments in the model.

1. If $Z \sim \mathcal{N}(0,1)$, the distribution of $U = Z^2$ is called the chi-squared distribution with one degree of freedom.

2. If $U_1, U_2, \ldots, U_k$ are independent χ^2_1 random variables, their sum, $V = \sum_{i=1}^{k} U_i$ follows χ^2_k, a chi-squared distribution with k degrees of freedom.

You'll often see the chi-squared distribution used as the sampling distribution for the sample variance in a variety of statistical hypothesis tests. It looks like this:



The parameter k, the **degrees of freedom**, controls the shape of the chi-squared distribution. The actual formula for the chi-squared distribution looks a bit intimidating, but I'm including it here so you can compare it to the other distributions we've seen:

$$p(x|k) = \frac{1}{2^{k/2}\Gamma(k/2)} x^{k/2-1} e^{-x/2}$$

$$E[x|k] = k \qquad \text{var}(x|k) = 2k$$

The gamma function shown in the denominator of the probability density,

$$\Gamma(z) = \int_0^\infty x^{z-1} e^{-x} dx,$$

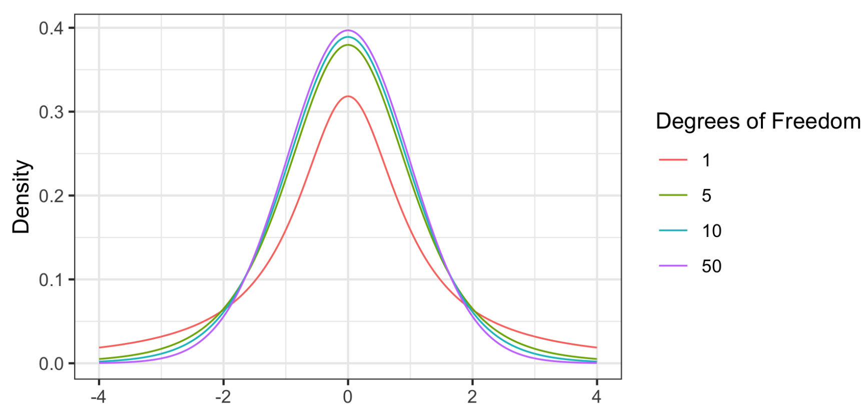
is a generalization of the factorial function to complex numbers. For any positive integer n, $\Gamma(n) = (n-1)!$.

4.9 Student's T Distribution

If $Z \sim \mathcal{N}(0,1)$ and $U \sim \chi^2_k$ and Z and U are independent,

$$T = \frac{Z}{\sqrt{U/k}} \sim t_k$$

or in words, the statistic T follows a t-distribution with k degrees of freedom. The T distribution plays an important role in a family of statistical hypothesis tests called T-tests.



Again, the functional form of the T distribution is a bit intimidating, but I'm including it for completeness:

$$p(x|k) = \frac{\Gamma\left(\frac{k+1}{2}\right)}{\sqrt{k\pi}\,\Gamma\left(\frac{k}{2}\right)} \left(1 + \frac{x^2}{k}\right)^{-\frac{k+1}{2}}$$

$$E[x|k] = 0 \quad \text{for } k > 1; \text{ otherwise undefined}$$

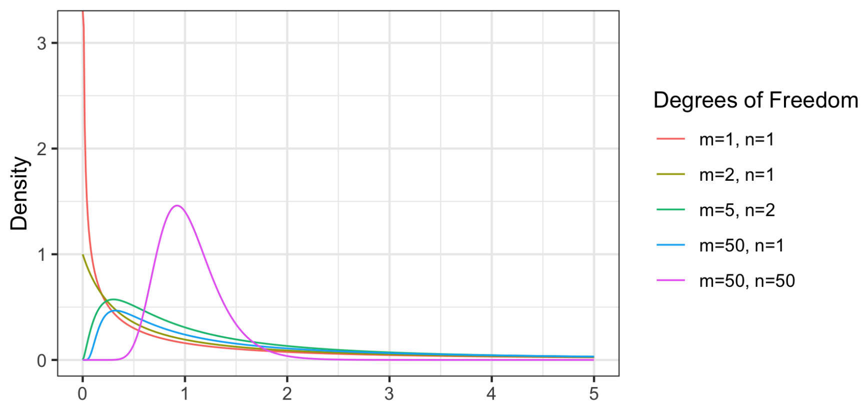
$$\text{var}(x|k) = \begin{cases} \frac{k}{k-2} & k > 2 \\ \infty & 1 < k \leq 2 \\ \text{undefined} & \text{otherwise} \end{cases}$$

4.10 F Distribution

If U and V are independent χ^2 random variables with m and n degrees of freedom,

$$W = \frac{U/m}{V/n} \sim F_{m,n}$$

or in words, the statistic W follows an F distribution with m and n degrees of freedom. I'm not writing out the functional form of the F distribution here because it's too awful-looking, but graphically it looks like this:



Note that if $T \sim t_k$, then $T^2 \sim F_{1,k}$. The F-distribution plays an important role in a class of statistical analysis techniques called **ANalysis Of VAriance**, or **ANOVA**.

Question 4.7

For each of the following experimental conditions, which distribution (from those listed above) provides the best model for how the data $x^{(1)}, \ldots, x^{(n)}$ are generated?

(a) You are observing several patients' skin in a clinical study to see how long it takes them to develop a rash. You take a picture each day. Let $x^{(i)}$ be the number of days of *no rash* before the rash occurs.

Patient ID (i)	$x^{(i)}$
1	4
2	1
3	0
4	2
5	2
6	4
7	3
8	1
9	0
10	1

(b) Same situation as above except that instead of taking a picture each day, the patient texts you at the moment he/she observes a rash. The data look like this, where $x^{(i)}$ is the time (in days) at which patient i develops a rash:

Patient ID (i)	$x^{(i)}$
1	2.25
2	3.43
3	0.68
4	0.04
5	3.78
6	5.65
7	2.88
8	3.88
9	2.83
10	1.87

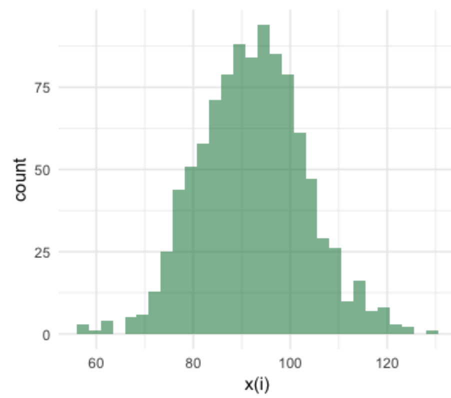
(c) Imagine you are Ladislaus Bortkiewicz, and you are modeling the number of persons killed by mule or horse kicks in the Prussian army per year. You have data from the late 1800s over the course of 20 years. Let $x^{(i)}$ be the number of people killed in year i.

Year (i)	$x^{(i)}$	Year (i)	$x^{(i)}$
1	8	11	9
2	10	12	7
3	5	13	10
4	3	14	12
5	10	15	8
6	8	16	7
7	7	17	8
8	2	18	8
9	6	19	10
10	11	20	7

(d) Every year, 10 scientists go to the same geographic area (same Lyme prevalence) and they each collect 40 ticks. They test each tick for Lyme disease and record the number of ticks that have Lyme. Let $x^{(i)}$ be the number of ticks with Lyme in the ith scientist's bunch.

Scientist ID (i)	$x^{(i)}$
1	8
2	9
3	14
4	15
5	12
6	7
7	6
8	8
9	8
10	14

(e) You have waist circumference data on 1045 men aged 70 and above (see Dey's 2002 paper in the Journal of the American Geriatric Society). It looks like this:



count
75
50
25
0
60
80
100
120
x(i)

Chapter 5

The Basics of Maximum Likelihood Estimation

Beneath our discussions of classification, regression, and probability distributions in Chapters 2, 3, and 4 lies the tricky problem of **model fitting**. We've seen what classification and regression models look like, but we still haven't addressed how to fit these models using training data.

Linear and logistic regression models are fit using a technique called **maximum likelihood (ML) estimation**, in which the model parameters are adjusted to maximize the joint probability of the observed data, or likelihood, given the model.

For example, consider the five different datasets from Question 4.7. In each case, you have some data and an assumption about which probability distribution the data are drawn from. The job of maximum likelihood estimation is to use the data to identify the correct distributional parameters, such as μ and σ (in the case of the normal distribution) or λ (in the case of the Poisson distribution). This process is a type of **statistical inference**.

5.1 The Likelihood and Log-Likelihood

Let $p(x|\theta)$ be the probability distribution that governs our data. Here, θ stands in for all of the parameters we want to fit.

If we draw independent[1] samples from $p(x|\theta)$, the **joint probability density function** for all n observations is:

$$p(x^{(1)}, x^{(2)}, \ldots, x^{(n)}|\theta) = \prod_{i=1}^{n} p(x^{(i)}|\theta).$$

Since the data are known but the parameter(s) θ are unknown, we will view this quantity as a function of θ. This is just a change in notation:

$$\mathcal{L}(\theta) = \prod_{i=1}^{n} p(x^{(i)}|\theta).$$

The higher the joint probability of the data (the more "likely" the data are) given θ, the higher the value of this function. We call $\mathcal{L}(\theta)$ the **likelihood**[2]. Frequently we will want to work with the logarithm of the likelihood, which we call the **log-likelihood**, because it has some nice properties, including allowing us to manipulate sums instead of products[3]:

$$\log \mathcal{L}(\theta) = \sum_{i=1}^{n} \log p(x^{(i)}|\theta).$$

In maximum likelihood estimation, we seek to find the θ for which the likelihood (or log-likelihood) is maximized. We do this by taking derivatives

[1] Independent sampling just means that the values of different samples do not depend on each other. When the samples are drawn independently from the same distribution, their joint probability density is just the product of the individual probability densities (which are all the same).

[2] The distributions we have discussed so far are from a broad family of probability distributions called the **exponential family**. One of the properties of this family is that the log-likelihood is concave. Practically speaking, this means that if we maximize the log-likelihood by setting derivatives equal to zero, we are guaranteed to (a) get only one solution, and (b) find a maximum (not a minimum or an inflection point).

[3] Note that if the function $f(z)$ has a maximum at z', the function $\log f(z)$ will also have a maximum at z', because the logarithmic function is monotonically increasing. So we will get the same parameter estimate(s) either way.

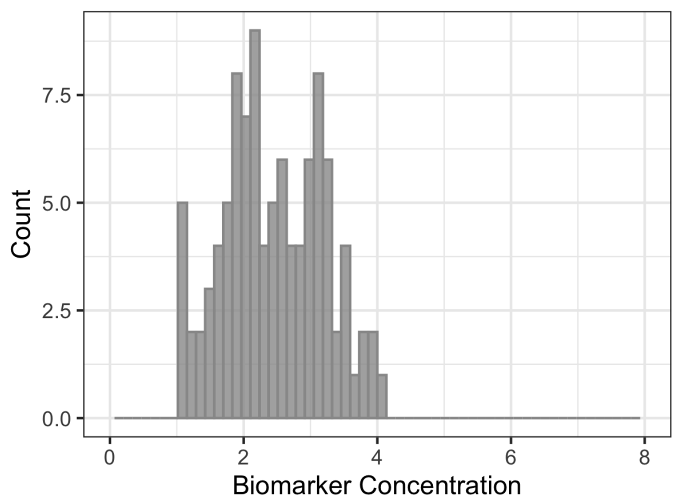
of the log-likelihood with respect to the various parameters and setting them equal to zero. The best-fit parameter estimates obtained in this way are called the **maximum likelihood estimates (MLEs)**.

Question 5.1

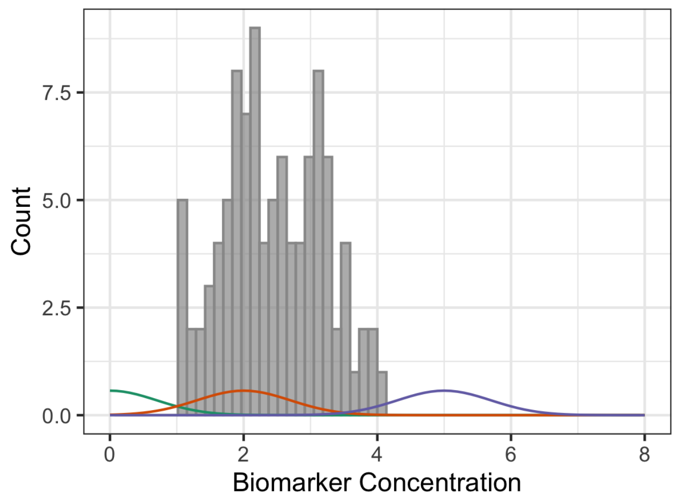
What are some reasons why we might want to fit data to a probability distribution?

5.2 Example: Fitting Data to a Normal Distribution

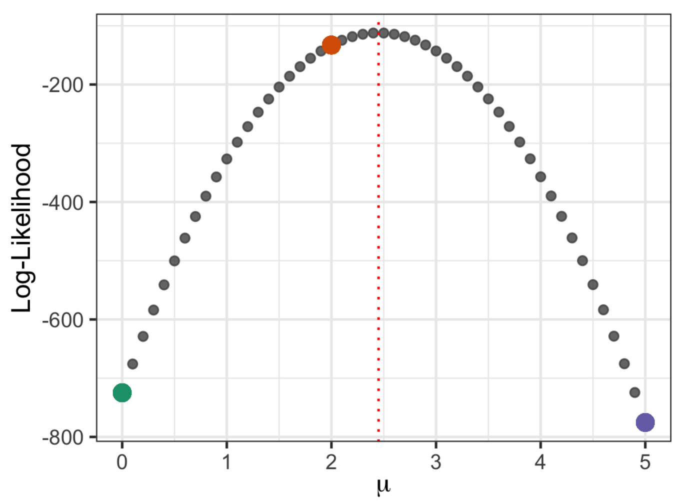
Imagine you have some data from a lab test that measures the concentration of a particular biomarker. You have data from 100 different subjects. A histogram of the raw data looks like this:



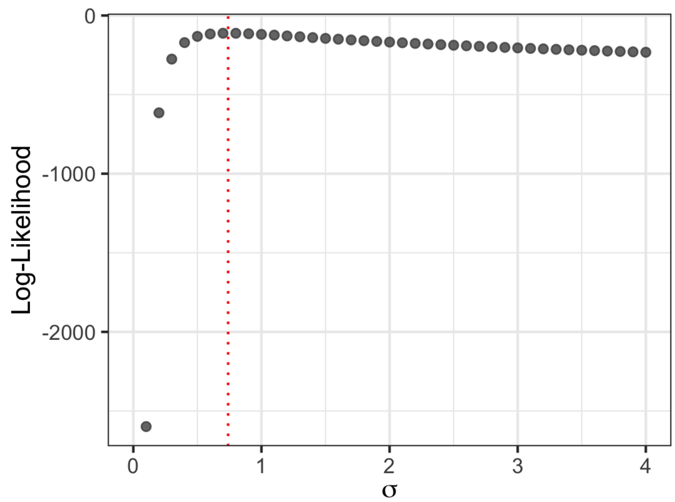
You want to find the normal distribution that best describes these data so you can create a reference distribution for this lab test. To do this, think about trying out several distributions with different values of μ and σ and choosing the one that maximizes the log-likelihood. For example, here are three different normal distributions with different values of μ and $\sigma = 0.7$:



Here is what happens to the log-likelihood as you vary μ. The log-likelihoods of the three distributions shown in the plot above are shown as dots with their corresponding colors, and the maximum likelihood estimate is shown as a vertical dotted line.



Now, here's what happens to the log-likelihood when we vary σ, keeping μ fixed at its maximum likelihood estimate from the graph above. Again, the maximum likelihood estimate is shown as a vertical dotted line.



For the record, I simulated these data from a normal distribution with $\mu = 2.5$ and $\sigma = 0.75$. The maximum likelihood estimates obtained from this dataset are $\hat{\mu} = 2.45$ and $\hat{\sigma} = 0.74$.

5.3 Analytical Calculations of MLEs

In some simple cases, the MLEs can be calculated analytically. We will now go through a bunch of examples of how to find the MLEs of the probability distributions we saw in Chapter 4.

5.3.1 Bernoulli Distribution

The Bernoulli distribution is described in Section 4.3. Our goal is to find the parameter, μ, of this distribution, given some observed data, $x^{(1)}, \ldots, x^{(n)}$. The data will consist of a list of 1s and 0s, since Bernoulli random variables can only take the values 0 or 1.

To find $\hat{\mu}$, our MLE for μ, we first write down the log-likelihood:

$$\begin{aligned}
\log \mathcal{L}(\mu) &= \sum_{i=1}^{n} \log p(x^{(i)}|\mu) \\
&= \sum_{i=1}^{n} \log \left(\mu^{x^{(i)}} (1-\mu)^{1-x^{(i)}} \right) \\
&= \sum_{i=1}^{n} \left[x^{(i)} \log(\mu) + (1 - x^{(i)}) \log(1-\mu) \right]
\end{aligned}$$

Then we take the derivative of the log-likelihood with respect to μ:

$$\frac{d}{d\mu} \log \mathcal{L}(\mu) = \sum_{i=1}^{n} \left[\frac{x^{(i)}}{\mu} - \frac{1-x^{(i)}}{1-\mu} \right]$$

The MLE of μ will occur when the likelihood is maximized, which happens when the first derivative equals zero. So to solve for $\hat{\mu}$, we set the derivative equal to zero and rearrange:

$$\begin{aligned}
\sum_{i=1}^{n} \left[\frac{x^{(i)}}{\hat{\mu}} - \frac{1-x^{(i)}}{1-\hat{\mu}} \right] = 0 \implies & (1-\hat{\mu}) \sum_{i=1}^{n} x^{(i)} = \hat{\mu} \sum_{i=1}^{n} (1 - x^{(i)}) \\
\implies & \boxed{\hat{\mu} = \frac{1}{n} \sum_{i=1}^{n} x^{(i)}}
\end{aligned}$$

We see that the MLE, $\hat{\mu}$, is simply the sum of our data – i.e. the number of data points where the outcome is 1 – divided by the total number of observations.

This makes sense: if you want to know the probability that a coin will come up heads, a good way to estimate it is to flip the coin a bunch of times and calculate the fraction of observations in which the coin comes up heads.

5.3.2 Binomial Distribution

The binomial distribution is described in Section 4.4. We will make one notational change from that section, which is to call the number of Bernoulli trials m instead of n, since we are using n to refer to the number of data samples. To keep things simple, we will assume that m is a known quantity.

As before, we first write down the log-likelihood:

$$\begin{aligned}
\log \mathcal{L}(\mu) &= \sum_{i=1}^{n} \log p(x^{(i)}|m, \mu) \\
&= \sum_{i=1}^{n} \log \left[\binom{m}{x} \mu^{x} (1-\mu)^{m-x} \right] \\
&= \sum_{i=1}^{n} \left[\log(m!) - \log(x!) - \log((m-x)!) + x^{(i)} \log(\mu) + (m - x^{(i)}) \log(1-\mu) \right]
\end{aligned}$$

Then we take the derivative of the log-likelihood with respect to μ:

$$\frac{d}{d\mu} \log \mathcal{L}(\mu) = \sum_{i=1}^{n} \left[\frac{x^{(i)}}{\mu} - \frac{m - x^{(i)}}{1-\mu} \right]$$

We set this equal to zero and solve for $\hat{\mu}$ (the maximum likelihood estimate of μ):

$$\begin{aligned}
\sum_{i=1}^{n} \left[\frac{x^{(i)}}{\hat{\mu}} - \frac{m - x^{(i)}}{1-\hat{\mu}} \right] = 0 &\implies (1-\hat{\mu}) \sum_{i=1}^{n} x^{(i)} = \hat{\mu} \sum_{i=1}^{n} (m - x^{(i)}) \\
&\implies \boxed{\hat{\mu} = \frac{1}{nm} \sum_{i=1}^{n} x^{(i)}}
\end{aligned}$$

Question 5.2

Interpret the MLE for the parameter, μ, of a binomial distribution, assuming fixed m (number of trials). Does the MLE for μ make intuitive sense to you? Think through a few of your examples from Question 4.3.

5.3.3 Normal Distribution

The normal distribution is described in Section 4.2. We will follow the same procedure as in the previous two sections, except that now we have two parameters to solve for, μ and σ, instead of one. First, we write down the log-likelihood:

$$
\begin{aligned}
\log \mathcal{L}(\mu, \sigma) &= \sum_{i=1}^{n} \log p(x^{(i)}|\mu, \sigma) \\
&= \sum_{i=1}^{n} \log \left(\frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x^{(i)}-\mu)^2}{2\sigma^2}} \right) \\
&= -\frac{n}{2}\log(2\pi) - \frac{n}{2}\log\sigma^2 - \frac{1}{2\sigma^2}\sum_{i=1}^{n}\left(x^{(i)} - \mu\right)^2
\end{aligned}
$$

To find the MLE for μ, we take the derivative of the log-likelihood with respect to μ:

$$
\frac{\partial}{\partial\mu} \log \mathcal{L}(\mu, \sigma) = \frac{1}{\sigma^2}\sum_{i=1}^{n}\left(x^{(i)} - \mu\right)
$$

We set this equal to zero and solve for $\hat{\mu}$ (the maximum likelihood estimate of μ):

$$
\frac{1}{\sigma^2}\sum_{i=1}^{n}\left(x^{(i)} - \mu\right) = 0 \implies \boxed{\hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} x^{(i)}}
$$

To find the MLE for σ, we then take the derivative of the log-likelihood with respect to σ:

$$
\frac{\partial}{\partial\sigma} \log \mathcal{L}(\mu, \sigma) = -\frac{n}{\sigma} + \frac{1}{\sigma^3}\sum_{i=1}^{n}\left(x^{(i)} - \mu\right)^2
$$

We set this equal to zero and solve for $\hat{\sigma}$ (the maximum likelihood estimate of σ)[4]. Note that the answer depends on our previously calculated MLE for μ:

$$
-\frac{n}{\hat{\sigma}} + \frac{1}{\hat{\sigma}^3}\sum_{i=1}^{n}\left(x^{(i)} - \mu\right)^2 = 0 \implies \boxed{\hat{\sigma} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(x^{(i)} - \hat{\mu}\right)^2}}
$$

[4]One detail: it turns out this estimate is biased because it depends on the MLE for μ. An unbiased version has $n-1$ in the denominator instead of n. The effect of this is minimal unless n is small.

Question 5.3

Interpret the MLEs for the parameters, μ and σ, of a normal distribution. Do these results make intuitive sense to you? Think through a few of your examples from Question 4.1.

5.3.4 Poisson Distribution

The Poisson distribution is described in Section 4.5. To find the MLE for λ, its mean, we first (as usual) write down the log-likelihood:

$$\begin{aligned}\log \mathcal{L}(\lambda) &= \sum_{i=1}^{n} \log p(x^{(i)}|\lambda) \\ &= \sum_{i=1}^{n} \log \left(\frac{e^{-\lambda} \lambda^{x^{(i)}}}{x^{(i)}!} \right) \\ &= \sum_{i=1}^{n} \left[-\lambda + x^{(i)} \log(\lambda) - \log(x^{(i)}!) \right]\end{aligned}$$

Now we take the derivative of the log-likelihood with respect to λ:

$$\frac{d}{d\lambda} \log \mathcal{L}(\lambda) = \sum_{i=1}^{n} \left[-1 + \frac{x^{(i)}}{\lambda} \right]$$

We set this equal to zero and solve for $\hat{\lambda}$ (the maximum likelihood estimate of λ):

$$\sum_{i=1}^{n} \left[-1 + \frac{x^{(i)}}{\hat{\lambda}} \right] = 0 \implies \boxed{\hat{\lambda} = \frac{1}{n} \sum_{i=1}^{n} x^{(i)}}$$

Question 5.4

Interpret the MLE for the parameter, λ, of a Poisson distribution. Does this result make intuitive sense to you? Think through a few of your examples from Question 4.4.

5.3.5 Geometric Distribution

The geometric distribution is described in Section 4.6. To find the MLE for μ, we first write down the log-likelihood:

$$\begin{aligned}
\log \mathcal{L}(\mu) &= \sum_{i=1}^{n} \log p(x^{(i)}|\mu) \\
&= \sum_{i=1}^{n} \log\left((1-\mu)^{x^{(i)}}\mu\right) \\
&= \sum_{i=1}^{n} \left[x^{(i)} \log(1-\mu) + \log(\mu)\right]
\end{aligned}$$

Now we take the derivative of the log-likelihood with respect to μ:

$$\frac{d}{d\mu} \log \mathcal{L}(\mu) = \sum_{i=1}^{n} \left[-\frac{x^{(i)}}{1-\mu} + \frac{1}{\mu}\right]$$

We set this equal to zero and solve for $\hat{\mu}$ (the maximum likelihood estimate of μ):

$$\begin{aligned}
\sum_{i=1}^{n} \left[-\frac{x^{(i)}}{1-\hat{\mu}} + \frac{1}{\hat{\mu}}\right] = 0 &\implies \frac{n}{\hat{\mu}} = \frac{1}{1-\hat{\mu}} \sum_{i=1}^{n} x^{(i)} \\
&\implies \boxed{\hat{\mu} = \frac{n}{\sum_{i=1}^{n}(x^{(i)}+1)}}
\end{aligned}$$

Question 5.5

Interpret the MLE for the parameter, μ, of a geometric distribution. Does this result make intuitive sense to you? Think through a few of your examples from Question 4.5.

5.3.6 Exponential Distribution

The exponential distribution is described in Section 4.7. To find the MLE for λ, we first write down the log-likelihood:

$$\begin{aligned}
\log \mathcal{L}(\lambda) &= \sum_{i=1}^{n} \log p(x^{(i)}|\lambda) \\
&= \sum_{i=1}^{n} \log \left(\lambda e^{-\lambda x^{(i)}}\right) \\
&= \sum_{i=1}^{n} \left[\log(\lambda) - \lambda x^{(i)}\right]
\end{aligned}$$

Now we take the derivative of the log-likelihood with respect to λ:

$$\frac{d}{d\lambda} \log \mathcal{L}(\lambda) = \sum_{i=1}^{n} \left[\frac{1}{\lambda} - x^{(i)}\right]$$

We set this equal to zero and solve for $\hat{\lambda}$ (the maximum likelihood estimate of λ):

$$\sum_{i=1}^{n} \left[\frac{1}{\hat{\lambda}} - x^{(i)}\right] = 0 \implies \boxed{\hat{\lambda} = \frac{n}{\sum_{i=1}^{n} x^{(i)}}}$$

Question 5.6

Interpret the MLE for the parameter, λ, of an exponential distribution. Does this result make intuitive sense to you? Think through a few of your examples from Question 4.6.

5.4 Summary of MLEs for Common Distributions

The table below contains a summary of the MLEs of various parameters from some common probability distributions.

Distribution	Parameter	ML Estimate	Domain of $x^{(i)}$
Univariate Normal	μ	$\frac{1}{n}\sum_{i=1}^{n} x^{(i)}$	$\mathbb{R}$
	σ	$\frac{1}{n}\sum_{i=1}^{n}\left(x^{(i)} - \hat{\mu}\right)^2$	$\mathbb{R}$
Multivariate Normal	μ	$\frac{1}{n}\sum_{i=1}^{n} x^{(i)}$	$\mathbb{R}^m$
	$\mathbf{\Sigma}$	$\frac{1}{n}\sum_{i=1}^{n}(x^{(i)} - \hat{\mu})(x^{(i)} - \hat{\mu})^T$	$\mathbb{R}^m$
Bernoulli	μ	$\frac{1}{n}\sum_{i=1}^{n} x^{(i)}$	$\{0, 1\}$
Binomial (fixed m)	μ	$\frac{1}{nm}\sum_{i=1}^{n} x^{(i)}$	$\{0, 1, \ldots, m\}$
Poisson	λ	$\frac{1}{n}\sum_{i=1}^{n} x^{(i)}$	$\{0, 1, \ldots\}$
Geometric	μ	$\frac{n}{\sum_{i=1}^{n}(x^{(i)} + 1)}$	$\{0, 1, \ldots\}$
Exponential	λ	$\frac{n}{\sum_{i=1}^{n} x^{(i)}}$	$\mathbb{R}^+$

Question 5.7

In Question 4.7, we examined several examples of experimental conditions and datasets and discussed which probability distribution best modeled each one. Using the formulas above and the actual datasets from Question 4.7, calculate the MLEs for the parameter(s) of your chosen probability distributions.

Chapter 6

Introduction to Hypothesis Testing

Hypothesis testing is a central idea underpinning much of the analysis in the clinical and biomedical research literature[1]. There are multiple approaches to hypothesis testing, but the most common is **null hypothesis testing**, which was developed by the statistician R.A. Fisher. In null hypothesis testing, one creates a model of how the data should look under default conditions and then quantifies the observed data's deviation from that model using a **test statistic**. If the test statistic is large enough, it means there is evidence that the default position is incorrect.

The statisticians Jerzy Neyman and Karl Pearson developed a different approach to hypothesis testing based on the idea of **model comparison**. In their approach, one sets up different models and then quantifies each model's fit to the data; the hypothesis test is used to see whether one model's fit to the data is significantly better than another's. We see the Neyman-Pearson philosophy reflected in techniques such as power calculations and likelihood ratio tests.

Most of the basic hypothesis tests we learn in introductory biostatistics courses (T-tests, chi-squared tests, etc.) follow Fisher's approach. We will

[1]I should state that there is still a lot of controversy around the whole idea of hypothesis testing and whether p-values should be used at all, etc.

focus on null hypothesis testing in this chapter and explore other ideas in subsequent chapters.

6.1 Basic Steps of a Hypothesis Test

1. *State the **null hypothesis***. The null hypothesis corresponds to the default, or baseline, position; for our example, the null hypothesis might be, "The events 'has mutation' and 'has cancer' are statistically independent." The **alternative hypothesis** is the hypothesis that is contrary to the null; for our example, it might be, "The events 'has mutation' and 'has cancer' are not statistically independent."

2. *List statistical assumptions*. All hypothesis tests make one or more assumptions about the data, and it's important to state them clearly. For example, **parametric** hypothesis tests assume the data follow a particular probability distribution under the null, while **nonparametric** tests do not make this assumption.

3. *Decide on an appropriate test and test statistic*. The **test statistic** quantifies the degree of deviation of the observed data from what one would expect under the null hypothesis[2].

4. *Derive the distribution of the test statistic under the null*. This is called the **null distribution**.

5. *Select a significance level under which you'll reject the null*. The **significance level**, usually written as α, is the probability of a type I error. A type I error is committed when one rejects the null even though it is true (false positive result).

6. *Compute the observed value of the test statistic from the data.*

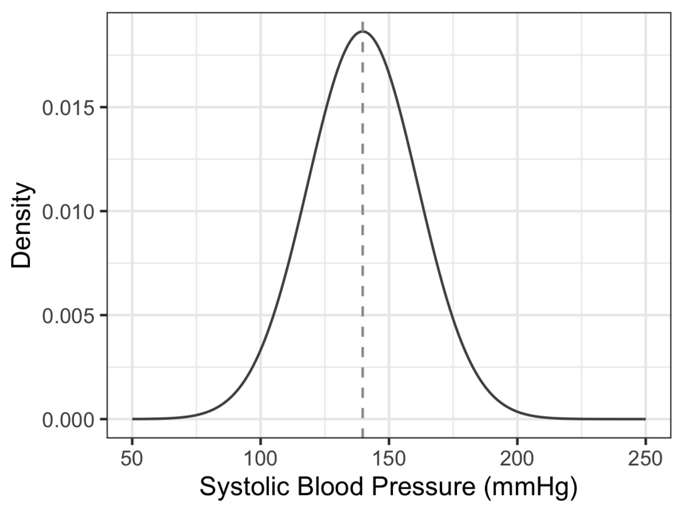
7. *Decide whether or not to reject the null hypothesis.*

[2]Some definitions: A **statistic** is just some quantity that summarizes a set of data, or gives some information about the value of a parameter. A **sufficient statistic** is a statistic that gives the maximum amount of information about a parameter that can possibly be obtained from the sample data.

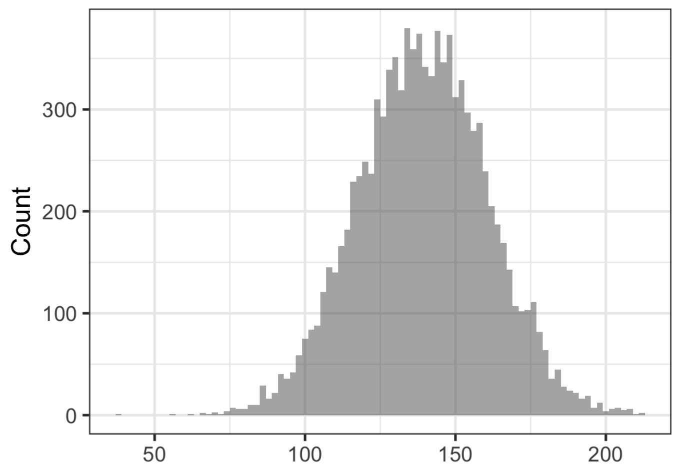
6.2 The Z-Test

A **Z-test** is a hypothesis test for which the null distribution is normal with known mean and standard deviation (i.e. known parameters μ and σ). It is most commonly used to compare the mean of a set of samples, $\overline{x}$, with a known population mean. It also appears in other contexts, such as significance tests of regression coefficients in generalized linear models (Chapter 12).

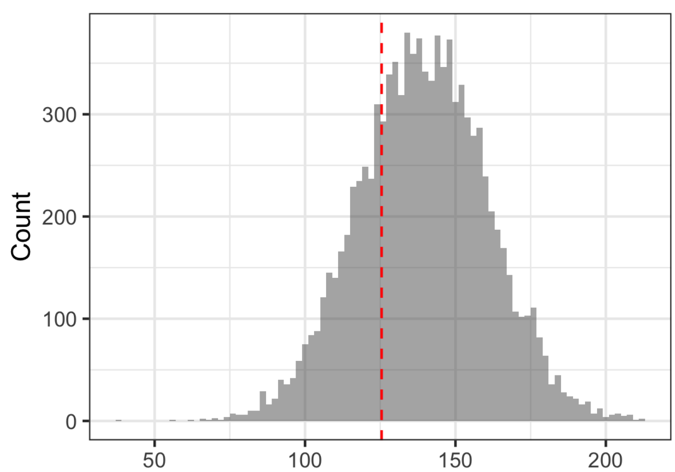
Example: SBP in an Appalachian Town The distribution of systolic blood pressure (SBP) among Caucasian males ages 55-64 in the United States is roughly normal with mean 139.75 mmHg and standard deviation 21.40 mmHg (Source: Int. J. Epidemiol. 2: 294-301, 1973). The following graph shows a normal distribution with those parameters.



Here is a histogram of 10,000 data samples drawn independently from that distribution (i.e., what we would expect if we sampled the SBPs of 10,000 men from the United States at large):

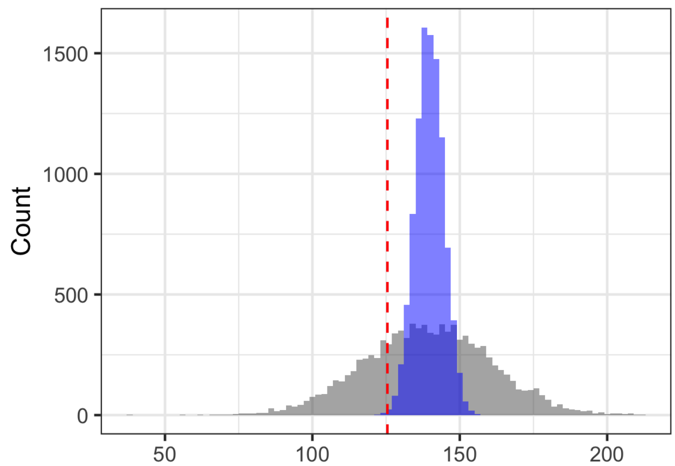


Now, assume some researchers find a small community in rural Appalachia and measure the SBP of 20 Caucasian males ages 55-64 there. Their mean SBP is 125.45 mmHg, illustrated by the red dashed line in the graph below.



At first glance, this may not appear that unusual. After all, the red line is sort of near the center of the gray distribution, right? This analysis is flawed, however, because our 125.45 mmHg value isn't for one man - it's an average over 20 men. The distribution of the **sample mean**, $\bar{x}$, is different from that of each individual sample.

To see this, imagine taking 20 samples from the gray distribution, taking their mean, and recording that value. Now repeat that process 10,000 times. If you do that, you get the **distribution of the sample mean**, which is skinnier than the gray distribution:



It turns out that the distribution of the sample mean will have the same mean, μ_0, as the population distribution, but its standard deviation will be $\sigma/\sqrt{n}$, where n is the number of samples over which the mean is taken.

Question 6.1

If $n = 1$, what is the standard deviation of the sample mean? If $n = \infty$, what is the standard deviation of the sample mean?

Question 6.2

The sample mean for our 20 sampled Appalachian men is shown as a vertical red dashed line in the figure above. Now that you know what the distribution of the sample mean looks like, do you think the observation from your Appalachian town is "weird"?

Let's conduct a hypothesis test to evaluate whether we have evidence that the mean SBP among men in this town is different from that of the general U.S. population.

1. *State the **null hypothesis***. Here the null hypothesis is going to be our default position: that there is no difference. Let μ_c be the true mean SBP for men in the community and μ_0 be the mean for the general population.

$$H_0 : \mu_c = \mu_0$$
$$H_a : \mu_c \neq \mu_0$$

2. *List statistical assumptions*. We make two assumptions. First, we assume that the SBPs of the different men in the sample are statistically independent. Second, we assume that under the null, SBP will follow a normal distribution with mean 139.75 and standard deviation 21.40, the same as the general population of men aged 55-64.

3. *Decide on an appropriate test and test statistic*. Our test statistic in this case is going to be the **Z-statistic**, which measures the deviation of the sample mean from the population mean in units of the standard deviation of the sample mean, $\sigma/\sqrt{n}$:

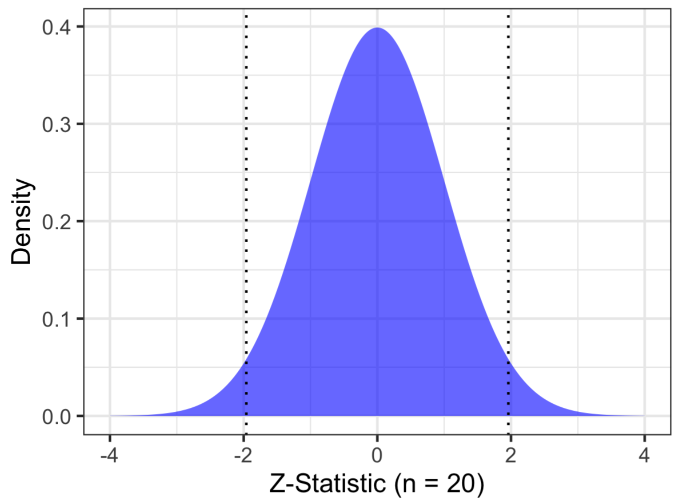
$$Z = \frac{\overline{x} - \mu_0}{\sigma/\sqrt{n}} \qquad \text{where} \qquad \overline{x} = \frac{1}{n}\sum_{i=1}^{n} x^{(i)}$$

   In our case, $n = 20$ because $\overline{x}$, our sample mean, is an average of 20 samples.

4. *Derive the distribution of the test statistic under the null*. The Z-statistic follows a **standard normal** distribution under the null, which is a normal distribution with $\mu = 0$ and $\sigma = 1$. To see this, remember that the distribution of $\overline{x}$ under the null is $\mathcal{N}(\mu_0, \sigma/\sqrt{n})$. When you calculate the Z-statistic, you shift that distribution by a distance μ_0 so it is centered at zero, then adjust its width (standard deviation) to 1.0 by dividing by $\sigma/\sqrt{n}$.

5. *Select a significance level under which you'll reject the null*. For the purposes of this example, we will choose $\alpha = 0.05$ (5% chance of a type I error).

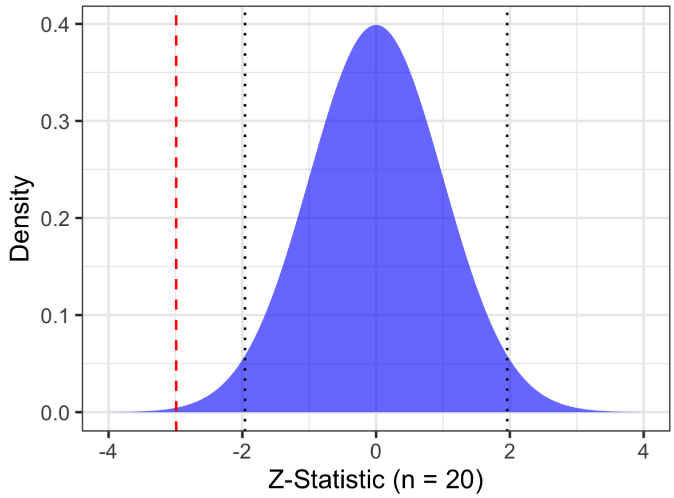
The null distribution of the Z-statistic is shown below. The vertical dotted black lines are situated at the **critical values** that produce $\alpha = 0.05$ (the area under the null distribution that is outside those lines is 0.05).



6. *Compute the observed value of the test statistic from the data.* The observed value of the test statistic is:

$$Z = \frac{\overline{x} - \mu_0}{\sigma / \sqrt{n}} = \frac{125.45 - 139.75}{21.40 / \sqrt{20}} = -2.99.$$

7. *Decide whether or not to reject the null hypothesis.* The value of our test statistic falls outside the region contained by the critical values (the **acceptance region**), so we reject the null at this value of α.



Question 6.3

As α gets smaller, are you more or less likely to reject the null for the same value of the test statistic? Hint: What does making α smaller do to the positions of the two black dotted lines in the figure, above?

6.3 Definitions

- **Type I Error:** When a hypothesis test rejects the null even though the null is true (also called a **false positive**). The type I error rate is usually denoted by α.
- **Type II Error:** When a hypothesis test fails to reject the null even though it is false (also called a **false negative**). The type II error rate is usually denoted by β.
- **P-value:** The probability of obtaining a test statistic at least as extreme as the one that was actually obtained, assuming the null is true. A p-value can be **one-sided** or **two-sided**. The difference lies in the definition of "extreme". In a one-sided test, we find the probability that the test statistic is at least as extreme *in the same direction* as the one we observed. In a two-sided test, we find the probability that the test statistic is at

least as extreme *in either direction* (positive or negative deviation). In most cases, this has the practical effect of doubling the p-value.

- **Power:** The probability that a hypothesis test will reject the null when the null is false (that the test will detect a true effect if the effect is there). Usually denoted $1 - \beta$.

6.4 Pearson's Chi-Squared Test

Imagine you have data on two discrete variables for n different subjects. You want to test whether the value of one covariate is independent of the value of the other. To do this, you can arrange your data in a **contingency table** where the rows and columns correspond to the values of the two variables. **Pearson's chi-squared test** can then be used to assess the independence of row and column values.

Example: Association of Genotype and Disease Imagine you want to test whether a person's genotype at a particular locus is associated with whether or not he/she has Disease X. You find 100 people with the disease and 100 healthy controls ($n = 200$) and genotype them:

	AA	Aa	aa	
X	52	43	5	100
Control	67	27	6	100
	119	70	11	200

Let's conduct a hypothesis test to examine this result.

1. *State the **null hypothesis***. We consider the genotype at this locus, G, to be a random variable (see Chapter 4) with three possible outcomes: AA, Aa, and aa. We likewise consider the patient's disease status, D, to be a

random variable with two possible outcomes: disease or no disease. We state our null hypothesis mathematically as:

$$H_0 : G \perp\!\!\!\perp D$$
$$H_a : G \not\perp\!\!\!\perp D$$

where the symbol $\perp\!\!\!\perp$ refers to statistical independence of G and D. We encountered statistical independence in our discussion of maximum likelihood in Chapter 5. Mathematically, statistical independence means that the joint probability of observing a particular value for G and a particular value for D is simply equal to the product of their individual probabilities:

$$P(G = g, D = d) = P(G = g)P(D = d)$$

Under these conditions, the expected values of the cells of our table are:

Under scenario of independence (E):

	AA	Aa	aa	
X	59.5	35.0	5.5	100
Control	59.5	35.0	5.5	100
	119	70	11	200

For example, consider the cell $G = AA, D = X$. Assuming the total number of patients is fixed at $n = 200$ and G and D are independent, the expected number of people in that cell is:

$$\begin{aligned} P(G = AA, D = X) \cdot n &= \left(\frac{119}{200}\right)\left(\frac{100}{200}\right) \cdot 200 \\ &= \mathbf{59.5} \end{aligned}$$

Our task now is to decide whether our observed table counts are different enough from what we expect under the null to cause us to reject the null.

2. *List statistical assumptions*. We assume that the data are sampled ran-

domly and independently from a fixed population where each member of the population has an equal probability of selection[3].

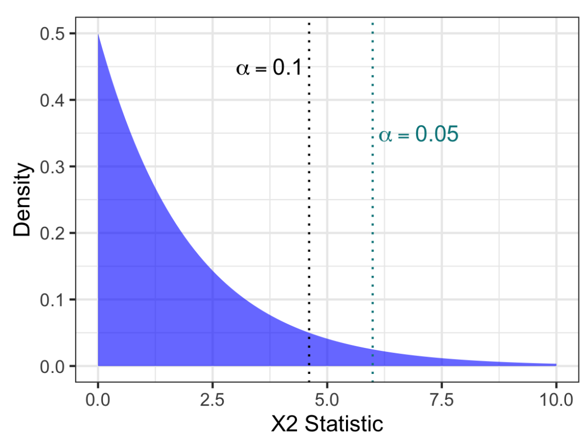
3. *Decide on an appropriate test and test statistic*. The chi-squared test works by calculating expected counts in all $r \times c$ cells of the table (r = number of rows, c = number of columns) and then measuring the data's deviation from those expected counts. The **chi-squared test statistic** has the form

$$X^2 = \sum_{i=1}^{r} \sum_{j=1}^{c} \frac{(O_{ij} - E_{ij})^2}{E_{ij}}$$

where O refers to "observed count" and E to "expected count". The expected counts are those that assume statistical independence of rows and columns (blue table, above).

4. *Derive the distribution of the test statistic under the null*. Under the null, the X^2 test statistic follows a chi-squared distribution (Section 4.8) with $(r-1)(c-1)$ degrees of freedom. In the case of our genotype example, there are $r = 2$ rows and $c = 3$ columns, thus 2 degrees of freedom.

5. *Select a significance level under which you'll reject the null*. The χ^2 distribution with 2 degrees of freedom is shown below. Two vertical lines are shown at different significance levels: $\alpha = 0.05$ and $\alpha = 0.1$.

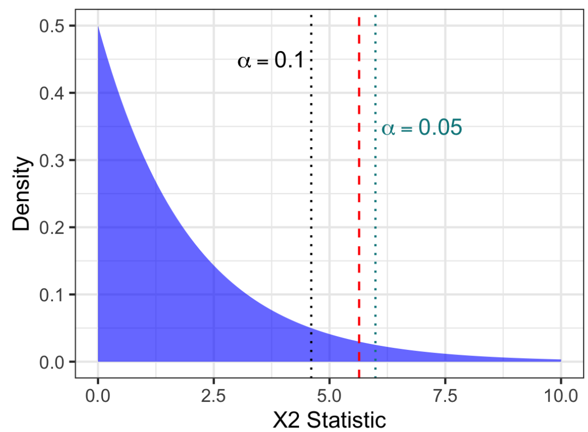


[3] A further assumption of the chi-squared test is that expected counts for each cell must be sufficiently high. A common rule is 5 or more in all cells of a 2×2 table, and 5 or more in 80% of cells in larger tables, but no cells with zero counts.

6. *Compute the observed value of the test statistic from the data.*

Question 6.4

Using the formula in step 4, above, compute the actual value of the chi-squared test statistic for this example. Hint: You should end up with a value that corresponds to the position of the red dashed line in the figure below.



7. *Decide whether or not to reject the null hypothesis.* Based on our calculated value of the test statistic, we will reject the null at $\alpha = 0.1$ and fail to reject the null at $\alpha = 0.05$.

Although it looks much different from the Z-test, the chi-squared test follows the same formalism: defining a null hypothesis, figuring out what the data should look like under the null, quantifying the deviation of the observed data from what's expected using a test statistic, and deciding if that test statistic presents strong enough evidence to cause us to reject the null.

6.5 Student's T-tests

The final example we will look at today is the **T-test**. Like the Z-test, the T-test (actually a family of tests) deals with situations where you have data that are assumed to be normally distributed under the null hypothesis. However, in this scenario, the population standard deviation, σ is not known and must be estimated from the data itself.

6.5.1 One Sample T-test

Assume you have a dataset $x^{(1)}, \ldots, x^{(n)}$, of real numbers that you can plausibly assume are normally distributed. You want to test whether the mean of your data is equal to a fixed value, μ_0. Under the null hypothesis that the means are the same, the test statistic

$$T = \frac{\overline{x} - \mu_0}{s/\sqrt{n}}$$

which we call a "T statistic", follows a T-distribution (Section 4.9) with $n-1$ degrees of freedom[4]. Here $\overline{x}$ refers to the sample mean, and s refers to the **sample standard deviation**:

$$s = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x^{(i)} - \overline{x})^2}$$

Question 6.5

Compare the formula for the sample standard deviation to the maximum likelihood estimate of the parameter, σ, of a normal distribution (Section 5.3.3). What is the same/different? Note in particular the use of $n-1$ in the denominator, rather than n. This arises because the MLE for σ, $\hat{\sigma}$, is a **biased** estimate of the population standard deviation (more on this later). For large n, however, the two are nearly identical.

[4]A one-sample T-test looks a lot like a Z-test. However, because we use s to estimate the population standard deviation from data, we must account for variation in our estimate. It turns out that the sample variance, s^2, follows a chi-squared distribution with $n-1$ degrees of freedom, where n is the sample size. In this case, by the definition of the T-distribution (Section 4.9), the T statistic follows a Student's T-distribution with $n-1$ degrees of freedom. As the number of samples, n, grows, the sample standard deviation approaches the population standard deviation and the T-test becomes a Z-test. But when n is small, the T-test is quite a bit more conservative.

6.5.2 Two Independent Samples, Equal Variance

Assume you have a dataset $x^{(1)}, \ldots, x^{(n)}$ and another dataset $y^{(1)}, \ldots, y^{(m)}$. You assume that both are drawn from normal distributions with equal variance but potentially different means. You want to test whether the means are equal.

The same basic machinery for the one-sample T-test can be deployed in this context with a slightly different test statistic. The test statistic

$$T = \frac{\overline{x} - \overline{y}}{s_p \sqrt{\frac{1}{n} + \frac{1}{m}}}$$

where

$$\begin{aligned}
s_p^2 &= \frac{(n-1)s_x^2 + (m-1)s_y^2}{m+n-2} \\
s_x^2 &= \frac{1}{n-1} \sum_{i=1}^{n} (x^{(i)} - \overline{x})^2 \\
s_y^2 &= \frac{1}{m-1} \sum_{i=1}^{m} (y^{(i)} - \overline{y})^2
\end{aligned}$$

follows a t-distribution with $m + n - 2$ degrees of freedom.

6.5.3 Two Independent Samples, Unequal Variance

Sometimes you have two independent samples but cannot assume the variances are equal. Again, similar machinery can be deployed. In this case, you can use **Welch's T-test**, which uses the test statistic

$$T = \frac{\overline{x} - \overline{y}}{s_{xy}}$$

where

$$s_{xy} = \sqrt{\frac{s_x^2}{n} + \frac{s_y^2}{m}}.$$

This test statistic approximately follows a t-distribution with degrees of freedom given by the Welch-Sattherwaite Equation

$$\text{d.f.} = \frac{\left(\dfrac{s_x^2}{n} + \dfrac{s_y^2}{m}\right)^2}{\dfrac{(s_x^2/n)^2}{n-1} + \dfrac{(s_y^2/m)^2}{m-1}}$$

6.5.4 Matched Pairs

Assume you have a data set of matched pairs. This could be a set of measurements of the same individuals taken at two different points in time, for example, or paired measurements taken from individuals with similar characteristics. You want to test whether the second set of values have changed relative to the first set of values.

To do this, you can use a one-sample T-test on the *differences* of the individual pairs. If no change has occurred, you would expect the mean of those differences to be zero. If we define $x^{(i)}$ as the difference of the paired observations for sample i and $\overline{x}$ as $\frac{1}{n}\sum_{i=1}^{n} x^{(i)}$, the sample mean of those differences, then

$$T = \frac{\overline{x}}{s/\sqrt{n}}$$

follows a T-distribution with $n-1$ degrees of freedom.

Question 6.6

Here are some sample data. They come from a study that looked at the effect of ozone, a component of smog, on the weight gain of rats. (Original source: Biometrika 63: 421-434, 1976, reproduced in Rice's *Mathematical Statistics and Data Analysis*, p. 465.) A group of 22 seventy-day-old rats were kept in an environment containing ozone for 7 days, and their weight gains were recorded. Another group of 23 rats of a similar age were kept in an ozone-free environment for a similar time and their weight gains were also recorded. Here are the data for the control group:

	group	original_weight	weight_gain
1	control	340.8	41.0
2	control	389.1	25.9
3	control	355.2	13.1
4	control	421.8	-16.9
5	control	377.1	15.4
6	control	404.3	22.4
7	control	321.2	29.4
8	control	447.5	26.0
9	control	305.9	38.4
10	control	335.9	21.9
11	control	386.3	27.3
12	control	377.0	17.4
13	control	357.2	27.4
14	control	441.7	17.7
15	control	383.7	21.4
16	control	373.7	26.6
17	control	336.0	24.9
18	control	419.4	18.3
19	control	287.1	28.5
20	control	602.8	21.8
21	control	325.4	19.2
22	control	452.4	26.0
23	control	398.9	22.7
Mean	control	384.4	22.4
St.Dev.	control	65.5	10.8

And here are the data for the ozone group:

	group	original_weight	weight_gain
1	ozone	437.4	10.1
2	ozone	275.9	7.3
3	ozone	296.3	-9.9
4	ozone	295.9	17.9
5	ozone	379.7	6.6
6	ozone	274.1	39.9
7	ozone	360.0	-14.7
8	ozone	331.9	-9.0
9	ozone	531.8	6.1
10	ozone	350.5	14.3
11	ozone	345.7	6.8
12	ozone	268.1	-12.9
13	ozone	339.9	12.1
14	ozone	352.4	-15.9
15	ozone	435.8	44.1
16	ozone	476.9	20.4
17	ozone	462.5	15.5
18	ozone	368.0	28.2
19	ozone	504.3	14.0
20	ozone	188.0	15.7
21	ozone	466.9	54.6
22	ozone	288.8	-9.0
Mean	ozone	365.0	11.0
St.Dev.	ozone	88.6	19.0

(a) Imagine that the population weight distribution of rats is known to be normal with $\mu = 350$ (grams) and unknown σ. How would you test the hypothesis that the mean of the control group is equal to the population mean? How would you test the hypothesis that the mean of the ozone group is equal to the population mean?

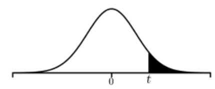
(b) How would you test the hypothesis that the mean original weights of the ozone and control groups are equal? Do not assume equal variance.

(c) How would you test the hypothesis that the mean weight gain in the ozone group is equal to the mean weight gain in the control group? Do not assume equal variance.

(d) How would your approach in part (c) change if you assumed the weight gains in the two groups had equal variance?

Plug in the relevant numbers from the tables above to perform each hypothesis test with $\alpha = 0.05$. The following table of critical values for the T-distribution[a] may help you:

Critical Values for Student's t-Distribution.



	Upper Tail Probability: $\Pr(T > t)$									
df	0.2	0.1	0.05	0.04	0.03	0.025	0.02	0.01	0.005	0.0005
1	1.376	3.078	6.314	7.916	10.579	12.706	15.895	31.821	63.657	636.619
2	1.061	1.886	2.920	3.320	3.896	4.303	4.849	6.965	9.925	31.599
3	0.978	1.638	2.353	2.605	2.951	3.182	3.482	4.541	5.841	12.924
4	0.941	1.533	2.132	2.333	2.601	2.776	2.999	3.747	4.604	8.610
5	0.920	1.476	2.015	2.191	2.422	2.571	2.757	3.365	4.032	6.869
6	0.906	1.440	1.943	2.104	2.313	2.447	2.612	3.143	3.707	5.959
7	0.896	1.415	1.895	2.046	2.241	2.365	2.517	2.998	3.499	5.408
8	0.889	1.397	1.860	2.004	2.189	2.306	2.449	2.896	3.355	5.041
9	0.883	1.383	1.833	1.973	2.150	2.262	2.398	2.821	3.250	4.781
10	0.879	1.372	1.812	1.948	2.120	2.228	2.359	2.764	3.169	4.587
11	0.876	1.363	1.796	1.928	2.096	2.201	2.328	2.718	3.106	4.437
12	0.873	1.356	1.782	1.912	2.076	2.179	2.303	2.681	3.055	4.318
13	0.870	1.350	1.771	1.899	2.060	2.160	2.282	2.650	3.012	4.221
14	0.868	1.345	1.761	1.887	2.046	2.145	2.264	2.624	2.977	4.140
15	0.866	1.341	1.753	1.878	2.034	2.131	2.249	2.602	2.947	4.073
16	0.865	1.337	1.746	1.869	2.024	2.120	2.235	2.583	2.921	4.015
17	0.863	1.333	1.740	1.862	2.015	2.110	2.224	2.567	2.898	3.965
18	0.862	1.330	1.734	1.855	2.007	2.101	2.214	2.552	2.878	3.922
19	0.861	1.328	1.729	1.850	2.000	2.093	2.205	2.539	2.861	3.883
20	0.860	1.325	1.725	1.844	1.994	2.086	2.197	2.528	2.845	3.850
21	0.859	1.323	1.721	1.840	1.988	2.080	2.189	2.518	2.831	3.819
22	0.858	1.321	1.717	1.835	1.983	2.074	2.183	2.508	2.819	3.792
23	0.858	1.319	1.714	1.832	1.978	2.069	2.177	2.500	2.807	3.768

Answers: (a) One-sample T-test of control group original weights vs. null of $\mu_0 = 350$; T-statistic is 2.5165, 22 d.f., two-sided p-value is 0.01964, reject null at $\alpha = 0.05$. One-sample T-test of ozone group original weights vs. null of $\mu_0 = 350$; T-statistic is 0.7961, 21 d.f., two-sided p-value is 0.4349, fail to reject null at $\alpha = 0.05$. (b) Welch's two-sample T-test of control vs. ozone group original weights; T-statistic is 0.8293, d.f is estimated using the Welch-Sattherwaite equation at 38.619, two-sided p-value is 0.4120, fail to reject null at $\alpha = 0.05$. (c) Welch's two-sample T-test of control vs. ozone group weight gains; T-statistic is 2.4629, d.f. is estimated using the Welch-Sattherwaite equation at 32.918, two-sided p-value is 0.01918, reject null at $\alpha = 0.05$. (d) You would use Pearson's two-sample T-test, which assumes equal variances; T-statistic is 2.4919, d.f. is 43, two-sided p-value is 0.01664, reject null at $\alpha = 0.05$.

[a] Borrowed with gratitude from https://www.stat.purdue.edu/ lfindsen/stat503/t-Dist.pdf

Chapter 7

Building a Decision Tree

Decision trees were developed as an alternative to neural networks in the 1970s. We already saw them in Chapters 2 and 3 as examples of supervised learning algorithms. Now we're going to get into a bit more detail about how these trees are learned from data.

Decision trees can be adapted to solve supervised learning problems with different types of outcomes. We will focus on **classification trees** in this chapter, but the same tree building principles can be applied to solve regression problems (see Chapter 3). Similar modifications can also allow us to construct **survival trees**, which model survival outcomes (represented as Kaplan-Meier curves; see Chapter 11). Decision trees can also handle count outcomes, modeling them using Poisson distributions (see Chapter 4).

7.1 Example: The Wisconsin Breast Cancer Dataset

The Wisconsin breast cancer dataset[1] contains information about imaging features of a fine needle aspirate (FNA) of a breast mass from 569 different subjects. Ten real-valued features are computed for each cell nucleus in the image:

(a) radius (mean of distances from center to points on the perimeter)

[1] You can download the dataset from the UCI Machine Learning Repository here: `https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic)`.

(b) texture (standard deviation of gray-scale values)

(c) perimeter

(d) area

(e) smoothness (local variation in radius lengths)

(f) compactness (perimeter^2 / area - 1.0)

(g) concavity (severity of concave portions of the contour)

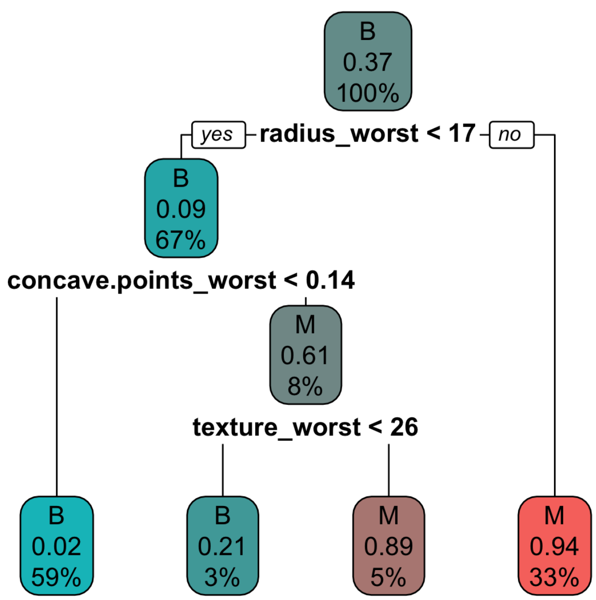
(h) concave points (number of concave portions of the contour)

(i) symmetry

(j) fractal dimension, a statistical index of complexity quantifying how detail in a pattern changes with the scale on which it is measured

The mean, standard deviation, and worst value of each feature are then computed, creating a total of 30 features. Each image is also labeled by its true diagnosis: *B* (for benign) or *M* (for malignant).

Here is a decision tree for this dataset, built using the `rpart` package in R with default parameters:



Question 7.1

What is the most important feature, as identified by the decision tree learning algorithm, for determining whether a breast mass is benign or malignant? What two other features are considered important by the tree? Which features are ignored completely?

Question 7.2

Looking at the decision tree for the Wisconsin Breast Cancer dataset, what do you think the advantages of a decision tree are for this problem over other classification methods, such as logistic regression and KNN?

Although there are several tree building algorithms, all of them are conceptually similar. They all try to reduce "impurity", or "uncertainty", in the outcome variable by intelligently splitting on the predictors. Trees are built recursively from root to leaves. Each internal node of the tree "contains" a subset of the overall dataset (this is why tree building is often called **recursive partitioning**, and why the R package is named `rpart`). At each stage, the algorithm will consider each of the existing leaf nodes and choose the split variable that maximally reduces uncertainty in the outcome. To understand how trees are built computationally, all we need to do is look at the math behind this idea.

7.2 The ID3 Algorithm

One of the earliest approaches to building decision trees was the **ID3 algorithm**[2]. The ID3 Algorithm uses the concepts of entropy and information gain to build trees.

7.2.1 Entropy

Entropy, usually abbreviated H, is a measure of the uncertainty in the value of a random variable. It is the number of bits (on average) required to describe

[2]Quinlan, J.R. "Induction of decision trees", Machine Learning, num. 1, pp. 81- 106, 1986.

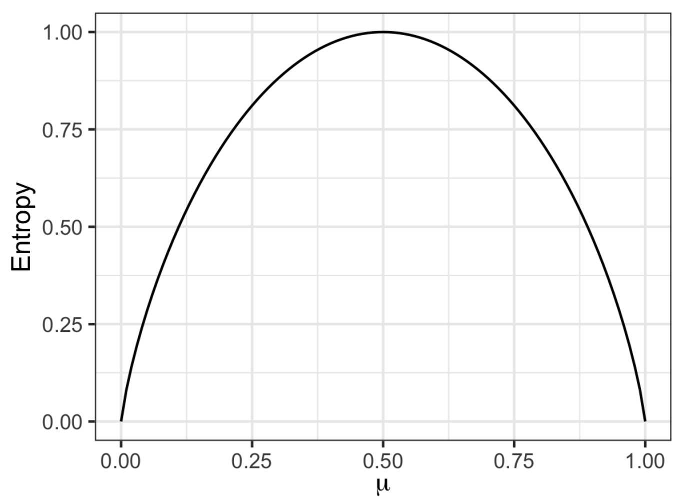
the outcome of the random variable. Here is the formula for the entropy of the discrete probability distribution governing the outcome of a random variable, X:

$$H(X) = -\sum_x P(X = x) \log_2 (P(X = x))$$

For a Bernoulli random variable (see Chapter 4, Section 4.3), there are only two possible outcomes: 0 and 1. The entropy of this random variable is given by:

$$H_{\text{Bernoulli}} = -\mu \log_2(\mu) - (1 - \mu) \log_2(1 - \mu)$$

where μ is the sole parameter of the Bernoulli distribution: the probability of a positive outcome. Here is what the entropy of a Bernoulli distribution looks like as a function of μ:



Question 7.3

At which value(s) of μ are we maximally uncertain about the outcome? At which value(s) of μ are we completely certain about the outcome? This should make intuitive sense if you consider the meaning of μ.

7.2.2 Information Gain

The goal of a decision tree is to reduce uncertainty about the outcome with every split. Let Y be the outcome variable of a training set. Let X be one of the predictors. **Information gain** is defined as:

$$\begin{aligned}\text{Gain}(Y, X) &= H(Y) - \sum_{x \in \text{Values(X)}} \frac{|Y(X = x)|}{|Y|} H(Y(X = x)) \\ &= H(Y) - H(Y|X)\end{aligned}$$

Since entropy is a measure of uncertainty, or impurity, in the outcome, information gain is the reduction in that uncertainty achieved by conditioning on the predictor, X. The tree will choose split variables for which the information gain is maximized.

Question 7.4

Say you have a dataset where the outcome is $Y = [0, 1, 0, 1, 0, 1]$ and there are two predictors: $X = [0, 0, 1, 1, 2, 2]$, and $Z = [1, 2, 1, 2, 1, 2]$. Intuitively, which predictor would make the better splitting variable and why? Calculate $Gain(Y, X)$ and $Gain(Y, Z)$. Which value is higher?

7.2.3 Using Entropy to Build a Decision Tree

By understanding the concepts of entropy and information gain, we arrive naturally at the ID3 Algorithm:

1. Start with a single node: the root of the tree.
2. At each current leaf node:
   (a) Compute the information gain for each feature.
   (b) Split on the one with the highest gain.
3. Return to Step 2. Stop when either the class distributions at all leaf nodes are entirely pure or there are no more variables left to split on.

7.3 Example: Happiness Dataset

Let's build a decision tree for a simple example, using the ID3 algorithm. Assume we surveyed 10 people and asked them whether they were happy or unhappy. We also asked whether they had friends (yes/no), money (poor/enough/rich), and free time (none/some). The data look like this:

Datapoint ID	friends (X_1)	money (X_2)	free time (X_3)	happy (Y)
1	1	1	0	0
2	1	1	1	0
3	0	1	1	0
4	0	0	0	0
5	1	0	0	0
6	0	0	0	0
7	1	2	1	1
8	1	0	1	1
9	0	0	1	1
10	1	0	0	1

$$x_1 = \begin{cases} 0 & \text{no friends} \\ 1 & \text{friends} \end{cases} \qquad x_2 = \begin{cases} 0 & \text{poor} \\ 1 & \text{enough money} \\ 2 & \text{rich} \end{cases} \qquad x_3 = \begin{cases} 0 & \text{no free time} \\ 1 & \text{some free time} \end{cases}$$

Question 7.5

Build a decision tree for this dataset using the ID3 algorithm. To get started, you need to know the entropy of the overall outcome distribution. It is:

$$H(Y) = -\frac{4}{10}\log_2\frac{4}{10} - \frac{6}{10}\log_2\frac{6}{10} = \mathbf{0.971}$$

We will go through the calculations below. As we go, you can start to draw the tree on another page.

(a) Perform the initial split at the tree root to determine which variable to split on first. Update the tree with this information.

$$\frac{|Y(X_1 = 0)|}{|Y|} H[Y(X_1 = 0)] =$$

$$\frac{|Y(X_1 = 1)|}{|Y|} H[Y(X_1 = 1)] =$$

$$\text{Gain}(Y, X_1) =$$

$$\frac{|Y(X_2 = 0)|}{|Y|} H[Y(X_2 = 0)] =$$

$$\frac{|Y(X_2 = 1)|}{|Y|} H[Y(X_2 = 1)] =$$

$$\frac{|Y(X_2 = 2)|}{|Y|} H[Y(X_2 = 2)] =$$

$$\text{Gain}(Y, X_2) =$$

$$\frac{|Y(X_3 = 0)|}{|Y|} H[Y(X_3 = 0)] =$$

$$\frac{|Y(X_3 = 1)|}{|Y|} H[Y(X_3 = 1)] =$$

$$\text{Gain}(Y, X_3) =$$

(b) We see that two of the leaves of our tree are "pure", meaning that all of the training examples that arrive there are of one outcome class. For those two leaves, we're done. However, for the third ($X_2 = 0$, or poor), we need to perform another split. Perform the split at the $X_2 = 0$ node to find which variable to split on next and update the tree with this information.

$$H[Y(X_2 = 0)] =$$

$$\frac{|Y(X_2 = 0, X_1 = 0)|}{|Y(X_2 = 0)|} H[Y(X_2 = 0, X_1 = 0)] =$$

$$\frac{|Y(X_2 = 0, X_1 = 1)|}{|Y(X_2 = 0)|} H[Y(X_2 = 0, X_1 = 1)] =$$

$$\text{Gain}(Y(X_2 = 0), X_1) =$$

$$\frac{|Y(X_2 = 0, X_3 = 0)|}{|Y(X_2 = 0)|} H[Y(X_2 = 0, X_3 = 0)] =$$

$$\frac{|Y(X_2 = 0, X_3 = 1)|}{|Y(X_2 = 0)|} H[Y(X_2 = 0, X_3 = 1)] =$$

$$\text{Gain}(Y(X_2 = 0), X_3) =$$

(c) We need to do one more split on the $X_2 = 0, X_3 = 0$ node. The only variable left to split on is X_1 (friends). Perform this split and add this information to the tree.

7.4 Alternative Splitting Criteria

Information gain is not the only criterion that is used in decision tree algorithms. In fact, the tree built for the Wisconsin Breast Cancer dataset in Section 7.1 used a different criterion, Gini impurity, because it is the default in R's `rpart` package.

The **Gini impurity** measures how often a randomly chosen element from a set would be incorrectly labeled if it were randomly labeled using the distribution of labels in the subset. It sums up the probability of an item with label i being chosen (p_i) multiplied by the probability $\sum_{j \neq i} p_j = 1 - p_i$ that a mistake is made in classifying it. The formula is:

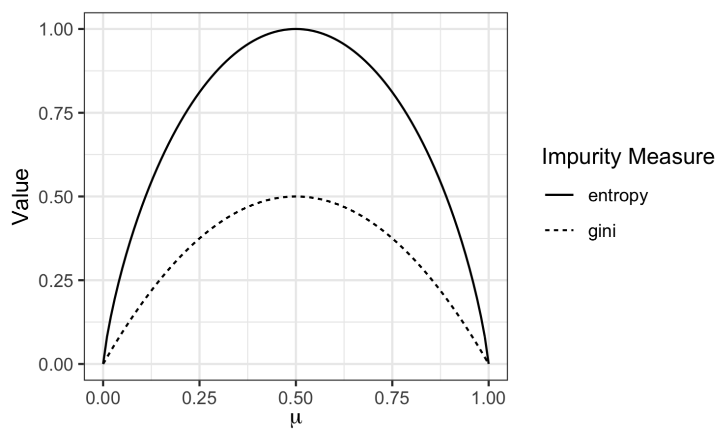
$$G(p) = \sum_i p_i(1 - p_i) = 1 - \sum_i p_i^2$$

which for a Bernoulli distribution is

$$G_{\text{Bernoulli}} = 1 - \mu^2 - (1-\mu)^2.$$

Question 7.6

Plots of the Gini impurity vs. entropy for a Bernoulli distribution are shown below. What do you notice about the value(s) of μ for which each is maximized or minimized?



Although the two metrics are similar and usually yield similar trees, depending on the dataset, the trees can look quite different. The Gini impurity is computationally faster because it does not make use of logarithms as the entropy does. This can make a difference as the number of features grows.

7.5 Decision Tree Regression

So far we've assumed that our outcome is discrete. But what happens if it's numeric? (That is, what if we want to perform regression instead of classification?)

In that case, we use **standard deviation reduction** instead of information gain to decide which variables to split on. The sample standard deviation of an outcome, y, is defined as:

$$S(Y) = \sqrt{\frac{\sum_i (y^{(i)} - \overline{y})}{n-1}}$$

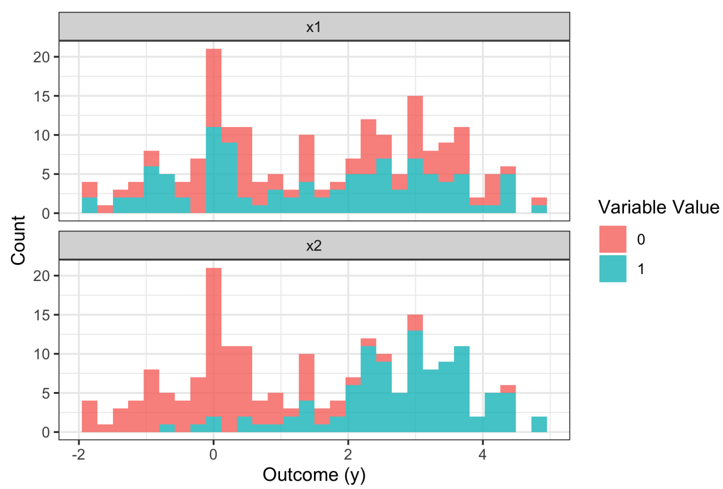
The procedure is identical to the ID3 algorithm except that it uses conditional standard deviation instead of information gain to decide on features. We define

$$S(Y, X) = \sum_x P(X = x)\, S(Y(X = x))$$

and at each current leaf node, we split on the variable where the reduction in standard deviation, $S(Y) - S(Y, X)$, is the highest.

Question 7.7

Imagine you have a dataset with two predictors, x_1 and x_2, each of which is binary (can only be 0 or 1). Here are the distributions of outcome values associated with x_1 and x_2:



Based on the idea of standard deviation reduction, which of these two variables, x_1 or x_2, would make the most sense for a decision tree to split on? What would such a split look like and what would the output value of the tree (the predicted value of y) be for each side of the split?

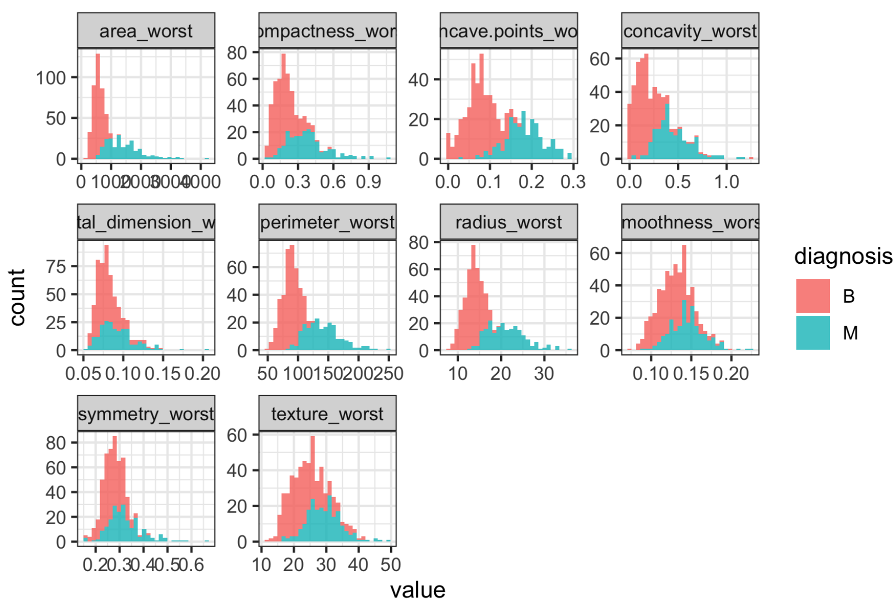
7.6 Numeric Predictors

Although the Happiness Dataset contained only discrete predictors, decision trees can also handle numeric predictors. We've seen this already with the Wisconsin Breast Cancer dataset.

There are different strategies for deciding on an optimal split for a predictor. The most common approach is to consider all possible splits. So for example, if a predictor has values $10, 11, 16, 18, 20$, and 35, the tree building algorithm would consider all $N - 1 = 5$ possible split points. If a dataset has a large number of numeric features or features with lots of different possible values, therefore, it can really slow down the construction of the tree.

Question 7.8

Here are histograms of 10 of the predictors in the Wisconsin Breast Cancer dataset. Only the "worst" variable version of each predictor is shown for clarity. Which variable, and which threshold, appears to show the clearest division of samples into B and M groups? Compare your choice to the first split of the tree in Section 7.1.



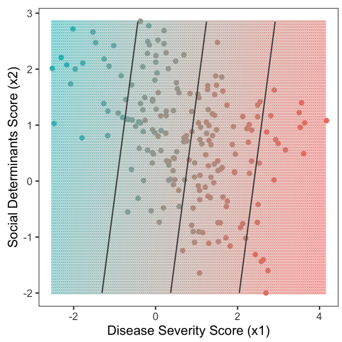
Chapter 8

Interpreting a Linear Regression Model

This chapter is devoted to understanding the structure of linear regression models. We first encountered them in Chapter 3 as "just one example" of a regression model. However, linear regression's overwhelming popularity in the clinical domain means that one cannot do clinical data science without fully understanding these models' structure and how to interpret the output produced by model fitting software.

8.1 Biomarker Example from Chapter 3

In Chapter 3, we saw an example where information about two predictors – a disease severity score (x_1) and a social determinants score (x_2) – was used to predict the level of a disease recurrence biomarker. One of the three supervised learning algorithms we tried was a **linear regression** model (Section 3.2.1). The output from that model is repeated below.




```
Call:
lm(formula = y ~ x1 + x2, data = df)

Residuals:
     Min       1Q   Median       3Q      Max 
-11.9218  -3.1032   0.2891   2.8316  12.5813 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  49.8600     0.5370  92.844  < 2e-16 ***
x1           10.4372     0.2855  36.555  < 2e-16 ***
x2           -1.8824     0.3609  -5.215 4.63e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.769 on 197 degrees of freedom
Multiple R-squared:  0.9026,	Adjusted R-squared:  0.9016 
F-statistic: 912.4 on 2 and 197 DF,  p-value: < 2.2e-16
```

Question 8.1

What are the number of samples, n, and the number of predictors, p, for this dataset?

But what do all these numbers *mean*?

8.2 Understanding the Model Summary

A linear regression model looks like this (see also Chapter 3, Section 3.2.1):

$$y = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p + \varepsilon$$

where we assume that the error, ε, is normally distributed, $\mathcal{N}(0, \sigma)$. Another way of saying this is that we are assuming the outcome, y, is normally distributed with mean $\beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p$ and standard deviation σ.

8.2.1 The Call

The first line of the output is just repeating the call you made to the `lm` function in R to fit the model. The `lm` package fits linear regression models using ordinary least squares. However, linear regression models are also a type of generalized linear model (see Chapter 12) and can be fit using maximum likelihood. In R, if you use the `glm` package with `family = "gaussian"` you should get identical coefficients and error estimates to what you get using the `lm` package.

8.2.2 The Residuals

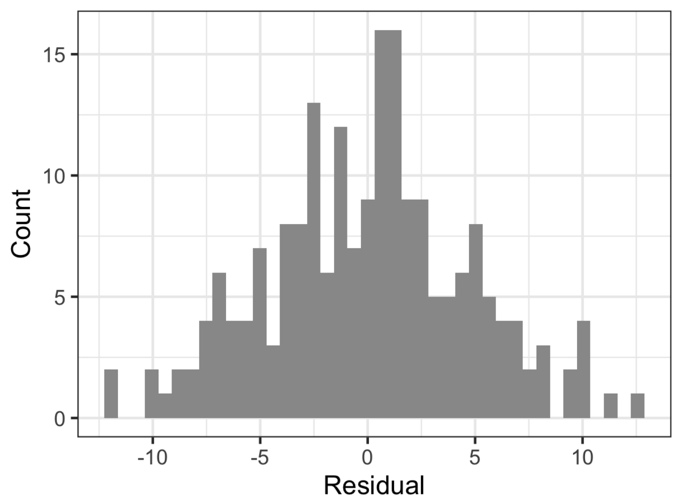
A **residual** is a measure of how much the true outcome value (y) of one datapoint differs from the model's prediction. For a linear regression model, the residual of training point i is:

$$\text{residual}^{(i)} = y^{(i)} - \hat{y}^{(i)}$$

where $\hat{y}^{(i)}$ is the model's prediction:

$$\hat{y}^{(i)} = \beta_0 + \beta_1 x_1^{(i)} + \cdots + \beta_p x_p^{(i)}.$$

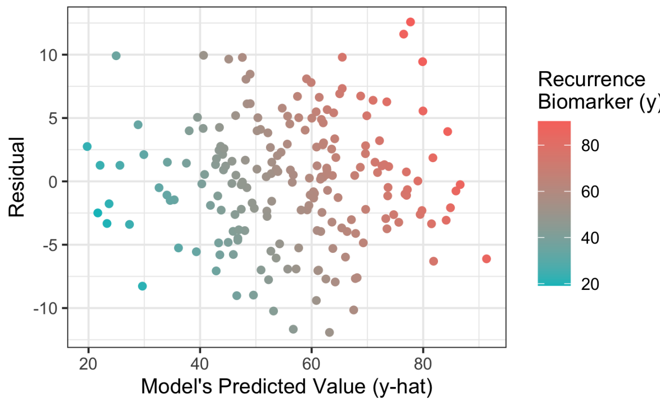
Here is a histogram of the residuals for this model:



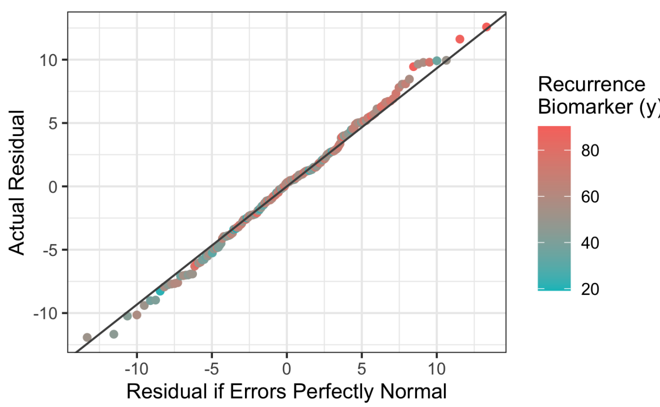
Question 8.2

Estimate the maximum, minimum, and mean residuals from this graph. Do they match what is in the model output?

The residuals play an important role in linear regression models because they are what allow us to estimate σ. They also play an important role in model diagnostics because they enable us to check one of the core model assumptions: the assumption that σ is constant. We can check this assumption by making a plot of the residuals vs. $\hat{y}$:



We can also check the normality of the residuals by making a plot of their values vs. what we would expect if the residuals were perfectly normally distributed with the same mean and standard deviation:

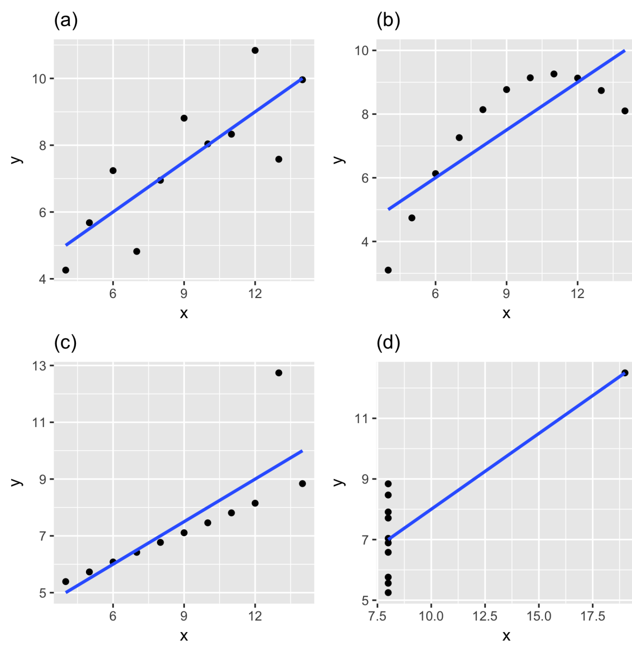


This is not quite a **QQ-plot**, one of the standard diagnostic plots of linear regression, because it's plotting the actual values of the residuals instead of their quantiles. But aside from the axis scales, it's exactly the same. We'll see formal QQ-plots later.

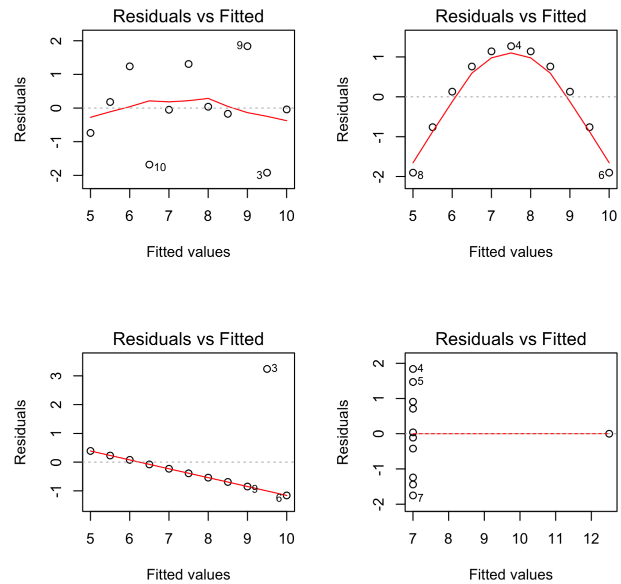
The fact that the points in the second plot lie close to a line is a good indication that the residuals are normally distributed. Another thing to notice from this plot is that the colors of the points (their y values) are unrelated to their residuals.

Question 8.3

The four plots below show a famous dataset called **Anscombe's quartet**. The regression lines produced by fitting a linear regression model to each dataset are identical, but only one dataset actually fulfills the assumptions of a linear regression model.



We can check these assumptions by examining plots of the residuals vs. fitted values of the model (here the "fitted value" of point i means $\hat{y}^{(i)}$).



Which of the four datasets fulfills the assumption of a linear regression model that the error has constant variance? How can you tell?

8.2.3 Coefficients and Standard Errors

Although residuals are important, the parts of the model output that will be scrutinized, reported in papers, etc. are the coefficients, along with their standard errors and hypothesis tests. The coefficients are what create the regression surface, which captures the model's prediction for every point in the feature space (see Section 3.2.1).

There are actually a few different ways to derive the coefficients in a linear regression model. The most common way uses **least squares**, which adjusts the coefficients $\beta_0, \beta_1, \ldots, \beta_p$ until the sum of the squared residuals is minimized:

$$\begin{aligned}\text{SSE} &= \sum_{i=1}^{n}(y^{(i)} - \hat{y}^{(i)})^2 \\ &= \sum_{i=1}^{n}(y^{(i)} - \beta_0 - \beta_1 x_1^{(i)} - \beta_2 x_2^{(i)} - \cdots - \beta_p x_p^{(i)})^2\end{aligned}$$

It turns out that you can find the optimal values of the βs analytically by taking derivatives of this thing and setting them equal to zero. Alas, this requires some matrix multiplication and the taking of matrix inverses, so we will save it for a later chapter. Suffice it to say that the βs are adjusted to minimize the SSE, and the values in the model output are the optimal values.

Question 8.4

Looking at the form of the linear regression model

$$y = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p + \varepsilon$$

what does the value of each of the βs mean? What is β_j telling us about how y varies with the predictor j, all else being equal?

The model's overall estimate of σ, the standard deviation of the error term, is obtained very naturally by averaging over the residuals, taking into account the number of predictors, p:

$$\hat{\sigma}^2 = \frac{1}{n-p-1}\sum_{i=1}^{n}(y^{(i)} - \hat{y}^{(i)})^2.$$

Question 8.5

The sum of the squared residuals for our model is 4480.678. There are $n = 200$ datapoints, and the number of predictors, p, is 2. Calculate $\hat{\sigma}$ for this model. Do you see this number anywhere in the model output? What is it called?

The standard errors of the model coefficients likewise require matrix multiplication to fully understand, but they are related to three factors: (1) the value of $\hat{\sigma}$, (2) the spread of the values of the corresponding covariate about its mean (more spread will decrease the standard error), and (3) correlations between that covariate and other covariates in the model (tighter correlations will increase the standard error).

Question 8.6

The standard errors attempt to capture how much we expect our estimates of the model coefficients to vary if we were to refit the model using a different dataset, provided that the new dataset is similar to (i.e., sampled from the same population distribution as) the one used to fit the model. On average, approximately how much would we expect β_0 (the intercept) to deviate from its fitted value of 49.8600? How much would we expect β_1 and β_2 to deviate from their fitted values?

8.2.4 Hypothesis Tests of Coefficients

Armed with our coefficients and standard errors, we can perform a hypothesis test on each regression coefficient. Our null hypothesis in each case will be that the true value of that coefficient is zero: that is, it has no effect on the outcome. Under the null hypothesis that $\beta_j = 0$ and assuming n, our number of samples, is large enough, the quantity $\hat{\beta}_j / \mathrm{se}(\hat{\beta}_j)$ will be distributed according to a Student's T distribution (see Chapter 4, Section 4.9) with $n - p - 1$ degrees of freedom.

Question 8.7

Sketch the null distributions for the hypothesis tests of our three regression coefficients, β_0, β_1, and β_2. Do you see why the p-values for these tests are so low?

8.2.5 Other Model Output

The software also provides some other output. The quantity **R-squared** is defined as:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y^{(i)} - \hat{y}^{(i)})^2}{\sum_{i=1}^{n}(y^{(i)} - \overline{y})^2}$$

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

or rather, the proportion of total variance in y explained by the model. The **adjusted R-squared** is almost exactly the same except it fixes a source of bias in R^2, namely that R^2 will favor models with more parameters. Adjusted R^2 penalizes models with more parameters. It is defined as:

$$R^2_{\text{adj}} = 1 - \frac{n-1}{n-p-1} \frac{\sum_{i=1}^{n}(y^{(i)} - \hat{y}^{(i)})^2}{\sum_{i=1}^{n}(y^{(i)} - \overline{y})^2}$$

$$= 1 - (1 - R^2)\frac{n-1}{n-p-1}$$

The **F-statistic** is the ratio of two variances: the variance in the outcome, y, that is explained by the model parameters ("sum of squares of regression", or SSR) and the residual, or unexplained variance ("sum of squares of error", or SSE). The corresponding **F-test** tests the null hypothesis that a model with no independent variables (that is, an intercept-only model with $\beta_1 = 0$ and $\beta_2 = 0$) fits the data as well as our model. The F-statistic follows an F distribution with p and $n - p - 1$ degrees of freedom (see Chapter 4, Section 4.10). The p-value reported in the model output is the p-value for this hypothesis test.

8.3 Example: Small Cities Pollution Dataset

The following data come from an early study that examined the possible link between air pollution and mortality. The authors examined 60 cities throughout the United States and recorded the following data:

MORT	Total age-adjusted mortality from all causes, in deaths per 100,000 population
PRECIP	Mean annual precipitation (in inches)
EDUC	Median number of school years completed for persons of age 25 years or older
NONWHITE	Percentage of the 1960 population that is nonwhite
NOX	Relative pollution potential of oxides of nitrogen
SO2	Relative pollution potential of sulfur dioxide

Note: "Relative pollution potential" refers to the product of the tons emitted per day per square kilometer and a factor correcting the SMSA dimensions and exposure.

We want to predict the value of MORT (y) using the predictors PRECIP, EDUC, NONWHITE, NOX, and SO2 (x_1, x_2, x_3, x_4 and x_5). Here is the output for a fitted linear regression model:

```
Call:
lm(formula = MORT ~ PRECIP + EDUC + NONWHITE + NOX + SO2, data = d)

Residuals:
   Min     1Q Median     3Q    Max
-91.38 -18.97  -3.56  16.00  91.83

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 995.63646   91.64099  10.865 3.35e-15 ***
PRECIP        1.40734    0.68914   2.042 0.046032 *
EDUC        -14.80139    7.02747  -2.106 0.039849 *
NONWHITE      3.19909    0.62231   5.141 3.89e-06 ***
NOX          -0.10797    0.13502  -0.800 0.427426
SO2           0.35518    0.09096   3.905 0.000264 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 37.09 on 54 degrees of freedom
Multiple R-squared:  0.6746,    Adjusted R-squared:  0.6444
F-statistic: 22.39 on 5 and 54 DF,  p-value: 4.407e-12
```

Question 8.8

Interpret the values of each of these coefficients. Based on the coefficient values and their standard errors, which predictor(s) do you think have the greatest impact on mortality?

Question 8.9

In this model, is the effect of one predictor (say, `PRECIP`) impacted by the value(s) of any of the other predictor(s)? How does this differ from the other regression algorithms we've seen (KNN and decision trees)? What are the advantages and disadvantages of this choice?

Question 8.10

Is a normal distribution the right distribution to model an outcome of age-adjusted mortality (MORT)? Why or why not? Look back at our discussion of the normal distribution in Chapter 4 if you need a refresher.

Chapter 9

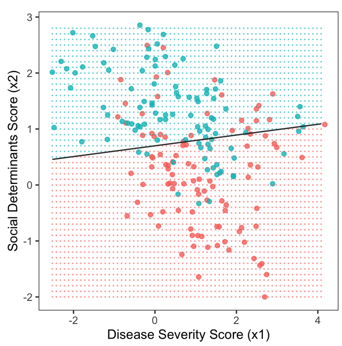
Interpreting a Logistic Regression Model

This chapter is similar to Chapter 8 but focuses on logistic regression models. As we saw in Chapter 8, linear regression models are used in situations where the outcome of a supervised learning problem, y, follows a normal distribution, conditional on the values of the predictors. **Logistic regression** models, in contrast, handle situations where the outcome, y, is binary: either 0 or 1. We first encountered these models as examples of classification algorithms in Chapter 2. Because of their popularity in the clinical domain, it's important to understand how these models are fit and how to interpret the summary output produced by software.

Unfortunately, a full understanding of logistic regression requires knowledge of maximum likelihood estimation. We will, therefore, skip over some of the details until we've had more time to explore this topic.

9.1 ER Readmissions Example from Chapter 2

In Chapter 2, we saw an example where information about two predictors – a disease severity score (x_1) and a social determinants score (x_2) – was used to predict a binary outcome: whether a patient would be readmitted to the ER within 30 days of discharge. We tried three different supervised learning algorithms, one of which was a logistic regression model (Section 2.3.1). The output from that model is repeated below.




```
Call:
glm(formula = y ~ x1 + x2, family = "binomial", data = df)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-1.88232  -0.90614  -0.05965   0.86579   2.28489

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   0.9780     0.2945   3.321 0.000897 ***
x1            0.1344     0.1372   0.980 0.327272
x2           -1.3981     0.2316  -6.035 1.59e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 277.26  on 199  degrees of freedom
Residual deviance: 209.54  on 197  degrees of freedom
AIC: 215.54

Number of Fisher Scoring iterations: 4
```

9.2 Understanding the Model Summary

A logistic regression model looks like this (see also Section 2.3.1):

$$\log \frac{\mu}{1-\mu} = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p \tag{9.1}$$

where μ is the mean of the Bernoulli distribution (see Section 4.3) governing our binary outcome, y; in other words, it is the probability that $y = 1$.

You will note that there is no independent error term here as there was in linear regression. That's because the variance and mean of a Bernoulli distribution are coupled and depend only on μ (again, see Section 4.3).

Question 9.1

In logistic regression, μ itself is not equal the sum of the predictors; instead, the **logit** of μ is their sum. Based on what you know about μ, why is a logistic regression model not of the form

$$\mu = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p?$$

We will explore this further in Chapter 12.

Question 9.2

The decision boundary in logistic regression (see picture above) occurs where the sum of the linear predictors, $\beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p$, is zero. What value of μ does this correspond to? Why does this make sense, intuitively?

9.2.1 The Call

The first line of the output repeats the call you made to the `glm` function in R to fit the model. The `glm` package fits a variety of different generalized linear models using maximum likelihood estimation (see Chapter 5; this will also be discussed in more detail in Chapter 12). The `family = "binomial"` argument tells the function to fit a logistic regression model.

9.2.2 Coefficients and Standard Errors

Logistic regression models, like other GLMs, are fit using maximum likelihood (see Chapter 5 for a brief introduction). We will skip most of the details for now, but you can gain intuition by staring at Equation 9.1. This equation says that the model's predicted value of μ, the probability that the outcome will be positive ($y = 1$), is controlled by the values of the predictors and their coefficients $\beta_0, \ldots, \beta_p$.

By adjusting the values of the βs, the model causes μ to be high in regions of the feature space where $y = 1$ and low where $y = 0$. The values of the βs that do this the best are called the **maximum likelihood estimates**, and they are the coefficients shown in the model output.

As with linear regression, a full understanding of the standard errors requires matrix multiplication. However, they are related to the same factors that drive the standard errors in linear regression: (1) the spread of the values of the corresponding covariate about its mean (more spread will decrease the standard error) and (2) correlations between that covariate and other covariates in the model (tighter correlations will increase the standard error).

Question 9.3

Looking at the form of the logistic regression model

$$\log \frac{\mu}{1-\mu} = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p$$

what does the value of each of the βs mean? What is β_j telling us about how y varies with the predictor j, all else being equal?

Question 9.4

The **odds** of something happening are defined as $\mu/(1-\mu)$, where μ is the probability that the thing occurs. In our example model, we are interested in the odds that $y = 1$ (the patient is readmitted). Does a unit increase in x_1 (disease severity score) increase or decrease the odds that a patient will be readmitted? What about x_2 (social determinants score)?

Question 9.5

What are the odds of readmission for a patient with:

(a) $x_1 = 0.1$ and $x_2 = 0.3$?

(b) $x_1 = 0.1$ and $x_2 = -1.3$?

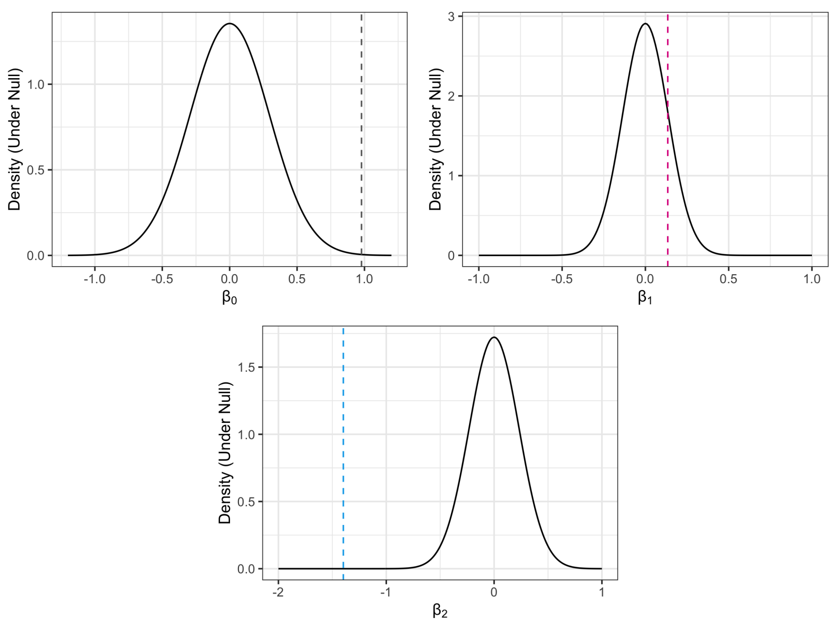
(c) $x_1 = 1.1$ and $x_2 = 0.3$?

9.2.3 Hypothesis Tests of Coefficients

Just as in linear regression, we can use our coefficients and standard errors to perform a hypothesis test on each regression coefficient, β_j, against the null hypothesis that $\beta_j = 0$ (the predictor x_j has no effect on the outcome). In logistic regression, the quantity $\hat{\beta}_j/\mathrm{se}(\hat{\beta}_j)$ will follow a normal distribution under the null.

Question 9.6

Below are the null distributions for the hypothesis tests of our three regression coefficients, β_0, β_1, and β_2. In each graph, the maximum likelihood estimate of the coefficient is shown as a vertical dashed line. Based on these graphs, can you tell why the p-values for β_0 and β_2 are low and the one for β_1 is high? What is the intuition behind this?



9.2.4 Deviance and Deviance Residuals

The **deviance** (called **residual deviance** in the model output) plays a role in GLMs akin to that of the residual standard error in linear regression; it is a measure of the residual variation in the outcome not explained by the model. The **null deviance** is the deviance for a model that only includes an intercept. Under the null hypothesis that all of the βs are zero except the intercept (i.e, a model with no predictors explains the data as well as our model), the difference

$$\text{null deviance} - \text{residual deviance}$$

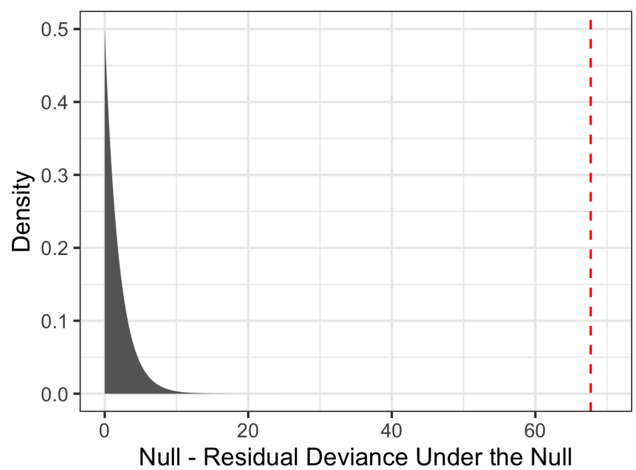
is distributed as χ^2_p, a chi-squared distribution (see Section 4.8) with p degrees of freedom, where p is the number of predictors.

Question 9.7

This test is a hypothesis test of the null hypothesis that a model with no predictors fits our data as well as our model, where goodness of fit is measured by the deviance (lower is better). What is this hypothesis test akin to in the linear regression model output?

Question 9.8

The difference in null and residual deviances in this case is 67.72. It follows a χ^2_2 distribution under the null. A plot of the χ^2_2 distribution and our test statistic is shown below. What do these findings indicate about the p-value of this goodness of fit test and what does it mean?



In the GLM context, there are multiple types of residual (more on this later). **Deviance residuals** quantify the contributions of the individual samples to the deviance. Unfortunately, the output from `glm` is confusing because what `glm` calls a deviance residual in the model summary is actually something called a **working residual**. We will ignore this part of the output until we understand more about the inner workings of GLMs.

9.3 Example: Low Birthweight Dataset

The goal of this study was to identify risk factors associated with giving birth to a low birth weight baby (a baby weighing less than 2500 grams). Infant mortality rates and birth defect rates are very high for low birth weight babies. A woman's behavior during pregnancy (including diet, smoking habits, and receiving prenatal care) can greatly alter the chances of carrying the baby to term and, consequently, of delivering a baby of normal birth weight.

Data were collected on 189 women, 59 of which had low birth weight babies and 130 of which had normal birth weight babies[1].

LOW	Low birth weight (0 = birth weight $\geq$ 2500 g; 1 = birth weight $<$ 2500 g)
AGE	Age of mother in years
LWT	Mother's weight in pounds at last menstrual period
RACE	Race (1 = white, 2 = black, 3 = other)
SMOKE	Smoking status during pregnancy (1 = yes, 0 = no)
PTL	History of premature labor (0 = none, 1 = one, etc.)
HT	History of hypertension (0 = no, 1 = yes)
UI	Presence of uterine irritability (0 = no, 1 = yes)
FTV	Number of physician visits during the first trimester
BWT	Birth weight in grams

We will build a model that predicts the value of `LOW` based on all of the other covariates except, of course, `BWT`. (Why not use `BWT`?)

[1]SOURCE: Hosmer and Lemeshow (2000) *Applied Logistic Regression: Second Edition*. Data were collected at Baystate Medical Center, Springfield, Massachusetts during 1986.

```
Call:
glm(formula = LOW ~ AGE + LWT + RACE + SMOKE + PTL + HT + UI +
    FTV, family = "binomial", data = d)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.8946  -0.8212  -0.5316   0.9818   2.2125

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.480623   1.196888   0.402  0.68801
AGE         -0.029549   0.037031  -0.798  0.42489
LWT         -0.015424   0.006919  -2.229  0.02580 *
RACE2        1.272260   0.527357   2.413  0.01584 *
RACE3        0.880496   0.440778   1.998  0.04576 *
SMOKE        0.938846   0.402147   2.335  0.01957 *
PTL          0.543337   0.345403   1.573  0.11571
HT           1.863303   0.697533   2.671  0.00756 **
UI           0.767648   0.459318   1.671  0.09467 .
FTV          0.065302   0.172394   0.379  0.70484
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 234.67  on 188  degrees of freedom
Residual deviance: 201.28  on 179  degrees of freedom
AIC: 221.28

Number of Fisher Scoring iterations: 4
```

Question 9.9

In this model, is the effect of one predictor (say, `AGE`) impacted by the value(s) of any of the other predictor(s)? How does this differ from the other classification algorithms we've seen (KNN and decision trees)? What are the advantages and disadvantages of this choice?

Question 9.10

Comment on how the variable `RACE` enters into the model here. Does this make sense in light of what that variable means and how it potentially interacts with the other study variables?

Question 9.11

Interpret the values of each of these coefficients. Based on the coefficient values and their standard errors, which predictor(s) do you think have the greatest impact on whether or not a woman has a low birthweight baby?

Chapter 10

A Brief Note on Feature Engineering

All machine learning algorithms and statistical models depend on the concept of a **feature**. A feature is some aspect of a dataset that, the model designer believes, represents the data in a way that is relevant to the problem he/she is trying to solve.

Before any algorithm can be applied, therefore, it is necessary to decide how to represent the data: which features to include and how to extract them from the raw data. This task is called **feature engineering**.

10.1 Study Design vs. Feature Engineering

We have seen a large number of features in Chapters 1–9, but we never stopped to consider them. That's because, in many datasets, the features are chosen at the **study design** stage. The analyst (statistician, data scientist, etc.) has no say in what the features look like or which features are included.

This paradigm is changing as data science increasingly focuses on large, observational datasets, like those from electronic medical records (EMRs). In these types of studies, the raw data were not collected for the study itself, but to fulfill some other purpose. The analyst must choose how to build features from the raw data and use them in models.

Question 10.1

The examples in Chapters 2 and 3 used the same two features. What were these features? How were they represented? What are some alternatives to this choice of features?

Question 10.2

In Chapter 7, we looked at the Wisconsin Breast Cancer Dataset, which includes 30 different imaging features relevant to predicting whether a tumor is benign or malignant. How were these features represented? What are some alternatives to this choice?

Question 10.3

In Chapters 8 and 9, we looked at two datasets that were collected for the purposes of answering particular questions. Do you agree with these study designers' choice of features? What other features could potentially have been relevant to answering each research question?

10.2 Turning Data into Numbers

A model is just a tool for learning relationships among sets of numbers. The first step in any data science problem, therefore, is deciding how to represent what is often a large, complex, noisy dataset as a set of numbers.

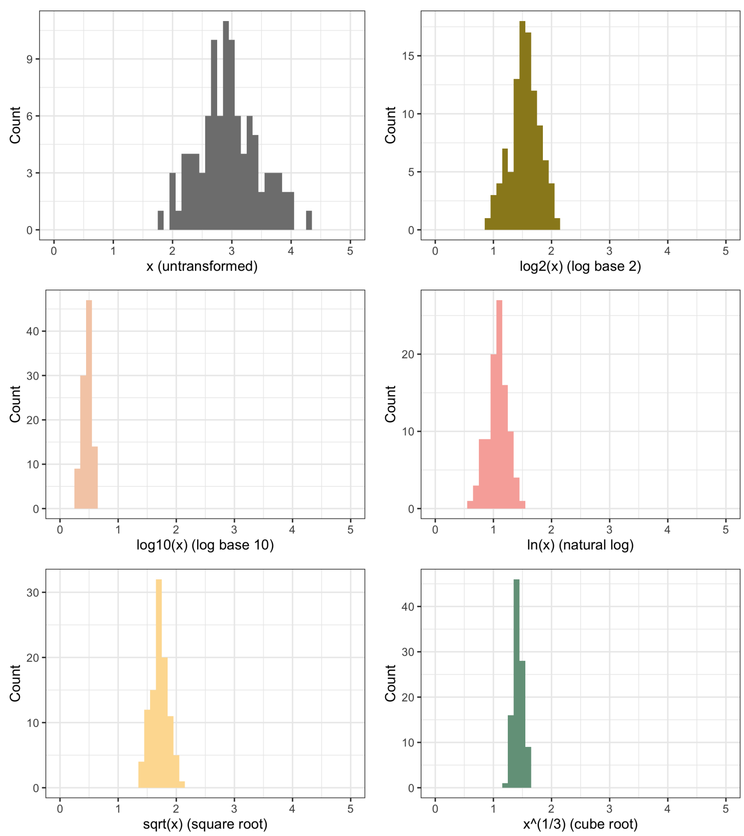
10.2.1 Numbers

Sometimes you get lucky and the feature you need is already a number, such as a vital sign measurement, lab value, or other biomarker. In that case, more often than not, the feature enters into the model as its raw value.

In some cases, you may also choose to apply a **transformation** to the feature before it enters the model. A transformation is simply the application of a deterministic mathematical function that changes the shape of the distribution of the feature. Transformations are often used to improve the interpretability of a model and/or to ensure that the model fulfills the assumptions of the statistical inference method(s) being used (e.g., a hypothesis test).

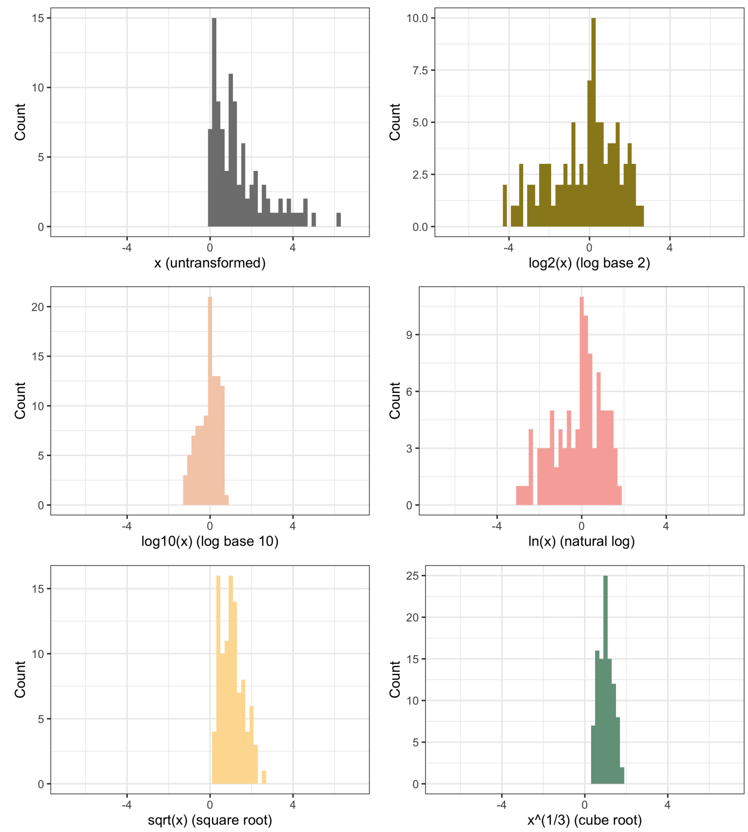
Question 10.4

Here are 100 random samples from a normal distribution with $\mu = 3.0$ and $\sigma = 0.5$ and five different transformations of those samples. What do you notice about the shape and position of the data under the different transformations?



Question 10.5

Here are 100 random samples from an exponential (see Section 4.7) distribution with $\lambda = 0.8$ and the same five transformations of those samples. What do you notice about the shape and position of the data under the different transformations?



Economics, the social sciences, and related disciplines, which are heavily dependent on the use of regression models and hypothesis tests, rely extensively on transformations. In my experience, machine learning folks spend almost no time on them because their primary concern is predictive accuracy, not model interpretation. Machine learning practitioners, however, very frequently **scale and center** their predictors (see footnote in Section 12.5), which is another type of transformation. We will get into more detail on transformations as we continue to learn about regression models.

10.2.2 Binary Variables

For features which are yes/no (e.g., presence/absence of a disease, symptom, physical attribute, etc.) the most common coding scheme is to use "1" for "yes" and "0" for "no". This is useful for interpretation, particularly in regression models. In a linear regression model using this coding scheme, for example, the model coefficient will be the shift in the mean of the normal distribution representing the outcome, y, when the feature is present.

10.2.3 Categories

Categorical features with $k > 2$ categories are generally represented using **indicator variables**. If a feature, x, has k **levels**, we can use $k - 1$ yes/no indicator variables to represent that feature. For example, assume $k = 3$ and the possible levels of our feature, x, are A, B, and C. We set:

$$x_1 = \begin{cases} 1 & \text{if } x = A \\ 0 & \text{otherwise} \end{cases}$$

$$x_2 = \begin{cases} 1 & \text{if } x = B \\ 0 & \text{otherwise} \end{cases}$$

If the value of x is A, $x_1 = 1$ and $x_2 = 0$. If it's B, $x_1 = 0$ and $x_2 = 1$. The value C is called our **reference category** and has $x_1 = 0$ and $x_2 = 0$. In this way, information about all three categories is captured using only two variables. Creating indicator variables is just another way of transforming the value of a feature.

Question 10.6

In Section 9.3, we saw an example of a model that predicts whether or not a mother will give birth to a low birthweight baby. One of the factors considered in that model is the mother's race, which was coded (crudely and probably inaccurately, I might add) as `1 = white, 2 = Black, 3 = other`. You can tell how the feature `RACE` was coded by examining the model output. How many indicator variables were used? Which level of the feature was used as the reference category?

Chapter 11

Survival Data and the Kaplan-Meier Curve

We have already investigated supervised learning models and hypothesis tests in cases where the outcome of interest is a category or number. But what if the outcome is a *time duration*? For example, what if we're comparing the effects of two treatments and our outcome is the time between treatment administration and disease progression?

Data where the outcome is a time duration are very common in clinical data science and are called **time-to-event** data or **survival data**. The field of **survival analysis** develops methods to analyze and interpret such data. We will examine one such method today and many more in subsequent chapters.

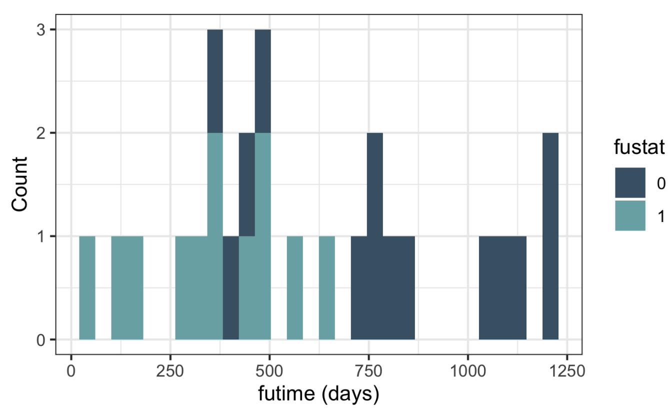
11.1 Example: Ovarian Cancer Survival Dataset

Today we'll examine some data from a study of ovarian cancer[1]. The dataset contains information on 26 women. The variables are:

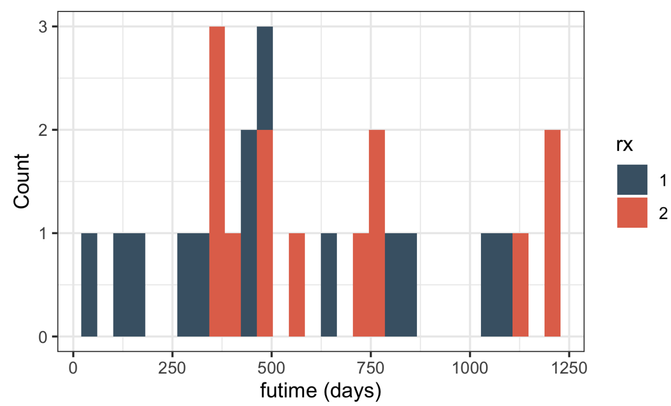
[1]The dataset comes from the `survival` package in R and is labeled `ovarian`. The original study is Edmonson JH *et al*, "Different chemotherapeutic sensitivities and host factors affecting prognosis in advanced ovarian carcinoma versus minimal residual disease", *Cancer Treatment Reports*, 63(2): 241-247; 1979.

- `futime`: The number of days from enrollment in the study until death or censoring, whichever came first
- `fustat`: An indicator of death (1) or censoring (0)
- `age`: The patient's age in years at the time of treatment administration
- `resid.ds`: Residual disease present at the time of treatment administration (1 = no, 2 = yes)
- `rx`: Treatment group (1 = cyclophosphamide, 2 = cyclophosphamide + adriamycin)
- `ecog.ps`: A measure of performance score or functional status at the time of treatment administration, using the Eastern Cooperative Oncology Group's (ECOG) scale. It ranges from 0 (fully functional) to 4 (completely disabled). Level 4 subjects are usually considered too ill to enter a randomized trial such as this. The patients in this dataset are all at Levels 1 and 2.

Here is a histogram of the follow-up times (`futime`) in days, colored according to whether the patient died or was censored (`fustat`):



And here is the same graph colored by treatment group (`rx`):



Now, imagine that we want to study the effect of the treatment group (`rx`) on the outcome of death or no death (1 = death, 0 = no death). We could think of this as a classification problem with only a single feature: treatment group. Unfortunately, this method of analyzing time-dependent data is fraught with problems:

1. How do you choose the time horizon at which to evaluate mortality?
2. How do you handle people who dropped out of the study before that time?

11.2 Definitions

Censoring occurs when the event of interest in a time-to-event analysis is not observed. It is a form of missing data problem (see Chapter **??**) and can be caused by a variety of factors, including inconsistencies in follow-up, the study's ending before all subjects have experienced the event, or a lack of knowledge about when, exactly, the event occurred. The type of censoring represented in the `ovarian` dataset is called **right-censoring**. We will focus on right-censoring today and investigate other types later.

> **Right censoring:** A situation that arises when the event of interest has not occurred by the end of the follow-up period. This may be because (a) the study itself ends, (b) a patient is lost to follow-up

during the study period, or (c) a patient experiences a different event that makes further follow-up impossible[2].

Survival data are generally described using two probabilities, called the survival and hazard.

Survival: Also called the **survival function** or **survival probability** and abbreviated $S(t)$, this is the probability that an individual survives to time t (i.e., does not experience the event by time t).

Hazard: Usually denoted by $h(t)$ or $\lambda(t)$, this is the probability that an individual who has not yet experienced the event at time t experiences it at that exact time. In other words, it is the instantaneous event rate for an individual who has already survived to time t.

We will focus on the survival function now and learn more about the hazard later.

11.3 The Kaplan-Meier Estimator

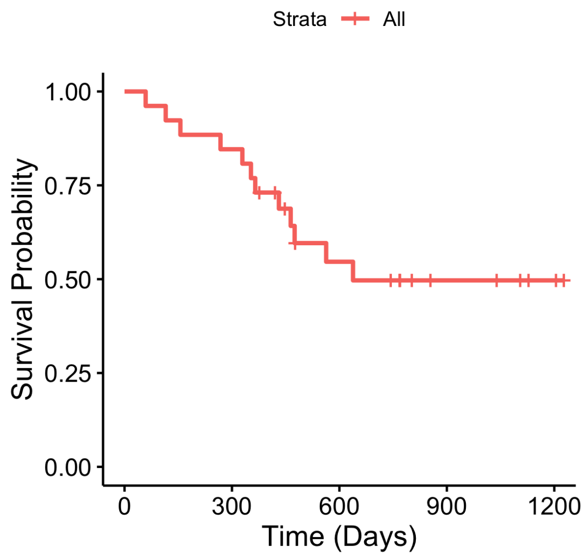
The **Kaplan-Meier estimator** is a nonparametric estimate of the survival function, usually represented graphically by a **Kaplan-Meier curve**[3]. The Kaplan-Meier estimator looks like this:

$$\hat{S}(t) = \prod_{j|t_j \leq t} \frac{n_j - d_j}{n_j}$$

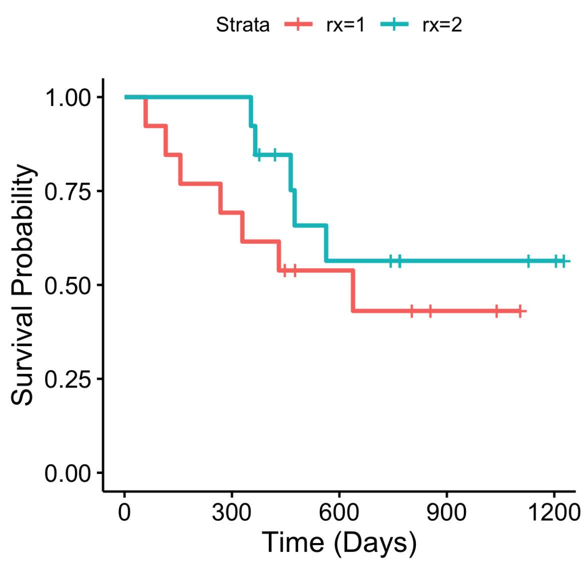
where d_j is the number of subjects who fail at time t_j and n_j is the number of subjects at risk just prior to t_j. Here is a Kaplan-Meier curve for the `ovarian` dataset. The little "+" signs correspond to censoring events.

[2]For more information, please see Clark TG *et al*, "Survival Analysis Part I: Basic Concepts and First Analyses", *British Journal of Cancer*, 89, 232–238; 2003.

[3]It can be shown mathematically that the Kaplan-Meier estimator is the maximum likelihood estimator (see Chapter 5) of the survival function in the case of censoring.



And here are Kaplan-Meier curves for the two treatment groups separately:



Question 11.1

Here are the raw data from treatment group 1 of the `ovarian` dataset. Using these data, fill in the remaining cells of the table below.

	rx	futime	fustat
1	1	59	1
2	1	115	1
3	1	156	1
4	1	268	1
5	1	329	1
6	1	431	1
7	1	448	0
8	1	477	0
9	1	638	1
10	1	803	0
11	1	855	0
12	1	1040	0
13	1	1106	0

j	t_j	n_j	d_j	$\hat{S}(t_j)$	Calculation
0	0	13	0	1.000	$\frac{13-0}{13}$
1	59	13	1	0.923	$\hat{S}(t_0)\left(\frac{13-1}{13}\right)$
2	115	12	1	0.846	$\hat{S}(t_1)\left(\frac{12-1}{12}\right)$
3	156				
4	268				
5	329	9	1	0.615	$\hat{S}(t_4)\left(\frac{9-1}{9}\right)$
6	431	8	1	0.538	$\hat{S}(t_5)\left(\frac{8-1}{8}\right)$
7	448	7	0	0.538	$\hat{S}(t_6)\left(\frac{7-0}{7}\right)$
8	477	6	0	0.538	$\hat{S}(t_7)\left(\frac{6-0}{6}\right)$
9	638	5	1	0.431	$\hat{S}(t_8)\left(\frac{5-1}{5}\right)$
10	803	4	0		
11	855	3	0		
12	1040	2	0		
13	1106	1	0		

Question 11.2

Based solely on the Kaplan-Meier curves for the two treatment groups, which treatment appears to prolong survival more effectively?

11.4 Assumptions of the Kaplan-Meier Estimator

The Kaplan-Meier estimator makes three important assumptions:

1. The probability of censoring is unrelated to the outcome of interest.
2. The survival probabilities are the same for participants recruited at different times during the study (e.g., circumstances that could alter the survival, such as treatments, do not change over calendar time).
3. The events occurred at exactly the times specified.

Question 11.3

What is one way each of these assumptions could be violated?

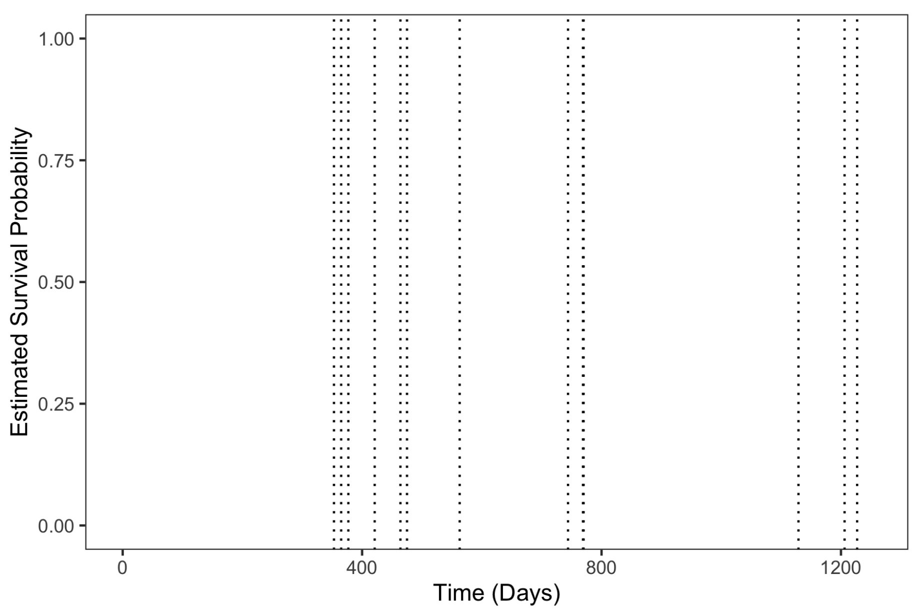
11.5 Comparing Kaplan-Meier Curves

Of course, now the question arises: How do we formally compare two Kaplan-Meier curves? There is a nonparametric hypothesis test for comparing Kaplan-Meier curves called the log-rank test; we will see it in Chapter **??**. There is also an entire family of linear models, called Cox proportional hazards models, that use the Kaplan-Meier curve as their backbone and model the effects of different covariates on this curve. We will see them in Chapter **??**.

Question 11.4

Here are the data for treatment group 2 of the `ovarian` dataset. Perform the calculations of $\hat{S}(t_j)$ for $j = 0, \ldots, 13$, starting with $t_0 = 0$. Draw the Kaplan-Meier curve, adding symbols for the censoring events.

	rx	futime	fustat
1	2	353	1
2	2	365	1
3	2	377	0
4	2	421	0
5	2	464	1
6	2	475	1
7	2	563	1
8	2	744	0
9	2	769	0
10	2	770	0
11	2	1129	0
12	2	1206	0
13	2	1227	0



Chapter 12

Generalized Linear Models

Linear and logistic regression, which we have seen already in Chapters 2, 3, 8, and 9, are members of a broader class of supervised learning models called **generalized linear models (GLMs)**. In GLMs, the outcome variable, y, is assumed to follow a probability distribution of a particular type. For example, in linear regression, y follows a normal distribution. In logistic regression, y is binary $(y \in \{0, 1\})$ and follows a Bernoulli distribution[1]. The expected value, or mean, of the outcome distribution is related to a **linear combination** of the predictors, $\beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p$, via a model-specific **link function**.

GLMs, like maximum likelihood (Chapter 5), are normally considered an advanced topic. However, they provide a nice example of how the same formalism – modeling the response variable using a probability distribution, assuming a certain form for the predictors, optimizing the whole thing using maximum likelihood – can be applied to solve different-looking problems. They are also a good entryway into the sorts of optimization tasks performed by graphical models and deep learning algorithms.

12.1 Model Assumptions

GLMs require us to make several assumptions which affect both our choice of model and our interpretation of model output:

[1]In **grouped** logistic regression, it follows a binomial distribution.

1. We assume that the outcome follows a certain type of distribution (e.g. Bernoulli distribution for a logistic regression model, normal for linear, etc.) conditional on the predictors. This assumption is baked into the model structure. It is, therefore, important to consider whether the outcome distribution you chose actually makes sense for your particular problem. It is generally not advisable to use a linear regression model, for example, when your outcome is a count.

2. We assume that the predictors are fixed and known, and thus have no error associated with their measurements[2].

3. We assume that the predictors enter the model as a linear combination, $\beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p$. This is why GLMs are referred to as "linear models".

4. We assume that the n samples in our dataset are collected independently, so that the errors of the n sample outcomes are uncorrelated[3].

12.2 Notation for the Predictors

As mentioned above, GLMs assume that the predictors enter the model as a linear combination. A linear combination is an expression constructed from a set of terms by multiplying each term by a constant and adding the results. We denote the number of predictors in the model by p and the vector of predictors by x, where

$$x = \begin{bmatrix} 1 \\ x_1 \\ x_2 \\ \vdots \\ x_p \end{bmatrix}$$

and we have included a "1" as the first element to allow for an **intercept**. We write $x^{(i)}$ to denote the vector of predictors associated with the ith training example. The coefficients of the linear combination (i.e. the model parameters

[2]Bayesian versions of these models relax this assumption.

[3]Think back to our formulation of the likelihood in Chapter 5 and how it depended on the samples' being independent and identically distributed, or iid.

we are hoping to learn) are denoted by:

$$\beta = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_p \end{bmatrix}$$

and we often express the linear combination as an **inner product**, or **dot product**, of the two vectors, written as

$$\beta^T x = \beta_0 + \sum_{j=1}^{p} \beta_j x_j.$$

This is just notational shorthand.

Question 12.1

We saw the details of linear and logistic regression models in Chapters 8 and 9 and discussed the limitations of predictors' entering as a linear combination. What are some of those limitations?

Question 12.2

Just to confirm that you understand this notation, write out the form of $\beta^T x$ for a model with (a) one predictor, (b) three predictors. Write both the general form and the form for one training example, $x^{(i)}$.

12.3 Modeling the Outcome

Generalized linear models model the expected value of the outcome, $E[y]$, as a function of this linear combination of predictors.

12.3.1 Linear Regression

In linear regression, we assume that the outcome, y, follows a normal distribution (see Section 4.2), whose mean is controlled by the values of the predictors.

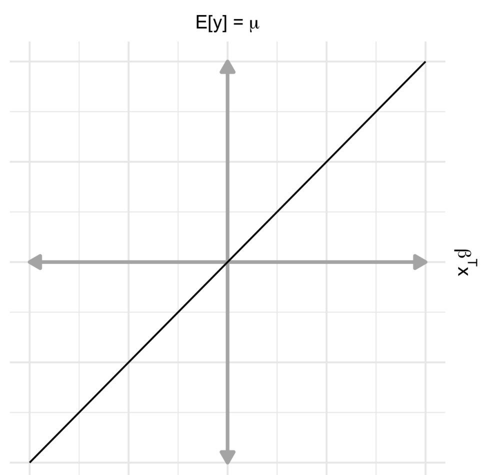
Recall that the normal distribution is a continuous probability distribution with the following properties:

$$p(y) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(y-\mu)^2}{2\sigma^2}} \qquad E[y] = \mu \qquad \text{var}(y) = \sigma^2$$

where $y \in \mathbb{R}$. Its mean, μ, can be any real number. To link μ to the predictors, therefore, we simply set it equal to $\beta^T x$, like so:

$$E[y] = \mu = \beta^T x \tag{12.1}$$

This is called using the **identity link**. The relationship between $E[y] = \mu$ and $\beta^T x$ is shown below.



Question 12.3

In this model, how much does the mean of the outcome distribution, μ, change as you vary each predictor? For example, if you have $p = 3$ predictors, by how much does μ change as the value of x_2 changes by one unit (for example, from 1 to 2)? How much does μ change as the value of x_2 changes from 3 to 4? What about x_1 and x_3?

12.3.2 Logistic Regression

In logistic regression the outcome, y, is either 0 or 1. We model it using the Bernoulli distribution (see Section 4.3), which is a discrete probability distribution with the following properties:

$$p(y) = \mu^y(1-\mu)^{1-y} \qquad E[y] = \mu \qquad \text{var}(y) = \mu(1-\mu)$$

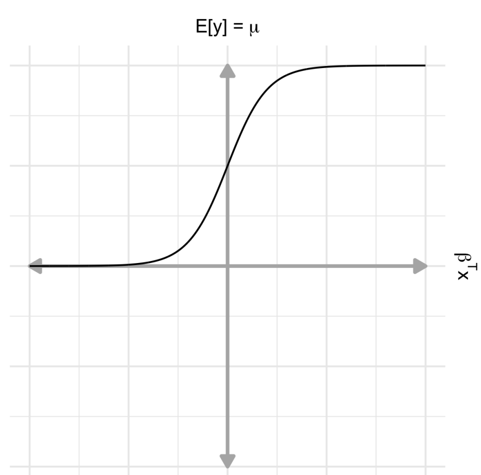
where $y \in \{0, 1\}$. Because μ is a probability, it must be a real number between 0 and 1. No matter how large or small $\beta^T x$ gets, the value of $E[y] = \mu$ cannot be outside this range. We therefore apply the **logistic function**, $f(x) = 1/(1+\exp(-x))$, which has the range $(0, 1)$, to $\beta^T x$ to squash it:

$$E[y] = \mu = \frac{1}{1+\exp\left(-\beta^T x\right)} \tag{12.2}$$

The relationship between $E[y]$ and $\beta^T x$ is shown below. We typically invert the model to write

$$\log \frac{\mu}{1-\mu} = \beta^T x.$$

The function $\log\left(\mu/(1-\mu)\right)$ is called the logit, and we say we use the **logit link**.



Question 12.4

Let's revisit Question 9.1. Now that we've described logistic regression in the framework of GLMs, what more can you say about why the model is not of the form

$$\mu = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p?$$

12.3.3 Poisson Regression

In Poisson regression, the outcome is a count. We model the outcome using a Poisson distribution, which is a discrete probability distribution with the following properties (Section 4.5):

$$p(y) = \frac{e^{-\lambda}\lambda^y}{y!} \qquad E[y] = \lambda \qquad \mathrm{var}(y) = \lambda$$

where $y \in 0, 1, 2, \ldots$. Because λ, the mean of the outcome distribution, is the expected value of a count, it must be a real number greater than or equal to zero. In particular, no matter how small $\beta^T x$ gets, the value of $E[y] = \lambda$ cannot be negative. We therefore exponentiate $\beta^T x$ to ensure that λ is greater than zero:

$$E[y] = \lambda = \exp(\beta^T x) \tag{12.3}$$

The relationship between $E[y]$ and $\beta^T x$ is shown below. We typically invert the model to write

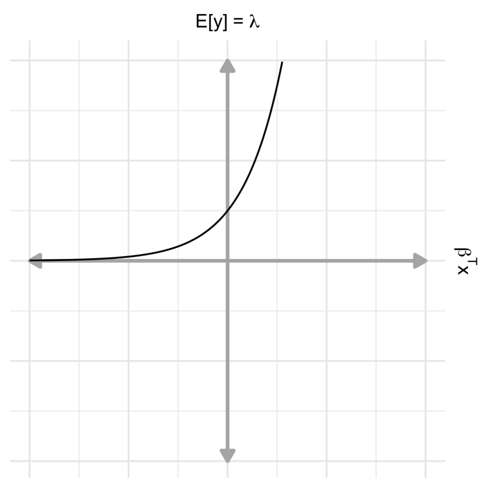
$$\log(\lambda) = \beta^T x$$

which is the standard form of the Poisson regression model. We say these models use the **log link**.

Question 12.5

There are many other generalized linear models. In each case, the mean (expected value) of a probability distribution is related, via a link function, to a linear combination of the predictors.

Knowing this, how would you create a GLM where the outcome follows an exponential distribution (Section 4.7)? Which link would you use?



12.4 Maximum Likelihood for GLMs

GLMs are fit using maximum likelihood estimation (see Chapter 5). A full treatment of MLE for GLMs is outside the scope of these notes, but I've put the start of the calculations for each type of model below. The only difference between these calculations and those in Chapter 5 is that now our parameters of interest, the means of our outcome distributions, are functions of our predictors $x_1, \ldots, x_p$. Our job is to find the coefficients on those predictors, $\beta_0, \ldots, \beta_p$, that provide the best fit between our model and our training data.

12.4.1 Linear Regression

The likelihood for the linear regression model is:

$$\mathcal{L}(\mu^{(1)}, \ldots, \mu^{(n)}, \sigma) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{(y^{(i)} - \mu^{(i)})^2}{2\sigma^2}\right]$$

where we use $\mu^{(i)}$ to represent the model's estimate of the mean of the outcome at the position of training example i. We can use Equation 12.1 to rewrite this as a function of the predictors:

$$\mathcal{L}(\beta, \sigma) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{(y^{(i)} - \beta^T x^{(i)})^2}{2\sigma^2} \right]$$

Taking the log, we obtain the log-likelihood:

$$\log \mathcal{L}(\beta, \sigma) = -\frac{n}{2} \log(2\pi) - \frac{n}{2} \log(\sigma^2) - \frac{1}{2\sigma^2} \sum_{i=1}^{n} \left(y^{(i)} - \beta^T x^{(i)} \right)^2$$

Taking derivatives of the log-likelihood with respect to the βs, we find that we can maximize the likelihood by minimizing the sum-squares: $\sum_{i=1}^{n} \left(y^{(i)} - \beta^T x^{(i)} \right)^2$.

Question 12.6

Take a minute to stare at this result. When most people learn linear regression, they learn that these models are fitted by minimizing the sum of squared residuals (see Chapter 8). Indeed, linear regression models predate GLMs and are typically fit using ordinary least squares, not maximum likelihood. If you fit a linear regression model in R using the `lm` package, you're using OLS. If you use the `glm` package with the argument `family = "gaussian"`, you're using maximum likelihood. However, both methods will produce the same fitted models. Do you see why this is?

12.4.2 Logistic Regression

The likelihood for the logistic regression model is:

$$\mathcal{L}(\mu^{(1)}, \ldots, \mu^{(n)}) = \prod_{i=1}^{n} {\mu^{(i)}}^{y^{(i)}} (1 - \mu^{(i)})^{1 - y^{(i)}}$$

Rewriting this as a function of the predictors, we get:

$$\mathcal{L}(\beta) = \prod_{i=1}^{n} \left(\frac{1}{1 + \exp(-\beta^T x^{(i)})} \right)^{y^{(i)}} \left(\frac{\exp(-\beta^T x^{(i)})}{1 + \exp(-\beta^T x^{(i)})} \right)^{1 - y^{(i)}}$$

Taking the log, we obtain the log-likelihood:

$$\log \mathcal{L}(\beta) = \sum_{i=1}^{n} \left[-y^{(i)} \log \left[1 + \exp(-\beta^T x^{(i)}) \right] + (1 - y^{(i)}) \log \left[1 + \exp(-\beta^T x^{(i)}) \right] \right]$$

Again, we will take derivatives of the log-likelihood with respect to the βs to maximize it. However, we cannot solve for the optimal βs analytically in this case. Numerical optimization methods are used to find the maximum likelihood estimates, $\hat{\beta}_0, \hat{\beta}_1$, etc.

12.4.3 Loglinear (Poisson) Regression

The likelihood for the Poisson regression model is:

$$\mathcal{L}(\lambda^{(1)}, \ldots, \lambda^{(n)}) = \prod_{i=1}^{n} \frac{\lambda^{(i)y^{(i)}} e^{-\lambda^{(i)}}}{y^{(i)}!}$$

Rewriting this as a function of the predictors, we get:

$$\mathcal{L}(\beta) = \prod_{i=1}^{n} \frac{\exp\,(y^{(i)} \beta^T x^{(i)}) e^{-\exp\,(\beta^T x^{(i)})}}{y^{(i)}!}$$

Taking the log, we obtain the log-likelihood:

$$\log \mathcal{L}(\beta) = \sum_{i=1}^{n} \left[y^{(i)} \beta^T x^{(i)} - \exp(\beta^T x^{(i)}) - \log(y^{(i)}!) \right]$$

As with logistic regression, we cannot solve for the optimal βs analytically; numerical optimization methods are used.

Question 12.7

Think of the log-likelihood as measuring the height of a hill. Your data,

$$\left\{ x^{(1)}, \ldots, x^{(n)} \right\}$$

don't change, so we don't care about their effect on the height. What we care about are the parameters, $\beta_0, \ldots, \beta_p$. For each combination of those $p + 1$ parameters, the height changes. We want to find the combination of parameters that puts us at the top of the hill.

The first derivative of the log-likelihood with respect to one of the parameters, β_j, is

$$\frac{\partial \log \mathcal{L}}{\partial \beta_j}$$

and the vector of all of these first derivatives for $\beta_0, \ldots, \beta_j$ is called the **gradient**. Evaluated at a particular set of parameters, the gradient tells you how steep your hill is in the direction of each of your $p+1$ parameters. How could you use this information to maximize the likelihood? You don't need to do any math. Just say how you would do it.

Question 12.8

There are many different numerical optimization algorithms that one can use to maximize the likelihood (i.e., find the top of the hill). One of them is called **Fisher scoring**. Examine the output of the logistic regression models in Chapter 9 and the Poisson regression model shown below in Section 12.6. Where do you see the term "Fisher scoring"? What do you think the term "Fisher scoring iterations" refers to?

12.5 Standard Errors and Hypothesis Tests

Here, once again, is the summary output from a logistic regression model of the ER readmissions example from Chapter 2, reprinted again in Section 9.1:

```
Call:
glm(formula = y ~ x1 + x2, family = "binomial", data = df)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-1.88232  -0.90614  -0.05965   0.86579   2.28489

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   0.9780     0.2945   3.321 0.000897 ***
x1            0.1344     0.1372   0.980 0.327272
x2           -1.3981     0.2316  -6.035 1.59e-09 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 277.26  on 199  degrees of freedom
Residual deviance: 209.54  on 197  degrees of freedom
AIC: 215.54

Number of Fisher Scoring iterations: 4
```

As we discussed in Chapter 9, the magnitudes of the coefficients in these models matter, but they are only important in relation to:

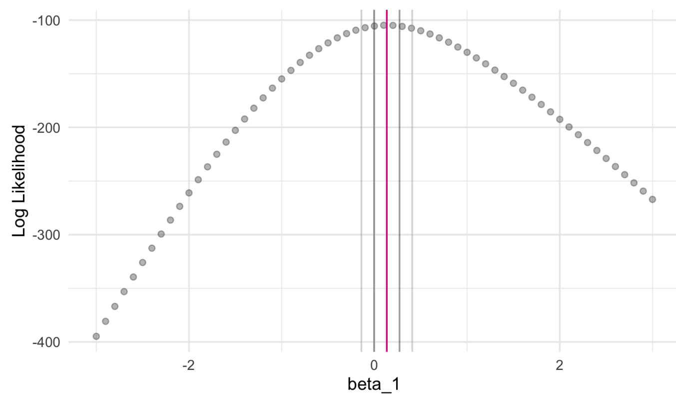
1. The scale on which the predictors are measured.
2. The amount of uncertainty the model has about their values.

For example, if a predictor varies only across a tiny range of values, its model coefficient may be large, since it quantifies the change in the link-function-transformed outcome when the predictor changes by 1.0. However, that doesn't mean that the predictor itself is important to the outcome[4].

Similarly, the model may be highly uncertain about a coefficient's value, owing to factors like a small dataset (small n) or collinearity (correlations) among the predictors. Mathematically, high uncertainty means that the value of the likelihood doesn't change very rapidly as you move away from the maximum likelihood estimate of a coefficient. For example, here is how the

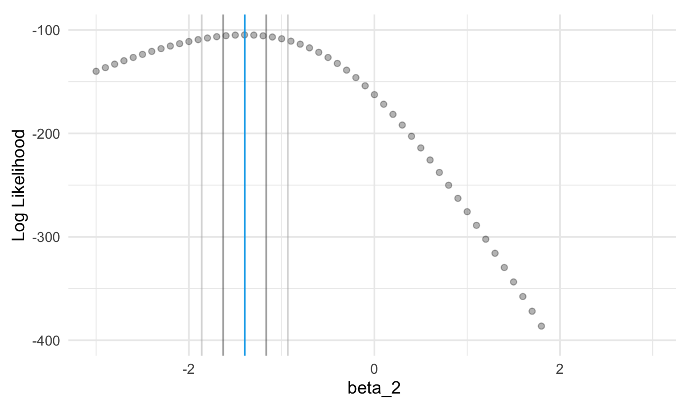
[4]This is one reason many advocate **scaling** and **centering** predictors before fitting a model. Centering means subtracting the mean value of a predictor from all of its individual measurements so that the mean of each centered predictor is zero. Scaling means dividing the values of each predictor by their standard deviation, so that the standard deviation of each predictor is 1.0. This enables the relative magnitudes of the model coefficients to be compared directly.

log-likelihood for the logistic regression example above changes when we vary β_1 (the coefficient of x_1), keeping β_0 (the intercept) and β_2 (the coefficient of x_2) fixed at their MLEs:



The gray vertical lines are related to the **standard error** of the model coefficient, which is in turn related to the “flatness” of the likelihood surface around the MLE. The gray lines are situated at 1 and 2 standard errors away from the MLE in either direction. You can see that in the case of β_1, the gray lines overlap zero. The value zero (no effect) is a plausible estimate of the impact of x_1 on the outcome.

Contrast this with how the log-likelihood varies around the MLE for β_2:



Here the standard error is larger, but the magnitude of the coefficient is also larger, so the range of the gray lines does not overlap zero.

Question 12.9

These findings are reflected in the relative values of the Z-statistic (`z value`) and P-value (`Pr(>z|)|`) in the model output for the two coefficients. With that in mind, let's reconsider Question 9.6. How do these likelihood plots and the null distributions shown in Question 9.6 convey the same information?

12.6 Example: Nesting Horseshoe Crabs Dataset

Let's examine some output from a Poisson regression model, which is a type of GLM with which you may not already be familiar.

These data come from a study of nesting horseshoe crabs. Each of the 173 observed female horseshoe crabs had a male crab resident in her nest. The study investigated factors affecting whether the female crab had any other males, called *satellites*, residing nearby. (Source: Agresti, *Categorical Data Analysis*, Table 4.3. Data courtesy of Jane Brockmann, Zoology Department, University of Florida; study described in *Ethology* **102**: 1-21, 1996.)

`SATELL`	`Number of satellites`
`COLOR`	`Color of the female crab (1 = light medium, 2 = medium, 3 = dark medium, 4 = dark)`
`SPINE`	`Spine condition (1 = both good, 2 = one work or broken, 3 = both worn or broken)`
`WIDTH`	`Carapace width of the female crab (cm)`
`WEIGHT`	`Weight of the female crab (g)`

The GLM output of this model is:

```
Call:
glm(formula = satell ~ color + spine + width + weight, family = "poisson",
    data = d)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.0126  -1.8846  -0.5406   0.9448   4.9602

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.3435447  0.9684204  -0.355  0.72278
color       -0.1849325  0.0665236  -2.780  0.00544 **
spine        0.0399764  0.0568062   0.704  0.48160
width        0.0275251  0.0479425   0.574  0.56588
weight       0.0004725  0.0001649   2.865  0.00417 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 632.79  on 172  degrees of freedom
Residual deviance: 551.85  on 168  degrees of freedom
AIC: 917.15

Number of Fisher Scoring iterations: 6
```

Question 12.10

Comment on how the variables `color` and `spine` are coded here. Does this make sense in light of what those variables mean?

Question 12.11

Interpret the values of each of these coefficients. Based on the coefficient values and their standard errors, which predictor(s) do you think have the greatest impact on the number of male satellites around a nesting female horseshoe crab?

Question 12.12

How could you use a decision tree to model the horseshoe crabs data? What are its advantages and disadvantages relative to Poisson regression (a type of GLM)?

Chapter 13

Model Complexity and the Bias-Variance Tradeoff

In classification, **model complexity** (i.e. the effective number of parameters the model must fit) is typically related to the intricacy and complexity of the decision boundary; the more parameters in the model, the more complex the boundary.

13.1 Goodness of Fit vs. Generalizability

Training vs. test error

13.2 Bias vs. Variance

This figure shows the training and test error for KNN as a function of K for a classification example similar to the one discussed in Chapter 2, as well as the training and test error for a linear model (which doesn't vary with K). You can see that the curves have characteristic shapes that vary with K. It turns out these shapes reflect a general principle for all supervised learning called the **bias-variance tradeoff**.

The bias-variance tradeoff: KNN example. The Bayes error rate, or **ir-**

reducible error, is the probability an instance is misclassified by a classifier that knows the true class probabilities given the predictors. From *Elements of Statistical Learning*, Figure 2.4.

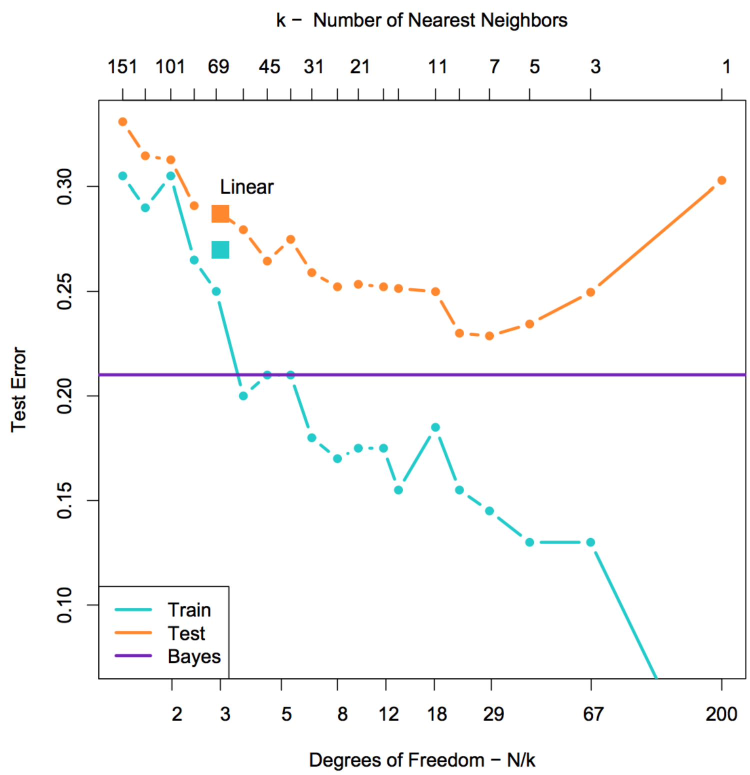
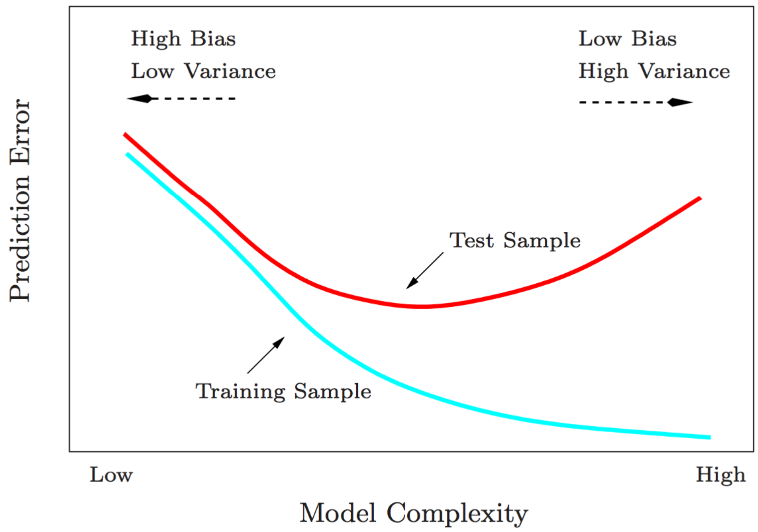
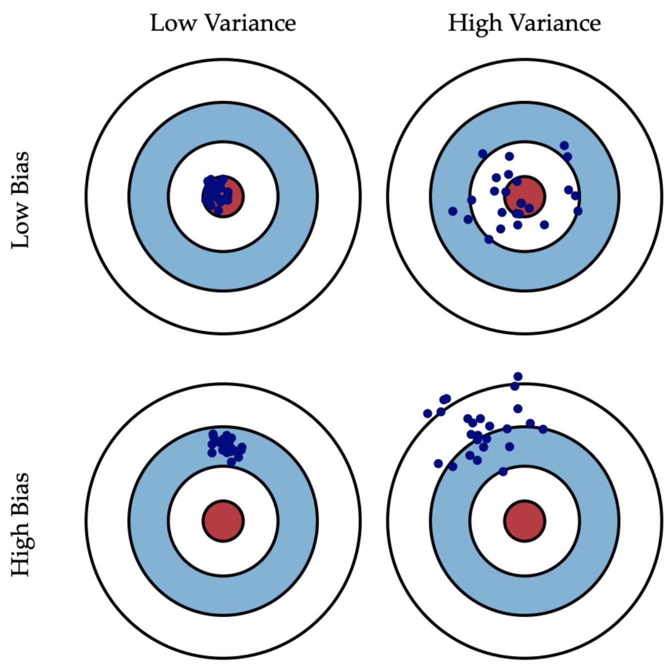


Illustration of training vs. test error as a function of model complexity, as well as the bias-variance tradeoff. From *Elements of Statistical Learning*, Figure 2.11.



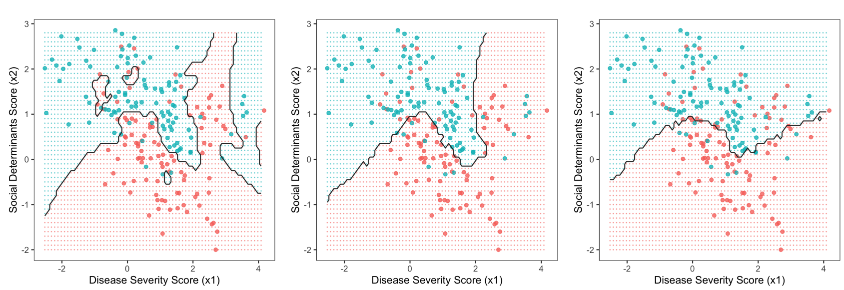
A graphical illustration of the difference between bias and variance. Think of each dot as representing a single test example evaluated under the same model trained on slightly different datasets. The center of the target is the prediction the model should make for that test example. In the case of high bias and low variance, all of the models are off, but they are "wrong in the same way". If you average their predictions, the answer is still way off the mark. In the case of high variance, the models all make very different predictions on the same training example. However, their predictions are off in random directions from the center, so if you average their outputs, you'll get closer to the right answer.



13.3 Overfitting vs. Underfitting

Question 13.1

What are the advantages and disadvantages of KNN with low K (e.g. $K = 3$) vs. high K (e.g. $K = 50$)? The decision boundaries for the previous example with (left to right) $K = 3, 15$, and 50 are shown below.



Question 13.2

We have discussed bias and variance in the context of classification (a yes/no outcome). How would training and test error, overfitting vs. underfitting, etc. be quantified if the outcome was a number, as in a regression problem (Chapter 3)?

Chapter 14

Random Forests DRAFT

A random forest is just a collection (or **ensemble**) of decision trees whose "votes" are uncorrelated. The trees vote to produce a final prediction.

Two details are important to the construction of random forests:

1. Each tree is built using a subset of training examples sampled with replacement from the original training set. This is called **bagging** (bootstrap aggregating). Typically around 2/3 of training examples are used per tree. Note that bagging is a general-purpose procedure that can be used for other models besides random forests.

2. For each split, the tree considers not all m predictor variables, but only a randomly-chosen subset, usually of size approximately $\sqrt{m}$ (for classification problems) or $m/3$ (for regression problems). This keeps you from building the same tree over and over again and ensures that the votes from different trees are uncorrelated.

Here are two bagged samples of size 6 from the dataset in Table **??**.

ID	friends (X_1)	money (X_2)	free time (X_3)	happy (Y)
5	1	0	0	0
4	0	0	0	0
2	1	1	1	0
10	1	0	0	1
8	1	0	1	1
10	1	0	0	1

ID	friends (X_1)	money (X_2)	free time (X_3)	happy (Y)
5	1	0	0	0
6	0	0	0	0
2	1	1	1	0
5	1	0	0	0
9	0	0	1	1
7	1	2	1	1

Question 3.11: Use a random forest to fit the data from the low birth-weight example used in the logistic regression model, above. Use the following commands exactly as shown to ensure it all runs smoothly and you can view the output:

```
library(randomForest)
d <- read.delim("../data/logistic-lowbwt-data.tsv")
d$RACE <- as.factor(d$RACE)  # <- ensure RACE coded as factor
d$LOW <- as.factor(d$LOW)    # <- ensure LOW coded as factor
r <- randomForest(LOW ~ AGE + LWT + RACE + SMOKE + PTL + HT + UI + FTV, data = d, ntree = 100, do.trace = TRUE)
plot(r)
```

The random forest will report a number called the **out-of-bag (OOB)** error as it runs. To calculate OOB error, the trees are allowed to vote on the points that were *not* used in their construction. This provides an ongoing estimate of the generalization error of the algorithm, so you can see if adding more trees is likely to help.

What is the (approximate) overall OOB error? What is it for the positive outcome class only? The negative outcome class only?