Data-Driven Decision-Making

Judith Künneke, Bjarne Brie, Harm Schütt

2025-03-21

Table of contents

# Preface



This web-book contains the lecture notes for the course **324070–Data-Driven Decision-Making** taught in the Master of Accountancy at Tilburg University.

This is a living document, we expect to improve on it regularly as the course grows. If you stumble on this site and find it useful, or if you have suggestions, we are always happy for feedback.

Best regards

Dr. Judith Künneke Instructor and course coordinator Office: Koopmans Buidling, K 205

Dr. Bjarne Brie Instructor Office: Koopmans Buidling, K 253

# 1. Introduction

## 1.1 Why have a data-driven decision-making course in a controlling track?

### 1.1.1 Data analytics to unlock the potential of data

To answer this question, consider, for example, the four common elements of any control process:

1. Establishing Performance Standards.
2. Measuring the Actual Performance.
3. Comparing Actual Performance to the Standards.
4. Taking Corrective Action.

In executing all these elements, decisions have to be made. Ideally, such decisions are based on data. Most firms these days collect data all the time. Valuable data, if used well. However, in its raw form data is useless. You cannot stare at a three-terabyte large table of customer data and expect to come up with better marketing strategies. Raw data needs to be structured and aggregated just the right way in order to yield insights. This is where data analytics comes in. Data analytics takes raw data and then structures and aggregates it to find meaningful data patterns. Those data patterns can be used to make better business decisions.

### 1.1.2 From data analytics to better decisions

However, data analytics is just one part of two. It is one thing to find regular patterns in the data. But it is a different skill set that helps you determine whether these are patterns that contain new insights—or how to make sensible decisions based on the patterns. For that, you need expert knowledge in the domain that the decision problem arose. And this is why we have this course.

Our course is tailored to management accounting and related questions, for which you need a unique combination of analysis techniques and accounting expert knowledge in order to come up with good decisions. It is of course not the only area where data analytics is useful. In fact, most business analytics books use examples like analyzing marketing campaign effectiveness or supply chain efficiency. This is another reason why we decided a course like this should be part of the curriculum.

## 1.2 Types of decision problems and accompanying questions

What are typical business decisions you’ll encounter? Consider the following job posting for a data analyst. Many of the duties listed here are closely related to controlling questions.

|  |
| --- |
| Figure 1.1: A data analyst job posting (April 5th, 2023) |

Controlling departments face very specific decision problems. For example, consider what you learned about the budgeting and forecasting process in the *Budgeting and Forecasting* course:

|  |
| --- |
| Figure 1.2: Budgeting and Forecasting Workflow |

A controlling department actively involved in setting and monitoring a firm’s strategy and performance will face numerous challenging questions. Each question is tied to a decision to be made. Here are just a few examples:

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Table 1.1: Questions and Decisions   | Question | Decision | | --- | --- | | What would be the operating profitability of opening a typical store in France? | Whether to expand operations to France? | | How would our revenues and costs be affected if Russia invades Ukraine? | How much effort to allocate to contingency planning? | | What are reasonable benchmarks for the cost allocation of overhead? | How to best allocate costs for controlling purposes? | | How would our most profitable customer segment react if we serve them exclusively via an online channel? | Whether to change a part of the business strategy? | | What is the most “at risk” driver of the net margin of our blockbuster product? | Whether profitability of the main product is “safe” enough or needs action? | |

For many of the decisions listed in [Table 1.1](#tbl-exampleqs), the question asked is just one of a collection of questions that is important for an overall decision to be made. In most cases, it is also not immediately obvious how you would use data to answer these questions. Exactly for that reason we recommend starting any analysis by carefully thinking about the question you need an answer to. Next, we will talk about a framework for decision-making in the next section [Chapter 2](#sec-decmb). The opposite, starting from the data, is often not advisable. Often you can (and often should) look at many different sources of data to find answers to a question. If you start with the data at hand, you might get misleading conclusions, because you analyzed inadequate, but readily available, data. For those and many other reasons, our framework starts with thinking carefully about the business question to be analyzed.

## 1.3 Summary: What this course teaches you

In today’s economy, most firms collect large amounts of data that have the potential to greatly increase decision-making quality inside firms. To properly digest and leverage data, a firm needs employees that are not only well -versed in data analytics (being able to find meaningful patterns in the raw data) but also draw the right conclusions from the patterns and formulate adequate decisions, and report the conclusions and decisions adequately. In this course, we will cover all of these aspects.

# 2. Decision-Making Basics

## 2.1 Learning goals

1. Introduce, apply, and evaluate a template for making data-driven decisions
2. Highlight the importance of connecting analysis goals to decision-making goals
3. Introduce a taxonomy of different types of analyses based on the type of questions that can be answered
4. Discuss mental patterns useful for critically evaluating analyses and reducing analysis errors

## 2.2 Rational decision-making

As stated in [Chapter 1](#sec-introduction), this course is about how to make sensible decisions with the help of data. Before we dive into doing data analysis though, we first want to discuss what “sensible decision-making” means. So we first need to talk about the building blocks of a decision problem. Some of the terms we’ll discuss (e.g., minimax) may sound strange or complicated. But, all we are doing is presenting some basic reasoning for how to best structure an analysis and draw conclusions from it.

The first thing to note is that most important business decisions involve some degree of uncertainty. Think of uncertainty simply as *missing knowledge about past, present, or future events (Walkter et al. 2003).* The knowledge that you would need in order to choose the best action under various potential actions.

We care about uncertainty because whenever there is uncertainty, there is a chance to make a mistake or make a sub-optimal choice. Also, in lots of real-world cases, some mistakes are more costly than others. For example, when prescribing a drug to a patient, the mistake “the drug didn’t work” is potentially less “costly” to the patient than the mistake “the drug had some serious side-effects on the patient”. This implies that a useful decision rule suggests not only an expected best action out of a set of actions. It also accounts for the costs of different kinds of potential mistakes.

Not always will we be able to quantify costs or probabilities of making mistakes though. That’s fine, we will still need to come up with simple decision rules that will help us make sensible decisions. Think of rules like: “If this course of action still performs best in the worst case, then let’s do it” (called maximin rule).

Ideally, though, we’d like to reduce the chance of making (costly) mistakes and that implies we want to reduce uncertainty in our decision-making. Data is the key input to reducing uncertainty. But data alone is not enough. We also need assumptions. Why? That will become apparent during the course. For now, let us discuss the decision-making process.

## 2.3 A decision-making process framework

The first part of the course is about learning a framework. Why spend a full chapter on it? Because Data analysis is not easy. It is easy to get stuck and stare at the data, not seeing the forest for the trees. Over the years of teaching and practicing data analysis in various forms, we have seen it happen frequently and to students at all levels–from bachelor to PhD students. It happens to us too still. Especially for beginners, the best remedy against getting stuck is to have a helpful framework of procedures. So, let us try to give you one.

If you google “business decision-making” then you will at some point stumble upon a process picture like this one:

|  |
| --- |
| Figure 2.1: A stylized rational decision process |

[Figure 2.1](#fig-dmdiag1) (based on Schoenfeld (2010)) shows our version of what a rational decision-making process looks like. While very stylized, this is not a bad schema for trying to find answers to important business problems. Let us explain each step in turn.

### 2.3.1 Detect the problem

Sometimes the problem is clear. Designing a new sales forecasting model or optimizing inventory are problems that are usually handed to you. However, sometimes we first need to realize that there is a problem in the first place (e.g., production quality is slowly decreasing). This does not always happen automatically. Well-designed monitoring systems that collect data and measure developments across multiple dimensions often throw the first warning signs that something needs to be looked at. (And somebody had to come up with the idea that it might be a sensible business precaution to install such a monitoring system.)

### 2.3.2 Identify the problem

To us, this is the most important step. Essentially, here we clarify what we want to know. Define what the current situation looks like. Then define the ideal outcome we are looking for. Describe the difference and then try to figure out what is causing the difference. The root-cause analysis will give you a path towards the problem.

It is key to slow down and properly identify what the problem really is. Do not go in blind and start digging. Even with exploratory analyses, first step back and take a moment to go through the steps described in the previous paragraph. This is not always easy. For example, is a sales decline because of demotivated salespeople or are salespeople demotivated because the product is outdated? It can easily happen that we mistake symptoms for causes. One of the most important messages of this chapter is: Guide your analysis by specifying the goal clearly and asking a succession of questions that are aimed at bringing you closer to the goal.

Often, we need to gather and analyze a significant amount of data to better understand what is happening. A lot of our data analysis will happen at this stage. Consider the following simple example.

You are working for a technology vendor that not only sells the technology but also service packages. The service packages are a big part of overall revenues. The CFO of the company comes to you and mentions that one client reached out to him being unhappy and not wanting to renew its service package. She asks you to look into what the overall satisfaction with the service is. You open the data on the latest customer survey you ran and look at the answers:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Table 2.1: Survey results   |  | Responses | Percent | | --- | --- | --- | | Highly Likely | 796 | 49.9 | | Likely | 350 | 22.0 | | Maybe | 253 | 15.9 | | Probably Not | 109 | 6.8 | | Certainly Not | 86 | 5.4 | | TOTAL | 1594 | 100 | |

[Table 2.1](#tbl-survey1) seems to suggest that 12% of respondents indicate that they are “probably not” or “certainly not” renewing the service package. You know from experience that such survey answers often overstate the true dropout rate. You also know that the numbers are in line with survey responses from previous years. So you could go to the CFO and tell her that nothing seems to have changed.

You don’t. Your company has different service tiers and you decide to split the survey responses by service tier. You are presented with the following data:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Table 2.2: Survey results stratified by service tier   |  | Service Tier 1 | Service Tier 2 | Service Tier 3 | Service Tier 4 | Service Tier 5 | | --- | --- | --- | --- | --- | --- | | Highly Likely | 218 | 212 | 163 | 111 | 92 | | Likely | 79 | 74 | 78 | 61 | 58 | | Maybe | 59 | 52 | 65 | 55 | 22 | | Probably Not | 16 | 19 | 13 | 39 | 22 | | Certainly Not | 24 | 15 | 7 | 11 | 29 | | TOTAL | 396 | 372 | 326 | 277 | 223 | |

You can see that most service tiers are fine. However, [Table 2.2](#tbl-survey2) shows that the high tiers (tier 4 and 5, the most expensive ones) have significantly higher frequencies of lower responses. *You found a meaningful pattern.* This is a very simple example of what data analytics is all about: finding meaningful patterns in raw data. (see [Chapter 1](#sec-introduction)). We will have much more to say about this in [Chapter 3](#sec-datapat).

You create a quick visualization that shows the pattern more clearly. This pattern was not obvious in the overall survey answers in [Table 2.1](#tbl-survey1). The reason is that there are fewer responses for high-tier service package than for low-tier packages. The pattern is this much more obvious in percentage terms ([Figure 2.2](#fig-survey)):

|  |
| --- |
| Figure 2.2: Percentage of respondents stratified by service tier |

Stop for a moment and consider how you would have possibly spotted this pattern if you had not known it existed (an unknown unknown). How could you have found it? We used some expert knowledge (there are multiple service tiers) and we had some type of *mental model* of the customers. That model suggested that customer siatifaction might vary across tiers (maybe higher tier customers are more demanding). Both led us to entertain the idea that mismatches might differ across tiers. We point this out to highlight the importance of models. Only if you combine the right inquisitive mindset with a bit of background/expert knowledge, will you be able to properly identify problems.

You probably do not want to stop right there either. Before going to your boss to tell her the bad news, you might want to better understand what is going on. In other words, we have a guiding question for the next analysis: What might be the reason for the low survey score for the expensive tiers? And how do we figure that out? Again, we now need to come up with a mental model of how the customer response came to pass and start testing for indications that are either consistent or inconsistent with this model–rinse and repeat.

The general steps remain the same for all types of decision problems you might face. If you need to come up with a sales forecast then “identifying the problem” becomes clarifying what exactly you are supposed to forecast (e.g., all sales? product, service, and advertising? How far into the future? On a yearly or monthly basis?). Often the right answers to those questions can be found by asking a slightly different question–one of purpose. “What is the sales forecast for?”.

The last example is thus also there to illustrate the power of asking the right questions. In fact, one can structure a decision problem into a series of questions to be answered. Example questions for this stage of the process include:

* Who is doing what to whom?
* Where/When does the problem appear to arise?
* What is the process behind what we observe?
* What is the reason for the suspected reason?
* What do we need to solve this problem?

### 2.3.3 Establish decision criteria

To make good decisions, we need to have an idea of what goal we are working towards! Thus, once we have a better grip on what exactly the issue is, we need to set some criteria that define what a good solution should look like.

It is hard to give good guidance here because these criteria are usually problem specific. For example, a sales forecast should have “expected accuracy” as a criterion to accept a forecast or not. But there are other potential criteria too. Typical decision criteria considered are financial benefits, resource usage, quality, riskiness, acceptability to others, etc. In the case of the sales forecast, another criterion might be complexity. Are we willing to entertain forecasts based on hard-to-reason or even black-box machine learning models or do we only entertain forecasts based on models we can reason about? This might be an important decision criterion depending on the costs of forecast errors and who is going to be held accountable for decisions made based on the forecast.

Again, this step can be quite tricky, but we believe it is vital to establish criteria at this stage of the process. Without criteria to describe the goal we are working towards, there is a big risk of wasting serious time and effort into designing solutions that will be thrown away immediately.

If there are multiple criteria, then it is important to discuss how to weigh them. Multi-criteria decision-making is not an easy topic, because the notion of a “best solution” becomes non-trivial (Pomerol and Barba-Romero 2000; Aruldoss, Lakshmi, and Venkatesan 2013). It is not always obvious what the right weights should be. Because of time constraints and scope, we likely won’t go into much detail here. But we want to raise awareness for the issue and cover some simpler examples.

Typical questions you might want to ask yourself at this stage:

* What is affected by the possible alternatives?
* What do we need to trade off (e.g., costs versus benefits)?
* What cost is more significant for us?

### 2.3.4 Develop alternatives

After you have set the relevant criteria for a good solution it is time to come up with potential alternatives that might lead to a solution to the problem at hand.

Creativity is often key here. The insights gathered from a careful root-cause analysis in Step 2: “identify the problem” are usually very useful at this stage too. Developing alternative solutions often involves building and entertaining different mental models of how the problem might have arisen. Sometimes this is done in brainstorm sessions with the whole team. Imagine you are part of a team and some team members disagree on what the best course of action is. In some cases (e.g., enough resources) it might be the best way forward to run both courses in parallel until one alternative clearly dominates.

This is another step where the large chunk of the data analytics resides. It is required for at least two reasons:

1. We need to collect and analyze raw data to develop mental models and propose alternative solutions based on them.
2. We need to collect and analyze raw data test and evaluate the assumptions underlying the alternative solutions. Are the mental models on which the solutions are resting supported by the data? This is often called diagnostic analysis and the topic of [Chapter 4](#sec-diag).
3. Properly fleshed-out solutions often require data-based inputs.

To give an example of different alternative solutions to a problem requiring data to analyze: imagine you are working for a supermarket chain. The top management is considering expansion strategies and you are charged with forecasting the potential sales of a new market at a new location. Since the chain has never had a store at that location, there is no historical data. Let’s further assume it was decided that there is only limited budget for external data and good solutions need to be transparent in their reasoning. How would you go about this?

Well, you have internal data on the other 140 supermarkets the firm runs. You can, for example, extract a table like this from the company databases:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Table 2.3: Supermarket example 1: Turnover data by market for 2022   | Market ID | Market size (m2) | No. Clients | Revenues (€) | | --- | --- | --- | --- | | 1 | 1,000 | 378,040 | 4,536,480 | | 2 | 800 | 300,395 | 1,501,975 | | 3 | 1,200 | 431,667 | 2,158,335 | | 4 | 800 | 337,552 | 4,050,624 | | 5 | 800 | 512,045 | 3,584,315 | | 6 | 1,000 | 670,372 | 8,044,464 | | 7 | 1,400 | 406,762 | 5,694,668 | | 8 | 800 | 278,441 | 2,227,528 | | 9 | 1,000 | 315,310 | 3,783,720 | | 10 | 1,000 | 540,726 | 8,110,890 | |

You know that the location available is ca. 1,200 square meters. That is one reason why your firm is considering setting up shop in that town. You could start by looking at what supermarkets of that size in your portfolio produce in revenues. You start by computing some average statistics:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Table 2.4: Supermarket example 2: Revenue(€) descriptives by market size group   | Size Group (sqm) | N | Avg. | SD | 5% Perc. | 95% Perc. | | --- | --- | --- | --- | --- | --- | | 700 | 16 | 4,874T€ | 3,326T€ | 1,944T€ | 10,849T€ | | 800 | 26 | 4,184T€ | 3,409T€ | 1,417T€ | 10,833T€ | | 1000 | 44 | 5,716T€ | 3,626T€ | 1,670T€ | 11,937T€ | | 1200 | 29 | 5,674T€ | 3,098T€ | 1,811T€ | 10,550T€ | | 1400 | 25 | 5,911T€ | 2,426T€ | 2,720T€ | 9,837T€ | |

The averages (means) are roughly as expected, with bigger stores having more revenues. But not always, The 29 1,200 square meter supermarkets have on average lower revenues than the 1,000 square meter markets.

Does it make sense to just report the mean of 5,674T€ as an estimate of the new market’s revenues? Probably not. The other columns in [Table 2.4](#tbl-supmart2) give us an indication of the variation among the 1,200 square meter markets. The standard deviation (SD) measures how dispersed the data is relative to the mean. The SD of 3,098T€, circa 3 million euros, is large for a “standard” distance from the mean. We also get a feeling of significant variation when examining the 5% and 95% percentiles. They tell us that 5% of the 29 markets have revenues lower than 1,8 million euros and 5% have revenues higher than 10,5 million euros.[[1]](#footnote-57). The important point is that the three descriptives SD, 5%, and 95% percentiles that there is a lot of variation around the mean. That tells us that “report the mean of similarly sized supermarkets” is not a good potential solution to the problem “provide a sales forecast for a store at the new location”. It is very likely to violate any accuracy-related decision criterion.

We learned from the data that we cannot just look at similarly sized supermarkets we already have. What alternatives do we have? We now need to start entertaining some mental models. You often do this automatically, but we want to highlight the importance of doing this a bit more explicitly. The question we want to start with when building a model is: What would a process look like that generates the outcome we are interested in? (In this case: supermarket revenues). Such a model is also sometimes called a “data-generating process”. We can start simple. We could split revenues into two determinants: the number of customers served times the average basket size of their shopping (volume times price). And then we think about possible determinants of these two determinants. What would determine the average basket size? The average basket size is probably determined by the average wealth of the customers (do they buy more expensive products) and the amount they buy. The latter can be driven by many things. Are the average customers busy people who eat a lot outside and only buy small quantities? Or do they buy for a whole family, etc.? What would determine how many customers buy in the market? The market size probably plays a role (you are more likely to find what you are looking for) and simply how close the market is to the customer. Then, stores in densely populated areas should have more customers. On the other hand though, it is probably more difficult and more expensive to find a large location for a store in densely populated areas. Thus, there should also be a negative relation between market size and population density in an area. We often like to draw DAGs (directed acyclical graphs) to visualize the data-generating process we have in mind. [Figure 2.3](#fig-dag-sm) shows one describing our reasoning thus far.

|  |
| --- |
| Figure 2.3: DAG of a simple model that generates supermarket revenues from location (Loc.) characteristics |

[Figure 2.3](#fig-dag-sm) just describes our reasoning in a simplified process graph. The node “Loc. lifestyle” is a catch-all for things like: family shopping or commuting frequency, and other things affecting your shopping behavior. For example, consultants jet-setting through the country most of the week will probably not buy many items but more expensive ones. For now, we just put a node like this in here, rather than expanding our thinking on how lifestyle affects shopping behavior.

Often visualizing your mental model like this helps you inspect it. Is our reasoning sound? Have we forgotten anything important? Probably. For example, we do not have competition from other supermarkets in the area in our model. But it is often better to start simple and add more detail later.

We want to test how descriptive this model is for the true data-generating process. Because if it can explain a decent part of the variation that we have observed in revenues of supermarkets of the same size ([Table 2.4](#tbl-supmart2)), then it could give us very helpful guidance on how to make better predictions.

How can we test the model’s descriptive power? We are going into much more detail on how to do this when discussing predictive analytics in [Chapter 5](#sec-pred). But often you can already get a decent idea with some simple descriptives. You have old data on the population density from previous planning projects readily at hand. It is sorted into categories though. Still, we can expand our descriptive statistics and see how much variation within size and density groups exists:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Table 2.5: Supermarket example 3: Revenue(€) descriptives by market size group and location density category   | Groups: Size (sqm) x Density | N | Avg. | SD | 5% Perc. | 95% Perc. | | --- | --- | --- | --- | --- | --- | | 1200 x A | 6 | 4,095T€ | 2,236T€ | 1,418T€ | 6,658T€ | | 1200 x B | 5 | 4,326T€ | 1,779T€ | 2,390T€ | 6,490T€ | | 1200 x C | 6 | 5,844T€ | 2,602T€ | 3,459T€ | 9,561T€ | | 1200 x D | 2 | 5,347T€ | 3,995T€ | 2,805T€ | 7,890T€ | | 1200 x E | 10 | 7,258T€ | 3,826T€ | 3,033T€ | 12,943T€ | |

Compared to the within-group variation shown in [Table 2.4](#tbl-supmart2), [Table 2.5](#tbl-supmart3) shows that we already do a better job by considering density in the area as a determinant of revenues. The variation across density categories shows itself in the variation across averages and is sizable (compare it to the mean of 5,674T€ for the 1,200 sqm supermarkets). We only focus on the relevant size category (1,200 square meters) here for ease of exposition, but you should of course always check whether your model holds for all areas of the data. The standard deviation and percentiles still indicate a sizable amount of variation though. If we can further improve on explained variation, it would be worthwhile to do so.

Let’s assume you also have categories of how wealthy people are on average in the area around the supermarket. These are also from previous planning projects (never throw away data). We can also expand our descriptives via this potential determinant:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Table 2.6: Supermarket example 4: Revenue(€) descriptives by market size group and location wealth category   | Groups: Size (sqm) x Wealth | N | Avg. | SD | 5% Perc. | 95% Perc. | | --- | --- | --- | --- | --- | --- | | 1200 x A | 6 | 2,283T€ | 715T€ | 1,418T€ | 3,065T€ | | 1200 x B | 5 | 4,271T€ | 953T€ | 3,295T€ | 5,238T€ | | 1200 x C | 13 | 6,407T€ | 1,927T€ | 4,105T€ | 9,295T€ | | 1200 x D | 4 | 7,792T€ | 2,994T€ | 4,616T€ | 10,536T€ | | 1200 x E | 1 | 15,021T€ | NA | 15,021T€ | 15,021T€ | |

[Table 2.6](#tbl-supmart4) shows that wealth also sorts well within the size category of 1,200 square meter markets in our portfolio. You can even combine all our splits and expand your descriptives by a three-way-split:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Table 2.7: Supermarket example 5: Revenue(€) descriptives by market size group, location density category, and location wealth category   | Groups: Size (sqm) x Density x Wealth | N | Avg. | SD | 5% Perc. | 95% Perc. | | --- | --- | --- | --- | --- | --- | | 1200 x A x A | 2 | 1,472T€ | 153T€ | 1,375T€ | 1,569T€ | | 1200 x A x C | 3 | 5,768T€ | 1,176T€ | 4,735T€ | 6,849T€ | | 1200 x A x D | 1 | 4,319T€ | — | 4,319T€ | 4,319T€ | | 1200 x B x A | 1 | 2,158T€ | — | 2,158T€ | 2,158T€ | | 1200 x B x B | 1 | 3,315T€ | — | 3,315T€ | 3,315T€ | | 1200 x B x C | 3 | 5,386T€ | 1,335T€ | 4,291T€ | 6,660T€ | | 1200 x C x B | 1 | 3,290T€ | — | 3,290T€ | 3,290T€ | | 1200 x C x C | 3 | 4,941T€ | 893T€ | 4,083T€ | 5,659T€ | | 1200 x C x D | 2 | 8,474T€ | 3,075T€ | 6,517T€ | 10,431T€ | | 1200 x D x A | 1 | 2,523T€ | — | 2,523T€ | 2,523T€ | | 1200 x D x C | 1 | 8,172T€ | — | 8,172T€ | 8,172T€ | | 1200 x E x A | 2 | 3,038T€ | 77T€ | 2,989T€ | 3,087T€ | | 1200 x E x B | 3 | 4,917T€ | 502T€ | 4,422T€ | 5,249T€ | | 1200 x E x C | 3 | 8,943T€ | 1,310T€ | 7,938T€ | 10,218T€ | | 1200 x E x D | 1 | 9,902T€ | — | 9,902T€ | 9,902T€ | | 1200 x E x E | 1 | 15,021T€ | — | 15,021T€ | 15,021T€ | |

Now, [Table 2.7](#tbl-supmart5) starts to become unwieldy. But it looks like the combo density x wealth of the area is important. As we will see in later chapters, regression models and other machine learning approaches essentially do what we did here–only more sophisticated. We are computing conditional means here. Means conditional on a given density and wealth level. More sophisticated models also estimate means (or other statistics) conditional on many possible conditions. The important takeaway here is to realize that we often want to know conditional means in forecasting problems. And as you can see, we can do this to some rudimentary degree just using tables.

We have a problem though: sample size. While It looks like the combo density x wealth of the area is important, we are getting into tricky territory. A lot of the cells in [Table 2.7](#tbl-supmart5) contain only one supermarket. You can also see that because of N = 1, most do not have a standard deviation associated with them. You can think of the cells with N = 1 as “anecdotes”. Maybe, if we would have more supermarkets in the “1200 x E x E” category, the average would be very different, and show significant variation. We do not know, we only have one observation. In technical terms, the average is an imprecise estimate of the expected value if the number of observations is small. With N = 1, we basically only have an anecdote. This is even more important for standard deviation and the percentiles. Measures of variation are even more dependent on a decent amount of observations. All this means, that while it seems that density x wealth explains a large amount of variation, we really do not know that for certain. We do not have enough supermarket observation to say much with certainty here. If we would use those cell averages to make predictions, we need to be very cautious. These averages are only “anecdotal”—we cannot have confidence that the pattern in averages represents a generalizable pattern. Thus, using cell averages would probably not yield reliable forecasts. We quickly ran into the issue that for precise extrapolations we need enough data to draw conclusions from.

Still, we need to make a decision whether to further explore building a supermarket at the offered location or not. And we need to decide how to proceed. We have not yet actually proposed any alternatives—any forecasts. From our previous analysis we can draw tentative conclusions: density and wealth in the area most likely matter for supermarket revenues. What next? Remember, guide your analysis by specifying the goal clearly and asking a succession of questions that are always aimed at bringing you closer to the goal. The following questions remain:

* What about lifestyle?
* Do we still want to use this model for forecasting, or do we need to expand or adjust it?
* Given the model we use for forecasting, how do we get the data for it? We need to collect the necessary data for the new location to make a forecast.

We have not yet tested whether lifestyle matters. We do not have data on that, so if we think this is important, then we need to think about whether collecting the data is worthwhile given our decision criteria *and* given how precise we would be able to forecast without. When discussing predictive analytics in session three we will learn more tools that help us in making such decisions.

Density and wealth seem to explain a decent amount of variation in revenues; we probably want to use the two determinants in our forecast model. But we are quite uncertain whether we can use the averages from our existing categories. One problem might be that we only have categorical data. We have five density categories and four wealth categories. If we could collect data on things like average income and amount of households in a certain radius around our markets, we could potentially get more precise estimates.[[2]](#footnote-65)

So carrying on with density and wealth, we have a few alternatives. In all we have to also think about how we collect the necessary data on the determinants for the new location:

* We ignore lifestyle, categorize the new location into the density and wealth categories and use the average as our forecast, ignoring what we discussed before. For this we need to collect data:
  + Infer this from public statistics if available
  + Pay someone to survey the location and thus generate the data we need.
* We investigate the importance of lifestyle before making any further decisions.
* We ignore lifestyle but explore how we can get better data, ideally continuous data, on density and wealth, then use a linear model (e.g., a regression model) if the relation between the variables turns out to be as roughly linear as [Table 2.5](#tbl-supmart3) and [Table 2.6](#tbl-supmart4) suggest. Then, we again need to do the same for the new location and apply the fitted model to the new data.

Which of these three paths is the best to follow next (or in parallel) depends on our decision criteria, which is why it is important to specify them first. For example, imagine that we are asked to deliver a rough estimate by tomorrow for a quick upper-management brainstorming session. Then you do not have time to collect more information. It might be optimal to extrapolate from the cell averages in [Table 2.7](#tbl-supmart5) to deliver a forecast (including reporting how imprecise this forecast). However, if this is part of the full feasibility analysis of building that supermarket and you have days to finish this (as well as a bigger budget), then spending some time on investigating lifestyle importance is probably worthwhile. The intermediate alternative could be fine too. We ignore lifestyle because density and wealth already explain a lot of variation in revenues across supermarkets. But we invest a bit of time and effort measuring these two predictors with more precision and test whether different forecast (resting on more, but defendable assumptions) helps us get more precise estimates and more readily interpretable confidence intervals. This could be the best combination of expected accuracy, uncertainty, and effort if we have a moderate (time) budget.

In summary, what we have done is to start with a simple model, and worked our way forward. How much variation does our simple model capture? Do the *expected* benefits of extending the model outweigh the costs of doing so? Iterating on these two questions will ultimately lead us to one or more forecasts.

### 2.3.5 Evaluate alternatives

Even if you only end up proposing one alternative (actually a quite common outcome), you still need to evaluate it. Three important questions to answer here:

1. How do the alternatives rank with regard to our decision criteria?
2. What is the uncertainty about the outcome of a decision given a certain alternative is chosen?
3. What are the potential costs of mistakes when choosing a certain alternative?

The first question is relevant whenever we have multiple alternatives and multiple decision criteria left. Then we need to weigh the importance of different criteria – We need a weighting scheme.

The second question concerns the uncertainty with respect to the outcome associated with a chosen alternative. This question is often tricky to answer and hence ignored. Going back to the supermarket example, assume we were able to ascertain that the new location would fall into the size x density x wealth categories “1200 x C x B”. We take the average: 3,290T€ as our forecast of revenues for the new supermarket location. However, the 3,290T€ is based on the revenue of one supermarket. We are basically extrapolating one anecdote for that cell. This makes it very hard for us to quantify the uncertainty/precision associated with 3,290T€ as a forecast. Sometimes that is a K.O. criterion and we would choose an alternative that makes it easier for us to quantify the uncertainty.

Equally important: What are the costs and benefits associated with each alternative solution? Stated another way. What is the cost associated with actual revenues in the new location being different from 3,290T€? And is understating revenues as costly as being overoptimistic and overstating potential revenues in the new location? The question of handling the differential costliness of mistakes can often appear very academic. It is nevertheless important, even if it is often hard to specify the exact costs of mistakes. Because even if you do not quantify the costliness of mistakes, it is often implicitly considered. For example, surveys quite clearly suggest that executives of publicly listed companies rather want to be seen as too pessimistic rather than too optimistic (larger stock price reactions to bad news and greater fear of subsequent litigation). Similarly, the “cost” of a patient adversely reacting to a drug is surely much higher than if a drug fails to help the patient. Thus, while it might be hard to quantify the exact costs, nevertheless many business decisions consider that not all mistakes are to be treated equally.

If one were able to quantify the costs of making a mistake, then one can compose what is called *loss functions*. And even if not, loss functions are a helpful mental model to think about the costs of making errors. [Figure 2.4](#fig-lossfuns) plots different archetypes of loss functions.

|  |
| --- |
| Figure 2.4: Different types of loss functions |

The blue line in [Figure 2.4](#fig-lossfuns) shows a (normalized) absolute loss function. It is symmetric, meaning predictions that are too low are as bad as predictions that are too high. The slope is also the same at all points, meaning larger error magnitudes are getting worse at the same “rate”. An asymmetric loss function is shown via the red line in [Figure 2.4](#fig-lossfuns). It shows a root-squared log loss function. It has the property that predictions that are too high are not as bad. In contrast, predictions that are too low have a higher loss. Plus larger negative errors get exponentially worse. You really do want to have a large miss on the downside with this loss function. A too-low forecast of 10 with a true value of 50 has an absolute error of 0.8 but a root-squared log error of 1.53—nearly twice as costly.

Again, we do not mean to suggest that you need to define loss functions for all your problems. We stray into scenario analysis territory here. But you should think about what errors will be more or less costly for your business—and decide accordingly! For example, in our supermarket forecasting example, being overly optimistic about the revenue prospects would mean a bad investment and lost money. Being overly pessimistic might result in not investing and potentially losing out on a profitable opportunity as well as a loss in market share if a competitor occupies that area. It depends on the competitive situation which mistake is worse. You might just want to assign lower decision weights to forecasts that have a higher chance to be optimistic (or pessimistic, depending on the situation).

Also, depending on what analytical methods you or your data scientists employ for predictions, you will implicitly use a loss function. That is because loss functions like these are at the heart of any machine learning algorithm. The “learning” is guided by a cost function for making mistakes. For example, an ordinary least squares regression, as the name suggests, has a squared loss function[[3]](#footnote-71). So, if you use any machine learning method, be aware that you have also chosen a loss function that comes with its predictions.

### 2.3.6 Select an alternative

Once you have decided on a weighting scheme of different alternatives, considered the costs of mistakes of pursuing an alternative, and scored each alternative according to the chosen criteria, then the next step would be to decide based on a final selection rule. For important decisions with sizable uncertainty, it is also important to do a scenario analysis. Scenario analysis is one way to quantify uncertainty and the costs of mistakes. In such situations, a common way to deal with the sizable uncertainty inherent in any forecast is to see how robust your alternative performs in the worst scenario. Two common rules that deal differently with uncertainty as expressed in scenarios are:

* **Weighted averages.** Pick the alternative with the highest weighted net benefit (score), with weights according to how probable different scenarios are. In this case, you choose the alternative that performs best in the most likely scenarios but also takes outliers into account.
* **Maximin/Minimax.** Also called the criterion of pessimism. Maximin is the maximum of a set of minima. According to this rule, we pick the alternative that is expected to yield the largest of a set of minimum possible **gains** (net benefit). You can think of this as answering the question “Which alternative does best in the worst scenarios of each alternative you can think of.” Minimax is the same just phrased in terms of **loss** (very often used in machine learning cause it can be tied directly to loss functions). You pick the option that minimizes the loss across the worst-case scenarios for each alternative.

There are other rules, of course, but these two suffice to illustrate the general problem. For complex and important problems, a scenario analysis is a common way to deal with the sizable uncertainty inherent in any forecast and a sensible decision rule seeks a solution that is robust to different states of the world.

## 2.4 Types of analyses

In the examples above we have already touched upon some types of analysis that are commonly distinguished. We will broadly distinguish four types of analysis.

* Descriptive: What happened in the past
* Diagnostic: Why did it happen?
* Predictive: What will happen?
* Prescriptive: What should we do?

### 2.4.1 Descriptive Analysis — Identifying data patterns

Descriptive analytics examines what happened in the past. Our decision steps “detect the problem” and “identify the problem” are very much reliant on well-executed descriptive analysis. That is because descriptive analysis is where we examine data about past events to spot patterns and trends in the data. A good descriptive analysis takes the raw data and aggregates and summarizes it in just the right way to isolate the pattern that reveals the insight we are looking for. In practice, it is the core of most businesses’ analytics, simply because it can already answer many important questions like: “How much did orders grow from a region?” or “How has client satisfaction developed, and what might be reasons for the trend?”

Descriptive analysis is powerful. It is often *exploratory*, requiring creativity and business expertise. But even for more complex decisions, it is a helpful first step for decision-makers and managers. You can go a long way in analyzing data from past events. Still, once we spotted relevant patterns in the past, it’s up to us to ask how or why those patterns arise and develop adequate responses to it. For example, in the supermarket example, we started building a mental model to *explain* the variation in revenues across supermarkets of different sizes. While we did so using simple descriptive statistics we also did some basic *diagnostic analysis*. We will cover descriptive analysis in [Chapter 3](#sec-datapat).

### 2.4.2 Diagnostic Analysis — Explaining patterns

Diagnostic analysis is more advanced and tries to find answers to the question “Why did it happen?” This is the sort of question a lot of academics are trained to answer, using for example the hypothesis testing framework. We put causal analysis and experiments into this category too. But often, you can already get quite far and rule out some explanations by simple data mining and correlation analysis. We will cover diagnostic analysis in [Chapter 4](#sec-diag).

### 2.4.3 Predictive Analysis — Predicting the future

Just as the name suggests, predictive analysis is about making forecasts (i.e., predicting likely outcomes). This is effectively done by comparisons and extrapolation. Even though the analysis methods become more complex (and can become very complex fast), any forecasting method still extrapolates from past data to make predictions. Based on past patterns, we predict what future data could look like (e.g., by computing probabilities, fitting trends, etc.).

Statistical modeling or machine learning methods are commonly used to conduct predictive analyses. In our supermarket example we did not do so, but we actually used the basic logic behind some machine learning approaches: we looked at similar observations (supermarkets) to the one we try to predict future sales for and extrapolated from what we observed in the past. We will cover predicitve analysis in [Chapter 5](#sec-pred).

### 2.4.4 Prescriptive Analysis — Finding the best action

Prescriptive analysis is complex and probably the most advanced of the three analysis types. There also seems to be two different definitions for it. Sometimes it is explained as “Understand why future outcomes happen”, which basically means: causal analysis.[[4]](#footnote-78) A different definition—and the one we adopt—is that predictive analysis revolves around the question: “What should happen?”. It encompasses the later parts of our decision-making process and encompasses all the analysis methods used to help come up with a sensible decision. For example, evaluating and optimizing between different alternative solutions to choose the best one is part of a prescriptive analysis. An example would be: Calculating client risk in the insurance industry to determine what plans and rates are the best to offer to a particular account. Decision modeling and expert system are parts of prescriptive analysis.

## 2.5 Uncertainty in decision-making

At the end of this chapter, we want to offer one last but important note of caution. Consider the following two quotes:

In contrast to complete rationality in decision-making, bounded rationality implies the following (Simon, 1982, 1997, 2009):

1. Decisions will always be based on an incomplete and, to some degree, inadequate comprehension of the true nature of the problem being faced.
2. Decision-makers will never succeed in generating all possible alternative solutions for consideration.
3. Alternatives are always evaluated incompletely because it is impossible to predict accurately all consequences associated with each alternative.
4. The ultimate decision regarding which alternative to choose must be based on some criterion other than maximization or optimization because it is impossible to ever determine which alternative is optimal.

–source: Lunenburg (2010)

and

As we know, there are known knowns – these are things we know we know. We also know there are known unknowns – that is to say, we know there are some things we do not know; but there are also unknown unknowns – the ones we don’t know we don’t know…. It is the latter category that tends to be the difﬁcult one. (Donald Rumsfeld, Department of Defense News Brieﬁng, February 12, 2002.)

The important message we want to reiterate is that—no matter how good the data are—there will always be uncertainty. If we ignore uncertainty, our decision-making can be far off the mark. A classic example is the channel tunnel revenues projections, as documented by Marchau et al. (2019):

The planning for the Channel Tunnel provides an illustration of the danger of ignoring uncertainty. Figure 1.1 shows the forecasts from different studies and the actual number of passengers for the rail tunnel under the English Channel. *The competition from low-cost air carriers and the price reactions by operators of ferries, among other factors, were not taken into account in most studies.* This resulted in a signiﬁcant overestimation of the tunnel’s revenues and market position (Anguara 2006) with devastating consequences for the project. Twenty years after its opening in 1994, it still did not carry the number of passengers that had been predicted. – (Marchau et al. 2019, 2)



Channel Tunnel Projections versus Actual. Source: Anguera (2006)

There are basically three big sources of uncertainty to deal with: Model uncertainty, data quality, and residual uncertainty. We want to discuss these types for two important reasons. For one, trying to get a handle on what you do not know or what you are uncertain about is important for making decisions that take many eventualities into account. For another, your data collection efforts should naturally be geared towards where you can reduce uncertainty the most.

1. **Model uncertainty.** Sometimes also called system uncertainty. We like the term model uncertainty because we want to include all the implicit mental models you might have—but never fully spelled out—about how the world works. All our (mental) models are simplifications and sometimes we simplified away from something important. In the channel tunnel example, the planners did not consider important competitive dynamics for travel routes to and from England. Another way of saying the same thing is: There is always the possibility you are operating under wrong assumptions. Unknown unknowns reside here.
2. **Data quality.** Say you have decided on one model like the one in [Figure 2.3](#fig-dag-sm) (or maybe a few alternative models) of the revenue generation process for supermarkets. You have a model of what drives the number of customers per month, the average basket size they buy. Now you need to start forecasting how these drivers will develop in the future. These forecasts will be based on assumptions and extrapolations from data. The quality of the data will be an important (but far from the only) driver of how much you can trust these forecasts. Common statistical methods provide means of quantifying aspects of this uncertainty and we will talk about this in more detail in [Chapter 5](#sec-pred). However, data quality has numerous facets, some hard to get a good handle on. Is data selected, censored, or missing in a non-random way, is current data even representative of the expected future dynamics, etc.?
3. **Residual uncertainty.** This is the biggest and often most difficult to grasp source of uncertainty. It arises from all the aspects of the problem we have either not included in the model or could not measure. Take a coin flip. It is in principle a well-understood physical mechanism. If we could measure all aspect of it, wind, angle, and force of the flip, etc., then we should be able to forecast quite accurately whether a coin lands heads or tails every time. However, even in this simple case, it is impossible to measure all aspects of the problem without any measurement error. Thus even here there will always be residual uncertainty about how the coin lands. In real-world problems, it is also common that certain dynamics are either too complicated (some competitive dynamics) to be forecast with any degree of precision, or simply unknown to us. Thus in many situations, it is impossible to quantify the residual uncertainty and we have to think about how to make sensible decisions in these cases too.[[5]](#footnote-84)

# 3. Identifying Data Patterns

## 3.1 Learning goals

1. Understand the importance of identifying data patterns
2. Evaluate the relevance of visualization and statistical techniques for analyzing data patterns
3. Explain and tackle several data challenges

## 3.2 Definition of data patterns

A pattern is a recognizable feature or tendency that can be observed in a dataset. Data patterns are recurring or consistent structures or relationships in data that can provide valuable insights into trends, relationships, and underlying structures, and help to carve out areas for further investigation using more advanced analysis.

## 3.3 Importance of identifying data patterns

Identifying data patterns is the starting point in data-driven decision-making because it lays the foundation for understanding the relationships, trends, and insights hidden within the data. This understanding is crucial for making informed, evidence-based decisions, developing effective strategies, and gaining a competitive advantage as a business.

One of the most important reasons for identifying data patterns is to gain insights into trends and relationships. For example, by analyzing sales data, we can identify patterns in customer behavior, such as seasonal fluctuations or changes in buying preferences. Businesses can further identify patterns in demographics related to buying decisions. This information can be used as input to optimize pricing and product offerings and improve customer service. In financial data, data patterns may reveal recurring trends in revenue, expenses, or cash flow. These patterns can help businesses forecast future financial performance, manage risks, and identify opportunities for growth. In operational data, businesses can identify patterns in production processes, supply chain logistics, and inventory management. Data patterns may reveal recurring inefficiencies, bottlenecks, or waste. These patterns give a starting point for businesses to optimize production schedules, reduce costs, and improve efficiency. In marketing data, data patterns may reveal recurring consumer behavior, such as responses to advertising, preferences for certain products or services, or responses to different pricing strategies. Building on these findings, it can help businesses to develop more effective marketing campaigns, improve customer acquisition and retention, and increase revenues. In employee data, data patterns may show insights about employee engagement and performance. This insight can help to identify areas for improvement and develop employee programs that help to increase engagement and performance.

Identifying data patterns is therefore essential to get a first glimpse of the underlying trends and relationships in data, and to generate hypotheses and guide further analysis. More specifically, data patterns help to establish hypotheses about the causes or drivers of the observed patterns. These hypotheses can guide more comprehensive analysis and research toward establishing causality (if desired). Data patterns and relationships identified during the initial analysis can inform the development of diagnostic and predictive models, which can then be used to forecast future trends, make recommendations, or identify optimal solutions to problems. By understanding the patterns in historical data, we can develop models that can help businesses to anticipate trends and „look into the future“, and make informed decisions already now. For example, by analyzing sales data, businesses can identify patterns in consumer behavior, which can help them to forecast demand and adjust production accordingly. This can help to reduce waste and optimize inventory management, which can lead to cost savings and increased profits.

Another important reason for identifying data patterns is to detect anomalies or outliers. Anomalies are data points that deviate significantly from the norm and can be indicators of errors or fraud. By analyzing financial data, businesses can identify patterns in transactions, which can help them to detect fraudulent activities and take corrective action to prevent further issues. Anomalies and outliers may also be the outcome of data quality issues that need further investigation in order to create reliable data for data-driven decision-making.

In summary, identifying data patterns is an important first step in data-driven decision-making because it enables decision-makers to get a first picture of underlying relationships in a company’s data. This process lays the groundwork for making informed, evidence-based decisions that can ultimately drive better outcomes for organizations.

While going through the chapter, please keep in mind that the examples form a basis for further investigations. Identifying data patterns using visualizations or basic statistics is usually the first step that paves the way for more advanced analysis which justifies the term “data-driven decision-making”. Of course, you could stop your investigation after analyzing data patterns and make your own assumptions about the origins of the pattern and what it implies for the company’s future, but you would miss out on the chance of what the true purpose of data-driven decision-making is: improving decision-making in an organization.

## 3.4 Examples of data patterns

### 3.4.1 Customer data

Suppose a business collects data on its customers’ purchasing habits, including the types of products they purchase and the frequency of their purchases. After analyzing the data, the business may identify a recurring pattern of customers who purchase a particular type of product every two weeks. Advanced investigation reveals that this pattern is due to the product’s shelf life or expiration date, as customers consume the product within two weeks of purchase. Armed with this knowledge, the business can adjust its inventory and ordering processes to ensure that they always have the product in stock, or consider promoting the product to encourage more frequent purchases.

Alternatively, the business may identify a pattern of customers who only purchase during certain times of the year, such as around holidays or during seasonal events. After further investigation, they may realize that these customers are motivated by seasonal promotions or discounts. Based on this analysis, the business can adjust its marketing and sales strategy to target these customers during those times of the year.

In both cases, identifying data patterns in consumer data is the first step to support the business to optimize its product offerings and increase sales.

### 3.4.2 Financial data

A business gathers data on its expenses, including the types of expenses and the amounts spent in each category. Data analysis may indicate a recurring pattern of increased expenses in a particular category, such as travel expenses. After diving deeper into the data, it may become clear that this pattern emerges due to a particular reason, such as frequent business trips or a lack of cost-saving measures. The business can adjust its policies to reduce expenses in that category, such as using video conferencing instead of traveling for meetings or negotiating discounts with travel vendors.

In another case, the business may identify a pattern of excessive expenses in multiple categories that exceed the budgeted amounts. More advanced analysis shows that this pattern is due to the lack of oversight and accountability in expense management. The business can establish more robust expense management policies and implement expense tracking systems to monitor and control expenses. At the same time, this may also be a signal that overall expenses are increasing and that budgets (for certain categories) have to be updated.

As you see, looking for patterns in expense data helps with the control of costs.

### 3.4.3 Operational data

Imagine a manufacturing business collects data on its production processes, including the time it takes to complete each stage of production and the number of defects per batch. After analyzing the data, the business may identify a pattern of increased defects during a particular stage of production. Upon further investigation, the business may discover that this pattern is due to a particular piece of equipment or a step in the production process that is causing the defects. Now the business can adjust its production processes, such as improving the equipment or modifying the process to reduce the likelihood of defects.

Alternatively, the business may identify a pattern of bottlenecks or delays during a certain stage of production that is causing the entire process to slow down. After further investigation, they may realize that this pattern is due to a lack of resources. The business can adjust its resources or optimize its supply chain to increase production efficiency and reduce delays.

Also here, monitoring production data and identifying data patterns helps the company to improve its production processes and reduce waste.

### 3.4.4 Marketing data

A company is collecting data on its marketing campaigns, including the channels used for advertising and the response rates from customers. Data analysis may indicate a recurring pattern of increased response rates from customers who receive personalized email campaigns. Further analysis shows that this pattern is due to the increased relevance of personalized email campaigns to the customer’s interests and preferences. Knowing this, the business can adjust its marketing strategy to focus more on personalized email campaigns or consider other personalized marketing tactics, such as targeted social media advertising.

In another example, the business may identify a pattern of decreased response rates from customers who receive marketing messages at certain times of the day or week. After further investigation, they may realize that these patterns are due to the customers’ work schedules or leisure time. The business can adjust its marketing schedule to better align with the customers’ availability and preferences.

In both cases, the pattern in marketing data is a first insight that can help the business to improve its marketing campaigns and increase its response rates and customer engagement.

### 3.4.5 Employee data

Imagine a company that has been tracking employee turnover. The company may notice a data pattern where certain departments or teams have higher turnover rates than others. By further analyzing the data with advanced methods, the company may find that employees who have been with the company for a shorter period of time are more likely to leave, or that employees who have not received promotions in a certain timeframe are more likely to leave. Armed with this information, the organization can take steps to address the underlying issues, such as increasing promotion opportunities or improving onboarding processes for new hires.

Alternatively, the organization may use employee satisfaction surveys to gather data on how employees feel about their jobs and the company as a whole. After analyzing the data further, they may notice a pattern where employees who have flexible work arrangements, such as the ability to work from home or adjust their work hours, report higher levels of job satisfaction. Based on this additional analysis, offering flexible work arrangements may be considered as an effective strategy for improving employee satisfaction, and the organization may decide to implement or expand these arrangements as a result.

Both examples show how the identification of simple patterns can kick off analyses to decrease turnover and increase job satisfaction.

## 3.5 Techniques for identifying data patterns

### 3.5.1 Start with the right mindset

First and foremost, it’s important to approach the data with a sense of curiosity and a desire to understand what it’s telling you. This means asking questions and exploring different angles, rather than simply accepting what you see at face value. By staying curious, you can uncover unexpected insights that might not be immediately obvious. At the same time, it’s important to maintain objectivity when analyzing data. Avoid making assumptions or jumping to conclusions based on preconceived notions. Instead, let the data guide your thinking, and be willing to challenge your own assumptions if the data suggests a different interpretation. This can be particularly challenging if you have pre-existing beliefs about what the data should show, but it’s important to keep an open mind and let the evidence speak for itself. It can be tempting to latch onto a particular idea or theory and look for evidence to support it, but this can lead to confirmation bias and blind you to other potentially important insights. Developing a structured approach to data analysis can also be helpful. By using a systematic approach, you can ensure that you’re analyzing the data in a consistent and rigorous manner, which can help you identify meaningful patterns more easily.

When analyzing the data, it is key to put yourself into the “shoes of the data-generating process”. A data-generating process helps business analysts to understand the underlying mechanism that produces the data they are working with and defines the factors that affect the data and how they contribute to its creation. For example, if you want to look into customer behavior, imagine yourself how you act as a customer. Imagine that you are making certain decisions and take actions, thereby actually creating the data (assuming that there is some type of variable measured and recorded that will reflect your decisions and actions). The figure below shows many factors that can influence our buying behavior.

|  |
| --- |
| Figure 3.1: Buyer decision process |

For example, what aspects would make you enter a drugstore and what aspects would keep you away from the store? Large price boards in the windows, commercial boards standing on the street, or employees handing out samples? What kind of aspects make you choose one hotel, but not the other? Price, proximity to the city center, or the cleanliness rating? And again, then hopefully you have data in your datasets that reflect or at least approximate these aspects such that you can use the data for your analysis. These examples are a bit easier as all of us are customers at some point. However, when you want to analyze bottlenecks in a production process, you probably have to consult your colleagues involved in the production. But even for the customer behavior example, it may be wise to contact colleagues from marketing to get a better picture of decisions customers typically make. You can get additional information from professional magazines or academic papers to put yourself into the data-generating process. Finally, it’s important to be patient when analyzing data. Finding patterns in data can take time and require persistence. Don’t expect to find all the answers right away, but be willing to put in the effort to uncover insights that can lead to better business decisions. s

### 3.5.2 Visualization techniques for identifying patterns

Now that we have the data, we want to start uncovering the story that the data tells us. Visualizing data and looking for distinct patterns is a way to accomplish this. There can be details that, when the data is given in tabular form, we are unable to perceive but which are made clear to us through visualization. The human brain is wired to process and interpret visual information quickly, and data visualization leverages this natural ability to help us understand complex data more easily and effectively. By using charts, graphs, and other visual representations of data, we can identify relationships, trends, and outliers that might be difficult to spot using only numerical summaries. Visualization can also help us communicate insights more clearly and effectively to others, making it an essential tool in data-driven decision-making (which is covered at a later moment in this course). Moreover, data visualization allows us to explore multiple variables and data sets simultaneously, enabling us to uncover complex relationships and patterns that might be difficult to see through other means. Before jumping into the creation of visualizations, there are two questions relating to the nature and purpose of our visualization which help us decide on the type of visualization (Berinato 2016):

1. Is the information *conceptual* or *data-driven*?
2. Am I *exploring* something or *declaring* something?

|  |
| --- |
| Figure 3.2: Nature and purpose of visualization. |

Source: https://hbr.org/2016/06/visualizations-that-really-work

The first question is quite easy to answer given the title of our course – the nature of our information is based on data. We are not exploring concepts or ideas that we want to visualize, but we want to use data.

The second question gets at the purpose of the visualization: do you want to provide information (declarative) or try to figure something out (exploratory)? Currently, we are still in the phase of figuring out things with the help of data visualization, so by putting the answers to the two questions together, we find ourselves in the bottom right corner, the visual discovery. Data visualization is a critical component of exploring data patterns because it allows us to visually identify relationships and patterns that may not be apparent from simply looking at raw data. Only at a later stage, we want to communicate information (such as the solution to the case), being in the everyday dataviz quadrant.

For the purpose of exploring data, we look at different types of patterns, each coming with typical forms of visualization.

#### 3.5.2.1 Distributions

Before looking at the relationships and patterns of our data and variables, it may be useful to get a “feeling” for the key variables in the dataset and visualize some characteristics of these variables. For these visual insights, we typically rely on histograms, bell curves, and box plots to understand the distribution of each variable.

A **histogram** is a type of graph that displays the frequency distribution of a continuous or discrete variable, such as sales, revenue, customer demographics, or product performance. By grouping the data into bins or intervals, histograms provide a visual representation of the underlying patterns in the dataset. Histograms are commonly used to identify the shape and spread of a dataset.

For example, let’s say you work with a marketing analyst at your company and you analyze the distribution of customer ages in your customer database. You create a histogram where the x-axis represents age intervals, and the y-axis represents the count of customers falling within each age interval.

s1 |>  
 ggplot(aes(x = Age)) +  
 geom\_histogram(  
 # binwidth = 5  
 breaks = seq(20, 80, 5)  
 ) +  
 scale\_y\_continuous(  
 expand = expansion(mult = c(0, 0.02)),  
 n.breaks = 6  
 ) +  
 scale\_x\_continuous(breaks = seq(20, 80, 5)) +  
 labs(  
 y = "Count by bin",  
 x = "Age",  
 ) +  
 theme(  
 panel.grid.major.x = element\_blank()  
 )

|  |
| --- |
| Figure 3.3: Histogram example |

By examining the shape of the histogram, analysts can identify trends and patterns, such as whether the distribution is symmetric, skewed, or has other notable characteristics. Histograms can help analysts to visualize various statistical measures, such as the mean, median, mode, and measures of dispersion (e.g., variance or standard deviation) (see 3.5.3). In the example above, you see that it seems that customer age follows a normal distribution (see also bell curve). You also see whether there is any interesting variation in the data useful for further statistical analysis. E.g., if all age bins would contain the same number of customers, there is no age variation that can be used to explain, e.g., customer behavior. However, we do not need to worry in this example as we can identify variation from the age bins. Furthermore, histograms can serve as a basis for fitting parametric models, such as normal, log-normal, or exponential distributions, to the data. These models can then be used for predictive analytics, forecasting, and other advanced statistical analyses, which can provide valuable insights for decision-making and strategic planning. Additionally, from a business decision perspective, you can identify any patterns that may exist in the age distribution of your customer base. In this example, you see that most customers are between 48 and 52 years old. This information can help guide marketing efforts towards the specific age group that is most likely to purchase your products or services. At the same time, if you believe that your product or service should theoretically be very interesting for people of the age around 35, you could further investigate reasons why people in that age category do not buy the product as much. Additionally, histograms can be used to identify any potential outliers or unusual patterns in the customer age data. For example, if the histogram reveals a sudden spike in a specific age group, it might indicate a data entry error, an anomaly in the dataset, or an emerging trend that requires further investigation. Another application of histograms in analyzing customer age data is to compare the age distribution of different customer segments, such as those who make online purchases versus those who shop in-store. By comparing the histograms, a business can identify differences in customer preferences and behaviors, allowing it to fine-tune its marketing and sales strategies accordingly.

Similar to a histogram, you may come across a **bell curve** (also known as normal distributions or Gaussian distributions).

s1 |>  
 ggplot(aes(x = Age)) +  
 geom\_density(  
 color = "red",  
 fill = "red", alpha = 0.1  
 ) +  
 scale\_y\_continuous(  
 expand = expansion(mult = c(0, 0.02)),  
 n.breaks = 6  
 ) +  
 scale\_x\_continuous(breaks = seq(20, 80, 5), limits = c(15, 85)) +  
 stat\_function(fun = dnorm,  
 geom = "area",  
 color = "dodgerblue",  
 fill = "dodgerblue",  
 alpha = 0.4,  
 xlim = c(0, quantile(age, 0.16)),  
 args = list(  
 mean = mean(age),  
 sd = sd(age)  
 )) +  
 stat\_function(fun = dnorm,  
 geom = "area",  
 color = "dodgerblue",  
 fill = "dodgerblue",  
 alpha = 0.8,  
 xlim = c(quantile(age, 0.16), quantile(age, 0.84)),  
 args = list(  
 mean = mean(age),  
 sd = sd(age)  
 )) +  
 stat\_function(fun = dnorm,  
 geom = "area",  
 color = "dodgerblue",  
 fill = "dodgerblue",  
 alpha = 0.4,  
 xlim = c(quantile(age, 0.84), 80),  
 args = list(  
 mean = mean(age),  
 sd = sd(age)  
 )) +  
 annotate(  
 "line",  
 x = c(quantile(age, 0.16), quantile(age, 0.84)),  
 y = c(0.01, 0.01)  
 ) +  
 annotate(  
 "text",  
 x = mean(s1$Age),  
 y = 0.011,  
 size = 3,  
 label = "One standard deviation (68%)"  
 ) +  
 labs(  
 y = "Density",  
 x = "Age",  
 ) +  
 theme(  
 panel.grid.major.x = element\_blank()  
 )

|  |
| --- |
| Figure 3.4: Bell Curve example |

Bell curves are relevant for further statistical analysis, as they come with specific properties that can simplify calculations and provide a foundation for hypothesis testing and other statistical procedures. When data follows a normal distribution, the mean, median, and mode are equal, and approximately 68% of the data falls within one standard deviation of the mean, 95% within two standard deviations, and 99.7% within three standard deviations.

A **box plot**, also known as a box-and-whisker plot, is a type of chart that displays the distribution of data by showing its median, quartiles, and outliers.

age2 <- c(13, age)  
q\_ages <- quantile(age2, c(0.25, 0.5, 0.75))  
lb <- q\_ages[2] - 1.5 \* IQR(age2)  
ub <- q\_ages[2] + 1.5 \* IQR(age2)  
data.frame(Age = age2) |>  
 ggplot(aes(y = Age)) +  
 geom\_boxplot(  
 width = 0.2,  
 fill = "dodgerblue",  
 outlier.colour = "orange", outlier.shape = 19, outlier.size = 3  
 ) +  
 scale\_x\_continuous(breaks = seq(-0.2, 0.2, 0.1), limits = c(-0.2, 0.2)) +  
 labs(  
 x = NULL,  
 y = "Age"  
 ) +  
 annotate(  
 "text",  
 x = 0.11,  
 y = c(13, lb, q\_ages, 72),  
 label = round(c(13, lb, q\_ages, 72))  
 ) +  
 theme(  
 panel.grid.major.x = element\_blank(),  
 axis.ticks.x = element\_blank(),  
 axis.text.x = element\_blank()  
 )

|  |
| --- |
| Figure 3.5: Boxplot example |

The box represents the interquartile range (IQR, see 3.5.3.1.4), which contains the middle 50% of the data. The box extends from the lower quartile (Q1) to the upper quartile (Q3). The median line is the line within the box that represents the median (Q2) of the dataset. The whiskers (probably coming from cat whiskers) are the lines extending from the box and represent the range of the data, excluding outliers. The whiskers typically extend to the minimum and maximum data points within 1.5 or 3 times the IQR from Q1 and Q3, respectively. Outliers are data points that fall outside the whiskers, typically represented by individual points or symbols. Using an adjusted dataset from the histogram, this box plot indicates a relatively normal distribution given that the median is in the middle and the same length of the quartiles. If the box is short and the whiskers are long, the data are spread out and there may be many outliers. If the box is tall and the whiskers are short, the data are tightly clustered and there may be few outliers. We see that indeed one outlier in terms of age exists, giving reason to further investigate how to deal with it. Alternatively, this could hint a **data quality issue**.

#### 3.5.2.2 Relative proportions

Proportions refer to the comparison of data between different groups or categories, and involves examining how data varies relative to other factors. Imagine a simple count of employees in different age categories, revenues per product line, or marketing expenses per marketing channel. Common visualizations to express proportions are pie charts or tree map charts (amongst others).

A **pie chart** is a type of chart used to display data in a circular graph. It is composed of a circle divided into slices, each representing a portion of the whole. The size of each slice is proportional to the quantity it represents, with the entire circle representing 100% of the data. Pie charts are commonly used to show the distribution of a set of categorical data, where each slice represents a different category or group. The size of each slice is determined by the percentage or fraction of the data that belongs to that category. Pie charts are useful because they provide a clear and intuitive way to compare the relative sizes of different categories. They also allow for the easy identification of the largest and smallest categories and can be effective in communicating the overall pattern or trend in the data.

Z <-  
 table(cut(age, seq(20, 80, 10))) |>  
 as.data.frame()  
Z$prop <- Z$Freq / length(age)  
Z <- Z %>%  
 mutate(csum = rev(cumsum(rev(prop))),  
 pos = prop/2 + lead(csum, 1),  
 pos = if\_else(is.na(pos), prop/2, pos))  
  
  
ggplot(Z, aes(x="", y=prop, fill=(Var1))) +  
 geom\_bar(stat = "identity", width = 1, color = "white") +  
 coord\_polar("y", start=0) +  
 # geom\_text\_repel(aes(y = pos, label = paste0(round(prop\*100), "%")),  
 # size = 3, nudge\_x = 1, show.legend = FALSE) +  
 theme(axis.ticks = element\_blank(),  
 axis.title = element\_blank(),  
 axis.text = element\_text(size = 10),  
 panel.background = element\_rect(fill = "white"),  
 panel.border = element\_blank(),  
 panel.grid = element\_blank(),  
 legend.position = "left"  
 )+  
 scale\_y\_continuous(breaks = Z$pos, labels = paste0(round(Z$prop\*100), "%")) +  
 scale\_fill\_brewer(palette = 1, name = "Age Group")

|  |
| --- |
| Figure 3.6: Piechart example |

Imagine you help out your colleagues from HR. The pie chart above displays HR firm data on age categories and proportion of employees in their respective age categories. This information helps you understand the age distribution of your workforce. You can quickly identify that the age group of employees between 41 and 50 is most frequently observed in that firm. In addition, you see that there is a relatively low inflow of younger people. This is alarming if you think about succession and business continuity, and further analysis is needed in that regard.

A **tree map** chart is a type of chart that displays hierarchical data as a set of nested rectangles. The size and color of the rectangles represent the relative sizes of the different categories or data points, with larger rectangles representing larger values and different colors representing different categories.

Z <- data.frame(  
 Campaign = c("TV ads", "Magazines", "Billboards", "Social media ads", "Email campaigns", "Influence"),  
 Budget = c(46, 6, 24, 23, 3, 15) \* 1000  
)  
  
  
my\_palette <- scales::brewer\_pal(palette = 1)(9)[4:9]  
  
ggplot(Z, aes(area = Budget, fill = Campaign,  
 label = paste(Campaign,  
 scales::label\_currency()(Budget), sep = "\n"))) +  
 geom\_treemap() +  
 geom\_treemap\_text(colour = "white",  
 place = "centre",  
 size = 10,  
 ) +  
 scale\_fill\_manual(values = my\_palette) +  
 theme(legend.position = "none")

|  |
| --- |
| Figure 3.7: Treemap example |

Suppose the marketing manager of your firm approached you to dig into their expense data. The tree map above includes marketing data on the expenses per marketing channel, with TV ads clearly making up the largest proportion of expenses. This information helps you understand how your marketing expenses are allocated across different channels and you can start further investigations into how far each channel is successful in, e.g., acquiring new customers, and whether the expenses allocated to the respective channels are justified.

#### 3.5.2.3 Ranking

Ranking in the context of data visualization refers to the process of ordering data points based on a specific variable or criterion. This allows for the quick identification of the highest or lowest values. The most common visualizations are bar charts and radar charts (amongst others).

A **bar chart**, also known as a bar graph, is a type of chart or graph that represents data using rectangular bars. The length or height of each bar is proportional to the value it represents. Bar charts can be used to display data in horizontal or horizontal orientation, with the latter usually called a column chart (see also 3.5.2.5).

Z <- data.frame(  
 product = paste("Product", LETTERS[1:5]),  
 failure\_rate = c(0.083, 0.015, 0.02, 0.032, 0.038)  
)  
Z$status = if\_else(Z$failure\_rate > 0.05, "bad", "ok")  
  
ggplot(Z,  
 aes(  
 y = fct\_reorder(product, -failure\_rate),  
 x = failure\_rate,  
 fill = status  
 ),  
) +  
 geom\_col(color = "white") +  
 scale\_x\_continuous(labels = scales::label\_percent(), expand = expansion(c(0, 0.02))) +  
 theme(  
 legend.position = "none",  
 panel.grid.major.y = element\_blank()  
 ) +  
 geom\_vline(xintercept = 0.05) +  
 annotate("text", x = 0.05, y = 1, label = "5% threshold", vjust = 0.5, hjust = -0.1) +  
 labs(  
 y = NULL,  
 x = "Failure rate after x months"  
 )

|  |
| --- |
| Figure 3.8: Barchart example |

Suppose you support a quality control manager at your manufacturing company. The bar chart above includes failure rates per product category. It becomes clear that Product A should be more closely investigated given the seemingly higher failure rate compared to the other products. Notice how small tweaks add to make the punch line of the plot as obivous as possible: the choice of two colors, annotating the plot with a vertical line, and ordering the products so that the “worst” product is at the bottom

A **radar chart**, also known as a spider chart or a web chart, is a type of chart that displays data on a polar coordinate system. Radar charts are useful for comparing multiple sets of data across different categories or variables. Each set of data is represented by a different line or shape on the chart, with each variable represented by a spoke. The length of each line or shape corresponds to the magnitude of the data for that variable, and the different lines or shapes can be compared to one another.

# remotes::install\_github("ricardo-bion/ggradar")  
  
Z <- data.frame(  
 product = c("A", "B", "C"),  
 Price = c(8, 10, 14),  
 Quality = c(10, 11, 9),  
 Popularity = c(8, 9, 11),  
 Durabil. = c(10, 4, 7)  
)  
lcols <- c("#EEA236", "#5CB85C", "#46B8DA")  
ggradar(  
 Z,  
 background.circle.colour = "white",  
 axis.line.colour = "gray60",  
 gridline.min.colour = "gray60",  
 gridline.mid.colour = "gray60",  
 gridline.max.colour = "gray60",  
 gridline.min.linetype = 1,  
 gridline.mid.linetype = 1,  
 gridline.max.linetype = 1,  
 legend.title = "Product",  
 legend.position = "bottom",  
 group.colours = lcols  
)

|  |
| --- |
| Figure 3.9: Radarchart example |

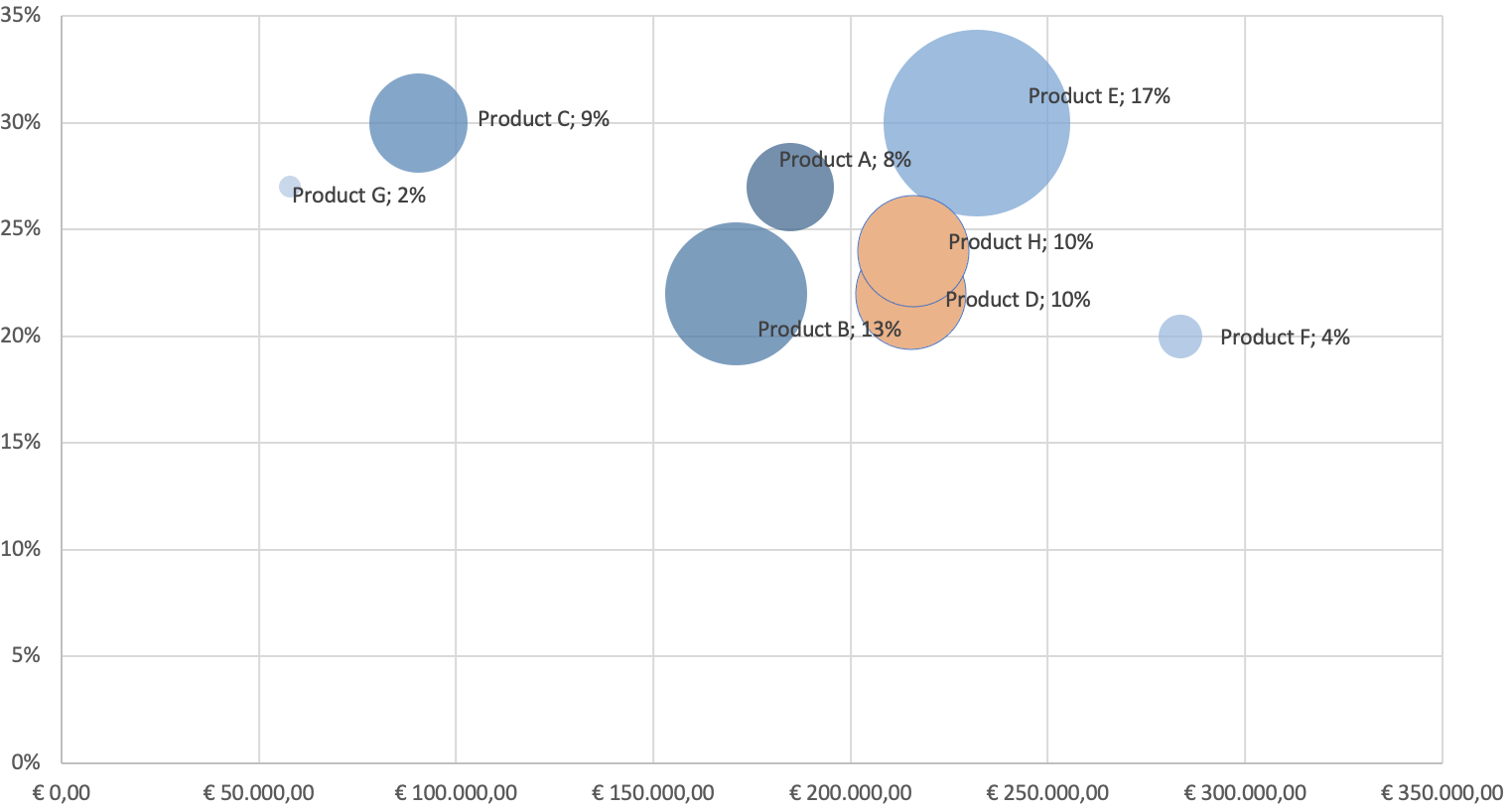
Imagine you help a product manager at your retail company. In the radar chart above, three products are compared based on their properties of price, quality, popularity, and durability. An interesting observation, e.g., is that Product C has the highest price and highest popularity while having the lowest quality. This is interesting because usually, one would expect that a high price and low quality would negatively affect popularity, so it is worth further investigating whether, e.g., a certain minimum requirement for quality is already met and therefore does not impact popularity.

**Another important use of visualizations:** This interesting finding could also hint at a data problem. Visualization can also help us identify data quality issues, such as missing or outlier values, which may be less noticeable when examining raw data. Any data points out of the order may provide a signal for you to review the quality of the data (see also section 3.6).

#### 3.5.2.4 Clustering

Clustering in the context of data visualization refers to the process of grouping data points together based on their similarities or proximity to one another. Common visualizations are bubble charts and scatter plots (amongst others).

A **bubble chart** is a type of chart that displays data points as bubbles or circles on a two-dimensional graph. Each bubble represents a data point and is located at the intersection of two variables on the graph. The size of the bubble represents a third variable, typically a quantitative value, and can be used to convey additional information about the data point. Bubble charts are useful for displaying data with three variables, where two variables are plotted on the x- and y-axes, and the third variable is represented by the size of the bubble. This allows for the visualization of complex relationships between variables and can be used to identify patterns or trends in the data.



Bubble chart

Imagine you help the controller in your retail company. The bubble chart above reflects financial data on each product’s revenue (x-axis), gross profit margin (y-axis), and operating profit margin (bubble size). The bubble chart allows you to quickly identify which products are generating the most revenue (Product F) and gross profit margin (Product C and E), as well as which ones have the highest operating profit margin (Product E). You could also identify any outliers, such as products with high revenue but low gross profit margin (Product F), or a product with low revenue but high operating profit margin (Product C). This information provides a first insight into which products to invest in, which ones to cut back on, and how to optimize your overall financial performance. Furthermore, you can color-code the bubbles based on the product category or production location to find patterns in other common characteristics, or clusters. Suppose that the orange bubbles represent the same production site: this could indicate that the production is similarly efficient as both products have an operating profit margin of 10% (and have similar revenue and gross profit margins). However, as with all initial evidence based on data patterns, this has to be further investigated as there might be many more reasons for such a similar operating profit margin.

A **scatter plot** is a type of graph that uses dots to represent values for two different variables. The position of each dot on the graph represents the values of the two variables.



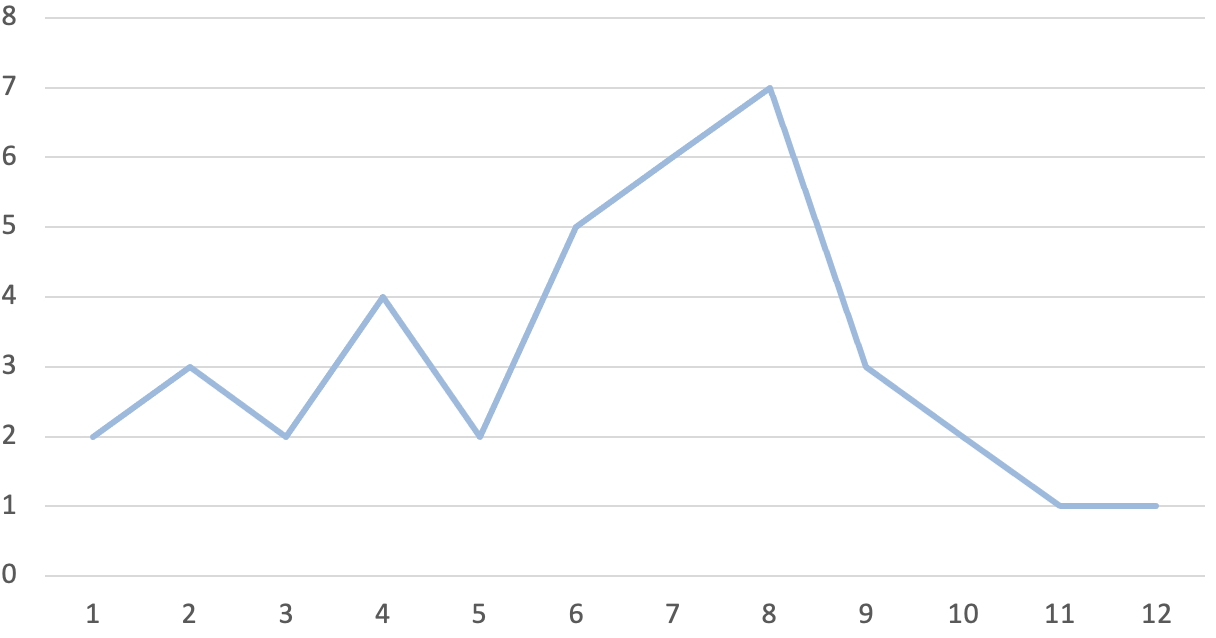
Scatter plot

Suppose you help a plant manager at your manufacturing company, and you want to analyze the relationship between production output and energy consumption. You have data on the daily production output (x-axis) and energy consumption (y-axis) for each production line (PL) over the past year. First of all, you notice that there appears to be a relationship between production output and energy consumption. In addition, several clusters of data points are tightly grouped together. It appears that the product lines in the cluster in the middle may be more energy-efficient than others in their production given that the cluster appears low on the y-axis. Scatter plots are also useful to detect outliers which would imply that one (or multiple) of the data points would appear further away from the other clusters.

#### 3.5.2.5 Change

In the context of data patterns, changes or trends, refer to a general direction or tendency in the data that shows a pattern over time. Trends can be further classified as upward, downward, or flat, depending on the direction of the trend. Line graphs and column charts (amongst others) are commonly used to depict changes and trends.

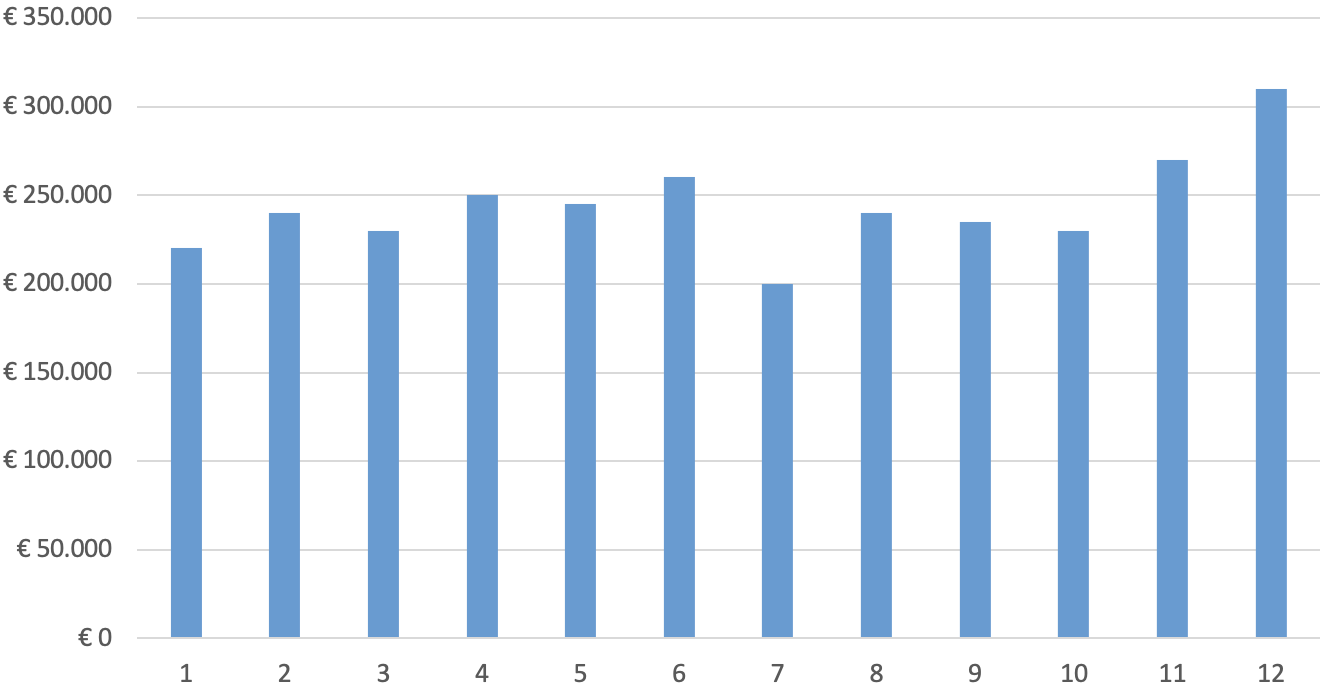
**Line graphs** are a type of graph that display data points as a series of connected points, or markers, that form a line. They are commonly used to show trends or changes in data over time. The x-axis typically represents the time period, while the y-axis represents the value of the variable of interest.



Line graph

For example, let’s say you help out a human resources manager at your company and you want to track changes in employee turnover over time. The line graph above shows the number of employees who have left the company each month over the past year. The x-axis would represent the months of the year, while the y-axis would represent the number of employees who left. You notice a sudden increase in employee turnover in June, July, and August, and you can further investigate the root cause of this upward “summer” trend.

**Column charts**, also known as vertical bar charts, are a type of graph that uses vertical bars to represent data. Each column or bar represents a category or group, and the height of the bar corresponds to the value or frequency of the variable being measured.

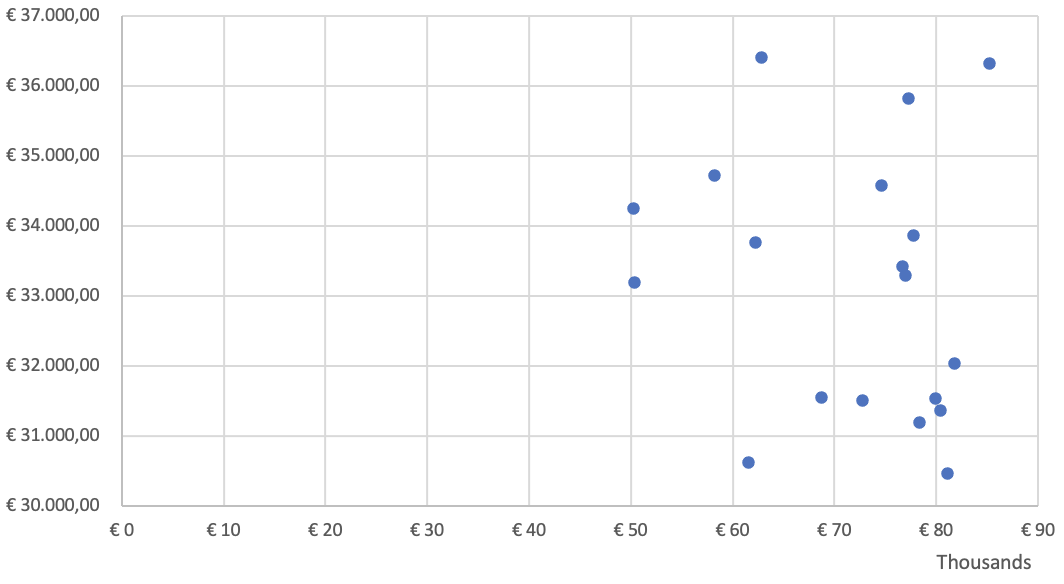


Column chart

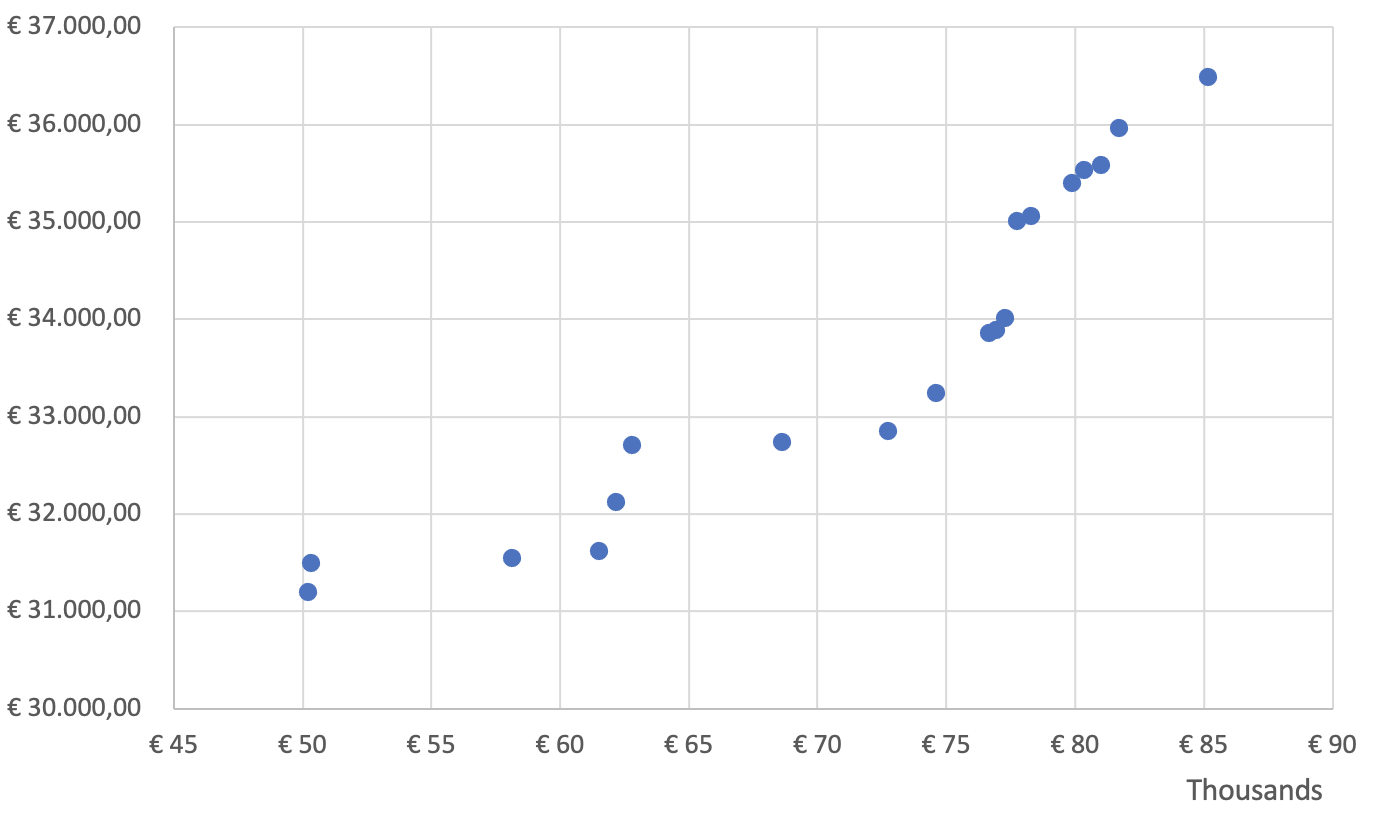
Imagine the marketing assistant approached you to help with some data analysis tasks. You should analyze the total sales revenue over the past year. You can create a column chart where the x-axis represents the different time periods, such as months, and the y-axis represents the total sales revenue for each month. In the above column chart, you quickly identify that sales are uncommonly low in July, and increasing in November and December. By repeating this analysis across multiple years, you may see that this is a consistent pattern across years, and you have a starting point to look closer into the causes of this pattern.

#### 3.5.2.6 Correlations

To identify correlations between variables visually, most often **scatter plots** are used as well.



Scatter plot



Scatter plot

Imagine a friend of yours is a financial analyst, and asks you to check the relationship between revenue and expenses of a company over the past year. You can create a scatter plot where the x-axis represents revenue, and the y-axis represents expenses. Let’s look at two different outcomes. If the outcome looks like the first scatter plot, you can most likely conclude from the visual that there is no correlation between revenues and expenses. If you consider the second scatter plot, it appears that there is a correlation between revenues and expenses. You conclude that both variables move in the same direction, and you can further investigate the underlying reasons.

And that was a journey throughout the most common charts and graphs used to identify data patterns. Yet, in general, it should be noted that some charts can be used for multiple purposes. For example, a bar chart can also be used to identify relative proportions. The number of categories represented in such charts also has an influence on the choice. For example, if a firm has 30 different products, it may indeed be wise to you use a bar chart or column chart to represent those products; a pie chart would be too cramped to read given many “pie pieces” reflecting the products. But some types of visualizations are also simply not useful for some purposes. For example, if we want to identify any changes over time, a pie chart is not going to help us reflecting such changes.

### 3.5.3 Statistical techniques for identifying patterns

Statistical techniques are essential for exploring data patterns because they provide us with a framework for analyzing and interpreting data in a rigorous and systematic way. While visualizations are great approach to start the analysis process, eventually we also want to “crunch” the numbers.

#### 3.5.3.1 Descriptive statistics

##### 3.5.3.1.1 Measures of central tendency (mean, median, mode)

Measures of central tendency are statistical measures that describe the central or typical value of a set of data. They are often used to summarize large datasets and provide insights into the characteristics of the data. The three most common measures of central tendency are mean, median, and mode.

The **mean** () is the arithmetic average of a set of numbers. It is calculated by adding up all the values in a set and dividing the sum by the number of values. The mean is sensitive to outliers in the data, which can greatly influence its value. For example, if we have a dataset of 5, 10, 15, 20, and 25, the mean would be (5+10+15+20+25)/5 = 15. The mean function in Excel is AVERAGE. For example, if you want to find the mean of a range of numbers in cells A1 through A10, you would enter the formula =AVERAGE(A1:A10) in a cell.

The **median** is the middle value in a set of numbers, arranged in ascending or descending order. It is less sensitive to outliers than the mean and provides a better representation of the typical value in a dataset. If the dataset has an odd number of values, the median is the middle value. For example, if we have an uneven dataset of 5, 10, 15, 20, and 25, the median is 15. If the dataset has an even number of values, the median is the average of the two middle values. For example, if the dataset is 4, 6, 8, 10, 12, 14, we would first order the values as 4, 6, 8, 10, 12, 14. The two middle values are 8 and 10, so we would take their average: Median = (8 + 10)/2 = 9. The median function in Excel is MEDIAN. For example, if you want to find the median of a range of numbers in cells A1 through A10, you would enter the formula “=MEDIAN(A1:A10)” in a cell.

The **mode** is the value that occurs most frequently in a set of numbers. It is useful for identifying the most common value in a dataset. For example, if we have a dataset of 5, 10, 10, 15, 20, and 25, the mode would be 10. Excel does not have a built-in mode function, but you can use a combination of functions to find the mode. For example, if you want to find the mode of a range of numbers in cells A1 through A10, you could use the following formula: “=MODE.MULT(A1:A10)”. This will return an array of all modes in the range.

##### 3.5.3.1.2 Measures of variation (range, variance, standard deviation)

Measures of variation are statistical measures that describe the spread or dispersion of a set of data. They are used to determine how much the individual values in a dataset vary from the central tendency. There are several measures of variation, including range, variance, and standard deviation.

The **range** is the difference between the highest and lowest values in a set of data. It provides a simple measure of the variability in a dataset but can be heavily influenced by outliers. For example, if we have a dataset of 5, 10, 15, 20, and 25, the range would be 25-5 = 20. The range function in Excel is simply the difference between the maximum and minimum values in a range. For example, if you want to find the range of a range of numbers in cells A1 through A10, you would subtract the minimum value from the maximum value: “=MAX(A1:A10)-MIN(A1:A10)”.

The **variance** () measures how spread out the data is from the mean. It is calculated by taking the average of the squared differences between each value and the mean of the dataset. The variance is useful for identifying the degree of variability in the data but is not easily interpretable due to its squared units. But it provides a more precise measure of variability than the range and is less sensitive to outliers. For example, if we have a dataset of 5, 10, 15, 20, and 25, the mean is (5+10+15+20+25)/5 = 15. The differences between each value and the mean are -10, -5, 0, 5, and 10. Squaring these differences gives us 100, 25, 0, 25, and 100. The variance is the average of these squared differences, which is (100+25+0+25+100)/5 = 50. The variance function in Excel is VAR. For example, if you want to find the variance of a range of numbers in cells A1 through A10, you would enter the formula “=VAR(A1:A10)” in a cell.

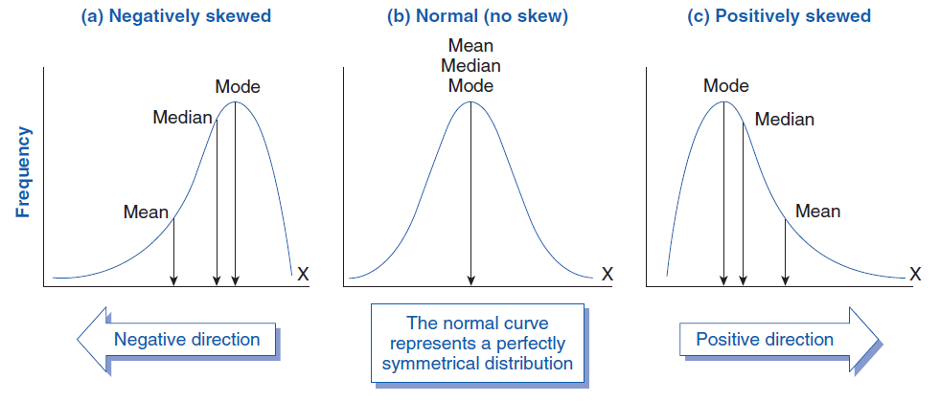
The **standard deviation** () is the square root of the variance. It is a commonly used measure of variation because it is expressed in the same units as the data (and hence more interpretable measure of dispersion than the variance) and provides a measure of how spread out the data is relative to the mean. For example, if we have a dataset of 5, 10, 15, 20, and 25, the variance is 50. The standard deviation is the square root of 50, which is approximately 7.07. The standard deviation function in Excel is STDEV. For example, if you want to find the standard deviation of a range of numbers in cells A1 through A10, you would enter the formula “=STDEV(A1:A10)” in a cell.

In summary, measures of variation such as range, variance, and standard deviation are important tools for identifying patterns in data. They provide valuable insights into the spread or dispersion of a dataset and can be used to detect potential patterns in the data.

##### 3.5.3.1.3 Skewness and Kurtosis

Skewness and kurtosis are two additional statistical measures that are often used to describe the shape and distribution of data.

**Skewness** measures the degree to which a dataset is skewed or distorted from a normal distribution, that is, the degree of asymmetry in a distribution. A normal distribution is a symmetric distribution where the mean, median, and mode are all equal. A dataset with positive skewness has a long tail on the right side of the distribution, meaning there are more extreme values on the right-hand side of the distribution. A dataset with negative skewness has a long tail on the left side of the distribution, meaning there are more extreme values on the left-hand side of the distribution.



Skewness

Source: https://www.biologyforlife.com/skew.html

Skewness is often measured using the coefficient of skewness, which is calculated as: (coefficient of skewness) = (3 \* (mean - median)) / standard deviation. For example: Let’s consider a dataset with the following 9 data points: 2, 4, 4, 4, 6, 6, 6, 8, 10

1. Calculate the mean (μ) of the dataset: μ = (2 + 4 + 4 + 4 + 6 + 6 + 6 + 8 + 10) / 9 ≈ 5.33
2. Calculate the median of the dataset: Since the dataset is ordered, the median is the middle value: Median = 6
3. Calculate the standard deviation (σ) of the dataset: σ ≈ 2.29 (calculated using standard deviation formula)
4. Compute the skewness using the formula: Skewness = (3 \* (mean - median)) / standard deviation Skewness = (3 \* (5.33 - 6)) / 2.29 ≈ -0.88

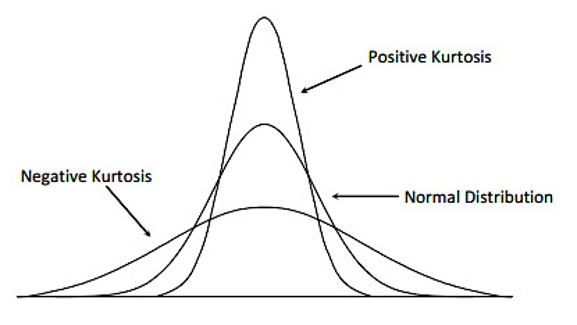
If the coefficient of skewness is positive, the dataset is positively skewed, while if it is negative, the dataset is negatively skewed. If it is zero, the dataset is normally distributed. A rule of thumb states the following: • If skewness is less than −1 or greater than +1, the distribution is highly skewed. • If skewness is between −1 and −.5 or between +.5 and +1, the distribution is moderately skewed. • If skewness is between −.5 and +.5, the distribution is approximately symmetrical.

The skewness function in Excel is SKEW. For example, if you want to find the skewness of a range of numbers in cells A1 through A10, you would enter the formula “=SKEW(A1:A10)” in a cell.

**Kurtosis** measures the degree to which a dataset is peaked or flat compared to a normal distribution. A distribution with a high kurtosis value has a sharp peak and fat tails, meaning there are more extreme values in the tails of the distribution. A distribution with a low kurtosis value has a flatter peak and thinner tails. The calculation is a little more complicated, so let’s save some space and rely on our programs to calculate it.

A normal distribution has a kurtosis of 3. If the coefficient of kurtosis is greater than 3, the dataset has positive kurtosis (more peaked), while if it is less than 3, the dataset has negative kurtosis (more flat).

In summary, skewness and kurtosis are measures that help describe the shape and distribution of a dataset. They provide additional information beyond measures of central tendency and variation, and can help identify patterns and anomalies in the data. The kurtosis function in Excel is KURT. For example, if you want to find the kurtosis of a range of numbers in cells A1 through A10, you would enter the formula “=KURT(A1:A10)” in a cell. Note that Excel’s KURT function returns the excess kurtosis, which is the kurtosis minus 3. Therefore, to get the actual kurtosis value, you need to add 3 to the result of the KURT function.



Kurtosis

Source: https://www.freecodecamp.org/news/skewness-and-kurtosis-in-statistics-explained/

As you can see from the figures, measures of central tendency and variation are typically related to visualizations such as histograms or scatters with lines and markers (additional options in Excel).

##### 3.5.3.1.4 Outliers

Outliers are data points that are significantly different from the other data points in a dataset. As outliers can have a significant impact on the results of statistical tests and models, we would also like to briefly mention statistical techniques to identify such outliers.

The **Z-score** is a standardized measure that represents the number of standard deviations a data point is away from the mean of the dataset. A high absolute value of the Z-score indicates that the data point is far from the mean, which can be considered an outlier. To detect outliers using the Z-score method, follow these steps:

1. Calculate the mean (μ) and standard deviation (σ) of the dataset. Let’s consider a dataset with the following 10 data points: 23, 25, 26, 28, 29, 31, 32, 34, 50, 55 μ = (23 + 25 + 26 + 28 + 29 + 31 + 32 + 34 + 50 + 55) / 10 = 33.3 σ ≈ 9.68 (calculated using standard deviation formula)
2. Compute the Z-score for each data point using the formula: Z = (X - μ) / σ, where X is the data point. Z(23) ≈ (23 - 33.3) / 9.68 ≈ -1.06 … Z(55) ≈ (55 - 33.3) / 9.68 ≈ 2.24
3. Identify outliers based on a chosen threshold for the Z-score, usually |Z| > 2 or |Z| > 3. Using Z| > 2, data point 55 has a Z-score of 2.24, which is greater than 2, so it could be considered an outlier.

Using the Z-score method assumes that the data is normally distributed. Outliers are identified based on their distance from the mean in terms of standard deviations. However, this method can be sensitive to extreme values, which might affect the mean and standard deviation calculations.

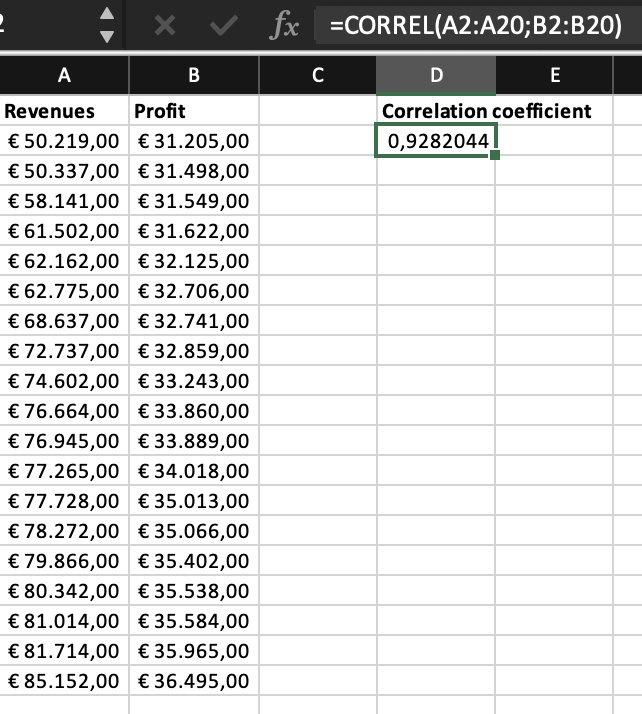
The **interquartile range (IQR)** is a measure of statistical dispersion that represents the difference between the first quartile (Q1, 25th percentile) and the third quartile (Q3, 75th percentile) of the data. The IQR is less sensitive to extreme values than the mean and standard deviation, making it a more robust method for detecting outliers.

1. Calculate the first quartile (Q1) and the third quartile (Q3) of the dataset. Let’s consider a dataset with the following 10 data points: 23, 25, 26, 28, 29, 31, 32, 34, 50, 55 Q1 (25th percentile) = 26 Q3 (75th percentile) = 34
2. Compute the interquartile range (IQR) using the formula: IQR = Q3 - Q1. IQR = Q3 - Q1 = 34 - 26 = 8
3. Define the lower and upper bounds for outliers: Lower Bound = Q1 - k \* IQR and Upper Bound = Q3 + k \* IQR, where k is a constant (typically 1.5 or 3). Lower Bound = Q1 - k \* IQR = 26 - 1.5 \* 8 = 14 Upper Bound = Q3 + k \* IQR = 34 + 1.5 \* 8 = 46
4. Identify data points that fall below the lower bound or above the upper bound as outliers. Data points 50 and 55 are above the upper bound (46), so they can be considered outliers.

The IQR method does not rely on the assumption of a normal distribution and is more robust to extreme values than the Z-score method. However, it might not be as effective in detecting outliers in datasets with skewed distributions. Therefore, using multiple methods to detect outliers can give a more complete picture.

#### 3.5.3.2 Correlation analysis

While correlations can be depicted visually, a correlation can also be measured statisticallay, quantifying the strength and direction of the relationship between two variables. The most widely used correlation coefficient is the Pearson correlation coefficient, also known as Pearson’s r. The Pearson correlation function in Excel is CORREL. For example, if you want to find the Pearson correlation coefficient between two variables in cells A2 through A20 and B2 through B20, you would enter the formula “=CORREL(A2:A20,B2:B20)” in a cell.



Correlation coefficient

The correlation coefficient is a numerical value that ranges from -1 to +1. A correlation coefficient of -1 indicates a perfect negative correlation, where one variable increases as the other variable decreases. A correlation coefficient of +1 indicates a perfect positive correlation, where both variables increase or decrease together to the same extent. This may help, for example, in a situation where you have to decide between using two measures. Imagine you have to decide whether you want to use revenues or profit to proxy for firm performance. If these variables correlate (nearly) a 100%, it does not matter which variable you use all analysis outcomes will be the same. A correlation coefficient of 0 indicates no correlation between the variables. Furthermore, we use the following rule of thumb to describe the strength of a correlation (this rule might differ slightly across disciplines):

* r = 0: No linear relationship between the variables.
* 0 < |r| < 0.3: A weak or negligible linear relationship.
* 0.3 ≤ |r| < 0.5: A moderate linear relationship.
* 0.5 ≤ |r| < 0.7: A strong linear relationship.
* 0.7 ≤ |r| ≤ 1: A very strong linear relationship.

Correlations help identify relationships between variables that might not be readily apparent through visual inspection alone. By quantifying the relationship, correlations provide a more objective basis for understanding the associations between variables in a dataset. Further, correlations help business analysts to generate hypotheses about the underlying causes or drivers of the observed relationships. While some common knowledge or gut feeling may be useful to start with, we want to train how to use data to guide the analysis process. Insights from correlations help us to complement our personal views, and provide more structured evidence rather than gut feeling does. These preliminary hypotheses can then guide further investigation and advanced analysis. That is, although correlations do not imply causation, they can provide initial evidence for potential causal relationships between variables. By identifying strong correlations, researchers can prioritize further investigation into causal relationships, using methods such as controlled experiments, natural experiments, or more advanced statistical techniques like instrumental variables.

An example is the correlation between advertising spending and sales revenue. A company might want to know whether its advertising efforts is related to increased sales. To test this correlation, the company could collect data on its advertising spending and its sales revenue over a period of time, such as a quarter or a year. The company could then use the Pearson correlation coefficient to calculate the strength and direction of the correlation between the two variables. If there is a strong positive correlation between advertising spending and sales revenue, the company can conclude that advertising efforts and sales move together. On the other hand, if there is no or a weak correlation, the company may need to rethink its advertising strategy or explore other factors that may be affecting sales revenue.

**Caution**: again, as stated above, correlation is not causation (and never ever write in your exams, assignment, or reports that one variable affects/impacts/influences/or increases another variable when you refer to correlations :))! While advertising spending and sales might move in the same direction, this does not automatically imply that the increase in sales is caused by the increase in advertising spending. Consider the following famous example on the correlation between ice cream sales and drowning deaths. Both variables tend to increase during the summer months, leading to a correlation between the two. However, this correlation is spurious, as there is no causal relationship between the two variables. In reality, the correlation is likely driven by a third variable, such as warmer weather. Warmer weather may lead to increased ice cream sales, as people seek cold treats to cool down, and may also lead to increased swimming and water activities, which could increase the risk of drowning deaths. Therefore, while the correlation between ice cream sales and drowning deaths may seem to suggest a causal relationship, it is actually spurious and driven by a third variable.

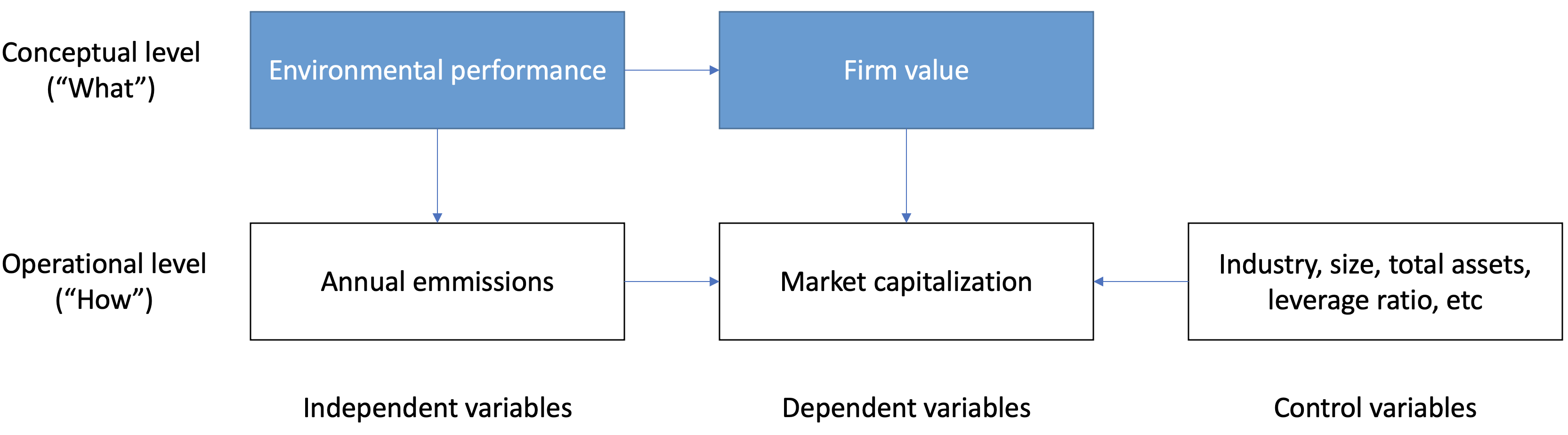
An example of spurious correlation in the business context could be the correlation between heating expenses and sales numbers. While there may be some anecdotal evidence to suggest that a comfortable office temperature can improve employee productivity, that does not imply that there is a structural causal relationship between the two variables. If the business analyst were to find a correlation between heating expenses and sales, this would likely be spurious. The correlation may be due to a third variable, such as seasonality: in the winter, employees might be less likely to take holidays, which may imply longer presence at the office (which increases heating expenses) and simultaneously imply more working hours–hence the increase in sales. So we need to be cautious and avoid drawing causal conclusions based on a potentially spurious correlation. It is essential to carefully consider the underlying data and any potential confounding variables before making any changes to office temperature or other policies that may impact sales; more advanced econometric analyses are required to determine causality.

## 3.6 Data challenges

It would be great if the data would come in a form that we can directly analyze. However, this is, unfortunately, literally never the case. Below we describe some of the common challenges we experience while preparing our data for analysis. The techniques explained earlier are actually helpful in figuring out several data issues. We also provide a couple of typical solutions to these issues. Disclaimer: please note that each of these solutions comes with its own challenges; challenges that go beyond this chapter and course.

### 3.6.1 Predictive validity

Measurement is a notable challenge when it comes to having data available that accurately represents the desired underlying construct. When we say something is valid, we make a judgment about the extent to which relevant evidence supports that inference as being true or correct. This issue can be better understood through the predictive validity framework, which emphasizes the importance of a measure’s ability to predict relevant outcomes or behaviors.



Predictive validity framework

Let’s look at the example above. At the conceptual level, the study investigates in how far a company’s environmental performance predicts firm value. Now at the operational level, the measurement level, the study uses annual emissions as proxy for environmental performance. Emissions are oftentimes used because that is relatively easy to measure, certainly in regulated industries. But is this the concept you want to get at? Are environmental concerns not broader than emissions? This is why when working with data, business analysts and researchers often encounter difficulties in finding suitable proxies for their desired constructs. This is because the data might have been collected for purposes unrelated to the research objectives, making it hard to ensure that the available data truly aligns with the target construct. Consequently, the proxy may lack predictive validity, reducing the overall quality of the analysis.

What to do about it? To enhance the predictive validity of your variables, various strategies help to ensure that the variables are accurate, reliable, and relevant to the desired construct. Firstly, carefully selecting variables is essential. This involves reviewing internal and external data sources, industry benchmarks, and best practices to identify variables that have a strong association with the construct of interest. This helps to establish a sound theoretical and practical foundation for the variables used in the analysis. Another approach is using multiple proxies for each construct, capturing different aspects of the construct and reducing measurement error.

Furthermore, validating the chosen variables is important. By ensuring that the variables have both convergent validity (measures of the same construct are highly correlated) and discriminant validity (measures of different constructs are not highly correlated), business analysts can enhance the overall predictive validity of their analysis.

Proper measurement is also essential for increasing predictive validity. Analysts should use established and validated measurement tools, or, if necessary, develop and validate their own instruments. This may involve piloting the measurement process, establishing internal consistency, and assessing the stability of the measures over time.

Sample selection plays a vital role in enhancing the generalizability of the analysis. A well-chosen, representative, and diverse sample of data points can help ensure that the relationships observed between variables are robust and applicable to the wider business context, thus increasing predictive validity.

Moreover, controlling for confounding variables is crucial. Analysts should identify potential confounding variables and account for them in their analysis or models. By controlling for these factors, analysts can reduce the likelihood of spurious relationships (see 3.5.3.2) and increase the predictive validity of their chosen variables. We will return to this topic in Chapter 4.

Lastly, regularly re-evaluating and updating the chosen variables is essential to maintain their relevance and predictive validity. This may involve incorporating new findings from industry research, revising measurement tools, or re-assessing the relationships between variables in light of changes in the business environment.

### 3.6.2 Structured and unstructured data

Structured and unstructured data can pose challenges because unstructured data often requires additional preprocessing and transformation to extract useful information. This can be time-consuming and necessitate specialized tools or techniques, such as natural language processing or image recognition. High dimensionality in (un-) structured data can also make it difficult to identify patterns or trends, requiring dimensionality reduction techniques. High dimensionality in data refers to a situation where there are a large number of variables or features in the data. It can occur in both structured and unstructured data, but it is particularly challenging in structured data where each row represents a record or observation, and each column represents a variable or feature.

For example, in a dataset containing information about customer purchases, each customer may have a large number of attributes such as age, gender, income, location, shopping habits, and so on. If each of these attributes is represented as a separate column, the resulting dataset can have a high number of columns, or dimensions, making it highly dimensional.

Dealing with challenges posed by structured and unstructured data, particularly preprocessing and transformation of unstructured data, can be addressed through a variety of methods. While it can be time-consuming and require specialized tools or techniques, a systematic approach can help make the process more efficient and effective.

Firstly, for unstructured data, it is essential to identify the specific information that needs to be extracted to address the business question. By narrowing down the focus, you can ensure that the preprocessing and transformation efforts are targeted and relevant.

Next, leveraging specialized tools and techniques is necessary to effectively work with unstructured data. For textual data, natural language processing (NLP) tools can be used to extract relevant information from text. For image data, image recognition techniques, such as convolutional neural networks, can be employed to classify, detect objects, or segment images. Similarly, for audio data, signal processing tools and techniques can be used to extract relevant features.

To tackle high dimensionality, dimensionality reduction techniques can be applied. For structured data, methods such as principal component analysis (PCA) or feature selection techniques like forward or backward selection can be used. For unstructured data, approaches like topic modeling for text data or autoencoders for image data can help reduce dimensionality.

### 3.6.3 Data sources

Problems in identifying patterns in data can arise from data sources because they may contain inconsistent, incomplete, and outdated information, or may not be accessible. Data collected from different sources or systems might have different formats, units, timeframes, or representations. Inconsistencies across data sources can complicate data integration and can make it difficult to use datasets for analysis and pattern identification. Data sources may not contain all the necessary information, thus only delivering incomplete data, and requiring additional data acquisition or integration of multiple sources to get a complete picture.Furthermore, data sources may be outdated or not up-to-date, making it difficult to identify current trends or patterns. In addition, gaining access to relevant data sources can be challenging due to privacy concerns, legal restrictions, or organizational barriers. Think of GDPR concerns not allowing you to use personnel data.

How to deal with it? Firstly, data cleaning and preprocessing are essential steps before conducting any analysis (see next section).

Secondly, ensuring the accuracy and reliability of the data through validation and verification is important. Cross-checking data sources, reviewing data collection methods, and confirming the data’s authenticity can help reduce the risk of errors and improve the quality of the analysis.

Combining multiple data sources can help address issues of incomplete or inaccessible data. By aggregating data from different sources, you can create a more comprehensive dataset that may reveal patterns that were previously hidden. It’s important to carefully assess the compatibility and quality of the different data sources before combining them. Updating your data sources regularly can help mitigate the issue of outdated information.

By incorporating the most recent data available, you can ensure that your analysis remains relevant and accurate. This may involve subscribing to data feeds, setting up automated data updates, or maintaining a schedule for manual data updates.

When dealing with inaccessible data, it may be necessary to explore alternative data sources or proxy variables. These alternatives can sometimes provide insights into the desired patterns, even if they are not the original variables of interest. It’s essential to evaluate the suitability and validity (see point 3.6.1) of these alternatives in the context of your analysis.

### 3.6.4 Data cleaning and formatting

Data cleaning and formatting can lead to problems in pattern identification because poor data quality can significantly impact the analysis process and insights derived. For example, handling different character encodings or converting between formats can be problematic, leading to unreadable text. Or ensuring the accuracy and consistency of data values may require additional validation checks or business rules. Time-consuming and labor-intensive data cleaning efforts may also divert resources from the actual analysis, potentially delaying the identification of valuable patterns.

What can you do about it? Firstly, establish a consistent data formatting and encoding policy across your organization or project. By enforcing standard formats and encodings, you can reduce the likelihood of encountering issues related to unreadable text or inconsistent data values. This can simplify the data cleaning process and reduce the risk of errors during data transformation.

Secondly, automate data cleaning and validation processes wherever possible. By using scripts or tools to automatically detect and correct inconsistencies, missing values, or errors in the data, you can save time and resources that would otherwise be spent on manual data cleaning efforts. Automation can also help ensure that the data is cleaned and validated in a consistent manner, reducing the risk of human error.

Additionally, prioritize data cleaning and validation tasks based on their potential impact on pattern identification. Focus on addressing issues that are most likely to have a significant effect on the analysis results, while deferring less critical tasks if time or resources are limited. This approach can help ensure that the most valuable patterns are identified, even if some data quality issues remain unresolved.

Collaborate with data providers or stakeholders to address data quality issues at the source. By working together to establish clear data quality standards, guidelines, and expectations, you can minimize the occurrence of errors and inconsistencies in the data. This proactive approach can help reduce the need for extensive data-cleaning efforts and enable quicker pattern identification.

Finally, invest in training and capacity building for your team in the areas of data quality management, data cleaning, and data validation. By enhancing your team’s skills and expertise in these areas, you can improve the efficiency and effectiveness of your data cleaning and formatting efforts. This can help ensure that your team is better equipped to identify patterns and derive valuable insights from the data.

### 3.6.5 Noisy data

Noisy data refers to data that contains irrelevant, random, or erroneous information, which can distort the true underlying patterns, relationships, or signal in the dataset. Noisy data can arise due to various reasons, such as measurement errors, data entry mistakes, data corruption, or natural variability in the data-generating process. In the context of statistical analysis, noisy data can have a detrimental impact on the performance, accuracy, and interpretability of models and algorithms. There are several types of noise that can affect data:

**Random noise:** This is the result of random fluctuations or errors in the data collection or recording process.

**Systematic noise:** This occurs when there is a consistent error or bias introduced into the data due to a flaw in the data collection or measurement process.

**Outliers:** Outliers are data points that deviate significantly from the overall pattern or distribution of the data. They can be considered as a form of noise when they result from errors or anomalies in the data collection process.

Dealing with noisy data often requires additional preprocessing steps. One approach to managing noisy data is applying smoothing techniques such as moving averages, which help reduce the impact of random variations in the data.

Outlier detection is another important aspect of handling noisy data. Identifying and dealing with outliers, which are data points that deviate significantly from the norm, can help minimize their impact on pattern identification. Visual methods such as box plots or various statistical methods, such as the Z-score or the IQR method, can be used to detect outliers.

Outliers can be caused by measurement errors, data entry errors, sampling problems, or naturally occurring variability. Outliers can have a significant impact on the regression analysis, as they can pull the regression line towards them, resulting in inaccurate estimates of the relationship between the variables.

**Measurement error** is a type of error that can occur when the measurement instrument used to collect the data is imperfect or when the measurement process is not accurate. Measurement error can cause outliers in the data because it can lead to values that are significantly different from the true value of the variable being measured. Measurement error in customer satisfaction data can occur due to errors in the survey process, such as asking unclear or biased questions, or due to errors in the data entry process. For example, if a respondent mistakenly selects the wrong answer on a customer satisfaction survey, this could result in an outlier in the data.

**Data entry errors** are mistakes that occur during the process of entering data into a computer or database. Data entry errors can cause outliers in the data because they can lead to values that are significantly different from the actual value of the variable being measured. For example, suppose that the accountant incorrectly records of sales transactions and types €200 instead of €100.

If the outliers are due to measurement errors, data entry errors, or sampling error, then it may be appropriate to remove them from the dataset. In that case, the data points do not belong into the data. You can use the z-Score o IQR (see section 222) to identify and delete the outlier.

However, things are different when considering data points which are extreme, but real. Sometimes outliers occur naturally, and are part of the population you are investigating. In this case, you should not remove the outlier, but handle it differently.

A common approach to do so is the transformation of the variables. This can be done by applying a mathematical function to the data that changes the scale or distribution of the data. For example, a log transformation of the respective variable can be created and used in the analysis to reduce the impact of outliers in positively skewed data.

Alternatively, winsorizing is a technique that involves replacing the outliers with the nearest values that are within a certain percentile range. For example, if the 99th percentile is used, any values above the 99th percentile are replaced with the value at the 99th percentile. This can be done on both ends of the distribution to deal with outliers in both tails.

Lastly, validating the results of the analysis using multiple methods or independent data sources can help ensure that the patterns identified are not merely artifacts of noisy data. By cross-validating the findings or comparing them to results from other studies or sources, you can increase confidence in the patterns and trends identified in the presence of noisy data.

### 3.6.6 Normalization and scaling

Normalization and scaling are related terms, often used interchangeably, but they do have some subtle differences in meaning. Both refer to the process of transforming data to fit within a specific range or distribution. More specifically, while normalization changes the shape of the distribution of the data, scaling changes the range of the data. This can be particularly useful in regression analysis because it can help to prevent variables with larger scales from dominating the analysis and can improve the accuracy and stability of the regression model. Typical approaches for normalization are log transformations or Box-Cox transformations; for scaling, we often use min-max scaling or the Z-score normalization (yes, confusing wording!).

However, but exactly for that reason, normalization and scaling of variables can also create a problem for identifying data patterns as it changes the underlying values of the data, which can obscure the original relationships between the variables. For example, if two variables are highly correlated in their original scale, but one variable is changed to a much smaller scale, the correlation between the variables may appear weaker than it actually is. This can lead to misinterpretation of the data patterns and incorrect conclusions about the relationship between the variables.

It can also result in the loss of important information in the data. For example, if a variable has extreme values that are outliers in the original scale, these values may be lost or diminished in the normalized scale. This can lead to a loss of information that may be important for understanding the relationship between the variables.

It is important to carefully consider the appropriateness of normalization and scaling based on the nature of the data and the research question. It may also not be necessary to change the original data, particularly when you are concerned with multivariate: many statistical packages offer the option to present coefficients which provides the advantage of comparability, but does not change the underlying data.

### 3.6.7 Missing data

Missing data can lead to problems in identifying patterns because it can result in biased or incomplete insights and may affect the performance of statistical models. To address this issue, firstly, it is essential to understand the underlying reasons for the missing data. This can help determine whether the data is missing at random, missing completely at random, or missing not at random. Understanding the nature of the missing data can guide the choice of appropriate techniques to handle it. For now, this goes beyond the scope of this chapter.

# 4. Diagnostic Analytics

## 4.1 Learning goals

1. Critically analyze identified data patterns
2. Evaluate the association between important decision variables
3. Evaluate business decisions and refine critical assumptions to improve decision-making by using advanced methods to analyze business data

## 4.2 What is diagnostic analysis?

Diagnostic analysis seeks to answer the question of why a particular event or outcome happened by uncovering the underlying factors, relationships, or determinants that led to the observed data pattern. This understanding can help decision-makers identify the root causes of issues, determine the key drivers of success, and thus guide future decision-making.

Diagnostic analysis is a natural extension of exploratory analysis—the identification of data patterns ([Chapter 3](#sec-datapat)). While exploratory analysis focuses on discovering trends, patterns, and relationships in the data, diagnostic analysis goes a step further by investigating the reasons behind those patterns. Consider the following question: *Why are some business units more productive than others?* By conducting diagnostic analysis, analysts can investigate the factors contributing to productivity differences among business units. Some potential reasons could be unit-specific competitive advantages, management styles, resource allocation, employee skills, company culture, or even measurement issues. In this case, diagnostic analysis techniques provide a principled way to disentangle many potential reasons and better understand the reasons for productivity differences. And of course, the insights gained from the analysis could in turn be employed to develop targeted strategies to improve overall firm performance.

Another example of a question lending itself for diagnostic analysis is: *What customers reacted well to a marketing campaign?* Analyzing differences in customer responses to marketing campaigns allows analysts to identify which customer segments reacted positively and why. This information can be used to tailor future marketing efforts, improve targeting, and increase the return on investment for marketing initiatives. Understanding the drivers of customer behavior can also inform product development, pricing strategies, and customer relationship management efforts.

## 4.3 Diagnostic analysis requires a bit of causal reasoning

A diagnostic analysis often requires more advanced statistical techniques than those used in exploratory analysis. Even more importantly, it requires an even stronger reliance on mental models. This is because mental models are required to design and interpret analyses able to speak to “why” questions. As a simple example, suppose we stumble upon a survey that finds a positive relation between respondents being married and respondents self-reported happiness score. What drives this relation? Is it that marriage makes people happy? Or that happier people are more likely to get married? Or is there even a third explanation (both variables are correlated with age). Analyzing data with the goal of answering “Why?” and “What drives…?” questions requires even more recourse to models that represent the relationships between variables and potential causal factors. The same holds for the development of the analysis itself. We need a model to guide the selection of relevant variables, the choice of appropriate statistical techniques, etc.

### 4.3.1 The importance of assumptions in causal reasoning

Before proceeding, we want to offer a word of caution. “Establishing causality” is often considered the gold standard in decision-making, as it allows for a better understanding of cause-and-effect relationships, which can lead to more effective policies and interventions. It is also impossible because every analysis rests on untestable assumptions. Even in a randomized experiment, analysts have to make assumptions about proper measurement of often difficult to define and measure constructs.

When data is not arising from a randomized experiment (i.e., all real world data), decision-makers face even more challenges when trying to infer the causes of patterns from data. Confounding, endogenous selection issues, and disagreement about the correct mental model are all issues that make it difficult to distinguish between correlation and causation. Especially in a complex and dynamic business environment we face the issue of multiple correlated factors that influence the outcomes of interest. Disentangling these factors to isolate relations is challenging.

The role of theory and hypothesis development thus becomes even more crucial. Theory and hypotheses serve as the foundation for reasoning about the underlying mechanisms and relationships between variables. Developing a well-thought through mental model allows decision-makers to make more informed assumptions and predictions about the potential outcomes of their policies. A good model makes it easier to develop useful hypotheses. Likewise, mental models guide the collection and analysis of data by specifying the expected relations between variables. Finally models help to identify potential confounders and biases.

In sum, it is crucial for any decision-maker to be very clear about what the mental model and the in-built assumptions are that guide any diagnostic analysis. Only with such an understanding can the analysis be directed and results be analyzed.

## 4.4 Developing a mental model

A mental model is a conceptual framework that represents the relationships between variables and potential causal factors. A mental model represents how different elements in a system are interconnected and how they influence each other. In the context of business relationships, mental models can help decision-makers better understand complex dynamics and make more informed decisions.

Consider the example of customer churn in a subscription-based business. The first step in developing a mental model involves identifying the key variables that are relevant to customer churn. These variables might include factors such as product quality, quality of customer support, pricing, and competitor offerings. This step of first identifying key determinants is essential, because it helps you focus the analysis on the most relevant factors.

After identifying the key variables related to customer churn, the next step is to reason about the relations between these variables. This involves reasoning about how changes in one variable might affect other variables in the system. For example, how does an increase in prices impact the likelihood of a customer churning? Or how does the quality of customer support influence the likelihood of churn? By modelling these relations, we develop the underlying structure of the problem. We need this structure for diagnostic analysis. As we will see in examples later, without it we would basically interpret correlations blindly and at great peril.

The more expert knowledge we have in an area, the more we can already resort to our experience to build mental models. Still, do not forget to reason about variables and patterns uncovered via the preceeding exploratory analysis. Often new insights were generated at that step. Now, for a more systematic approach, we recommend trying to take each step one at a time to not get off track, and evaluate after each step whether you believe you completed the respective step. First, list all variables that you believe may be connected with your variable of interest that you want to explain and that popped up as interesting during the exploratory stage. Then describe the nature of the relations between the variables. We actually encourage you to draw a diagram. A visual representation of your mental model is often a great help for organizing your thoughts, critically analyzing then, and communicating them to others. The example below illustrates the point that text only may not be enough:

|  |
| --- |
| Figure 4.1: Mental model |

Source: Schaffernicht (2017)

Let’s go back to our customer churn example. With regard to our key variable of interest, customer churn, we **hypothesize** the following:

1. Quality of customer support is negatively associated with customer churn (because better customer support quality makes customers happy, and they are less likely to leave).
2. Product quality is negatively associated with customer churn (because higher product quality makes customers happy, and they are less likely to leave).
3. Competitor offerings are positively associated with customer churn (because interesting offers from competitors might trigger our customers to switch to our competitors).
4. Price is positively associated with customer churn (because customers do not like to pay more, and they are more likely to leave).
5. Product quality is positively associated with price (because higher quality is usually costly, sales prices need to be higher as well).
6. Competitor offerings are negatively associated with price (because of the pressure in the market and the offer by our competitors, we might need to reduce our prices).

It is easy to loose the bigger picture and a quick sketch makes our reasoning much more transparent:

|  |
| --- |
| Figure 4.2: Mental model |

And now you are ready to test your mental model against real-world data and observations to ensure its accuracy and relevance. The sketch (as we will explain later) also already tells you that you cannot analyze the relation between churn and price without holding the influence of product quality and competitor offerings fixed! Something that your analysis of a price effect will need to do, if your theory is correct. The example thus highlights, why having a mental model is essential as basis for any decisions you want to make. Without a clear idea of **why** certain variables are associated, you will not be able to interpret any results that your analyses will provide you. It needs sound theoretical reasoning to give meaning to your results.

Developing a mental model is an iterative process, where the model is continuously refined based on the insights gained from the analysis. As new relations and factors are identified, the mental model can be updated to better represent the true underlying structure of the data. As new information emerges or the business environment changes, update your mental model to reflect these developments. This iterative process allows for a deeper understanding of the factors influencing, for example, customer churn and more effective decision-making regarding this issue.

## 4.5 There is no such thing as chance

Before diving into the details of analyzing why questions, we need to clarify some basics. You might have heard phrases like: “This relation between x and y is unlikely due to chance”. What this means, is that the analyst believes the pattern observed in the data between x and y would also be present in general and not just in this particular data set. That it is generalizable. But how could “chance” (or “noise”) even create a pattern? What does this mean?

To understand what is actually meant by randomness or chance in our context, think of a coin flip. We usually assume the chance of a coin landing heads or tails to be even (50% it lands heads and 50% it lands tails). We often say: the chance of a coin landing heads or tails is purely *random*. Notice though, that randomness here is a statement about our *ignorance*, not a property of the universe. We know the physics of a coin flip quite well. If we could measure the angle of the coin flip, the wind around the coin, the distance to ground, etc. We could combine our knowledge with the measure to built a prediction model and estimate the probability of the coin landing heads. We would very likely do much better than 50%. Randomness is thus simple due to the fact that either, we have not measured all determinants of the outcome (e.g., coin lands heads), or we do not even know enough about the problem to specify the determinants with certainty (a mental model question). In regression models, all the unmeasured and unmodelled influence are packed into what is called an error term. The following is a typical regression equation, specifying a linear relation between and :

The is the error term. It captures all the other things that also influence and which we didn’t care to model and measure explicitly. The fact that there will be always an error term—we can never hope to name and measure all influences of even simple problems—is the source of “chance” patterns.

The following simulation tries to illustrate the issue. We draw 50 x variables at random. Its values are:

[1] -6.81405021 0.29080315 -16.64506129 -16.93622258 -19.55388221  
 [6] 27.82578770 -2.12722250 -4.42818912 6.06723243 -4.86329299  
[11] -12.99773334 3.66039111 -15.46757532 0.15252543 -0.47744402  
[16] -3.97635456 -5.01853836 -21.89194429 -4.70215525 5.51044371  
[21] -1.93933150 4.45463912 -0.39409296 18.12015075 -20.29820415  
[26] -5.13356070 -5.03222157 2.08585323 -9.37122698 -12.85983170  
[31] -4.86984684 3.48049542 -0.02823887 7.00464237 -17.87917138  
[36] -4.87859963 -7.17129903 10.38264712 -18.20612595 17.46602851  
[41] 8.98895014 13.38321090 -16.12558303 -1.94650403 4.71490692  
[46] 3.92105297 -7.14457750 -7.37191957 -18.61520235 -14.39777436

We then we draw some other influences at random, pack them into an error term and compute a variable according to the equation above . We do this 20 times, but we always use the same values above. Importantly, this means that the true slope for all 20 samples is 2. If you need an example, think of 20 surveys where each time 50 people were surveyed about how happy they are ( = happiness). The survey firm tried very hard to always get the same distribution (say x is age of the respondents), but it can obviously never find exactly the same type of people. Because people of the same age are still very different, the error term will always differ quite a bit among these 20 surveys. [Figure 4.3](#fig-rand1) shows how much the *sample* relation (the slope) between x and y varies between the 20 surveys:

|  |
| --- |
| Figure 4.3: Imprecisely estimated relations as a function of unmeasured influences |

Remember the slope underlying the relation between x and y for all 20 samples is 2! All the variation you see here in slope (some are even negative!) is due to the unmeasured influences in . Remember we always used the same x values. If you pay careful attention, you can see that the dots in the plot only move up and down. The values are fixed. It is thus the changing amounts of that are added to x to arrive a that camouflage the true relation. This is what people mean when they say a relation can be due to chance.

The “chance” of chance patterns is thus higher when there is a lot of unexplained variation of the outcome of interest left. The best remedy against this is: more data. In [Figure 4.4](#fig-rand2), we start with a sample of 50 observations and increase it step by step by 10 observations up to a maximum of 500 observations.

|  |
| --- |
| Figure 4.4: Increasing sample size reduces noise |

Note how the regression line becomes more and more certain what the relation could be (it slowly converges to 2).

Let’s consider an example from the business world to illustrate again what to look out for. Suppose you are a sales manager at a retail company and you want to determine if sales performance varies significantly between two groups of sales representatives: those who work weekdays (Group A) and those who work weekends (Group B). After analyzing the data, you find that Group B had, on average, 20% higher sales than Group A. However, you need to determine if this difference is due to the different work schedules or if it could have occurred by chance. Given all the other influences that might determine sales, it is probably not unreasonable to be skeptical that this pattern might be related just to differences in people. We thus need some protocol or workflow to decide how to interpret this pattern. One such workflow are classical statistical **hypothesis tests**, which we discuss next.

## 4.6 Hypothesis testing

Hypothesis testing is an extension of the logic we used above to provide intuition for chance patterns in the data. More formally, it is a statistical procedure used to make inferences about a *population parameter* based on a *sample* drawn from a *population*. Population just means: all relevant units of interest. The sample is then those units we have data on. A population parameter is then an unknown parameter describing a relation of interest, derived from our mental model. In [Figure 4.3](#fig-rand1) the population parameter (technically unknown) was a slope parameter between y and x of 2. The sample estimates (the blue line that keeps jumping around) is our sample estimate of that population parameter. And as [Figure 4.3](#fig-rand1) shows, it can be quite off sometimes.

The problem that hypothesis testing tries to deal with is this: suppose that our sample estimate of the slope parameter is 0.3, what can we learn about the population parameter (in our example: 2, but again, we wouldn’t know that normally). It’s main component is a hypothesis concerning what the population parameter is, often called the null hypothesis. For example, we could pose a null hypothesis (H) of no-relation (the slope is hypothesized to be zero). However, a null hypothesis could also be: the slope is 15. We decide what hypothesis to test. Let us stick with H: the slope is zero. The “testing” then involves whether the data is more or less consistent with our chosen null hypothesis. Intuitively, if there is a lot of unmeasured influences in , then a slope of 0.3 is probably not unrealistic to occur by chance (again, see [Figure 4.3](#fig-rand1) for how much a parameter can change from sample to sample). Classical hypothesis testing formalizes this intuition and condenses it into two numbers, a chosen significance level and a derived p-value (or equivalently a confidence interval). Here is the thought experiment behind it.

Suppose we would run 1,000 experiments and each time record the slope. Then we count how often we see certain magnitudes. Given our null hypothesis of the slope being 0 and an estimate of the amount of unexplained influence that lead to variation around 0, we would expect a histogram like the left one in [Figure 4.5](#fig-rand3):

|  |
| --- |
| Figure 4.5: Hypothesis testing: how likely is it to see certain slopes, given the null hypothesis? |

If you look at the left plot, the chance of observing a slope of +/- 0.3 is quite high. We would observe a lot of counts in this range, if H would be true. When seeing a 0.3 slope, we would thus *not* be able to rule out H, would not be able to rule out that a slope value of 0.3 is due to chance, and thus not be able to rule out that the pattern is due to chance. But, because we simulated the date, we know that it would be mistake to conclude that the pattern is due to chance. The true slope is different from 0; it is 2. As shown on the right plot of [Figure 4.5](#fig-rand3), we are less likely to draw a slope value of 0.3 when we get a sample from the true distribution (but it is not unlikely!).

A p-value and significance level quantify this reasoning. We need to find a cut-off after which we say, “Now, slope values higher than that are too unlikely to be due to a population with H as its parameter”. When choosing such a cut-off, we need to also consider the chance of making a mistake. As we can see in [Figure 4.5](#fig-rand3), there are some unlikely values. E.g., the slope value of 5 is rare under H: the slope is zero. But it is not impossible. Still, the chance of making an error is small when we see a value of 5 and say, “ok, the population parameter is probably not zero”. The amount of error we are willing to consider is called the significance value. You can see it in the figure. It is the shaded region in the left plot. The shaded region shows the 97.5 and 2.5 percentile—the most extreme values occurring with 2.5% or less of a chance on both sides. If we get a sample whose slope value lies in the shaded region, we would only expect to see such a value with 5% chance (the chosen significance value). So we would make an error in 5% of the cases when deciding to reject H. Similarly, the p-value is the chance of seeing a slope value as the one observed in the sample if H is true (and the regression model is correct). If it is below our chosen significance level (our chosen maximum error level), we reject H in favor of the alternative.

The actual mathematics of computing p-values, test statistics, and so on are very important. However, these are covered in detail in any standard regression textbook and we do not see a lot of value in repeating them here. Instead we focused on the intuition. We hope it has become clear that *hypothesis testing is in fact a decision making framework*. One that tries to control a certain error rate with the significance level: The chance of incorrectly rejecting a hypothesis.

In summary, the general steps involved in hypothesis testing thus are:

1. Formulate the null and alternative hypotheses: Clearly state the null hypothesis (H). The alternative H is usually that that the population parameter is not equal to the population parameter. For example, H: There is no difference in sales performance of weekday- and weekend-workers. H: There is a difference in sales performance of weekday- and weekend-workers. There are also of course, directional hypotheses, such as the population parameter is greater than 0. The same logic applies to them.
2. Choose an appropriate significance level (): It is the error level we allow ourselves; the point at which we say, values that occur with a probability of or less we consider to not arise from H. A common choice for is 0.05, indicating a 5%. But that number is to some extent ad-hoc. Think carefully about what it should be in your application.
3. Select a suitable statistical test: Depending on the research question, data type, and assumptions, choose a statistical test (e.g., t-test, chi-square test, ANOVA) to analyze the data. We have not talked much about this yet. We will talk more about it in the exercises.
4. Calculate the test statistic and p-value: Apply the selected statistical test to the sample data and calculate the test statistic and corresponding p-value. Let’s say we find a p-value of 0.02. This means that there is only a 2% probability of observing a difference as large as the one we found (or larger) if the null hypothesis were true (i.e., if there was no real relation between the work schedule and sales performance).
5. Compare the p-value to the chosen significance level (): If the p-value is less than or equal to , reject the null hypothesis in favor of the alternative hypothesis. If the p-value is greater than , fail to reject the null hypothesis, meaning there is not enough evidence in the data to rule out H. Note: When you cannot reject H it does **not** mean that H is true! It simply means, we cannot rule it out. Absence of evidence is not the same as evidence of absence.

## 4.7 The Regression framework

We basically introduced half of what a regression analysis is about already in the preceding two sections. Regression analysis is a statistical method used to model the relationship between a dependent (response) variable and one or more independent (predictor) variables. It is an incredibly powerful tool in any data analysts toolbox. In addition, it provides a great teaching aid where we can introduce may important concepts that generalize to all other analysis settings. To do so we start with the difference between correlation coefficients and coefficients in a regression model.

### 4.7.1 Revisiting correlations

In [Chapter 3](#sec-datapat), we touched upon correlations. A correlation coefficient only assesses the strength of an assumed linear relationship in a sample—*how linear* it is. It does not tell us anything about the *magnitude* of the relation. If we want to learn something about the magnitude of the linear relation—and maybe compare it to the magnitude in other settings or other variables, we need a regression model.

To illustrate, we use an example of a retail company that analyzes the performance of its employees at each branch. The company wants to understand if there is a correlation between average employee tenure at the branch level and the branch profit they generate.[[6]](#footnote-214) The company collected data on the number of months an employee works for a branch and the total profit of the respective branch. We start our analysis by calculating the correlation coefficient between these two variables. We are first interested in the strength and direction of a possible linear relationship. Similarly to earlier, we would have two competing hypotheses:

* H: There is no significant association between employee tenure and profit
* H: There is a significant association between employee tenure and profit

To determine whether this correlation is “not by chance,” the company could perform a statistical test to determine if the observed correlation coefficient is statistically significant (using the correlation:: R package).

correlation(d4[, c("profit", "employee\_tenure")])

# Correlation Matrix (pearson-method)  
  
Parameter1 | Parameter2 | r | 95% CI | t(73) | p  
-------------------------------------------------------------------  
profit | employee\_tenure | 0.26 | [0.03, 0.46] | 2.28 | 0.026\*  
  
p-value adjustment method: Holm (1979)  
Observations: 75

This test would compare the observed correlation against what would be expected if there were no relationship between the two variables (i.e., the correlation is purely random). We can again use a t-test.[[7]](#footnote-215) The results present a correlation coefficient of 0.26 and a p-value of 0.026. This implies there is indeed a weak positive correlation (r = 0.26) which is significant (p-value of 0.026 is smaller than the conventional significance level (α) of 0.10 - in the accounting discipline).

All the correlation coefficient tells us, however, is that there is an indication of a linear relation, *not* the magnitude of the relation (see this [wikipedia graph](https://en.wikipedia.org/wiki/Pearson_correlation_coefficient#/media/File:Correlation_examples2.svg)). If we are interested in that, we need to think about a regression model.

### 4.7.2 Regression coefficients

A regression expresses relations between an outcome (also sometimes called response or dependent variable) and one or more predictor variables (also sometimes called independent variable).

A regression coefficient represents the estimated change in the dependent variable associated with a one-unit change in an independent variable, holding all other independent variables constant. In other words, it quantifies not the direction of the relationship between the independent variable and the dependent variable, but also its magnitude in unit-changes. For example, a positive coefficient indicates a positive relationship, whereas a negative coefficient suggests an negative relationship.

[Figure 4.6](#fig-emp1) plots a regression line. It shows the slope of the following linear equation:

|  |
| --- |
| Figure 4.6: A regression line |

Most data tools these days allow you to estimate regressions. [Figure 4.7](#fig-excelreg) shows regression output, estimating the line above, for a random sample of branches using excel. Below you find R output (using the jtools:: summ function because it creates cleaner summaries).

|  |
| --- |
| Figure 4.7: Simple regression |

m1 <- lm(profit ~ employee\_tenure, data = d4)  
summ(m1, confint = TRUE)

MODEL INFO:  
Observations: 75  
Dependent Variable: profit  
Type: OLS linear regression   
  
MODEL FIT:  
F(1,73) = 5.19, p = 0.03  
R² = 0.07  
Adj. R² = 0.05   
  
Standard errors:OLS  
-------------------------------------------------------------------------  
 Est. 2.5% 97.5% t val. p  
--------------------- ----------- ----------- ----------- -------- ------  
(Intercept) 258178.44 232639.45 283717.43 20.15 0.00  
employee\_tenure 1301.74 163.14 2440.34 2.28 0.03  
-------------------------------------------------------------------------

The row labeled employee\_tenure contains the slope, the of the regression model above. It is the parameter of interest here. First, note that the t-value and p value in this row are the same as that of the correlation coefficient. That is true for a simple linear regression with just one variable. The coefficient itself has value of 1,301.7. It is a measure of *magnitude*. It means that one additional month of average employee tenure is *associated* with 1,301.7 euros more in branch profits. This is the slope of the line in [Figure 4.6](#fig-emp1). Do not be too excited about the magnitude however. Looking at the confidence interval, we can see that there is a lot of uncertainty about the actual magnitude (The 95% range is from 163 to 2,440). 1,301 is the best guess, but not a great guess.

The intercept (also known as the constant or the y-intercept) is the value of the dependent variable when all independent variables are equal to zero. It simply represents the point where the line intercepts with the y axis and usually does not have an interesting interpretation. In this case for example, do we really care that of a branch with an average tenure of zero months, the average profits are 258K euros?

Another feature of fitted regressions is that one can use them for (linear) predictions. If you want to make a prediction about the change in branch profits for let’s say a branch with an average employee tenure of 12 months, we simply use the regression formula and plug in the estimated coefficients: 1,301 \* 12 + 258,178 = 273,790, i.e., the predicted incremental profit after 12 months is 15,612 Euros

### 4.7.3 Isolating associations via multiple regressions

The real use case for linear regression models is not assessing the magnitude of one single variable. It usually is either of two things: For diagnostic analysis it is the ability to estimate a linear association between the outcome and a variable of interest while holding the influence of other variables (sometimes called covariates) fixed. For prediction tasks it is the ability to make make predictions taking a multitude of variables into account. This is the domain of multiple linear regression analysis. It is a straight-forward expansion of the one-variable case.

We will focus on the reasons for including multiple variables into regression for diagnostic purposes and cover the prediction case in the next [Chapter 5](#sec-pred). Generally, there are two reasons for why we would want to include a variable into a regression for diagnostic purposes:

1. If two variables and are related and we want to isolate the association of one with while holding the other one fixed.
2. To take out unmeasured influences out of the error term and thus improve the precision of our coefficient estimates. Remember from [Figure 4.3](#fig-rand1) that regression slopes bounce around more from sample to sample (are less precisely estimated), if there is a lot of unmeasured influences left in the error term. Including big determinants of into the regression takes them out of and thus helps against this.[[8]](#footnote-227)

Why would one do either of these two steps for diagnostic purposes is best explained by example. Put yourself into the shoes of a business controller at a retail store chain and doing some exploratory analysis you notice that there is considerable variation in retail store performance in a certain region. You want to figure out the reason for that (diagnostic analysis).

Before you start throwing data at a regression, you do the smart thing and first come up with a first mental model of what should drive store revenues. It looks like this:

|  |
| --- |
| Figure 4.8: DAG of a simple model of store revenues |

Your data (stored in a data.frame called d5) looks like this:

# A tibble: 6 × 5  
 StoreID foot\_traffic sick\_calls staff revenues  
 <int> <dbl> <dbl> <dbl> <dbl>  
1 1 143 1 4 31839  
2 2 216 2 8 48804  
3 3 78 4 1 17631  
4 4 184 4 6 41208  
5 5 214 2 7 48516  
6 6 214 1 9 48747

revenues is weekly revenues, foot traffic is the number of customers per week. sick calls is how many employees called in sick that week. staff is how much sales personnel was working (not sick) during the week. You think that for your shops the amount of staff is important. But you also know that if a store is busier, you staff more sales personnel. So there is a direct connection. Just looking at the relation between revenues and amount of staff will thus be contaminated by foot traffic. The coefficient might just pick up that stores with more staff are busier stores. You want to know what the importance of staff is, holding foot traffic (how busy the store is) fixed. Stated another way, foot traffic is a confounding variable for estimating the relation between number of staff and revenues. You can see that in [Figure 4.8](#fig-dag-rs). According to your mental model, foot traffic has an arrow going into revenues and an arrow going into staff. It causally influences both you think. Thus you need to hold its influence constant to have a chance to estimate the direct arrow going from staff to revenues. Multiple regression is one approach to do so.

Here is what would happen if we would ignore the confounding influence of foot traffic. We first estimate a simple regression of store revenues on number of staff working.

m2 <- lm(revenues ~ staff, data = d5)  
summ(m2, confint = TRUE)

MODEL INFO:  
Observations: 28  
Dependent Variable: revenues  
Type: OLS linear regression   
  
MODEL FIT:  
F(1,26) = 13.79, p = 0.00  
R² = 0.35  
Adj. R² = 0.32   
  
Standard errors:OLS  
-----------------------------------------------------------------  
 Est. 2.5% 97.5% t val. p  
----------------- ---------- --------- ---------- -------- ------  
(Intercept) 19774.52 9653.48 29895.56 4.02 0.00  
staff 2604.77 1163.11 4046.44 3.71 0.00  
-----------------------------------------------------------------

We see a large and significant coefficient on staff working. One more staff working in that week is associated with 2.605 weekly revenues on average. However, your mental model in [Figure 4.8](#fig-dag-rs) suggests that this number is incorrect. The number of staff is itself determined by foot traffic, so the 2,605 is likely partially driven by stores with higher foot traffic. We put foot traffic itself into the regression now.

m3 <- lm(revenues ~ staff + foot\_traffic, data = d5)  
summ(m3, confint = TRUE)

MODEL INFO:  
Observations: 28  
Dependent Variable: revenues  
Type: OLS linear regression   
  
MODEL FIT:  
F(2,25) = 17526.33, p = 0.00  
R² = 1.00  
Adj. R² = 1.00   
  
Standard errors:OLS  
--------------------------------------------------------------  
 Est. 2.5% 97.5% t val. p  
------------------ -------- -------- --------- -------- ------  
(Intercept) 761.44 332.92 1189.97 3.66 0.00  
staff 352.95 295.45 410.45 12.64 0.00  
foot\_traffic 209.98 207.12 212.84 151.31 0.00  
--------------------------------------------------------------

All of a sudden the coefficient drops down dramatically to €352 weekly revenues per staff member. This change is as expected if foot traffic is indeed a confounding influence for the staff variable.[[9]](#footnote-232)

The main takeaway here is: one big advantage of multiple regression is that it allows you to examine the role and magnitude of individual determinants while holding others fixed. As we saw in this example, it can really matter to do so. If we would not have included foot traffic, we might have gotten the impression that staff members are way more important to store performance than they really are.

But notice, we made the judgment call to include foot traffic based on a mental model. Repeating our call from above, we need mental models to know which variables to include and when not. Notice for example that we have a variable called sick\_calls in our dataset. what happens if we include this variable into the our regression model:

m4 <- lm(revenues ~ staff + foot\_traffic + sick\_calls, data = d5)  
summ(m4, confint = TRUE)

MODEL INFO:  
Observations: 28  
Dependent Variable: revenues  
Type: OLS linear regression   
  
MODEL FIT:  
F(3,24) = 11217.80, p = 0.00  
R² = 1.00  
Adj. R² = 1.00   
  
Standard errors:OLS  
---------------------------------------------------------------  
 Est. 2.5% 97.5% t val. p  
------------------ -------- --------- --------- -------- ------  
(Intercept) 755.95 250.81 1261.08 3.09 0.01  
staff 353.42 290.76 416.09 11.64 0.00  
foot\_traffic 209.97 206.98 212.95 145.11 0.00  
sick\_calls 2.58 -115.25 120.41 0.05 0.96  
---------------------------------------------------------------

Things barely change. But that doesn’t mean that sick calls do not have an influence. They do! they reduce the number of staff working in that week and thus affect revenues. But, controlling for staff working, sick calls cannot affect revenues anymore! Your mental model tells you that including this variable into the regression does not add anything. This is a benign example, in some cases adding a variable can distort all the other variables’ coefficients. All of this is to say that we need strong theory and mental models to analyse determinants.

### 4.7.4 Some terms to know

Let’s now also take a closer look at some other information in typical regression tables.

* **Standard errors** measure the variability or dispersion of the estimated coefficients (remember [Figure 4.3](#fig-rand1)). They provide an indication of how precise the estimated coefficients are. Smaller standard errors imply that the coefficients are estimated with greater precision and are more reliable, whereas larger standard errors suggest less precise estimates and more uncertainty. And uncertainty is something we need to live with as we are only estimating the coefficient of a sample.
* The **T statistics** (t) are calculated by dividing the estimated coefficient by its standard error and help you determine the statistical significance of the coefficients. They are test statistics for a default hypothesis test H: the coefficient is zero in the population. In essence, the t statistic here measures how many standard errors the coefficient estimate is away from zero. A large t statistic indicates that the coefficient is unlikely to have occurred by chance, suggesting that the independent variable has a significant association with the dependent variable. (see [Figure 4.5](#fig-rand3)).
* We explained **p-values** above, but for completeness: P-values are associated with t statistics and are used to test the null hypothesis that the true population coefficient is zero (i.e., no relationship between the independent and dependent variables). A small p-value (typically less than 0.10) indicates that you can reject the null hypothesis, suggesting that the independent variable has a significant association with the dependent variable. On the other hand, a large p-value implies that there is insufficient evidence to reject the null hypothesis, meaning that the relationship between the independent and dependent variables may not be statistically significant (and therefore “by chance”).
* The **confidence intervals** provide a range within which the true population coefficient is likely to lie with a certain level of confidence, often 95%. If the confidence interval does not include zero, it suggests that the coefficient is statistically significant at the specified confidence level. Conversely, if the confidence interval includes zero, the coefficient is not considered statistically significant. The chosen significance level matches the confidence interval: if you decide to use a significance level of 5%, then the corresponding confidence interval is 95% (see regression output). You usually have to specify this explicitly in your statistical package.
* R. The r-squared is a measure of variation in the outcome explained by the estimated linear combination of the variables. It is sometimes used as a measure of model fit or quality. But that is dangerous interpretation. We will say more about the r-squared later.

### 4.7.5 Relative importance of variables

It is tempting to base the relative importance of variables on the coefficients obtained from the regression analysis. And one can do so, but only with extreme caution.

First, as our example with staff and foot traffic shows. Any interpretation first and foremost rests on the regression model capturing confounding effects. This also requires that our mental model is not too far off reality and that we can approximate relations in a linear way (e.g., after transformations)

Second, often, the units of measurement usually vary between the variables, making it very hard to to directly compare them. For example, the meaning of a one-unit change in distance is different from a one-unit change in weight. There are ways around this, but in complicated situations the whole idea of comparing coefficients in a table is dangerous and often a big mistake. Be very careful when trying to do this. The reason is the same as with the foot traffic and staff example. In complicated situations a complicated regression equation with many variables is often required to isolate the association between and just one ! All the other variables (called control variables) are in there just to unbias the coefficient on that one . Thus the coefficients on the control variables are not guaranteed to be unconfounded or easily interpretable.[[10]](#footnote-235) If you are sure (based on your mental model) that both coefficients can indeed be interpreted properly, one approach to assess the relative importance of variables is to compare the **standardized coefficients** in a multiple regression model. Standardized coefficients are the estimates obtained when all the variables are standardized to have a mean of zero and a standard deviation of one. By standardizing, the coefficients are expressed in units of standard deviations, which allows for a direct comparison of the effect of each predictor on the dependent variable. As mentioned in [Chapter 3](#sec-datapat), if you standardize the original data, this can lead to complications. Instead, you can simply take the regression output and multiply the unstandardized coefficient with the standard deviation of the respective variable.

Another common approach to assess the relative contribution of each variable to the R-squared (more on the R-squared below) is calculating the **incremental R-squared**. To achieve this, you first run the regression with all independent variables and note down the R-squared (0.4962 in our multiple regression above). To determine the incremental contribution of employee tenure, you exclude this variable from the regression, run it again and note down the new R-squared, which is 0.4247. The difference is the incremental R-squared of employee tenure, that is, 0.4962 - 0.4247 = 0.0715. Repeating this for service quality would yield an incremental R-squared of 0.1316, again ranking the relative importance of service quality higher than employee tenure. We like this approach even less though. Again, the issue is complicated interdependence between variables that make it very difficult to judge whether an lower increase in R-squared is meaningful or not. Often the order of variables matters a lot as a result of that. Also remember the example with revenues being explained by foot traffic, staff working, and sick calls. If you would want to estimate the impact of sick calls, you would have to throw out the staff variable in order to unblock the path between sick calls and store revenues. That will drastically lower the R-squared because staff working in our simulation explains revenues quite well. It explains y much better than sick calls. Sick calls has no incremental R-squared to staff working. But for our question of interest, that is irrelevant. Again, the main message is, when comparing R-squared contributions, carefully examine your mental model if the comparisons are apples to apples or even sensible.

## 4.8 What is a “good” (regression) model?

This is almost a philosophical, but nevertheless a very important question. When building a model to answer a business question, you need some criteria to say: “This model is good enough, let’s run with it”. Some often used criteria for this purpose are dangerous. We recommend to let yourself be guided by theory first and out-of-sample prediction accuracy second. Only have a cautious look at measures like the in-sample R-squared.[[11]](#footnote-238)

The **R-squared**, also known as the coefficient of determination, is a measure that represents the proportion of the variance in the dependent variable that can be explained by the independent variables in a regression model. It “quantifies the goodness-of-fit” of the model *to the sample* by comparing the explained variation to the total variation in the sample. We emphasized *fit to the sample* to already highlight one trap when interpreting an R-squared. It is usually an in-sample measure. One can easily build a bad model with lots and lots of variables fitting noise, which will have a high R-Squared (R ranges from 0 to 1). Models that are *overfit* (see [Chapter 5](#sec-pred)) have high R that are not indicative of their fit to new samples. The standard R increases with the addition of more variables, regardless of their relevance. Therefore, we typically look at the **adjusted R-squared**. The adjusted R-squared is a modified version of the R-squared that takes into account the number of independent variables and the sample size. It can help to penalize overly complex models, making it a more reliable measure of goodness of fit, but even the adjusted R-squared suffers from the in-sample problem.

R is also not a good measure of model quality when the goal is to identify a relation of interest. In that case, we do not need to explain a lot of variation necessarily. On the contrary, if we find a proxy for a determinant of interest that does not explain much of , but which is unconfounded by other things, then we will get a cleaner estimate of the coefficient and a low R. That is fine, in this case, we care more about the clean estimate of the coefficient.

So, what is a good model. We believe it depends on the purpose of your analysis. If it is diagnostic analysis, a good model can again only be assessed via theory: your mental models. If the model manages to isolate the variables your care about, blocks confounding influences, etc. it is a good model. If the purpose is prediction, then *out-of-sample* prediction accuracy is the relevant criteria. A model that fits well on data it hasn’t seen is a much more believable model. We will spend quite some time in [Chapter 5](#sec-pred) on discussing how to assess this.

## 4.9 Logistic regression

Next to linear regressions, logistic regressions are the next most often used regression tool. It belongs to the family of generalized linear models (GLM). The main differences between these two types of regression lie in the nature of the dependent variable, the function used to model the relationship, and the interpretation of the results. Linear regression is used when the dependent variable is continuous, meaning it can take any real value within a certain range. Logistic regression is specifically designed for binary dependent variables, which have only two categories or outcomes, such as success/failure or yes/no. This is very useful for determinant analysis like: What determined whether a customer did respond to a campaign (a yes/no outcome).

Because the outcome is binary(0, 1), using a normal linear regression could result in predictions that lie far away from the (0; 1) interval. A logistic regression thus needs to ensure that the outcome can never be below 0 and never above 1. It does so by taking a the linear part of the model (the part) and pushes it through a function (the so-called logistic function) that pulls large values to 1 and too small values to 0. It gives the relation a characteristic S-shaped curve that maps any real-valued input to a value between 0 and 1 ([Figure 4.9](#fig-logreg)). Very useful when modeling probabilities.

|  |
| --- |
| Figure 4.9: The form of the logistic function sqaushing the linear model into the 0-1 range |

The only difficulty that the logistic transformation brings about is that it complicates the interpretation of coefficient magnitudes. That is because the S-curve now makes the relation between and non-linear. In logistic regression, the coefficients are expressed in terms of the logarithm of the odds ratio, which describes the change in the odds of the event occurring for a one-unit change in the independent variable.

To facilitate interpretation, these coefficients can be exponentiated to obtain the odds ratios, which provide a more intuitive understanding of the relationship between the independent variables and the binary outcome. Let’s consider a hypothetical example to illustrate logistic regression and its interpretation. Assume we are a broker and send an offer around for a counseling on retirement saving. We wanted to know whether older customers were more likely to respond to this campaign. Here is some simulated data and the result of throwing the data into a logistic regression:

set.seed(0374)  
# simulating an x variable with 100 random obs ofage between 24 and 58  
x = sample(24:58, size = 100, replace = TRUE)  
# simulating some probabilities  
y = inv\_logit(-2 + 0.3 \* (x - mean(x)) + rnorm(100, 0, 1.3))  
# turning probabilities into a binary 1, 0 variable  
y = ifelse(y > 0.5, TRUE, FALSE)  
# estimating a model  
m5 <- glm(y ~ x, family = binomial(link = "logit"))  
summ(m5)

MODEL INFO:  
Observations: 100  
Dependent Variable: y  
Type: Generalized linear model  
 Family: binomial   
 Link function: logit   
  
MODEL FIT:  
χ²(1) = 86.39, p = 0.00  
Pseudo-R² (Cragg-Uhler) = 0.79  
Pseudo-R² (McFadden) = 0.66  
AIC = 49.40, BIC = 54.61   
  
Standard errors:MLE  
-------------------------------------------------  
 Est. S.E. z val. p  
----------------- -------- ------ -------- ------  
(Intercept) -18.85 4.42 -4.27 0.00  
x 0.41 0.09 4.33 0.00  
-------------------------------------------------

where p is the probability of of coming to the counsel meeting, and odds represent the ratio p / (1 - p).

Plugging in the values from our fitted model, we get:

Now, let’s suppose we want to predict the probability of getting counsel for a 40 year old person. We can use the logistic regression equation to calculate the log odds of passing:

To convert the log odds to a probability, we apply the logistic function:

So, the probability of passing of a 40 year old responding to the campaign was approximately 8%.

To interpret the coefficients, we can calculate the odds ratio for the independent variable (age) by exponentiating :

The odds ratio of 1.51 indicates that for each year of age, the odds of showing up to the counselling increase by a factor of 1.51, holding all other variables constant. You can use the respective functions in your statistical packages to get at these more intuitive numbers.

For presentations it is often easier though to set all variables , etc to their average values, plug in relevant values for the variable of interest and plot the change in probabilities. We often found this to be the most intuitive way of presenting the non-linear nature of the - relations in logisitic regressions. As you can see, after a certain age their is a steep increase in the probability of showing up that you would not have expected based on the coefficients.

|  |
| --- |
| Figure 4.10: Presenting logisitic regression outcomes |

## 4.10 Conclusion

The material we covered in this chapter is quite complex and deep. Rather than focusing on the mechanics we tried to focus on important intuition. We still think it is worth, as you get more and more familiar with the tools to dive deeper into it by practicing. Fully understanding these concepts takes practice. With the tools we introduced here, linear and logistic regressions, many applications will be served and if you will already make a large step forward to data-driven decision-making if you have a rudimentary grasp of them. Particularly to find explanations for **why** you observe certain patterns in your data. For the purpose of creating predictive models, we will continue to discuss regressions and more advanced models in Chapter 5.

# 5. Predictive Analytics

## 5.1 Learning goals

1. Learn how to do principled data-based forecasting for decision making
2. Evaluate a prediction models quality based on standard metrics and theory

## 5.2 Scope of this chapter

In the language of machine learning, this chapter is about supervised learning.[[12]](#footnote-252) Supervised learning is what we mostly think of as prediction models, where we have inputs (called independent variables, features, or predictors) and have one or more outputs (often called, responses, dependent variables, or outcome variables). Depending on the type of the outcome variable, two types of prediction (or learning) problems are distinguished: regression problems (predicting a continuous response) and classification problems (predicting a binary or categorical response). Both categories feature an ever expanding set of models and algorithms. We do not have the time to cover this expanding model space in any way that does it justice. That would be a course (or two) by itself. Instead, our goal for this chapter is to provide you with the foundation of predictive analysis and some practice with generalized linear regression models. Even given the plethora of more advanced algorithms, these are incredibly useful model families for most applications. You will be surprised how far the standard approaches will carry you.

After introducing the basic concepts about the sources of prediction errors, we will discuss how to assess whether a particular model predicts well. The trend towards machine learning and AI has brought with it a rich body of knowledge on how to assess the prediction quality of even complex models and we will draw on more recent advances in this field. Once you know how to assess model performance, we will showcase a full prediction pipeline and close with an extension: a brief treatment of the powerful toolkit enabled by penalized regression models.

## 5.3 What is predictive analysis?

Predictive analysis is asking questions, such as “What will happen?”, “What is the most likely response, given a set of conditions?”, etc. It encompasses a huge and growing field of prediction and forecasting models—from classic time-series models to the most complex large language models. In this chapter we will focus mostly on the core concepts behind data-driven predictions. We will explain what exactly the issues are that make extrapolating from a historical sample to (yet) unseen data so difficult. These issues are quite broad and but provide a good foundation not only for designing useful predictive analyses, but also for a basic understanding of modern AI models behind driverless cars, generative Art, or generative text models, such as ChatGPT. At their heart, all of these applications are prediction models.

It is important to understand that predictive analysis has a different goal than diagnostic analysis. Thus it also has different criteria to assess model quality. In [Chapter 4](#sec-diag), we stressed that diagnostic model quality can ultimately only be assessed via theory (your mental model). That is because diagnostic analysis asks “Why?” questions. Predictive analysis is less strict. You can easily have a prediction model that provides quite accurate forecasts without being even close to modelling causal relations. For example, the developers of the by now infamous early “Google Flu Trends” model, which predicted flu outbreaks using Google search terms, reported weeding out highly predictive but unrelated search terms related to basketball (Ginsberg et al. 2009). Basketball terms were highly predictive of flu outbreaks, because the main US basketball season is March, which is also the main flu season. So basketball search terms were just predicting season, not causally related to flu outbreaks itself. (See also the Lazer et al. 2014 discussion of GFT as a parable for traps utilizing big data.). Still, while weeding out search terms to basketball helps making the model more robust, a predictive model does not need to isolate causal relations. Consider a somewhat trivial example similar to the one used in [Chapter 4](#sec-diag) ([Figure 4.8](#fig-dag-rs)). Imagine we are interest in predicting next week’s retail store sales and have as inputs the number of staff and number of purchases in the previous week. If more store staff is associated with more purchases, including both number of purchases and number of staff is fine for prediction purposes. But you would not include number of purchases into your model if you would run a diagnostic analysis of the importance of the number of staff for sales. That is because part of the expected effect of staff on sales is via more purchases. Again, the point here is that models for predictive purposes and diagnostic purposes have different but equally important goals. Thus, they need to be designed based on different criteria.

## 5.4 The source of prediction errors

### 5.4.1 Overfitting vs underfitting

In the previous section we argued that models for diagnostic analysis and predictive analysis are not constructed using the same criteria. In [Chapter 4](#sec-diag), we argued that just adding variables is a bad idea, as adding a variable to your model can sometimes bias the coefficient that you are after. But in predictive analysis we do not care so much about the coefficients–about understanding how the determinants of our outcome of interest relate–only that they are determinants of the outcome. So, is it okay to just throw all of the hypothesized determinants into a model? The answer is still no. Here is why.

By adding more variables, a model becomes more complex, and will often better explain variation in the sample that the model is fitted on. But the reason might not be that the model actually predicts (new data) better. Instead, the more variables (parameters) a model has, the more likely it is that a newly included variable was chosen in error and accidentally fits noise. This issue is called **overfitting**. It is a very important concept, and we will try to explain it with some illustrations.

|  |
| --- |
| Figure 5.1: Overfitting of data with more complex models |

For [Figure 5.1](#fig-overfit), we simulated five samples of the form with the same x variables and drawn from a distribution. Each model we fit with two types of models. A classic linear one and a polynomial one . Since we know the true relation, we know that the polynomial version has an unnecessary variable in there . Usually, we do not know that, however. We never know for sure what the real world dynamics look like. That is why we used simulations here; to make the following points: Notice how the (wrong) polynomial regression always has the same or better than the (correct) linear model. This is overfitting. Even though the polynomial version is wrong. It fits the sample it got trained on better. Why does overfitting arise? This is because the additional variable starts fitting the noise part of the sample. The x values are always the same in each sample, the only thing that changes, and that gets picked up by to a bigger or smaller extend is the unmeasured influences in the error term . And since these change from sample to simple, picking up the sample is bad news for out-of-sample predictions!

Overfitting is a serious issue (also for diagnostic models, btw). Notice how especially the fitted lines from the really curved samples with high are off. These will very likely be a terrible fit for the other samples. *Overfitted models will likely fit the sample they got trained on better than less overfit models, but they will predict new data worse*. And, since the theme is prediction, we really only care about how good a model fits new data.

To illustrate the severity of the issue, we computed a common prediction error metric, the mean squared error (MSE) for each of the five samples. The MSE is simply the average of all aquared prediction errors: . In the [Table 5.1](#tbl-mse1), you see the difference in in-sample MSE between the polynomial model and linear model. In the last column, we also report the average difference in *out-of-sample MSE* for the other samples. The logic behind these out-of-sample averages is illustrated in [Figure 5.2](#fig-overfit2). For each of the five samples and each model (poly or linear), we fitted to model to a sample (say sample 2 in row 2). We then used the fitted model to make predictions for each observation in the other four samples (for row 2, samples 1, 3, 4, and 5).

|  |
| --- |
| Figure 5.2: Using sample 2 fit to predict outcomes in the other samples |

From those predictions, we computed the out-of-sample MSE for each of the four samples for each model. We then computed the difference in MSE between the polynomial and the linear model. Finally we averaged the difference in out-of-sample MSEs for all samples not used to fit the models (again, for row 2, the average of the differences for samples 1, 3, 4, and 5).

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Table 5.1: Differences in prediction errors (polynomial model MSE - linear model MSE)   | Sample | In-Sample MSE Difference | Avg. Out-of-Sample MSE Difference | | --- | --- | --- | | 1 | -8.5 | -15.1 | | 2 | -1419.8 | 1770.0 | | 3 | -778.7 | 901.7 | | 4 | -2735.2 | 4602.0 | | 5 | -8.4 | -15.1 | |

The important thing to note here is that for all samples where the polynomial model fits better (has much lower mean squared error as shown by a negative in-sample MSE difference), it fits worse out-of-sample to the same degree! If we would average over the out of sample MSE difference we find that the polynomial model is on average worse by 1448.7 in terms of MSE difference.[[13]](#footnote-264)

In [Chapter 4](#sec-diag) we discussed that one should not use in-sample measures of goodness-of-fit such as to judge the quality of a model for diagnostic analysis purposes. You can only judge a diagnostic model using theory. In-sample overfitting is another reason why you do not want to trust in-sample s to judge model quality—for diagnostic or predictive analysis purposes.

### 5.4.2 Determinants of overfitting risk

The above simulation shows that model complexity increases the chance of overfitting. This is always also relative to the amount of data. It is difficult to give concrete guidelines here. But if you have a lot of data, then you can get away with more complex models too. At the same time, complex data, with lots of complicated interactions and where a lot of the variation in the data is left unexplained (relegated to the error term) are also data where it is also easier to accidentally fit in-sample noise. We thus need a way to judge overfitting risk by a model, which is what we discuss next.

### 5.4.3 The bias variance trade-off

A nice feature of MSE as an error metric is that it can be easily decomposed into two components that tell us more about the reasons for prediction errors. Imagine we have some prediction model with inputs and trained on some data . Then the mean squared error of such a model is:

There are two sources of errors according to this formula. One is variance, and this is the term that is affected by overfitting. Variance reflects errors arising from sensitivity to small fluctuations in the sample used to fit the modes. The second term, Bias, reflects systematic errors, such as those arising from an *underfitted* model (one that misses a systematic determinant and thus makes a systematic error). If it helps, you can think of the bias variance trade-off as an overfitting/underfitting trade-off. Sometimes it is better to have simpler models that are likely to underfit or make a systematic mistake (a bit of Bias), than complex models that might be prone to large overfitting (high variance).

This trade-off is quite general, irrespective of the error metric. And there often is no clear ex-ante guideline when to expect that a certain model will have a better trade-off than another. For that reason, we often have to evaluate different models and simply choose the one that seems to have the better bias variance trade-off in our particular setting. To do so, we need to discuss how to assess prediction performance in a principled way next.

## 5.5 Assessing prediction performance

### 5.5.1 Cross-validation

One can find two approaches to judge the quality of prediction models for their out-of-sample prediction quality. One is cross-validation, the other is looking at information criteria, such as the Akaike Information Criterion ([AIC](https://en.wikipedia.org/wiki/Akaike_information_criterion)). In this course, we will focus on cross-validation, since it is the most widely used approach in machine-learning.

The idea behind cross-validation is similar in spirit to what we did to generate the results reported in [Table 5.1](#tbl-mse1). And this is the method we will focus on in the remainder of this course for evaluating prediction model quality.

Cross-validation is a resampling method. We take a sample of data and randomly partition it into k parts. We then fit the model on one part (consisting of k-1 parts, called the training set) and fit the model to the left-out part k. We do this for all k parts, compute MSE (or another suitable error metric) on the left-out part each time, and finally compute the average of all k error metrics.

Of course, by randomly partitioning the data into training and testing sets and then averaging across partitions, we do not really evaluate completely new data. Instead, the purpose of cross-validation is to *estimate* how accurate the model would be if it were to be thrown at truly new data. The cross-validated average out-of-sample MSE (or other error metrics) is thus only an estimate. But a very useful one. Let’s illustrate this with another simulated example. We will use the [tidymodels](https://www.tidymodels.org/) packages for our prediction exercises from now on:

library(tidymodels)

If you want to learn more about using tidymodels for machine learnings, the open [webook](https://www.tmwr.org/) by Max Kuhn and Julia Silge is a great start.

We will use a simulated data set used by IBM to illustate a Watson data science pipline ([source here](https://github.com/IBM/employee-attrition-aif360)).

data(attrition)  
glimpse(attrition)

Rows: 1,470  
Columns: 31  
$ Age <int> 41, 49, 37, 33, 27, 32, 59, 30, 38, 36, 35, 2…  
$ Attrition <fct> Yes, No, Yes, No, No, No, No, No, No, No, No,…  
$ BusinessTravel <fct> Travel\_Rarely, Travel\_Frequently, Travel\_Rare…  
$ DailyRate <int> 1102, 279, 1373, 1392, 591, 1005, 1324, 1358,…  
$ Department <fct> Sales, Research\_Development, Research\_Develop…  
$ DistanceFromHome <int> 1, 8, 2, 3, 2, 2, 3, 24, 23, 27, 16, 15, 26, …  
$ Education <ord> College, Below\_College, College, Master, Belo…  
$ EducationField <fct> Life\_Sciences, Life\_Sciences, Other, Life\_Sci…  
$ EnvironmentSatisfaction <ord> Medium, High, Very\_High, Very\_High, Low, Very…  
$ Gender <fct> Female, Male, Male, Female, Male, Male, Femal…  
$ HourlyRate <int> 94, 61, 92, 56, 40, 79, 81, 67, 44, 94, 84, 4…  
$ JobInvolvement <ord> High, Medium, Medium, High, High, High, Very\_…  
$ JobLevel <int> 2, 2, 1, 1, 1, 1, 1, 1, 3, 2, 1, 2, 1, 1, 1, …  
$ JobRole <fct> Sales\_Executive, Research\_Scientist, Laborato…  
$ JobSatisfaction <ord> Very\_High, Medium, High, High, Medium, Very\_H…  
$ MaritalStatus <fct> Single, Married, Single, Married, Married, Si…  
$ MonthlyIncome <int> 5993, 5130, 2090, 2909, 3468, 3068, 2670, 269…  
$ MonthlyRate <int> 19479, 24907, 2396, 23159, 16632, 11864, 9964…  
$ NumCompaniesWorked <int> 8, 1, 6, 1, 9, 0, 4, 1, 0, 6, 0, 0, 1, 0, 5, …  
$ OverTime <fct> Yes, No, Yes, Yes, No, No, Yes, No, No, No, N…  
$ PercentSalaryHike <int> 11, 23, 15, 11, 12, 13, 20, 22, 21, 13, 13, 1…  
$ PerformanceRating <ord> Excellent, Outstanding, Excellent, Excellent,…  
$ RelationshipSatisfaction <ord> Low, Very\_High, Medium, High, Very\_High, High…  
$ StockOptionLevel <int> 0, 1, 0, 0, 1, 0, 3, 1, 0, 2, 1, 0, 1, 1, 0, …  
$ TotalWorkingYears <int> 8, 10, 7, 8, 6, 8, 12, 1, 10, 17, 6, 10, 5, 3…  
$ TrainingTimesLastYear <int> 0, 3, 3, 3, 3, 2, 3, 2, 2, 3, 5, 3, 1, 2, 4, …  
$ WorkLifeBalance <ord> Bad, Better, Better, Better, Better, Good, Go…  
$ YearsAtCompany <int> 6, 10, 0, 8, 2, 7, 1, 1, 9, 7, 5, 9, 5, 2, 4,…  
$ YearsInCurrentRole <int> 4, 7, 0, 7, 2, 7, 0, 0, 7, 7, 4, 5, 2, 2, 2, …  
$ YearsSinceLastPromotion <int> 0, 1, 0, 3, 2, 3, 0, 0, 1, 7, 0, 0, 4, 1, 0, …  
$ YearsWithCurrManager <int> 5, 7, 0, 0, 2, 6, 0, 0, 8, 7, 3, 8, 3, 2, 3, …

This is a fictional dataset used to examine the question of what possible determinants of employee attrition are. Note, this is usually a diagnostic question. Instead, we turn to the purely predictive question: Can we predict who left the organization, irrespective of the reason?

The models we are about to fit, are not so much the focus right now. Here we only want to illustrate the use of cross-validation. First, we use the vfold\_cv() function out of the tidymodels’s rsample package to partition the sample five times. Again, we do this so that we can come up with an estimate of which model will likely do better if used to predict actual dropouts of future employees.

set.seed(456)  
folds <- vfold\_cv(attrition, v = 5)  
print(folds)

# 5-fold cross-validation   
# A tibble: 5 × 2  
 splits id   
 <list> <chr>  
1 <split [1176/294]> Fold1  
2 <split [1176/294]> Fold2  
3 <split [1176/294]> Fold3  
4 <split [1176/294]> Fold4  
5 <split [1176/294]> Fold5

As you can see, each cut has cut the 1,470 data points into 1,176 observations for training and 294 observations for evaluating model performance. Five times 294 yields you the full sample of 1,470 data points again.

We now fit a few ad-hoc models to each fold. The tidymodels package has a nice pipeline structure that allows us to fit a list of different model types and preprocess steps. For what we are doing right now, it is a bit overkill, but we will still start introducing it already. Our outcome variable is binary (Attrition: “left organization” yes / no), so we start with a simple logistic regression. We start with WorkLifeBalance and MonthlyIncome as predictors for the outcome variable Attrition.

# define the model type and a workflow including the predictors and the outcome  
log\_model <- logistic\_reg(mode = "classification", engine = "glm")  
  
logistic\_1 <-   
 workflow() |>   
 add\_model(log\_model) |>   
 add\_formula(Attrition ~ WorkLifeBalance + MonthlyIncome)  
  
fit\_rs\_1 <-   
 logistic\_1 |>   
 fit\_resamples(folds)  
  
collect\_metrics(fit\_rs\_1)

# A tibble: 3 × 6  
 .metric .estimator mean n std\_err .config   
 <chr> <chr> <dbl> <int> <dbl> <chr>   
1 accuracy binary 0.839 5 0.00989 Preprocessor1\_Model1  
2 brier\_class binary 0.131 5 0.00665 Preprocessor1\_Model1  
3 roc\_auc binary 0.647 5 0.0135 Preprocessor1\_Model1

We get outputs for three metrics: accuracy, brier\_class, and roc\_auc (ROC area under the curve). These are default error metrics to judge the performance of classification problems (We will discuss what that means in the next section). Tidymodels automatically chose to report those because we told it we are fitting a classification model.

The mean column shows the average of both metrics across the five 294-observation folds, we generated with cross-validation. The average accuracy is 83.8%, which means we got 84% of labels correct. That is not as great as it sounds though. Accuracy is a tricky error metric whenever you have a disbalance in labels. We do have much

The std\_err column shows you the standard error across the five folds. It is an indication of the variation of both metrics across the five folds, which is going to be important when comparing different models. In fact, the distribution of our outcome variable in the full dataset is like this

attrition |>   
 summarize(percent\_label = n() / nrow(attrition),  
 .by = Attrition)

Attrition percent\_label  
1 Yes 0.1612245  
2 No 0.8387755

So, if we would just use “no”-model, just predict every employer to stay in the organization, we would also get 83,8% of our predictions correct! Our first model is really is not better than an obviously lazy model.

We will discuss how to interpret roc\_auc in a minute. First, we also fit another model for comparison. In this one we add two more predictors: DistanceFromHome and BusinessTravel, assuming that both might affect the desirability of a job.

logistic\_1 |>   
 update\_formula(  
 Attrition ~ WorkLifeBalance + MonthlyIncome + DistanceFromHome + BusinessTravel  
 ) |>   
 fit\_resamples(folds) |>   
 collect\_metrics()

# A tibble: 3 × 6  
 .metric .estimator mean n std\_err .config   
 <chr> <chr> <dbl> <int> <dbl> <chr>   
1 accuracy binary 0.838 5 0.00935 Preprocessor1\_Model1  
2 brier\_class binary 0.128 5 0.00621 Preprocessor1\_Model1  
3 roc\_auc binary 0.673 5 0.0136 Preprocessor1\_Model1

We get a very similar accuracy score and a slightly higher roc\_auc. Now, does this mean the second model is better or not that different? What do these metrics tell us?

### 5.5.2 Common error metrics

Mean squared error, accuracy, roc\_auc, are all different error metrics. Which on to use depends on the learning problem and the data. In machine learning lingua, the type of analysis we just did is called a *classification* problem. We tried to classify observations into one of two groups, those the left the organization and those that did not. When your have an outcome variable that is binary or categorical (e.g, “delayed”, “on-time”, “cancelled”) you usually have a classification problem. For those, measures like accuracy and roc\_auc are commonly used. Learning problems involving continuous outcomes are often called *regression* problems. For those error metrics like MSE or mean absolue error (MAE) are often appropriate.

**Accuracy**. We already discussed accuracy as a classification error metric and some of its issues. Another one is that often, not all classification errors are equal. Is it worse for us to label people who left the firm as staying or people who stayed as leaving? Often it makes sense to decompose accuracy based on the type of misclassification.

**ROC AUC (Receiver Operator Characteristic Area Under The Curve)**. Generally, an roc\_auc value is between 0.5 and 1, with 1 being a perfect prediction model. To explain what roc\_auc measure takes a bit of time, but it is worth it as it highlights some of the unique issues with classification predictions. For simplicity, let us fit our first model to the whole data and look at the in-sample predictions (normally you would again examine predictions out-of-sample). We can see what the true classification label is (“Yes” or “No”) and what probabilities our logistic regression assigns to each class:

Attrition .pred\_No .pred\_Yes  
1 Yes 0.7026551 0.2973449  
2 No 0.8493398 0.1506602  
4 Yes 0.7933363 0.2066637  
5 No 0.8097954 0.1902046  
7 No 0.8204417 0.1795583  
8 No 0.7810288 0.2189712

We could then pick a probability threshold to assign an observation a prediction. Say, if .pred\_Yes is greater or equal 25% we classify the observation as of the class “Yes”

Attrition .pred\_No .pred\_Yes .pred  
1 Yes 0.7026551 0.2973449 Yes  
2 No 0.8493398 0.1506602 No  
4 Yes 0.7933363 0.2066637 No  
5 No 0.8097954 0.1902046 No  
7 No 0.8204417 0.1795583 No  
8 No 0.7810288 0.2189712 No

With this threshold (“Yes” if .pred\_Yes >= 25%), we would get the following classification result.

Attrition .pred N Condition  
1 Yes Yes 36 True positive  
2 No No 1179 True negative  
3 Yes No 201 False negative  
4 No Yes 54 False positive

The terms true positive, true negative and so on are important. In a binary classification, you define one class as the “positive” class and we assigned the “Yes, has left the organization” condition to it. As you can see, the simple model does not do a good job of classifying the “Yes” condition correctly. we have 201 false negatives (people who have left that we incorrectly labeled as stayed). Out of all “Yes” cases we only labeled correctly. This is called the *true positive rate* (another common name is: *recall*). You can also think of such rates and tables as further digging into the accuracy statistic.

What a ROC curve does is it plots the true positive rate and false positive rate for a range of classification thresholds between 0 and 1. In our example, we picked one threshold: .pred\_Yes >= 25%. If we computed true positive and false positive rates for a range of probabilities between .pred\_Yes >= 0% (Always assume “Yes”) and .pred\_Yes >= 100% (bascially always say “No”), then we can plot the true positive and false positive rates in a graph that usually looks like a curve:

|  |
| --- |
| Figure 5.3: Comparing ROC curves for two models |

This is the ROC curve. To understand this graph, think about the corners of it. Say, we use as threshold .pred\_Yes >= 0% (always predict “Yes”). That is the top right corner of the graph. Intuitively, when we always predict “Yes”, then we will always get all truly positive observations classified correctly (true positive rate will be 1). But, we will always misclassify all true “No” cases (our true negative rate will be zero or, equivalently, our false positive rate will be 1) The other extreme case is the .pred\_Yes > 100% (always predict “No”). Again, intuitively, when we always predict “No” then we can never predict the true “Yes” cases correctly but we will always predict the true “No” cases correctly. That is the 0, 0 bottom left of the graph.

A good prediction model thus has a curve that is as high as possible. Compare for example, the red line, the logistic regression model with DistanceHome and BusinessTravel as additional variables. For almost each level of false positives (i.e., for the same level of getting “Yes” wrong) it gets more “Yes” correctly classified. This is what the higher sitting curve implies and it is a good sign.

The roc\_auc is simply a summary measure of ROC curve graphs. It measures the area under the curve. Since the graph is usually plotted on the x and y axis from 0 to 1, the maximum area under the graph can also only be 1. The higher the curve is, the closer the area under the curve will be to 1. So higher values are better. When we compare the cross-validated average roc\_auc of the simple model (0.647, SE: 0.0135) and the model including DistanceHome and BusinessTravel (0.673, SE: 0.0136) then the expanded model seems to do a better job out-of-sample too. Is this difference () meaningful though? Looking at the standard errors of both cross-validation tables (0.0135 and 0.0136), it is borderline, as it is not too implausible to get such a difference also from sampling variation (rather than a true difference in predictive ability).

We spend quite some time on ROC because it also allowed us to introduce some important terms and thinking related to classification problems. Turning to regression problems that deal with continuous outcomes, things become about easier.

**Mean squared error (MSE)**. MSE is a standard error metric for continuous outcomes that—because of squaring the errors—penalizes large errors more severely than small ones.

MSE is the most common error metric, probably also because it is mathematically very convenient to work with. For example, as we have seen above, it is easy to decompose the MSE into a component due to a bias in the model and a component that is due to the variance of the model. The mean squared error is obviously appropriate, if we have at a prediction problem with a squared loss function (it is the risk function corresponding to the expectation with a squared loss function). But we are not always in such situations. For example, MSE weighs outliers very strongly, which is not often desirable. In such cases, metrics like the mean absolute error, or versions based on the median, are commonly used. A final note regarding MSE is that it is often easier to take the square root of it and interpret the root mean squared error RMSE, because it the RMSE is in the original unit scale of .

**Mean absolute error (MAE)**

The median absolute error is another common error metric, and defined very similarly to MSE. In contrast to MSE, it does not penalize large error more severely, which is why it is less affected by outliers.

### 5.5.3 The connection between error metrics, loss functions, and models

So far, we only discussed the most common default error metrics. There is a large list of alternatives that we have no chance to give credit to in the course. That does not mean they are not important, but they are usually tied to specific loss functions and prediction problems. We believe you will encounter them naturally in those settings. What you should be aware of is that your choice should be partly informed by what loss function you think you are operating with. In fact, loss functions not only dictate what error metric to choose but also which models to choose. Every algorithm behind a model is a combination of an algorithm that can fit certain functional forms to data and a loss function that dictates what criteria the best fitting function should satisfy. For example, it is quite easy to show that if a) you care about squared losses, b) you want to fit linear models, and c) you want the *best unbiased* linear predictor, then math tells you to use a classic linear regression model.

The point here is to highlight how these different concepts tie together from a decision making angle (after all, the course is called data-driven decision making). Loss functions, quantifying what errors are more or less costly to a decision, are an important driver of the data analysis methods used to make decisions.

## 5.6 A full prediction example

### 5.6.1 The data

We covered a lot of ground in this chapter. We will now put it all together in an example. Imagine you are working for a large European electronics market chain. In recent years, inspired by Apple’s genius bar, the chain has started to install and offer a counter that offers laptop and mobile phone repair services for items bought in its shops. These repair counters have been rolled out to a few stores during the last years and you are tasked with predicting the amount of FTEs needed to staff repair counters in those stores that do not have them yet. A crucial factor for staffing is the amount of time a repair service needs, which varies widely. Your firm’s service software tracked the self-reported time of repair services and some other information regarding the service job. You want to test whether you can build an accurate prediction model that would help you predict service hours and thus FTEs needed. This would greatly help you calculate the cost estimates for these new repair counters. You look at the data from last month:

library(poissonreg)  
glimpse(repair\_dta)

Rows: 4,072  
Columns: 7  
$ months\_since\_purchase <dbl> 18, 11, 5, 12, 18, 18, 4, 22, 7, 19, 15, 16, 10,…  
$ month <int> 12, 12, 1, 9, 9, 12, 1, 1, 1, 3, 10, 11, 8, 2, 1…  
$ has\_ext\_warranty <dbl> 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, …  
$ item\_type <chr> "appliances", "laptop\_premium", "mobile\_mp", "la…  
$ employee\_id <chr> "53", "34", "4", "7", "29", "9", "49", "7", "11"…  
$ purchase\_price <dbl> 794, 1598, 471, 1131, 165, 568, 932, 643, 278, 5…  
$ time\_to\_repair\_hrs <dbl> 14, 14, 1, 28, 18, 1, 7, 23, 0, 14, 4, 4, 3, 8, …

You first look at the outcome variable: time\_to\_repair\_hrs, which is the time it takes to repair the item (in hours).

repair\_dta |>   
 ggplot(aes(x = time\_to\_repair\_hrs)) +   
 geom\_histogram(binwidth = 5) +   
 scale\_y\_continuous(expand = expansion(c(0.01,0.05))) +   
 scale\_x\_continuous(n.breaks = 20) +  
 theme(panel.grid.major.x = element\_blank())

|  |
| --- |
| Figure 5.4: Always check how your outcome variable is distributed |

We can see substantial variation. Most repairs take less than 20 hours, but there are a rare few that seem to take forever. Let us see whether we can predict repair time with a model. We start by building a tidymodels workflow. First, we prepare the data. First, we will put 20% of the sample away for later.

### 5.6.2 Building a model testing pipeline

set.seed(123)  
splits <- initial\_split(repair\_dta, prop = 0.8)  
repair\_test <- testing(splits)  
repair\_other <- training(splits)  
splits

<Training/Testing/Total>  
<3257/815/4072>

It is good practice to keep a part of your sample away for a final evaluation at the end. Imagine fitting and examining multiple models and specifications on a training set. Even using cross-validation to estimate out-of-sample performance of the models, decide on preprocessing steps, and so on. Even then, you still want hold part of the data back. So that when you are done, you have a fresh out-of-sample comparison that you did not use for model tuning and selection. This is most important when you have more advance models that require you to tune model parameters, like a penalty parameter.

Next we prepare our cross validation folds.

folds <- vfold\_cv(repair\_other, v = 5)  
print(folds)

# 5-fold cross-validation   
# A tibble: 5 × 2  
 splits id   
 <list> <chr>  
1 <split [2605/652]> Fold1  
2 <split [2605/652]> Fold2  
3 <split [2606/651]> Fold3  
4 <split [2606/651]> Fold4  
5 <split [2606/651]> Fold5

Now we define preprocessing steps and a model type. The code below is quite a bit different to how we normally write code for linear regressions. This is the tidymodels way of setting up a model fitting pipeline, consisting of data pre-processing steps, model definitions, and then assessment, fitting, and prediction. While it is definitely a bit verbose and more lines of code, writing pipelines such as this, helps a lot in evaluating and testing a larger number of more involved models without making copy/paste mistakes. We thus want to introduce you to this type of coding as well.

# Recipe  
# A recipe is a description of the steps to be applied to a data set in order   
# to prepare it for data analysis.  
# Here we just standardize all numeric predictors to have zero mean and standard  
# deviation one.  
base\_processing <-   
 recipe(# define the outcome and denote all other variables as predictors  
 time\_to\_repair\_hrs ~ .,   
 data = repair\_other  
 ) |>   
 step\_normalize(all\_numeric\_predictors()) |>   
 step\_dummy(item\_type)  
base\_processing

── Recipe ──────────────────────────────────────────────────────────────────────

── Inputs

Number of variables by role

outcome: 1  
predictor: 6

── Operations

• Centering and scaling for: all\_numeric\_predictors()

• Dummy variables from: item\_type

summary(base\_processing)

# A tibble: 7 × 4  
 variable type role source   
 <chr> <list> <chr> <chr>   
1 months\_since\_purchase <chr [2]> predictor original  
2 month <chr [2]> predictor original  
3 has\_ext\_warranty <chr [2]> predictor original  
4 item\_type <chr [3]> predictor original  
5 employee\_id <chr [3]> predictor original  
6 purchase\_price <chr [2]> predictor original  
7 time\_to\_repair\_hrs <chr [2]> outcome original

Building on the base processing, we define six different model specifications. The six models are a compbination of three different regression equations and two model types

# We want to fit classic regressions for now:  
lin\_model <- linear\_reg(mode = "regression", engine = "lm")  
pois\_model <- poisson\_reg(mode = "regression", engine = "glm")  
  
# By selecting different predictors we define different specifications  
regspec1 <-   
 base\_processing |>   
 step\_select(time\_to\_repair\_hrs, has\_ext\_warranty, purchase\_price, skip = TRUE)  
  
regspec2 <-   
 base\_processing |>   
 step\_select(time\_to\_repair\_hrs, has\_ext\_warranty, purchase\_price,   
 starts\_with("item\_type"), skip = TRUE)  
  
regspec3 <-   
 base\_processing |>   
 step\_select(time\_to\_repair\_hrs, has\_ext\_warranty, purchase\_price,   
 starts\_with("item\_type"), months\_since\_purchase, skip = TRUE)

regspec3

── Recipe ──────────────────────────────────────────────────────────────────────

── Inputs

Number of variables by role

outcome: 1  
predictor: 6

── Operations

• Centering and scaling for: all\_numeric\_predictors()

• Dummy variables from: item\_type

• Variables selected: time\_to\_repair\_hrs has\_ext\_warranty, ...

### 5.6.3 Testing the model candidates

Next we create a so called workflow set. This set captures basically all the combinations of pre-processing recipes and models families that we want to evaluate.

set1 <-   
 workflow\_set(  
 preproc = list(m1 = regspec1, m2 = regspec2, m3 = regspec3),  
 models = list(LR = lin\_model, PR = pois\_model),  
 cross = TRUE  
 )  
  
wf\_rs\_fits <-   
 set1 |>   
 workflow\_map("fit\_resamples", resamples = folds)

collect\_metrics(wf\_rs\_fits)

# A tibble: 12 × 9  
 wflow\_id .config preproc model .metric .estimator mean n std\_err  
 <chr> <chr> <chr> <chr> <chr> <chr> <dbl> <int> <dbl>  
 1 m1\_LR Preprocessor1… recipe line… rmse standard 17.8 5 0.180   
 2 m1\_LR Preprocessor1… recipe line… rsq standard 0.137 5 0.0137  
 3 m1\_PR Preprocessor1… recipe pois… rmse standard 17.8 5 0.135   
 4 m1\_PR Preprocessor1… recipe pois… rsq standard 0.143 5 0.0175  
 5 m2\_LR Preprocessor1… recipe line… rmse standard 17.2 5 0.220   
 6 m2\_LR Preprocessor1… recipe line… rsq standard 0.198 5 0.0174  
 7 m2\_PR Preprocessor1… recipe pois… rmse standard 17.2 5 0.195   
 8 m2\_PR Preprocessor1… recipe pois… rsq standard 0.202 5 0.0198  
 9 m3\_LR Preprocessor1… recipe line… rmse standard 16.9 5 0.218   
10 m3\_LR Preprocessor1… recipe line… rsq standard 0.223 5 0.0130  
11 m3\_PR Preprocessor1… recipe pois… rmse standard 16.8 5 0.164   
12 m3\_PR Preprocessor1… recipe pois… rsq standard 0.239 5 0.0147

collect\_metrics shows us cross-validated performance of our 3 x 2 model combinations: 3 regression equations and 2 model families (classic regression versus poisson regression). The default error metrics are ok here. We believe that in this setting, large prediction errors are more severe in terms of cost they produce. Thus, a linear regression is fine and the RMSE a measure in alignment with this logic.

A few observations: - The errors RMSE of 17.8 to 16.9 hours seems quite big. Being of by 16 hours on average is basically being of by more than two working days on average. It looks like adding the predictors, especially how long ago an item was purchased seem to help predictions. - Poisson regressions seem to be working slightly better. This could be because hours to repair is basically count data, which is what poisson regressions are designed for.

rank\_results(wf\_rs\_fits, rank\_metric ="rmse", select\_best = TRUE)

# A tibble: 12 × 9  
 wflow\_id .config .metric mean std\_err n preprocessor model rank  
 <chr> <chr> <chr> <dbl> <dbl> <int> <chr> <chr> <int>  
 1 m3\_PR Preprocessor1… rmse 16.8 0.164 5 recipe pois… 1  
 2 m3\_PR Preprocessor1… rsq 0.239 0.0147 5 recipe pois… 1  
 3 m3\_LR Preprocessor1… rmse 16.9 0.218 5 recipe line… 2  
 4 m3\_LR Preprocessor1… rsq 0.223 0.0130 5 recipe line… 2  
 5 m2\_PR Preprocessor1… rmse 17.2 0.195 5 recipe pois… 3  
 6 m2\_PR Preprocessor1… rsq 0.202 0.0198 5 recipe pois… 3  
 7 m2\_LR Preprocessor1… rmse 17.2 0.220 5 recipe line… 4  
 8 m2\_LR Preprocessor1… rsq 0.198 0.0174 5 recipe line… 4  
 9 m1\_PR Preprocessor1… rmse 17.8 0.135 5 recipe pois… 5  
10 m1\_PR Preprocessor1… rsq 0.143 0.0175 5 recipe pois… 5  
11 m1\_LR Preprocessor1… rmse 17.8 0.180 5 recipe line… 6  
12 m1\_LR Preprocessor1… rsq 0.137 0.0137 5 recipe line… 6

### 5.6.4 Fitting the chosen model

We choose the best combination via its workflow id wflow\_id, fit it to the whole training data this time, and use it to predict the time to repair on the 20% test sample we have not used yet.

chosen\_wf <-   
 set1 |>   
 extract\_workflow("m3\_PR")  
  
chosen\_wf\_fit <-   
 chosen\_wf |>   
 fit(data = repair\_other)  
  
chosen\_wf\_pred <-   
 chosen\_wf\_fit |>   
 predict(new\_data = repair\_test) |>   
 bind\_cols(repair\_test)  
  
head(chosen\_wf\_pred)

# A tibble: 6 × 8  
 .pred months\_since\_purchase month has\_ext\_warranty item\_type employee\_id  
 <dbl> <dbl> <int> <dbl> <chr> <chr>   
1 6.00 5 1 0 mobile\_mp 4   
2 8.32 18 12 0 laptop\_budget 9   
3 5.11 4 2 0 laptop\_budget 23   
4 25.1 22 12 0 mobile\_hp 56   
5 11.8 16 5 0 laptop\_gaming 44   
6 28.2 19 10 0 laptop\_premium 37   
# ℹ 2 more variables: purchase\_price <dbl>, time\_to\_repair\_hrs <dbl>

Finally, we can see what the RMSE is on the held-out test data

rmse(chosen\_wf\_pred, truth = time\_to\_repair\_hrs, estimate = .pred)

# A tibble: 1 × 3  
 .metric .estimator .estimate  
 <chr> <chr> <dbl>  
1 rmse standard 15.8

mae(chosen\_wf\_pred, truth = time\_to\_repair\_hrs, estimate = .pred)

# A tibble: 1 × 3  
 .metric .estimator .estimate  
 <chr> <chr> <dbl>  
1 mae standard 10.2

chosen\_wf\_pred |>   
 ggplot(aes(y = .pred, x = time\_to\_repair\_hrs, color = item\_type)) +   
 scale\_y\_continuous(limits = range(chosen\_wf\_pred$time\_to\_repair\_hrs)) +   
 scale\_x\_continuous(limits = range(chosen\_wf\_pred$time\_to\_repair\_hrs)) +   
 geom\_abline(intercept = 0, slope = 1, color = "cornflowerblue") +   
 geom\_point(alpha = 0.9, shape = 21) +  
 guides(color = guide\_legend(nrow=3, byrow=TRUE)) +  
 coord\_fixed(ratio = 1) +   
 labs(y = "predicted time\_to\_repair\_hrs", color = NULL)

|  |
| --- |
| Figure 5.5: Comparing actual outcomes to predicted outcomes (1) |

RMSE, MAE, and the plot all show us that this is a bad model. RMSE of 16 hours and MAE of 10 hours is probably too much (we will later do a cost analysis). The graph also tells us that we a doing a bad job at getting the really long jobs correct (A perfect prediction model would have all points lined up on the blue line). And we also see that our dummy variables does not capture the variance in the data well.

### 5.6.5 A better model with a problem

Let us try another model, which as its main new input is employee\_id. We have the suspicion that time\_to\_repair might also depend a lot on how able the employee is. We also show a bunch of other steps like taking the square root of the price, just to show you that things like this are also possible.

set.seed(123)  
repair\_dta2 <- repair\_dta |>   
 add\_count(month, name = "Busy")  
  
splits2 <- initial\_split(repair\_dta2, prop = 0.8)  
repair\_test2 <- testing(splits2)  
repair\_other2 <- training(splits2)  
  
folds2 <- vfold\_cv(repair\_other2, v = 5)  
  
rec2 <-   
 recipe(# define the outcome and denote all other variables as predictors  
 time\_to\_repair\_hrs ~ months\_since\_purchase + purchase\_price + item\_type + Busy + employee\_id,   
 data = repair\_other2  
 ) |>   
 step\_normalize(months\_since\_purchase, Busy) |>   
 step\_dummy(item\_type, employee\_id) |>   
 step\_sqrt(purchase\_price)  
  
wf2 <-   
 workflow() |>   
 add\_recipe(rec2) |>   
 add\_model(pois\_model)  
  
wf2\_fit <-   
 wf2 |>   
 fit(data = repair\_other2)  
  
wf2\_pred <-   
 wf2\_fit |>   
 predict(new\_data = repair\_test2) |>   
 bind\_cols(repair\_test2)

rmse(wf2\_pred, truth = time\_to\_repair\_hrs, estimate = .pred)

# A tibble: 1 × 3  
 .metric .estimator .estimate  
 <chr> <chr> <dbl>  
1 rmse standard 7.14

mae(wf2\_pred, truth = time\_to\_repair\_hrs, estimate = .pred)

# A tibble: 1 × 3  
 .metric .estimator .estimate  
 <chr> <chr> <dbl>  
1 mae standard 4.21

wf2\_pred |>   
 ggplot(aes(y = .pred, x = time\_to\_repair\_hrs, color = item\_type)) +   
 scale\_y\_continuous(limits = range(chosen\_wf\_pred$time\_to\_repair\_hrs)) +   
 scale\_x\_continuous(limits = range(chosen\_wf\_pred$time\_to\_repair\_hrs)) +   
 geom\_abline(intercept = 0, slope = 1, color = "cornflowerblue") +   
 geom\_point(alpha = 0.9, shape = 21) +  
 guides(color = guide\_legend(nrow=3, byrow=TRUE)) +   
 coord\_fixed(ratio = 1) +   
 labs(y = "predicted time\_to\_repair\_hrs", color = NULL)

Warning: Removed 2 rows containing missing values or values outside the scale range  
(`geom\_point()`).

|  |
| --- |
| Figure 5.6: Comparing actual outcomes to predicted outcomes (2) |

That is a drastically better model. We still don’t get the large times fully right, maybe we need a model that accounts for a bigger tails than a poisson regression allows. But we are down to a MAE of 4 hours. And the plot looks much much cleaner.

### 5.6.6 Making a decision

This is a problem for us. Why? Because it is mainly employee id that gave us this increase in predictive power. We cannot use employee effects to predict and estimate the staffing costs! We do not know the new employees that we will hire to staff the repair counters. Maybe we could collect new data to analyse what it is about the employees that drive time efficiency. But we would still then need to either assume we can hire employees with the right traits or predict what traits new employers would come with. This seems too involved and brittle for our staffing cost problem. We build this problem specifically this way to illustrate the issue that sometimes, you can figure out who to make better predictions, but you cannot use that information for various reasons. Often there is variation that you just cannot predict, even if you knew how. And we still need to make a decision on whether to use one of the models to inform our staffing costs. And if so, how to proceed.

Based on our model we could use estimates of the typical type of cadence of different types of items, busyness, etc. to predict the average amount of repair time in a store. From this we could deduce how many FTEs we need. We also know, however, that we have a fairly big error rate (ca. 10 hours) for each. These errors do not simply add up though. In fact, if our tendency to underestimate or overestimate time is random, they might cancel out to a large extent (we already saw though, we have a tendency to underestimate the big time killers). To examine the aggregate error, we can run a prediction simulation to see what the error for the sum of expected time\_to\_repair is.

We will fit a final model without employee effects:

rec3 <-   
 recipe(# define the outcome and denote all other variables as predictors  
 time\_to\_repair\_hrs ~ months\_since\_purchase + purchase\_price + item\_type + Busy,   
 data = repair\_other2  
 ) |>   
 step\_normalize(months\_since\_purchase, Busy) |>   
 step\_dummy(item\_type) |>   
 step\_sqrt(purchase\_price)  
  
wf3 <-   
 workflow() |>   
 add\_recipe(rec3) |>   
 add\_model(pois\_model)  
  
wf3\_fit <-   
 wf3 |>   
 fit(data = repair\_other2)  
  
wf3\_pred <-   
 wf3\_fit |>   
 predict(new\_data = repair\_test2) |>   
 bind\_cols(repair\_test2)

And now, we are going to create a few *bootstrap* samples from the 20% test sample predictions. Each time we will sum up the predictions and the true time\_to\_repair values. We then look at the variation the man absolute error across all bootstrapped samples.

set.seed(47)  
  
gen\_sum\_descs <- function(split) {  
 split |>   
 as.data.frame() |>   
 summarize(  
 sum\_pred\_hrs = sum(.pred),  
 sum\_true\_hrs = sum(time\_to\_repair\_hrs),  
 abs\_diff\_sum\_hrs = abs(sum\_pred\_hrs - sum\_true\_hrs),  
 pct\_abs\_diff = round(abs\_diff\_sum\_hrs / sum\_true\_hrs, 3)  
 )  
}  
  
boot\_res <-   
 as.data.frame(wf3\_pred) |>   
 bootstraps(times = 1000) |>   
 mutate(res = map(splits, gen\_sum\_descs)) |>   
 select(-splits) |>   
 unnest(res)

summary(boot\_res)

id sum\_pred\_hrs sum\_true\_hrs abs\_diff\_sum\_hrs   
 Length:1000 Min. :10857 Min. : 9562 Min. : 0.4015   
 Class :character 1st Qu.:11370 1st Qu.:10841 1st Qu.: 190.4397   
 Mode :character Median :11552 Median :11184 Median : 441.2819   
 Mean :11557 Mean :11180 Mean : 480.5079   
 3rd Qu.:11729 3rd Qu.:11528 3rd Qu.: 706.5152   
 Max. :12347 Max. :12870 Max. :1580.1642   
 pct\_abs\_diff   
 Min. :0.000   
 1st Qu.:0.017   
 Median :0.039   
 Mean :0.044   
 3rd Qu.:0.065   
 Max. :0.165

In absolute, terms, we are actually not far off and often too high.

Visualizing the results, we get [Figure 5.7](#fig-sumerr):

|  |
| --- |
| Figure 5.7: Absolute error of predicting the sum of time to repair as percent of true sum of time to repair |

We see that there is quite a bit of variation, but if we want to be conservative (we seem to mostly overestimate the sum of hours needed anyway), then something like an 8% markup might make sense. What you actually do here will mostly depend on your other decision criteria (remember, [Chapter 2](#sec-decmb)). But you have all the information now to make an informed decision.

## 5.7 Advanced stuff: shrinkage

It is quite easy to show that if a) you care about squared losses, b) you want to fit linear models, and c) you want the *best unbiased* linear predictor, then math tells you to use a classic linear regression model. But that does not mean it will be the best predictor, just the best predictor that is unbiased. You might get even better predictions by allowing a bit of bias because a slightly biased predictor might have a much lower sensitivity to noise (lower variance). This is precisely the motivation for penalized regression models such as lasso regressions.

**To be completed.**

# 6. Reporting Design

## 6.1 Learning goals

1. Understand the importance of dashboards for reporting user-oriented results
2. Apply dashboard design principles
3. Evaluate the design of existing dashboards

## 6.2 Data visualization to speed up decision-making

The importance of data visualization in data-driven decision-making cannot be overstated. As organizations continue to generate and accumulate massive amounts of data, the ability to effectively visualize and communicate this information becomes critical in making informed decisions. Data visualization serves as a **bridge between raw data and human understanding**, making complex datasets more accessible, understandable, and usable for decision-makers.

As already discussed in [Chapter 3](#sec-datapat), data visualization helps identify trends, patterns, and outliers that may not be apparent in raw data alone. By using charts, graphs, and other visual elements, data visualization can reveal hidden relationships and connections between data points, and support exploratory data analyses. This newfound understanding can lead to the discovery of new opportunities, the identification of potential risks, and the ability to make better predictions about future events.



Nature and purpose of visualization.

Now we have moved forward and are at the declarative stage. At this stage, our aim to facilitate faster decision-making. By presenting data in a visual format, decision-makers can quickly grasp the key insights and trends in the data. This rapid comprehension enables them to make informed decisions with greater confidence and speed, as opposed to spending valuable time sifting through spreadsheets or dense reports.

What enables this **speediness** when it comes to visualizations? The human brain’s ability to quickly process visualizations plays a crucial role in comprehending complex information with ease. This rapid processing capability can be traced back to the evolutionary development of our brain, which has been wired to prioritize visual processing in response to the environmental demands faced by our ancestors. Our ancestors relied heavily on their ability to perceive and interpret visual cues from the environment for survival, such as identifying predators, locating food, and navigating the terrain. Consequently, the human brain evolved to prioritize visual processing, resulting in a highly efficient and effective system for interpreting visual information.

Furthermore, the human brain is particularly adept at recognizing patterns, which is a key aspect of data-driven decision making. Visualizations facilitate pattern recognition by presenting data in a way that highlights trends, correlations, and anomalies. This enables decision-makers to identify insights and make data-driven decisions more effectively and efficiently than if they were to rely solely on textual or numerical data. The solution is not one or the other, but using text and numbers as complements to visuals.

## 6.3 Interactive dashboards for business analysts

Interactive dashboards are versatile and dynamic data visualization tools that provide users with the ability to engage with, manipulate, and explore data through various user interface elements and features. As the name suggests, these dashboards are called “interactive” because they allow users to actively engage with the data, rather than passively consuming static visualizations. The interactivity enables users to dive deeper into the data, customize views, and uncover insights that may not be immediately apparent in a static representation.

These dashboards are the most common form of visualizations for business analysts and controllers, who are responsible for monitoring and managing an organization’s (financial) performance, operational efficiency, and overall business health. Controllers ensure accurate financial and non-financial reporting, analyze performance metrics, identify potential risks, and provide actionable insights to support decision-making across the organization. Therefore, the consumer of the dashboard is the analysts/controller and/or other stakeholders who require insights from the underlying data.



Example dashboard.

Source: https://www.thesmallman.com/dashboards

One of the primary advantages of interactive dashboards is their ability to provide **real-time** or near-real-time data updates (if well-integrated into the company’s information systems). This feature allows users to monitor financial and operational performance as it occurs, enabling them to identify and address potential issues more quickly and make timely decisions based on the most current information available. The rapid access to up-to-date data ensures that users can stay ahead of any emerging trends or anomalies, allowing them to proactively manage the organization’s health.

Efficient **monitoring** of KPIs is another crucial aspect of interactive dashboards. Users often need to track and analyze multiple KPIs to assess the organization’s financial performance and operational efficiency. Interactive dashboards provide a centralized platform for monitoring these KPIs, making it easier for users to spot trends, identify areas for improvement, and evaluate the impact of strategic initiatives on the organization’s performance.

A range of **interactive elements** is another significant benefit of such dashboards. Features such as drill-down capabilities, tooltips, and filtering enable controllers and other users to delve deeper into the data where considered necessary. This deeper level of analysis leads to a better understanding of the organization’s health.

Interactive dashboards offer **customization and flexibility**, empowering users to tailor the dashboard to their unique needs and preferences. With the ability to easily customize views, filters, and parameters, users can focus on the specific data and metrics most relevant to their role and responsibilities. This flexibility makes it easier for users to track and analyze the performance of various financial and operational aspects of the organization.

Interactive dashboards also foster improved **collaboration and communication** among stakeholders within the organization. By presenting data in a visually appealing and easily digestible format, controllers can share their findings and insights more effectively with other stakeholders, such as senior management, investors, and department heads. This improved communication encourages a shared understanding of the data, allowing everyone involved to participate in the decision-making process and contribute their unique perspectives and insights, driving more informed and data-driven decision-making across the organization.

For these benefits to materialize, we should understand some important design principles when creating such a dashboard.

## 6.4 Principles of effective dashboard design

### 6.4.1 Identify your audience

To create a truly effective dashboard, it is essential to understand your target audience (again, don’t forget, you yourself might be the target audience for your own daily work!). The user is always the most important factor in dashboard design. To cater to their needs, it is essential to develop an intimate knowledge of your audience. Identifying the relevant information to display on a dashboard can be challenging, but understanding users’ expectations, current status, and any urgent information they may need can streamline the process.

Creating so-called **personas** can help. A persona is a representation of a significant portion of your users, often incorporating the characteristics of thousands of individuals. Personas help designers understand the mindset of the people who will use their products and services, or in our case, consume the dashboard. Start by conducting both qualitative and quantitative user research to gather data about your users. Use various research methods like surveys, web analytics, user interviews, focus groups, or contextual interviews. Analyze the collected data to identify patterns and segment users into distinct groups. Once you have a better understanding of your user groups, decide on a number of personas, ideally between one and four, and categorize them into primary and secondary personas. For each persona, you can give a name, image, demographics, psychographics, and a summarizing quote. Make sure these details are based on your research and relate to your dashboard. These personas will help you empathize with your users and design a better dashboard that caters to their needs and motivations.

### 6.4.2 Prioritize your goals

The second principle emphasizes prioritizing your goals. Once you understand your user personas, you should be able to answer the key question: “What will my users expect from this dashboard?” A great rule of thumb for data disclosure in dashboards is to always begin with a high-level overview and offer easy paths for users to increase the level of granularity. Dashboards should save users time and help them become more efficient by only showing relevant information and simplifying how it’s presented.

For instance, if a text-heavy document is difficult to read and understand, most people might skim the content or not read it at all. The same applies to analytics dashboards; if there’s too much content or an overly complex design, users might not bother using the analytics. It’s essential to remember that less is more, and once you’ve prioritized your goals and identified key takeaways, consider the **five key takeaways** users want to see in the dashboard.

Selecting the right KPIs is also part of the process. Depending on your audience, different KPIs are important. Here is a great overview of typical KPIs in various functions: https://www.datapine.com/kpi-examples-and-templates/

A rule of thumb states to not include more than **8-10 KPIs in one dashboard**, otherwise the main message becomes blurry and users do not understand what to concentrate on. Of course, your possibilities to choose KPIs depend to your data and data availability.

### 6.4.3 Data sources and integration

A crucial step in creating an interactive dashboard is identifying and connecting to relevant data sources. These sources can include databases, spreadsheets, or APIs that provide access to data from various systems or platforms. Handling various data formats and structures may require integrating multiple sources and ensuring compatibility between them. This process may involve using connectors, APIs, or custom scripts to establish connections and retrieve data from the desired sources. This may require the help of data warehouse specialists in your company, certainly if you want to go real-time.

Once data sources have been identified and connected, the data must be prepared and transformed for optimal visualization. This process can involve cleaning, aggregating, and transforming data to ensure it is presented in a format that is easy to understand and interpret. In the best case, much of this has been done before you started the data analysis, but visual aspects will often require you to adapt the data further. Ensuring data quality and consistency is also crucial, as inaccurate or inconsistent data can lead to misleading visualizations and unreliable insights.

### 6.4.4 Tell a story

In essence, dashboard storytelling is the art of visually presenting data to convey the entire narrative of the data analysis process, enabling users to better understand business strategies and objectives. Effective storytelling ensures that your message is communicated as clearly as possible.

This step is crucial because a well-crafted data story bridges the gap between more technical users and those less familiar with analytics. Your dashboard should present a comprehensive story to the user. It should integrate all relevant metrics to create a cohesive story that offers a snapshot of the current situation, prioritizing information based on user needs.

To achieve efficient data storytelling, it’s helpful to plan your dashboard design in advance. Determine which charts to include based on your target audience and objectives, which will help you maintain focus when building your dashboard. This approach prevents you from combining a random assortment of visualizations and hoping they make sense together. Instead, you create valuable reports that consider users’ levels of understanding and the ultimate goals.

### 6.4.5 Selecting the right charts

When visualizing your data, it is essential to use the right charts to effectively present the information. Here is a guide in which situation which chart may be mostly suitable:



Guidance chart selection.

Source: https://biuwer.com/en/blog/how-to-choose-the-right-chart-for-your-data/

**Comparison** focuses on contrasting various values or attributes within the data. The choice of visualization depends on factors such as the presence of a time variable, the number of time periods, and the quantity of variables and categories in the data. **Composition** deals with understanding the overall characteristics of the data set. The choice of visualization depends on factors like changes over time, the number of time periods, and whether static data contains cumulative values. **Distribution** aims to examine how individual data points are spread throughout the data set. The choice of visualization, such as bar charts, line charts, or scatter charts, depends on the number of variables being analyzed in the distribution. **Relationship** explores the connections between values and attributes. Scatter charts are commonly used for examining relationships between two variables, while bubble charts are utilized when three variables are involved.

In most cases, particularly at the stage of declaring knowledge, comparison and composition will serve you well. Particularly comparisons are necessary to **provide context**, otherwise it is difficult to determine if the figures presented are positive or negative, or if they represent typical or exceptional situations. That is, numbers on a dashboard may become meaningless to users without comparative values. More importantly, users are not able to discern whether any action needs to be taken. This website contains a comprehensive overview and explanation of charts and when to use which type: https://www.datapine.com/blog/different-types-of-graphs-charts-examples/

### 6.4.6 Information architecture

Consider the principles of information architecture when deciding which charts to display and their positions. Utilizing information architecture helps limit the amount of content and data you present, allowing you to organize content for improved usability and findability. Begin with the most critical takeaways and let your dashboard flow from there. While all elements are important, some are more vital than others, so limiting the content and data presented is key to creating a visually effective dashboard. Designing a dashboard will be a trade-off between aesthetic appeal, information richness, and functionality. Consider the following aspects:

#### 6.4.6.1 Establishing a visual and hierarchy

Hierarchy pertains to the organization of information in a coherent and visually appealing manner relative to other elements. An effective hierarchy enables users to effortlessly identify crucial information and delve deeper into details as required.

A **logical hierarchy** involves the arrangement and organization of information according to its relevance or importance. It is based on the relationships and connections between various pieces of information, often resembling a tree-like structure. For instance, a dashboard might present a top-level overview displaying key metrics and trends, with the option to investigate more specific details for each metric.

A **visual hierarchy** refers to the positioning of elements on a dashboard, guiding the user’s focus towards the most significant elements on the screen. It is determined by factors such as size, color, placement, and other visual attributes of the elements.

#### 6.4.6.2 Use of size, color, and position

**Size**: Varying the size of elements within a visualization is an effective way to guide users’ attention and establish visual hierarchy. Larger or bolder elements naturally draw attention and can be used to emphasize critical information, while smaller or subtler elements can be used to convey secondary or supporting information. By manipulating size, designers can ensure that the most important insights are easily noticed and understood by users.

**Color**: The strategic use of color is another powerful tool for guiding users’ attention and creating a visual hierarchy. Colors can be used to differentiate between data categories, highlight important elements, or indicate relationships within the data. Particularly if you want to highlight certain aspects, then use one single color, and skip using colors or categories; too many colors do not create attention. High-contrast colors or bright hues can be used to emphasize critical information, while more muted or similar colors can be used for secondary or less important data points. Opt for two or three colors and experiment with gradients. A common pitfall is the overuse of highly saturated colors. While vivid colors can effectively draw users’ attention to specific data points, a dashboard saturated with too many different colors can leave users feeling overwhelmed and disoriented, unsure of where to focus their attention. It is generally advisable to use more muted colors for the majority of the dashboard. And do not forget that many people associate red with “bad” and green with “good”.

**Position**: The position of elements within a visualization can also impact users’ attention and help establish visual hierarchy. The human eye typically scans visual content from left to right and top to bottom, so placing important information in these areas can help ensure that it is noticed and understood more quickly. When structuring your dashboard, you may apply the **F and Z reading patterns**, as these create visual hierarchy in designs. Research using eye-tracking experiments has shown that those patterns are mostly followed when looking for information. The F pattern is an eye movement pattern commonly observed in users when they scan content, particularly when reading textual information. The F pattern involves users scanning the content horizontally across the top and then scanning horizontally again, but slightly shorter. Finally, users scan the content on the left side in a vertical movement.



F-patterns in eye movement.

Source: https://www.bluegranite.com/blog/design-principles-dashboard-layout-is-crucial

The Z pattern is based on the natural reading habits of individuals in cultures that read from left to right. When users view a screen or a visualization, their gaze tends to move in a pattern resembling the letter “Z”. The Z pattern is better to use when simplicity is a priority and there’s a main call-to-action takeaway.



Z-patterns in eye movement.

Source: https://www.tableau.com/blog/how-design-thinking-will-affect-todays-analysts-93507

Additionally, elements that are grouped or positioned close together are often perceived as related or interconnected, so designers can use positioning to emphasize relationships or groupings within the data.

For a successful and user-friendly design, visual and logical hierarchies must collaborate seamlessly. If these two hierarchies conflict with each other, the dashboard can become messy and challenging to comprehend because the information’s organization (logical hierarchy) does not correspond with the user’s attention towards various elements on the screen (visual hierarchy).

#### 6.4.6.3 Grouping related information

Grouping involves organizing related information cohesively on a dashboard, enabling users to efficiently and effortlessly compare and derive insights from various metrics and trends presented collectively.

For instance, a financial dashboard may have separate sections displaying all revenue metrics, expense metrics, and profit metrics together. This organization enables users to obtain comprehensive insights about each specific area.

Another prevalent method of grouping on a dashboard is categorizing different metrics or trends. For example, a customer dashboard may have a set of metrics repeated for each customer segment, such as first-time customers, active customers, or high-value customers. This arrangement allows users to seamlessly compare metrics among customer segments and identify emerging trends and patterns.

**White space**, or the empty space between elements, can help create a sense of order and separation. Be mindful of how you use white space to delineate different sections or groupings of charts within the dashboard.

#### 6.4.6.4 Maintain consistency in style

Consistency in design elements such as fonts, colors and color schemes, typography, icons, and chart types helps create a cohesive and professional look. It also makes it easier for users to understand the dashboard, as they can recognize patterns and relationships more quickly.

#### 6.4.6.5 Labeling

Clear, concise, and consistent labeling is essential for maintaining a coherent and easily understandable information architecture. Labels should accurately represent the data they describe and be consistent across individual charts and dashboards. This consistency helps users quickly grasp the meaning of each metric or trend and reduces the cognitive load required to interpret the data.



Inferior dashboard.

Source: https://www.slideteam.net/inventory-and-logistics-order-and-delivery-on-time-dashboards.html

This example probably violates multiple of the aforementioned principles:

* In terms of positioning, it appears that we need to follow a column order, making us read top-down several times…irritating.
* And while doing so, the grouping in terms of color groups does not seem always clear. For example, why are the two “Delivery On Time” charts not next to each other? If you decide to use colors for grouping purposes, it is most natural to also group the elements together.
* Apart from that, the chart on the left bottom does not seem to add any information value which makes is unnecessary in this dashboard.
* In addition, there is no clear visual hierarchy. But it may be that all items are equally important?
* Speaking about color, the dashboard uses many highly saturated colors, again making it difficult to focus.
* We are missing some benchmarks: e.g., is 96,5% on time delivery good or bad?
* Some charts can also not be understood stand-alone. For example, the “Overdue” chart only makes sense due to its relation to the chart above. Relatedly, many labels are not informative, particularly the ones called “View Details”. These issues increase cognitive load if we have to make efforts to understand the content and relation between the charts, that is, the opposite of what we want to achieve with a dashboard.

### 6.4.7 Navigation

If you prepare multiple dashboards, users should be able to move seamlessly between different dashboards or different sections within a single dashboard. This includes global navigation components such as menus, which provide a consistent way to access different parts of the application, and local or contextual navigation elements like tooltips, links, and drill-downs, which help users dive deeper into specific data points or explore related information. Effective navigation enhances the overall usability and efficiency of the dashboard.

### 6.4.8 Possibilities for interactivity

An effective and comprehensive dashboard should enable users to effortlessly delve into specific trends, metrics, or insights. There is a vast amount of options, so here a selection of the most common ones:

**Drilling** is a method that allows users to examine their data from various angles to glean significant insights. This allows users to access more detailed dashboard information related to a specific element, variable, or key performance indicator without cluttering the overall design. It’s crucial to note that the value of data drilling is directly connected to the quality and structure of the data, as with any analytical process or technique. For example, when performing a drilling down analysis, it is vital to have data that can be broken down into finer levels for the method to be effective. This feature is tidy, interactive, and allows users to view or hide crucial insights at their discretion, instead of sifting through a mass of digital information.

**Filters** enable users to refine the data displayed on a dashboard by selecting specific criteria, such as date ranges, categories, or geographic regions. This helps users focus on the most relevant information for their analysis. Imagine you want to display revenue data by country. Using click-to-filter, you just click on a particular country (or multiple countries) on the dashboard map. This action serves as a filter for the entire dashboard, and the data displayed now corresponds solely to the selected country. Users can effortlessly unclick to return to an unfiltered view or click on another country to view its data. With just a few simple clicks, users can interact with the data and identify information that is relevant to them.

**Tooltips and hover effects** provide additional context and information when users interact with the dashboard elements. For example, hovering over a data point on a line chart might display a tooltip with more information about that specific data point, such as exact values, percentage changes, or labels.

In general, giving users the ability to personalize their dashboard by rearranging, adding, or removing elements can enhance user experience and encourage deeper engagement with the data. Users can tailor the dashboard layout to meet their specific requirements, ensuring that the most relevant information is easily accessible.

### 6.4.9 Responsiveness and accessibility

Designing dashboards that are responsive and accessible is essential for providing a seamless user experience across different screen sizes and devices. **Responsiveness** ensures that the dashboard adapts to the user’s screen size, rearranging and resizing visualizations as needed for optimal viewing. This can be achieved through responsive design techniques or by using visualization tools and libraries that support responsive layouts.

**Accessibility** considerations, such as color contrast and screen reader compatibility, should also be taken into account to ensure that the dashboard is usable by a diverse audience, including users with visual impairments or other disabilities. This may involve using accessible color palettes, providing alternative text for images and visualizations, and ensuring that interactive elements are keyboard navigable.

## 6.5 Dashboard evaluation and iteration

A comprehensive approach to dashboard evaluation and iteration is essential to ensure that interactive dashboards remain effective, user-friendly, and relevant over time. This process includes assessing dashboard effectiveness through metrics, KPIs, user feedback, and usability testing, as well as ongoing maintenance and updates to ensure data quality and optimal performance.

### 6.5.1 Assessing dashboard effectiveness

To measure the success and effectiveness of a dashboard, it’s important to define specific success criteria and performance indicators. These metrics and KPIs should be aligned with the overall objectives of the dashboard and the decision-making processes it supports. Monitoring usage patterns, user satisfaction, and the impact on decision-making processes can provide valuable insights into the dashboard’s effectiveness and inform potential improvements. Metrics such as the number of active users, session duration, and goal completion rates can help gauge user engagement and the overall value of the dashboard.

Gathering feedback from users is a crucial aspect of dashboard evaluation. This can be achieved through various methods, such as surveys, interviews, or focus groups. By collecting user feedback, designers can identify areas where the dashboard may be lacking clarity, causing confusion, or failing to meet user needs. Usability testing, where users interact with the dashboard while performing specific tasks or scenarios, can also help identify pain points and areas for improvement. These insights can be used to refine the dashboard, making it more user-friendly, effective, and better suited to its intended purpose.

### 6.5.2 Ongoing maintenance and updates

Ensuring the accuracy and reliability of the data presented in the dashboard is crucial for maintaining its effectiveness and credibility. Continuously monitoring data sources and pipelines is necessary to identify any potential issues or discrepancies in the data. Implementing data validation and error checking procedures can help maintain data consistency and accuracy across the dashboard. Regularly reviewing and updating data sources can also ensure that the dashboard remains relevant and reflects the most recent and accurate information available.

As data volume and complexity grow, dashboard performance can become a concern. Slow load times and unresponsive interfaces can negatively impact the user experience and diminish the value of the dashboard. Monitoring dashboard performance and identifying potential bottlenecks or performance issues can help guide optimization efforts. Techniques such as data aggregation, caching, and query optimization can be employed to improve dashboard load times and responsiveness. Regular performance reviews and updates can help maintain a smooth and efficient user experience, even as the data landscape evolves.

In addition, you regularly review and update the dashboard to align with changing organizational goals and priorities. Incorporate new data sources, metrics, or visualizations as needed to maintain relevance and continue supporting data-driven decision-making. This ongoing adaptation ensures that the dashboard remains a valuable tool for users and keeps pace with the dynamic nature of the business environment.

# 7. Case 1: Decision Making

In this chapter, we have two goals. First, we want to review the decision making process we introduced in [Chapter 2](#sec-decmb).

|  |
| --- |
| Figure 7.1: A stylized rational decision process |

Second, we want to introduce basic R data manipulation steps using a simple decision problem that is to be solved with a data analysis.

# Loading additional 'packages'  
library(tidyverse)  
library(palmerpenguins)  
library(ggdensity)  
source("hhs-ggtheme.R")

## 7.1 Why R?

Because it is easy:

# a good life motto  
if (sad == TRUE) {  
 stop(sad)  
 be\_awesome()  
}  
# if you can read this, then you can R

## 7.2 R Basics

### 7.2.1 Basic R components.

Basically there are only two things: functions or objects (variables, data.frame, matrix, text, etc.).

Functions “do things”. So every time, you want to “do stuff”, you need a function. Objects are glorified value stores. A function is code that takes inputs and produces output. The simplest version is something like the add function. It takes two inputs and produces the sum as output:

1 + 3

[1] 4

x <- c(-3, 1, 35, 2, 5, 6, 23, 3, 5, 7) # create some input (here, a vector object)  
y <- mean(x) # do something with the input  
y # let's look at the output

[1] 8.4

As we said before an object is a glorified value store.The most widely used one is a vector. It stores values of the same *type*. You can easily build vectors using the c() function (for “concat”).

x\_nums <- c(1, 0.3, 200)  
x\_mixed <- c(2, "test example")  
x\_char <- c("a", "bw", "02")  
x\_nums

[1] 1.0 0.3 200.0

x\_mixed

[1] "2" "test example"

x\_char

[1] "a" "bw" "02"

typeof(x\_mixed)

[1] "character"

As you can see x\_mixed has converted all its element the type “character”. Vectors always contain the same type of elements. There are five basic types (Logical, integer, numeric, factor, and character. In addition, there are a few special ones like “date-time” types. These types are distinguished because of the difference in operations allowed on them. For example, Multiplying to integer numbers is a common operation. But what does ‘multiplying two sentences’ mean? Type distinctions thus help us (among other things) to check whether functions are executed with the right inputs.

An interesting case to highligh here are variables of type factor. Suppose you asked six people how they feel this morning and you get the responses

mood <- factor(  
 x = c("good", "good", "great", "ok", "great", "don't ask"),   
 levels = c("don't ask", "ok", "good", "great"),  
 ordered = TRUE   
 )  
mood

[1] good good great ok great don't ask  
Levels: don't ask < ok < good < great

With factors we can define what is called categorical variables. Here we did so and also told R that there is an order to the possible responses. Now, we might want to ask: What is the average mood this morning? Let is throw the mean() function at the data and see what happens.

mean(mood)

Warning in mean.default(mood): argument is not numeric or logical: returning NA

[1] NA

That obviously did not work, nor should it. Categorical variables do not have “distance” between the responses, even though they are ordered. We do not know whether the step from ‘ok’ to ‘good’ is as big as from ‘good’ to ‘great’. We could have also used integers to express the different responses and then computed a mean:

mood\_in\_numbers <- as.integer(mood)  
mood\_in\_numbers

[1] 3 3 4 2 4 1

mean(mood\_in\_numbers)

[1] 2.833333

Note, however that this is dangerous. Now we put an additional assumption in here. Namely that the steps between each category are equally long. Only then can we compute an average number. If we do not want to assume that, we should leave the variable as factor and then we are safe from accidentally doing operations that are not sensible for a categorical variable.

Often you want to store data of different types in one storage object (like a table in Excel). The go to object for that in R is a data.frame (or the tibble–for “tiny table” I think?–, which is a data.frame with less optionality. It is used in many tidyerse packages)

df\_example <- data.frame(SubjID = c(1,2,3),  
 Gender = factor(c("male", "female", "female")),  
 Treat = c(FALSE, TRUE, FALSE),  
 Hospital = c("Idaho", "Oklahoma", "Los Angeles")  
 )  
df\_example

SubjID Gender Treat Hospital  
1 1 male FALSE Idaho  
2 2 female TRUE Oklahoma  
3 3 female FALSE Los Angeles

data.frames are opinionated. As you can see, it turned the hospital column into a factor automatically. That is partly why some people prefer tibbles which don’t do that.

### 7.2.2 Working with tables and the grammar of data manipulation

R—and especially newer packages included in the tidyverse package ecosystem—have very expressive data verbs that make code readable. What people have realized is that most data manipulation steps that concerns data tables are really a combination of a few basic actions. The most common are listed below with their names as used in R’s [dplyr](https://dplyr.tidyverse.org/) package:

|  |
| --- |
| Figure 7.2: The basic grammar of data manipulation. Image [source](http://perso.ens-lyon.fr/lise.vaudor/dplyr/) |

* select() picks columns of a table based on their names.
* filter() picks rows of a table based on their values.
* arrange() changes the ordering of the rows.
* mutate() adds new columns that are functions of existing columns
* summarise() reduces/aggregates multiple rows down to a single summary.

If you add join and grouping actions to this list, then 98% of everything you want to do is a combination of the above actions.

Let’s work on actual data though. We’ll use the palmer penguins data set for the basic introduction, because penguins are awesome. This is how the data looks like:

head(penguins) # head() shows the first 6 rows of a table

# A tibble: 6 × 8  
 species island bill\_length\_mm bill\_depth\_mm flipper\_length\_mm body\_mass\_g  
 <fct> <fct> <dbl> <dbl> <int> <int>  
1 Adelie Torgersen 39.1 18.7 181 3750  
2 Adelie Torgersen 39.5 17.4 186 3800  
3 Adelie Torgersen 40.3 18 195 3250  
4 Adelie Torgersen NA NA NA NA  
5 Adelie Torgersen 36.7 19.3 193 3450  
6 Adelie Torgersen 39.3 20.6 190 3650  
# ℹ 2 more variables: sex <fct>, year <int>

str(penguins) # helpful function to give you a quick description of any R object

tibble [344 × 8] (S3: tbl\_df/tbl/data.frame)  
 $ species : Factor w/ 3 levels "Adelie","Chinstrap",..: 1 1 1 1 1 1 1 1 1 1 ...  
 $ island : Factor w/ 3 levels "Biscoe","Dream",..: 3 3 3 3 3 3 3 3 3 3 ...  
 $ bill\_length\_mm : num [1:344] 39.1 39.5 40.3 NA 36.7 39.3 38.9 39.2 34.1 42 ...  
 $ bill\_depth\_mm : num [1:344] 18.7 17.4 18 NA 19.3 20.6 17.8 19.6 18.1 20.2 ...  
 $ flipper\_length\_mm: int [1:344] 181 186 195 NA 193 190 181 195 193 190 ...  
 $ body\_mass\_g : int [1:344] 3750 3800 3250 NA 3450 3650 3625 4675 3475 4250 ...  
 $ sex : Factor w/ 2 levels "female","male": 2 1 1 NA 1 2 1 2 NA NA ...  
 $ year : int [1:344] 2007 2007 2007 2007 2007 2007 2007 2007 2007 2007 ...

Back to data verbs and combing different verbs to arrive at any data manipulation you want. For example, *counting*:

penguins |>   
 count(species, name = "n\_penguins")

# A tibble: 3 × 2  
 species n\_penguins  
 <fct> <int>  
1 Adelie 152  
2 Chinstrap 68  
3 Gentoo 124

Under the hood, the count() function does something like:

penguins |>   
 group\_by(species) |>   
 summarize(n\_penguins = n()) |>   
 ungroup()

# A tibble: 3 × 2  
 species n\_penguins  
 <fct> <int>  
1 Adelie 152  
2 Chinstrap 68  
3 Gentoo 124

# In newer versions of dplyr, this more succinct version also works  
penguins |>   
 summarize(n\_penguins = n(), .by = species)

# A tibble: 3 × 2  
 species n\_penguins  
 <fct> <int>  
1 Adelie 152  
2 Gentoo 124  
3 Chinstrap 68

Once you understand these basic steps, you can translate them into any other language. They might use different verbs, but the steps are the same. For example, in SQL, the most common database query language, you would write the above as:

# SQL code, not R  
SELECT   
 species,   
 COUNT(\*) AS n\_penguins   
 FROM penguins   
 GROUP BY species;

**A note on pipes ( |> or %>% ):**

In our code you see a lot of |> (or %>%). Both mean: “Take whatever is left of |> and put it as a first argument in what is right of |>”. We think it makes for much more readable, chunked code.

function1(function2(function3(data)))

becomes

data |>   
 function3() |>   
 function2() |>   
 function1()

which, to me, shows the order of execution and the separate steps in a more obvious fashion.

The rest of the common data manipulation steps we will illustrate by example using the following case.

## 7.3 Planning a festival

### 7.3.1 Case description

The following case uses the well-known Chinook sql [database](https://www.sqlitetutorial.net/sqlite-sample-database/). It is inspired by an exercise of an [Udacity](https://www.udacity.com/course/business-analytics-nanodegree--nd098) course, which we modified and expanded to train and illustrate the decision-making process more clearly. Here is the case:

You work for Chinook, an internet music store. Your boss has the idea to organize a promo music festival and asks you to come with a good location and line-up.

### 7.3.2 Identifying the problem and decision criteria

The first step is always to clarify and clearly identify what the goal is. As we discussed before, a series of questions is usually helpful. What is the purpose of the festival: promotion. Of what? The music store. For what purpose? For the purpose of generating more revenues? Probably. So let us assume that is the goal. Organizing a festival with the goal of helping to generate future revenues.

The next step is the think about what criteria such as festival should fulfill.

* Location should enable us to promote our store and generate revenues
* Budget? Not specified. Can we decide on the location without? Maybe, not ideal, but at least for a first best scenario, we can do without.
* Same for artists. Artist choice would probably very much depend on budget. Takeaway: all we do here is a perfect world festival with unlimited budget.

### 7.3.3 Developing a plan and alternatives

We have a goal (organizing a festival, which promotes the store and generates new revenues) and decided to not impose further restrictions for now. Location and line-up are obviously important parameters here. How should they be chosen? A few thoughts:

* To maximize promotional effect, we ideally want to put the festival somewhere where we can reach many customers. So put it where most existing customers are or none?
  + We probably want an area with a lot of existing customers (we can target them and they are more likely to come). We need to make sure the festival is not empty. Empty festivals are negative promotion! But we also want potential new customers to come and give them a favorable impression of the store.
  + So, what is a good location? Where there are lots of existing customers + many potential new customers
  + Should we focus on high revenue customers? Or would any customer do?
* What is a good line-up?
  + You probably have something like 10 hours of show potential (12:00 to 22:00). 4 \* 1 + 4\* 1.5 is 10 hours meaning 8 artist slots.
  + Do we want the most popular artists only? Probably not. We need some popular artists in order to draw a crowd. But those are expensive. And people own their songs already. If possible, we should also have promising, young artists or popular ones with new albums in there. Those will, if well-received, be bought and generate revenues.
  + What is a good mix? Very much depends on budget, which we do not know. Qualify, but for now, half and half is probably a good default.

### 7.3.4 Finding the location

Now that we have a plan, let’s look at the data and see whether we can execute on it. The following code loads the tidyverse package and a package to access an sqllite database. Then it connects to the chinook database and lists the tables inside the database.

library(tidyverse)  
library(RSQLite)  
  
conn <- DBI::dbConnect(SQLite(), "analysisgen/chinook.db")  
conn

<SQLiteConnection>  
 Path: /Users/judith/Library/CloudStorage/Dropbox/324070-M-6-3DM/DDDMbook/analysisgen/chinook.db  
 Extensions: TRUE

dbListTables(conn)

[1] "albums" "artists" "customers" "employees"   
 [5] "genres" "invoice\_items" "invoices" "media\_types"   
 [9] "playlist\_track" "playlists" "sqlite\_sequence" "sqlite\_stat1"   
[13] "tracks"

You can view the database diagram [here](https://www.sqlitetutorial.net/sqlite-sample-database/). In order to figure out where our customers are concentrated, we need to find data about which cities or regions they are from. Looking at the diagram, this data is most likely stored in the invoices table. Let’s have a look:

tbl(conn, "invoices") |> glimpse()

Rows: ??  
Columns: 9  
Database: sqlite 3.47.1 [/Users/judith/Library/CloudStorage/Dropbox/324070-M-6-3DM/DDDMbook/analysisgen/chinook.db]  
$ InvoiceId <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 1…  
$ CustomerId <int> 2, 4, 8, 14, 23, 37, 38, 40, 42, 46, 52, 2, 16, 17, …  
$ InvoiceDate <chr> "2009-01-01 00:00:00", "2009-01-02 00:00:00", "2009-…  
$ BillingAddress <chr> "Theodor-Heuss-Straße 34", "Ullevålsveien 14", "Grét…  
$ BillingCity <chr> "Stuttgart", "Oslo", "Brussels", "Edmonton", "Boston…  
$ BillingState <chr> NA, NA, NA, "AB", "MA", NA, NA, NA, NA, "Dublin", NA…  
$ BillingCountry <chr> "Germany", "Norway", "Belgium", "Canada", "USA", "Ge…  
$ BillingPostalCode <chr> "70174", "0171", "1000", "T6G 2C7", "2113", "60316",…  
$ Total <dbl> 1.98, 3.96, 5.94, 8.91, 13.86, 0.99, 1.98, 1.98, 3.9…

‘BillingCity’ looks like a variable we could use to decide on the location of our festival. Let us compute how much revenues we do by city. For that we need to group all the invoices by city and sum up the revenues (the ‘Total’ column). This is an aggregation operation. A standard data manipulation. Group then compute a summary measure. The dplyr package offers the summarise function to do just that. We need to call the collect() function in the end, because everything in code lines 2-7 is done inside the database. Line 8 then pulls the data from the database into R.

rev\_by\_city <-  
 tbl(conn, "invoices") |>  
 summarise(  
 total\_billings = sum(Total, na.rm = TRUE),  
 .by = c(BillingCity, BillingCountry)  
 ) |>  
 arrange(-total\_billings) |>  
 collect()  
  
head(rev\_by\_city)

# A tibble: 6 × 3  
 BillingCity BillingCountry total\_billings  
 <chr> <chr> <dbl>  
1 Prague Czech Republic 90.2  
2 Mountain View USA 77.2  
3 Paris France 77.2  
4 Berlin Germany 75.2  
5 London United Kingdom 75.2  
6 São Paulo Brazil 75.2

As an aside, since dplyr operates on a table connection in this specific case, it actually transforms the dplyr code into SQL code and sends that code to the SQL database. We can see that code by calling show\_query():

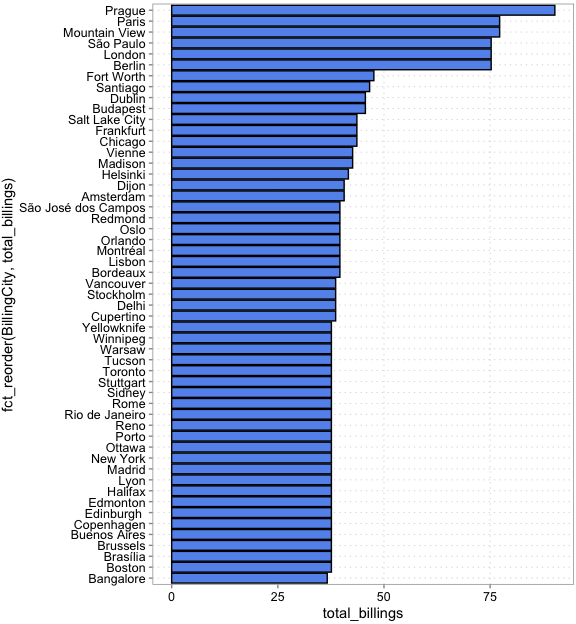
tbl(conn, "invoices") |>  
summarise(  
 total\_billings = sum(Total, na.rm = TRUE),  
 .by = BillingCity  
) |>  
arrange(-total\_billings) |>   
show\_query()

<SQL>  
SELECT `BillingCity`, SUM(`Total`) AS `total\_billings`  
FROM `invoices`  
GROUP BY `BillingCity`  
ORDER BY -`total\_billings`

Not terribly important to know, it just shows the flexibility of dplyr to work on many different data sources. Plus it shows the similarity to other programming languages.

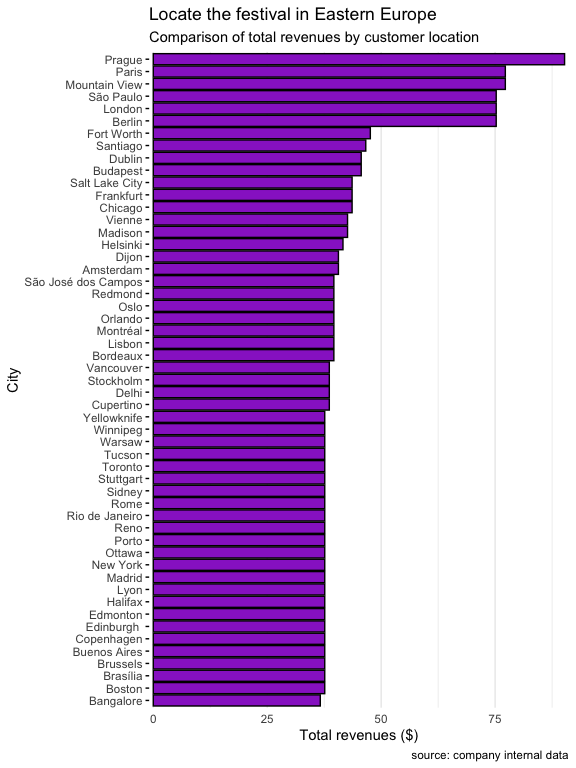
Back to the problem. Here is our result again in figure form.

rev\_by\_city |>  
 ggplot(aes(y = fct\_reorder(BillingCity, total\_billings),   
 x = total\_billings)) +  
 geom\_col()



Or if you would put this into a presentation later, add an action title and make it pretty with a few more lines:

rev\_by\_city |>  
 ggplot(aes(y = fct\_reorder(BillingCity, total\_billings),   
 x = total\_billings)) +  
 geom\_col(fill = "darkorchid")+  
 theme\_minimal() +  
 theme(panel.grid.major.y = element\_blank(),  
 axis.ticks.y = element\_line()) +  
 scale\_x\_continuous(expand = expansion(0.01)) +   
 labs(  
 title = "Locate the festival in Eastern Europe",  
 subtitle = "Comparison of total revenues by customer location",  
 caption = "source: company internal data",  
 y = "City",  
 x = "Total revenues ($)"  
 )



But let’s stop here. Is Prague really where we want to put our festival? Is it the most densely populated city for example? Given our discussion above, we probably want to take density into account to also attract as many new customers as possible. (Again, we are ignoring costs …). After a bit of googling, you find city data here at [simplemaps](https://simplemaps.com/data/world-cities):

library(readxl)  
world\_cities <- read\_excel("analysisgen/worldcities.xlsx")  
world\_cities <-   
 world\_cities |>   
 mutate(  
 country = case\_when(  
 country == "United States" ~ "USA",   
 country == "Czechia" ~ "Czech Republic",  
 .default = country)  
 )  
head(world\_cities)

# A tibble: 6 × 11  
 city city\_ascii lat lng country iso2 iso3 admin\_name capital population  
 <chr> <chr> <dbl> <dbl> <chr> <chr> <chr> <chr> <chr> <dbl>  
1 Tokyo Tokyo 35.7 140. Japan JP JPN Tōkyō primary 37732000  
2 Jaka… Jakarta -6.18 107. Indone… ID IDN Jakarta primary 33756000  
3 Delhi Delhi 28.6 77.2 India IN IND Delhi admin 32226000  
4 Guan… Guangzhou 23.1 113. China CN CHN Guangdong admin 26940000  
5 Mumb… Mumbai 19.1 72.9 India IN IND Mahārāsht… admin 24973000  
6 Mani… Manila 14.6 121. Philip… PH PHL Manila primary 24922000  
# ℹ 1 more variable: id <dbl>

We need to clean this one up a bit

Let’s merge this to our city data and compare

city\_data <-   
 rev\_by\_city |>   
 left\_join(  
 world\_cities |> select(city, country, population),  
 by = join\_by(BillingCity == city, BillingCountry == country)  
 )  
head(city\_data)

# A tibble: 6 × 4  
 BillingCity BillingCountry total\_billings population  
 <chr> <chr> <dbl> <dbl>  
1 Prague Czech Republic 90.2 1335084  
2 Mountain View USA 77.2 82409  
3 Paris France 77.2 11060000  
4 Berlin Germany 75.2 4473101  
5 London United Kingdom 75.2 11262000  
6 São Paulo Brazil 75.2 23086000

Now, how do we trade off our customer base in a location and the population there?

city\_data |>   
 arrange(-population) |>   
 head()

# A tibble: 6 × 4  
 BillingCity BillingCountry total\_billings population  
 <chr> <chr> <dbl> <dbl>  
1 Delhi India 38.6 32226000  
2 São Paulo Brazil 75.2 23086000  
3 New York USA 37.6 18972871  
4 Buenos Aires Argentina 37.6 16710000  
5 Bangalore India 36.6 15386000  
6 Rio de Janeiro Brazil 37.6 12592000

Just eye-balling, Sao Paulo seems a good candidate as it is the only one in the top 6 of both categories.

### 7.3.5 Finding the line-up

To decide on the four popular artists and the four up-and-coming bands we want in our line-up, we need to analyze our sales. Looking at the diagram, the data is dispersed across various tables. That is often the case to reduce the size of the database. What do we need now:

* We need to figure out which artists are popular in Sao Paulo
* We need to figure out which music genres are popular in Sao Paulo
* From the popular genres, we need to pick four artists

Let’s look at some of the other tables

invoice\_items <- tbl(conn, "invoice\_items")  
glimpse(invoice\_items)

Rows: ??  
Columns: 5  
Database: sqlite 3.47.1 [/Users/judith/Library/CloudStorage/Dropbox/324070-M-6-3DM/DDDMbook/analysisgen/chinook.db]  
$ InvoiceLineId <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 1…  
$ InvoiceId <int> 1, 1, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 4, 4, 4, 4, 4, 4, 4,…  
$ TrackId <int> 2, 4, 6, 8, 10, 12, 16, 20, 24, 28, 32, 36, 42, 48, 54, …  
$ UnitPrice <dbl> 0.99, 0.99, 0.99, 0.99, 0.99, 0.99, 0.99, 0.99, 0.99, 0.…  
$ Quantity <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,…

tracks <- tbl(conn, "tracks")  
glimpse(tracks)

Rows: ??  
Columns: 9  
Database: sqlite 3.47.1 [/Users/judith/Library/CloudStorage/Dropbox/324070-M-6-3DM/DDDMbook/analysisgen/chinook.db]  
$ TrackId <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17…  
$ Name <chr> "For Those About To Rock (We Salute You)", "Balls to the …  
$ AlbumId <int> 1, 2, 3, 3, 3, 1, 1, 1, 1, 1, 1, 1, 1, 1, 4, 4, 4, 4, 4, …  
$ MediaTypeId <int> 1, 2, 2, 2, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, …  
$ GenreId <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, …  
$ Composer <chr> "Angus Young, Malcolm Young, Brian Johnson", NA, "F. Balt…  
$ Milliseconds <int> 343719, 342562, 230619, 252051, 375418, 205662, 233926, 2…  
$ Bytes <int> 11170334, 5510424, 3990994, 4331779, 6290521, 6713451, 76…  
$ UnitPrice <dbl> 0.99, 0.99, 0.99, 0.99, 0.99, 0.99, 0.99, 0.99, 0.99, 0.9…

We need to merge the TrackId in invoices\_items. We can match this to the invoices table via the InvoiceId. With the TrackId we can match the data in the tracks table to the invoices, which is what we are really after. First, we only need the São Paulo invoices and then we join TrackId from the invoices\_items table to the data. The we join the tracks data using the TrackId:

bought\_tracks <-  
 tbl(conn, "invoices") |>  
 filter(BillingCity %in% c("São Paulo")) |>  
 inner\_join(invoice\_items, join\_by(InvoiceId == InvoiceId)) |>  
 select(InvoiceId, TrackId) |>  
 inner\_join(  
 tracks |> select(TrackId, AlbumId, GenreId),   
 join\_by(TrackId == TrackId))  
head(bought\_tracks)

# Source: SQL [6 x 4]  
# Database: sqlite 3.47.1 [/Users/judith/Library/CloudStorage/Dropbox/324070-M-6-3DM/DDDMbook/analysisgen/chinook.db]  
 InvoiceId TrackId AlbumId GenreId  
 <int> <int> <int> <int>  
1 25 738 57 7  
2 25 744 57 7  
3 25 750 58 1  
4 25 756 59 1  
5 25 762 60 1  
6 25 768 61 1

With the AlbumID we can get the artist from the albums data.

genres <- tbl(conn, "genres") |>  
 rename(Genre = Name)  
  
artists <- tbl(conn, "artists") |>  
 rename(Artist = Name)  
  
popular\_tracks\_data <-  
 bought\_tracks |>  
 inner\_join(tbl(conn, "albums"), join\_by(AlbumId == AlbumId)) |>  
 inner\_join(genres, join\_by(GenreId == GenreId)) |>  
 inner\_join(artists, join\_by(ArtistId == ArtistId)) |>  
 collect()  
head(popular\_tracks\_data)

# A tibble: 6 × 8  
 InvoiceId TrackId AlbumId GenreId Title ArtistId Genre Artist  
 <int> <int> <int> <int> <chr> <int> <chr> <chr>   
1 25 738 57 7 Cássia Eller - Sem Li… 77 Latin Cássi…  
2 25 744 57 7 Cássia Eller - Sem Li… 77 Latin Cássi…  
3 25 750 58 1 Come Taste The Band 58 Rock Deep …  
4 25 756 59 1 Deep Purple In Rock 58 Rock Deep …  
5 25 762 60 1 Fireball 58 Rock Deep …  
6 25 768 61 1 Knocking at Your Back… 58 Rock Deep …

popular\_artists <-  
 popular\_tracks\_data |>  
 summarise(  
 nr\_tracks\_bought = n(),  
 .by = Artist  
 ) |>  
 arrange(-nr\_tracks\_bought)  
  
# Now, what are the 4 most popular artists?  
head(popular\_artists, 4)

# A tibble: 4 × 2  
 Artist nr\_tracks\_bought  
 <chr> <int>  
1 Pearl Jam 11  
2 Deep Purple 7  
3 R.E.M. 7  
4 U2 6

Now let’s try to find the up-and-coming artists in the popular genres. First we need to figure out which these are.

popular\_genres <-  
 popular\_tracks\_data |>  
 summarise(  
 nr\_tracks\_bought = n(),  
 .by = c(Genre, GenreId)  
 ) |>  
 arrange(-nr\_tracks\_bought)  
head(popular\_genres)

# A tibble: 6 × 3  
 Genre GenreId nr\_tracks\_bought  
 <chr> <int> <int>  
1 Rock 1 40  
2 Latin 7 18  
3 Metal 3 5  
4 Alternative & Punk 4 5  
5 Reggae 8 3  
6 Hip Hop/Rap 17 2

Looks like we want to focus on Rock and Latin. Now we need to think about how we can identify new but interesting Artists. What are the characteristics of such Artists?

* Not popular yet
* Few, but recent Albums
* Should show signs of recent popularity.

Think about these characteristica for a second. Did you realize that these are essentially, filtering conditions? The first bullet point translates into: should not be in the upper part of our popular\_artists data.frame. The second means, should have few but recently released Albums. Maybe we can filter for this using InvoiceDate. The third point we could translate as have most purchases in the last six months.

Once we have the relevant data collected from all tables, we only need to apply the correct filters that describe the characteristics above.

artists\_tracks\_link <-   
 tracks |>   
 filter(GenreId %in% c(1, 7)) |>   
 select(TrackId, AlbumId) |>   
 inner\_join(tbl(conn, "albums"), join\_by(AlbumId == AlbumId)) |>  
 inner\_join(artists, join\_by(ArtistId == ArtistId)) |>   
 select(TrackId, Artist, ArtistId)  
  
head(artists\_tracks\_link)

# Source: SQL [6 x 3]  
# Database: sqlite 3.47.1 [/Users/judith/Library/CloudStorage/Dropbox/324070-M-6-3DM/DDDMbook/analysisgen/chinook.db]  
 TrackId Artist ArtistId  
 <int> <chr> <int>  
1 1 AC/DC 1  
2 2 Accept 2  
3 3 Accept 2  
4 4 Accept 2  
5 5 Accept 2  
6 6 AC/DC 1

young\_artists <-   
 tbl(conn, "albums") |>   
 count(ArtistId) |>   
 filter(n == 1) |>   
 select(-n)  
head(young\_artists)

# Source: SQL [6 x 1]  
# Database: sqlite 3.47.1 [/Users/judith/Library/CloudStorage/Dropbox/324070-M-6-3DM/DDDMbook/analysisgen/chinook.db]  
 ArtistId  
 <int>  
1 3  
2 4  
3 5  
4 7  
5 9  
6 10

young\_artist\_invoices <-   
 tbl(conn, "invoices") |>   
 select(InvoiceId, InvoiceDate) |>   
 inner\_join(invoice\_items, join\_by(InvoiceId == InvoiceId)) |>   
 inner\_join(artists\_tracks\_link, join\_by(TrackId == TrackId)) |>   
 inner\_join(young\_artists, join\_by(ArtistId == ArtistId)) |>   
 select(InvoiceId, InvoiceDate, TrackId, Artist) |>   
 collect()  
  
head(young\_artist\_invoices)

# A tibble: 6 × 4  
 InvoiceId InvoiceDate TrackId Artist   
 <int> <chr> <int> <chr>   
1 3 2009-01-03 00:00:00 24 Aerosmith  
2 109 2010-04-16 00:00:00 25 Aerosmith  
3 214 2011-07-25 00:00:00 26 Aerosmith  
4 3 2009-01-03 00:00:00 28 Aerosmith  
5 320 2012-11-06 00:00:00 30 Aerosmith  
6 109 2010-04-16 00:00:00 31 Aerosmith

young\_artist\_lineup <-   
 young\_artist\_invoices |>   
 mutate(InvoiceDate = ymd\_hms(InvoiceDate),  
 year = year(InvoiceDate)) |>   
 # filter to only look at most recent purchases  
 filter(year == max(year)) |>   
 # counting purchases  
 summarise(  
 nr\_buys = n(),  
 .by = Artist  
 ) |>   
 # make sure to take out the top 6 popular artists:  
 filter(!(Artist %in% head(popular\_artists$Artist, 6))) |>   
 # now sort and show the top 4  
 arrange(-nr\_buys) |>   
 head(4)  
young\_artist\_lineup

# A tibble: 4 × 2  
 Artist nr\_buys  
 <chr> <int>  
1 O Terço 8  
2 Men At Work 5  
3 The Who 5  
4 Soundgarden 4

### 7.3.6 Wrapping up

In summary:

* We decide to propose Sao Paulo as a location because it is the only city in the top six by number of existing customers and population to draw new customers from.
* We propose 8 artists based on time constraints
* We propose the top 4 artists based on what is popular with existing customers in the Sao Paulo:
* We propose the top 4 newcomers (artists with only one album in our db) and most track purchases during the most recent year.

We could think of other alternatives, but we stop here and evaluate the current one. It fits our criteria established above. The big issue is that we weren’t given any budget constraints. We could potentially also do a better job at identifying newcomers. Or we could also consider not only the city but define a radius of influence around a city when deciding on a location. We can still propose the current solution. We have solid arguments to justify our choices. But should keep these limitations in mind.

## 7.4 Merging and aggregating data in Excel

Everything we have done here, we can do in Excel, as long as the data we are dealing with is not too big. In this section we want to give a short intro into tables, queries, and pivot tables in excel. These tools have become quite powerful, so we showcase them here for comparison.

With the right tools, you can query SQL databases directly from excel too. To keep things simple, we put the tables in the chinook database into separate excel sheets in one file (see [Figure 7.3](#fig-excel1).)

### 7.4.1 Make use of the table feature

The first thing we want to turn the data cells in these sheets into “tables” as these makes it easier to reference them later. Doing so is simple. Below we show how to do this for the invoices data sheet. Select any cell within the data range that constitutes your table (see [Figure 7.3](#fig-excel1)), then click on “Format as Table”. Pick a format, then make sure the cell range is correct and “has headers is checked”.

|  |
| --- |
| Figure 7.3: Turn a cell range into a table |

Finally, we click on the Table Design section in the Excel ribbon and choose a name for the table (see [Figure 7.4](#fig-excel2)).

|  |
| --- |
| Figure 7.4: Labeling tables |

### 7.4.2 Pivot tables as a powerful aggregation feature

Everything we have done so far with grouping and summarizing using the summarise() function in R can be done with the help of excel pivot tables to. Let us repeat the exercise of finding the city where most of Chinook’s revenues are originating from.

Click on the “Summarize with PivotTable” button you see in [Figure 7.4](#fig-excel2). Choose to put the table into a new sheet. A new sheet will open and on the side you will see a Fields menu as shown in [Figure 7.5](#fig-excel3). We want a similar output as we got from the summarise() function in R. We want the sum of the revenues grouped by billing city. So the rows of the output table should be billing cities. To get that, drag the BillingCity field into the Rows section (see [Figure 7.5](#fig-excel3)). The values in each row should be the sum of the “Total” field. So drag the “Total” field into the Values section. It usually defaults to “Sum of ..” which is what we want in this case. If it is not click on the down arrow, then on “Value Field Settings” in that menu, you can choose what aggregation is performed (sum, count, mean, min, max, etc.)

|  |
| --- |
| Figure 7.5: Pivot table options |

We almost got what we want. We just need to sort the resulting pivot table from city with highest total revenues to lowest. For whatever reason, that sorting feature is a bit hidden. Right-click on any cell in the Sum of Total column and there you find a sort option (see [Figure 7.6](#fig-excel4)).

|  |
| --- |
| Figure 7.6: Sorting pivot tables |

This is the result:

|  |
| --- |
| Figure 7.7: Most important cities |

### 7.4.3 Joining data in Excel

Joining data in Excel used to be a very error prone and cumbersome operation. It still is not as straightforward as with a programming language. But the current capabilities make it much much easier and clearer to debug. Here is one approach we like. We will illustrate merging using the question: “What are the genres of music that are bought by customers in Sao Paulo?” To figure this out, we need to merge three tables: invoices, invoice\_items, and tracks. First we need to define a table for invoice\_items and tracks—just as we did above for Invoices. Second, we need to create a query link for each table. This is because we will use power query to do the merging. Here is how we define a query for each table.

Click on a cell for the table we want to link, then go to the Data section and click on “From Table/Range” (see [Figure 7.8](#fig-excel6))

|  |
| --- |
| Figure 7.8: Pivot table options |

This will open the power query editor. We are not going to do much there yet. Hit “close and load to”

|  |
| --- |
| Figure 7.9: Close and load to |

and then click “Only Create Connection”

|  |
| --- |
| Figure 7.10: Connections option |

Do this for all three tables we want to merge. You should then have three queries in your query sidebar:

|  |
| --- |
| Figure 7.11: Queries and Connections |

Now we are ready to do the merging. We have to do this in two steps because we want to do two merges. Click again on the Data section, then click the “Get Data” drop-down menu and choose “Combine Queries”.

|  |
| --- |
| Figure 7.12: Get data menu |

A menu will open where we can select two queries to join. We choose Invoices and InvoiceItems. Then we click on the columns in each table we want to join by (“InvoiceID”). Finally, we select “inner join” under “Join Kind”. Your selection should look like this ([Figure 7.13](#fig-excel11)):

|  |
| --- |
| Figure 7.13: Join menu |

Once you click “OK”, you will enter the power query editor again. It will show a new query called Merge1. Rename that to something like “SaoPauloTracks” by right-clicking on “Merge1” and choosing rename.

|  |
| --- |
| Figure 7.14: Name your queries so you know what they are for |

We called it SaoPaulo tracks, because we are going to filter this query a bit now too. Click on the dropdown arrow right next to the column “BillingCity” and select only Sao Paulo. Then click on the Icon right next to the column header InvoiceItems (see [Figure 7.15](#fig-excel13).)

|  |
| --- |
| Figure 7.15: Including join columns |

This contains all the columns from the joint invoice\_items table. When you click on it, it asks you what columns to keep. We only need “TrackId”. We also do not need a lot of the other columns still in this query. Click on Choose columns and get rid of all but InvoiceId, BillingCity, and TrackId. You final query should look like this and list the following steps on the right (see [Figure 7.16](#fig-excel14).)

|  |
| --- |
| Figure 7.16: A finished joint query |

Click on “close and load to” again (see [Figure 7.9](#fig-excel7)) and again choose “Only Create Connection”. This is only an intermediate join that we do not want to output into our excel worksheets. We do need still merge the third table onto this. Now we are essentially repeat the join steps again, starting from “Get Data” and “Combine Queries” ([Figure 7.12](#fig-excel10)). This time we merge the SaoPauloTracks query and the Tracks query. And we join them by TracksId and choose “inner join” again.

|  |
| --- |
| Figure 7.17: Join menu again |

We click ok and click on expand right next to the header of the “Tracks” column and select only the GenreId and AlbumID columns. We rename the query to something like SaoPauloGenres. Now we again hit Close and load to. But this time we export to a new table and a new worksheet. The result should look like this ([Figure 7.18](#fig-excel16)):

|  |
| --- |
| Figure 7.18: Final join output |

From here now we can produce produce a pivot table that produces the most popular genreIds and so on. We leave the rest of the case as an excercise.

# 8. Case 4: Predictive Analysis

library(tidymodels)  
library(poissonreg)  
library(ggdag)  
library(patchwork)  
library(collapse, include.only = c("descr"))  
source("hhs-ggtheme.R")

## 8.1 Your task: predicting renovation costs

You are working for a hotel chain, Eden resorts. Eden resorts owns 598 hotels in various big cities and tourist destinations all across Europe. You are working on the financial planing for the upcoming year. Right now, you are tasked with predicting the group-level costs for renovating rooms. When looking at the financials of recent years, there seems to be quite a bit of variation across hotels and locations, so you decide to take a closer look at what predicts renovation costs. Not only will it help you with your current planing, but this might be useful information for future expansion plans.

## 8.2 Theory as the starting point

You know that you can collect some data from the hotel chain’s systems, so you first start sketching out a mental model of what might be important predictors. You start with something like this [Figure 8.1](#fig-hoteldag):

|  |
| --- |
| Figure 8.1: DAG of possible determinants of the N of hotelroom renovations |

## 8.3 The data

It is quite usual to adjust your mental model as you go and learn from what the data tells you. The sketch above is just a rough starting point and you should adjust it, once you start thinking more about the problem. But it is good enough to get started. For now, you use [Figure 8.1](#fig-hoteldag) to guide your data collection efforts. You extract the following data from Eden’s systems:

dta <- readr::read\_csv("analysisgen/case-hotel-renovations.csv")  
dta$price\_rank <- ordered(dta$price\_rank, levels = 1:5)  
dta$hotel\_id <- factor(dta$hotel\_id)  
dta$location\_id <- factor(dta$location\_id)  
glimpse(dta)

Rows: 4,784  
Columns: 11  
$ hotel\_id <fct> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, …  
$ year <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, …  
$ location\_id <fct> 27, 51, 7, 30, 36, 1, 20, 31, 43, 16, 5, 42, 25, 43…  
$ business\_traveller <dbl> 0.4999054, 0.5015014, 0.6021005, 0.2023258, 0.60034…  
$ young\_tourists <dbl> 0.0000000000, 0.0980487616, 0.1988999322, 0.1001583…  
$ family\_guests <dbl> 0.30020121, 0.30034938, 0.09910840, 0.59785546, 0.2…  
$ other <dbl> 0.19989338, 0.10010047, 0.09989118, 0.09966038, 0.0…  
$ n\_rooms <dbl> 166, 275, 191, 339, 214, 345, 436, 185, 291, 329, 3…  
$ n\_guests <dbl> 4328, 14339, 8714, 7279, 8679, 12592, 10609, 4502, …  
$ price\_rank <ord> 3, 1, 1, 2, 3, 3, 3, 4, 3, 4, 4, 4, 2, 3, 3, 1, 1, …  
$ n\_renovations <dbl> 5, 17, 12, 23, 18, 18, 19, 4, 18, 11, 42, 25, 35, 1…

## 8.4 Exploratory analysis to get familiar with the data

A good first step is to examine the outcome variable of interest. A histogram to examine to univariate distribution and some descriptive statistics are usually enough to give you a sense of the data:

dta |>   
 ggplot(aes(x = n\_renovations)) +   
 geom\_histogram(binwidth = 5) +   
 scale\_x\_continuous(breaks = scales::breaks\_width(10)) +  
 scale\_y\_continuous(expand = expansion(c(0.0,0.05))) +  
 theme(panel.grid.minor.x = element\_blank())

|  |
| --- |
| Figure 8.2: Empirical distribution of the outcome variable |

# here we use the collapse::descr function  
descr(dta$n\_renovations)

Dataset: n\_renovations, 1 Variables, N = 4784  
--------------------------------------------------------------------------------  
n\_renovations (numeric):   
Statistics  
 N Ndist Mean SD Min Max Skew Kurt  
 4784 114 18.88 15.57 1 220 4.73 41.36  
Quantiles  
 1% 5% 10% 25% 50% 75% 90% 95% 99%  
 3 5 7 10 16 22 31 42 76  
--------------------------------------------------------------------------------

## 8.5 Preparing the data for our comparison of different prediction models

When we compare many different models, we will always run the risk of choosing an overfit model when selecting a model based on *estimates* of out-of-sample performance. Even, though we will use cross-validation (discussed in [Chapter 5](#sec-pred)), we can never really get rid of the fact that we are re-using and re-using the training data and will eventually be fooled by noise. So, we want to keep a bit of the data out of the training routine, so that we can asses the out-of-sample performance of the model we eventually select a bit better. We do this via the initial split function. Here we actually use a group\_ split function. The reason is that we have a panel. We have hotel data for multiple years. We want to make sure that we randomly select hotels and \*not’ hotel-years. By defining hotel\_id as a grouping variable we make sure that all years of a hotel are in the same split (either in training or in testing).

We could also stratify slightly by the outcome variable. What this does is it sorts n\_renovations in 3-4 buckets (think: low, medium, high) and ensures that training and test are roughly comparable in their proportions of low, medium, and high number of renovations. This is just a precaution to make sure our testing set is not randomly tilted in terms of the outcome. In our case, we cannot do that however, because we do not have enough data to do such stratification. So we will have to live without

set.seed(47)   
splits <- group\_initial\_split(dta, prop = 0.8, group = hotel\_id)  
dta\_test <- testing(splits)  
dta\_train <- training(splits)  
splits

<Training/Testing/Total>  
<3824/960/4784>

Next, we prepare our cross-validation folds. Again, because we have a smallish sample, we use five folds. If we had more, we’d probably go up to ten folds

set.seed(48)  
folds <- group\_vfold\_cv(dta\_train, v = 5, group = hotel\_id)  
print(folds)

# Group 5-fold cross-validation   
# A tibble: 5 × 2  
 splits id   
 <list> <chr>   
1 <split [3056/768]> Resample1  
2 <split [3056/768]> Resample2  
3 <split [3056/768]> Resample3  
4 <split [3064/760]> Resample4  
5 <split [3064/760]> Resample5

## 8.6 Deciding on the preprocessing and model steps

Because number of renovations is a positive integer number, classic and poisson regressions are natural starting point for a predictive analysis. We define both model classes for use later.

Next, we define our different models. These are a combination of 1) outcome variable, 2) predictors (think x-variables) to use and 3) pre-processing (transformation) steps to be applied to the outcome or predictor variables. Such pre-processing steps could be turning a factor variable into dummy variables, adding squared terms of a particular x variable as an additional predictor, changing the scale of some predictor, and so on.

Below we just tried a few combinations to illustrate how to set up different models. We start with a recipe that defines what the outcome variables are (the variables to the left of ~) and what the predictor variables are (the variables to the right of ~). Then we apply different (pre-processing) steps.

lin\_model <- linear\_reg(mode = "regression", engine = "lm")  
pois\_model <- poisson\_reg(mode = "regression", engine = "glm")  
  
# basic steps we want in all models:  
r0 <- recipe(dta\_train) |>   
 step\_normalize(all\_numeric\_predictors()) |>   
 update\_role(n\_renovations, new\_role = "outcome")  
  
# Individual versions of inputs for each model  
# Each one updates the preceding version.   
r1 <- r0 |>   
 update\_role(c(n\_rooms, n\_guests), new\_role = "predictor")  
  
r2 <- r1 |>   
 update\_role(  
 c(family\_guests, young\_tourists, business\_traveller),  
 new\_role = "predictor"  
 )  
  
r3 <- r2 |>   
 update\_role(price\_rank, new\_role = "predictor") |>   
 remove\_role(c(family\_guests, business\_traveller), old\_role = "predictor")   
  
r4 <- r3 |>   
 remove\_role(c(young\_tourists), old\_role = "predictor") |>   
 step\_dummy(price\_rank)  
   
r5 <- r4 |>   
 step\_poly(n\_guests)

Once we have defined all our models, we combine them into what tidymodels calls a workflowset. With the cross = TRUE option, we will have every combination of model specification and model class as one model in our workflow set.

model\_set1 <-   
 workflow\_set(  
 preproc = list(m1 = r1, m2 = r2, m3 = r3, m4 = r4, m5 = r5),  
 models = list(LR = lin\_model, PR = pois\_model),  
 cross = TRUE  
 )

We now take the defined workflowset and test each model in it on the cross validation folds. Afterwards we use the rank\_results function to rank the different models according to the root mean squared error metric. We chose RMSE because we are concerned about large errors. We use the verbose option in case there are model hiccups. If there is something going awry (e.g., a rank deficit matrix because we accidentally selected fully co-linear variables) then the verbose option will tell us which work flow had an issue.

model\_set1\_fits <- model\_set1 |>   
 workflow\_map("fit\_resamples",   
 # Options to `workflow\_map()`:   
 seed = 1101, verbose = TRUE,  
 # Options to `fit\_resamples()`:   
 resamples = folds, control = control\_resamples(save\_pred = TRUE)  
 )

i 1 of 10 resampling: m1\_LR

✔ 1 of 10 resampling: m1\_LR (1.4s)

i 2 of 10 resampling: m1\_PR

✔ 2 of 10 resampling: m1\_PR (1.5s)

i 3 of 10 resampling: m2\_LR

✔ 3 of 10 resampling: m2\_LR (375ms)

i 4 of 10 resampling: m2\_PR

✔ 4 of 10 resampling: m2\_PR (385ms)

i 5 of 10 resampling: m3\_LR

✔ 5 of 10 resampling: m3\_LR (331ms)

i 6 of 10 resampling: m3\_PR

✔ 6 of 10 resampling: m3\_PR (369ms)

i 7 of 10 resampling: m4\_LR

✔ 7 of 10 resampling: m4\_LR (845ms)

i 8 of 10 resampling: m4\_PR

✔ 8 of 10 resampling: m4\_PR (508ms)

i 9 of 10 resampling: m5\_LR

✔ 9 of 10 resampling: m5\_LR (639ms)

i 10 of 10 resampling: m5\_PR

✔ 10 of 10 resampling: m5\_PR (536ms)

results\_ranked <- rank\_results(model\_set1\_fits, rank\_metric ="rmse")  
results\_ranked

# A tibble: 20 × 9  
 wflow\_id .config .metric mean std\_err n preprocessor model rank  
 <chr> <chr> <chr> <dbl> <dbl> <int> <chr> <chr> <int>  
 1 m5\_LR Preprocessor1\_… rmse 5.10 0.114 5 recipe line… 1  
 2 m5\_LR Preprocessor1\_… rsq 0.892 0.0105 5 recipe line… 1  
 3 m5\_PR Preprocessor1\_… rmse 5.60 0.178 5 recipe pois… 2  
 4 m5\_PR Preprocessor1\_… rsq 0.871 0.00981 5 recipe pois… 2  
 5 m2\_LR Preprocessor1\_… rmse 6.46 0.507 5 recipe line… 3  
 6 m2\_LR Preprocessor1\_… rsq 0.837 0.00979 5 recipe line… 3  
 7 m3\_LR Preprocessor1\_… rmse 6.65 0.534 5 recipe line… 4  
 8 m3\_LR Preprocessor1\_… rsq 0.828 0.00877 5 recipe line… 4  
 9 m1\_LR Preprocessor1\_… rmse 6.66 0.531 5 recipe line… 5  
10 m1\_LR Preprocessor1\_… rsq 0.827 0.00891 5 recipe line… 5  
11 m4\_LR Preprocessor1\_… rmse 6.67 0.532 5 recipe line… 6  
12 m4\_LR Preprocessor1\_… rsq 0.827 0.00888 5 recipe line… 6  
13 m3\_PR Preprocessor1\_… rmse 9.30 2.93 5 recipe pois… 7  
14 m3\_PR Preprocessor1\_… rsq 0.824 0.0205 5 recipe pois… 7  
15 m4\_PR Preprocessor1\_… rmse 9.30 2.94 5 recipe pois… 8  
16 m4\_PR Preprocessor1\_… rsq 0.824 0.0200 5 recipe pois… 8  
17 m2\_PR Preprocessor1\_… rmse 9.34 2.84 5 recipe pois… 9  
18 m2\_PR Preprocessor1\_… rsq 0.817 0.0170 5 recipe pois… 9  
19 m1\_PR Preprocessor1\_… rmse 9.76 3.20 5 recipe pois… 10  
20 m1\_PR Preprocessor1\_… rsq 0.816 0.0222 5 recipe pois… 10

We can also look at the results in figure form:

p1 <-   
 autoplot(model\_set1\_fits, metric = "rmse") +  
 ggrepel::geom\_text\_repel(aes(label = wflow\_id),   
 nudge\_x = 1/8, nudge\_y = 1/100, size = 3.5) +  
 theme(legend.position = "none")  
  
p2 <-   
 autoplot(model\_set1\_fits, metric = "rsq") +  
 ggrepel::geom\_text\_repel(aes(label = wflow\_id),   
 nudge\_x = 1/8, nudge\_y = 1/100, size = 3.5) +  
 theme(legend.position = "none")  
  
(p1 + p2)

|  |
| --- |
| Figure 8.3: Prediction performance of different models in terms of RMSE and R-squared |

Apparently, the last model with a polynomial specification and price rank dummies does best. Let us take this as our chosen model. We now take it and fit it to the whole training data. And then we will finally assess how well it does on the test sample.

### 8.6.1 Fitting the chosen model

We choose the best combination via its workflow id wflow\_id, fit it to the whole training data this time, and use it to predict the time to repair on the 20% test sample we have not used yet. To better explore the performance of our model it often helps to add our predictions to the test sample:

best\_wf\_id <- results\_ranked$wflow\_id[1]  
chosen\_wf <- extract\_workflow(model\_set1\_fits, best\_wf\_id)  
  
# we want to refit the best model/workflow on the full training set  
chosen\_wf\_fit <- fit(chosen\_wf, data = dta\_train)  
  
# Now we can use the fitted model to predict on the test set  
chosen\_wf\_pred <-   
 predict(chosen\_wf\_fit, new\_data = dta\_test) |>   
 bind\_cols(dta\_test)  
  
head(chosen\_wf\_pred)

# A tibble: 6 × 12  
 .pred hotel\_id year location\_id business\_traveller young\_tourists  
 <dbl> <fct> <dbl> <fct> <dbl> <dbl>  
1 21.3 2 1 51 0.502 0.0980  
2 20.7 6 1 1 0.598 0.101   
3 20.4 7 1 20 0.302 0.300   
4 10.3 25 1 15 0.400 0.102   
5 11.8 30 1 57 0.600 0.201   
6 13.0 34 1 30 0.199 0.0987  
# ℹ 6 more variables: family\_guests <dbl>, other <dbl>, n\_rooms <dbl>,  
# n\_guests <dbl>, price\_rank <ord>, n\_renovations <dbl>

Finally, we can see what the RMSE is on the held-out test data

rmse(chosen\_wf\_pred, truth = n\_renovations, estimate = .pred)

# A tibble: 1 × 3  
 .metric .estimator .estimate  
 <chr> <chr> <dbl>  
1 rmse standard 5.39

mae(chosen\_wf\_pred, truth = n\_renovations, estimate = .pred)

# A tibble: 1 × 3  
 .metric .estimator .estimate  
 <chr> <chr> <dbl>  
1 mae standard 3.69

So on average, we are off by 3.6 renovations by year or 5.3 root squared renovations. Given that the mean number of rennovations in the full dataset is 18.9 and the median is 16 that MAE is ca 23% of the median. So, in terms of magnitude it is still sizable. How expensive this error is, we can only assess once we know the cost of a renovation though.

We should also always look at a plot of predicted versus actual outcomes. This looks quite decent. We do seem to struggle a bit with the outliers. Also, while a bit hard to see, we seem to not do so well at the very low end ever. We seem to systematically overshoot the hotel-years with a very low number of renovations.

chosen\_wf\_pred |>   
 ggplot(aes(y = .pred, x = n\_renovations, color = n\_rooms)) +   
 scale\_y\_continuous(limits = range(chosen\_wf\_pred$n\_renovations)) +   
 scale\_x\_continuous(limits = range(chosen\_wf\_pred$n\_renovations)) +   
 geom\_abline(intercept = 0, slope = 1, color = "coral1") +   
 geom\_point(alpha = 0.9, shape = 21) +  
 labs(y = "predicted n\_renovations") +   
 theme(legend.position = "right")+   
 coord\_fixed(ratio = 1)

|  |
| --- |
| Figure 8.4: Predicted values versus actual outcomes |

## 8.7 An alternative set of model with transformed outcomes

Sometimes we can improve predictive performance by transforming the outcome variable. This can sometimes make the functional form tying the outcome variable to predictors more easy to fit. In our case, the number of renovations is obviously heavily dependent on the number of rooms the hotel has. Maybe a better question to ask is: what proportion of hotel rooms is renovated? We can then also frame the number of guests as a proportion of rooms to get to a predictor that reflects how often the average room is used (something like room booking frequency). These transformed variables might be be closer to the underlying mechanism that determines how fast a room “wears out”. Let us do these transformations and try everything again:

dta2 <- dta |>   
 mutate(  
 prop\_renov = n\_renovations / n\_rooms,  
 room\_freq = n\_guests / n\_rooms  
 )

Let’s see how the new outcome variable is distributed

dta2 |>   
 ggplot(aes(x = prop\_renov)) +   
 geom\_histogram(binwidth = 0.025) +   
 scale\_x\_continuous(n.breaks = 10) +  
 scale\_y\_continuous(expand = expansion(c(0.01,0.05))) +   
 theme(panel.grid.major.x = element\_blank())

|  |
| --- |
| Figure 8.5: Empirical distribution of the outcome variable |

descr(dta2$prop\_renov)

Dataset: prop\_renov, 1 Variables, N = 4784  
--------------------------------------------------------------------------------  
prop\_renov (numeric):   
Statistics  
 N Ndist Mean SD Min Max Skew Kurt  
 4784 2367 0.07 0.05 0.01 0.57 4.63 33.4  
Quantiles  
 1% 5% 10% 25% 50% 75% 90% 95% 99%  
 0.02 0.03 0.04 0.05 0.06 0.08 0.11 0.15 0.25  
--------------------------------------------------------------------------------

This is still a pretty skewed distribution. The descriptives show you that we have one year where more than 50% of the rooms were renovated. So, visually it is hard to tell whether this outcome might or might not be hard to get right. Let us carry on nevertheless. It seems a bit easier to reason about it.

We do our data preparation steps again. This time we do a split that is aware of hotel\_id and we also do a cross-validation that is aware of hotel\_id. This means, that we effectively randomly choose hotels for the test and training set and for the different folds. This is important because we want to make sure that we do not accidentally train on the same hotel in the training and test set.

set.seed(47)  
splits2 <- group\_initial\_split(dta2, prop = 0.8, group = hotel\_id)  
dta\_test2 <- testing(splits2)  
dta\_train2 <- training(splits2)  
  
set.seed(48)  
folds2 <- group\_vfold\_cv(dta\_train2, v = 5, group = hotel\_id)  
print(folds2)

# Group 5-fold cross-validation   
# A tibble: 5 × 2  
 splits id   
 <list> <chr>   
1 <split [3056/768]> Resample1  
2 <split [3056/768]> Resample2  
3 <split [3056/768]> Resample3  
4 <split [3064/760]> Resample4  
5 <split [3064/760]> Resample5

And we define new models

r0 <- recipe(dta\_train2) |>   
 update\_role(prop\_renov, new\_role = "outcome")  
  
r1 <- r0 |>   
 update\_role(room\_freq, new\_role = "predictor")   
  
r2 <- r1 |>   
 update\_role(c(family\_guests, young\_tourists, business\_traveller),   
 new\_role = "predictor")  
r3 <- r1 |>   
 update\_role(price\_rank, new\_role = "predictor") |>  
 step\_dummy(price\_rank)  
  
r4 <- r1 |>   
 update\_role(price\_rank, new\_role = "predictor") |>  
 step\_dummy(price\_rank)  
  
r5 <- r3 |>   
 step\_sqrt(room\_freq)  
  
r6 <- r4 |>   
 step\_sqrt(room\_freq)  
  
# Here we "accidentally" add another transformation to room\_freq after taking  
# the square root. This is just to illustrate that you can chain transformations  
r7 <- r6 |>   
 step\_poly(room\_freq)  
  
# Here we leave out the root transformation step  
r8 <- r4 |>   
 step\_poly(room\_freq)  
  
model\_set2 <-   
 workflow\_set(  
 preproc = list(m1 = r1, m2 = r2, m3 = r3, m4 = r4, m5 = r5,   
 m6 = r6, m7 = r7, m8 = r8),  
 models = list(LR = lin\_model),  
 cross = TRUE  
 )  
  
model\_set2\_fits <-   
 model\_set2 |>   
 workflow\_map("fit\_resamples",   
 # Options to `workflow\_map()`:   
 seed = 1101, verbose = TRUE,  
 # Options to `fit\_resamples()`:   
 resamples = folds2, control = control\_resamples(save\_pred = TRUE)  
 )

i 1 of 8 resampling: m1\_LR

✔ 1 of 8 resampling: m1\_LR (264ms)

i 2 of 8 resampling: m2\_LR

✔ 2 of 8 resampling: m2\_LR (272ms)

i 3 of 8 resampling: m3\_LR

✔ 3 of 8 resampling: m3\_LR (490ms)

i 4 of 8 resampling: m4\_LR

✔ 4 of 8 resampling: m4\_LR (474ms)

i 5 of 8 resampling: m5\_LR

✔ 5 of 8 resampling: m5\_LR (493ms)

i 6 of 8 resampling: m6\_LR

✔ 6 of 8 resampling: m6\_LR (449ms)

i 7 of 8 resampling: m7\_LR

✔ 7 of 8 resampling: m7\_LR (551ms)

i 8 of 8 resampling: m8\_LR

✔ 8 of 8 resampling: m8\_LR (461ms)

results\_ranked2 <- rank\_results(model\_set2\_fits, rank\_metric ="rmse")  
results\_ranked2

# A tibble: 16 × 9  
 wflow\_id .config .metric mean std\_err n preprocessor model rank  
 <chr> <chr> <chr> <dbl> <dbl> <int> <chr> <chr> <int>  
 1 m8\_LR Preprocessor1… rmse 0.0175 1.13e-4 5 recipe line… 1  
 2 m8\_LR Preprocessor1… rsq 0.893 5.63e-3 5 recipe line… 1  
 3 m7\_LR Preprocessor1… rmse 0.0195 2.67e-4 5 recipe line… 2  
 4 m7\_LR Preprocessor1… rsq 0.870 5.42e-3 5 recipe line… 2  
 5 m2\_LR Preprocessor1… rmse 0.0247 9.62e-4 5 recipe line… 3  
 6 m2\_LR Preprocessor1… rsq 0.793 6.46e-3 5 recipe line… 3  
 7 m1\_LR Preprocessor1… rmse 0.0254 9.27e-4 5 recipe line… 4  
 8 m1\_LR Preprocessor1… rsq 0.781 5.59e-3 5 recipe line… 4  
 9 m3\_LR Preprocessor1… rmse 0.0254 9.24e-4 5 recipe line… 5  
10 m3\_LR Preprocessor1… rsq 0.781 5.63e-3 5 recipe line… 5  
11 m4\_LR Preprocessor1… rmse 0.0254 9.24e-4 5 recipe line… 6  
12 m4\_LR Preprocessor1… rsq 0.781 5.63e-3 5 recipe line… 6  
13 m5\_LR Preprocessor1… rmse 0.0307 1.36e-3 5 recipe line… 7  
14 m5\_LR Preprocessor1… rsq 0.679 1.16e-2 5 recipe line… 7  
15 m6\_LR Preprocessor1… rmse 0.0307 1.36e-3 5 recipe line… 8  
16 m6\_LR Preprocessor1… rsq 0.679 1.16e-2 5 recipe line… 8

Because we have transformed the outcome variable, the RMSE is now in terms of renovation proportions and not renovations anymore. So, when we want to compare our new predictions with the old ones we need to create a new variable that is the predicted proportion of renovations times the number of rooms (line 15 below)

best\_model\_id2 <- results\_ranked2$wflow\_id[1]  
  
chosen\_wf2 <- extract\_workflow(model\_set2, best\_model\_id2)  
chosen\_wf\_fit2 <- fit(chosen\_wf2, data = dta\_train2)  
chosen\_wf\_pred2 <-   
 predict(chosen\_wf\_fit2, new\_data = dta\_test2) |>   
 bind\_cols(dta\_test2) |>   
 mutate(pred\_renov = n\_rooms \* .pred)  
  
head(chosen\_wf\_pred2)

# A tibble: 6 × 15  
 .pred hotel\_id year location\_id business\_traveller young\_tourists  
 <dbl> <fct> <dbl> <fct> <dbl> <dbl>  
1 0.0727 2 1 51 0.502 0.0980  
2 0.0577 6 1 1 0.598 0.101   
3 0.0518 7 1 20 0.302 0.300   
4 0.0549 25 1 15 0.400 0.102   
5 0.0723 30 1 57 0.600 0.201   
6 0.0520 34 1 30 0.199 0.0987  
# ℹ 9 more variables: family\_guests <dbl>, other <dbl>, n\_rooms <dbl>,  
# n\_guests <dbl>, price\_rank <ord>, n\_renovations <dbl>, prop\_renov <dbl>,  
# room\_freq <dbl>, pred\_renov <dbl>

Just out of curiosity, do we have any observations, where we predict a negative proportion? (Which would be obviously non-sensical?)

chosen\_wf\_pred2 |>   
 filter(.pred < 0)

# A tibble: 0 × 15  
# ℹ 15 variables: .pred <dbl>, hotel\_id <fct>, year <dbl>, location\_id <fct>,  
# business\_traveller <dbl>, young\_tourists <dbl>, family\_guests <dbl>,  
# other <dbl>, n\_rooms <dbl>, n\_guests <dbl>, price\_rank <ord>,  
# n\_renovations <dbl>, prop\_renov <dbl>, room\_freq <dbl>, pred\_renov <dbl>

It does not look like it. That is a good sign for our model fit. For the range of predictors we have in the test sample, we do not get obvious non-sense

rmse(chosen\_wf\_pred2, truth = n\_renovations, estimate = pred\_renov)

# A tibble: 1 × 3  
 .metric .estimator .estimate  
 <chr> <chr> <dbl>  
1 rmse standard 4.35

mae(chosen\_wf\_pred2, truth = n\_renovations, estimate = pred\_renov)

# A tibble: 1 × 3  
 .metric .estimator .estimate  
 <chr> <chr> <dbl>  
1 mae standard 3.34

p1 <-   
 autoplot(model\_set2\_fits, metric = "rmse") +  
 ggrepel::geom\_text\_repel(aes(label = wflow\_id),   
 nudge\_x = 1/8, nudge\_y = 1/100, size = 3.5) +  
 theme(legend.position = "none")  
  
p2 <-   
 autoplot(model\_set2\_fits, metric = "rsq") +  
 ggrepel::geom\_text\_repel(aes(label = wflow\_id),   
 nudge\_x = 1/8, nudge\_y = 1/100, size = 3.5) +  
 theme(legend.position = "none")  
  
(p1 + p2)

|  |
| --- |
| Figure 8.6: Prediction performance of different models in terms of RMSE and R-squared |

Our transformation did seem to help. We managed to get the RMSE down quite a bit and also slightly reduced the MAE. We can double check the coefficients of our regression model as well to see whether the coefficients are looking reasonable. However, there is an important caveat here. The coefficients might not be intuitively interpretable. That is because we have not designed the regression with isolating specific associations in mind. We only cared about prediction performance when designing the model. You need a DAG again, to see whether and how you can interpret the coefficients.

chosen\_wf\_fit2 |> tidy()

# A tibble: 7 × 5  
 term estimate std.error statistic p.value  
 <chr> <dbl> <dbl> <dbl> <dbl>  
1 (Intercept) 0.0737 0.000283 260. 0   
2 price\_rank\_1 -0.0000395 0.000633 -0.0624 0.950  
3 price\_rank\_2 0.000293 0.000633 0.463 0.643  
4 price\_rank\_3 -0.000655 0.000632 -1.04 0.300  
5 price\_rank\_4 -0.000831 0.000634 -1.31 0.190  
6 room\_freq\_poly\_1 2.94 0.0175 168. 0   
7 room\_freq\_poly\_2 1.14 0.0175 65.1 0

chosen\_wf\_pred2 |>   
 ggplot(aes(y = pred\_renov, x = n\_renovations, color = n\_rooms)) +   
 scale\_y\_continuous(limits = range(chosen\_wf\_pred$n\_renovations)) +   
 scale\_x\_continuous(limits = range(chosen\_wf\_pred$n\_renovations)) +   
 geom\_abline(intercept = 0, slope = 1, color = "coral1") +   
 geom\_point(alpha = 0.9, shape = 21) +  
 labs(y = "predicted n\_renovations") +   
 theme(legend.position = "right") +   
 coord\_fixed(ratio = 1)

|  |
| --- |
| Figure 8.7: Predicted values versus actual outcomes |

It looks like the transformed regression really helped us doing a better job at the outlying observations (That is also why the RMSE went down more than the MAE). We also seem to do a better job with the hotel-years with low number of observations. This is a common reason why we want to think about transformations in the first place. Whenever we see that we have trouble predicting a specific part of the distribution correctly, we might want to see whether we can transform the distribution to something easier to fit.

## 8.8 Takeaway

Similar to our example in [Chapter 5](#sec-pred), we still need to decide whether we can use the last model for our cost forecasts problem. The individual error still seems large. It is 22% of the median of number of renovations. So, how should we proceed. We could carry on and try to improve the model. Rethink our mental model based on what we learned and maybe, if possible, collect additional predictors that we have not yet considered. Before doing that though, we should first remember that our goal is to forecast group-level renovation costs. So we a) also need costs pre renovation (which likely varies by location and hotel) and b) need to look at the error in agreggate renovations. Because that is ultimately what we want to forecast. If individual errors cancel out, we might do okay with the model we have already. So let us have a look at aggregate errors

chosen\_wf\_pred2 |>   
 summarize(  
 agg\_n\_renov = sum(n\_renovations),  
 agg\_pred\_n\_renov = sum(pred\_renov),  
 .by = year  
 )

# A tibble: 8 × 3  
 year agg\_n\_renov agg\_pred\_n\_renov  
 <dbl> <dbl> <dbl>  
1 1 2179 2186.  
2 2 2426 2460.  
3 3 2205 2201.  
4 4 2255 2243.  
5 5 2225 2256.  
6 6 2088 2149.  
7 7 2075 2106.  
8 8 2473 2499.

That looks quite okay for all the years we have date in our test sample. We need cost per renovation to make the ultimate decision, but this model might be good enough to carry on with your budget planing.

# References

Anguera, Ricard. 2006. “The Channel Tunnel—an Ex Post Economic Evaluation.” *Transportation Research Part A: Policy and Practice* 40 (4): 291–315.

Aruldoss, Martin, T Miranda Lakshmi, and V Prasanna Venkatesan. 2013. “A Survey on Multi Criteria Decision Making Methods and Its Applications.” *American Journal of Information Systems* 1 (1): 31–43.

Gelman, Andrew, and Jennifer Hill. 2006. *Data Analysis Using Regression and Multilevel/Hierarchical Models*. Cambridge university press.

Ginsberg, Jeremy, Matthew H Mohebbi, Rajan S Patel, Lynnette Brammer, Mark S Smolinski, and Larry Brilliant. 2009. “Detecting Influenza Epidemics Using Search Engine Query Data.” *Nature* 457 (7232): 1012–14. <https://www.nature.com/articles/nature07634>.

Lazer, David, Ryan Kennedy, Gary King, and Alessandro Vespignani. 2014. “The Parable of Google Flu: Traps in Big Data Analysis.” *Science* 343 (6176): 1203–5. <https://www.science.org/doi/full/10.1126/science.1248506>.

Lunenburg, Fred C. 2010. “The Decision Making Process.” In *National Forum of Educational Administration & Supervision Journal*. Vol. 27. 4.

Marchau, Vincent AWJ, Warren E Walker, Pieter JTM Bloemen, and Steven W Popper. 2019. *Decision Making Under Deep Uncertainty: From Theory to Practice*. Springer Nature.

Pomerol, Jean-Charles, and Sergio Barba-Romero. 2000. *Multicriterion Decision in Management: Principles and Practice*. Vol. 25. Springer Science & Business Media.

Schaffernicht, Martin FG. 2017. “Causal Attributions of Vineyard Executives–a Mental Model Study of Vineyard Management☆.” *Wine Economics and Policy* 6 (2): 107–35.

Schoenfeld, Alan H. 2010. *How We Think: A Theory of Goal-Oriented Decision Making and Its Educational Applications*. Routledge.

1. Again, disclaimer, these are simulated data [↑](#footnote-ref-57)
2. And if the relation between the variables is roughly linear. More on that in [Chapter 5](#sec-pred). [↑](#footnote-ref-65)
3. OLS minimizes the sum of squared errors, which in turn means we care about mean values. [↑](#footnote-ref-71)
4. Some view causal analysis as distinct, some as very special and case of predictive analysis. See Gelman and Hill (2006), chapter 9. [↑](#footnote-ref-78)
5. There are interesting philosophical problems here that we don’t have the time to cover. For example, if we didn’t include competitive dynamics into our model, is this model uncertainty or residual uncertainty? It depends. We think of a model to cover the relevant dynamics. For example, if competitive dynamics affect the relation between our main variables of interest, then we have a wrong model. If competitive dynamics only affect the main variable of interest but not others, then we might not necessarily have a bad model for the problem of interest. [↑](#footnote-ref-84)
6. As we perform our investigation at the branch level, average employee tenure implies that the tenure of all employees at the respective branch is averaged. Technically, we need to be careful here. As the average tenure is itself estimated, there is additional noise in our x variable that we should account for. We ignore this in the example, for ease of exposition. [↑](#footnote-ref-214)
7. Meaning the test statistic follows a t-distribution. It is computed a slightly differently than, for example, a differences-in-means t-test. The formula is: [↑](#footnote-ref-215)
8. There is sometimes a trade-off however. If the variables we take out of the error term are highly correlated with our variables of interest, we create a new problem called colinearity. [↑](#footnote-ref-227)
9. We simulated this data and the true coefficient on foot traffic is 210€ in weekly revenues per weekly customer and the true coefficient on staff is 360€ in weekly revenues per staff member. [↑](#footnote-ref-232)
10. Some call this the “table 1 fallacy”. [↑](#footnote-ref-235)
11. There is also a model F-test, but we ignore that. It is rarely useful for model selection. [↑](#footnote-ref-238)
12. In our lecture, we have not focused much on unsupervised learning methods, such as clustering, mainly due to time constraints. We will see over time whether we can add them at some point. [↑](#footnote-ref-252)
13. The interpretation of the actual number of the MSE, the 1448.7, depends on the scale of your outcome variable . Whether a particular number is high or low thus always depends on context. [↑](#footnote-ref-264)