Introduction

Hello everyone, welcome to this course, Data Analytics: Hands On. My name is Ben Sullins, and I'm going to walk you through this module now on getting started. First let's talk about our course goals. So, we're going to learn the basics of data analytics from start to finish. You're going to understand all the pieces, how they all fit together, and a good functional understanding of each different component. The best way to do that, I think, is to follow a story, something that is a real world example of something you might encounter in the world today in the process of analyzing data to glean information out of it. There are two methods of learning that this course offers you. There's the start to finish idea where if you're new to the data analytics world or really just want to get an understanding of the whole thing, I encourage you to watch from start to finish, starting with this module all the way through the last one. It's really going to be a cohesive experience, where all the pieces throughout are going to fit together, and you'll gain a complete holistic view of how the world of data analytics actually works. The other way, is to just fill in the gaps. If you're familiar with the data analytics world, or you're new to it, but have some experience, and you just want to learn about little pieces, you can also use this as a reference library. So each module is compartmentalized to where it in itself is going to teach you about that topic. So you don't have to watch end to end if you already have a lot of other experience in a lot of other areas, and you just want to learn bits about different aspects that you may be unfamiliar with.

Module Layout

Each module's going to cover four basic questions. What is it? So each topic here will answer what it is. We'll talk about why it exists, or what function does it actually serve, why is it a part of this world of data analytics. We'll understand how it works, so how whatever the topic is actually functions, and how it fits in with the rest of the world of data analytics. And then finally, we'll talk about where to learn more. We have a lot of resources on Pluralsight, and it may be hard to navigate and find which ones are specific to each area, at the end of every module, I'm going to point you to those.

Skills Overview

In this course you're going to learn several skills, starting with the most basic, fundamental one, I believe, for data analysts, and that's Excel. Hopefully you have some Excel background, but if not, no worries, I'll get you ramped up in there. And even if you are an experienced Excel user, there's probably still some value, some little hidden gems, and the ways of using Excel to help us with our analytics, that maybe is new to you. So I encourage you to watch those pieces, even if you already are familiar with Excel. Next, we'll go a little bit deeper, and we'll learn SQL. SQL stands for Structured Query Language. It's how we talk to data when it lives inside of a database. Now, you may wonder, well, what about Excel? We have our data in Excel. Well at some point, the data is too big and too cumbersome to manage in Excel, and it comes in the form of a database. This is how we organize and structure data in large quantities. And all databases out there generally have the ability to understand SQL. It's a universal language, so no matter what system you have, whether Oracle, or SQL Server, or MySQL, there are ways that you can understand this language, and query it for data, and get information out of it for your analysis. We'll also go through this concept of databases and data warehousing. Where do databases come from, why do they exist, how are they formed, and what are they structured like? Same with data warehousing, what's the difference? It's another type of database, but why is it different? Why does it exists? Then we'll look at when it comes to visualizing our data, how do we choose the right chart types? There are lots of them to choose from and increasingly more and more start popping up. How do we know which ones are actually going to effectively communicate our findings or our insights? We'll look at Tableau software, this is a visualization tool that plugs into Excel, as well as many other data sources, and it's really great in terms of just helping you visualize and explore your data. And last one, we'll get into presentation design. All of this is great, but if you can't stand up in front of a room and communicate your idea effectively, and really bring home the message of an answer to a question or an idea that you have, then none of this matters. So, I'm going really cover and emphasize that piece as well, that when design presentations, you do it in an effective way.

Our Story

Let's start now with our story. Cogsley Services Inc. was founded in 2008, and is a startup focusing on the technical services sector. They provide consulting services to various industries, and they have lots of different systems to manage their business processes. So, they're having a hard time pulling all that data together. Maybe they have data in Salesforce, and maybe data in Google Analytics, and they have all these different systems that the data is housed in, and they need a way to analyze it. Because often when you have a business question, it involves more than just a specific thing, especially if you're a fast growing company that has a ton of different sprawl of these different systems that manage the different parts of your business. So Cogsley is going to be our company that we're going to follow throughout this journey, and we're going to answer some questions for them. We're going to some analysis on some data, and all of that will be downloadable via the demo files, and we'll actually have the tangible working results in the end. Even if you aren't able to download these files, the demo portions of the course, they're going to be really easy to follow along with your own data.

What Is Data Analytics?

So now let's dive right in. What is data analytics? Well data analytics is the process of converting data into useful information. See without this process, the data itself is useless, it just sits there, there's nothing really meaningful coming out of it. So this is the process that actually converts those raw materials into something that is of value. Now when we go through this process it looks something like this. You have the analyst at the center of it. And it generally starts with acquiring some data. Once you have an idea that you want to vet, you want to have a hunch, or maybe you have a question, or somebody asking for some information to help them make a decision, we need to go get our data. Once we have our data, it's time to explore it. So we want to look at the data, we want to check it for things like blanks or errors in it, if we have country listed, and we have something like Chicago, the country is probably going to be USA, so things like that, make sure that those all things line up, and that the data itself is of a good quality. As we go through it we're going to visualize it, it's a great way to explore it, humans themselves are visual creatures, so what better way is there to actually explore your data than to look at it in a visual form? Once we have that, and we have a good structure visually, we're going to test that, we're going to take a look and sort and filter and see if the meaning is really there. Once we have that, we're going to study the findings. So we have something at this point, we have a visualization, we have something that answers our question. And often we'll give this back or we'll look at it and try to learn from it, and it will often lead to more questions. So the analytic cycle, it is all based around the analyst, and the analyst doing this kind of continuous iterations through this process. And the challenge is is that it's kind of a never ending cycle, in that once you answer one question, often another question pops up. So, we need to understand that it's an ongoing thing, and throughout the process, we're going to learn a lot, about our business, about our customers, about every aspect, and as an analyst you're responsible for all of them. The process generally doesn't follow this perfect cyclical nature. Usually what happens is, you start to get some data, you start to explore it, maybe at that point you realize you need more data, so you go back to acquire stage. At any rate through here, you can jump to another step that was in the process. But in general, and a perfect scenario, this should be the process we'd flow. So when you think about analytics, and how it works, this is the mindset you want to have. Let's go get some data, let's explore it, let's visualize it, let's test our results, see if they hold up, and then let's study it and try to learn something.

Why Data Analytics Exists

Now let's try to answer the questions, why does data analytics exist? Well data analytics exist because hidden in the data are these great treasures just waiting to be discovered. Let's just walk through an analogy here. As I mentioned before, data in itself is kind of like raw material, it's a bunch of bricks laying around, a bunch rock, some metal, some steel. It really takes some digging, and some churning, and some work on the data, in order for us to uncover things and discover them. Once we've uncovered the form and the structure, we can start to build things, and we can start to analyze them, and learn, and it's like this layer of learning, just learning on top of learning as we go through that cycle. And throughout the process, what we can do, in the end, if we really focus on being objective, using data and having the right mindset, and continuously learning and continuously exploring, we can build magnificent, fantastic data sets that provide all sorts of value, huge economics of scale. Think about Google for a second. Google, a multi-billion dollar company, one of the largest tech companies in the world, doesn't charge for the majority of its services. Google Search is free, Gmail is free, Google Analytics, the web analytics package is free. They have literally dozens of services that, in themselves are incredibly expensive to deliver. If they were selling their products, they would be charging enormous sums money. But they thought of it in a different way. Google puts data first, and to them, they think data is the product. So, while you search, and the cost of providing you with search results and a way to navigate the web is very expensive, the information they learn by tracking and understanding what people are searching for helps them target ads so that way when you search for something it gives you relevant information, and relevant advertisement, which then, for the people that are actually advertising, leads to a better use of their dollars. So data in the learning, throughout the process of search, as well as Gmail, as well as Google Maps, as well as many of their other services, comes from the actual use of the data and the learning that they are able to get out of it. So, these algorithms, and these data sets, and the learning, all of that becomes the product. And if there's any other shining example of it, it might be Facebook. Facebook is also a huge tech company that makes tons of money and delivers a great value to a lot of people, some people love it, some people hate it, but regardless, they are of the same mindset, Facebook is free, Facebook has tons of services that traditionally we would charge millions and millions of dollars for. But the deal is, the data that they collect, and the information they gather, all of that leads to a learning about how to better serve their customers, how to provide ad platforms, and other types of platforms for other types of business. So data really is the fuel that is the fire to most of these very large companies. So if you have that mindset, or if your company has that mindset, this is a really key process to understand, is the value of how you turn data into cash.

How It Works

Okay so how does it work? Let's get down to brass tacks here, let's really kind of dig in. Data undergoes a long journey before it's ready for analysis. Now this overview I'm going to show here describes that process. So it starts something like this. You have data, and let's talk about types of data, just in general. We have external events. A great example would be Facebook or Twitter. Somebody tweets about your company, somebody tweets about you. There's external information happening. People are meeting face to face, and are selling, and are talking about other things, and then those things are leading to effects that happen to your business, sales, or PR, things like that. Internally, we also have business systems, so external events such as marketing activities lead towards other things like leads, and conversions, and sales. So, all of those are tracked on business systems. Some common ones you may come across are Salesforce. com, Infusionsoft, or customer relationship management, possibly Google Analytics, or Adobe Analytics for digital marketing use cases. Lots of other cases there for business systems like NetSuite or QuickBooks for financial data. There's so many that it's impossible to name them all, but think that all businesses have some amount of computer software, some amount of technology powering their operations. Also, a lot of companies build internal apps. So some companies, their product is an app, so in that case, that would be an external source. But other companies have their own internal systems. Sometimes it makes more sense to build your own type of internal wiki, for example, or your own internal customer relationship management software. Whatever the case may be, there are many companies that spend a lot of time investing in their own internal apps. So all of these are places where data can be generated. And it's not ready for analysis yet in this point, maybe it is in little pockets, but if you want to look holistically, and really do analytics across the business or look at things outside of just one of these individual spots, you need to bring all that data in somewhere, so it can be commensurated and pulled together. This is often referred to as the back office. What I'm describing here is a very traditional approach, so, in general these are logical things as I get into the big data space, and I talk about in one of the later modules, you'll understand how maybe this same process occurs, but it's just in a maybe different format. So once the data's in our back office, we need to process that, we need to transform it. And think of this almost like a kitchen. So, the ingredients come in on a truck, or somebody goes to the farmers market, hopefully buys some beautiful vegetables and fruits and things to cook and prepare. In the kitchen, they are going to be preparing them. This is your construction workers, these are your chefs, these are your business intelligence team, these are your data warehousing teams. And then once that data is ready for use, it's going to go to the front office. And this is where it can be analyzed and consumed by the end users. So, almost like a buffet if I am just ready to eat I just grab my plate and I go get whatever I want. So those are the kind of scenarios we have, where data in the back office really isn't ready for mainstream consumption yet, it's still in a raw form. After it gets processed it's going to head up to the front office, and that's where we're going to be able to actually consume it. Some use cases we have for this data once it's there, is we can target our customers in terms of our marketing efforts. If you are consuming Pluralsight content per se, maybe we want to send you an email about the type of content that is relevant to you. It doesn't make sense if your IT person is sending you stuff about creative content, or developer content, maybe we should send you IT related content so we can target our efforts towards our customers to add value to what we're doing for them. We also can do business monitoring functions, this is a common one. We want to know what our sales like, what are our web traffic on our website look like. What are all these different things happening in the business that are happening in those business systems, and how can we monitor those just to keep a pulse on how the business is functioning? And then lastly, a common one is a strategic decision making. So if I'm planning for the next year, or if I'm wondering if we would change our price from, you know, 5. 99 to 3. 99 what would happen? How much would our profits go up or down? How many more people would we get? All those things can happen at the strategic level, so this data can power a lot of that, and hopefully is, because if not, you're really kind of flying blind, you're just going off of assumptions. And that's where this whole process really helps, is it gives you some truth in any of those decisions that you need to make. So some terms here are data providers, these are also called data sources, things that happen on the left. These could be databases, they could be web services, things like Salesforce that live in the cloud. In the middle we have our data warehouse, and as I mentioned when I get into the big data module, I'll explain kind of how that's evolving and how that's changing. These concepts are still pretty much there, where you have the raw unprocessed stuff, and then the processed ready to use stuff. And on the right here you just have some basic use cases which are common to almost all companies, ways of using data to extract value and improve business and all of those kind of things.

Where to Find More

Now let's learn where we can find more. Recently, Smarterer joined the Pluralsight family, and they're an assessment tool that you can go through and it's a really interesting and adaptive way of helping you assess your level of skill with a specific area. They have an analytics one, all of this is crowdsourced, so there are thousands of people uploading questions and taking them. So it's a very crowdsourced kind of social way of understanding where you stand in kind of the general population, in terms of your analytic skills. I would encourage you now, since it's the beginning of this course, to go take a look at that. Run through it, it shouldn't take more than 5 minutes, sometimes as little as 10 questions to assess where you are in terms of your skills. The other one is the Data Warehouse Institute. This is a mature organization that's been around for many years, that has helped grow the business intelligence data warehousing and data analytics practices and industry. They have many papers there and many resources for you to use and understand. A lot of it is very vendor agnostic, so it doesn't talk about a specific tool, it talks more about logically and conceptually how these things all work, and how they fit together. It can be a bit daunting, some of the topics they can be very advanced, so take a look, and I encourage you to join and maybe join a local chapter wherever city you're in, and you can kind of network with other people, and really just get some knowledge in this space. On our blog we do have a learning path for a data analyst, so there's a list of several courses there. Each one of those courses are in depth, and I'll probably be listing most of them throughout this course, but if you just want to jump over to that, feel free. And if you really want to try something fun, we can go over to Code School, which is also a company that joined the Pluralsight family recently, and with this they have a course called Try R. R is a statistical programming language, so it's a really fun interactive way of learning a specific data science type skillset. But if you're interested, maybe take a look at the videos and see if you can test your luck on that. It's really fun and an immersive way of learning, so I encourage you to at least have a go at it.

Excel Basics

Introduction

Hey everyone, welcome to this module on Excel Basics. I'm Ben Sullins, and I'm going to teach you here today all the basic knowledge that you need to be a good data analyst and understand where Excel fits in. In this module, we're going to talk about what Excel is, why people use Excel, how to use Excel, we'll go through some basic examples, and where you can find more information about Excel.

What Is Excel?

First, what is Excel? Excel is a desktop spreadsheet program that is the most versatile data management platform ever built. It's a strong statement, but it is very true. By far, Excel is the most widely used program in the world to manage, analyze, and understand data. Some of the key features are data exploration. With Excel it's very easy to visually analyze your data, to sort, filter, to jump around and look at different things with pivot tables, and summaries, and charts. All of these are built in. In fact, the next big feature, there is the pivot table. Pivot table is like a summary. So if you have a bunch of raw data, you can aggregate that and summarize it down. Imagine if you have a list of all of your sales orders for your company, you might want to see those grouped by customer, or grouped by month, you can do that using a pivot table. Excel also provides a very robust calculation engine. We won't get into too much of that detail today, but I will show you how some of the calculations can help us massage and manage the data, and come up with things like profit ratios, or conversion rates for marketing campaigns, all very meaningful metrics that you may want to use in your decision making in your business. Excel also offers a robust charting platform. The charts here are a bit more cumbersome to manage than some of the newer tools that have been focused solely on this aspect, but in general, Excel provides all of these features at a very competent level, and can really be your one-stop shop for your data platform.

Why Use Excel?

Why use Excel? Well Excel is an end-to-end data management system that runs on many platforms. That means that all you need is this one piece of software, and you can do everything end to end that you need to with a lot of your data. Now Excel does have some limitations, but over the years it has grown to be able to handle more and more data volumes in more and more complexity. One danger is that Excel is a desktop application primarily. So if you were to mess something up or lose data or your laptop were to die, all of those things would go away, and you wouldn't have the data backed up on the cloud, or any of those other features that the more robust systems give you. There are of course ways to mitigate that. There's Office 365, there are things like Dropbox where you can sync up to the cloud, or Microsoft has several options like SkyDrive. All of those things help you with that, but in the end, you're relying on your laptop or your desktop machine as the powerhouse that does all the data managing, so it can be cumbersome and it can be unwieldy when you get larger data volumes. But overall it is the most flexible thing and it helps everyone work with their data by their hands, really kind of get into the weeds there, and understand things, so it's really valuable in that regard. To diagram why you would use Excel a little bit, simply put, is you start out with raw data, you go through some processing, and you come out with insights. It's really that simple. You can do all of this with a single desktop application that is very inexpensive.

How It Works

How does Excel work? Well rather than get into the details of the software, I'm actually going to talk through a story, and we're going to jump over to Excel and do some demos, and I'm going to walk you through step by step on how to do many of these features I just described. Let's remember Cogsley Services. Cogsley Services is a consulting company that provides training, as well as some development work, and consulting for a lot of companies in the tech sector, as well as several others. They have been growing tremendously since 2008, and they have some data, some sales data in Excel, and we're going go through and analyze it now. First we're going to explore the data and we'll take a look at that. We're going to format some of the columns, and then we're going to use some of the basic functions. So if you haven't already, go install Microsoft Excel, and then we'll download some data and start working with it.

Demo: Formatting Columns

Okay let's take a look at formatting some columns now in Excel. From the demos that come with this module I'm going to open the CogsleyServices-SalesData-Start-Excel file. And with this, I'm presented with a pretty robust spreadsheet with lots of things going on. So first, let's just take a look around and see what we have here. So, on top, if you're not familiar with Excel that much, you have a ribbon. And the ribbon has all kinds of menus, specifically, there's a Data menu. And we'll be looking at some of these features a little bit later on. Right now we're actually just going to use some of the basic ones. I also have an ad in here for Tableau, you can go get this from the web, and this helps with massaging some of the data for Tableau. We'll take a look at that a little bit later on in this course in terms of visualizing data with Tableau. Right now though, my screen resolution is pretty small so I'm actually going to double-click on the ribbon and remove that, or minimize it. I can single-click, get it back to use it, and then it disappears again, or I can double-click and it comes back for good. So let's get rid of it for now, and let's just take a look at our data here. So I'm in a tab called Sales, and I have a lot of fields here, I have the Row ID, the Order ID, the Order Date, the Quantity of the sales, the Quote, a Discount, a Rate, a Sale Amount, and then, so this gives me all of the quantitative bits about the sale. Then I have the Customer Name, the Company Name, and some more demographic type information, the Location, some geographic info, the Region, when the project was completed, so we're a services business remember, so we offer up our consulting services and we have a Project Complete Date is really when we're able to recognize revenue or get paid. We have a Key which identifies the type of service that was delivered, and then we have the Consultant that actually delivered the service. So this is our main data set here, the Sales data set. And we have some other tabs down at the bottom. So we have Companies, which have all this other information from the NASDAQ so we can actually take a lookup who our customer is, find out what sector they're in, what industry they're in, what the IPOyear was. We have Managers, so we have this Region and Manager assignment so we can do things like look up sales by region. Then we also have our Consultants, which each get paid a different hourly wage. This can help us in figuring out the amount of profit, so what we're paying for the services versus what the sale amount was. And then we have the Pricebook, so this is our list of services, the category, and broken down by subcategory, and then the price per hour of each service. Okay, so here have an Order Date column that doesn't look quite right. As you can tell, it has a number instead of the actual date value. So what I'm going to do, I'm going to double-click and get my formatting menu back, and here I actually see that if I highlight the entire column by clicking on the column name C, I can go up to this number format and change this to a Short Date. This is also where I can do things like change things to Currency, Accounting, Number format, Time format, Percentage, etc., there's several more here. So for this I'm just going to change it to Short Date, and there you go. So I now have an actual order date that I can read. The Quote itself is a currency, and I have a little shortcut here, so I can just click on that, I get the currency. Since these are all whole numbers, I'm going to go ahead and remove the decimals by clicking on this little decimal guy here. Under Discount, this is a percentage, so I'll click on the percent. And the Rate and the Sales Amount, I'll highlight them both, I can click and drag across, or I can hold Shift on my keyboard, and I can click and select both of those. Then I can do the same thing, I can call those that. And then for the Rate however, they're whole numbers, so I'm just going to click on this, and get rid of the decimals. There you go. So now I've done some formatting there, pretty basic, right? I can kind of scroll through my data and see what I have. Some keyboard shortcuts that are useful, if you want to jump to the end of this column, I can hold Ctrl on my keyboard and press the down arrow, and I go to the very end. If there are gaps in my data, I would go to the last row that has data before that gap occurs. These are some real common things that you'll get familiar with as you browse around. So I'm just holding Ctrl right now using my arrows to jump around this data set so I can see the full corners of it. And then if I scroll down, I can see what's going on there. Good. Okay, let me keep going here. Now we have Zip Code. Zip Code is an interesting one, because Zip Code, while most times is just a regular number, sometimes like this one down here for New Jersey, it does have an actual 0 in front of it, and I get a little error that says, Number Stored as Text. Well, we want to keep it as text, because this is how we'll actually be able to map it, and see where this location is. So I'm going to click on the L for the column reference here, and I'm going to go back to my format, and I'm going to format the entire thing as text. There we go, so now there's no confusion, some of them are numbers and some are text, they're just all text. As I go along here, I see I have another date value, so I'm going to click on this guy and go back to my format, Short Date, there we go. And I've completed my formatting assignment. So that's good, that's step one, now I can actually start to read and understand what the data looks like.

Demo: Basic Functions

Now we're going to take a look at some basic functions. Excel as I mentioned is a pretty complete data platform, and as most data platforms go, you need to be able to do things like math, or text manipulation, or date calculations, all these kind of logical functions. So we'll take a look at some of the real basic ones here. And in the next two modules we'll get kind of deeper and deeper into that space. Back in the spreadsheet I've been working on, I have all of my columns here. And what I'm going to do, I'm actually going to first start off by using one of those functions on that data menu. So I'm going to go over it, and notice I have a Product Key. So this has the product category, a dash, and then the product subcategory. Well if I wanted to do some analysis on this data and see maybe sales by product category or product subcategory, this format isn't ideal. I need to split these into separate columns. So if I go up to my Data window here, you'll notice that there's a function called Text to Columns. Before I use Text to Columns, I'm going to insert some new columns here, so I can make room for this. So I'm going to right-click on my column Q and click Insert, and right-click on it again, and click Insert. So now, I'm going to go ahead and copy everything from column P over to column Q. So I'm going to hold Ctrl+Shift on my keyboard, and select it in the row 2, I'm going to hit the down arrow. So you remember that went all the way down to the bottom of this column, so now I've selected all the data in the column. I'm going to hit Ctrl+C, or I can right-click and Copy, you see I get the little dash line floating around it. And now, I'm just going to go over here and do something simple, I could hit Ctrl+V or I could just paste. I'm going to call this column Product Category. And then, I'm going to split these into two ones, so I'm going to call this one Product Sub Category. Okay, now what I'm going to do is I'm going to highlight all the data again in this column by holding Ctrl+Shift, and then hitting the down arrow. And then I'm going to click on my Text to Columns. Now this is one wizard, and it's a really useful one that allows us to split out columns based on a delimiter. So notice that I have options here, Delimited or Fixed width. Fixed width would be if I had data where at the tenth position in the cell it always meant it was a new column, I could choose that and just give it kind of a very fixed width in terms of how the columns are delimited. So now what I'm going to do, now here are the delimiter options that it comes with. Well, none of those are mine, I have a dash, so I'm going to go ahead and hit Other and just type a dash. And notice in my little Data preview what it did there. It actually went ahead and gave me an idea of what this is going to look like. I can scroll up and down, I can see that it looks pretty good. So I'm just going to click Next, now it's going to ask me if I wanted to format these in any specific way, so I can change them to Text or a Date or etc. In this case it doesn't matter, they're just going to be general. So I'm going to click Finish, and there you go, that easy. Now I have two separate columns that are split out for Product Category and Product Sub Category. Pretty cool, huh? Okay, now, there's some issues with this, it's not complete. Notice that, I have at the beginning of this an extra space. There's a little extra space because if you recall, I split on the dash, but there's also a space after that. So, I'm going to jump over to column T here, I'll expand column S and give myself a little bit of room to work. And in column T what I'm going to do is I'm going type a formula. Now this is our first function, so how do you enter a function in Excel? You start by entering an equal sign. So if I type = what's going to happen is I get this menu up here, which allows me to actually see functions. So this is the most important thing that I want to convey about this, besides these specific functions, is how to find functions that you need to use. So, there is a massive library of functions that you can use in Excel. This one, I happen to know, is called Trim, but here this little function finder is a way to actually figure that out. So, if you thought about it, if I wanted to trim extra space, let's see what happened here. It did a search and it said I recommend this TRIM function, it says removes all space from text string except for single spaces between words. So what that means is that it will get rid leading and trailing extra spaces. And that's exactly what I want. So I'm going to double-click on this guy, and then it's going to ask me, what do I want to trim? Well, luckily enough if I just want to continue using my mouse, I can just click on the field that I want to trim. There you go, and it even gives me a nice little example of what that means. I can also click for Help on this function and go out to the web and learn more, but I think that this is pretty much all I want. Now what I have there, is I have the column of Q duplicated into column T, and I've trimmed the space. Notice that the field here, while I'm seeing the word Development, it's actually a formula. You'll notice that, a lot of times, you'll see something, but the actual value is coming as the result of a formula. So I'm going to go ahead and copy this, and just paste it right next to it. And look at what happened now. It didn't give me the same one, it gave me Big Data. Well, if you noticed, when I switch over here, it's looking at column Q. Well, when I copied and pasted that over to the right, it actually gave me a column R, so it is a relative reference as we would say. So if I copy this formula all the way around anywhere, it's going to look back a couple columns and give me the relative reference, the trimmed value of that field over a couple columns to the left. So I could keep pasting this and pasting it and trim all the stuff if I wanted to. These are the only two I really care about right now, though. To complete the picture here, if I wanted to duplicate this exact formula and I didn't want to have the relative reference move as I pasted it, I would have to put a $ in front of the bit that I wanted. This is maybe a bit of an advanced concept, but it's one that I think you should be able to grasp pretty easily. So if I were to put a $ in front of Q, that's my column reference here, Q, what that means is that now as I paste this field, I'm going to copy, and paste it over where it says Big Data. It's going to keep looking at column Q, it's not going to move with the relative reference. Notice when I pasted that there, it still has column Q in it. Now see, that's one of the interesting things about this. So, you have two types of things, you have a relative reference, where when you paste the formula around the reference cells that it's using in its formula are moving with it, and then you have what this is called an absolute reference. An absolute reference is where it always sticks to the exact same column. So I could undo or I could try to figure that out and redo the steps I took, or I could just go manually edit this, type in R2, and give me the field I want. So this is very flexible. Think of all the types of things and manipulations that you may want to do in data, and how flexible and dynamic this type of function can be. So I'm going to go ahead and double-click on the bottom corner here. What that's going to do is that's going to copy this formula all the way down as long as I have a value in this column S. So it's going to copy it down using column S as kind of a guidepost of how far down to go. And luckily I know that column S is populated for the entire data set, so if I were to go to the very bottom of the data set, I can see that column S was populated there, and when I double-clicked that, it copied this all the way down as well. So all I wanted to do here was to trim those spaces and then go replace columns Q and R. So I'm going to go ahead and copy this here, I'm going to highlight columns T and U, and hold Ctrl+Shift on my keyboard, press the down arrow, and I get the entire data set. I'm going to right-click and Copy this. I'm going to go all the way back up to the top, make sure I'm on the very top one there, I'm going to right-click, and this time I'm not going to Paste, because Paste would actually paste the formula. I'm going to do something called Paste Values. So Paste Values is a way to insert the result of the formula, instead of the formula itself when you do a Paste. So that's what I wanted, now I've cleaned up my columns, I'm going to go ahead and delete these guys, they're no longer needed. And, there we go. Now there's a few more things here, I need to do dates to complete. So I'm going to go, we have an Order Date, and then we have a Project Complete Date. And with the Project Complete Date, what I want to know is how long did it take to complete this project. I'm going to go ahead and insert a column here, and say Days To Complete. I could also call this Duration maybe, I could do lots of other things. So this is a simple formula, this is an arithmetic one, and luckily Excel makes date arithmetic very easy. I'm going to start as I always do by typing =, and I essentially just want to see the difference between these days. So I'm going to use my arrows on my keyboard to go over to the Order Date, and I've selected that. And then I'm going to type a -, because I'm going to subtract the Project Completion Date. It looks like I've got a number formatting issue here, so no worries, I know how to do that. I'm going to go back and format this as a number, and it looks like -7 days. So something might be off with our data quality here. Or, we simply just did our subtraction wrong. So let's take this and try it again, and this time I'm just going manually type it in. I know it's column O and I know it's column C. So I'm just going to type in O2-C2, and hit Enter. This is a number, and first let's format this entire column as a number. And then once it's all formatted as a number, we can change it down to not having decimals. Okay great. So, as I could do before, I could double-click on the corner here, and copy this all the way down. Or, let me show a nifty other little trick where I'll use my Ctrl button and hit the down arrow, I jump to the bottom of the data set. Now I'm going to select all these blank rows, I'm going to hold Ctrl+Shift, hit the up arrow, and I selected up until I found a value. And now I'm going to press Ctrl+D on my computer. And Ctrl+D, think of it as duplicate. I just duplicated that formula all the way down, so you can see it copied it all the way down. And just like before, we have the relative references, so there's no dollar signs here, which means if we copied this around we would actually, possibly have the wrong type of calculations. So we could keep this locked in if we want, but for now let's just leave it here and move on.

Where to Find More

Okay, now let's take a look where we can find more info about Excel. So Excel obviously has an incredibly large community online, but I'll point you to some really interesting references here. One is the Smarterer test for Excel. So if you go out to that URL, you'll see here the Smarterer tests, and you can start. We have 88, 000 users that have taken this, and this is a really excellent way that you can assess your relative skill. It takes a second to sign up, it's free, we don't share your information or anything like that, and then you go take your test, and you can see exactly where you sit relative to the others. The next place I'll point you is to our own library. If you just go search for Excel, you'll see that we have 82 results, so we have lots of fundamentals courses and you can see the length of these are pretty lengthy, 5 hours plus. There's some little shorter ones that are more specific, and then there's some really in-depth ones. One I like is Andrew Brust's course, the Analytic Superhub for Excel. It's really a good way to think about Excel as kind of a real strong data platform. There's also a bunch of others there that have similar concepts, so I encourage you if you really want to get deep into Excel to go take a look at the library. And the last one I'll point you to is one of my other courses which is the Business Dashboard Fundamentals course. This course I go into depth about the types of ways you should visualize your data, the types of charts you should choose, and even building them out in Excel, pretty robust level of detail there, and then also building them out in Tableau. So it's a really great way if you're familiar with Excel and you want to get into to Tableau, or you do dashboarding and some other tool and you want to learn some conceptual design, practices, and things about the right way to use color and all that, that's a really good course for that type of content as well.

Massaging Data in Excel

Introduction

Continuing on our journey here to become awesome data analysts, we're going to go through this module now on Massaging Data in Excel. In this module we have some goals. First, we're going to talk about what massaging data is, why we would massage data to begin with, how to do it, we'll go through some functional examples, and lastly, we'll figure out where to find more info if you want to get deeper into this category.

What Is Massaging Data?

So what is massaging data? Well, massaging data, or transforming it is the process of cleaning, normalizing, and preparing the data for analysis. You see, a lot of data is very messy, and most systems aren't designed to be there for analytics, meaning that the data is meant for the processing of it, point of sale systems, or email marketing campaigns, and tracking things online. The things aren't designed for us to have this beautifully, nice, clean data, and they often allow a lot of free form input. So, the data coming in can often be quite messy. And this is the process of taking that messy data and making it cleaner and neater for analysis.

Why Massage Data?

So why would we want to massage data? Well simply put, this enables more accurate and faster methods of analysis. If every time you try to run a query, or to drag something onto a pivot table, and look at the data, you get real messy results. It's going to take a lot more time to clean it up after the fact. So we spend a little bit of time up front cleaning and preparing everything, just like we would if we were chopping vegetables for a meal, and then we go and it's easier and faster to get the answers and the insights that we want. Remember our goal is to take raw data and turn that into insights. And massaging of the data is really the first step in this process.

How It Works

So how does it work? How does massaging data happen? Well the best way to do this is to work through some examples. So let's talk about that. Remember our company Cogsley Services. Well they have some key questions that they're trying to answer. They have four basic needs, and we'll go through these. So, the first one is to look at sales by industry or sector. We know the companies and who the companies are, but we don't know the other bits about the company, so what industry are they in, what sector is that, all that kind of stuff. The next thing we'll look at is how many days it took to complete a project grouped by manager, some analysis, figuring out are there trends, or are there patterns, are some people better than others at certain types of projects? Next thing, we'll look at hourly wage by bill rate, so tell us some information about how much we're paying or what the cost is for the projects we're delivering. And lastly, we'll some actual cleanup work and just remove some duplicates. So these are the things that we're trying to solve. We want to make sure that this is all there. So in the process of massaging data, we just have to keep these in mind, and we're going to through some functional examples now.

Demo: Vlookup

Let's take a look at how to do a VLOOKUP. Now VLOOKUP is about the most common or most popular function that you're going to use in Excel when you're massaging data. It's going to allow us to bring two different data sets together. So I'm going to switch over and I'm going to open our Excel file that we just finished up with the last module, and we'll start adding some fields in there. Okay so here I have my file CogsleyServices-SalesData-Start from the demos in this module. And if you recall from the last module, we added a lot things, we formatted, we created some calculations. Here what we're going to do, we're going to add that industry and sector info. So if you recall, we have Company Name here in column J. And the question we're looking for is the sector or industry of the company. We can guess some of these, but we probably are going to have thousands and thousands of customers, if we're successful, so we need a way to programmatically add that bit of information. So this is the transformation, or the massaging of the data. I'm going to insert a couple of columns here, just to the right. What this is going to do, is just give me some space to add in my new fields. So, first one is going to be Industry, and the next one is going to be Sector. Okay, pretty simple here. So I'm going to do our VLOOKUP. Now, the VLOOKUP, I just start typing, and you can see it pops up, I can hit Tab, and it gives me the nice little formula here, lookup\_value, table\_array, col\_index. So the first question is, what am I trying to look up? Well, I want to look up the company. So I'm going to click on my cell that has the value for company name in it. Then I'm going to hit comma, and it's going to tell me where do I want to look up, what table or array am I going to use to actually get the data back? Well we have our separate tab here for Companies. So I'm going to click on that, and notice what it did, it instantly added the name of the sheet with an!. Now this is just its syntax on how to reference a different sheet. Don't worry about it too much, we'll figure it out in a second here. Now what we have though is the name of the company which is column A. So I'm going to click on column A, and I'm building out my formula, and as I click and drag over, you can see that I was able to highlight this entire area over to column G. Now, there's a little trick here where I'm going to lock this in place, and I'll explain it in a second, but I'm going to hit F4 on my keyboard, and that adds a $ in front of the A and in front of the G. I'll explain what that's for in just a second. Okay, now that we've got our reference for our table, I'm going to hit a comma, it's going to ask me which column do I want to return? So if I'm starting with column A and this is 1, E would be 5. So I'm looking for Sector right now, and I'll hit 6. Then it's going to ask me, do I want an Approximate match or an Exact match? I want an Exact match, and so I'll either enter FALSE, I could use my arrows and select one, or I'm kind of old school, so I just hit 0, 0 and 1 are basically the same ideas. Okay, and it looks like it returned something, that's awesome. One of the problems is, as you noticed, I have Industry and Sector mixed, I'm pulling back Sector here versus that. So if I wanted to change that, I'll simply go in here and pull back column 7, which is the next one over. There you go, so now I have my actual industry. Now just to validate, I'll go over to my Companies tab, I'll go here and find CA Inc., found him on row 422 there, and you can see that the sector is Technology, and the Industry is Computer Software: Prepackaged Software. So, it worked. So what it did, is it actually went over and it took this value, CA Inc., it looked up, using my VLOOKUP formula, found the reference that I wanted, and returned the seventh column over. Now, real quickly, I want to basically do the same thing for Sector. Now, let me show you what would have happened if we didn't lock our actual range, and I didn't have those dollar signs there. Well if I were to just copy this and paste it over here, you'd see it gave me a total messed up error there, an N/A error. Well, as you can see what's happening when I click on this, it's actually looking no longer at column J, it's looking at column K for the first one, which is not the right one, and it also moved from A through G, it's B though H in my lookup. So, the way we prevent that are we create, what you call, absolute references. So if we put a $ in front of the J, and then I'll copy this over again, you can see that between the two, the J stayed consistent. Now if I paste it down below, you can see that the number changed. So, the locking is relative to whatever column or row is referenced just after the $. So if we want to keep something consistent, meaning as we copy it from left to right, we still want it to look at column J, we just put the $ in front of the J. We do want it to change as we go down so you can look up the appropriate company. Now in our range however, this columns A through G, we don't want that to change at all, that's our entire table, so I'll just hit that, I'll use F4, or I could manually type them in, and I will just copy and paste, that still seems to work, now copy and paste, and copy and paste. So you can see what's happening here is that it's returning the exact same thing, because the formula didn't really change. So if Sector, if you recall, was in column 6, so I'll just go in and change that to 6, and copy this guy down, and you can see there, that's how that works. To wrap this out, let's just copy this down all the way to the bottom of our spreadsheet. Take a look, I'm using my Ctrl and up and down arrows to jump around, that's a real quick way to navigate. And I'll use my filter here to see if there's any issues. So as I scroll down, it looks like everything was found, I don't see and #N/A or #Error, or any kind of Excel errors. That means that all of these companies were found in my list, which is awesome. If they weren't, I would need to go back and find those companies, add them there, and then it would populate. So it looks good. So now I can do analysis based on industry and sector. Let's go over now, and we're going to add a couple more of these. We're going add Manager. So if I have my Consultant here, and if I go to our Managers tab it's based on Region, so I'm going to do a look up on Region, and find out who the manager is. And I'm going to call this Manager. This time, I'm going to do my VLOOKUP, just like before, some more practice for you. And I'm just going to use my arrows to go over and find my region, there it is, hit comma, go to my Managers, columns A and B, pretty simple. I'll use F4 to lock that, and I'm going to return column 2, and I want an Exact match. There we go, so now I've got the manager, copy that down. Now the next question was about hourly wage. So, let's go look that up, and we're going to look it up by consultant. So, I'll just add a new column here, Hourly Wage, and we're going to do VLOOKUP, just like before, go find our consultant, hit comma, go over to Hourly Wages, Consultant, and Hourly Wage, just need those two columns, I'll lock it in place, I'll return the second one, and I'll get an Exact match. Notice these were a bit easier. In the next demo I'm going to show you how to make it a bit more complex to where it is more resilient, but for now, we can keep this pretty simple. So let's just scroll down and see if we have any errors, doesn't look like it. Okay, perfect. So always double check that you don't have any errors in there, that the formula is actually working, it's also sometimes good just to double check, spot check a few of them, because sometimes you may be pulling in the wrong column, but it appears to be the right one, and you just want to make sure that everything's good there.

Demo: Vlookup & Match

Now let's take a look at how to make our VLOOKUP formula a little bit more resilient. I'm going to jump back over to my spreadsheet, and in here, you remember, I added Industry and Sector, and when I first did that, I actually punched in the wrong number for the column to return. The way this works is, I tell it, hey return column 7 of my table or array that I put here, which is in the Companies sheet, columns A through G. Well, if I wanted to make this more dynamic, I can actually tell the VLOOKUP, hey, go take a look at the column header, in this case, Industry, and then go look and figure out which column that is in my table or array. And this works using a formula called match. So I'm going insert a column here, just to show you how the match formula works, and then I'll basically bring it into the other one, building what's known as a compound formula. I'm going to type match, and it's going to be pretty similar to the VLOOKUP. It wants a lookup value, in this case, I'm going to give it the column header, and then it says, okay where do you want me to look that up, and I'm going over to go over to my Companies tab. And if you recall, we were in A over to G. Now I'm just going to select the first row, because the way this works, it doesn't do the whole scan of all of the table. It only looks at one specific array and says, okay, which column, you know, do I want to return? What number column do I actually have here? I'll give it a comma, and say I want an Exact match, and close it out. And you can see what it did, it gave me a 7, and in my VLOOKUP, that's exactly what I want in there. So I want this formula right where that 7 is. Now let's make this sure that it doesn't break when I copy it down. So first off, I want it to change as I copy it from column to column, but it always needs to look at the first row. So I'm just going to add that there. And then just like the other one, the array itself of where it looks up needs to remain static. So I'm going to make sure that both of these guys are completely locked in place. Okay, perfect. Now it's pretty simple, I'm just actually going to go and copy this entire formula, without the =, and then I'm going to go over into column K here, where my VLOOKUP is, and where the 7 is, I'm going to go ahead and just replace that, delete that, and paste in my other formula. Notice now it highlighted the column header because it's adding that as a point of reference. When I copied it in, look at that, it still works. So if I have just to copy this down, and I'm using Ctrl+D to duplicate the formula to the cells from the very top one down to the bottom ones, you can see that it worked. You can see where the formula changes as I scroll, but everything there is still functioning properly. Now if I delete this column, what I should be able to do now is just copy this entire formula over, and it'll automatically recognize that it's looking for Sector, no longer Industry. So as I copy this over, it shouldn't change anything, and there you go, it came back with the exact same result as it did before. So, I'm going to highlight both of these, click on the bottom right corner, and it will copy it down to the rest of my sheet. And everything looks good. It looks like it's still working, I don't have any errors, all is good. Now, just to test it out, let's see what'd happen if I tried to change this to Sector. You can see the entire column changed. And if I wanted this one to be Industry. You can see how quick and easy that is. So we're making very resilient spreadsheets, data sources here. If I wanted to add a new column, you can see there's a lot more data here, the Symbol, the IPOyear, the MarketCap, all of that, I could actually just add that column, copy this formula over, and make sure the other column header matched, and it would automatically populate it for me, I wouldn't have to go through that same exercise. So I'm saving myself a few clicks, and I'm also making it a bit more dynamic. So, as the needs or the questions of the data change, it doesn't take nearly as long to rehash and re-massage all the data in all the formulas that I've already built.

Demo: Removing Duplicates

Alright, now we're going to take a look at how to remove duplicates. Data comes in very messy a lot of times, so it's always good to double check and make sure that you don't have duplicates, and you're not going to end up analyzing or reporting false numbers. Let's jump back over to the workbook we've been working on here. And remember that, we have a column for Row ID and a column for Order ID, Order Date. So we can do some double checking here. One thing I like to do, over at the far right, or just at the end of my data set, is add something called a row count. RowCount, give it one word, and just add a 1, and then we can actually count things up. And I'll duplicate that all the way down, so it's just a 1 for every row, okay. I'm going to go ahead and do a pivot table now by clicking on Insert and Pivot Table. It'll automatically select the range. I hit OK, now I can actually drag Row ID down onto ROWS, and I can take my RowCount and add that, and see how many times the Row ID appears. Okay, so it looks like sum, we've just got 1, let's scroll all the way down, okay, this looks pretty good. So, oh, towards the bottom it looks like we actually have some duplicates, look at that. So these Row IDs, 8378, 79, appear 2 or sometimes 3 times, so that's an issue. So, this basic first step tells us we do have duplicates. Alright. So, let's go back to our sheet, and let's figure out how we can identify them. Obviously, we have them here, so we could just do a lookup on that, and we know how to do VLOOKUP, so let's try that first. And then what we'll do is, I'll show you a more sophisticated way of how to do it. So, I'm going to insert a column on the very far left, and I normally don't do that, because what happens is, a lot of times a formula or something may reference one of these columns, so I don't want to mess with it, but right now we're cleaning it, so we're not quite ready to analyze, some type of column here called Dupe? And I can do a VLOOKUP, just like I did recently, and I'm going to look up the Row ID. I'm going to click on my sheet here. Now I'm going to select columns A and B. It's important that I select the column and not one of the values because what happens is, if I click inside of the pivot table, a lot of times I'll get, Get Pivot Data Formula, and that's not what I want here. So what I want to do is I want to make sure that I click on the column all the way at the top, and then I hold Shift, and I click on the second column. So just like before, I'm going to lock that in place, and I will return the second one with an exact match. So this is telling me how many times this row shows up. So I have that, so that's one piece, it doesn't quite yet tell me if it's a dupe, because I basically want it to say yes or no. Okay, so now I'm going to have a formula around this. So, VLOOKUP is the first one, the next one I want to do is, very simply, is this greater than 1, FALSE. So if it were greater than 1 it would say TRUE, and that would tell me that my duplicate was there. So I'll copy this guy down, and I'll go all the way to the bottom, and you can see these ones down here at the bottom are all TRUE, or several of them are. So, very simply, I've been able to identify where my duplicates are. So step one was, build a pivot table, make sure that to see if I have duplicates, and if I don't, then I'm good, but here I do, so let's go identify them. Step one, we've been able to identify them. Now, this hasn't quite helped us in actually eliminating them. All we know now is that there is a duplicate. Well, one thing that we can do now is we need to identify the first one as not a duplicate, and then all the subsequent ones as duplicates. So what that'll do is, that will allow us to filter, and then delete all of our duplicate data. So I'm going to insert another column here, just by selecting in column A, and saying Insert. And, I'm going to say, Remove? So this will tell me if I should remove the column, so it's going to be a bit more complex of a formula, but I'll take it slow and walk you through it so you can understand, because this is going to be something you're going to want to do a lot of times, over and over again. So there's a formula called COUNTIF, and you can see what it does, it counts the number of cells within a range that meet a given condition. So what do you think we're going to do there? We're going to say, COUNTIF the range, in this case it's going to be column C, equals the value in the cell C2. And just like the absolute versus relative references as we copy this down, it's going to move that cell reference down. So I hit Enter, and it says, this cell, C2, appears once in the column C, meaning it's not a duplicate. We have a double check there that it says FALSE, so that is a good indicator. Now, if I were to copy this down, it essentially would still do the same thing, it would just identify which ones have more and more duplicates, which ones are duplicates. So, I'll do that now, just to see how it works and validate that our formula is working. You can see this one takes a little bit more time to calculate. And down here you can see all these guys that are the duplicates, and of course they say TRUE next to it because they're looking up on our pivot table, and you can see how many times they appear. Okay, so how are we going to get this to tell us which one to remove versus not? Well, what I'm going to do, is I'm going to ask this formula to tell me how do I know which one to remove? And so the way I'm going to do that is still using COUNTIF, but in a little bit of a different manner. So, as you recall, the relative reference, as I copy that down, it's going to move the cell range that I'm looking at. So I'm going to start out with COUNTIF again. And this time, I'm going to scroll over, and I'm going to select from here, up. Then I'm going to hit comma, and tell it the criteria is column C2. So this will tell me how many times does this number appear from the cell I'm on all the way to the top. Now, you're probably guessing if I were to copy this down, that it would only look at the top two cells above it, you are correct. This is where we need to actually add in the locking of the top cell, so while the second part of it copies all the way down, the first one stays consistent. Okay, so let's lock the top cell here, using our F4 shortcut, just like we were. And then, we'll leave it as C2 for the second part of that table or array because as we copy it down, you'll see what's going to happen. I'll just copy this down a few times and you can see how it changes now. So you can see what it's doing here is it's looking at from C1 to C3, how many times does C3 appear? So as we copy this down to the duplicates, what's going to happen is, you're going to see it go one, two, three, four, five, or up to as many times as it appears. So with that, I'll just copy this all the way down, Ctrl+Shift, and then Ctrl+D, and that stands for duplicate, so it's going to duplicate the formula all the way down. And so now as I go down towards the bottom here, let's go ahead and sort our columns in order, or sort Row ID, so that way we can see them and how they're happening. So we're going to sort smallest to largest. And so they're all in order now, and so what we're going to do is just go down, I'll jump all the way down to the bottom because I know there's some duplicates there, and you can see what happens here. So, where the duplicate is you'll see it's a 1 and then 2 right below it. So you can see that, you can see that we know this one is a duplicate because it's looking up off of our pivot table, and then it's counting the first one as it appears one time from here and above, and the next one, from two times and above. So what that's saying is, we can actually just filter on this Remove? column, or change the formula to say remove yes or no, because if it equals anything more than one, then we should remove it. So let's go back up and make that adjustment. So now by adding the greater than 1 to our formula here, when we hit Enter it's going to tell us if this is the first occurrence, and if so that's okay, otherwise if it's the second or third occurrence, then we need to remove it. So if I jump all the way down to the bottom, you can see where we have some cases here. In this case, the Row ID 40 is being repeated, and so the first one is okay, but the second one needs to be deleted. So what I'm going to do now, is I'm going to add the filter to this, that way we can actually just find those, and then remove them. So I'm going to click over on Sort & Filter, say Filter, that'll remove the previous ones, click it again, and it adds it to all of the rows. Now I can just go here and say TRUE. And there you go. I'll highlight all these rows here, and I will right-click, and Delete Row. Okay, now when I go ahead and clear my filter, (Working), you can see where there are still ones that were showing up from the previous table that were duplicates, but now they're only showing up once in our list. So we have now an actual clean list, and if we wanted to check it looks like the Row ID, which is just incremented for each row, matches the row count with the addition of the header row. So, we're good, we've been able to successfully remove duplicates. Now that probably took about five too many steps in here, but I wanted to explain all the different things happening so as you go through this, you can understand maybe shortcuts, or different ways that are more effective to you to give you confidence that you've transformed the data into a clean data set that you can then analyze.

Where to Find More

Okay, and to wrap up this module here, I'll just talk about some additional resources. Similar to the last one, we have a test on Smarterer for Excel, it's a great place to go and try to assess your level of skill with everyone else's. We also have tons of courses on our library that have Excel, and one thing I would recommend there, take the Fundamentals course, there's several different versions, and we're always on top of this one. Excel is a wildly popular subject, so we'll always have new content coming out, that's why I didn't put a link to a specific course there. And I actually have another course called Business Dashboard Fundamentals, and this one actually goes through a lot of Excel stuff and how to build dashboards and understand things from a visual stance, conceptually, and then practically how to build them out in Tableau and in Excel.

Analyzing Data in Excel

Introduction

Hello, and welcome to this module on Analyzing Data in Excel. This is our fourth module in the course on Data Analytics: Hands On. To this point, what we've done is collected and organized some data. We looked how to massage it, and now it's time to actually bring home the bacon here and bring some visualizations and some insight into what the data is going to try to tell us. In this module we're going to understand what data analysis is, why we would analyze data, how we can do it, and we'll get into a lot of demos in that section, and then we'll learn about where to find more. As always, this is a huge topic and way too much to go into depth on every little nook and cranny of it just for this module, so I'll point you in the right direction in case this is really an area you want to dive deep into.

What Is Analysis?

So what is analysis? Well, I think of analysis as this process of iterating through these questions until you can uncover some insights from the data. There's a process here that is the analytic cycle, and it starts with you, the analyst. And generally the first step is you'll have a question, and you'll want to get some data to help answer that. Let's look at the drought going on in the west coast of the United States right now. If I wanted to understand how severe it is, I might want to go look on websites like data. gov or the Census Bureau, some of these other ones, and find some data to help me understand really what's happening. And then I would like to explore that data. So maybe I want to churn through it, I want to do some summary statistics, I want to see what kind of anomalies there are to see what options I have, what columns and fields, what types of dimensions, what types of things can I learn from it. Then I'll go through this process of visualizing the data. We're humans, and humans are visual creatures, so the best way for us to learn or really consume information is visually. It's how we as a species have evolved over thousands of years, how we've understood the stars in our patterns across the sea and all of these things. It's all visually. It's all recognizing patterns in visual displays. And that doesn't change. We're still humans. We still have that innate ability to discern stuff visually versus all the other forms, so auditory or sensory or whatever. Then we want to test it. So let's say we come up with some visualization and it makes sense with one cut of the data or one perspective. We want to test out a few different scenarios and see if it really rings true. That way we can kind of weed out any false positives. Sometimes when you're looking at data you may come to a conclusion super quick, and it may be a sensational claim, but when you really try to look at it from different angles and see if your theory or hypothesis tests well with other scenarios that's when you can uncover some flaws in it, so we really need to do a rigorous process of making sure whatever conclusions we think we have are valid by testing those. And after that we study it. This is the act where we really kind of let all the things we've learned throughout the process absorb, and we gain a sense of knowledge. Over time, understanding the similar topics, we have this ability to predict things and have sort of an intuition about how things are going to go. Now it's good to really build that up over time on a solid foundation of understanding and learning through data, but it's also good to reevaluate so we never get assumptive in ways that actually lead to disasters. So this is the whole process of analyzing data in general. Now the thing about it is that you can go from one step to the other at any moment. We could look at some data and quickly visualize it and come up with an answer that we think we have, but before we test it and study it we have to learn we have to get more data, or in the act of exploring we have to go back and acquire more, or we visualize and then we have to go back and explore, so any of these angles or any of these different paths here could intersect at any moment in time leading us towards that ultimate goal of gaining an understanding based on solid information.

Why Analyze Data?

Then that leads us to why should we analyze data? Why not just rely solely on intuition? Why not just give it that college try or rely on your gut instinct? Well, in Excel, specifically, it provides this really incredible capability for us to use the data without any coding necessary, and it's really economical. I mean it's really affordable here considering the power that you get with Excel itself. Let's take a look at some analytical tools out there and some options. Now some of these vary in lots of different categories, and this is my survey of my experience using these different ones. But if you go look online at any of those research facilities you'll find that there are similar comparisons there. So when I look at Excel I think that the big thing here is its ease of use. It is relatively ubiquitous in the business world, and all over the world it is used for major decision making to the most basic of household budgets for our family. Then we have another popular one in the Microsoft area called SQL Server. It's a database, and we'll talk about databases a little bit later. And BI stands for business intelligence, so business intelligence is kind of the evolution of this idea of data analytics or taking data and making sense of it. Now the big thing there I think is its power to model the data and to come up with these complex structures that allow all kinds of flexibility and stuff. In the other realms it's not so strong, but it definitely has its place. Now Tableau we'll get into later. Tableau's main bread and butter is this charting area where you can visualize the data and explore it very naturally and intuitively. And in all the other areas it's fairly strong as well. It doesn't do so hot in the statistical modeling, but overall it's one of the most powerful ones. And if you look at any studies online or read testimonials you'll understand why. Then we also have R. R is a statistical modeling tool. It's an open source one. It is widely used across the globe by all the major companies that you're aware of. So anybody that does real hard-core statistical work usually is using R or a couple other's packages. There are very few that really dominate that market. And this one, of course, it's biggest area is statistical modeling. You can run all kinds of simulations and do all kinds of really, really advanced analysis in there. It also has some cool things with charting in other ways, but its bread and butter is that statistical side. Now one other I'll mention is MicroStrategy. This is probably one of the biggest players in the business intelligence space or been really popular for a long, long time. And they actually, I believe their value is in the data modeling, as well as mobile, which I didn't really have on here. But, you know, they're another really awesome business intelligence or data analytics platform. The challenge I've had with them in the past is that it's kind of all or nothing. You have to go all in with them and use every bit of the platform whereas some of these other ones are a bit more piecemeal. So Excel and Tableau play really well together, Tableau and R play really well together, SQL Server and Tableau play really well together, SQL Server and Excel, etc. So a lot of these other ones know that they are really good in one area and not so good in the others, so rather than try to do everything they integrate with the other tools whereas MicroStrategy is one of those that doesn't. In my experience I haven't seen it really integrate well with any other tools. It's kind of like you do everything in it or you don't use it. So overall this is kind of where I rate Excel. It is the most general purpose tool out there, so it's one of those things where it's really the base component. So if you want to get into data analytics, this is my opinion that you should start with Excel, really feel what it's really like to get your hands on the data and how to work with it by hand. Think of it almost like gardening, really getting your hands in the soil there to work with the plants whereas some of these other ones are much more advanced, but the learning curve is going to be a lot higher.

How It Works

Now let's take a look at how this actually works. So we're going to go through some demos here, and we're going to build on what we've been doing to date, and we're going to use that Cogsley Services Inc. file from the demos. So we're going to try and answer some questions for them. Namely we're going to answer this sales by industry. We want to analyze that. We want to look at the average days to complete by manager. If you recall, what we are doing is looking at the projects and how long they take. Then we're going to look at the hourly wage to bill rate ratio. So are we billing properly in certain areas or should we assign people and shift them around so we can optimize our profit ratio there?

Demo: Pivot Tables

Okay, now we're going to start by using a pivot table, which I consider the base unit of measure for analysis in Excel. This is going to give us the ability to do summary level statistics, create hierarchies, and see at a high level just some summary statistics about our data. So first let's take a look at our sales data set that we're going to start with. It's similar to the one we've been working on to date, and you can download it from the demos of this module. So here I am in the starting analysis, and you can see I have my several tabs here. I have my Sales, and then I have all my look up ones, the Companies, Managers, HourlyWages, and Pricebook. And in here what I have is every order and the order date. I've created a calculation to format it into a month and year. I've got the Quantity, the Quote, the Discount, Sales Amount, customer information, location information, some key pieces here, when the project was completed, as well as when it was started, the Sub Category, the Consultant, and then I get all the way over to the Wage Margin, the percent margin of hourly wage. If you recall, that's one of the questions we want to answer for Cogsley Services. Okay, so to do a pivot table we first need to select our data, our table of data if you will. So here I am in a tab that has all the data I'm interested in analyzing, so I'm going to go up to INSERT and choose Pivot Table. Now before I go on, this highlight, this ribbon here has a lot of the things that you're going to want to do when it comes to analyzing data in Excel. We have a Recommended Pivot Tables. It'll try to guess some ideas based on the data you have. You also have the ability to do Recommended Charts, pick a specific type of chart or do a PivotChart, which we'll do in a minute here. You can do maps. Power View is this add-on to Excel that'll allow you to analyze the data in kind of a more interactive way. We won't get into that here. There are other courses on that already in the library, and that itself is a whole course in terms of how complex and deep it actually gets. And then you have these spark line ideas. These are like those little mini charts you can put into a scorecard or just right in line with your data to show a mini trend. Then you have slicers and timelines. These are the things we'll use in a second to get interactive with our data. And then of course you can just have more basic stuff like a hyperlink or other text and symbols. So once we have our data here now we could select it all, but I know that if I just click on this Excel is going to select the entire range that I'm looking at, so it'll go all the way over to the right all the way down based on the cell I'm in, and that'll give us everything we need here so it's selecting the table or range. I could also name this if I wanted to. There are lots of different ways to treat your data in terms of a management of it. This being kind of a light level course and intro we're not going to get into that, but just know that this act in itself can be pretty in-depth. And we're going to do in a new worksheet. We could put it in an existing worksheet at a specific location, but we're just going to dump this into a new one. I'll click OK. And here I have a new worksheet, and in this worksheet I have basically a blank canvas to analyze my data. For this one, what we're going to do is we're going to look at sales, and we're going to look at sector. So the way these work, and there are lots of different things that you can do here, you can drag a field or you can just check the box, and it'll guess where to put it. I like to drag it. So when I drag this down to rows, see over in my pivot table here what it did is it gave me all of the unique different sectors we have, so it deduplicates a lot, which another way you can use a pivot table. And then I have my sales which I'm going to add here, Sales Amount, over to my VALUES. It's not quite formatted right, so I'm going to click on the VALUES, choose Value Field Settings, it's going to Sum, that's correct, but the number format I want is Currency. Let me just click on that and click OK, and now I have something that looks more like the numbers I want. Now of course I may not want decimals here or I may want to roll this up to 1000s, so I'm going to go back there and do this again, choose Number Format. So I just got rid of my decimals there and hit OK. So now I have a bit more readable of a number. And this is one basic way I could say that I've answered that question about sector. I also could create a hierarchy by dragging Industry below my Sector, and I could switch these around. And what I can do here now is see how it added this minus sign and it expanded all of them? Well, when I'm looking at this what I can do is I can actually group these together. And if I go up in my window here under the Active Field for my ANALYZE tab of my ribbon, I can collapse all of them. So now what I do is I have basically that same answer that I was looking for, the sales by sector, and now I can actually drill down. And let's say I was interested in Technology because that looks like a big one, I can dig in there and actually see exactly where those sales are being distributed across that sector, so kind of a cool way to show stuff at a high level with the ability to drill down. This is where I could do some of the basic stuff. I could do like a highlighting type thing, conditional formatting. Here's a Color Scale or an Icon Set. I can do these kinds of things. Data Bars, I kind of like these ones. You can see what it does there. It kind of puts a bar chart inside of the data itself so you can see a visual indicator of how relatively different they are. You can also play with the Color Scales, that kind of a thing, different ways of looking at it. And then of course there are Icon Sets. Some of these make sense, some of them don't, so play with this and look at all the different options you have. I think it's kind of a cool way to just add some visual queues to it without making it too messy or fussy on the distraction. Okay, so that's the first pivot we're going to build, and we're going to call this Sales Pivot. Now the second question we wanted to answer was the days to complete, and we're going to look at this by manager. So I'm going to go back to my data set here. I'll jump all the way back up to the top. I'm going to choose INSERT, Pivot Table, same data set. Now what I want this time is Manager in my rows, and I want my average days to complete, so I'm going to take Days to Complete. And when I drag it on there notice it's a sum meaning it's going to add all of them together. Well, that's not what I want because some manager may have had more projects than another one, so it's not a fair measure. Instead, I'm going to change my Value Field Settings, and in here I have the ability to look at a different type of aggregation. They say Summarize Values By. The way most people in the data analytics world look at it is what type of aggregation is this. So I'll go ahead and click that. Now I get a much more reasonable number that's better analysis measure, and of course I'll go ahead and change this. I'm going to choose Number, and let's just do two decimals. We don't need a bunch of more decimals in that. Okay, so we have something that's a bit easier to understand by manager. How long does it take to complete a project? It looks like they're all pretty quick. Now let's take a look just for a sense of reference about how many projects each one of them has completed. So we have a field that I added there called RowCount, so this is just a one for every row that's in the data set. And so what that's going to do is it's going to let us list out just how many they have. Again, I could do stuff like add those conditional formatting pieces to it or any of the other types of indicators I want, but here I'm going to go ahead and leave it alone just like that. Okay, and this one we're going to call Days Complete Pivot. The third thing we were looking at was the Wage Margin, so I'm going to go back to my data set here, jump up to the top again, INSERT, Pivot Table, and here what I want to look at is by day what is the average wage margin. So for that what I need to do is I need to pick a date to look at, and then from there I can actually see a trend over time. So, if you recall, I added this Order (M/Y), so this is the date I'm going to look at. And it's aggregated up so if I were in the middle of the month or the beginning they would all show up under that same month. And then I added that field at the bottom, which is Wage Margin, and I can look at that there. Now this one is a percentage, and so what we're going to do is going to see an average of it again because we're not going to aggregate this; we're not going to sum it up. And for the Number Format I'm going to choose Percentage, and there we go. So I have some data here. This will look much better when we get into the chart and you can see, but just overall I can see kind of how it's changing. We're always right around 50 something percent, and down at the bottom you can see I'm at 55. 74%. So I could sort and filter by year and choose this down or aggregate this up, all those kinds of things. So this is a really good one. You know think about if you're answering questions to an investor about your margin on your projects. Here you have that, and it's super easy to get to, and you can slice and dice it a dozen ways. So let's call this one Wage Margin Pivot.

Demo: Pivot Charts

Now let's take a look at taking what we just built there with those pivot tables and visualizing them in what are known as pivot charts. So the difference between a pivot chart and a regular chart is that the pivot chart is based on a pivot table, and with a pivot table, as you can see on the right, I can sort, filter, and I can do all these kinds of things and interact with it, and that'll update the chart as well whereas if I just had regular data set I'd have to set up a bunch of different things in order to be able to have that same functionality. So what we want to do in this case if we want to make it interactive and be able to answer lots of questions for somebody instead of just a single question with our data, then we'd want to make it more interactive. So it's real simple. I can click on my Pivot Table here, and with it selected I can say INSERT Pivot Chart. And for this one it's real simple. I'm going to do an Area chart over time looking like that. It looks pretty good to me, and I can see what that looks like. We'll go ahead and click on this STYLE here. I want the dark one. It has different options there. And with that I can color it differently. I can choose like a color palette. The ones I kind of like are this orange, so we'll leave it like that. You probably want to give it a title. It's generally good to do that, Wage Margin by Month. You can see when I click on this I have the PivotTable Fields, which I can still interact with, and I also have the ability to format the Chart Title, so I have this FILL gradients, all this kind of stuff. I can put a border around it if I wanted to, something like that just for the sake of demoing it. And then I can form at my axes; I can change the types of dates I see. I can do all that kind of stuff here. I'm not going to get too deep into this. The basic idea is that once you have something there you can actually just easily visualize it in something that is just a bit easier to understand what's going on. Now let's go over to our Days Complete Pivot, see what kind of visualization we can come up with here. This one I'm going to do a combo chart. So I'm going to go up to my INSERT again and Pivot Chart, and there's a Combo option down at the bottom, and you can kind of preview and see what that's like. It's real nice how it just does it for you. And you can change the different type and all that just by clicking a couple buttons. And that's what I'm going to leave this one as, kind of good. And I kind of like this visualization, you know, because it shows you the average days complete, and then it shows you how many they've done. So it's a really interesting thing where looking at Bob Turner here his days complete are a bit higher, but his number of rows, so the number of projects he's completed really isn't much different than say Allen Young over to the left there. So something is going on here, something interesting. The data is certainly much easier to consume this way than over to the chart on the left. I certainly wouldn't be able to understand the variance here versus that. So one thing I may want to do is edit this axis. Since it doesn't start at 0, it's not always telling the true story. So it may look like Frank here, he's at 1. 95, and that looks like about as half as the time as Bob Turner, but we can see that Bob is merely at 2. 09, so it's really we're talking about not much of a difference between these two. But because this axis doesn't start at 0, it really kind of skews the truth. It makes it look unrealistic, so this is where you may want to format the axis and dive deeper and make sure that the data is being represented exactly how it should be. So to do that I just right-clicked on the axis, and then I can change it from a minimum of that to 0, tab out, and you can see how it updated. See, now it's much more of a true representation of the data. And in other courses out there, I have a couple, and a few others do, talk a lot about just the idea of visualizing data and how the default settings may not be exactly what you want. You know, it would probably make sense to on the right side as well change this axis so I have thousand separators or I just have 2. 5K, 3K, that kind of abbreviation. You want to basically remove any chart junk, as it's known, to where things that really aren't necessary aren't there because you want the data to shine through. You don't want to be distracted by all this other chart junk I guess is the best way to put it. Okay, now let's take a look at our Sales Pivot. In here we're going to do a bar chart, so I'm going to do INSERT, and I'm going to choose Pivot Chart. And I'm going to do not a Column chart, but an actual Bar chart. In other tools these are both known as bar charts. One would just be horizontal versus vertical, but in Excel's world we call this one a column versus a bar. So I'm going to click OK, it gives me this nice bar chart, pretty much exactly what I wanted right out of the gate. Get rid of my format options, get rid of my PivotTable Fields. Let's just jump over here so I can see what that looks like. Looking pretty good. So now what I'm going to do is I'm actually going to try to sort this. So if I go to sort my Sector here, I'm going to sort descending, not like that, but with the more sort options, and I'm going to sort descending by Sum of Sales Amount. And you can see what it did there. It's interesting how these are kind of opposites. So usually what I'd want to do here is I'd want to choose the A to Z and then do the more sort options so I can do an ascending on Sum of Sales Amount so it gives me kind of the top to the bottom on the right side here. We know that this is total or we know that this is sales amount, so I'll go I'll go ahead and delete that. I'll give it a title here, Sales by Sector. And I'll format my axis a little bit so that way I don't really need to see all those zeros there. We're talking in millions here, so I should be able to do something like that. There you go. So that kind of makes it a bit cleaner. There we go; looking pretty good.

Demo: Going Interactive

Okay, in this demo we're going to take what we've done and make it interactive. Remember, this is where we're going to allow our users, as well as ourselves to explore the data and look at it from many different perspectives. This is when I think about testing, when I think about really understanding where the meaning lies is an interactive visualization that allows me to think about things from lots of different angles without having to go back and recode or rebuild. So here on our Sales Pivot what I'm going to do is I'm going to click on this guy and see that I get this ANALYZE and DESIGN and FORMAT so I can change a lot of the ways that this looks and feels. And I can just kind of highlight those, and it'll do it interactively. It's pretty cool. And now I want to analyze it, and this is where I can say Insert Slicer or Insert Timeline. So I'm going to insert a slicer, and I'm going to insert the slicer by Region. So what this is going to do is it's going to give me this little box here that I can work with, and I can size this and format this and change the different look and feel of it, etc. And when I click on this now, right now it's selected on all meaning my chart and my data are representing all regions, but if I uncheck that notice how it changed, and I can see that across the board, for example, Technology continues to be the most popular sector for us in terms of our services. I can hold Shift and select all of them or hold Ctrl and just select a couple of them, those kinds of things. So I can interact with the data in ways that allow me to answer lots of different questions, and it's this really nice, fluid experience where as I click on it the chart instantly updates, and it makes it really easy for me just to kind of without having to pause and think about how I'm going to build something, just kind of at the speed of thought analyze my data. Let's go ahead and add another slicer here. This time I'm going to look at Product Category, and I can then see what types of products and services we're offering and where those are popular, etc. So just similarly I can click on that, and I can see exactly where all of these things are being done. So that's a fun little way to do it. I'm now going to make this guy a little bit bigger, drag this guy down here, and I'm going to add up here a timeline, like a date filter. So just like before under ANALYZE I can go Insert Timeline, and I'm going to base it on Order (M/Y). And it dropped it right on top of my chart because that's where I wanted it. Just kidding. And I can scrunch this up, I can look at this, make it bigger. Now it's going to default to months. Let me make this just a little bit bigger here. I'm going to hide my ribbon on top so I get some more screen real estate since the resolution here is kind of small. So here what I'll do is I'll make this a bit bigger. And now it's going by months, and you can see I can scroll all the way back, and that's not an ideal experience. You know it might be cool that I can look at it at a monthly level, but because my data is so wide ranging it isn't great for me to have to scroll horizontally. So this little timeline has a cool feature where I can actually change my aggregation or my level of detail up to quarters, and now on quarters it's a lot easier because I don't have to scroll horizontally. And if I wanted to filter say just up to Q2 of 2012 and starting at Q2 2011, I can easily do that, and then down below you can see exactly how it is. Just like the slicers when I update, it automatically changes it. So those are kind of the two main ways to add interactivity here. I could go in and add it to all of them, but I'm not going to because that would just be repetitive, and I think you get the point there. There are other things you can do here like these Filter Connections, and it can filter different pieces. So if you have lots of pivot charts, you can say click on this one, it filters this other one, and really control the interactive nature of it. There's also the POWERPIVOT, which is the thing I mentioned before where you can really come into a new experience where you're able to drill down and do much more complex calculations. There's also this Power View thing, which if I click back on my data set I should see INSERT Power View. Again, more advanced ways of interacting and looking at your data that are well beyond the scope of this course. I just wanted to point them out so that you're aware if you really want to dig in deeper here or in case you're already an Excel user and maybe those are new to you. I would encourage you to explore those and see if they give you want you want.

Where to Find More

Okay, so now that we're wrapped up here, let's take a look at where you can find some more info. First there's an Excel test on Smarterer, which is a great way to test your skills and see how you fair against others. There's also an analytics one, which is pretty useful. Then we have this Excel course here, the Analytics Superhub. I recommend checking this out. You have here lots of different things. They get really, really deep into Excel as a BI tool, the data management of it, the data modeling, PowerPivot and Power View as I mentioned, and then getting into mapping and how it ties to SharePoint if your company uses SharePoint. So this is great course by a great author, Andrew Brust, so I recommend that if you really want to dig deeper into Excel. There's also this PowerPivot course, which talks about self-service BI and PowerPivot. That's just another example of where you can really take Excel to a whole new level. This is where you have this idea of DAX, this data expression language that allows you to come up with complex calculations that do things at different aggregations. So if you're really digging in deeper, this would be a great place to go as well from Bill Pearson.

Databases Overview

Introduction

Welcome to this module on Databases. Here we're going to talk about what databases are, go through some examples conceptually to try to understand the difference between that and say a spreadsheet. We'll talk about why we would use a database, look at the different types of databases there are, figure out how they actually work, and then we'll conclude by showing some additional resources for you if you really want to dig into the databases' world. Okay, let's get started.

What Are Databases?

So what are databases anyways? Well, I think of them as computer systems that we design to store and organize our data. And there are kind of two main areas, the transactional and analytical purposes. So the simplest way I think I can explain this is by starting out looking at our spreadsheet that we used just a minute ago for Cogsley Services. In here we had all of our data that we needed to do our analysis, and it fit nicely and neatly in the size and scope of our Excel document. And if we think about these tabs, each different tab had essentially a different set of data. You can think of those in the database world as tables. Tables are just like sheets. They contain different relatively autonomous units of data. And, other than that, inside of a table we have columns, so think of columns just like you would in a spreadsheet. In fact, it conceptually looks and feels exactly like it does here. And then the third main concept is rows, and rows are exactly what you would think. Those are the actual lines of data that are stored within our tables. So when we talk about tables and we talk about databases, these are kind of the main three tenants of what you're talking about, and they generally apply to almost any database you're going to work with no matter which vendor it's from, SQL Server, Oracle, MySQL, etc. Some of the more newer NoSQL databases have different concepts that don't totally apply to this, but in general this is a good foundation that anybody working with a database should understand. So what happens when we want to put our data into a database is we need to take these tables and essentially combine them into a singular vehicle. In this diagram you can see that we have several different tables all representing the different tabs of our spreadsheet, and they all kind of fall in directly into this data structure that allows us to write queries and interact with it programmatically. And that's really the goal here. That's the reason why we would use a database as well, so we'll talk more about that in just a second.

Why Use a Database?

So why use a database anyways? Well, databases provide a tremendous amount of capacity and flexibility in working with our data far beyond that of a spreadsheet. So when we need to store large quantities of data of we have complex relationships in the data or we just simply need to interact with it in more sophisticated or advanced ways, databases provide all the capabilities we need, whereas a spreadsheet is very limited to what you can do kind of by hand working with it in Excel. So think about a situation like this. This is a real famous photo of the Office of Veterans Affairs in the United States, and these are all claims. These are all backlogs of veteran's claims for medical insurance and other things like that. And think about this system and how difficult this is to work with. It's almost impossible, hence the backlog, hence the issues, so honestly this is why you need a database, to alleviate this type of a scenario. So instead we put that data into servers and hard drives and racks, and then we can sift through it at lightening speeds and financers and process claims and get people healthy. So that's just one example of how and why you would use a database there.

Types of Databases

Now let's talk about different types of databases. The first one I want to mention is relational databases. These are the most common traditional ones that have powered and still power many applications and computer programs around the world today. They're highly normalized, which means that you take data and you kind of spread it out so there's no duplication. Imagine if you have customers and you have your orders. Well, you may not want to have the customer information in different tables because what'll happen is if you need to update that customer information now you'd have to update it in multiple places. This can lead to failures and anomalies in the data or orphaned records as they're called. There are also these transactional designs. So the idea here behind these processing systems is that they will be there for handling things like orders coming in or people registering on websites or sending things. They're transactional in nature and as opposed to the analytical databases we'll talk about in a second. They eliminate redundancy, so they're actually quite efficient when it comes to storing the data because the data isn't duplicated in multiple places. But that also leads to this issue of scaling. They don't really scale to support these high or massive volumes that we're seeing in technology services today. And the reason is because, well, there's almost a single-threaded nature of them where a transaction must occur, it must be written, and all these things have to happen before it can move onto the next transaction. So the challenge there is that when you have a website like Facebook or YouTube or Google there are billions and billions of these transactions happening every second, so the challenge is you can't really scale a relational database to handle that load. So we'll talk about what option we have there in a second, but that's the main idea here is that relational databases have been around forever. They're really valuable, and they're really great in what they do, but they are kind of very focused in their intent nowadays. We have analytical databases. These are the ones that we'll be talking about a little bit more in this course as we get into the data analytics side of it. These are de-normalized. So if you think about instead of spreading the data out into each separate table for each different part of it, which is great for efficiency of storage and transactional nature in eliminating redundancy, it also can add a great amount of complexity when it comes to trying to query or analyze that data. And so these ones are de-normalized and what I like to think of them as human oriented, so they're based around what you as a person would want to get from the database when you think about it. If you ask tell me about your customers, for example, from our database, you'd want of all the information about the customer to be there instead of having to go to 15 different places to get it and join it all back together, so analytical databases are designed around this kind of human way of thinking. They simplify your queries. A query is a way you can ask a question of a database. Later we'll talk about SQL and that query language, which is how we actually ask questions. And these are often referred to as data warehouses and data marts. There are differences between all of them, and the schools of thought go way deep on every angle that you could imagine, so I won't touch too much on it, but later in the course we will talk about it a fair amount so you should have a solid understanding coming out of this. The last one I'll mention here are NoSQL databases, and this is a very broad simplification of the different types here. Some of them are document databases, key-value databases, and graph databases. All of these are things that are really kind of the new evolution of the relational database systems of the past, and they're helping support all of the insane amount of technology innovation that we've seen, different systems doing different things at insane volumes and scales, so these types of databases are just the evolution of those as the way I think it. And they also have different ways of handling data, so we don't talk about them in the same way. And I won't go too deep into it here, but I will point you to other places where you can learn more about it. One of the key attributes of a NoSQL database is that they're generally schema-less. A schema as a definition, imagine our Excel doc having rows and columns, and we know a certain column is currency, the other one is text, the other one is a date value, and they have a very specific name and nomenclature, well these databases, because they want to be scalable and not have to worry about that up front, they allow you to store data without having to predefine what the columns and rows and everything should look like, which is one of the key advantages and why people tend to use these for a lot of different development efforts today. Now NoSQL isn't the anti-SQL. Some people tend to think that, but it truly stands for Not Only SQL. Most of them still support SQL in some flavor or fashion, but they also support many additional programming interfaces beyond just SQL, so that's really the intent here. So if you're building an application and maybe SQL isn't the best language or way to interface with your database, these ones will support other additional languages and interfaces. And they do scale incredibly well. There's this thing called the CAP Theorem, and that kind of defines the different types of database and where they shine and where they don't. The idea is you can have two, but never three of the C, the A, and the P. So I won't go too deep into that again, but you can go research that, and there's a lot to learn there, really interesting and fascinating stuff happening in the database world today.

How Databases Work

Now let's take a look at how databases actually work. Well, first it starts with the user. And imagine a user is interacting with a website. They are on the computer; they enter some information. Let's say they sign up for something. Well, that request is going to be sent over to a server that's processing the logic of the action with the user. Then the data is going to be sent over to a database. The database will write the data and possibly read back some data facilitating these transactions, whatever is occurring here, and then it will kind of cycle through this process and send the information back to the user. Hey, congratulations! You signed up for our service. And this is basically the flow of most web applications and most computer systems in general is that the backend database is continually processing information and transactions coming in and retrieving things, and it's just this never-ending cycle. As long as people are using systems, this is what's happening. So this is kind of a simplification of that process, but one that hopefully visualizes it and conceptualizes it to you as it's maybe an entirely new realm of working with a database and not understanding say how user information gets in there or how data gets updated. It's generally always happening through some sort of interface that is driven by the user.

Demo: Setting up SQL Server

Now let's take a look at setting up SQL Server. SQL Server is Microsoft's most popular database platform that they offer. It scales from small desktop applications up to hundreds of servers and thousands of users. So step one here is to go to the download page on microsoft. com. And I would give the URL to you, but it seems to change a lot, so if you just Google for this, SQL Server Express is the one we're looking for. This is the free version. There are also other levels of it. There's Standard, Enterprise, and there's a separate one for Business Intelligence alone now. And so what we're going to do in this course is we're actually going to do SQL Server Express. So when you click here, go to the link to download. It'll ask you to log in with your Microsoft account. If you don't have one, that's okay. You'll just need to create one at that time. It's free. And then once you're there it'll ask you to log your information. And then you're going to go choose the SQL Server Express with Tools, either 32 or 64-bit based on your application. Now we'll mention this is only for Windows, so if you have a Mac or something else, you'll need to a virtual machine in order to follow along for this section here. So once you select your type that you want to download, there'll be a few additional pieces of information you need to fill out. And then you eventually get to the bottom, you click Continue, and it's going to download it. I'll warn you this does take quite a bit of space on your machine, so it may take a while to download. I'm going to fast forward to that point to where we're looking at the install, and then we'll continue after that with looking how to set some things up. And once you've downloaded, just double-click on the exe file. We'll click Run. You'll need to extract it. And I've already done that, so I'm just going to double-click in there. And from here it'll kick off the Installation Center. We're going to choose a new stand-alone installation. As we go through this setup here there'll be lots of dialogs that we're just going to mostly click Next through. And then once we get to the install part I'm just going to fast forward that so you don't have to sit there, but I wanted you to see what it looks like so if you're having any issues on your end you can compare in contrast to what I have going on here. For the features we'll just leave the defaults selected. Probably not all of these are needed, but what we'll be able to do is just see all the tools and everything that comes with it out-of-the-box. For the locations we'll leave it at the defaults there. More advanced installations you can put that in different locations, but for now we'll just click Next. We're going to keep the named instance here of SQLExpress. In more advanced situations you may want to name it something else, but here we'll just leave the default again. Again just we'll keep the default settings here. Click Next. Here is where we could add a separate system administrator user and give it local authentication as it's known. We're not going to do that for this installation because it's just demoing exactly how to get going. The other option is Windows authentication where you can log in with your Windows account. And this is going to work good for us here, so we're just going to keep that the way it is and move on. Okay, now is when we're going to run through the installation here. I'm going to fast forward until it's all done, and we'll regroup once it's complete. Okay, now that the installation is complete we should get this nice success message here. You can check all the services, make sure everything succeeded properly. Go ahead and close that, and we're good to go. Now let's move on and learn how to load some data into SQL Server.

Loading Data to SQL Server

Okay, now that SQL Server is stood up, let's take a look at loading some data into it. I'm starting here with the DBImport Excel file. It comes with the downloads of this module. And what you can do here simply is save it as a CSV. So we're going to do Save As, Computer, find a place, and instead of saving it as a Excel workbook we're going to do a CSV file. And a CSV stands for comma-separated values. Now before we do that, if you're doing this on a different sheet, you're going to want to make sure that all of your formatting is kind of removed with the exception of dates. What'll happen is if I were to save this as a CSV now with this percent sign here it would actually treat it as a piece of text, and that's not going to be good later when we want to write a query against it. Same goes for our dollar signs here and any of the other number formatting columns that we have going on. The dates, however, we do want to keep as date values because if we were to change this to General it would actually show as a number because this is how Excel actually stores this. Now instead of that, what we're going to do is we're going to leave it as a date and then save it as a CSV there. So here's what it looks like after we've saved as a CSV. Notice the formatting is gone, and the columns aren't calculations any more. They're just all the raw values. And looking at all these here it looks like everything is stored correctly. So this is how we want to get our format for this example where we're going to load the data into SQL Server. And before I move on there's one additional step I want to highlight, and that is removing any additional spaces or anything inside of the column headers. With this it's going to cause a problem when we try to import it and wants to create our table, so I'm just going to go ahead and make these a bit more database friendly. So by that I mean there are no spaces and that everything is using the kind of CAML case here. Notice how this one has a space before and after. We'll get rid of those, get rid of these guys. We're good to go. So whatever file you're using here, make sure that you've removed any spaces, leading, trailing, in between, etc. This way when we go to import to the database everything is going to be real nice and ready to go. Here in this example make sure our recommendation is that you use an uppercase for everything to separate the words. That way it's real easy to read. Okay, good to go, so I'm going to go ahead and save this now. It's going to give you the errors like it likes to. Go ahead and close it. I'll say Don't Save because I've already done that. Now we're going to open Management Studio. So from your program menu here you should be able to go to All Programs, SQL Server 2014, and Management Studio. Management Studio is where we do everything basically in SQL Server. And so this you should be able to connect immediately with your Windows account. And we're in. We're in here, and I'm going to go ahead and create a database called CogsleyServicesDB. I had done this before, so let me just show you how to redo that. I'll delete this one first, hit OK, right-click and say New Database, CogsleyServicesDB. Leave everything default for now. And there you go. Now when we open this guy up here we have tables and we have Views and we have Programmability such as Stored Procedures, really useful things when building kind of more complex applications. I won't get into any of this detail here because this course is just kind of at the high level, but there are many, many courses on SQL Server in our library you can use. So I'm going to right-click on this CogsleyServicesDB and go to Tasks, and down here I'm going to say Import Data. This'll open up my Import/Export Wizard. I'm going to click Next, and what we're going to look for for our data source is a Flat File. The Flat File here is going to be the CSV, and with that that's the one from the demos for this module or whichever one you happen to be using would be fine. If you generated it from Excel, you're going to need to use double quotes as a text qualifier. That's what it uses any time it finds text with other commas or things in it. It puts double quotes around it to make sure it doesn't mess up the kind of ordering. We'll click Next. You can see here that we're previewing some rows, and everything looks to be in order. Sometimes if you didn't have that text qualifier some of the other pieces of these won't line up properly. That all looks good. If I had a different delimiter as well, I could change that. I'm going to go to Advanced because I think that I would want to show all of the details here. So this is exactly what it's doing. It's generating, it's assuming a lot about the metadata here, so each one of these fields it has the width of the column, the type of column, all that kind of thing. I can go to Preview, and I can see what it looks like. All looks good to me. Now we're going to choose a destination. Here I'm going to look at SQL Server. I'll connect to the database that already is there. And viola! It actually created a table for me. So here we have create destination table. We have Edit SQL. This is the code it's actually running. Don't worry about that right now, but if you did want to edit that you can. I can scroll down. I can see the data types. So like in Excel we have formats for currency and dates and those kinds of things. A varchar is a text-based format that is variable in length. So that's fine for now. We could change that and mess with it, but I'm just going to leave it as is because I just want to get the data in there. Click Next a couple more times, and then click Finish. We got an error. Let's see what's going on here. It's saying the data conversion for the column Industry returned status value 4 and status text. Text was truncated, and one or more characters had no match, etc. So truncated. What this is telling me is that the field Industry I'm trying to put it into a column and a database that it doesn't fit. And it happens to be specified that it could truncate it, but in this case that gives us an error and fails because it wouldn't be able to load the data properly or completely. So what I could do here is I can go back and edit the mappings. Now I'm going to have to in this case make sure I do drop and re-create destination tables so that it recreates it. And let's go ahead and go down to Industry, just add a 0 to the end of this. And we'll do it for Sector as well. I'm guessing that one is probably a bit long. Click Next, get all the way down there. Still got an error. So still getting that same error, which means it's not our destination table. It's actually all the way back to our Flat File Source under Advanced. So what's happening here are that the SQL Server tools, and the one behind the scenes here is called Integration Services, is very particular about data types. It's very, very picky about this, so it doesn't recognize that if this field that it's pulling from is greater than 50 the OutputColumnWidth that's it's expecting it's going to fail, and so we have to edit that there as well. So now let me go click Next, click Next, edit the mappings. Now here we are doing an append rows to the destination table. It's not exactly what I want, so what I'm actually going to do is I'm going to cheat it a little bit. I'm going to go back one. If I jump over to Management Studio, click on the dropdown and refresh, you can see that my table is here. I'm going to go ahead and delete this table. Okay, now let's jump back to our Import Wizard. Here we go. Now let's edit some mappings. Let's refresh, click Next. Here we go. Now we're going to create the new table, and let's see what kind of metadata it has. So it picked up the metadata coming from the file, which is what we wanted. Click Next, click Next and Finish. And there we go. You got about 8400 rows transferred. We can look at a report, and we can see what kind of stuff was happening. You can see all the different steps that occurred and messages and any warnings or anything like that. But in the end that's that. The data is in there. So let's go to Tables, Refresh. Now I can expand this, and I can see, as I mentioned before, columns, so it picked up all the names and the data types. If I right-click on this, I can say Select Top 1000 Rows. This is how I retrieve data with a query here. This is getting into SQL, kind of scripted it for me. But you can see there we go. The data is now in our database. So we've accomplished that. And I wanted to specifically not show a demo that just works perfectly without any errors. I intentionally wanted to show the ability to troubleshoot and focus on because I know it's always frustrating when you see something online and it worked perfect, then you go to do it yourself, and you get lost, and there's an error, and you're completely lost, so that exercise was there as a way for you to understand kind of the different ways and places to look, at least one example. As you work more and more with this, what you'll see is that you just get to the point where it's kind of intuitive. You see a message, and you know exactly where to go look. And these tools, just like all computer systems, are prone to these types of errors, so don't be surprised if you hit them. Don't imagine everything is going to go smoothly every time.

Where to Find More

Okay, now let's take a look on where we can find some more info. On Pluralsight we have a bunch of different courses here. We have our Postgres SQL, which is a different type of database just like SQL Server, relational database that's open source, really popular. We have a really good course there on getting started with it. We have this Introduction to Graph Databases and Neo4j. As I mentioned, the NoSQL database's graph databases are one of the really interesting ones that allow all kinds of interesting things about suggested searches and recommendations. Imagine when you go to Amazon and you're looking for something, and it says people that bought this also bought this. Chances are it's using a graph database behind the scenes to come up what's called a vector map, so interesting stuff there if you kind of want to go off into the more advanced interesting realm. Then we have MySQL and SQL Server Fundamentals. These are both really popular relational databases that are a great starting point as you get into this database world here. On Code School we have a course called Try SQL. This course is a really good way to get introduced here, as well as you'll see that again when we actually start to do some SQL ourselves. By the time this course publishes they may have more on that, but I recommend taking a look at Code School as a really good way to get introduced to these kind of new concepts for you.

Intro to SQL

Introduction

Welcome to this module, Intro to SQL. We're getting deeper down the path on our data analytics journey here, and here we're going to actually write some code and interact with the data we just loaded into our SQL database. We're going to talk about what SQL is, talk about the definition of it, figure out why we would use SQL in the first place, go through some SQL basics, talk about how SQL actually works under the covers, and then we'll learn where to find more. Throughout this module we're going to have demos following along with our story of Cogsley Services Inc, so I hope you're ready to get started here and really get down into the details and gain some new super powers using this programming language known as SQL.

What Is SQL?

So what is SQL? Well, SQL is a programmatic way to interface with data inside of a database. If you recall the previous module, we loaded some data from Cogsley Services into our SQL Server database, and SQL is the programming language we're going to use to interface with that. It stands for Structured Query Language and is actually designed around the English language and how it is structured as a way to get data and retrieve it or interact with it inside of our database. Thinking back to our example of our spreadsheet and how when we loaded the spreadsheet data into a database we loaded the data into a table how in the columns that's how we defined those tables, and the data themselves lived in rows. So these are the terminology we use when we are thinking about getting data out in a database, as well as how we retrieve it using SQL. We get data from a table, and we select which columns we want out of that table, and the results are in the row format.

Why Use SQL?

So why use SQL? Well, SQL offers this incredible capability for us to interact with our data and perform all types of operations. Think about things like writing a new record to our database such as a new user that's signed up or potentially a new sale from our point of sale system, as well as do things like look at averages over time or answer complex questions about which types of customers are the ones that we should be marketing to more because they tend to be more profitable or stick with us longer as in terms of retention. All these are things that you can use SQL for to answer questions of and perform different types of operations.

SQL Basics

So let's look at the SQL basics here. We have our tables, row, and columns like I just mentioned, and this is how we're actually going to write a query. A query is a question that we can ask of our database. It's a piece of SQL code that we write, which will actually retrieve data or manipulate data inside of our database. So it starts out, and just to recall SQL stands for Structured Query Language, we start with this select statement is what it's called. This is the first word that we're going to write in retrieving our data, and it's one of the most common fundamental things that you'll do when you're writing SQL is start with select. Then we're going to tell it which columns that we're interested in. So just like our columns in our spreadsheet, in our tables we're going to retrieve a certain amount of columns from them. Then when we define where we're getting our data, we're going to choose from as the word and put the table name after it. Now this is interesting. This is important because we can pull data from multiple tables. And that's a bit more complex, we're not going to get into that section here, but we start out with select and from, and these are the two basic building blocks of a SQL query. Here's an example. So let's say we want to pull back CompanyName and Region from our table. Well, we say select CompanyName and Region from sales, sales being the table. Now that's the most basic type of query, and I'll demo that for you here in a second so you'll see it in action. But what if we wanted to filter this? What if we wanted to say I only want to see this data where a certain condition exists? Well, this is the next building block of a SQL query where we define that condition in something like where Region = 'west'. So you can understand how this becomes a language in itself, and it's modeled after English and how you may actually use English to ask a question. In here we're asking a question of a database. It seems to flow along with the style of language and the syntax of the English speaking world. So just to review our statement again, select CompanyName and Region from sales, that's our table, where the Region equals west, so we're filtering our data down. There's a lot more to select statements, but I want to go through this example here because I think this is one of the basic building blocks. And if you get this, it'll take you a long way. And then as you go and become more familiar with SQL, you can dig deeper and learn all of the other nuances and additional capabilities here, but this is the basic unit here that we want to start with.

How SQL Works

So how does SQL work? Let's look back at our example here where we have our user interacting with our website. Then that website interacts with a server or a computer system on the backend. That backend server interacts with a database, which stores the data in tables and organizes it and interacts and returns some results back to the user. These things continue to happen as the user interacts with our product, and we're also going to have some operations inside of the database itself, so we have lots of moving parts here. Well, there's SQL going on inside of the database, and often the back and forth between the website and the database itself can also be done with SQL. So SQL is an integral part of how most modern web applications or computer systems in general work. It is how you can interact with the database, but it's not the only way. There are many other interfaces and types, but here are some common areas where you'll see SQL happening.

Demo: Basic Select Statement

Now let's take a look at writing an actual select statement. I'm going to open up SQL Server Management Studio, and in Management Studio I'm g got connect to our server, which we installed locally. In here I have my CogsleyServicesDB, which we created in the last module, and we have a table. This table was originally named something like CogsleyServicesDBImport. I went ahead and renamed that to sales just to simplify. That way when we write our query we can keep it real easy. So, as I mentioned, each table has columns in it, and the data in them is represented in rows. So, if you see here, I have all of the columns. These are what I imported. Each row here is its own sale, its own order number. And the easiest way to start out is just to right-click on my table and say Select Top 1000 Rows. And there we go. We have a basic select statement. Now I'll point out that this TOP 1000 is a SQL Server specific syntax or keyword. SQL is an ANSI standard meaning that all databases that support SQL will at least support some standard version of it. Most of them have their own additions to it, other ways and capabilities that they've baked into how the code is interpreted, but just know that unlike a lot of other programming languages this is one where there is a standard across all database platforms, so it's really powerful. You're able to nail this down because it doesn't matter as much which database you're then working with, Oracle, SQL Server, MySQL, or anything else. So I'm going to remove this top one there, and I just want to show you a couple things. I'll keep RowID OrderID, OrderDate, and then I'm going to delete everything up to CustomerName and Industry and then Region, and I'll delete the rest of these here. So just like my example earlier we have a SELECT statement, we're retrieving columns from our table, and then we have a FROM. It's retrieving it from a specific table in our database. Now SQL Server does this where it wants you to point out exactly which database, and this is the owner of the object, the owner of the table, and that's because that up here in the top left I'm looking at the master database is where I'm trying to execute my query. If I change that to CogsleyServiceDB because that's the one we're working in, I can delete all that and just have sales. Now these brackets that are around all these are because SQL, when it auto-generates a query, it needs to be overly cautious. So if there was a space or a special character in one of these column names, which isn't usually recommended in the databases, then you would have to put these brackets around it. We don't have to do that, but rather than having to re-go code and change all of it, I'll just leave it as is. It doesn't make a bit of difference when I run my query. So now if I were to execute this query again, you'd see that I have a slimmer data set, and I still have one row per order, and I have just the data that I want. Now if I wanted to eliminate the RowID in order to just get a list of customers, industries, and regions, I can do that, but when I run this query what's going to happen is I'm going to get duplicates, so we're going to have to figure out how to remove duplicates. Well thankfully in SQL there is a clause called DISTINCT. It's common practice to put the keywords in uppercase, and most languages here will highlight them for you so you know it's a keyword. If I add DISTINCT here, what it's going to say is give me basically all the unique combinations of the rows and columns that I'm selecting. So where I see Liz Greer Television Services in the West, now I'll only see a unique one of those, so I won't see three, I'll see one. And indeed I only have 1106 rows now, not 8399, which is the entire data set.

Demo: Filtering Results

Now let's take a look and filter our results. Jumping back to the query we were just working on, if I only wanted to see sales from the west region, what I can do is add a where clause. Now where is going to accept a condition, so where Region, I can click on that, equals West. And if I execute the query now, you can see down below that Region everything equals West. If I scroll all the way down, it's always West, and in fact I'm only looking at 261 rows. So you can add multiple Where conditions. So if I wanted to say where Region= 'West' AND Industry=, and I'll just copy one of these here, Business Services, so I want to see all of my West customers that are in the Business Services industry. I click on that, and I have a list of seven folks here. So that's a way you again can just narrow down and filter your query, so you can start to see how interactive and how dynamic you can be.

Demo: Ordering Results

Now let's do another thing here where we order our results. I'm going to remove the Industry filter, and what I want to do now is right now it looks like they're ordered by CustomerName, and that's cool, but I also could order them by Industry let's say. So there's a clause ORDER BY, and I just type the column that I want to order by. And there we go. Now we're ordering by Industry. I also can put the column number. This is a bit trickier, but sometimes it's better if you're doing dynamic stuff. I could say ORDER BY 2, and it'll order by the second column. And there you go. It ordered by Industry.

Demo: Aggregations

Now let's dig a little bit deeper and ask a more complex question and do some aggregations. Here I have Customer, Industry, and Region. Well, let's remove Customer, and let's just keep Industry and Region. And the question I want to ask is which industry and region are the most profitable? Well, if I have just those two columns in here, what I can do, I'll format this a little bit, I can add an aggregation here. So I'm going to add a comma, and I know I have a field here called SaleAmount, which represents the total amount of the sales, and if I ran this query the way it is now, and I'll get rid of the ORDER BY, it's going to give me everything in the region West and all of the sales here, but it's not going to aggregate them. So notice Television Services isn't there, and all of these are here. So if I added DISTINCT like we had before, it would treat each one of these as separate entries because the SalesAmount is different. Well, if we want to aggregate and say okay well sum those up, give me the total for Television Services in the West, I'm going to add an aggregation type. Just like in Excel I have things like sum, average, min, max, and I put that around the column name, give it an alias by typing the word as, and I can say TotalSalesAmount. Now when I execute this query I get an error, and this time it's showing me that the varchar is invalid for the sum operation. What that's saying is if you recall when we imported the data we treated everything as varchar because it was easy, and now we're paying for that mistake. So the point of this is that we need to know how to convert this field into something that can be summed. Well, we know that SalesAmount is probably going to have decimals, so we need to use a function called cast. I'll say cast around this, and I will say as decimal. Now the next error we're going to get is around the fact that Industry is not in the group by clause. This is the last one I promise. We need to tell SQL here that I know that this one is going to be aggregated, so we're good there. What we need now is Industry and Region to be our grouping clauses. This just says that basically like when we were asking for a distinct list of things here we're telling SQL that we want to aggregate this number and group it by these, so these will be our kind of areas in a pivot table that we would pull into the rows and columns. So now when I execute this I get the result that I was looking for. Now I see just the Industry once and the Region West because I'm filtering on it. If I got rid of that filter clause and executed this again, I would now see all of them. Just like before I can order by. So if I wanted to see where my largest sales was I could say order by 3, and I need to tell it descending because it's ascending by default. And there we go. Television Services in Central is the winner here where we have the most amount of sales out of all of our data. You can see here there are several things that happen, and a lot of this is the trial and error process, and I really want to introduce you to that early because I believe it's paramount to being a successful analyst. As well as learning SQL is learning how to troubleshoot these things. So as you get deeper and deeper, also don't expect that you'll write a query and it'll always work on the first try. There will always be issues, almost always from your very first go, so don't get discouraged. It happens to the best of us, and it happens for your entire career, so you just come to expect it that you're going to write some SQL and you're going to get hit some speed bumps, but keep going, and eventually you'll get your answer as we did here.

Where to Find More

Okay, so to wrap things up we're going to take a look at where we can find more. On Pluralsight we have a couple really awesome courses. One is Introduction to SQL. Another one is SQL Server Transact-SQL Basic Data Retrieval. Transact-SQL is Microsoft's version of SQL. As I mentioned before there's an ANSI standard, but then there's also the vendor's choice, and there are vendor's additions to it. Microsoft's one is called Transact-SQL. On Code School we also have a couple courses. We have Try SQL and Sequel to SQL. Both of these are really great introductory ways to get your hands on. It doesn't require installing anything. You just start typing in your browser, and you get working right there.

Intro to Data Modeling

Introduction

Welcome to this module on Intro to Data Modeling. We're getting pretty far down our path and becoming an awesome data analyst here, and in this module what we're going to do is take a look at what data modeling is, why we would use data modeling, some key concepts, and then where we can find more.

What Is Data Modeling?

So what is data modeling? Well, it's this process of structuring our data inside of a database for a specific purpose. There are transactional systems that we design for, analytical systems that we design for, all types of different purposes that we may want to design our database around, and this is just the process of putting our data into a structure that will fit that purpose. Think about our example here where we have our columns, rows, and tables. Well, in this case what we're talking about is organizing these tables in a way that they have relationships, and if we want to query them with SQL, which we just went over, we'll be able to write SQL to join the data together. So thinking of that you have all these different tables here, and I represent them with these bubbles, but just to conceptualize that they are separate entities that are all related living inside of the same database. So this is the process here that we'll use to actually structure those and build the relationships and connection points so that we can make sense of it and work with it for our purposes.

Why Use Data Modeling?

So why use data modeling? Well, by modeling it what we're going to do is enable this perfect system, or we hope to have a perfect system, for handling our data workloads. Again, if we're doing transactional processing like a sales order or a website doing some transactional systems there, we're going to design our database and model it in a way that facilitates that need exactly. For analytics we're going to do it a bit differently than we would for transactional processing. And this is the key difference between relational database design with normalization and data warehousing with dimensional modeling, and we'll talk a little bit about that in the next module. So here's our example of how SQL works and how databases work. If you recall, we have our users interacting with our website. Then that website is interacting with our server, which is interacting with the database. SQL is facilitating all these connection points. Well, down here in the bottom right is where we have the data model, and that's what we're going to talk about today is how this actually structures the data so that way you can facilitate this entire process.

Key Concept: Relationships

Now let's talk about this key concept of relationships. The idea here is this is how tables will actually relate to each other so that you can do your transactional processing or do your analytics or whatever you're using your database for. The different types of relationships: There's the one to one, the one to many and then conversely the many to one, and the many to many. And I'll show you can example here of what that is, but these are just the different ways in which tables can relate to each other. Let's look at an example real quick. If you recall, we have customers as one of our data sets, and it lists all of the different companies in there with their sector, industry, the date they went public, etc. We also have a sales database or a sales table. Well these have a relationship, and in fact it's that one customer can have many sales, but each sale is only for one customer, so we visualize this and we represent it with this logo here, with this little tree, this chicken foot on one end and just a straight line to the other. There is a whole language and syntax for how to model these things. This is just the most basic one I wanted to share with you, and at the end of this course I'll point you where you can dig in deeper if you really want to get into database modeling. There's something called UML, which is a universal markup language as well. So there's really a lot to it here, but just know that when you see this it's basically saying that one customer can have many sales. Now how does this impact us with SQL? Well, if we did a query here and we said Select \* From Customers, we'd get one row per customer assuming that's the granularity or the grain of our table. Well, if we try to do a similar query and join it to Sales like this where we pull from Sales and then join to Customer, conversely we're going to get one row per order and then duplicate customer info because we're pulling back all of the orders and then all of the customer info, so one customer could have many orders. So what'll happen then is you'll have many rows with customer information if that makes sense. So this is where this really hits home and really becomes real is when you start to interact with the data. These relationships greatly define exactly how you can interact with it and actually how the processes work.

Key Concept: Normalization

So now let's talk about normalization. Normalization is this idea that we should reduce redundancy in our database so that way we don't have any anomalies or anything else floating up where customer information was updated in one place but not the other. We should store the element only once and in only one place. It also enables processing efficiency. If I need to retrieve a small bit of information, I have one place to look and one place to go and theoretically a smaller data set to churn through because it's much more consolidated. This is also the separation between logical and physical design. The logical design may have an entity such as customers, and customers may have lots of different attributes about them, say their industry and sector and their address and those kinds of things, and how you physically design that in the database may differ from what the logical entity is, and I'll talk about what that means. And that especially plays in when we talk about data warehousing and how data warehousing basically is a physical representation of the logical model. There are also different levels. I'm just going to mention this. I'm not going to go into the details of each of them. But if you want to get deep into this or if you're becoming a SQL DBA, for example, or a data modeler, you're going to want to know about all the different levels of normalization. So the most common one for something to be considered normalized is 3NF, which stands for third normal form. So let's take a look at our example with our data set here and just keep working with it. So before we talked about tables and how they relate to each other with relationships. Now we're going to talk about the columns and how we actually separate out the columns into new tables to normalize that data set and eliminate redundancy and facilitate processing efficiency. Here are all of the fields from our sales data set. This is the same one we've been working with since the earlier stages of this course. And what I want to do is going to actually parse these out and give you an idea of how you would normalize this. This is a common thing where a lot of companies or a lot of people will build their data set, which is all the fields they want, and this is really good often for analysis. This is how you want things for analytics because it's easy. You just have to go to one place to answer all of your questions. However, when you want to process the data, when you want to use a transactional system to insert and update information, you want to build an application on top of this, it's not ideal to have one table with everything in it, so you separate those things out. First let's just gray out the fields that are kind of superfluous here, the RowID and the RowCount. We don't need those. Those are just added as a factor of our spreadsheet. Now let's look at the other data here, and let's take out our order information. So what I did there is I just simply parsed out all the pieces that basically you'd say are an order themselves; it is its own unit. Now we'll take customer information, which fits neatly right there, and it has all the bits about a customer themselves. Then we separately have product information. And lastly we have our staff or our employee information. So each of these four buckets kind of represent these entities that are related into what you would call a sale. And so we may have in our database model the concept of a sale, and these may be all the different physical representations of that in separate tables. So we give each one of these its own name essentially, and then we would define relationships, which actually defines how our business runs. Now you could further normalize this by breaking out things like city, zip code, state, and region because those may apply to customers, as well as employees, as well as department stores, as well as many other things, so it may make sense to eliminate that from this and have a separate table for either each one of those elements or such as an address table.

Demo: Creating a Data Model in SQL Server

Now let's go ahead and create a data model in SQL Server using the data set we've been working with so far from Cogsley Services. I'm going to open up SQL Server Management Studio now and jump over there, and we'll get started. So once I'm in SQL Server Management Studio I'm going to go ahead and connect to my database. I'm going to take a look. If you recall, we have CogsleyServicesDB. Under this we have one table, sales. That happens to have everything in it. So what I'm going to do is create some other tables that would be related to this similar to the ones we have in our spreadsheet. So step one right here is going click on Database Diagrams and say New Database Diagram. We only have this one table right now, so we'll go ahead and add that. I'll hit Close. Our resolution here is a little tough, so let me shrink this up a little bit, go to 75%. Okay, so here's our table, here's the first table we're going to have, and what I want to do is build all the other tables from our spreadsheet. So, if you recall, we have Customers. I think it was called Companies, but I'm going to call it Customers here, and in there we're going to add a field called CustomerID. It's common in database modeling to always have an ID field that represents every row uniquely. This is called a primary key, and it's how we actually build our relationships. I'll show you that in a second. And then let's just do some basic stuff here, CustomerName, let's give it a varchar of 50, that's pretty common, and then we'll say Industry varchar50, Sector varchar50, and we can do something like IPODate, and this one we would choose as natural date. Okay, so I have a Customers table easy enough. Now I'm going to add another one called Products. And again I'm going to add a ProductID. This one will be an int. Let me go change the other one in CustomerID to be an int as well, do some basic stuff, ProductName varchar50, ProductType, ProductCategory, ProductPrice, call it that. There we go. So we've got another table there. We've got products, customers, sales. Looking pretty good. Let's create another one here for Employees, and we'll have EmployeeID int, EmployeeName how about FName for first name, EmployeeLName for last name, EmployeeStartDate, we'll call that a date, and EmployeeSalary so it will help us with our wage calculations to figure out our profit ratios later. Okay, and that's good for now. We have basically the workings of a data model here. Now a really cool feature of SQL Server is I can just save this model, and it will create all these tables for me. Notice these tables don't already exist. So if I wanted to do this I could go one-by-one and manually create them and then build my model, but it's really easy in SQL Server just to do this here. So I'm going to hit Save and call this SalesDB, and it's going to say the following tables will be saved to your database. Do you want to continue? Yes. Now there are no little stars next to them. If I go over here and refresh over my tables, you can see I have those there now. Let's go ahead and rename sales here so it fits our naming convention. There we go. And I'll go ahead and close this model here. Now if I refresh under Database Diagrams, if this was you're first time creating a database diagram and you're following along, you may have saw a warning about installing the bits you need to create them. That's fine. Go ahead and click that. It just adds the additional components you need to basically do this part here. So that was step one. We created some tables. Now we need to create some relationships here.

Demo: Creating Relationships in SQL Server

So let's now go in and create some relationships in the model we just built. Okay, with this formatted a little bit I think it's time to just talk about maybe what would be appropriate in terms of how to continue to build this data model. This sales table is not normalized, and that gives us a challenge in terms of how to implement this model or set up these relationships. So what I'm going to do is actually remove this from our model, and I'm going to create a new one called Orders. So now I'm going to build this correctly so that I can have relationships here. So remember we'll have OrderID, which will be an int; we'll have OrderAmount let's say, which will be a decimal; we'll have OrderDate, which will be a date; and we could say EstCompletion, which will also be a date. And then now let's relate it to some stuff, so the product that was sold, so we're going to add ProductID, which will be an int; we'll add a CustomerID, which will also be an int; and we'll have an EmployeeID, which will also be an int. So these last three bits here are going to join to these other tables, and it's how we're going to build out these relationships. You may have caught that I misspelled Product in our ProductID table there, so I'll go ahead and change that. Okay, so I'm just going to save this and have it update those tables for me. Done. Now I'm going to create a relationship here, so I can just simply drag ProductID from Orders over to ProductID from Products. Now SQL Server wants us to enforce uniqueness when we create these joins; otherwise, we can have a mini to mini join, which can result in something called a Cartesian product. It basically means that the tables, because they're not unique, could forever join to each other basically eating all of our memory crashing the server, so let's not do that. That's a bad, bad thing. So it said hey you need to have a unique constraint. So we're getting a little bit deeper into SQL Server and how it works, but commonly with data models this is how we do things. So we would have a primary key defined here for products. Now what that means is that this value must be unique. It cannot be duplicated in that table. Similarly, we'll go ahead and do this for employees, for customers, and for orders so each one of these has a primary key. Now since we're adding these keys after the fact, it requires SQL Server to actually drop and recreate the table, and SQL Server Management Studio out-of-the-box won't let you do that. So what you could do is script all these changes and then run that. But rather than get into that, I'm going to show you how to get around this so that way you can do this on a continuous basis if you so choose. Especially when you know what you're doing this early on it's really not very difficult or there's not like a high degree of uncertainty or danger. So I'm going to go to Tools and Options, I'm going to go to Designers, and under Table and Database Designers we have prevent saving changes that require table re-creation. If I uncheck that and click OK and now click Save, it allowed me to do it, so that's how we can get around that to make it just much easier. Especially this early on when we're just building our database we don't necessarily need to have that level of constraint in order to protect our database quite yet. Okay, so one other last thing to mention here. So I was just typing in these data types. These define the types of data that can be stored in these columns. It also helps define things like how the data is treated when you interact with it from SQL or from other programming interfaces. And we also have on the far right here this Allows Nulls. This is basically saying can this field be blank? Can it not have a value for every row? And in this case we're going to say no to OrderDate and OrderAmount, but some of the other ones are allowed to be blank. Again I'll hit Save, it'll recreate that for me, good to go. Okay, so now let's try to recreate that relationship. We have ProductID here, and I'm just going to simply drag that over to ProductID on top of Products. And it popped up this dialog, which does the join for me or creates the foreign key relationship. So there's the primary key, which defines a table's granularity, and then there's the foreign key, which is the one that it's linked to for this relationship. It got it correctly. All of these settings here are a bit more advanced. I'm not going to get into those now, but later I'll point you where you can find those. And now look at that. We have this really interesting symbol here, and this is how SQL Server chooses to do this where it has a little key icon and this little mini icon here. And you can click on that. You can go in and edit the relationship. You can understand how it actually works. So let's just create a couple more here. We have CustomerID, drag that over to Customers. It gives me that. Good to go. EmployeeID goes over to Employee. There we go. There we go. So what we have now, and I'll try to fit this, I apologize for the resolution, it will actually allow us to have a data model that we can join things together, and we can enforce what's known as referential integrity. The system basically is protecting itself from harm by making sure that these relationships do exist. So that'll come up when you're trying to insert data into orders, and if you have a ProductID that doesn't exist in the Products table, in SQL Server's world it would actually give you an error, so that's where these relationships really, really do matter and are a big part of the physical design of how the database works. We'll go ahead and save this, and we're good to go. We have a legitimate database model that we built, and it has been implemented as well in our database. If I go refresh our tables, you'll see that I have more tables in here now. There we go. I have the Orders table. So we could fill this up with data, we could build an application, we could actually build systems that rely on this now that we have a basic system here that would take orders for us.

Where to Find More

So where can we find more? Well, on Pluralsight we have a lot of courses on this. We have this Practical Data Modeling with C# and SQL Server. Great course if you are a. NET developer. We also have this Logical and Physical Modeling for Analytical Applications. Now we're going to talk a little bit about that with data warehousing in the next module, but this is just a full course on that. So if you're in that space and you're doing the modeling work, this would be a great way for you to dive in. Also we have the Introduction to Data Warehousing and Business Intelligence course. We'll talk about little bit about those concepts in the next module, give a light intro, but that is a full course that actually goes into the detail of how data modeling works and just data warehousing and business intelligence in general work.

Data Warehousing Basics

Introduction

Welcome to this module on Data Warehousing Basics. Here we're going to learn what data warehousing is, why we would use data warehousing, how it really works, we'll cover some of the key concepts so you have a basic understanding, and then I'll show you where you can find more info. Let's get started.

What Is Data Warehousing?

So what is data warehousing? Well, it's the process that we use to collect and organize disparate sources of information for analysis. In many systems and many companies there are different systems that house all the different types of data and information. In Cogsley Services we ask questions, and we're thinking about sales by industry, we're looking at average days to complete by manager, and we're looking at hourly wage to bill rate ratio. Thinking about each one of these, there are some measures here, the sales, the days to complete the wage, and they are sliced or segmented by different types of information, different contextual pieces we want to add, so each one of those pieces may come from a different system. In an ideal world all of our data would live in one computer system that generated everything and it was perfectly clean, but in reality that's just not the case. Especially in today's world it seems that it's really easy for a business team to go to a website, click a couple of buttons, enter a credit card, and be off and running with a new system housing data that is critical to your business success, but isn't managed or even considered by other teams, and this is a problem and an opportunity. It's one that's difficult for people to wrangle and get the data together so they can make it usable, but it's also an incredible opportunity for businesses to move fast and deliver things much quicker than they were ever to before. So considering this perspective in mind, we have data living in different places, how are we going to do this? Well, that's what data warehousing is. So see we start with our sales system let's say, and then we have our HR system so if we wanted to know our sales by department. And then we have our company website, which maybe tracks web visitors or users. And then we have our accounting system, which is the one that actually has the financial data, the actual numbers of what people paid over time. Maybe there was a dispute, maybe there was discount, maybe there was a return. All of these things occur in different systems, and this is just a basic idea of an order. And all of these things can flow into a single place called a data warehouse. Once they're in there they make analysis really easy because we have all the data collated in one easy spot. Think about like going to the grocery store. If you had to go to 15 different stores for every different thing, it would make it much more difficult. And what we're trying to do is build that warehouse essentially of all the different data that you're going to need so that when you do need it or you have questions or processes that rely on it you just have to go to one place and it's all there. It's all nice and neat and clean and ready to use.

Why Use Data Warehousing?

So why would we use data warehousing? Well, it opens up these opportunities to gain value from your raw data through this process of analysis. So think about that for a second. There's raw data out there in these systems that's not collated, it's not normalized, it's not cleaned up, state names aren't matching, maybe user IDs aren't matching, those kinds of things. And we need to do that. We need to go through this process to gain value from it; otherwise, all the analysts and all the processes that are dependent upon it are going to have to spend the time themselves. So instead, what we do is we try to offload all of that heavy lifting onto the data warehousing team so they can get it done for us, and then we can just get the value out of it, you know the food is ready to eat so to speak. So in a typical scenario if we didn't do data warehousing it would look something like this. Each different system would produce its own set of reports and analysis, and this is how a lot of systems operate. It's kind of basic. They need to produce those systems anyways. And then if you wanted to call all that data together and you wanted to understand let's say the sales by the department where you needed to combine sales and the HR system, you'd have to try to cobble it together after the fact, and that makes it really difficult. In fact, what happens is sometimes people try to send the data between systems so HR data lives in our sales system and vice versa, and then what happens often if two people come to the meeting with different sets of data trying to answer the same question, and just confusion ensues and distrust of the data and overall just a breakdown of productivity. That's what we want to avoid, and that's why we do this is to help the consistency and the cleanliness and the accuracy of the data be here and true so everybody can benefit from it. So in the new world the data all flows into one, and then you just have one place to do your analysis. And it doesn't mean that things will always agree, but it does mean that people will at least be drinking from the same pond so that there's no confusion about who has the right numbers.

How Data Warehousing Works

So how does data warehousing work? By collecting data from multiple sources and reorganizing it into these simple structures, analysts and programs can leverage it for better decision making. That's kind of the key we're trying to get after here is better decision making, trying to make it so when we want to say spend more money on a marketing campaign we can analyze our past campaigns and understand maybe what was successful and what wasn't, make better decisions about how to spend money, about how to hire, about how to organize, about which projects were fruitful, which ones weren't. All of these things are truth seeking measures that are validated and backed up by data. So it starts like this. You have different types of data happening in every business. It doesn't really matter who you are. Everybody has these external events. Think about social media. People are talking about your company out there, and you probably want to participate in that conversation. You have business systems that run your business. You have a way to take orders. You have a way to fulfill services depending on what your business is. You probably also have some internal systems. This could be something as simple as an Excel spreadsheet with some little macros that you built, or it could be more complex like an internal website that you built to schedule vacation let's say. All of these things are just different general categories of types of data that your business probably needs in order to actually function or understand what's going on. So the process here of how it works is we take that data, and we feed it into a back office system. This back office system is just like a database just like the ones we've been working with to date. It's just different designation. So this one we know isn't really ready for prime time; people aren't going to be hitting this. This is more like in the back of the kitchen of a restaurant where all the vegetables and everything are being prepared. You don't really want the customers back there yet because it's not ready for them. So once the data warehousing team does process that and make it ready now you have the front office system. Now it's ready to be consumed, and out of that we get all of the analysis like targeting our customers, looking at the business monitoring of what's happening, as well as things like our strategic decision making. These are just some basic examples of the things that we can actually do with this data. So in the broad stroke categorization here these things on the left are called data providers, in the middle we have the data warehouse, this is what we're talking about in this module, and on the right we have some use cases. There are many more, like I said, but these are some common ones that seem to be true for all different businesses that I've worked with.

Key Concept: ETL

Now let's talk about this key concept of extract, transform, and load, ETL for short. In our diagram where the data is just seemingly moving around, the extract, transform, and load process is the one that actually facilitates that movement. What happens is these systems that are computer systems that have code and ways of connecting to different data sources such as your external events, your business systems, your internal apps, pulling that data out, and landing it in your back office system. Then you have routines that you can either run in that back office system or through your ETL frameworks and your ETL software that move that data or cleanse that data and put it into the front office system. And then from there it's more on the analytics side. But let's say you had different places where that data needed to go. Your ETL system would be the one that would pipe that data and move it from the front office into whatever other system needed it. So this is a key, key piece. Without this none of it works, and it has to be built in a way where people understand how to move the data properly and effectively and efficiently. In general, the three broad categories or steps are first to stage the data. When you stage the data, what you do is you basically pull it out and just land it in your system. I prefer or I recommend to people not to do anything to the data in this stage. All you do is pull it and land it in your system. This also begs that you should have a system which can support all different formats and all different types of data. As systems grow and as the evolution of technology has turned, it means that different formats are to be expected, so this is where often people opt for NoSQL solutions because they are the most flexible in storing any type of data. Then you have this cleansing process. This is where we look at things like state name, for example. I could have two data sets, and if I wanted to join them on the name of the state, and one of them for California listed it as CA and the other one listed it as Calif. for some sort of abbreviation, I would need to somehow match those up so that when an analyst wanted to join these data sets those fields were all clean and neat and actually lined up properly. So this is cleansing phase. This is where most of the heavy lifting occurs. And then this is all back office at this point. Now we move it to the front office in the delivery stage, and the delivery stage is where the data actually gets presented to the analysts. This happens in the form of what are known as star schemas often. We'll get into that in just a second here. In general, those are the three broad steps of the extract, transform, and load process, and it's a critical piece, and I'll point you where you can find more information to that at the end of this module.

Key Concept: Dimensional Modeling

Now let's take a look at some dimensional modeling. This is another key concept in data warehousing. Dimensional modeling is designed to reduce complexity. If you recall, normalization was the kind of opposite of this where what it did is it reduced redundancy, and that's for transaction processing efficiency. Here what we're trying to do is make it most efficient for analysts to get answers to their questions, so we can't have a really complex design that slows them down. It needs to be simple and easy for them to use. It also provides a historical view. As accounts for our customers or as our products change over time, if we want to accurately report on what that was like in the past, we have to keep a historical record of what that is. This happens through a process known as a slowly changing dimension. And I won't get into that in this module, but I will point you to where you can find more information about how that process works. This dimensional model also enables analysts. This is essentially how they interact with the data. It is the design of your database that they interact with. It's also reusable. One thing that's great about dimensional modeling if you do it right is that in the initial stages it's going to be really slow to build, and the cost is going to be high, but over time you're going to be able to reuse all of those things that you built hopefully, if you did it right, and you'll be able to then move much faster in the future. So let's talk about this dimensional modeling. What is it? Well, it's a technique that it focused on analysis or Business Intelligence. In the world of Business Intelligence this is a key, key concept here. And why would we do it? Well, it's a great way to enable the analyst to be effective. Like I said, it simplifies things, it gives them access to clean data, and we do it by creating fact and dimension tables creating what are known as data marts. It's like a miniature data warehouse, and some would argue that a collection of data marts is effectively what you could call a data warehouse. The semantics here aren't that important. I think what's more important is to understand how these things are designed and built, so I'm going to jump in and show you that now. So let's talk about facts and dimensions real quick. A fact or a measure is generally a numerical thing that we want to aggregate. It's the subject of our analysis I like to think of it as, such as Sales Amount. We want to do a sum or an average, a min, max, median or something like that on one of these fields, one of these attributes. Think of it like a column from our spreadsheet that you would use in a pivot table. Well dimension is kind of the opposite. It's the context by which we analyze our facts. So if we're looking at Sales Amount, we want to see Sales Amount by industry or by customer. The customer or industry by which we're analyzing, that's the context of our analysis, and those are what are known as dimensions. These are generally pieces of text that are categorical in nature, not necessarily things that are continuous such as a time series, but you can think of them as just two key ingredients for good analysis.

Key Concept: Star Schema

So when you combine the facts and dimensions, what you come up with is known as a star schema. And a Star Schema is also known as a data mart as I mentioned, and this is the key concept or key tenant behind Business Intelligence. It's creating these data marts to then analyze them. They're very simple, but they are sustainable if you do them right. The data modeling effort can be quite cumbersome up front and quite costly, but it will yield a much greater return down the road as you're able to reuse these data sets. The data marts are also pretty flexible. You can answer lots and lots of questions with them, so let's get in and build one. So looking at our star schema we have facts and dimensions as I mentioned. We have our sales fact for example, and if we wanted to analyze our sales fact we could add something like date, a date dimension. With the date dimension, we could look at things like quarterly sales, monthly sales, day of week sales. Any different types of options for a date we would put in that date dimension, and then we'd be able to join it to sales based on a key and then actually see the different cuts of the database on date options. We also could look at staff, thinking about who actually sold something, who fulfilled it. There could be multiple joins here. We could look at shipping information if we were in the business that actually required shipping products. So we could do something like shipping line, shipping method, days to ship, shipping container, etc. You could have a whole list of characteristics about the shipping part of the sale that are listed in a single table just one join away. And then thinking about it further we could add product. Of course it makes sense to look at the sales and the different product lines we have. Maybe in that product table you have category, subcategory, product type, those kinds of things. All that information pulled together in a single table joined directly to sales allows us to analyze it in numerous ways. And lastly for our example here I just had customer, thinking about okay it probably makes sense we want to look at who bought stuff and who we sold it to, if there were returns or anything like that, and we could add that to our analysis here. So if we look at this design and this layout, and of course it's kind of this way on purpose, you can think of it as a star. It kind of matches right up with it. So this is where the concept or the name of star schema came from, and it's a key tenant in data warehousing that you should be familiar with. I think in more modern methods and other situations in tech companies where some of these traditions are kind of left behind or reevaluated you may not find this as prevalent, but I believe it's key or still relevant for people to understand where these things came from even if it's not exactly how you're using or designing your system today.

Demo: Building a Star Schema in SQL Server

Alright, so here I am in SQL Server Management Studio. If you recall, we have our CogsleyServicesDB we already built earlier, and now I'm going to create a new database called CogsleyServicesDM for data mart. One of the interesting things about when you build these is that you don't necessarily just want to take the database you have and figure out how to model that differently. You want to start by actually analyzing how your business works and what those facts and dimensions may be and then work backwards and let the ETL process handle the process of transforming the database that transacts or records the transactions into the star schema that you build. So the act of building a data mart is very similar to the act of building any other system other than you're starting with that data model. So here what we're going to do, I'm just going to start by doing a database diagram, and when I click that it's going to say would you like to install these bits? Yes, I would. And I'm just going to say New Database Diagram. We don't have any tables here, so I can just hit Close. And for our resolution's sake I'm going to hide the Object Explorer. Here I can start adding some tables, and the first one I'm going to add is my SalesFact. You'll find it common that in data warehousing or in data marts you either start or end a table name with the type of table it is, such as fact or dimension. Now it also could be common where people do something like this, f\_sales or d\_sales or customers, that kind of a thing. Now it depends on, I guess, how you think about it in terms of the design or the naming conventions. If you have other systems you've built, you might want to stick with those so people that use it are also familiar with this. I tend to be more on the analytics side, so I prefer things that are a bit more self explanatory. So I'll start with something called SalesFact. Hit OK. Now a common thing we do is we have an ID. This is called a surrogate key. It's just a random ID for every row that gets written here. So we have SalesFactID. We'll just make that an integer, and we'll go ahead and click on this guy and make it the primary key. So that's just an increment it every time a row gets inserted. Now we'll have all of the dimensions that we wanted to add, so we'll add something like CustomerDimID. This is an int. ProductDimID. This is also an int. EmployeeDimID. And I'll pause on this one second. So employee could be referenced multiple times here, right? We have a person that maybe sold something, and in CogsleyServices' world we're a consulting company, so we may have a primary consultant again. So you can add the same table basically to your star schema multiple times, but when you have a separate join you're going to have to have a separate key to join on. So EmployeeDimID isn't self explanatory, right? So let's say I wanted to add SalesPersonDimID, and then I wanted to add ConsultantDimID like that. That would make much more sense since we have two roles essentially that the employee table, the physical table which has all of our employees is actually playing in this model. So when you think about that it's really important to model your data after the roles and the business types and those kinds of things, not so much based on the physical design of the transactional system which maybe you're pulling your data from. I'll add a DateDimID to joint to our dates so we can see that. And again, we have to think about the role that this play, so that'll be OrderDate is what I'll call it. And then what I'll have here is a DeliveryDateDimID int. Now I'll add a few bits about the sales here, so we'll look at SalesAmount. So this is actually the first column that actually has data in it about the sales, and for this example I'm just going to leave that as the last one. So essentially all of these other fields here are just joins out. They're just keys referencing other dimensions in our design that we haven't built yet, but you can see how basically a fact table is a very simple way to just log transactions that occurred. I'll mention that there are other types of fact tables. They're much more complex, and they get much deeper into the idea about how to model these things, so I'll point you where you can learn more about that and leave it there for now. Okay, so we have our SalesFact. Let's shrink this guy up, and I'll go ahead and just move him over to the left. Now I'm going to go ahead and add a new table here, CustomerDim. And when I create this what we're going to have here is we're going to have the CustomerDimID, which is just an integer, and it is the primary key for our table, so we'll go ahead and set the primary key. Now we also probably have a system that has an actual key or identity for the customer, so I'm going to add a field for CustomerID. Now CustomerID here in this case should line up with whatever the source system was that houses all of our customer information, whereas we're creating one called DimID, which is our own what we call a surrogate key. It's our own additional abstraction, our own additional identity just for the data warehouse here. And the reason we do this is because we may have something here like multiple versions of a customer. So the reason we would have multiple versions of a customer is maybe let's say that the billing address changed and that was significant for us. Well, if we had a field here, let's call it just Address1, and this is varchar(50), and then we had City, it's another varchar, State, and Country, well here what we could do is we could also have a StartDate and an EndDate. And so what this would basically allow us to do is have multiple records or multiple versions of this same customer, and we would know that because the CustomerID, the business key as it's known, would actually be the same, so we could say oh this is the same customer, customer A, but they have different attributes here, let's say City and State changed, so they would get a new CustomerDimID and a different Start and EndDate. That's the concept behind a slowly changing dimension. I'm going to leave this here for now, and just know that is the kind of key concept here about how a data warehouse can track historical changes to your things such as customers, whereas most transactional systems only care about the current value of that. Now this becomes really important when you want to look at historical analytics. If I want to see sales for a different sales team a year ago, I need to know all the sales people that were on that team a year ago, and that team may look very different today, so that's really a key, key thing that this data, this concept adds value to your business to be able to look back in time and have an accurate understanding of how things looked back then. So I'll go ahead and leave this here. I'll add one more field, CustomerName, and go ahead and call that one done. I'm going to add Product now. I'll add a ProductDimID, call it an int, right-click on the field, and say Set Primary Key. I'll have a ProductID, I'll have a ProductName, ProductCategory, ProductSubCategory, ProductDescription, call it ntext, which is just another data type here. Similarly we could have a StartDate and an EndDate. As our products change, product lines, we can get new IDs and look at things historically and how they were, so it's a good way to design that table there. You may notice that we could also do something like move ProductCategory and SubCategory off to a separate table because they themselves have a relationship and save space. And this is where there's a lot of debate about the right way to do this. For simplicity sake for our analyst, it's going to be certainly easier to just have the data all in one table, but for maybe the physical design we could actually spread that out into multiple tables to make it easier to manage. It will actually store less data, so it would be more efficient. Then we could do something like create a database view, which combine them into this dimensional or this logical view that we're looking at now. So there's a lot of debate around that. That's just something to point out that if you were looking at something and it wasn't quite designed this way, that may be why, that there are other considerations here. Okay, so we've got our product table, we've got a customer table, let's build our employee table. Similarly we'll add EmployeeDimID, interface. This is a primary key. We'll have our EmployeeID. This is the source system ID. Have EmployeeFirstName, EmployeeLastName, EmployeeStartDate, EmployeeStatus, EmployeeTermDate. And then you could also have something here like ManagerID, which would join back to the same table. And the way we would do things here maybe is you'd have a ManagerDimID, which would join back to the dimensional key because there could be multiple versions of an actual employee just like there can be multiple versions of a product. Let's say somebody changes their name, for example, and that was a significant change that you wanted to track, you could have multiple versions for every time that occurred for any employee that's in your company. Okay, we're getting there. So we've got three, and this one we're going to join to twice, so I'll show you how that'll work in a second. So the last one for us to add here is actually the date dimension, so DateDim we'll call it. And here's where there's a bit difference, and dates in general are pretty special. So instead of having a surrogate key, just an integer that randomly increments, because dates don't change, for example February will always be the second month of the year, we don't have to have this arbitrary other abstraction layer because there won't be ever two records of the same date. The same date will always be the same date. So often what people do they'll have a DateID, and this will be an integer, but this won't be the random number. It'll actually be an integer representing the date. So, for example, 20140101 would be January 1st of 2014. And you can use a format like that and very easily create joins across all your different database tables and all your facts that need dates, so that's kind of a common thing people do here. So for that we'll have DateID, then we'll have the actual Date, then we'll have maybe Month let's say, varchar, and Quarter varchar we'll call that, Year. This could be an int because it's a whole number. And you could add lot of things. You could add FiscalYear, add that as an int. You could add all those kinds of things. You could add IsHoliday and all that an int for true or false or 1 or 0. You could add all kinds of things. You could say is it a company holiday, is it a federal holiday, etc. So in your dates this can be a really rich dimension, and there are lots of other examples online. You could probably go find one that fits kind of what you're looking for and just use that as your example, but I'm just going to leave this here kind of as a very basic one so we can get going on our star schema. Okay, so we've built all the tables necessary. Now it's time to link them all together, to create those relationships. If you recall from the previous module we did that. So I'm also going to right-click on this guy and say Set Primary Key. And I have all of my dimensions here. What I'm going to do is I'm actually going to save this diagram, and I'm going to call this SalesDM for data mart. It's going to build these tables for me. There we go. The little asterisks disappeared from all of them. And now that we're there we can actually do some joins. So because my resolution is tough here I'm going to shrink this down quite a bit. Maybe we'll go to 75. Alright, so you can still kind of see it. So try to make this look like a star just like we did before. So let's look at dates, and this is where what we're going to do is we're going to drag OrderDate over to DateID, and it's going to create the relationship for me. It did it on DateID OrderDateDimID. That's cool. That's what I want. I've got that. I'm also going to drag DeliveryDate over to Dim DateID and hit that guy so now I have two relationships, two joins going on there. Then I also have customer, so let's add customer in here, so CustomerDimID drags over. Building everything for me. This is pretty awesome. I'm going to drag this guy over. Maybe I'll drag a whole lot of these tables over here. So we have EmployeeDim, so this is where I'm going to have ConsultantID joined up to EmployeeID. And then we'll also have SalesPersonID, so we'll have a multiple join here. There you go. And we'll drag our customer one down, so that's there. We've got our dates up here, and we've got our products down here, so we'll add product's ProductDimID and ProductDimID. There we go. Okay, so let me just try to fit this to the view so you can see what's happening here. I apologize for the resolution. But hopefully this makes sense that you have your SalesFact, or in general your fact tables are kind of at the center of your design, and then related to those are all the different dimensions. So if we wanted to enhance this later on we could add new dimension keys inside of our SalesFact table and add those dimension tables as necessary. Similarly, if we add another fact table that say needed employee information, I could just use that same EmployeeID and actually join to that table there. So these assets become reusable, which is how they add value over time. But like you may have noticed here, having to build them all from scratch the very first time is kind of costly, and that's what seems slow, but overall in the end it becomes much faster and much better as you kind of collate all the different data sources, and in the end hopefully you have a really complete picture of your business and how everything fits together.

Where to Find More

Okay, now let's take a look at where we can find more info. Well, we've got a bunch of courses on Pluralsight. We've got this Logical and Physical Modeling for Analytical Applications. It goes through basically all these bits, but just in a lot more detail. We also have the Introduction to Data Warehousing and Business Intelligence, basically a whole course covering this same type of topic that we talked about in this module. Then we also have this Optimizing OLTP and Data Warehousing with SQL Server 2014. OLTP is that transactional processing system, so it's the opposite of what we've been doing here. It's the more normalized version built for a transactional system like ordered processing, that kind of a thing, and the data warehousing side with SQL Server 2014, which is the current version as of date of production.

Big Data

Introduction

Welcome to this module on Big Data. I'm Ben Sullins, and today we're going to learn what big data is, why big data exists, how big data works, some of the key concepts behind big data, and where we can find more info. Let's get started.

What Is Big Data?

So what is big data anyways? You've probably heard this term all in marketing materials and all kinds of blogs, but I like to set it out there that big data is this set of technologies and the methods that we use to handle these large volumes of data. We need to handle them at rapid speeds and in various formats. These are known as the three Vs, volume, variety, and velocity. So big data really isn't one thing, it's many things, and I'll try to explain some of what that is here, but think of it more as the evolution of data processing systems. We've been talking a lot about databases to this point, and this is just the evolution of that to handle these new types of data that are coming in faster than ever and are at much larger volumes than ever. The idea is that some events are happening in real time, and you need to capture and process that data while other data sits there at rest. It's sitting there waiting to be picked up. All of these will flow into some sort of system. It's a whole ecosystem generally that handles these multiple types of data. Sometimes we'll stream that data in; other times it'll come in batch mode. A popular trend nowadays is instead of doing real-time streaming we're doing micro batches, so think every two seconds we'll write a batch of new data coming in versus doing a pure real-time stream. It's much more efficient and a lot more cost effective, so that's a new thing that's really popping up as a trend now. On the other side, our systems need to be able to support analysis and SQL and tools that will connect to these big data systems. We need to build applications and use APIs. An API is an interface, an application programming interface that a website or another application will use to interact with another application. And our data scientists need ways to interact with the data as well in terms of using different tools to explore the data, as well as tools to develop machine learning algorithms. Think of that as when you go onto Amazon or Google there's this suggested search or people that bought this item also may have bought this item. We suggest or recommend that. Those are all the machine learning tools that just get better and better. It's a light form of artificial intelligence if you can think of it that way where the system itself is becoming smarter about how to recommend things or how to do whatever function it has. All of these things need to be supported by a data system, and this is the ecosystem that is generally known as big data.

Why Big Data Exists

So why does big data exist? Today's technology generates incredible volumes of data in a variety of formats needing to be processed in frequent intervals. Try to encompass all of that into a system or an ecosystem of tools that can handle that. Think about data variety, for example. Well, we have web log information, we have sensor data, we have multimedia information, and we have cloud applications. See years ago data wasn't so prevalent everywhere in every aspect of our lives. Business data specifically was generally very curated and highly consolidated into a few systems that had all the key pieces of information that you'd need to run your business. Well nowadays, especially considering things like Twitter or Facebook where external parties are generating data about your company, they're talking about your company and you cannot control that, so what you need to do is you need to be plugged into it. You need to be able to analyze that information. Web logs are a way where our web server can log and track things much quickly and easier than building every little thing we want to track into the website, so those can be parsed into meaningful bits of information. This also comes across as text usually. Sensors, if you're in the manufacturing industry, obviously in the manufacturing line there are sensors all over the place, how hot certain parts of the system are, how many transactions per second are happening or how many things are getting stamped. All of those bits can be coming in in real time, and we may need to react to them automatically. Think about what would happen on a manufacturing line if one piece started failing and you didn't have somebody there to either shut it off immediately or a computer system or a machine that would automatically recognize that it needs to be shut off to prevent other failures or other catastrophes from happening. Sensor data is very fast, and it's generally very small, little bits of data streaming in, but very quickly, and you need to respond to them very quickly. With regards to multimedia, there's a whole new world here. Think about how Pandora, the music streaming service works. It will recommend new content for you based solely on your first input of a station, an artist, or a genre, and with you and the millions of other people using their platform it comes up with a way to analyze the audio itself to determine what other songs you might like, and it does this with an incredible precision. Cloud applications are kind of the latest one to come into the realm here where businesses are able to spin up new systems and just start plugging away and running a new part of their business using a cloud-based application. These are things that the business doesn't necessarily need IT for; however, that data is still valuable and relevant to your decision making and to understanding how your business or how your system is operating, so we really need to make sure that we're capturing that data for analysis and bringing into a way so we can look at things holistically. The other challenge is that why big data exists is around volume. Think about Facebook, for example. Facebook has one of the largest databases in history, and it's arguably one of the most valuable from all kinds of different angles from studying societies to understanding psychology of humans. Think about every time somebody shares something on Facebook. They get 2. 4 million shares that happen per minute. That's per minute every day 365 days a year. That is a huge amount of data that's being generated on their platform. YouTube similarly has another huge number here, 72 hours of video. This is how much new video is uploaded to YouTube every single minute. So think about the data volumes, and they're dealing with video, not just things like text or links or those kinds of things. They're dealing with actual multimedia files. And then of course Google with 4 million searches per minute. So every time somebody searches for something they need to respond with a suggested answer, they need to run their algorithm, and they need to do all these things to get exactly the right answer for you in the milliseconds that you expect it to load in. All of these are happening per minute, so the volumes of data are astronomical, so the systems that you build are massive and need to scale, and it's this whole idea of distributed computing, which I'll talk about in a second. And the last thing I'll mention is the velocity of data. There are lots of examples here, the stock market obviously, things happening in millisecond transactions, manufacturing with all the sensors on the assembly line and those examples, business events happening where possibly something goes down or somebody buys something and you want to react very quickly. And telecom providers have been in this game for a long, long time thinking about call volumes and staffing and where to route internet traffic and those kinds of things. All of these things need to be happening and responding to in real time, and so the velocity of data is important that we're able to respond to that and work with that. Well, traditional data systems couldn't really handle this very well, and this is where a lot of the idea of the velocity that we're trying to capture with big data system's needs have come into play.

How Big Data Works

So how does big data work? Well by handling the data in any format in real time at large scales, big data systems can now meet these needs of current technologies. Let's take a look at this. So there's a concept here, and this is one way of looking at it, there are many as with all technologies, that you have an enterprise data hub, and inside of the data hub you store your raw data. So as I mentioned the real-time streaming data coming in, you want to land that data in there for archival purposes and so then you can work with it. Once the data is there, you can build a data warehouse, and a data warehouse, as we've talked about in previous modules, is the idea about aggregating the data and modeling it and storing it for easy analysis for decision making. Well, that decision making could happen in real time in milliseconds based on things like stock markets and those kinds of things, or it could happen over a nightly process or quarterly meetings, those kinds of things, but that data hub, this big data system will have both of these types of data in there. We'll also have real-time APIs, ways you can interact with the data in real time, and also it'll support machine learning. This is the algorithms that we generate to have the intelligent answers or intelligent products that are helping us just make things much more efficient and convenient for our customers and our users. Some examples here would be pulling in data of all types, the system logs, web logs, databases, flat files; business system data itself, so an HR system you could imagine; event streams, so if you have your website or you have other events happening out there you can pull that data all in; and then you can interact with it with an API or a SKD, which is a software development kit. It's like toolsets for working with the data hub. The data science team can do the modeling in here and actually coming up with statistical modeling and predictive analytics and all these kinds of really awesome tools that help us just better our business and better our product for our customers. And then our analysts. They can interact with it using SQL. We've talked a little bit about this where SQL is kind of the standard query language that everyone uses for interacting with databases. And then business users can just do it directly with the different analytics that are generated out of the system meaning the system needs to support analytics for non-analysts, for people that don't know how to write SQL. And down here we also need to be able to build web and mobile applications. Of course we can't forget our admins who need to configure the thing and set it up with security and all that. So this is kind of a deep dive into the previous slide that just had the big globe looking thing for the big data. This is an example of what all those pieces might actually fit and how you can interact with it and how something like this is so diverse and can support so many use cases. It's also why a lot of companies maybe are hesitant to get into it because you know it's going to be a significant investment. You're going to have to spend a lot of time and money, and the value of it may be down the road because it's not something that tomorrow you can start selling. It's something that has to get up and running for a little bit before you actually start to realize the value, but this is an example here of how all the pieces would fit together.

Key Concept: Distributed Computing

Now let's talk about distributed computing. This is a key concept in big data, and it stemmed from Yahoo and Google, companies that were trying to index the entire internet and interact with those indexes. And so for this example, because it can be so complex, I'm going to boil it down to kind of the simplest form and talk about our story with Cogsley Services. So Cogsley Services started a new support service. This support service allows customers to actually get help about products just by simply clicking a button in their web browser. They can click a button and be connected to one of our experts available to help them with whatever their needs are. Well, to support this service, what we did is we stood up a data node here, and this data node handles all those requests. It stores the data, it does the machine learning, it processes everything that's happening, and then it seemed to work well. So as our support service grew over a year, we needed to increase the size of our data node to support the additional customers, the transactions, all the machine learning that's happening, everything there, so what the system is now is it's able to support it, but we have a key problem here, and this is the difference between the traditional approach of scaling up our systems versus scaling out or distributing the load. Here we have a single point of failure. If this data node were to die, the entire system shuts down, and as this has become a mission critical part of Cogsley Services revenue, we really need to make sure that we have a system that will support it no matter what happens. So instead of doing that, what we did is we bought a bunch of little systems, and now we have the load spread across all of our data nodes instead of just one main one. This is the key idea behind distributed computing. So that way if we had 10, 000 nodes in our system supporting it, distributing the load, letting us scale tremendously and we have a 1% expected failure rate, that means we're always going to have about 100 nodes that are down that are not working that are failing, and so we need to design a system that will be able to support expecting certain pieces of it to always be failing or always be not working as optimal as they could be. And this is something that is a key part about why we would use a distributed system versus a traditional system, which is clustered.

Key Concept: CAP Theorem

Now let's talk about the CAP theorem. The CAP theorem is this idea that when you have a distributed system like this you can only have two of the three guarantees across the reads and writes. The C, the A, and the P all stand for something different, and I'll talk about those in a second. But you can only have two of the three. You can never have all three. And so when you design these systems you have to think about what is the intent, what is the user behavior, what is okay versus not okay. So first let's talk about the first one, consistency. So consistency means that when somebody writes or updates a piece of data, anyone else looking for that same piece of data gets the exact same response, the actual truth of what occurred. So in an example where somebody maybe bought something online and then somebody else was looking at a sales report, when they updated the sales report did all the most recent transactions, every single one of those transactions actually appear in that report? If so, then you'd have a consistent design. If it's not important that they are exactly in sync, that maybe a few second delay or a day delay or even less than that is okay, then you don't need consistency in your design. Facebook works this way in part. Some parts of the system are consistent while others are not. If I were to unfriend you on Facebook right now, you may still see me as a friend for a few minutes depending on our location and depending on how fast the system is updating at the time. It's not critical that those two things are exactly consistent, so they've sacrificed that of the CAP in the CAP theorem. The next one is availability. Availability means that the system will always return a response. There will never be an error or a timeout. The response will always be valid meaning the system is rock solid. It will always be there no matter what else is happening to the system, no matter what the load is, no matter if there are network failures or anything else that this system will be available and will return a response. The third one now is partition tolerance. In partition tolerance the idea is that your network itself of where all of your data is being distributed and how all the nodes are computing things is not guaranteed. No computer system is 100% guaranteed, so you have to have a system that can handle these faults. As I mentioned before, if you had 10, 000 nodes with a 1% failure rate, you'll have 100 nodes down at any given time. So what that means is that you need a system that can handle that scale and that partitioning of the data and of the load, so this is where the C the A and the P come from in the CAP theorem. So let's look at this in a diagram to maybe explain it a little bit better. So you have this triangle here with the C, the A, and the P, and remember you can only have two, never three. So on the left you could have a C and an A, consistency and availability meaning the system as long as it's up and running will always return a response in a reasonable time and that the data will be consistent. So any writes or updates that have occurred at the moment that the next read happens, it will get the most updated information. These are your typical relational databases: SQL Server, MySQL, Oracle, PostgresSQL. There are many others here. These are the standard databases that have been around forever. Notice the piece missing here is the partition tolerance. So what that means is while you can cluster these systems and have the load spread across many of them, they can't support partitioning. So that means that if one of the nodes dies or if several of the nodes die or if a hundred of the nodes die that the whole system will actually crash, so you don't have that fault tolerance, that resiliency. On the other side you have the availability and the partition tolerance. These are known as eventually consistent systems where if I were to write something, a new piece of data, or update something and somebody else were to come in a second later and read or try to pull that information out, they may not have the most up-to-date information, and that's okay in some cases. It's not always critical. Sales processing systems, it's not generally acceptable, but in a lot of other cases it generally is. So this is the NoSQL database that you're familiar with: Cassandra, CouchDB, DynamoDB. There are dozens of them out there. Those are some of the most popular ones. Now down on the bottom we have an interesting one where we have this consistency and partition tolerance where the things that are being read or queried may not always be there. There may be an availability issue some times, but the consistency will be there and the partition tolerance will be there meaning the system will live on. The whole thing isn't going to come crashing to a halt, but you may have little intermittent errors here and there, so you have to build the system to support that. Some of the really popular databases in this realm are MongoDB, Hbase, Memcache, and Redis. They all operate in a very similar way.

Where to Find More

Okay, so let's look at where we can find more info. Well, on Pluralsight we have several courses on big data. One that I've actually done before, the Big Data Analytics with Tableau talks a bit deeper about the architectures and then goes into actually analyzing data with Tableau against a big data system. And it's all fun and interactive. You can do that pretty easily and pretty lightly. The Big Data: The Big Picture. This is a good course that kind of talks more in-depth about all the things that I've covered here. There's an example working with Big Data and Reporting with MongoDB, as well as one on Amazon Web Services. Amazon Web Services is a hosted version of a lot of different infrastructure, and so this talks about how to actually run your big data systems on Amazon Web Services.

Visualizing Data in Charts

Introduction

Welcome to this module on visualizing data in charts. I'm Ben Sullins, and in this module we're going to learn what data visualization is, why we would visualize data, how it really works, some of the key concepts that make up this whole world of data visualization, and where you can find more. Let's get started.

What Is Data Visualization?

So what is data visualization? Well, it's this process that we use to take raw information and display it in a way that helps us more easily understand the underlying meaning, and you see this in regular life. Some examples I've come across on a daily basis are dashboards that show me things like my fitness levels and the activity that I've been on, my competition with friends. The names have been redacted to protect the innocent. This is a simple example here of a dashboard of a data visualization. On the top we see my number of steps by day, and there's a color coding there meaning the activity level or the calories burned, and it goes down by hour. I see some totals on the right. I see some trend lines around my recent heart rate and then my recent exercises and even a hike and everything there. On the top left I have these beautiful radial indicators that display some totals. They're really simple metrics. But all of this is data that in this case is being generated by a little fitness tracker I wear, and it's being displayed here in a way that just makes it easier to understand. This is a common thing that lots of people see. And you probably don't even realize it, but you see data all throughout your regular life nowadays. Another example is maps. Maps here really help us understand where we are. Also, things like this can help us understand where we need to be. All of these are not simple data points like in the dashboard we saw, but they are relative geo-spatial data points that are drawn for us that give us a sense of direction. They tell us where we're going, they tell us where we've been, they help us search and find new places to explore in the world. All of these are just different data points that are being displayed in different ways. This is one of the more creative, more intensive ways, but still a data visualization nonetheless. And of course I can't leave this segment without talking about some of the more famous data visualizations, this one here from Henrietta Leavitt. She came out of Harvard in the early 1900s where they were studying the luminosity of stars. And the thing there that she discovered in these trend lines is that by looking at the stars we were actually able to determine how far away they were. And the interesting thing about that is up until this point we on this planet had no idea how far away all these twinkling little lights in the sky were. We had no sense of understanding of where Earth sits in the vastness of the cosmos. That wasn't until Henrietta Leavitt put down some data points and graphed them manually. And the reason I wanted to show this and talk about the importance of this data visualization is because this in itself is a very simple graph, and it was done by hand. So you don't need complex sophisticated technology, you don't need big data, you don't need machine learning or statistical packages to understand how visualizing the data can truly benefit you and lead to all kinds of new discoveries that can help the world, that can help yourself, help your team. In whatever your situation is, data can be used to understand things that weren't prevalent there when all you saw was a list of numbers. When you visualize the data, it really brings through the underlying meaning.

Why Visualize Data?

So it may seem obvious, but why would we visualize data? Where did this idea come from? We visualize data because humans, we process visual input above all else and faster than any other method of input. This is the key reason here. So humans ourselves have been looking at patterns in shapes, sizes, and colors since we were cave people. Think about looking at the plains on the safari and being able to detect exactly where a lion could be approaching or where your dinner could be sitting. All of these things have trained our brains to be incredibly perceptive at understanding the slight minute difference in our visual input. This is why our brains have become able to process this information much faster than any other type of input, so no sensory, no auditory, no other form of input or information coming into your brain can be processed faster than visual information. Let's take a look at an example. Here we have some data, and we have four different plots. You have I, II, III, and IV in the Roman numerals. Inside of there you have X and Y variables. These variables have different numeric values, and each one of these plots doesn't share the exact same numbers anywhere. So if we look at this data, we can't really tell what's going on. Nothing here really seems to add up. It looks like a bunch of different numbers that are grouped together in different ways. Well, what if we actually look at the statistics on this data? Well, in fact they all come out the exact same. So the mean of X is 9 exactly, the variance of X in each case, the mean of Y, the variance of Y, all of these are if not exact very, very similar. The correlation between them, the linear aggression, all of the summary level statistics, all the typical ways that we would look at this stuff doesn't really tell us much. It's basically saying these two data sets are the same, and from a standpoint of summary statistics that's correct. What if we visualize this data? We if we took this data and we actually wanted to plot it on a graph? What would it look like? Well, if we summarized it, we would have this statistical visualization here where, again, everything is the same. They all add up to about the exact same value. Well, what if we actually then instead of aggregating it up and doing summary level statistics showed each data point individually? What would that look like? So now I see a very different story. All of these different plots come out to about the same result. No matter which way you slice it statistically, they're even. However, when you visualize the data, there's an entirely different story being presented. I think of this as, you know, the team that won the championship in the Super Bowl and how they got there is incredible. The summary is they won. They won championship. They won the Super Bowl. They won whatever they accomplished, the same thing as everyone else that has accomplished and came before them; however, their story, their path on which lead them to that achievement is all unique to them, and this is an example here of that exact data being visualized and telling that story. This is why often when you research this topic people talk about data story telling or data visualizations or narratives. It all stems from this idea that when you truly see the data you can understand something new and different that you typically wouldn't get from just summary-level statistics. This is called Anscombe's Quartet. For instance, Anscombe came out with this in 1973, and he was an economist that had this idea about look the numbers you need to look at them in a way because if all you do is run summary stats on it again you won't get the true picture, the whole story, and he was right. And this where this entire world of data visualization comes from is this idea that when you can see the data you can understand it in new ways that wouldn't be possible otherwise.

How Data Visualization Works

So how does data visualization really work? By plotting the data points on charts and graphs, this is now typically done by software, the data takes shape for your eyes to consume, so we take the data and we plot it. Now I'm going to show you can example of this in the next module where we actually build something, but the basic workflow is you have data living in a database or a spreadsheet or wherever, and that data flows in, and an analyst, a person, and I want to emphasize the importance of a human in this process, takes it and then visualizes it, and typically this is done to help with some business question or some monitoring activity or trying to understand something because, again, that raw data is almost impossible to comprehend, especially when it gets large and you have millions of data points or potentially billions of data points. The only real way for a human to consume that information is visually, and it's important to note that a person will always have a better sense of intuition and different ways of looking at things than a machine will. Now machines can run algorithms on things that a human couldn't as well, so there's kind of this balance of machine learning or statistics and running models on things versus a person with their exploratory nature and their intuition about a business or a problem or anything like that. That experience you can't really program into computers yet, so a human is definitely a value add in this whole workflow.

Key Concept: Visual Perception

Okay, so let's talk about one of the key concepts here, visual perception. Visual perception is how we as humans try to look at things and consume them, and this matters a whole lot when you're talking about visualizing data, so let's do some fun examples here. Go ahead and count how many times the number five appears in the chart below. Okay, did you get it? Now let's reformat this data, and let's see if we can answer the question again. That time was a lot easier, right? So what we've done here is we've simply modified our display to help us get to the answer that we were seeking much faster. So think about when you're making charts for your business. This is an easy one here of emphasizing the points of interest that help us just quickly and easily, it's unavoidable, you cannot get away from answering this question. It's almost like the rest of the data disappears, and your answer is just staring you right in the face. This is the goal for our data visualizations. We want people to have a question or have something they're trying to understand look at a visual representation of that data and then have it instantly grab them to where the answer is so obvious. Now this of course is a prime example, so not everything is this easy or this obvious, but think about that and think about the power of this example and compare it to the other charts and other graphs and other things that you've built for people in the past, think about how you'll do them differently in the future. And what this plays on are two types of processing. The first one is known as attentive, so this is when your brain has to scan essentially every line here and every row to find what it's looking for. In the database terminology this could be known as a table scan or a full table scan where every bit of data in a table must be scanned for that one piece of information you're looking for. Thankfully databases help us with that; otherwise, we would just spend our entire lives looking at raw numbers, which is no fun. So the other form is called pre-attentive processing. In this form we've already preprocessed it. It is impossible---in fact, before you can even read the numbers that are being displayed, your brain has already consumed the information because the visual input is much faster than the auditory input of you telling yourself or you counting in your head, so it's a really, really powerful tool to use and to know about, and this is the key idea behind visual perception.

Key Concept: Color

Now let's talk about color. Now we just used color in that example there where we were bolding something, but what about other uses of color? Well, remember this whole concept of visualizing data comes from human nature, and in nature we have a couple of cases in which color is prominently used. Here's an example where we have a Golden Dart Frog. Two micrograms, hardly any, can kill an adult. That is incredible, so nature has helped us out by making this color bright so that we can know that this frog demands attention. It's not something to be taken lightly. Any time you see a bright colored animal like this, it's often because there's something that demands your attention. Same goes for other types of color in nature such as fruit. Brightly colored fruits are often very, very sweet because they want us to consume them so we can spread their seeds and they can reproduce. So color in nature I used to draw our attention, and we as humans haven't evolved that greatly that when we look at a spreadsheet or a chart color is still going to be the thing that draws our attention the most. Here's a good use of color. So in this example the author of this chart, which was Stephen Few, and this is from one of his books that I highly recommend called Information Dashboard Design, this is the first version I prefer over the second one, the second one is just a lot harder to get through, the first version here had this example where he's using a red dot next to the Profit metric in order to draw the reader's attention there. Why is he doing that? Because that one is the one that's in the most jeopardy, and it's also the one that demands the most attention right now. Notice how everything else on this chart, on this graph is very subtle. It's very gray, it's very light, it's kind of in the background. That's on purpose. When you design these charts and graphs, always act with intention. Whenever you use any color or any symbol or any amount of decimal points or anything, think about what you're doing. Don't just accept the defaults. Think about how can I design this to where it just jumps off the page as it does in this example. Now let's take a look at an example of bad use of color. Here we have a similar type of chart where it's showing key performance indicators. In this case we're looking at profit ratio. And most people, and commonly this would probably be acceptable by most standards, a lot of companies would probably see this and think this is great. The challenge here though is that the green dot and the red dots are so crowded in by the yellow triangles that it's more difficult to actually pull those out. If you just got rid of them, it would be much easier to see so the red and green would just stand right out, and we wouldn't need all this unnecessary color.

Key Concept: Chart Types

Now let's talk about chart types. The basic ways that we look at data are these six categories of visualization here, geo-spatial such as maps; correlation, which is a great way to look at scatter plots; hierarchical data where you have maybe a product category and a product subcategory; the distribution. Think about home prices. This is a classic one where you don't necessarily want an average. You typically run off a median because there's that one 30 million dollar home, but really most of them are around 400, 000. You have the time series data, and you have categorical data. So these translate very well to typical types of charts, maps, scatterplots, drill-downs, histograms, line charts, and bar charts. And I'll put some caveats on here. The drill-down is more of a capability, more of a functionality than a chart type, and the histogram itself is just a way of organizing your bar chart. So really when you think about it there are kind of only four. There are maps, scatterplots, lines, and bars. The other two are kind of nuances to these. And of course there are many more advanced or complex ways of looking at it, but honestly in all the years that I've been visualizing data and helping people understand it better those four chart types can get you 99% of the way. It really---most business cases bar charts and line charts answer those questions. Scatterplots are another great way to look. And maps, if you're in any industry that really has a lot to do with geo-spatial data, let's say you're in an oil industry and you need to know exactly where oil rigs are and which ones are performing better than others, or if you're in real estate or something like that where the actual physical location matters a whole lot, maps are an excellent choice of chart to use.

Key Concept: Dashboards

Okay, now let's talk about the infamous dashboards. So dashboards are basically a collection of charts. And we already looked at one in the opening of this module, but here let's take a look at one that I built for this example. So on the top here what I have is a simple chart that is just a map, and it's showing me profit by state. I've color coded the state, and I've put labels on them and so you can see right there exactly which ones are high, which ones are low, and in between. Now if I wanted to correlate that data with category sales or anything else, I could add another chart down below. So let's take a look here, and we have Quarterly Sales. So on this it's very simple. It's kind of year over year by quarter, so really quickly I can see and notice the colors are all kind of agreeing with each other with the more recent years being the darker color because that's the one I want you to focus on, and the further back in time it kind of gets lighter and grayer. Depending on what industry you're in, but typically anything three or four years out from today is probably not nearly as relevant as the previous one or two years. And then of course we can add another chart down there by Sales by Category. This is just a very simple way, and this is something I built off of the data set we're using for Cogsley Services. So here I'm using color to distinguish between the product category, so the color isn't a heat map kind of a thing that is used on the Profit by State one. Instead, it's just simple way of distinguishing which categories so you can see the breakdown. This is an example of a bar chart, as well as that drill-down chart that I just talked about. So the simple example here is a dashboard where in one page we have lots of different charts, and we have lots of different data, and it's all being displayed in ways that really help us understand what's going on.

Where to Find More

Okay, so where can you find more information? This is a big topic, and we have a lot of courses on it, a couple of them by me, and one by Dan Appleman who has an excellent course there. So the first one, Business Dashboard Fundamentals, is a course where I talk through all the examples really in detail about color, shape, size, proximity, different chart types, when to use different charts and the concepts behind how to plan and how to build the dashboard and where to start and how to end up including a form that you can use to ask people to fill out. And then I build one in Excel with you, and then I build one in Tableau with you. So that one kind of takes you through soup to nuts specifically around dashboards. Data Visualization for Developers is a great one from Dan Appleman where he talks especially to the developer audience, which I imagine some of you are, and you're probably looking at okay data visualization it's just an easy way to throw data on there and make pretty charts, but there's really a lot more to it, and he kind of breaks it down for you as to why to use which charts, when, and all those kinds of things, so it's an excellent course. I highly recommend it. And then if you're looking to actually implement some of these things, you can take one of my other courses here, the D3. js. It's a JavaScript charting library, and if you already know what that is you'll be familiar with it. And this is a way for you to build those charts using D3. And again, I get a little bit deeper into which charts to use when. In a tool like D3 it's really important because you're going to do a lot of coding, and you need to figure out which charts and everything you want first before you actually get into the build phase because it's a lot of code to write, and it just sucks if you have to go back and change that often. Also, there's a book I recommend, Information Dashboard Design. I really like the first version. It's by Stephen Few. I highly recommend you going and picking it up. He also has a second version, but I would definitely start with the first one. It's an easy read, and it really goes through dashboard examples, and he talks about all the different nuances of chart types and everything. So if this is going to be a big part of your job and you really want to dig in, I would highly recommend that book. Even if you're just mildly interested, I think it's a good read and a good one to have on your shelf at your desk.

Building Charts in Tableau

Introduction

Welcome to this module on Building Charts in Tableau. I'm Ben Sullins, and today we're going to learn what Tableau is, why we would use Tableau, how it works, we'll go through so you'll get really hands on here, we'll follow our story with Cogsley Services, and then we'll point you where you can find more information. Let's get started.

What Is Tableau?

So what is Tableau anyways? Tableau is a complete data analytics platform enabling everyone to gain insights from raw data. The emphasis here is on the word everyone. It is a simple intuitive tool that anyone can use. If you know how to use Excel at all, you can use Tableau and elevate your use of data by visualizing it and exploring it in a much deeper more natural way. In the Tableau product family we have four main areas here. Tableau Desktop is the one we're going to be talking about, and we'll demo with a free version of it called Tableau Public. On the left we have Tableau Server. This is where you can share and you can create and organize all your content on a server. This is great for companies or organizations that need to have a centralized location for all their analytics or their data. You can also do this via Tableau Online, which is on the bottom here. It's the same idea as Tableau Server; it's just hosted by Tableau. So you no longer need to host any of it. You just pay them a fee, and everything is in the cloud. And on the right is Tableau Reader. Think of it like Adobe Reader where somebody can create a workbook for you or a visualization, and they can send it to you, and you can open it without having to have Tableau Desktop installed. Some of the key features and reasons why people use Tableau Desktop are that you can pull in data from anywhere. Really any data source that you can get your hands on you can pull into Tableau. Some of those are easier than others, but a lot of cases Tableau automates or makes this process as simple, as beautiful as possible. And once the data is in there, you can analyze it very easily and simply. It doesn't matter where the data came from. Once it's in Tableau, Tableau has its own engine to treat the data in a similar fashion no matter where its origin. Also, you can build dashboards, the combinations of different charts and analysis in interactive ways of digging in and really understanding what their true meaning is that the data is trying to tell you. And you can also do some advanced analytics functions in here, not quite as robust as some of the statistical packages out there, but enough that will get you to understand your data and look at it in a very legitimate and very scientific way to where you can come up with really close to conclusions or very close to causality and understanding how things are related to each other.

Why Use Tableau?

So why would you use Tableau? Tableau makes working with data universally accessible with our platform. This idea of universally accessible or democratizing data is one that has caught on in recent years, especially in the analytics realm. No longer do you need to send requests into a data team or a business intelligence team living within IT and get on some priority or project list. Now with tools like Tableau everybody can do it, and you can pull data from virtually anywhere. It also offers IT and whoever owns the data systems control to where they can actually point you and have the certified data sources in a Single Source of Truth, but give you the flexibility you need to really get your work done. Think of the workflow that you typically would go through. You'd start with some data either in a database or on a server or in a cloud system or data living in a file such as a spreadsheet or a comma separated values or one of those types of files, you'd pull that data in and you'd visualize it in Tableau, you'd analyze it, and now you want to share that information. You want to broadcast it to the world. Well how do you do that? Well, Tableau using Tableau Reader and Tableau Server allows you to do this. So what you can do then is you can publish this, and this data and these visualizations and the tools that you build really can be available on mobile, web, and tablets. There are editing capabilities, you can share this, people can interact with it, they can create their own copies of it or their own versions with their own filter sets. It truly is a really awesome for anyone starting out with some simple data sets to visualize it, understand it, and share it with everyone else that needs to see it and use it.

How Tableau Works

So how does Tableau work? Tableau's platform pulls data from virtually anywhere and allows anyone to explore and visualize it. Let's look at some details of how this might happen. On the left here we have some actual data sources, or these are symbols that represent data sources. You have anything from big data systems, the elephant there is kind of what that represents, NoSQL systems, cubes, spreadsheets, cloud applications like salesforce. com or Google Analytics, and regular databases, things you might connect to internally or externally even. All of that can be pulled and connected to live or extracted meaning that the data is pulled out of those systems and landed inside of Tableau's own data engine, and this is where the self-service visual data analysis happens. Inside of here you have your analysts and your data engineers pulling the data in and building it up and coming up with dashboards and things that people can use and these tools to run the business. You can then share that information out as a static file. This is how Tableau Reader works. You information consumers on the right here would need to have the Tableau Reader system installed, and then they could open these dashboards or these visualizations and interact with them just as you designed it for them. As well, you can publish and move this data over to Tableau Server. So we won't talk too much about Tableau Server here, just the notion that it exists, and this is where you really use Tableau as an enterprise platform. There's governance, there's collaboration, there are ways to distribute it, you have a security model, there's automation. So think if you pulled your data from a database and you did some analysis to it and some massaging or some calculations and then shared that out, you can publish that to Tableau Server and have Tableau Server do all that heavy lifting for you so you don't have to open up the reports every week and refresh them and send them out, that kind of thing. From here the end users have a much better experience or one that at least is much more fluid where they can access it from all kinds of different locations. They can subscribe to the visualizations over email, they can embed those into other wikis or other web pages such as a company intranet or that kind of thing, and you can also download the files. You can export to CSV or export to Tableau even. So these are the two main ways to think about Tableau. There's a desktop tool, and that's kind of where the authoring and creating occurs, and then there's the server one. Now the server is much more the enterprise feature, and as I mentioned before Tableau Online is basically Tableau Server hosted by Tableau, so you don't even have to have IT set anything up. You just need a credit card to get going.

Demo: Installing Tableau

Okay, let's dive in here. Let's install Tableau, and I'm going to walk you through some of the basic UI elements. So the first step in installing Tableau is to go download it. Now on tableau. com, this is where you can learn all about the different products and all they do. There are great videos here explaining it and tutorials and everything. Tableau Desktop is the one we're talking about, so if you wanted to download that you could go here, get a free trial. We're actually going to use Tableau Public for this version, so I'm going to point you at that. Instead of going to tableau. com, you go to public. tableau. com. Here on tableaupublic. com what you have is a similar interface. You can sign in and create an account. If you've never used it before, you can download the app here. Enter your address; click Download the App. It's going to download it here. I'll click Save. Once that's downloaded, I'm just going to click on the download and run the installer. Go ahead and read the license agreement, and once you do that you can accept. If you wanted to customize the install location, you can do that here. I'll leave it as the default. And we're running on Tableau version 9. There are lots of other versions that are still kind of in the wild, but 9 is the current one as of the publication of this course. It has some recent updates to it that are pretty new and dramatic versus the old version, so it's exciting to see how fast and quickly this product is evolving. Okay, now that Tableau is installed, it went ahead and opened, and I'm looking at Tableau Public here. As I mentioned, you can connect to virtually any data source with Tableau. Tableau Public, however, you are limited to these three options here, the Excel file, the Text File, or the Microsoft Access file. Also, there's this OData feed option if you happen to have access to one of those. On the right we have our How-to Videos. These are quick little videos to understand the interface and all that. And you have something down here called VIZ OF THE DAY. This is a really cool way where you can see what other people have done and reverse engineer those. Everything on Tableau Public is available to everyone, so if you find something on there that you like or something that's interesting you can click Download and use that and reverse engineer it and make it your own. There's also usually some sample data sets here and some live training that you can schedule where you could actually go to a class.

Demo: Connecting to Data

Okay, now let's connect to some data in Tableau and start looking around and see what we've got. In the demos of this module, there is a file you can download, which is from a previous module in which we built up an Excel file for Cogsley Services, so we're going to take that file, and we're going to visualize it in Tableau during this module here. But first I want to open the Excel file just to explore it, and then I'll connect to it in Tableau so you can see how Tableau handles it. So here in the CogsleyServices-SalesData what we have are several spreadsheets, Sales, Companies, Managers, HourlyWages, and Pricebook. All of these have different data elements, the main one here being the Sales one, which has the majority of my data with all of the columns here that really make up a good solid data set for me to understand how sales are doing. The other tabs here have additional related data. Some of the data is also pulled in, but there are ways to bring that in and join it together. I'll show you how to do some of that in Tableau here in a second, but I just wanted to show you what we're looking at. So we're working with an Excel file, and there's nothing fancy here that is different than any other Excel file. And so when I go into Tableau and I connect to this it's going to do some really interesting things. It's going to auto-categorize these fields, and it's going to try to understand the data types and do all kinds of things with them without me having to do anything. Tableau has this intuitive engine that does that for me, and that's one of the value adds here, so let's go pull this file into Tableau and see how it works. Here in Tableau I'm going to click on Excel under my Connect options, I'm going to go browse to the workbook folder and find my CogsleyServices-SalesData, hit Open, and I'm in the connect to data window. This is where Tableau allows me to analyze my data. And I can do a lot here. There's probably more than I'm going to show you here that you can do, but you can see right off the bat what it did is it treated the Workbook kind of as a database and the Sheets similar to how tables in a database are treated. And that makes a lot of sense because those things are relatively synonymous. Here I have the Sales data set, and I can preview this by clicking the little viewer icon on the right. It'll load up to 10, 000 rows at a time, and I can see what my data looks like. It has column headers, which is nice. It looks like it pulled all the data in the right spots. When I drag this onto my sheets here, which is drag sheets here, you can see I get an automatic preview. There are icons here for numeric indicators and date icons for date indicators, green and blue ones for ones it thinks that are aggregates or numbers that we're going to use in our analysis versus the blue ones, which it thinks are the categories or the context of our analysis. I can scroll all the way over and see that it picked the column headers, and it did a pretty good job here. Now at this stage if I wanted to do something I could rename these fields, I could hide them, I could add a description, which is really helpful if you're going to be sharing this with other people that want to use the same data set. If you share the description there, they'll understand what that field is. There's also something here called the data interpreter. This is a feature you can turn on in the latest version of Tableau that will do things like remove extra headers or extra footers, basically clean up an ugly spreadsheet into just a pure data source. Similarly, I can click on this icon here, the little hamburger, and manage my metadata. I can change the data types, I can look at the tables, I can do all this kind of stuff. So here is the preview of the data, here is my metadata manager, and once I have done that I am ready to go. I could also add filters at the top. So if I wanted to say only show me a specific region or only specific manager, I could filter on either of those. So in the connect to data icon you have a lot. There's a lot going on here, and it's really important to understand that for when you afterwards build something knowing where the data came from and anything that could've happened along the way. Once we've connected to our data, it's time to go look at it inside of Tableau. I'll click on Sheet1 where it says Go to Worksheet, and here I am. I'm inside of Tableau, Tableau has connected to my data set, and on the left I can see that. I have Dimensions on top and Measures on the bottom. I like to think of Measures as the subject of our analysis, Sales Amount, the Rate, the Quote, the Quantity, and the Dimensions is the context of our analysis, so sales amount by Manager, sales amount by Customer, sales amount by Order Date over time, for example. These are just two segmentations of how Tableau thinks of data elements, and they do impact how Tableau works with them and interacts with them. Notice these indicators here. There's even one for City, which means that this is a geographic property. It recognizes the name City, and it automatically geo-coded it. So if I were to visualize this, I could do it by city on a map simply by clicking on this. Now recall I didn't ever tell Tableau that was a geographic property. Sometimes it guesses right; sometimes it guesses wrong. It's really simple to change and manipulate that. I can see all the fields I have here, and I can see all the Measures. Now I can add other data sources as well, I'm not limited to just one, and if I had them they would pop up just below this Sales one here.

Demo: Building Charts

Now that we have our data in Tableau, let's go ahead and build some charts. Here I am back in Tableau, and I want to build four charts here, and later we'll put those together as a dashboard. If you recall, some of the questions we had were year over year sales and sales by location and sales by industry. So let's do this. Let's take Order Date, and as I click on that I'm going to drag it up, and as I drag notice these areas here. They're called shelves, Columns shelf, Rows shelf, Filters shelf, Pages shelf, etc. They highlight orange indicating where this field or this pill can be placed, so where can I place this field here is highlighted in orange. Down here I have these Marks options where I can place it as well. For now I'm just going to place this on the Columns. And when I do that date is a real interesting one. Notice how it defaulted to YEAR. Now back in my data set I didn't have a separate field for year. I just had the actual date value being displayed. In fact, if I go back and look at my raw data here and I go over and find my Order Date, you can see that it's at the date level. There's not a year there; it's the full date. So what Tableau did, because it knew that this was a date field, it built a natural hierarchy for me. So what that means is that it's going to aggregate that data up to year. Now this is useful; this is good. My data is going back a couple years, so what I'm going to try to do now is drill down. I can click on the plus sign here, and I can see how I have YEAR and QUARTER. If I wanted to drag that down, I could do something like this, do that to MONTH. Let's get rid of QUARTER, so I broke the hierarchy. Not a problem. Tableau lets you work with it in any way you like. Let's go ahead and drag the MONTH back up and then YEAR over to Color. So you can see January across there, and now we have dots for every year, and they're colored by that. If I click on Sales Amount, I'm going to not drag that into my visualization yet, but highlight that. On the far right here I have something called Show Me. Now this is Tableau's intuitive way of trying to help you visualize your data using all the best practices in the data biz world. Now when I'm working with a date and a measure, such as sales, Tableau and all the other institutes out there that look at data visualization and the kind of neuroscience behind how our brains perceive things, if you remember our module on visual perception, there come to be trends or best practices for how to look at data. In this case it happens to be that the best way to understand data across time is with a line chart. Now Tableau is suggesting I use this line chart. There's also this one. And it's saying that because it highlighted it in blue by default. If I were to click off of Sales Amount or click on a different field here, notice how it changes. Now if I go click on any other measure here though, it's going to go right back to that. So that's the interesting thing. Tableau is trying to tell you hey, why don't you try this one, and to get there it's a single click. And now I have my line chart over time broken out by year and month. Now I wanted it to show year over year, so what I'm going to do is I'm going to drag YEAR from my Columns shelf down to my Color shelf, and then I'm also going to take Order Date again and drag it onto my Label. Now this is an interesting thing where it tries to do the hierarchy for me again. I don't want that, so I'm going to choose Year. So here you can see all the different years, year over year comparisons for my sales by month. If I wanted to change that to Quarter and make it a little bit easier to consume, I can do that, make this a little bit bigger. You can see how it's going there. So this is a real interesting way, and only with a couple of clicks we have a relatively complex visualization. This is year over year by quarter sales comparisons, and the color coding is just defaulted to this palette here. You can change that by clicking on Edit Colors. I like to keep them all the same color typically when I'm working on something here, so maybe a blue, and I'll assign that. It automatically chooses 2009 as the lightest and 2012 as the darkest. That's good because that's the one that I want to highlight the most. And in fact I'm just going to drag these guys up and rearrange them a little bit so that they overlap each other like that. So I can see 2012 on top of all the other ones, then 2011, then 2010, then 2009. So here's an interesting way of visualizing some simple data. I'm going to rename this sheet. Sheets are like a view or a tab in an Excel sheet, and the workbook itself combines multiple sheets or it contains multiple sheets. So I'll call this Sales over Time. Now let's build another one. I just clicked on the New Worksheet icon to get a new sheet here, and here I want to look at any trends among industry, so I'm going to drag Industry on, and I'm going to drag Sales Amount over. That gave me a bar chart. If I wanted to do that again, it'll undo up here. Tableau has a forever undo, which is awesome, so never get worried about doing something and not being able to go back to the previous state. When I click on Sales Amount now, instead of having a line chart because it knows that it don't have a date value, it suggests a bar chart. If you recall in the previous one about visualizing data and charts, I mentioned line charts and bar charts are kind of the best ways to look at things often, and Tableau is kind of showing us that based on all the research that they've done along with all the other folks in the industry. So here I have a very simple bar chart. I can see that the Television Services and the Semiconductors, etc. are the top two industries that we service. If I wanted to look at the rate or the quantity, I could do something like this where I can drag one of these fields in such as rate and put it on color. You can see the higher rate. the Television Services are the higher anyways, so that's great. Let's look about Quantity, and let's see---or Days to Complete, how often or how hard are these coming. Okay, it makes sense. So the larger sales are for the ones that take longer to complete. That seems to make sense to me. So we'll call this Industry Breakdown. I'll go ahead and get another sheet here, and this time I want to see a map of where my customers are. So at the level here I have City, State, and Zip Code, and I didn't build a hierarchy, but I could do that real simply. But what I want to do is instead just visualize, put a dot on a map every zip code that we have for a customer. So I'll double-click Zip Code, and it automatically draws a map. There's one unknown. I can click on this and click Edit Locations, and here's a zip code it doesn't recognize. Now if I had that or I were to go find that, I could just pop in exactly what it is. Let's say that was 90210 kind of a thing. You could just punch in the zip code there, and it would automatically recognize that, so you can adjust your data or manipulate it in a way that Tableau understands it. You don't change the underlying data, but you are telling Tableau how to handle that or how to treat it. So this is a pretty cool thing. Now I have an actual visualization of looking at where my customers are, and it looks like we have a lot of coverage all across the United States here. For the color, I'm going to go ahead and make this a little bit transparent so where we have more coverage it'll be darker versus others it'll be a bit lighter. And I'm going to use my data again. I'm going to show the Quantity of orders based on the Size, and the Sales Amount on Color. And instead of a sum of sales amount I'm going to see an average, so I can see the average, or I can also choose median. You can see that there. And for the color again I could change that to say a different color palette such as red and black diverging. We can do something like that. If I have a specific color pattern or color combo that I want to use, do something like this. I tend to like this orange and blue. It's very color blind friendly, which is great. And there we go. I have a real interesting way to see all of the dots and how all my customers are ordering and where they live. I can shrink these dots down a little bit if I want. There we go, something like that. And I can also make them a bit more transparent. Let's just do that. So the darker dots mean that there's a lot of concentration in that area. I have map options here I can play with. I can wash the map out and make it even more transparent. I can remove things, like I don't really need the country region or the country name. The state names are fine, we'll leave those on, but you can see exactly kind of where everyone is at and how this all fits, so---. For my indicator down here I'm going to right-click on that and hide that, and I'm going to call this Customer Map. So I've got three charts there. I've visualized the data. There are lots of other options. Show Me is your friend. You can click on this and just simply click through all these different chart types. If I want to see this as a heat map instead or a scatterplot or a word cloud, you can kind of come up with artistic expressions sometimes, bullet graphs and all kinds of different visualizations, heat maps, these kinds of things. And I'll just go straight back to my map there. And again, I probably lost all of the customization I was doing, so I'll just keep hitting undo until I get back to where it is. Notice how easy it is to switch between visualizations. That's the idea here that just because you have the data and you have a way that you want to visualize it doesn't mean that that's going to give you the answer that you want, so sometimes you need to be able to quickly flip through the visualizations and understand the different ways of looking at the data until you find the meaning buried inside of it. Let's do one more visualization here, and what I want to see is a scatterplot of customers, and I want to look at the average Discount %, double-click on that, change the aggregation to an Average, and the average Sales Amount and see if these two are correlated. What I'll do here is I'll actually add Customer Name to my Detail shelf, which will give me a dot for every customer on there. I could also do something like this and change that to Order ID. You can see how there are some outliers here where this one got an overly high discount for the average sales amount there, but you can go back and see the customers there and how they're all kind of right around the middle. So it looks like there are some groupings in here, specifically around this Discount amount and the Sales Amount. So let's take a look, and we can even color code this if we wanted to. Let's color code it by Sector and see if there are any trends there. Well, it looks like the green is kind of a popular one, but it doesn't look like any of them are living outside of this. We don't have any interesting or varying ways of coming up with this discount or this correlation. Let's see how it looks by Product Category. Drag Product Category onto Rows here, and it'll give me a different one for every---drag it onto Columns too. I can see how they're all different there. Yeah, it looks they're all fairly similar. What if I color bit it? Ahh, interesting. So things are getting a little bit diverse here. So this is how Tableau wants you to work. They want you to be able to visually explore the data, change aggregations, add color, sort, filter, do all of these things very quickly and easily. I'm going to leave this one here, and I'm just going to call it a development one, so I'll call this Beta-Customer Scatterplot.

Demo: Calculations

Okay, so let's build some calculations now that help us understand things a bit in more detailed way, some of the more advanced ways of looking at things. I'm going to jump back to my visualizations in Tableau. In here I'm going to go to my Sales over Time, and what I'm going to do is duplicate this sheet by right- clicking on it and saying Duplicate Sheet. And instead of showing the way I wanted to here, I'm going to filter out the previous two years by highlighting them and saying Exclude. And now instead of showing just a basic sum, what I want to see is year over year comparison, and visually I can see that; however, I want to see the actual difference. I don't really care about the line. I want to see the difference between them. So a way to do these types of calculations is known as a Quick Table Calculation. So if I click on a measure such as SALES, go down to the Quick Table Calculation option here, I can choose one of these many options, a Running Total, the Difference, Percent Difference, Year over Year, or Compound Growth Rate. What I'm going to do is the Year over Year Growth. And because of the way it's sorted it's choosing the wrong year. I wasn't interested in that one. Now there are four nulls because what it's showing me there is that 2011 doesn't have anything to compare to, so it's going to be null, so I'm going to go ahead and hide that. And here I have an interesting thing. So it looks like if I go back, and let's go ahead and just filter out the previous two years, I'm not interested in those, I can see that here yeah that Q1, 2, and 3 2012 was above 2011, and then it dropped down below, and this sheet here is essentially telling me how much above and how much below. So let's make this a bit easier to understand. Let's change it to a bar, and for the visualization I'm going to hold Ctrl on my keyboard and click and drag this down to Label. And I'm going to get rid of this one for the label because I just want to see the difference there, so now I can see the percent difference by quarter year over year. Let's move YEAR over to Detail so we're not coloring by it. Now I'm going to copy this guy again and put it on Color, and you can see what it does there. It makes it quite a bit easier to read. So the red one or the negative one really jumps out. Think about what we talked about in the last module about color and how to use it. Here your eye is instantly drawn to that dot there, that bar. If I wanted to, I could break this down by month instead, switch it to fit the entire view, and you can see November and December were the really bad months. So this is just one type of calculation. This is the one that it automatically builds for you, so I'm going to call this one Year over Year Comparison. Now I want to take my Sales Amount and create a calculation on it that would allow me to guess what next year's forecast would look like, so I'm going to right-click on this and choose Create from the context menu and Calculated Field. In here what I'll do is I'll take Sales Amount and do some simple math \* 1. 2, give it a name, and down at the bottom I have a little indicator saying this is valid, so that's good. If I wasn't sure about the options, I can click on this context menu on the right, and it'll show me all of the different calculation options I have. I can search for those, and I can categorize them, etc. This is a real basic one, so we're just going to take this times a number, hit OK, and now what we're going to do is drag Sales Amount on, drag Forecast Amount right next to it, and you can see that yeah it's about 20% higher. It makes sense. And now what I'd like to do is actually see this over time, so I'll drag Order Date up here. I can see what that would look like. Let's go ahead and get rid of Sales Amount because really I just want to know what my forecast would be. And so if I wanted to do something, let's say at the month level, and I can filter on this using a quick filter to where I can actually see the past 12 months, so in this case that's from December, the data ends in December of 2012, so I'm just going to go through January of 2012 through December of 2012, and this is essentially what I could use as a forecast. It's a simple guess, but I created a calculation based on another field using very simple math and simple syntax. So if I wanted somebody to be able to edit this using a variable just like this quick filter out here, I would need to create a parameter, so let me go ahead and right-click in the same area here, and I'm going to go ahead and create a parameter this time. It's going to be called Forecast Amount Parameter, it has a value there of Float, and what I'm going to do is leave it as Float, give the value a. 2, let's say a minimum and a maximum, or let's just let it be. We'll do a step size of. 02 or. 01, either way, current value. 2. Okay, we've set up a parameter here. We can change data types. We can have a List or a Range or All. We're going to use this in our calculation, so once I hit OK it's there. The next step would be to show the parameter so I can see what it looks like over here. I used that one so I can create a type in. Now I'm going to right-click on my calculation, click Edit, and here instead of this I can just start typing Forecast Amount Parameter. It auto-completes, and it gives me this number there. So now it's actually doing. 2, so 20% of the total, not 1. 2 is what I wanted, so let's edit that again, put some parentheses here, and do 1 + that amount. There we go. Now we have that 1. 2 over our previous sales. So if I wanted to change this to. 5 or. 4, you can see how it changes there. This number is on the right. If I gave this a minimum and a maximum, let's say it was between 0 and 1, I could click OK, and now I'd have the option to actually make it a slider, so this is an interesting one where you can just kind of move this guy around like this. You can also format this. If I go back and edit the parameter again, there's a display format here. I could change that to a percentage. Let's leave it as 0 decimals there and hit OK. Okay, cool. So I have something here that is a pretty unique thing; call it a Sales Forecast Calculator. And there are just two simple examples of how to create a calculation in Tableau.

Demo: Dashboards

Okay, let's bring this home now by building an actual dashboard that combines multiple sheets, and then we'll actually look at how to share that work with others via Tableau Public. Back in Tableau, instead of choosing a new worksheet down at the bottom, I'm going to choose New Dashboard. When I do that, Tableau shows me all the sheets I've built, gives me these hover thumbnails as I go over them, and it's really simple to do this. For right now it has a size. This is a range of which it wants to expand or contract. I can change this. So I can look at it Automatic where it would fill the entire screen, a Laptop size, a Desktop size, etc. And I'm going to start by double-clicking on Sales over Time. When I do that, it brings the data on. I don't need this legend here because the labels already exist. Go ahead and delete that. Now instead of double-clicking this time, I'm actually going to click and drag, and you can see how it highlights different sections of the sheet, which is where this sheet will fill up the screen. So if I want it on top, I could drop it there, if I want it on the side, etc. I'm going to drop it on the bottom, and I'm going to get rid of this legend here. And I'm actually going to drag the Customer Map on as well right next to the other guy and get rid of those so I can see that there. So I have here a very simple dashboard. It combines multiple sheets. I can add other elements like an image, or I can add other web pages embedded into this. I can set containers or just add text. Let's say I want to put something at the very top, Sales Dashboard. I can format that, something like this, hit OK. Maybe I want to make this white, format this, give it like a dark gray, etc. There you go. So I can do this kind of a thing. I have now my Dashboard title. I have this. If I wanted to add a piece of text on the bottom for any kind of legal notice or anything like that, I could do so. And I've done it. I've built a simple dashboard. Now the next thing I would do here is I would probably add some interactivity. So let's say I wanted to be able to see sales over time and have it filter everything else or vice versa. Let's say that if I wanted to click on an industry, I wanted the map and the chart to show just those that I clicked on, well there's a really handy feature here called Use as Filter. So if I click on one of these charts, I choose Use as Filter, and now when I click on Television Services both the map and the line chart above are filtered specifically to Television Services. When I click off of it, it goes back to normal. Similarly I can do that for this one. So if I wanted to see let's say just this zip code here, I can do that, see what that's like. I can do some kind of a lasso effect if I wanted to see okay let's look at everybody in this part of the country here. I can see their sales, what in industries they're in, etc, and I can click off of it and go back to normal. So Tableau in a very simple way allows you to come up with some really cool interactive pieces.

Demo: Sharing Your Work

Okay, we're bringing it home here, and we're going to show how to share your work using Tableau Public. I have my dashboard, and if I wanted to save this, it's going to actually ask me to log into Tableau Public and save it. So if you don't have a profile yet, you'll need to go create one. That's just a simple set up process. Once you do though, you just go ahead and sign in, give it a name, click Save, and it's going to push all the data directly up to Tableau Public. Tableau Public is their hosted version of the server, and anything you publish there it can be accessed by others, so if you're using any sensitive data be very careful. You could essentially download the workbook and then go delete it from there if you wanted to, but this is just really a simple idea of how this piece works. It's very similar to how Tableau Server works, so if you have Tableau Server or Tableau Online you can publish and share your work the same way. I now have a URL. I could pass that along to somebody. And the cool thing is that all those interactive pieces still work. So when I click on this, it does those filters for me. So I can do all that interactive stuff, and I can export this and work with it just how I would on any other place, and from there it would actually just work on the web instead of having to have the tool installed. You can see all this data down here. And there you go. So that's the basic way to share your work. There's also a way to embed this. There's a Share option down here. I could get some embed code that I could embed into an HTML page, or I can copy this link. If I were to go to a new tab, you can see that it's embedded here. So this is a different way of doing this. It's pretty neat and pretty awesome. Tableau Public does have some limitations, but you're probably not going to hit them. You can have over a million rows of data and up to a gigabyte's worth of data being uploaded, so don't feel shy or bashful about publishing stuff there. It should all work fairly well, especially since you're limited to Excel or CSV files as a data source.

Where to Find More

Okay, so let's figure out where we can find more info. On Pluralsight we have quite a few courses on this. A few of them I've published, another one is published by another guy, and they're all focused around Tableau, Tableau Desktop, Tableau Server. Business Dashboard Fundamentals, as I mentioned before, gets really deep into the concepts behind dashboards and planning for them. The Data Analysis Fundamentals is all about Tableau and how to work with it. Big Data Analytics talks about big data. So if you're a company or you're already using some of those systems, it's a really interesting thing in looking at how Tableau interacts with it. And then of course there's the Data Visualization Using Tableau Public, which is specific to Tableau Public. So if you're thinking about using the Desktop version, you'll probably do one of the other courses, but if you're going to stick with the free version then Tableau Public is the way to go for you.

Presenting Findings

Introduction

Welcome to this module on Presenting Findings. In this module we're going to talk about what presentations are, why we would use presentations and why they're so important, how we can actually do this, how the act of presenting findings can convey and move our audience, we'll go through some of the key concepts, and lastly we'll point you where you can find some more info. Let's get started.

What Are Presentations?

So what are presentations? I think presentations are this mix of entertainment and information. They should enlighten and move your audience all at the same time. Think about where we stand here. On one side you have a document. This is something maybe you read, and it's like a memo or some document manual, something like that. On the other end of the spectrum you have a movie, something that is pure entertainment, something that just is there to enlighten you and to share some stuff and maybe communicate a few things, but mostly entertain and engage. Well, I think presentations are somewhere in the middle of this spectrum, and this is where I'd put you, the analyst. Your job here, working with data as a data analyst, is to both move and enlighten your audience. You should inform them, and you should encourage them. They should see what you do and be inspired to act. That is the ultimate goal here anyways is to learn as the process of analysis works, and then in the end realize something that actually can change in your business or in the world for the better.

Why Use Presentations?

So why would we use presentations? Why use this format versus a different format? Why not just email everything or have text-based or have just on the other side pure marketing material, pure movies? Well, I think that presentations, when done right, will have this lasting impact on the audience and can really spark a change in the world. Think about some of the most famous presentations or speeches in history. Think about Martin Luther King's I Have a Dream speech, which wasn't called I Have a Dream, but you get the point. It moved and changed the Civil Rights Movement. Think about other forms of Steve Jobs presenting the iPhone for the first time and how that sparked this whole revolution of smart phones. Or even think about Al Gore's presentation, love him or hate him or disagree or agree, his presentation with An Inconvenient Truth was incomparable, and its quality and its ability to entertain and move the audience, that's what we want to do here. Let's think about this a little bit. How does the process work, and how does it fit in with our process as data analysts? We start with our data, we go through our analytics cycle, we explore the data, we have a question, we answer the question, we learn something, we have new questions. It's this never-ending cycle of learning and continually evolving our understanding of something. This is all happening because we are here to do this. We are the analysts, and it takes a human to actually take their experience and their intuition and apply it using these methods to the data to come up with this understanding. Well, none of this matters if we don't present it, if we don't have a way to share this information. And I don't just mean putting a link where somebody can look at a dashboard. I'm talking about a way that actually moves people, something that actually inspires them to act.

How to Present Findings

So let's get in now and talk about how to present the findings. I have some fun examples here. By using a clean style and a great story, you can design your presentations that will truly move your audience. I believe this whole-heartedly, and let's look at some examples. When building a presentation, it's key to know your audience. The audience in the end are the ones that you're trying to affect. You're trying to move them, you're trying to encourage them, to inspire them, so knowing who they are, knowing what a day in their life is like, knowing why they're there, all these things are critical to understanding how to actually present your information. Write them down. Have a form, have a sheet that you fill out every time you go to give a presentation so that way you can understand your audience better because every audience that comes to hear you speak will bring a different perspective or a different vibe to the whole situation and understand and communicate and be receptive in different ways, so this is absolutely critical to know your audience. You need to find a common ground, something that you both can relate to, and this goes back to why we might use a story, and I'll talk about that in a second. But having a common understanding about something, whether you're both at the same company facing the same problem, whether you've all had shared experiences in the past that you can relate to, bring those up. Include those in your story and in your presentation so people can feel like they empathize with you. Now structure to the story is critical, and we'll talk a little bit about that here in a minute having both emotional and analytical bits, ups and downs. Think about movies are structured. This is where film and television really help us with our presentations by giving us that structure, that story arc. There's a whole world to discover in this realm. And lastly, make sure you have a really solid design for your slide. I'll talk about that in a minute. You want to make sure your slides are effective in communicating and they're not putting people to sleep, they're not over-communicating, and they're not distracting. This should be a visual aid, not a visual hindrance.

Key Concept: Audience

So let's talk about this now. What about your audience? Who are they? Where did they come from? Why are they here? Are they your boss? Are they people that work for you? Are they peers? Are they other folks in the organization? Are they the public? Are they people that want to listen to you? Are they people that are there because they're being told to? Why are they there, and who are they? Understanding all of this is key to being able to communicate effectively with your audience. Think about where your audience is coming from. When they start your presentation or they start listening to your talk, think about where they are now and where then you would want them to be. So how are you going to affect that change? Write those things down. Where are they now? Where do you want them to be after your talk or after your presentation? These are two key pieces and two states. A lot of times when book reviewers and book publishers look at a new piece of work to decide if they want to publish it or if they want to buy it, what they'll do is they'll look at the beginning of the book to identify the protagonist and who they are and what their state is, and then they're look at the very end of the book and see how that person has changed. And if there's not enough transformation in there, they know that in the middle of the book there's not going to be enough substance, and people are going to lose interest. So understanding the beginning and the ending of your presentation in terms of where your audience is at is critical to making sure that what you present is actually relevant and going to drive that change or inspire them to act in the way that you want. Find common ground. This is of course key. If you're presenting to people you work with, this is super easy. Everyone you work with probably has similar problems to you. They may be frustrated with the same things or they enjoy the same things. Maybe the physical space you're in is something that you can all joke about and relate to, maybe how the Wi-Fi sucks or something like that. There are lots of different things that you can share, and every little tidbit, every little piece of how you can tie that into your conversation, even if it appears just to be ad-hoc or random can actually influence people's perception of your talk and then can lead to a more meaningful outcome later.

Key Concept: Story

Now let's talk about the story. This is one of my favorites. So if you recall, in this course what I have done is I introduced you at the very beginning to Cogsley Services. I told you that they are a startup, I've told you that they've seen tremendous growth in the technical services sector, I told you about their challenges pulling together data and analyzing it because it's across multiple systems. This is not a coincidence. This story is easy to relate to. We can keep going back to this, and it allows you as the audience to understand or to relate to something and hold it in a place where you can come back to it. It really is a great learning tool and a great communication tool. Now in your story when you're using a story like this and whether it's made up like Cogsley Services or whether or not it's real in your real business, it's important to include emotional and analytical bits. So when you're thinking about that, this slide right here does a lot of emotional stuff. It's talking about how it's a startup and they've seen tremendous growth, but now they're having a hard time and they really need your help, so it sets up this whole emotional connection with you and them, and that emotional piece should not be underrated. That piece is super important mostly because we're humans, we're not machines, and so you can never avoid or ignore the emotional connection or the power that you can use that tool to communicate with your audience about your story or about your message. Also, you should include the analytical things, the more practical things as I did in one of the later modules. I introduced you to the exact things that we were trying to accomplish here. We wanted to analyze the data by these three ways, and we wanted to look at them and understand different things. We wanted to remove duplicates. We wanted to do all these things that are very practical and analytical. And a lot of this doesn't make a whole lot of sense or you probably wouldn't care nearly as much if you didn't already have this emotional connection. When you set it up to where the story makes sense and you can associate with it, it's a growing company, they have these data problems, that could be the exact situation you're in right now, or it could be one you faced in the past in your career or maybe one in the future, but everybody that's looking at this, I'm assuming as the teacher here, that you will be able to relate to this and understand it, and from that point you have a foundation to build upon. Now with these analytical bits and these practical things we have an interesting way of how to go about it and move forward, and you have that context and that foundation so you understand exactly why we're doing something, which just makes it much more meaningful.

Key Concept: Slides

Now let's talk about slides. This is one that is a bit challenging for most people. Most people have the basic understanding of how they may go about doing slides, but I've got some shortcuts for you here I think are going to be really helpful. So when looking at the slides you have a message, and then you want to present that message in your slide. Well, it's important to keep it clear so that one message equals one slide. You should never have more than one message on one slide. Furthermore, the slide itself, you're going to want to minimize your text. Now if you're presenting something and it's more of a training thing or you need reference material, whatever, but in general you want to minimize the amount of text on the screen as possible. Use images, use diagrams. Look at the slide right here. Think about how I came up with this kind of a layout. This was all based around this idea of this 1:1 where you have one slide for one message. Also, you should consider using full-screen images. We've seen a few already in this module, and think about all the other ones and when that full-screen image really draws you in. You can use emotion or a moving image, that kind of a thing, but really think about using images, diagrams. They're so much more powerful. Hopefully throughout this course you've learned about the power of visual perception and how as humans we're just designed, we're wired for visual input much more than we are any kind of text or auditory or any other input. So try to use those to your advantage, and you will get a lot further and convey a lot more meaning with actually a lot less writing. It's really simple. You put some images that represent what you want to say, and then you just say it. Now let's take a look and actually go through an example. So if I'm designing a slide, the step one that I'm going to go through here is have my slide ready and put some points on there. This is pretty common, right? In fact, the software we use usually, Keynote, PowerPoint are the most common ones, encourage this type of behavior. Well, unfortunately this is a really bad way of communicating. If you've heard death by bullet points, that's exactly this. People say death by PowerPoint. I think that's unfair. I think PowerPoint itself is just a tool. Although it does encourage some poor choices in design, it is just a tool, and you can make it do whatever you want, so don't let somebody tell you that PowerPoint is the devil here. It's the bullet points in my opinion that are the devil. So this is typical, right? You go through, and you add Point 1, Point 2, Point 3, Point 4, you add some more detail, you add more and more detail, and sooner or later it looks like a dissertation. And now if I were to show you this and put it on the screen well in advance of me talking, most people what's going to happen is they're going to start reading it. They're going to start going though point-by-point reading ahead. Now that's destructive. You want your message to come across at the exact right cadence. At the time that you want to throw the punch line in there is super important, so don't let that happen. Now once I've done this and I've bullet pointed or outlined my message, what I want to do is actually find the most important one. Here that happens to be Point 4. And so I'm going to take all of these, I'm going to go down to my Notes section, and I'm going to put them in there. So now I still have all that detail, I still have that structure that I came up with, and I'm just going to get rid of what was actually on the slide. So now I'm not distracting anybody, I'm not causing any problems with people trying to read ahead, my message can still be delivered exactly on time, and instead what I'm going to do is replace that with an image. Here's a graphic that represents all the locations where Pluralsight users are located, and my message in the slide was that we're actually in 150+ countries. So you can see here how I took all this text and all this raw detail, simply copied it down, found the most important bit, found an image to represent what I was trying to talk about, and I can leave that image up there for a long time and talk because you're going to be visually exploring it but still able to process what I'm saying. And then lastly I put just a couple pieces of text that is the exact point I'm trying to make. Think about the difference between those two versions of that slide there. Think about how effective this one would be over the other one.

Key Concept: Data

Now let's talk about data. Rounding this out here, data is of course super important when we're communicating. But it's not just how we actually analyze the data or the summary statistics we came up with. It's about how it is actually presented. And when we think about the visual perception we've already talked about, we already should have some of this concept there. I'm just going to go through some examples and talk about the data-to-ink ratio. Now this is the idea here where we have a chart like this, and in the data viz world we would say this has a ton of chart junk. And chart junk is basically dots on the page or ink in the old days that isn't data. So every dot on here, every piece of ink used to display something that isn't data is wasted ink. It isn't something that we want. We want to reduce it as much as possible. So think about what we could do here. On the left we have an absurd about of extra zeros. Now this example, of course, is a bit over the top because I wanted to illustrate the point. And think about these grid lines. What are they, these kinds of prison bars that we've put the data behind? All of this stuff is not healthy. It's not helping the viz at all. In fact, it's distracting. Down on the bottom we have some super big, bold year labels. That's actually not horrible. Then you have this text on the bottom. I could do without that in this example. On top we have some shading around our title. All of these things aren't great. They're not really adding value. And then on the right we have our Region label. This actually is necessary for this because otherwise you'd have no idea what we're looking at. But there really isn't a message here. In fact, choosing the right chart type is also a key part of displaying data correctly. In this case I'm looking at data over time, and all this research suggests that the best way to look at data over time is using lines. And furthermore, this is a stacked chart. So if I actually wanted to understand a trend, for example which of these are trending up or down, it's almost impossible to visually do that. I would literally have to write down a bunch of ideas and come up with it and kind of redraw it in my head, which really isn't efficient. So what can we do here? Well, we can change the labels on the left, we can remove the grid lines, change the chart type, change the colors, do a lot. Let's make one sweeping change here and see what we could do in the first pass of cleaning this up. So here what I've done I've shrunk down the axes so they're not super bold and big like they were before, I've removed all those extra zeros and added a B to indicate that we're looking at billions, I've converted it a line chart because that was the key here is that we actually want to look at data over time, so it's not something that we would even want a bar chart for necessarily, and I've added the labels directly onto the chart so I don't have to bounce back and forth. My eye doesn't have to track and try to remember which color I'm looking at and associate it to a color legend. I really don't like having to have color legends or those things on there if you can just label directly on the chart. That's much more easy, much more efficient. Here you can easily determine exactly what's going on. Europe, of course, is killing it. Americas is on the upward trend. Asia in 2012 surpassed the Americas, which is maybe the first time in at least the past 12 years on this chart. And then down below everything is kind of low, the Africa and Oceana and all these other areas. So depending on what my analysis is, this could be meaningful or not, but certainly this is a less chart junky way to look at it. There is a higher data to ink ratio here, and it's easier to understand and consume visually for sure. Now if I wanted to focus this analysis, and this is really where the transaction is, if you built this chart, I would say you're doing okay. Now if you wanted to have a specific point such as Europe dominates when it comes to tourism or Asia just took over Americas in 2012, you want to highlight and reduce all of the other colors, remove all of the other labels, and focus solely on that single point, so let's do that here in this example. Now here on this last one what I've done is I've removed the shading around the title, I've removed the borders around the line chart, I removed all of the labels except for the one I want you to focus on, which is Europe, and I also have changed all the colors so there's only one contrast here, Europe and everything else. The point here is that Europe has dominated in the tourism income by region since 2000. Now notice I took out the axes on the left side because on the title of the chart I'm saying this is tourism income, so you can assume that when I have a number there like 400B that it's 400 billion in tourism income by region. So easy enough, stripped all the way down, and the message just shines directly through. Going back to that visual perception talk when we looked at how to find the number of fives, think about that. This is the representation of that, the same affect of the pre-attentive processing. When you looked at this chart, it was probably impossible to avoid looking at the orange color and the word Europe, which means that the message of Europe is dominating here shined right through before you could even comprehend what was going on. That's what we want to do when we present data. When we build interactive dashboards or things that we're going to use in regular meetings that need 50 buttons on them, that's a different type of analysis. Here what we want to do is present the information and share it in a way where our message shines through.

Where to Find More

Okay, where can we find more info? There's a ton of this out there on the web. We've got some good courses, and I'll point you to some other good resources as well. So for Pluralsight we have PowerPoint Essentials, Speaking Fundamentals, Presentation Design, PowerPoint 2007 Part 2: Making a Speaker-Led Presentation. I thought this was a really interesting one that I'd recommend. We have other resources out there, Duarte Training from duarte. com. This is Nancy Duarte. She is kind of a goddess in this space, and she has an incredible amount of content, and her company is fantastic at doing this for big companies, so they actually work on things like presentations for Al Gore and all those. So they're training, they have online stuff, and they also have a two-day onsite seminar. It is a great use of your time and resources, so I highly recommend checking them out. Garr Reynolds is another prolific author in this space. He has a book called Presentation Zen, probably the only book I've ever read cover to cover where I could not put it down because it was that good. And then The Back of the Napkin by Dan Roam. This is another great way just thinking about how to sketch things out, how to diagram things. All of these people are incredibly knowledgeable and incredibly helpful, and I encourage you to really make the idea of presenting a key part of being a good analyst because, again, without the ability to present the data and to get people to act, really it's just a bunch of numbers that we're just playing with. I hope you've enjoyed this course, and I hope you've learned a lot. If you have any questions or anything, please feel free to leave comments in the discussions, or you can find me online on Twitter at @bensullins and we can take it from there. Thank you.