1 Software for modeling

Models are mathematical tools that can describe a system and capture relationships in the data given to them. Models can be used for various purposes, including predicting future events, determining if there is a difference between several groups, aiding map-based visualization, discovering novel patterns in the data that could be further investigated, and more. The utility of a model hinges on its ability to be *reductive*. The primary influences in the data can be captured mathematically in a useful way, such as in a relationship that can be expressed as an equation.

Since the beginning of the twenty-first century, mathematical models have become ubiquitous in our daily lives, in both obvious and subtle ways. A typical day for many people might involve checking the weather to see when might be a good time to walk the dog, ordering a product from a website, typing a text message to a friend and having it autocorrected, and checking email. In each of these instances, there is a good chance that some type of model was involved. In some cases, the contribution of the model might be easily perceived ("You might also be interested in purchasing product *X*") while in other cases, the impact could be the absence of something (e.g., spam email). Models are used to choose clothing that a customer might like, to identify a molecule that should be evaluated as a drug candidate, and might even be the mechanism that a nefarious company uses to avoid the discovery of cars that over-pollute. For better or worse, models are here to stay.

> There are two reasons that models permeate our lives today: an abundance of **software** exists to create models and it has become easier to record **data** and make it accessible.

This book focuses largely on software. It is obviously critical that software produces the correct relationships to represent the data. For the most part, determining mathematical correctness is possible, but the reliable creation of appropriate models requires more.

First, it is important that it is easy to operate software in a proper way. The user interface should not be so poorly designed that the user would not know that they used it inappropriately. For example, Baggerly and Coombes (2009) report myriad problems in the data analyses from a high profile computational biology publication. One of the issues was related to how the users were required to add the names of the model inputs. The user interface of the software made it easy to offset the column names of the data from the actual data columns. This resulted in the wrong genes being identified as important for treating cancer patients and eventually contributed to the termination of several clinical trials (Carlson 2012).

If we need high quality models, software must facilitate proper usage. Abrams (2003) describes an interesting principle to guide us:

> The Pit of Success: in stark contrast to a summit, a peak, or a journey across a desert to find victory through many trials and surprises, we want our customers to simply fall into winning practices by using our platform and frameworks.

Data analysis and modeling software should espouse this idea.

The second important aspect of model building is related to scientific methodology. When working with complex predictive models, it can be easy to unknowingly commit errors related to logical fallacies or inappropriate assumptions. Many machine learning models are so adept at discovering patterns that they can effortlessly find empirical patterns in the data that fail to reproduce later. Some of these types of methodological errors are insidious in that the issue can go undetected until a later time when new data that contain the true result are obtained.

As our models have become more powerful and complex, it has also become easier to commit latent errors.

This same principle also applies to programming. Whenever possible, the software should be able to protect users from committing mistakes. Software should make it easy for users to **do the right thing**.

These two aspects of model development are crucial. Since tools for creating models are easily obtained and models can have such a profound impact, many more people are creating them. In terms of technical expertise and training, their backgrounds will vary. It is important that their tools be *robust* to the experience of the user. Tools should be powerful enough to create high-performance models, but, on the other hand, should be easy to use in an appropriate way. This book describes a suite of software for modeling which has been designed with these characteristics in mind.

The software is based on the R programming language (R Core Team 2014). R has been designed especially for data analysis and modeling. It is an implementation of the S language (with lexical scoping rules adapted from Scheme and Lisp) which was created in the 1970s to

> "turn ideas into software, quickly and faithfully" (Chambers 1998)

R is open-source and free of charge. It is a powerful programming language that can be used for many different purposes but specializes in data analysis, modeling, visualization, and machine learning. R is easily *extensible*; it has a vast ecosystem of packages, mostly user-contributed modules that focus on a specific theme, such as modeling, visualization, and so on.

One collection of packages is called the ***tidyverse*** (Wickham et al. 2019). The tidyverse is an opinionated collection of R packages designed for data science. All packages share an underlying design philosophy, grammar, and data structures. Several of these design philosophies are directly informed by the aspects of software described in this section. If you've never used the tidyverse packages, Chapter 2 contains a review of its basic concepts. Within the tidyverse, the subset of packages specifically focused on modeling are referred to as the ***tidymodels*** packages. This book is an extended software manual for conducting modeling using the tidyverse and tidymodels. It shows how to use a set of packages, each with its own specific purpose, together to create high-quality models.

1.1 TYPES OF MODELS

Before proceeding, let's describe a taxonomy for types of models, grouped by purpose. While not exhaustive, most models fall into *at least* one of these categories:

DESCRIPTIVE MODELS

The purpose of a descriptive model is to describe or illustrate characteristics of some data. The analysis might have no other purpose than to visually emphasize some trend or artifact in the data.

For example, large scale measurements of RNA have been possible for some time using *microarrays*. Early laboratory methods placed a biological sample on a small microchip. Very small locations on the chip can measure a signal based on the abundance of a specific RNA sequence. The chip would contain thousands (or more) outcomes, each a quantification of the RNA related to some biological process. However, there could be quality issues on the chip that might lead to poor results. A fingerprint accidentally left on a portion of the chip might cause inaccurate measurements when scanned.

An early method for evaluating such issues were *probe-level models*, or PLM's (Bolstad 2004). A statistical model would be created that accounted for the *known* differences in the data, such as the chip, the RNA sequence, the type of sequence, and so on. If there were other, unknown factors in the data, these effects would be captured in the model residuals. When the residuals were plotted by their location on the chip, a good quality chip would show no patterns. When a problem did occur, some sort of spatial pattern would be discernible. Often the type of pattern would suggest the underlying issue (e.g. a fingerprint) and a possible solution (wipe the chip off and rescan, repeat the sample, etc.). Figure 1.1(a) shows an application of this method for two microarrays taken from Gentleman et al. (2005). The images show two different colors; red is where the signal intensity was larger than the model expects while the blue color shows lower than expected values. The left-hand panel demonstrates a fairly random pattern while the right-hand panel exhibits an undesirable artifact in the middle of the chip.

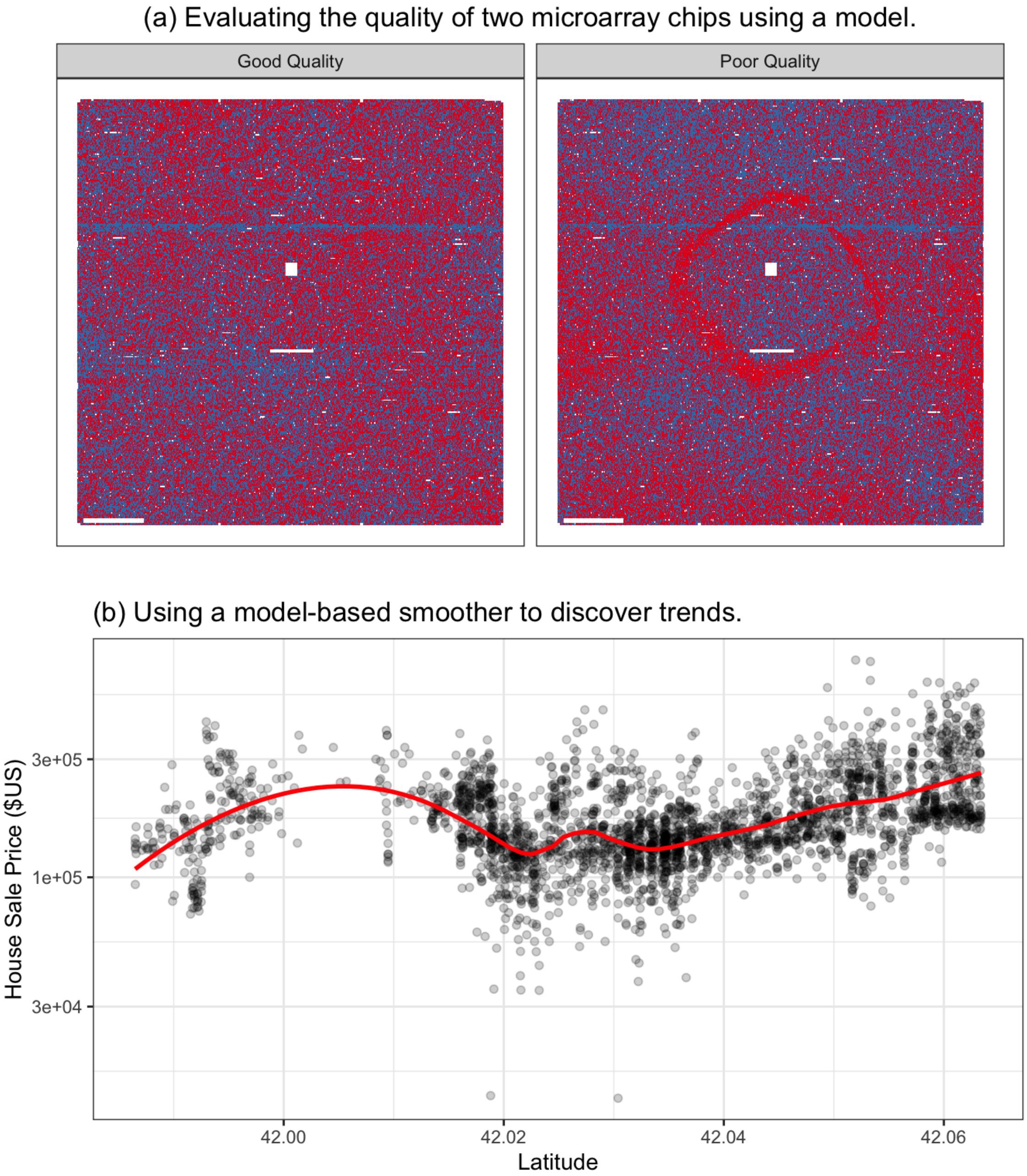

Figure 1.1: Two examples of how descriptive models can be used to illustrate specific patterns.

Another example of a descriptive model is the *locally estimated scatterplot smoothing* model, more commonly known as LOESS (Cleveland 1979). Here, a smooth and flexible regression model is fit to a data set, usually with a single independent variable, and the fitted regression line is used to elucidate some trend in the data. These types of smoothers are used to discover potential ways to represent a variable in a model. This is demonstrated in Figure 1.1(b) where a nonlinear trend is illuminated by the flexible smoother. From this plot, it is clear that there is a highly nonlinear relationship between the sale price of a house and its latitude.

INFERENTIAL MODELS

The goal of an inferential model is to produce a decision for a research question or to test a specific hypothesis, in much the way that statistical tests are used[1]. The goal is to make some statement of truth regarding a predefined conjecture or idea. In many (but not all) cases, a qualitative statement is produced (e.g., a difference was "statistically significant").

For example, the goal of a clinical trial might be to provide confirmation that a new therapy does a better job in prolonging life than an alternative, like an existing therapy or no treatment at all. If the clinical endpoint was related to survival of a patient, the *null hypothesis* might be that the two therapeutic groups have equal median survival times, with the *alternative hypothesis* being that the new therapy has higher median survival. If this trial were evaluated using traditional null hypothesis significance testing via modeling, the significance testing would produce a p-value using some pre-defined methodology based on a set of assumptions for the data. Small values for the p-value in the model results would indicate that there is evidence that the new therapy helps patients live longer. Large values for the p-value in the model results would conclude that there is a failure to show such a difference; this lack of evidence could be due to a number of reasons, including the therapy not working.

What are the important aspects of this type of analysis? Inferential modeling techniques typically produce some type of probabilistic output, such as a p-value, confidence interval, or posterior probability. Generally, to compute such a quantity, formal probabilistic assumptions must be made about the data and the underlying processes that generated the data. The quality of the statistical modeling results are highly dependent on these pre-defined assumptions as well as how much the observed data appear to agree with them. The most critical factors here are theoretical in nature: "If my data were independent and follow distribution *X*, then test statistic *Y* can be used to produce a p-value. Otherwise, the resulting p-value might be inaccurate."

One aspect of inferential analyses is that there tends to be a delayed feedback loop in understanding how well the data matches the model assumptions. In our clinical trial example, if statistical (and clinical) significance indicate that the new therapy should be available for patients to use, it still may be years before it is used in the field and enough data are generated for an independent assessment of whether the original statistical analysis led to the appropriate decision.

PREDICTIVE MODELS

Sometimes data are modeled to produce the most accurate prediction possible for new data. Here, the primary goal is that the predicted values have the highest possible fidelity to the true value of the new data.

A simple example would be for a book buyer to predict how many copies of a particular book should be shipped to their store for the next month. An over-prediction wastes space and money due to excess books. If the prediction is smaller than it should be, there is opportunity loss and less profit.

For this type of model, the problem type is one of *estimation* rather than inference. For example, the buyer is usually not concerned with a question such as "Will I sell more than 100 copies of book *X* next month?" but rather "How many copies of book *X* will customers purchase next month?" Also, depending on the context, there may not be any interest in *why* the predicted value is *X*. In other words, there is more interest in the value itself than evaluating a formal hypothesis related to the data. The prediction can also include measures of uncertainty. In the case of the book buyer, providing a forecasting error may be helpful in deciding how many to purchase. It can also serve as a metric to gauge how well the prediction method worked.

What are the most important factors affecting predictive models? There are many different ways that a predictive model can be created, so the important factors depend on how the model was developed.

A **mechanistic model** could be derived using first principles to produce a model equation that is dependent on assumptions. For example, when predicting the amount of a drug that is in a person's body at a certain time, some formal assumptions are made on how the drug is administered, absorbed, metabolized, and eliminated. Based on this, a set of differential equations can be used to derive a specific model equation. Data are used to estimate the unknown parameters of this equation so that predictions can be generated. Like inferential models, mechanistic predictive models greatly depend on the assumptions that define their model equations. However, unlike inferential models, it is easy to make data-driven statements about how well the model performs based on how well it predicts the existing data. Here the feedback loop for the modeling practitioner is much faster than it would be for a hypothesis test.

Empirically driven models are created with more vague assumptions. These models tend to fall into the machine learning category. A good example is the *K*-nearest neighbor (KNN) model. Given a set of reference data, a new sample is predicted by using the values of the *K* most similar data in the reference set. For example, if a book buyer needs a prediction for a new book, historical data from existing books may be available. A 5-nearest neighbor model would estimate the amount of the new books to purchase based on the sales numbers of the five books that are most similar to the new one (for some definition of "similar"). This model is only defined by the structure of the prediction (the average of five similar books). No theoretical or probabilistic assumptions are made about the sales numbers or the variables that are used to define similarity. In fact, the primary method of evaluating the appropriateness of the model is to assess its accuracy using existing data. If the structure of this type of model was a good choice, the predictions would be close to the actual values.

Broader discussions of these distinctions can be found in Breiman (2001b) and Shmueli (2010).

> Note that we have defined the type of a model by how it is used, rather than its mathematical qualities.

An ordinary linear regression model might fall into any of these three classes of model, depending on how it is used:

- A descriptive smoother, similar to LOESS, called *restricted smoothing splines* (Durrleman and Simon 1989) can be used to describe trends in data using ordinary linear regression with specialized terms.
- An *analysis of variance* (ANOVA) model is a popular method for producing the p-values used for inference. ANOVA models are a special case of linear regression.
- If a simple linear regression model produces highly accurate predictions, it can be used as a predictive model.

There are many examples of predictive models that cannot (or at least should not) be used for inference. Even if probabilistic assumptions were made for the data, the nature of the KNN model makes the math required for inference intractable.

There is an additional connection between the types of models. While the primary purpose of descriptive and inferential models might not be related to prediction, the predictive capacity of the model should not be ignored. For example, logistic regression is a popular model for data where the outcome is qualitative with two possible values. It can model how variables are related to the probability of the outcomes. When used in an inferential manner, there is usually an abundance of attention paid to the *statistical qualities* of the model. For example, analysts tend to strongly focus on the selection of which independent variables are contained in the model. Many iterations of model building are usually used to determine a minimal subset of independent variables that have a “statistically significant” relationship to the outcome variable. This is usually achieved when all of the p-values for the independent variables are below some value (e.g. 0.05). From here, the analyst typically focuses on making qualitative statements about the relative influence that the variables have on the outcome (e.g., “There is a statistically significant relationship between age and the odds of heart disease.”).

This can be dangerous when statistical significance is used as the *only* measure of model quality. It is possible that this statistically optimized model has poor model accuracy, or performs poorly on some other measure of predictive capacity. While the model might not be used for prediction, how much should inferences be trusted from a model that has significant p-values but dismal accuracy? Predictive performance tends to be related to how close the model’s fitted values are to the observed data.

> If a model has limited fidelity to the data, the inferences generated by the model should be highly suspect. In other words, statistical significance may not be sufficient proof that a model is appropriate.

This may seem intuitively obvious, but is often ignored in real-world data analysis.

1.2 SOME TERMINOLOGY

Before proceeding, we outline here some additional terminology related to modeling and data. These descriptions are intended to be helpful as you read this book but not exhaustive.

First, many models can be categorized as being *supervised* or *unsupervised*. Unsupervised models are those that learn patterns, clusters, or other characteristics of the data but lack an outcome, i.e., a dependent variable. Principal component analysis (PCA), clustering, and autoencoders are examples of unsupervised models; they are used to understand relationships between variables or sets of variables without an explicit relationship between predictors and an outcome. Supervised models are those that have an outcome variable. Linear regression, neural networks, and numerous other methodologies fall into this category.

Within supervised models, there are two main sub-categories:

- **Regression** predicts a numeric outcome.
- **Classification** predicts an outcome that is an ordered or unordered set of qualitative values.

These are imperfect definitions and do not account for all possible types of models. In Chapter 6, we refer to this characteristic of supervised techniques as the *model mode*.

Different variables can have different *roles*, especially in a supervised modeling analysis. Outcomes (otherwise known as the labels, endpoints, or dependent variables) are the value being predicted in supervised models. The independent variables, which are the substrate for making predictions of the outcome, are also referred to as predictors, features, or covariates (depending on the context). The terms *outcomes* and *predictors* are used most frequently in this book.

In terms of the data or variables themselves, whether used for supervised or unsupervised models, as predictors or outcomes, the two main categories are quantitative and qualitative. Examples of the former are real numbers like `3.14159` and integers like `42` . Qualitative values, also known as nominal data, are those that represent some sort of discrete state that cannot be naturally placed on a numeric scale, like “red”, “green”, and “blue”.

1.3 HOW DOES MODELING FIT INTO THE DATA ANALYSIS PROCESS?

In what circumstances are models created? Are there steps that precede such an undertaking? Is it the first step in data analysis?

> There are always a few critical phases of data analysis that come before modeling.

First, there is the chronically underestimated process of **cleaning the data**. No matter the circumstances, you should investigate the data to make sure that they are applicable to your project goals, accurate, and appropriate. These steps can easily take more time than the rest of the data analysis process (depending on the circumstances).

Data cleaning can also overlap with the second phase of **understanding the data**, often referred to as exploratory data analysis (EDA). EDA brings to light how the different variables are related to one another, their distributions, typical ranges, and other attributes. A good question to ask at this phase is, "How did I come by *these* data?" This question can help you understand how the data at hand have been sampled or filtered and if these operations were appropriate. For example, when merging database tables, a join may go awry that could accidentally eliminate one or more sub-populations. Another good idea is to ask if the data are *relevant*. For example, to predict whether patients have Alzheimer's disease or not, it would be unwise to have a data set containing subjects with the disease and a random sample of healthy adults from the general population. Given the progressive nature of the disease, the model may simply predict who are the *oldest patients*.

Finally, before starting a data analysis process, there should be clear expectations of the goal of the model and how performance (and success) will be judged. At least one *performance metric* should be identified with realistic goals of what can be achieved. Common statistical metrics, discussed in more detail in Chapter 9, are classification accuracy, true and false positive rates, root mean squared error, and so on. The relative benefits and drawbacks of these metrics should be weighed. It is also important that the metric be germane; alignment with the broader data analysis goals is critical.

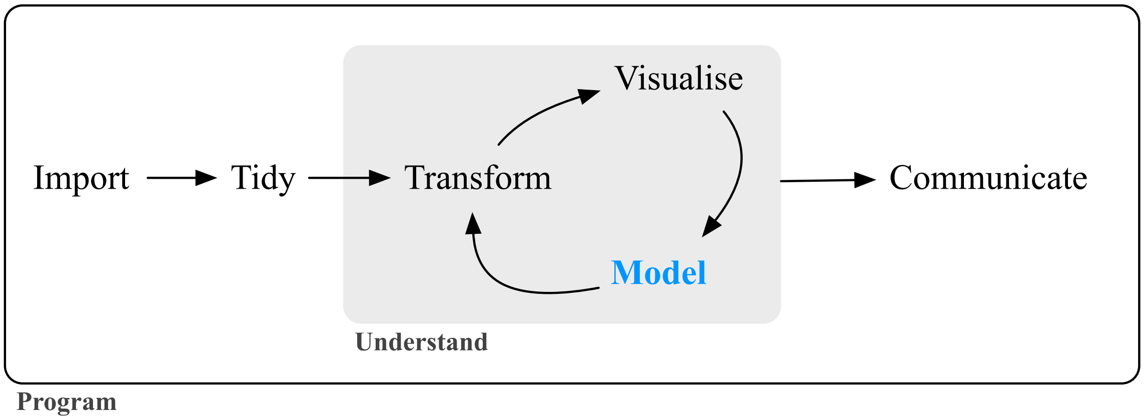

Figure 1.2: The data science process (from R for Data Science).

The process of investigating the data may not be simple. Wickham and Grolemund (2016) contains an excellent illustration of the general data analysis process, reproduced with Figure 1.2. Data ingestion and cleaning/tidying are shown as the initial steps. When the analytical steps for understanding commence, they are a heuristic process; we cannot pre-determine how long they may take. The cycle of analysis, modeling, and visualization often requires multiple iterations.

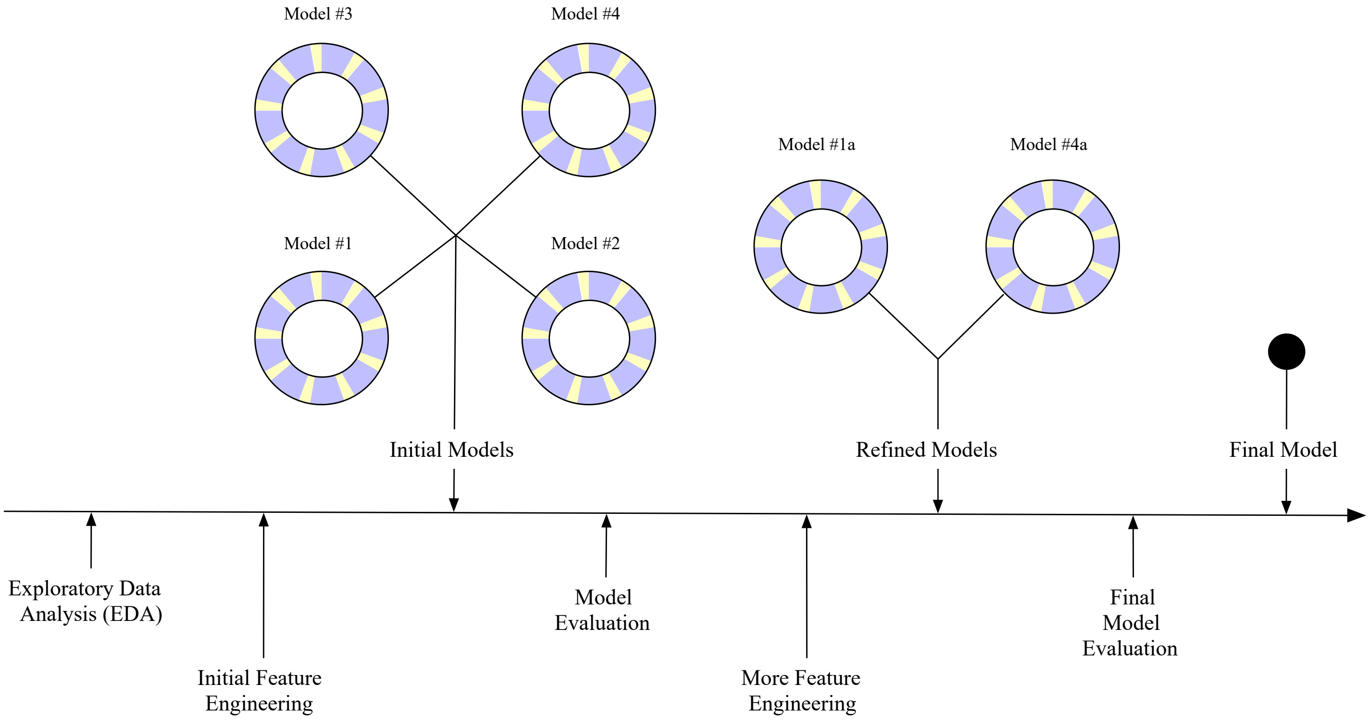

Figure 1.3: A schematic for the typical modeling process.

This iterative process is especially true for modeling. Figure 1.3 is meant to emulate the typical path to determining an appropriate model. The general phases are:

- **Exploratory data analysis (EDA):** Initially there is a back and forth between numerical analysis and visualization of the data (represented in Figure 1.2) where different discoveries lead to more questions and data analysis "side-quests" to gain more understanding.

- **Feature engineering:** The understanding gained from EDA results in the creation of specific model terms that make it easier to accurately model the observed data. This can include complex methodologies (e.g., PCA) or simpler features (using the ratio of two predictors). Chapter 8 focuses entirely on this important step.

- **Model tuning and selection (circles with blue and yellow segments):** A variety of models are generated and their performance is compared. Some models require *parameter tuning* where some structural parameters are required to be specified or optimized. The colored segments within the circles signify the repeated data splitting used during resampling (see Chapter 10).

- **Model evaluation:** During this phase of model development, we assess the model's performance metrics, examine residual plots, and conduct other EDA-like analyses to understand how well the models work. In some cases, formal between-model comparisons (Chapter 11) help you to understand whether any differences in models are within the experimental noise.

After an initial sequence of these tasks, more understanding is gained regarding which types of models are superior as well as which sub-populations of the data are not being effectively estimated. This leads to additional EDA and feature engineering, another round of modeling, and so on. Once the data analysis goals are achieved, the last steps are typically to finalize, document, and communicate the model. For predictive models, it is common at the end to validate the model on an additional set of data reserved for this specific purpose.

As an example, Kuhn and Johnson (2020) use data to model the daily ridership of Chicago's public train system using predictors such as the date, the previous ridership results, the weather, and other factors. An approximation of these authors' "inner monologue" when analyzing these data is, in order:

Thoughts	Activity
The daily ridership values between stations are extremely correlated.	EDA
Weekday and weekend ridership look very different.	EDA
One day in the summer of 2010 has an abnormally large number of riders.	EDA
Which stations had the lowest daily ridership values?	EDA
Dates should at least be encoded as day-of-the-week, and year.	Feature Engineering
Maybe PCA could be used on the correlated predictors to make it easier for the models to use them.	Feature Engineering
Hourly weather records should probably be summarized into daily measurements.	Feature Engineering
Let’s start with simple linear regression, K-nearest neighbors, and a boosted decision tree.	Model Fitting
How many neighbors should be used?	Model Tuning
Should we run a lot of boosting iterations or just a few?	Model Tuning
How many neighbors seemed to be optimal for these data?	Model Tuning
Which models have the lowest root mean squared errors?	Model Evaluation
Which days were poorly predicted?	EDA
Variable importance scores indicate that the weather information is not predictive. We’ll drop them from the next set of models.	Model Evaluation
It seems like we should focus on a lot of boosting iterations for that model.	Model Evaluation
We need to encode holiday features to improve predictions on (and around) those dates.	Feature Engineering
Let’s drop K-NN from the model list.	Model Evaluation

and so on. Eventually, a model is selected that is able to achieve sufficient performance.

1.4 CHAPTER SUMMARY

This chapter focused on how models describe relationships in data, and different types of models such as descriptive models, inferential models, and predictive models. The predictive capacity of a model can be used to evaluate it, even when its main goal is not prediction. Modeling itself sits within the broader data analysis process, and exploratory data analysis is a key part of building high-quality models.

For all kinds of modeling, software for building models must support good scientific methodology and ease of use for practitioners from diverse backgrounds. The software we develop approaches this with the ideas and syntax of the tidyverse, which we introduce (or review) in Chapter 2. Chapter 3 is a quick tour of conventional base R modeling functions and summarize the unmet needs in that area.

After that, this book is separated into parts, starting with the basics of modeling with tidy data principles. The first part introduces an example data set on house prices and demonstrates how to use the fundamental tidymodels packages: **recipes**, **parsnip**, **workflows**, **yardstick**, and others.

The second part of the book moves forward with more details on the process of creating a good model. This includes creating good estimates of performance as well as tuning model parameters.

REFERENCES

Abrams, B. 2003. “The Pit of Success.” https://blogs.msdn.microsoft.com/brada/2003/10/02/the-pit-of-success/.

Baggerly, K, and K Coombes. 2009. “Deriving Chemosensitivity from Cell Lines: Forensic Bioinformatics and Reproducible Research in High-Throughput Biology.” *The Annals of Applied Statistics* 3 (4): 1309–34.

Bolstad, B. 2004. *Low-Level Analysis of High-Density Oligonucleotide Array Data: Background, Normalization and Summarization*. University of California, Berkeley.

Breiman, L. 2001b. “Statistical Modeling: The Two Cultures.” *Statistical Science* 16 (3): 199–231.

Carlson, B. 2012. “Putting Oncology Patients at Risk.” *Biotechnology Healthcare* 9 (3): 17–21.

Chambers, J. 1998. *Programming with Data: A Guide to the S Language*. Berlin, Heidelberg: Springer-Verlag.

Cleveland, W. 1979. “Robust Locally Weighted Regression and Smoothing Scatterplots.” *Journal of the American Statistical Association* 74 (368): 829–36.

Durrleman, S, and R Simon. 1989. “Flexible Regression Models with Cubic Splines.” *Statistics in Medicine* 8 (5): 551–61.

Gentleman, R, V Carey, W Huber, R Irizarry, and S Dudoit. 2005. *Bioinformatics and Computational Biology Solutions Using R and Bioconductor*. Berlin, Heidelberg: Springer-Verlag.

Kuhn, M, and K Johnson. 2020. *Feature Engineering and Selection: A Practical Approach for Predictive Models*. CRC Press.

R Core Team. 2014. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. http://www.R-project.org/.

Shmueli, G. 2010. “To Explain or to Predict?” *Statistical Science* 25 (3): 289–310.

Wickham, H, M Averick, J Bryan, W Chang, L McGowan, R François, G Grolemund, et al. 2019. “Welcome to the Tidyverse.” *Journal of Open Source Software* 4 (43).

Wickham, H, and G Grolemund. 2016. *R for Data Science: Import, Tidy, Transform, Visualize, and Model Data*. O’Reilly Media, Inc.

[1] Many specific statistical tests are in fact equivalent to models. For example, t-tests and analysis of variance (ANOVA) methods are particular cases of the generalized linear model.↩