# Backwards Design

## Contents


- Overview
- 1) Define Your Problem
- 2) Define the Question You Wish to Answer
- 3) Write Down What An Answer Would Look Like
- 4) What Data Do You Need?
- 5) Where Can You Get That Data?
- Wrapping Up
- A Digression on Supervised Machine Learning


Backwards Design is a way of developing an efficient strategy for completing a new data science project, and in my view it is one of the most important skills of a professional data scientist.

If you don't have a lot of professional data science experience, it may not be obvious why this is an important skill, or even why I call it a "skill." That's because most data science students' experience with project development comes from classroom exercises or sites like kaggle. These types of projects are excellent opportunities for learning, but it is usually the case that – unbeknownst to the student – these projects have been carefully tailored to have clearly defined goals, and they come with data sets that have been cleaned and filter to provide only relevant variables. This is usually done for good reasons – the instructors design these exercises in a way that focuses student attention on the skills that they are trying to develop (like model selection or model interpretation). But as a result, students often come away with the impression that most of what data scientists do is work with statistical models.

In reality, however, often the most important thing that a data scientist does is (a) develop and articulate a concrete, feasible objective of a data science project, and (b) develop a strategy for achieving that objective efficiently. And Backwards Design is one of the best ways to go about

Skip to main content

# Overview

As the name implies, the idea Backwards Design is to *start* by clearly defining where you want to end up at the end of the project, and then working backwards to figure out exactly what you need to do to get there. Backwards Design is actually a common project management strategy and a range of different domains, and so you may already be familiar with the strategy and broad terms. In this class, however, we will focus on a five-step strategy for doing Backwards Design in data science:

1. Define the problem you want to solve.
2. Define a *question* that you wish to answer to help you solve this problem.
3. Articulate exactly what an answer to your question would look like.
4. Determine the variables you would need in order to generate that answer.
5. Identify data sets with those variables, and develop a strategy for bringing them together.

# 1) Define Your Problem

This first step should be the most straightforward, and yet you will be surprised at how often it is never actually explicitly addressed. People get so excited about the idea of data science that they will often come to you (the data scientist) with the data set and say "do some data science with this!" So the first thing you should always do when starting a data science project is make sure that you can clearly articulate the objective of the project. In addition, you should always make sure that *your stakeholder* agrees with that articulation of the problem you are trying to address! There's nothing worse than spending weeks on a project and then discovering that it's not actually a value to your stakeholder.

Here are a few examples of defined problems:

- We don't know how to reduce mass incarceration.
- My business can't identify potential new customers.
- We don't know who is going to develop Alzheimers, so we can't test early interventions.

Skip to main content

# 2) Define the Question You Wish to Answer

Although not everyone will agree with us, it is my view the data science is fundamentally the practice of using data to quantifiably answer questions about the world.

For example, when we ran our regressions of birth weight on various demographic variables and whether the mother smoked during pregnancy, those models were answering the question "is maternal smoking associated with lower birth weight (at a statistically significant level after controlling for other confounds)?"

If someone who runs a commerce website runs an A/B test where users visiting the site are randomly assigned to see two versions of a new landing page, and we then track their purchasing behavior, then when we analyze that data statistically what we're doing is answering the question "Which of these designs is more effective at getting customers to buy things?"

And finally we can think of supervised machine learning algorithms as answering two types of question: there's the broad question that you answer by building a model and evaluating it ("can we identify cancer from patient x-rays, and if so, how well?"), and then the narrow question the is answered each time the model is run ("given this x-ray, how likely is this patient to have cancer?"). (There's a small digression on supervised machine learning and this "data science is about answering questions" conceptual framework at the end of this if you're interested).



So the next step in backwards design is to ask yourself: what question, if answered, would help you solve the problem that motivates you?

## Defining a Good Question

A key feature of a good question is one that it is *concrete*, *tractable* and *answerable* by a data science project. Your question meets these criteria if it **directly implies how you should approach the data science project, and that approach seems feasible.** If your question is so vague that you don't immediately start thinking about the data you want to collect, it's not a good motivating question.

To illustrate, here are a set of *bad* questions to the three problems described above:

- What policies reduce mass incarceration?

Skip to main content

- What indicates Alzheimers?

By contrast, here's a set of concrete, tractable, and answerable questions:

- Does the availability of grand juries result in longer sentences? (Grand juries are a pool of citizens prosecutors can ask for guidance on sentencing. In theory they're supposed to hold prosecutors accountable, but in reality its often said that prosectors can shape the information grand juries get so they reach the recommendations that prosecutors wanted to begin with).
- What attributes are common to the customers who buy the most from my business?
- Are there lab results common in patients who later develop Alzheimers (diagnosed post-mortem) that we don't see in patients who don't go on to develop Alzheimers?

For the first set, we've asked questions, but they're so vague that it's not clear how you should approach answering the question. In the second set, by contrast, likely first steps are very clear. For example, the first good question clearly implies we need to find data on sentencing from places with and without grand juries. For the second question, its clear we need data on our current customers, and data from the general population for comparison. And for the last question we clearly want lab data from patients with and without diagnosed Alzheimers!

Moreover, for these answerable questions, you can imagine what the answer to the question will look like: for the first, you could have a regression that regresses sentences on grand jury availability controlling for crime committed; and for the second, you could imagine a table that compares various demographic characteristics of customers to non-customers.

# Why is Having a Good Question Important?

- It will save you from getting lost in your data, since it helps you focus you energy.
- Being able to articulate the question you wish to answer allows you to make sure that answering that question will actually help address your motivating problem (/make sure that your stateholder agrees that answering your question will help them). There's *nothing* worse than getting excited, diving into your data, doing lots of work, and then getting a result that doesn't actually help address the problem that motivated you (but it happens *all the time*).

Skip to main content

# 3) Write Down What An Answer Would Look Like

Seriously. Do it. Not abstractly: I mean draw the graph, figure, table, dataset with columns of predicted values and predictors, or set of model diagnostics you want to generate as a way of answering your question. Literally draw the graph, label the axes, etc.

Why?

- If you can't, then it turns out your question wasn't sufficiently concrete.
- You can then show this to your stakeholder to ensure they think this constitutes an answer that would help them solve their problem (and avoid later being told your work doesn't actually help them)
- It makes it even clearer what steps you need to do next to generate this result.

## Falsifiability

OK. So now you're written down what an answer to your question looks like. But there's one other key feature of a good question that we didn't get into above: it should be *falsifiable*, which means that (a) you should be able to articulate a hypothesis about the answer to your question *and* (b) know what a result to your question would look like.

So when writing down what your answer should look like, do the following:

1. State a hypothesis about what you think the answer to your question is likely to be.
2. Draw what an answer to your question would look like *if your hypothesis is true*.
3. Draw what an answer to your question would look like *if your hypothesis is false*.

For example, consider our question about grand juries and sentecing. My hypothesis might be that grand juries result in longer sentences because they insulate prosectors from accountability.

My result, as described above, could be a regression table where I regress sentences on whether a county has a grand jury along with controls for crime committed.

The answer if my hypothesis is true is that we'd have a positive coefficient on the presence of

Skip to main content

The answer if my hypothesis is false is that we'd have a zero or negative coefficient on the presence of grand juries.

Why do all this? Because it helps ensure that the way we're planning to answer our question will actually answer it by generating different results in different states of the world.

# 4) What Data Do You Need?

Congratulations! You've now completed the really hard part of a data science project: defining your goals. Now we turn to the easy stuff.

First, now you know your goal – the result described above – we turn to how we will actually answer our question. So ask yourself:

- What variables do I need to make the result I described above,
- What population do I need represented in that data

So let's think about our business trying to find new customers. Clearly, we need data on (a) customer spending, and (b) the demographics of those same customers.

We also need data on *both* people who spend a lot, and people who don't spend a lot so we can compare these two populations. We could do this either by getting data on all our customers and comparing big spenders from people who don't buy much (if there's a lot of variation in the data on level of spending), or we could compare current customers to the general public (most of whom aren't customers).

# 5) Where Can You Get That Data?

OK, now you know the variables you need measured and the populations for whom you need those variables. Now let's figure out:

- Where can I get that data, and
- If the data will come from different datasets, how will I combine them?

In the customer example above, for example, we can start by looking for the data the company already has on its current customers. What's in that data?

Skip to main content

We can also look for similar demographic data for non-customers using a resource like the American Community Survey (the annual survey run by the US Census bureau).

If we use government data for our comparison group, we'll then have to make sure we get comparable samples, so we'll want to make sure we can match our observations on things like age, gender, and where people live, so we'll need to make sure we have those variables in both datasets.

# Wrapping Up

Congratulations!

By the time you've done these 5 steps, you've managed to develop a concrete plan for exactly where you'll focus your time, and you've nearly guaranteed that the result you're working to generate will actually be useful in solving the problem that motivates you or your stakeholder. There's still a lot of data wrangling, model selection, etc. ahead, but at least you won't get lost in your data, or do lots of work that ends up no helping anyone!

**Want a template for this?** Great news! You can download one here!

# A Digression on Supervised Machine Learning

As noted above, we can think of superived machine learning as a tool that answers two types of questions: the first is the broad question of "can we predict [outcome of interest] using [variables we have] and the training set we have access to?", and the second is the more narrow prediction question "for a given set of predictors, what value of [outcome of interest] would the model predict for a given observation"?

The first, I think, is pretty straightforward. But there's a nuance to the second question that's super important to understand: when we ask a supervised machine learning it's prediction for a given observation, what we're fundamentally asking your model is: **"how do you think the entity who labeled the data in your training data set would label this new observation?"**

Because that's all that supervised machine learning does: it develops models that are designed to replicate the behavior that gave rise to the data set due used for training your model. For example, if you train a supervised machine learning algorithm to label pictures with the name of

Skip to main content

undergraduates at an American university, than what you are training that machine learning algorithm to do is answer the question "how would an American undergraduate label this photo" every time it sees an unlabeled photograph.

Obviously different supervised machine learning algorithms go about trying to answer this question in different ways, and some will be more successful than others depending on the context (which is why we spend so much time studying model selection in machine learning courses), but answering this question is *always* the goal to which they aspire.

This is a bit of a digression, but I think it's an important one: recognizing that this is all supervised machine learning algorithms do is important because it helps you, the data scientist, understand the limitations of supervised machine learning algorithms. For a surprisingly long time, people thought that machine learning algorithms were incapable of harboring racial or sexist prejudices. They are, after all, just built of math, and math can't be racist, can it? And so companies like amazon tried to build supervised machine learning algorithms to help them decide who to hire. The problem, though, is that they trained them using data on which people human employees had decided to hire in the past, and data from subjective employee evaluations that had been made by human supervisors. And because this gave rise to an algorithm that looked at people's resumes and asked itself "what would Amazon's (very human) hiring staff and supervisors have thought of this person?," the algorithm of course inherited all the biases of those humans. And so, OOPS!, when Amazon suddenly realized that "their new recruiting engine didn't like women," they had to abandon the project.

OK, digression on bias in data science complete. For now. :)