# Nanodegree of Data Science

#### **Introduction to Data Science**

##### Lesson1: The Data Science process

**The CRISP-DM Process (Cross Industry Process for Data Mining)**

The lessons leading up to the first project are about helping you go through CRISP-DM in practice from start to finish. Even when we get into the weeds of coding, try to take a step back and realize what part of the process you are in, and assure that you remember the question you are trying answer and what a solution to that question looks like.

1. **Business Understanding** - this means understanding the problem and questions you are interested in tackling in the context of whatever domain you're working in. Examples include

* How do we acquire new customers?
* Does a new treatment perform better than an existing treatment?
* How can improve communication?
* How can we improve travel?
* How can we better retain information?

2. **Data Understanding** - at this step, you need to move the questions from **Business Understanding** to data. You might already have data that could be used to answer the questions, or you might have to collect data to get at your questions of interest.

3. **Data Preparation**

Luckily stackoverflow has already collected the data for us. However, we still need to wrangle the data in a way for us to answer our questions. The wrangling and cleaning process is said to take 80% of the time of the data analysis process. You will see that will hold true through this lesson, as a majority of the remaining parts of this lesson will be around basic data wrangling strategies.

We will discuss the advantages and disadvantages of the strategies discussed in this lesson.

————-

When looking at the first two questions:

1. How to break into the field?
2. What are the placement and salaries for those who attended a coding bootcamp?

we did not need to do any predictive modeling. We only used **descriptive** and a little **inferential** statistics to retrieve the results.

Therefore, all steps of CRISP-DM were not necessary for these first two questions. CRISP-DM states 6 steps:

1. Business Understanding

2. Data Understanding

3. Prepare Data

4. Data Modeling

5. Evaluate the Results

6. Deploy

For these first two questions, we did not need step 4. In the previous notebooks, you performed steps 3 and 5 without needing step 4 at all. A lot of the hype in data science, artificial intelligence, and deep learning is integrated into step 4, but there are still plenty of questions to be answered not using machine learning, artificial intelligence, and deep learning.

**All Data Science Problems Involve**

1. Curiosity.
2. The **right** data.
3. A tool of some kind (Python, Tableau, Excel, R, etc.) used to find a solution (You could use your head, but that would be inefficient with the massive amounts of data being generated in the world today).
4. Well communicated or deployed solution.

**Extra Useful Tools to Know But That Are NOT Necessary for ALL Projects**

* Deep Learning
* Fancy machine learning algorithms

With that, you will be getting a more in depth look at these items, but it is worth mentioning (given the massive amount of hype) that they do not solve all the problems. Deep learning cannot turn bad data into good conclusions. Or bad questions into amazing results.

When looking at the first two questions:

1. How to break into the field?
2. What are the placement and salaries for those who attended a coding bootcamp?

we did not need to do any predictive modeling. We only used **descriptive** and a little **inferential** statistics to retrieve the results.

Therefore, all steps of CRISP-DM were not necessary for these first two questions. The process would look closer to the following:

1. Business Understanding

2. Data Understanding

3. Prepare Data

4. Evaluate the Results

5. Deploy

However, for the last two questions:

1. How well can we predict an individual's salary? What aspects correlate well to salary?
2. How well can we predict an individual's job satisfaction? What aspects correlate well to job satisfaction?

We will need to use a predictive model. We will need to pick up at step 3 to answer these two questions, so let's get started. The process for answering these last two questions will follow the full 6 steps shown here.

1. Business Understanding

2. Data Understanding

3. Prepare Data

4. Model Data

5. Evaluate the Results

6. Deploy

**Dealing with missing data**

Dropping the rows/columns with missing data

Though **dropping** rows and/or columns holding missing values is quite easy to do using numpy and pandas, it is often not appropriate.

Understanding why the data is missing is important before dropping these rows and columns. In this video you saw a number of situations in which dropping values was not a good idea. These included

1. Dropping data values associated with the effort or time an individual put into a survey.
2. Dropping data values associated with sensitive information.

In either of these cases, the missing values hold information. A quick removal of the rows or columns associated with these missing values would remove missing data that could be used to better inform models.

Instead of removing these values, we might keep track of the missing values using indicator values, or counts associated with how many questions an individual skipped.

In the last video, you saw cases in which dropping rows or columns associated with missing values would not be a good idea. There are other cases in which dropping rows or columns associated with missing values would be okay.

A few instances in which dropping a row might be okay are:

1. Dropping missing data associated with mechanical failures.
2. The missing data is in a column that you are interested in predicting.

Other cases when you should consider dropping data that are not associated with missing data:

1. Dropping columns with no variability in the data.
2. Dropping data associated with information that you know is not correct.

In handling removing data, you should think more about why is this missing or why is this data incorrectly input to see if an alternative solution might be used than dropping the values.

One common strategy for working with missing data is to understand the proportion of a column that is missing. If a large proportion of a column is missing data, this is a reason to consider dropping it.

There are easy ways using pandas to create dummy variables to track the missing values, so you can see if these missing values actually hold information (regardless of the proportion that are missing) before choosing to remove a full column.

Imputing missing data

Imputation is likely the most common method for working with missing values for any data science team. The methods shown here included the frequently used methods of imputing the mean, median, or mode of a column into the missing values for the column.

There are many advanced techniques for imputing missing values including using machine learning and bayesian statistical approaches. This could be techniques as simple as using k-nearest neighbors to find the features that are most similar, and using the values those features have to fill in values that are missing or complex methods like those in the very popular [**AMELIA library**](https://cran.r-project.org/web/packages/Amelia/Amelia.pdf).

Regardless your imputation approach, you should be very cautious of the **BIAS** you are imputing into any model that uses these imputed values. Though imputing values is very common, and often leads to better predictive power in machine learning models, it can lead to over generalizations. In extremely advanced techniques in Data Science, this can even mean [**ethical implications**](https://intelligence.org/files/EthicsofAI.pdf). Machines can only 'learn' from the data they are provided. If you provide biased data (due to imputation, poor data collection, etc.), it should be no surprise, you will achieve results that are biased.

One of the most common methods for working with missing values is by imputing the missing values. Imputation means that you input a value for values that were originally missing.

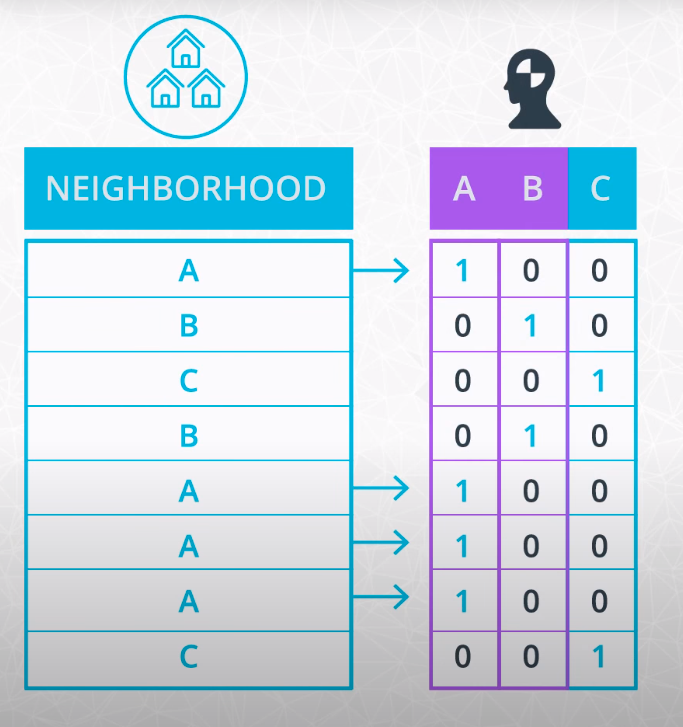
It is very common to impute in the following ways:

* Impute the mean of a column.
* If you are working with categorical data or a variable with outliers, then use the mode of the column.
* Impute 0, a very small number, or a very large number to differentiate missing values from other values.
* Use knn to impute values based on features that are most similar.

In general, you should try to be more careful with missing data in understanding the real world implications and reasons for why the missing values exist. At the same time, these solutions are very quick, and they enable you to get models off the ground. You can then iterate on your feature engineering to be more careful as time permits.

**Categorical variables**

The best way to work with categorical variables is with ones and zeros as seen in the image.



**Overfitting**

Overfitting is a common problem when our model does not generalize to data it has not seen before. Assuring you build models that not only work for the data the model was trained on, but also generalize to new (test) data, is key to building models that will be successful to deploy and that will become successful in production.

#### **Software Engineering**

##### Lesson1: Software Engineering Practices

###### Clean and Modular Code

* **PRODUCTION CODE:** software running on production servers to handle live users and data of the intended audience. Note this is different from *production quality code*, which describes code that meets expectations in reliability, efficiency, etc., for production. Ideally, all code in production meets these expectations, but this is not always the case.
* **CLEAN:** readable, simple, and concise. A characteristic of production quality code that is crucial for collaboration and maintainability in software development.
* **MODULAR:** logically broken up into functions and modules. Also an important characteristic of production quality code that makes your code more organized, efficient, and reusable.
* **MODULE:** a file. Modules allow code to be reused by encapsulating them into files that can be imported into other files.

###### Refactoring Code

* **REFACTORING:** restructuring your code to improve its internal structure, without changing its external functionality. This gives you a chance to clean and modularize your program after you've got it working.
* Since it isn't easy to write your best code while you're still trying to just get it working, allocating time to do this is essential to producing high quality code. Despite the initial time and effort required, this really pays off by speeding up your development time in the long run.
* You become a much stronger programmer when you're constantly looking to improve your code. The more you refactor, the easier it will be to structure and write good code the first time.

**Writing Modular Code**

***Tip: DRY (Don't Repeat Yourself)***

*Don't repeat yourself! Modularization allows you to reuse parts of your code. Generalize and consolidate repeated code in functions or loops.*

***Tip: Abstract out logic to improve readability***

*Abstracting out code into a function not only makes it less repetitive, but also improves readability with descriptive function names. Although your code can become more readable when you abstract out logic into functions, it is possible to over-engineer this and have way too many modules, so use your judgement.*

***Tip: Minimize the number of entities (functions, classes, modules, etc.)***

*There are tradeoffs to having function calls instead of inline logic. If you have broken up your code into an unnecessary amount of functions and modules, you'll have to jump around everywhere if you want to view the implementation details for something that may be too small to be worth it. Creating more modules doesn't necessarily result in effective modularization.*

***Tip: Functions should do one thing***

*Each function you write should be focused on doing one thing. If a function is doing multiple things, it becomes more difficult to generalize and reuse. Generally, if there's an "and" in your function name, consider refactoring.*

***Tip: Arbitrary variable names can be more effective in certain functions***

*Arbitrary variable names in general functions can actually make the code more readable.*

***Tip: Try to use fewer than three arguments per function***

*Try to use no more than three arguments when possible. This is not a hard rule and there are times it is more appropriate to use many parameters. But in many cases, it's more effective to use fewer arguments. Remember we are modularizing to simplify our code and make it more efficient to work with. If your function has a lot of parameters, you may want to rethink how you are splitting this up.*

**Experimental Design and Recommendations**

Lesson2: Concepts in experimental design

**Types of Study**

There are many ways in which data can be collected in order to test or understand the relationship between two variables of interest. These methods can be put into three main bins, based on the amount of control that you hold over the variables in play:

* If you have a lot of control over features, then you have an **experiment**.
* If you have no control over the features, then you have an **observational study**.
* If you have some control, then you have a **quasi-experiment**.

While the experiment is the main focus of this course, it's also useful to know about the other types of study so that you can use them in effective ways, especially if an experiment cannot be run.

**Experiments**

In the social and medical sciences, an experiment is defined by comparing outcomes between two or more groups, and ensuring equivalence between the compared groups *except* for the manipulation that we want to test. Our interest in an experiment is to see if a change in one feature has an effect in the value of a second feature, like seeing if changing the layout of a button on a website *causes* more visitors to click on it. Having multiple groups is necessary in order to compare the outcome for when we apply the manipulation to when we do not (e.g. old vs. new website layout), or to compare different levels of manipulation (e.g. drug dosages). We also need equivalence between groups so that we can be as sure as possible that the differences in the outcomes were only due to the difference in our manipulated feature.

Equivalence between groups is typically carried out through some kind of randomization procedure. A **unit of analysis** is the entity under study, like a page view or a user in a web experiment. If we randomly assign our units of analysis to each group, then on the whole, we should expect the feature distributions between groups to be about the same. This theoretically isolates the changes in the outcome to the changes in our manipulated feature. Of course, we can always dig deeper afterwards to see if certain other features worked in tandem with, or against, our manipulation.

**Observational Studies**

In an experiment, we exert a lot of control on a system in order to narrow down the changes in our system from one source to one output. Observational studies, on the other hand, are defined by a lack of control. Observational studies are also known as naturalistic or correlational studies. In an observational study, no control is exerted on the variables of interest, perhaps due to ethical concerns or a lack of power to enact the manipulation. This often comes up in medical studies. For example, if we want to look at the effects of smoking on health, the potential risks make it unethical to force people into smoking behaviors. Instead, we need to rely on existing data or groups to make our determinations.

We typically cannot infer causality in an observational study due to our lack of control over the variables. Any relationship observed between variables may be due to unobserved features, or the direction of causality might be uncertain. (We'll discuss this more later in the lesson.) But simply because an observational study does not imply causation does not mean that it is not useful. An interesting relationship might be the spark needed to perform additional studies or to collect more data. These studies can help strengthen the understanding of the relationship we're interested in by ruling out more and more alternative hypotheses.

**Quasi-Experiments**

In between the observational study and the experiment is the quasi-experiment. This is where some, but not all, of the control requirements of a true experiment are met. For example, rolling out a new website interface to all users to see how much time they spend on it might be considered a quasi-experiment. While the manipulation is controlled by the experimenter, there aren't multiple groups to compare. The experimenter can still use the behavior of the population pre-change and compare that to behaviors post-change, to make judgment on the effects of the change. However, there is the possibility that there are other effects outside of the manipulation that caused the observed changes in behavior. For the example earlier in this paragraph, it might be that users would have naturally gravitated to higher usage rates, regardless of the website interface.

As another example, we might have two different groups upon which to make a comparison of outcomes, but the original groups themselves might not be equivalent. A classic example of this is if a researcher wants to test some new supplemental materials for a high school course. If they select two different schools, one with the new materials and one without, we have a quasi-experiment since the differing qualities of students or teachers at those schools might have an effect on the outcomes. Ideally, we'd like to match the two schools before the test as closely as possible, but we can't call it a true experiment since the assignment of student to school can't be considered random.

While a quasi-experiment may not have the same strength of causality inference as a true experiment, the results can still provide a strong amount of evidence for the relationship being investigated. This is especially true if some kind of matching is performed to identify similar units or groups. Another benefit of quasi-experimental designs is that the relaxation of requirements makes the quasi-experiment more flexible and easier to set up.

**Types of Experiment**

Most of the time, when you think of an experiment, you think of a **between-subjects** experiment. In a between-subjects experiment, each unit only participates in, or sees, one of the conditions being used in the experiment. The simplest of these has just two groups or conditions to compare. In one group, we have either no manipulation, or maintenance of the status quo. This is like providing a known drug treatment, or an old version of a website. This is known as the **control group**. The other group includes the manipulation we wish to test, such as a new drug or new website layout. This is known as our **experimental group**. We can compare the outcomes between groups (e.g. recovery time or click-through rate) in order to make a judgement about the effect of our manipulation. (Since we have an experiment, we'll randomly assign each unit to either the control or experimental group.) For web-based experiments, this kind of basic experiment design is called an **A/B test**: the "A" group representing the old control, and "B" representing the new experimental change.

We aren't limited to just two groups. We could have multiple experimental groups to compare, rather than just one control group and one experimental group. This could form an A/B/C test for a web-based experiment, with control group "A" and experimental groups "B" and "C".

If an individual completes all conditions, rather than just one, this is known as a **within-subjects** design. Within-subjects designs are also known as *repeated measures* designs. By measuring an individual's output in all conditions, we *know* that the distribution of features in the groups will be equivalent. We can account for individuals' aptitudes or inclinations in our analysis. For example, if an individual rates three different color palettes for a product, we can know if a high rating for one palette is particularly good compared to the others (e.g. 10 vs. 5, 6) or if it's not a major distinction (e.g. 10 vs. 8, 9).

Randomization still has a part in the within-subjects design in the order in which individuals complete conditions. This is important to reduce potential bias effects, as will be discussed later in the lesson. One other downside of the within-subjects design is that it's not always possible to pull off a within-subjects design. For example, when a user visits a website and completes their session, we usually can't guarantee when they'll come back. The purpose of their following visit also might not be comparable to their first. It can take a lot more effort in control in order to set up an effective within-subjects design.

**Side Note: Factorial Designs**

*Factorial designs* manipulate the value of multiple features of interest. For example, with two independent manipulations "X" and "Y", we have four conditions: "control", "X only", "Y only", "X and Y". Experimental designs where multiple features are manipulated simultaneously are more frequently seen in engineering and physical sciences domains, where the system units tend to be under stricter control. They're less seen in the social and medical realms, where individual differences can impede experiment creation and analysis.

**Types of Sampling**

While web and other online experiments have an easy time collecting data, collecting data from traditional methods involving real populations is a much more difficult proposition. If you need to perform a survey of a population, it could be unreasonable in both time and money costs to try and collect thoughts from every single person in the population. This is where sampling comes in. The goal of sampling is to only take a subset of the population, using the responses from that subset to make an inference about the whole population. Here, we'll cover two basic probabilistic techniques that are commonly used.

The simplest of these approaches is **simple random sampling**. In a simple random sample, each individual in the population has an equal chance of being selected. We just randomly make draws from the population until we have the sample size desired; your sample size depends on the level of uncertainty you are willing to have about the collected data. Since everyone has an equal chance of being drawn, we can expect the feature distribution of selected units to be similar to the distribution of the population as a whole. In addition, a simple random sample is easy to set up.

However, it is possible that certain groups are underrepresented in a simple random sample, especially those that make up a low proportion of the population. If there are certain rarer subgroups of interest, it can be worth adding one additional step and performing **stratified random sampling**. In a stratified random sample, we need to first divide the entire population into disjoint groups, or strata. That is, each individual must be a part of one group, and only one group. For example, you could divide people by gender (male, female, other), or age (e.g. 18-25, 26-35, etc.).

Then, from each group, you take a simple random sample. In a *proportional sample*, the sample size is proportional to how large the group is in the full population. For example, if you require 1000 data points, and stratified individuals of proportion {0.5, 0.3, 0.2}, then you would take 500 people from the first group, 300 from the second, and 200 from the third. This guarantees a certain level of knowledge from each subset, and theoretically a more representative overall inference on the population.

An alternative approach is to take a *nonproportional sample* from each group. For example, we could simply sample 500 people from each group. Computing the overall statistics in this case requires weighting each group separately, but this extra effort offers a higher understanding of each subgroup in a deeper investigation.

**Side Note: Non-Probabilistic Sampling**

As noted at the start, the goal of sampling is to use a subset of the whole population to make inferences about the full population, so that we didn't need to record data from everyone. To that end, probabilistic sampling techniques were described above to try and obtain a sample that was representative of the whole. However, it's useful to note that there also exist non-probabilistic sampling techniques that simplify the sampling process, at the risk of harming the validity of your results. (We'll discuss the term 'validity' later in the lesson.)

For example, a *convenience sample* records information from readily available units. Studies performed in the social sciences at colleges often fall into this kind of sampling. The people participating in these tasks are often just college students, rather than representatives of the population at large. When performing inferences from this type of study, it's important to consider how well your results might apply to the population at large.

One notable example of a convenience sample resulting in a grave error comes from the prediction made by magazine "The Literary Digest" on the 1936 U.S. presidential election. While they predicted a healthy victory by candidate Alf Landon, the final result ended with a landslide victory by opposing candidate Franklin D. Roosevelt. This major error is attributed to their methods capturing a non-representative sample of the population, which included looking at the results of a mail-in survey from their magazine readers. Since the mail-ins were voluntary, and the magazine subscribers were already not well-representative of the general population, focusing on the people who returned surveys gave a large bias toward Landon.

**Measuring Outcomes**

The goals of your study may not be the same as the way you evaluate the study's success. Perhaps this is because the goal is something that can't be measured directly. Let's say that you have an idea of a website addition that improves user satisfaction. How should we measure this? In order to evaluate whether or not this improvement has happened, you need to have a way to objectively measure the effect of the addition. For example, you might include a survey to random users to have them rate their website experience on a 1-10 scale. If the addition is helpful, then we should expect the average rating to be higher for those users who are given the addition, versus those who are not. The rating scale acts as a concrete way of measuring user satisfaction. These objective features by which you evaluate performance are known as **evaluation metrics**.

As a rule of thumb, it's a good idea to consider the goals of a study separate from the evaluation metrics. This provides a couple of useful benefits. First, this makes it clear that the metric isn't the main point of a study: it's the *implications* of the metric relative to the goal that matters. This is especially important if a metric isn't directly attached to the goal. For example, measuring students' confidence going into a standardized test might be a proxy for the goal of test preparedness, in the absence of being able to get their test scores directly or in a timely fashion.

Secondly, having the metric separate from the goal can clarify the purpose of conducting the study or experiment. It makes sure we can answer the question of *why* we want to run a study or experiment. From the above example, we aren't measuring confidence just to make people feel good about themselves: we're doing it to try and improve their actual performances.

**Side Note: Alternate Terminology**

You might hear other terminology for goals and evaluation metrics than those used in this course. In the social sciences, it's common to hear a "construct" as analogous to the goal or objective under investigation, and the "operational definition" as the way outcomes are measured. For example, the construct of "reaction time" could be operationally defined as "time in milliseconds to click on the correctly indicated button."

In general company operations, you might encounter the terms "key results" (KRs) or "key performance indicators" (KPIs) as ways of measuring progress against quarterly or annual "objectives." These objectives and KRs / KPIs serve a similar purpose as study goals and evaluation metrics, and might even be driving factors in the creation of an experiment.

**Creating Metrics**

**Funnels**

There are additional concepts and terms that are commonly used for designing experiments, especially for web-based studies. In a web experiment, you'll often think of the user **funnel**. A funnel is the flow of steps you expect a user of your product to take. Typically, the funnel ends at the place where your main evaluation metric is recorded, and includes a step where your experimental manipulation can be performed. For example, we might think of the following steps for someone to purchase a product in an online store:

* Visit the site homepage
* Search for a desired product or click on a product category
* Click on a product image
* Add the product to the cart
* Check out and finalize purchase

One property to note about user funnels is that typically there will be some dropoff in the users that move from step to step. This is much like how an actual funnel narrows from a large opening to a small exit. Outside of an experiment, funnels can be used to analyze user flows. Observations from these flows can then be used to motivate experiments to try and improve the dropoff rates.

It's also worth noting that the flow through a funnel might be idealized compared to actual user practice. In the above example, users might perform multiple searches in a single session, or want to purchase multiple things. A user might access the site through a specific link, subverting the top part of the funnel. Refining the funnel and being specific about the kinds of events that are expected can help you create a consistent, reliable design and analysis.

**Unit of Diversion**

Once you have a funnel, think about how you can implement your experimental manipulation in the funnel. If the goal of the above experiment was to change the way the site looks after a user clicks on a product image, we need to figure out a way to assign users to either a control group or experimental group. The place in which you make this assignment is known as the **unit of diversion**. Depending on the type of experiment you have, you might have different options for diversion, each with its own pros and cons:

* Event-based diversion (e.g. pageview): Each time a user loads up the page of interest, the experimental condition is randomly rolled. Since this ignores previous visits, this can create an inconsistent experience, if the condition causes a user-visible change.
* Cookie-based diversion: A cookie is stored on the user's device, which determines their experimental condition as long as the cookie remains on the device. Cookies don't require a user to have an account or be logged in, but can be subverted through anonymous browsing or a user just clearing out cookies.
* Account-based diversion (e.g. User ID): User IDs are randomly divided into conditions. Account-based diversions are reliable, but requires users to *have* accounts and be logged in. This means that our pool of data might be limited in scope, and you'll need to consider the risks of using personally-identifiable information.

When it comes to selecting a unit of diversion, the consistency of the experience required can be a major factor to consider. For the example provided, we need something more consistent than pageview events. So we then consider the cookie-based diversion. If the differences in interface between control and experiment are fairly minor, then we're probably okay with cookie-based diversion. But if we think that users will notice the change and we believe that it will have a major effect on experience, then we might be inclined to choose an account-based diversion.

**Invariant and Evaluation Metrics**

A funnel will also be of benefit when it comes to deciding on metrics to track and analyze as part of the experiment. The immediate features that come out of a funnel come in the form of counts and ratios. For example, we could count the number of times a search results in a product being selected (a count), or the ratio of selections to searches as adjacent slices in the funnel (a ratio).

There are two major categories that we can consider features: as *evaluation metrics* or as *invariant metrics*. **Evaluation metrics** were mentioned in the previous page as the metrics by which we compare groups. Ideally, we hope to see a difference between groups that will tell us if our manipulation was a success. We might want to see an increased click-through-rate from search results to products, or an increase in overall revenue. On the flip side, **invariant metrics** are metrics that we hope will *not* be different between groups. Metrics in this category serve to check that the experiment is running as expected. For example, in an experiment with cookie-based diversion, the number of cookies generated for each condition would be a good invariant metric. Another metric could compare the distribution of times in which cookies were generated, to check the bias in the randomization procedure.

We're not limited to tracking just one metric of each type. It's not unusual to track multiple invariant metrics as checks on the system's integrity, or multiple evaluation metrics to check different potential facets of a manipulation's effects. Don't think that you need to track every possible metric, however. It's better to focus on a few key metrics, ignoring features that might be less reliable or highly correlated to other, more informative features. We'll discuss statistical considerations surrounding metrics in the next lesson.

**Controlling Variables**

If we want to determine causality between two features, there are two main things to control. First of all, we need to enact the manipulation on one of the features of interest, so that we know that it is causing the change in the other feature. In order to know that it was our manipulated variable and not any other, the second major control point is that we want to make sure that all other features are accounted for. These two requirements make the arguments for causality much stronger with an experiment compared to a quasi-experiment or observational study.

If we aren't able to control all features or there is a lack of equivalence between groups, then we may be susceptible to **confounding variables**. The correlation observed between two variables might be due to changes in a third variable, rather than one causing the other. Another possibility is that there is a causal relationship between the two features, but it is an indirect relationship mediated by a third, intermediate variable. This intermediate variable might be a larger driver of the changes in the output, with the manipulated variable only having a direct effect on the intermediate feature.

For the case where we see a relationship but don't perform a manipulation, we also need to be careful about the direction of effect. A relationship between variables "A" and "B" might be due to "A" having an effect on "B" or the reverse, "B" having an effect on "A". It might even be the case that "A" and "B" are related through some other function like a third variable.

**Checking Validity**

When designing an experiment, it's important to keep in mind validity, which concerns how well conclusions can be supported. There are three major conceptual dimensions upon which validity can be assessed:

* Construct Validity
* Internal Validity
* External Validity

**Construct Validity**

Construct validity is tied to the earlier discussion of how well one's goals are aligned to the evaluation metrics used to evaluate it. Poor construct validity can come about when an evaluation metric does not actually measure something related to the desired outcome concept. Alternatively, it might be that a metric is ill-constructed, such that it does not make clear distinctions on the outcome concept.

**Internal Validity**

Internal validity refers to the degree to which a causal relationship can be derived from an experiment's results. Controlling for and accounting for other variables is key to maintaining good internal validity. The previous page on controlling variables shows ways in which internal validity might not be met.

**External Validity**

External validity is concerned with the ability of an experimental outcome to be generalized to a broader population. This is most relevant with experiments that involve sampling: how representative is the sample to the whole? For studies at academic institutions, a frequent question is if data collected using only college students can be generalized to other age or socioeconomic groups.

**Checking Bias**

Biases in experiments are systematic effects that interfere with the interpretation of experimental results, mostly in terms of internal validity. Just as humans can have a lot of different [**biases**](https://en.wikipedia.org/wiki/List_of_cognitive_biases), there are numerous ways in which an experiment can become unbalanced.

**Sampling Bias**

Many experimental biases fall under the sampling bias umbrella. Sampling biases are those that cause our observations to not be representative of the population. For example, if assignment to experimental groups is done in an arbitrary fashion (as opposed to random assignment or matched groups), we risk our outcomes being based less on the experimental manipulation and more on the composition of the underlying groups.

Studies that use surveys to collect data often have to deal with the **self-selection bias**. The types of people that respond to a survey might be qualitatively very different from those that do not. A straight average of responses would not necessarily reflect the feelings of the full population; weighting responses based on the differences between the observed responses and properties of the target population may be needed to come to reasonable conclusions.

One type of sampling bias related to missing data is the **survivor bias**. Survivor bias is one where losses or dropout of observed units is not accounted for in an analysis. A key example of this was in British World War II operations research, where engineers *avoided* using survivor bias when they considered where to add armor to their planes. Rather than add armor to the spots where returning planes had bullet holes, armor was added to the spots where the planes *didn't* have bullet holes. That's because the planes that took shots to those places probably crashed, due to those locations being more vital for maintaining flight, so they didn't "survive" and weren't available for observation.

**Novelty Bias**

A novelty effect is one that causes observers to change their behavior simply because they're seeing something new. We might not be able to gauge the true effect of a manipulation until after the novelty wears off and population metrics return to a level that actually reflects the changes made. This will be important for cases where we want to track changes over time, such as trying to get users to re-visit a webpage or use an app more frequently. Novelty is probably not a concern (or perhaps what we hope for) when it comes to manipulations that are expected to only have a one-shot effect.

**Order Biases**

There are a couple of biases to be aware of when running a within-subjects experiment. Recall that in a within-subjects design, each participant performs a task or makes a rating in multiple experimental conditions, rather than just one. The order in which conditions are completed could have an effect on participant responses. A **primacy effect** is one that affects early conditions, perhaps biasing them to be recalled better or to serve as anchor values for later conditions. A **recency effect** is one that affects later conditions, perhaps causing bias due to being fresher in memory or task fatigue.

An easy way of getting around order biases is to simply randomize the order of conditions. If we have three conditions, then each of the six ways of completing the task (ABC, ACB, BAC, BCA, CAB, CBA) should be equally likely. While there still might end up being order effects like carry-over effects, where a particular condition continues to have an effect on future conditions, this will be much easier to detect than if every participant completed the task in the exact same order of conditions.

**Experimenter Bias**

One bias to watch out for, especially in face-to-face experiments, is the experimenter bias. This is where the presence or knowledge of the experimenter can affect participants' behaviors or performance. If an experimenter knows what condition a participant is in, they might subtly nudge the participant towards their expected result with their interactions with the participant. In addition, participants may act differently in the presence of an experimenter, to try and act in the 'right' way – regardless of if a subject *actually* knows what the experimenter is looking for or not.

This is where design steps like blinding are important. In [**blinding**](https://en.wikipedia.org/wiki/Blinded_experiment), the administrator of a procedure or the participant do not know the condition being used, to avoid that subconscious bias from having an effect. In particular, the **double-blind** design hides condition information from both the administrator *and* participant in order to have a strong rein on experimenter-based biases.

**Ethics in Experimentation**

Before you run an experiment, it's important to consider the ethical treatments to which you subject your participants. Through the mid-20th century, exposure of questionable and clearly unethical research in the social and medical sciences spurred the creation of guidelines for ethical treatment of human subjects in studies and experiments. While different fields have developed different standards, they still have a number of major points in common:

* **Minimize participant risk**: Experimenters are obligated to construct experiments that minimize the risks to participants in the study. Risk of harm isn't just in the physical sense, but also the mental sense. If an experimental condition has potential to negatively affect a participant's emotions or mentality, then it's worth thinking about if the risks are really necessary to perform the desired investigation.
* **Have clear benefits for risks taken**: In some cases, risks may be unavoidable, and so they must be weighed against the benefits that may come from performing the study. When expectations for the study are not clearly defined, this throws into question the purpose of exposing subjects to risk. However, if the benefits are shown to be worth the risks, then it is still possible for the study to be run. This often comes up in medicine, where test treatments should show worthy potential against alternative approaches.
* **Provide informed consent**: Building up somewhat from the previous two points, subjects should be informed of and agree to the risks and benefits of participation before they join the study or experiment. This is also an opportunity for a participant to opt out of participation. However, there are some cases where deception is necessary. This might be to avoid biasing the participant's behavior by seeding their expectations, or if there is a dummy task surrounding the actual test to be performed. In cases like this, it's important to include a debriefing after the subject's participation so that they don't come away from the task feeling mislead.
* **Handle sensitive data appropriately**: If you're dealing with identifiable information in your study, make sure that you take appropriate steps to protect their anonymity from others. Sensitive information includes things like names, addresses, pictures, timestamps, and other links from personal identifiers to account information and history. Collected data should be anonymized as much as possible; surveys and census results are often also aggregated to avoid tracing outcomes back to any one person.

In the formal sciences, an experiment proposal must go through a review board before it can be run, to ensure that ethical principles have been followed. It's likely that you won't have a review board to submit your designs to prior to running an experiment. You'll need to evaluate these principles for yourself or with your colleagues to check for potential ethical issues before going forward with a study design.

One particular point worth further discussion is that of informed consent for web-based experiments. It's often the case that when an experiment is run, users who are included in an experiment often don't know that they're participating in an experiment. If a manipulation carries no risk and is so minor as to be hidden away from the user (e.g. a change in recommendation engine), perhaps there is no need for informed consent. And when it comes to bias, it's known that peoples' behaviors can change when they know they are under observation. In practice, informed consent is often not considered when performing a web experiment.

However, informed consent is still an important ethical principle, so there is continuing debate on how to best obtain consent for users of a website. One option could be to allow users to opt out of experiment participation, with the default user agreement implying consent to participation in unobtrusive experiments. An opposing option would only run experiments on users who opt-\_in\_ to participation, asking the user to set their preference on their initial visit or registration. The opt-in approach is more in line with the core idea of informed consent, but also risks fewer users available for testing changes.

**Examples in Experimental Ethics**

Here are three studies in the social and medical sciences that are often brought up as examples of violations of experimental ethics and progenitors of the movement to establish ethics guidelines and boards:

* [**Tuskegee Syphilis Study**](https://en.wikipedia.org/wiki/Tuskegee_syphilis_experiment): This study was started in 1932, where hundreds of African-American men were tracked over the course of up to forty years to study the natural progression of syphilis. The subjects were denied treatment, even after the development of effective syphilis treatments like penicillin, and there were active steps taken to hide the truth of their conditions and treatments to the subjects.
* [**Milgram obedience study**](https://en.wikipedia.org/wiki/Milgram_experiment): During the mid 1960s, psychologist Stanley Milgram tested to what degree people follow authority figures. Participants were asked to administer gradually increasing shocks to an acting confederate participant at the behest of a lead experimenter based on mistakes made on a dummy memory test used as a cover story. While no shocks were *actually* administered, the study did bing forth questions on what constitutes adequate debriefing and what level of information and informed consent needs to be provided to a participant in a study that includes necessary deceptive elements.
* [**Stanford Prison Experiment**](https://en.wikipedia.org/wiki/Stanford_prison_experiment): This 1971 study conducted by psychologist Philip Zimbardo was built to test the dynamics and effects of power differences, using a prison scenario. Study volunteers were divided into prisoner and guard groups and their dynamics observed; the study had to be ended after less than a week due to the increasingly harsh conditions the 'guards' had settled into treating the 'prisoners'. It has since served as a major ground for ethical criticisms, violating now-established guidelines around risks to participants and clarity of purpose. There are also methodological criticisms, as experimenter biases may have shaped the behavior of the 'guards' group in their interactions with 'prisoners'.

And here's a [**Techcrunch article**](https://techcrunch.com/2014/06/29/ethics-in-a-data-driven-world/) discussing the ethics of the 2014-published Facebook study on the impact of changing the affect of posts seen on users' feeds to their own posting habits.

**SMART Mnemonic for Experiment Design**

There's a mnemonic called **SMART** for teams to plan out projects that also happens to apply pretty well for creating experiments. The letters of SMART stand for:

* **S**pecific: Make sure the goals of your experiment are specific.
* **M**easurable: Outcomes must be measurable using objective metrics
* **A**chievable: The steps taken for the experiment and the goals must be realistic.
* **R**elevant: The experiment needs to have purpose behind it.
* **T**imely: Results must be obtainable in a reasonable time frame.

There are also other words possible for the mnemonic, such as **A**ctionable and **R**ealistic. Ultimately, however, the message is pretty much the same, with the dimensions switched around a bit. Note, however, that the mnemonic doesn't cover certain considerations important to experiment design. Considerations of ethical issues or bias will need to be considered separately, so don't just take the mnemonic as the final judge of whether your experiment is ready to proceed!

Lesson3: Statistical Considerations in Testing

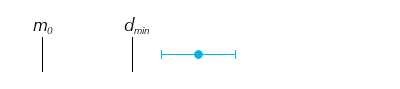
**Practical Significance**

Even if an experiment result shows a statistically significant difference in an evaluation metric between control and experimental groups, that does not necessarily mean that the experiment was a success. If there are any costs associated with deploying a change, those costs might outweigh the benefits expected based on the experiment results. **Practical significance** refers to the level of effect that you need to observe in order for the experiment to be called a true success and implemented in truth. Not all experiments imply a practical significance boundary, but it's an important factor in the interpretation of outcomes where it is relevant.

If you consider the confidence interval for an evaluation metric statistic against the null baseline and practical significance bound, there are a few cases that can come about.

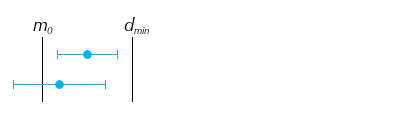
**Confidence interval is fully in practical significance region**

(Below, m\_0*m*0​ indicates the null statistic value, d\_{min}*dmin*​ the practical significance bound, and the blue line the confidence interval for the observed statistic. We assume that we're looking for a positive change, ignoring the negative equivalent for d\_{min}*dmin*​.)



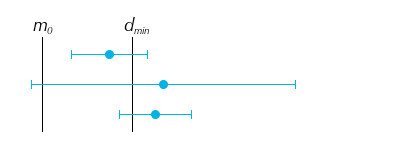
If the confidence interval for the statistic does not include the null or the practical significance level, then the experimental manipulation can be concluded to have a statistically and practically significant effect. It is clearest in this case that the manipulation should be implemented as a success.

**Confidence interval completely excludes any part of practical significance region**



If the confidence interval does not include any values that would be considered practically significant, this is a clear case for us to not implement the experimental change. This includes the case where the metric is statistically significant, but whose interval does not extend past the practical significance bounds. With such a low chance of practical significance being achieved on the metric, we should be wary of implementing the change.

**Confidence interval includes points both inside and outside practical significance bounds**



This leaves the trickiest cases to consider, where the confidence interval straddles the practical significance bound. In each of these cases, there is an uncertain possibility of practical significance being achieved. In an ideal world, you would be able to collect more data to reduce our uncertainty, reducing the scenario to one of the previous cases. Outside of this, you'll need to consider the risks carefully in order to make a recommendation on whether or not to follow through with a tested change. Your analysis might also reveal subsets of the population or aspects of the manipulation that **do** work, in order to refine further studies or experiments.

**Experiment size**

After computing the number of observations needed for an experiment to reliably detect a specified level of experimental effect (i.e. statistical power), we need to divide by the expected number of observations per day in order to get a minimum experiment length. We want to make sure that an experiment can be completed in a reasonable time frame so that if we do have a successful effect, it can be deployed as soon as possible and resources can be freed up to run new experiments. What a 'reasonable time frame' means will depend on how important a change will be, but if the length of time is beyond a month or two, that's probably a sign that it's too long.

There are a few ways that an experiment's duration can be reduced. We could, of course, change our statistical parameters. Accepting higher Type I or Type II error rates will reduce the number of observations needed. So too will increasing the effect size: it's much easier to detect larger changes.

Another option is to change the unit of diversion. A 'wider' unit of diversion will result in more observations being generated. For example, you could consider moving from a cookie-based diversion in a web-based experiment to an event-based diversion like pageviews. The tradeoff is that event-based diversion could create inconsistent website experiences for users who visit the site multiple times.

**Using Dummy Tests**

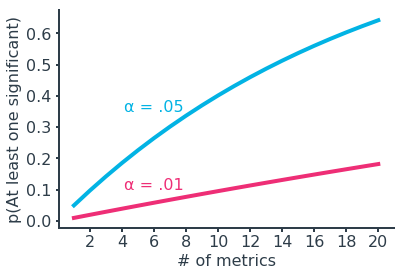
When it comes to designing an experiment, it might be useful to run a dummy test as a predecessor to or as part of that process. In a dummy test, you will implement the same steps that you would in an actual experiment to assign the experimental units into groups. However, the experimental manipulation won't actually be implemented, and the groups will be treated equivalently.

There are multiple reasons to run a dummy test. First, a dummy test can expose if there are any errors in the randomization or assignment procedures. A short dummy test can be worth the investment if an invariant metric is found to have a statistically significant difference, or if some other systematic bias is identified, because it can help avoid larger problems down the line. A second reason to run a dummy test is to collect data on metrics' behaviors. If historic data is not enough to predict the outcome of recorded metrics or allow for experiment duration to be computed, then a dummy test can be useful for getting baselines.

Of course, performing a dummy test requires an investment of resources, the most important of which is time. If time is of the essence, then you may need to just go ahead with the experiment, keeping an eye on invariant metrics for any trouble. An alternative approach is to perform a hybrid test. In the A/B testing paradigm, this can take the form of an A/A/B test. That is, we split the data into three groups: two control and one experimental. A comparison between control groups can be used to learn about null-environment properties before making inferences on the effect of the experimental manipulation.

**Analyzing Multiple Metrics**

If you're tracking multiple evaluation metrics, make sure that you're aware of how the Type I error rates on individual metrics can affect the overall chance of making some kind of Type I error. The simplest case we can consider is if we have \_n\_ independent evaluation metrics, and that seeing one with a statistically significant result would be enough to call the manipulation a success. In this case, the probability of making at least one Type I error is given by \alpha\_{over} = 1 - (1-\alpha\_{ind})^n*αover*​=1−(1−*αind*​)*n*, illustrated in the below image for individual \alpha\_{ind} = .05*αind*​=.05 and \alpha\_{ind} = .01*αind*​=.01:

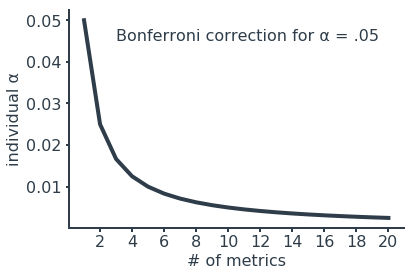


To protect against this, we need to introduce a correction factor on the individual test error rate so that the overall error rate is at most the desired level. A conservative approach is to divide the overall error rate by the number of metrics tested:

\alpha\_{ind} = \alpha\_{over}/n*αind*​=*αover*​/*n*

This is known as the **Bonferroni correction**. If we assume independence between metrics, we can do a little bit better with the **Šidák correction**:

\alpha\_{ind} = 1-(1-\alpha\_{over})^{1/n}*αind*​=1−(1−*αover*​)1/*n*



The Šidák correction is only slightly higher than the line drawn by the Bonferroni correction.

In real life, evaluation scenarios are rarely so straightforward. Metrics will likely be correlated in some way, rather than being independent. If a positive correlation exists, then knowing the outcome of one metric will make it more likely for a correlated metric to also point in the same way. In this case, the corrections above will be more conservative than necessary, resulting in an overall error rate smaller than the desired level. (In cases of negative correlation, the true error rate could go either way, depending on the types of tests performed.)

In addition, we might need multiple metrics to show statistical significance to call an experiment a success, or there may be different degrees of success depending on which metrics appear to be moved by the manipulation. One metric may not be enough to make it worth deploying a change tested in an experiment. Reducing the individual error rate will make it harder for a truly significant effect to show up as statistically significant. That is, reducing the Type I error rate will also increase the Type II error rate – another conservative shift.

Ultimately, there is a small balancing act when it comes to selecting an error-controlling scheme. Being fully conservative with one of the simple corrections above means that you increase the risk of failing to roll out changes that actually have an impact on metrics. Consider the level of dependence between metrics and what results are needed to declare a success to calibrate the right balance in error rates. If you need to see a significant change in all of your metrics to proceed with it, you might not need a correction factor at all. You can also use dummy test results, bootstrapping, and permutation approaches to plan significance thresholds. Finally, don't forget that practical significance can be an all-important quality that overrides other statistical significance findings.

While the main focus of this page has been on interpretation of evaluation metrics, it's worth noting that these cautions also apply to invariant metrics. The more invariant metrics you test, the more likely it will be that some test will show a statistically significant difference even if the groups tested are drawn from equivalent populations. However, it might not be a good idea to apply a correction factor to individual tests since we want to avoid larger issues with interpretation later on. As mentioned previously, a single invariant metric showing a statistically significant difference is not necessarily cause for alarm, but it is something that merits follow-up in case it does have an effect on our analysis.

**Early Stopping**

As the above workspace shows, there are significant risks for peeking ahead and making an early decision if it is not planned for in the design. If you haven't accounted for the effects of peeking on your error rate, then it's best to resist the temptation to look at the results early, and only perform a final analysis at the end of the experiment. This is another reason why it's important to design an experiment ahead of any data collection.

Note that there *are* ways of putting together a design to allow for making an early decision on an experiment. In the workspace, we showed how to treat the problem like a multiple comparisons problem, adjusting the individual test-wise error rate to preserve an overall error rate. For continuous tracking, [**this page**](https://www.evanmiller.org/sequential-ab-testing.html) describes a rule of thumb for rate-based metrics, tracking the number of successes in each group and stopping the experiment once the counts' sum or difference exceeds some threshold. More generally, tests like the [**sequential probability ratio test**](https://en.wikipedia.org/wiki/Sequential_probability_ratio_test) can be developed to make an early stopping decision while an experiment is running, if it looks statistically unlikely for a metric to move past or fall back against the statistical significance bound.

Lesson6: Introduction to Recomendation Engines

**Introduction**

Recommendations are being used to recommend everything from movies to music to friends to new destinations. There are three main methods for implementing recommendations that you will become familiar with throughout this lesson:

* Knowledge Based Recommendations
* Collaborative Filtering Based Recommendations
* Content Based Recommendations

After completing this lesson, you will be ready for the upcoming lessons where you will

* Learn about more advanced techniques.
* Deploy your recommendations in a web application.

These three lessons will aim to be extremely practical. The lessons will require that you write code to implement a number of different recommendation techniques.

**Knowledge Based Recommendations**

A knowledge based recommendation is one in which knowledge about the item or user preferences are used to make a recommendation.

Knowledge based recommendations are pretty common when purchasing luxury items. Take a look at the filters available on Zillow in the image below. This is an example of building in a knowledge based recommendation, as users can add their own preferences to the items that are provided.

Often a rank based algorithm is provided along with knowledge based recommendations to bring the most popular items in particular categories to the user's attention.

In the next concept, you will get some practice implementing this type of recommendation for the MovieTweetings dataset.

**Collaborative Filtering**

There are two main ways to implement collaborative filtering:

1. **Model Based Collaborative Filtering**
2. **Neighborhood Based Collaborative Filtering**

In this lesson, we will cover Neighborhood Based Collaborative Filtering, which is used to identify items or users that are "neighbors" with one another.

There are a number of ways we might go about finding an individual's closest neighbors - the metrics we will take a closer look at include:

1. Pearson's correlation coefficient
2. Spearman's correlation coefficient
3. Kendall's Tau
4. Euclidean Distance
5. Manhattan Distance

On the next page, you will work through a few examples to get more familiar with how each of these metrics is computed, and why you might use one over another.

**Three Types of Recommendations**

There are three methods that you have now implemented for making recommendations. These are the three most recognized methods in industry:

1. Knowledge Based Recommendations

Knowledge based recommendations frequently are implemented using filters, and are extremely common amongst luxury based goods. Filters that you might see when purchasing items like cars or homes are examples of knowledge based recommendations. In knowledge based recommendations, users provide information about the types of recommendations they would like back.

2. Collaborative Filtering Based Recommendations

Collaborative filtering uses the connections between users and items to make recommendations. Even the content based recommendation you just implemented used some collaborative filtering techniques, as you were not treating items and users independent from one another. In this lesson, you used neighborhood based collaborative filtering to find users who were alike and then recommend new movies based on these similar users.

Even in the content based recommendation, you were using collaborative filtering. You were finding items that were similar and making recommendations of new items based on the highest ratings of a user. Because you were still using the user ratings of an item, this was an example of a blend between content and collaborative filtering based techniques.

3. Content Based Recommendations

In the previous notebook, you created a matrix of similarities between items (movies) based only on the content related to those movies (year and genre). The similarity matrix that was used, was completely created using only the items (movies). There was no information used about the users implemented. For any movie, you would be able to determine the most related additional movies based only on the genre and the year of the movie. This is the premise of how a completely content based recommendation would be made.

Often blended techniques of all three types are used in practice to provide the the best recommendation for a particular circumstance.

There are still more advanced techniques that are related to the methods that you learned about here, and they will most likely fall in one of the three buckets below.

**Different Scales**

If you are in control of choosing your rating scale, think of what might be most beneficial to your scenario. If you are working alongside a team TO design the interfaces for how data will be collected, there are number of ideas to keep in mind.

* Do you need to ask questions of your user or can you collect data about their interactions with items?
* If you need to ask questions, how many do you ask?
* How do you word the questions?
* And finally, the question in the above video: what type of scale should you use?

In general, I suggest using the simplest rating that allows you to get whatever questions of interest you have, but there are some important ideas to keep in mind when choosing a particular type of rating. Ratings are a necessary part of working with different recommendation systems, but they aren't a central part of our focus. A good overview of types of ratings and when to use them is also provided [**here**](https://conversionxl.com/blog/survey-response-scales/).

Most of these ideas are specific to your use case, and are easy to notice in hindsight. It is simply important to think of this in advance and not completely gloss over possible issues with the data you are collecting and how it connects to the questions you want answered.

**NEXT**

**Business Goals of Recommendations**

In general, recommendations are important because they are often central to driving revenue for a company. In the new world of the Internet, you don't always need a friend to personally recommend you an item you never heard of - instead, the recommendations can do this.

There are ultimately 4 goals to keep in mind when performing recommendations:

1. **Relevance**
2. **Novelty**
3. **Serendipity**
4. **Diversity**

Often people only think of the first goal, and it is a good reminder that focusing only on **relevance** can lead to drops in use, and therefore loss of revenue. We also need to consider the other three goals, so that recommendations can be most effective in driving sales. [**In this article, Lab41 shows how they worked to quantify these metrics.**](https://gab41.lab41.org/recommender-systems-its-not-all-about-the-accuracy-562c7dceeaff)

Lesson7: Matrix Factorization for Recommendations

**Training and Testing Data For Recommendations**

In the last lesson, you were making recommendations by providing a list of popular items, or a list of items that the user hadn't observed but that someone with similar tastes had observed. However, understanding if these recommendations are good in practice means that you have to deploy these recommendations to users and see how it impacts your metrics (sales, higher engagement, clicks, conversions, etc.).

You may not want your recommendations to go live to understand how well they work. In these cases, you will want to split your data into training and testing portions. In these cases, you can train your recommendation engine on a subset of the data, then you can test how well your recommendation engine performs on a test set of data before deploying your model to the world.

However, the cases you saw in the last lesson, where just a list of recommendations was provided, don't actually lend themselves very well to training and testing methods of evaluation. In the next upcoming pages, you will be introduced to matrix factorization, which actually does work quite well for these situations.