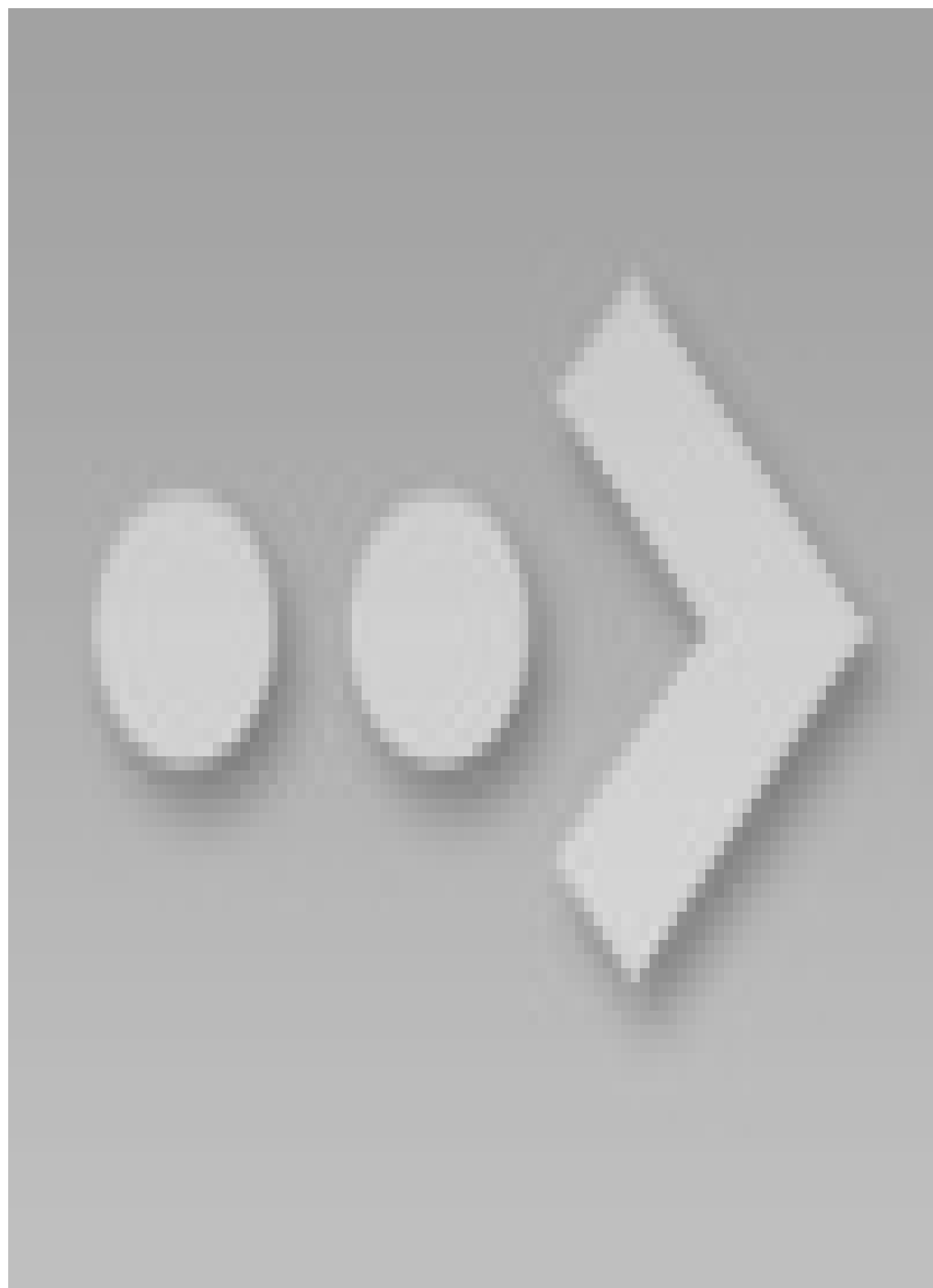


Practical Tableau, 1st Edition



Practical Tableau

Ryan Sleeper

Copyright © 2017 Ryan Sleeper

All rights reserved.

Published by O'Reilly Media, Inc., 1005 Gravenstein Highway North,
Sebastopol, CA 95472.

ISBN-13: 9781491977316

4/20/17

Part 1: Fundamentals

Chapter 1 - How to Learn Tableau: My Top Five Tips - AVAILABLE

Chapter 2 – Which Tableau Product is Best for Me? - AVAILABLE

Chapter 3 – An Introduction to Connecting to Data - AVAILABLE

Chapter 4 – Shaping Data for Use with Tableau - AVAILABLE

Chapter 5 – Getting a Lay of the Land - AVAILABLE

Chapter 6 – Dimension vs. Measure - AVAILABLE

Chapter 7 – Discrete vs. Continuous - AVAILABLE

Chapter 8 – 5 Ways to Make a Bar Chart / An Introduction to Aggregation - AVAILABLE

Chapter 9 – Line Graphs, Independent Axes, and Date Hierarchies - AVAILABLE

Chapter 10 – Marks Cards, Encoding, and Level of Detail - AVAILABLE

Chapter 11 – An Introduction to Filters - AVAILABLE

Chapter 12 – An Introduction to Calculated Fields - AVAILABLE

Chapter 13 - An Introduction to Table Calculations - AVAILABLE

Chapter 14 – An Introduction to Parameters - AVAILABLE

Chapter 15 – An Introduction to Sets - AVAILABLE

Chapter 16 – An Introduction to Level of Detail Expressions - AVAILABLE

Chapter 17 - An Introduction to Dashboards and Distribution-AVAILABLE

Part 2: The INSIGHT Framework for Data Visualization

Chapter 18 – Introducing the INSIGHT Framework for Data Visualization - AVAILABLE

Chapter 19 – Identify Business Question - AVAILABLE

Chapter 20 – Name KPIs - AVAILABLE

Chapter 21 – Shape Data - AVAILABLE

Chapter 22 – Initial Concept - AVAILABLE

Chapter 23 – Gather Feedback - AVAILABLE

Chapter 24 – Hone Dashboard - AVAILABLE

Chapter 25 – Tell the Story

Part 3: Storytelling

Chapter 26 – Introduction to Storytelling - AVAILABLE

Chapter 27 – An Analogy for Data Visualization: Tableau's Iron Viz Championship - AVAILABLE

Chapter 28 – Tip 1: Know your Audience - AVAILABLE

Chapter 29 – Tip 2: Smooth the Excel Transition - AVAILABLE

Chapter 30 – Tip 3: Leverage Color - AVAILABLE

Chapter 31 – Tip 4: Keep it Simple - AVAILABLE

Chapter 32 – Tip 5: Use the Golden Ratio - AVAILABLE

Chapter 33 – Tip 6: Retell An Old Story - AVAILABLE

Chapter 34 – Tip 7: Don't Neglect the Set-Up - AVAILABLE

Chapter 35 – Tip 8: Don't Use Pie Charts - AVAILABLE

Chapter 36 – Tip 9: Use Comparisons - AVAILABLE

Chapter 37 – Tip 10: Use Callout Numbers - AVAILABLE

Chapter 38 – Tip 11: Allow Discovery - AVAILABLE

Chapter 39 – Tip 12: Balance Data and Design - AVAILABLE

Chapter 40 – Tip 13: Eliminate Chartjunk (but Not Graphics) - AVAILABLE

Chapter 41 – Tip 14: Use Freeform Dashboard Design - AVAILABLE

Chapter 42 – Tip 15: Tell a Story-AVAILABLE

Part 4: Chart Types

Chapter 43 – A Spreadsheet is Not a Data Visualization - AVAILABLE

Chapter 44 – How to Make a Highlight Table - AVAILABLE

Chapter 45 – How to Make a Heat Map - AVAILABLE

Chapter 46 – How to Make a Dual-Axis Combination Chart - AVAILABLE

Chapter 47 – How to Make a Scatter Plot - AVAILABLE

Chapter 48 – How to Make a Tree Map - AVAILABLE

Chapter 49 – How to Make Sparklines - AVAILABLE

Chapter 50 – How to Make Small Multiples - AVAILABLE

Chapter 51 – How to Make Bullet Graphs - AVAILABLE

Chapter 52 – How to Make a Stacked Area Chart - AVAILABLE

Chapter 53 – How to Make a Histogram - AVAILABLE

Chapter 54 – How to Make a Box-and-Whisker PlotChapter 55 – How to Make a Symbol Map with Mapbox Integration - AVAILABLE

Chapter 56 – How to Make a Filled Map - AVAILABLE

Chapter 57 – How to Make a Dual-Axis Map - AVAILABLE

Chapter 58 – How to Map a Sequential Path - AVAILABLE

Chapter 59 – How to Map Anything in Tableau - AVAILABLE

Chapter 60 – How to Make Custom Polygon Maps - AVAILABLE

Chapter 61 – How to Make a Gantt Chart - AVAILABLE

Chapter 62 – How to Make a Waterfall Chart - AVAILABLE

Chapter 63 – How to Make Dual-Axis Slope Graphs - AVAILABLE

Chapter 64 – How to Make Donut Charts - AVAILABLE

Chapter 64 – How to Make Donut Charts - AVAILABLE

Chapter 65 – How to Make Funnel Charts - AVAILABLE

Chapter 66 – Introducing Pace Charts in Tableau - AVAILABLE

Chapter 67 – How to Make a Pareto Chart - AVAILABLE

Chapter 68 – How to Make a Control Chart - AVAILABLE

Chapter 69 – How to Make Dynamic Dual-Axis Bump Charts - AVAILABLE

Chapter 70 – How to Make Dumbbell Charts - AVAILABLE

Chapter 71 – How and Why to Make Customizable Jitter Plots - AVAILABLE

Part 5. Tips & Tricks

Chapter 72 – How to Create Icon-Based Navigation or Filters - Not available

Chapter 73 – How to Make a What-If Analysis Using Parameters - Not available

Chapter 74 – 3 Ways to Add Alerts to Your Dashboards - Not available

Chapter 75 – How to Add Instructions or Methodology Using Custom Shape Palettes - Not available

Chapter 76 – 10 Tableau Data Visualization Tips I Learned from Google Analytics - Not available

Chapter 77 – Tableau Pie Chart: A Better Approach - Not available

Chapter 78 – How to Compare and Create Segments - Not available

Chapter 79 – Five Ways to Balance Data and Design in Tableau Without Graphic Design - Not available

Chapter 80 – Leveraging Color to Improve Your Data Visualization - Not available

Chapter 81 – 3 Creative Ways to Use Dashboard Actions - Not available

Chapter 82 – How to Create a Custom Shape Palette - Not available

Chapter 82 – How to Conditionally Format Individual Rows or Columns - Not available

Chapter 83 – 5 Tips for Creating Efficient Workbooks - Not available

Chapter 84 – Using Level of Detail Expressions to Create Benchmarks - Not available

Chapter 85 – Designing Device-Specific Dashboards - Not available

Chapter 86 – How to Make a Stoplight 100-Point Index - Not available

Chapter 87 – The Case for One-Dimensional Unit Charts - Not available

Chapter 88 – How to Highlight a Dimension - Not available

Chapter 89 – Allow Users to Choose Measures and Dimensions - Not available

Chapter 90 – How to Dynamically Format Numbers - Not available

Chapter 91 – How to Change Date Aggregation Using Parameters - Not available

Chapter 92 – How to Equalize Year Over Year Dates - Not available

Chapter 93 – How to Filter Out Partial Time Periods - Not available

Chapter 94 – How to Compare Two Date Ranges on One Axis - Not available

Chapter 95 – How to Compare Unequal Date Ranges on One Axis - Not available

Chapter 96 – How to Make a Cluster Analysis - Not available

Chapter 97 – 5 Tips for Making Your Tableau Public Viz Go Viral - Not available

Chapter 98 – 3 Ways to Make Beautiful Bar Charts in Tableau - Not available

Chapter 99 – 3 Ways to Make Lovely Line Graphs in Tableau - Not available

Chapter 100 – 3 Ways Psychological Schemas Can Improve Your Data

Chapter 1 - How to Learn Tableau: My Top Five Tips

Tableau's mission is to help people see and understand their data, and I can tell you that after you've mastered a few of the fundamentals, it is an extremely easy way to do just that. For basic analyses, such as looking at a measure such as sales, and slicing and dicing that measure by a dimension such as region, I'm not sure anything could be easier than Tableau. However, there can be a substantial learning curve required to get exactly what you want out of the software. In fact, for me personally it has been a career-long education spanning eight years using Tableau. While I've been through some growing pains and experienced some frustration learning the tool, I mostly view my lack of perfection as good news. The challenge keeps my job interesting and I continue to get excited discovering innovative solutions to complex problems that have led to several successful visualizations.

This chapter shares my top tips for how to learn Tableau, whether you have a budget of \$0 or \$5,000.

5. Follow the community

Cost: \$0

The first tip in my list of top five is to follow the Tableau community. I have learned several software programs during my career in digital analytics and data visualization, and bar none, Tableau has the most selfless community of any of them. The great thing about following the community is that you can tailor the list of users you focus on to align with your own uses of Tableau. Perhaps you want to follow users sharing advanced technical know-how, members of the community who are applying Tableau in your own industry, or users more focused on design and user experience.

I've put together a Twitter list, [Data Viz Heroes](#), that might be a good starting point for you. These are just a few of my favorite users to learn Tableau from.

Remember, this is not a comprehensive list of every outstanding Tableau user, but a short list of users whose style aligns with how I want to use the software.

Some of these users have created aggregated learning resources from several users in the community. One of my favorites is Jeffrey Shaffer's ([@HighVizAbility](#)), [Data + Science Tableau Reference Guide](#).

Lastly, get involved with a local Tableau User Group. This is a free resource where you can meet local Tableau users and learn from what others are doing. Many of my 'Data Viz Heroes' mentioned above often speak at these meetings. These user groups are all over the world – use this handy [Tableau User Group map](#) to find the one closest to you and reach out to the leader to get involved.

4. Take a training class

Cost: \$13 - \$6,000 / day

No matter how many blog posts you've read, sometimes you just need to talk to somebody who can help you connect the dots between what you are learning. Attending a Tableau training or data visualization workshop can help you take your skills a significant step forward in a short amount of time. Tableau training comes in many shapes and sizes, and as with the community tip above, you should choose your Tableau training based on what you are hoping to get out of the software at this point in your development.

If you would like a recorded training, I recommend the resources available at [Link to Come]O'Reilly's Safari Udemy, Lynda, or Pluralsight. Several of these recorded trainings are taught by qualified instructors including my data viz heroes.

If you are in need of in-person training, you can attend a one-day training at Tableau's annual customer conference, a group training conducted by Tableau, an on-site training conducted by Tableau, or an on-site training conducted by a third-party trainer. I attended a group training conducted by Tableau during my second year using Tableau, and a condensed "analyst" training at a Tableau conference in my third year using Tableau. I can personally attest to the value that attending an in-person training provides.

I'd be a bad consultant if I didn't mention that if you are interested in my personal training services, please take a look at [my speaking schedule](#) and get in touch.

3. Read up

Cost: \$35 - \$45

It may be a cliché, but there are simply some good books on Tableau available to learn from. This is a great starting point for learning Tableau, and one we sometimes take for granted. When I started using Tableau ("Back in my day...."), there was only one book that I can remember, and it was a very short book. There are now dozens of such resources available. I'm obviously biased towards the book you are currently reading, but among several other great options, here are two that I vouch for: [Tableau Your Data!](#) by Dan Murray This book is possibly the best all-around – *Non-Practical Tableau* - resource for getting started with Tableau. It provides some of the basic fundamentals, but also discusses more advanced features and Tableau Server.

[Communicating Data with Tableau](#) by Ben Jones In my opinion, Ben's book is the best second step as it is more strategic and provides some ways to think about your approach to data visualization after you have the fundamentals down. It also offers several hands-on walkthroughs for different applications of

Tableau.

2. Practice

Cost: \$0

There is no substitute for on the job training with your own data and unique business problems. The more challenges you come across and push through to an eventual solution, the more unique tools you get to add to your toolbelt to solve increasingly complex problems that emerge. This may sound obvious, so I will offer an extra tip to help you get the most out of your practice: Start a weekly internal meetup; I call mine Tableau Tuesday. During Tableau Tuesday, a group of 5 – 10 internal Tableau users get together to train, share case studies of our own work, and/or work collaboratively through challenging situations. These Tableau Tuesday events lead to valuable discussion and ensure that the entire team is continuously learning.

1. 1. Tableau Public

Cost: \$0

I credit Tableau Public as the primary reason for my personal success with Tableau, and thus, it is my number one tip for how to learn Tableau. [Tableau Public](#) is a free tool that has almost all of the same functionality as Tableau Desktop (Personal). You can currently connect to and explore Excel, text files, and Google Sheets with up to 15 million rows.

The only catch with Tableau Public is that your files have to be saved to the web, and external audiences can potentially find your work. For this reason, it is not a suitable option for private business data. I actually view this as a positive. This forces you to find topics and data outside of your normal work environment. As I described in [my Tableau customer story](#), Tableau Public is my sandbox to try new approaches to data visualization that may not be as - let's say,

appreciated - in a business setting. The cool thing is, these ‘attempts to fly’ are often eventually figured out, and often make it into my daily corporate work.

You can also download many of the workbooks you find on Tableau Public. This provides an amazing bevy of dashboards that you can use as a learning resource by downloading, looking under the hood, and reverse engineering. There is an option for the publisher to disallow this feature, but there are still thousands of downloadable dashboards – including [every single one of mine](#). I previously had just one dashboard that was not downloadable, *The Cost of Attending the 2015 World Series*, and that was because it included stadium data of Kauffman Stadium in Kansas City and Citi Field in New York worth thousands of dollars to create. Well, I’m proud to report that even the 2015 Tableau Public Visualization of the Year is now available to download for free!

I unlocked this dashboard for two reasons, which I’ll relate here because they illustrate the spirit of Tableau Public. Steve Wexler of Data Revelations wrote a post called [In Praise of Tableau Public](#). In the post, Steve describes all of the things that I love about Tableau Public. Then I came to a line that said, “Unless you indeed have proprietary data please, please, please don’t stop your workbooks from being downloaded.” That’s three pleases. It reminded me of how important Tableau Public is as a resource for people to learn from and have discussions around approaches to data visualization.

Second, after the announcement that this viz received the honor of Tableau Public Viz of the Year, I was immediately asked personally from a new user for the original copy so they could see how it was created. It simply didn’t feel right to keep the dashboard locked. My hope is that Tableau users of any experience level have the opportunity to learn from Tableau Public dashboards so they can incorporate innovations into their own work and continue pushing the envelope in their own ways.

That’s it my for my top five tips for how to learn Tableau. Trust me when I say that *everybody* is learning! The key is to be persistent. Tableau is user-friendly enough and has so many resources available that anybody who is committed can

become an expert.

Chapter 2 – Which Tableau Product is Best for Me?

The first thing I wish I knew the first day I used Tableau is which product or products should I download to get started. Tableau is growing at a rapid pace and there are still regular updates to all of their products, as well as their product ecosystem itself, making product selection a potentially confusing topic for a beginner.

The first thing to know is that Tableau is a brand, not a specific product. When somebody asks you to “download Tableau”, they could be talking about Tableau Reader, Tableau Public, Tableau Desktop (Personal), Tableau Desktop (Professional), Tableau Server, Tableau Online, and so on.

This chapter provides an introduction to Tableau’s product ecosystem so you can make the choice that best suits your individual requirements.

Which Tableau product is best for me?

The decision on which Tableau product to download comes down to four key attributes:

1. Connectivity – what data sources do you need to access?
2. Distribution – who do you want to see your dashboard and how will you share it with them?
3. Automation – do you need your work to update automatically on a refresh schedule?
4. Security – do you require an on-premise level of security or can your work be saved in the cloud?

From here, I will share a brief synopsis of each product, how each answers the four questions just mentioned, and who might get the best use out of each product.

Tableau Reader

Tableau Reader is a free download that allows you to open “packaged workbooks”, which are Tableau workbooks that are saved in a special way by Tableau Desktop users so the data and visualizations are in the same file. Tableau Reader allows you to open and interact with Tableau workbooks, *but not develop them*. Development capabilities could be considered an obvious fifth key attribute, but as Tableau Reader is the only product listed that does not provide these capabilities, I have not listed it as a key consideration. This product works much like a PDF-viewer, where a developer of a document saves it in a certain way so that it can be opened by a PDF-reader.

Connectivity: .twbx files only (packaged workbooks)

Distribution: Offline

Automation: Not available

Security: As good as your personal computer / server’s security

Best for: People that need an affordable way to view and interact with colleagues’ Tableau workbooks

Tableau Public

Tableau Public is another free download, but this product actually provides development capabilities. The catch is that the workbooks have to be saved to Tableau’s public cloud, making this an unsuitable choice for proprietary business data.

Connectivity: Excel, text files

Distribution: Cloud (Public)

Automation: Not available

Security: Limited; your workbooks are potentially accessible by anyone on the Internet, but you are able to restrict the ability for someone to download your files.

Best for: Journalists; sharing publicly available data (and Tableau know-how) with the world; practicing Tableau for free; trying the software

Tableau Desktop: Personal

Tableau Desktop: Personal is the entry point for the paid development versions of the software. It allows you to keep your workbooks private, but connection and distribution options are limited.

Connectivity: Excel, text files, Access, statistical files, shape files, spatial files, and Tableau files

Distribution: Offline or Tableau Public

Automation: Not available

Security: As good as your personal computer / server's security

Best for: Those that only need to connect to flat data files; those that need the most cost-effective version that will keep their data private

Tableau Desktop: Professional

Tableau Desktop: Professional is similar to Tableau Desktop: Personal in that it is a development version of Tableau. Both the Personal and Professional versions have all of the same development capabilities, but the Professional version provides full access to every data type and distribution channel.

Connectivity: All possible connections in Tableau

Distribution: Offline, Tableau Server, or Tableau Public (all possible distribution options in Tableau)

Automation: Not available

Security: As good as your personal computer / server's security

Best for: Those that need to connect to data in databases; those that need the capability to publish to Tableau Server

Tableau Server

Tableau Server provides a central repository for all of your Tableau workbooks that can be accessed by your business users via a web browser. Tableau Server also has the advantage of data refresh capabilities as well as a way for your organization to keep their data and workbooks on premise in the case that your organization requires that level of security. Tableau Server requires additional user licenses, even if you already have a Tableau Desktop license.

Connectivity: Workbooks that have been published to Tableau Server and that you have been granted access to

Distribution: Cloud

Automation: Available via data refresh schedules

Security: As good as your on premise or server host's security

Best for: Those that need to access / distribute workbooks in the cloud; those that want to automate workbook refreshes; those that want to edit workbooks in the cloud (limited capability); those that need to keep their data and workbooks on premise

Tableau Online

Tableau Online is similar to Tableau Server, but it is hosted via a third-party partner of Tableau. This product still has the advantages of cloud distribution and automatic refreshes, but it is hosted off premise, which can result in security challenges for certain organizations. Like Tableau Server, Tableau Online requires additional per-user licensing, even if those users already have access to Tableau Desktop.

Connectivity: Workbooks that have been published to Tableau Online and that you have been granted access to

Distribution: Cloud

Automation: Available via data refresh schedules

Security: As good as Tableau's third-party host

Best for: Those that need to access / distribute workbooks in the cloud; those that want to automate workbook refreshes; those that want to edit workbooks in the cloud (limited capability); those that are okay having their data and workbooks hosted off premise Tableau consistently invests in research and development so it is a good idea to keep an eye out for updates and new products at Tableau's product page at tableau.com/products. Also, if you are a non-profit organization and just getting started with any of the paid products mentioned above, be sure to ask Tableau for special pricing!

Chapter 3 – An Introduction to Connecting to Data

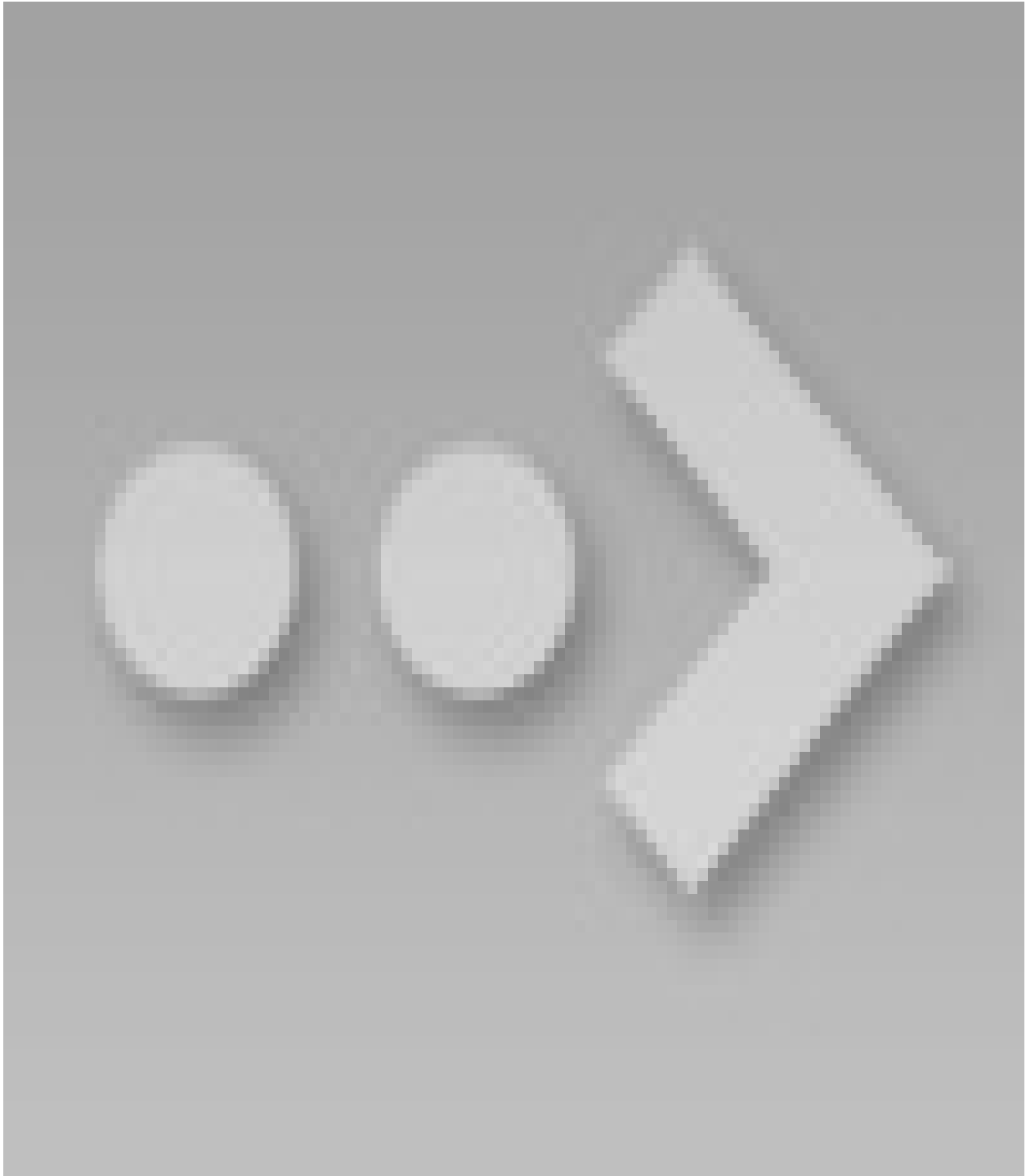
Once you have chosen the best Tableau product for you, it is time to start finding insights in your data! Much like Tableau's suite of products, data connections come in many shapes and sizes. As of this writing, Tableau Desktop: Personal has five different types of data connections, and Tableau Desktop: Professional adds another 54 native ways to connect to data. That doesn't even count the ability to access web data through customized connectors or Open Database Connectivity (ODBC).

As you can imagine from the breadth of connection options, you can connect to almost any type of data in Tableau and if you don't see the connection you are looking for, somebody is likely working on a customized solution that will help. I could write an entire book on the different data connections alone, but they all work similarly and are fairly intuitive. So for the purposes of this chapter, I will show you how to get started with one connection type and a few of the ways you can prepare to work with the data.

An introduction to connecting to data in Tableau

When you open Tableau, you will see a screen that looks like this, where you have the option to choose your data connection:

[Insert Image 3-1]



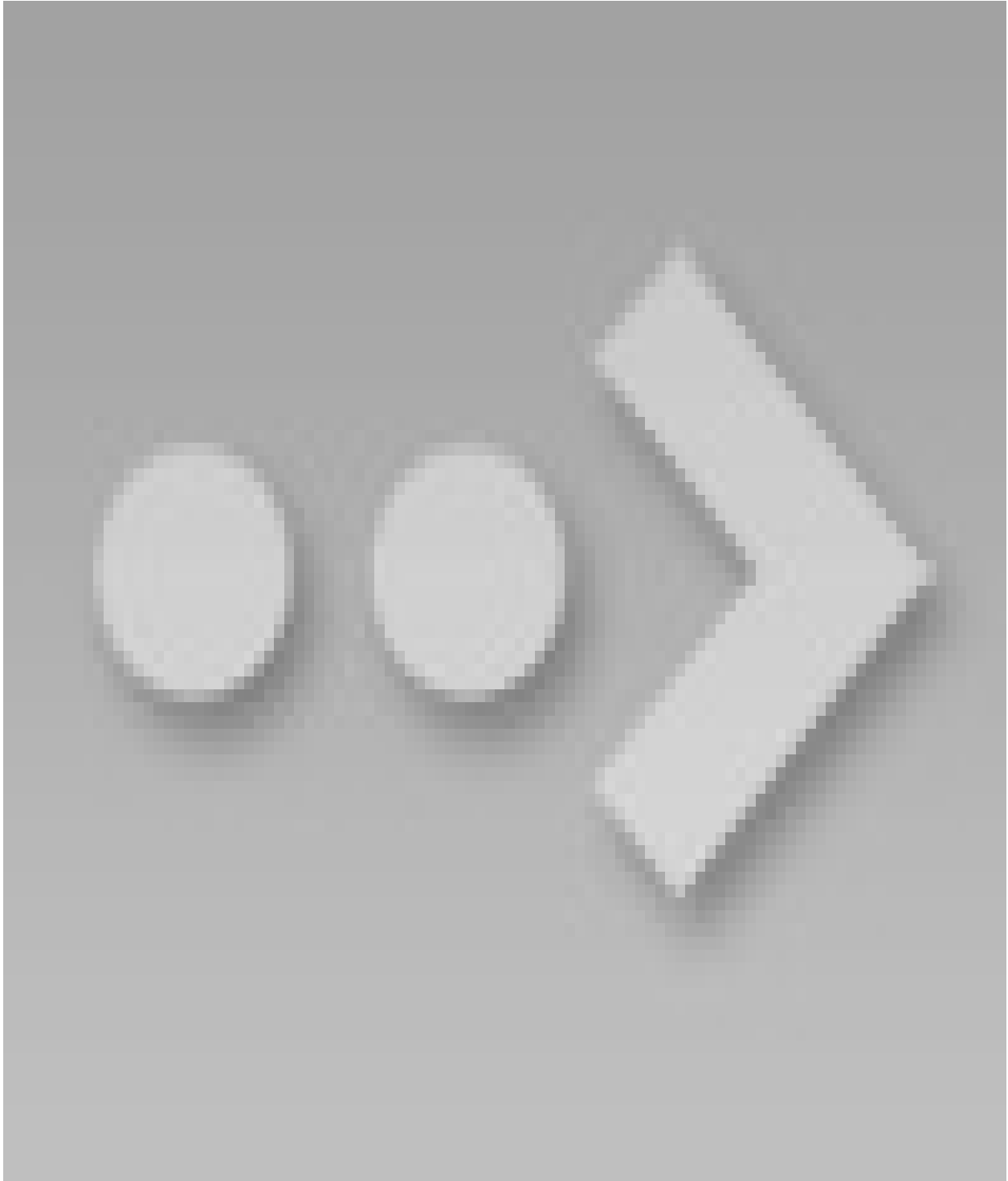
The options under the navigation heading “To a File” can be accessed with Tableau Desktop: Personal. All possible data connections, including to data that resides on a server, can be accessed with Tableau Desktop: Professional.

At the bottom of the left navigation, there are a couple of data sources that come with every download of Tableau. The first, Sample – Superstore, is actually an

Excel file, so you can connect to it whether you are using Tableau Desktop: Personal or Tableau Desktop: Professional. I like to train with this data source because it is the most common data source used in online tutorials and during Tableau’s own training. To start using it, click on it.

After clicking on a saved data source, you are immediately thrown into the authoring interface. We discuss getting a lay of the land [[link to chapter 5 – Getting a Lay of the Land](#)] in another chapter, but I actually want to take a step back to show you what happens when you normally connect to a new data source. To get to the data editing interface, click the “Data Source” tab in the bottom left corner of the authoring interface. You should be taken to a screen that looks like this:

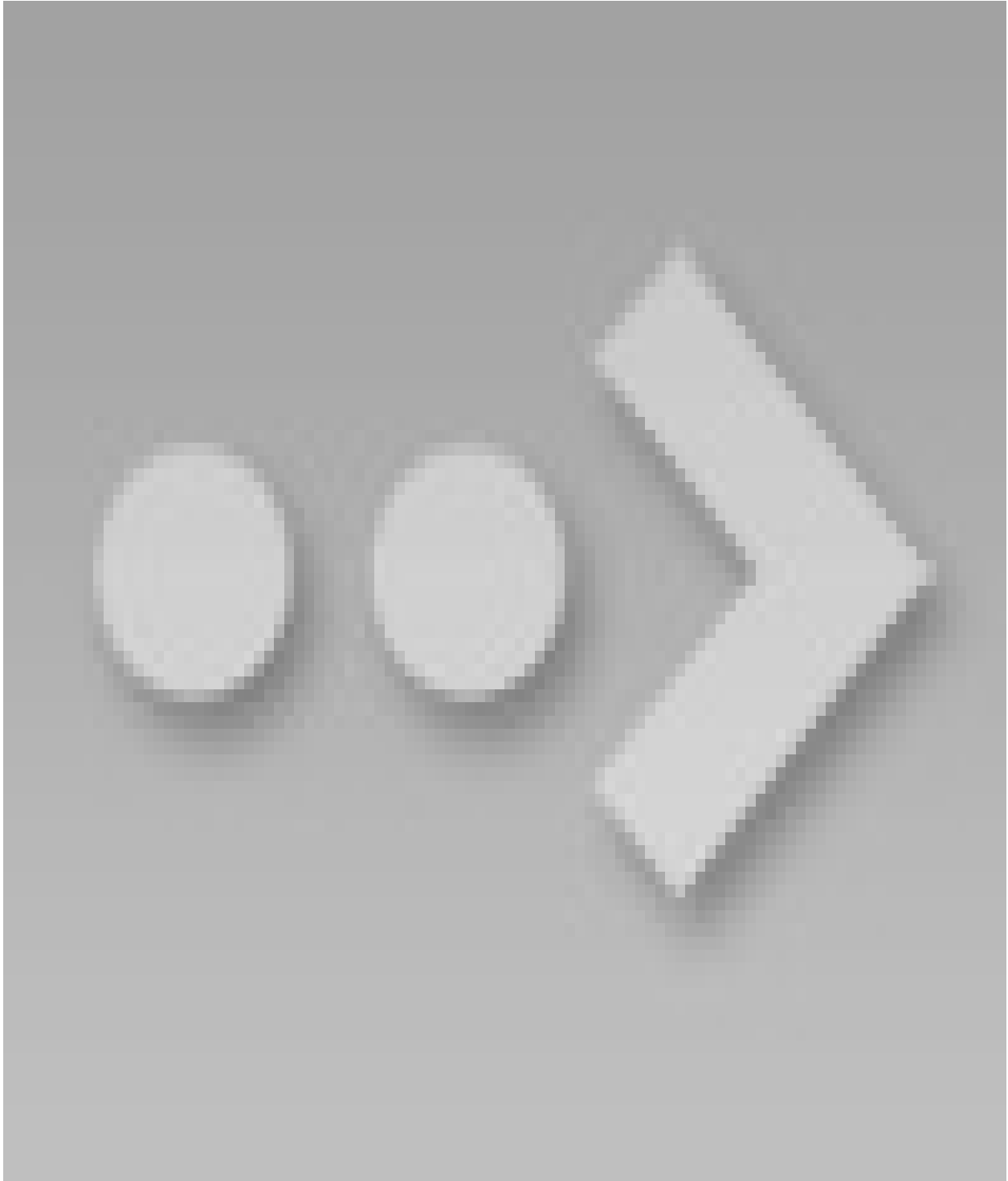
[Insert Image 3-2]



This is the screen you will be presented with when connecting to an Excel or database connection. In Tableau, the Excel workbook is treated as a database and the individual tabs are treated as individual tables within that database. For this reason, you can join tabs to each other if they have at least one field in common.

When you join tables, you are appending additional fields to your data source based on shared fields. To do so, simply drag the table (tab) that you want to join into the data editing interface and tell Tableau what the two tabs have in common. Here's what the Sample – Superstore dataset looks like after I dragged the Returns table into the view and set up a *left* join on Order ID. This means that every field that has a matching Order ID in the table on the right will be appended to the table on the left.

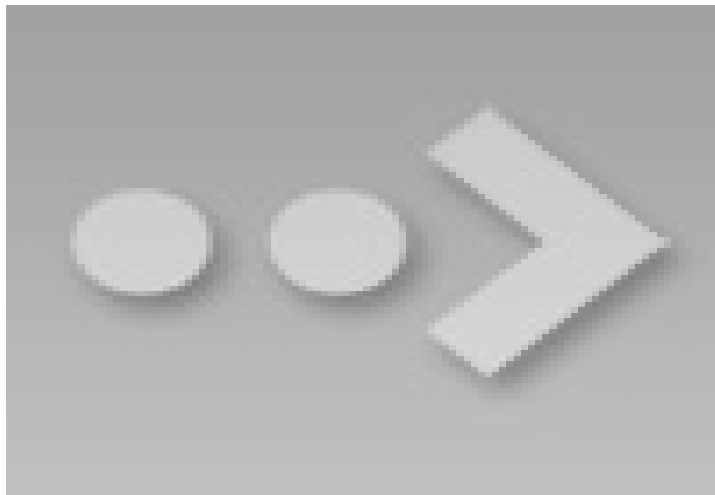
[Insert Image 3-3]



You can even do cross-database joins, even if the data come from different types of data connections. To do this, you would click “Add” to the right of “Connections”, connect to your additional data source, and set up a join just as pictured in the previous image.

If you're working with multiple tables that all have the same column headers, it may make more sense to union, or stack, the tables instead of joining them. Maybe you've got twelve months of web analytics data in one Excel file, but each month's data live on a separate tab. To union the twelve tabs, you would drag "New Union" from the left navigation onto the data editing interface, then drag the tables that you want to union into the box that appears. When you create a union in Tableau, a column will be added that tells you what sheet the data came from.

After you've retrieved the data you want to work with, there are a few more options for preparing each column. To access them, click the down arrow next to the data type icon for each column: [Insert Image 3-4]



The options are:

Rename: Allows you to rename the field

Copy Values: When nothing is selected, places the value in the first row on your clipboard (preselecting rows before choosing this option will copy your selection)

Hide: Hides the entire column

Aliases...: Allows you to assign new names to individual dimension members

Create Calculated Field...: Allows you to create a new field before you start using it within Tableau

Create Group: Allows you to group different dimension members. This can be handy for quick data clean-up.

Split: Tableau will look at the dimension members in your column and guess the most appropriate way to split them into multiple columns

Custom Split...: The same as split, but you determine how to separate the dimension members

Pivot: When you have multiple columns selected, you can transpose them. Note you can only do one data pivot per data source.

Describe: Gives you additional information about the field

When going through this process for quantitative fields, the string functions are not available, and one additional option is available: “Create Bins...”. This creates equally-sized bins, which can be used to make histograms. We will discuss how to make histograms in a later chapter.

Lastly, you can also change the data type of a column by clicking the data type icon at the top of the column.

It’s important to note that any changes you make to the data at this point creates metadata and has no impact on your underlying data source. This means you can make rapid progress in Tableau without the risk of messing up your existing infrastructure.

Another notable task that you may choose to do when connecting to a new data source is to either extract or filter the data source. By default, most data source connections will be live with no filters; these options can be seen in the top right corner of the data editor: [Insert Image 3-5]



Extracts create a snapshot of your data at whatever point they were created. They are typically faster than a live data connection, especially when connecting to a live database, and are my general recommendation. Just remember that extracts have to be refreshed periodically so that you are working with the latest data possible. From within Tableau Desktop, extracts can be refreshed by navigating to “Data” in the top navigation, hovering over the data source you want to refresh, then hovering over “Extract”, and clicking “Refresh”. If you eventually use Tableau Server, the cloud-based version of Tableau, you will see option to automate the refresh process when you publish from Tableau Desktop.

The final option discussed in this chapter is the ability to filter the entire data source before you start working with it in Tableau. These filters can be created

source before you start working with it in Tableau. These filters can be created with any combination of fields by clicking the “Add” button under “Filters”. This is an easy opportunity to make your workbooks more efficient because you have the ability to filter out the data you don’t need for your analysis. For example, if your analysis is about this year’s data, don’t pull in the last ten years of data! Or maybe you are building the workbook for a stakeholder that is only responsible for one division and they’re not allowed to see the performance of other divisions. Adding a filter in this scenario not only makes the workbook process more efficiently, it will help you manage security to ensure data does not fall into the wrong hands.

With all of these choices, you should be able to set your data up exactly as you wish before you start working with it. However, if you are trying to transition existing Excel reports or working with irregularly shaped data, you may benefit from reading on into chapter 4, Shaping Data for Use with Tableau before you get seriously down to work.

Chapter 4 – Shaping Data for Use with Tableau

The second thing I wish I knew the first day I used Tableau is that there is an optimal way to shape data for use with the software. I’ll never forget the day I was introduced to Tableau. The boss walked in and asked three of us in the office to try out this new tool they had heard of for creating data visualizations. The first thing every one of us tried to do is connect to an existing Excel report and recreate it in Tableau. After all, this was supposed to be intuitive – perhaps even *magical* - software, right? We quickly found out that nothing worked as we expected, we couldn’t figure out how to make a single chart, and we had to fight the temptation to immediately revert back to our familiar Excel experience.

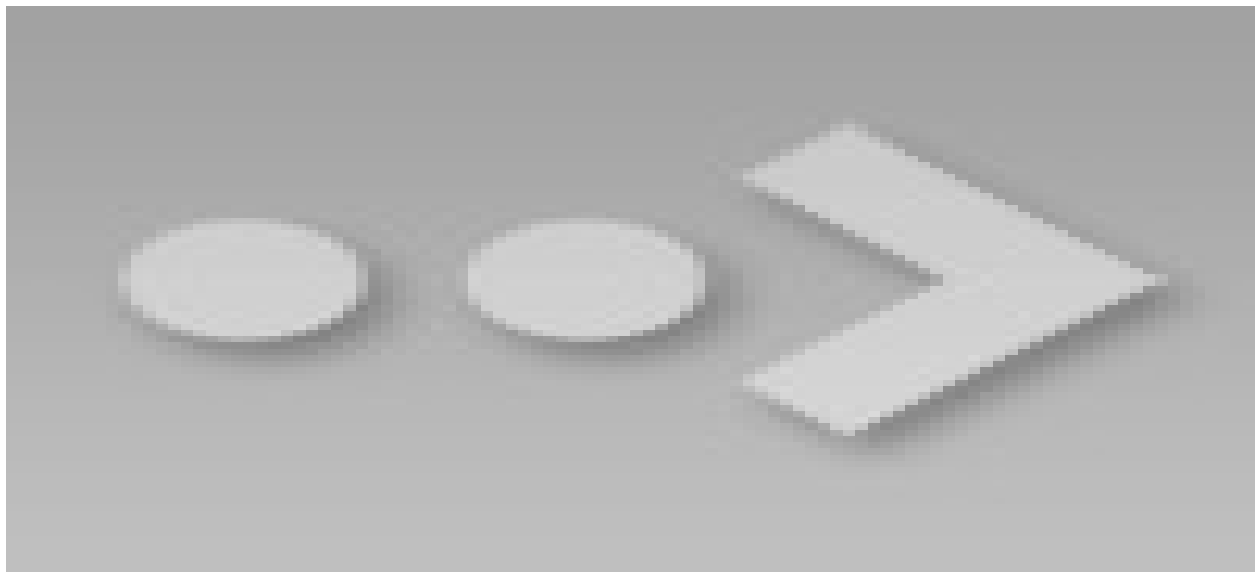
It’s fun to look back, and this now seems like a simple problem to solve, but the scenario I experienced my first time with Tableau is not uncommon. In fact, it’s both the most common Tableau adoption scenario that I come across—and the most difficult: First-time users connecting to an existing Excel report without any consideration to the format of the data.

Most existing Excel reports are not set up to work well with Tableau, and if this is the first data source that a first-time user attempts to work with, they are setting themselves up to fail. But don't despair - I can personally attest to what it's like to start using Tableau without any data or visualization software experience. I've always said that there are three or four key things to know when getting started with Tableau, and sometimes you just need somebody to share them with you so you can connect the dots and get started.

Shaping data for use with Tableau

This topic is easiest to illustrate, so take a look at the first image, which is meant to be similar to a typical report in Excel:

[Insert Image 4-1]



There is a title along the top, a column header for each quarter, and a row for each KPI (Sales, Profit, and Orders). In addition, there is a total for each row on the right-hand side of the table.

The format of this report poses several problems for Tableau which, upon connecting, will try to interpret the data source, classify the fields, and set up your workspace:

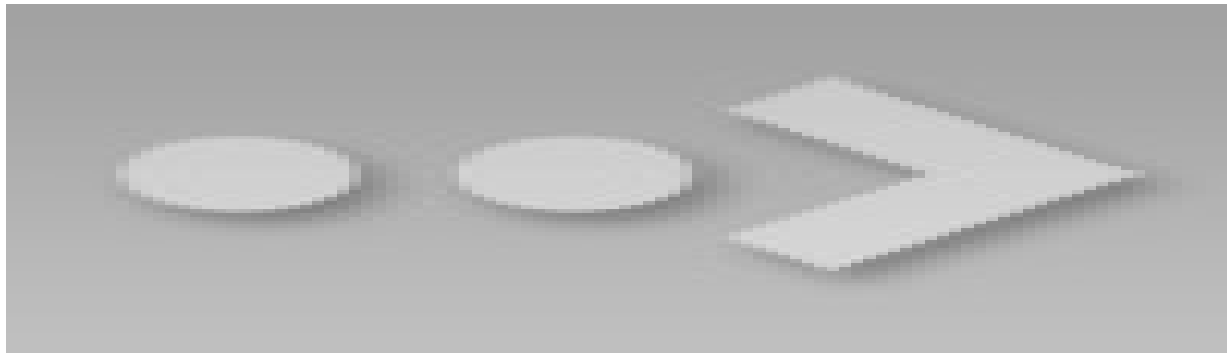
1. There is a title in the first row. The first two rows are critical for Tableau to interpret the data source, so we've immediately gotten off on the wrong

foot.

2. 2. The column headers are quarters, which will cause Tableau to create a field for each quarter, when in fact the quarters should all be consolidated into one field for date / quarter.
3. 3. The KPIs are running down the first column so, by default, Tableau will not interpret these KPIs as unique fields.
4. 4. There is a total in the right column. As Tableau totals fields for you, not only is this unnecessary, it will likely lead to double-counting.

The ideal format for Tableau looks like this:

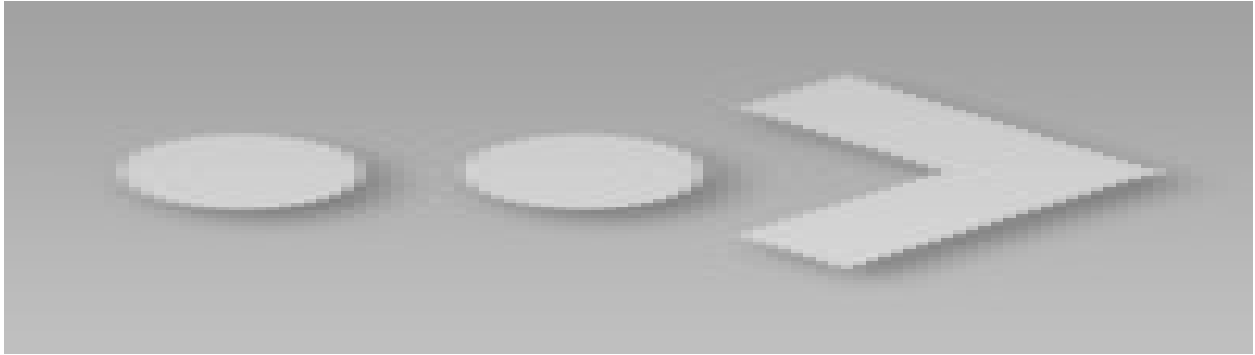
[Insert Image 4-2]



Each column now represents a unique field, so the layout is vertical instead of horizontal. The title and totals have also been removed.

With the data in this shape, Tableau will be able to look at the first row to determine the fields and the second row to classify the data (i.e. type, discrete vs. continuous, dimension vs. measure). We will discuss the ways Tableau classifies data in the next two chapters in this series.

As one additional tip, if your dataset includes a date field that is not in a traditional date format (as we've shown here with quarters), I recommend adding a column that looks like an actual date. In this case, I've added a column for quarter as date, and chosen the first date in each quarter as the entries: [Insert Image 4-3]



Dates are a special data type in Tableau and by having dates in a date format that the software recognizes, the full functionality of date fields is unlocked.

Finally, if data reshaping is required for you to work with a dataset in Tableau, you can reshape it prior to connecting – which is my personal preference - or use Tableau’s data interpreter and data pivot tools when you connect. Regardless of the method you choose, putting some thought into the shape of your data will help you get off to a strong start with your analyses in Tableau.

Chapter 5 – Getting a Lay of the Land

This chapter provides an overview of the Tableau interface, terminology, and a couple of things that I like to do first whenever I start working with a new dataset. While this is certainly not an exhaustive list, it will help you get started authoring in Tableau immediately and will provide a foundation for what’s to come.

Tableau terminology

To walk through some of the most important Tableau terminology, we will use the following key followed by names and definitions:

[Insert Image 5-1; O’Reilly production to add key numbers from original image in own style]

1. Data Window – Displays all of the data connections in the workbook. Note that only one data connection (Sample – Superstore) is being displayed in this example, but you can connect to more than one data source at a time.

2. Dimensions Shelf – A list of all of the fields in the data source classified as

2. Dimensions Shelf – A list of all of the fields in the data source classified as dimensions (discussed in the next chapter).
3. Measures Shelf – A list of all the fields in the data source classified as measures (discussed in the next chapter).
4. Pages Shelf – The Pages Shelf allows you to pivot through dimension members and/or add animation to a view. For example, you can put a dimension for Month of Order Date onto the Pages shelf and have the view rotate through one month of data at a time.
5. Filters Shelf – Any dimension or measure that you filter a view by will be displayed here.
6. Marks Card (Marks Shelf) – Each square in this area is called a Marks Card, which are called that because they influence the marks on the view. Each Marks Card resides on the Marks Shelf. Note that depending on the chart type you are creating, additional cards will show up such as for Shape or Path.
7. Legend – There are several different legends that will appear here to show how the marks are encoded including Color (pictured), Size, and Shape.
8. Columns Shelf – Fields placed here will create columns on the view.
9. Rows Shelf – Fields placed here will create rows on the view.
10. Worksheet / View – Each tab in a Tableau workbook is called a worksheet and the area that displays a data visualization is a view.
11. “Pill” – The slang term for fields being used on a worksheet.
12. Mark – Each data point on the view.

I will close this chapter by sharing two things that I like to do when I open a new dataset in Tableau:

View the underlying data

If I am working with a new dataset that has been provided to me, I like to get a general feel for the types of data each field provides. You can easily view this in a tabular form in Tableau in a few different ways:

- Right-click on the data connection in the data window and choose “View Data...”
- Click Data in the top navigation, hover over the data connection, and choose “View Data...”
- Click on the first tab in the bottom of the worksheet view

Whichever method is used, this provides a snapshot of the data so that you can

get a general feel for the types of data that are available, if some fields contain nulls, what fields may be most useful to ‘slice and dice’ your measures, *etc.*

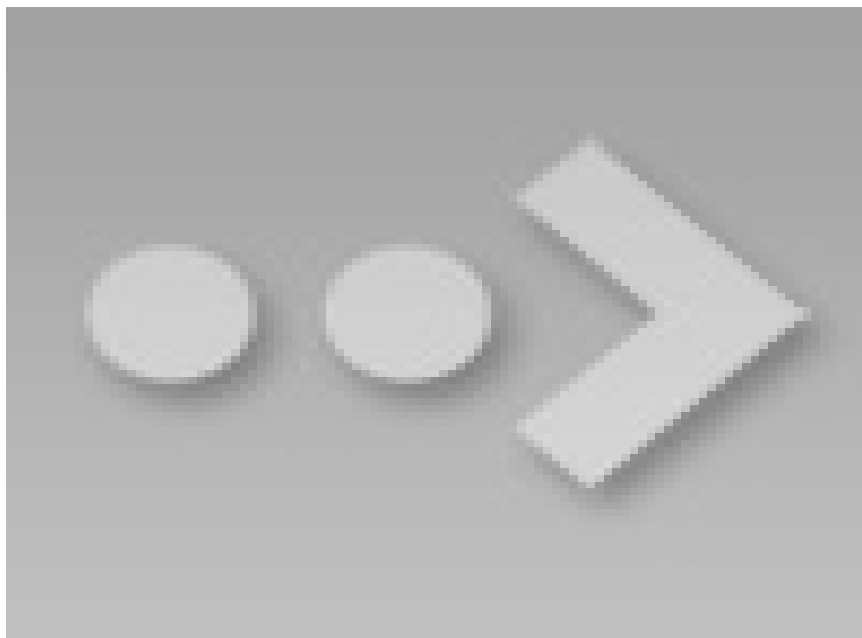
If you want to look at one field at a time, a handy trick is to right-click on that field on the Dimensions / Measures Shelf and choose “Describe...”. A window will pop up to provide you with helpful information about the respective field.

View the number of records

If you look at the bottom of the list of measures on the Measures Shelf, you will see a field called Number of Records. This is a special field that Tableau automatically generates for you, which is indicated by the italic formatting. Number of Records is actually a calculated field that simply equals 1. What this does is adds a column with an entry of 1 to each row of your data, so Tableau can count the number of records in the dataset.

To view how many records are in your dataset, go to a new worksheet and drag the Number of Records field to the Text Marks Card.

[Insert Image 5-2 (uploaded to Google Drive)]



I like to do this when I am getting started with a new dataset for two reasons:

- This provides a quick quality assurance check. For example, if the number

of records is lower than you expected, you may need to ensure you haven't filtered the dataset or that the data provider gave you the correct output.

- Having a ballpark idea of the number of records in a dataset can provide a clue on whether I need to take extra steps to keep the workbook running efficiently. If I find that I am working with a very large dataset, I may consider additional ways to aggregate the data and / or keep the size of the data in mind when creating calculated fields (as some are more efficient than others).

After I've got a feel for the size of the data and the type of data I have available, I start creating views and doing some discovery analytics, which we will discuss in future chapters.

Chapter 6 – Dimension vs. Measure

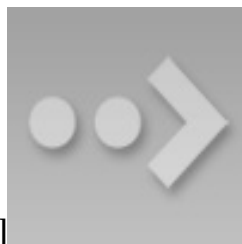
The third thing I wish I knew the first day I used Tableau is that when I connect to data, the software will classify each field as a dimension or a measure.

Tableau will then group the fields by their dimension or measure classification on the left-hand side of the workspace. Having a good understanding of the differences between dimensions and measures makes it much easier to work with the data in Tableau.

What is a measure?

According to Tableau's Knowledge Base, a measure is a field that is a dependent variable; that is, its value is a function of one or more dimensions. Tableau treats any field containing numeric (quantitative) information as a measure.

Consider the following bar chart, created in Tableau with the Sales measure from the Sample – Superstore dataset:



[Insert Image 6-1]

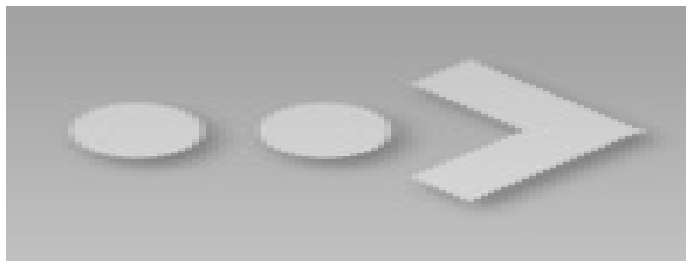
Sales is quantitative, so by default, Tableau will guess that the field is a measure. It can be considered a dependent variable, because a measure by itself does not provide much value. The value of \$2,297,201 is meaningless by itself. It is dependent on context that comes in the form of being broken down by dimensions.

What is a dimension?

According to Tableau's Knowledge Base, a dimension is a field that can be considered an independent variable. By default, Tableau treats any field containing qualitative, categorical information as a dimension.

Here is the same Sales measure from above, broken down by the dimension of Region:

[Insert Image 6-2]



Now that our sales total has been broken down by region, we are able to start gaining insights from the data. One insight that emerges is that the South region has relatively low sales compared to the other regions. This is a descriptive insight that materialized only when we combined measures and dimensions together.

Generally, the measure is the number; the dimension is what you “slice and dice” the number by.

That being said, there can be exceptions to this rule, so it helps to understand how Tableau treats these types of fields. Consider a business that has unique numerical order IDs for each sale (i.e. Order 1 is assigned the number 1, order 2 is assigned the number 2, and so on). Looking at the definition above, Tableau will classify this Order ID field as a measure the first time you connect to a

will classify this Order ID field as a measure the first time you connect to a dataset containing the field. However, Order ID is a dimension because we would ‘slice and dice’ a measure, such as Sales, by Order ID to see how much revenue we brought in per order.

Another rule of thumb I follow is that if it doesn’t make sense to sum up a number, it is likely a dimension. That’s the case with the hypothetical Order ID field just mentioned. There would never be any value in adding up all of our Order ID numbers to get the total, and sure enough, this field should be a dimension instead of a measure.

One more case that comes to mind where Tableau can misclassify fields is when you have a field that should be a measure, that has the word NULL in the first entry under the column header in your data. The word NULL would be seen as a string value to Tableau, and thus qualitative, which would cause Tableau to classify the field as a dimension.

The good news is that any field that is misclassified can easily be reclassified by right-clicking on the field from within the Dimensions Shelf or Measures Shelf and choosing “Convert to dimension” or “Convert to measure” as appropriate. The same thing can be achieved by dragging and dropping the field into the Dimensions Shelf or Measures Shelf.

An understanding of how dimensions and measures work in Tableau, combined with the basic data preparation just mentioned (when applicable), will make it easier to create visualizations moving forward.

Chapter 7 – Discrete vs. Continuous

Another thing I wish I knew the first day I used Tableau is the difference between discrete and continuous fields. This classification has an impact on what types of visualizations you can create as well as how they will look, so having a good grasp on what this distinction means is core to your understanding of how Tableau looks at your data.

It is easy to know if a field is being used as discrete or continuous based on its color. Blue indicates that a field is discrete, while green indicates that a field is continuous. If your first guess was that these colors represented whether a field was a dimension or measure, you are not alone. The thought that blue represents

was a dimension or measure, you are not alone. The thought that blue represents dimensions and green represents measures is the most common myth in Tableau. It's easy to understand why because, by default, dimensions are categorized as discrete variables, and thus have a small blue icon in front of them in the Dimensions Shelf. Measures are categorized as continuous variables, so they are prefaced with a green icon in the Measures Shelf.

I assure you that the color-coding identifies discrete vs. continuous fields and not dimensions vs. measures. Measures can actually be used as discrete fields or continuous fields, and the same is true for some dimensions, such as dates.

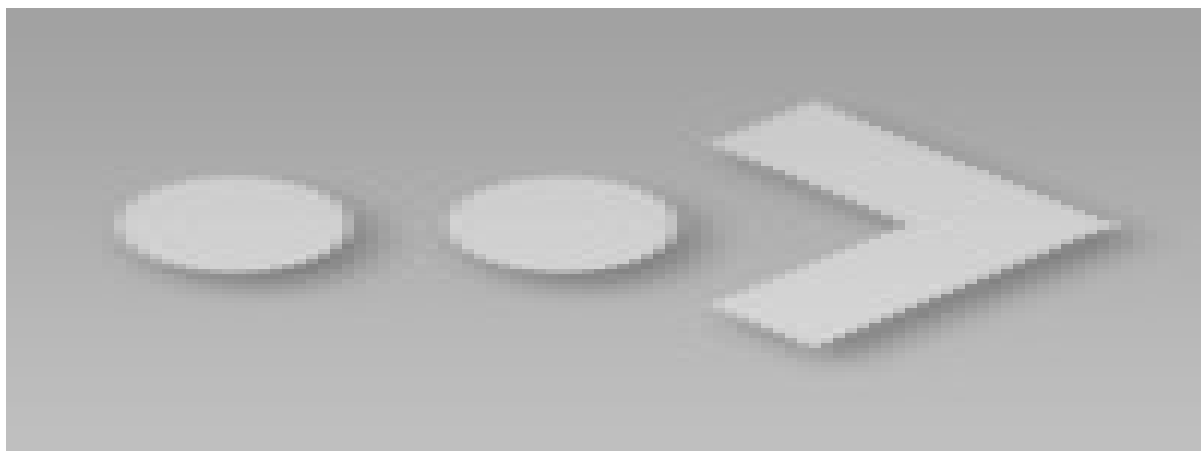
So what does this mean for your visualizations? I will illustrate using two rules of thumb I have when considering if a field should be used as discrete or continuous.

Discrete fields draw headers; continuous fields draw axes

Take a look at the following visualizations that looks at sales by month.

In the first chart, I am using date as a *discrete* field:

[Insert Image 7-1]

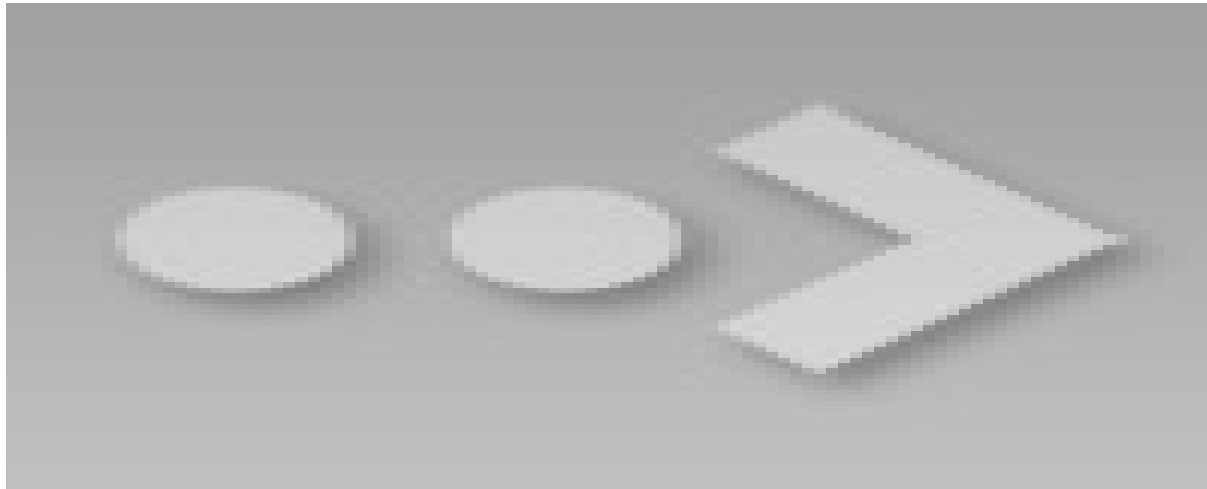


Notice that there is a *discrete* header for each month.

In the second chart, I am using the same exact data, but I have changed the Date

dimension from discrete to *continuous*:

[Insert Image 7-2]



As you can see, I now have a *continuous* axis of time. Since the axis is continuous, I cannot change the order of the dates; they follow a chronological order from oldest date on the left to most recent date on the right. On the other hand, when the Date dimension is being used as discrete as pictured in the first image, I am able to change the order of the dates. For example, I could sort the bars in descending order, with the month with the highest sales first, and the month with the lowest sales last. Which brings me to my second rule when determining whether I should use a field as discrete or continuous:

Discrete fields can be sorted; continuous fields cannot

So how you might use this in the real world. If you know that you want to look at a trend over a continuous period of time, you would want to use a continuous date, which will be colored green on the view. If your analysis requires you to have discrete marks that can be sorted, you would use the field as discrete, which will be colored blue on the view.

This date example is just one of many possibilities, but remembering the two rules outlined in this chapter will help you understand how the use of discrete and continuous fields are impacting the data visualizations you create in

Tableau.

Chapter 8 – 5 Ways to Make a Bar Chart / An Introduction to Aggregation

Now that we've gone through some fundamental topics such as dimension vs. measure, discrete vs. continuous, and you have an overview of the authoring interface, you're ready to start creating visualizations in Tableau. This chapter shares five different ways to create a bar chart and provides an introduction to the topic of aggregation.

Perhaps the most important lesson from this chapter is a line I hear myself saying almost every day: *there is always more than one way to do something in Tableau*. You will find your own techniques, form your own habits, and hear different opinions – and they likely will all have merit. You truly can take multiple paths to get to the same end result in Tableau. We are about to discuss *five* different ways to create a bar chart, and it's not even a comprehensive list!

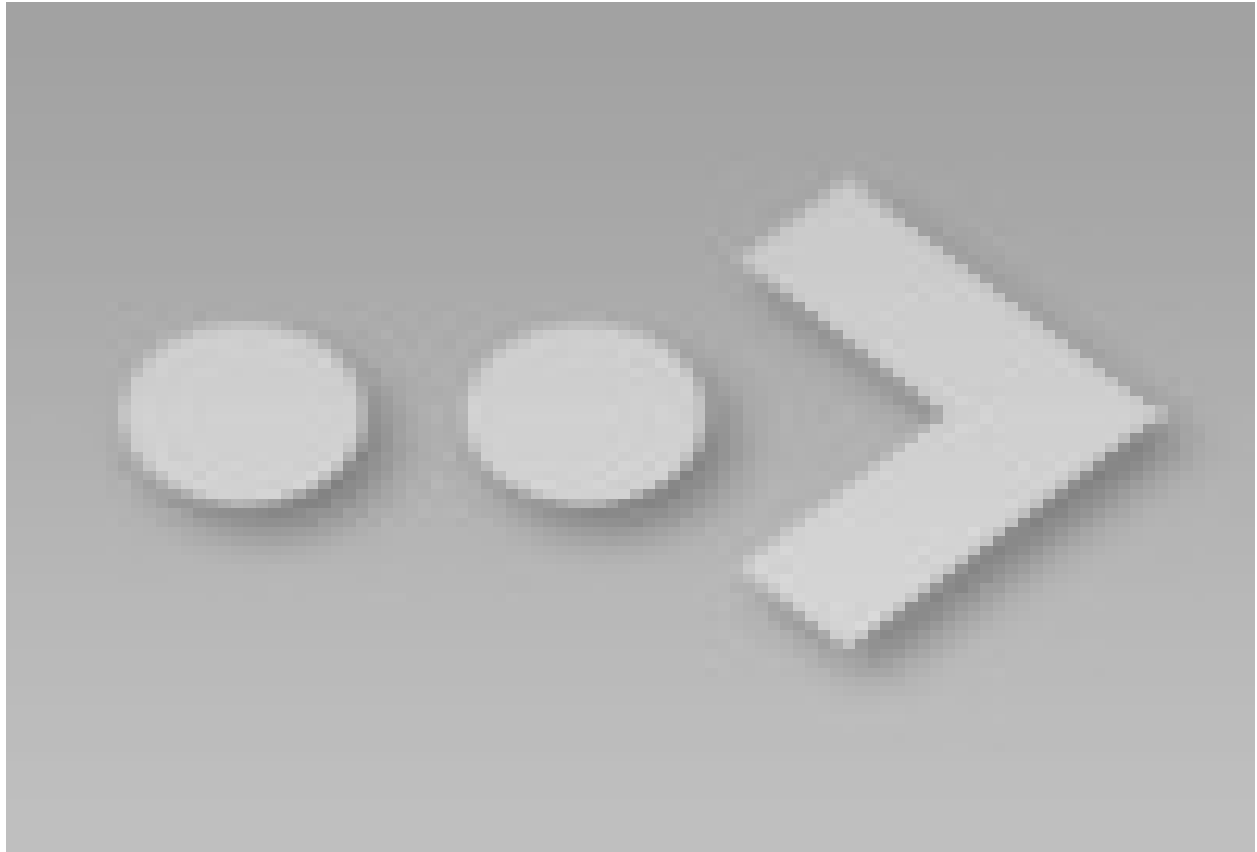
This tutorial will be using the Sample – Superstore dataset in Tableau and will show five techniques for creating a bar chart out of the Sales measure.

Five ways to create a bar chart in Tableau

1. The easiest way to start a bar chart in Tableau is to simply double-click on the measure you want to visualize from the Measures Shelf. Let's double-click on the Sales measure. By default, this will place a continuous pill for Sales on the Rows Shelf, which creates a vertical bar.
2. You could have got to this same place by left-clicking and dragging the Sales measure from the Measures Shelf to the Rows Shelf.
3. 'Preselect' the Sales measure by clicking on it, then click 'horizontal bars' in the Show Me options. This creates a different orientation than the first two approaches because the Sales measure is placed on the Columns Shelf instead of the Rows Shelf. If you prefer the vertical orientation, you

can click the Swap icon (pictured below), use the Ctrl+W shortcut, or drag and drop the Sales measure from the Columns Shelf to the Rows Shelf.

[Insert Image 8-1]



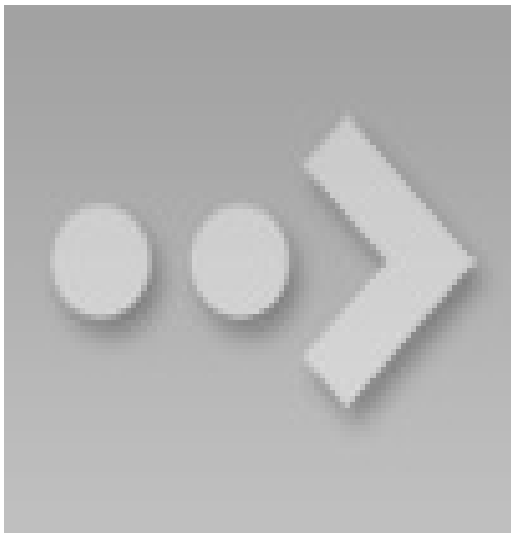
1. 4. You can change the mark type on an existing view to Bar. Let's say you are looking at the Sales measure by Year of Order Date as a line graph. You can convert the line graph to a bar chart by changing the mark type on the Marks Shelf from Automatic (line) to Bar.

[Insert Image 8-2]



1. 5. Similar to approach 2 above, but if you *right-click* and drag the Sales measure from the Measures shelf to the Rows Shelf, you will be presented with the option to choose the data aggregation before the bar chart is created.

[Insert Image 8-3]



An introduction to aggregation in Tableau

Every measure on a view in Tableau is aggregated in some way. It can be easy to not notice this when you're getting started because the default aggregation is SUM, and that works for most situations. Look back at the views created through the five approaches above; the Sales measure is preceded by the word SUM to show you how the field is being aggregated. There are several other aggregation options in Tableau, and the choice will influence your analysis. Here is a list of some of the options and the results you can expect to see for each choice (using the Sales measure): *SUM*: All of the sales added up together

AVG: All of your sales added up, divided by the number of records

MEDIAN: When sorted, the sales amount for the record in the exact middle of your data

CNT: A count of all records with sales

CNTD: A count of distinct sales amounts

MIN: The smallest sales value in your data

MAX: The largest sales value in your data

Knowing that there are always multiple approaches to the same solution in Tableau and having an understanding of aggregation will help tremendously as we start working with different visualizations and creating calculated fields in future chapters.

Chapter 9 – Line Graphs, Independent Axes, and Date Hierarchies

In the last chapter, I shared five different ways to make a bar chart in Tableau. This chapter walks you through how to build another kind of data visualization: the line graph. For all the fancy visualizations I enjoy building with Tableau, at the end of the day, bar charts and line graphs are two of, if not, the most effective options available. While we're at it, we'll touch on some related topics including date hierarchies and independent axes so you can master this foundational graph.

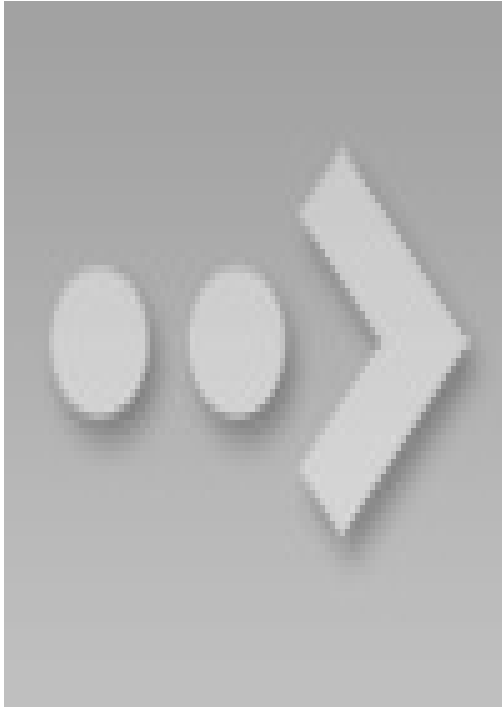
How to make a line graph in Tableau

A line graph is similar to a bar chart in Tableau in that you are looking at one or more measures with the option to ‘slice and dice’ it by one or more dimensions. The important distinction between the bar chart and line graph is that the line graph should include an element of time. Let’s build out a line graph using the Sample – Superstore dataset using Order Date as our element of time.

First, use any of the five methods discussed in the last chapter to create a bar chart that adds SUM(Sales) to the Rows Shelf. I’m going to simply double-click on the Sales measure to start the view.

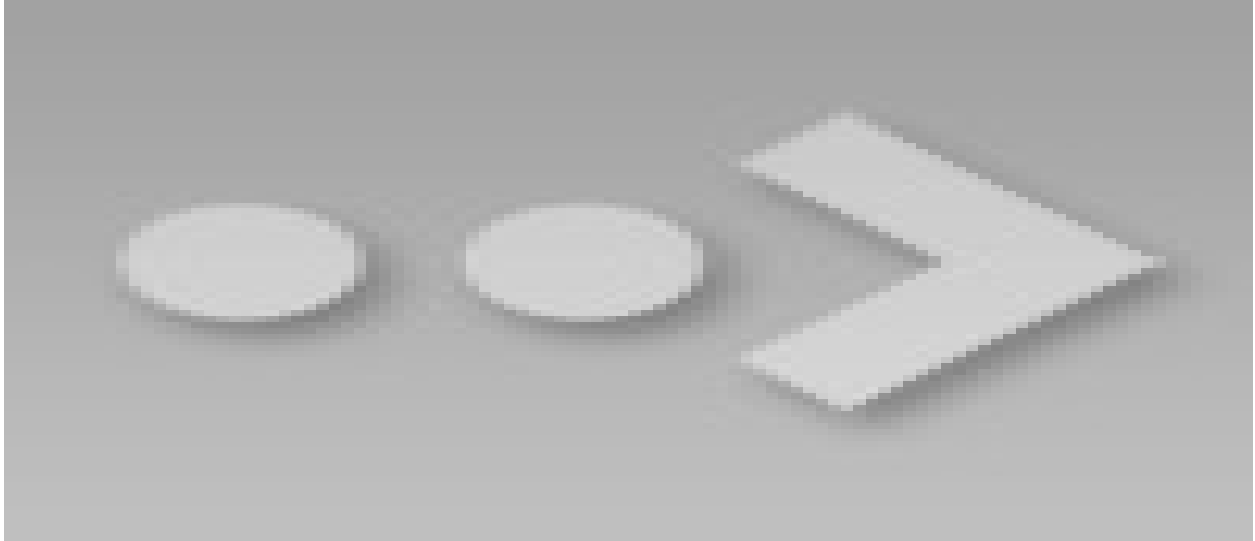
At this point, you could also double-click on your element of time (Order Date) to start a line graph, but there is a better way. As with measures, dates also have an aggregation and can be used as continuous or discrete fields. The choices you make for these two classifications will impact how the visualization will look. Fortunately, it is easy to see all of the options if instead of double-clicking on the Order Date dimension, you right-click and drag it to the Columns Shelf. Before Tableau generates the visualization, you will see the following options:

[Insert Image 9-1]



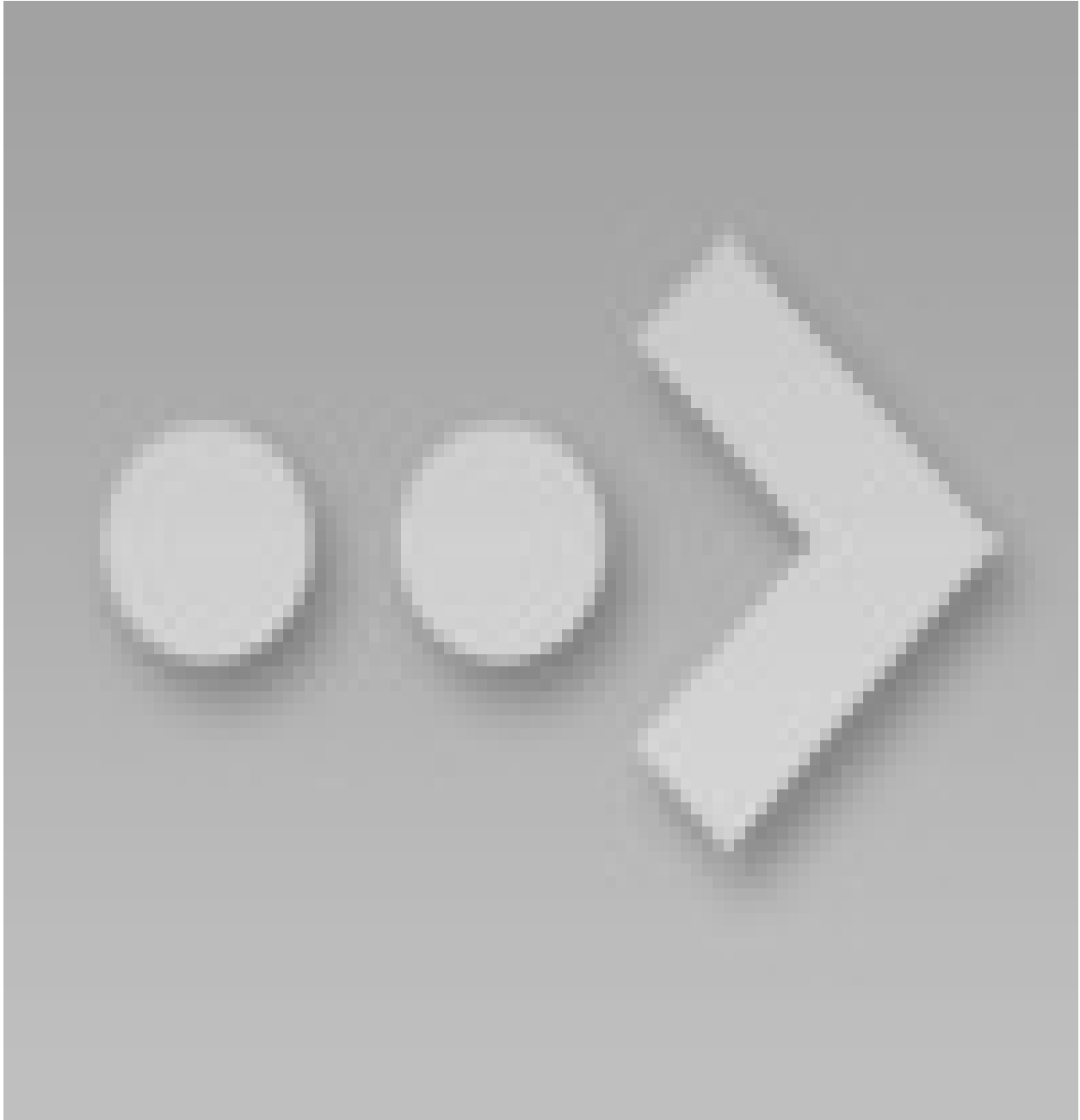
As you can see, each option has a blue or green icon immediately preceding it. Remember, the blue color coding in Tableau indicates discrete, while green color coding indicates continuous. The choice of whether you select a discrete or continuous option should be based on what type of visualization you want to create. Let's say that we want to look at a *continuous* trend over time, which means we can ignore the blue options for now.

The next choice we have is how granular we want the continuous trend to be. For this analysis, let's say we want to see a monthly trend over time. With these two requirements in mind, we know that we should pick the choice with a green icon with an aggregation of Month, which is the fourth choice from the bottom. After making this selection, the line graph looks like this: [Insert Image 9-2]

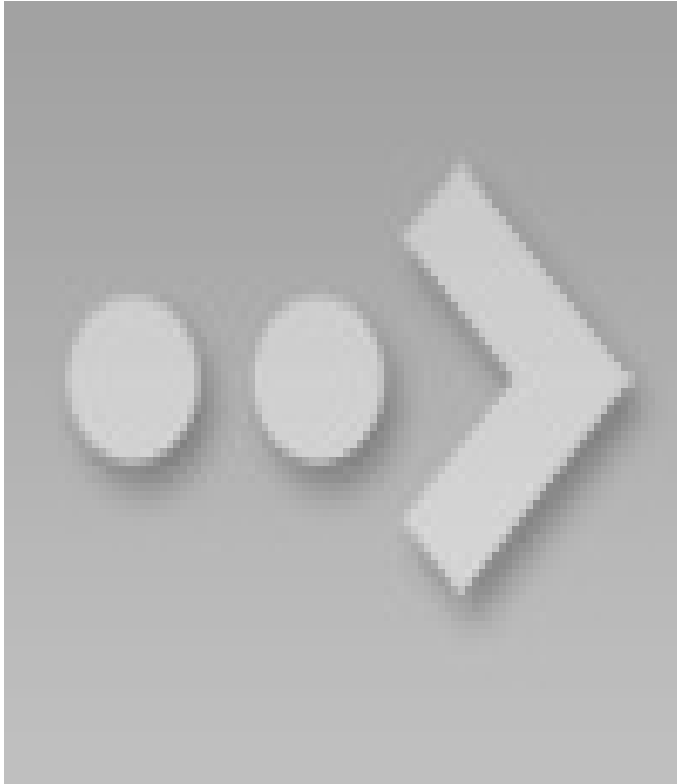


Independent axes in Tableau

The line graph created in the previous section can now be sliced and diced by additional dimensions by simply dragging them to the Columns Shelf or Rows Shelf and dropping them in front of the continuous fields on the view (currently MONTH(Order Date) or SUM(Sales)). To demonstrate how axes work across multiple columns or rows in Tableau, I will drag the Ship Mode dimension to the Rows Shelf: [Insert Image 9-3]

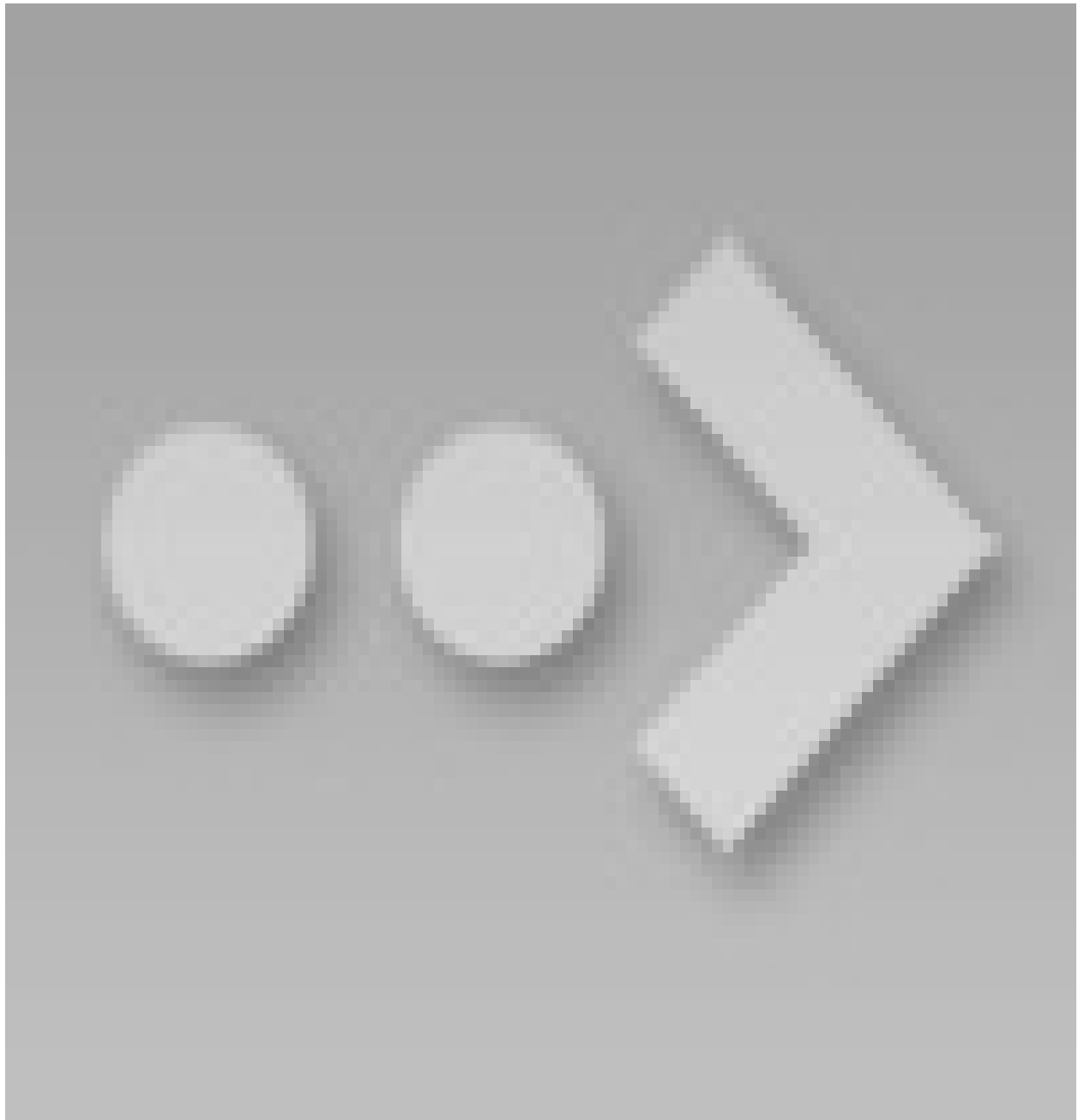


Notice that, by default, each row shares the same axis range from \$0 to approximately \$60,000, which is the largest range across all four ship modes. This default setting where the axes share the same range is helpful because it provides an ‘apples to apples’ comparison across the four rows. However, it can be difficult to see the trends for each individual row. If you would like the four axes to have their own axis range, right-click on any of the axes and choose “Edit axis...”. You will be presented with a dialog box where you can choose “Independent axis ranges for each row or column”: [Insert Image 9-4]



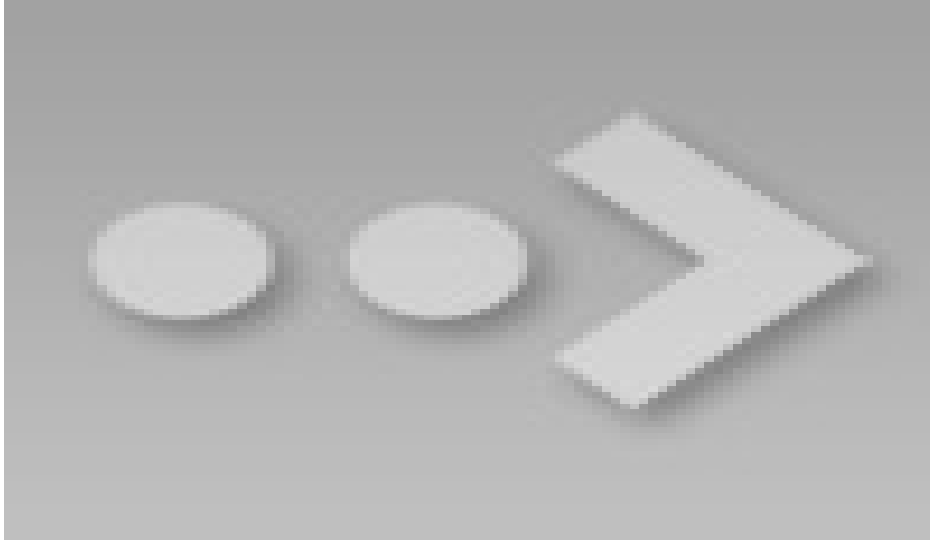
Upon making this selection, you will see your trend graph change so that each row has its own unique range:

[Insert Image 9-5]



Date hierarchies in Tableau

For the final section in this chapter, let's take a step back by building the line graph again, this time by first double-clicking on the Sales measure and Order Date dimension. At this point, your line graph should look similar to this: [Insert Image 9-6]



The blue YEAR(Order Date) pill is telling us that date is being aggregated by year and being used as a *discrete* field. When dates are being used as discrete fields, you will see a '+' sign on the field, which indicates this field has a hierarchy. Dates are one of the few special dimensions in Tableau which automatically receive a hierarchy, and it goes in this order: Year > Quarter > Month > Day.

To utilize the hierarchy, simply click the '+' sign on the field. The field will expand to the next level of the hierarchy and provide a different view. Here's how this same view looks if I click the '+' on Year, then click the '+' on Quarter: [Insert Image 9-7]



You can also drill back ‘up’ by clicking the ‘-’ sign that appears on fields where the ‘+’ has already been clicked. What’s nice about these fields is that, because they are discrete, they can be moved around into different orders to quickly create different analyses. The fields are processed in order, so with the last line graph pictured, the Sales measure is first cut by Year, then Quarter, then Month. To demonstrate a couple of possibilities, let’s first remove Quarter from the view, leaving us with Year and Month: [Insert Image 9-8]



At this point, we have a pretty standard analysis, where our years and months go in chronological order from oldest to most recent. This creates a seasonal analysis where you can compare four calendar years to each other. Look what happens when you simply change the order of Year and Month by dragging the Month pill in front of the Year pill: [Insert Image 9-9]



The fields are processed in order, so now the Sales measure is first cut by Month, then Year. This completely changes the analysis into a four-year trend per month.

For example:

While line graphs may seem simple on the surface, they are extremely powerful visualizations, and Tableau provides the flexibility to use them for many different analyses.

Chapter 10 – Marks Cards, Encoding, and Level of Detail

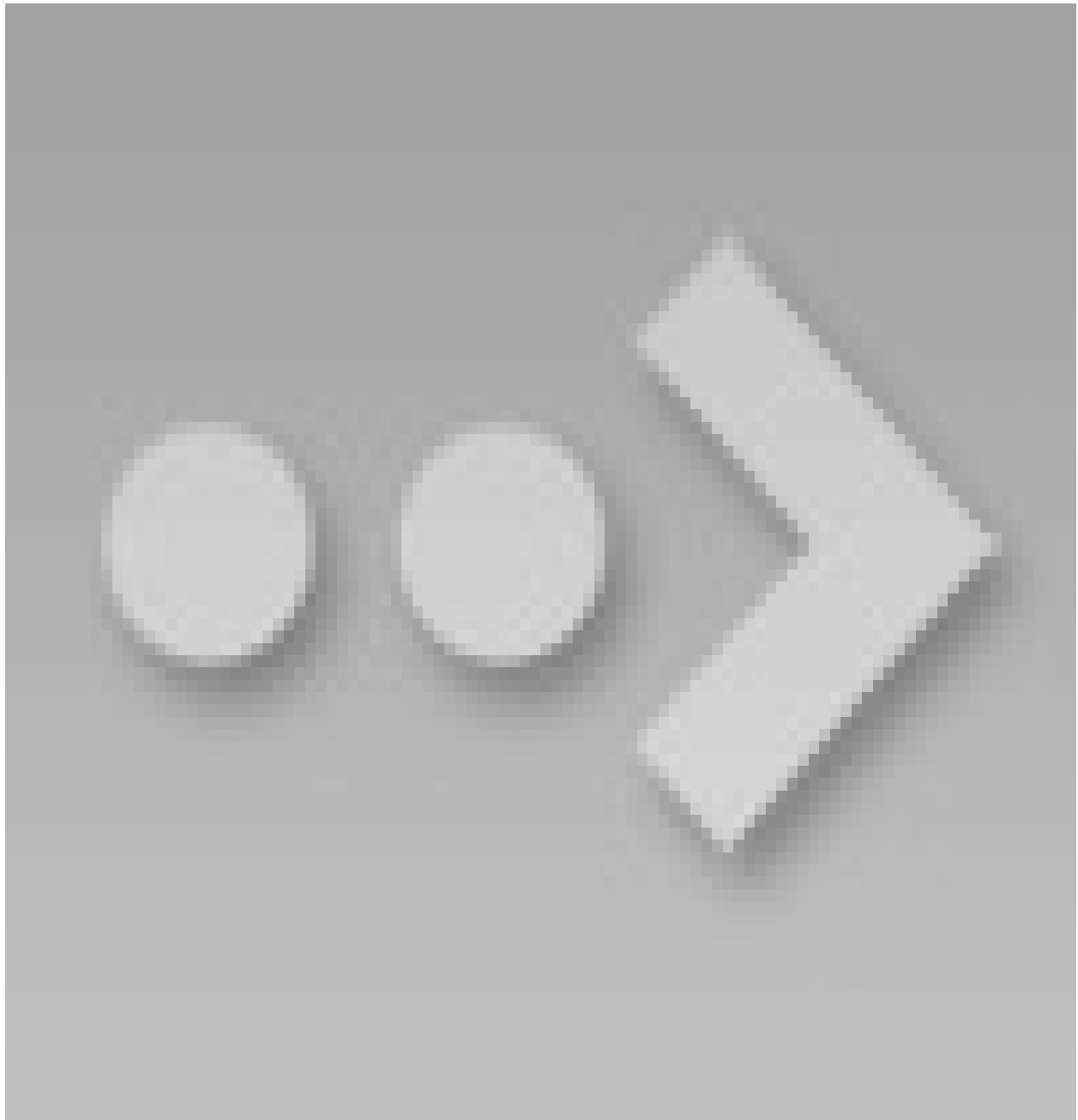
The Marks Cards in Tableau provide some of the most powerful functionality in the program because they allow you to modify a view's design, visualization type, user experience, and granularity of analysis all in one place. This chapter provides an overview of the Marks Cards available in the authoring interface as well as an introduction to the concept of a view's *level of detail*.

In the chapter, Getting a Lay of the Land, you saw that the default location of the Marks *Shelf* (or Shelves), which contain the Marks *Cards*, are located on the left side of the view to the right of the Dimensions Shelf and under the Filters Shelf. The Marks Shelf and Marks Cards include the word 'marks' because they change the marks, or data points, on a view.

An explanation of level of detail

To help illustrate how each Marks Card impacts the marks on a view, we will be using this simple scatter plot looking at Profit Ratio and Sales:

[Insert Image 10-1]

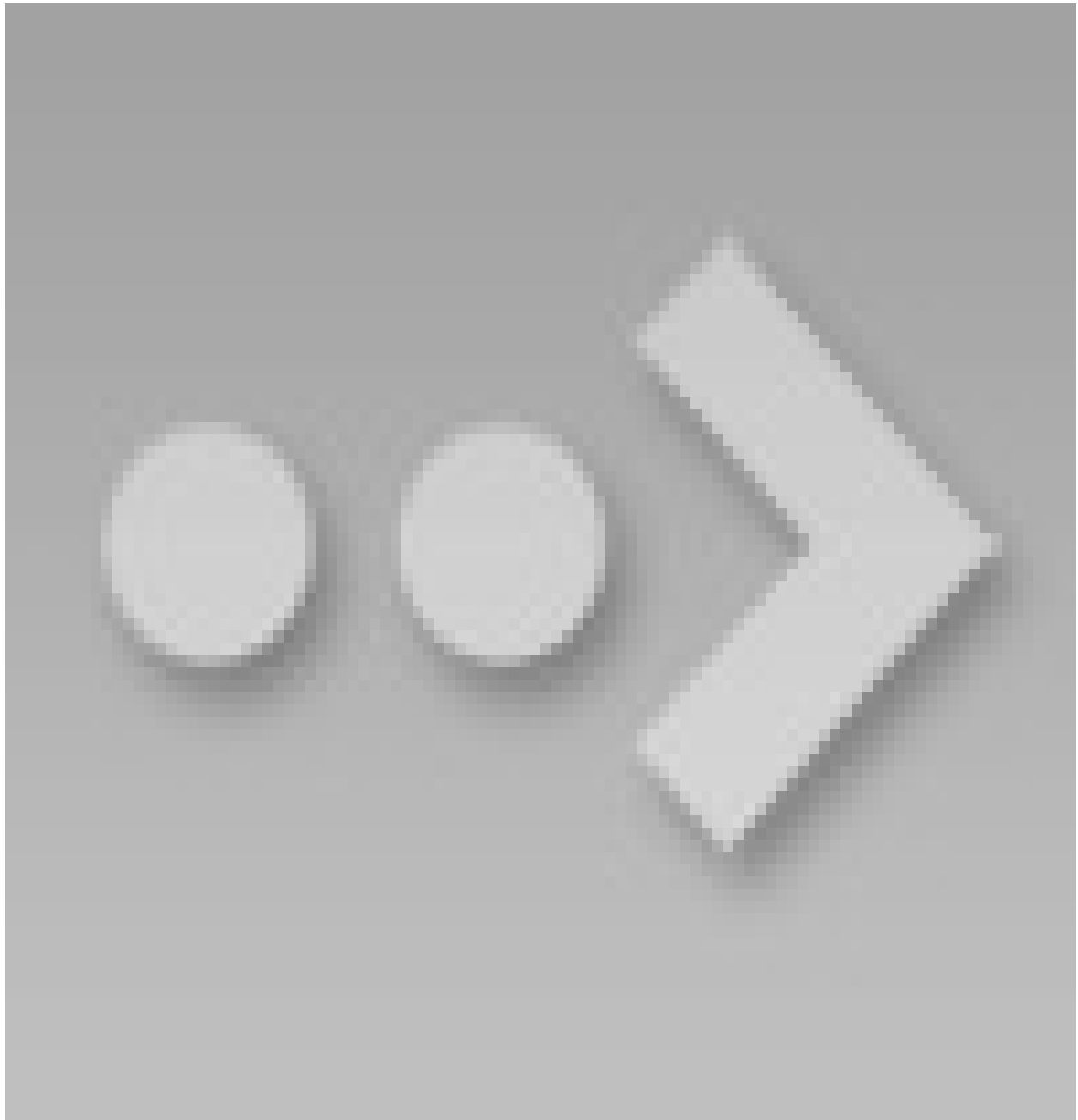


As you can see on the Marks Shelf, there are six different Marks Cards: Color, Size, Label, Detail, Tooltip, and Shape. Note that the Shape Marks Card is not available for every view, but appeared because we are creating a scatter plot in this example. To help explain what each of these cards does, I'm going to start a little out of order and discuss the Detail Marks Card.

I'm starting with the Detail Marks Card because level of detail is a critical aspect of analyzing data in Tableau, but it took me a while when I started using Tableau to realize its importance. It's not a difficult concept, but just one of those that

wasn't explained to me early on. Once you wrap your head around the topic of level of detail, your authoring in Tableau will feel much more fluid and it will be easier to use the powerful Level of Detail expressions functionality.

The first thing you need to know is that every visualization has a level of detail. It helps me to think about level of detail as the most granular level where the analysis takes place. Consider the scatter plot pictured above. At this point, we are not slicing and dicing the Profit Ratio and Sales measures by any dimension. For that reason, you see just a single mark, which represents the intersection of Profit Ratio and Sales for every record combined in the Sample – Superstore dataset. As you add dimensions to the view, the analysis becomes more granular, so the level of detail changes. For example, if we wanted to do this Profit Ratio versus Sales comparison at the customer level, we can drag and drop the Customer Name dimension from the Dimensions Shelf to the Detail Marks Card, which results in the following: [Insert Image 10-2]

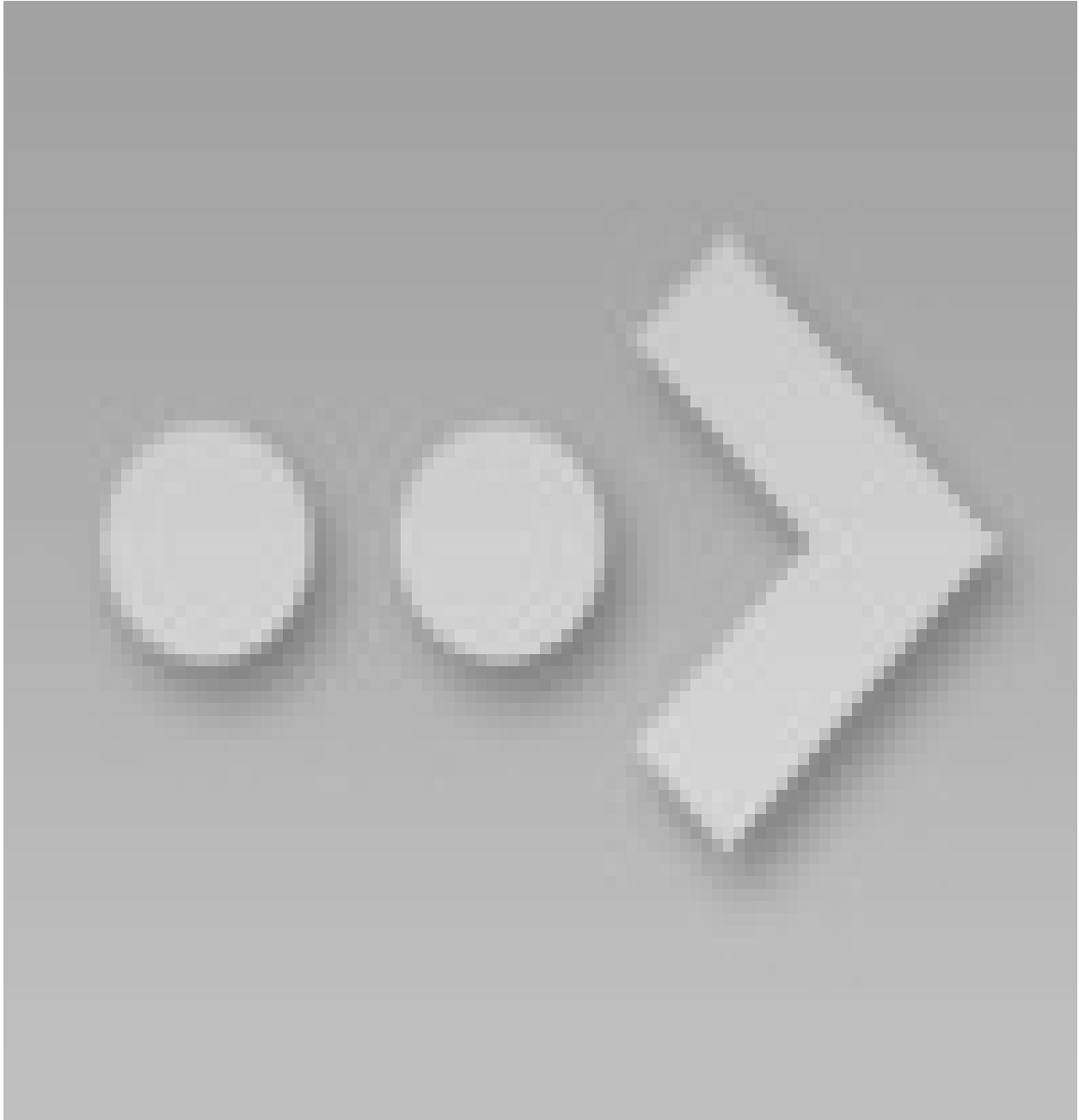


We have changed the most granular level of the analysis from the entire file to customer. Another handy feature of Tableau is you can look in the bottom left-corner of the worksheet for a summary of the view. When we changed the level of detail for this scatter plot, the mark count changed from 1 (the entire file) to 793 (the number of customers).

Note that adding dimensions in other places on the view will also change the level of detail, but the Detail Marks Card is available to make the analysis more granular if needed.

An introduction to encoding

The Color, Size, and Shape Marks Cards all allow you to “encode” the marks on a view. Encoding marks adds depth to an analysis by mapping marks to colors, sizes, and / or shapes to add context to a view. Here’s how the scatter plot view looks if we put Segment on the Color Marks Card, Sales on the Size Marks Card, and Category on the Shape Marks Card: [Insert Image 10-3]



There are now three different legends corresponding with the encoding that was just added. For consistency, this encoding will conveniently carry through on other views as they are created. For example, if I color a new view by the Segment dimension, Consumer will still be identified as blue, Corporate will still be identified as orange, and Home Office will still be identified as green. These colors can be changed by clicking on the color legend and mapping new colors.

These three Marks Cards can also be used to change all the marks on the view instead of mapping to a specific dimension. Instead of placing a dimension on

the Marks Cards, click each card to experiment with changing the color, size, or shapes for all of the marks at the same time. Simple changes to the Marks Cards can substantially improve the design of a data visualization.

Label and Tooltip Marks Cards

The Label and Tooltip Marks Cards can both be used to add written information to a view. The difference is that whatever information is added as a Label will show up on the view itself, while any information added to the tooltips will only show up when an end user hovers over marks on the view.

This is an important distinction that should be considered when you are authoring in Tableau. For example, if your visualization will be printed or copied and pasted as a screenshot, you would want to add the information to Label to ensure the information is shown on the view. On the other hand, if you know your end users will be interacting with Tableau, you may opt to save some on-screen real estate by providing the information through tooltips.

As with the other Marks Cards, labels and tooltips can be customized with specific information by dragging and dropping fields onto the Label and Tooltip Marks Cards, respectively. You can click into each of these two Marks Cards to toggle them on and off, change the formatting, and even type in additional information.

The six Marks Cards introduced in this chapter can dramatically improve the depth, design, and user experience of your visualizations. Utilizing this functionality will not only help your analyses, it will make your final product more effective with end users.

Chapter 11 – An Introduction to Filters

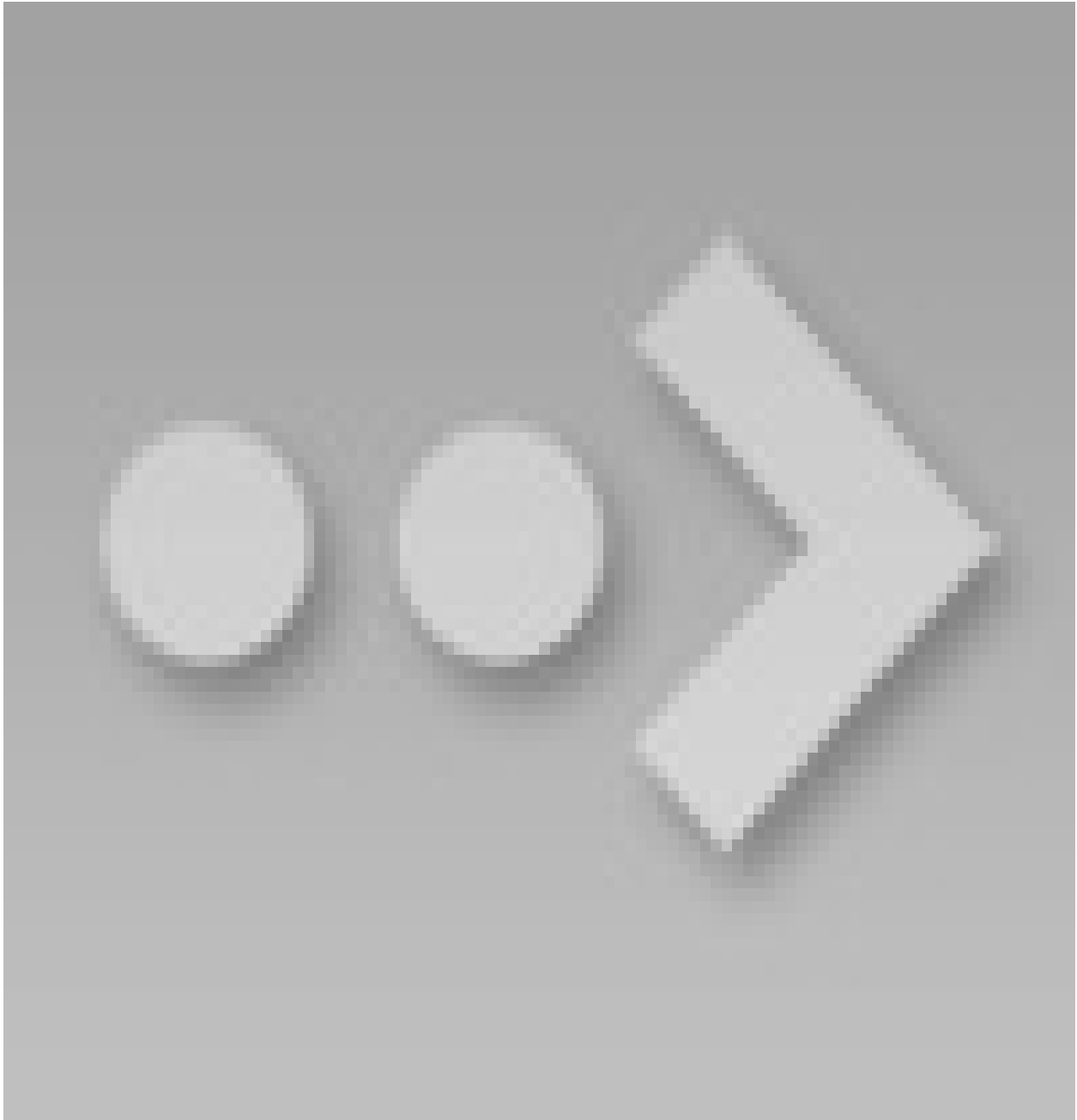
Tableau provides the ability to filter individual views or even entire data sources on dimensions, measures, or sets (discussed later in this section). This filtering capability can serve a variety of purposes including minimizing the size of the data for efficiency purposes, cleaning up underlying data, removing irrelevant

dimension members, and setting measure or date ranges for what you want to analyze. What's more, most of these filters can be put into the hands of you and your end users to change – a powerful tactic for finding stories in the data. This chapter offers an introduction to filters in Tableau including extract filters, data source filters, dimension filters, and measure filters.

Dimension filters in Tableau

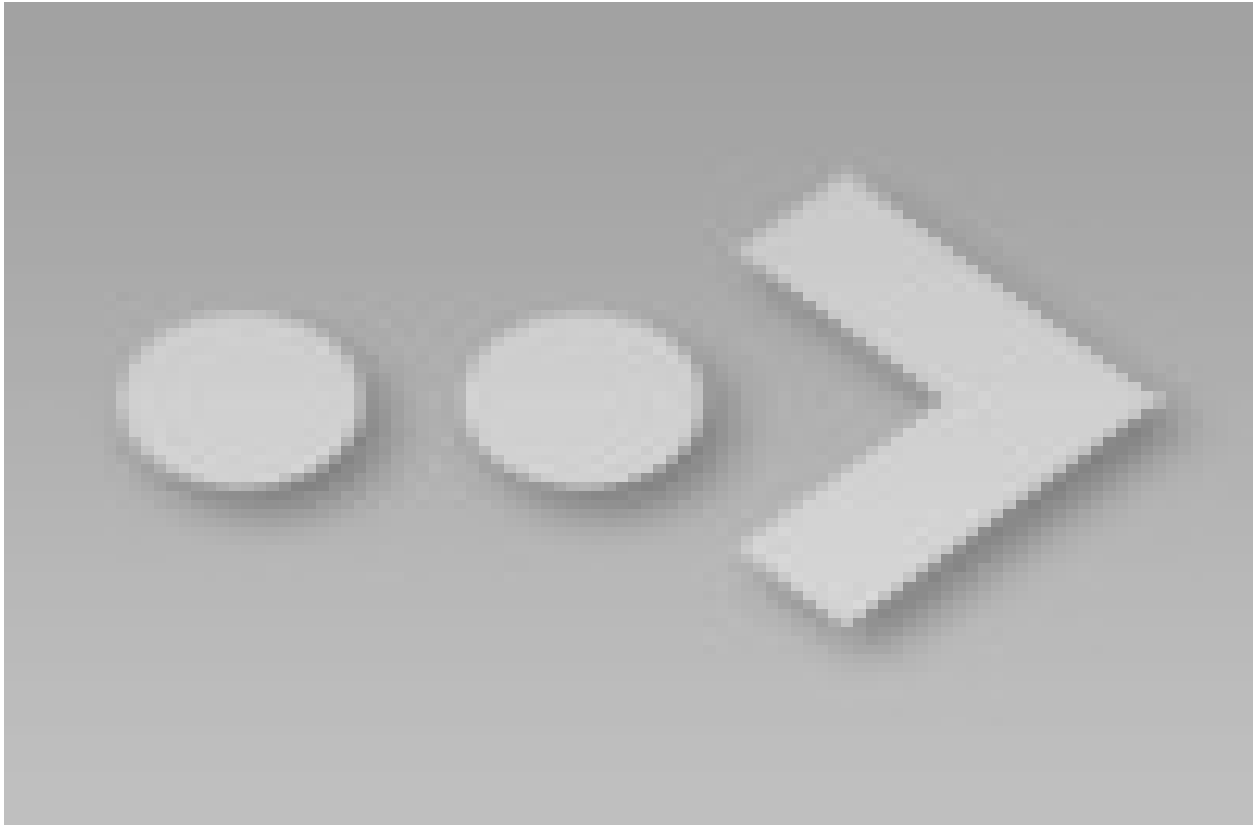
For this chapter, we will start with the most granular filters, dimension filters and measure filters, and then work our way out to the more 'macro' filters. To help illustrate dimension filters and measure filters, we will use this simple bar chart, which is showing the sales per customer from the Sample – Superstore dataset in descending order.

[Insert Image 11-1]



The first, and most basic way to filter out marks on a view, is to select the marks, and then choose to keep or exclude them. Let's say that our boss has an extreme aversion to names that start with the letter "C", so we want to make sure we remove those customers from the report. To do so, you can go through the list of customers, and use multi-select (Ctrl + Click on a PC) to highlight them. After selecting the names that you want to filter, hover over one of the dimension members (in this case, customer names), and click "Exclude".

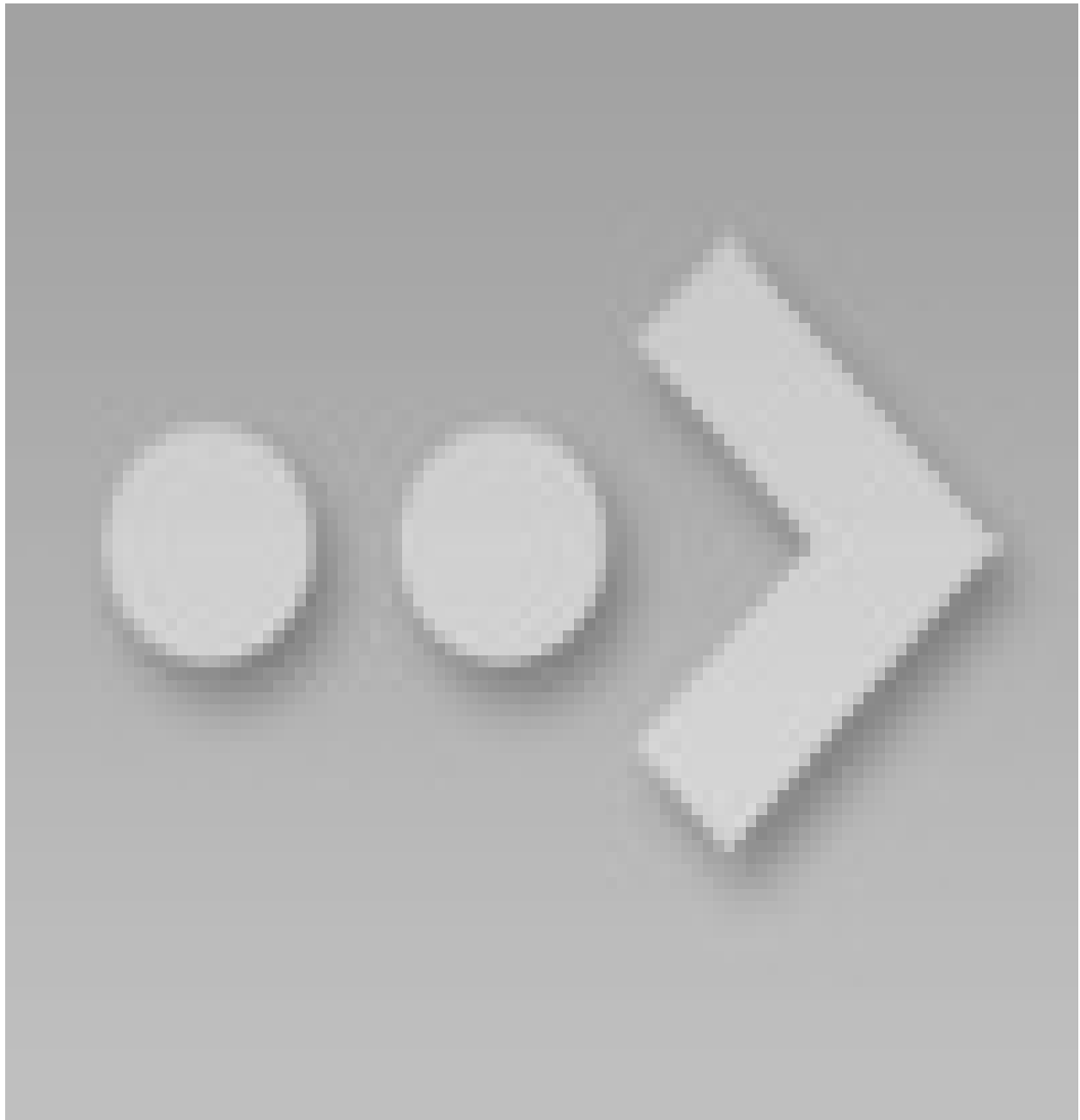
[Insert Image 11-2]



Note that there is also an option to “Keep Only”, which would do exactly what it sounds like, and keep only the selected names on the view.

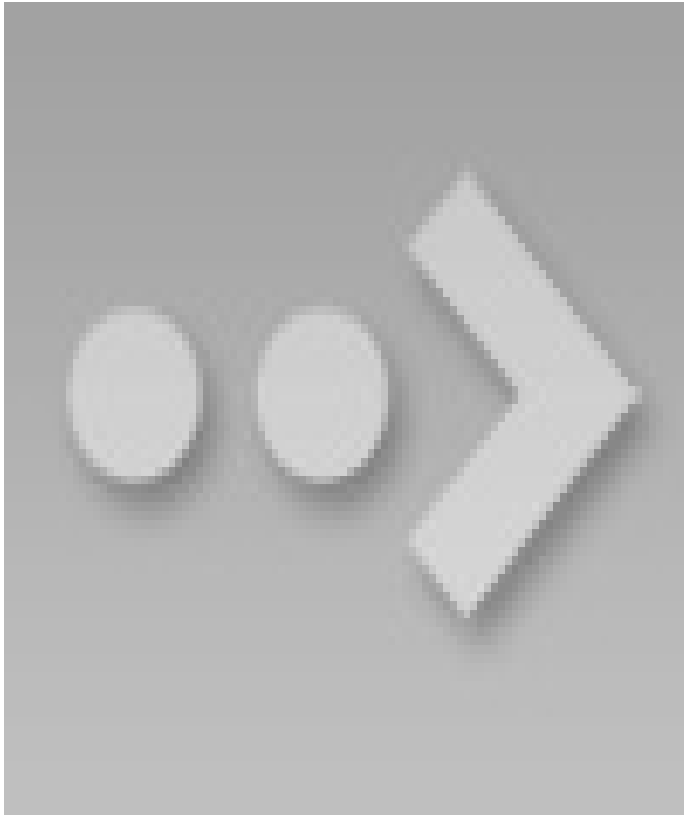
After choosing “Exclude”, notice that the Customer Name dimension is added to the Filters Shelf in the top left corner of the view.

[Insert Image 11-3]



That's because Tableau created a dimension filter for you when you excluded the customer name dimension members from the view. You could have got to this same point by dragging the Customer Name dimension from the Dimensions Shelf to the Filters Shelf and manually setting up the dimension filter. Let's take a look under the hood at the options for manually setting up a dimension filter by right-clicking on the Customer Name filter and choosing "Edit Filter..."

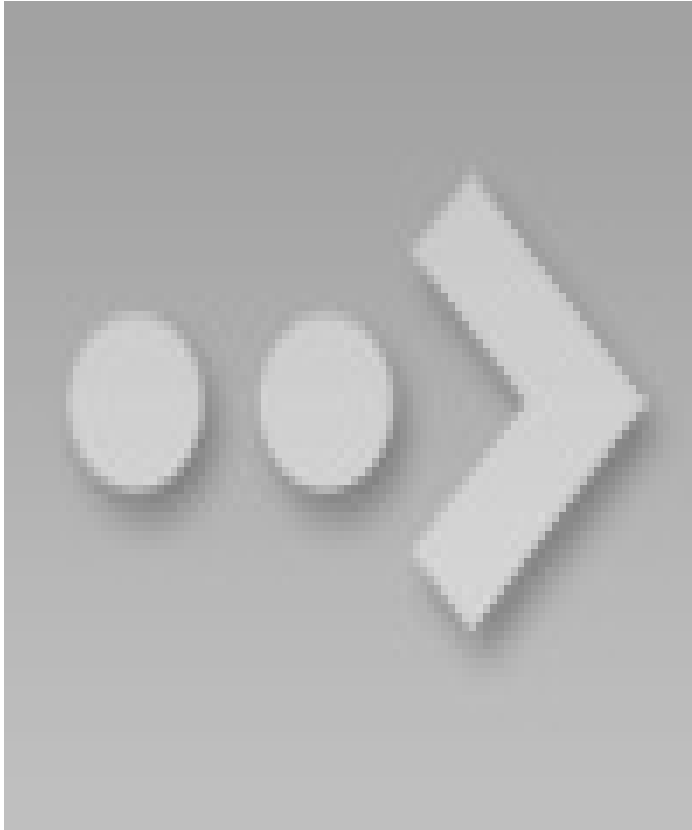
[Insert Image 11-4]



Notice that Tableau created an “Exclude” dimension filter for us when we chose to exclude the three names that start with the letter “C” from the view, as indicated by the box for “Exclude” being checked. There is a summary box at the bottom of the filter that is telling us the criteria for our filter. We currently are excluding 3 of 793 total marks, or customer names, from the view. This manual selection of individual dimension members is the most precise form of filtering, but can also be quite tedious. This is especially true when filtering on the Customer Name dimension, which has 793 individual dimension members.

Fortunately, there are three other tabs at the top of our dimension filter dialog box, which help us refine the filter criteria. Instead of manually selecting all names that start with the letter “C”, I’ll navigate to the ‘Wildcard’ tab and leverage the starts with function.

[Insert Image 11-5]



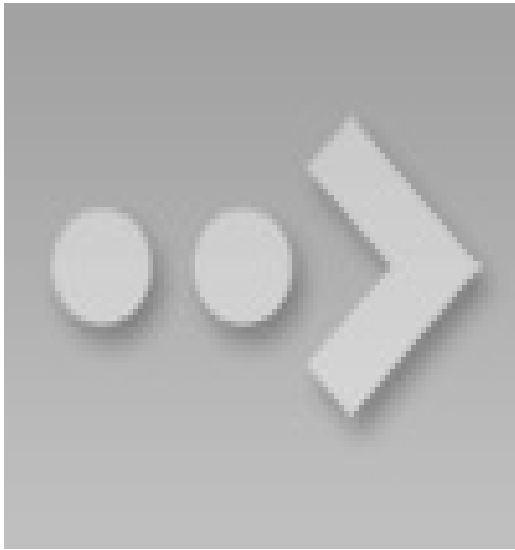
After applying this logic, I am left with 726 of 793 customers – and none whose first names start with “C” so our boss can rest easy! After adding any criteria to one of the four tabs, you can always navigate back to the Summary card on the General tab to see all the rules in the filter. Each rule acts as an AND statement, meaning that the dimension members must meet all criteria to be included or excluded from the view.

There are two additional tabs that can be used to add rules to a dimension filter. The Condition tab allows you to add quantitative thresholds that must be met and the Top tab allows you to focus only on the Top or Bottom dimension members based on a measure of your choosing.

Measure filters in Tableau

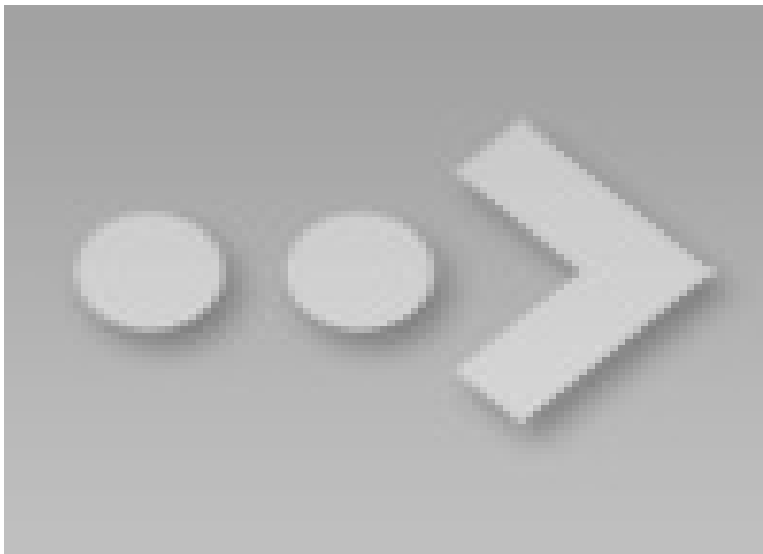
Measure filters are similar to dimension filters, but the filtering options are different between the two. To illustrate, drag the Sales measure from the Measures Shelf to the Filters Shelf. The first difference you’ll notice is that before you select the measure filter criteria, you are asked to choose the aggregation of the measure.

[Insert Image 11-6]

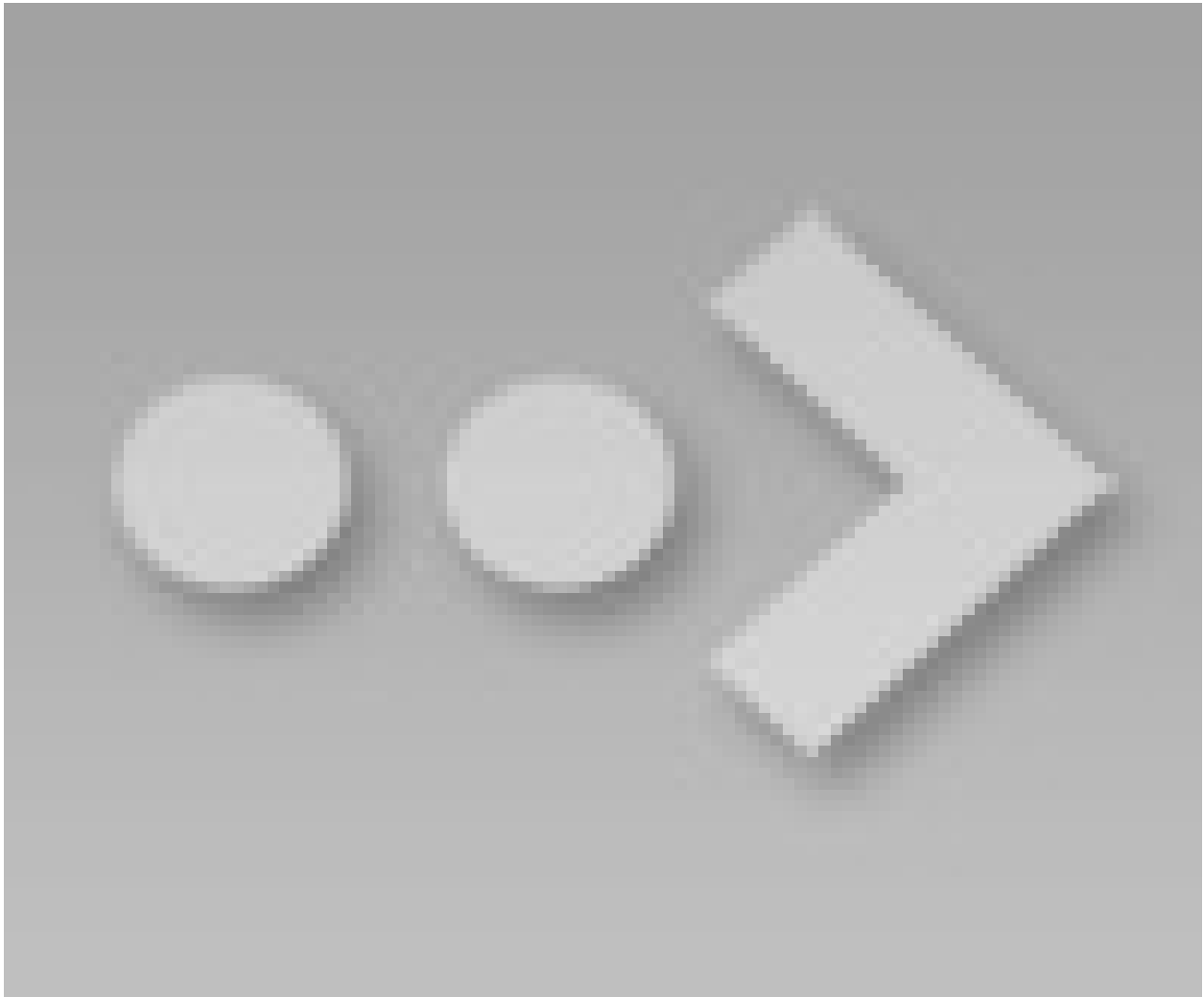


I will choose 'Sum' which is the default for most measures. After making the choice of aggregation, you can choose to filter on a range of values, a minimum threshold, or a maximum threshold.

[Insert Image 11-7]



Range of values provides the most flexibility and is the best (and only) choice if you want you and your end users to be able to control both the bottom end and top end of the range. For now, I'll change just the bottom end of the range to be \$10,000. After changing the low end to \$10,000 and clicking 'OK', my view looks like this: [Insert Image 11-8]



Note that I now have two filters on the Filters Shelf; one dimension filter and one measure filter. All of the filters that you place here will act as a condition in an AND statement. So at this point, in order for a customer name to be shown on the view, it must not start with the letter 'C' and sales must be between \$10,000 and \$25,043.05.

Note that you can add additional dimension or measure filters, even if the fields are not used to generate the view.

More options with filters

Sometimes, you want a filter to be permanent and do not have any interest in changing its criteria. Other times, you would like to change the criteria or even let your end users decide what is filtered on the view. Any filter can be shown to

let your end users decide what is filtered on the view. Any filter can be shown to you and your end users for easy manipulation by simply clicking on a filter from the Filters Shelf and choosing “Show Filter”. Upon doing so, you will see the filter appear in the top-right corner of the view. If you’re not happy with the default format of the filter, you can click the down arrow in the top right corner of the filter being shown on the view and change it. Dimension filters have seven different formats to choose from and measure filters have three different formats.

Lastly, dimension and measure filters are applied to only the worksheet you added it to by default. However, filters can be changed to apply to additional individual worksheets, every worksheet that shares the data source, or as of Tableau 10, even all related data sources. This provides the ability to make filters “global” so that changing their criteria in one place filters the views throughout the workbook. To change the worksheets that a filter is applied to, click the filter on the Filters Shelf, hover over “Apply to Worksheets”, and make your selection.

Macro filters

Dimension and measure filters are the most granular form of filtering in Tableau, and will be used most regularly. Occasionally though, you may want to apply a filter at a higher level, such as the data source or extract. This type of ‘macro’ filter provides the benefit of reducing the size of the dataset, one of my top five efficiency tips for working with data in Tableau.

To add this type of filter, navigate to the data source editor by either (1) right-clicking on the data source from the Data Window and choosing “Edit Data Source...”, (2) clicking on Data in the top navigation, hovering over the data source of interest and choosing “Edit Data Source...”, or (3) clicking on the Data Source tab at the bottom of the authoring interface. Once there, you will see an option to add filters to the live data source or data extract in the top-right corner. Once you choose a dimension or measure to use as a filter, the filter dialogs will look very familiar to the filters introduced to this point. Only now, the filters will be applied to the entire data source before you create individual views.

Chapter 12 – An Introduction to Calculated Fields

Possibly the most powerful feature of Tableau is its ability to allow authors to create new data from existing data through calculated fields. Calculated fields can be used to create new dimensions such as segments, or new measures such as ratios. They can also be used with any data type, a multitude of functions and aggregations, as well as logical operators, making the calculated results virtually limitless. This chapter will introduce calculated fields by walking you through how to create two new measures and one new dimension in the Sample – Superstore dataset.

Why use calculated fields?

You may be wondering what's with all the praise regarding calculated fields? After all, I've also suggested that it's best to prepare your data as much as possible before it gets to Tableau, and that data should include all of the required fields for your analysis. Right?

There are many reasons to leverage the calculated fields functionality in Tableau. Here are just a few:

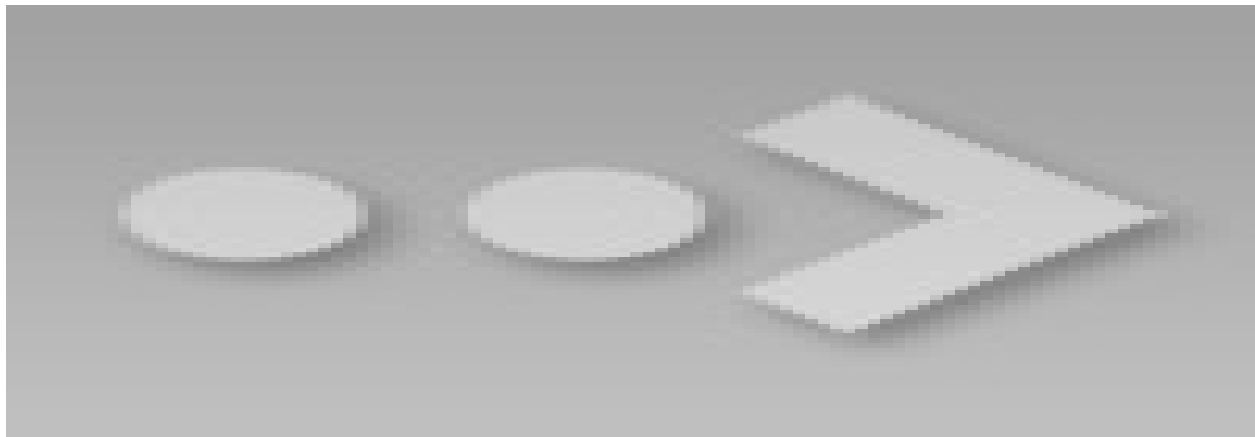
- To segment your data in new ways on the fly
- To prove a concept such as a new dimension or measure before making it a permanent field in the underlying data
- To filter out unwanted results for better analyses
- To take advantage of the power of parameters, putting choice in the hands of your end users
- To calculate ratios across many different variables in Tableau, saving valuable database processing and storage resources

To bring the concept of calculated fields to life, let's pretend we are working with the Sample – Superstore dataset, and want to evaluate the average order value (AOV) for the product sub-categories we manage compared to the rest of the company. AOV is defined as total sales revenue divided by the number of orders. While this dataset has Order ID as a dimension, it does not have the number of orders as a measure, which is the denominator for the AOV calculation.

To isolate the number of orders, I will first create a calculated field to count the number of orders. The formula to count the distinct number of orders will be COUNTD of the Order ID dimension. To start a calculated field, you can either:

- Click the down arrow in the top-right corner of the Dimensions Shelf and pick “Create calculated field...”
- Right-click a blank space on the left sidebar and choose “Create calculated field...”
- Click Analysis in the top navigation and choose “Create Calculated Field...”
- Right-click on one of the fields you want to use as part of your calculated field, hover over “Create”, and choose “Calculated field...”

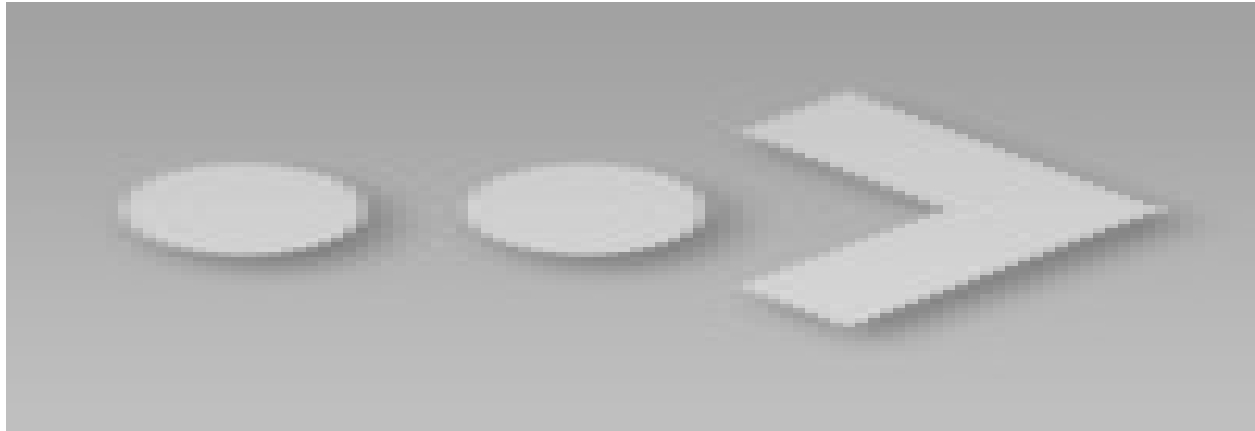
For this first calculated field, because I know that I need the Order ID as part of my calculated field, I will start the calculated field by right-clicking on the Order ID dimension on the Dimensions Shelf, hovering over “Create”, and clicking “Calculated field...”. This opens a new dialog box where I can enter the formula for my calculated field: [Insert Image 12-1]



On the right-hand side of the calculated field dialog box there is a definition of the COUNTD expression. Any time you see a blue color-coding when creating a calculated field, you can click on the blue-colored word to get a definition and example of that particular function; this is a great way to learn the syntax. After clicking the OK button, you will see a new measure for Orders appear on the Measures Shelf – a brand new measure that we can now use throughout the workbook!

Now that I have my Orders measure, which is the denominator of the AOV calculation I’m after, I can create another calculated field to calculate the total

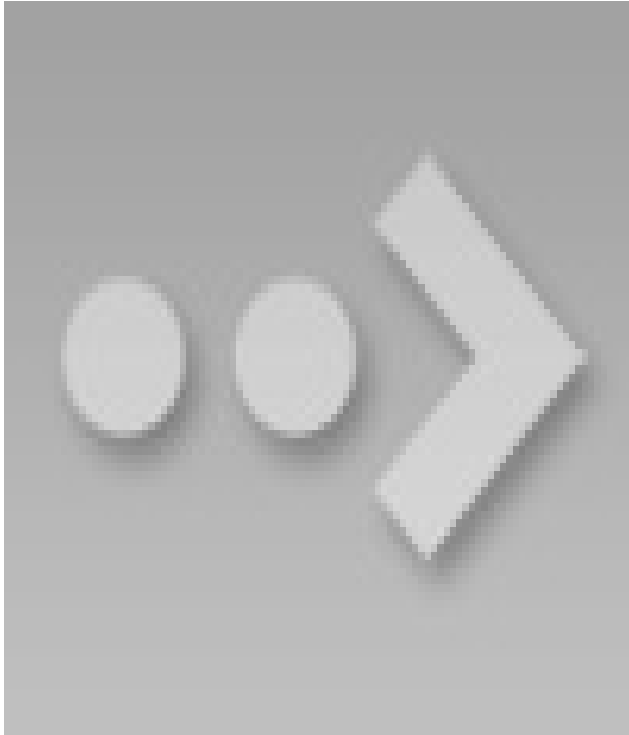
sales revenue divided by the number of orders. To start this calculated field, I will right-click on a blank space in the left sidebar and choose “Create calculated field...”. I’m presented with a blank calculated field dialog box where I can enter my formula for AOV: [Insert Image 12-2]



Notice this time that the measure of Sales is aggregated as SUM, but Orders appears to have no aggregation. That’s because Orders already has an aggregation of COUNTD in the underlying calculated field that we created in the previous step. Tableau does not allow you to mix aggregated and non-aggregated fields within a calculated field. If you create an invalid calculated field, Tableau will display a red indicator at the bottom of the dialog box that reads “The calculation contains errors”. You can click on the error message to be provided with a clue for what may be wrong with your calculation.

Also remember that even when Tableau says that a calculation is valid, that’s in terms of the syntax, and not in terms of the result you are looking for. To quality check a calculated field and ensure it’s properly calculating the result, I like to put the raw ‘ingredients’ on the view and calculate the answer manually. In the case of this tutorial, I would create a quick table showing orders, sales, and the newly created AOV measure. I would then divide the sales amount by the number of orders to make sure the correct answer is reflected as AOV.

[Insert Image 12-3]



Based on the image above, it looks like Tableau has calculated the correct answer: \$2,297,201 in total sales divided by 5,009 total orders equals an average order value of \$459. By default, calculated fields do not receive number formatting. In the case of AOV, which should be displayed as a currency, this can be changed by right-clicking on the calculated field on the Measures Shelf, hovering over “Default properties”, and clicking “Number format...”.

More on aggregating calculated fields

Aggregation, introduced in the chapter, 5 Ways to Make a Bar Chart / An Introduction to Aggregation, is an important concept to consider when creating calculated fields. A calculated field for `SUM([Profit]) / SUM([Sales])` will give you a very different answer than `[Profit] / [Sales]`, even though both formulas are valid. If you do not provide the aggregation within the calculated field, Tableau will calculate the equation for every record in your analysis, then aggregate the answers for all of the rows together when the calculated field is added to the view. It is critical to quality check calculated fields and ensure you are seeing expected results before integrating the new measures in your work.

To this point, we have created a measure for Orders and used the Orders

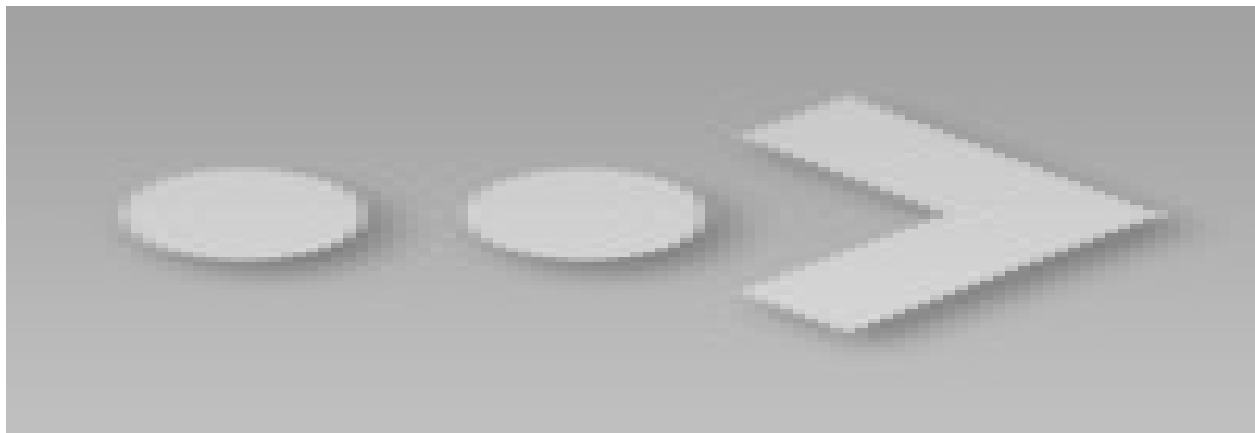
to this point, we've created a new measure for Orders and used the Orders measure within a second new calculated field for AOV. Now let's put the new fields to work by answering our business question:

What is the AOV of the product sub-categories I manage compared to everything else?

For the purposes of this illustration, I'll pretend that I manage the Copiers, Machines, and Supplies sub-categories. To answer this question, we will create a third calculated field, this time creating a new Sub-Category Segmentation dimension.

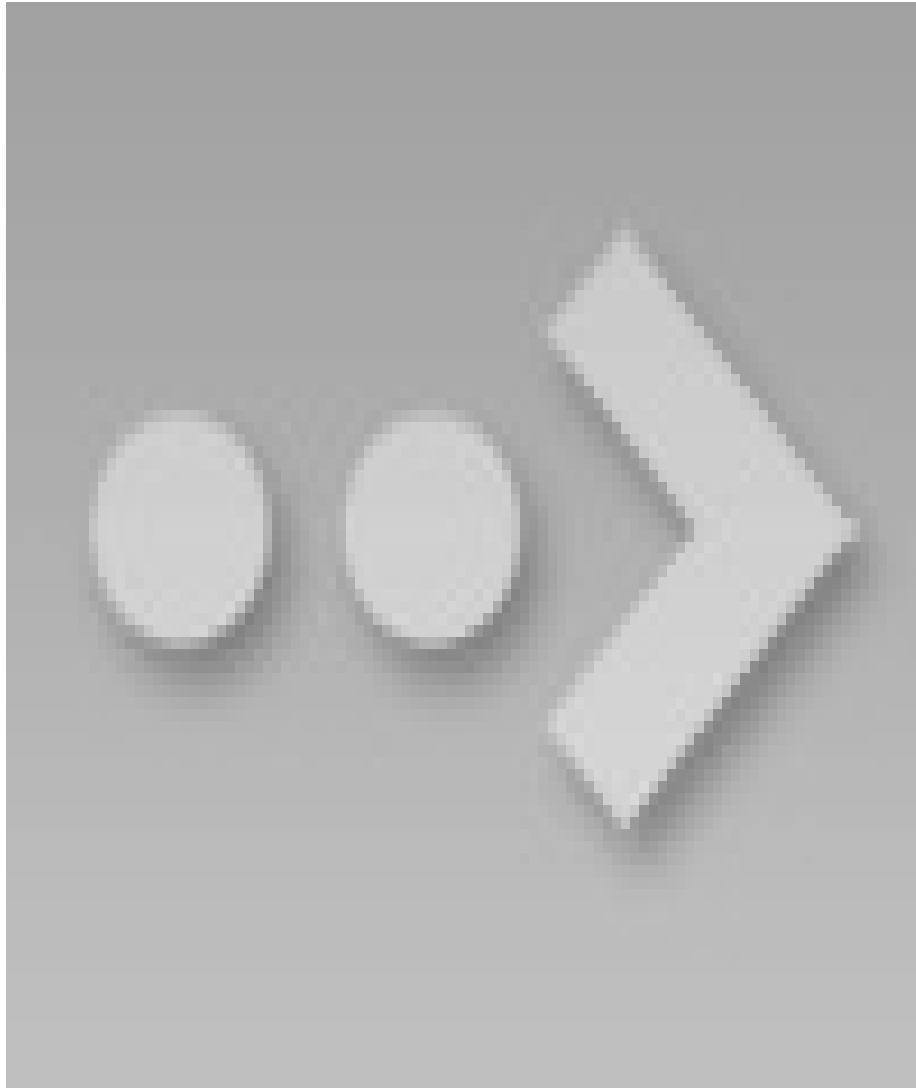
This calculation is slightly different in that we will be incorporating IF / THEN logic to create a segmentation: one segment for the sub-categories I manage (Copiers, Machines, and Supplies) and the other segment for every other sub-category. The formula looks like this:

[Insert Image 12-4]



All this calculation is saying is that if the dimension member of Sub-Category matches copiers, machines, or supplies, I want the data to be classified as “My Sub-Categories”. If the dimension member is anything other than those three, I want the data to be classified as “Other”. To quality check a dimension calculation such as this one, I like to first place the original dimension on the rows shelf, followed by the newly created calculated field to make sure the dimension members are being properly calculated:

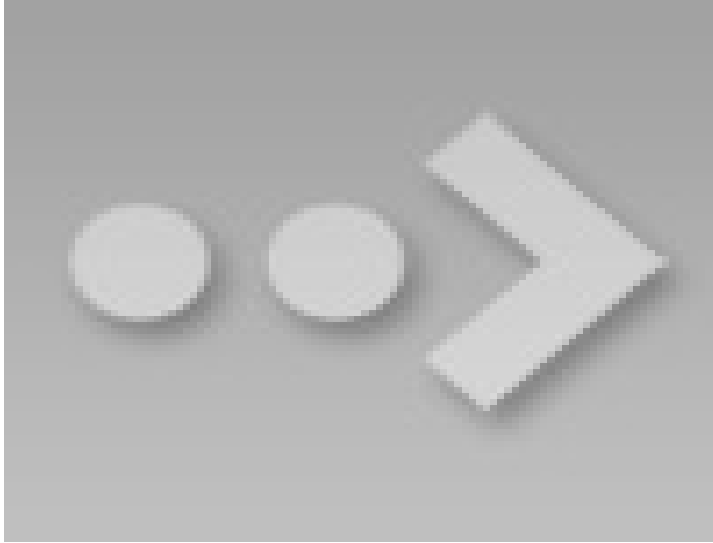
[Insert Image 12-5]



As you can see, my three sub-categories are classified as “My Sub-Categories” in the second column, and everything else is classified as “Other”.

I’m now ready to answer my business question by putting the AOV calculated field on the view, then slice and dice the measure by the Sub-Category Segment dimension:

[Insert Image 12-6]



The answer clearly shows the AOV for the product sub-categories that I manage is much higher than the AOV for all of the other sub-categories – maybe it's time I ask for a bonus!

Chapter 13 - An Introduction to Table Calculations

Tableau comes with several preset calculations that you can compute with the numbers on a view including running total, difference, percent difference, percent of total, moving average, and more. These predefined calculations are called table calculations because they compute the result based on a virtual table that includes only the numbers on the view. Table calculations provide several benefits including:

- A fast way to create advanced calculations even without knowing the underlying syntax
- Table calculations can be saved for future use as calculated fields, and because calculated fields can be edited, this is a great way to learn the syntax and the different functions available in Tableau
- Efficient processing; table calculations are computed on a very small subset of the data source, making them an efficient solution for calculating results

To help introduce table calculations and how to use them, we will be using this simple crosstab created with the Sample – Superstore dataset:

[Insert Image 13-1]

Figure 13-2

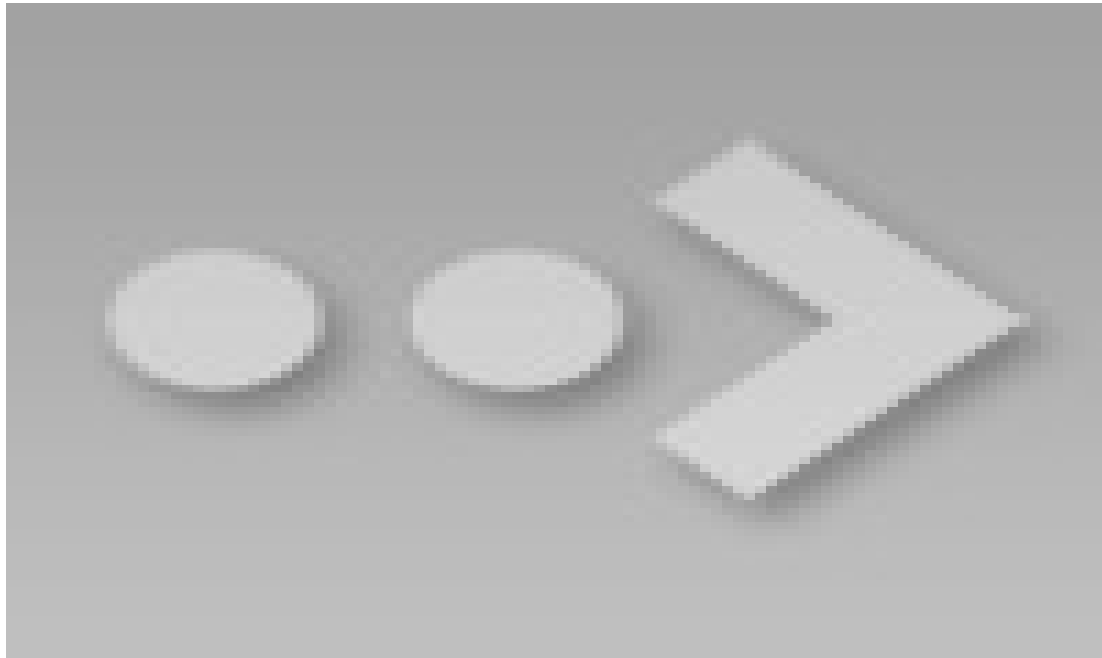
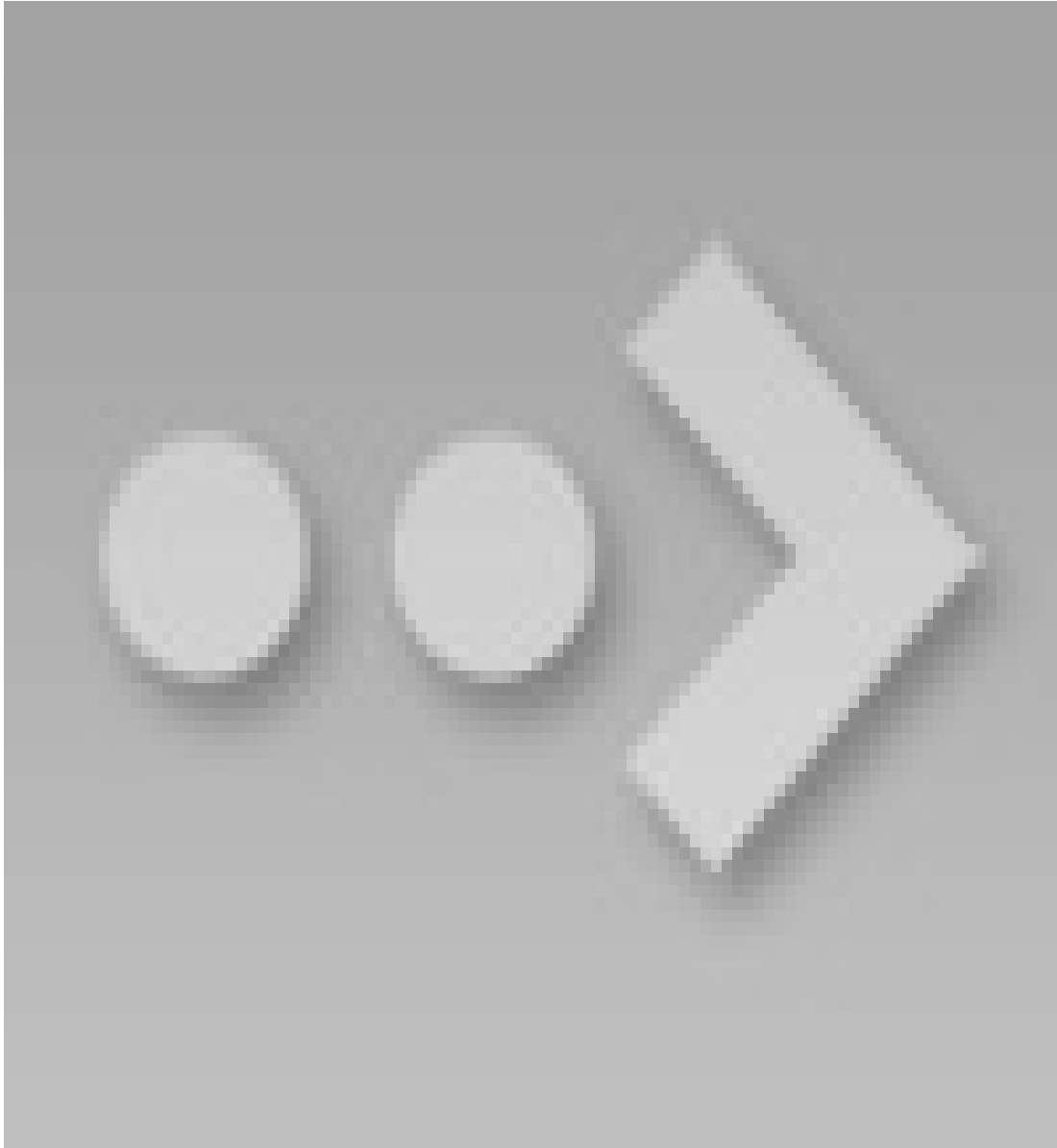


Table calculations are added to measures, so in order to add a table calculation, click a measure that's on the view. The fastest way to add a table calculation is to hover over “Quick table calculation” and choose an option: [Insert Image 13-2]

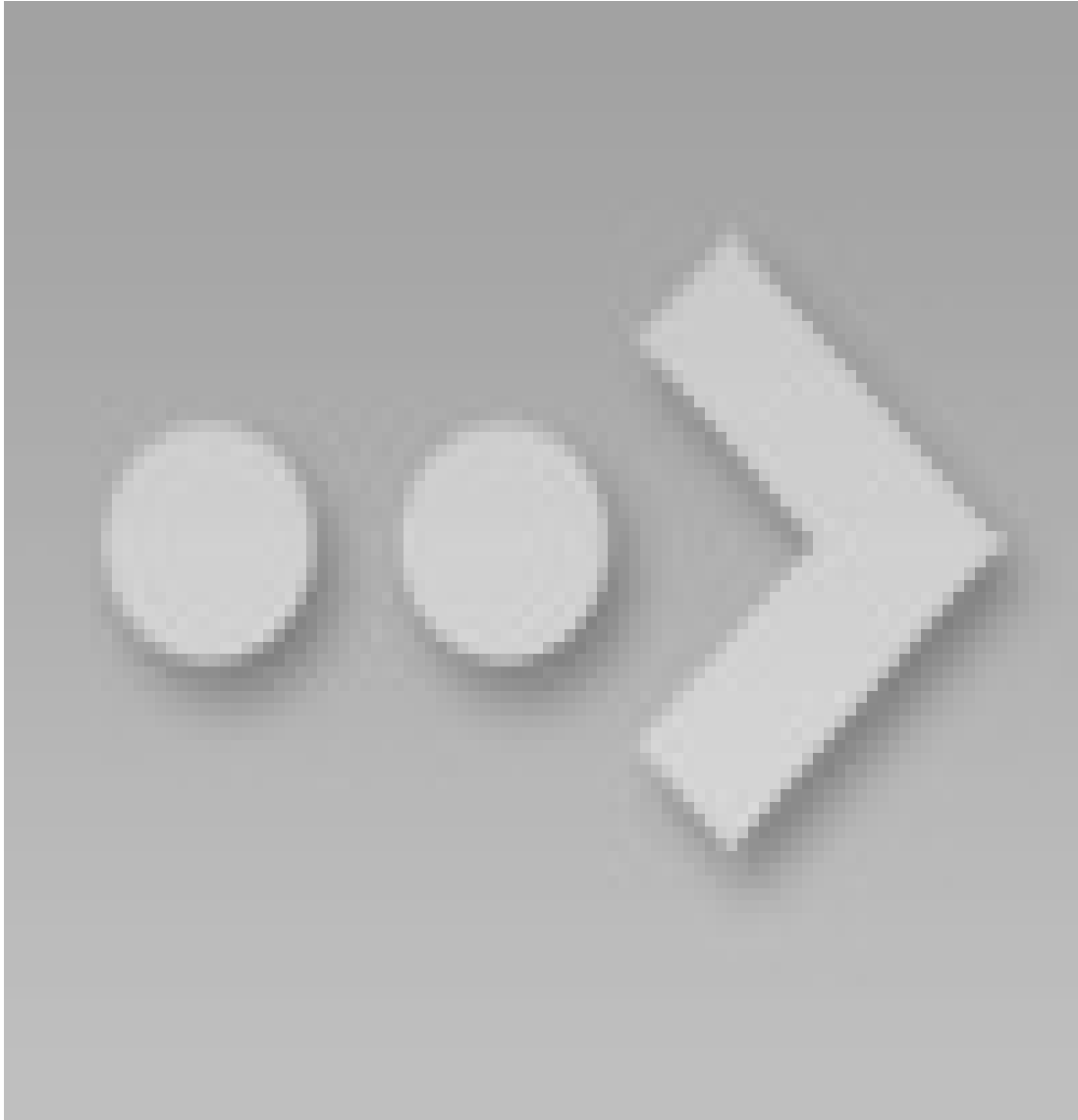


Here's how the view looks after choosing the "Running total" table calculation:

[Insert Image 13-3]

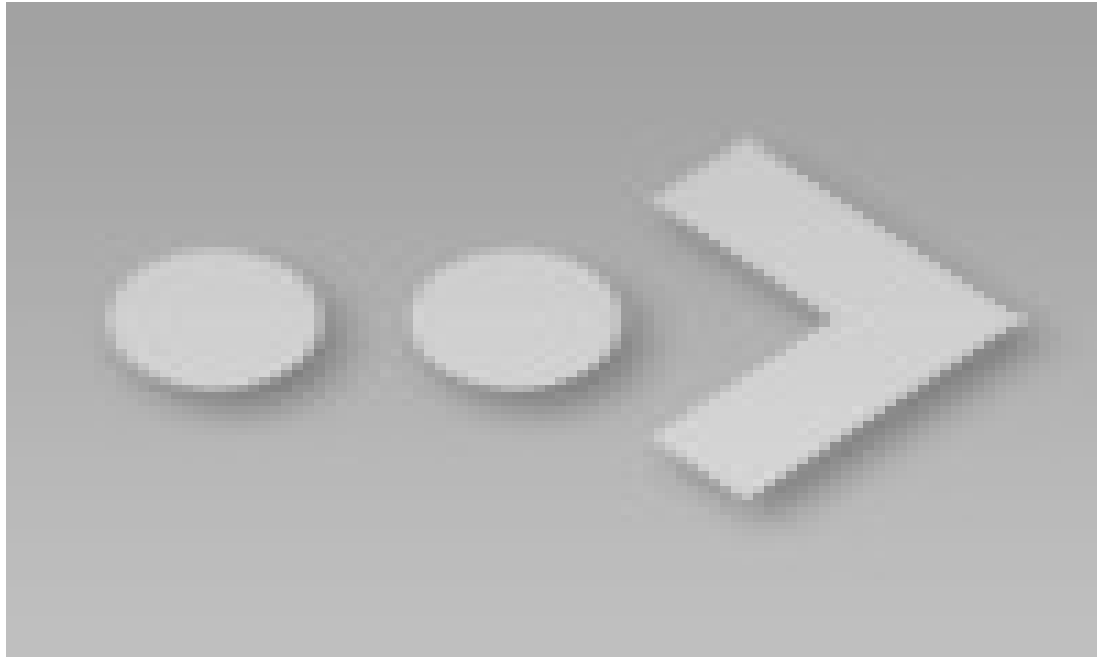


Table calculations are defined by how they are (1.) partitioned – or grouped, and (2.) addressed – or how they are computed. Notice in the example above the running total is being computed from left to right, which is the default addressing. This would mean that, by default, the table calculation is being addressed by the Product Category dimension. This leaves the Month dimension as the partitioning field. For a running total calculation, this doesn't make a lot of sense. It is easy to change the addressing by changing how the table calculation is being computed. To do this, click on the measure with the table calculation again, now identified with a delta symbol, hover over compute using, and change how the calculation should be computed (or addressed): [Insert Image 13-4]



Here's how the crosstab looks after changing the addressing / compute using to Table (Down):

[Insert Image 13-5]

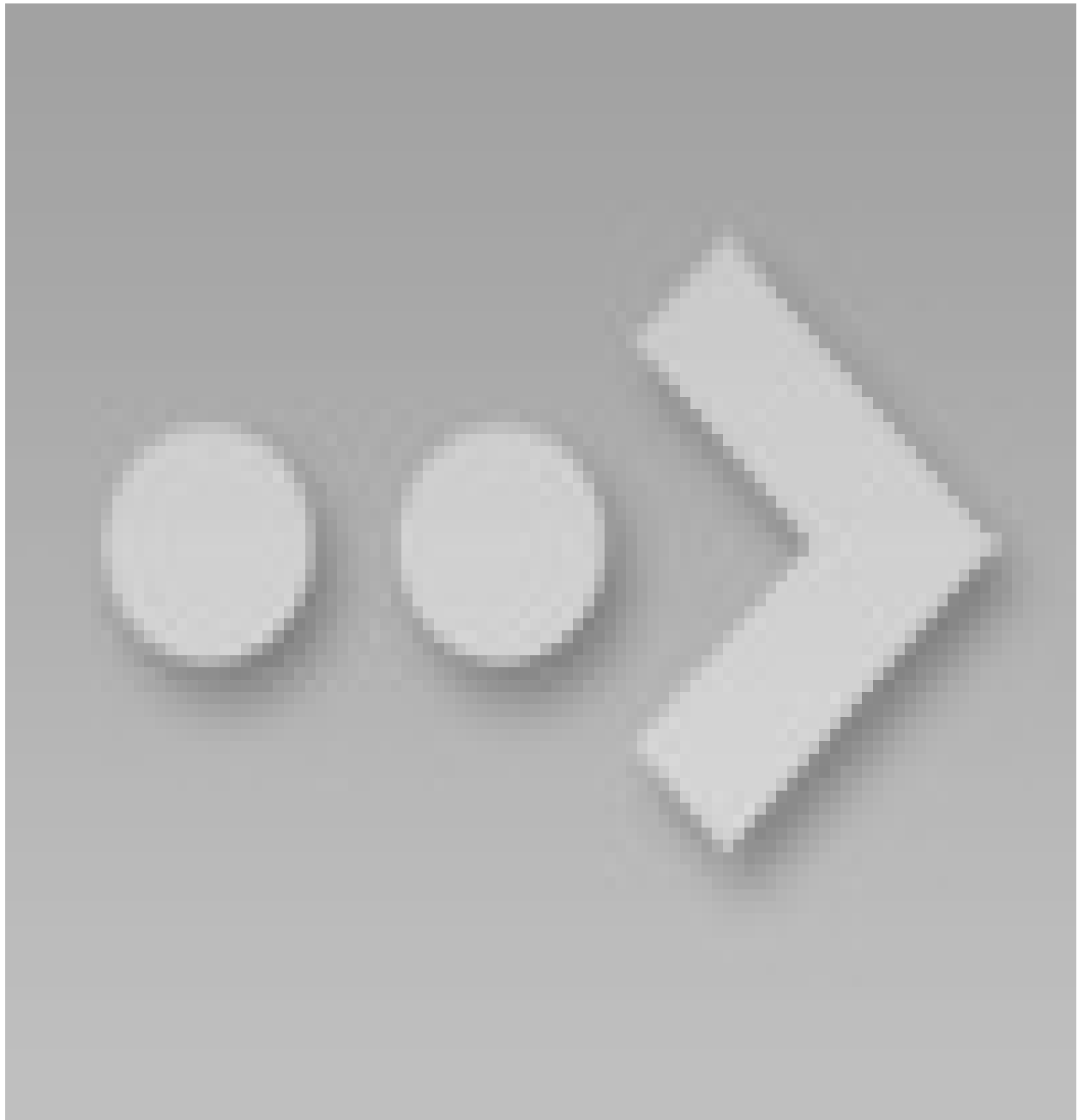


Now that the addressing field has been changed to Month and the partitioning field has been changed to Product Category, my result makes a lot more sense for my analysis. I can look at each Product Category column, and look down across months to see how the sales built up throughout the year.

There are many different options available within table calculations that can be accessed by clicking a measure with a table calculation and choosing “Edit table calculation...”. For just a few possibilities, see one of Tableau’s most popular posts, [Top 10 Tableau Table Calculations](#).

I mentioned in the introduction to this chapter that one of the benefits of table calculations is that they provide a way to learn the syntax. To do so, simply double click on a measure with a table calculation to see the underlying formula. This code can be copied into a calculated field so you can take a closer look.

Lastly, even though they are called table calculations, table calculations do not literally have to be used on a table, or crosstab, view. Here is one example I regularly use to add value to a basic monthly line graph: [Insert Image 13-6]



Notice that the first row is a basic monthly trend, while the second Sales pill has a delta symbol, indicating there is a table calculation being applied. This time, I am using the second row to show the month over month *difference* in sales – another of the quick table calculation options available in Tableau.

To create this visualization, I started by simply making a line graph showing the sum of sales by continuous month of order date. I then placed a second occurrence of the Sales measure on the Rows Shelf, which created a second row with the exact same trend. Now that there were two independent pills for the

Sales measure, I was able to add a table calculation for month over month difference (the second option under “Quick table calculation...”) to the second pill only.

Each measure also gets its own set of Marks Cards, which means they can be edited independently. After changing the second row so the mark type is bar and the marks are colored by the month over month change, the result is the visualization above. Now in addition to a typical monthly sales trend, I’ve provided the month over month difference values and visualization which allows the end user to quickly compare the spikes and drops across months; a value-add made possible through Tableau table calculations.

Chapter 14 – An Introduction to Parameters

As you make your way through *Practical Tableau*, and especially the Tips & Tricks section, there is one functionality that you will hear about over and over: parameters. Parameters are the solution to several of the handiest tricks in Tableau because of their flexibility. Other than filters, parameters are also one of the most powerful tactics in Tableau for transferring control from the author to the end user. As discussed in the upcoming data-driven storytelling section, providing this type of user interaction not only improves the engagement with your dashboards, but it improves the retention of insights and improves the likelihood that they will be shared.

This chapter will use a simple algebra example to illustrate how to create a parameter, how to allow end users to control the parameter you’ve created, and how parameters work.

An introduction to parameters in Tableau

For this tutorial, imagine that you want to set up a simple algebraic equation for 2 multiplied by X, where X is the parameter, and the end user gets to choose the value of X. Tableau will then display the answer of 2 multiplied by X based on

the parameter selection.

For this introduction, it does not matter what dataset you are following along with, but for consistency with the rest of *Practical Tableau*, I will be building a parameter in the Sample – Superstore dataset. There are several ways to create a new parameter in Tableau including (1) Clicking the down arrow in the top-right corner of the Dimensions Shelf and choosing “Create parameter...” or (2) Right-clicking in a blank space on the Dimensions, Measures, Sets, or Parameters Shelf and choosing “Create parameter...”. Once you have done this, you will be presented with a dialog box where you can design the parameter: [Insert Image 14-1]



Instead of “Parameter 3”, named that by default in the Sample – Superstore dataset because there are already two other parameters, I will change the name to “Algebra Parameter”. By default, the data type for a new parameter is float, which means any number including decimals, but there are five additional data types to choose from: *Integer*: Any whole number (i.e. no decimals)

String: Text

Boolean: True or false

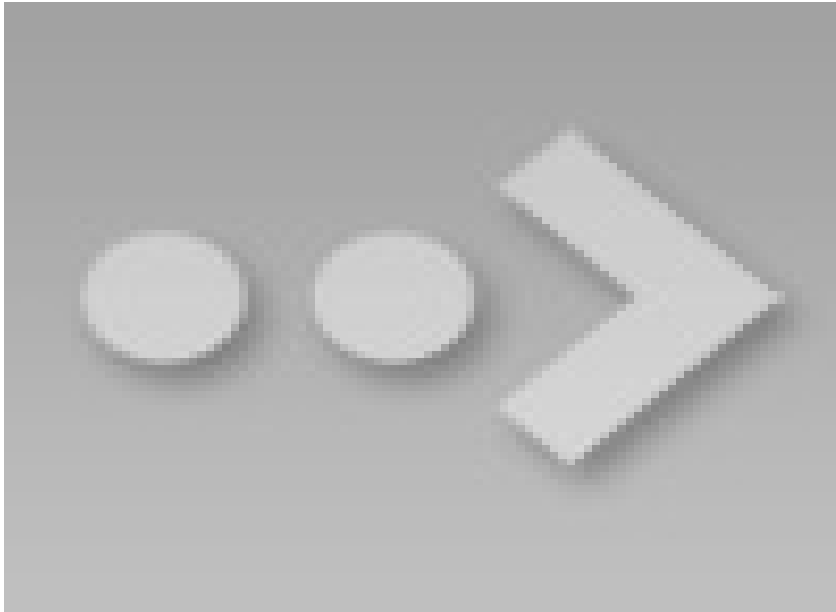
Date: Date without a timestamp

Date & Time: Date with a timestamp

The data type you choose for the parameter depends on your use case. For this 2 multiplied by X example, let’s say that we eventually want the parameter to include only whole numbers, so I will change the data type from Float to Integer.

Within the parameter creation dialog box, you also have the ability to designate the allowable values for the parameter. The entries that you choose at this step will eventually determine what choices you and the end users have for the

parameter in the final product. The choices are all or any integer, a specific list of integers, or a range of integers. I will choose range so that I can set a minimum and maximum number as well as a step size, or multiple, for the parameter. After setting up the parameter to have a range of 1 to 20 with a step size of 1, my parameter looks like this and is ready to be saved by clicking the “OK” button: [Insert Image 14-2]

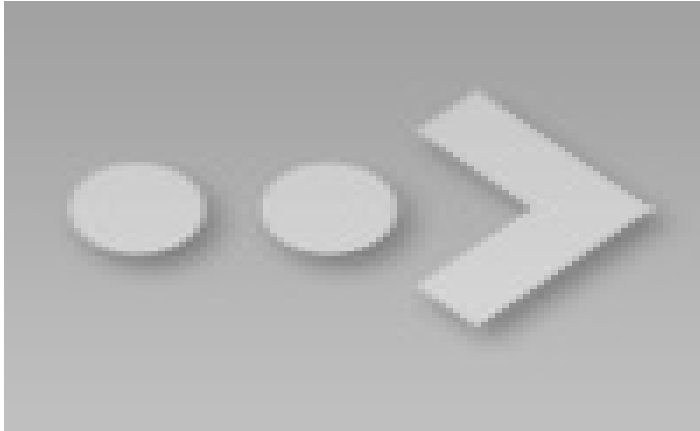


In order to use a parameter, you must provide Tableau instructions on how to use the parameter. This is accomplished through [calculated fields](#). Being that the equation for our use case is 2 multiplied by X, or our algebra parameter, your first instinct may be to create a calculated field that says: $2 * [\text{Algebra Parameter}]$

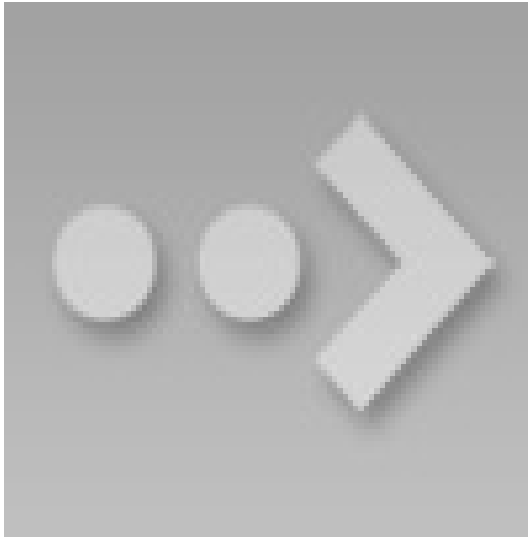
Being that the current value of the parameter is 1, we would expect the answer to be $2 * 1 = 2$. However, this calculation will be computed for every row in the data, which would not provide the answer we are looking for. Should we add this calculated field to the view, we would get an answer of 19,988, which is 2 multiplied by 1, multiplied by the number of records in the dataset (9,994 for the Sample – Superstore dataset).

To alleviate this, add an aggregation of MIN to the 2 so the Algebra Parameter is guaranteed to be multiplied by 2:

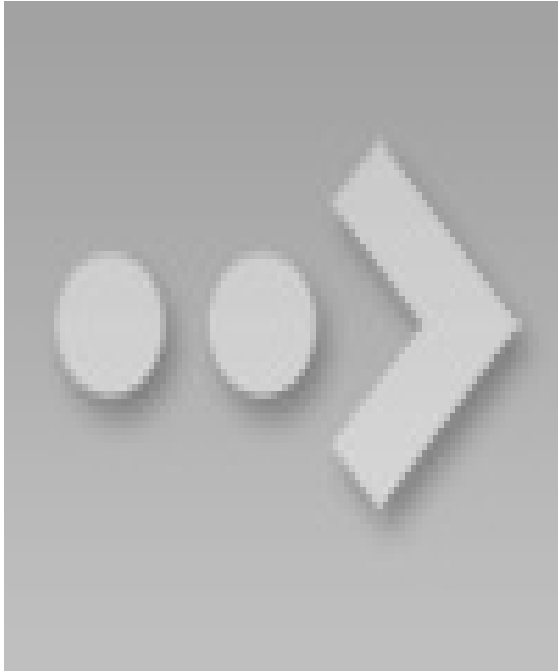
[Insert Image 14-3]



Now if I add my newly created Algebra Equation calculated field to the Text Marks Card to display the answer on a view, I see the answer I am looking for, MIN(2) multiplied by 1 equals 2: [Insert Image 14-4]



Finally, to provide the power of selecting the value of X in our algebra example to the end user, you must provide access to the parameter. To do this, right-click on the newly created Algebra Parameter and choose “Show parameter control”. A new selector will appear in the top-right corner of the view that allows the user to choose from the allowable values that were set up when the parameter was created. Notice that the current value is 1, the maximum value is 20, and the numbers can be selected in multiples of 1, just like we set up. Here’s what my final view looks like when I’ve shown the parameter control, moved the parameter control from its default location to below the Marks Cards, and changed the selection from 1 to 7: [Insert Image 14-5]



In this example, the end user has chosen the number 7, which powered the underlying calculation of $2*7$, to produce the answer of 14. Imagine all of the power that can be put into the hands of your end users through the use of parameters!

For just a few other possibilities, see the following chapters:

- How to Compare Two Date Ranges on One Axis
- How to Change Date Aggregation Using Parameters
- How to Make a What-If Analysis Using Parameters
- How to Compare and Create Segments
- Allow Users to Choose Measures and Dimensions

Chapter 15 – An Introduction to Sets

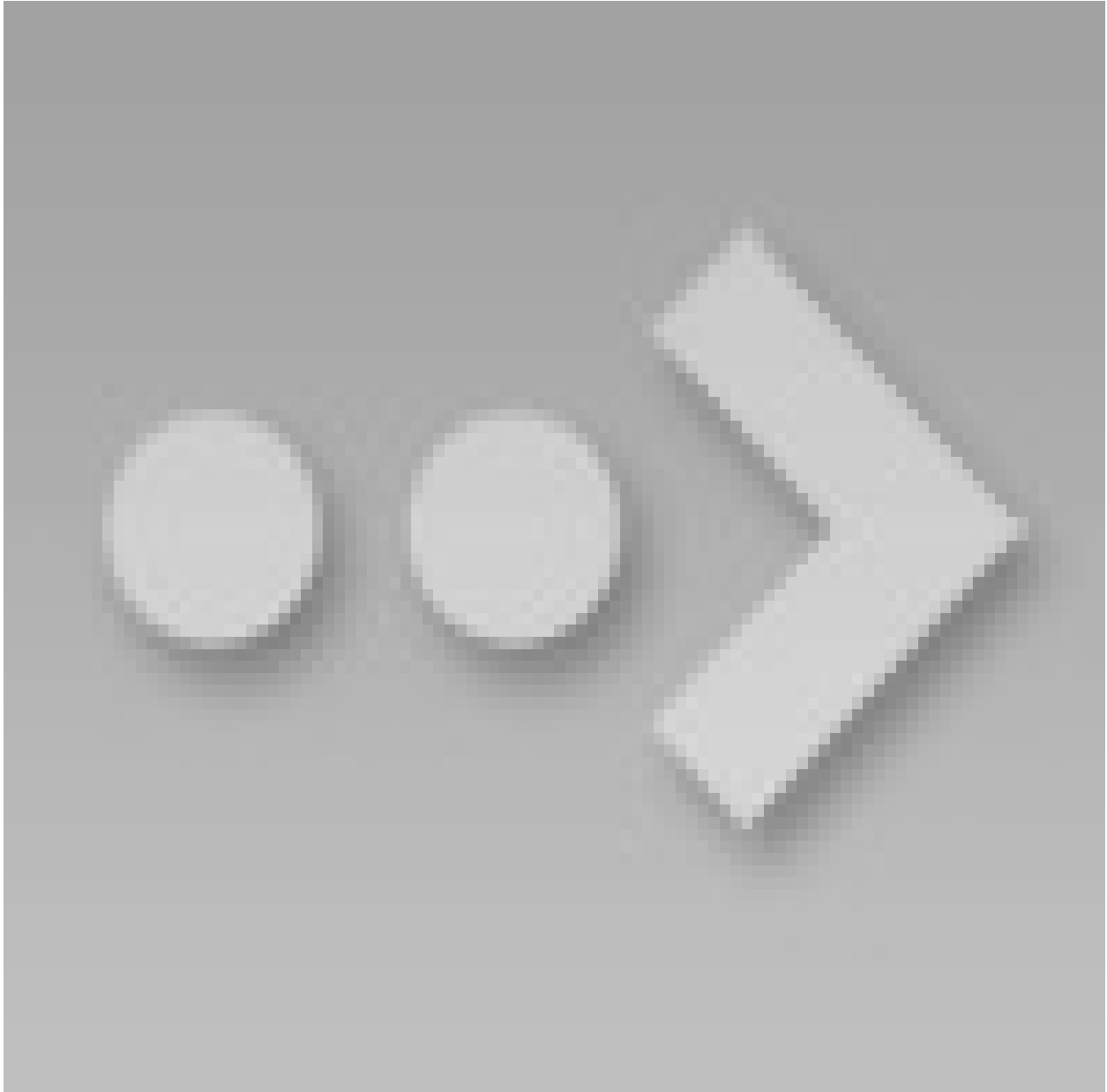
Tableau sets allow you to isolate specific segments of a dimension, which can then be used in several different ways to find insights in your data. This chapter provides instructions on how to build sets as well as five different ways they can be used to enhance your analyses.

Sets can be thought of as custom segments, but unlike dimension fields, they are always binary. In other words, you are either in the set or not. Other than that one restriction, sets can be created for just about anything. You can pick individual dimension members to place in a set, have sets be based on quantitative thresholds, created with the top or bottom performing dimension members, and more.

How to create a set in Tableau

Similar to creating filters in Tableau, sets can be created in a couple of different ways. The first, and most straight-forward method is to simply select the dimension members on a view, then hover over one of the dimension members, click on the Venn diagram icon that appears, and choose “Create Set...”. Here’s what that would look like if I wanted to create a set from the top 20 customers by sales in the Sample – Superstore dataset.

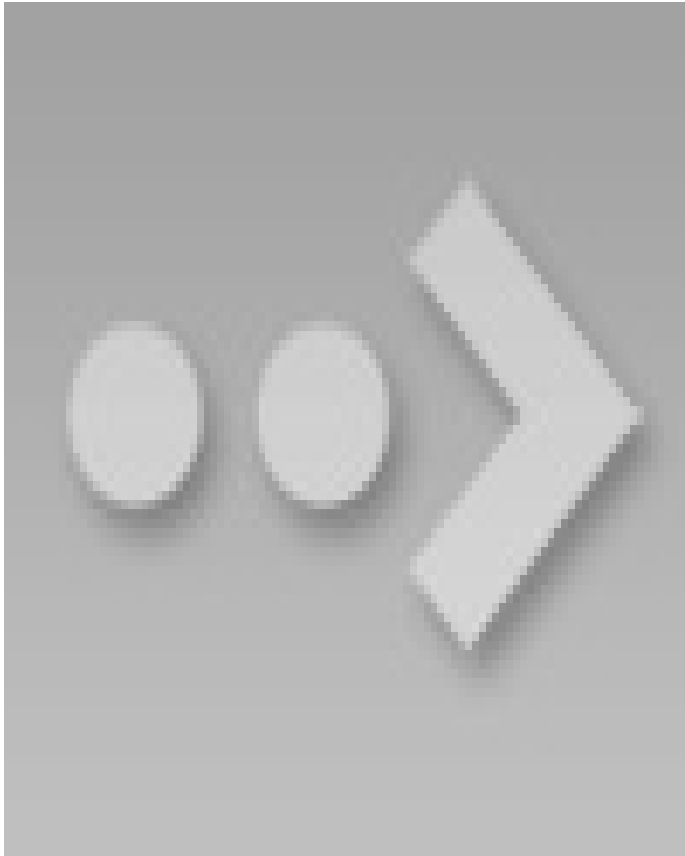
[Insert Image 15-1]



After clicking on “Create Set...” and giving the set a name, the set will appear on the Sets Shelf on the left side of the authoring interface. This set that tells us whether each customer name is in or out of the top 20 customers by sales is now available to use in our analyses. Note that sets created with this method are static, so the top 20 will not dynamically change should a new customer enter the top 20.

You can also create a set by right-clicking on the dimension the set will be created from on the Dimensions Shelf, hovering over “Create”, and clicking “Set...”. After following the preceding steps on the Customer Name dimension, the following dialog box appears (Figure 15-21).

the following dialog box appears: [insert image 15-2]



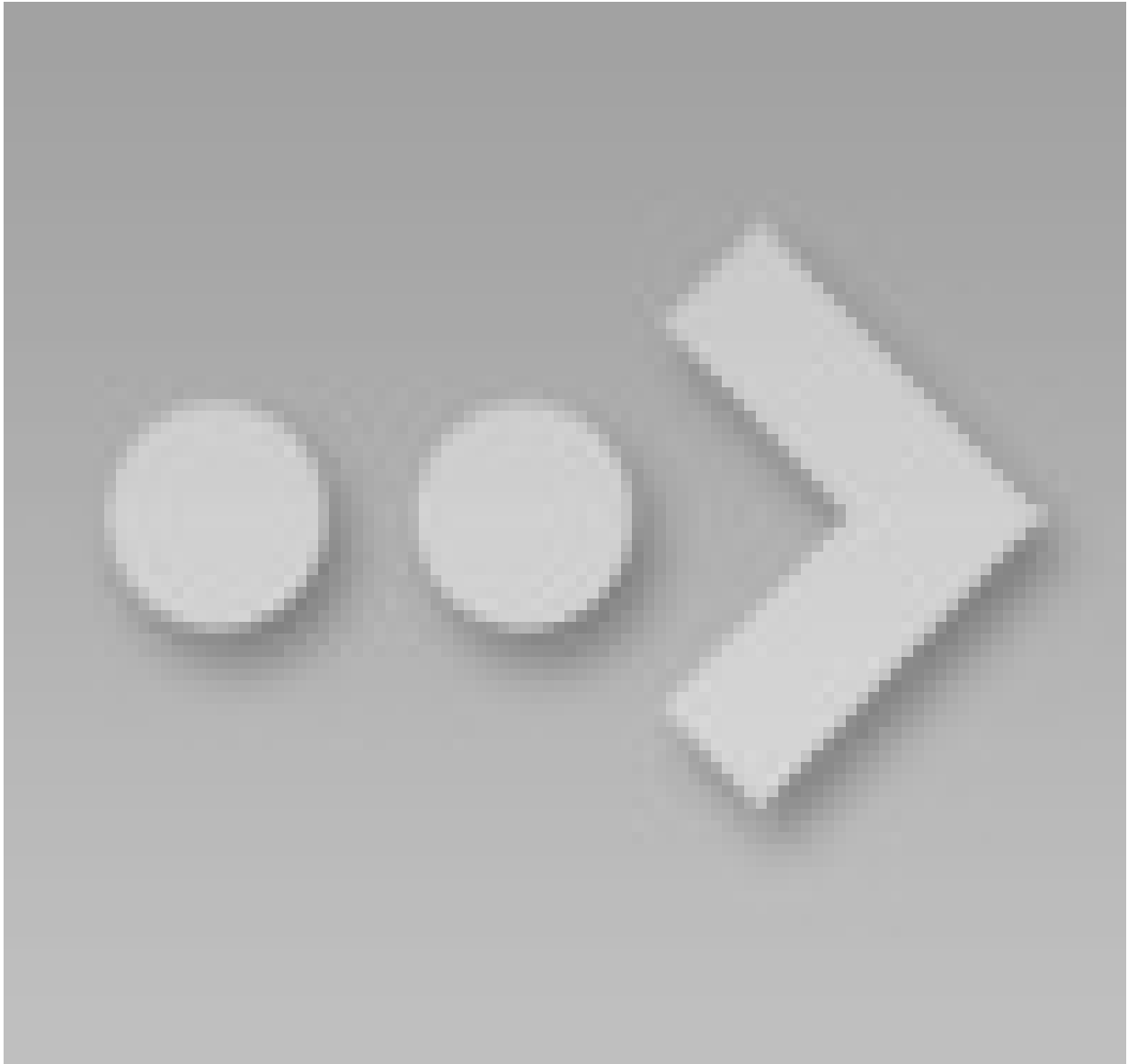
The first tab provides the ability to manually pick and choose the dimension members to be included in the set as we have done with the first method above. The remaining two tabs can be used to base the set on a condition or ‘Top N’, respectively. Computed sets like this will dynamically change when the dataset is updated. Whichever method you choose to create your set, a new field will be created to use in the following examples.

Five ways to use Tableau sets

1. As a filter

Sets can be used as a filter by right-clicking on a set from the Sets Shelf and choosing “Show Filter”. Sets are binary, so when you add them as a filter, you will only have the option to choose whether the marks on the view are in or out of the set. Here’s how my bar chart of sales by customer name in descending order looks after showing the filter for Top 20 Customers by Sales and keeping only the names that are in the set.

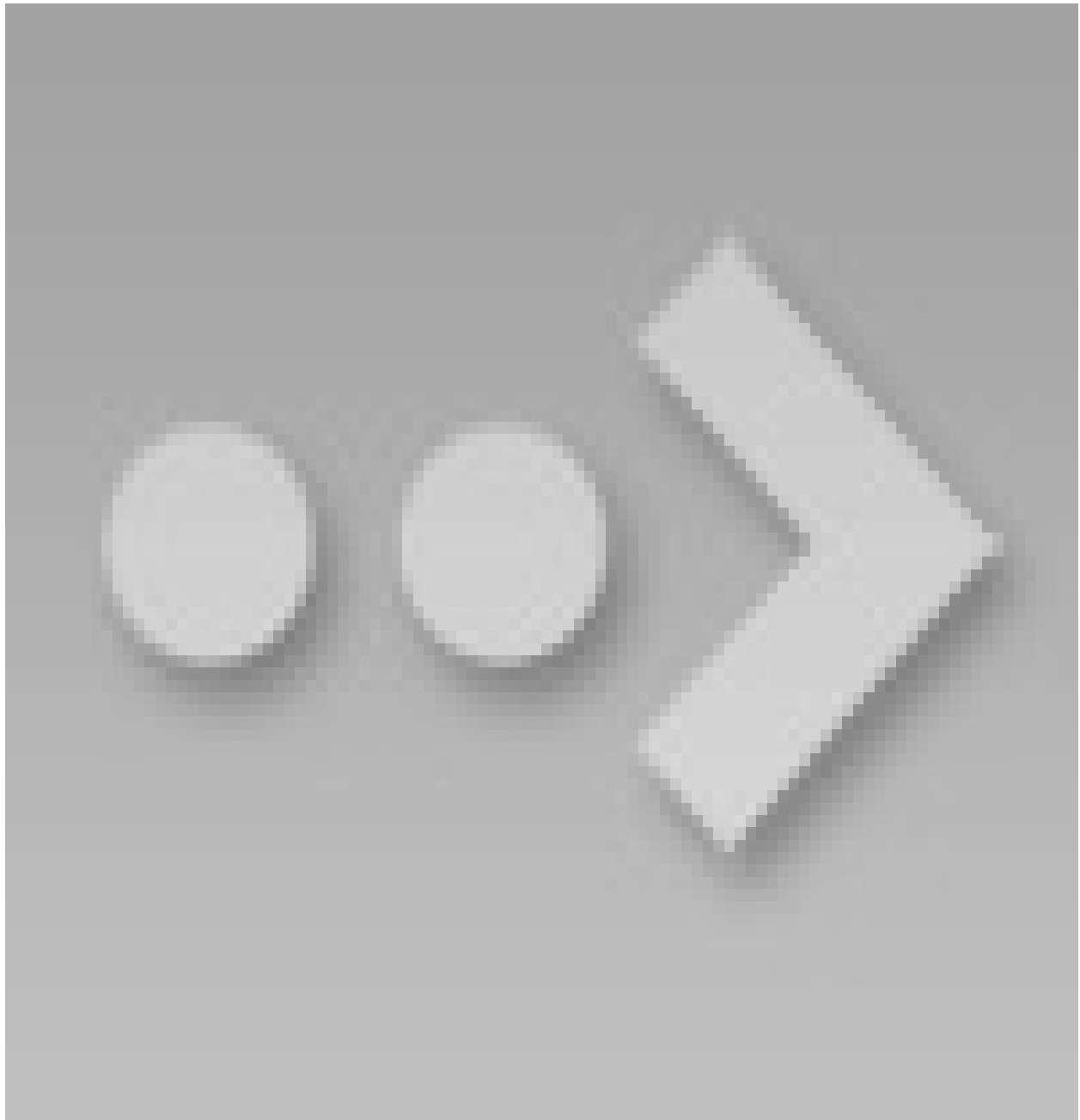
[Insert Image 15-3]



2. To encode marks

You can encode marks on a view by a set by dragging the set from the Sets Shelf to the Color Marks Card. Here's an example where I've done this to make the top 20 customers by sales stand out on a scatter plot.

[Insert Image 15-4]



3. In calculated fields

Sets can be used in calculated fields just like dimensions or measures. This way you can treat dimension members differently based on whether or not they are in a set. Here is a simple example being used to rename the sets: [Insert Image 15-5]

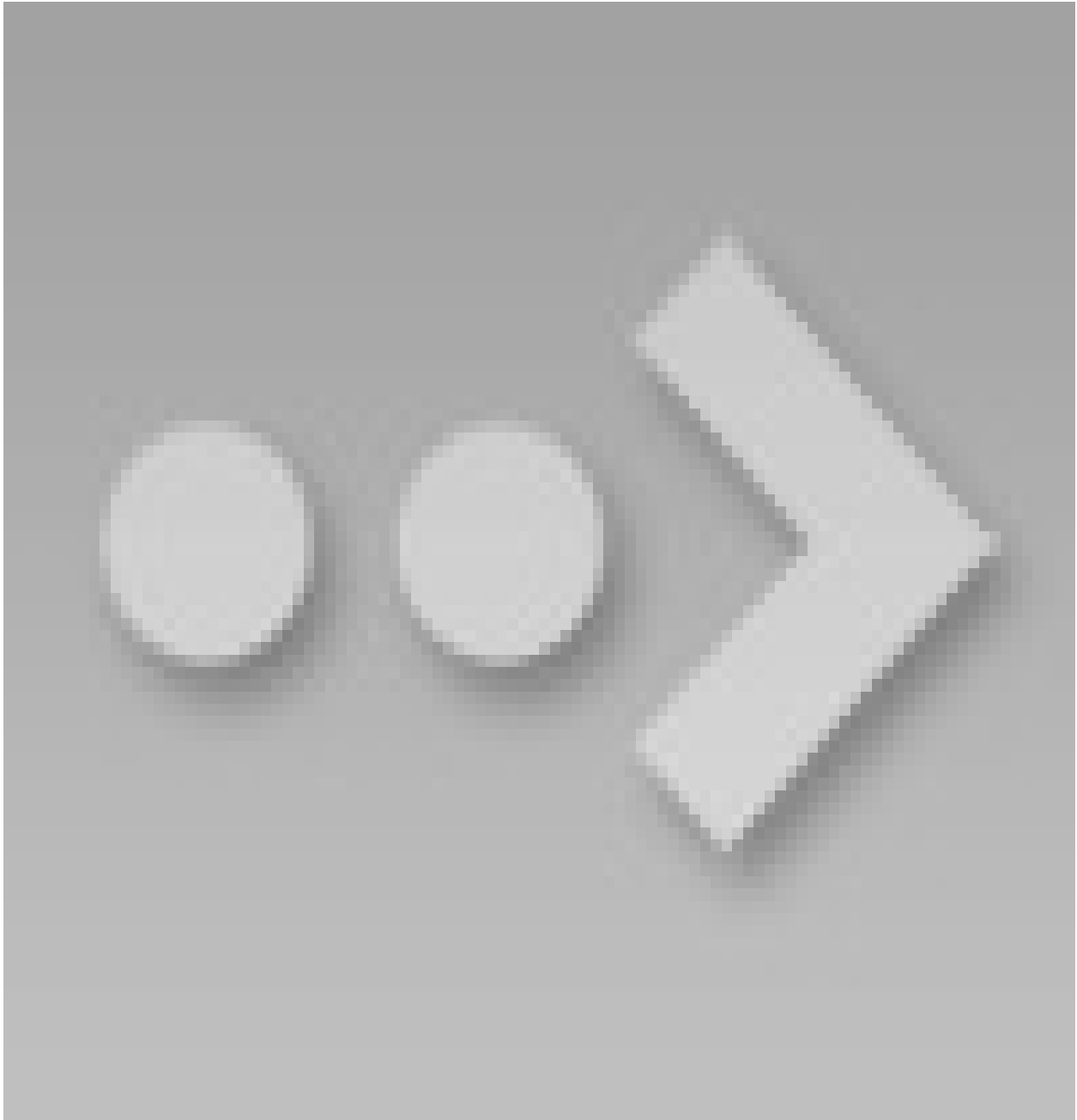


This formula is saying that if a customer name is in the set, name the set “Top 20 Customers”; otherwise call it “Other”. Note that this example is used to show you that sets can be used within calculated fields, but this same renaming could have been accomplished by editing the aliases of the set.

4. As dimension fields

Sets can be used just like dimensions in that if you add a set field to the Columns Shelf or Rows Shelf, the view will be ‘sliced and diced’ by that field. Here is how my scatter plot from earlier looks if I create two separate columns for in and out of the set by dragging the set from the Sets Shelf to the Columns Shelf.

[Insert Image 15-6]

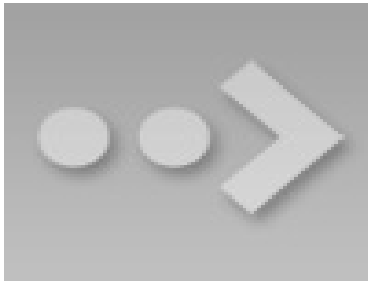


5. Within a custom hierarchy

Sets can be used as part of a custom hierarchy in Tableau, which allows you and your end users to easily drill down and back up across different dimensions. For example, you may want to create a hierarchy with customer names that starts with the Top 20 Customers by Sales set, then drills down to individual customer names, then to their segment.

Create a custom hierarchy by selecting the Customer Name and Segment dimensions, right-click, hover over “Hierarchy”, and choose “Create Hierarchy...”. Once the custom hierarchy is created, you can drag the Top 20 Customers by Sales set into the hierarchy on the Dimensions Shelf and re-arrange them to the desired order of the drilldown. At this point, I see this hierarchy on the Dimensions Shelf.

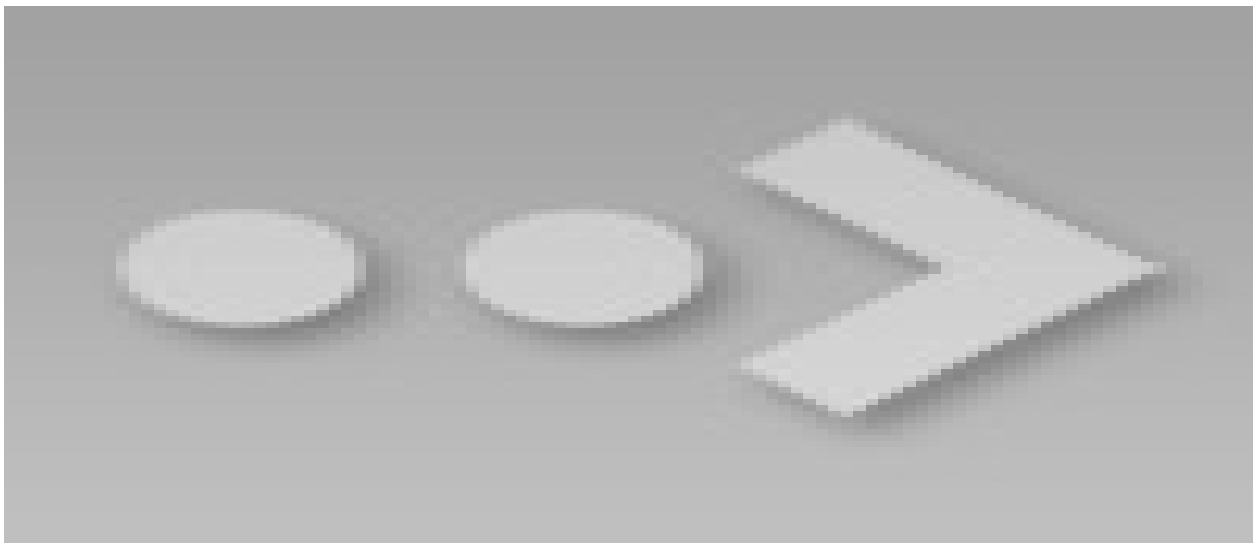
[Insert Image 15-7]



Now that the hierarchy is in place, if I replace the Customer Name dimension with the Top 20 Customers by Sales set dimension in the bar chart example from above, I will be able to drill down from the top 20 set, to customer name, to segment. This is accomplished by clicking the “+” symbol on the field(s) on the Rows Shelf.

Sales by in / out of set.

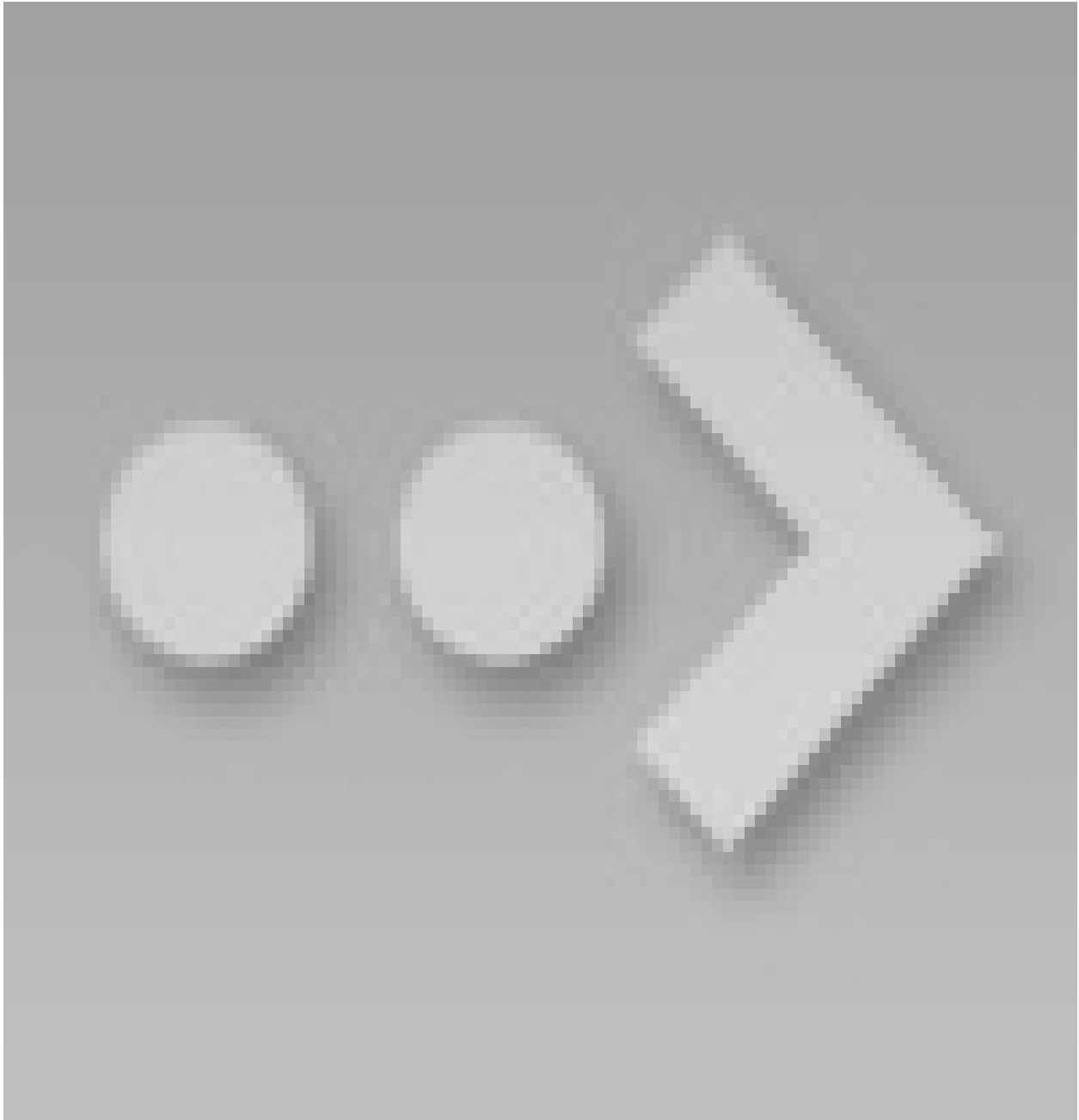
[Insert Image 15-8]



Sales by in / out of set and customer name

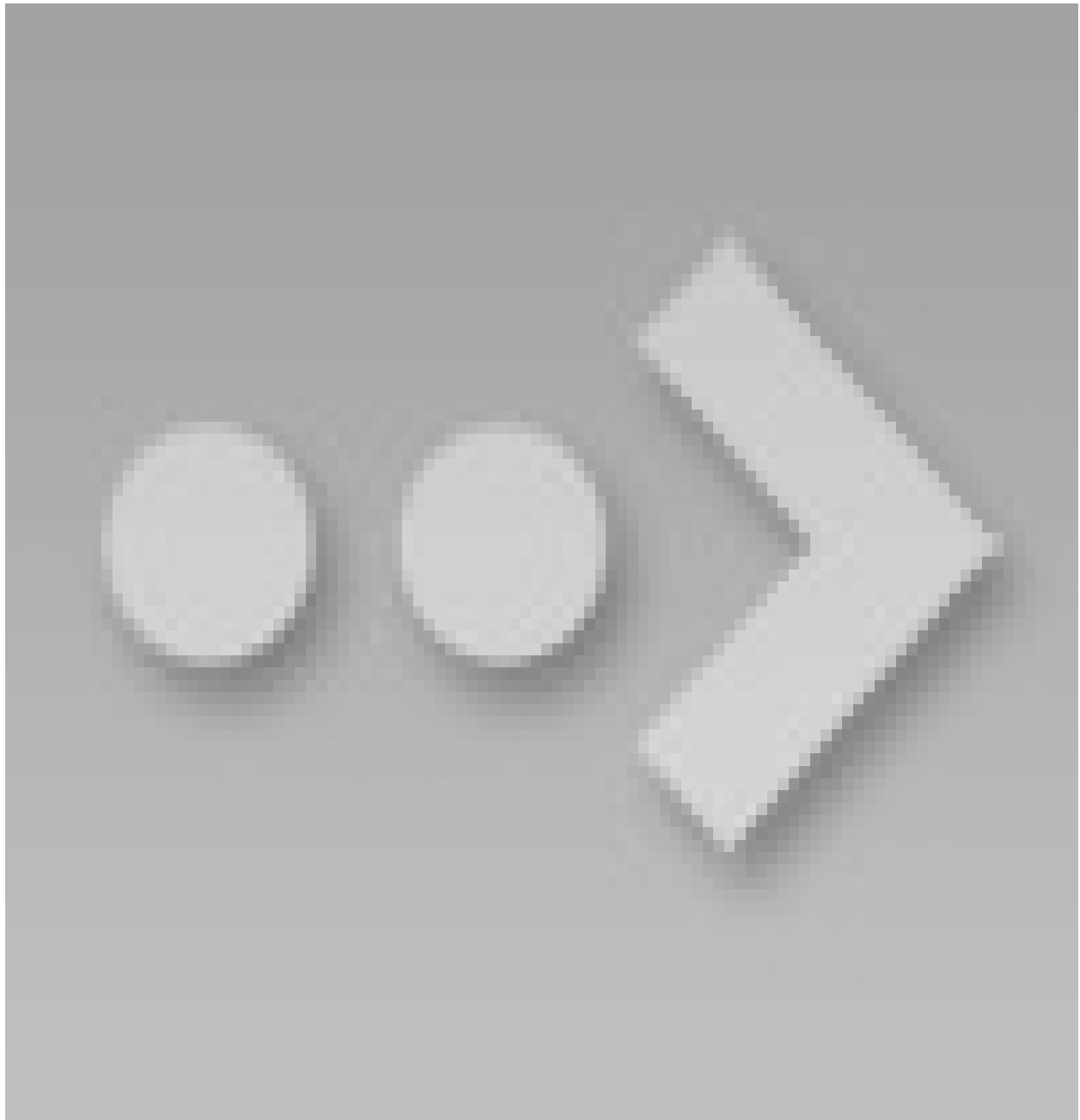
Sales by in / out of set and customer name.

[Insert Image 15-9]



Sales by in / out of set, customer name, and segment

[Insert Image 15-10]



Chapter 16 – An Introduction to Level of Detail Expressions

Tableau level of detail expressions allow you to change the most granular place where an analysis takes place. An analysis, such as `AVG(Sales)` by State includes the dimension that you are slicing the dicing the measure by as well as the aggregation of the measure. Prior to Tableau version 9.0, and without some clever technical hacking such as duplicating a data source or leveraging table

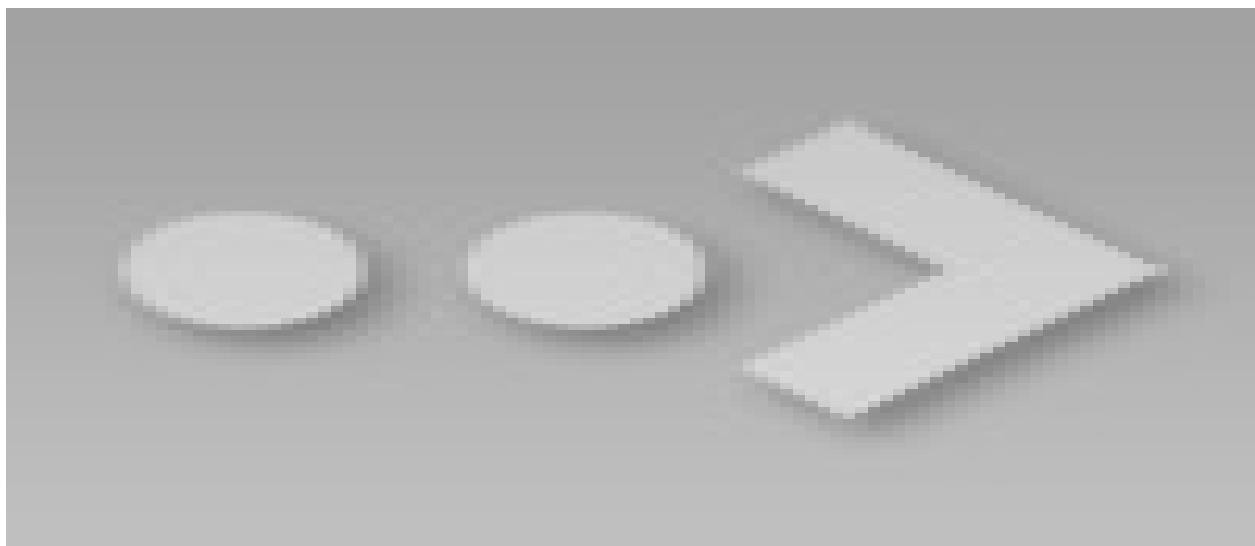
calculations, you were stuck using the same level of detail for an entire view. Now that you can alter the level of detail for specific measures, you can compare and contrast numbers at different granularities within the same chart.

With level of detail expressions in Tableau, you can now compute things such $\text{AVG}(\text{Sales})$ by State minus $\text{AVG}(\text{Sales})$ for the entire dataset to see how the sales per state compare to the overall average.

This unlocks a great deal of flexibility in your analyses because you can explicitly define the level of detail for different numbers in your view. [Here are just 15](#) of limitless ways to take advantage of Tableau LOD expressions. I will also share a couple of resources that show you how I've used level of detail in my own Tableau workbooks at the end of this chapter, but this chapter mainly serves as an introduction to what level of detail is and the syntax needed to control it.

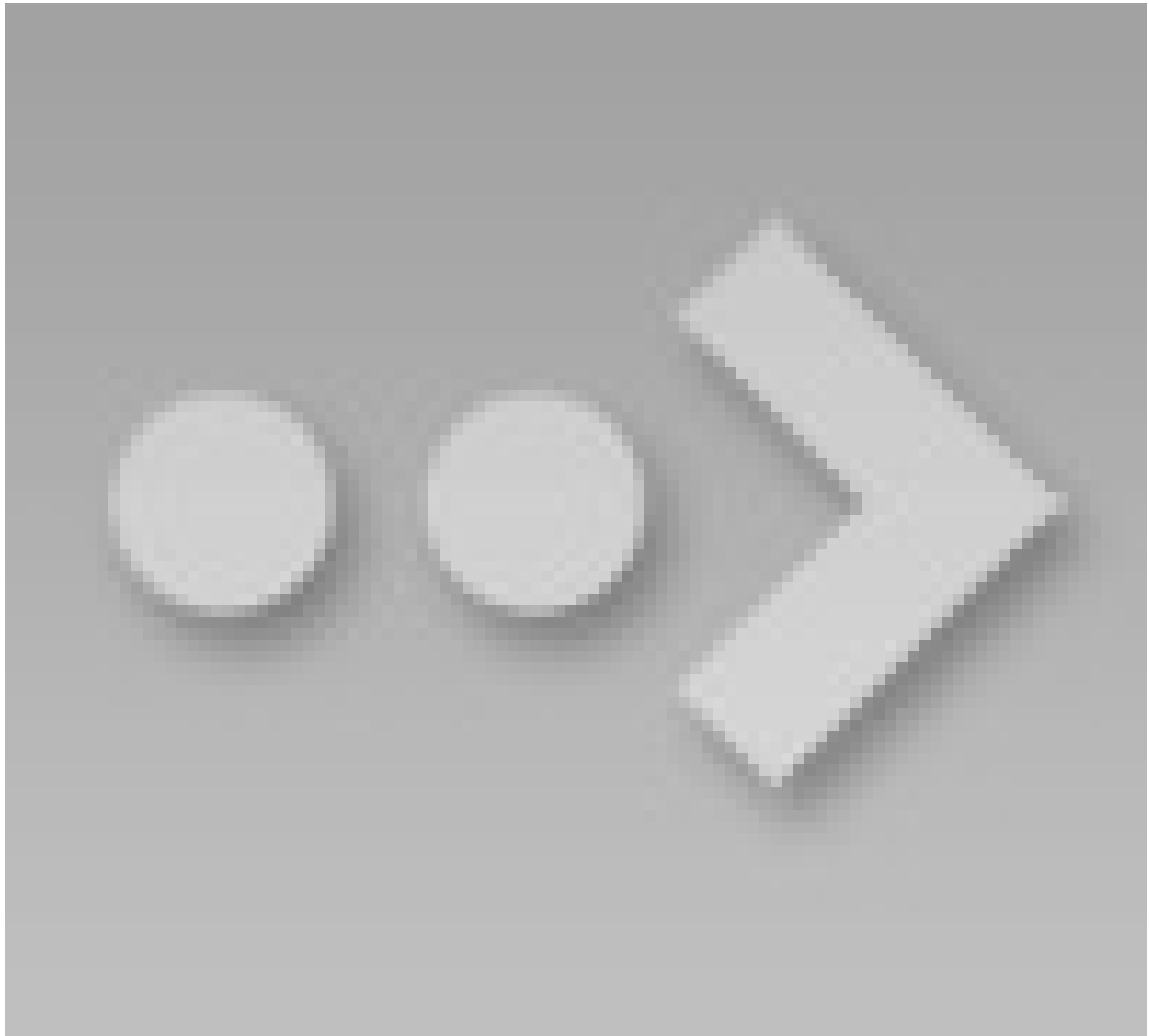
An introduction to Tableau Level of Detail expressions

Earlier in *Practical Tableau*, we introduced the idea of level of detail and how you can make a visualization more granular by [adding dimensions to the Detail Marks Card](#). Here is another example to help illustrate what a Tableau visualization's level of detail is: [Insert Image 16-1]



In this simple, Sales by Category bar chart from the Sample – Superstore dataset, the most granular level of detail in the analysis is Category. Since we are only slicing and dicing the Sales measure by the Category dimension, there are only

slicing and dicing the Sales measure by the Category dimension, there are only three marks; one for each category. We could make this analysis more granular by adding an additional dimension to the view. Here's what the bar chart looks like if I add the Sub-Category dimension to the Rows Shelf: [Insert Image 16-2]



We are now slicing and dicing the Sales Measure by both the Category and Sub-Category dimensions. We have changed the level of detail for this view and made our analysis more granular.

But what if we wanted to look at it both ways? Perhaps we want to show the sales per sub-category in one column, the sales per category in a second column, and even divide the two by each other to determine how much each sub-category is contributing to their respective category.

That's where Tableau's Level of Detail expressions come in. Level of Detail expressions have their very own syntax in Tableau, which looks like this:

- An open curly bracket "{"
- Followed by one of three level of detail expressions: FIXED, INCLUDE, or EXCLUDE
- Followed by zero or more dimensions you want to be fixed, included, or excluded from the computation
- Followed by a colon ":"
- Followed by an aggregated measure
- Followed by a closing curly bracket "}"

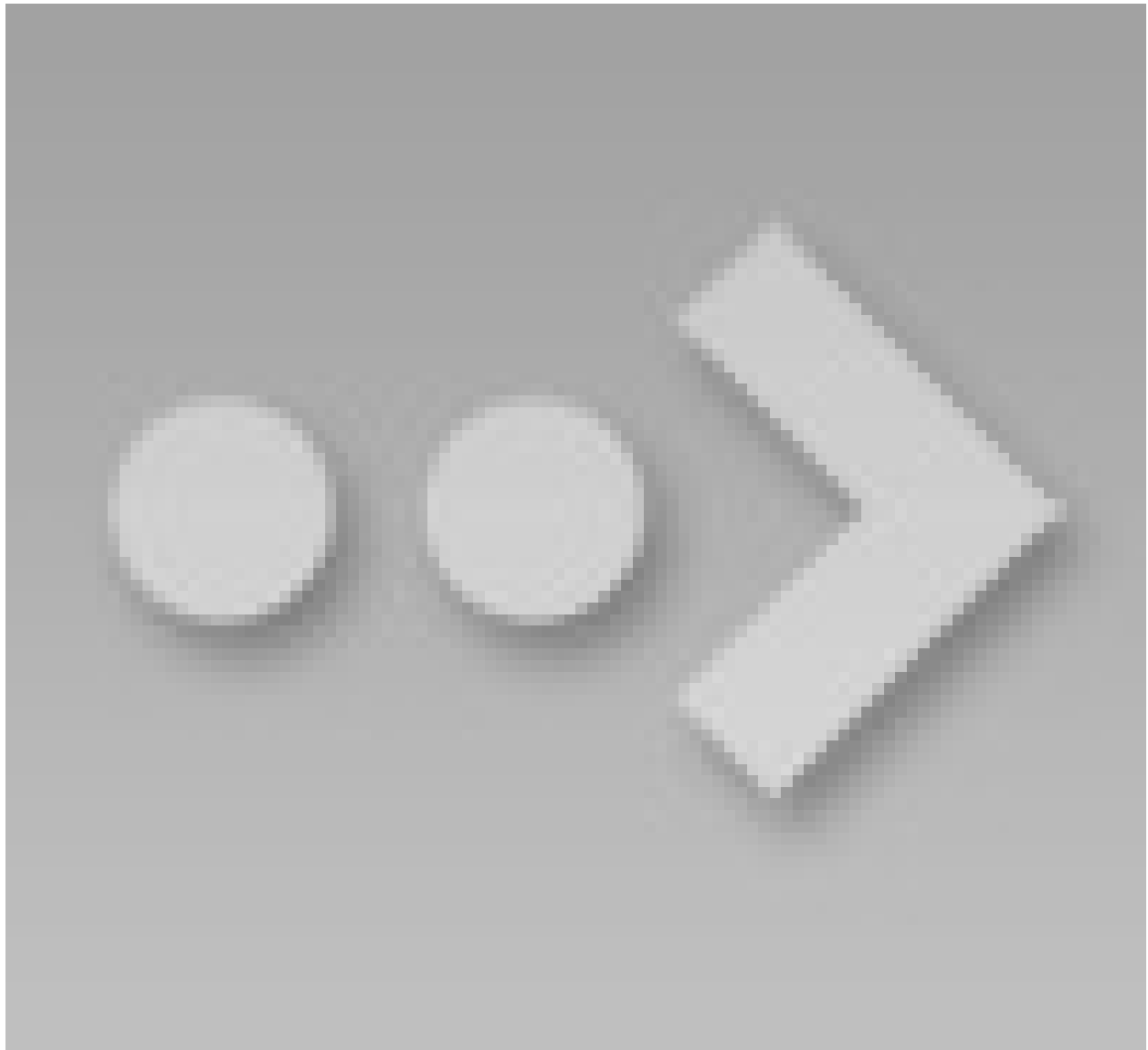
The three level of detail expressions, FIXED, INCLUDE, and EXCLUDE, are fairly intuitive names for what they're going to do (i.e. fix the measure at a certain level of detail, include dimensions that are not on the view, or exclude dimensions that are on the view). As with many things in Tableau, there are multiple ways to get to the same answer, and there is not always a "best" way to go about a certain solution.

Your Level of Detail expressions may take some experimenting to get the result you want, but I have found the EXCLUDE expression to align best with the way I think about aggregations. The reason this works well for me is because I can make a view as granular as I'd like as I normally would, then "back out" to a comparison metric by excluding certain levels of granularity.

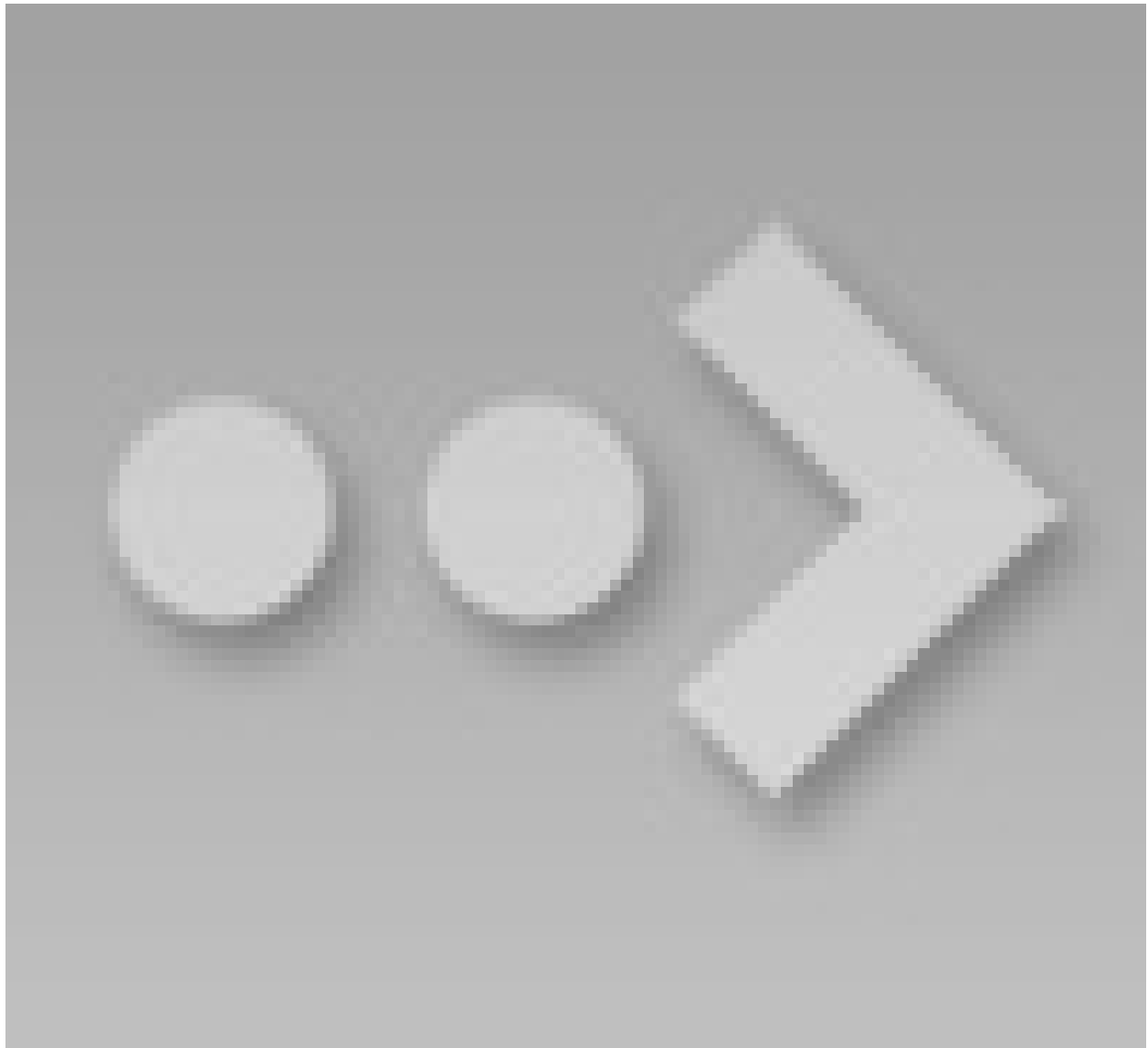
Take another quick look back at the last bar chart. I've made the analysis as granular as I'd like, slicing and dicing the Sales measure by the Category and Sub-Category dimensions. If I wanted to compare each row by the Sales at the Category level only, so excluding the Sub-Category dimension, I would create a calculated field with this formula: {EXCLUDE [Sub-Category]: SUM([Sales])}

Now when I add my new calculation to the Columns Shelf, the left side shows my original analysis, while the right side "backs out" the level of detail to the Category level, excluding the Sub-Category dimension.

[Insert Image 16-3]



We're now displaying two different levels of detail on one view! This new measure can now be used for further analysis. If I wanted to calculate the contribution of each sub-category to its respective category, I could create another calculated field that divides SUM(Sales) by the newly created SUM(Sales Excluding Sub-Category): [Insert Image 16-4]



This third column could have been computed using a [Tableau table calculation](#) that took the percent of total per pane, but using Level of Detail expressions provide several benefits including:

- You can “show your math”; in the example above, we would not have been able to show the first and second column together without LOD expressions. You could have also used a different table calculation called WINDOW_SUM, but with LOD expressions...
- You no longer need to worry about setting and maintaining the direction and scope of a table calculation; set the level of detail once in the LOD calculation and don’t worry about it again
- You can definitely get creative with table calculations in Tableau, but LOD

calculations are even more flexible, allowing you to specify the precise level of detail and aggregation for a measure

We have barely scratched the surface on arguably the most powerful feature to ship with Tableau version 9, but this example should have provided a good foundation about what level of detail is and how it can be changed to benefit your analysis. You will have to experiment with your own use cases and LOD calculations, but in addition to the 15 uses of Tableau LOD expressions linked to above, here are a couple more from *Practical Tableau*:

- Using LOD Expressions to Create Benchmarks
- How to Compare Unequal Date Ranges on One Axis

Chapter 17 - An Introduction to Dashboards and Distribution

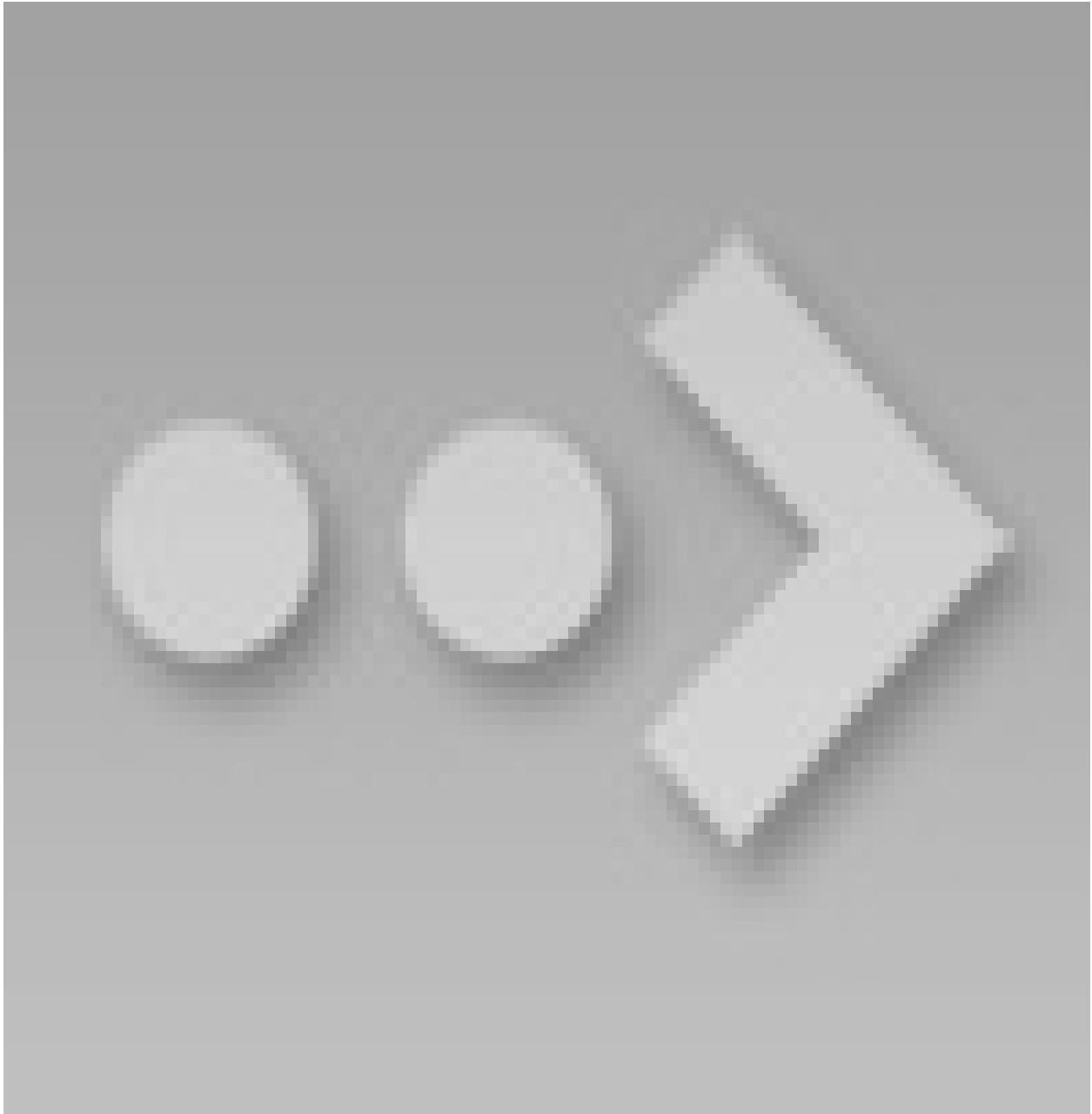
Individual worksheets in Tableau can lead to powerful insights that help your business, but many times, it makes sense to combine the worksheets into a single dashboard. By combining varying visualizations into a dashboard, you and your audience are able to analyze different aspects of the data in context of each other. This is a much more intuitive experience than viewing the visualizations individually.

In addition to this one obvious benefit, Tableau comes with several technical features that allow you to control the user experience and even the ability to have the individual components of the dashboard interact with each other. This chapter offers an introduction to dashboards in Tableau and several ways to distribute the dashboard after it has been created.

An introduction to dashboards in Tableau

For this introduction, we will recreate this dashboard in Tableau:

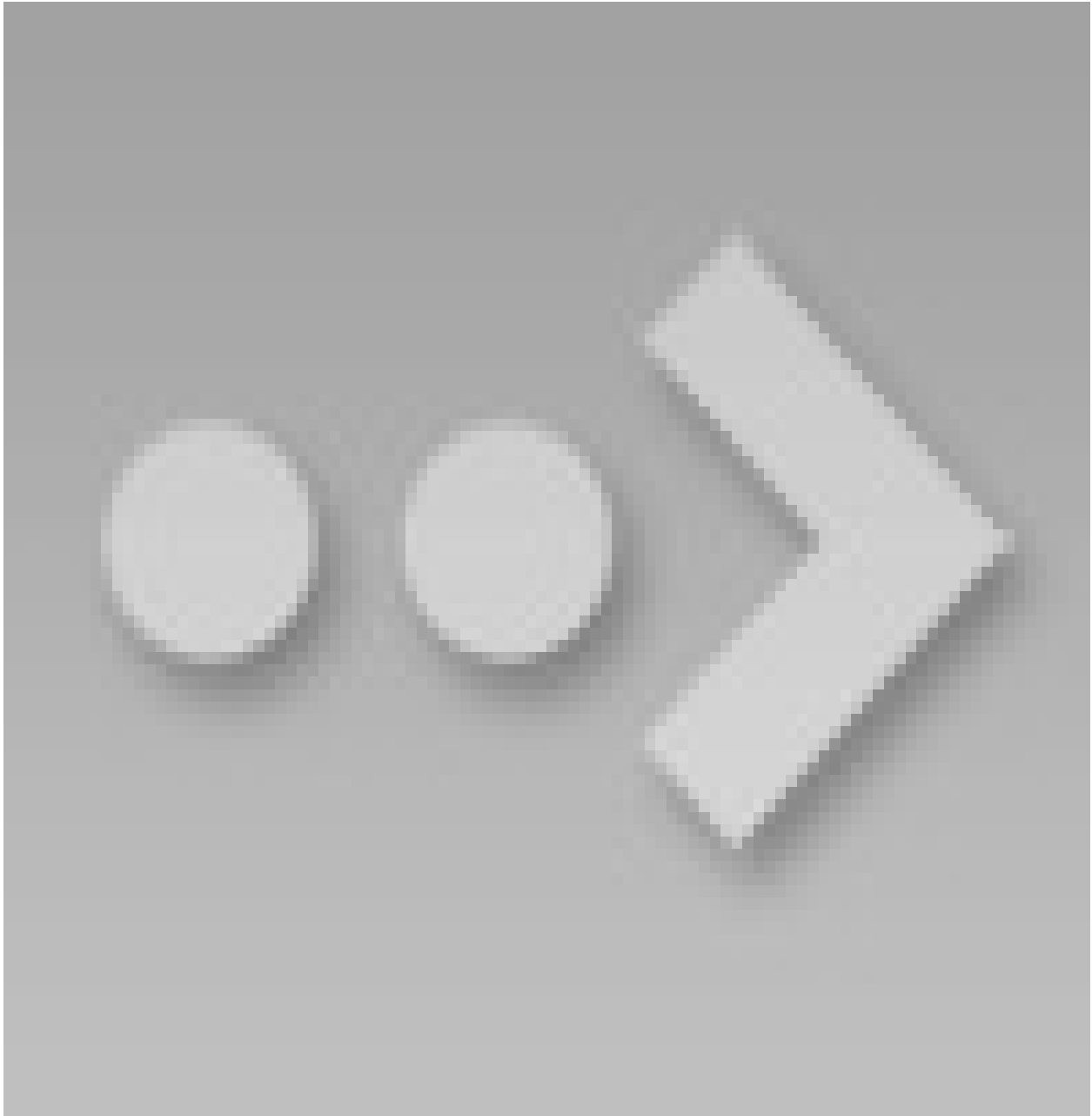
[Insert Image 17-1]



To create a new dashboard in Tableau, either click “Dashboard” in the top navigation and then “New Dashboard” or click the New Dashboard icon at the bottom of any worksheet. The New Dashboard icon is the second icon immediately following the existing worksheets in the workbook.

Upon creating a new dashboard, you will be provided a blank slate that looks like this:

[Insert Image 17-2]



Here's a quick overview of all the different dashboard options in the left navigation:

Dashboard and Layout tabs

By default, you will be working on the dashboard tab which allows you to set most aspects of the dashboard. The layout tab allows you to set the dimensions and location of individual dashboard components. All sizes on the Layout tab are in pixels.

Device Preview Button

The Device Preview button allows you see what the dashboard will look like on different devices and you can even save different versions of the dashboard so that it looks different depending on what device it is displayed on. For more, see the chapter, *Designing Device-Specific Dashboards in Tableau 10*.

Size

This is where you can set the height and width of the dashboard in pixels. There are several preset size options or you can set an exact size of your choosing. If you choose the Automatic option, the dashboard will change to fill all available space on the screen it is being displayed on and resize the individual components of the dashboard accordingly. While this option sounds good on the surface, beware that it is not truly “responsive”, and the display can be somewhat unpredictable.

Sheets

These are the individual worksheets in your workbook that can be added to the dashboard. Note that to create the example pictured above, there are three sheets in the workbook: Map, Trends, and Bar Chart. It helps to give the worksheets good names so you can easily find them, but you can also get a thumbnail preview of the worksheet by hovering over the name in the left navigation.

Objects

Horizontal: Adds a horizontal layout container that additional objects can be added to.

Vertical: Adds a vertical layout container that additional objects can be added to.

Text: Opens a mini word processor where you can add and format any text you wish.

Image: Adds an image from your computer to the dashboard.

Web Page: Embeds a web page in the dashboard (requires an Internet connection to display the web page).

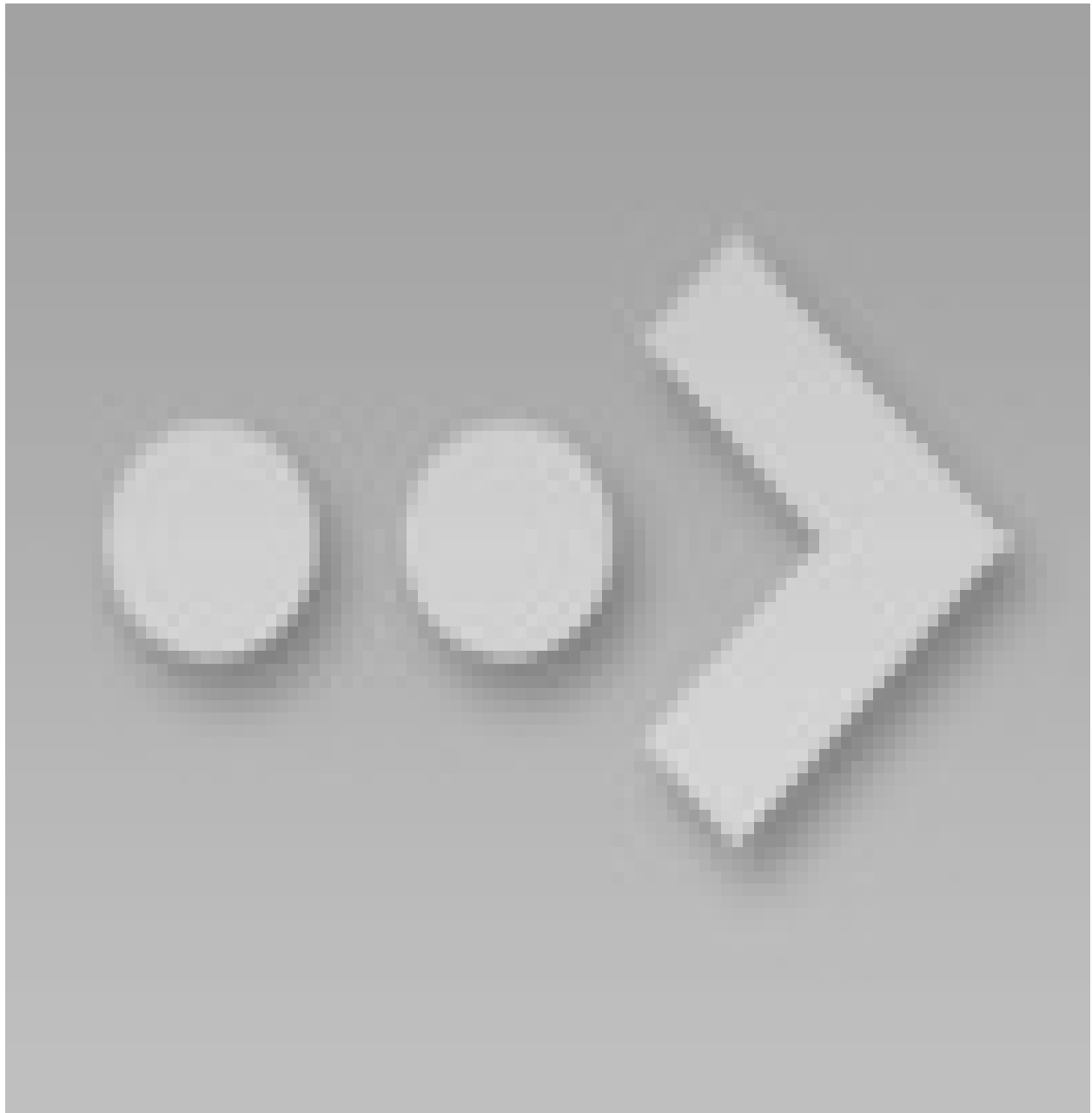
Blank: Adds blank space to the dashboard which can be helpful when dashboard elements are too close to each other and in a tiled layout.

Tiled or Floating

When dashboard elements are tiled, they fill all available space in their respective tiles. When dashboard elements are floating, you control their exact size and location on the dashboard. Each of these layouts comes with their own pros and cons and the choice is largely dependent on your specific use case. Most Tableau users prefer the automatic resizing that comes with a tiled layout; I personally prefer the predictability and precision that comes with floating elements.

The dashboard pictured above is tiled in the Desktop Browser size, which is 1000 pixels wide by 800 pixels tall. To create the dashboard, I first ensured the size was 1000 by 800 by clicking on Size in the left navigation, choosing Fixed Size, then choosing the preset Desktop Browser option.

Tiled is the default layout option, so I am ready to go ahead and add individual sheets to my view. I will add the map first, and when there is nothing else on the dashboard, it will fill all the available space on the view: [Insert Image 17-3]



To add a worksheet to a dashboard, left-click and drag the worksheet name from the left navigation to the dashboard. Note that when I added a worksheet that has legends, the legends were added for me automatically. The color legend can be used as a highlighter by simply clicking a color in the legend. To learn how to create the map pictured, see the chapter, [How to Make a Dual-Axis Map](#).

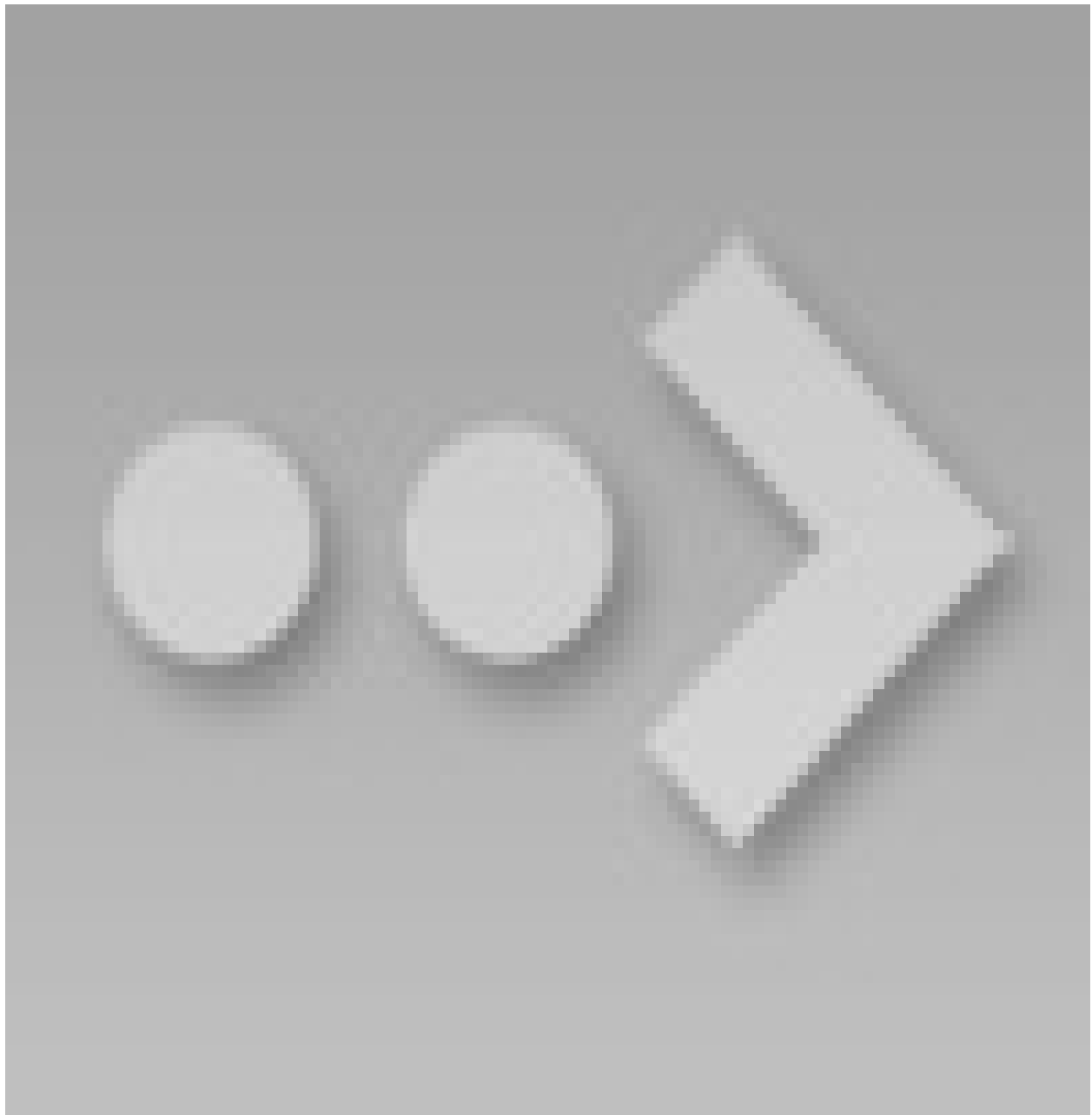
I will then add the Trends worksheet in the space below the map by dragging it

towards the bottom of the dashboard. To learn how to make a line graph, see the chapter, Line Graphs, Independent Axes, and Date Hierarchies. To learn how to calculate the month over month difference pictured, see the chapter, An Introduction to Table Calculations in Tableau.

I will then add the Bar Chart worksheet by dragging it near the bottom right-corner of the dashboard. To learn how to make a bar chart in Tableau, see the chapter, 5 Ways to Make a Bar Chart / An Intro to Aggregation.

Lastly, I added a title to the top by dragging a text object to the top of the dashboard, and ended up with this view:

[Insert Image 17-4]



This is a more than serviceable dashboard that we were able to make in a matter of minutes in Tableau. You could add even more value by building in some user interactivity. Perhaps you want the trend lines and bar chart to update when a region is clicked on the map. To learn how to do this, see the chapter, *3 Creative Ways to Use Dashboard Actions*. Another option would be to add filters that allow the end user to filter the three worksheets. For more on filters, see the chapter, *An Introduction to Filters*.

Distributing Tableau dashboards

After you've created a dashboard in Tableau, there are several ways the dashboard can be distributed.

Packaged Workbooks

If you have created a dashboard using Tableau Desktop, you can package the workbook for offline distribution. To package a workbook, navigate to File in the top navigation and click "Export Packaged Workbook...". This will package the data with the instructions for how to visualize the data. Anybody with Tableau Desktop or Tableau Reader can open the file and interact with the visualizations you have created. Note that packaged workbooks do not automatically update so the data within the workbook will only be as recent as the last update.

Tableau Public

Any dashboard built in Tableau Desktop or Tableau Public can be published to the web for public consumption. This is not a realistic approach for sensitive business data, but if you are able to make your data public, this is the perfect solution for distributing your dashboard to the widest audience possible. To publish a dashboard from Tableau Desktop to Tableau Public, navigate to "Server", hover over "Tableau Public", and choose "Save to Tableau Public As...".

Tableau Server / Tableau Online

You can also distribute your workbooks privately on-premise or in the cloud by using Tableau Server or Tableau Online, respectively. Tableau Server requires incremental licenses for you and your end users, but is the most scalable Tableau solution for distributing your business-related workbooks. Publishing to Tableau Server is very similar to publishing to Tableau Public, but you will be presented with several additional options including where to publish the workbook within Tableau Server, who has permission to view and interact with the workbook, and if / when you want the data in the workbook to update. To publish a workbook to Tableau Server, navigate to "Server" in the top navigation, and choose "Publish Workbook...". Tableau Online works similarly, but it is hosted by Tableau in the cloud.

Chapter 18 – Introducing the INSIGHT Framework for Data Visualization

To this point in *Practical Tableau*, we've introduced the basics, provided how-to tutorials for dozens of chart types, and covered some of my favorite tips and tricks. All of this will help you on your journey to becoming a Tableau Jedi, but it is not enough. It doesn't matter how good you are at the technical aspects of Tableau if you don't consider some intangibles that will help tie it all together. This section is the missing link to help bring your data visualization from good to great.

Over the past eight years, I have personally designed and constructed hundreds of data visualizations. To this day, I have yet to build a dashboard that *every* viewer thought was perfect. Data visualization is an art, and its value is in the eye of the beholder. There are limitless possibilities and variables involved in creating a data visualization, too; no wonder that many people, when confronted with a data visualization issue, stick to an inoffensive template that does ... okay. But you can do better! With a structured approach to the problem of data visualization, you can maximize the chances that your work will be well-received—and communicate the information inside the data.

The strategic framework I've developed—the INSIGHT framework—has saved me hundreds if not thousands of hours of iteration, increased value for stakeholders, and led to countless business insights – and now I'm sharing it with you!

[Insert Image 18-1]



Identify the Business Question – one of the most common pitfalls in data visualization is trying to answer too many questions in one view. By prioritizing what the data visualization will answer, you are able to keep the view simple and make it clear to the end user what stories they should be looking for in the dashboard.

Name the KPIs – these Key Performance Indicators (KPIs) should all help answer the business question that the data visualization is trying to answer.

Shape the Data – in order to work with the data in a data visualization software, data should be prepared in tabular form in advance. For the most flexibility, each column header should represent a unique field, with subsequent rows representing every entry for each of those fields.

Initial Concept – get an idea documented, even if it is simply a sketch of the direction you are planning to take. By not committing too much time to the initial concept, you will have the flexibility to pivot based on stakeholder input.

Gather Feedback – ask end users what they think of your initial concept and if they believe it will meet their needs. This step helps instill some stakeholder ownership during the design process so they are more likely to support the final product.

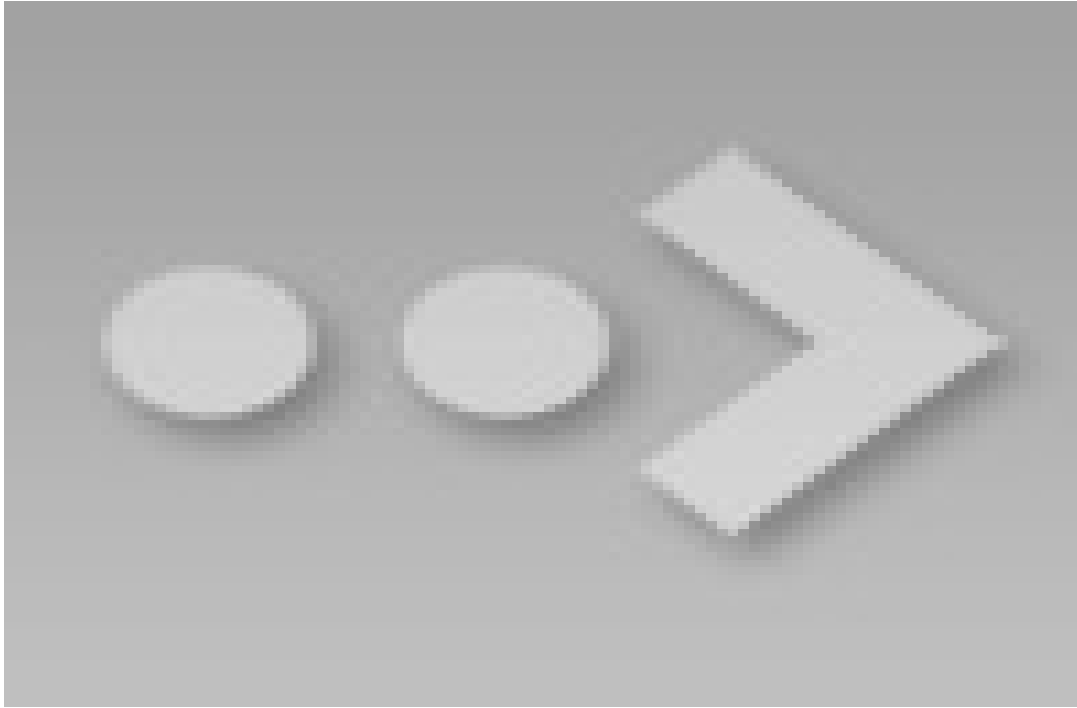
Hone the Dashboard – this is your opportunity to incorporate stakeholder feedback and finalize the data visualization.

Tell the Story – finally, you are ready to distribute the dashboard and allow the stories in the data to begin making an impact.

The following chapters will take you through each step of the process.

Chapter 19 – Identify the Business Question

[Insert Image 19-1]



The first - and most critical step in the INSIGHT framework for data visualization - is to identify why you are making the data visualization to begin with! As a data visualization practitioner in a corporate environment, I call this step “Identify the Business Question”, but the concept can be applied to any question you are attempting to answer through visualizing data, whether that’s a business question or just something you’re curious about.

What is contributing to our business growth?

Where is the best place to get a cup of coffee in Seattle?

What are the odds of becoming a professional athlete?

It’s incredible to see how often this fundamental step is overlooked in the real

It's incredible to see how often this fundamental step is overlooked in the real world. My theory as for why is that it is becoming increasingly easy to access data and it can be tempting to look at too much at one time. While it may sound convenient to put every number you care about on a single dashboard, this approach typically does more harm than good.

By having a single business question in mind going into a data visualization design, you set yourself and your end users up for success by providing a clear purpose for the dashboard. Furthermore, most times it's appropriate to literally state the question to open a dashboard. This not only provides a purpose, but gives you an opportunity to frame what an end user will be looking for in your dashboard even when you're not there to explain it to them.

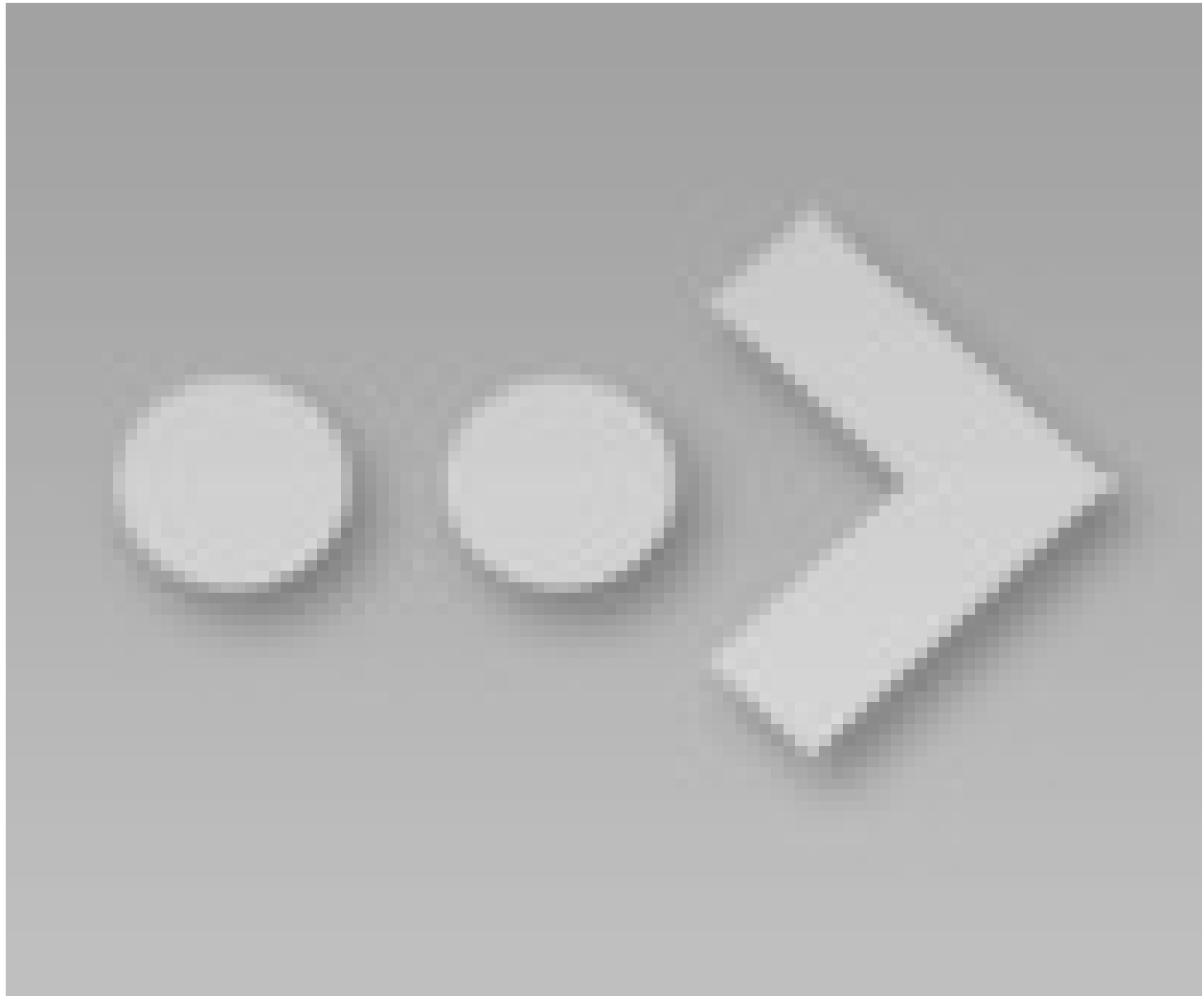
Notice that this step is called *Identify the Business Question*, and I can tell you that the singularity of the word 'question' is not coincidental. Some of the best data visualizations I've seen as well as some of my own most popular dashboards state a single question at the top. From there, the rest of the dashboard attempts to answer that single question. Sure, additional insights *context* 'rabbit holes' are likely to emerge, but by going into a data visualization design with one question in mind, you improve the chances of your work providing value. This is because when you identify a single question, you're almost guaranteed that your dashboard will serve at least one purpose: answering the question.

Throughout the chapters about the INSIGHT framework for data visualization, I will use my dashboard, "Do Old Movies Get Better with Age?" to illustrate how the framework can be applied in a real-life scenario. This was the winning entry in Tableau's 2013 Iron Viz Championship, a popular data visualization contest held live at the Tableau Conference each year. I'm using this as my example to show both that I practice what I preach, and that you can have success adopting this framework.

Here is a screenshot of the viz:

Here is a screenshot of the viz:

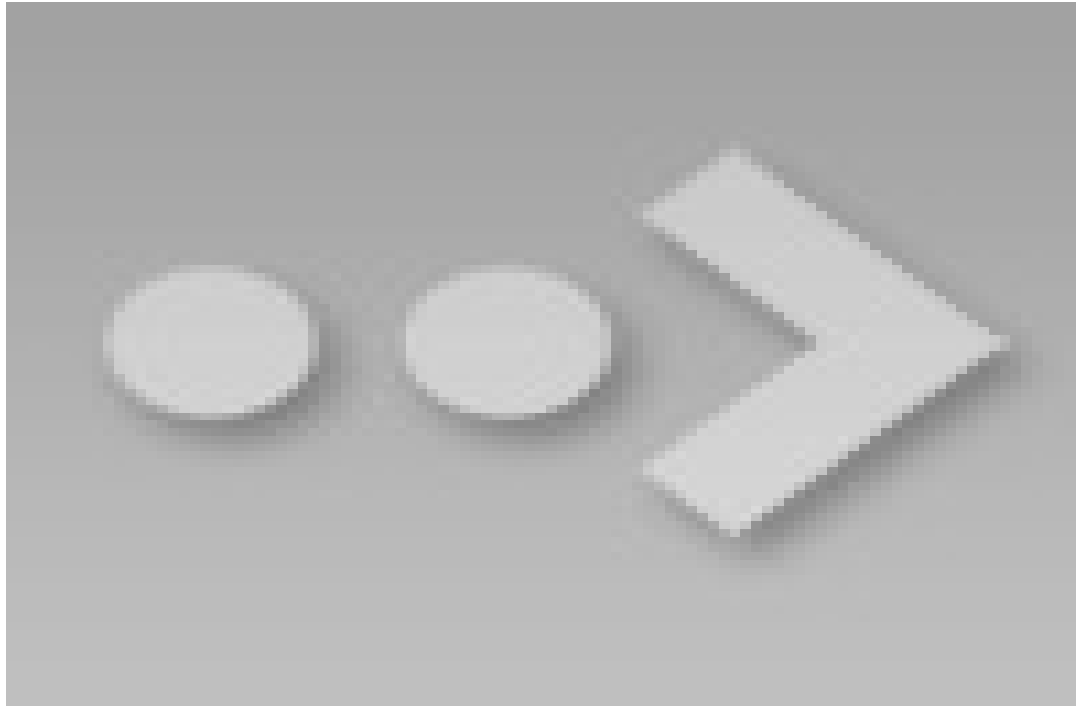
[Insert Image 19-2]



Notice that the dashboard asks a single question in the title, which makes it clear what the audience should be looking for throughout the rest of the dashboard. The next chapter will share how you help answer the business question.

Chapter 20 – Name KPIs

[Insert Image 20-1]



The second step in the INSIGHT framework for data visualization is to name the key performance indicators, or ‘KPIs’, that will help answer the business question identified as the focus for the dashboard. More specifically, the KPIs should be documented, which serves three key purposes:

- Aligning stakeholders on the data they agree will help answer the business question that is important to the organization. It is much easier to debate the merit of each KPI and make modifications to the list at this point rather than waiting until you have a data visualization developed.
- Informing the dashboard design. Once a consensus is reached on the KPIs that should be included in a data visualization, you basically have a blueprint in terms of what should be included in the final product.
- Providing a historical record so you can review how requirements change over time. It is common for a business environment to change to the point where new KPIs or ways of looking at the data emerge. You may discover a

customer segment you want to focus on or realize that a KPI is no longer helping answer the business question. Allowing some room for change is okay, but it helps to maintain a record of the dashboard evolution.

The KPI-naming exercise is not a one-size-fits-all task. It's possible that a single KPI can answer the question at hand. Other times, several more KPIs will be required. I personally try to limit my dashboards to twelve total components, including KPIs. They are called *key performance indicators* for a reason. If you are finding you have more than five or six key metrics, you likely need to put some thought into what is really moving the needle for your business and differentiate between key and secondary indicators. The secondary indicators may be better served in a drilldown type of view, or you may run the risk of them muddying up the primary focus.

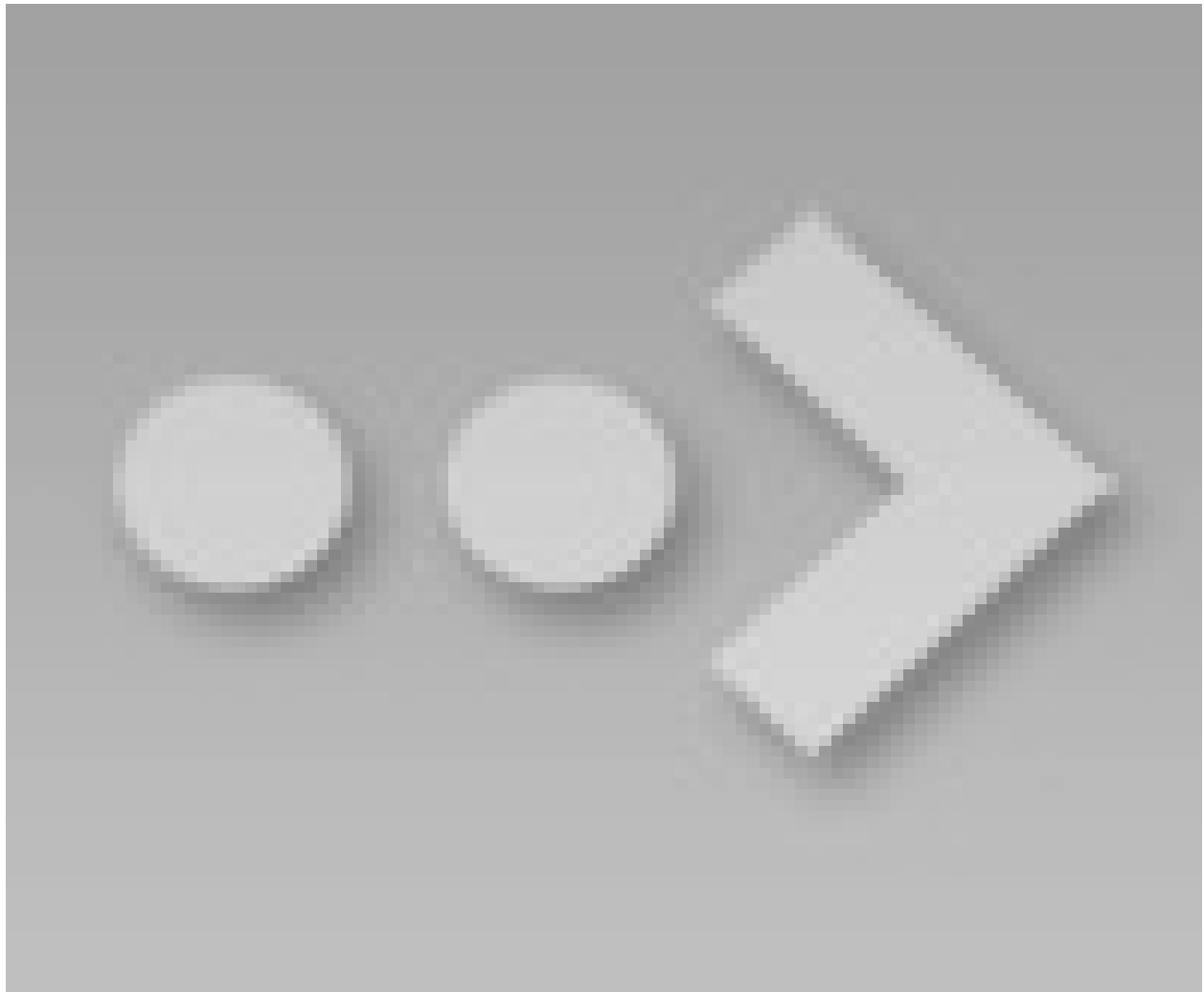
So how do you name the KPIs?

I find the best KPIs come from one of two places, or a combination of both:

1. Measurement planning. Putting some thought into how success should be measured and what information is required to answer a business question is half the battle when it comes to building valuable data visualizations for your organization. Many times, this step alone provides enough to inform a dashboard's requirements.
2. Discovery analytics. If you're attempting to answer a new business question, it is not always known what KPIs are most equipped to help answer that question. By using data visualization tools such as Tableau, there is an opportunity to quickly pivot through many metrics to get a better feel for which may be most helpful.

Let's take another look at my data visualization, *Do Old Movies Get Better With Age?*; this time to see how the KPIs ladder up to answering the identified question.

[Insert Image 20-2]



In the case of this question, “Do old movies get better with age?”, there is at least one metric that should be obviously required to help answer the question: (1) Age. From here, there was a choice on how success should be measured, or what constitutes a movie being “better”. Should the measurement of success be money the film grossed? Oscar nominations? Times it made me laugh? All

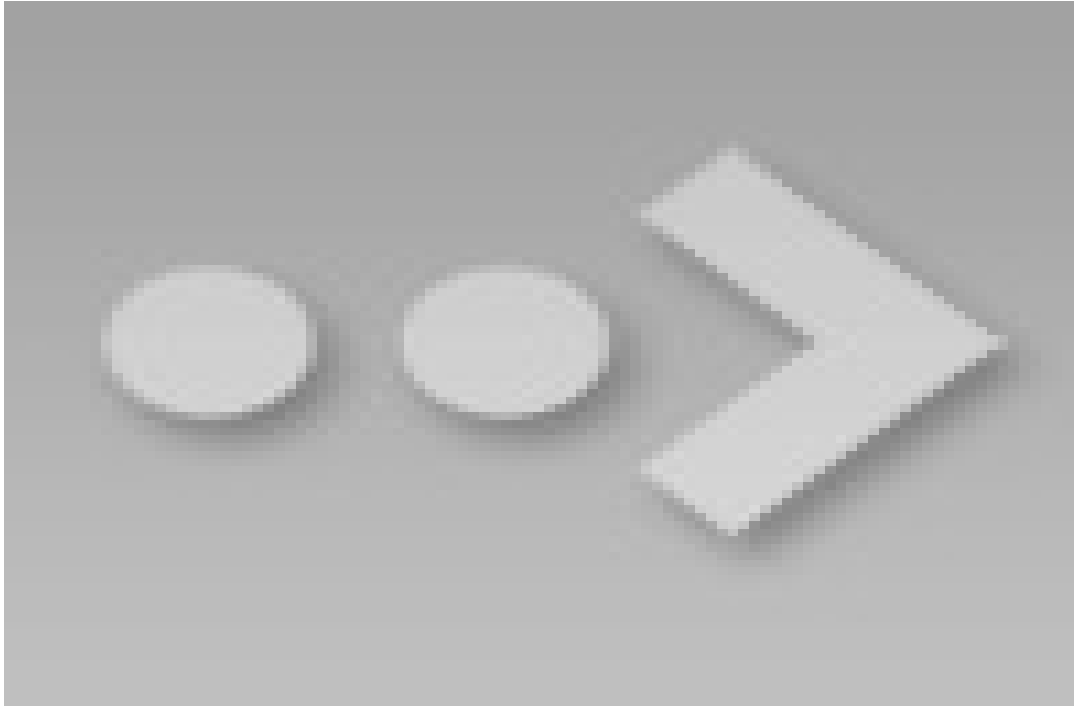
potential choices (assuming the data are available), but I chose to focus on (2) average IMDb movie rating. I also included a third metric for (3) number of reviews because I thought an important component of the analysis was that older movies tended to have a smaller sample size of reviews. But that's it – three “KPIs”.

The rest of the dashboard looks at those three KPIs in different ways in an attempt to answer the question. The first and second parts of the dashboard are descriptive in nature, providing the answer and a trend right off the bat. The final third is more prescriptive in that it slices and dices average rating by demographics, which helps answer the ‘why’ behind the results.

The next chapter will share how to prepare the KPI dataset so that it's usable in your data visualization.

Chapter 21 – Shape the Data

[Insert Image 21-1]



The first two steps in the INSIGHT framework for data visualization, (1) [Identify the Business Question](#) and (2) [Name KPIs](#), will inform what should be included in the third step: Shape the Data. In the Fundamentals section, I discussed the ideal way to shape data for use with Tableau, but this step is also about accessing the required data, doing any preparation such as joins or aggregations, and making sure the dataset is laid out for your data visualization requirements.

Mastering this step is truly half the battle when it comes to data visualization, and this chapter will touch on different tactics for ensuring your analyses are set up for success.

Shaping data for use with Tableau

As mentioned in the introduction to this chapter, we have already covered the ideal way to shape data for use with Tableau, but it's worth repeating some of this critical concept. In general, it is easier to work with 'vertical' tables than with 'horizontal' tables. In most cases, each column should represent one field

with the field name in the column header, and each subsequent row should be an entry under that column header.

One common type of data that usually requires reshaping is survey data. In survey data, each column corresponds to a different question, such as “on a scale of 1 to 5 indicate the degree to which you agree or disagree with the following statement”, and each row represent the answers to every question from a single respondent. This is a recipe for disaster as the data is wide. It may be easy for humans to look at but Tableau will hate it and you will be fighting to glean any insights. Now, if you reshape, or what Tableau calls “pivot”, the data so that there are only a few columns with lots of rows, you may not like looking at the data in a spreadsheet but Tableau will be much happier.

Joining and aggregating data

As your analyses become more sophisticated, it is not uncommon to require access to data across multiple sources—and when possible, it is often best to consolidate the data into one centralized data connection. Further, as the size of your data grows, you will need to start thinking strategically about how to aggregate your data so it continues to be manageable to work with.

One of the most common questions I receive is if it’s better do this type of data preparation *in* Tableau or *before* you connect to the data with Tableau. Tableau comes with some useful data preparation functionality including the ability to do joins, blending, interpretation, reshaping, and aggregation. My general recommendation is to prepare the data as much as possible before it gets to Tableau, but these capabilities are extremely handy for trying new things on the fly before making them more permanent in the underlying data source or stream.

One big exception to pre-Tableau preparation that I would be remiss if I didn’t mention, is with calculated measures. One of the most powerful features of Tableau is its ability to do quantitative calculations on the fly across any

dimension you throw on a view. If you are aggregating your dataset before it gets to Tableau to the point where every number is predetermined, you are basically working with an OLAP, or cube, data source. This takes away a lot of the exploratory value of Tableau. I suggest having the raw calculated measure inputs in the dataset before connecting with Tableau, but creating calculated measures themselves within Tableau.

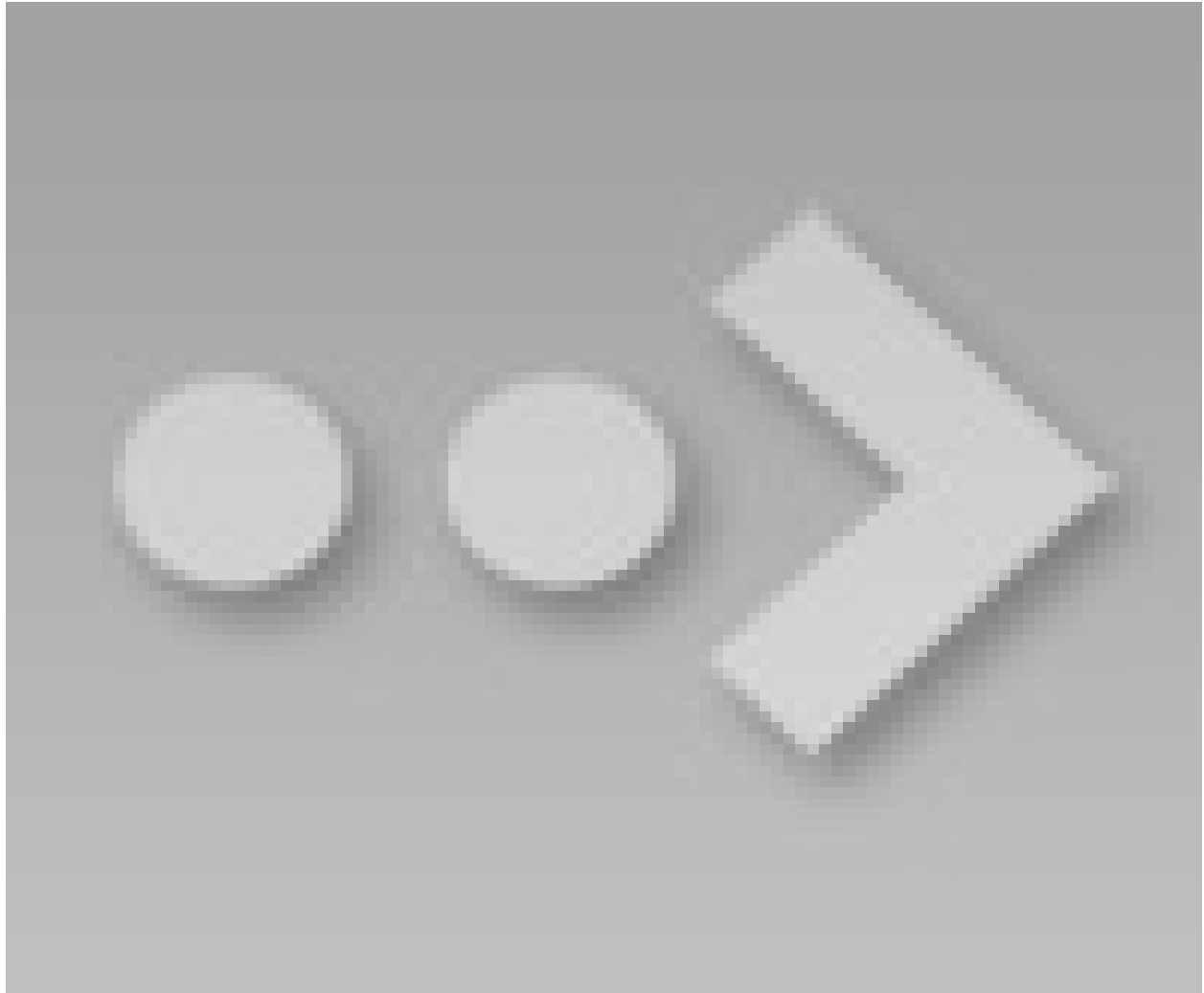
Laying out data for specific analyses

Certain chart types, such as Sankey Diagrams and [Funnel Charts](#) (link to chapter), are much easier to create if there is some thought put into how the underlying dataset is laid out. Situations like this are a case by case basis, but I do want to point out again that the dataset itself can actually help you create data visualizations more efficiently. If you come across a situation like this where a certain chart type requires a unique data format, I advise keeping a master data connection for the rest of your workbook, but creating a second data connection for these unique requirements. The second dataset may have the exact same data, but it can be laid out in a way that makes it easier to accomplish the special use case.

Shaping data for the Iron Viz example

Sticking with the “Do Old Movies Get Better With Age?” dashboard example, let’s take a look at how the ‘Shape the Data’ step was applied.

[Insert Image 21-2]



In the case of the Iron Viz competition, all three contestants were provided with a clean and easy to work with dataset. After all, this contest is meant to be a data visualization contest, and not a data preparation contest. So I admit, this dataset was a lot easier to work with compared to many of the datasets I come across in my day job. That also means that I did not literally need to transpose or reshape the dataset in order to use it. That being said, there was one notable hole in the underlying data: Age. As mentioned in the last chapter, Age was my number one KPI for my analysis, and I simply could not create the visualization I had envisioned without this field.

Rather than move on to a different concept, I evaluated the existing fields that I had to work with. Fortunately, I found that the titles of each movie included the year it was released in the text. If I could strip out the year of release from each

title, I would have a numeric field. From there, I could subtract the year of release from the current year, creating my field for Age.

So I created a calculated field to isolate year of release that looked like this:

```
Year of Release = int(mid([Main Movie Title],find([Main Movie Title],  
")+1,4))
```

This calculated field tells Tableau to look at each dimension member in the Main Movie Title dimension and find the “(“ character. The number four in the calculation tells Tableau to return the four characters following the “(“ character.

After isolating the year of release, I created another calculated field for Age:

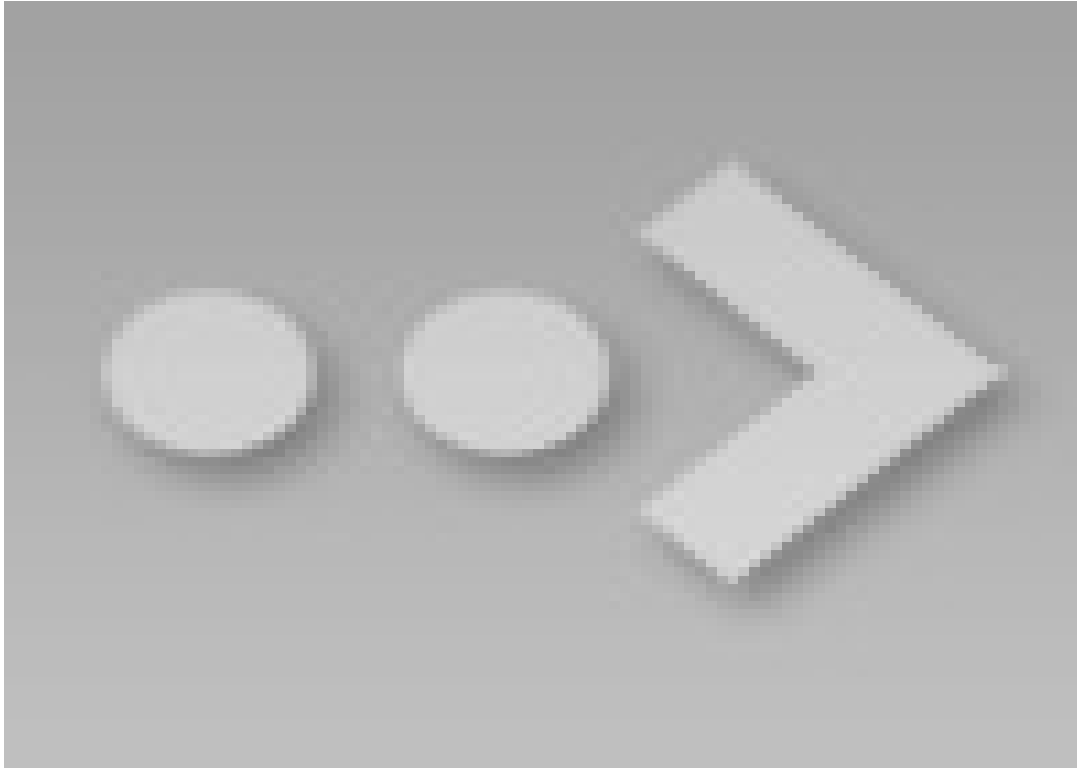
Age = Current Year – Year of Release

I was then able to use these fields as part of my analysis. This is a perfect example of creating something on the fly in Tableau by getting creative with the existing data. If this were a situation in my day job, upon proving the value of this field, I would try to make Year of Release and Age permanent dimensions in my underlying dataset before it gets to Tableau.

In the next chapter, we will finally start to visualize data!

Chapter 22 – Initial Concept

[Insert Image 22-1]



Are you ready to finally start visualizing data? I feel like Mr. Miyagi in *The Karate Kid*—making us spend months “waxing on” and “waxing off” before getting to the reason we came here. The truth of the matter is the [Identify Business Question](#), [Name KPIs](#), and [Shape Data](#) steps in the INSIGHT framework for data visualization are the foundation required to make your visualizations as effective as possible.

They also make the visualization component itself more efficient, and dare I say *fun*, because these steps help reduce frustration and potential pitfalls. The strategic thinking put in during the I and N steps inform what should be included in the dashboard and the S step makes working with the data more seamless.

After this groundwork has been laid, you're ready to create an initial concept for your data visualization.

Creating an initial concept

There is no single best way to create a data visualization. There are seemingly endless factors that can influence a dashboard concept, which was actually my motivation behind creating this framework to begin with. If I had to choose the two largest factors though, I would say they are the data visualization's (1) audience and (2) purpose. These two factors should at least provide a starting point for what your data visualization should look like. When creating an initial concept, think about these questions:

Who is my audience?

- If the audience is fellow analysts, you may be able to provide more advanced chart types (i.e. Box and Whisker Plots, Pareto Charts, etc.)
- If the visualization is meant for a mass audience, consider simplifying the concept both in terms of layout and chart selections to make your story clearer
- If the visualization is being created specifically for a C-level audience, more weight should probably be put on boiling explicit insights to the top before diving into any in-depth prescriptive analysis

See the chapter, Know Your Audience, for more on this topic.

What is the purpose of the data visualization?

We have covered this in the Identify Business Question step, as well as put some thought into answering the question in the Name the KPIs step. During the initial concept step, you have a chance to prioritize the KPIs in order to communicate the answer in a way that is intuitive to the end users.

See the chapter, Use the Golden Ratio, for more on prioritizing and laying out dashboard content.

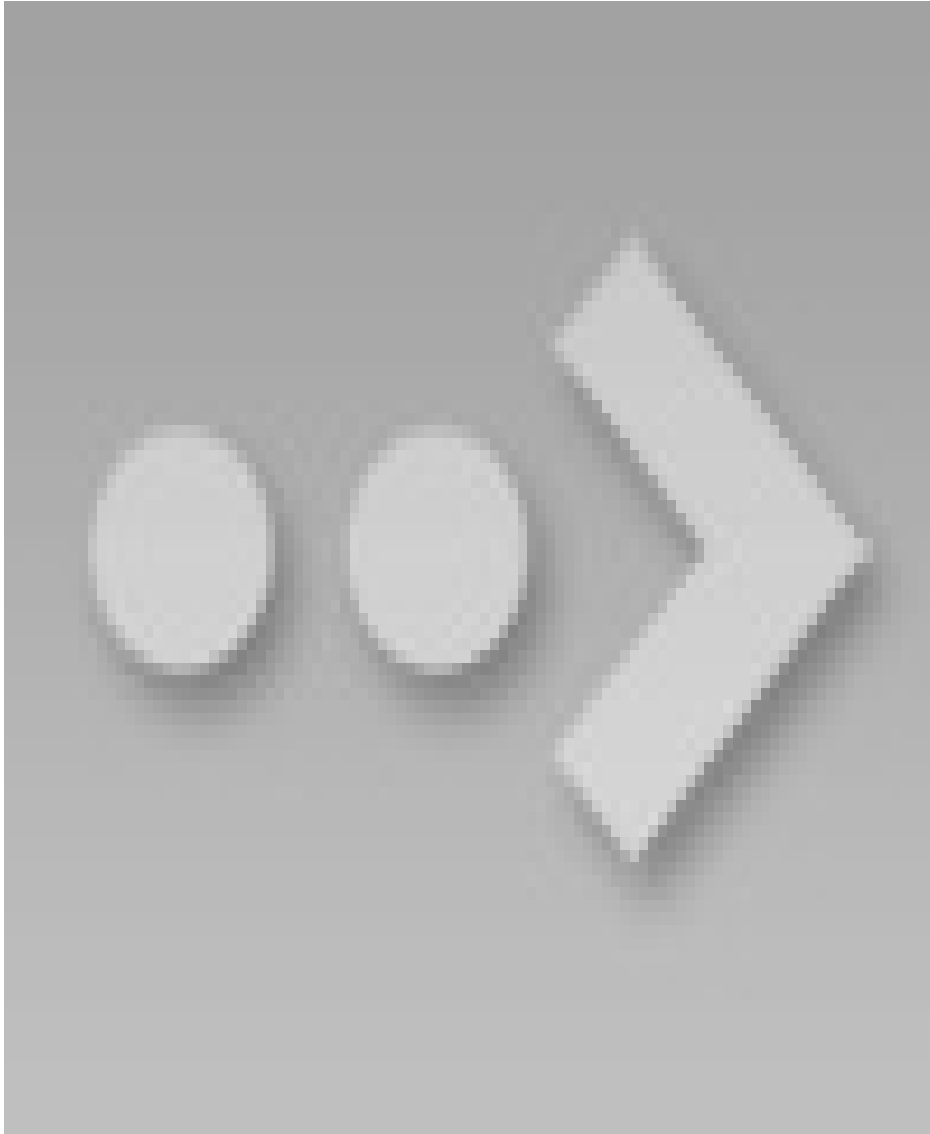
With the answers to these two questions in mind, I create an initial concept in one of two ways:

1. A hand-drawn sketch

That's right – I will literally draw out a concept. This has become my preferred method of creating an initial concept because it is extremely efficient and since I am not investing much time building an idea out, I do not get frustrated when I inevitably have to make a change.

I cannot draw, but that is okay! The outline, prioritization, and chart selections will come through and it will make it much easier to pivot if and when you receive feedback. Here's a real-life example from a project with one of my data visualization partners:

[Insert Image 22-2]



2. Exploratory

In some cases, there are certain dashboard elements that I know will be included in my final deliverable. If so, I don't mind spending some time building these out and will experiment with sizing, layout, and flow on a dashboard "canvas". When I take this approach, I usually at least have an idea of how the dashboard will be distributed, which will inform the appropriate canvas size (i.e. optimized for mobile, optimized for printing, etc.). Once I have a blank canvas in my data

visualization software, I can begin placing elements to create an initial concept before requesting critiques.

No matter which approach to creating an initial concept you choose, the key is to be open to change at this point in the process. I recommend not investing a great deal of your effort during this step because *your concept will change* – which is discussed in the next step - and you will be happy you saved your effort for later in the framework.

Chapter 23 – Gather Feedback

[Insert Image 23-1]



As mentioned above, there is no one best way to create a data visualization and there are almost infinite factors that can influence a dashboard concept. I have always said that data visualization is a form of art, and with that, ‘beauty is in the eye of the beholder.’ One anecdote that I enjoy sharing is that regardless of the Tableau contests I’ve won, and the popular publications that have picked up my work, and the *hundreds* of times I have created data visualizations...

I have yet to build a dashboard that every single viewer thought was perfect.

And that’s okay! I understand that it is impossible to please everyone all the time. Among a diverse audience, different people will have different ideas of what the business questions are, opinions on which KPIs should be used to answer those business questions, varying levels of knowledge on data visualization best practices, personal design preferences, etc., *etc.*

Gathering feedback should be an expected – and welcome - step in your data visualization process. Gathering feedback on your initial concept from the audience that will be using it provides at least three major benefits:

Improved Efficiency – By taking time to gather feedback, you are able to get to the final product faster. When you dedicate a step to gathering feedback in an organized fashion, you are much less likely to receive sporadic, overly-detailed feedback after you bring the concept to life.

Reduced Frustration – In my personal experience, when I know that incorporating audience feedback is part of the design process, it makes it less frustrating when somebody requests that we move in a different direction.

I suggested in the last chapter that you shouldn't invest a lot of time on the initial concept. It is easier to hear criticism or varying opinions when you haven't spent time actually building out a vision before gathering feedback.

Ownership – I have found that when you proactively gather feedback from end users, they take ownership in your final dashboard. Things tend to go smoother when the audience has some stake in your work. I don't have any science behind this, but I think subconsciously, people are less likely to criticize something if they had some say in its creation.

How you gather feedback is up to you and is largely dependent on your audience. For example, if you are building a dashboard for a small audience of one to three, you may simply send them an image of your initial concept and ask them to respond with their thoughts. If the audience is slightly larger, you may opt to set up an hour-long brainstorming meeting to think through different approaches as a group.

When designing for a mass audience, consider sending the initial concept to two or three individuals in your personal network to get a sample of how the overall population may look at your visualization. The key with your “preview” audience is that you choose individuals with diverse backgrounds that better reflect a larger, and diverse, public audience.

Even if you are creating a visualization for yourself, I encourage you to at least sleep on your initial concept. By stepping away from the design process, you can determine if you are still happy with the approach or have some ideas for improvement.

In the next chapter, I will share how to incorporate the feedback that you’ve gathered and hone the dashboard.

Chapter 24 – Hone Dashboard

[Insert Image 24-1]



After gathering feedback from your stakeholders, you are ready to finalize your data visualization. I call this step “hone the dashboard” because you are refining your initial concept based on the audience input provided in the last step. At this stage in the INSIGHT framework, the business question (or purpose of the data visualization), the KPIs that will be used to answer that business question, and the dataset that contains those KPIs, should all be in a solid – if not final – state.

The task that is left to complete is investing the time to actually build out the components of the dashboard and lay them out. Depending on how much time you invested in the [Initial Concept](#) step, you may have quite a bit of effort left to get to a working product. In fact, I recommended that in most situations, the initial concept should have very few working components or even none (in the case that your initial concept was a hand-drawn sketch). While this step leaves quite a bit of work to bring your concept to life, it is much more efficient because you only have to build the dashboard elements once (versus building everything only to have the audience make you move in a completely different direction).

While there is no one best way to create a dashboard, here are a few general rules I have in the back of my mind when trying to finalize a view:

- Dashboard dimensions should be determined by the method of distribution. For example, if you know the dashboard will be printed out or attached to an email, consider making the dashboard the same size as an 8 ½” by 11” piece of paper. If you know the dashboard will be consumed on mobile devices, make the dashboard the same size as a tablet or mobile phone. If you can’t decide, the Golden Ratio is a good way to get started.
- Keep dashboard elements to 12 components or fewer. This is a rule I learned from Google Analytics’ limits on custom dashboards. By capping the number of dashboard elements, you keep the design focused and help the story in your data emerge more efficiently and effectively. If you find yourself requiring more than 12 components, consider breaking the dashboard up into multiple views or even multiple business questions. Occasionally I will start with a descriptive overview, but use interior pages to provide more prescriptive analyses.
- *Keep It Simple* in terms of chart selections, user experience, and design. Being clear and concise is another best-practice method for telling your data-driven story, which we will discuss in the next and final step of the INSIGHT framework.

I will close this chapter by addressing one of the biggest questions in any design process: *What do you do if the audience feedback is conflicting or outside of data visualization best practices?* Remember, part of honing the dashboard is the opportunity to incorporate the stakeholder feedback, and there are several benefits to doing so which were [outlined in the last step](#). So what should you do if the audience disagrees with you or each other? Or worse, they want to create a dashboard with 15 pie charts on it?

Ultimately, you are the final gatekeeper of the dashboard and need to prioritize which feedback makes sense and how it fits together. The designing outside of

best practices question is one I have struggled with because I really do find it hard to put my name on views that are just a huge spreadsheet or include charts such as pies or packed bubbles. My solution is to provide the views in both ways: (1) the way that was requested and (2) the way that I recommend. This way everybody gets what they want, and I at least have the opportunity to share the best practice approach to a specific situation. Sometimes my advice is taken; sometimes it's not. But by providing both approaches, I don't lose any sleep because I know I tried to help.

Honing the dashboard can be an iterative step itself, so don't worry if you don't get it perfect on the very first try. Following the steps in the INSIGHT framework to this point will help make this step as efficient as possible, and you soon will be ready to distribute your work.

Chapter 25 – Tell the Story

[Insert Image 25-1]



Once you have honed your dashboard and possibly iterated a couple of times to land on a final product, you are ready to distribute your data visualization. I call this step, “Tell the Story”, because you have hopefully integrated some storytelling tactics into your dashboard and your data has a compelling story to share. Further, I believe that the entire *purpose* of data visualization is to find and share actionable stories that are based in quantitative evidence. If your dashboard does not provide insights that can help inspire action in your organization, it’s probably not worth sharing to begin with.

No, there’s not always an earth-shattering realization that comes from every single weekly report you may create. That being said, if you have put some strategic thought into identifying the business question that your dashboard is answering and how the KPIs you’re measuring will answer that question, you drastically increase the chances of your dashboard providing actionable insight.

So assuming you’ve followed the INSIGHT framework to this point and your

So assuming you've followed the INSIGHT framework to this point and your dashboard is sure to provide valuable insights, your method of distribution once again comes back to my largest factor when creating a data visualization: the audience. In my experience, the audiences for my dashboards are almost as diverse as the factors that go into creating the dashboard itself. They truly live on any and all points of a spectrum in terms of both their analytical sophistication and their technical infrastructure.

For example, I have created dashboards for audiences that have never heard of Tableau. With the analytics partner that I have in mind, I refreshed the dashboard data once per month and saved the dashboard as a PDF that I attached to an email. On the other end of that spectrum, I've worked with several partners that have a very robust Tableau Server implementation in place where I can post the dashboards and set them up to automatically update. This makes it a little easier to tell the story because users can subscribe for periodic email updates, and starting with Tableau 10, you can even subscribe other people to receive your dashboards with the most up-to-date data available.

Regardless of your audience, there are a few specific tactics you can use to help tell the story that emerges in your dashboard:

- Context – You won't always be around to explain your dashboard to end users. Be sure to provide some context for your dashboard so your end users have an idea of what they should be looking for as they navigate your work. This context can be as simple as a title and subtitle on the dashboard itself, but could include additional information such as the source of the data or the date range being applied to the visualizations.
- Inline Insights – One of the most effective methods of providing findings and recommendations for next steps is to explicitly write them out within the dashboard. One way to do this is to design in a text box for "inline insights" that the developer or analyst can modify with their own thoughts. Some insights can even be automated through the use of calculated fields that will display a sentence based on the performance of the data.
- Annotations – If a dip on a line graph, for example, is influenced by an

event that is not readily apparent in the dashboard, add an annotation to the line graph to provide additional context. In just one hypothetical example, let's pretend that the line graph is looking at daily production. If you know the factory was closed for two days due to unforeseen circumstances, but this is not data that is normally tracked or reported to the audience, help provide this context by annotating the graph.

I'll close the INSIGHT framework for data visualization section by using the *Do Old Movies Get Better with Age?* example – the winning visualization in the 2013 Tableau Iron Viz Championship - one more time. The three Iron Viz finalists each year are among the best of the best Tableau users in the world. While I pride myself on balancing data and design, I considered one of my competitors more skilled at design than myself, and the other competitor more skilled at data than myself. While I think my balance between the two was a large factor, I think the primary reason I was able to win was *storytelling*.

At the end of the Iron Viz contest, each contestant gets to explain their data visualization. While the other two contestants used their time to walk through the technical aspects and user experience of their dashboards, I literally told the story behind what motivated my approach. I believe this helped me connect with the judges and give purpose to my data visualization which gave me the edge needed to pull off the win. In a corporate environment, the judges from the Iron Viz are your audience, and pulling off the win equates to finding and sharing a valuable insight that causes positive change for your business.

The next section of *Practical Tableau* will discuss more storytelling tactics you can integrate into your dashboard development process.

Chapter 26 – Introduction to Storytelling

So how exactly do you tell a data-driven ‘story’? I fear that storytelling in the context of data visualization has become a bit of a buzzword. What’s worse is the concept is often mentioned without practical examples of how you can apply storytelling techniques to data visualization. This section offers an introduction to storytelling and aims at providing several tangible tips for incorporating storytelling techniques into your data visualization. If the INSIGHT strategy section is the key to taking your work from good to great; storytelling is the key to taking your data visualization from great to *actionable*: the ‘Holy Grail’ of the profession.

Storytelling has been used in every culture as a means of engaging audiences and communicating everything from entertainment to moral values. Stories work because humans are wired to retain stories; not independent facts or statistics. In fact, after a presentation, 63% of attendees remember the stories presented, while just 5% of attendees remember the individual statistics. (Heath, 2007) This is particularly relevant to data, where business insights risk being lost amidst a deluge of increasingly large sets of numbers. Think about what you remember after seeing an effective data visualization compared to a raw spreadsheet of numbers. The difference is inherent storytelling at work.

The parallels between data visualization and storytelling are undeniable. Much like stories, data storytelling includes three critical elements: (1) Characters, (2) Plot or Storylines, and (3) Narrative. With data, KPIs and contextual metrics can be considered the characters, storylines can be considered the insights, , and the narrative is data visualization – or the style in which the insights are being communicated.

Data visualization practitioners have a lot of say with all three elements of data storytelling. (1) Your strategy informs the KPIs that are being used to answer the business questions at hand. (2) Your ability as an analyst helps unearth the insights from the data. (3) The style in which you communicate those insights goes a long way in determining if your visualization will be actionable and cause a change.

Interestingly, not to mention amusingly, data storylines tend to follow another parallel, *The Seven Basic Plots* from Christopher Booker:

1. 1. Overcoming the Monster; *the protagonist (us), use the power of data insight to make a recommendation that overcomes an obstacle to the business.*
2. 2. Rags to Riches; we leverage data visualization to add value and maximize our company's ROI.
3. 3. The Quest; we have a business question or know something is impacting the business, but must use data visualization to locate the answer.
4. 4. Voyage and Return; we travel to lengths only data visualization can bring us, returning heroically with business-changing insights.
5. 5. Comedy; it really can be funny when data visualization reveals an insight that was seemingly under our nose for quite some time, but was disguised as a wall of numbers in our monthly Excel report.
6. 6. Tragedy; unfortunately, this may be the most common storyline. At least data visualization helps us find the 'tragedy' as efficiently as possible and, when done well, helps us know what to do about it.
7. 7. Rebirth; this one is my favorite. This is when the main character realizes the error of their ways, and uses the power of data to become a better employee.

As our strategy generally informs the 'characters' in our analyses and the plot is one of the seven just outlined, these tips will focus on narrative, or style of communication, to maximize the effectiveness of two types of stories:

1. 1. You, the visualization creator, know the outcome to the story. In this case, it is your responsibility to communicate the insight as effectively as possible so that action is taken.
2. 2. You make the end user part of the story. This happens when you include interactivity that allows the user to find storylines about themselves or something they care about.

References

Heath, C. and Heath, D. (2007). *Made to Stick: Why Some Ideas Survive and Others Die*. New York: Random House

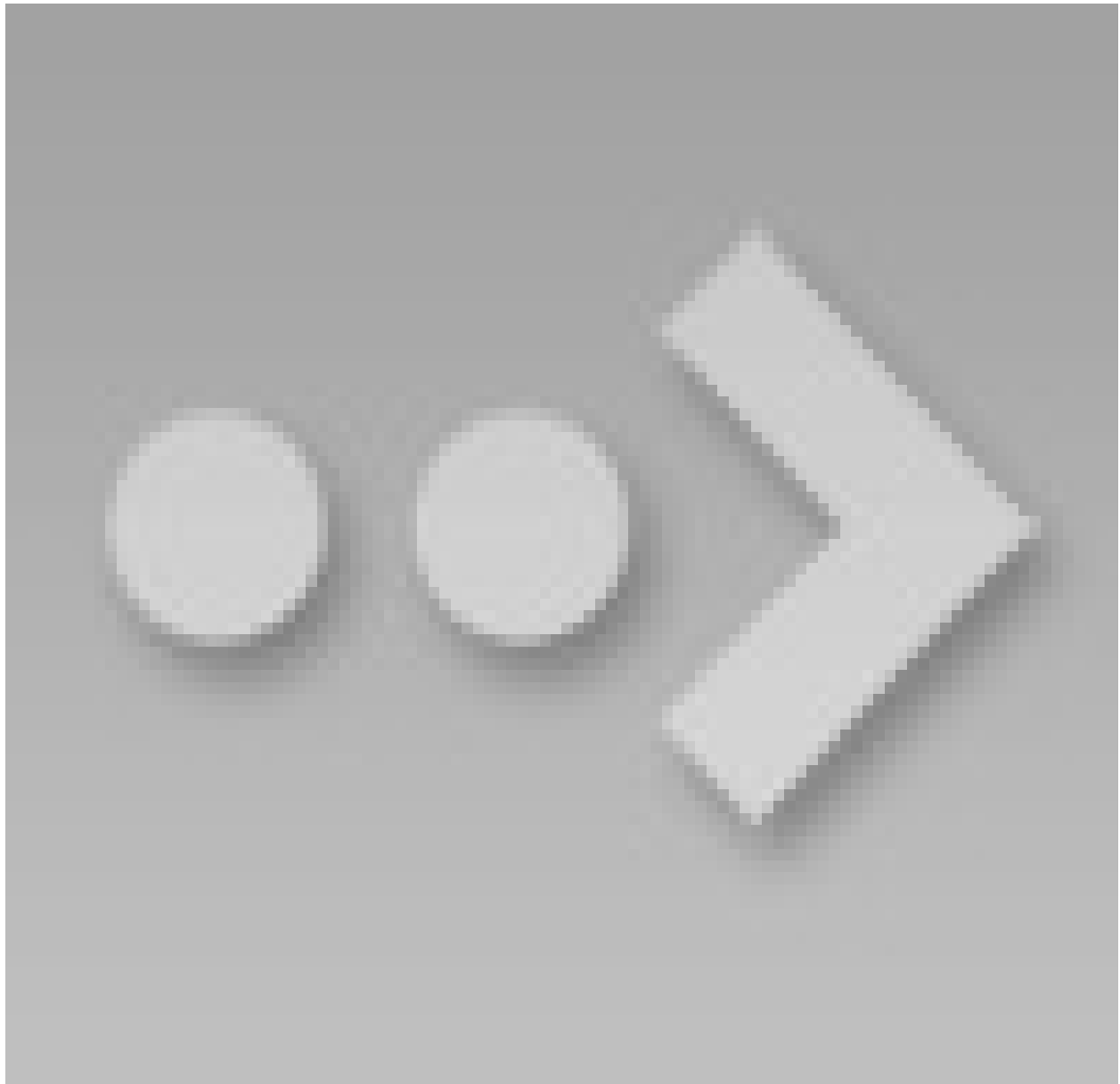
Booker, C. (2004). *The Seven Basic Plots: Why We Tell Stories*. London: Continuum

Chapter 27 – A Data Visualization Competition—that’s also an analogy for the data visualization process

Before getting into the tips, I want to use the Iron Viz competition one more time to share a bit more about how I think about data visualization. If you have ever seen me present on the topic of visual analytics, you have heard me use Tableau’s Iron Viz Championship as an analogy for good data visualization. The ‘Iron Viz’ is a competition held annually at Tableau’s conference at various locales throughout the United States. It is the culmination of a year’s worth of Tableau users using the latest in Tableau product developments to create innovative Tableau Public dashboards. There is only one way to receive an invite to compete in the Iron Viz, and that is to win one of three Tableau Public feeder contests held throughout the year leading up to the annual conference. During the contest, the three ‘Iron Vizzers’ compete live in front of several thousand conference attendees. Each contestant is given the same dataset, and they have exactly 20 minutes to create the best data visualization they can before being judged by the audience, the previous year’s winner, and a panel of Tableau product experts.

It's quite a spectacle—and a lot of fun. As the contest begins, each competitor walks to the stage through dim lighting and heavy fog. Dramatic music is blaring. The competitors even don chef-like jackets - a nod to the contest's format borrowed from the Iron Chef television show. After some announcements and introductions from two MC's, the contest is played out on huge projection screens so the audience can watch every competitor's move in real-time.

[Insert Image 27-1]



The reason I love this event as an analogy for data visualization is because it is a perfect balance of sizzle and steak – both at their highest levels. The presentation aspect is needed to get the largest audience possible interested in the event and make it an entertaining experience from start to finish. It truly transforms the type of work I do every day into must-see theatre. It guarantees that the event will be (literally) remarkable, enticing the audience to share on social media, write blog posts about it when they return home, and of course, come back in the future for more. At the same time, the three contestants have been thoroughly vetted and are among the best in the world at what they do. The visualizations they are about to pull together promise to be top quality, and are the core of the event. As with data visualization, the data, analysis, and the designer's ability to communicate that analysis, are the heart of the work. These key elements are strong enough to stand on their own, but the Iron Viz maximizes their effectiveness by helping make them memorable and shareable.

I argue that the best data visualizations are those that not only incorporate data visualization and analysis best practices, but those that also possess a sense of storytelling – whether that be through usability, graphic design, inline insights, or innovative visual approaches that help make the message of the dashboard stick.

My goal in this Data-Driven Storytelling section is to provide actionable tips that will help you round out your existing corporate dashboards in a way that maximizes their effectiveness in your workplace. I'll refer back to my experience at the Iron Viz championship because, in a very compressed time frame (20 minutes!) I went through the process of storytelling—in many jobs you'll go through these steps slower, or in a different order.

Chapter 28 – Tip 1: Know your Audience

There are not many tactics that are going to improve the chances of your data visualization making a difference around the workplace more than tip number one: *Know Your Audience*. Before you create a single chart that you intend to share, putting some thought into who will see that chart will dramatically improve your data visualization.

My favorite analogy for this concept is the old Southwest Airlines *Wanna Get Away?* campaign. In one of the commercials, a rapper performing at a concert mistakenly and passionately thanks the audience by saying “Thank you Detroit – we love you!” The crowd begins to boo and another performer informs the speaker that “Detroit was last night.” I have actually been at several concerts where the artist on stage thanks the home crowd by name. Even though the performer surely thanks every stop’s hometown in the same way, hearing your own city’s name provokes a great sense of pride and is usually an easy way to make a connection with the audience. Get it wrong, however, and the city shout-out has an equally, if not more, negative impact on the crowd.

This same principle can be applied to your own corporate dashboards. Knowing your audience goes a long way to making a connection and maximizes the chances that your end user will understand and happily adopt the reports that you have created. Get it wrong, and you risk permanently damaging the chances of getting your visualizations off the ground around the office. Think about this the next time you undertake a visualization project. Here are just a few examples:

- If your dashboard is intended for a C-level audience, keep your work simple and direct. Focus on KPIs and the progression toward goals. You may also consider creating dashboards that are optimal for being saved as PDFs, to improve the chances that your work is either attached to an email or printed out and handed to a C-level executive.
- If your work is intended for fellow analysts, build in interactivity that

allows them to find their own stories in the data. In Tableau, quick filters, dashboard actions, and parameters are my go-to tools for achieving this.

- If you are using Tableau Public and attempting to make your data visualization connect with a mass audience, don't be afraid to use freeform dashboards, story points, and even incorporate some graphic design elements. Believe it or not, some people think of data as dry! Leveraging some of these tactics will make your visualizations much more shareable.

Data visualization is not a one-size-fits-all practice. Regardless of the storyline you have to share, knowing your audience will help you prioritize and make the most of the data-driven storytelling tips to follow.

Chapter 29 – Tip 2: Smooth the Excel Transition

Tableau is not Excel. Excel is not Tableau. One of the most common barriers to Tableau adoption is the belief that similar data visualizations can be created more easily in Excel. This may be true for certain situations, but if your goal is to master the art of data-driven storytelling, you must leverage the strengths of each software individually. I do not use Tableau to create and store my datasets. While it's possible, the thought alone makes me cringe with frustration. That is not what Tableau is best at. On the other hand, I don't attempt to use Excel to make beautiful, interactive dashboards. I'm sure with some elbow grease, this could be achieved, but the one million row limit in Excel alone makes this solution impossible in the enterprise-level visual analytics projects I work on. That is not what Excel was designed to do.

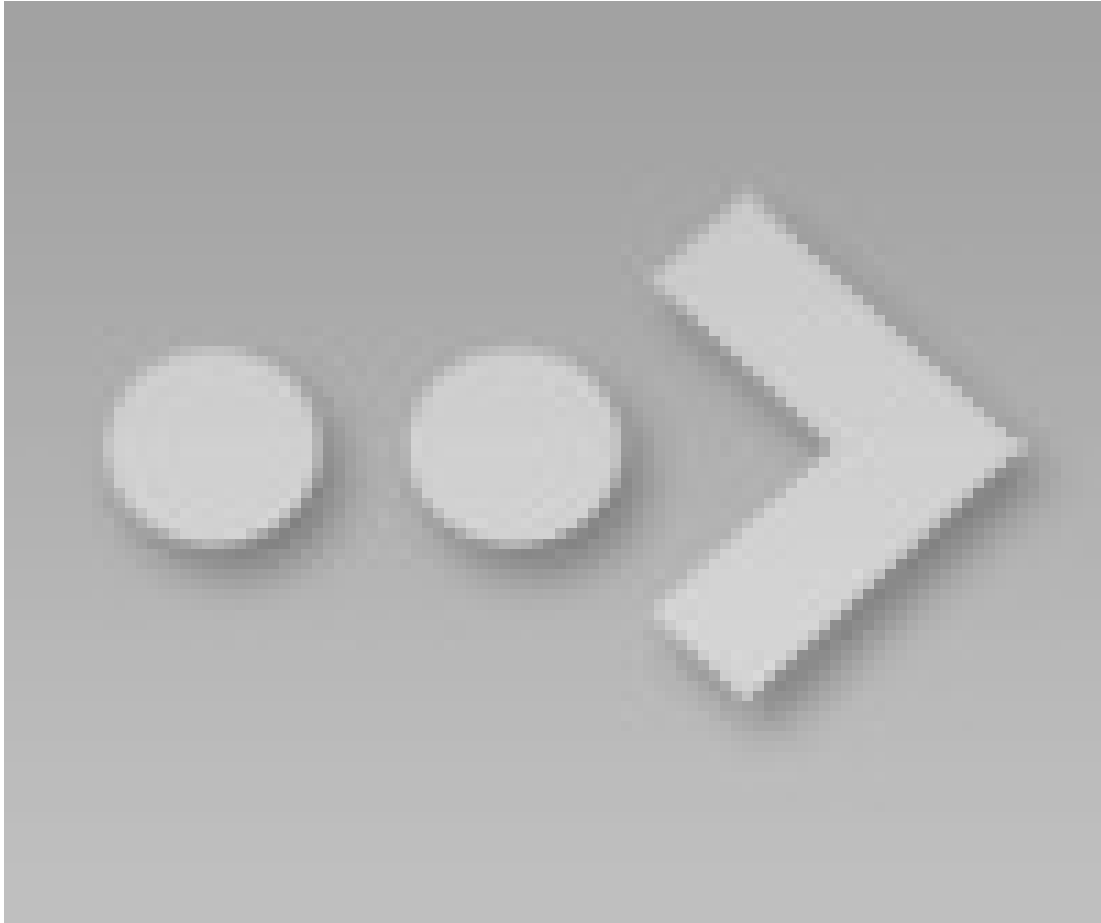
And that is okay!

I love both of these programs and use them almost every day, but for different reasons. If Tableau's goal was to replace Excel, Excel would not be one of the primary connection types available in their software. Tableau Personal Desktop users, who are restricted to using flat data files, rely heavily on the ability to interact with their Excel files in Tableau. Tableau knows that it is not Excel.

Nevertheless, the Tableau vs. Excel debate is one I encounter regularly. My recommendation for anybody experiencing the same pushback to leveraging the data visualization capability of Tableau, or moving from crosstab views to visual analytics, in general, is to start with highlight tables. The following exercise can be used to help illustrate the power of applying even a very simple visualization to your data. You'll find that even with the most basic visualization, you'll transition your forgettable spreadsheet to a remarkable data story.

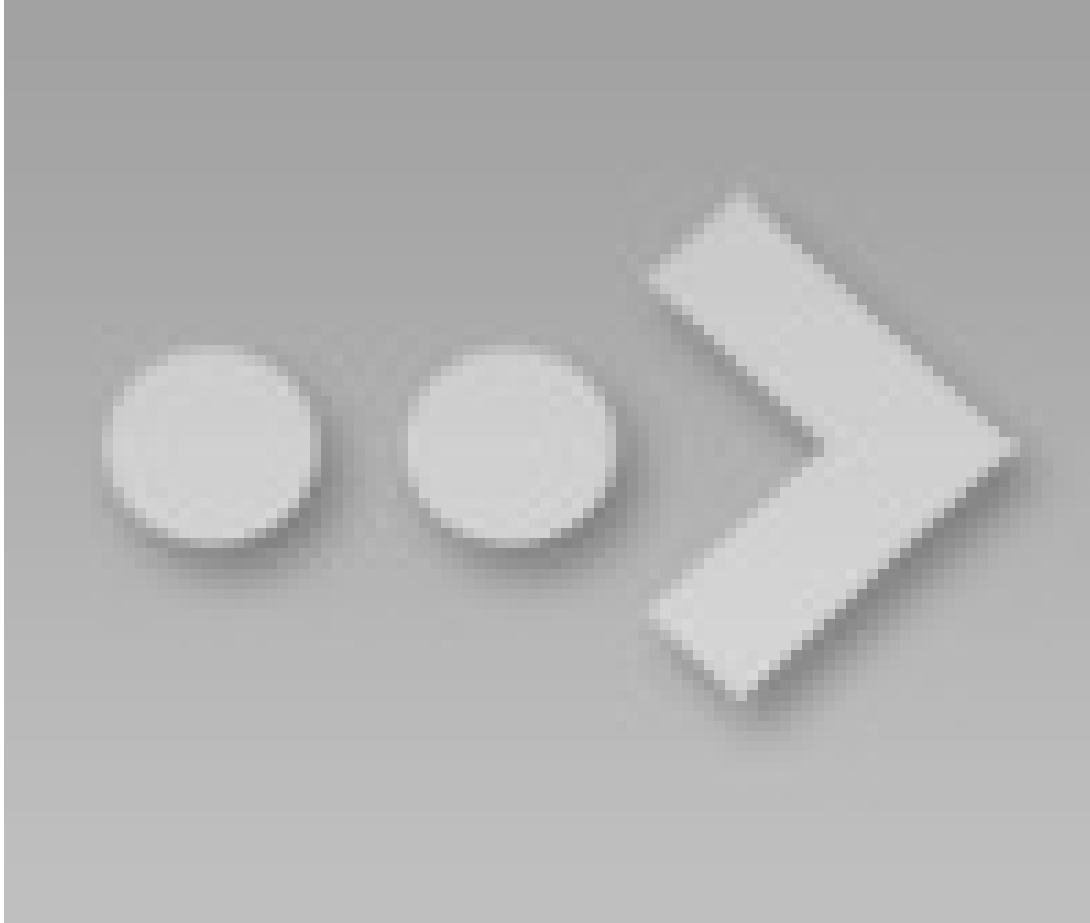
First, take a look at the image below, which is what an Excel spreadsheet looks like with no conditional formatting. See how many seconds (minutes, maybe?) it takes you to determine the top three values in the table:

[Insert Image 29-1]



Next, take a look at the same data, encoded by color. In this case, the higher the sales, the darker the green. Now count how many seconds it takes you identify the three highest values:

[Insert Image 29-2]



Much less time, right? This is the power of data visualization. The above image is called a highlight table, and as you can see, even the simplest forms of data visualization can lead to much shorter time to insight. You can easily create highlight tables in Excel or Tableau, and I have found they are a great way to introduce the power of data visualization. In the tips to follow, I will introduce more complex data visualizations and data-driven storytelling techniques that can be integrated more efficiently in Tableau, but many times, you have to start by helping your audience understand why data visualization plays an important role in analytics.

Chapter 30 – Tip 3: Leverage Color

“I found I could say things with color and shapes that I couldn't say any other way - things I had no words for.” - Georgia O'Keeffe

During his keynote presentation at the [2014 Tableau Conference](#), Christian Chabot (Tableau co-founder and former CEO) talked about data analysis being a *creative* process. As an analyst, you may not think of yourself as an artist, but by its nature, data visualization is an art form. Visual analysts use data to express insights and provoke action. As Chabot put it, “Analysts and artists are both on a mission to reveal something new — to discover truth, to find meaning.” So if you are doing visual analytics, congratulations – you’re an artist! If that label makes you feel slightly uneasy, don’t worry, this chapter covers several tips on utilizing one of the most powerful forms of artistic expression: Color.

While leveraging color is relatively easy to put into practice, it is also one of the most effective tools for discovering and sharing insights.

A few benefits of leveraging color in your data visualization

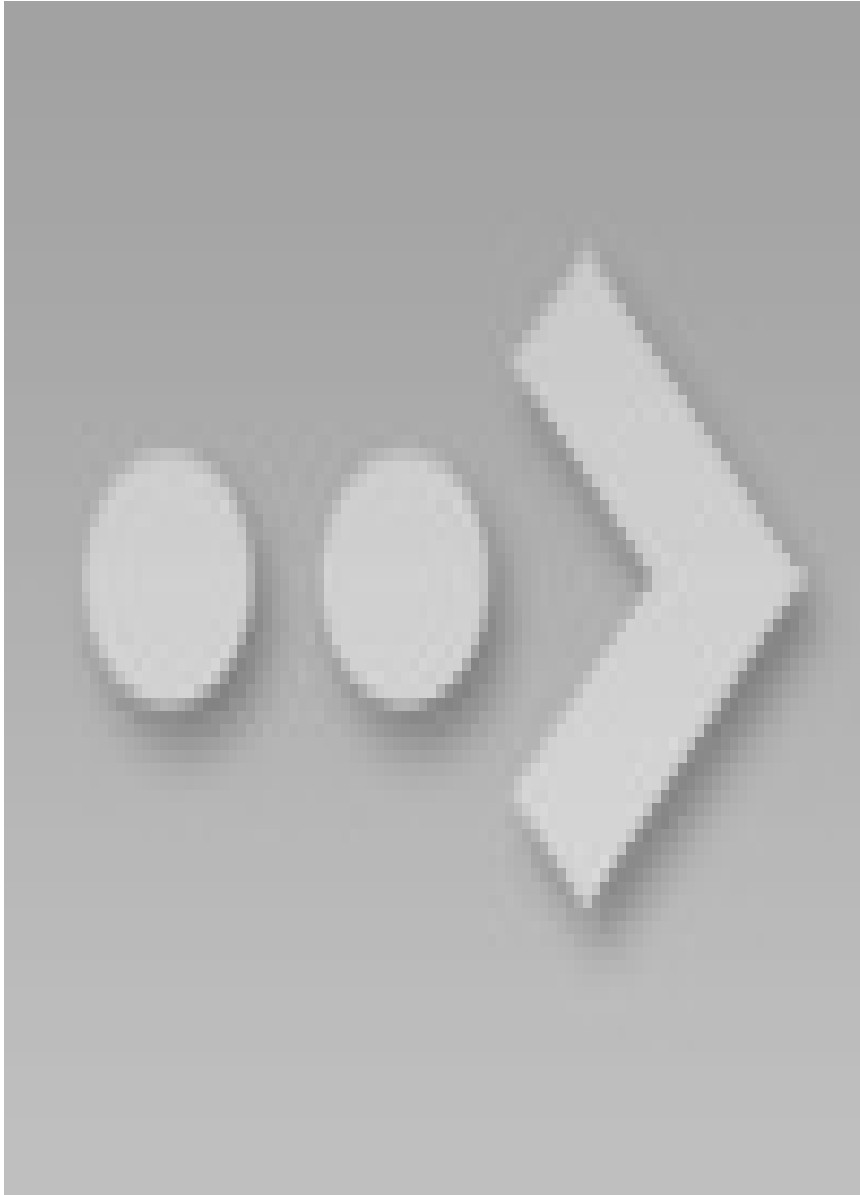
1. Color makes the stories in your data pop. The primary use of color is a practical one: it helps the insights in your data emerge, both for you and your audience. Color helps accomplish a shorter time to insight by providing (a) a means to identify discrete dimensions so you can quickly recognize strong and poor performers (i.e. Each categorical value can have a different color such as West is colored orange, and East is colored blue) and (b) a scale to illustrate relative performance (i.e. A sequential palette that shows Sales turning a darker green as they increase or a divergent palette which shows positive sales as blue and negative sales as orange.)

1. 2. Color helps engage your audience. Careful color selections, or even the use of color at all, is an easy way to subliminally capture and keep the attention of your visualization users. One concept that I always have in mind when designing data visualizations is Seth Godin's idea of [being 'remark'-able](#). Being remarkable means that your work is so unique, interesting, insightful, or awesome enough in some way, that your audience will want to make a remark about it. In today's world, that may mean sharing it on social media, or passing your report up the corporate ladder. I have found that the use of color is often a key ingredient in achieving remark-ability in data visualization.
1. 3. Color provides an opportunity for you to show your narrative style. Perhaps your company's brand colors work well as primary or secondary palettes (this is not always the case, so please consider with caution!). Maybe you want to use color to make an emotional alignment with the story the visualization is sharing (i.e. a darker look and feel when addressing a darker topic).

Customizing your use of color is easy with Tableau

Below is one colorful example from my Tableau Public portfolio. Remember, if you ever see something you like in a Tableau Public viz, you can download it to find out how it was created. The same is true with colors – if you see a color you like, you can download the workbook to find the values that generated the color. More on color values below.

[Insert Image 30-1]

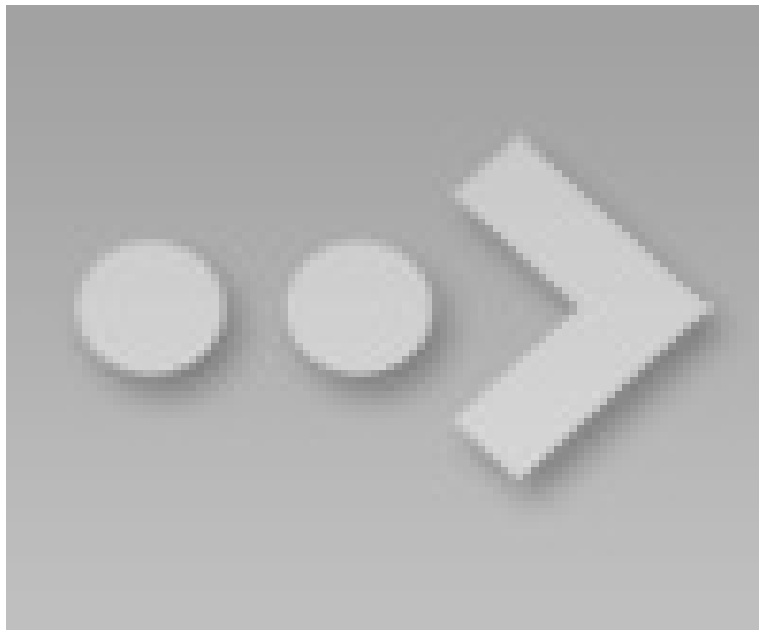


This visualization uses the out-of-the-box “Temperature Diverging” color palette in Tableau. Even though it is typically not best practice to use red and green due to a common color-blindness that impacts around one out of ten men, I couldn’t resist using these colors for a viz specifically about traffic. Fortunately, the particular hues of red and green are actually color-blind friendly. The only downside was that using the color palette in Tableau was not as flexible as I would like; this certain color palette is only available for use when coloring continuous measures (i.e. lower values colored red; higher values colored green). Fortunately, Tableau allows you to customize colors and how you can use them

if you know the values used to generate them.

There are three types of color values that you can use to customize colors in Tableau. The quickest way to customize colors in Tableau is to know either the (1) Hue-Saturation-Luminance or (2) Red-Green-Blue values. Both HSL and RGB use a combination of three, three-digit numerical codes. You only need to know one of combinations because when you use one, the other will automatically be generated. For example, if you type in the RGB values of a color, the corresponding HSL values will be generated. Any color can be added and used in Tableau by double-clicking a color in Tableau color legends until you see a color dialog box and enter the HSL or RGB values. The custom color dialog box looks like this on a PC:

[Insert Image 30-2]



Here are the RGB values for Tableau's temperature diverging color palette if you would like to use them in your own data visualization. The first value is the level of red, the second is green, and third is blue:

Green: 82.153.133

Blue: 210.207.71

Yellow: 219.207.1

Red: 194.107.81

Orange: 239.180.78

The third type of color value that can be used with Tableau is called hex, which is a six-digit numerical code. Note that all colors have all three types of values (among others): HSL, RGB, and hex. Hex values can be used to create a variety of [custom color palettes in Tableau](#). You can use a tool like [ColorSchemer](#) to find the color values you need.

Tips for getting the most from color in your data visualization

1. 1. Use limited color palettes. Ideally, two-color schemes are the most effective. These types of visualizations make the story in your data very clear and don't require the use of a color legend. Sometimes, the volume of dimensions dictates how many colors you need to use, but I generally try to keep the colors on any dashboard to five or fewer. Just keep in mind that the more colors you add, the more your dashboard user has to work to look back and forth to the color legend to understand what they are looking at.
1. 2. Use muted colors. Muted colors are less intense due to a lower hue saturation. Muted colors as shown in the example above are more pleasing on the eye and make it easier for your audience to consume your visualizations. Try to avoid harsh *sharp* bright colors which distract from your data rather than add to it.

There are several color-related topics that I have not touched on here including how the color wheel works, selecting color schemes, and the psychology of

color. For more on color, see [Link to chapter on using color in the Tips & Tricks section]

Chapter 31 – Tip 4: Keep it Simple

“I didn't have time to write a short letter, so I wrote a long one instead.” – Mark Twain

I love this quote from Mark Twain because the author is putting a value on prioritizing content. It is said that Twain's “complete” bibliography remains incomplete due to the volume of his writings, and the fact that they were often completed for obscure publishers – not to mention under a variety of pen names. However, even as one of the most prolific writers of all time, the quote above implies that Twain believed the most effective storytelling was done by being clear and concise.

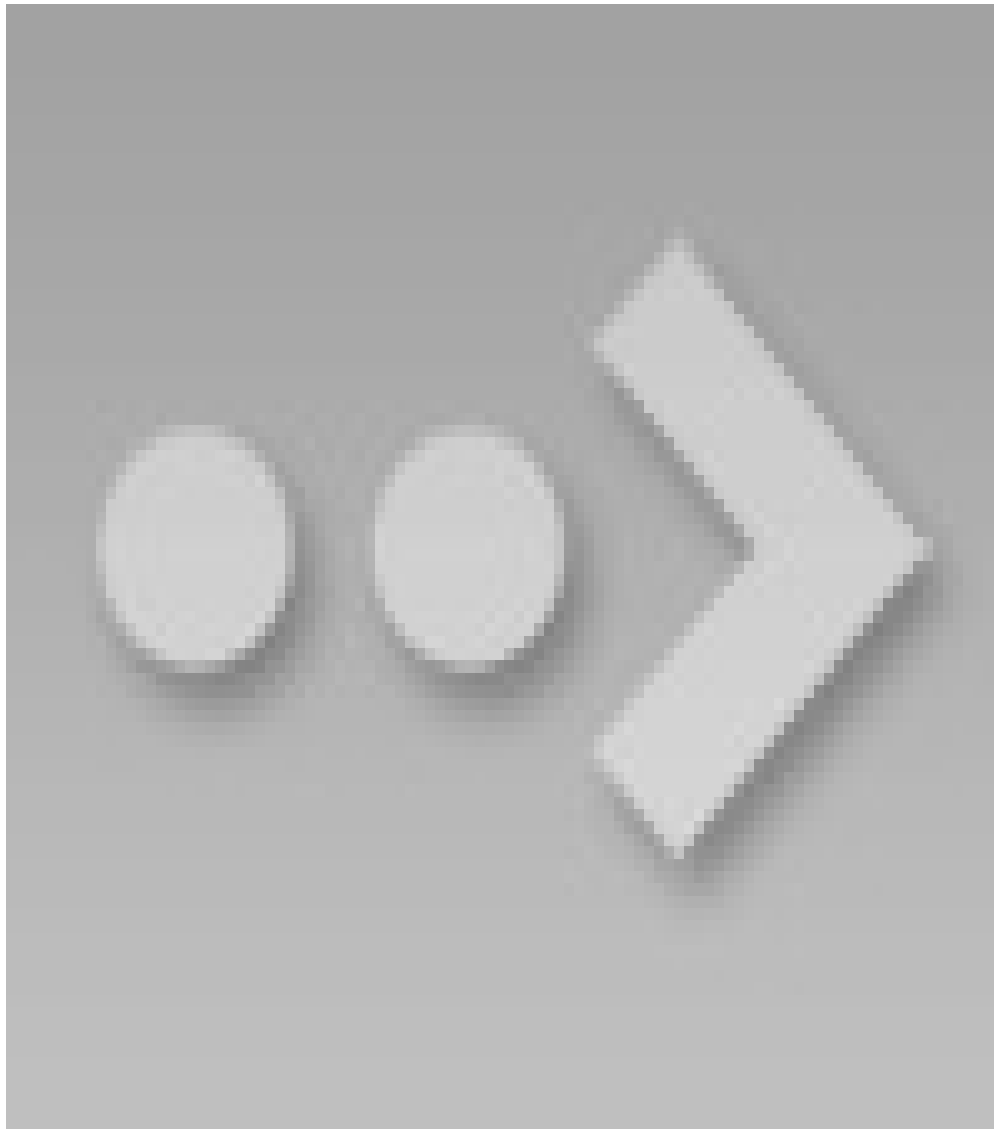
This same idea applies to data visualization: Keep It Simple. One of the most common mistakes I see in dashboard layout and design is attempting to create silver bullet dashboards that provide every possible answer to the business question at hand – all in a single view. One of the things I find myself saying often is ‘just because it is possible in Tableau does not mean you should do it.’ Tableau makes it extremely easy to add filters, charts, and widgets to a dashboard, but there is a point when too many options for the end user actually detracts from your visualization, making it harder for the *story* in the data to emerge.

In the previous tin, Leverage Color, I shared a driving principle behind my work:

in the previous tip, leverage color, and use a guiding principle coming from Seth Godin's idea of being 'remarkable'. Another concept that I always have top of mind is Occam's Razor, usually described as "the simplest answer is usually the correct one." William of Ockham did not have data visualization in mind when he devised this principle in the early 13th century, but I believe the concept fits quite nicely with the practice. Think about how bar charts have withstood the test of time, and despite their simplicity, continue to be one of the most effective ways to communicate the differences in data.

Less is almost always more when it comes to communicating your data-driven story. To help me prioritize what I share within a single view of my data visualizations, I often stick to a general rule of thumb of including no more than twelve widgets. I adopted this threshold from Google Analytics, which allows you to add a maximum of twelve widgets to a custom dashboard. The twelve widgets include charts, titles, and filters. To help illustrate this tip, let's take a look at one of my most-viewed Tableau Public vizs to date, which was built using four widgets: One title, two filters, and one chart.

[Insert Image 31-1]



I primarily credit the success of this visualization with its simplicity. The visualization is simple in several different ways:

- It asks and answers a single question.
- It offers only two filters - one with two options; the other with four.
- The story is communicated using a single chart. (For more on creating this simple and effective chart type, see the chapter, How to Make Funnel Charts)

The story in this visualization is almost impossible not to understand – and understand very quickly – by analysts and non-analysts alike. By keeping it simple, you maximize the effectiveness of your data-driven storytelling across the largest audience possible.

Chapter 32 – Tip 5: Use the Golden Ratio

“Without mathematics there is no art.” – Luca Pacioli

The Golden Ratio, or 1 : 1.618, is a number found in patterns that we see all around us every single day. In fact, the shape of this book and likely the screen you’re reading this on (if you’re an ebook reader) are probably pretty close to a Golden Rectangle, which is a rectangle where the long sides are 1.618 times longer than the short sides. The Golden Ratio is used in the design of the televisions we watch, architecture, paintings from the likes of Salvador Dali, and even occurs naturally in plants and our own DNA.

Shapes and patterns created with the Golden Ratio have been found aesthetically pleasing for centuries, and when it comes to data visualization, these patterns offer the added benefit of helping us prioritize the content we share in our dashboards. In the last tip, keep your data visualization simple, I shared that I try to stick to twelve dashboard components or fewer, including titles, charts, and filters. The Golden Ratio is one of the tools I use to help guide how to select and layout those twelve widgets.

The gist of the layout is that the most important data you are communicating should be toward the top and to the left, with lower priority information appearing further toward the bottom and right of your view. Just as you start reading a page in a book, you can expect your audience to approach your data visualization the same way.

[Knowing your audience](#) will help you choose which content earns the prime real estate on your dashboard. For example, if your audience is C-level executives, you should likely have your main measurements of success and your progress toward achieving that success near the top left of your dashboard. If you are creating self-serve reports that your audience is going to use to help answer their own needs, you may want to consider prioritizing the filters, which will be a natural starting point for this type of end user.

The Golden Ratio is effective in both a horizontal layout as well as a vertical layout. The choice of a vertical or horizontal dashboard should be based on how you intend to distribute your data visualization. For self-serve dashboards with many filters that other analysts are going to use to help find stories in the data, I typically opt for a horizontal layout. I prefer vertical layouts for dashboards that I know are going to be saved as PDFs and passed around, and for all of my Tableau Public dashboards, which are embedded in blogs that typically have vertical designs.

Using the Golden Ratio in your data visualization is an easy way to subconsciously provide a familiar and well-balanced layout to your audience. It also serves the practical purpose of helping you prioritize content and think about how your audience is going to consume your dashboard.

Chapter 33 – Tip 6: Retell An Old Story

“Our stories are not new; yet, in the retelling we are reborn as heroes.” – Harley King

I think we've all been there – a few months after introducing a new reporting format, you may start to get the sneaky suspicion that your weekly or monthly updates are not being opened by stakeholders. Or perhaps they are being opened, but that fresh new reporting format that was so well-received upon release is no longer being used to gain as much insight as it is capable of providing. This may be due to rarely fluctuating KPIs, a lack of understanding on how to leverage the reporting, or maybe simply a shortage of time to dedicate to finding stories in the data.

The nature of many KPIs used to answer business questions is that they are steady and / or predictable. Think about a website's bounce rate and average time on site – for better or worse – two KPIs often used to measure a company's ability to engage website visitors. Without a site redesign, these two KPIs may never fluctuate more than 5 – 10 percent over the entire life of the website. Once you have seen these numbers two or three times, they become stale, and end users become less and less motivated to use the reporting you're creating for them. In the same vein, you may have KPIs that follow a seasonal pattern, and your end users know exactly what to expect from week to week or month to month; this may make them indifferent to large spikes or dips spotted in your reporting.

The good news is that data visualization provides a means for re-engaging your audience by retelling your data's stories in new ways. Data visualization tools like Tableau enable you to (1) make your dashboards more aesthetically pleasing, (2) add filters and functionality that allows end users to find their own stories in the data, and (3) makes your reporting more usable so that finding insights is more intuitive for the user.

As just one of infinite possible examples of retelling an old story, I took a shot at reimagining sports standings. As a data visualization evangelist, I view data tables as the least effective way of communicating data. As a sports fan, I am forced to view tables almost exclusively to see how my team is doing – and this is true for every league: NFL, NBA, MLB, NHL, MLS, among others. Tables have been used to track league standings in Europe's "football" leagues for over a hundred years. They even affectionately call their standings, "The Table." To show how much more value visualizing sports standings adds compared to the

traditional table, I recreated the table in North America's Major League Soccer as a Tableau Public data visualization.

First, take a look at the traditional standings from mlsoccer.com:

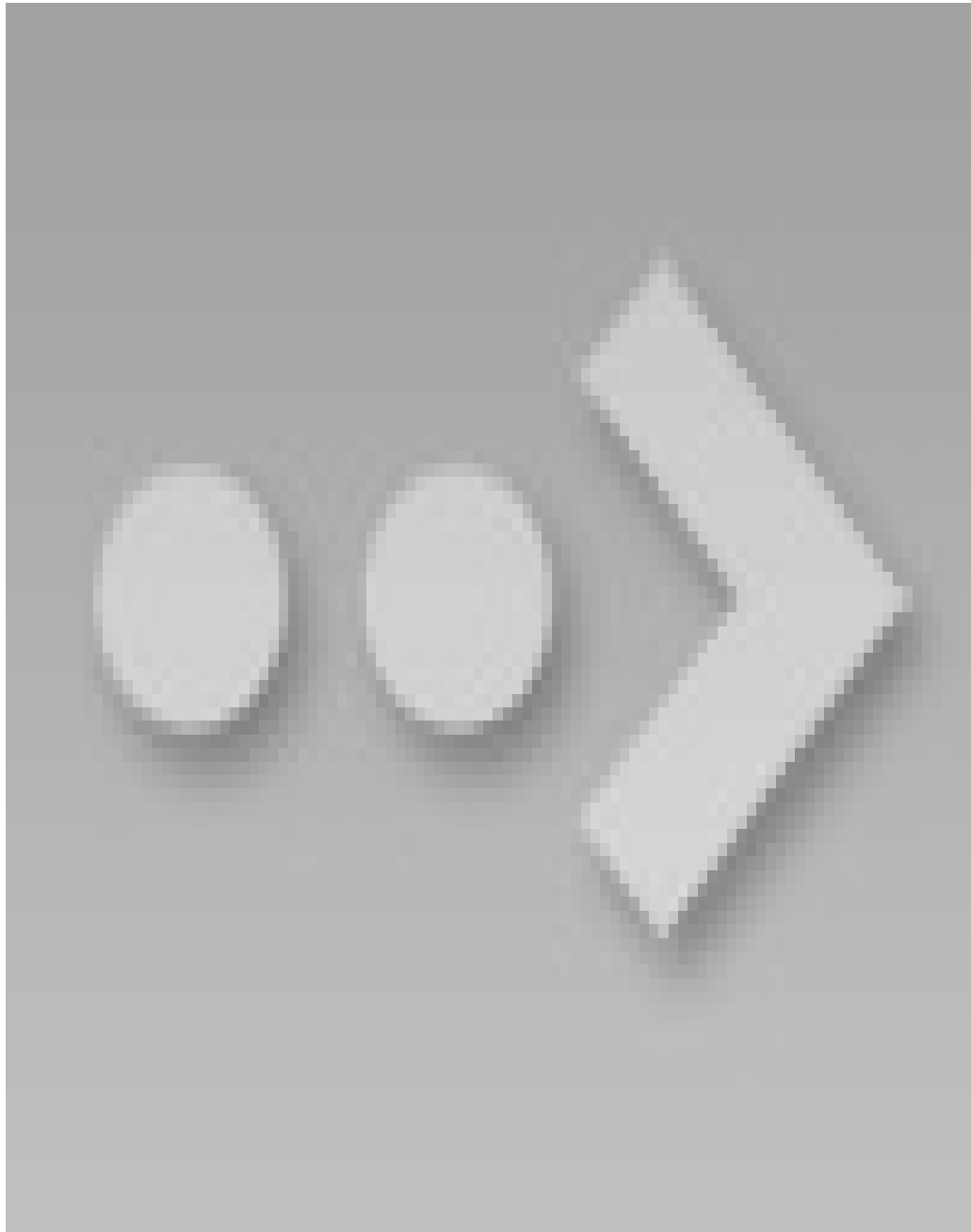
Eastern Conference	Points	Games Played	PPG	W	L	T
Kansas City	35	20	1.75	10	5	5
D.C.	34	19	1.79	10	5	4
Toronto	26	17	1.53	7	5	5
New York	24	20	1.2	5	6	9
New England	23	19	1.21	7	10	2
Philadelphia	23	21	1.1	5	8	8
Columbus	23	20	1.15	5	7	8
Chicago	20	19	1.05	3	5	11
Houston	19	20	0.95	5	11	4
Montreal	14	19	0.74	3	11	5

Western Conference	Points	Games Played	PPG	W	L	T
Seattle	38	18	2.11	12	4	2
Salt Lake	32	20	1.6	8	4	8
Colorado	30	20	1.5	8	6	6
Dallas	29	20	1.45	8	7	5
Los Angeles	27	17	1.59	7	4	6
Vancouver	27	19	1.42	6	4	9
Portland	24	20	1.2	5	6	9
Chivas	23	20	1.15	6	9	5
San Jose	20	18	1.11	5	8	5

Now, take a look at the data visualization created using the same data:

[Insert Image 33-11]

[Insert image 55.4]



In addition to the table, you now have the ability to do the following in a single view:

- Filter between conference and league standings.
- See how many games each team has played and how many games they have remaining.
- Hover over every single game to see the teams that played and the final

score.

- Determine what it will take for each team to make the playoffs.
- Gain better understanding into how teams are performing relative to each other.

The same principles used to add value to this century-old story can be used to reinvigorate your reporting at work, engage your audience, and maximize the impact of your data visualization. I'm willing to bet that by retelling old stories, you will discover many new 'Rags to Riches' storylines in your existing data!

Chapter 34 – Tip 7: Don't Neglect the Set-Up

“If content is king, then context is God.” – Gary Vaynerchuk

When you tell a story, you don't jump to the climax without first building up the characters and working your way through the plot. If you skipped the set-up, the story would not be as effective as possible. Data visualization is the same way.

When you work closely with a set of data for a long time, the insights that emerge may begin to feel intuitive. It is natural to think that the audience for your data visualization will find the seemingly ‘obvious’ stories in the data as easily as you. This may be true occasionally, but why not make it a guarantee by setting the user up for success by adding some context to your work? This will also allow you an opportunity to guide the way your data visualization is consumed and ensure that you and your audience will align on those aforementioned ‘obvious’ stories in your data.

If you don't believe your data visualization will be interpreted differently by each stakeholder, you have likely never seen an Iron Viz Championship. I am always amazed by the diversity of stories that emerge from the same dataset each and every time the competition is held. The designers not only find unique stories in the data, but they communicate the stories using varying chart types, fonts, colors, and *context*.

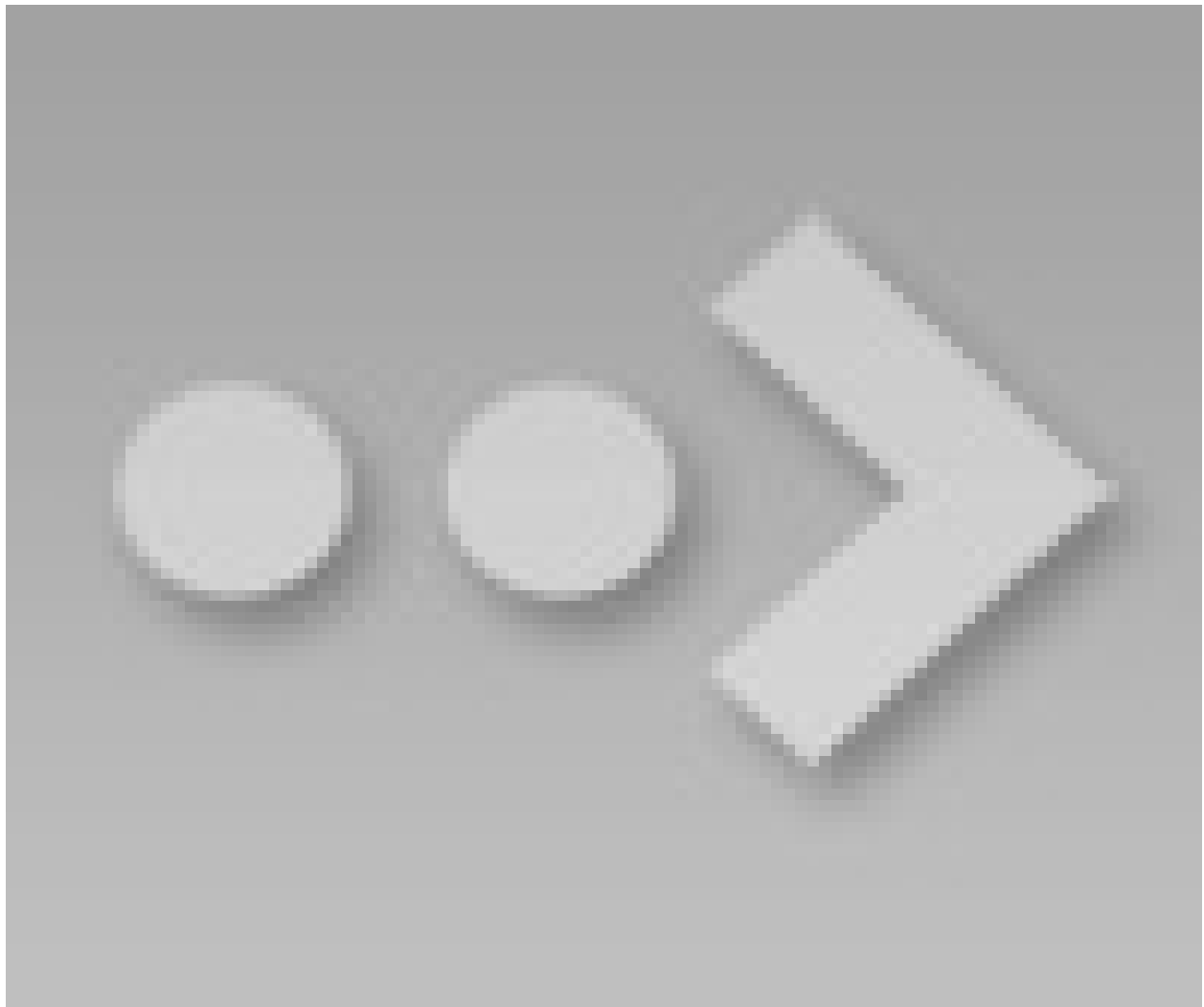
Here are just a few tips on setting up your data visualization:

- *Always include a title.* This helps set an expectation for what your work is about and can provide valuable information to the end user such as data sources and the date range covered.
- *Ideally, you can ask a single question to open your data visualization.* Each item in your dashboard will then ladder up to answering that single question. Oftentimes, one set of data can be used to answer many different questions, but by stating up front what question you are focusing on answering, you guide how your audience will consume your data visualization.

- *Explain the features available in your dashboard up front.* For example, if your visualization is interactive, explain the filtering options available.

Here is an example of a set-up I like from Anya A'Hearn in her 2012 Iron Viz winner, *Does Tornado Alley Deserve Its Moniker?*. Notice that the viz asks one question, and the rest of the viz attempts to answer the question. I also like how she saved real estate by building her color legend into the set-up. This also helps the end user understand how to read the dashboard right from the start:

[Insert Image 34-1]



By not neglecting the set-up, you help align your message with the audience's expectations by guiding the story that the data is telling.

Chapter 35 – Tip 8: Don't Use Pie Charts

“...the only worse design than a pie chart is several of them...” – Edward Tufte

Pie charts have become one of the most recognizable and widely adopted chart types in business and data journalism. Being that the message that pie charts convey is almost universally understood, it is tempting to incorporate pie charts into your own data visualization. However – data visualizer beware - while the message that pie charts convey is immediately recognized (parts of a whole), *the insights that pie charts provide are not*. Using pie charts work against you and your end users because when compared to other visualization choices, they make it more challenging to glean *accurate* stories from your visualization.

The pie chart's primary limitation is that people are much better at comparing lengths and heights, as you would see in a bar chart or line graph, than they are at comparing areas within a pie. Further, the long tail results, or the thinner pieces of a pie, tend to become unreadable. Pie charts are also a very poor vehicle for communicating changes over time.

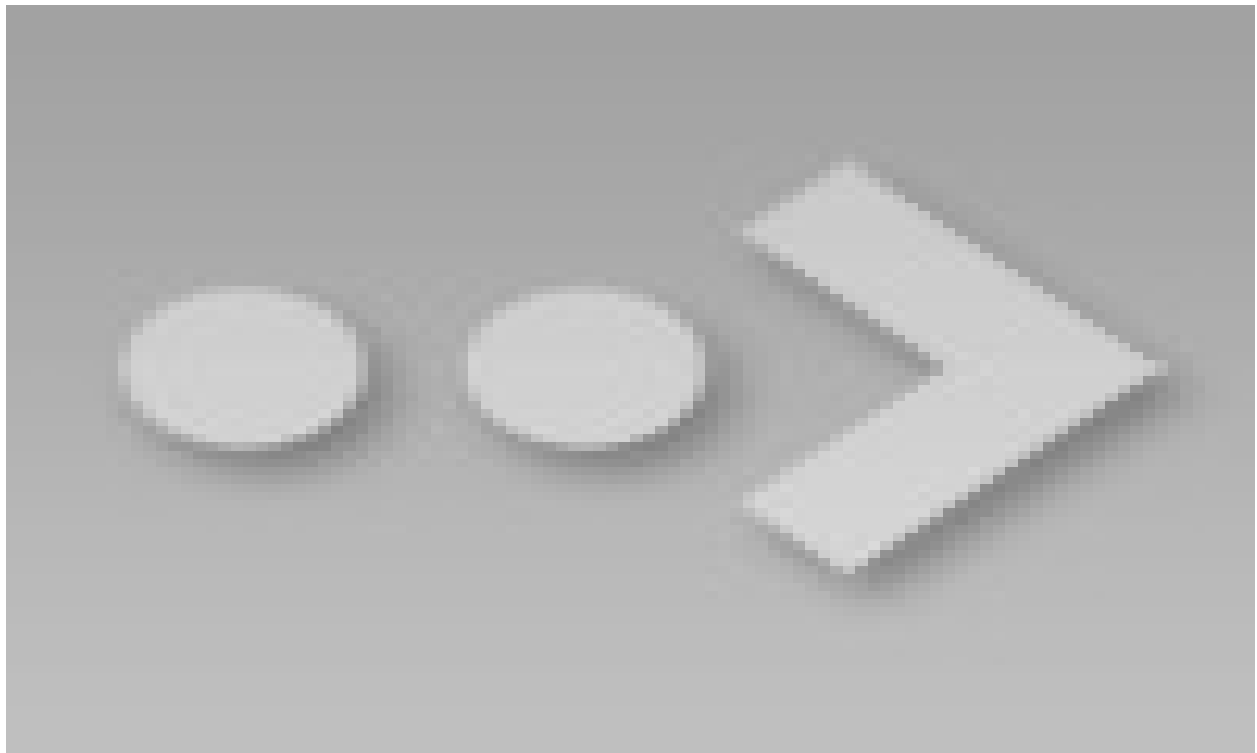
Despite the empirical evidence against pie charts, I continue to see them in some of the world's most trusted news sources, stock reports, and corporate dashboards, among many other places. It is easy to make fun of people's use of pie charts ([like this one!](#)), but I want to be clear that when you hear people recommend avoiding pie charts, it is not some kind of elitist-data-visualization-specialist credence – it is a best practice based in evidence. I've always viewed the prolific use of pie charts as a huge opportunity to educate and improve the data visualization space. While graphical methods of statistics date back over 200 years, we still have a lot to improve – which is a great thing for you if you are reading this and are interested in evangelizing the benefits of data

visualization.

To help, here is a simple exercise to illustrate the reduction in time to insight from a pie chart to more appropriate chart types.

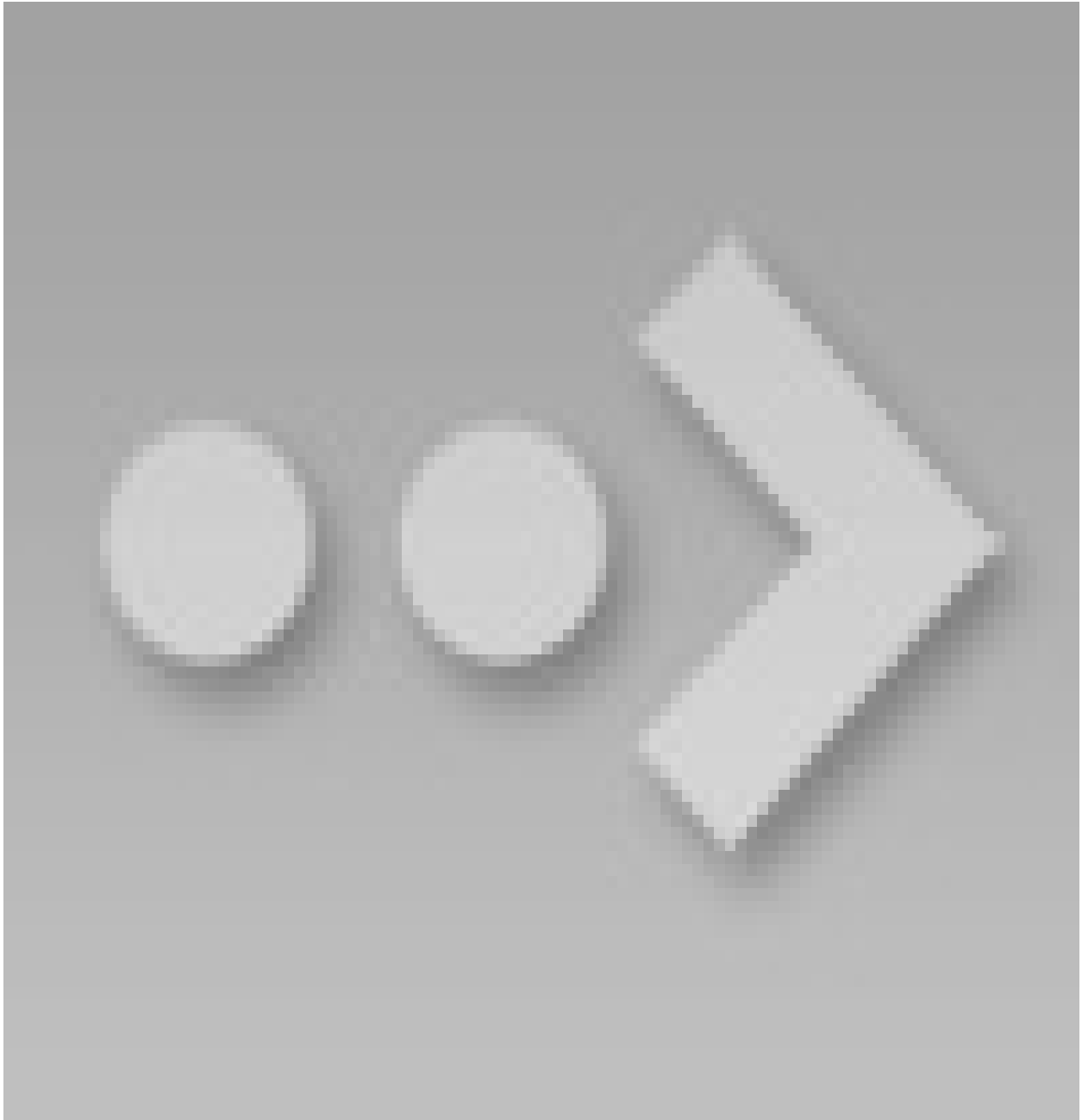
First, have a look at this series of pie charts, showing the sales by department over a four-year period. Take 15-30 seconds and think of a couple of insights that the charts provide:

[Insert Image 35-1]



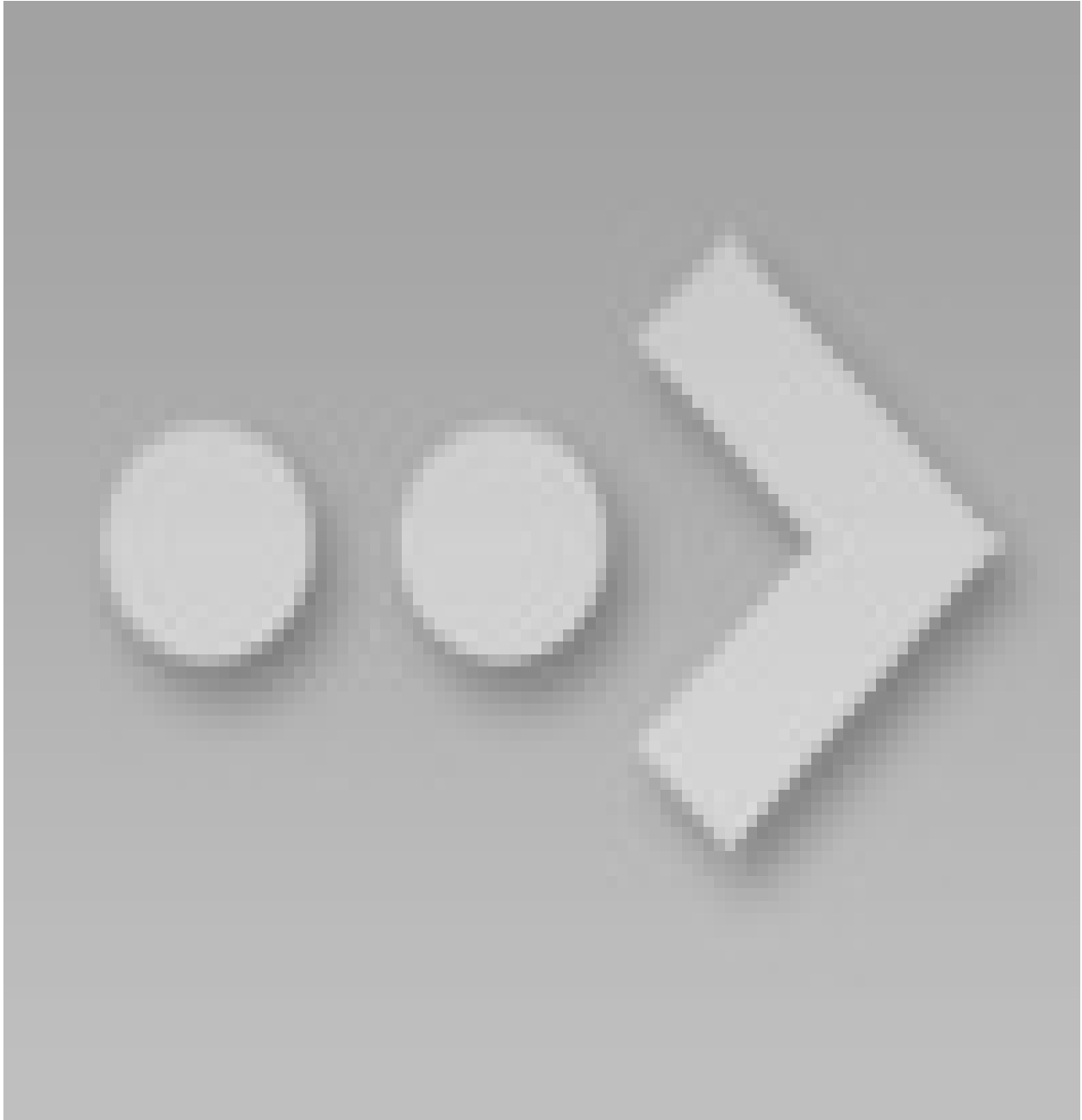
You likely gathered that the entire pie was getting slightly larger over time (though it was hard to tell by how much and what that means), but it is pretty difficult to ascertain which pieces of the pie are causing the growth. With an easy tweak in Tableau, look at the exact same data as a line graph. Take the same 15 - 30 seconds to understand what the data is telling you - though it likely won't take you nearly that long.

[Insert Image 35-2]



Much easier, right? By converting our pie chart to a line graph, we can easily see that all three categories are on the rise, and by how much. We were able to process the visual and get to the insight much faster. If you are not as interested in the performance over time, and want to see how each category performed compared to each other each year, use a bar chart:

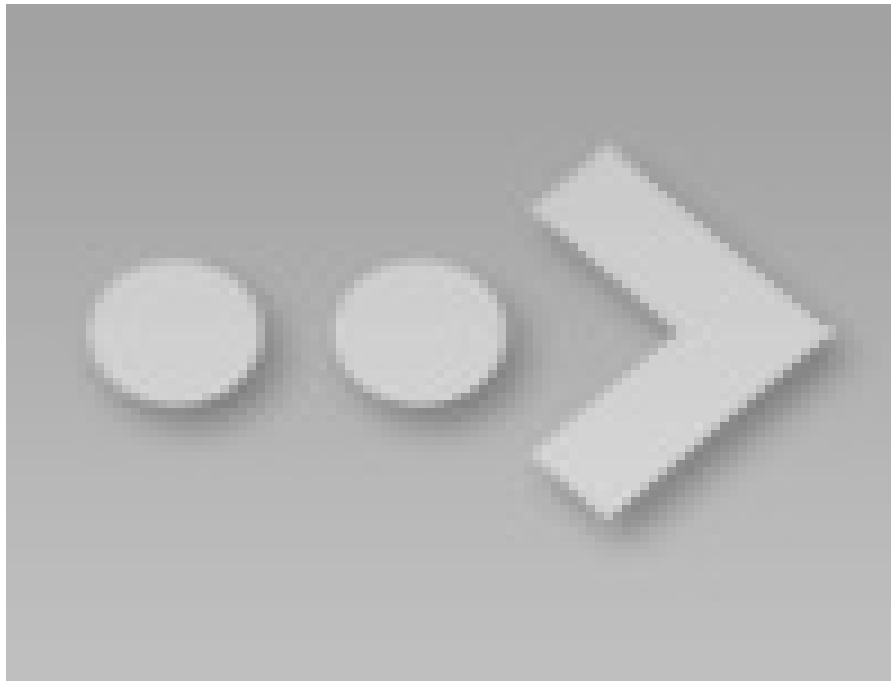
[Insert Image 35-3]



Again, your brain processes heights of bars more easily than it process areas in a pie. Using a bar chart also provides you more real estate to display the values, either as dollar amounts and/or percentages of the whole (as you were trying to convey by using a pie chart). To take this one final step further, if you only care about comparing how each category performed against the others for one period

in time, you can isolate the bar chart to show just the year in question:

[Insert Image 35-4]



I understand that you may have a stakeholder that has a hard time letting go of pie charts. If you or a boss can't quite go cold turkey, here are a couple of tips for using pie charts while you transition away from them:

- Stick to five slices or fewer, including the “other” category. As mentioned previously, the thinner, long tail, slices become unreadable. If you find yourself saying, “but I have to represent all fifteen of my categories”, that is another vote for you moving away from pie charts.
- Only use pie charts to show comparisons for one point in time. Avoid using them in a time series analysis as shown above.

Context is useful in data visualization because it helps you avoid the dreaded question, “So what?” As powerful and simple as they are, fundamental chart types such as bar charts and line graphs are often used in a vacuum. Lack of context makes it difficult for an end user to understand, engage, and eventually *take action*. By building context into the visuals themselves, you help the stories in the data emerge, even if you aren’t there to explain them.

For this reason, context helps us tell both types of stories discussed in the introduction to this section. If you as the visualization creator know the endgame, you can use context to make your story stand out and ‘stick’. If you are building a self-service interactive data visualization, adding in context will help your end users analyze the data and find the storylines that are most relevant to them.

There are several ways to add context to your work including:

- Index scores; not only do these scores provide comparisons, they help normalize the data.
- Sparklines; this chart type is essentially a small set of line graphs that allow you to view high level trends across several metrics at the same time.
- Bullet graphs; this chart type is a variation on the bar chart that adds context in the form of comparison points such as prior period performance and/or goals.
- Small multiples; this chart type is similar to a table of visualizations instead of raw numbers, which allows the end user to compare results across several different dimensions and/or measures at the same time.

For an example of small multiples, see the following visualization which featured a set of small multiple stadium maps to help illustrate how the cost of ticket prices changed throughout the 2015 MLB World Series.

[Insert Image 36-1]



Chapter 37 – Tip 10: Use Callout Numbers

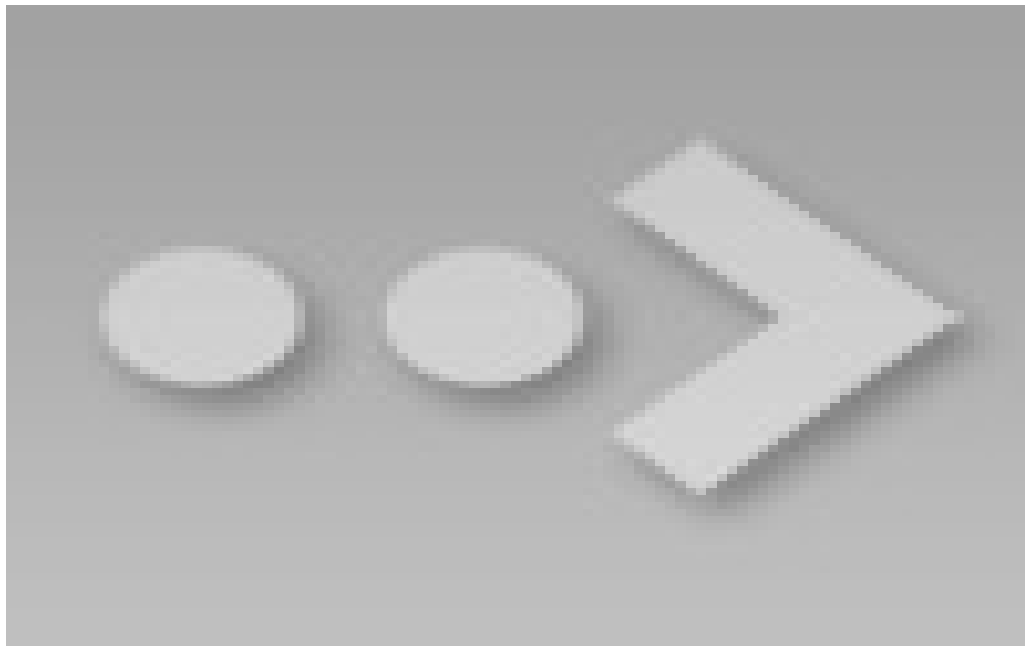
Callout numbers are one of the easiest ways to communicate what is most important to you in your data-driven story. A Google Images search for “callout” provides several examples of speech bubbles, thought bubbles, and Batman-style “BAM” / “POW” graphics. While I typically do not enclose my callout numbers in such an illustration, they share the purpose of explicitly telling the main story in your data visualization.

Callout numbers, as I’m calling them, are simply oversized numbers that should be in a legible font that is easy to consume. As these callout numbers will be communicating the most important numbers in your view, they should be prioritized near the top and left of your dashboard. For more on prioritizing and laying out content, see tip five, Use the Golden Ratio.

In tip four, I discussed keeping your data visualizations simple. I also shared a general rule of thumb to keep the widgets on your dashboards to twelve or fewer. This is a critical concept if you want to cut through the noise and be as effective as possible at communicating the story in your data. Callout numbers provide a means for cutting through the noise, even within your own views.

Here is an example from my portfolio using callout numbers. This viz includes two numbers that stand out: (42) Jackie Robinson's number that is retired league-wide and (71) the number of seasons without integration in MLB. The end user can see instantly what the dashboard is about and why I think the content is important. From there, they can decide whether or not they want to explore the data.

[Insert Image 37-1]



If you use callout numbers in your view, they will likely be the first thing that your end users are drawn to. This serves two important purposes: (1) it instantly tells the main point of your dashboard and (2) gives the end user a natural place to decide whether or not they need to continue searching for the answers or context they are looking for. This could come into play with C-level executives, who may only want a top-level data point before leaving your visualization – and this is not necessarily bad.

I compare this behavior to a website's bounce rate. There is a common misconception that it is *always* bad to have a high bounce rate (or rate of people that view only one page on your website before leaving). However, a perfectly optimized webpage that provides exactly what the visitor is looking for without making them search for it on a second page will have a very high bounce rate. In the corporate world, you may have numbers that do not fluctuate very often, and callout numbers will be a valuable tool for either providing what users are looking for or communicating that it's important for them to investigate further.

Chapter 38 – Tip 11: Allow Discovery

“Tell me and I forget, teach me and I remember, involve me and I learn.” – Benjamin Franklin

I've always loved the above quote from Benjamin Franklin, and especially in the context of data visualization, because I think it applies to the practice in a couple of ways. First, simply telling a stakeholder your results is the least effective way to get your data-driven story to stick. By its nature, visualizing data provides additional context beyond verbally sharing findings, so practitioners are already a step ahead in communicating actionable insights. Second, tools like Tableau allow data designers / artists / developers to build in interactivity that lets end users find stories in the data on their own. When end users are involved, whether it be in the iterative process of a dashboard design or in interacting with a dashboard, the shared sense of ownership goes a long way towards making your data visualization a success. When end users find an insight on their own, they are more likely to remember it, and what's better, *do something about it*.

Allowing discovery is a tip I take personally because, as a huge sports fan in a relatively small market (Kansas City), my teams do not get as much coverage as some of the teams in larger cities on the coasts. With tools like Tableau, there is no longer an excuse to not include relevant information for every fan. This is something I keep in mind every time I create a new data visualization. This same principle applies in a corporate setting, as you now have the capability to provide

relevant information to a variety of stakeholders in the same amount of space. For example, you can provide filters that change the view based on product categories or regions. You may have data size considerations, but it is theoretically possible to allow end users to look at every order or every customer individually.

In Tableau, there are three easy-to-implement ways to allow discovery, which gives you the power to make your end user part of the story.

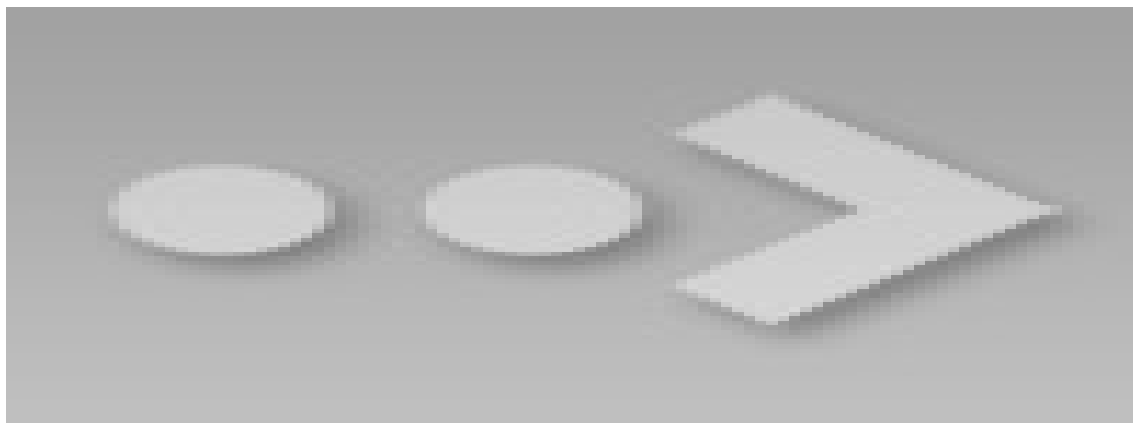
Filters

We'll start with the most obvious tool for allowing discovery: Filters. Any filter used in the making of a view can be added to a dashboard. Filters can be shown in two ways:

1. From within a sheet view, right-click a filter on the filters shelf and select "Show Filter". When the sheet is added to a dashboard, the filter will appear with the sheet.
2. From within a dashboard view, (a) click the down arrow that appears when you hover over a sheet (b) hover over "Filters" and (c) make the appropriate selection.

What Filters look like:

[Insert Image 38-1]



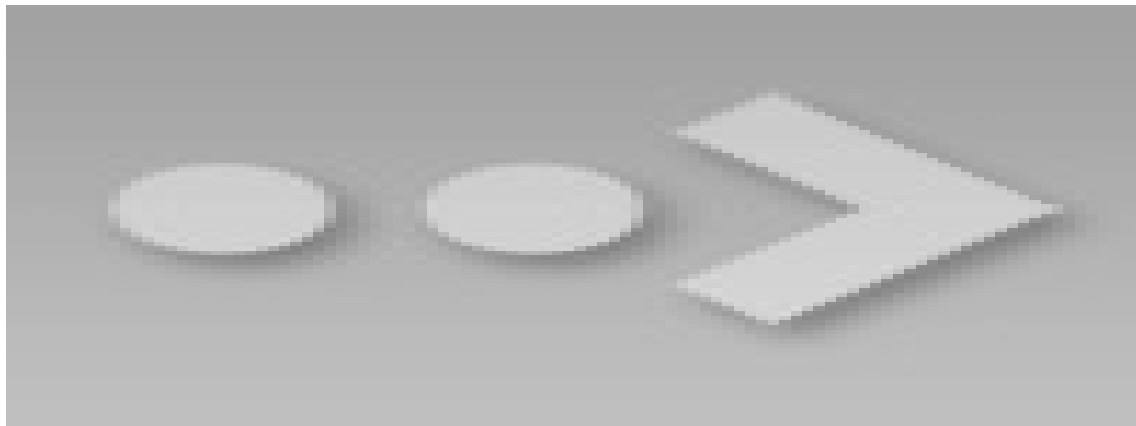
Dashboard Actions

Dashboard actions are a means by which users can add interactivity to a dashboard

Dashboard actions are a more subtle way to add interactivity to a dashboard. They also have the added benefit of saving processing time if your Tableau workbooks are published to Tableau Server, Tableau Online, or Tableau Public. Dashboard actions are easy to create, but there are many options on how they can be utilized. The easiest way to implement dashboard actions is to click the down arrow that appears when you hover over a sheet and select “Use as Filter”. This will create a simple dashboard action behind the scenes that will filter your entire dashboard based on the item clicked in the sheet that you opted to use as a filter.

What Dashboard Actions look like: (Clicking Florida filters the area chart on the right)

[Insert Image 38-2]



Many people do not realize that web-based Tableau workbooks are interactive. This will improve as more and more people are exposed to the functionality of the software and end users become comfortable exploring a dashboard on their own. You may want to consider using hover actions so end users that are new to the software may stumble into the interaction capabilities that you’ve built in. For example, they may accidentally hover over the map in your dashboard, causing a change which makes them realize how to use your data visualization. This option is set when you build a dashboard interaction; instead of having the action execute on select (or click), choose to have the action execute on hover.

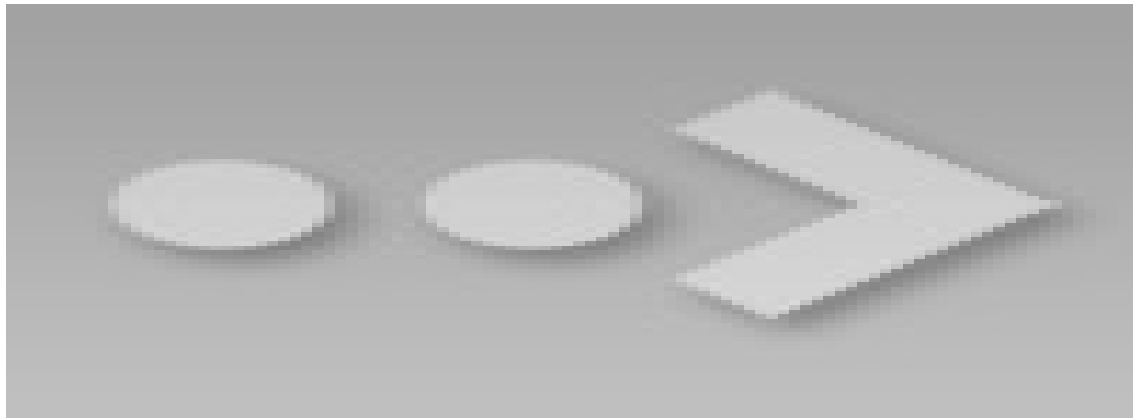
Parameters

Parameters act as a wildcard that can be used in calculations. By showing the

Parameters act as a wildcard that can be used in calculations. By showing the inputs for the parameters on your dashboards, end users can experiment with different scenarios on their own.

What Parameters look like:

[Insert Image 38-3]



Parameters are a slightly more complex way of allowing discovery. To learn how to build this simple what-if analysis, see the chapter, *How to Make a What-If Analysis Using Parameters*.

Chapter 39 – Tip 12: Balance Data and Design

Data should always be the heart of any data visualization. Obviously, you cannot have a *data* visualization without data – and it is imperative that the data is accurate and honest. If I had to choose one element of a new data visualization to focus on between its data or design, I would make sure the data is right first every single time.

That being said, I'm going to explain how without making some design considerations, your data visualization is doomed to falling short of its full potential. Balancing data and design is another tip that I take personally because I often hear criticism from data purists that do not see the value in complementing the data with an appealing design. My opinion on the topic

complementing the data with an appealing design. My opinion on the topic stems from Seth Godin's principle of being remarkable – which means your work is good / unique / interesting enough to cause the audience to make a remark about it. In today's age, making a remark means sharing it on social media, or in a corporate setting, it may mean passing your story up the ladder.

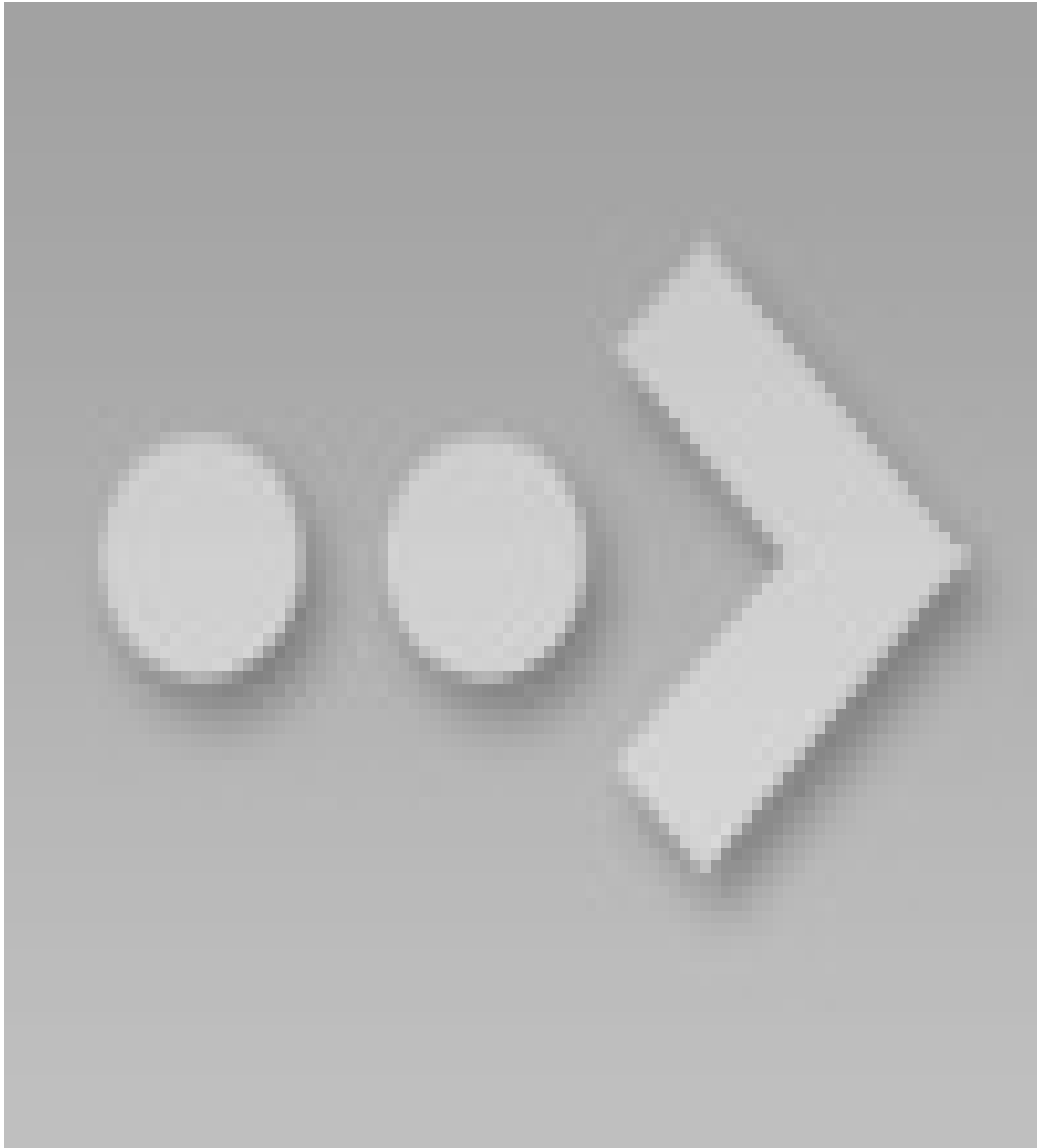
What is the purpose of data visualization?

My answer is that the purpose of data visualization is to find and share actionable stories that are based in quantitative evidence. To make your insights actionable, this means sharing them with the most relevant, and many times, largest audience possible. If you don't provide your data stories in a well-packaged design, you drastically minimize the chances of your work spreading, and thus, it is consumed and acted on by fewer people.

To help illustrate, I stripped out most of the design elements in my viz, *Your Salary vs. a MLB Player's Salary*. In this dashboard, the end user can enter their own salary and compare it to the salary of any Major League Baseball player across several different hitting and pitching statistics. The data is extremely fascinating - if not depressing - on its own, but I believe it is the design that helped this become my most shared data visualization on Twitter to date.

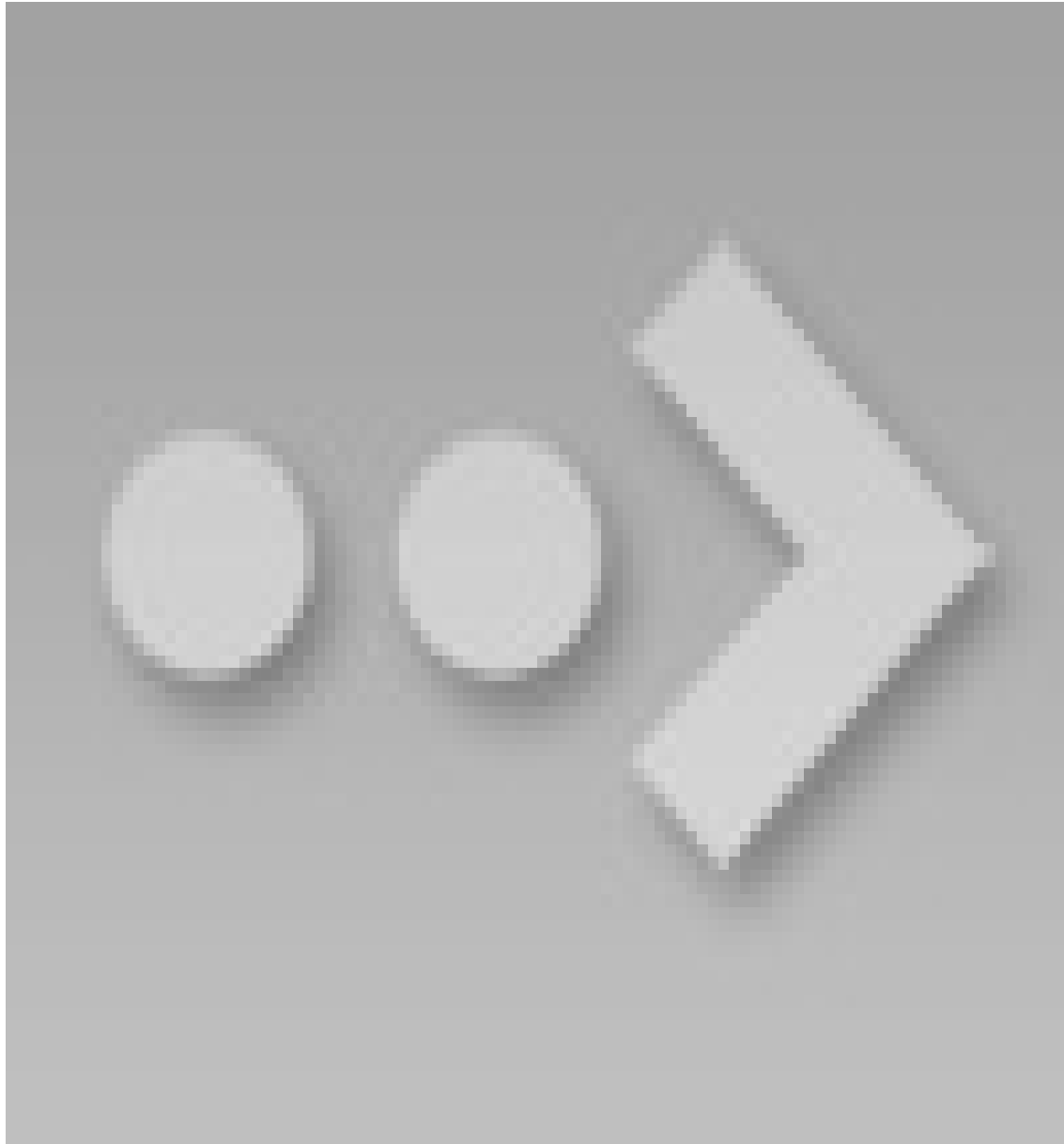
Before

[Insert Image 39-1]



After

[Insert Image 39-2]



I want to point out that the *only* elements in this entire design that required any Adobe Photoshop or Illustrator skills are the three icons, and even the foundation of those were built using stock Illustrator files from [iStockPhoto.com](https://www.istockphoto.com). The chart type surrounding the icons were built in Tableau. If you would like to learn how, see the chapter, [How to Make Donut Charts in Tableau](#). There are plenty of techniques you can use to balance data and design, even if you've never used Photoshop or Illustrator.

Design enhancements you can make immediately without any graphic design experience:

1. Put some thought into your color selections. Choose simple color palettes and “mute” them by adding some transparency. The latter can easily be accomplished by clicking on the Color Marks Card and changing the opacity to 80 – 90%. For more on color, see tip three, Leverage Color.
2. Switch up the typography. Choose some unique fonts that are easy to read and visually-appealing. Make key elements different point sizes to denote your priorities in a subtle way.
3. Lay your data visualization out in a way that guides the end user through your story and provides some spacing for each element. For more on dashboard layout, see tip five, Use the Golden Ratio.
4. Pay attention to the details. This is perhaps the easiest way to make your data visualization stand out. Whenever end users think a data visualization looks really good, but they can’t quite explain why, it’s because of the details. Clean your dashboards up by minimizing lines. Add borders to marks such as bars and shapes. Add some transparency to show overlapping values.

These are just a few of probably dozens of tiny changes you can make to your work. These details add up quickly, resulting in a polished, professional piece that will maximize the chances of your story being shared.

Chapter 40 – Tip 13: Eliminate Chartjunk (but Not Graphics)

Chartjunk is a term coined by Edward Tufte in his book, *The Visual Display of Quantitative Information*. In it, Tufte defines chartjunk as “non-data-ink or redundant data-ink”. My interpretation of chartjunk is that it is any design element that is counterproductive, actually detracting from a data visualization rather than adding value to it. Chartjunk can be as brash as charts that are made to fit within a graphic, with the data almost a complementary element to an illustration (see the work of Nigel Holmes). It can also be as subtle as a three-dimensional bar or extra gridlines. Calling non-essential elements ‘chartjunk’ is

a fair criticism, but the same person who invented the term also said the following:

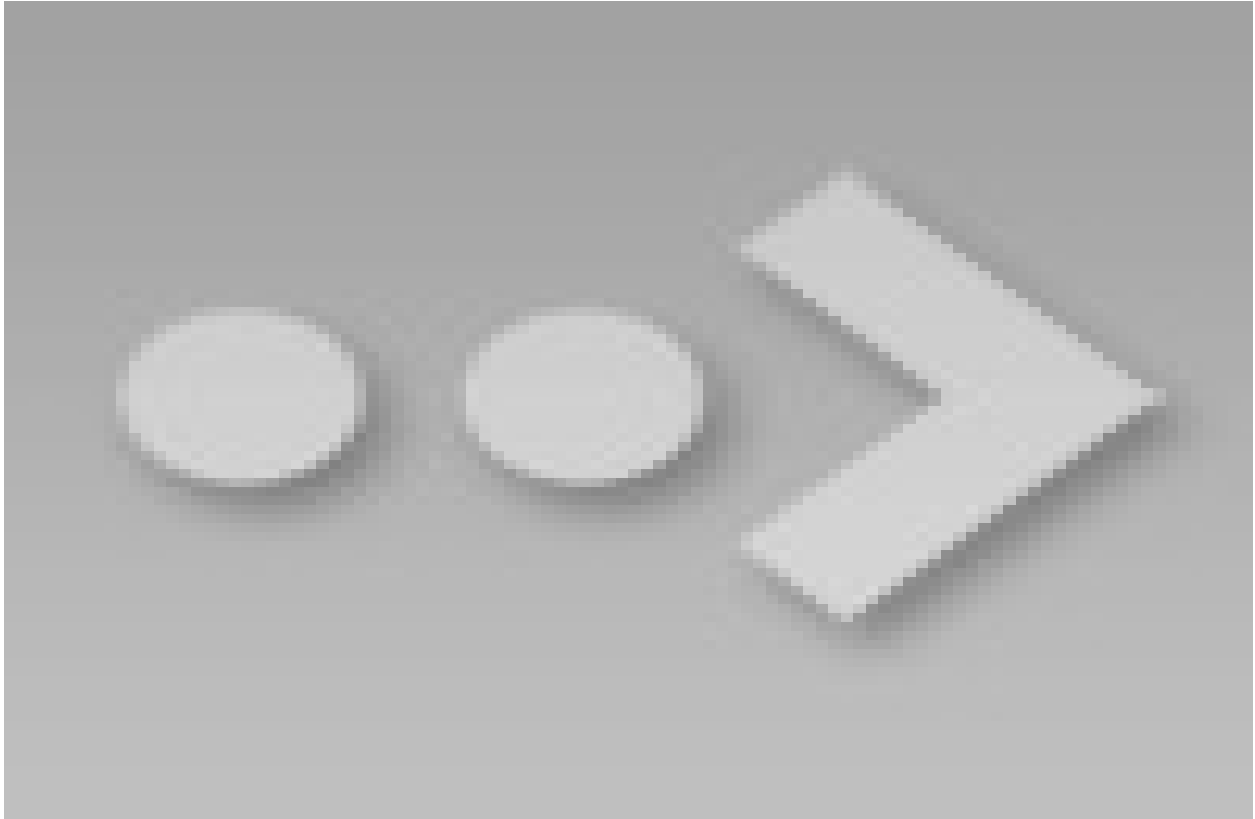
“Only a picture can carry such a volume of data in such a small space.” - Edward Tufte

As with most debates, the truth probably lies somewhere in the middle. When the design becomes the primary purpose of a data visualization, it can become distracting and even misleading. Neglecting design completely and a data visualization can be unengaging. In the last tip, I explained the importance of balancing data and design in data visualization. The tasteful use of graphics is one way to do that.

Chartjunk is not always composed of graphics. There are several examples of ‘non-data-ink’ outside of graphics including extra gridlines, shading, and three-dimensional marks. In the same way, graphics do not always create chartjunk. In fact, graphics are one of the most powerful ways to communicate information in a small space, and they also help engage an audience.

Consider the following ‘symbol map’ showing sales by US state. This is technically a scatter plot with a circle on each combination of latitude and longitude. The circle is sized and colored by the sales amount. Here is the scatterplot with no image:

[Insert Image 40-1]



Now see what happens when we add a background image to the exact same scatter plot:

[Insert Image 40-2]



The map graphic helps you decode dozens of latitude / longitude pairs almost instantly, leveraging schemas for state locations that you have been building up in your mind over many years. The map also provides geospatial context which can lead to additional insight that you may not have uncovered looking at a crosstab or bar chart view. This would be especially powerful with an additional data layer, such as showing sales by zip code in relation to store locations.

Research shows that people process visuals 60,000 times faster than text descriptions. To help communicate more efficiently and engage your audience, consider using graphics to help complement your data visualization in these three ways:

Background

An attractive background can be a great way to contain a data visualization while communicating what the piece is about. In my opinion, a background with a clear interior container for the charts and graphs is not a distraction, but I do caution you to leave the background as the background. Once background

caution you to leave the background as the background. Once background elements overlap or become the primary message of a data visualization, it is counterproductive and takes the end user longer to find the story that your data is sharing.

Icons

By using icons instead of words to communicate what a number represents helps end users understand what's happening in your data visualization without them even having to read. For example, a dollar icon could be used to represent sales. Icons also help reduce language barriers in the case that your data visualization is viewed internationally.

Marks

Think about a scatter plot showing the sales by three departments: furniture, office supplies, and technology. You could use a unique shape to represent each of the three departments, but this would cause the end user to look back and forth to the legend to determine which shape represents which department. One alternative would be to use graphics as your marks to cut out the middle man and help your end user process the view faster. For example, a chair, stapler, and phone could represent furniture, office supplies, and technology, respectively.

Using graphics is about first engaging your audience, and then helping them process your data visualization more efficiently.

Chapter 41 – Tip 14: Use Freeform Dashboard Design

“Conformity is the jailer of freedom and the enemy of growth.” - John F. Kennedy

Using freeform dashboard design is an easy way to tie several of these data visualization tips together including Use the Golden Ratio and Balance Data and Design. Starting with Tableau version 8.0, data visualization developers have the option to not only select the overall height and width dimensions of their

dashboards, but the ability to place charts, titles, filters, and graphics on the exact X and Y coordinates of their choosing – down to the pixel.

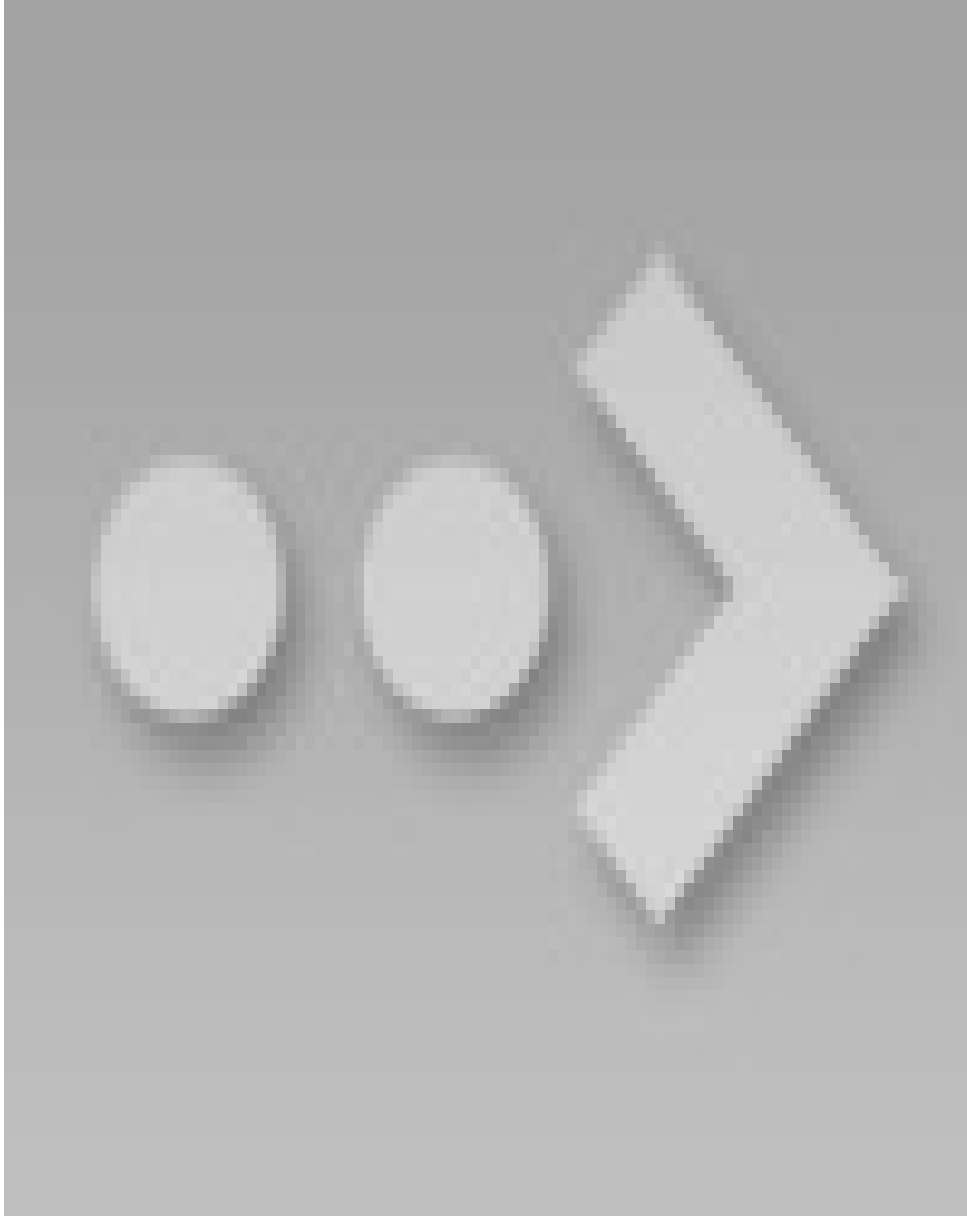
This feature is extremely powerful in that it removes all spatial restrictions and offers designers the freedom to use an entire dashboard canvas. Prior to Tableau 8, all dashboard elements had to literally fit inside of a box. The creative freedom that comes with freeform dashboard design provides additional means for engaging end users and making your data visualization worthy of sharing. The biggest benefit that I see from this update is that Tableau users can now create what are essentially *interactive* infographics. Here are just a few of the possibilities:

1. 1. Design a background image for your data visualization that is the same height and width as your final dashboard. Place the image in the dashboard first, then float all of the elements over the image.
2. 2. Experiment with the sizing and layout of icons and other graphics to create the best look and feel for your data visualization.
3. 3. Place dashboard elements in better relation to associated content.

To help illustrate, take a look at a data visualization I originally created in Tableau 7, but gave a makeover in Tableau 8. In fact, this was a contest with the specific objective of showing off the new features in Tableau's latest product release – including freeform dashboard design.

Original

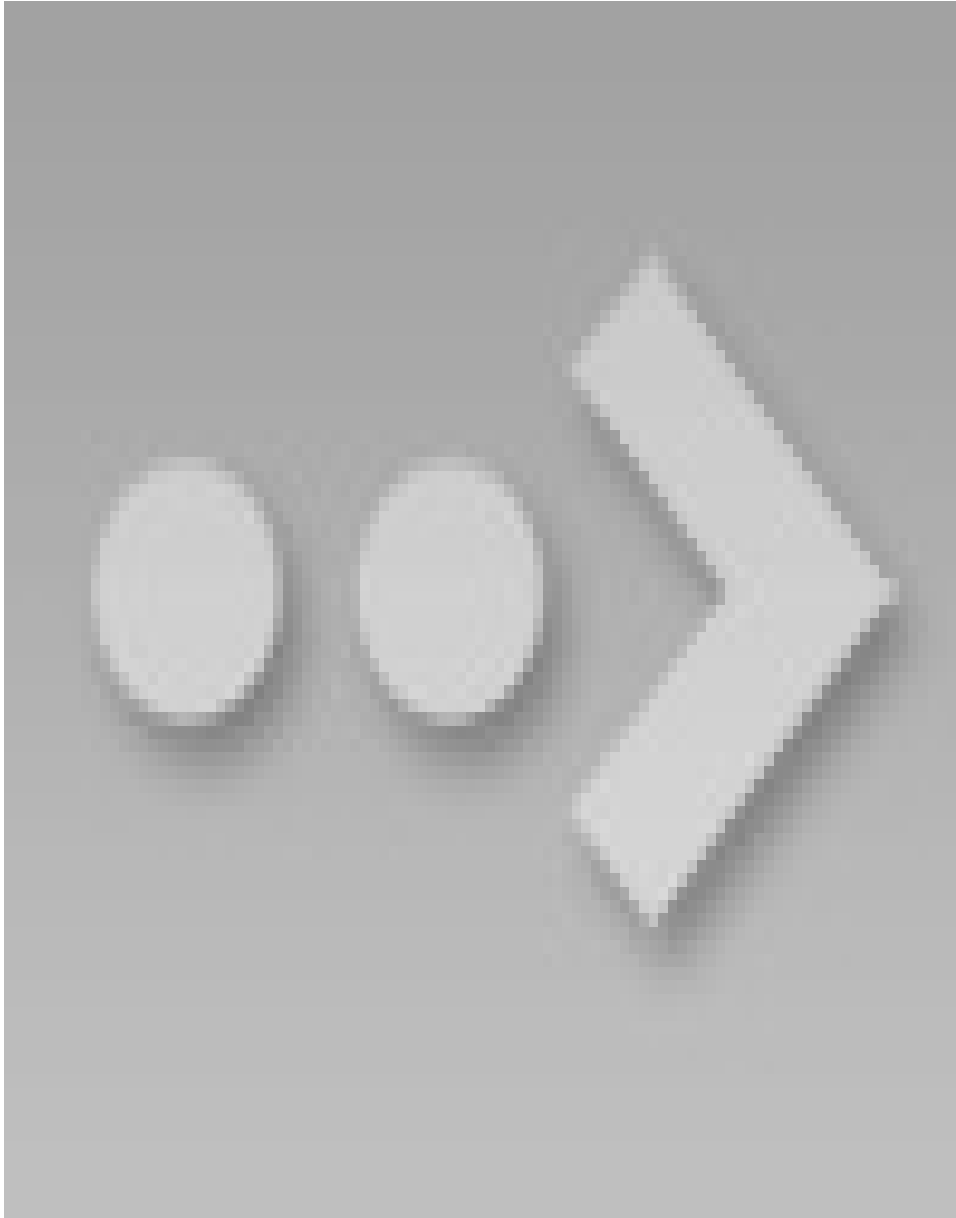
[Insert Image 41-1]



Notice that every item is in its own container, with no overlap. I did not have much choice but to waste a great deal of real estate in the center of the dashboard where the main player comparison takes place.

Freeform dashboard design

[Insert Image 41-2]



With freeform dashboard design, I was able to create an attractive background image and float everything on top of that image exactly where I wanted it, resulting in an interactive infographic. I was also able to place icons for Salary, WAR (an advanced baseball statistic), and XVAL (an advanced baseball statistic) in better relation to their titles and values.

Chapter 42 – Tip 15: Tell a Story

“Storytelling is the most powerful way to put ideas into the world today.”

- Robert McAfee Brown

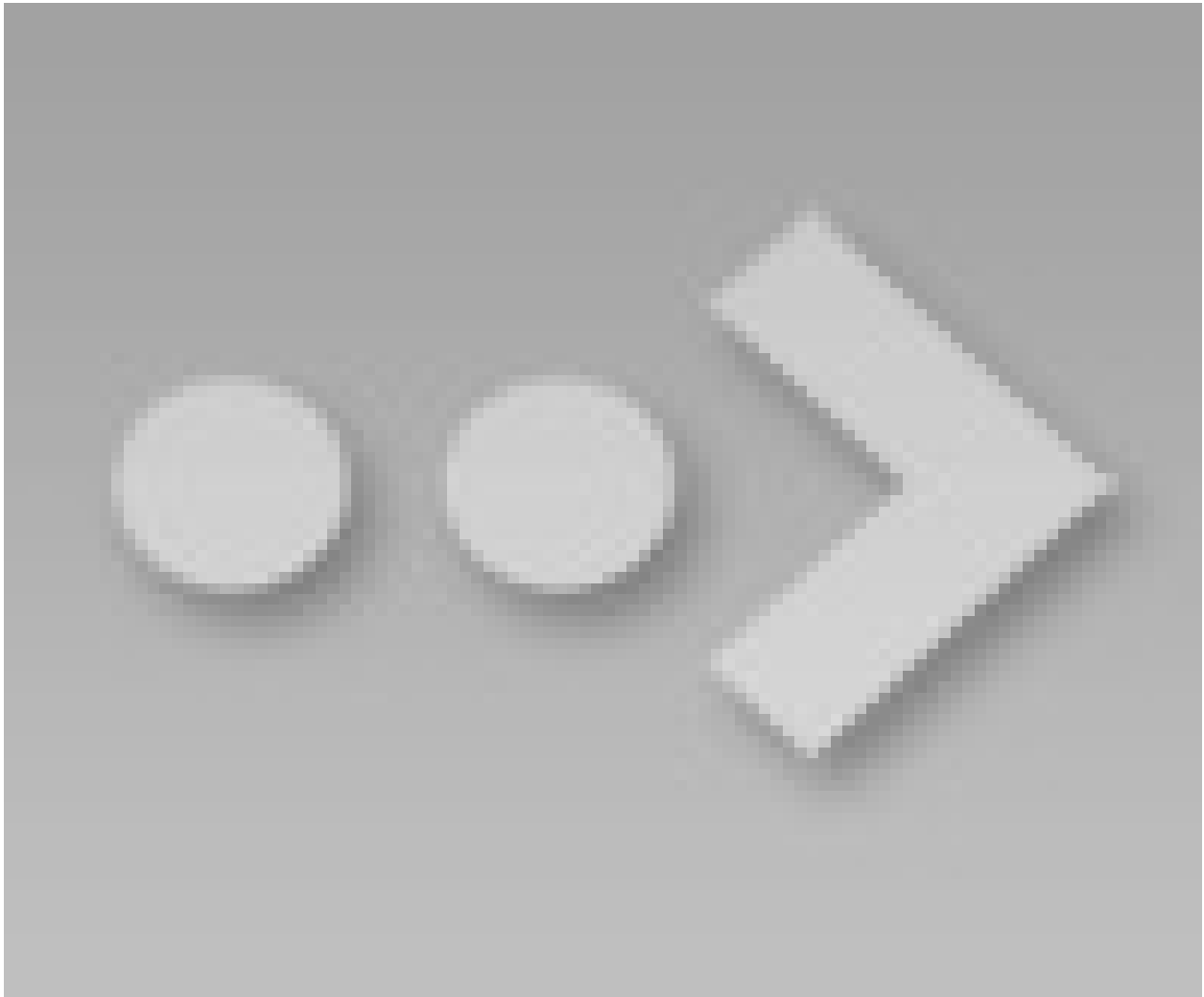
In an ideal scenario, the stories in your data are so compelling that they are self-explanatory. Unfortunately, this is rarely the case. What's worse, at times we are so close to the data and insights that we don't realize that our data visualizations need additional context in order to be properly understood.

If we want our data visualizations to be understood, elicit sharing -- and eventually -- cause action, we need to help tell the data-driven stories in our dashboards. Here are just a few ways to help complement a data visualization with written anecdotes.

1. 1. In the seventh tip, I remind data visualization practitioners to '[not neglect the set-up](#)'. By adding some context to open a dashboard, you clearly communicate what your work is about. This works even better if you are able to ask a single question that the rest of your dashboard attempts to answer. This guides end users and helps them find the answer on their own, giving the discovered insights a better chance to stick with your audience.
2. 2. I also use 'inline insights' as a tactic for communicating my analysis of the data. To do this, I will simply build a text box into my data visualizations that provides real estate for me to add my own two cents about my findings and recommendations.
3. 3. Thirdly, don't underestimate the power of annotations. Many times, the practice of data visualization is extremely dependent on context. For example, if I am analyzing web analytics data and see a large spike in traffic, I would like to know what offline tactics may have driven the spike. I may be able to see that the spike was a result of an increase in direct traffic, but without input from advertising stakeholders, I will not be able to fully explain the trend. Perhaps our company launched a television commercial during the timeframe in question. These are the pieces of context that I like to add in the form of annotations.
4. 4.

Finally, if you have a chance to be in the room when an end user interacts with your data visualization, try to – literally - tell a story. Why did you make the

visualization? What were the key findings? What *actions* should we take next?



With a firm grasp on the technical tactics in Tableau, a strategic approach to data visualization, and some elements of storytelling, your work will have the best-possible chance at causing positive actions.

Chapter 43 – A Spreadsheet is Not a Data Visualization

Like the movie *Groundhog Day*, but without its Certified Fresh 96% score on Rotten Tomatoes, there is one conversation I find myself having over and over again in my career as a data visualization consultant. It typically goes something like this:

like this:

Excel fan: “Thanks for this – looks great. Would it be possible to take your world-class data visualization and turn it into an unreadable wall of numbers?”

(Okay, the ‘world-class’ part is an exaggeration – no one has ever said that.)

Ryan (me): “But Excel fan, look how much easier it is to gain valuable business insights more accurately and efficiently when the numbers are visualized! Isn’t that really cool?”

Excel fan: “Steven, my stakeholders are more comfortable seeing the exact numbers.”

Ryan (me): “It’s Ryan, actually. Let me just show...”

Excel fan: “Brian, please provide a text table of this information. Also, if there’s room along the bottom, please add at least one pie chart.”

At least this is how I sometimes feel the conversation goes when I’m having these discussions. I admit it can be discouraging, but mostly because I feel I have let my stakeholders down by my own failure to educate them on a better way of looking at data. It can also be encouraging, because it reminds me there is still a long way to go before best practice techniques are widely adopted and that I have a long career in data visualization ahead of me. That means I have quite a bit of time to earn money that I can use to purchase Bar-B-Q.

In 2016, I attended the Tapestry Conference in Estes Park, Colorado – an inspiring one-day event full of storytelling and networking with an amazing group of data journalists and data visualization practitioners. One of the most memorable sessions for me was the opening keynote by Scott Klein of ProPublica.

During the session, Klein shared several fascinating examples of early data visualization. What I found most interesting was that William Playfair, inventor of the bar chart and line graph, was not celebrated in his time. In fact, most people did not understand the value in visualizing quantitative information and were critical of not using tables.

In other words, data visualization practitioners have been fighting this same fight

since, quite literally, *the beginning of data visualization time*.

A spreadsheet is not a data visualization

Spreadsheets are extremely valuable in business. They can be used in accounting, to store information, or to do calculations, just to name a few purposes.

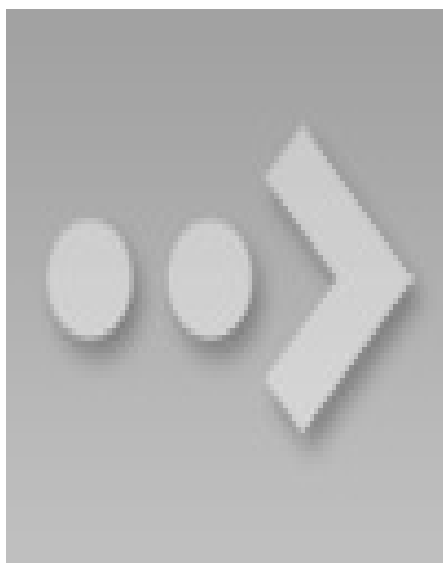
However, a spreadsheet or text table view of data is not a data visualization. In my opinion, putting one or more of these text tables in one place does not make for a dashboard. To me, dashboards and data visualizations help users interpret data by leveraging the power of visualization. Further, once there is more than just *one* number in a report, I firmly believe that a text table is the worst possible choice for communicating the data.

Consider this example of just two numbers:

23.4K vs. 3.42M

Even when the numbers are in units, you can likely tell that the first number is smaller than the second number, but it is challenging to consider the scale of the difference. Take a look at the same two numbers visualized as a bar chart:

[Insert Image 43-1]



The values can still be included as labels on the bars if the end users need those

The values can still be included as labels on the bars if the end users need those in order to sleep at night, but now they have a much better insight into what is happening with the numbers. Not only is the bar chart easier to interpret than the raw numbers, it is more memorable, and thus, effective.

I am not saying that raw numbers provide no value, but I am saying that visualizing raw numbers always provides additional value.

Are text tables or crosstabs ever the best choice?

If I am ever critical, I try to provide either a better solution to what I'm criticizing and/or a way the thing I'm criticizing can still be used. In this case, I'm providing the latter, with three ways to use crosstab views in Tableau.

1. Crosstabs can be used to export raw data

Sometimes there are legitimate reasons to get to the raw data. If an end user is more comfortable using a tool like Excel and wants to add their own calculations, for example, it would be nice to provide the data in an Excel-ready format. Fortunately, any view in Tableau can easily be duplicated as a crosstab. The crosstab view can then be opened in Excel. Here are the steps:

- From any view in Tableau, hover over 'Worksheet' in the top navigation and select 'Duplicate as Crosstab'
- Once the crosstab view is generated, hover over 'Worksheet' again, hover over 'Export', and choose 'Crosstab to Excel'

2. Crosstabs can be used to create "Callout Numbers"

In the [data-driven storytelling section](#), I encourage data designers to use "callout numbers" to clearly communicate the most important points of the story. These numbers are standalone, which makes them descriptive in nature (typically with no prescriptive value), but they can help engage the viewer so they can immediately see what KPIs are prioritized by the designer. From here, hopefully the end user will continue exploring your dashboard so they can help answer why a callout number is the value that it is.

Callout numbers are created by making a crosstab view, filtering it down to one

number, and formatting the number so that is oversized (think a minimum of 36-point font). These numbers are then typically placed along the top or left side of a dashboard to provide a natural starting point of an analysis.

3. Crosstabs can be used to create dashboard filters or navigation

One clever application of text table views in Tableau is to use the view as a dashboard filter or navigation. This approach has an added advantage of making your dashboard process more efficiently.

For example, if you wanted to create a dashboard filter for Region out of a crosstab, you would first make a crosstab view that displayed the four regions: North, East, South, and West. You would then place this view on a dashboard and add a dashboard action that filters the rest of the dashboard when a Region selection is made.

Not only does this type of dashboard action process more quickly than a filter in Tableau, but it can be used for other purposes. Perhaps you want to add a dashboard action to the table view that takes end users to interior pages of your workbook or even conducts an online search.

For more dashboard action ideas, see the chapter, 3 Creative Ways to Use Dashboard Actions in Tableau.

Chapter 44 – How to Make a Highlight Table

In the data-driven storytelling section, I demonstrate an exercise for “Smoothing the Excel Transition” by introducing the value of data visualization. In the example, we simply encode the values in a large text table of data by color - the higher the sales values, the darker the green. The result is called a highlight table.

Highlight tables are one of the simplest chart types to create, but are also among the most powerful. When compared to a crosstab (aka a text table) view, this

the most powerful. When compared to a crosstab (a.k.a. text table) view, this basic data visualization helps reduce the time to insight and improve the accuracy of insights. Highlight tables are definitely worth making a part of your chart type toolbox, right alongside bar charts and line graphs. They are best for:

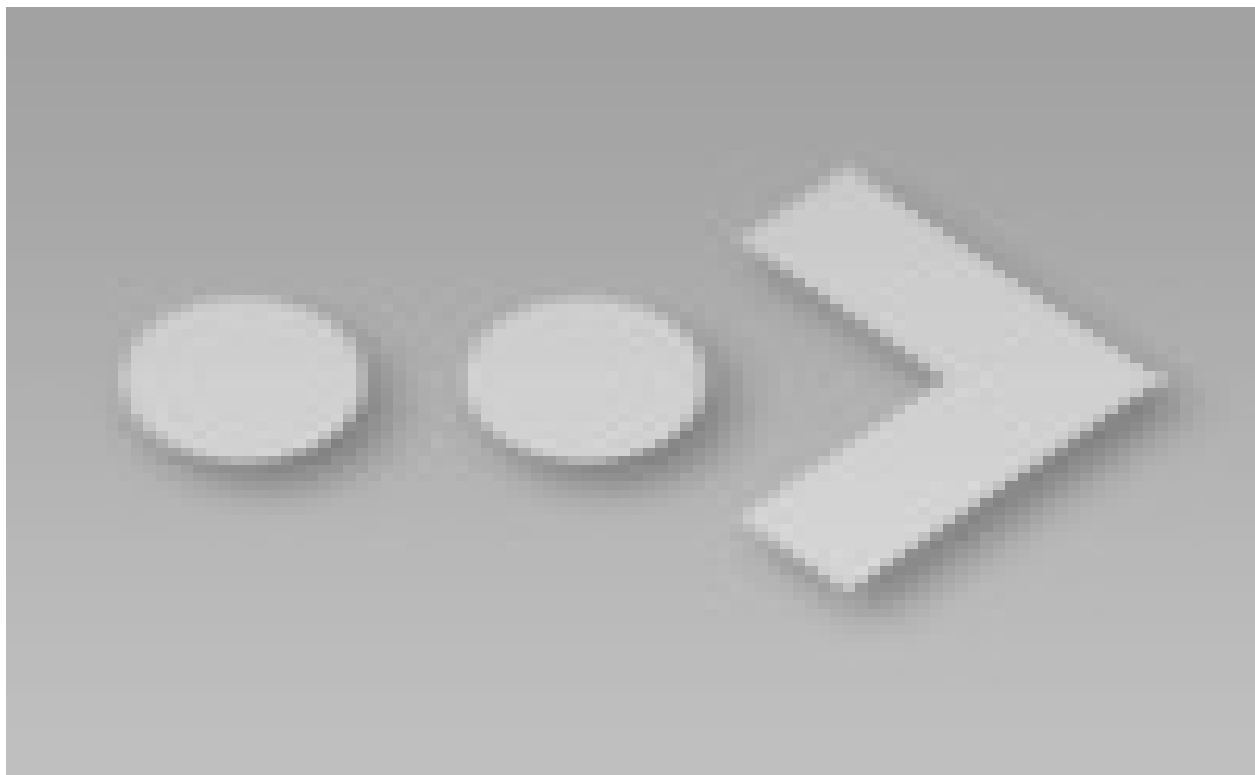
- Quickly identifying highs and lows or other points of interest in your data.
- A means of enhancing a crosstab.
- A tactic for helping smooth the Excel transition.

Highlight tables consist of one or more dimensions and exactly one measure (the color). Here's how they're made:

How to make a highlight table in Tableau

Before we begin, let's see how the final product will look:

[Insert Image 44-1]

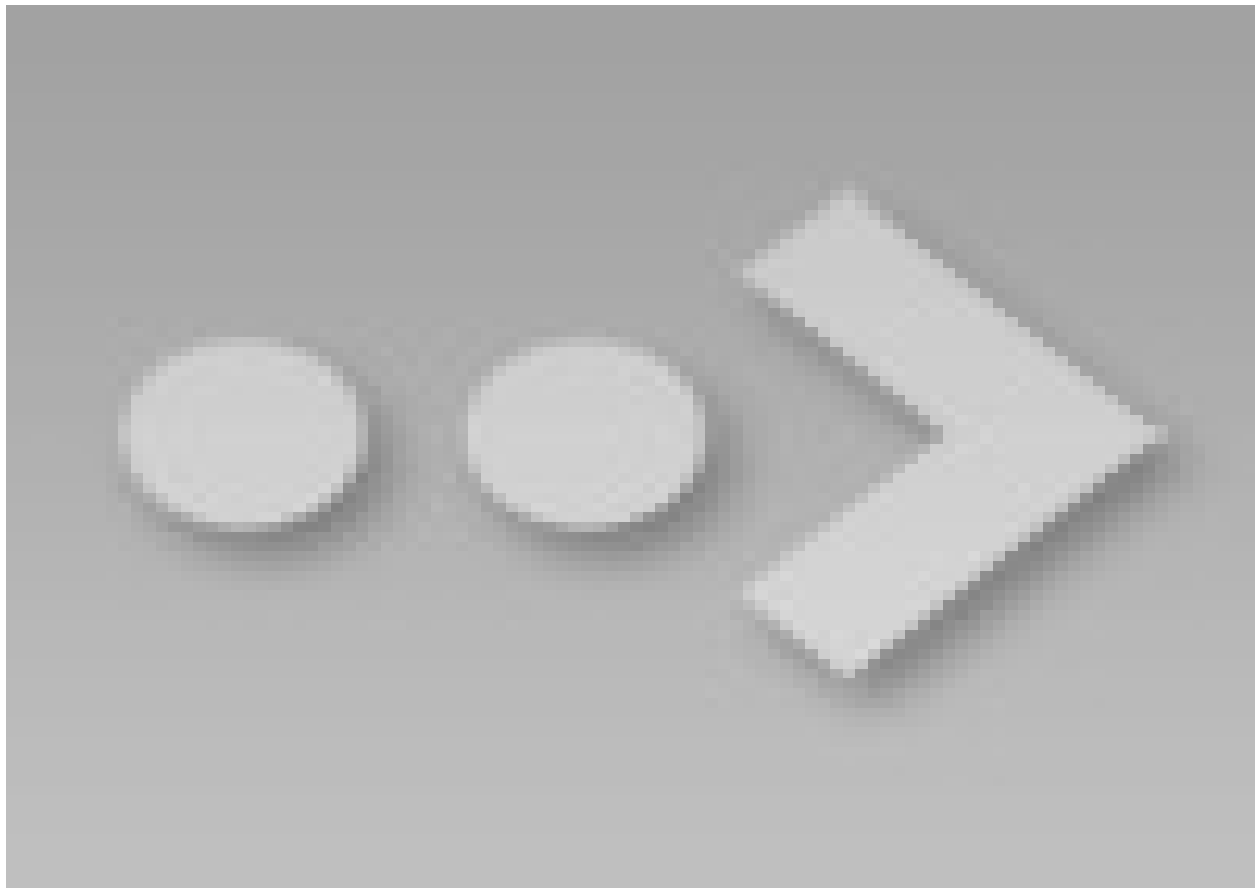


You can create a highlight table with any combination of one or more dimensions and one measure, but if you would like to follow along, the following example will use the Sample – Superstore data that comes with Tableau. You can also create a highlight table using Show Me, but I prefer to learn by creating chart types manually so I understand how Tableau is generating the data visualization.

Step 1 – Make a Crosstab of Data

First, make a basic text table. In my example, I'm looking at the measure of Sales by the dimensions of Sub-Category and Month of Order Date (Discrete). At this point, my view looks like this:

[Insert Image 44-2]

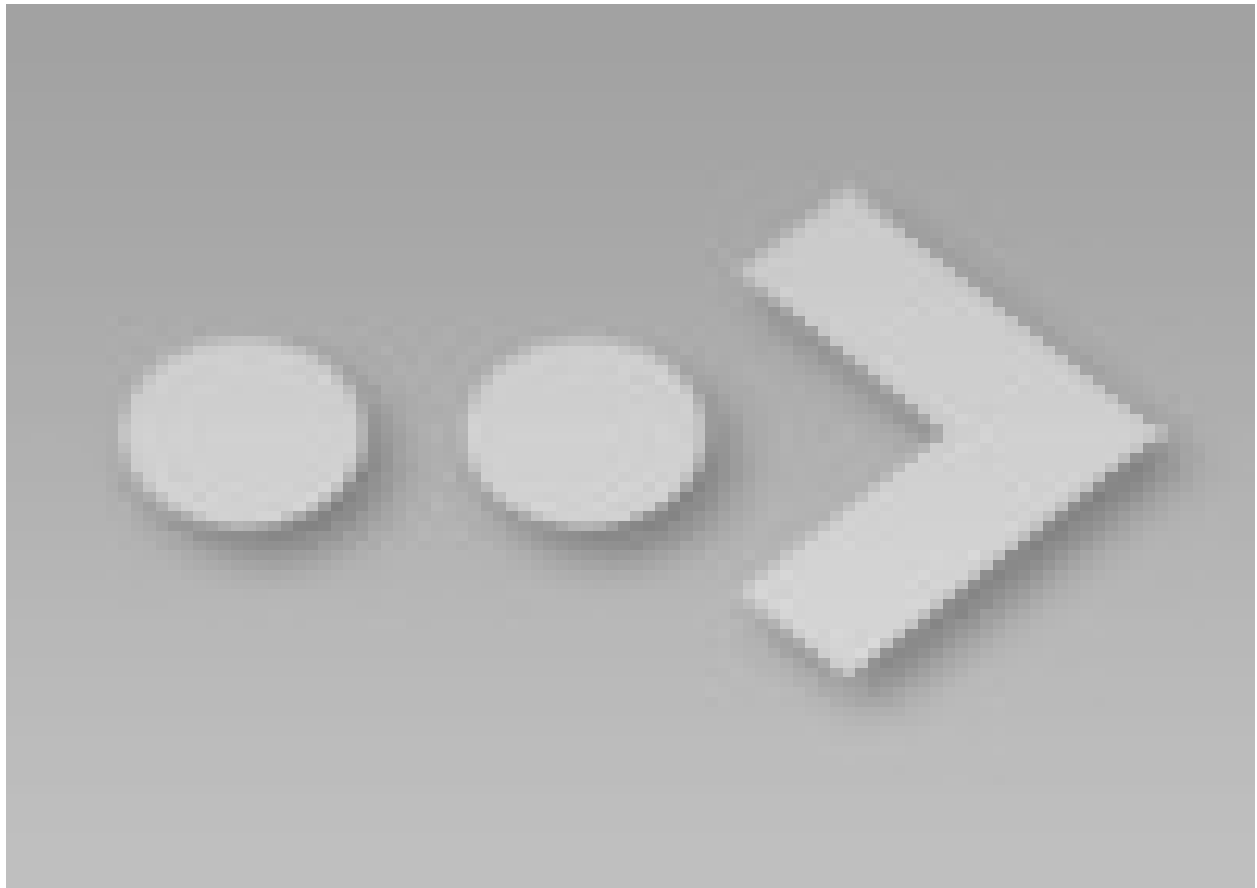


Step 2 – Color the Numbers by Sales

- - - - -

Second, color the displayed sales values in the text table by dragging the sales measure from the list of measures to the Color Marks Card. Note that the Sales measure is now on the marks shelf twice: once for text and once for color. At this point, my view looks like this:

[Insert Image 44-3]



Step 3 – Create a Highlight Table by Changing the Mark Type

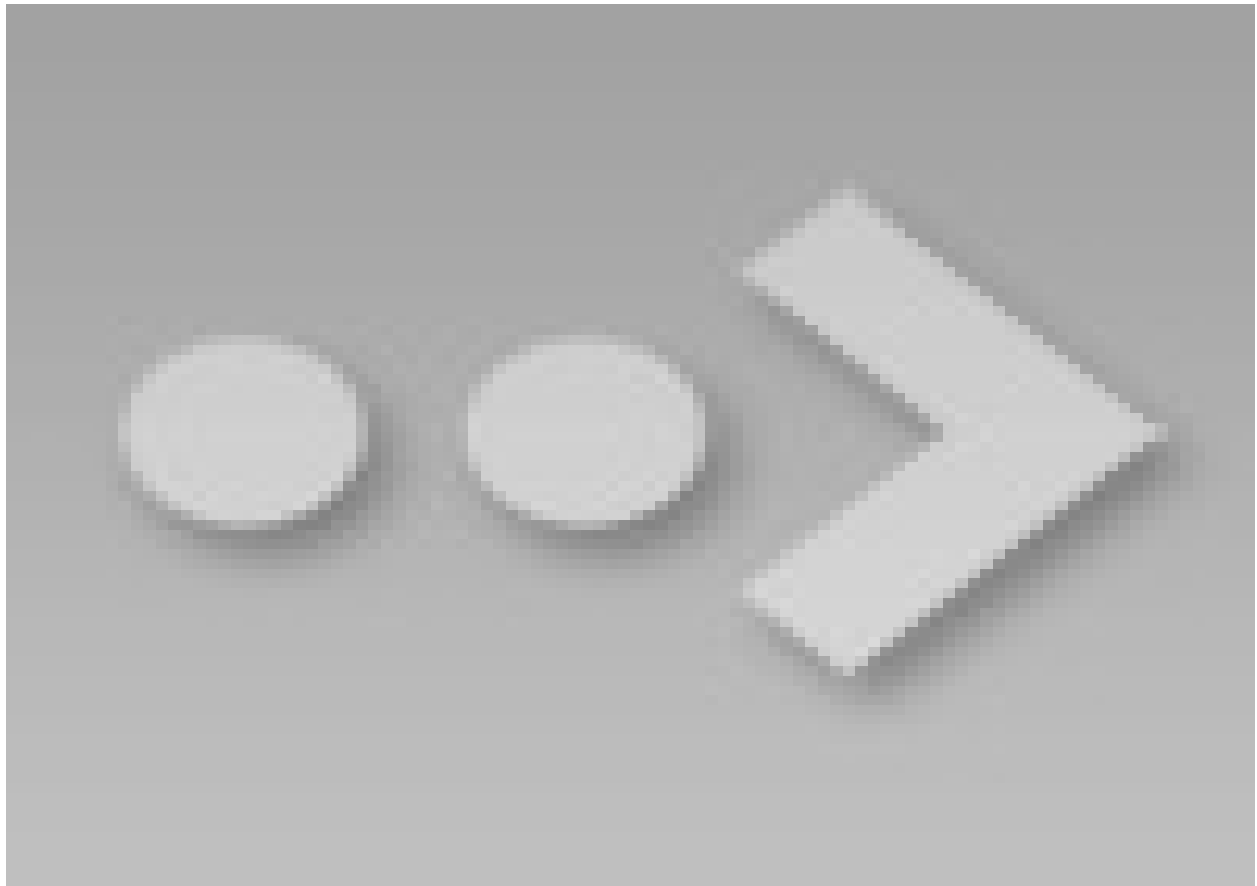
While more helpful than a raw table of numbers, the view we have created so far can still be improved. While the numbers themselves are encoded by color, which helps us recognize highs and lows in the data more quickly than we could with a crosstab, there is a lot of white space around each number.

The final step in converting this view to a highlight table is to simply change the mark type from Automatic (which at this point is Text) to Square. When you convert the mark type from Text to Square in a crosstab view, the “Square” mark

type fills each cell, creating a highlight table. When building text tables, I also like to add a white border around each cell; this can be accomplished by clicking on the Color Marks Card and changing the border.

My final highlight table looks like this:

[Insert Image 44-4]



Chapter 45 – How to Make a Heat Map

Heat maps are a visualization where marks on a chart are represented as colors. As the marks “heat up” due to their higher values or density of records, a more intense color is displayed. These colors can be displayed in a matrix / crosstab,

which creates a highlight table, but can also be displayed on a geographical map or even a customized image – such as a webpage used to show where users are clicking.

That being said, heat maps are defined somewhat differently in Tableau, and this chapter shares how to create a Tableau heat map. If you are interested in creating a traditional heat map using a custom image, see the chapter, *How to Make Small Multiple Stadium Maps in Tableau*.

To first get more specific about how Tableau defines heat map, let's take a look at the requirements to draw a heat map under Tableau's Show Me options.

“For heat maps try 1 or more dimensions and 1 or 2 measures”

This is very close to the requirements for drawing a highlight table with Show Me:

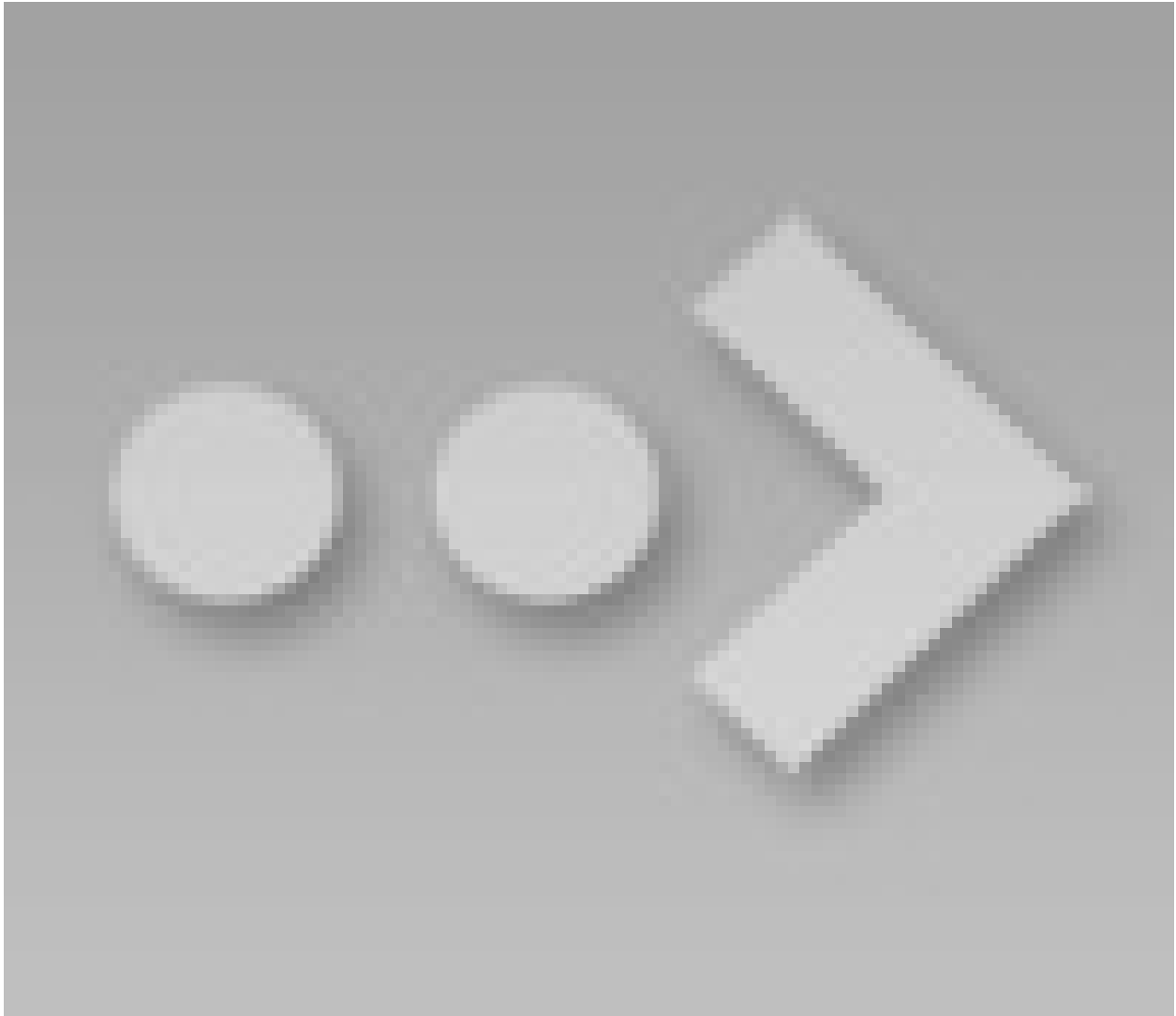
“For highlight tables try 1 or more dimensions and 1 measure”

The key distinction between the two chart types is that with a heat map, you are able to encode the marks by one additional measure. With a highlight table, you're only option is to color the marks by one measure. Since you can only color marks by one thing at a time, your encoding is limited to exactly one measure. With a heat map in Tableau, you can color the marks by one measure, but also size the marks by a second measure. Depending on your analysis, this additional encoding can add value to your visualization.

How to make a heat map in Tableau

Let's say we've been tasked with evaluating the product sub-category sales in the Sample – Superstore dataset by Month of Order Date to see if we can identify any seasonal trends in the data. The element of time (Month of Order Date) may give you the instinct to go with a line graph for this analysis, which would look like this:

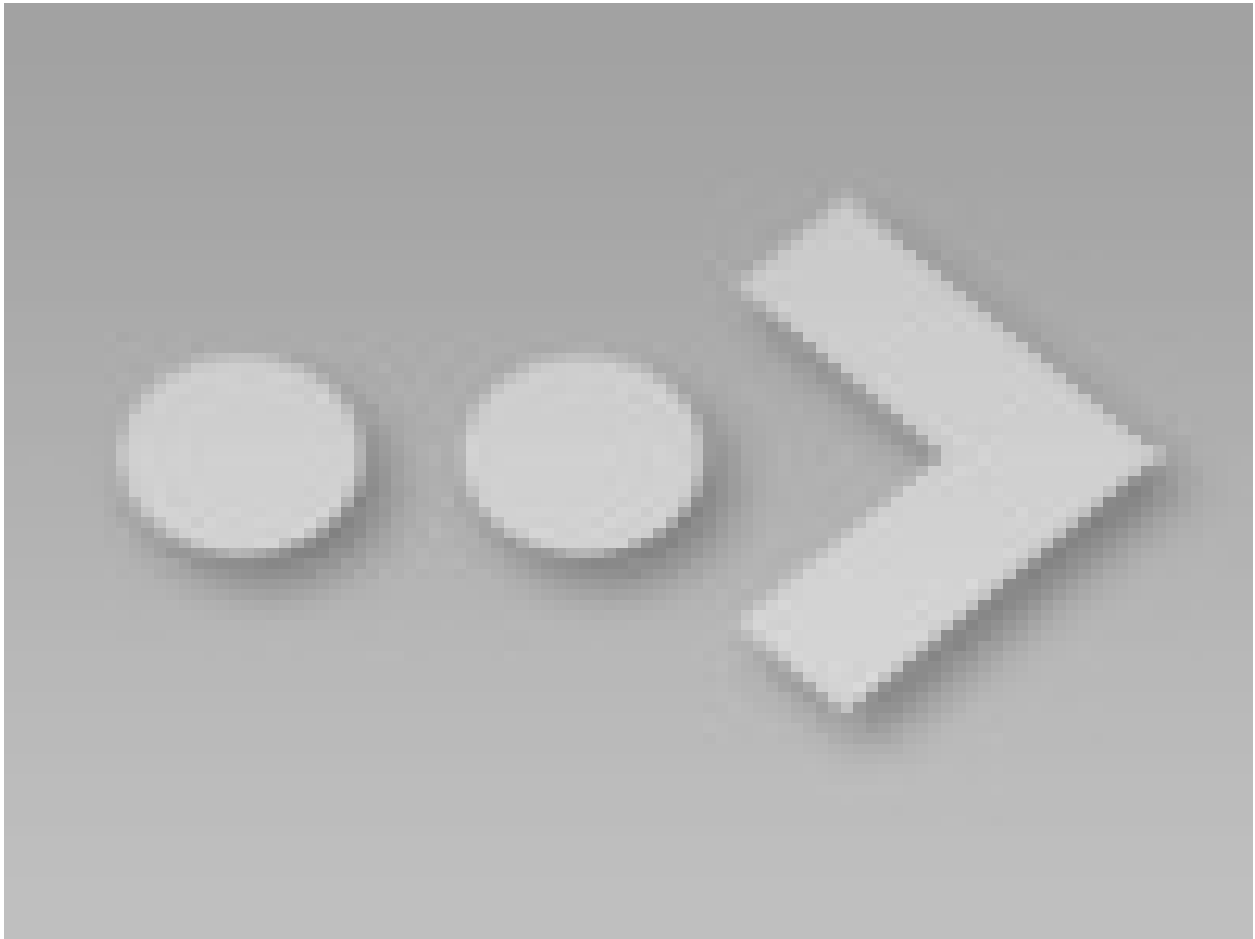
[Insert Image 45-1]



As you can see, this graph is a bit of a mess. The 17 lines are causing a lot of overlap and several of the sub-categories at the bottom are on a much smaller scale than the rest, making it challenging to gain insights. In this case, a heat map may be a better option.

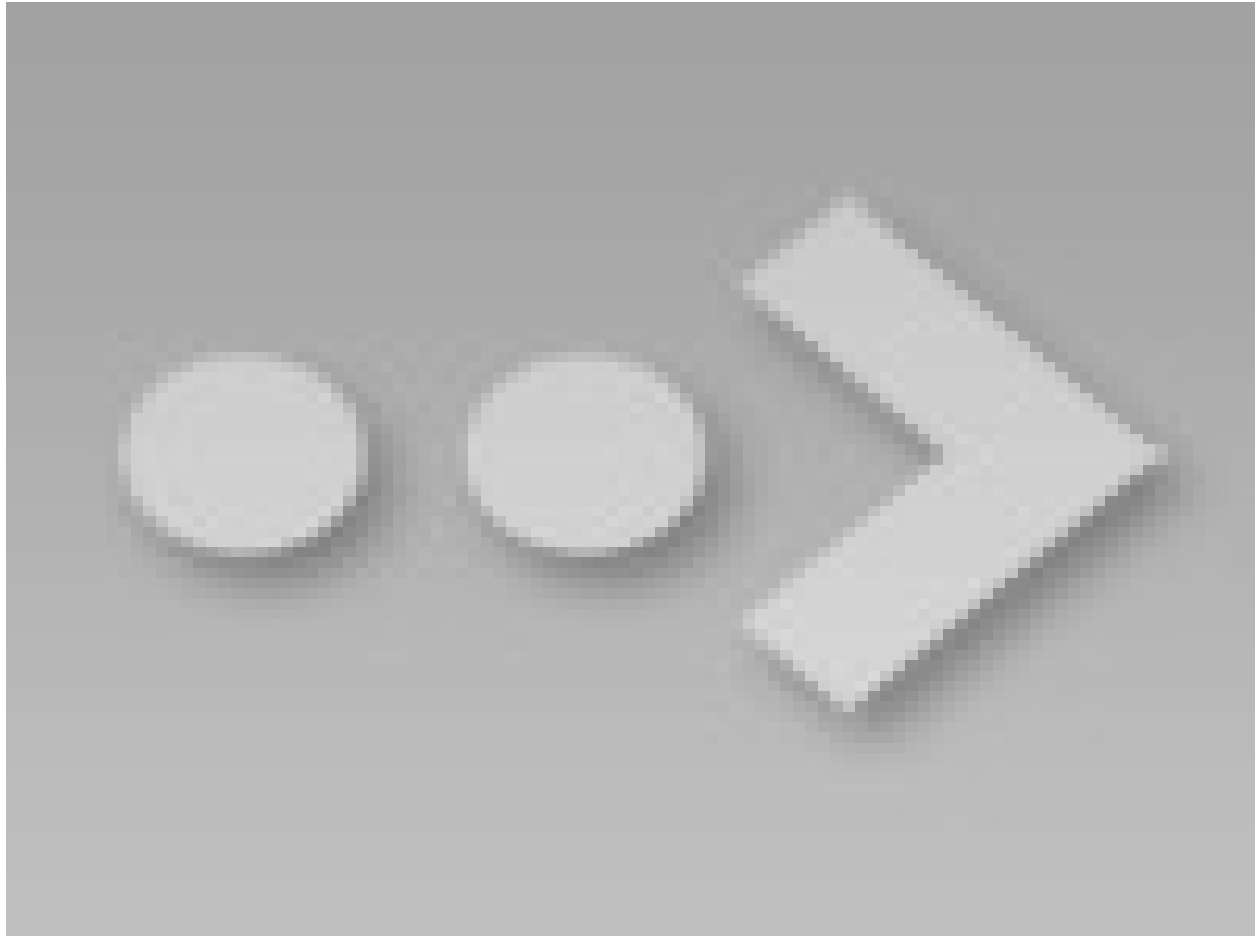
To create a heat map in Tableau, start by laying out the rows and columns which will serve as the grid for the visualization. I would like the months in this analysis to be listed along the top of the view. Since the months will create columns, I know that I should put the Month of Order Date dimension on the Columns Shelf. Conversely, I would like each sub-category to have its own row, so I will place that dimension on the Rows Shelf.

[Insert Image 45-2]



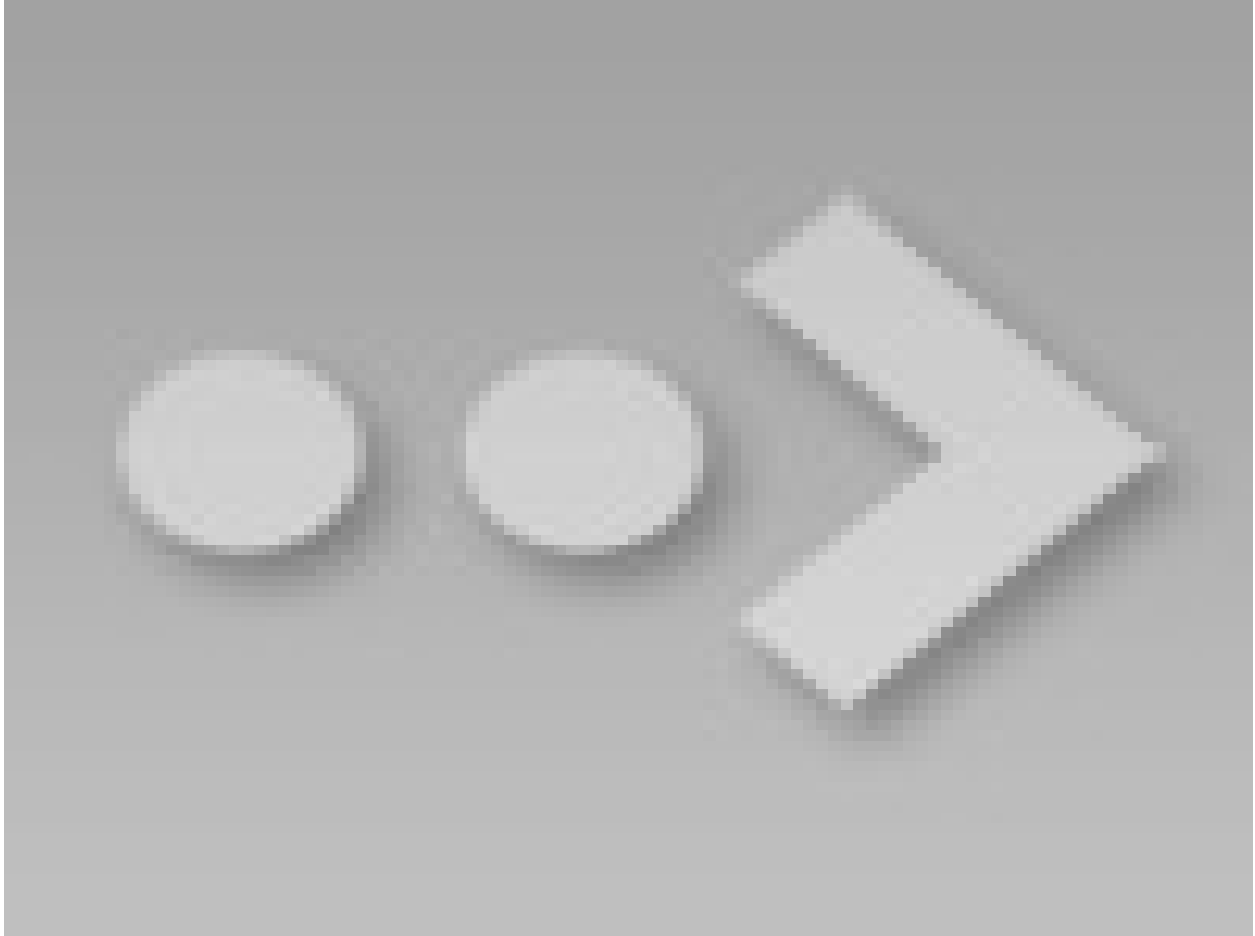
By default, the mark type is set to Text. I personally prefer my Tableau heat maps to be created with circles, so I will change the mark type to circle to lay the foundation for the view. The Shape or Square mark types are also good choices.

[Insert Image 45-3]



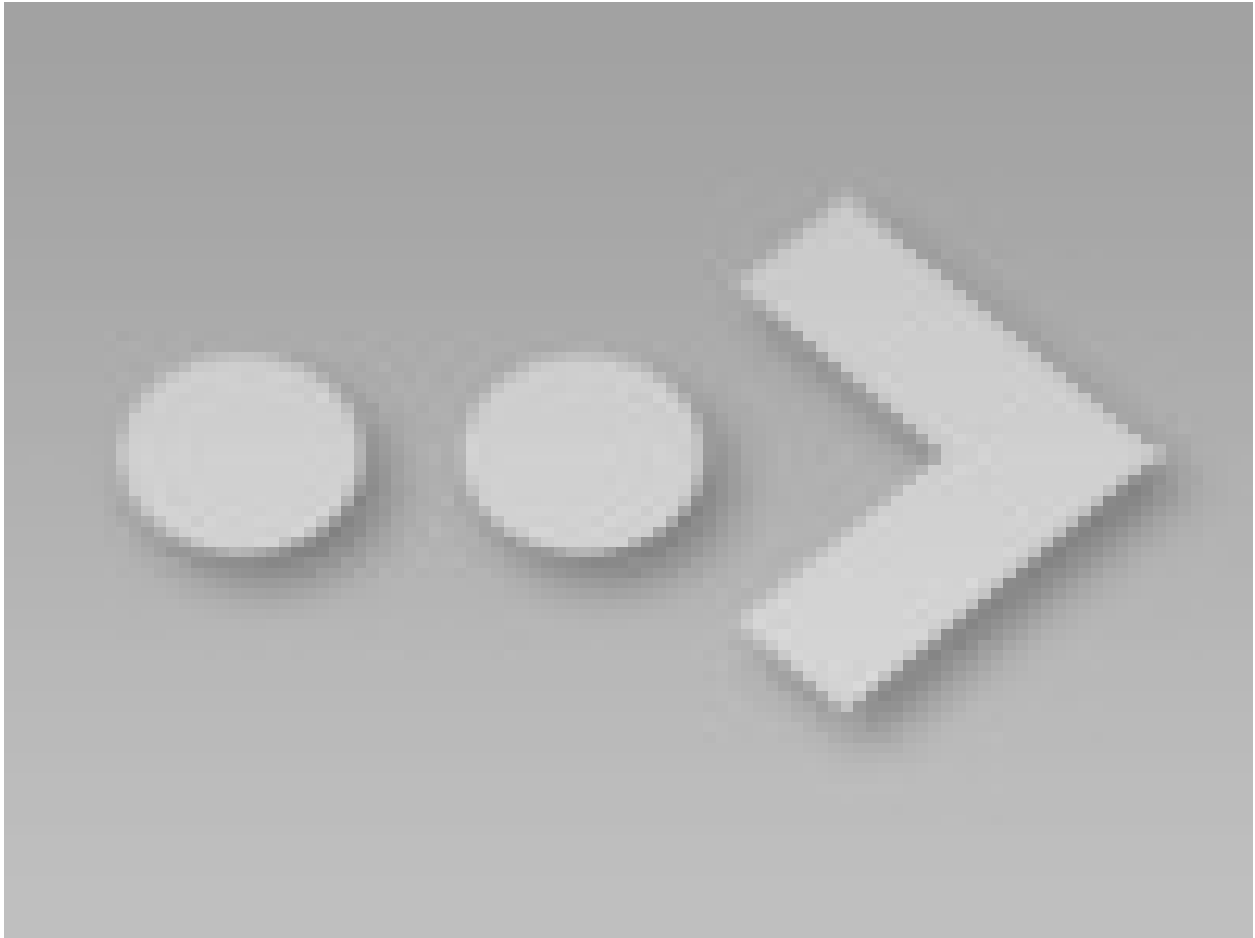
Now that I have a mark at each intersection of Sub-Category and Month of Order Date, I can encode them by two measures; one which will determine the size of the marks and one which will determine the color intensity of the marks. This encoding is produced by placing the measures we want to encode the marks by onto the Size Marks Card and Color Marks Card, respectively. For my first analysis, I will size and color the circles by the same measure: Sales.

[Insert Image 45-4]



This visualization uses the exact same fields as the line graph above, but it is now much easier to compare sub-categories within a specific month (reading the chart vertically) or compare the seasonality across each sub-category (reading the chart horizontally). The “double-encoding”, where the size and color are both based on the same field, are meant to help the insights ‘pop’. However, you have the option to use one measure for the size, and a different measure for the color. For example, here is what the heat map looks like if I size the circles by the Quantity measure instead of sales.

[Insert Image 45-5]



With this analysis, large and light circles would mean that a sub-category sold a relatively large quantity but made a relatively low amount of revenue: low sales per item. Conversely, small and dark circles would mean the sub-category sold a relatively small quantity, but generated a relatively high revenue: high sales per item.

Beware that this type of mixed encoding can be confusing for end users unless you explicitly state what the size and color represents. Despite some of their limitations, Tableau heat maps provide a viable alternative to a line graph or highlight table if you need to compare dimension members with varying scales across multiple measures.

Dual-axis combination charts, or Combo Charts, are an effective chart type for showing related information while saving real estate by combining views. This chart type is created with one shared axis, such as an X-axis for date, and two separate axes, such as Y-axes for two different measures. This chapter will show you how to make a dual-axis combo chart in Tableau which looks at Sales and Discount by Year and Product Category.

How to make a dual-axis combo chart in Tableau

Before we begin, let's take a look at how our final product will look:

[Insert Image 46-1]



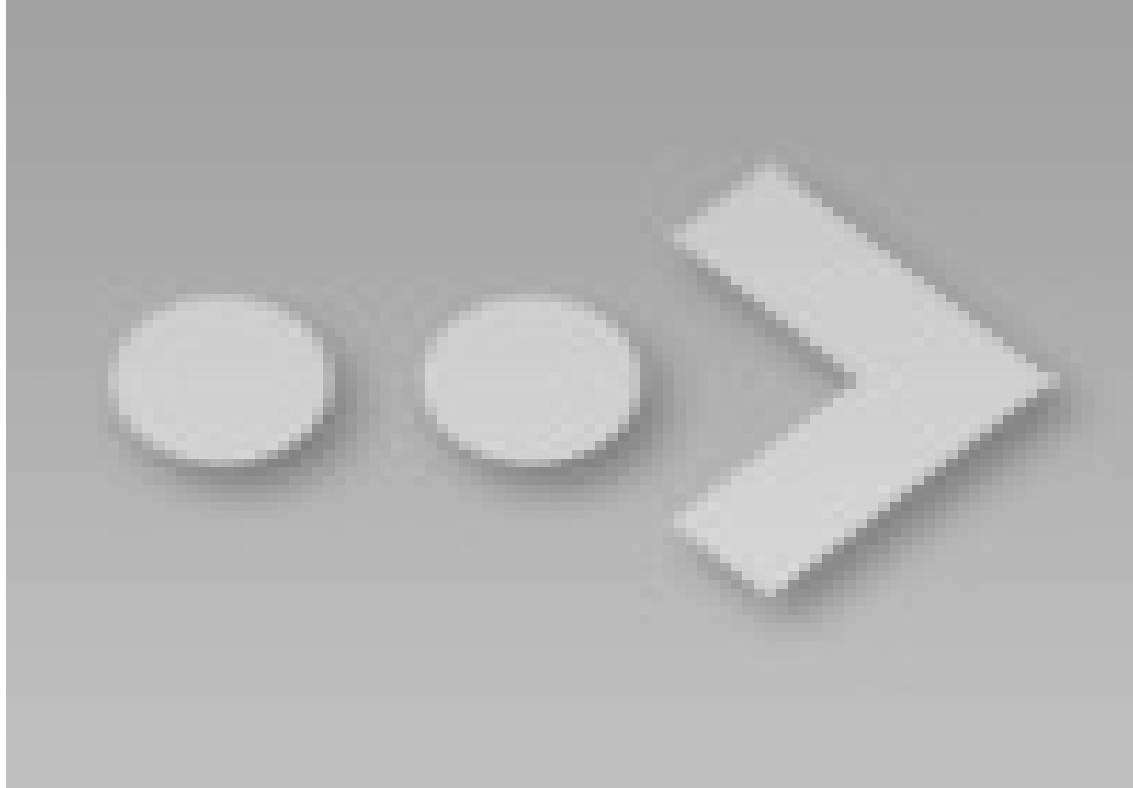
If you would like to follow along, I'm using the Sample – Superstore dataset that comes with every download of Tableau.

Step 1 – Make a graph for one of the measures

The first step is to make a graph for one of your measures. It doesn't matter too much which of your two measures you begin with, so let's make a line graph for Sales by year. To create the first graph, drag the Order Date field to Columns

Sales by year. To create the first graph, drag the Order Date field to Columns Shelf with an aggregation of Year Continuous. Then drag the Sales field to the Rows Shelf. At this point, the view looks like this:

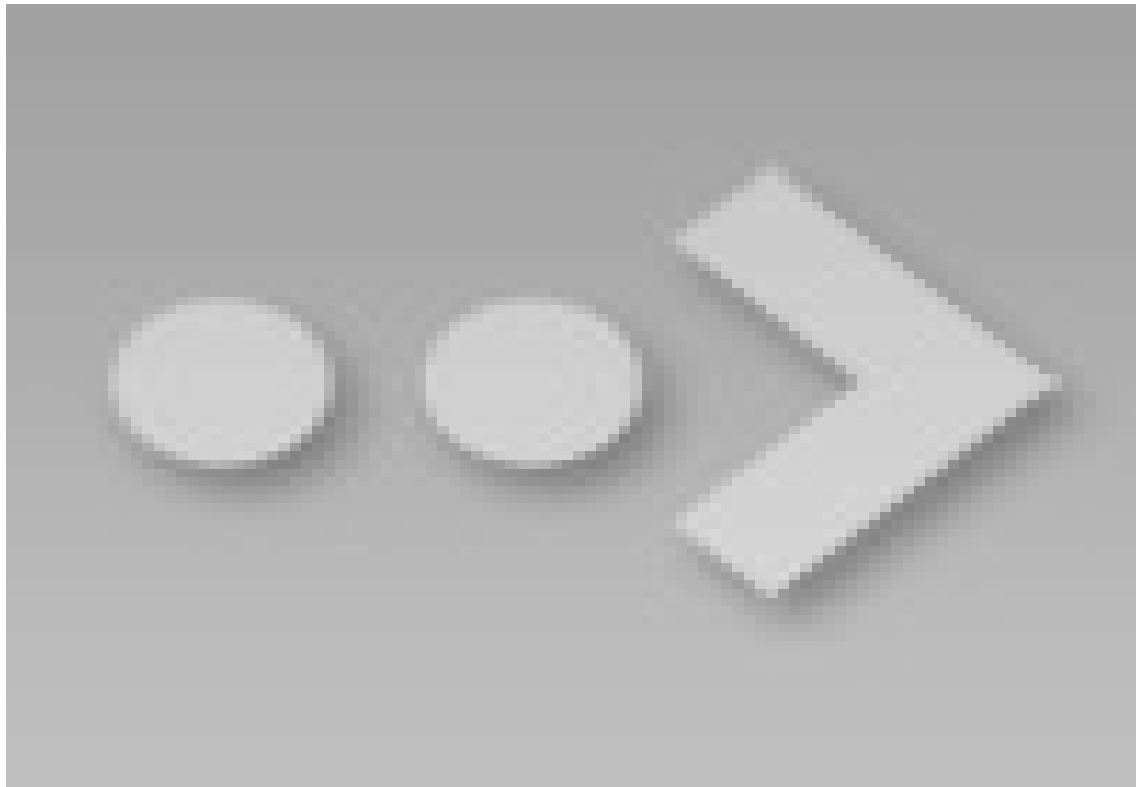
[Insert Image 46-2]



Step 2 – Drag the second measure onto the opposite axis

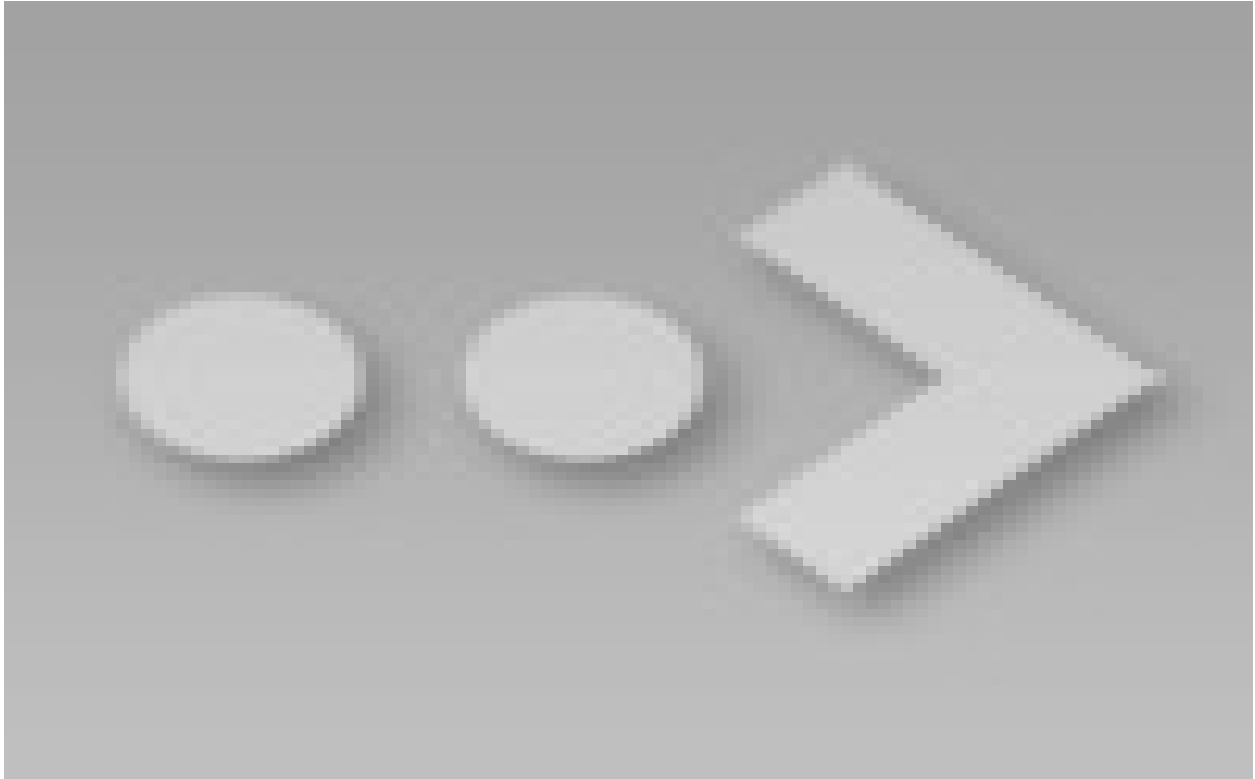
In order to create a dual-axis chart, you will drag the second measure onto the opposite axis from your first measure. In my case, Sales is on the left Y-Axis. I want to add the Discount measure to the view, so I will drag it to the right Y-Axis. As I get close to the opposite axis, Tableau is giving me a hint of where I can drop the measure as indicated by a dashed line:

[Insert Image 46-3]



Once I drop the field on the opposite axis, Tableau generates a dual-axis chart. Note that this is a dual-axis chart at this point, but not a dual-axis *combination* chart. The name combination chart comes from using a combination of mark types, and so far, we only have one mark type (line).

[Insert Image 46-4]

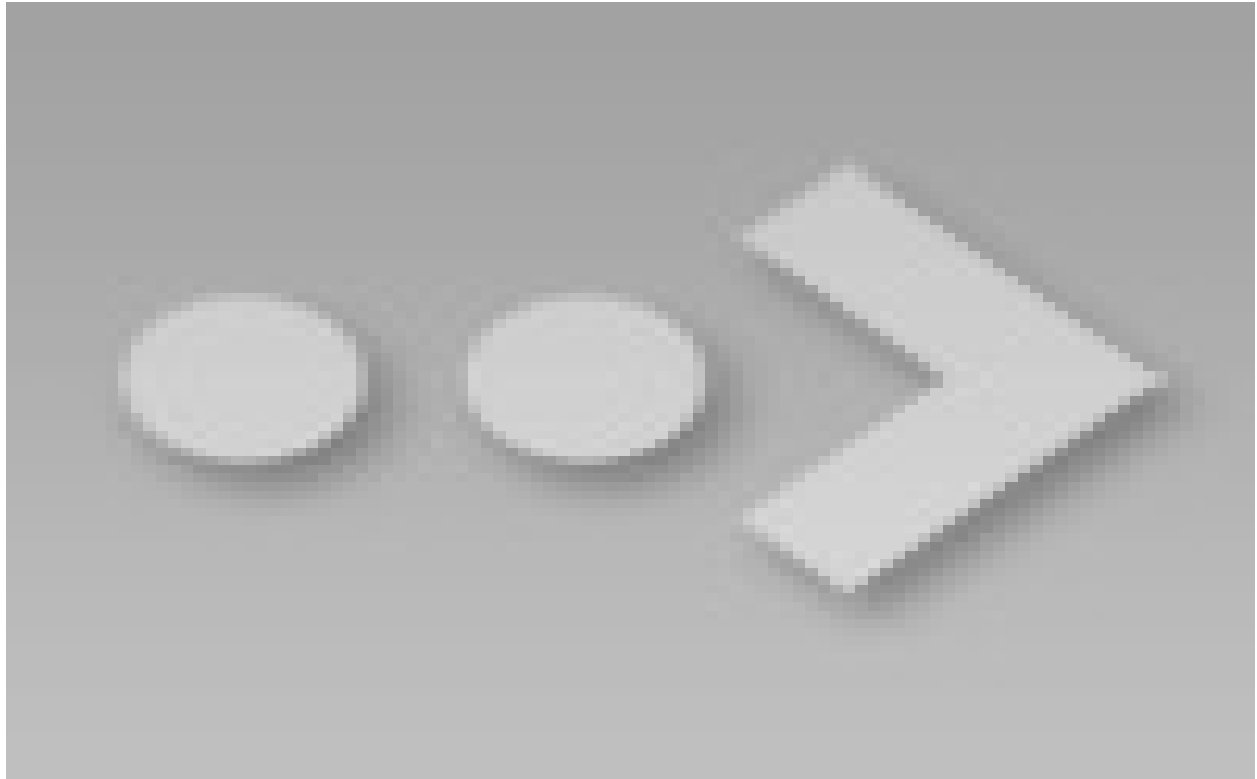


Step 3 – Create a dual-axis combination chart by changing one of the mark types

When the second measure was dropped onto the view, not only did the field appear on the rows shelf, but a new marks shelf was generated. The marks for each measure can now be edited independently of each other, allowing you to display a *combination* of mark types on your view.

Let's change the mark type of the Sales field to bar and leave the mark type for Discount as line. This is accomplished by clicking on the Marks Shelf for SUM(Sales) and changing the mark type from Automatic to Bar. At this point, the view looks like this:

[Insert Image 46-5]



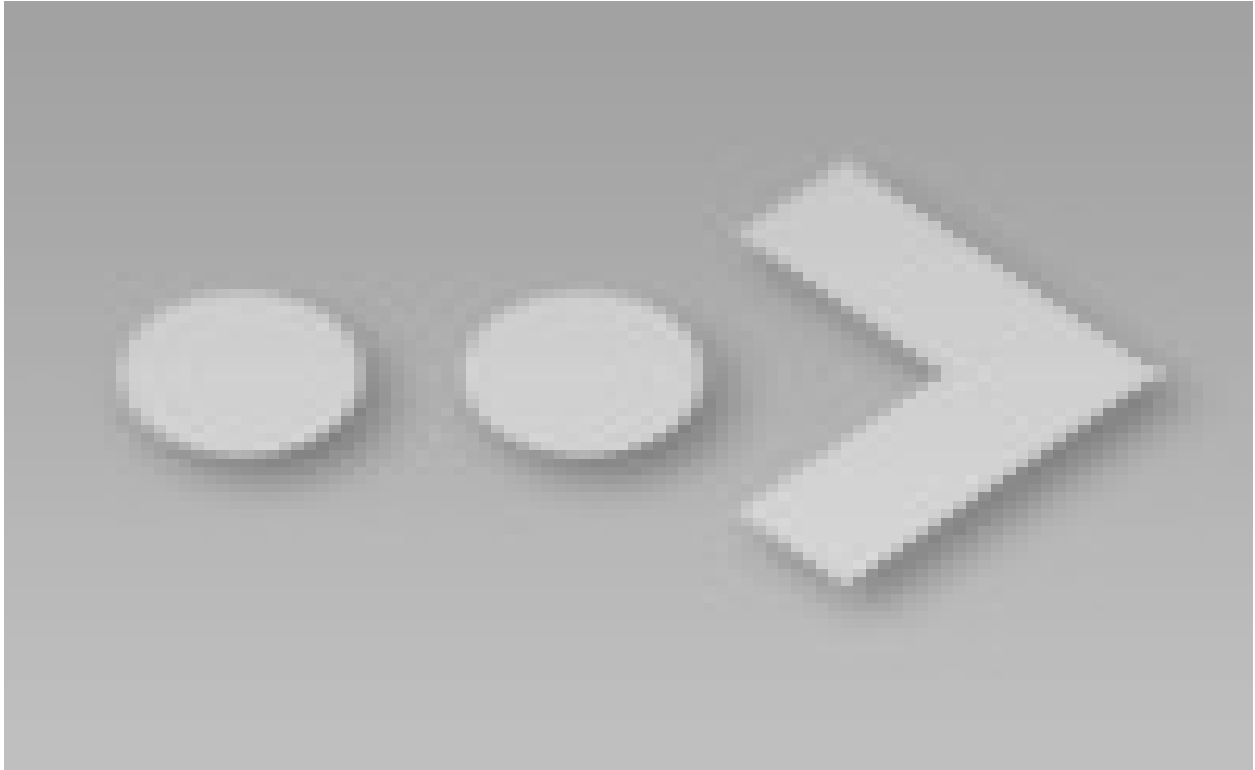
Some Additional Thoughts

- I decided to make the first graph, Sales by *continuous* Year, because it was meant to be a continuous (i.e. chronological) trend over time. Many people ask why the bars are so skinny when year is continuous. You can get a better look by changing the Year field from continuous to discrete by right-clicking on the field on the Columns Shelf and choosing “Discrete”. The catch with discrete date dimensions is that the dates can now be sorted out of chronological order. If you decide to make this change for formatting purposes, be careful not to sort the view in an unintended way.
- A mark type of line should only be used when connecting dots between continuous dates. This is because a line implies that there is a chronological relationship between points. This works well for our example, where discount is being connected by years in chronological order. However, dual-axis combination charts do not necessarily need to use a date as the shared axis, so choose your mark types wisely.
- You can create small multiples, or comparison views, by dragging another

dimension onto the Rows Shelf or Columns Shelf.

Here is my final view after changing Year to Discrete, adding the Category dimension to the Columns Shelf, and cleaning up the formatting:

[Insert Image 46-6]



Chapter 47 – How to Make a Scatter Plot

After the bar chart and line graph, I find the scatter plot to be one of the most effective visualization options for analyzing data. A scatter plot displays data points at their respective intersections of two measures, and displays every data point on the same view. The marks can then be further encoded by up to two additional measures and/or dimensions by leveraging the Color and/or Size Marks Cards.

This ability to slice and dice data points in several ways within a condensed space provides an effective means for identifying patterns. Not only that, scatter plots provide a natural way to segment the marks into four quadrants by simply adding a reference line to each of the two axes. You can even take this a step further in Tableau by creating sets for each of the four segments to use for deeper analysis later.

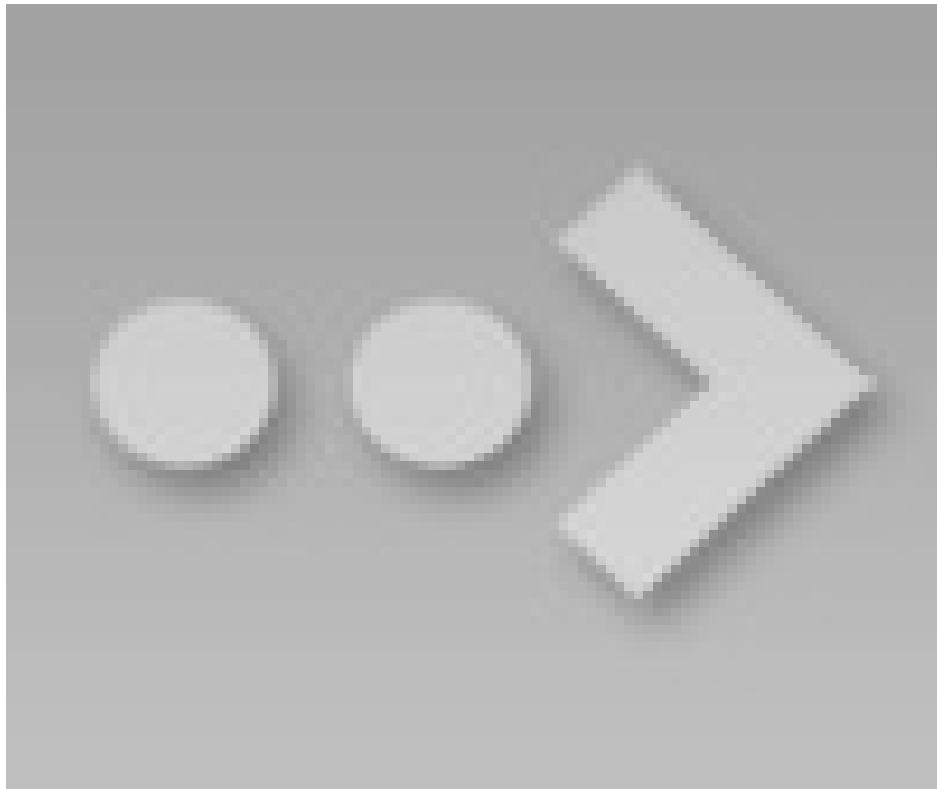
This chapter shares how to create a scatter plot in Tableau and use the results to create segments.

How to make a scatter plot in Tableau

For this walkthrough, we'll be using the Sample – Superstore dataset to evaluate all of our products across the Sales and Profit Ratio measures. When you build a scatter plot, one measure will form the Y-axis and one measure will form the X-axis. The marks on the view will then be plotted at the intersection of the values on the two axes.

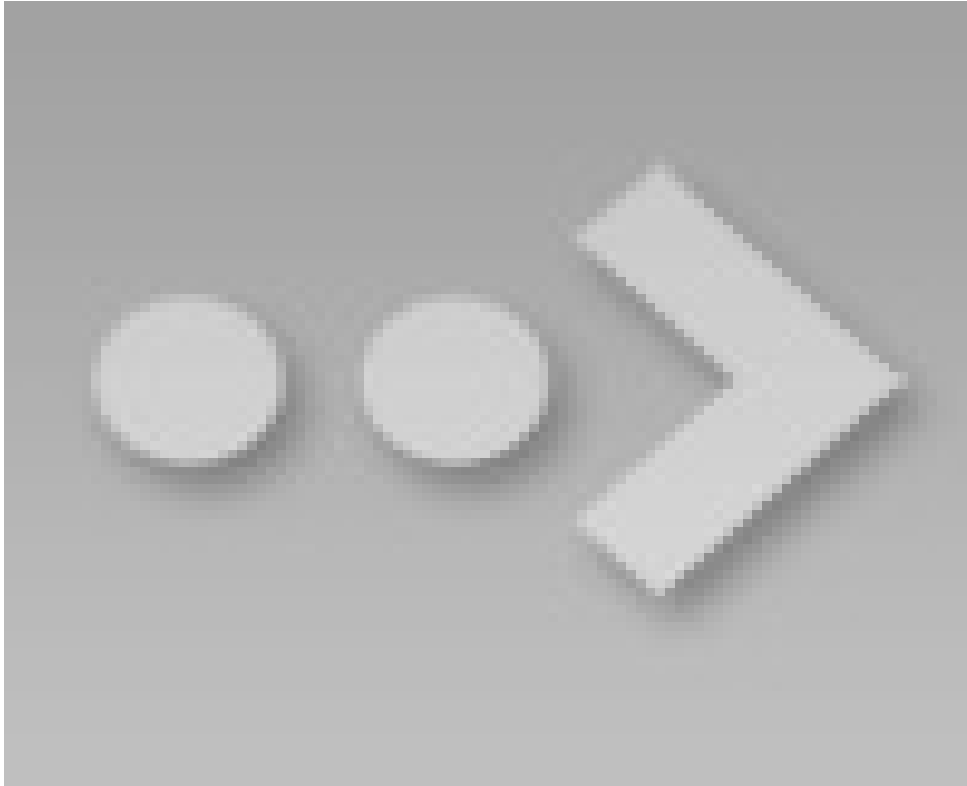
To create a scatter plot, drag and drop the Profit Ratio measure to the Rows Shelf and the Sales measure to the Columns Shelf. Scatter plot is the default chart type in Tableau when two measures are used, so you could have got to this same point by just double-clicking Profit Ratio, then double-clicking Sales to add them to the view. At this point, your view should look similar to this:

[Insert Image 47-1]



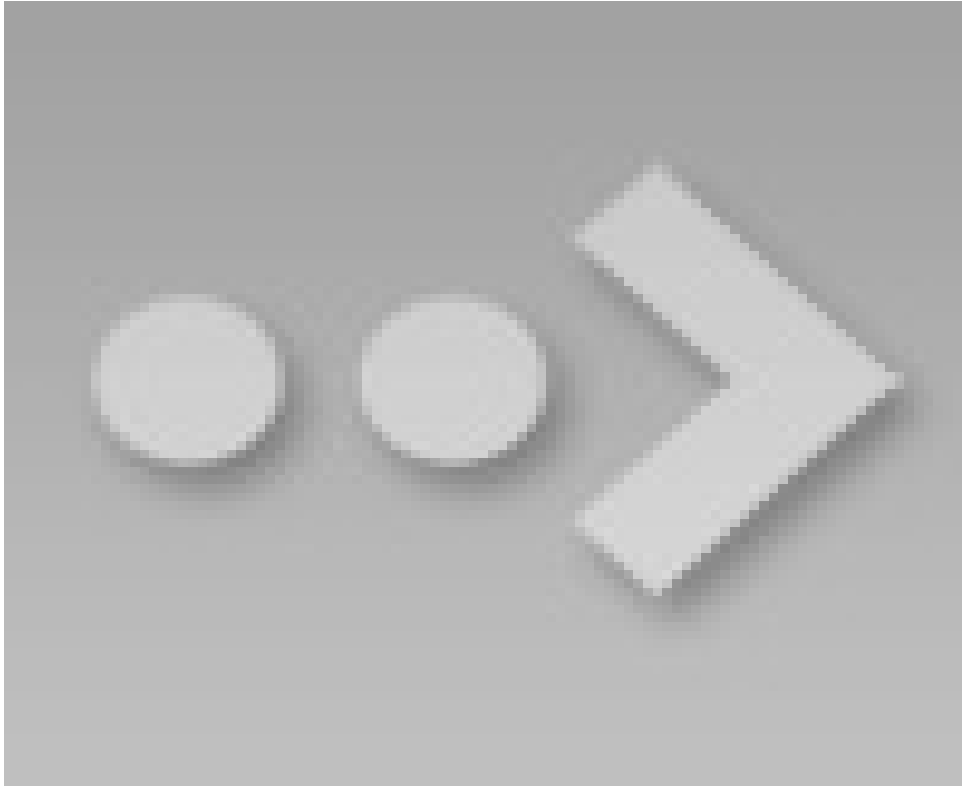
So far, we have just one point that represents the intersection of Profit Ratio and Sales for all of the records in our dataset. This is because we have yet to specify a level of detail for our analysis. For more on this topic, you can review the fundamentals chapter, Tableau Marks Cards, Encoding, and Level of Detail. Our analysis is going to look at products, so change the level of detail by dragging the Product Name dimension to the Detail Marks Card. The view has now been changed to this:

[Insert Image 47-2]



What's powerful here is that we are looking at all 1,850 of our products at once, which helps us evaluate them quickly in context of each other. You can see outliers, unprofitable products, and segments are beginning to emerge (i.e. high sales *high profit ratio*, *high sales* low profit ratio, and so on). One way to make the segments more apparent is to add reference lines to each axis. Here's what the view looks like when I add a reference line for the average of each axis by right-clicking on each axis and choosing "Add reference line":

[Insert Image 47-3]



These reference lines create four quadrants on the view that can be used to segment the data:

- Top-left quadrant: High Profit Ratio & Low Sales
- Top-right quadrant: High Profit Ratio & High Sales
- Bottom-left quadrant: Low Profit Ratio & Low Sales
- Bottom-right quadrant: Low Profit Ratio & High Sales

This provides some areas to focus on. For example, we don't want to have high sales of products that are causing us to lose money. Also, are there opportunities to increase the sales of our most profitable items? This example created segments using the averages of each measure, but you can easily change the reference lines to the thresholds of your choice for segmentation.

Lastly, these segments or products of interest can be made more permanent by placing them into sets. In the above example, our two highest-selling products are also above average in profitability. Let's put these two products into a "positive-outlier" set by selecting them on the view (either through multi-select or dragging a box around them), right-clicking, and choosing "Create set...". Once in a set, these positive outliers can be highlighted in different visualizations, which can lead to valuable insights for the business.

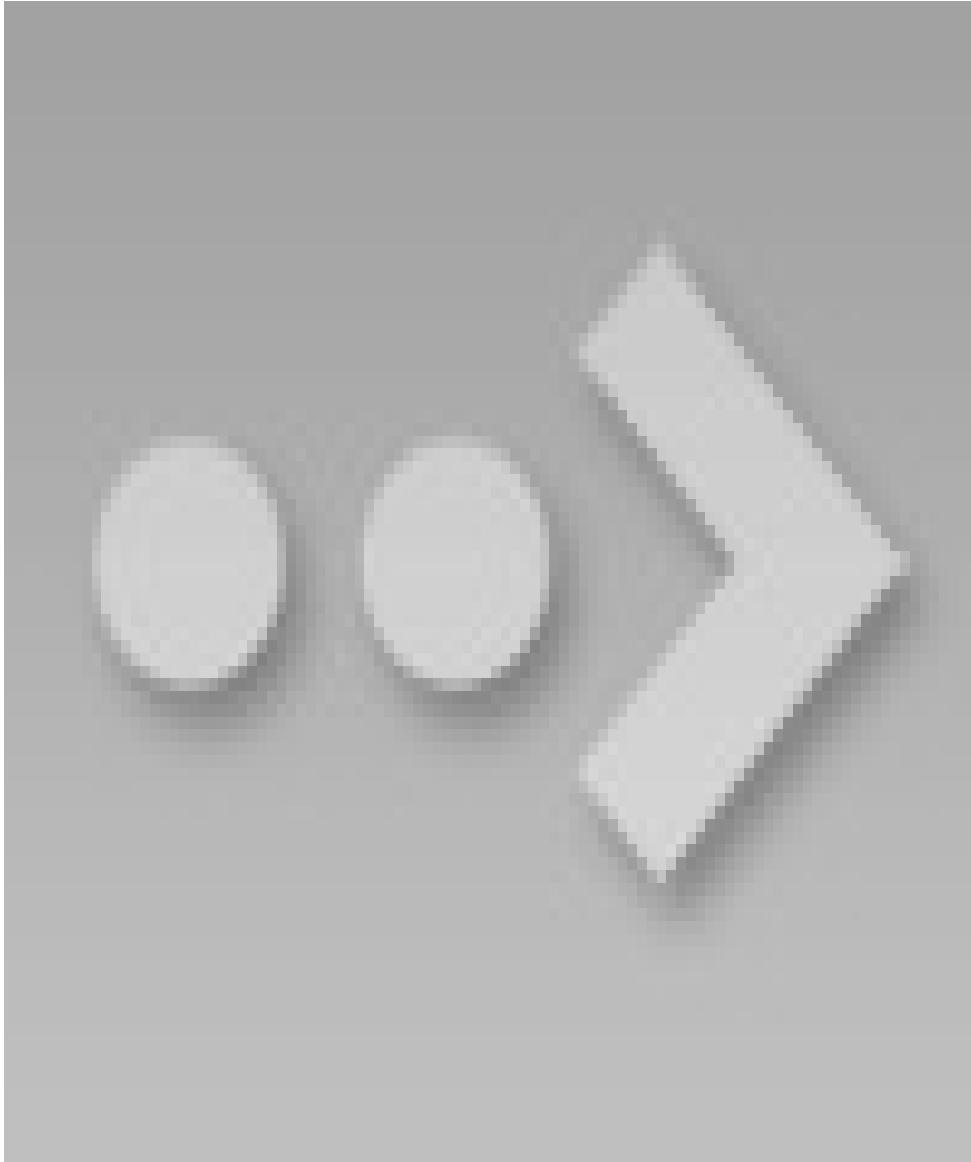
Chapter 48 – How to Make a Tree Map

A tree map is a visualization that nests rectangles in hierarchies so you can compare different dimension combinations across one or two measures (one for size; one for color) and quickly interpret their respective contributions to the whole. When used poorly, tree maps are not much more than an alternative pie chart. When used well, they provide at least two big benefits:

- Depending on the analysis, some portions of the tree map will be composed of large rectangles where additional context can be added as labels. This is beneficial when the visualization will not be interactive and you still want the written information represented.
- In addition to the scatter plot, tree maps are one of the only visualization types that allow you to reasonably communicate and consume hundreds of marks on a single view. This makes it easier to spot patterns and relationships that you would not otherwise be able to see.

For this tutorial, we will be making the following set of tree maps using the Sample – Superstore dataset:

[Insert Image 48-1]



This analysis shows not only the sales amount by sub-category, but the sales contribution of each US state per sub-category and whether or not those states were profitable within each sub-category.

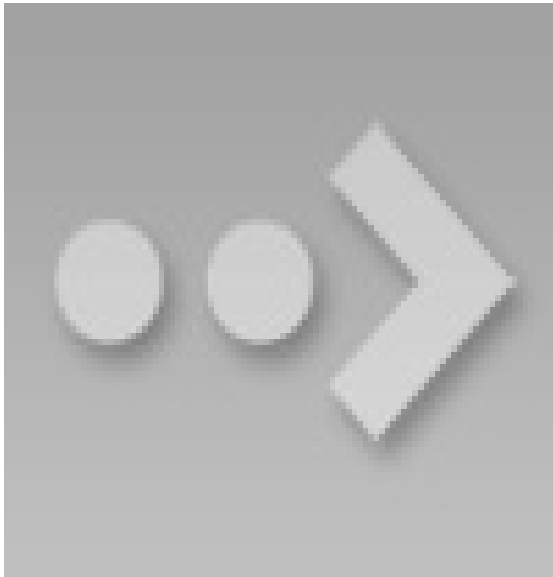
By changing the level of detail to make our analysis more granular and encoding the marks by a second measure of profit ratio provides more context to the view and helps us avoid the dreaded question, “So what?”.

How to make a tree map in Tableau

Tree maps are one of the out-of-the-box Show Me options in Tableau, but as with most charts in Tableau, I find building them from scratch helps me not only understand how they work, but also helps me get to my desired output faster.

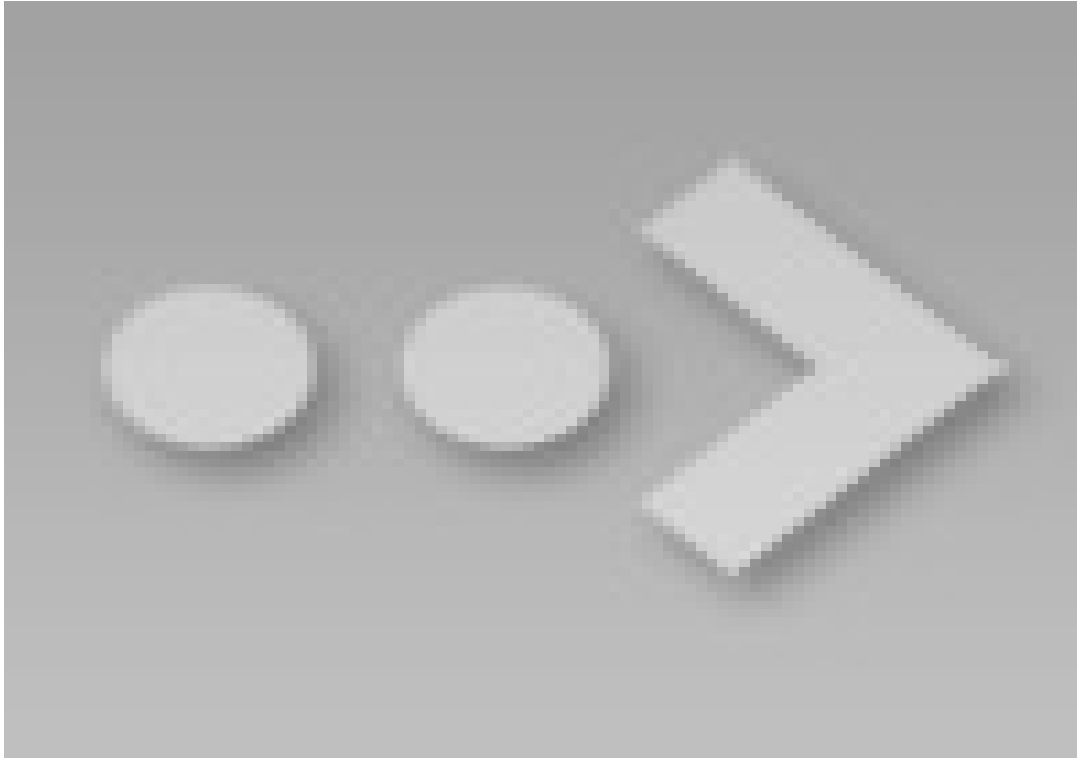
To make a tree map in Tableau, begin by changing the mark type for a view from Automatic to Square. Then drag the primary measure that you want to evaluate to the Size Marks Card and the secondary measure to the Color Marks Card. Just as it sounds, the primary measure will control the size of the squares on the view, and the secondary measure will control the color of the squares. In our case, I've dragged Sales to size and Profit Ratio to color.

[Insert Image 48-2]



At this point, we've simply laid the foundation of the tree map. There is no detail, so we just see one square colored by the overall profit ratio in the Sample – Superstore dataset. To create a square for each state, drag state to the Detail Marks Card.

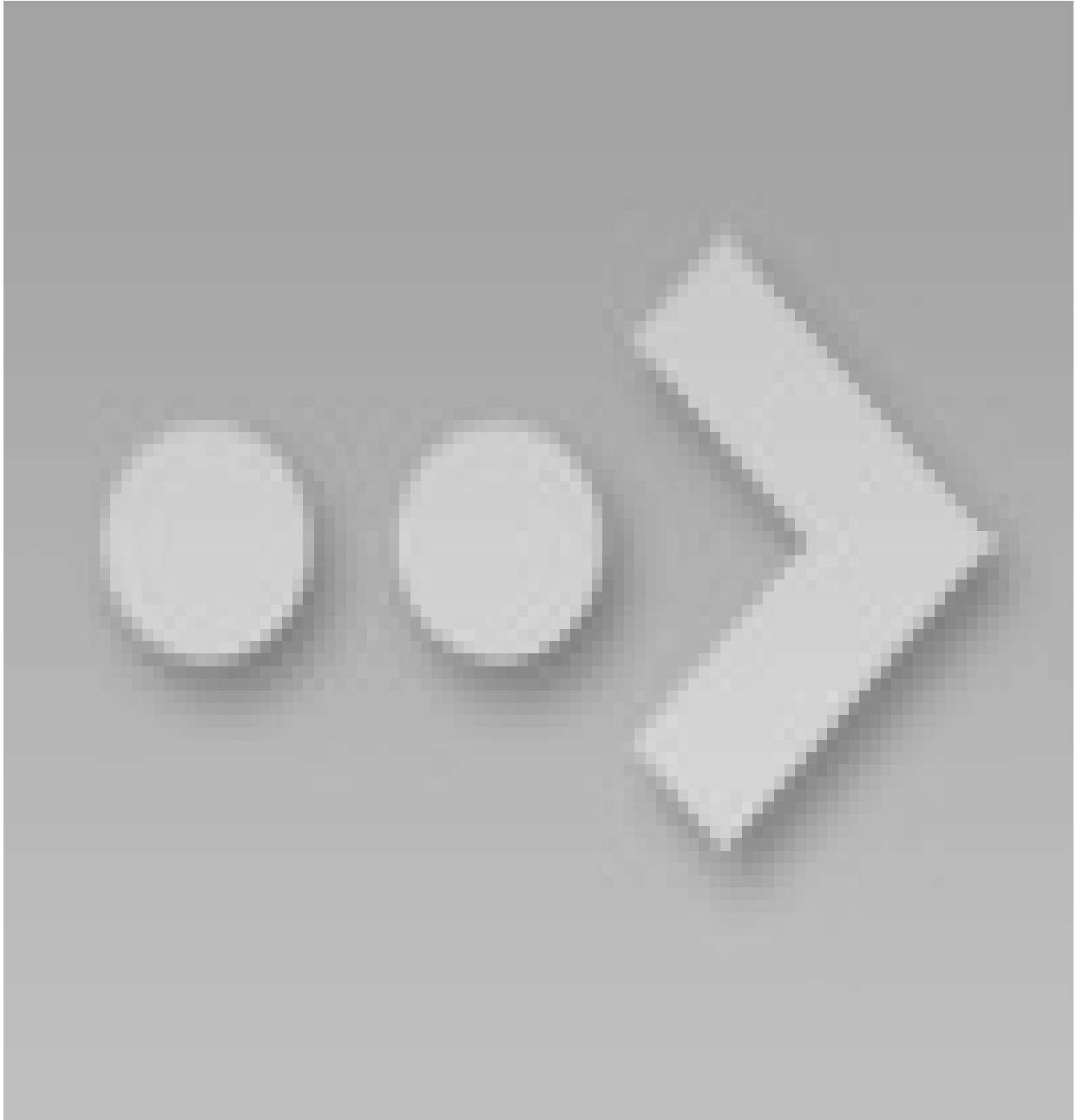
[Insert Image 48-3]



At this point, we have a tree map and a solid analysis. You can see there is quite a bit of real estate to add information to the Label Marks Card and have it displayed on the view; the first benefit mentioned in the introduction.

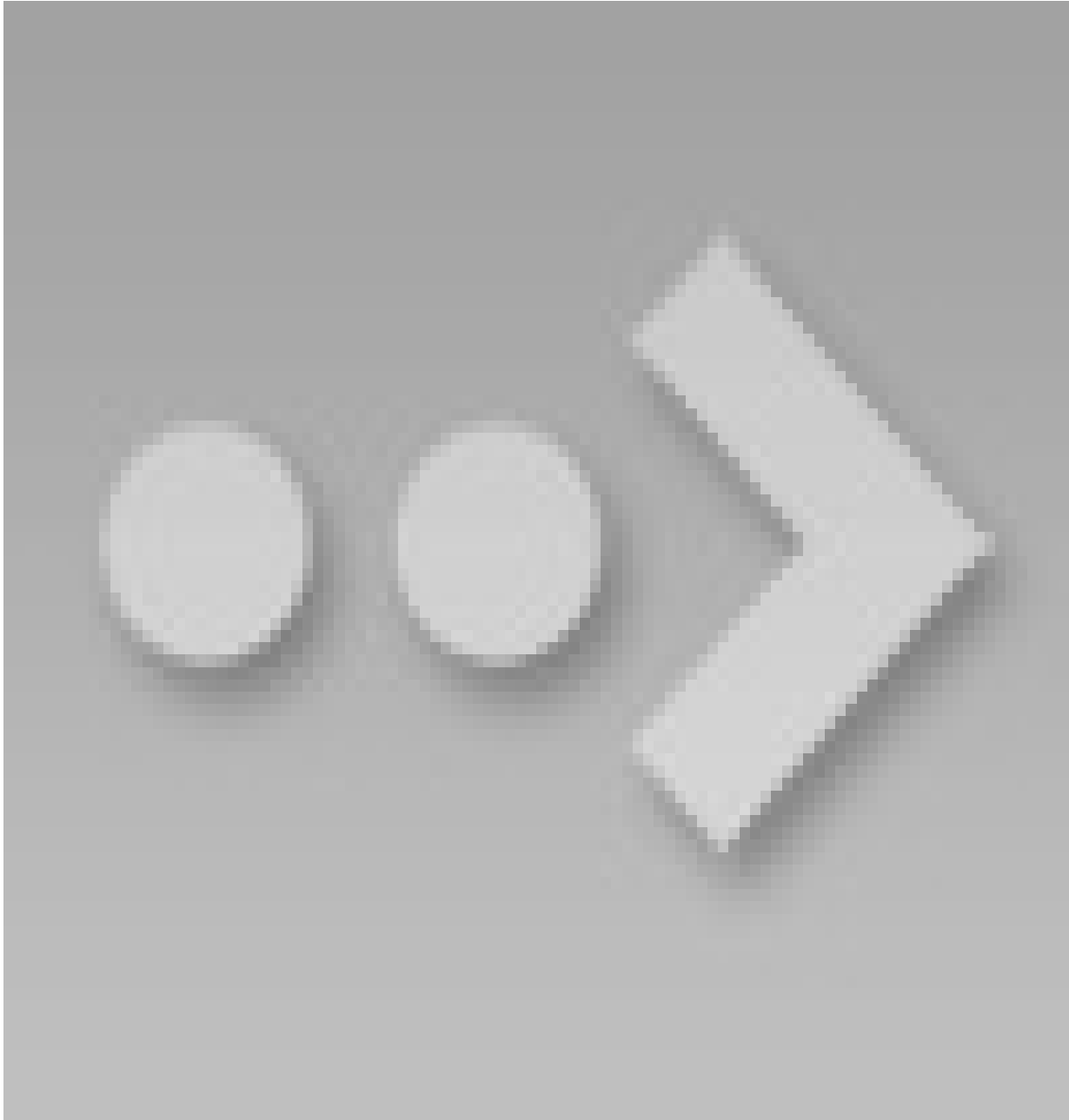
If I wanted to see this same analysis done at the Sub-Category level as pictured above, I will drag the Sub-Category dimension to the Rows Shelf. This will create a row with the sales and profit ratio by state tree map for each sub-category.

[Insert Image 48-4]



The final steps required to match the example would be to drag the State dimension to the Label Marks Card and double-click on the color legend to choose the colors and steps. I changed the steps to 2 and changed the colors so that any state with a negative profit ratio per sub-category would be colored red and any state with a positive profit ratio per sub-category would be colored navy.

[Insert Image 48-5]

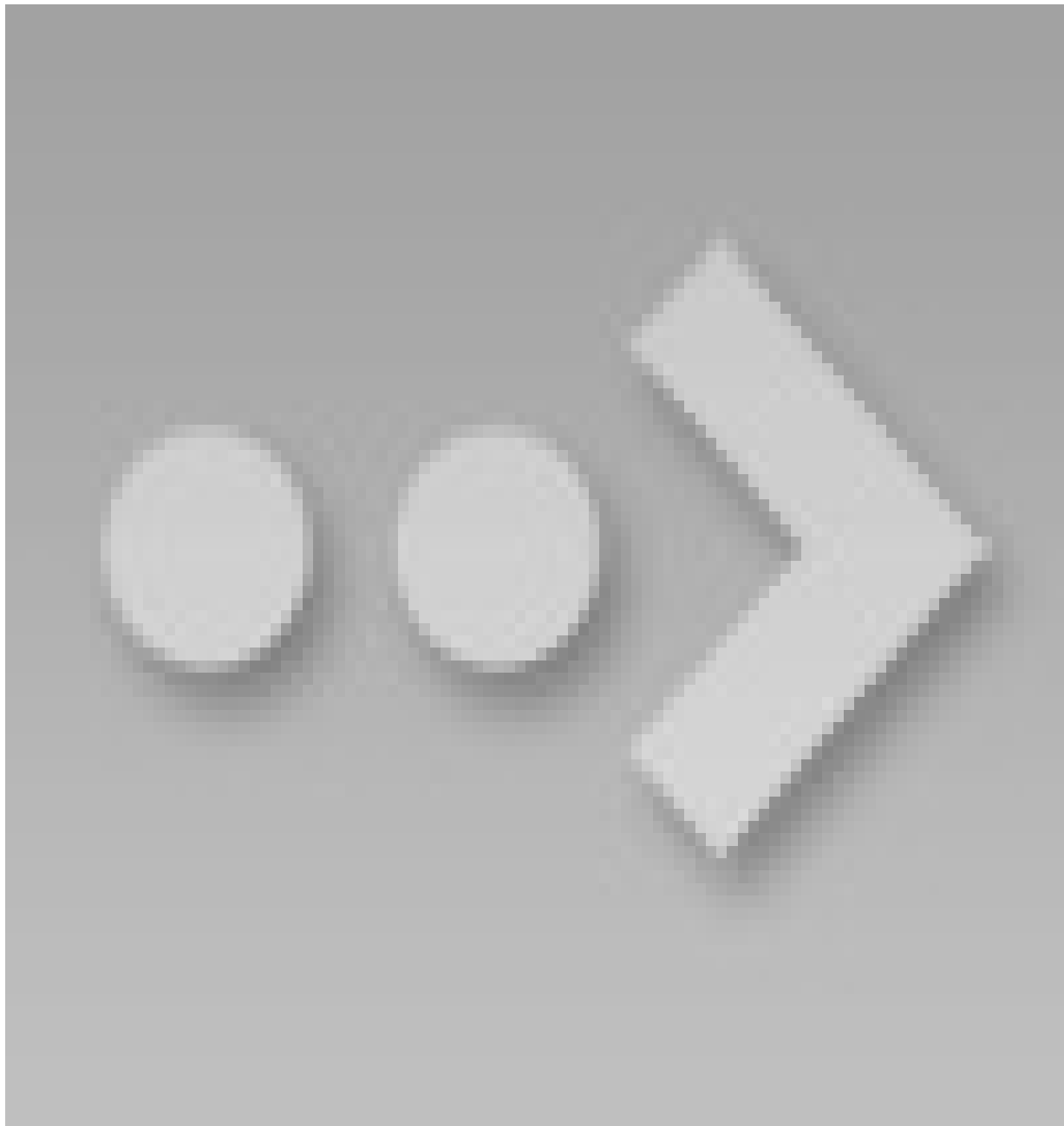


This tree map provides several insights, including:

- The overall sales are strongest in the Chairs and Phones sub-categories. This is the main insight we would see when looking at a simple Sales by Sub-Category bar chart.
- The Tables sub-category is largely composed of unprofitable states, and two out of the top three highest selling states are unprofitable.
- The overall best-selling state and sub-category combination is Phones in California, which was profitable.

I also see that California is the highest selling state in all sub-categories except for Machines and Binders. Note that I was not able to see that insight in the screenshot alone, but by hovering over marks in the interactive version of the workbook. You can also simply provide a filter that looks at one sub-category at a time and add even more context to the mark labels:

[Insert Image 48-6]



These are just a few examples of the insights gleaned from this visualization, and a couple of limitless applications of tree maps.

Chapter 49 – How to Make Sparklines

Sparklines are condensed graphs or charts that can be used in-line with text or grouped to show trends across several different measures. The term sparkline was introduced by Edward Tufte, a data visualization pioneer, and proponent of the chart type. Sparklines are typically so small that the chart itself usually does not contain familiar context that you would find in a full-size chart, such as axes. Despite this limitation, I find sparklines to be one of the most effective corporate chart types for quickly communicating trends across KPIs. After all, you can always provide context in the surrounding text, and if you are using Tableau, context can be added through other approaches such as tooltips (the information that appears when you hover over a data point).

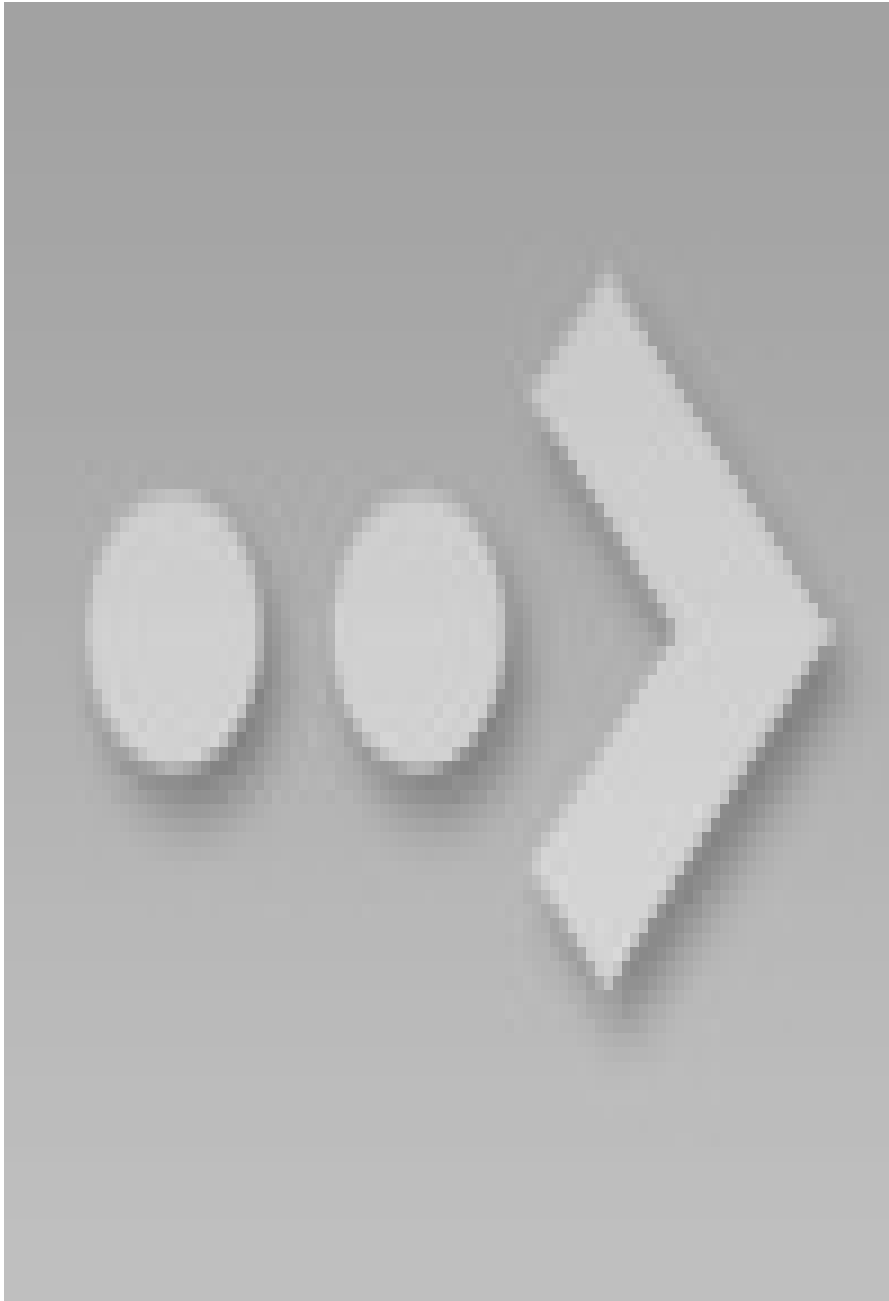
It is very common for sparklines to be a foundational piece of the corporate dashboards I create. I think they are a great place to guide an end user to start, so I usually place sparklines in a prominent area of my dashboards, such as down the left side. For more on dashboard layout and design, see tip five in the data-storytelling tips section, Use the Golden Ratio.

How to make sparklines in Tableau

Sparklines are easy, and (dare I say?) fun, to create in Tableau. If you would like to follow along, I am using the Superstore sample dataset that comes out of the box with Tableau. To get started, you will leverage two special fields in your data, *Measure Names* and *Measure Values*. These fields are automatically generated in your data by Tableau so they will be available to use even though they do not exist in your underlying data.

Step 1 - Place 'Measure Names', then 'Measure Values' on your 'Rows' shelf.

[Insert Image 49-1]



Notice that *every* measure name appears in your chart, whether it is relevant to your analysis or not - more on filtering out specific measure names later. Since you placed 'Measure Values' on your view as well, each measure name has an

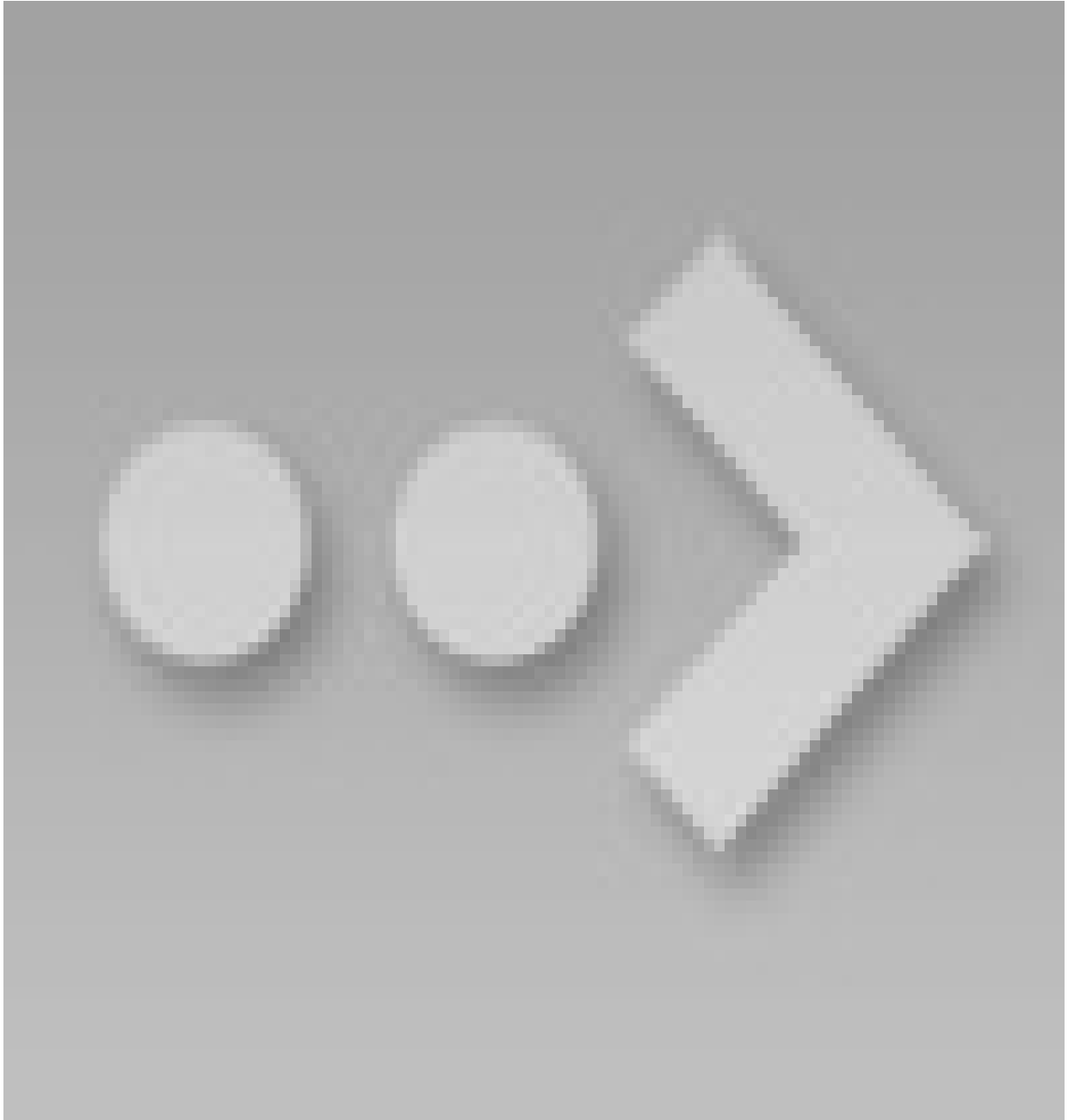
accompanying value, shown by default as a bar in each measure's default aggregation (i.e. SUM, AVG, etc.). I have also changed the view's fit to "Fit Height" so that I can see all of the measure names and values without scrolling.

This is a good start, but now we need an element of time to trend the measure values.

Step 2 - Place a date field on the 'Columns' shelf.

By right-clicking and dragging my 'Order Date' field onto the 'Columns' shelf, I was given an extra option to select the date aggregation (i.e. Year, Month, Week). To get the view to look as it does below, I chose the 'MONTH' option that was colored green. The green indicates that the date will be continuous.

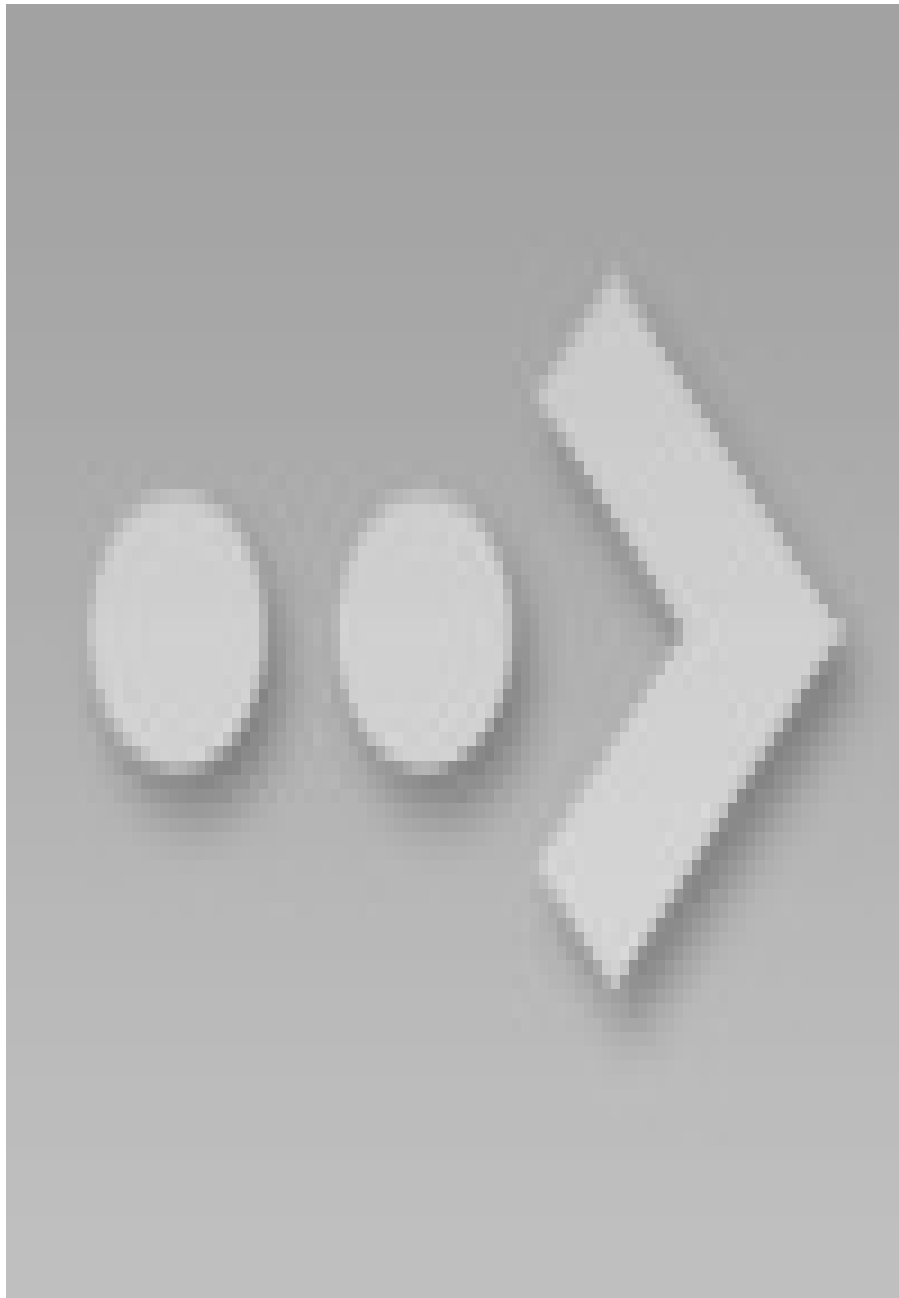
[Insert Image 49-2]



We have now essentially made a series of line graphs, but they are not very 'sparky', making it difficult to quickly glean insights. This is an easy fix in Tableau by clicking and dragging the right side of the graph to the left to reduce the width of the view.

Step 3 - Reduce the width of the sparklines view to make the trends pop.

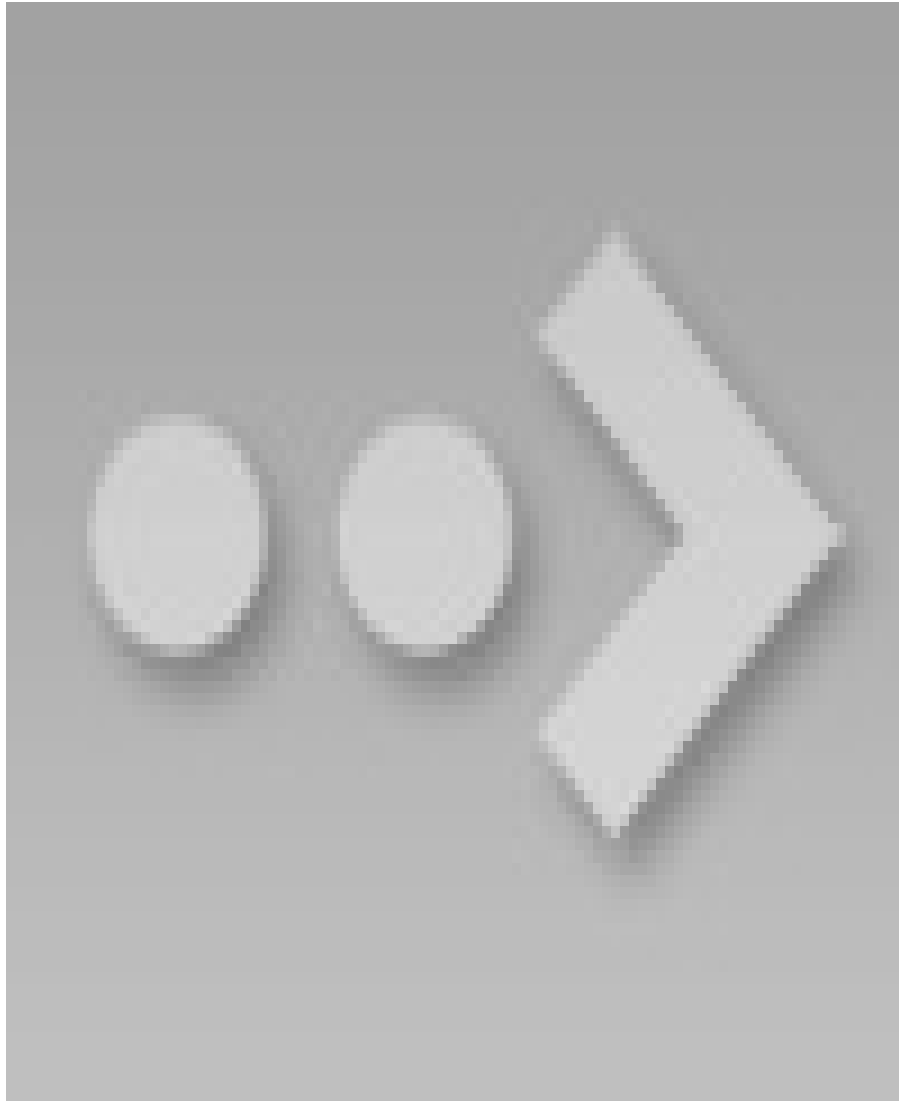
[Insert Image 49-3]



You can see at this point that the sparklines are coming together, but as previously mentioned, we have an irrelevant measure name that is not adding much to our analysis.

Step 4 - Remove irrelevant measures from your view.

[Insert Image 49-4]



For the purposes of this analysis, I removed the Number of Records measure by dragging its green 'pill' from the 'Measure Values' shelf. Number of Records is a generated field and not needed for this analysis.

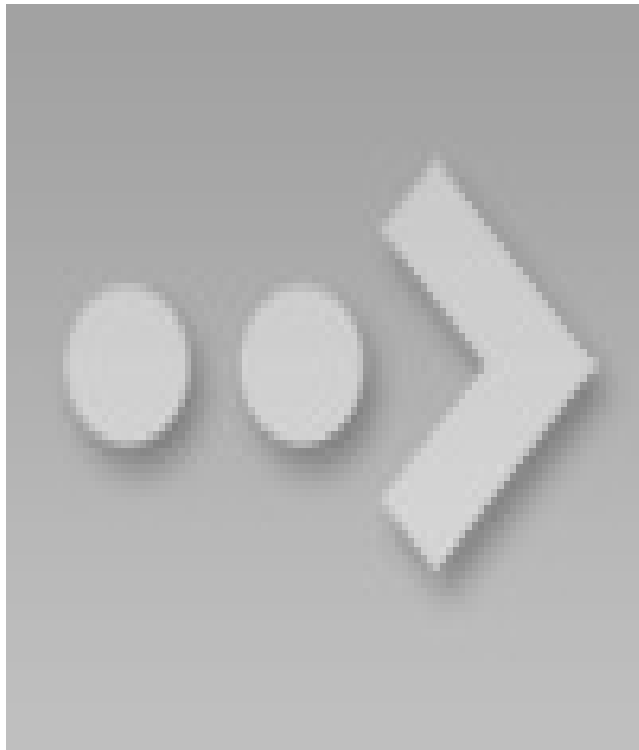
We're getting even closer now, but notice that Discount is not providing much insight because this measure has very little fluctuation. For this reason, I typically remove zero from my axes in sparklines. There is much debate around whether it is ever appropriate to exclude zero from your axes, because it is easy to mislead your audience when an axis starts anywhere but zero. In the case of sparklines, and measures with little to no volatility, I recommend either excluding zero on the axes, or removing these types of measures completely

from your view. I say this because if you cannot gain any insight from these measures in your sparklines, they are not adding any value.

Step 5 - Exclude zero from your axes or remove measures that have little to no fluctuation.

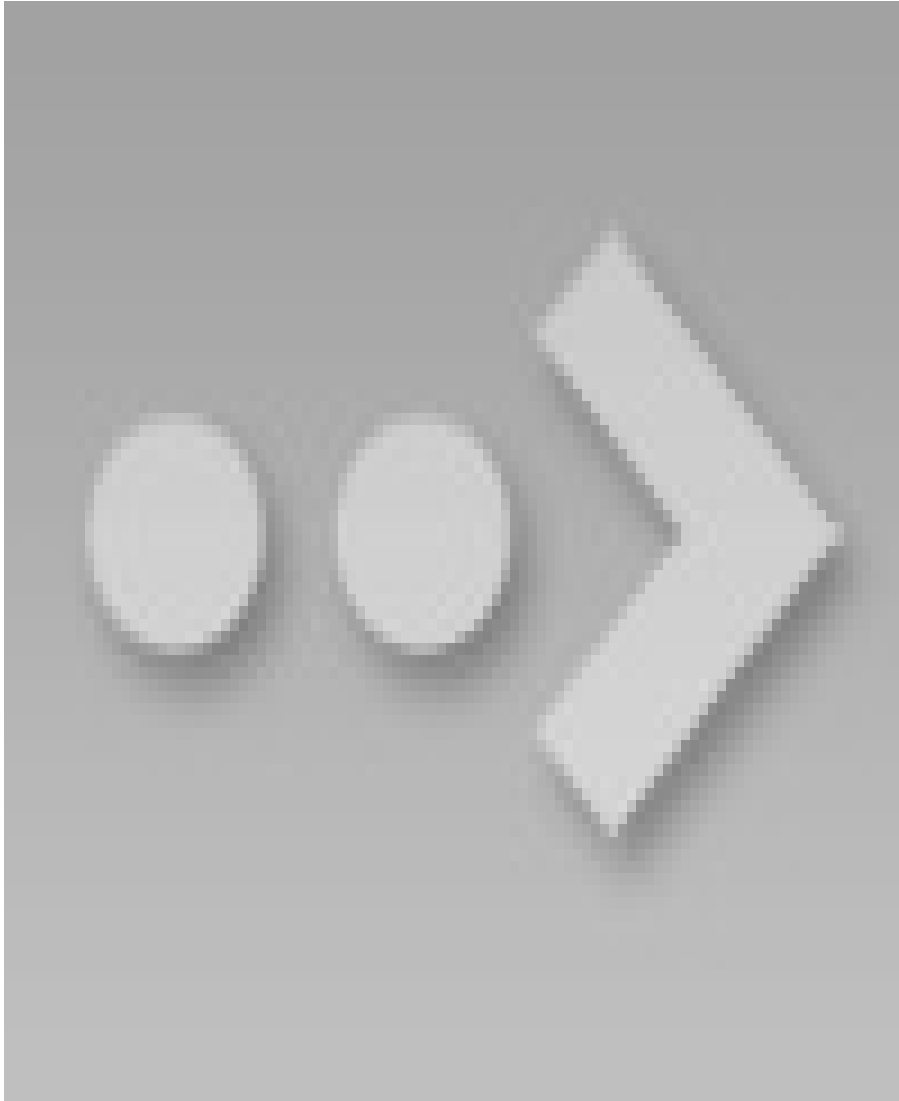
To do this in Tableau, right-click on any of the axes in your sparklines and choose "Edit Axis". You will see a box specifically created to give you the option to include or exclude zero in your axes. By default, the box to "Include zero" is checked. To exclude zero, uncheck this box.

[Insert Image 49-5]



You can now see the fluctuation in Discount.

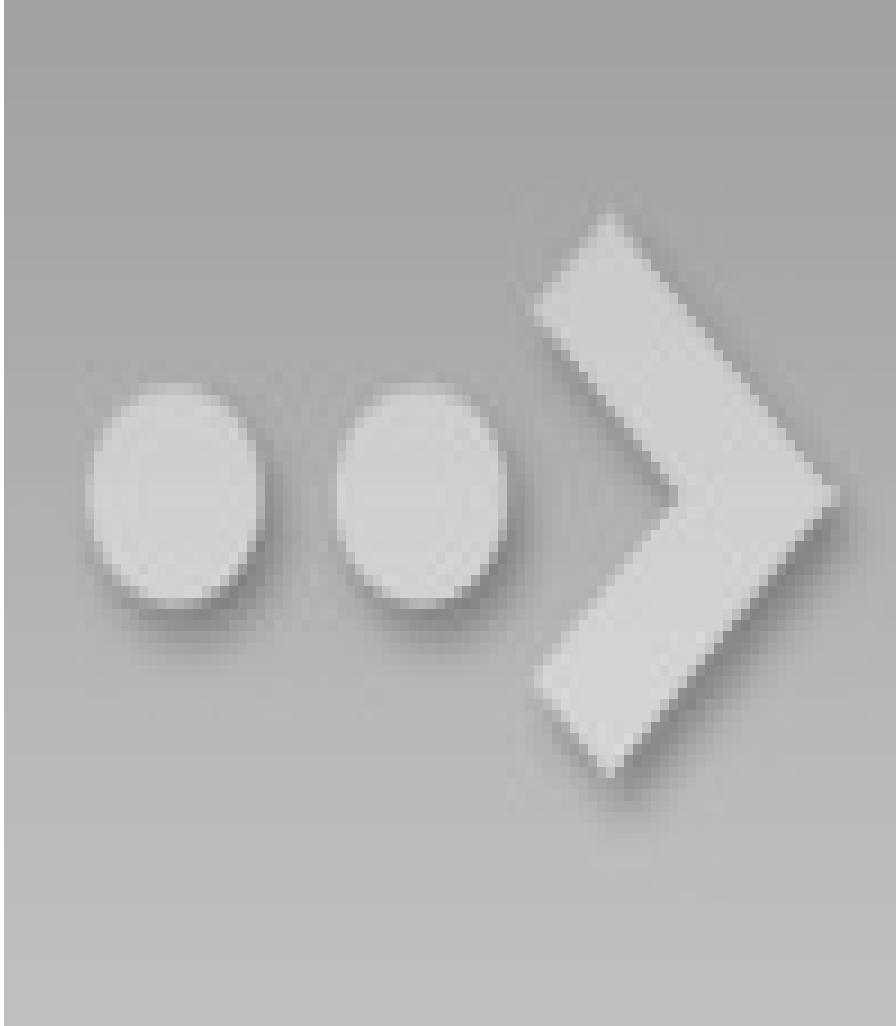
[Insert Image 49-6]



From here, all that's left is to format the sparklines to your preference. Remember, sparklines are not quite like regular charts or graphs in that they are meant to provide quick trends at a glance. They don't usually contain typical context, such as the axis values.

Step 6 - Hide axes and format your view

[Insert Image 49-7]



Formatting changes I made to finalize this view include:

- Hid the Y axis by right clicking on the axis and deselecting "Show Header".
- Hid the X axis. Some prefer to keep the axis that shows the date range. I personally exclude this from my sparklines, but if you need to show it, I recommend only showing the start and end points.
- Removed the gridlines.
- Removed the column separators.
- Softened the row separators by choosing a dotted line instead of a solid line.

Final considerations

The sparklines pictured above can definitely stand on their own and provide a great deal of insight and a starting point for any dashboard. That being said, Tableau allows you to customize from here in many different ways. For example, you may choose to show labels for the minimum and maximum values for each sparkline. This approach provides an alternative to showing the Y axis because it shows the end user how low and high the axis goes. You can also customize the tooltips to provide additional context such as the exact values and dates that the end user is viewing.

Chapter 50 – How to Make Small Multiples

Small multiples are a group of charts or graphs that share the same axes and scales, which allows the user to compare trends across dimensions in a single view. They have been praised for their ability to provide a great deal of context, reducing the need for end users to ask the dreaded, “...So what?”. The term small multiples was popularized by Edward Tufte, who puts it best in his book, *Envisioning Information*:

“At the heart of quantitative reasoning is a single question: *Compared to what?*”

While smaller series of small multiples can work well on an executive summary dashboard, I typically like to use them as a second layer in an analysis. My executive level view, or ‘first layer’, may provide higher level information about a particular measure, such as the overall sales trend and progress towards goal. This is a ‘descriptive’ view of the data answering the question, “What is happening with sales?” From here, I may provide an option to view sales across different dimensions and sub-categories as a series of small multiples in a second layer of the dashboard (often located away from the first view). While small multiples is still a ‘descriptive’ view, it helps answer the question, “Compared to what?”

How to make small multiples in Tableau

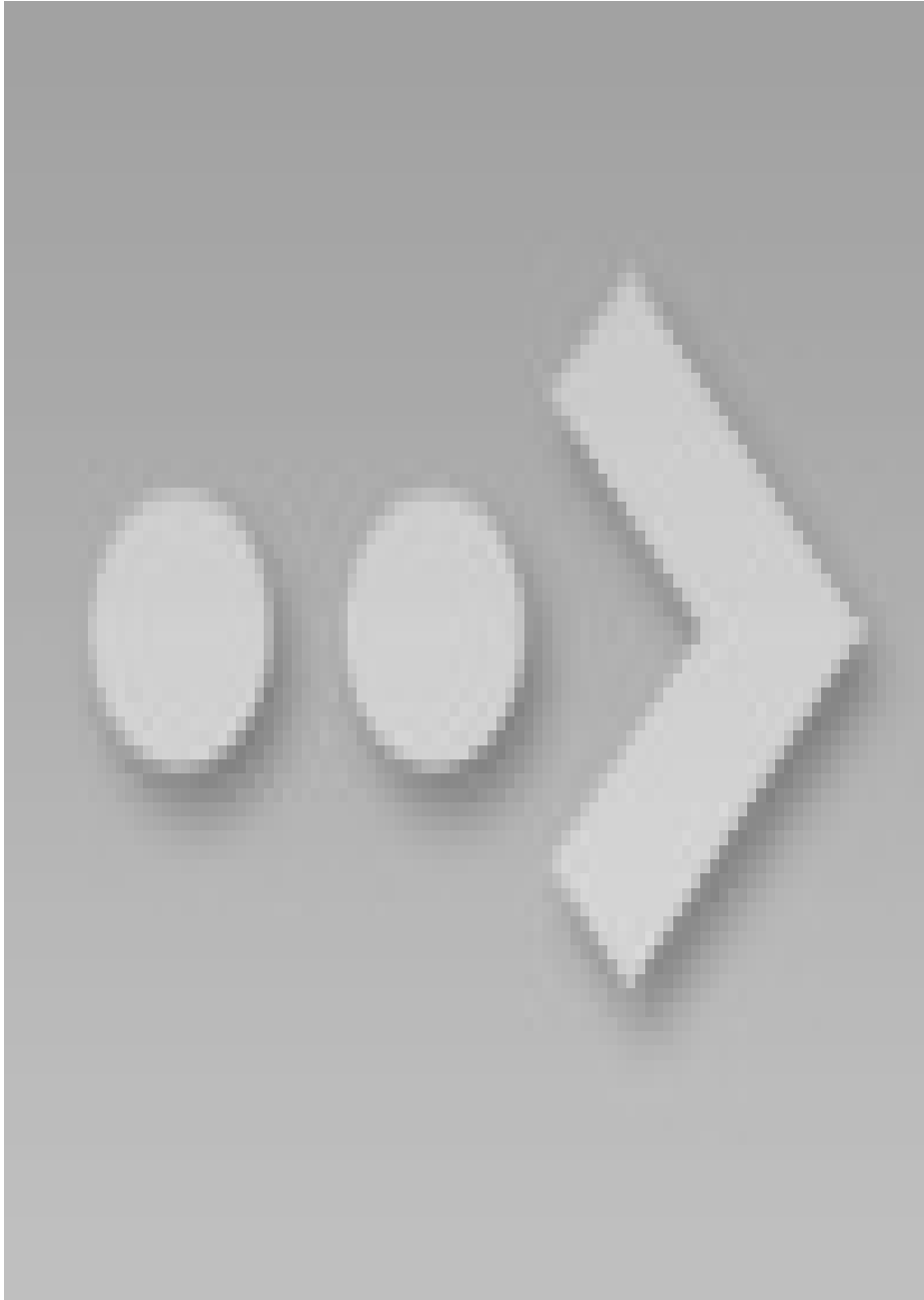
Small multiples can come in several different forms, but for the purposes of this tutorial, I am going to show you how to compare a single measure across two

different dimensions. I am using the Superstore sample data that comes packaged with every version of Tableau Desktop. To get started, select two dimensions and a measure. I have chosen the dimensions of Segment and Region and the measure of Sales for the view to follow.

Step 1 – Place one dimension on the ‘Columns’ shelf and the other dimension and your measure on the ‘Rows’ shelf.

By default, Tableau has created a bar chart for you. By placing Segment on the Rows Shelf, performance for each segment is read left to right, in rows. By placing Region on the Columns Shelf, each region is represented vertically, in columns.

[Insert Image 50-1]



Small multiples may be a series of bar charts, but we still need to add an element of time to trend the Sales measure.

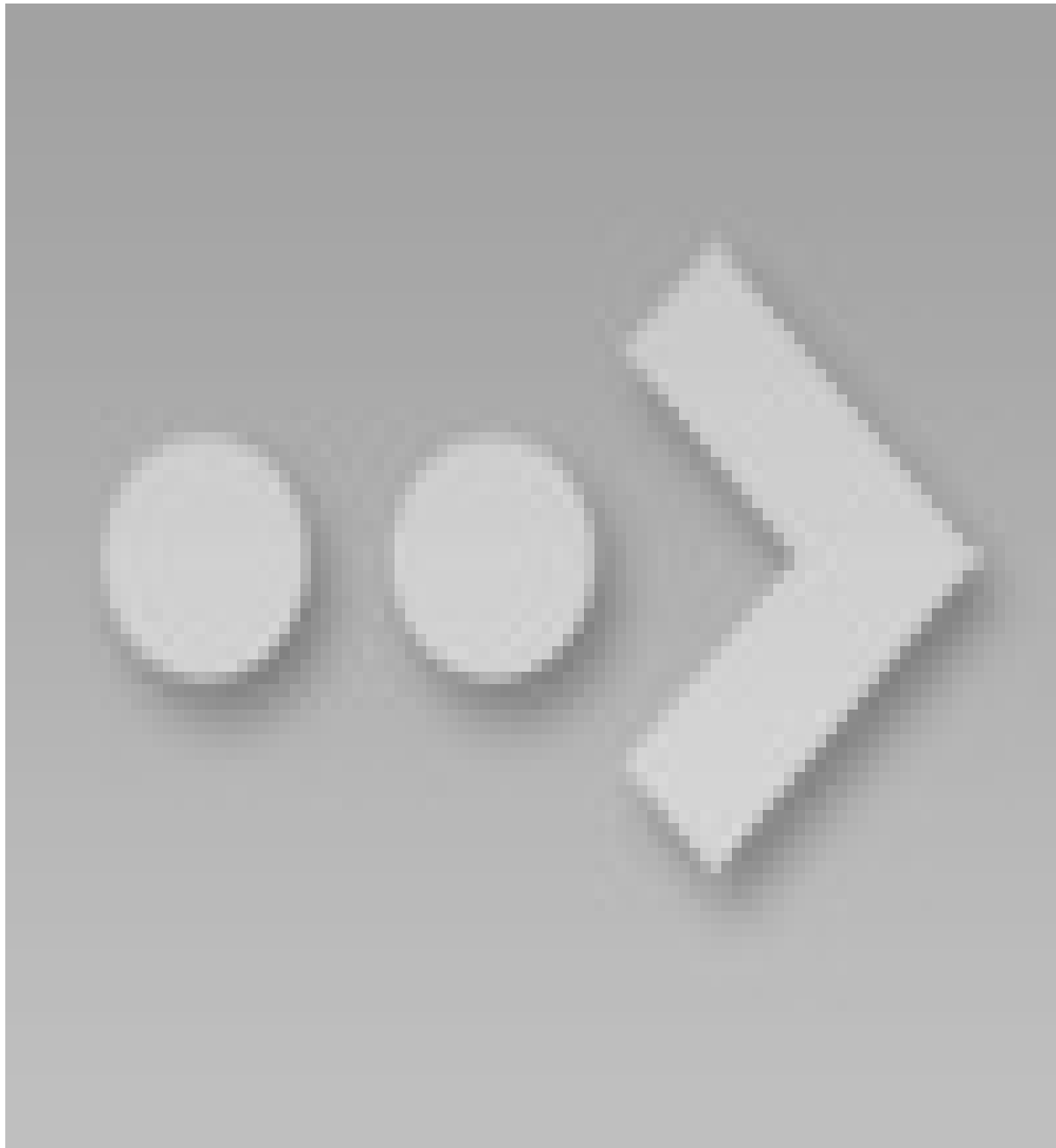
Step 2 – Place a date field on the ‘Columns’ shelf.

Let’s pretend that we would like to evaluate the seasonality of sales to answer a question such as, “Is there a certain month that I can expect a spike in sales?” For this type of analysis, we will use a discrete date field with a monthly

aggregation, which will always show us the performance per distinct calendar month. By using discrete months, I know that I will always have twelve data points per small multiple, one for each calendar month (January, February, March, and so on). The Superstore sample dataset contains four years of data, so if we used a continuous date field with a monthly aggregation, we would have up to 48 data points per small multiple (four years x 12 months = 48 points).

By right-clicking and dragging the Order Date field onto the Columns Shelf, I was given an extra option to select the date aggregation (i.e. Year, Month, Week). To get the view to look as it does below, I chose the MONTH option that was colored blue. The blue indicates that the date will be discrete.

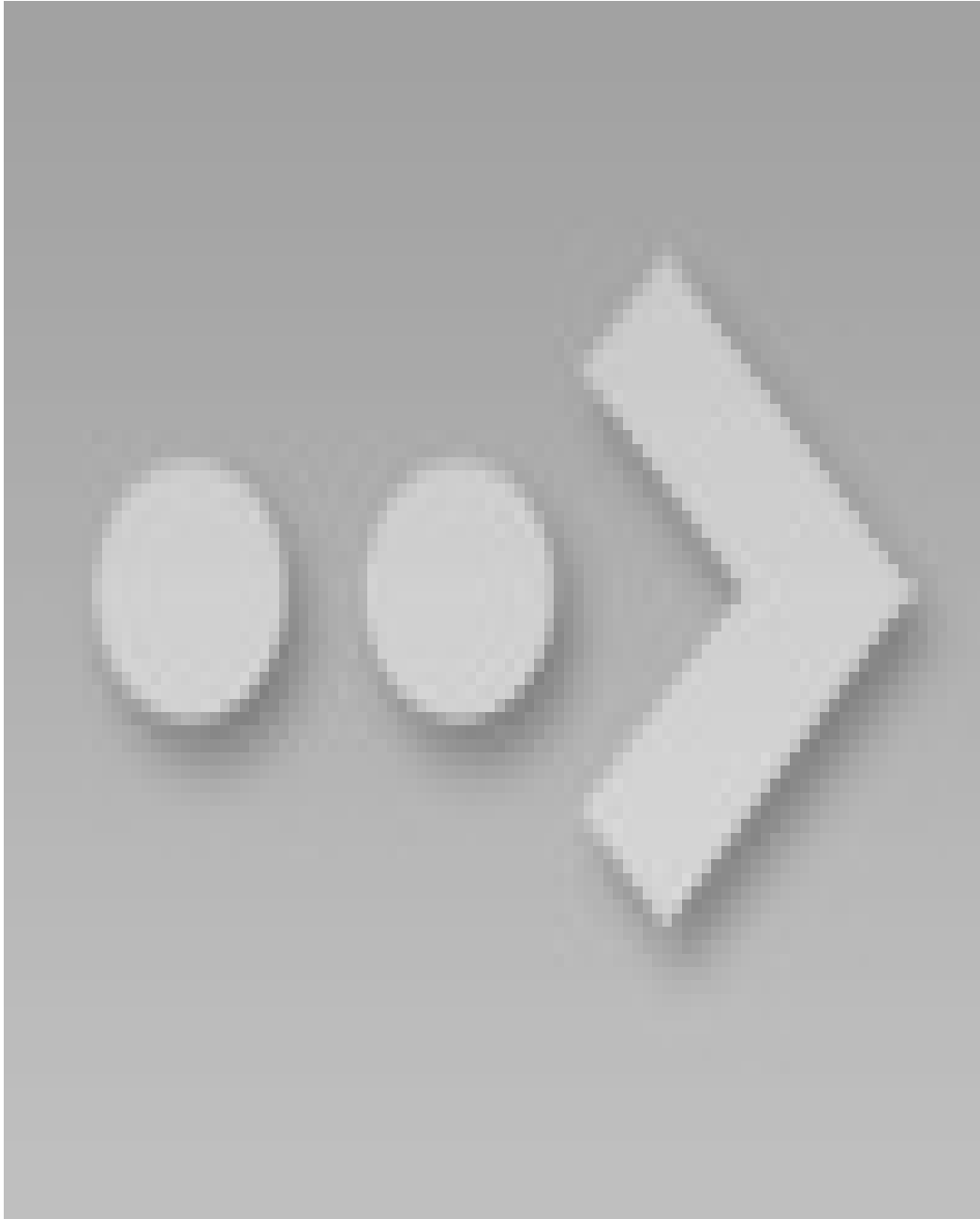
[Insert Image 50-2]



Step 3 – Format the small multiples to your preference.

From here, all that is left to do is format the view to your liking.

[Insert Image 50-3]



Formatting changes I made to finalize this view include:

- Hid the field labels for columns by right-clicking on Region / Order Date.
- Hid the field labels for rows by right-clicking on Segment.
- Hid both the X and Y axes by right-clicking on them and deselecting “Show Header”. This is a personal preference of mine. You may choose to keep the axes in the view for additional context. To keep the view as clean as possible, I opted to provide this context in the form of a title, and would

typically provide additional information in the tooltips that appear when an end user hovers over different data points.

- Increased the font size in the headers.

Final considerations

In this tutorial, we have laid a strong foundation for using small multiples, but Tableau makes it easy to build on this view by customizing it in many different ways. For example, instead of using lines, you can change the mark type to bars, which are particularly useful for measures that can be positive or negative, or areas – which provide a beautiful design alternative. I have also seen small multiple views that display year over year data nicely and Ben Jones of Tableau has shown [how to create small multiple maps in Tableau](#). So get creative, and build small multiples that help provide the answer to “Compared to what?”.

Chapter 51 – How to Make Bullet Graphs

Bullet graphs are a variation of the bar chart invented by Stephen Few. Bullet graphs are one of my go-to chart types and are often featured prominently throughout my dashboards because I find that when I use them, I hear the question, “So what?”, less often. As powerful as bar charts are at communicating data, when viewed in a vacuum – or without additional context – their comparisons are limited to only the fields that are displayed in the chart. Bullet graphs improve on the bar chart by providing additional points of comparison. For example, in addition to showing a bar for widget sales, a bullet graph would also include a point of comparison that shows either last year’s sales or a target sales amount. Further, bullet graphs will include shading to illustrate how close your sales number is to last year’s number or your target.

While I find this to be one of the most powerful visualization types, it took me some time to wrap my head around how to make them when I was getting started with Tableau. Bullet graphs are a “Show Me” option in Tableau, but I found that this option was not always providing the results that I expected. Once I figured out that bullet graphs are simply a combination of bars and reference lines, it was much easier for me to create them. The good news is that if you can create a bar chart in Tableau, you can create bullet graphs!

How to make bullet graphs in Tableau

Bullet graphs are only an appropriate chart type to use if you have a point of comparison, such as last year's performance or goals. For this walkthrough, I am using the Superstore sample dataset that comes prepackaged with every download of Tableau. This dataset does not contain goal information, so my point of comparison will be last year's performance.

Step 1 – Break out this year's performance and last year's performance.

For this bullet graph, I am going to look at Sales by Sub-Category. In order to create a bullet graph, I will need to break out this year's sales and last year's sales. The Superstore dataset only runs through 2016 and 2016 outperformed 2015 in almost every sub-category, so for the purposes of this tutorial, I am going to pretend that this year is 2014 and last year was 2013.

The isolation of sales for these two years is achieved through calculated fields. To create a field that contains only 2014 sales, right-click on the 'Sales' measure and select "Create Calculated Field". This approach provides a small shortcut because sales is already part of your formula when the 'Calculated Field' dialog box opens. Once the dialog box is open, enter a formula like this to isolate the current year's sales:

```
IF YEAR([Order Date]) = 2014 THEN [Sales] END
```

[Insert Image 51-1]

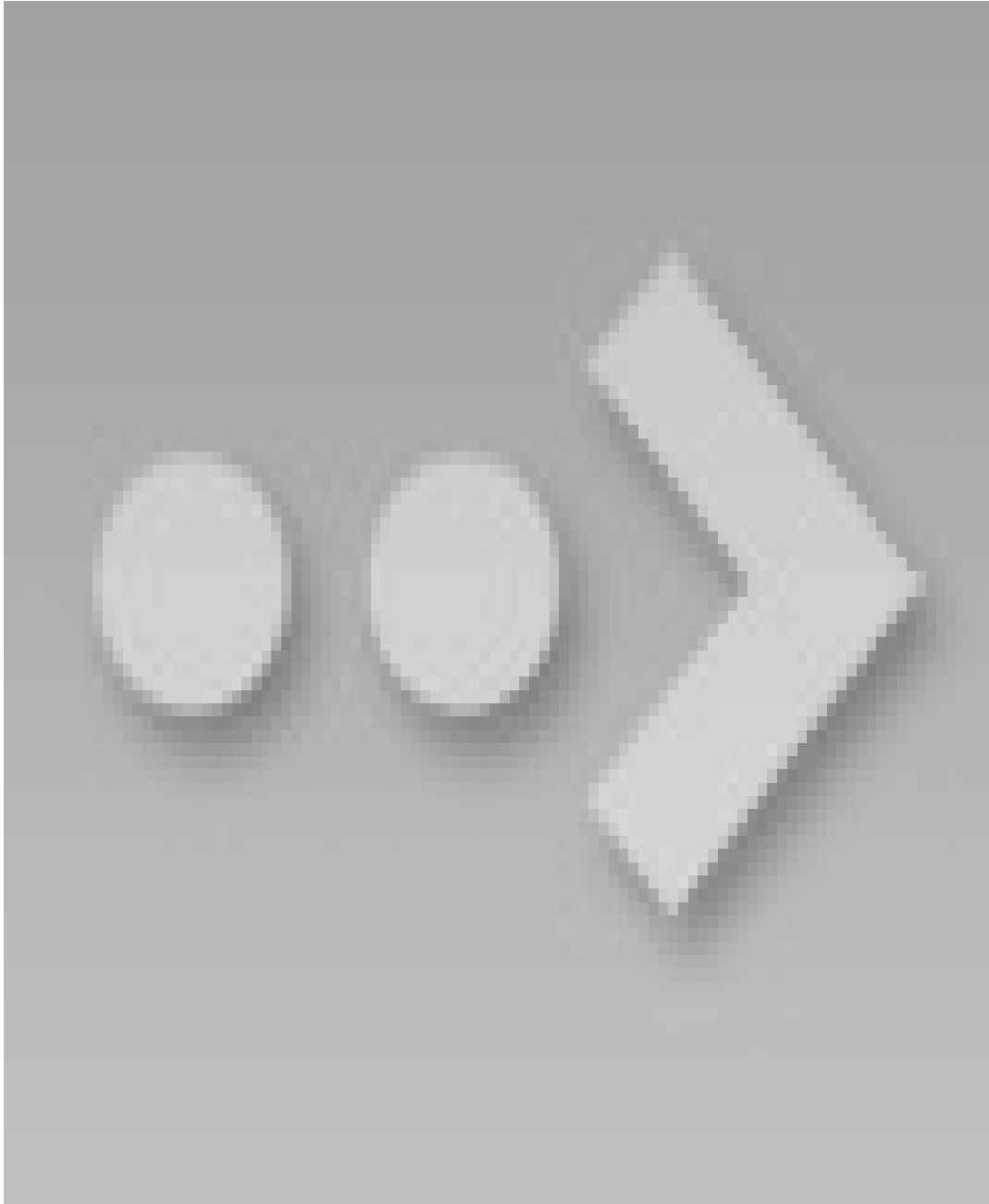


Repeat the above step to isolate last year's sales (in this case, 2013).

Step 2 – Create a bar chart that will serve as the foundation for your bullet graph.

Create a bar chart as you normally would by placing the current year's sales on the Columns Shelf and a dimension on the Rows Shelf. I am looking at sales by sub-category, so I have placed the Sub-Category dimension on the Rows Shelf. I have also fit the height for more visibility.

[Insert Image 51-2]



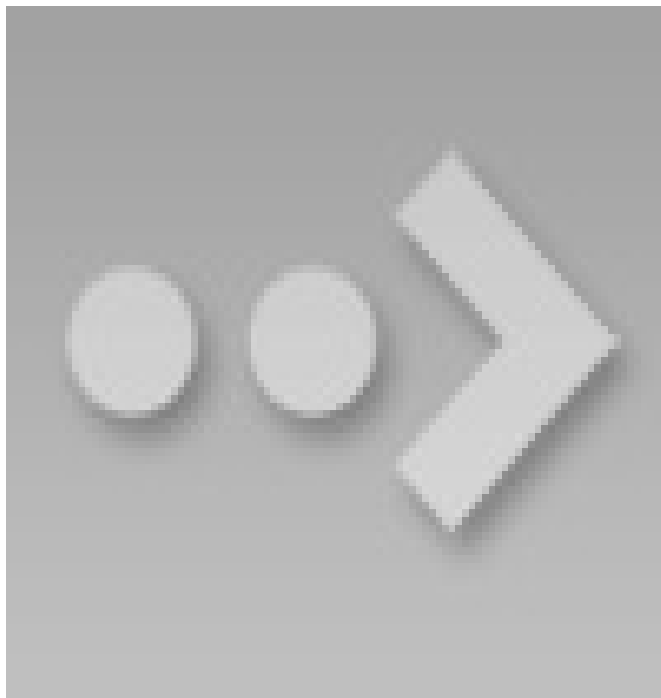
Step 3 – Add a reference line for last year's sales.

In order to use last year's (2013) sales as a reference line, the isolated calculation that you created in Step 1 needs to be a part of your view. Even though it is not yet visible, you can make Last Year's Sales part of your view by dropping the field on your Detail Marks Card. Notice that when you place Last Year's Sales on the Detail Marks Card, your view does not change, but now that data is

available to use as a reference line.

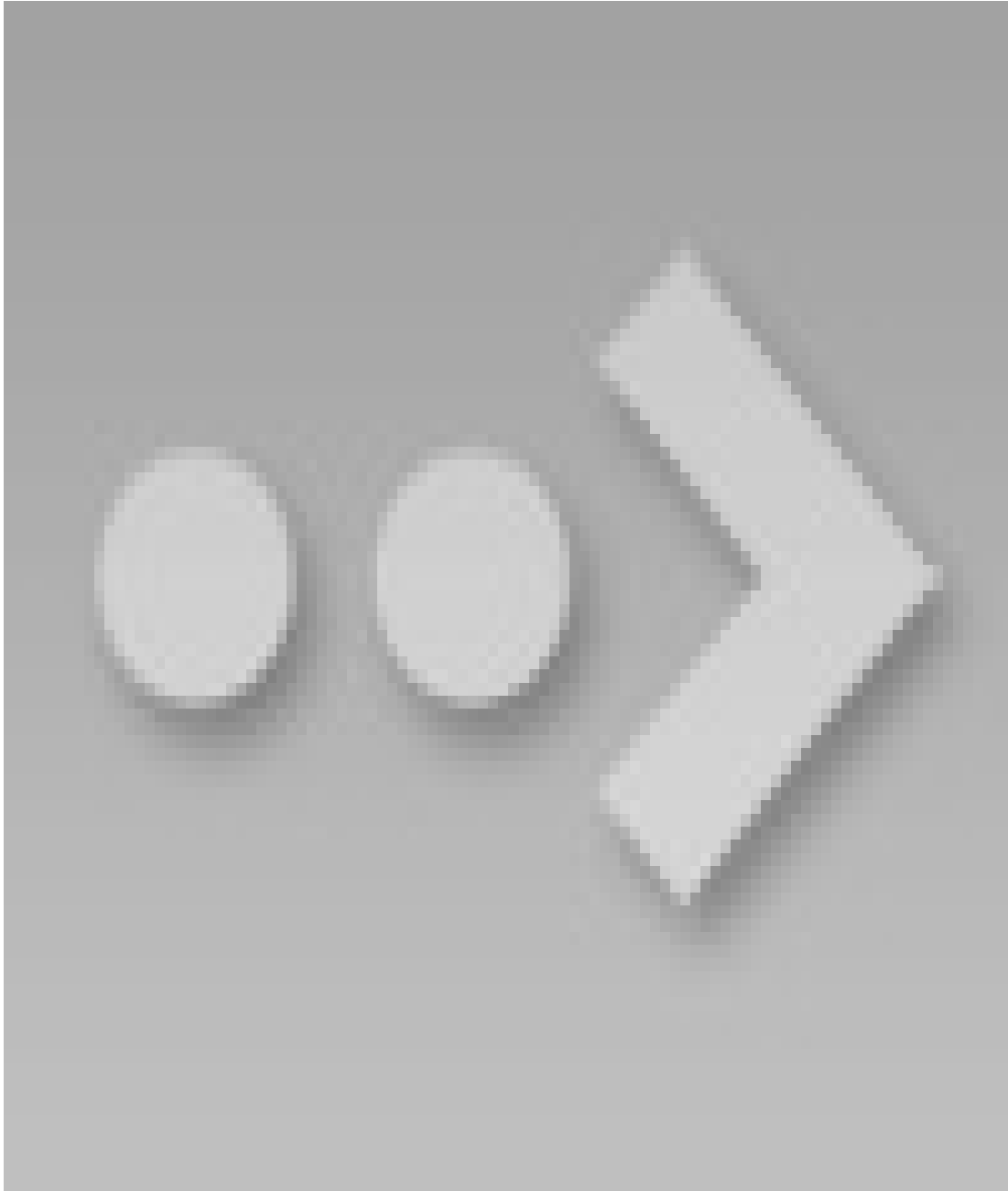
To create a reference line, right-click on the X-axis and select “Add Reference Line”. Change Value to SUM(Last Year’s Sales) and Label to None. The most important change is to toggle the Scope radio button from Per Pane to Per Cell. This will give you a reference line for each distinct sub-category. You may also choose to make the line a bolder color and heavier weight to make it stand out on your view. After the appropriate selections, your reference line dialog box should look something like this:

[Insert Image 51-3]



At this point, your view should look like this:

[Insert Image 51-4]



Step 4 – Add a reference distribution for last year's sales.

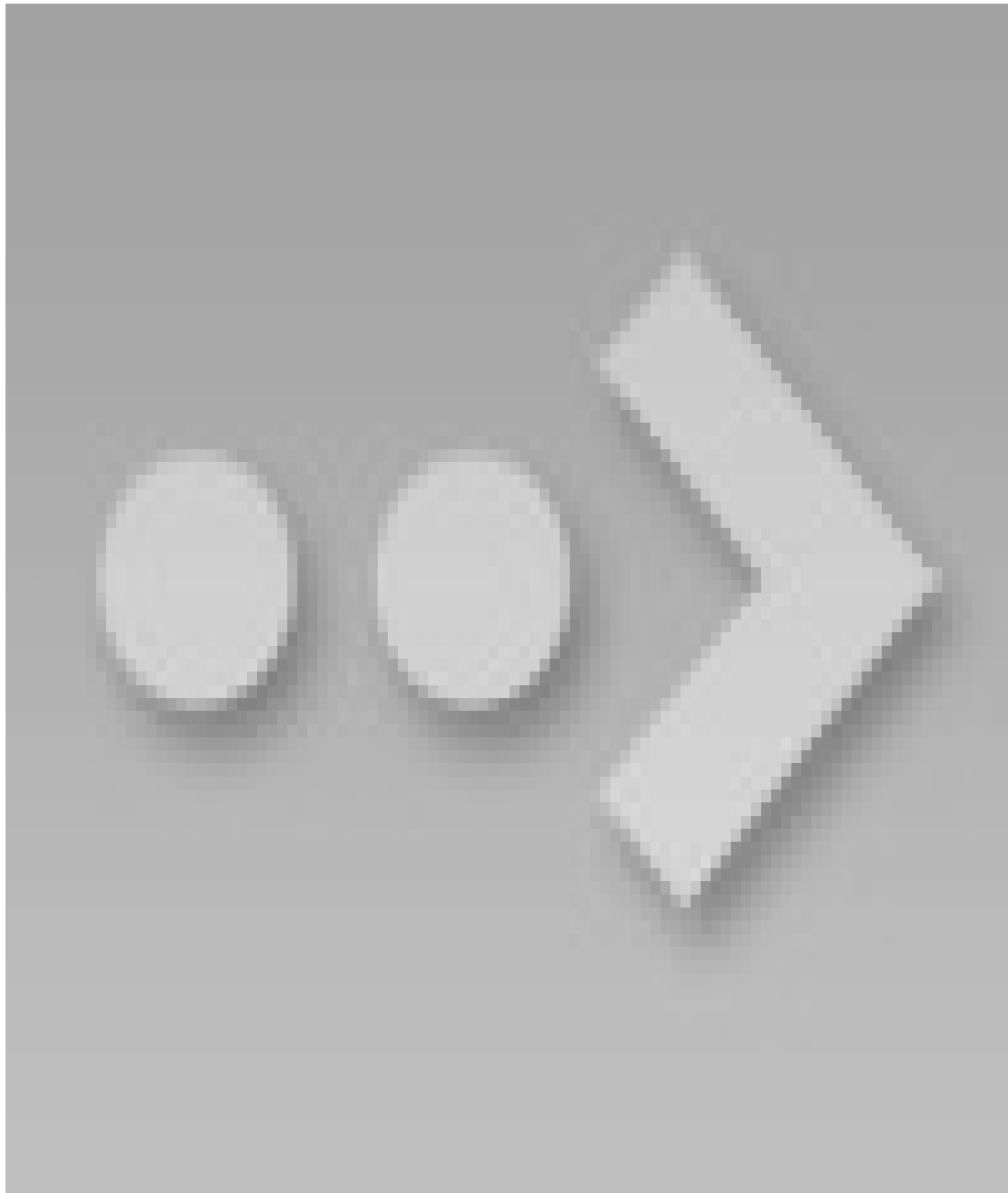
At this point, you have already added a comparison point to your bar chart which shows whether each category is outperforming or underperforming last year's sales. To take this a step further, you can add a reference distribution to show how close this year's sales are to last year's for the underperforming sub-categories.

This reference type is slightly more complicated, but still easy to do in Tableau. To create a reference distribution for your bullets, do the following:

1. 1. As you did before, right click on the X-axis and select “Add Reference Line”.
2. 2. Select Distribution at the top.
3. 3. Change the ‘Scope’ from Per Pane to Per Cell.
4. 4. This is the trickiest part. You need to change the Computation Value from This Year’s Sales to Last Year’s Sales. You can do this by clicking the down arrow on the Computation > Value box. Then where it says ‘Percent of:’, make the appropriate selection. Notice that the default percentages are 60% and 80%. This means that it will show shading for 0 – 60% of last year’s sales and 61 – 80% of last year’s sales.
5. 5. Change Label to None.
6. 6. Check the box that says ‘Fill Below’. This will provide the correct shading when you apply the reference distribution.

After hitting okay, you should see a well-done bullet graph that looks like this:

[Insert Image 51-5]



Now that you have a reference distribution, you can quickly determine not only whether or not each category is on pace with last year, but you can see how far behind pace underperforming sub-categories are relative to a year prior.

Another great application of bullet graphs is to use them to show progress toward goals. In that case, the bars would be current year's sales, and the reference lines and distributions would be your goals. By using year-end goals, you can track movement throughout the year and determine which sub-categories are progressing the fastest. This type of context may be enough to

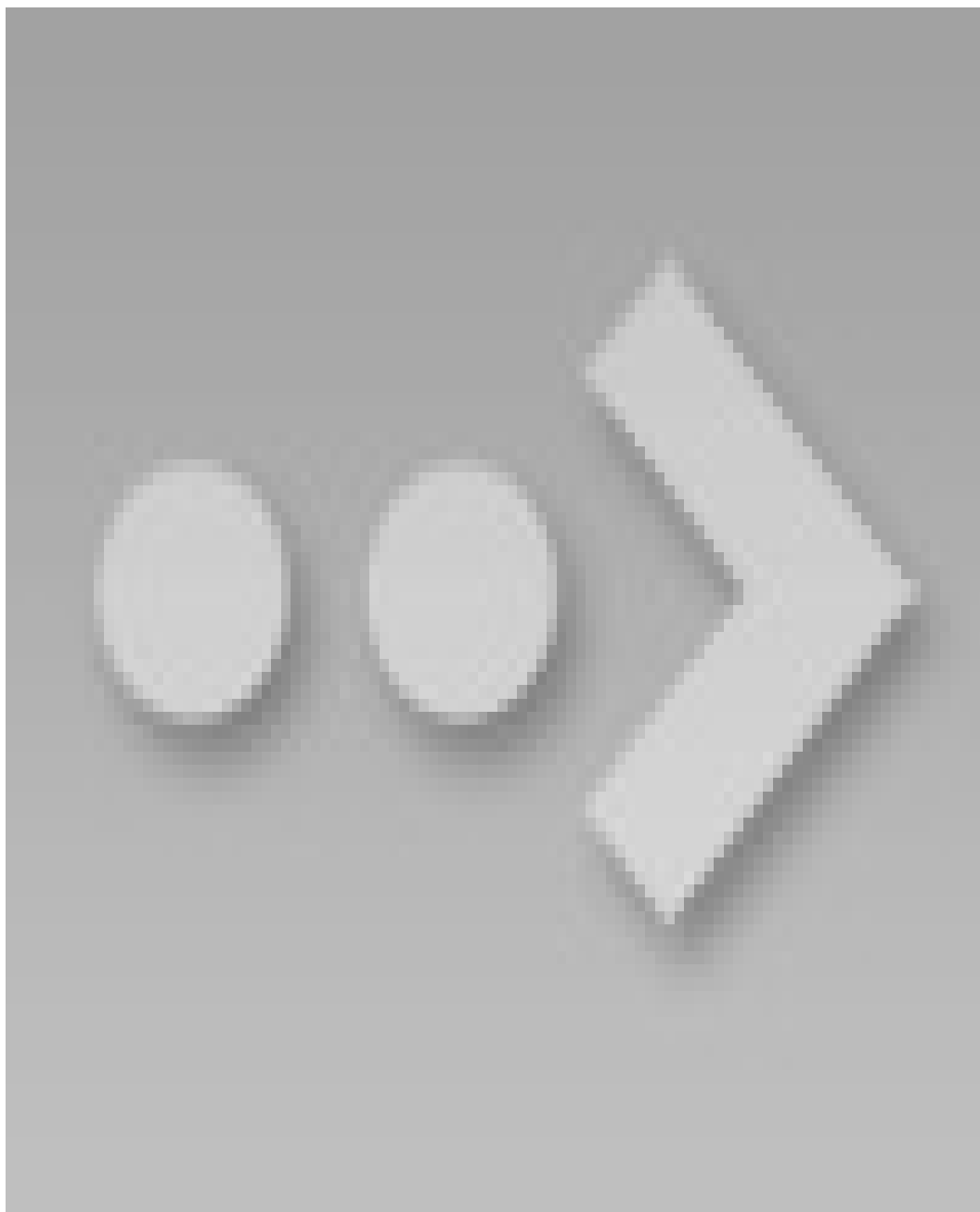
help you pivot resources to help underachievers catch up – actionable insight that may not have come from a bar chart alone.

Chapter 52 – How to Make a Stacked Area Chart

A stacked area chart ‘stacks’ trends on top of each other to illustrate how a part-to-whole distribution is changing over time. When combined with a table calculation that computes the percent of total for each dimension member in the visualization, stacked area charts are an effective way to evaluate distributions.

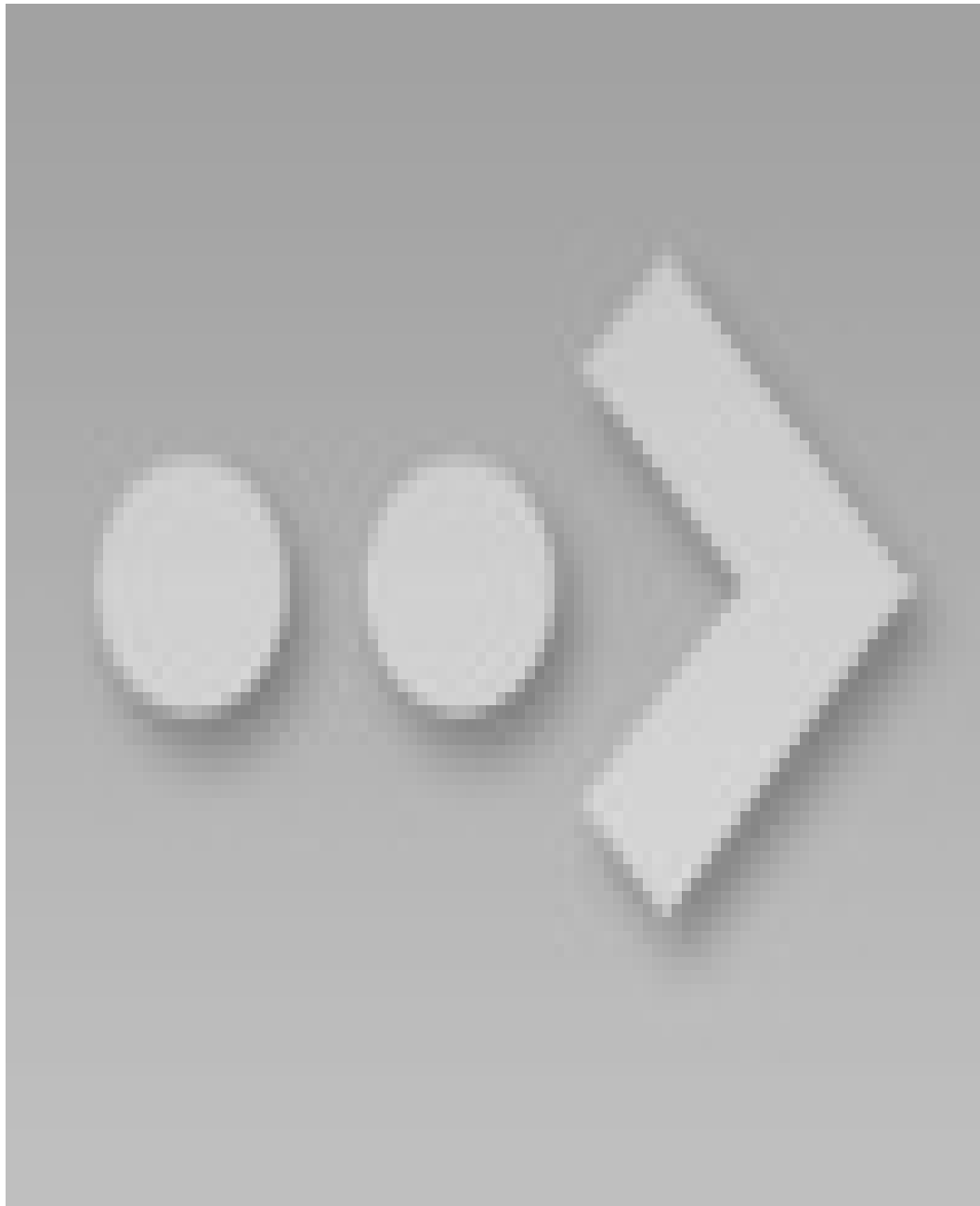
To create a stacked area chart, I will start with this line graph showing Sales by Sub-Category by discrete month of Order Date in the Sample – Superstore dataset.

[Insert Image 52-1]



To create a stacked area chart in Tableau, simply change the mark type from Automatic, which is currently set to Line, to Area.

[Insert Image 52-2]



As stacked area charts on one of the foundational Show Me options and are extremely easy to create in Tableau, this chapter will focus more on the best applications of this chart type and a trick for using them most effectively.

As you can see in the previous image, when the mark type was changed from line to area, the values for each of the dimension members for Sub-Category were stacked on top of each other. This provided a new insight to us that was not available when the mark type was line: the monthly total across all 17 sub-categories. If it was more important to know the monthly total versus the exact

contribution of each sub-category, the stacked area chart is the right tool for the job.

However, if it is more important to know the individual trends of each sub-category, that becomes difficult to assess with a stacked area chart. The reason is that after the trend on the bottom of the stack, each subsequent trend inherits the trend below it. So by the time you get to the slice on top, the value equals the value of the first 16 slices combined, plus the value of the 17th and final dimension member. You can turn off the stacks by navigating to Analysis > Stack Marks and choosing 'Off', but at that point, you're better off with the line graph we started with.

One other big word of caution if you are using stacked area charts: never use them to stack rates, such as a click-through rate, open rate, or another type of conversion that divides a numerator by a denominator. This is again because the trends are stacked on top of each other, so if you have three dimension members, each with a conversion rate of 5%, the top slice would display the value of the first two stacks combined (10%) plus the value of the third and final dimension member (5%). In this scenario, the slice on top would display a value of 15%, far from its true 5% performance.

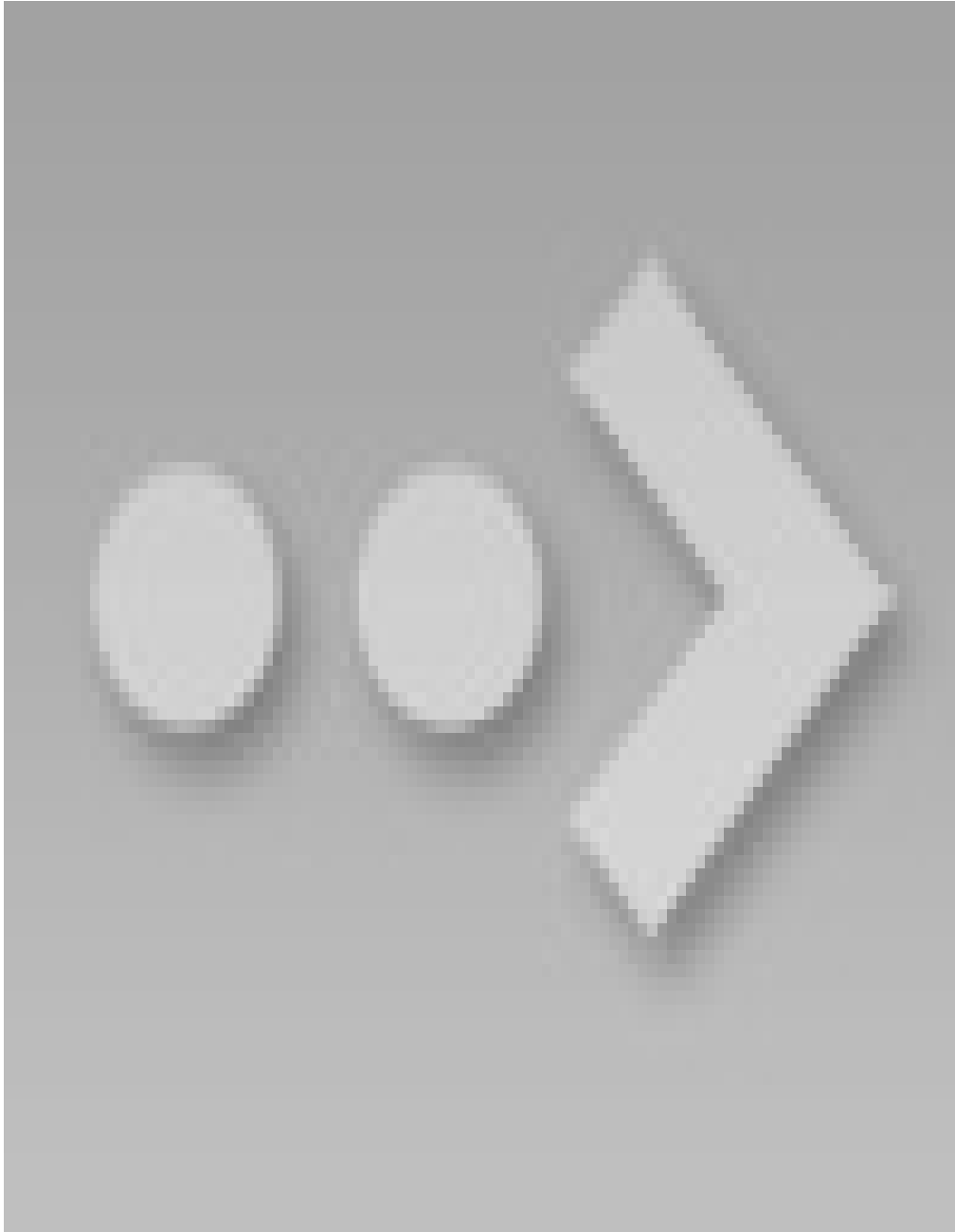
So when is a stacked area chart a good choice?

I like to use stacked area charts when the total axis equals 100% and each individual dimension member is displayed as a percentage of the total. This can be achieved by adding a quick table calculation for 'Percent of total' to the measure being displayed.

For a refresher on table calculations, see the chapter, An Introduction to Tableau Table Calculations.

By default, Tableau calculates table calculations across the table. So if I were to add a quick table calculation for percent of total to the stacked area chart in the previous image, I see this:

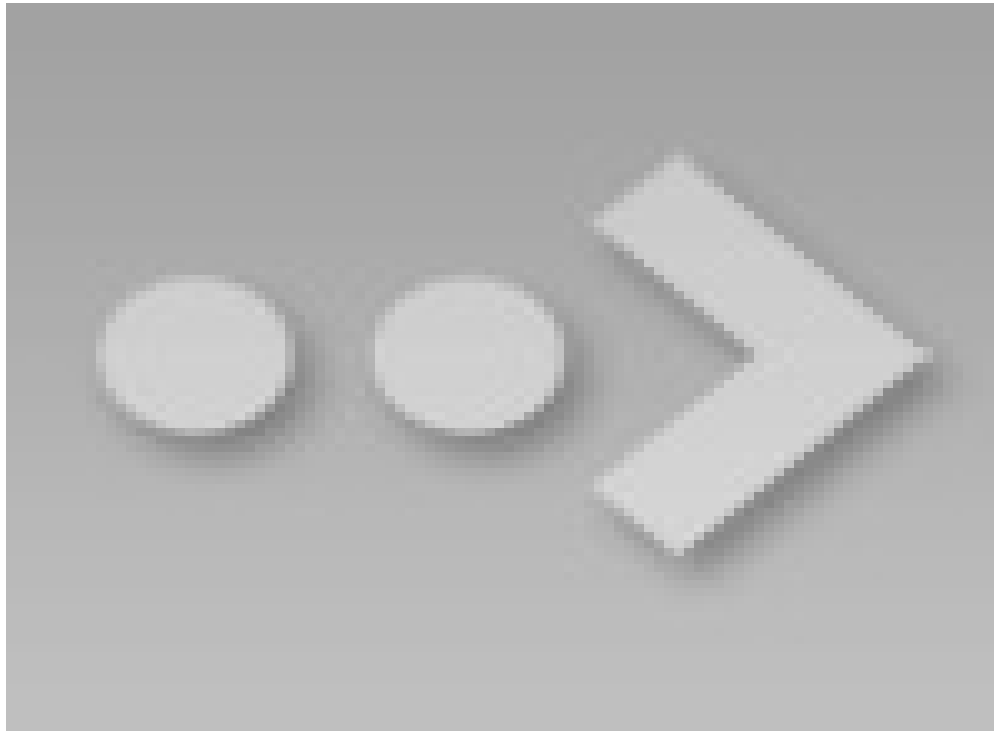
[Insert Image 52-3]



With this view, the trends look exactly the same, but the axis is being displayed as a percent of total instead of the raw revenue values. To change the view so that the axis totals 100% and the areas represent each dimension member's contribution to each month's total, the table calculation needs to be changed to compute using Table (Down). This is changed by clicking on the measure with the table calculation for percent of total, now designated with a delta symbol, hovering over 'Compute using', and choosing 'Table (Down)':

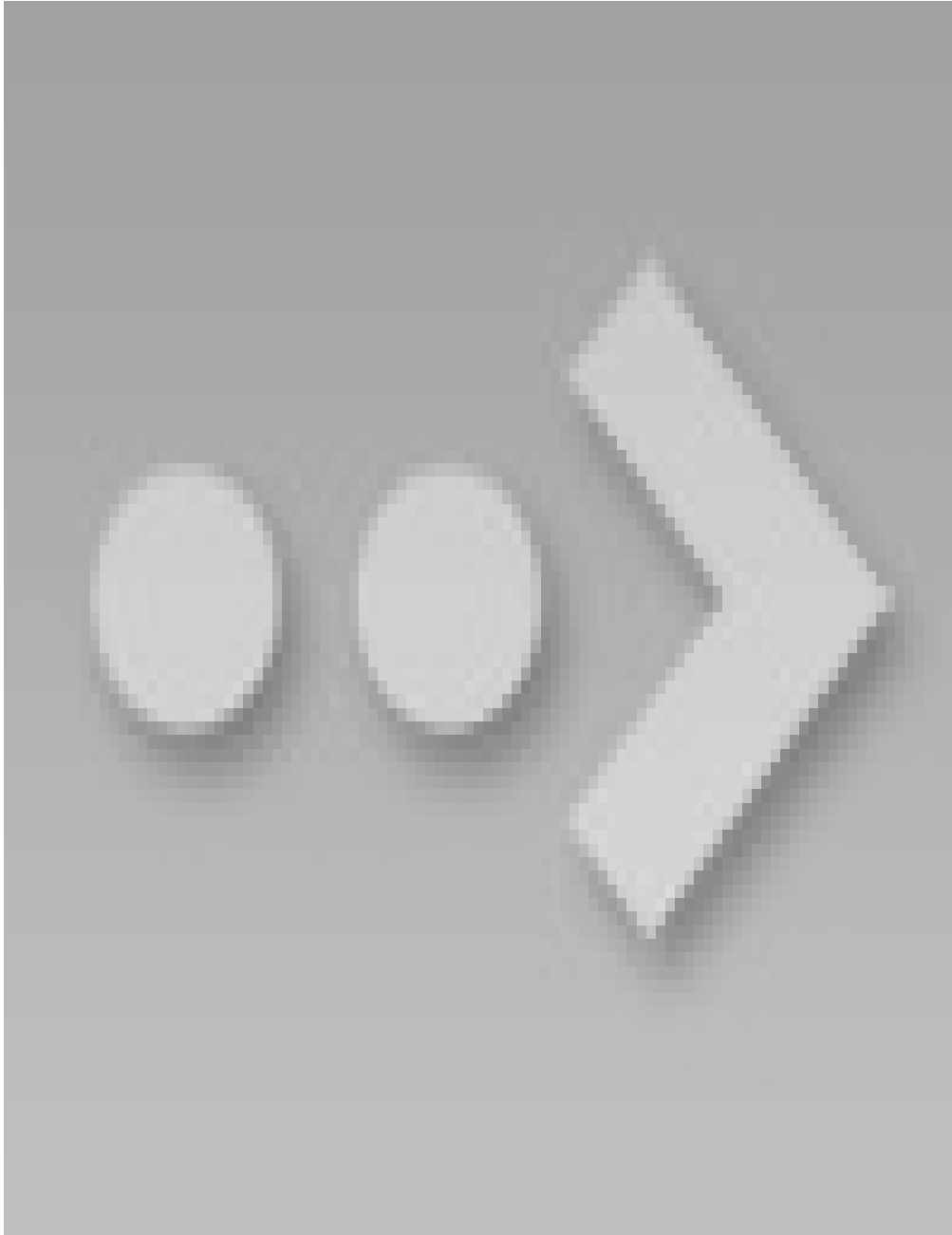
[Insert Image 52-4]

Figure 52-4



The view looks as follows after changing the direction of the table calculation, making it easier to see the monthly contribution for each individual dimension member:

[Insert Image 52-5]



Unless the slice is on the top or the bottom, it can still be challenging to evaluate the trends for individual dimension members with a great deal of precision. At the very least, it should be easier to evaluate which dimension members are gaining or losing share from month to month. As with most analyses in Tableau, the chart selection should be informed by the business question at hand. Stacked area charts, especially when combined with table calculations, are a good visualization to have in your toolbelt to display how distributions in your business are changing over time.

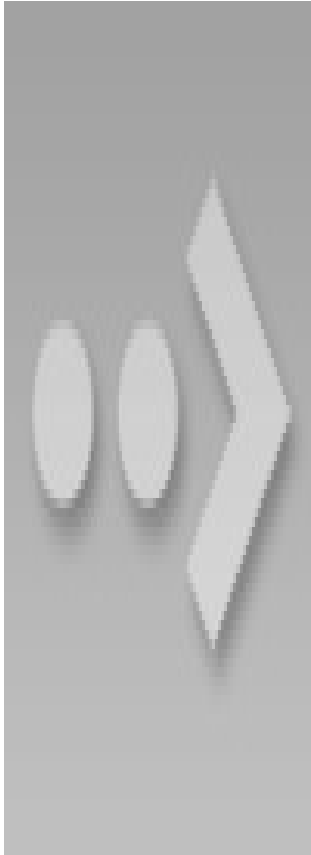
Chapter 53 – How to Make a Histogram

Histograms are one of the most effective chart types for showing a distribution of quantitative data at one point in time. Similar to a bar chart, the important distinction is that histograms are used to plot continuous, numerical data while the bar chart is used to plot discrete, categorical data. When you create a histogram, bins are created to group equally-sized numerical ranges. Despite being continuous and quantitative, these bins can be thought of as the dimension that you slice and dice the count of records by to create the histogram.

How to make a histogram in Tableau

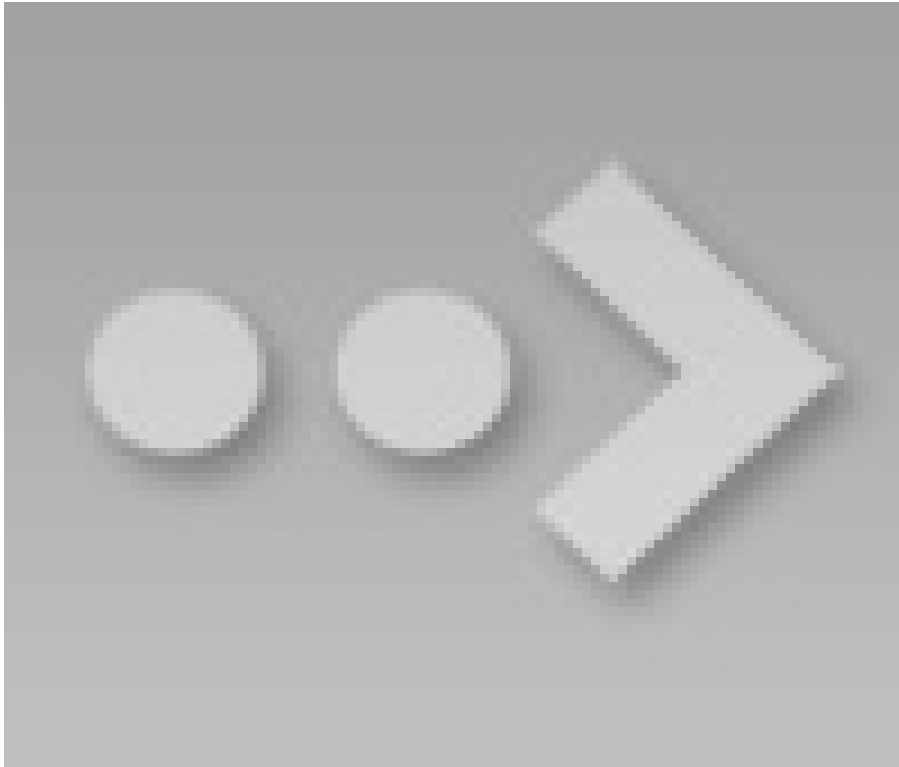
Histograms are one of the 24 chart types in Tableau that can be created using the Show Me tool in the top right corner of the Tableau Desktop interface. I usually like to explain how a visualization is created manually in Tableau instead of relying on Show Me, but histograms are one of my few exceptions. Histograms are created in Tableau using just one measure. To create a histogram, pre-select the measure that you want to visualize the distribution for by clicking on it in the Measures Shelf. Then navigate to the Show Me options and choose “histogram”.

[Insert Image 53-1]



Here's how a histogram looks if you follow the steps above for the Quantity measure in the Sample – Superstore dataset:

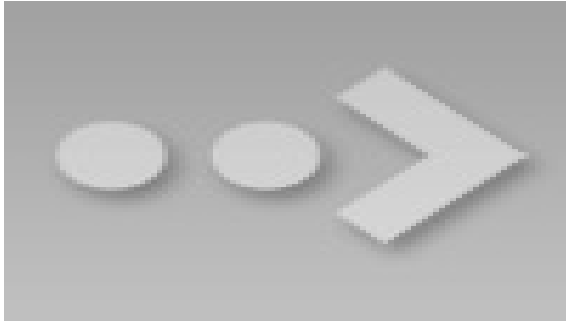
[Insert Image 53-2]



When this chart was created, Tableau conveniently created the dimension for “Quantity (bin)” to provide the equally-sized ranges which are required to create a histogram. The bars then represent the count of records within each of these ranges. Notice that the bars have no space between them; this is how some authors like to differentiate the bars with continuous data in a histogram from the bars with categorical data in a bar chart.

Depending on how the distribution looks, you may choose to change the size of the bins. This is easy to accomplish by locating the newly created Quantity (bin) dimension on the Dimensions Shelf, right-clicking on the field, and choosing “Edit”. A new interface will appear that shows you the current bin size and allows you to change the equal ranges:

[Insert Image 53-3]



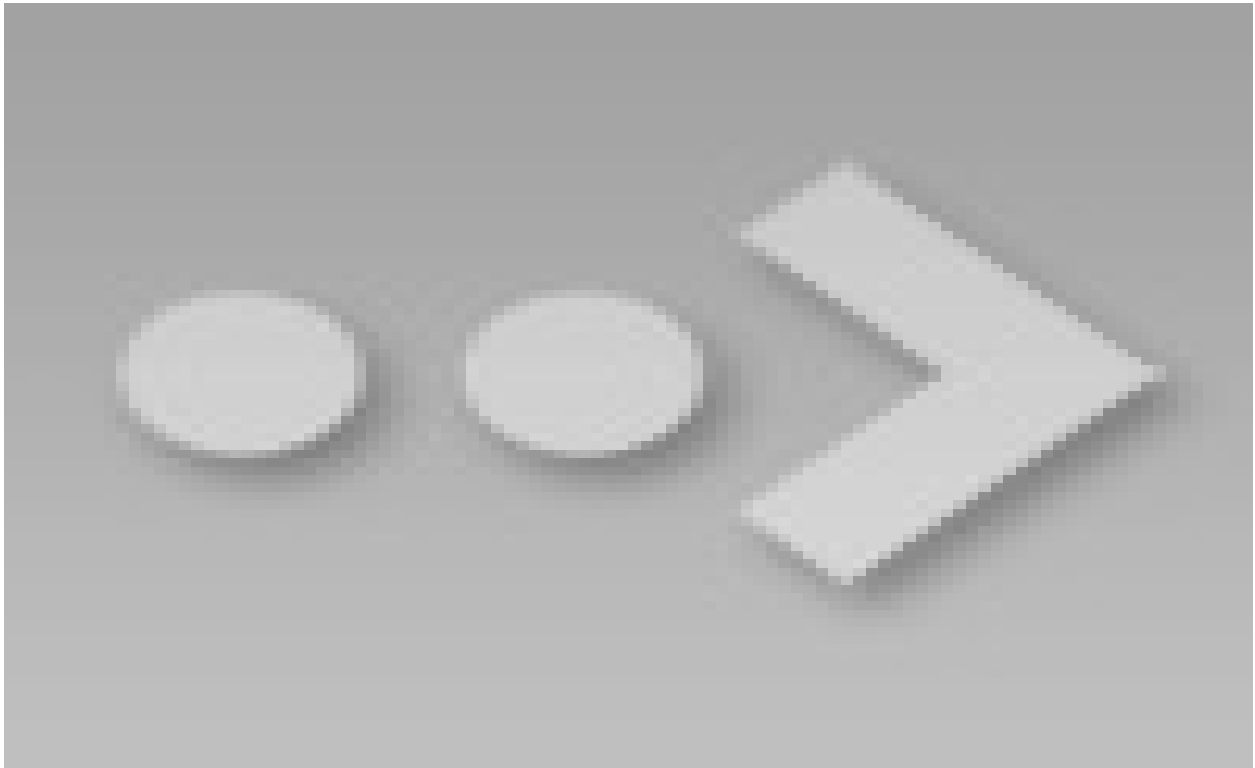
Here, you can either type in a number, or click the dropdown arrow to have the bin size be based on a parameter. Let's say for simplicity that we want to change the bin size for the quantity measure to one. This analysis would show the end user how many records (i.e. sales) included one item, two items, three items, and so on. Here's how the histogram looks after editing the bin size:

[Insert Image 53-4]



Lastly, because the bin ranges are continuous, we have a continuous axis that begins at zero and ends at one bin past our maximum bin size. If you need to change the axis range, right-click on the X-axis and choose "Edit axis...". Here is how my final histogram for quantity looks after fixing the axis range from 1 to 15:

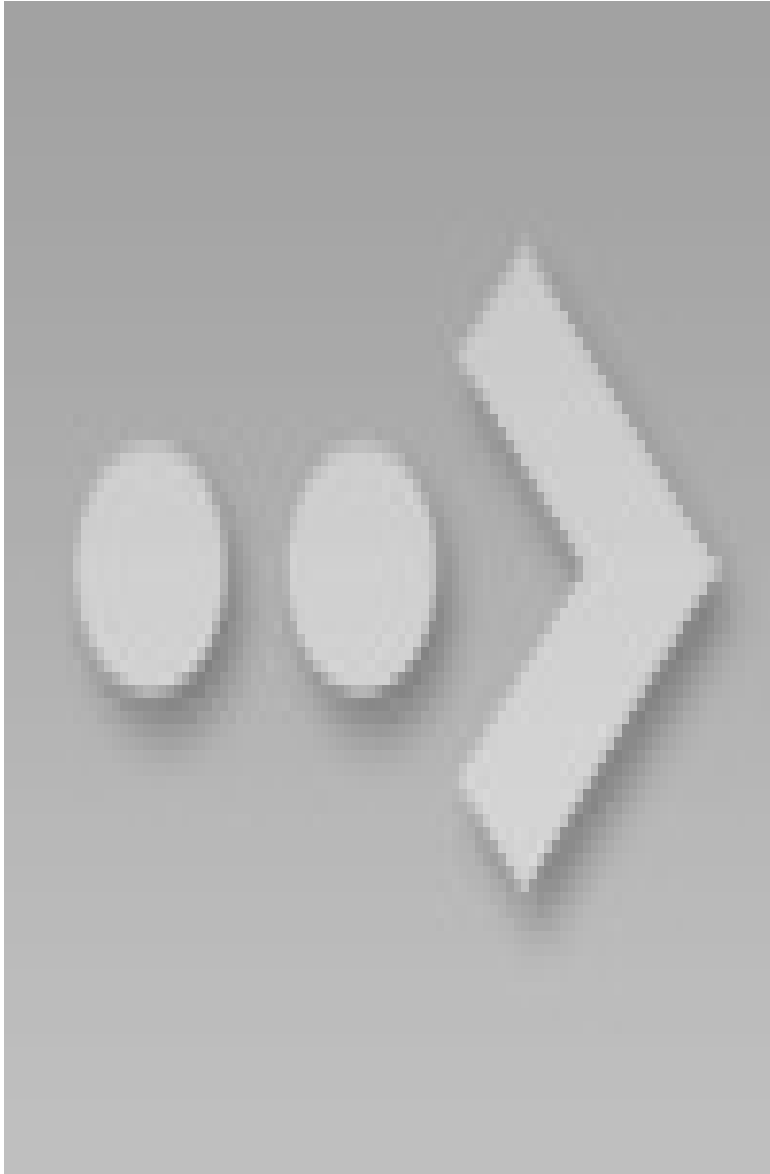
[Insert Image 53-5]



Chapter 54 – How to Make a Box-and-Whisker Plot

The Box-and-Whisker Plot, or Box Plot, is another effective visualization choice for illustrating distributions. Along with histograms and stacked area charts, Box-and-Whisker plots are among my favorite chart types used for this purpose. They work particularly well when you want to compare the distributions across two different dimension members side-by-side, where one set of dimension members make up the X-axis, and the other dimension member is used as the visualization's level of detail. To help illustrate, here's the box-and-whisker plot we will create with the Sample – Superstore dataset during this tutorial:

[Insert Image 54-1]



As you can see, each set of circles corresponds to the dimension members on the X-axis for the Sub-Category dimension. The level of detail, or most granular level of the analysis, is Month of Order Date. Since the level of detail is month of order date, each Sub-Category column has 12 circles, one for each month of the year.

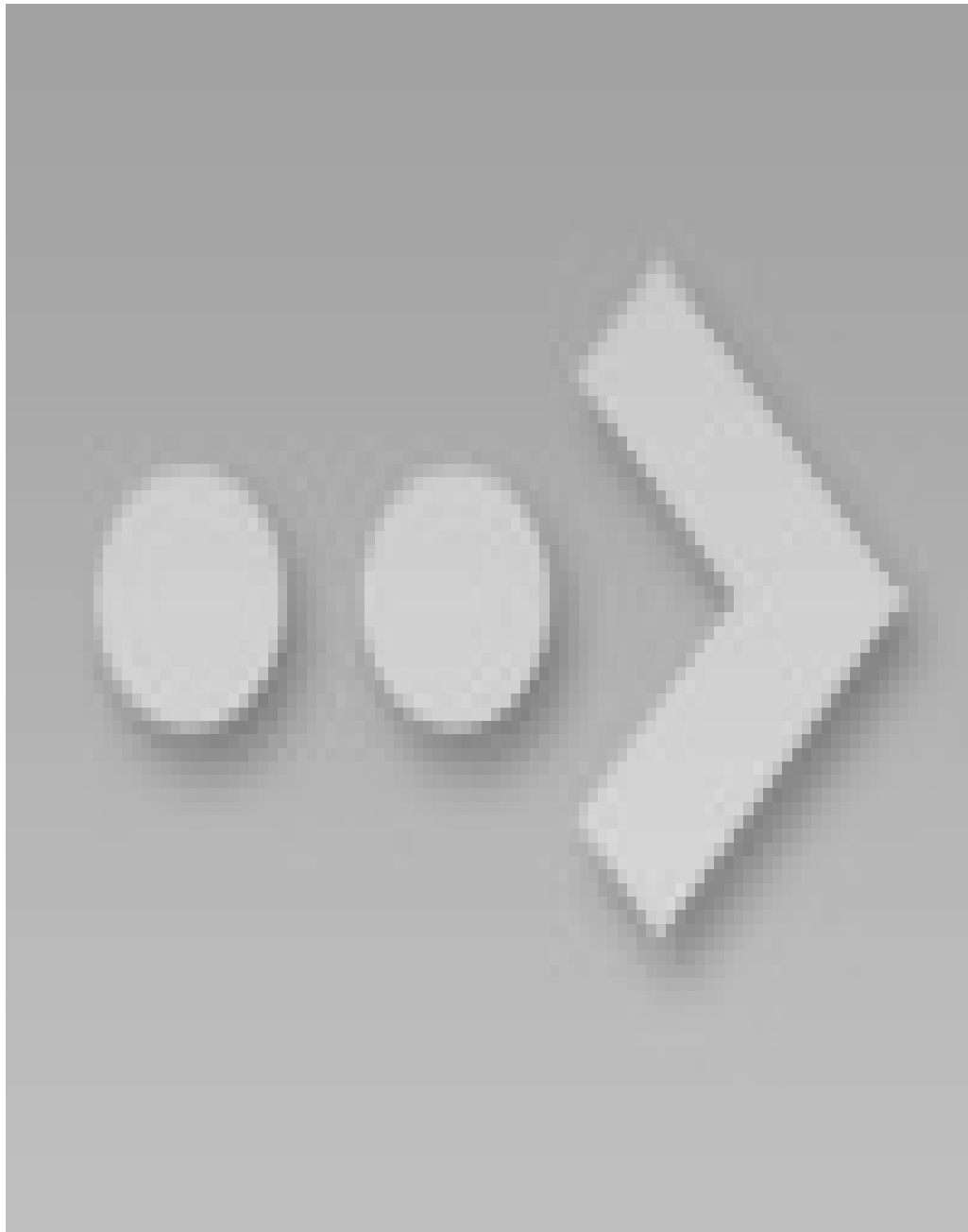
In short, this visualization is showing how the distribution of monthly sales vary between product sub-categories. While I can easily find several insights in this visualization and believe box-and-whisker plots to be among the most effective ways to communicate distributions, I find them to be one of the most misunderstood chart types when I attempt to share them with an external audience.

For this reason, this chapter shares not only how to make Box-and-Whisker Plots in Tableau, but how to read them.

How to make a box-and-whisker plot in Tableau

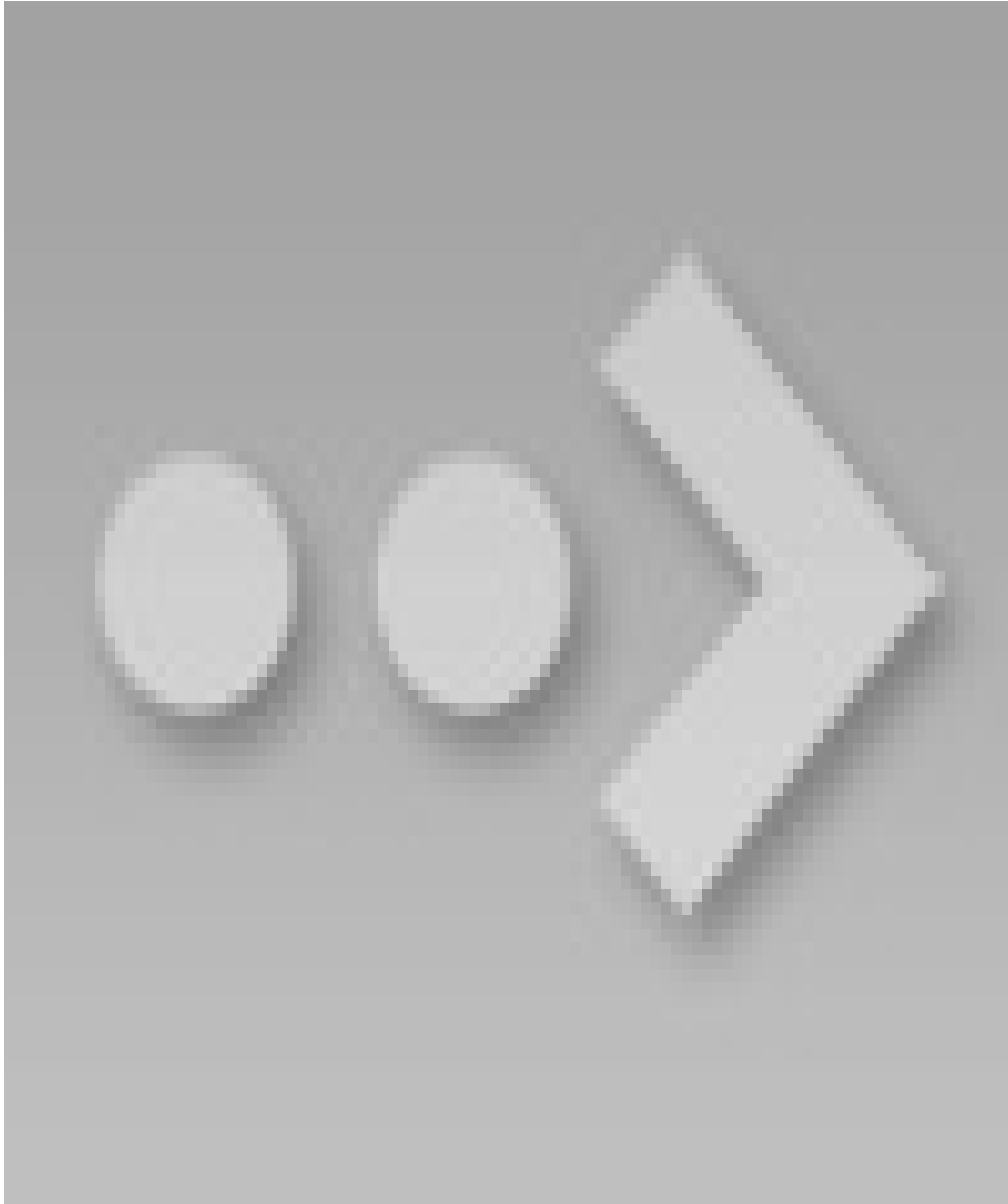
Box-and-Whisker Plot is one of the out-of-the-box Show Me options in Tableau, but they are actually created with reference lines – which is what we'll show here. To create a box-and-whisker plot, start by creating a bar chart with the dimension and measure of interest. In the example above, we are looking at Sales by Sub-Category.

[Insert Image 54-2]



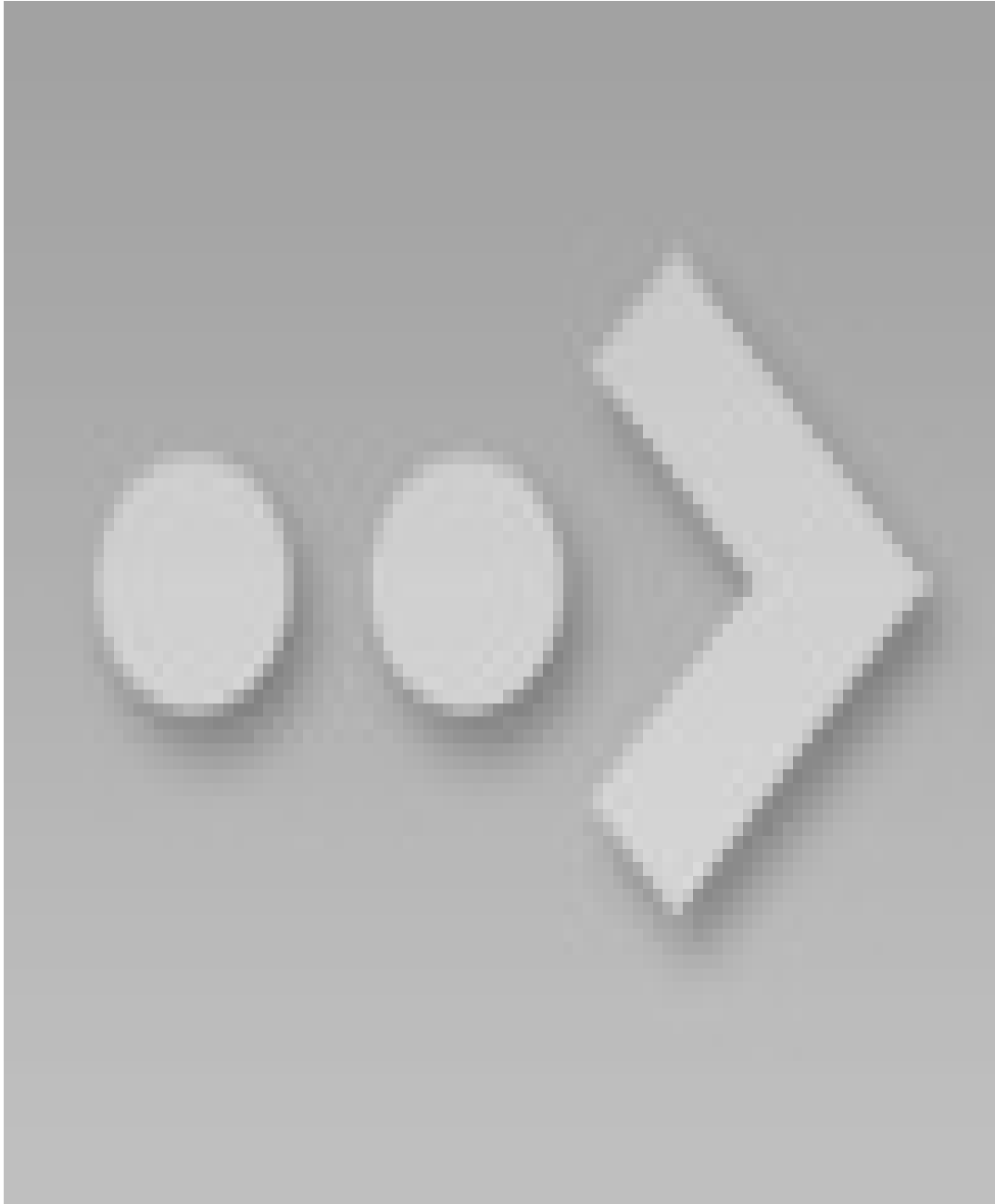
Next, add the distribution that you care about to the Detail Marks Card. In this case, we are looking at how Sales are distributed by Sub-Category, by *Month of Order Date*. So in this example, Month of Order Date is added to the Detail Marks Card.

[Insert Image 54-3]



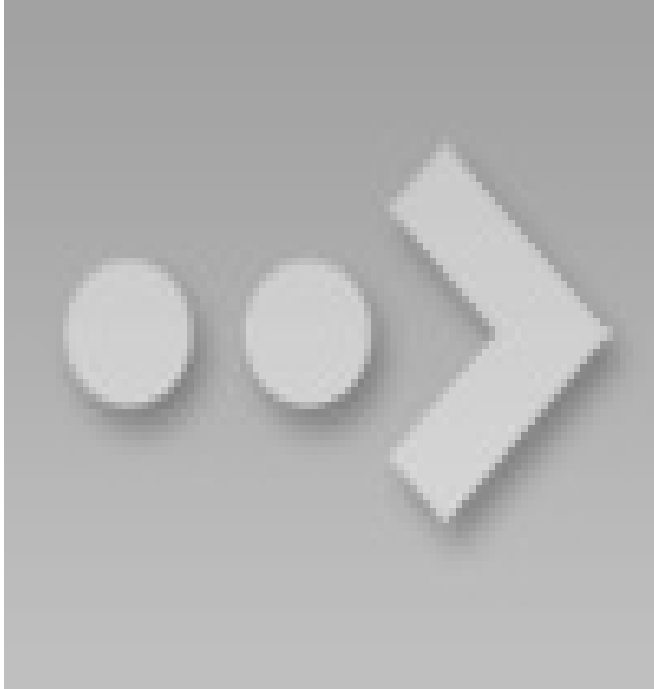
By increasing the level of detail, a stacked bar chart is created, with each stack per sub-category representing one of the twelve months of the year. To set the foundation of the box-and-whisker plot, convert this stacked bar chart to a dot plot by changing the mark type from Automatic (Bar), to Circle.

[Insert Image 54-4]



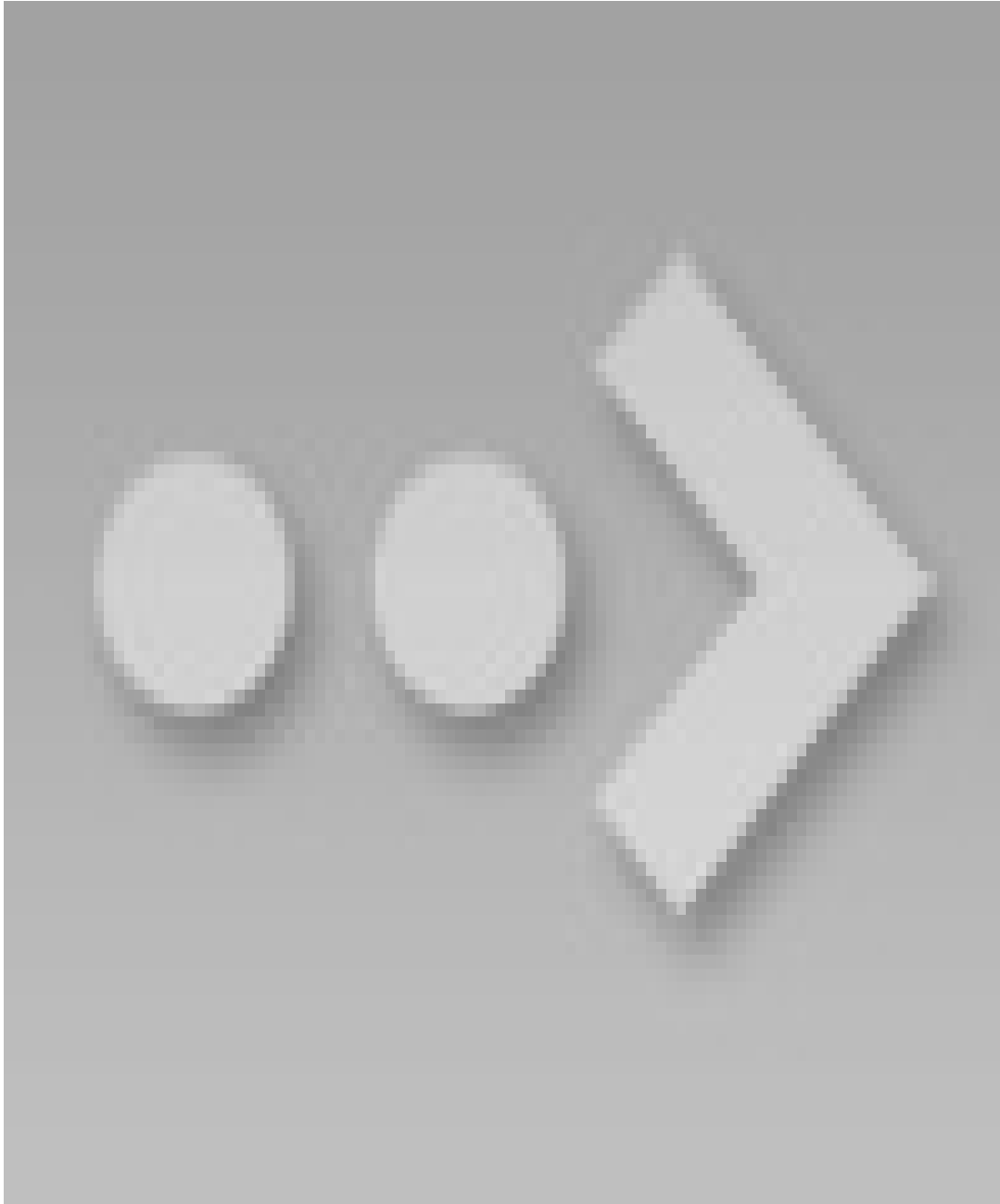
Lastly, to create a box-and-whisker plot, right-click on the Y-Axis, and choose “Add Reference Line”. When the add reference line dialog box appears, click on the choice for Box Plot. There are some formatting options available, but the default settings are usually best:

[Insert Image 54-5]



IQR stands for Interquartile Range, which are the data points between the first and third quartile. So the default options are telling Tableau to make all of the data points on the box-and-whisker plot fit into 1.5 times the IQR; anything outside of that range is an outlier. This sounds confusing and is probably why I don't see this chart type getting much traction, but this chart type provides a lot of helpful context and is easy to read once someone explains it to you. Let's take another look at the final box-and-whisker plot from this tutorial:

[Insert Image 54-6]



Each line on the box-and-whisker plot provides a piece of statistical context. The most important line is the one right in the middle of each “box”, which represents median. With median displayed, you can quickly look across the dimension members and compare medians, regardless of how big or small the range of values is within each column.

That alone is very useful for an analysis, but the rest of the lines also have a meaning. Working out from the median, the next set of lines is showing plus or minus one quartile from the median.

Lastly, the upper whisker is 50% higher than the IQR, or “middle fifty”, which are the data points within the first and third quartile. The lower whisker is 50% lower than the IQR. Any data points outside of the box-and-whisker are considered outliers.

Now not only can you make a box-and-whisker plot in Tableau, you know how to use them to get the most out of your analyses!

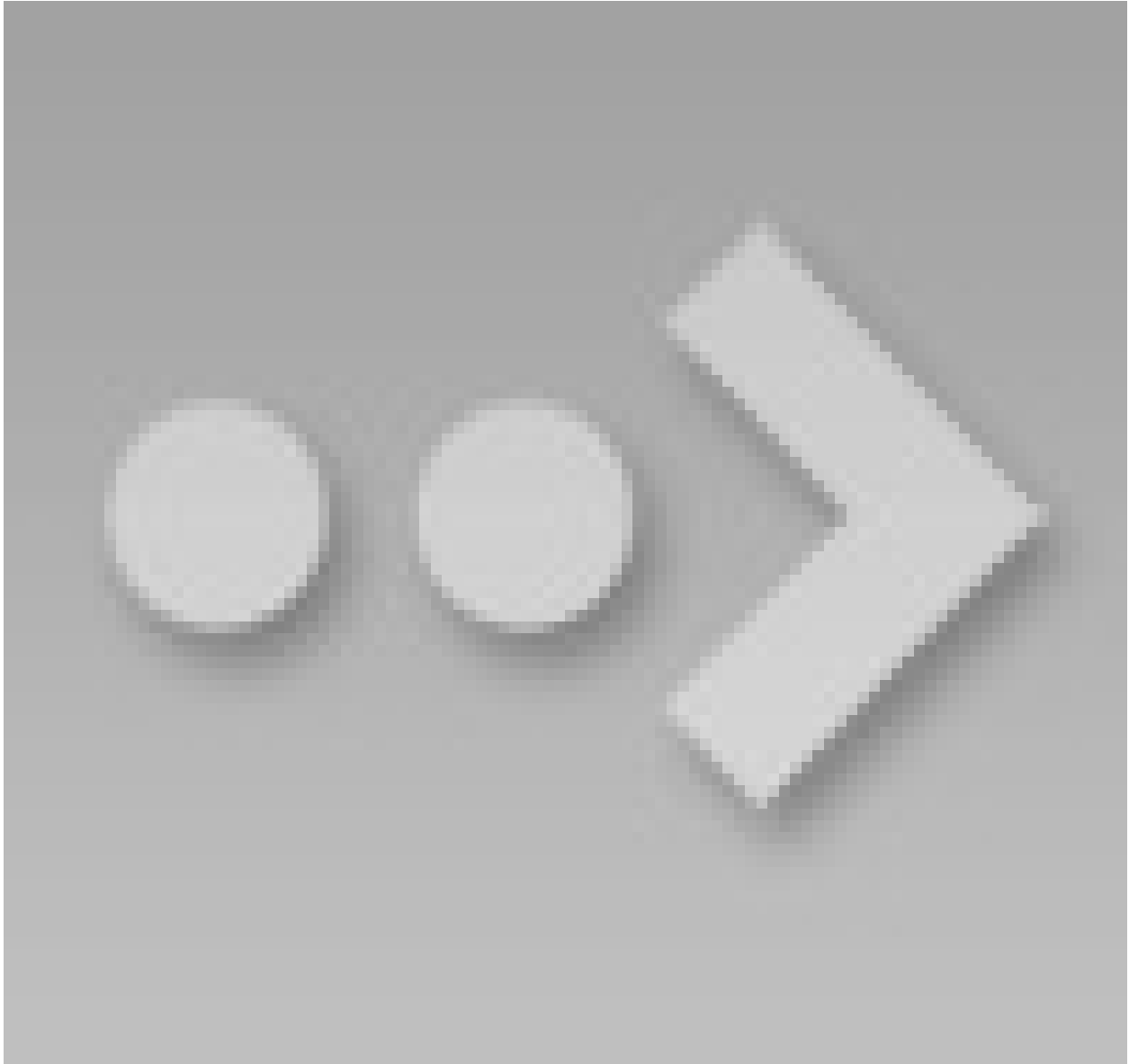
Chapter 55 – How to Make a Symbol Map with Mapbox Integration

Maps, which were introduced with Tableau 4.0 in August 2008, are one of the most powerful visualization types available. The power of maps comes from their inherent ability to leverage schemas that you have been building up for many years. In the Data-Driven Storytelling tip, Eliminate Chart Junk (but Not Graphics), I showed how an image of a map helps you decode dozens of Latitude / Longitude pairs almost instantly.

This chapter shares how to harness this power by creating symbol maps and how to take your maps a step further by integrating Tableau with Mapbox maps.

First, let’s take a look at how my final map example will look:

[Insert Image 55-1]

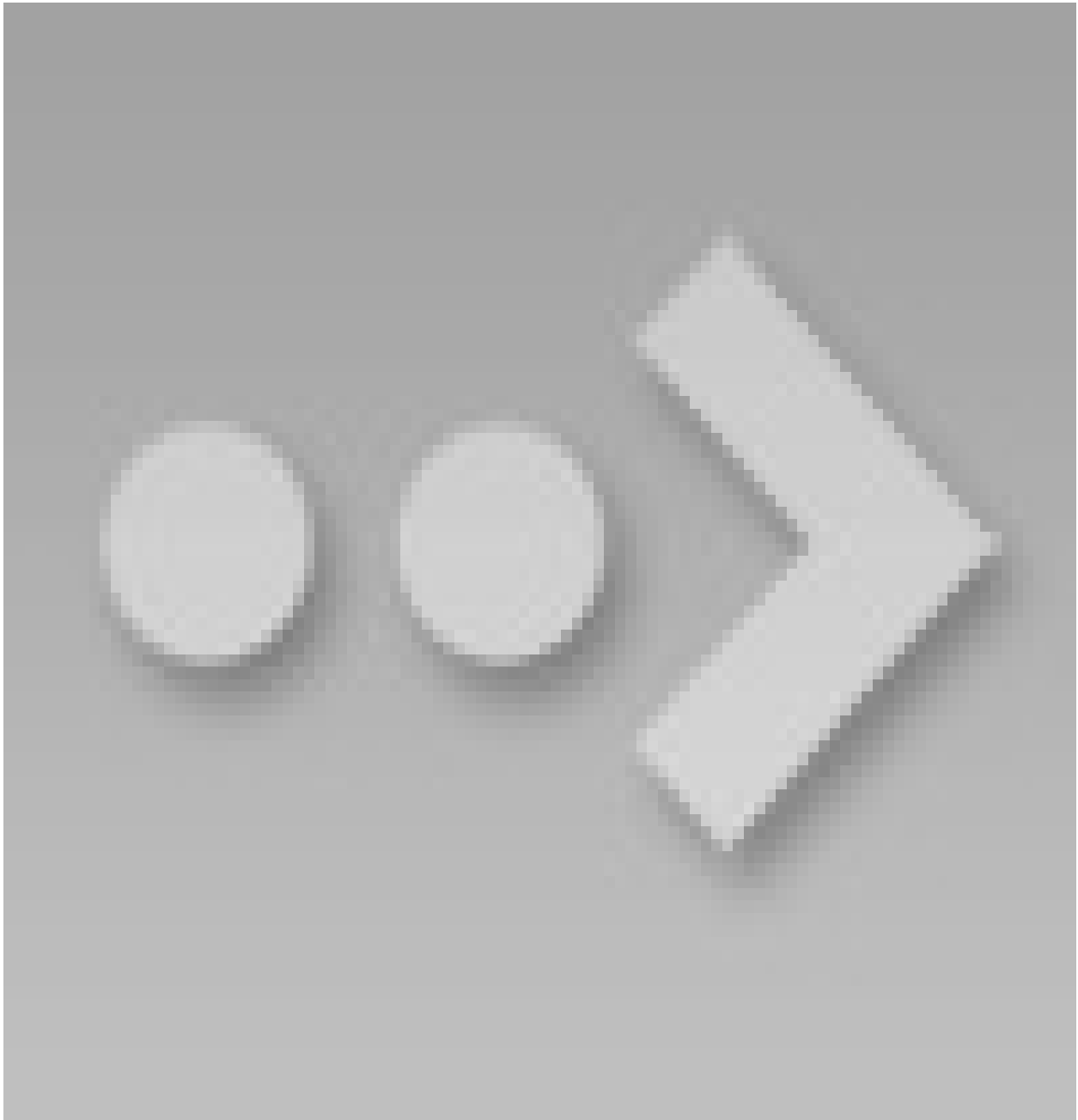


This tutorial uses the Sample – Superstore dataset.

How to make a symbol map in Tableau

The easiest way to start a symbol map in Tableau is to double-click on a geographic dimension from the dimensions shelf on the left-side of the interface. You know that Tableau recognizes your geographic fields as map-compatible if there is a globe icon next to the dimension. In my example, I am going to double-click on Postal Code to start the view.

[Insert Image 55-2]

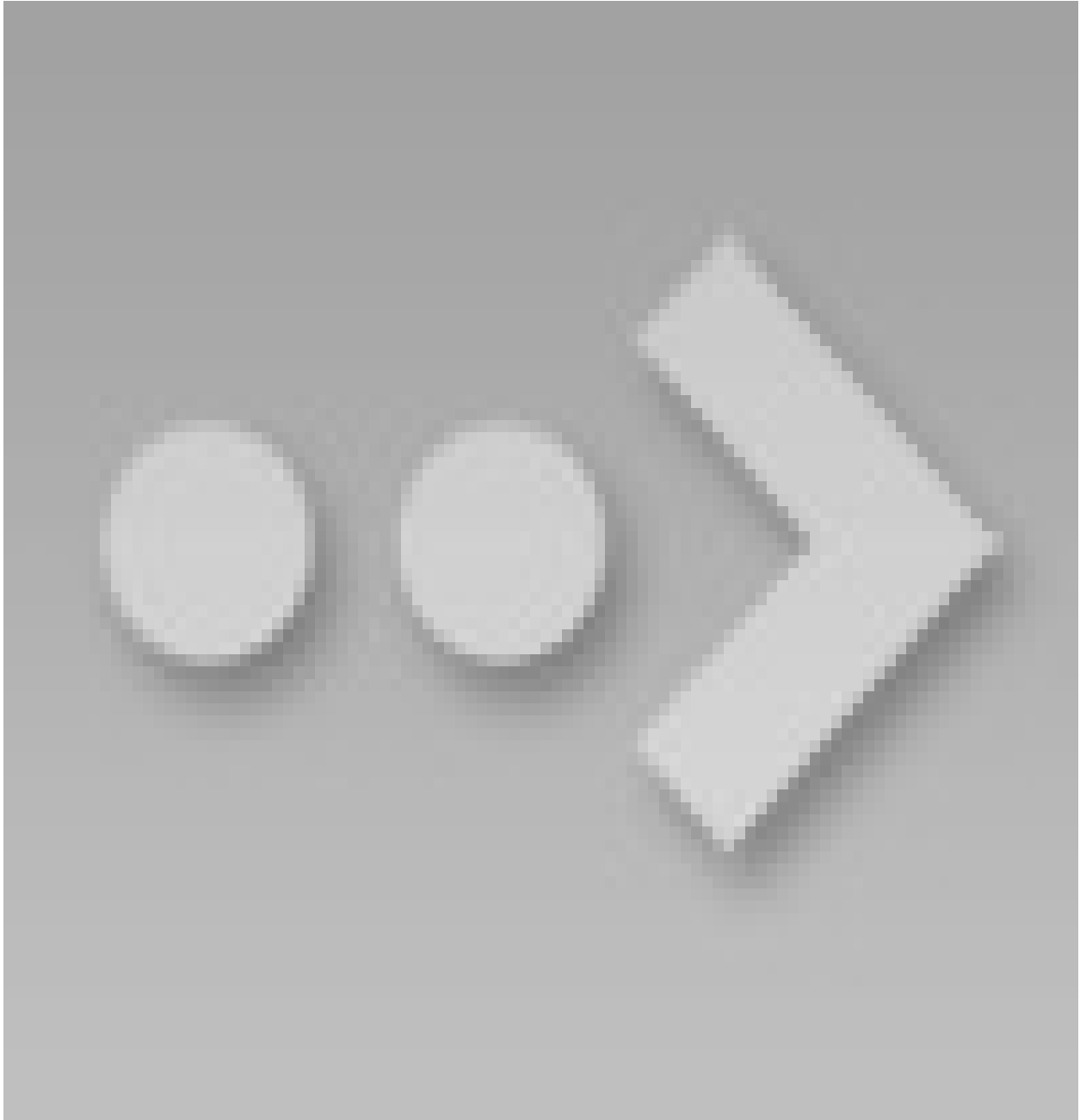


Notice that Tableau put Longitude on the columns shelf and Latitude on the rows shelf, and each circle represents the intersection of each pair. At this point, I technically already have a symbol map, but there are several ways I can add value to the visualization. You may want to change the ‘symbol’ from the default circle to a square or other shape. Perhaps you want to size and/or color

the symbols by a measure such as sales by placing measures on the appropriate marks cards.

I personally like the default circles because you can add a border and they look nice with transparency, so I am going to stick with that. I will change the color to something that pops more by clicking on the color marks card and size the bubbles by sales by placing that measure on the size marks card. At this point, my view looks like this:

[Insert Image 55-3]



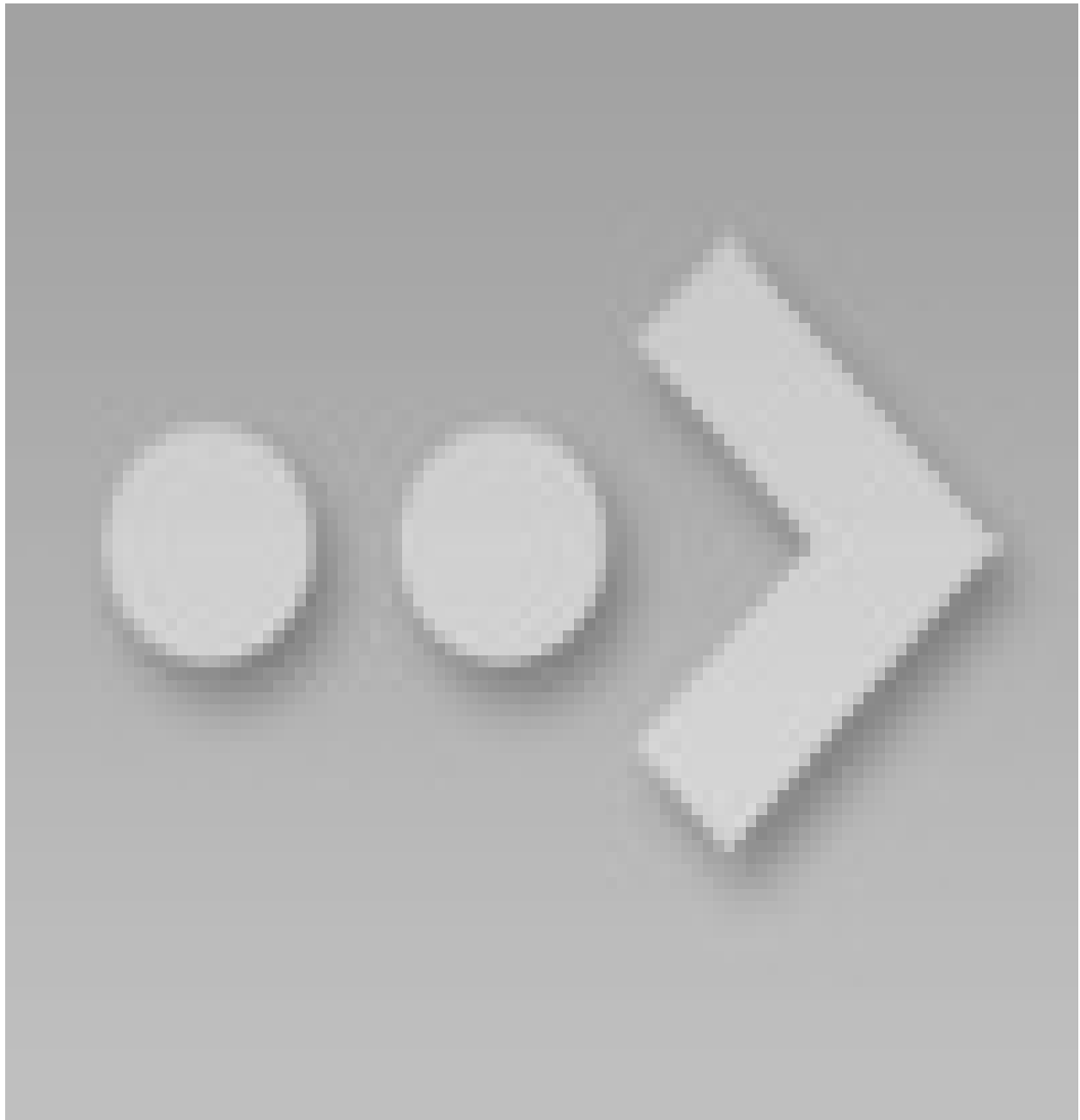
This is a solid symbol map at this point, but you can also add additional data layers or choose from three map styles by navigating to Map > Map Layers.

How to add Mapbox maps

Tableau provides three map styles out of the box, but starting in Tableau version 9.2, it is very easy to integrate your Tableau workbooks with Mapbox, a custom map designing service, to access 14 additional map styles. In order to access these new styles, follow these simple steps:

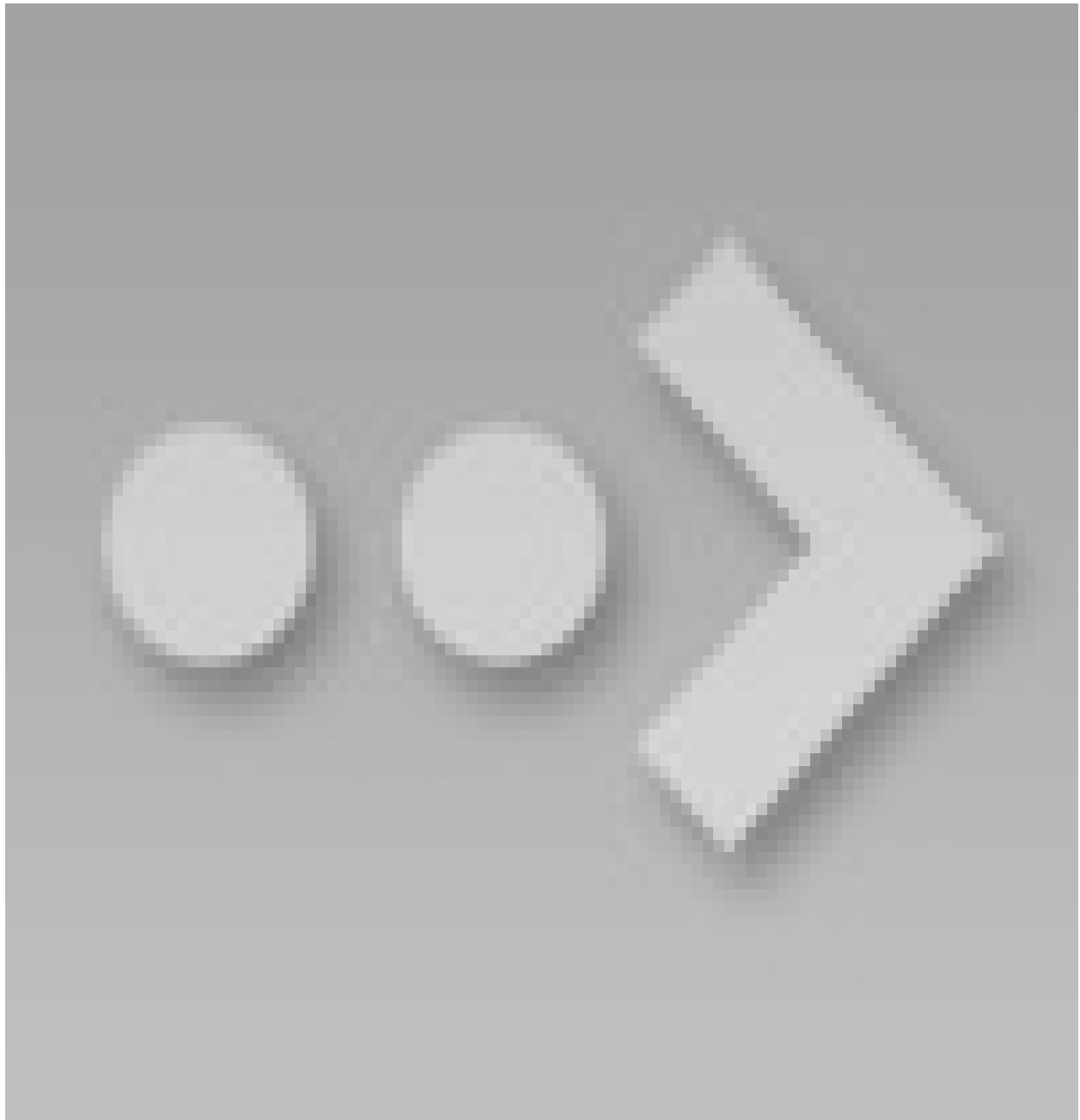
- Go to mapbox.com
- Signup via the button in the top right corner
- Once you're signed up, navigate to the Home tab, and copy your access token on the right side
- From within your map view in Tableau, navigate to Map > Background Maps > and click Map Services
- From the new dialog box, click 'Add...' and choose 'Mapbox Maps'
- This is where you give your custom map a name, paste your access token, and choose one of the styles

[Insert Image 55-4]



For my example, I chose the high contrast Mapbox style, but there are 13 additional styles to choose from! After clicking ‘Okay’ and closing the dialog box, I am left with a nice-looking Tableau symbol map with an integrated Mapbox style.

[Insert Image 55-5]



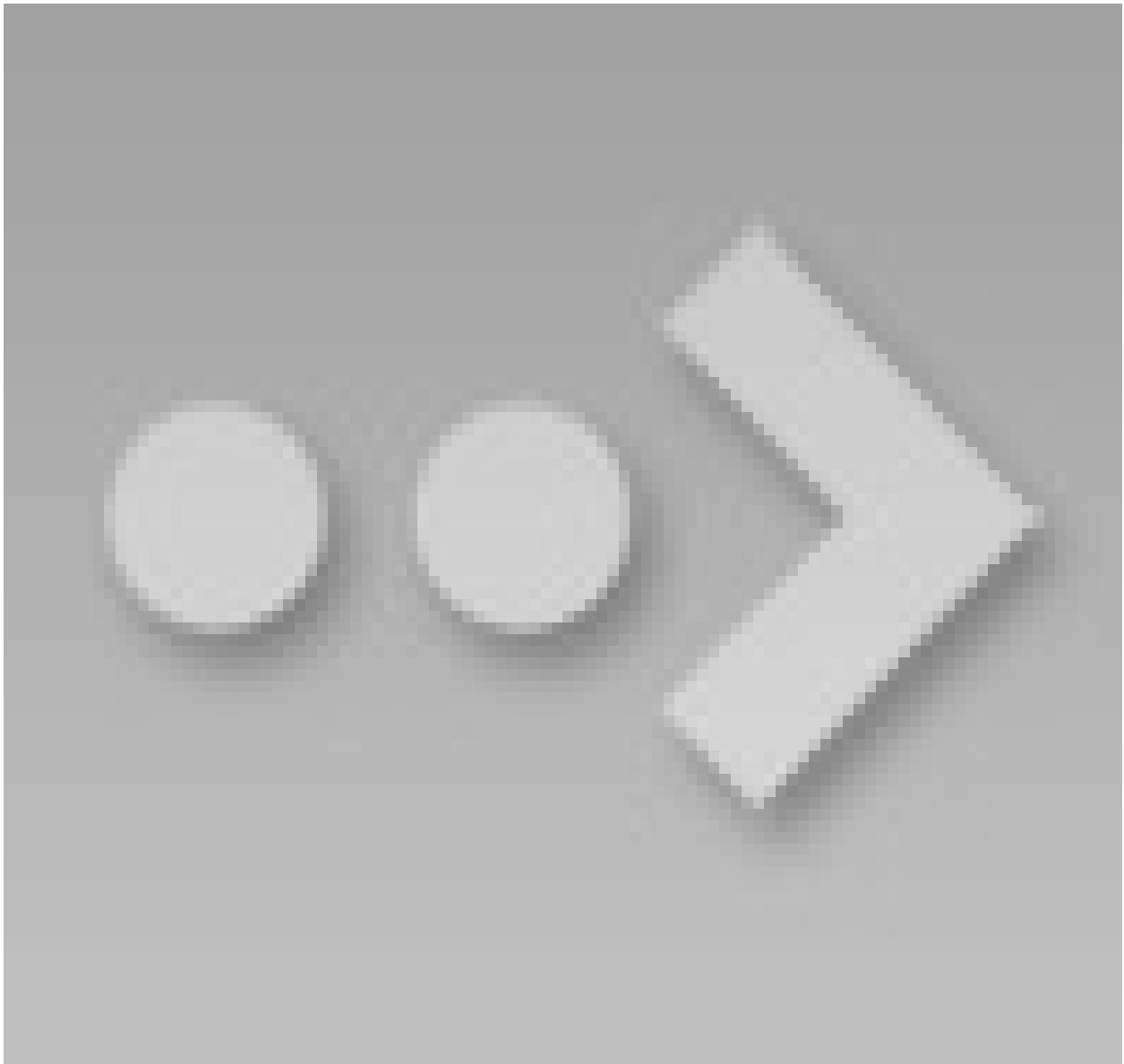
Chapter 56 – How to Make a Filled Map

In the last chapter, I shared how to create symbol maps in Tableau with custom Mapbox maps, and how mapping in Tableau leverages schemas that you are familiar with to help you rapidly decode latitude and longitude pairs. Maps in

Tableau can be thought of as scatter plots, but the background image of a map helps you orient yourself almost instantly with the different geographic territories being displayed.

Filled maps in Tableau are similar to symbol maps, but they include many more data points. While a symbol map draws a symbol at the intersection of each latitude and longitude pair, filled maps draw a polygon around the entire border. Here's one example of a filled map colored by region in Tableau:

[Insert Image 56-1]

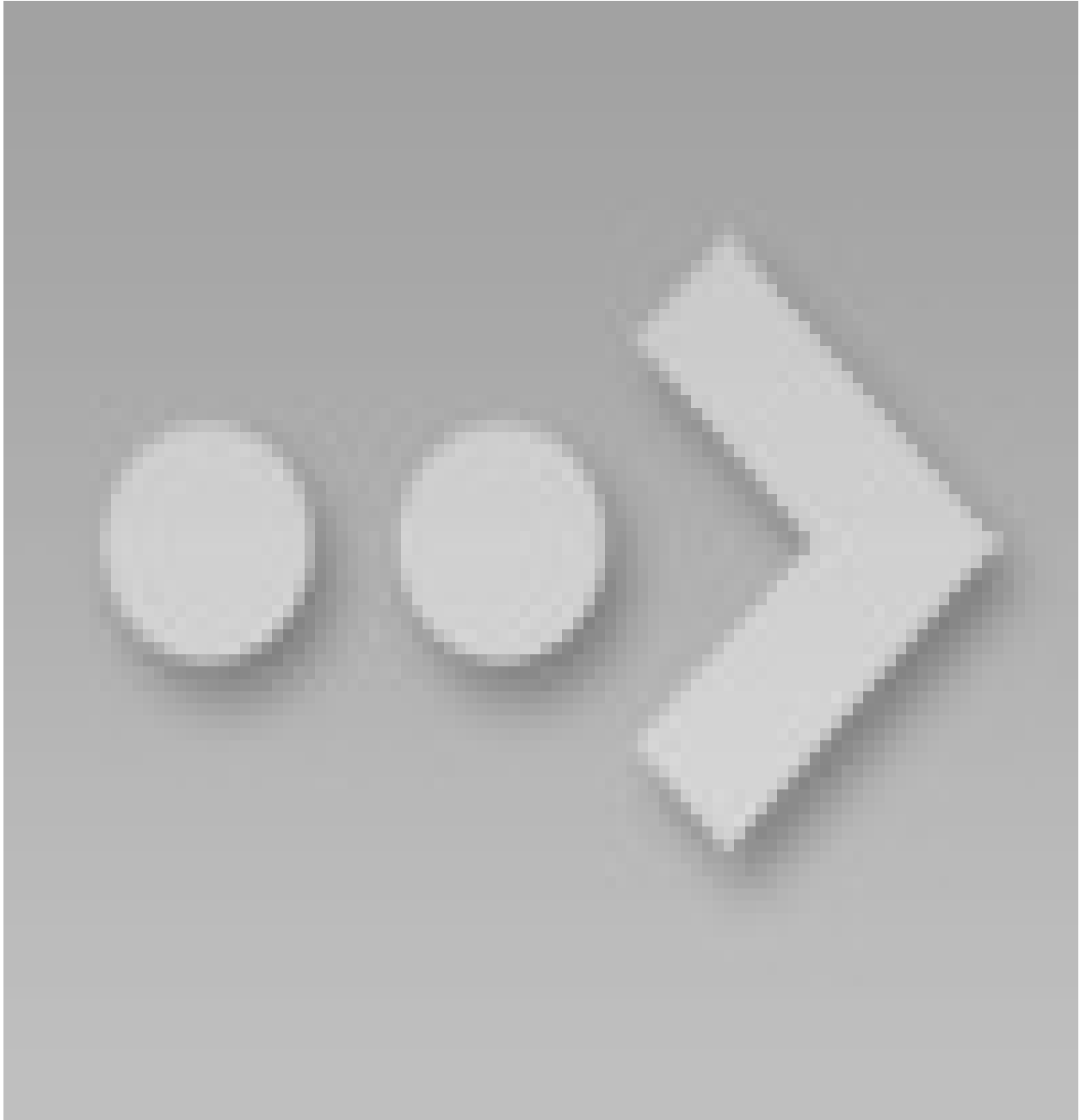


How to make filled maps in Tableau

Filled maps are one of the easier chart types to create in Tableau using Show Me. To do so, simply click a geographic dimension (identified by a globe icon) from the Dimensions Shelf and choose 'filled maps' under Show Me. That being said, I prefer to create chart types manually, especially when I am learning. By creating the charts without relying on Show Me, it helps me understand what Tableau is doing behind the scenes to generate each chart type.

To create a filled map manually, I will double-click on the geographic dimension for State in the Sales – Superstore dataset that comes with Tableau.

[Insert Image 56-2]



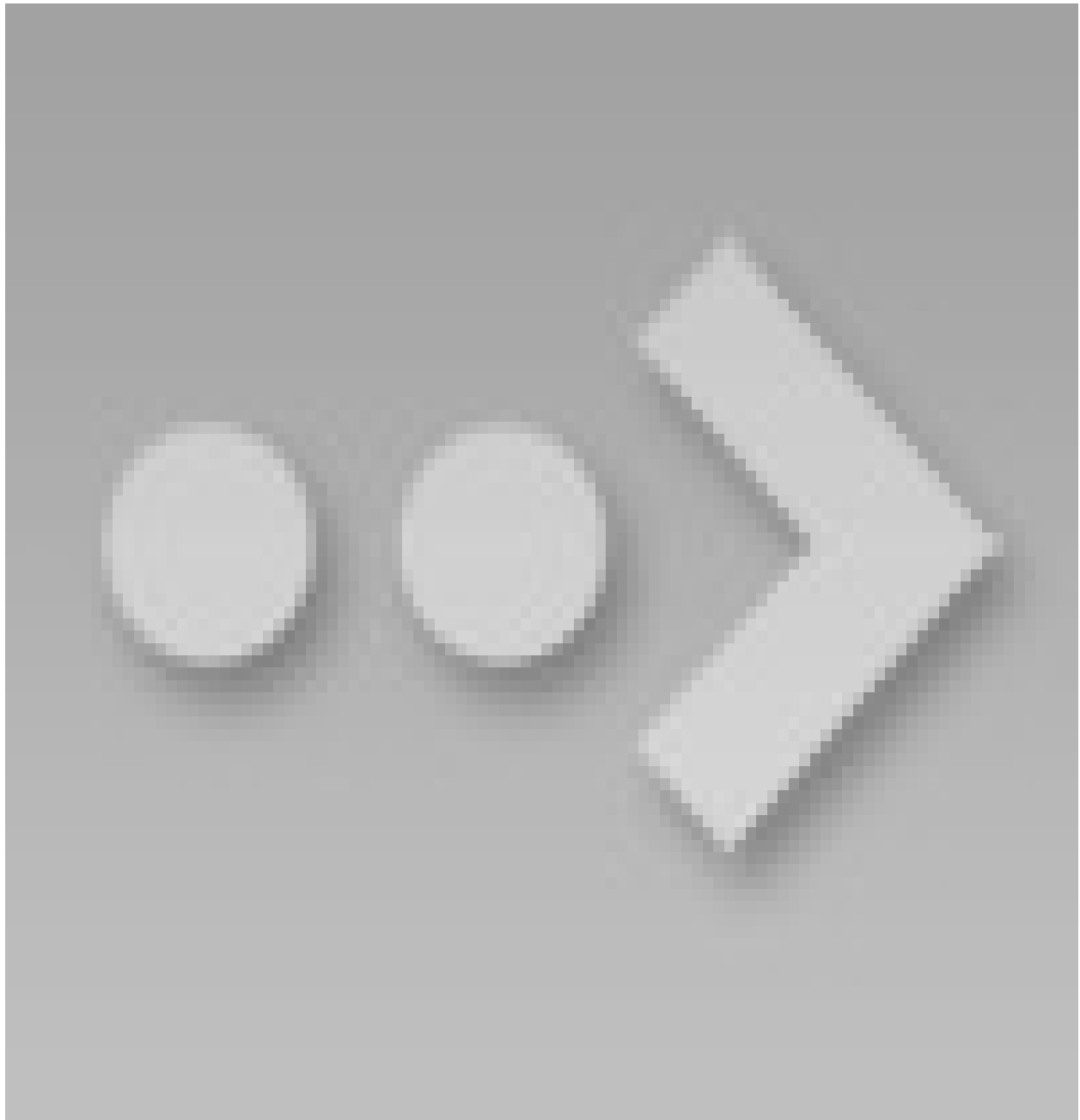
By default, Tableau generates a symbol map, placing a circle at the intersection of Longitude and Latitude for each state. In keeping with the “looking under the hood” theme of not relying on Show Me, take a look at the location of each dimension and measure. First, Longitude is on the Columns Shelf, which can also be thought of as the X-axis. Conversely, Latitude is on the Rows Shelf, or the Y-axis. On the Marks Shelf, we can see that State is the most granular level of detail in the view. Even without seeing the map in the view, we should be able

to guess what Tableau will display just by seeing the geographic fields on the Columns Shelf and Rows Shelf, the level of detail on the Marks Shelf, and the mark type of 'Automatic' (which is 'Circle' by default).

In order to change this from a symbol map to a filled map, change the mark type from 'Automatic' to 'Filled Map'. By selecting this special mark type in Tableau, you will see that the single circles on each state have been converted to nice, smooth polygons that trace the entire border of each state.

From here, you can encode the filled map by color by placing a field on the Color Marks Card. Perhaps you want to color the territories by a measure such as Sales or Profit. You can also color the marks by a dimension, as is the case in this example, where the states are colored by the Region dimension. The map view ends up looking like this:

[Insert Image 56-3]

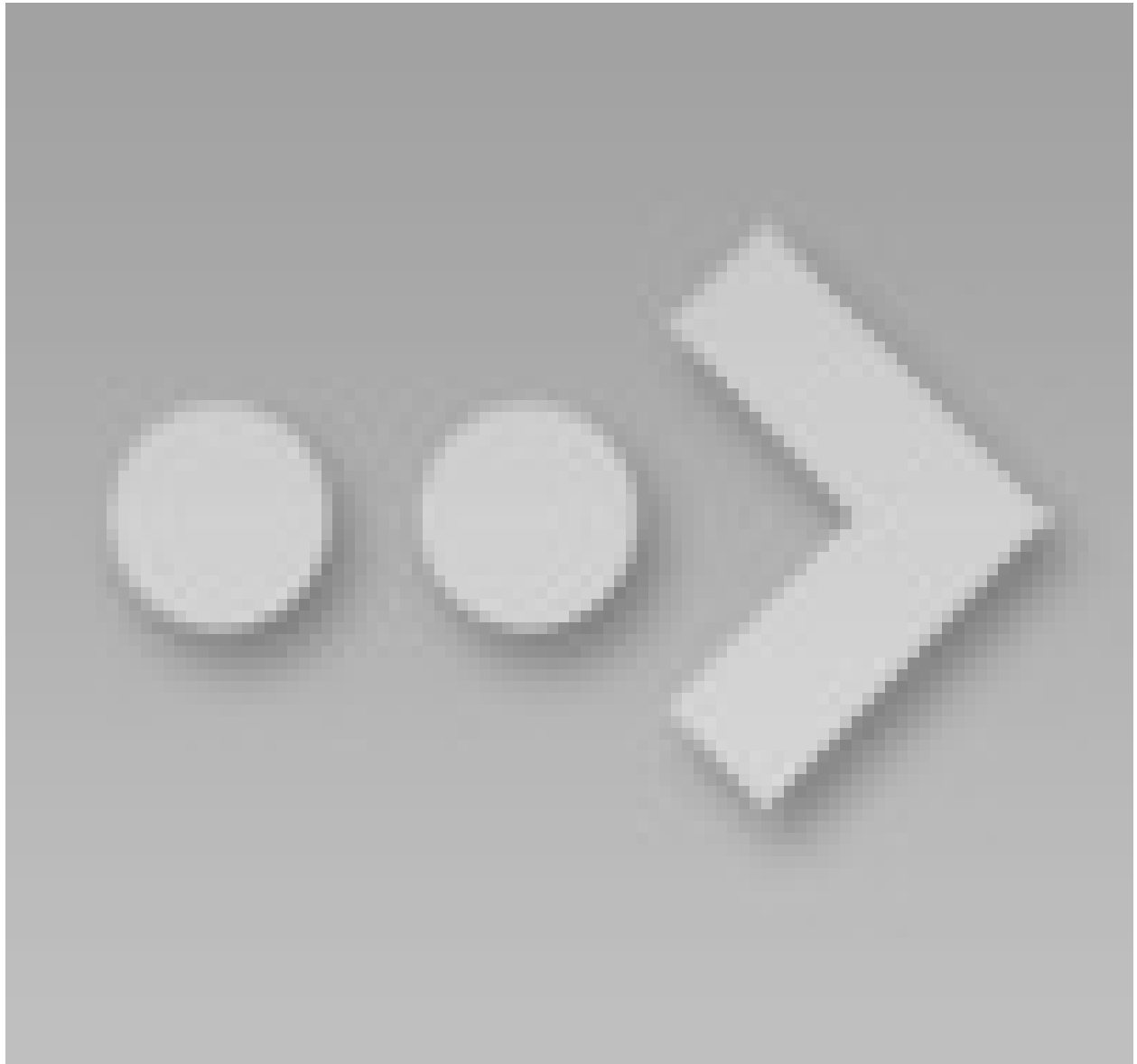


Chapter 57 – How to Make a Dual-Axis Map

In the last two chapters, you've read how to make a symbol map with Mapbox maps and how to make a filled map in Tableau. This chapter provides a step-by-step tutorial on how to combine the two into a single, dual-axis map. The final

view in this example will display sales by city as a symbol map and US States colored by region as a filled map:

[Insert Image 57-1]

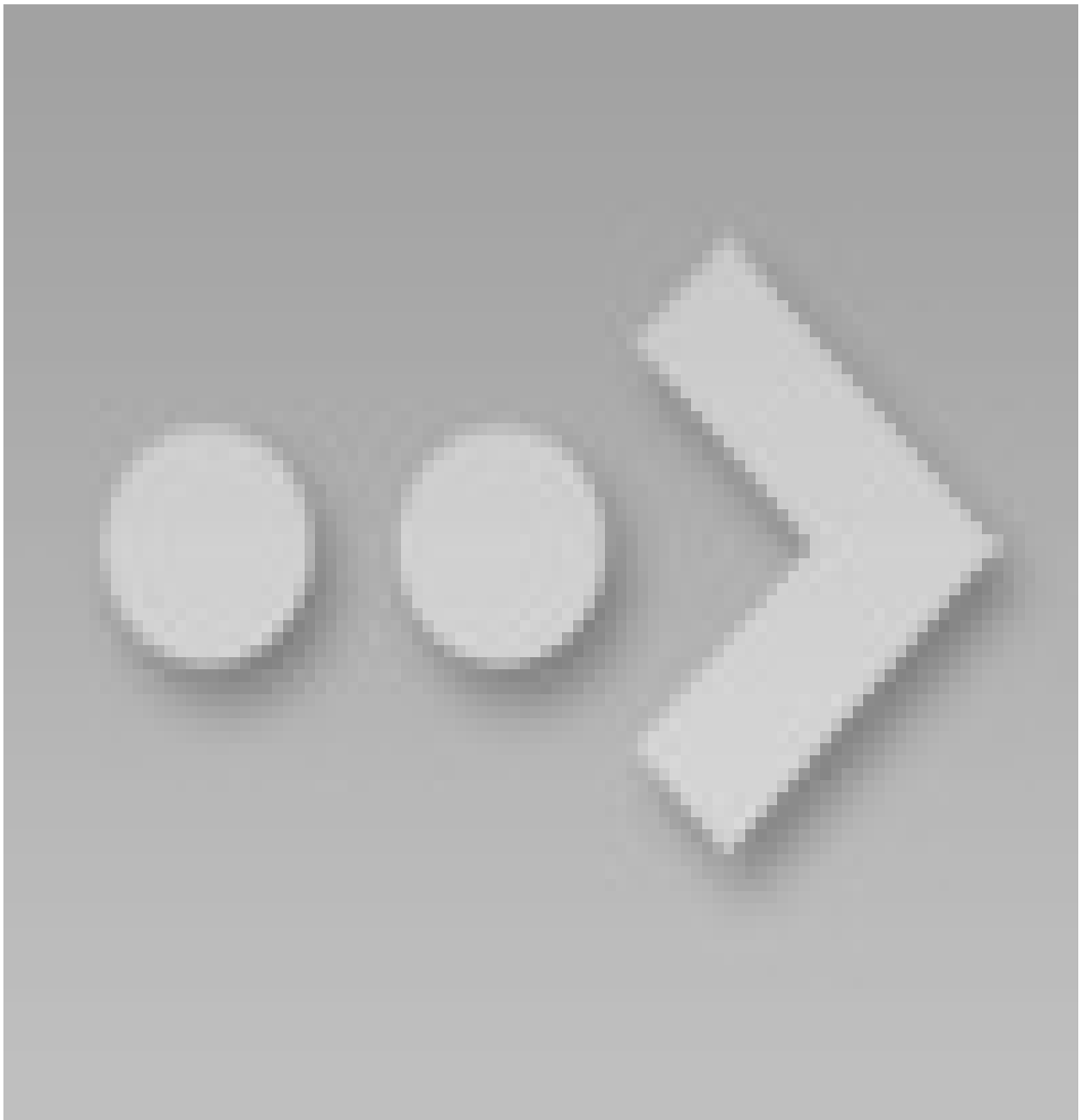


This type of map is useful any time you want to combine two maps into one to help provide additional context that would not be available when showing only one map or the other. One of the best real-life use cases I have seen for a dual-

axis map in Tableau was when an analytics partner wanted to see how the sales of their product by territory compared to the brick and mortar locations of a competitor.

To create a dual-axis map in Tableau, start by creating one of the two maps that you want to combine. I will start this example by creating a symbol map for sales by city.

[Insert Image 57-2]



Once the first map is created, you can create a duplicate of that map in one of two ways:

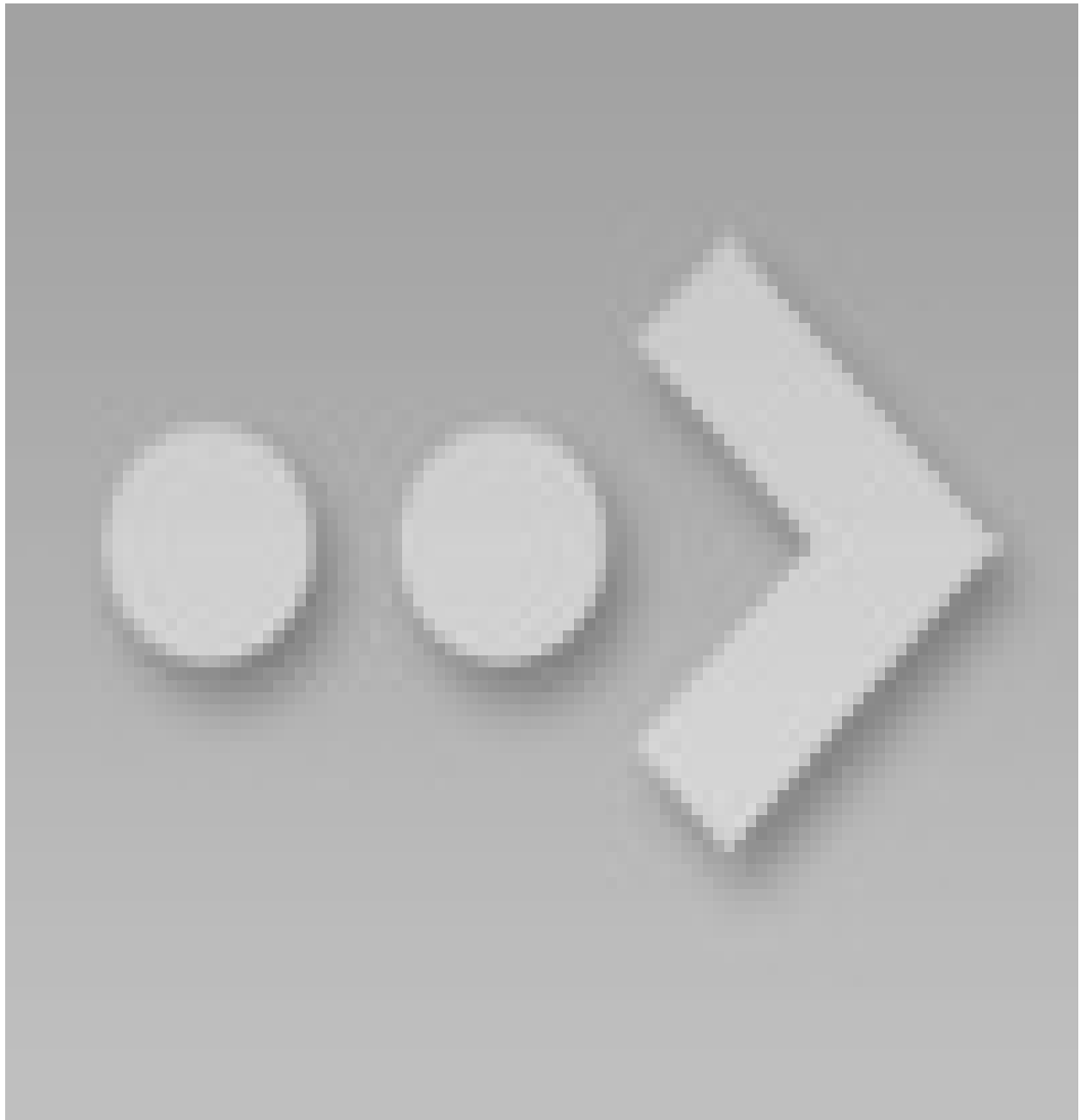
1. While holding the Control key, left click and drag the Longitude field on the Columns Shelf or the Latitude field on the Rows Shelf right next to itself. This is a shortcut for duplicating fields that works with any blue or green pill in Tableau.
2. Drag Longitude from the Measures Shelf to the second position on the Columns Shelf or Latitude from the Measures Shelf to the second position on the Rows Shelf.

Whichever method you select, the key is that by placing a second pill on the Columns or Rows shelf, a second Marks Shelf is generated. Now that there are two separate Marks Shelves, one for each map, the maps can be edited independently. That means a symbol map can exist on the first Marks Shelf, and the second Marks Shelf can be changed to a filled map. In this example, I will leave the first map as is, but change the second map to be a filled map by region. To format the map for my example, follow these steps:

- Navigate to the second Marks Shelf
- Remove the Sales Measure
- Remove the City Dimension
- Add the Region Dimension from the Dimensions Shelf to the Color Marks Card

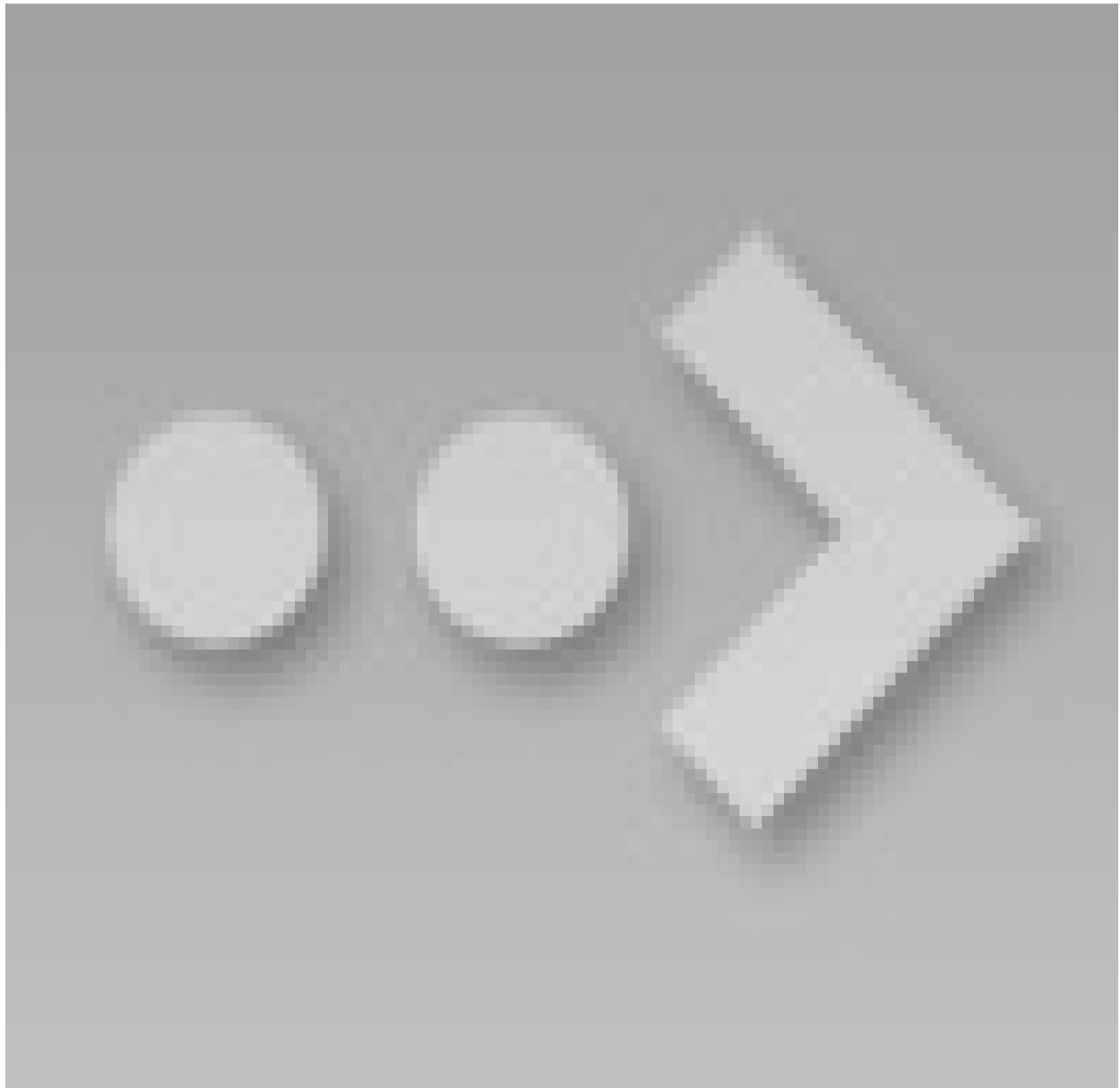
I've now got two distinct maps in my view.

[Insert Image 57-3]



To combine the maps, click on the second geographic measure on the Rows or Columns Shelf (depending on whether you duplicated Longitude or Latitude) and click “Dual-Axis”. If you want to change the float order of the maps, drag the second geographic measure on the Rows or Columns Shelf to the first position. My final map looks like this:

[Insert Image 57-4]



Chapter 58 – How to Map a Sequential Path

As with most features in Tableau, maps are flexible enough to meet many different analysis needs. One of the most popular uses of maps in Tableau that I have seen involves evaluating traffic through a certain travel hub, such as an airport or train station. These powerful visualizations are created using hub and spoke paths, with the airport or train station acting as the hub, and the paths to the destinations acting as the spokes. See the [Tableau Path Mapping](#) article on

Knowledge Base if you are interested in creating this type of map.

This is a great application of path maps in Tableau, but I recently had a need to display multiple destinations in sequential order instead of from a single origin to multiple destinations. My wife and I make it a priority to travel as much as we can, so I wanted to create a visualization to document our travels between stops. This tutorial walks you through how to create sequential paths on a map in Tableau.

How to map paths in Tableau

Before we get started, let's take a look at the visualization that inspired this chapter:

[Need an image from the viz at:

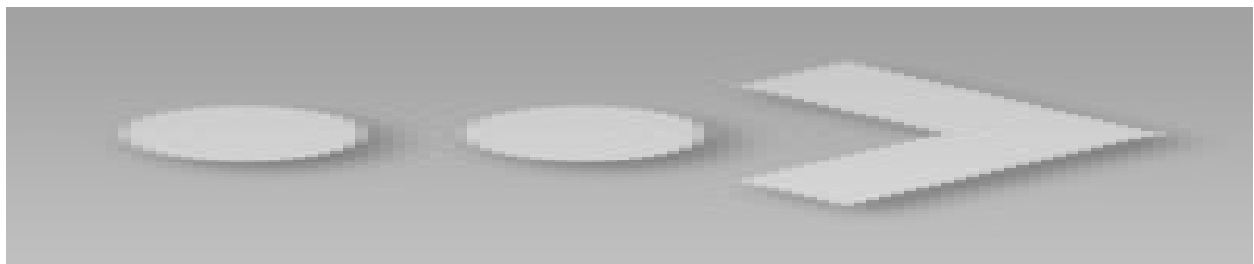
<https://public.tableau.com/views/WereNotInKansasAnymore/WereNotInKansasAnymore?d=8&v=1&f=t&rs=1>

Notice that within each trip, a path is displayed not only from our hometown of Kansas City to each destination, but from each stop in between. Here are the steps required to get this effect.

Step 1 – Prepare Your Data

Create your dataset so that each stop has a latitude, longitude, and path order. By looking up the latitude and longitude sets, you will have more flexibility when you create maps than you would if you use the automatically generated coordinates that Tableau applies to geographic fields. The path order is what tells Tableau how to connect the dots. Here's how my underlying data looks for one of our trips ("Stop" = Path Order):

[Insert Image 58-1]



Note: When you open this data for the first time in Tableau, your Stop field will be classified as a measure because it is quantitative. Make sure you change this field to a dimension by either dragging and dropping it from the Measures Shelf to the Dimensions Shelf, or by right clicking on the measure and choosing “Convert to Dimension”.

Step 2 – Start Your Map

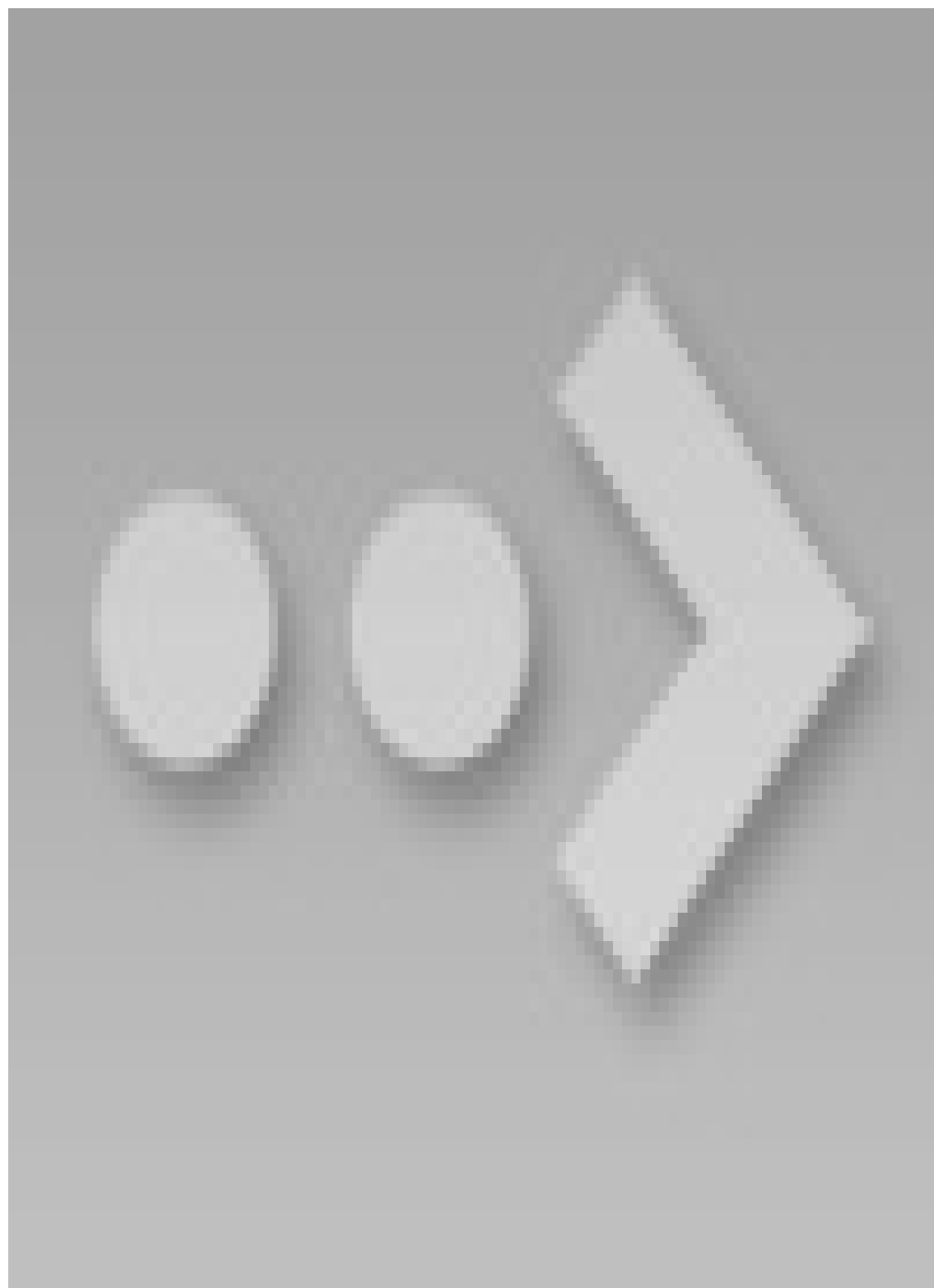
Using the dataset from step 1, begin your map in Tableau by placing your longitude field on the columns shelf and your latitude field on the rows shelf. It is important in this step to use the coordinates that you added to your dataset instead of the generated latitude and longitude in Tableau. You can always tell the difference because generated fields in Tableau are *italicized*.

Step 3 – Change the Map Type

At this point, you should see a map with a single point. That single point represents the average for all of the coordinates in your data. Change the mark type from Automatic, which is set to circles by default, to line. This change will reveal a new marks card called “Path”. The final step is to add the “Stop” (or path order) *dimension* to the Path Marks Card. This should connect the dots in the proper order, creating a sequential path.

Using the sample dataset from above, your final product should look like this:

[Insert Image 58-2]



Remember, many Tableau Public authors allow you to download their workbooks by clicking in the bottom right corner of an embedded Tableau Public viz. This is a great way to reverse engineer techniques that you want to incorporate into your own work.

Chapter 59 – How to Map Anything in Tableau

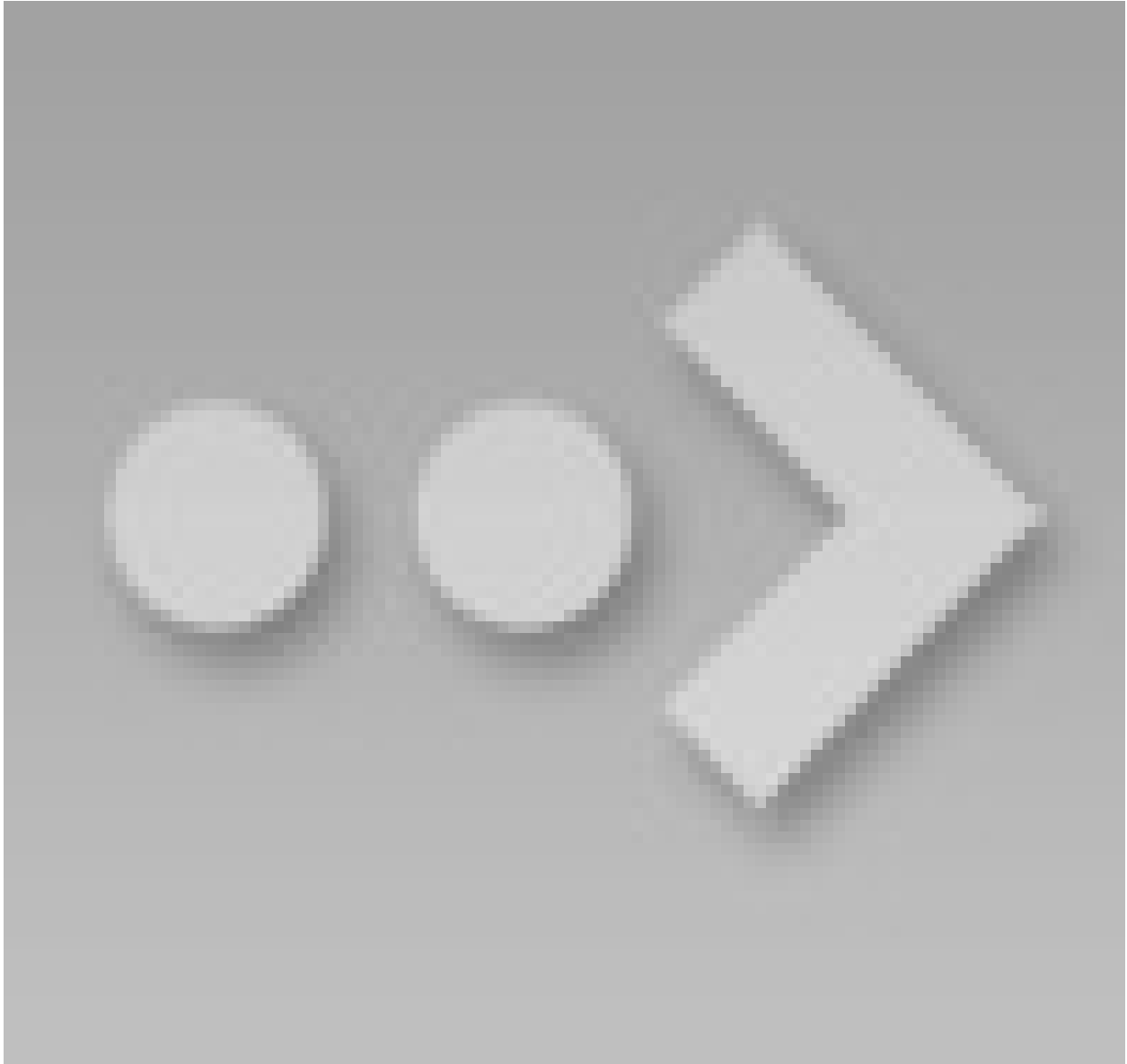
The out of the box Tableau symbol and filled maps are some of the most powerful visualization types available in the software. As illustrated in the Eliminate Chart Junk (but Not Graphics) tip, maps provide a means of decoding hundreds of latitude / longitude pairs almost instantly – all with a single image. Not only that, Tableau does a lot of the heavy lifting for us by providing the geographic coordinates of locations all over the world – this way we get to enjoy making maps instead of looking up longitudes and latitudes. Maps became even better in Tableau 9 with lasso and radial selections.

But wait – there's more! You can create custom maps in Tableau for just about anything. This tutorial will walk you through how to map *any* background image in Tableau.

Maps in Tableau can be thought of as scatter plots with a background image. Tableau will plot a point at each combination of the latitude on the Y-axis and longitude on the X-axis. To help illustrate the scatter plot concept – take a look at the following two images:

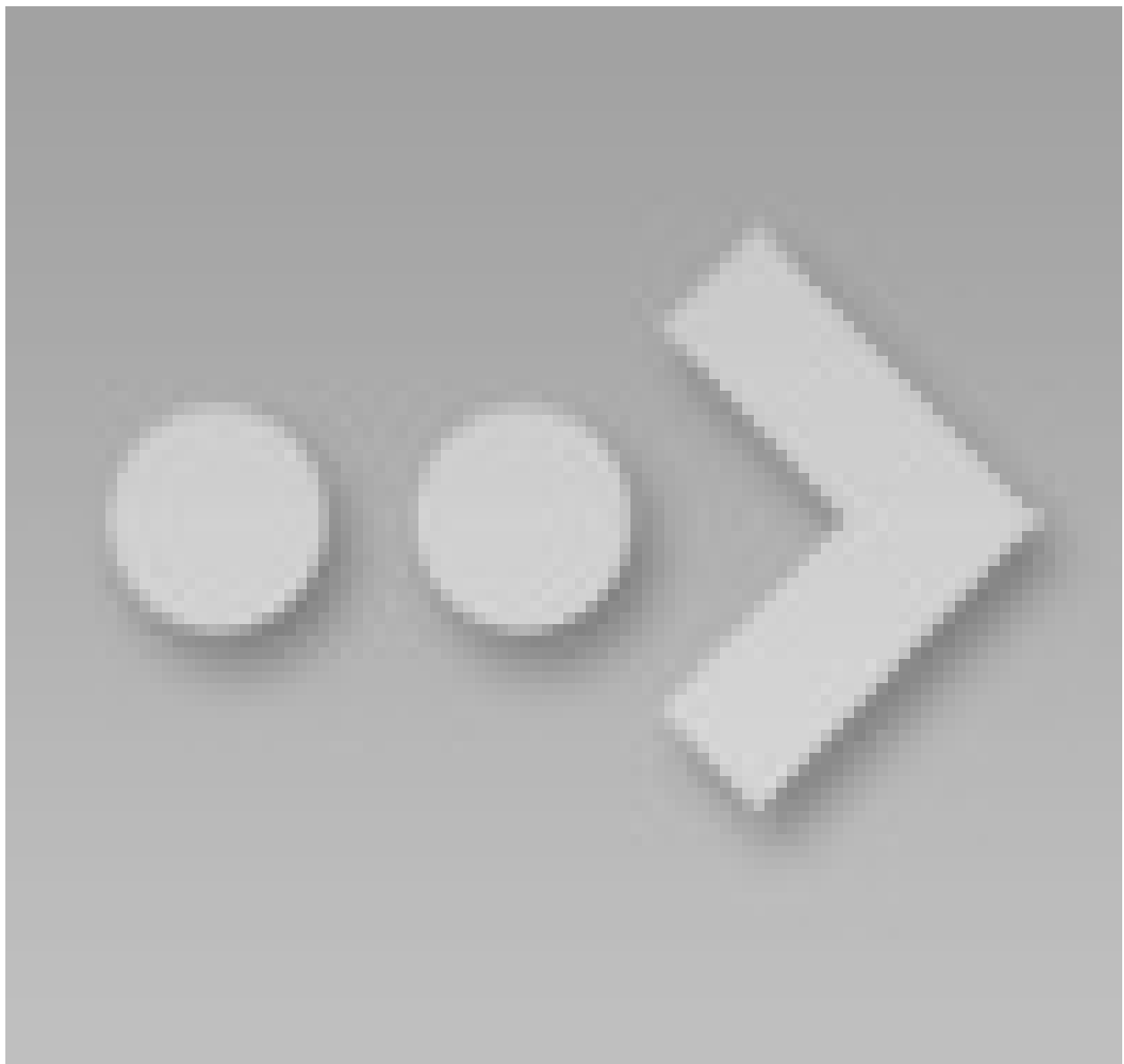
Latitude and longitude plotted with no background image

[Insert Image 59-1]



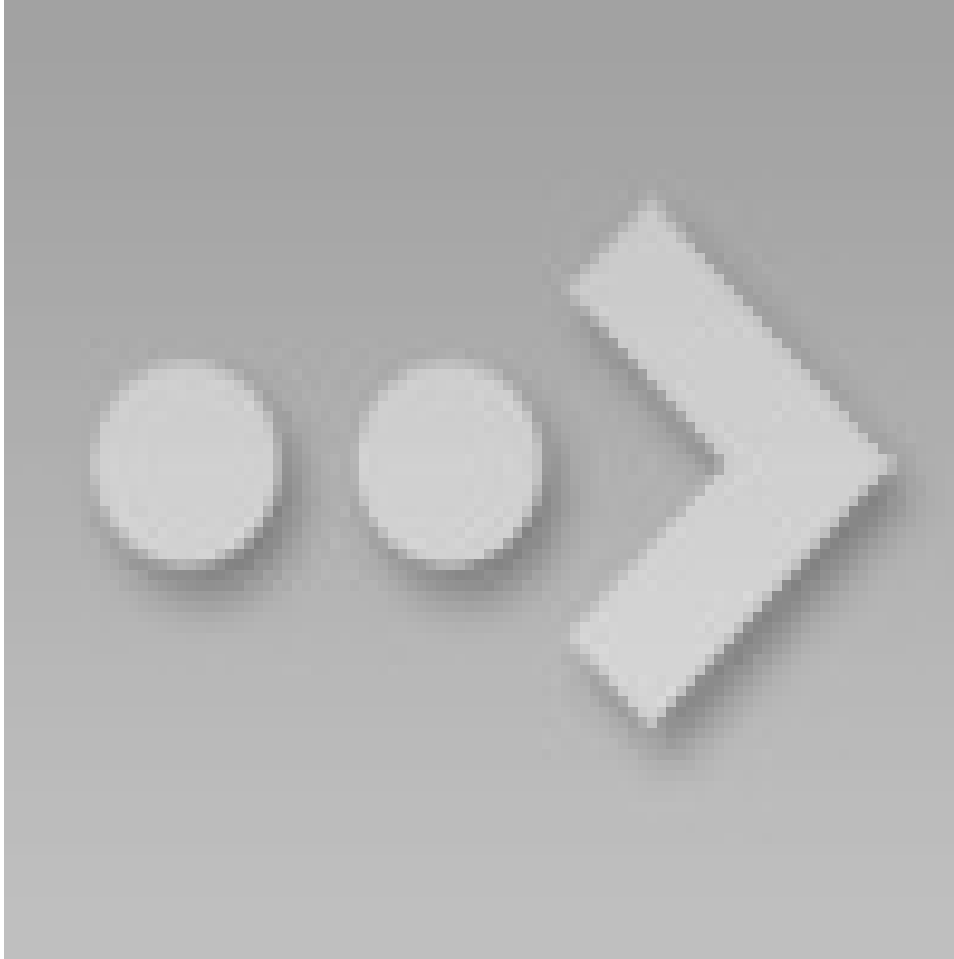
Same data points plotted with a background image of the United States

[Insert Image 59-2]



Due to the way maps function in Tableau, as long as you know the coordinates of a point on the vertical Y-axis and horizontal X-axis, you can map anything you would like! This provides some unique possibilities. So let's build one of these. For this tutorial, I will use this image of a baseball diamond.

[Insert Image 59-3]



Let's imagine that we want to visualize different baseball statistics by player position. To create this visualization, we first need to plot the nine positions on the field.

Building custom background maps in Tableau

Step 1 – Locate image and image dimensions

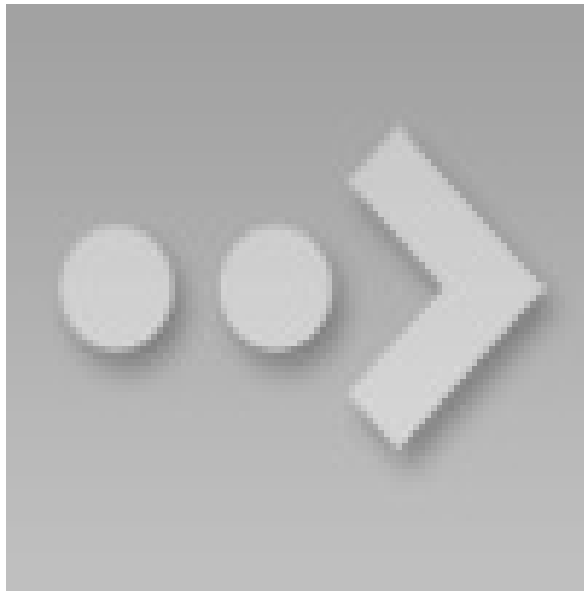
First we will need an image. If you want to follow along, download an image of a baseball diamond. For best results, pre-size the image in the desired dimensions and take note of the height and width of the image in pixels. The baseball diamond image above is 500 pixels wide by 500 pixels high.

Step 2 – Create dataset with fields for X and Y coordinates

The next step is to create a dataset that has at least three fields: 1. each point you

want to plot (in our case it's player position) 2. X Position 3. Y Position. For the first row, just put an arbitrary name in the cell for the point you want to plot, such as "Lookup". The X value will be the width of the image, and the Y value will be the height of the image. At this point, my dataset looks like this (the player position coordinates still need to be looked up):

[Insert Image 59-4]



Step 3 – Add the background image

Start a new workbook in Tableau and connect to the data from step 2. Navigate to Map > Background Images, and click on the data source. This will open a new dialog box where you can choose 'Add Image' in the bottom left corner. This is where you can choose which image will act as the background. If you're following along with the baseball example, navigate to the image that you downloaded from above.

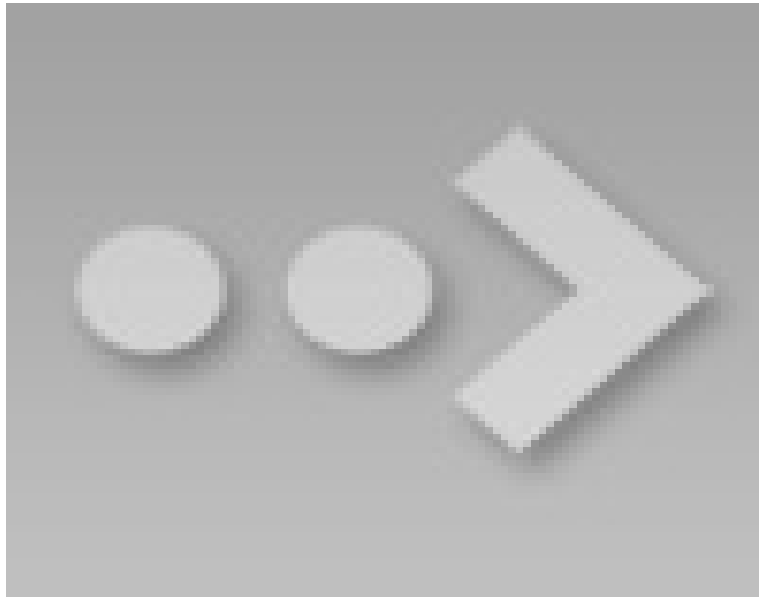
All that is left in this step is to put the maximum value for the X and Y coordinates. If our image is 500 wide by 500 high, we would make the following entries:

X Position: 0 Left, 500 Right

Y Position: 0 Bottom, 500 Top

Also be sure to change the 'Y Field' from 'X' to 'Y'. At this point, your screen should look like this:

[Insert Image 59-5]



Click OK to apply your changes.

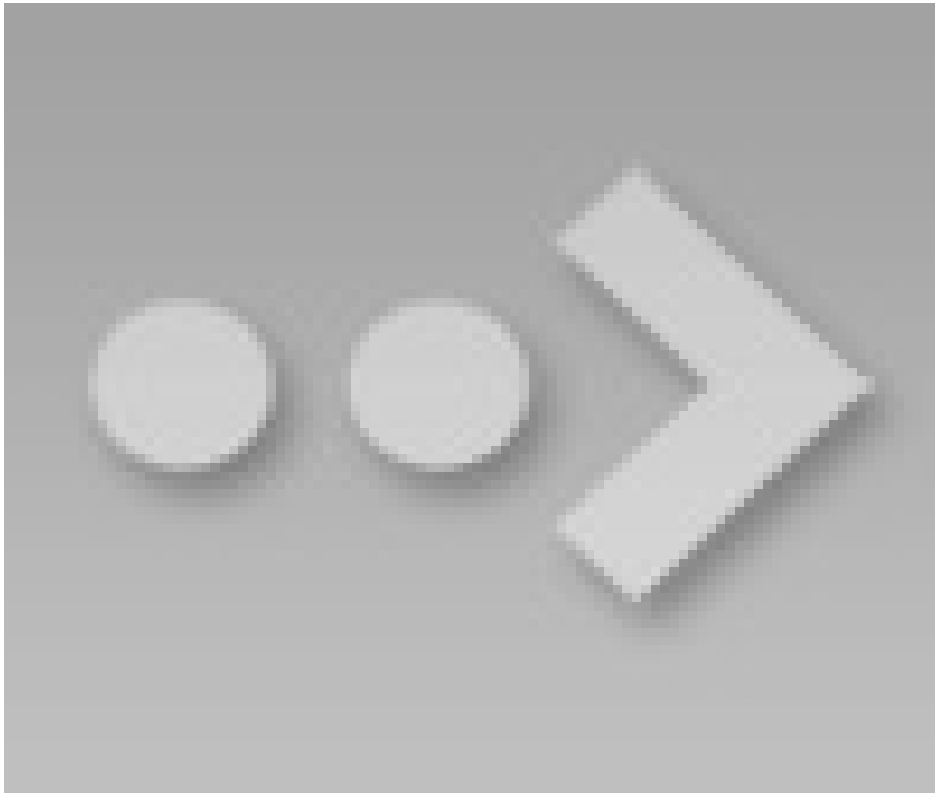
Step 4 – Look up X and Y position for each coordinate

Start your new custom map by placing the X measure on Columns Shelf and the Y measure on the Rows Shelf. Ensure both axes are fixed at the maximum X and Y values; which are 500 for both in our case. If you need to change an axis, simply right click on the axis, choose “Edit Axis”, and fix the range at 0 to 500.

In order to look up the X and Y coordinates for each position player (or whatever you are plotting), right click on the view, hover over ‘Annotate’, and click ‘Point’. If the X and Y measures are the only fields on your view, the annotation will show you the X and Y values by default. This is what we want, but you may want to make the font larger so that it’s easier to read.

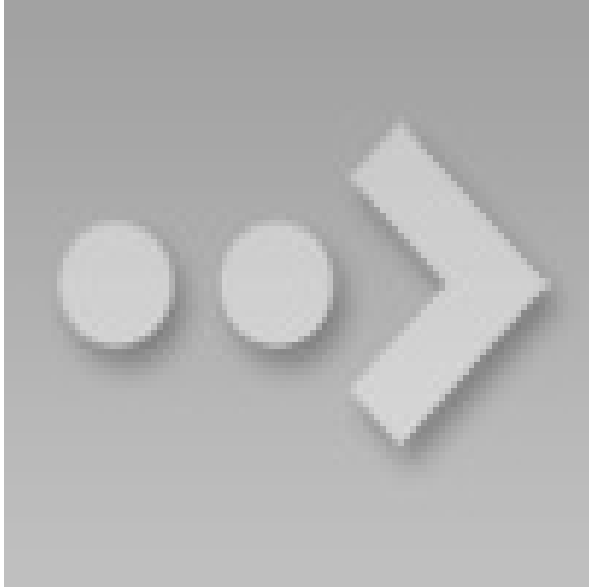
Now drag the end point of the annotation to each location you want to plot, and see how the X and Y values change. For best results, format the annotation so the line end is a circle; this way you can precisely drag the circle to the exact location you want plotted. Here is one example showing the coordinates for the Catcher:

[Insert Image 59-6]



Drag the annotation to each point you want to plot, and record the coordinates in your dataset. Note that you don't need to record the decimals. Here is a look at my data with the X and Y coordinates for each player position:

[Insert Image 59-7]

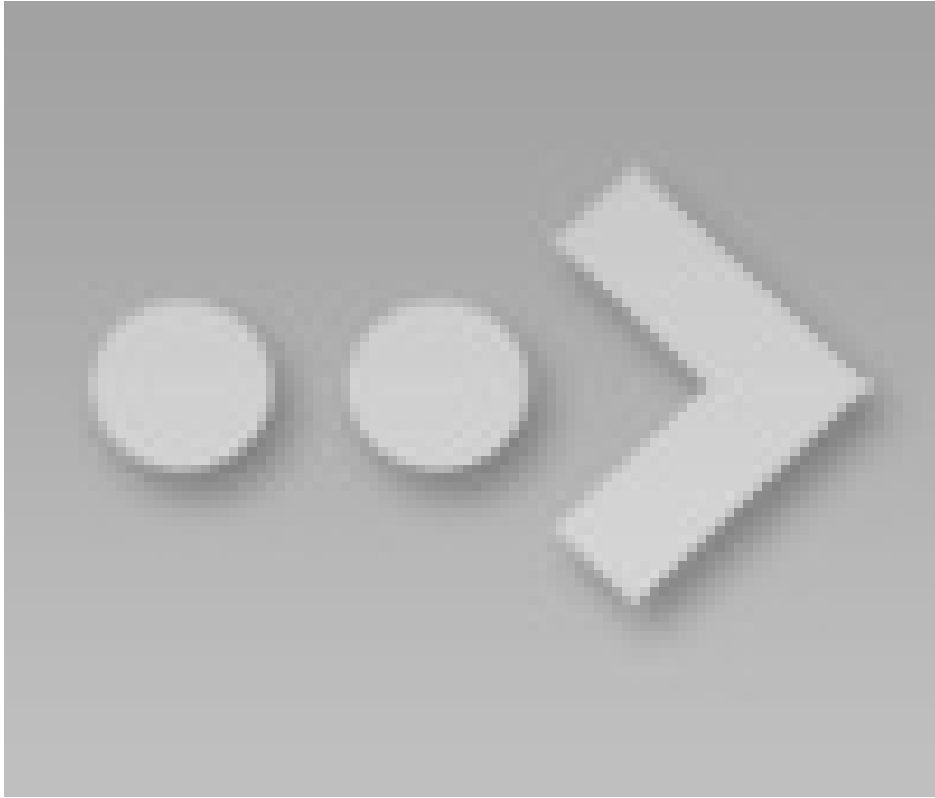


Step 5 – Refresh your map

After all coordinates have been recorded in your dataset, save your data and refresh the data source in Tableau.

Drag the field you are plotting, such as position in the baseball example, to the Detail Marks Card. You should now see a circle at each position, like this:

[Insert Image 59-8]



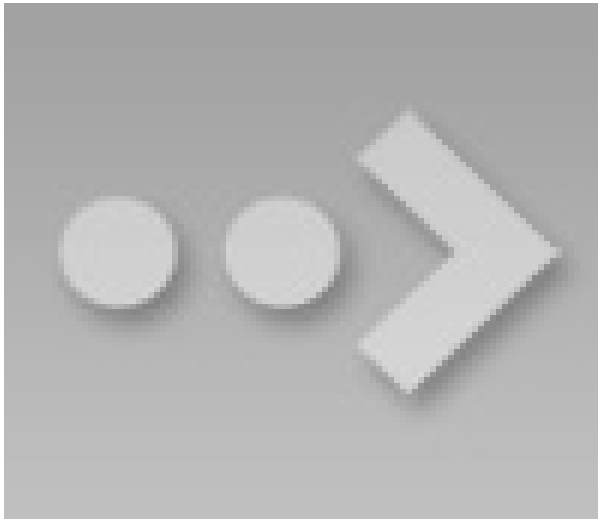
Step 6 – Finalize your map and/or add metrics

There are just a few items you can do to finalize your custom Tableau map:

- Remove the annotation by right-clicking on it and selecting ‘Remove’
- Hide the axes by right-clicking on each one and deselecting ‘Show Header’
- Format your marks; my favorite is a filled circle with a custom color and white border
- Filter out the ‘Lookup’ X and Y coordinates. You will notice a mark in the top right corner of your new custom map. You can filter this out by either right-clicking on it and choosing ‘Exclude’, or for better long-term results, remove this row from your underlying data and refresh your view.
- Optional: You can now add measures to your underlying data for each position to visualize different metrics on your customized view (i.e. size the circles by fielding %).

The final custom Tableau map will look like this:

[Insert Image 59-9]



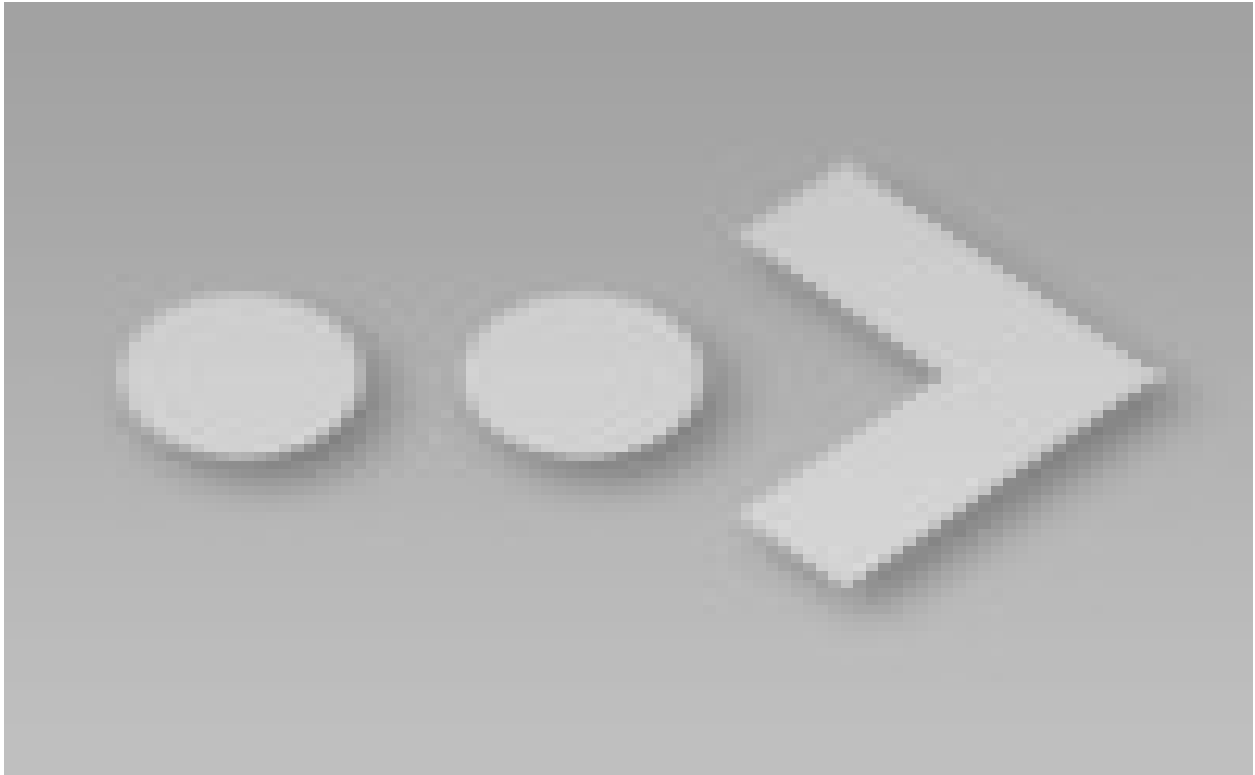
Chapter 60 – How to Make Custom Polygon Maps

In the last two tutorials about mapping, we have discussed pathing maps in *How to Map a Sequential Path* and custom symbol maps in *How to Map Anything in Tableau*. There is a third type of map in Tableau called a polygon map that allows you to map custom shapes. If you're not familiar with these types of maps, the best example to think of is a "filled map". Imagine a map of sales by U.S. state where each state is colored by their respective sales volumes. With these filled maps, Tableau is essentially looking up the latitude and longitude coordinates all the way around the border of each state, and plotting a custom polygon for each territory.

This tutorial will use one my most asked about visualizations, *The Cost of Attending the 2015 World Series*, to illustrate how you can create custom polygon maps with any shapes – including stadiums!

First, let's take a look at the screenshot of the original viz:

[Insert Image 60-1]



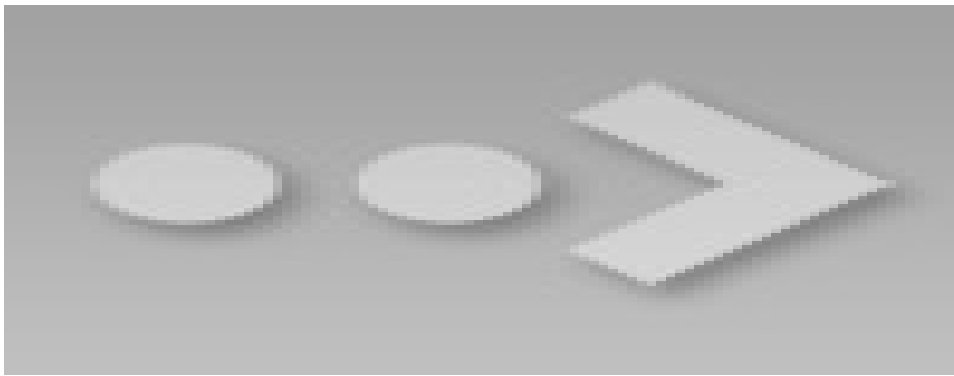
Polygon maps in Tableau are created by looking up the coordinates of the shape you want to draw, and then connecting the dots by drawing a path between them. I encourage you to read the first two chapters in this series mentioned in the opening paragraph if you are not familiar with these concepts. The main difference between a polygon map and a path map is that you “close the loop” around the dots by choosing a polygon mark type instead of a line mark type. Here are the steps:

How to make custom polygon maps in Tableau

1. Find an image of the stadium that you want to map. StubHub is a good source for these images because they already have the sections broken out, but you can use any image, including an aerial photo of the stadium. Being a visualization about the 2015 World Series, I used Kauffman Stadium in Kansas City and Citi Field in New York.

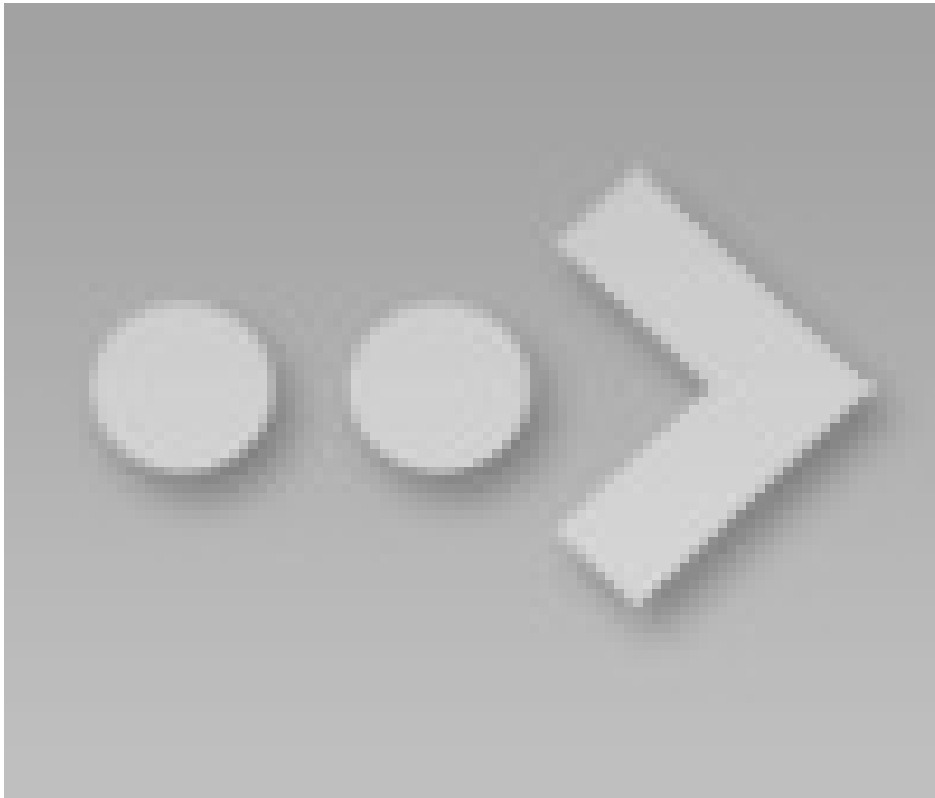
1. 2. Follow the steps in How to Map Anything in Tableau to set up your map in Tableau and record the coordinates for your shapes.
1. 3. Similar to [mapping a sequential path](#), each combination of coordinates should be given a “point order”. This is a field in your underlying data that tells Tableau what order the “dots” are connected. At this point, my underlying data for one section of Kauffman Stadium looks like this:

[Insert Image 60-2]



1. 4. Once you have the X and Y coordinates for each point of each shape you want to plot, we are ready to build the stadium map in Tableau. To start, put your X measure on the columns shelf and Y measure on the rows shelf; both with an aggregation of AVG.
1. 5. Change the mark type from Automatic to Polygon, and place your Point Order dimension on the Path Marks Card.
1. 6. Place the section dimension (and/or the dimension with the most granular level of stadium detail) to the Detail Marks Card. Note that my maps are actually plotted by zone and zone section (which was required when the same zone name existed in two places in the stadium) so I had to add both of these to the Detail Marks Card. If you are just building one stadium at the section level, you would place section on the Detail Marks Card. At this point, my map looks like this:

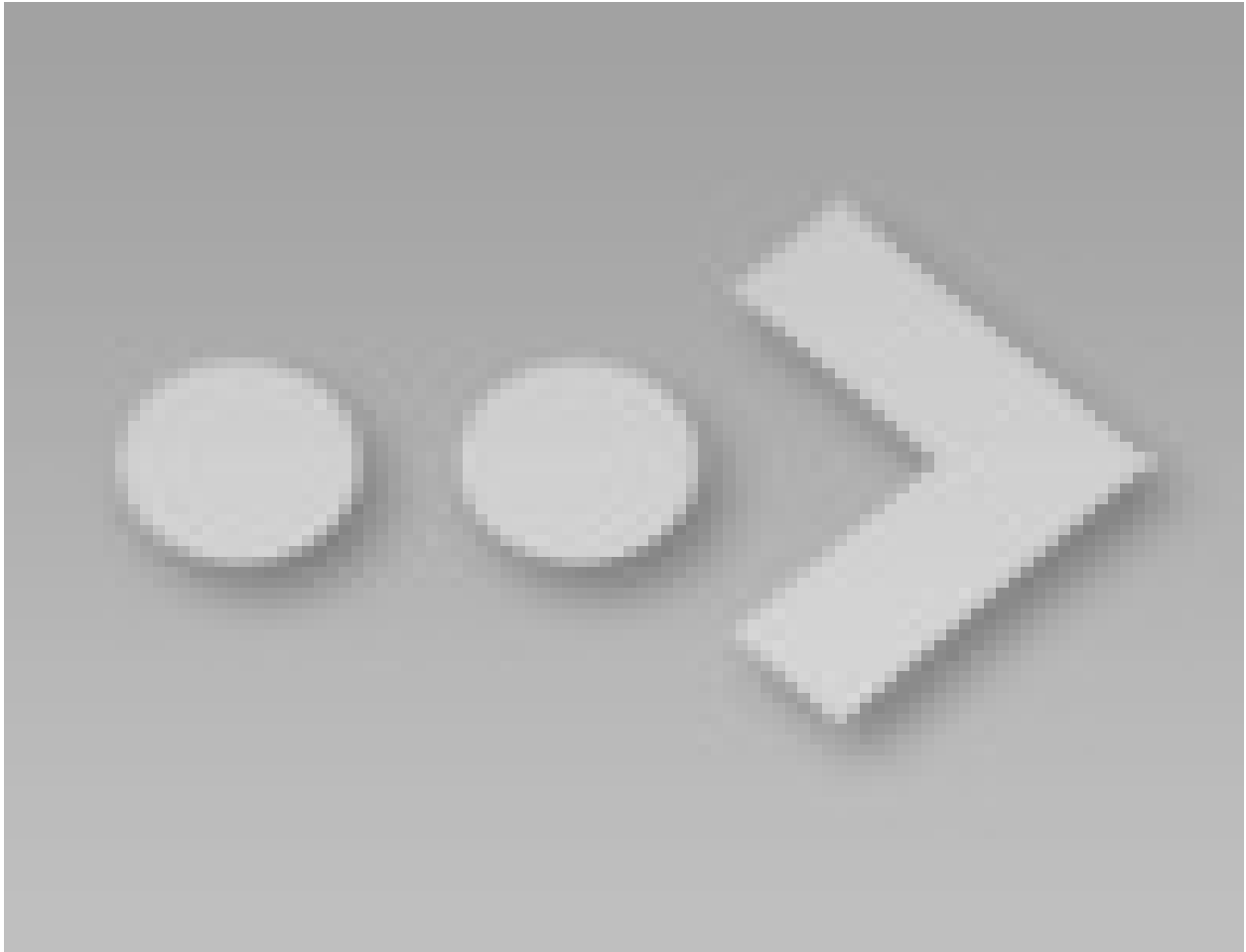
[Insert Image 60-3]



1. 7. This can now be made into a small multiples view by “slicing” the map coordinate measures by a dimension. In my case, I was looking at the maps for each game of the World Series, so I put Game on the columns shelf to create a column for each game with a stadium in each cell.

From here, you can hide the axes and encode each section by putting a measure on the Color Marks Card. Here’s what my final product looks like under the hood:

[Insert Image 60-4]



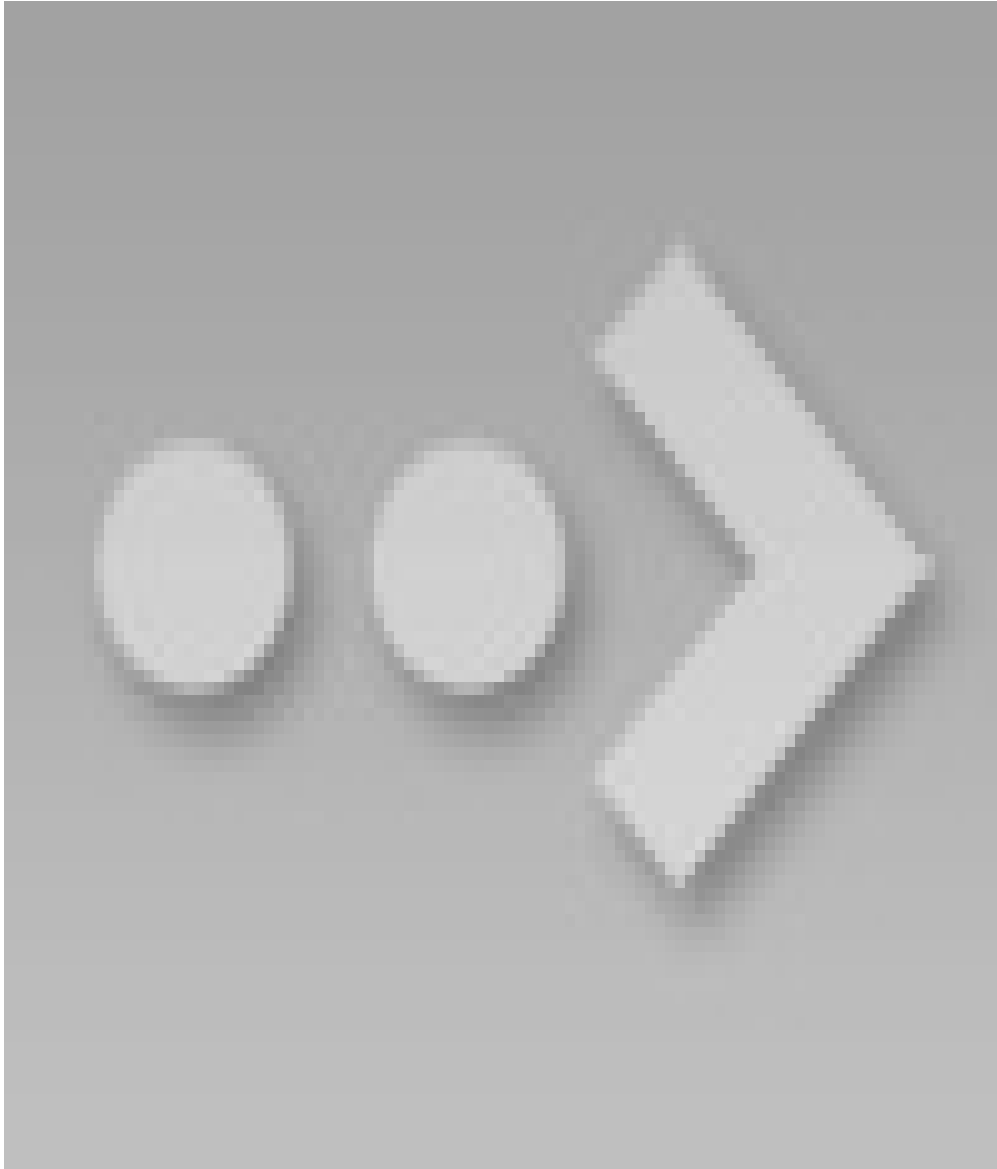
I admit that the map creation process requires painstaking attention to detail, but this step only has to be done once. The map coordinates can then be used over and over in future analyses and the results are great!

Chapter 61 – How to Make a Gantt Chart

Gantt charts are traditionally used for visualizing project schedules. They are effective for illustrating task durations and dependencies in context of the larger business operation. As with several visualizations that can be built with Tableau, the fact that ‘Gantt Bar’ is a mark type option unlocks some flexibility that allows you to use this chart type in many other ways. To start demonstrating the usefulness of the Gantt Bar mark type, we will build a Gantt chart with a traditional application of illustrating project schedules. The final product will

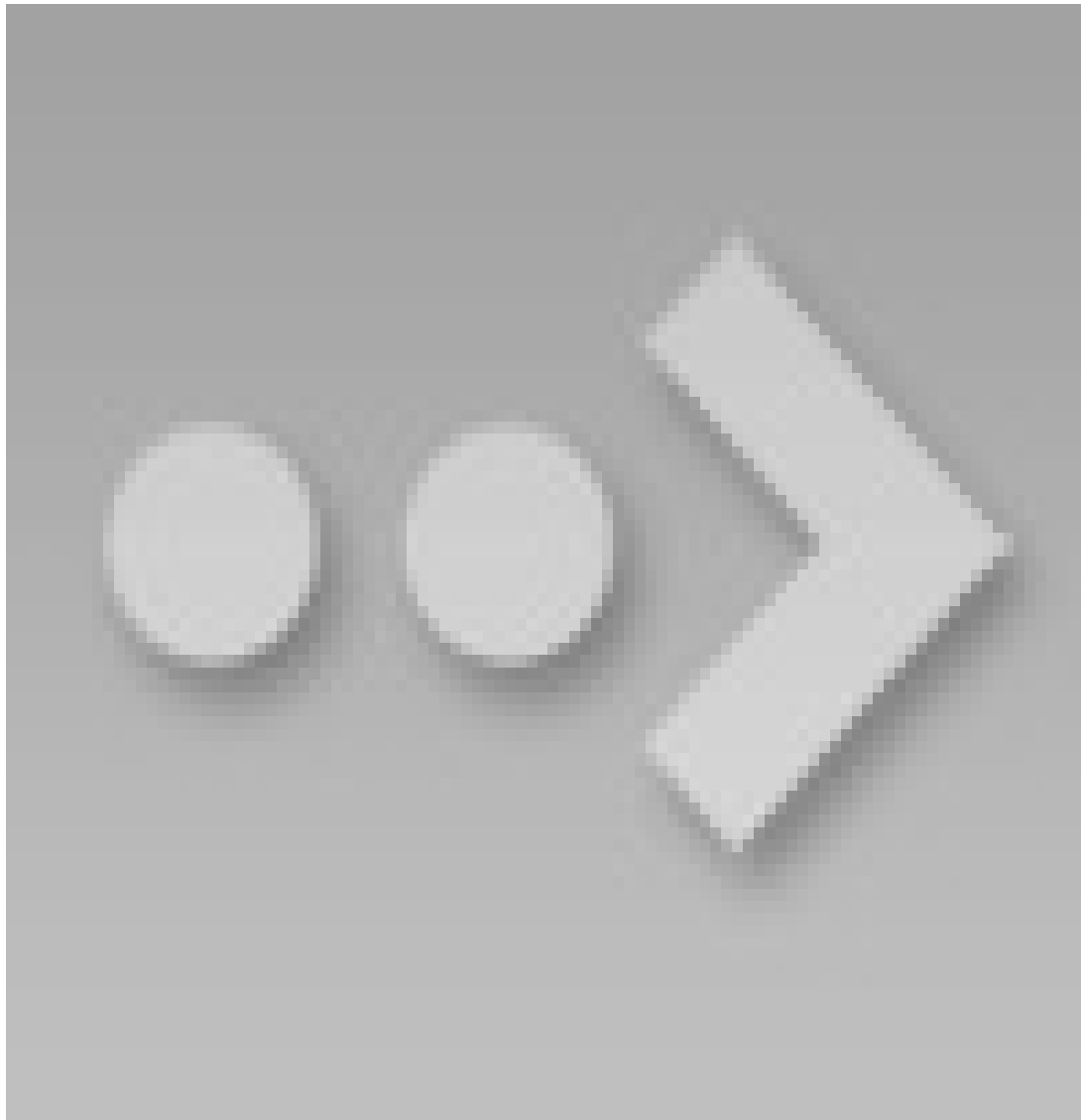
look like this:

[Insert Image 61-1]



This is one of the few examples that cannot be illustrated using the Sample – Superstore data source, so I will be using the following dataset:

[Insert Image 61-2]

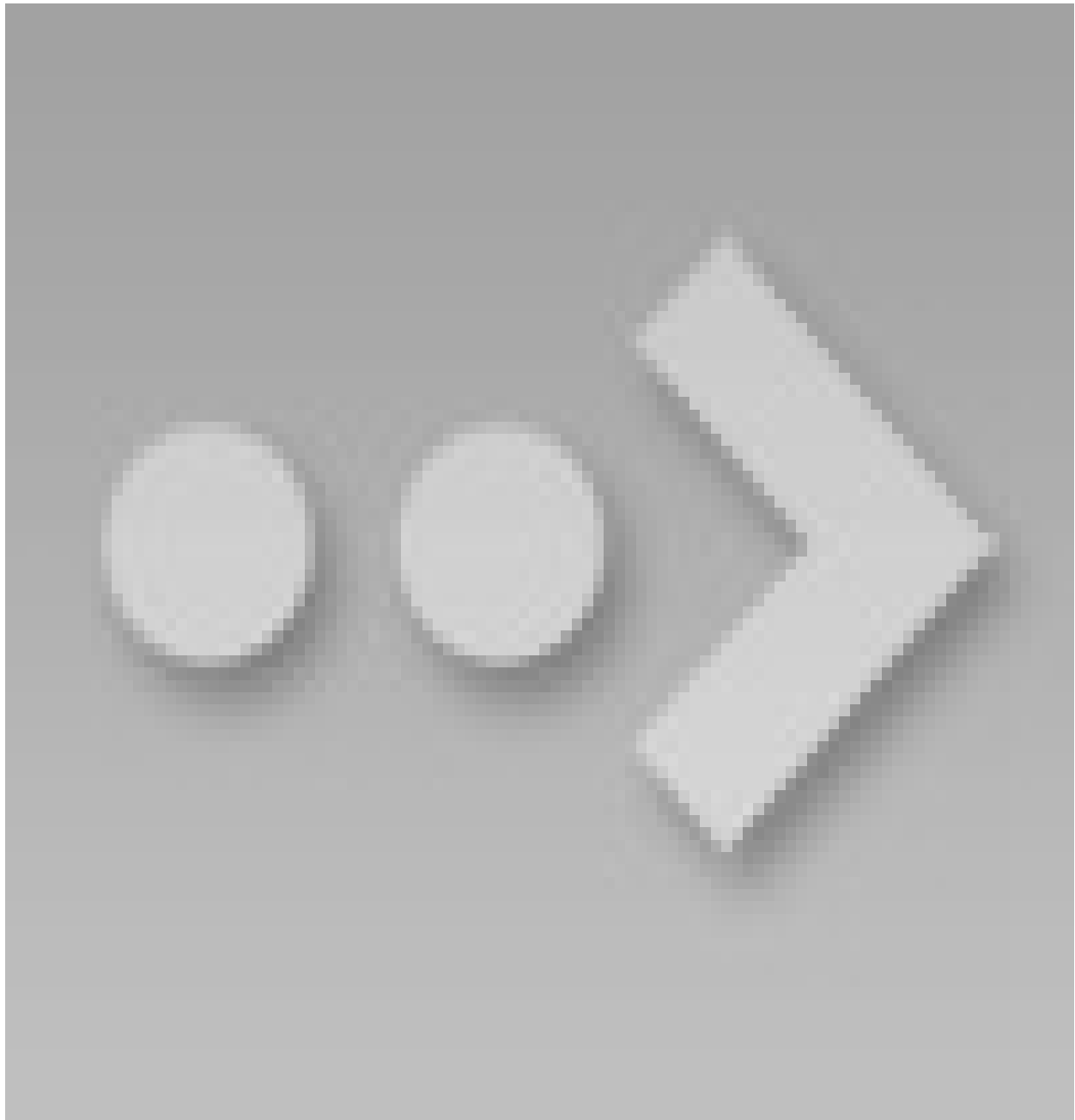


For the purposes of this illustration, let's pretend that we are a manager with three employees and we are plotting out their schedules across twelve projects for the year. We have created the schedules in Excel, but want to leverage data visualization to (1) get a better sense of how our projects and employees line up, (2) track progress to completion for each project, and (3) provide a visual schedule for our employees to use. All three of these features can be provided through a Gantt chart.

Gantt charts are created with one date, one or more dimensions, and zero to two measures. The date provides the axis; the dimensions provide the breakdowns we want to visualize and/or the encoding of the Gantt bars; the measures create the length of the Gantt bars and/or their encoding.

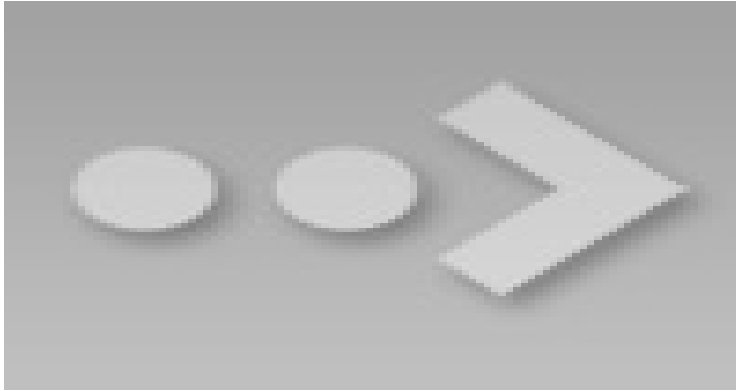
For this Gantt chart, I would like to have a continuous axis running left to right, so I will put the Date dimension (being used as a continuous field) on the Columns Shelf. Looking at the final product above, there are rows for each combination of Project and Person, which tells us those breakdowns reside on the Rows Shelf. Gantt chart is the default mark type in Tableau for this combination of dimensions and measures, so placing the fields on the view results in this foundation of a Gantt chart:

[Insert Image 61-3]



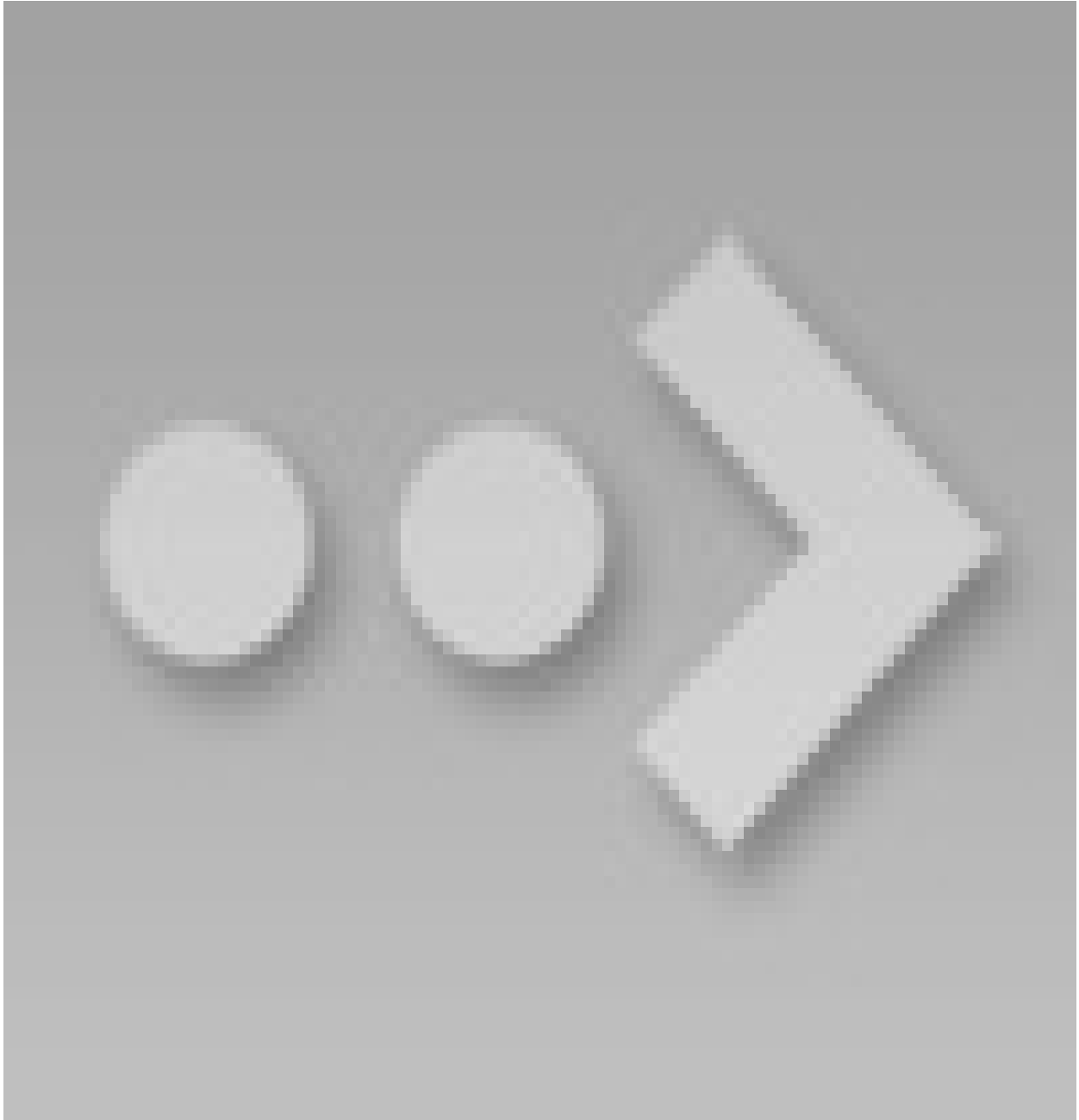
What Tableau has done is place a Gantt mark at the start date of each combination of Project and Person. To extend the Gantt bars to illustrate task duration, we need to size each mark by the number of days in each respective project / person combination. You may have a field for duration in the underlying dataset, but it can also be created in Tableau with a calculated field. In this case, duration simply equals [End Date] – [Start Date]

[Insert Image 61-4]



Placing the duration on the Size Marks Card will extend the Gantt marks so that project / person combinations with longer durations will be longer bars and project / person combinations with shorter durations will be shorter bars.

[Insert Image 61-5]



At this point, we have a usable Gantt chart, but there is still room to add a lot of value in Tableau through formatting, encoding, and reference lines. This step is flexible based on your own requirements, but as just one example, I have done the following:

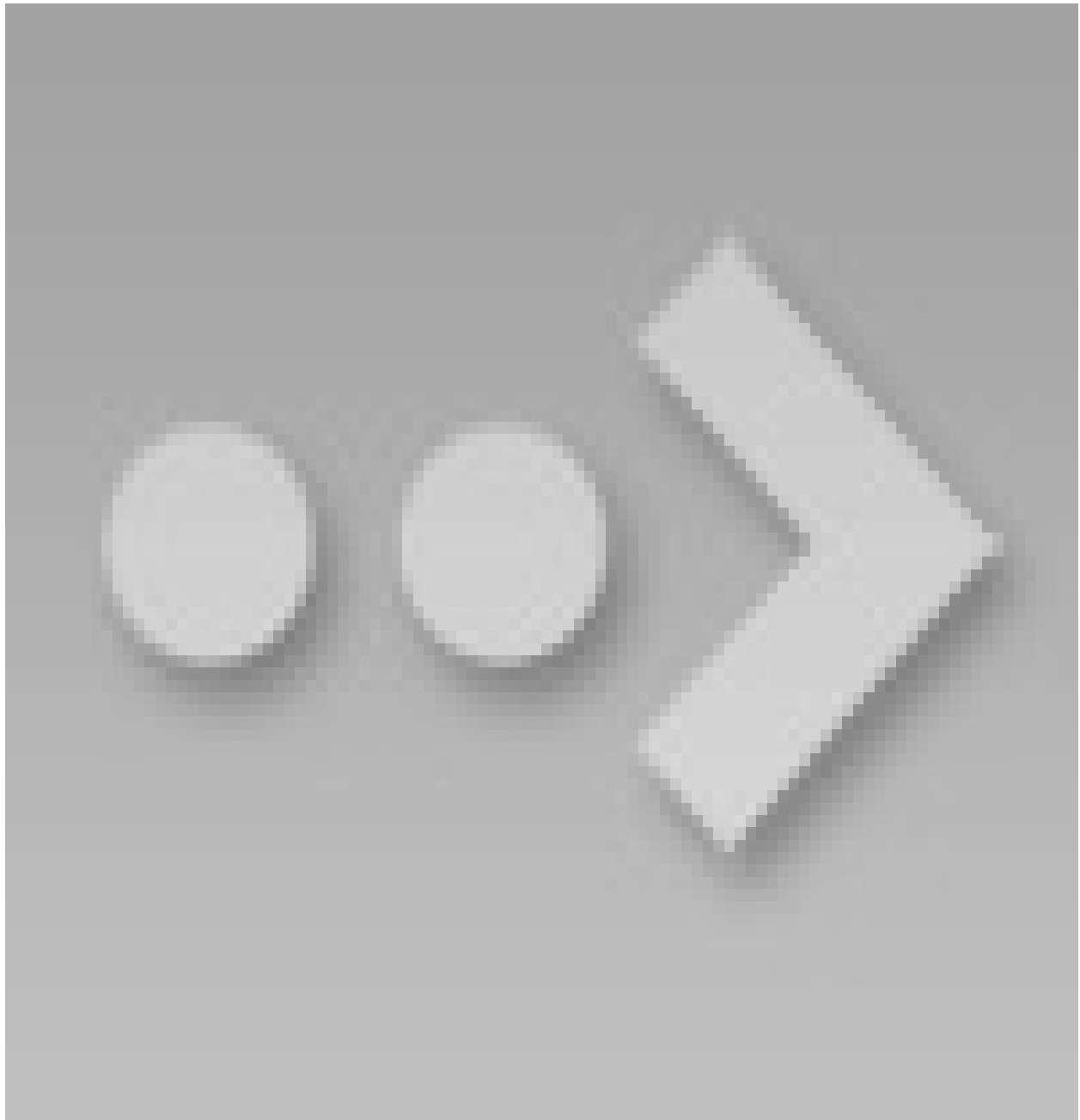
- Colored the bars by employee by adding the Person dimension to the Color Marks Card. One of my goals was to provide a visual schedule to my team and by doing this, they can quickly view their own schedule and even

highlight themselves in Tableau by clicking their name on the color legend.

- Added a reference line for “Today” (pretending it’s June 20, 2018) to show each project in context of today’s date.
- Added percent complete to the Label Marks Card to help determine if we are ahead or behind pace for each project. This number was hardcoded in my underlying dataset but in the real world it may also come from a time tracking system. Another good use of color would have been to color the Gantt bars by progress to completion.

Here is my final Gantt chart after making these changes and cleaning up the formatting:

[Insert Image 61-6]



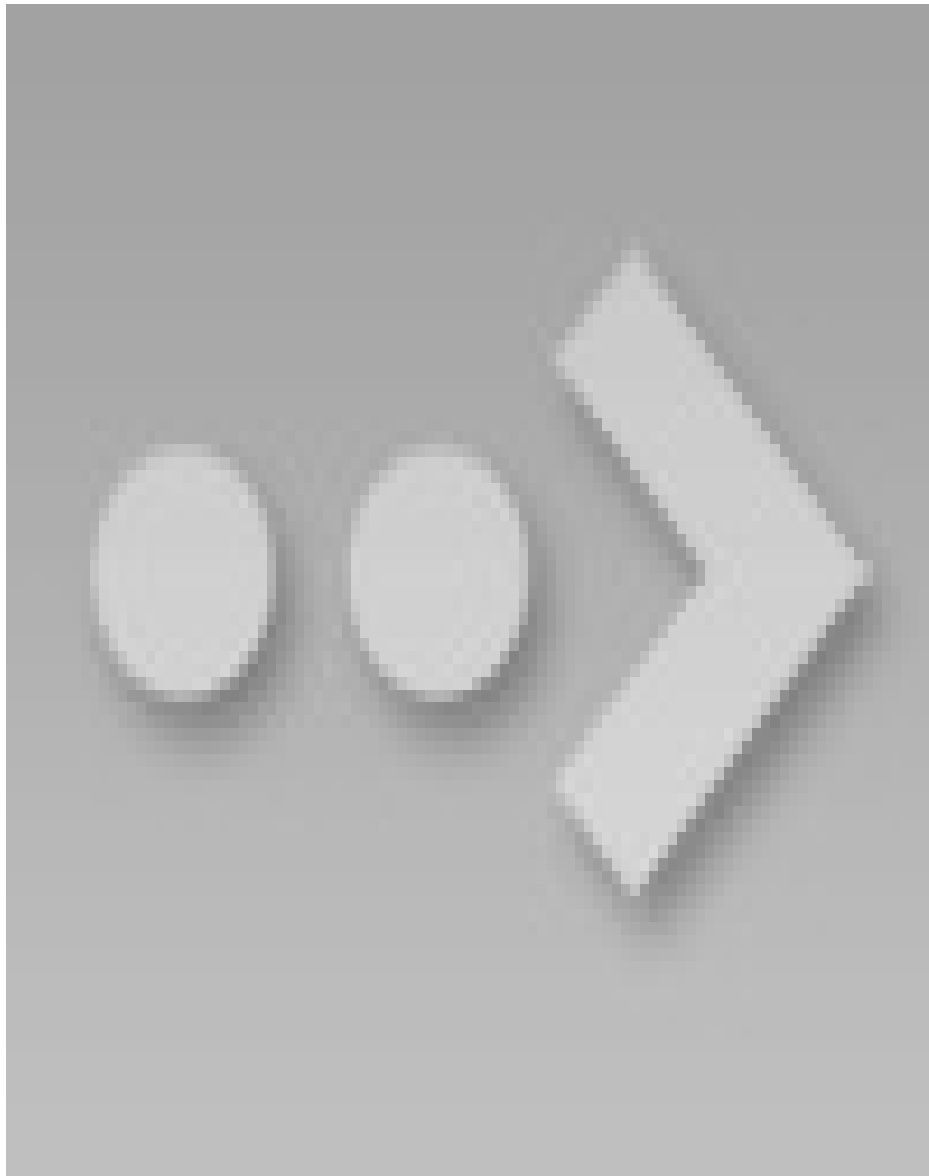
In the next chapter, I will illustrate one special use of the Gantt mark type: Waterfall Charts.

Chapter 62 – How to Make a Waterfall Chart

This chapter shares how to make a waterfall chart in Tableau – a visualization that helps understand how positive and negative values of dimension members

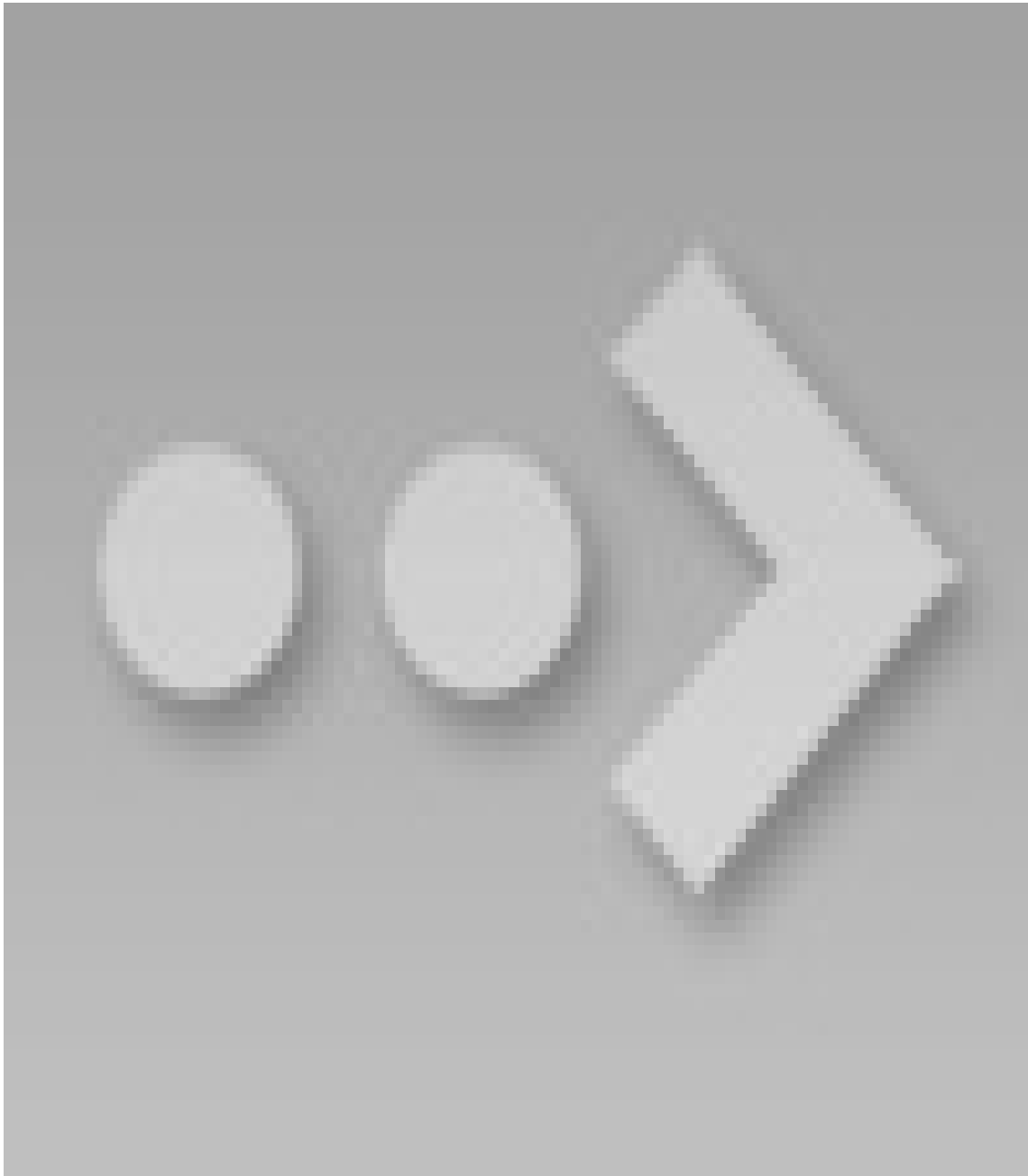
are contributing to a total. What makes waterfall charts different from a simple running total calculation is that they illustrate how each dimension member with a positive value adds to a running total and each dimension member with a negative value detracts from a running total. For this tutorial, we will build the following waterfall chart in Tableau which visualizes how each Sub-Category in the Sample – Superstore dataset is contributing to total profit:

[Insert Image 62-1]



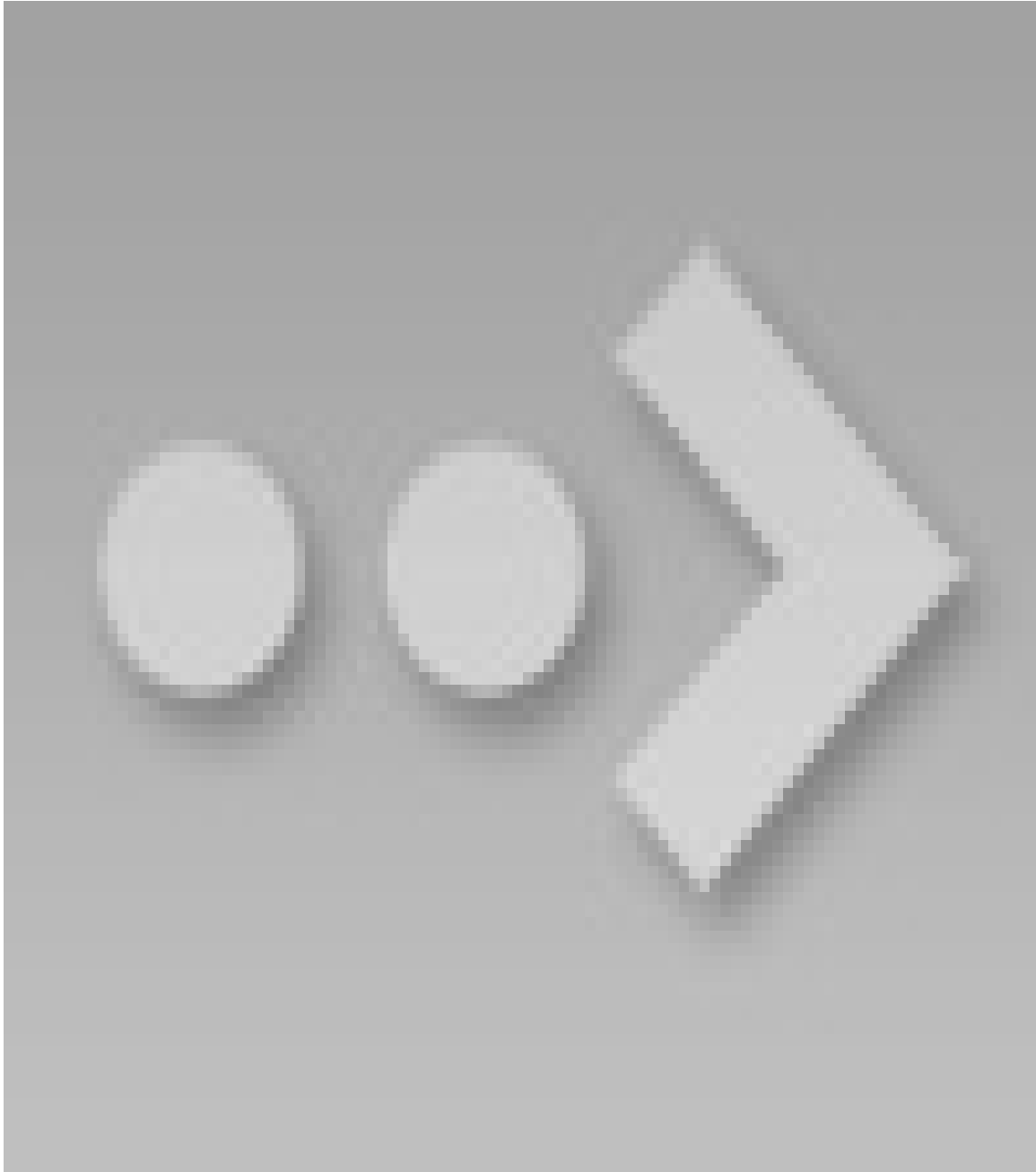
To start a waterfall chart in Tableau, create a vertical bar chart showing the measure, Profit, by the dimension, Sub-Category:

[Insert Image 62-2]



Next, add a table calculation to the Profit measure so that it calculates a 'Running total' on 'Table (Across)'. For a refresher on table calculations, see the chapter: An Introduction to Tableau Table Calculations. After adding the quick table calculation, the view looks like this:

[Insert Image 62-3]

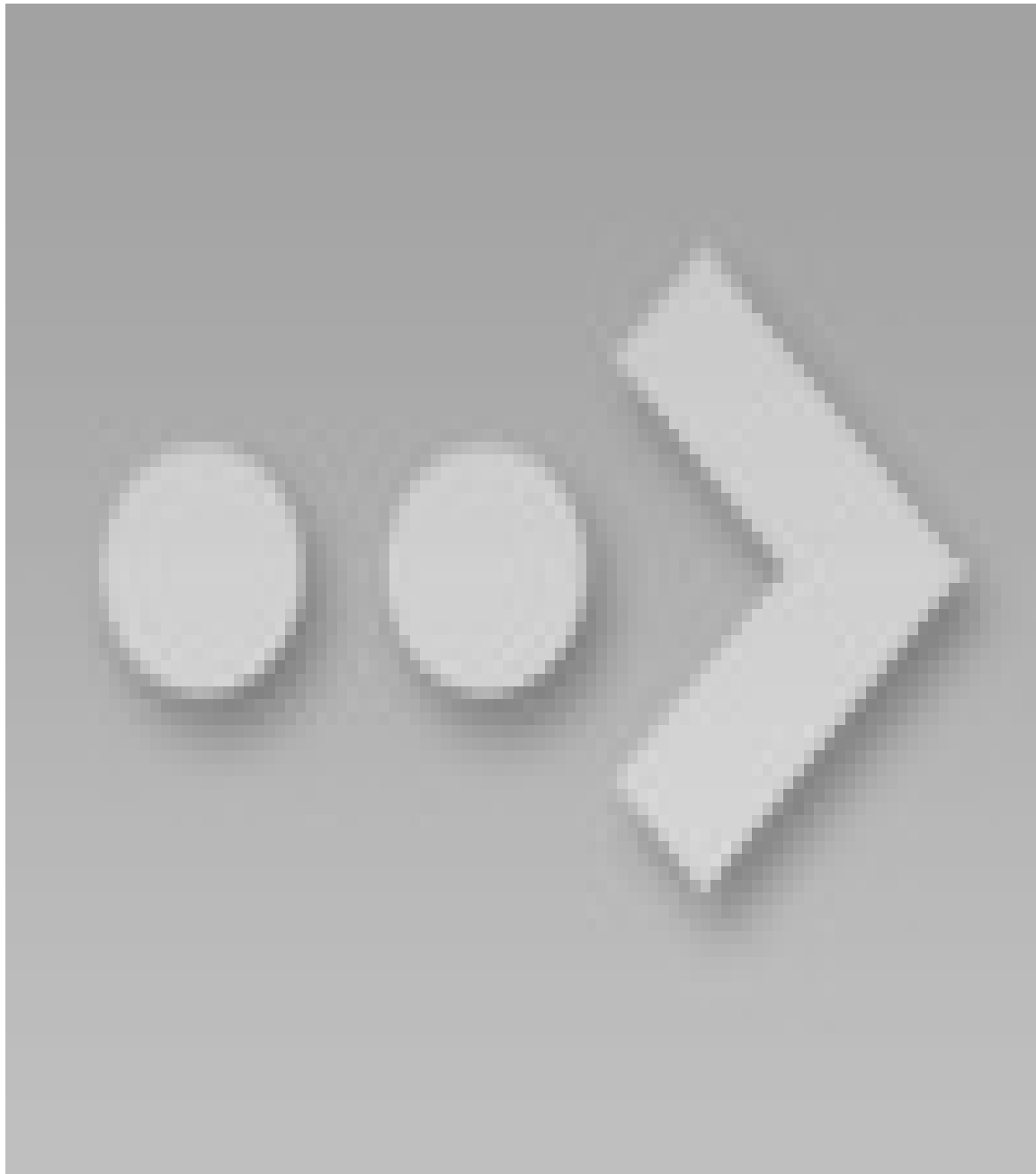


As mentioned in the introduction, while at this point we see how the running total has cumulated across our different sub-categories, it is not easy to determine the positive or negative contribution of each individual dimension member. To make this easier, we will convert this bar chart showing running total to a waterfall chart with a couple of additional steps.

First, change the mark type from ‘Automatic’, which is currently bar, to the

‘Gantt Bar’ mark type.

[Insert Image 62-4]



To get this view to look like the first image above, we need the tops or bottoms of each bar to line up at the same points on the Y-axis. To accomplish this, we have to size the Gantt bars by something in order to extend them. While your first instinct may be to size the Gantt bars by the Profit measure or even the Profit measure on the view which includes a table calculation for running total,

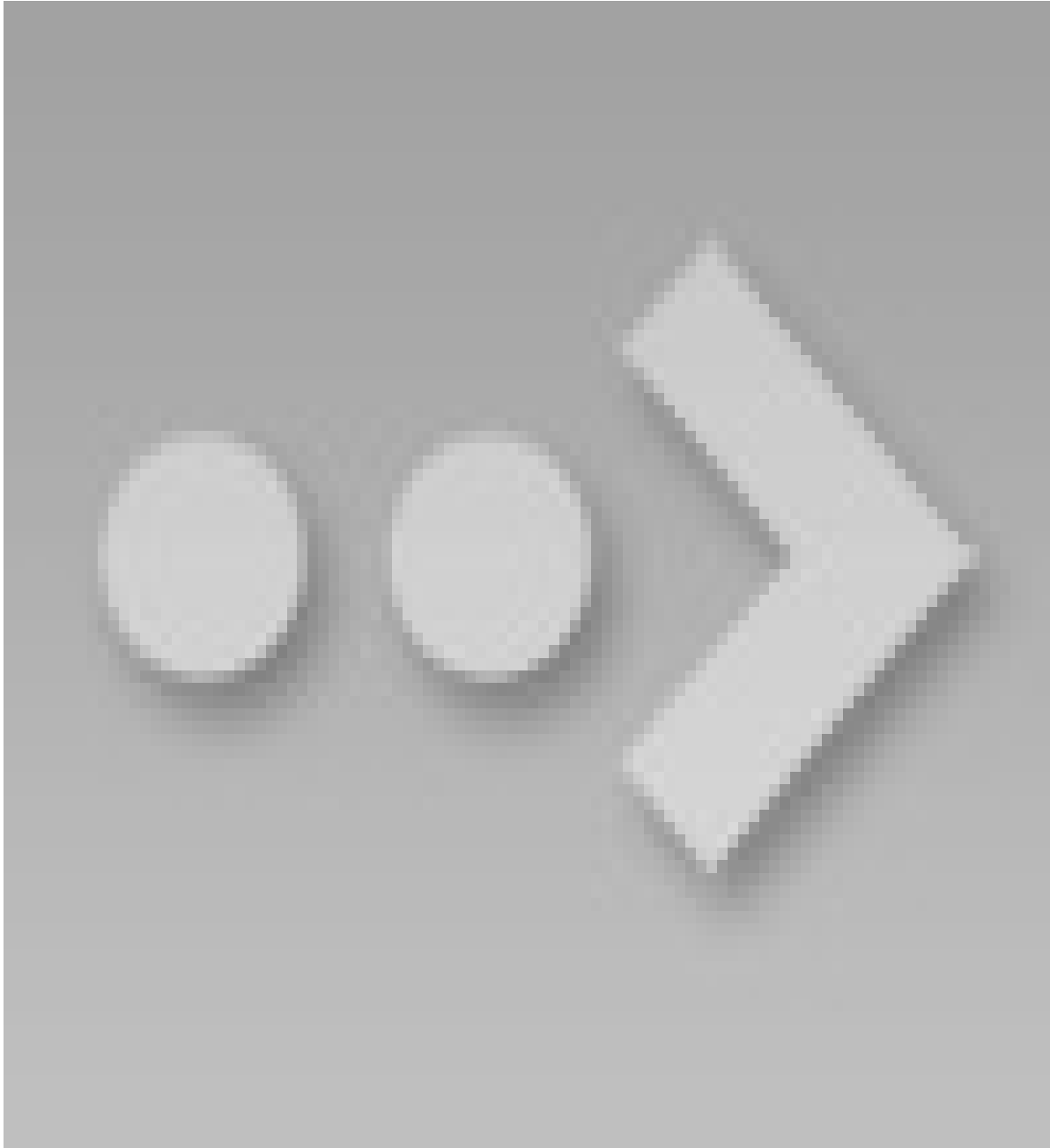
there is a trick involved with this step to get the desired effect. In order to get the Gantt bars for each dimension member to properly line up, you first have to create a new calculated field which takes the measure in the waterfall chart multiplied by negative one. This example is using the Profit measure, so I will create a new calculated field that equals $-\text{[Profit]}$:

[Insert Image 62-5]



Once this calculated field has been created, this is the measure that you drag to the Size Marks Card to create the waterfall effect:

[Insert Image 62-6]



At this point, we have an effective waterfall chart, but there are a few things I did to polish the final product as shown at the beginning of this chapter:

- Cleaned up the axis formatting.
- Colored the Gantt bars by Profit by dragging the Profit measure to the Color Marks Card; this created the blue and red color coding which represents positive and negative values, respectively.
- Added a total to the far right side of the visualization by navigating to

Analysis > Totals in the top navigation and choosing “Show Row Grand Totals”.

Lastly, you may choose to sort the dimension members by the sequence in which they were introduced or their values by ascending or descending order. As with many uses of Tableau, there is a great deal of inherent flexibility. These types of choices will depend on your analysis, business requirements, and business questions. Now that you know how waterfall charts are constructed in Tableau, experiment with the sort order of the dimension members to get the visualization that works best for you.

Chapter 63 – How to Make Dual-Axis Slope Graphs

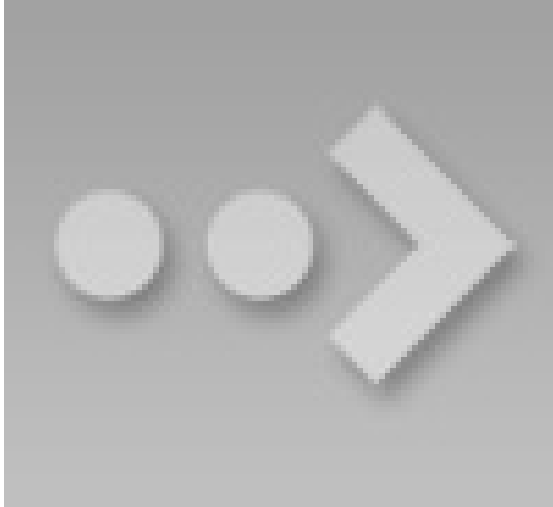
Slope graphs, or essentially line graphs between two points, are one of my favorite Tableau charts when my analysis requires a comparison between two data points. They work so well, in fact, that they are the one chart I will use to connect lines between discrete categorical variables. In almost every scenario, lines should only be used to connect points in time, but with slope graphs, I am okay drawing a line between ‘Thing 1’ and ‘Thing 2’, as well as ‘Time 1’ and ‘Time 2’.

This chapter shares not only how to make slope graphs in Tableau, but how to enhance them by leveraging a second axis to provide additional context.

How to make slope graphs in Tableau

First, let's knock out the traditional slope graphs using this data:

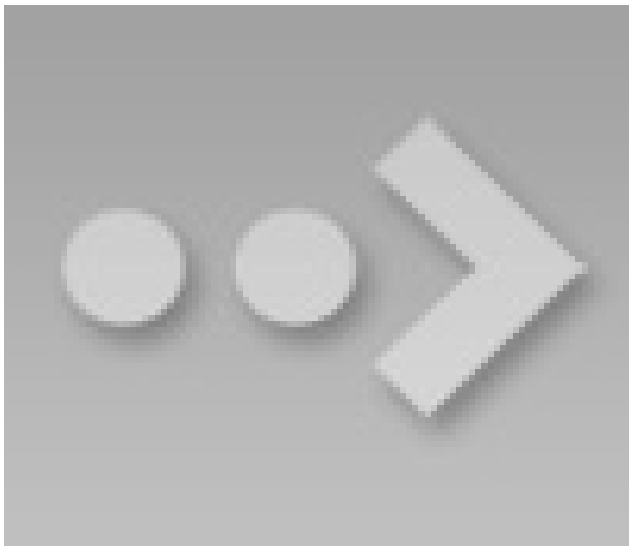
[Insert Image 63-1]



Step 1 - Create a Bar Chart

Create a bar chart with your 2-point time comparison on the Columns Shelf as a discrete dimension and your measure on the Rows Shelf.

[Insert Image 63-2]

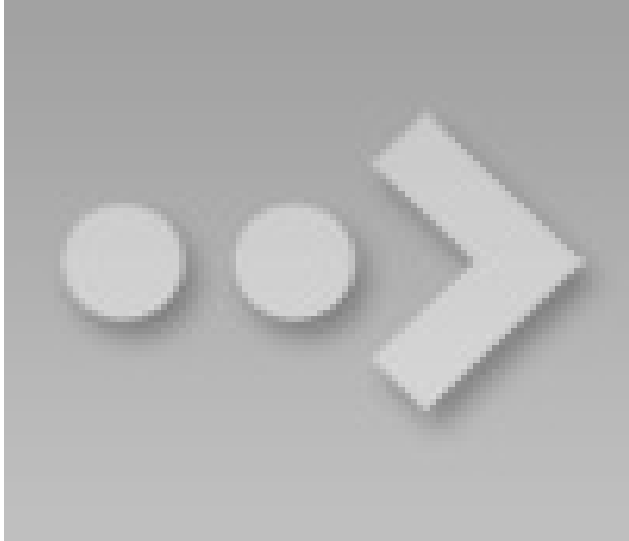


Step 2 - Add Dimension to Level of Detail

Add the dimension you are wanting to compare to the Detail marks card. This

Add the dimension you are wanting to compare to the Detail marks card. This will eventually create one line per categorical variable.

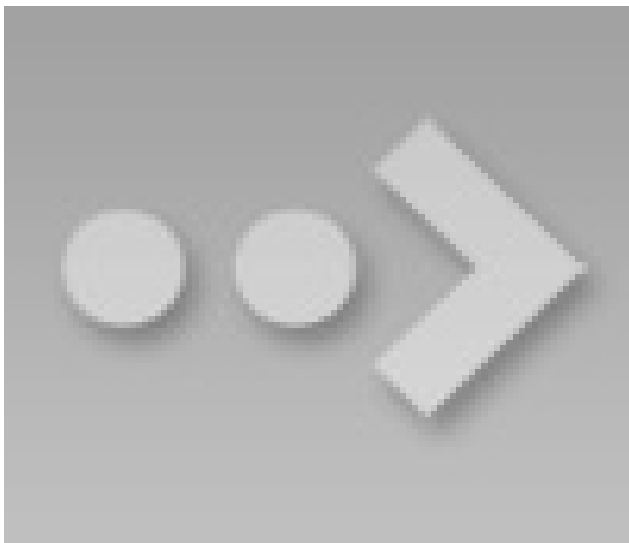
[Insert Image 63-3]



Step 3 - Change the Mark Type to Line

Change the mark type from Automatic (currently set to bar) to line to create slope graphs.

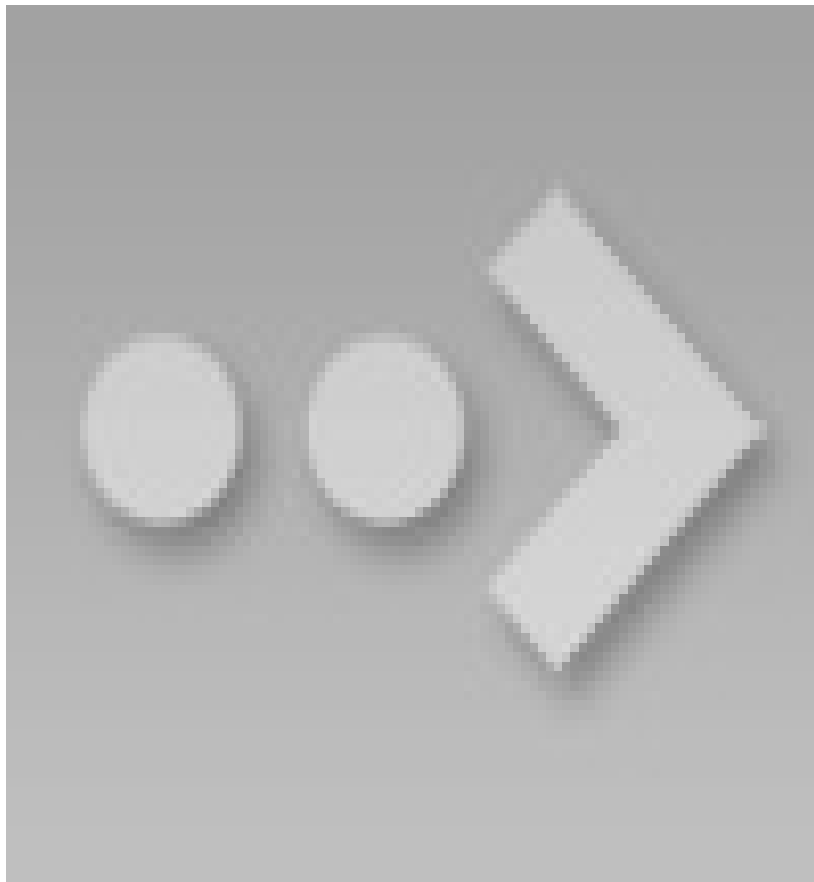
[Insert Image 63-4]



Step 4 – Format

To finalize the view, add labels and markers to the line ends. To update the labels, place the fields you want on the view to the Label Marks Card, and edit the Label Marks Card to show labels on the line ends. The markers are added by editing the Color Marks Card.

[Insert Image 63-5]



How to make dual-axis slope graphs in Tableau

Disclaimer before I let the genie out of the bottle: In the Thing 1 vs. Thing 2 scenario I'm about to share, I only think this works well if you are comparing two things. One may get bored from this as the animation officially

exactly two things. Once you get beyond two things, the visualization officially becomes a line graph, and should never be used to connect dimensions that are not elements of time. The best practice is a simple bar chart.

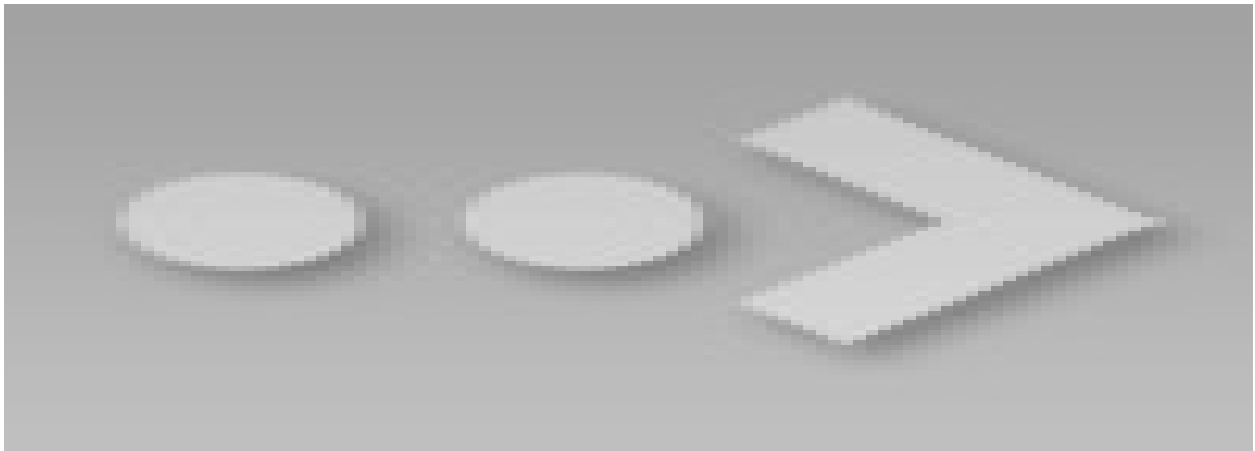
For the purposes of this exercise, the underlying data looks like this:

[Insert Image 63-6]



Our final product will look like this:

[Insert Image 63-7]

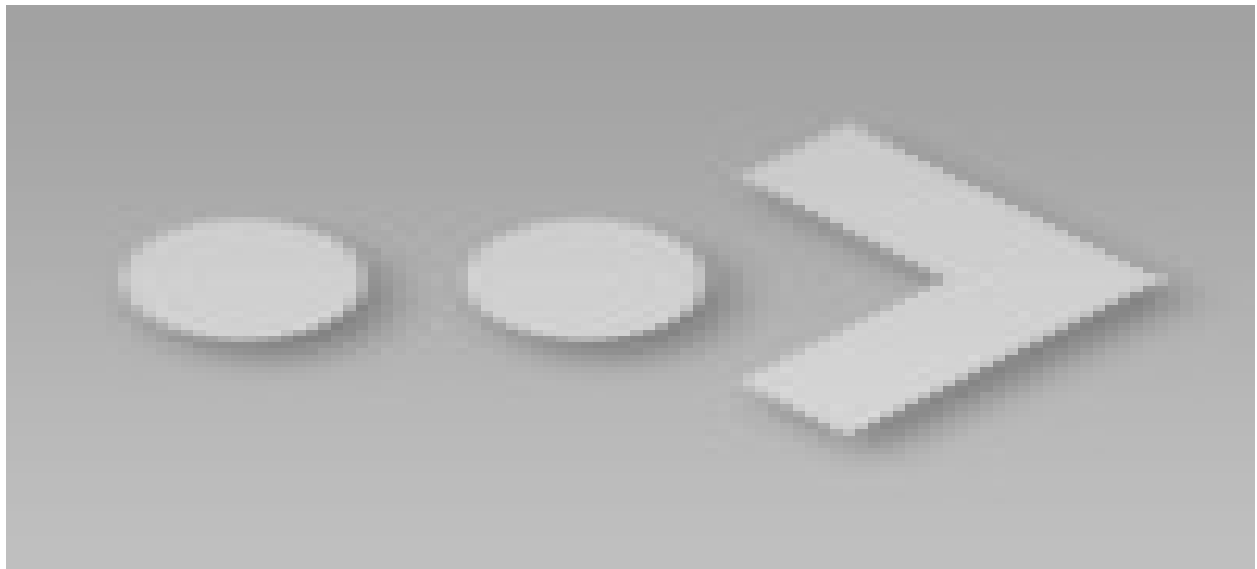


I'm using a Company A vs. Company B scenario just to show a slightly different use case, but this same exact process is applicable if you are comparing two points in time.

Step 1 – Create a Bar Chart

As mentioned above, the first step to creating slope graphs is to get the columns set up with whatever dimensions we are comparing. Sometimes with traditional slope graphs, there can be overlapping points and labels if data points are too close together. In that case, and when comparing two things (vs. times), I prefer to place the more granular dimension (in our case, social network) on the columns shelf first, followed by the dimension that includes 'Thing 1' and 'Thing 2' or 'Time 1' and 'Time 2'. The measure we are analyzing goes on the rows shelf, to end up with a bar chart view like this:

[Insert Image 63-8]



Again, this chart type is best practice when we are comparing two discrete categorical variables such as Company A vs. Company B, so stop here if you are feeling any discomfort with looking at this in a different way.

Step 2 – Change Mark Type to Line

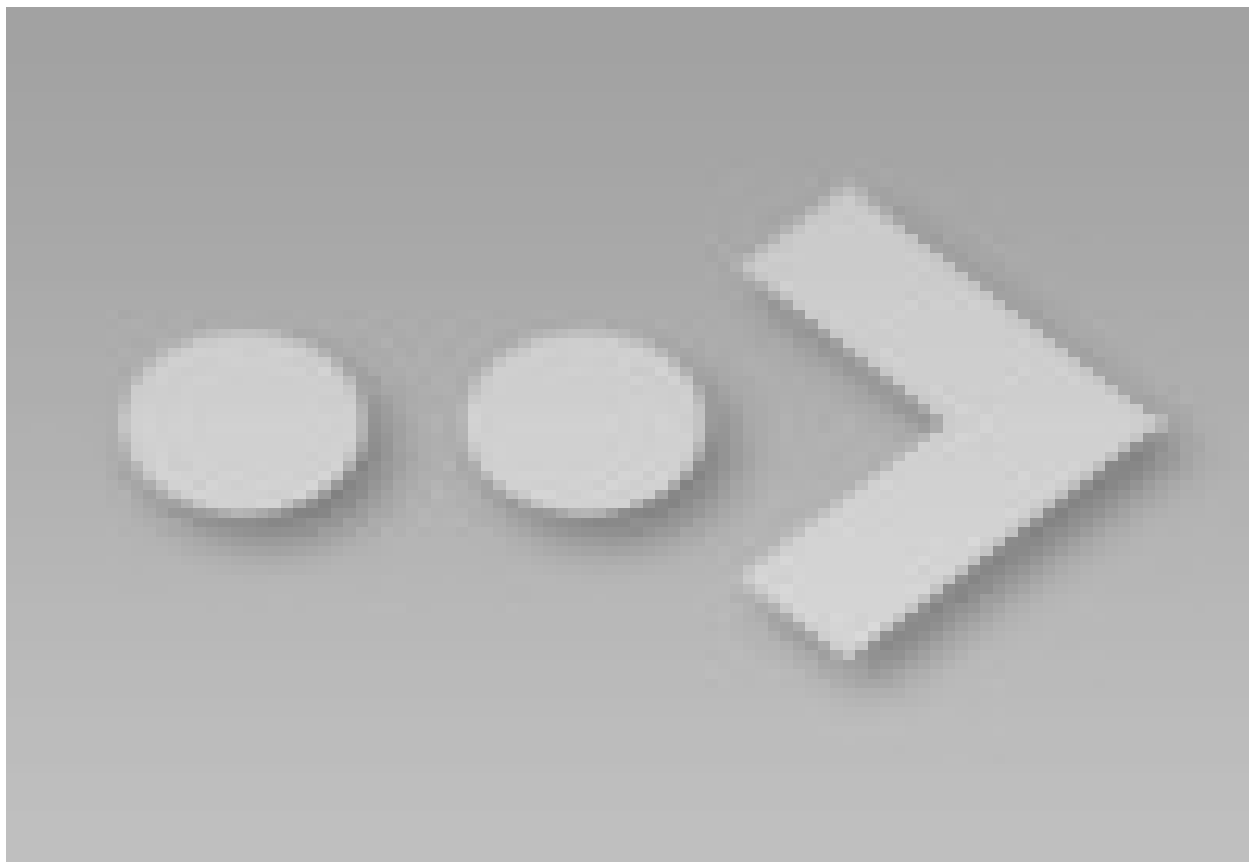
From here, a bar chart is converted into a slope graph by simply changing the mark type on the view from bar to line.

[Insert Image 63-9]



You can get a nice look from here by adding markers to the line ends.

[Insert Image 63-10]



At this point, you already have slope graphs in Tableau. This is an awesome chart type. If you want an incredibly awesome chart type, keep reading.

Step 3 – Create a Dual-Axis Combo Chart

Create a dual-axis combo chart by dragging the measure you're interested in (in our case, Size of Audience) to the opposite axis, and change the mark type on the second marks card to circle. If you are not familiar with building dual-axis combo charts, be sure to check out the chapter, How to Make Dual-Axis Combo Charts in Tableau.

For best results, also be sure to synchronize the second axis by right-clicking on the right-axis and choosing “Synchronize Axis”. At this point, my chart looks like this:

LINE CHART.

[Insert Image 63-11]



Step 4 – Customize the Second View

Now that we have two measures on the rows shelf, we have two marks shelves that we can edit independently of each other. This means we can keep the slope graph on one marks shelf, but change the size, colors, and context on the circles in the secondary marks shelf.

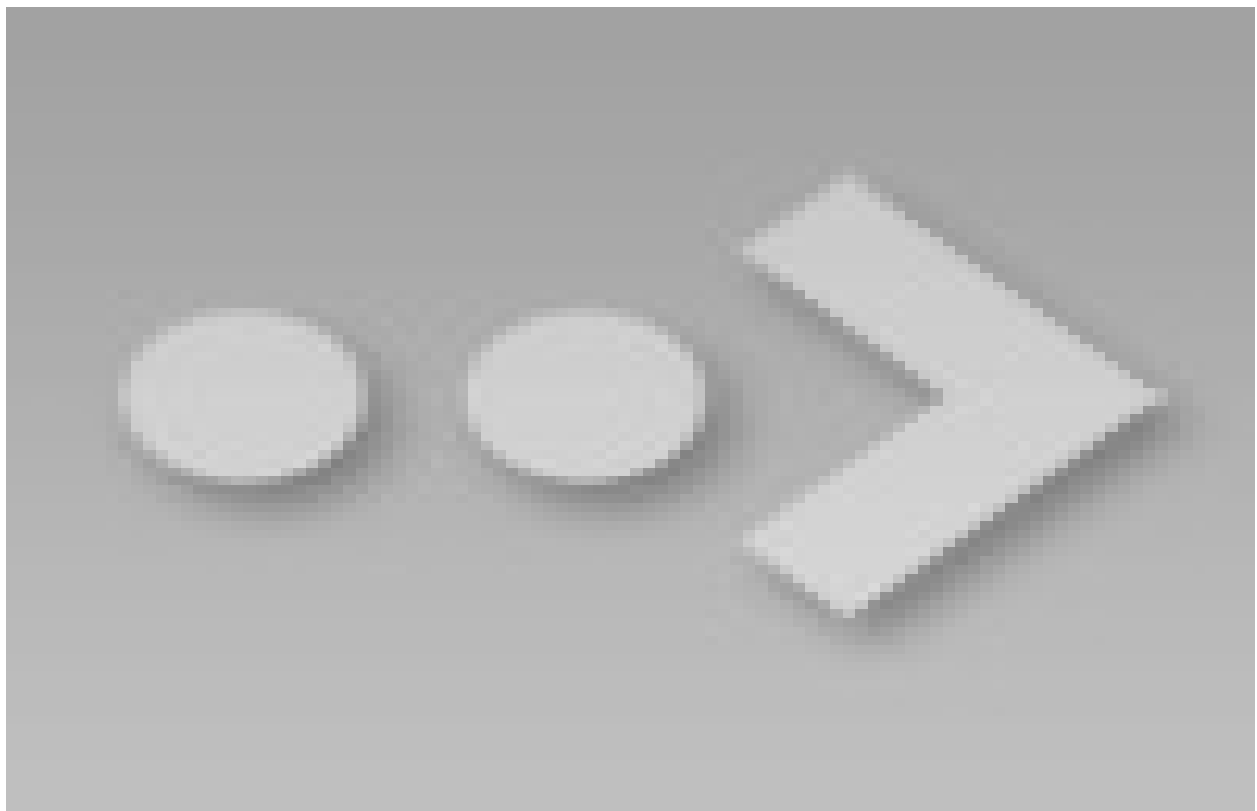
As just one example, I'll use the circles to display the delta between ourselves and our main competitor. I will also color both the slope graphs and circles to make our brand clear on the view. Here are the steps:

1. Change the sort order so the top competitor is listed first and Our Awesome Company is listed second. This is more intuitive when you are comparing things across a table. The sort order can be changed by clicking on the Company dimension on the Columns Shelf.
2. Add the Size of Audience measure to the Label Marks Card.
3. Add a quick table calculation to the Size of Audience measure for Percent Difference.

4. 4. Edit the table calculation to compute the percent difference along Pane (Across). This will execute the table calculation for each social network column.
5. 5. Add the Company dimension to the color marks card of both the slope graph and the circles (which is actually a dot plot) mark cards.

After some basic formatting, I end up with this:

[Insert Image 63-12]



This is just one example, but you can use this approach with other mark types, to show different KPIs, and/or to color based on performance such as a stoplight index. Slope graphs are an excellent choice for comparing two things - especially changes over two points in time - and they are made even stronger when additional context is added by leveraging a dual-axis in Tableau.

Chapter 64 – How to Make Donut Charts

When used properly, donut charts can be an effective way to communicate comparisons in a unique way. When used improperly, they are the butt of jokes mentioned in the same vein as pie charts. In fact, donut charts are essentially pie charts with a circle in the middle, and in the data storytelling section, I explain why you shouldn't use pie charts.

So what's the difference with donut charts?

One of my recommendations for practitioners that cannot quite let go of pie charts is to use five slices or fewer. Personally, I would not use more than two. When used for the specific purpose of showing a metric's progress to goal, with one "slice" being the current state of the KPI and one "slice" being the remainder to goal, I think a donut chart works well.

I admit, a bullet graph would be the most efficient way to communicate the progress to goal scenario above. In other words, a bullet graph would be processed faster by end users than a donut chart and also take up less real estate on a dashboard. If you think a bullet graph may be a more appropriate choice for your data visualization, see [How to Make Bullet Graphs in Tableau](#).

That being said, I don't mind the extra real estate that a donut chart takes up because that space can be used to provide additional context in a visually appealing way. For example, you can place an icon that denotes what the donut chart applies to or a callout number that shows the percent your KPI has progressed to goal. Donut charts are an acceptable alternative chart type that can make your data visualization stand out. For more on making your work 'remarkable', see tip twelve in the Data-Driven Storytelling section, [Balance Data and Design](#).

How to make donut charts in Tableau

Step 1

For the purposes of this exercise, we will make a gauge showing how our sales are progressing towards goal. I am using the Superstore sample data in Tableau. This data already contains a measure for 'Sales', but we will need to create one more field that calculates the sales goal minus the actual sales. The data you are working with may contain a measure for 'Sales Goal', but I am going to hard code a goal of \$8,000,000 into my calculated field: $\$8,000,000 - \text{SUM}([\text{Sales}])$.

[Insert Image 64-1]



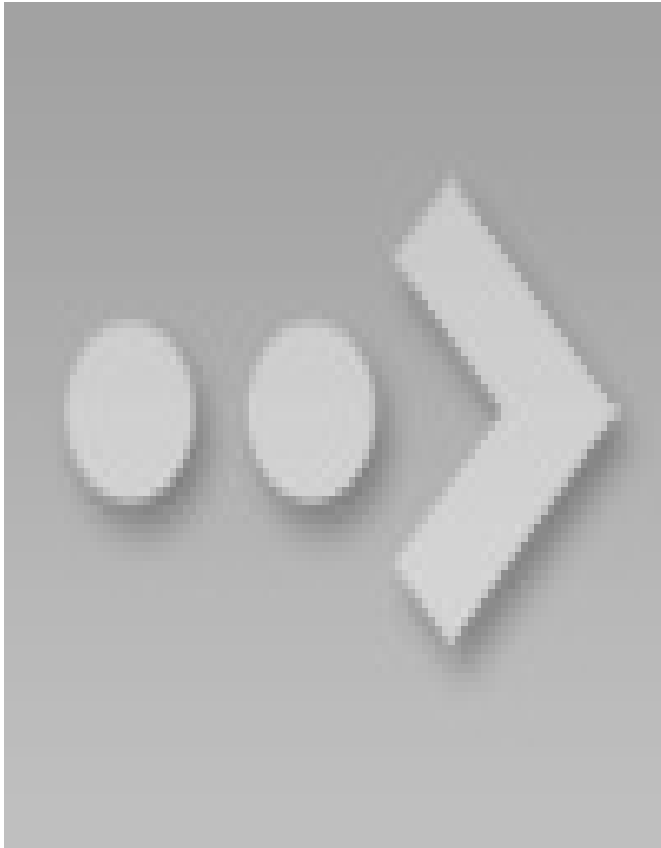
The reason we make this calculated field is so the two slices in our donut chart will always add up to the total goal, and actual sales will always be a correctly sized portion of the total goal.

Step 2

You will now create a pie chart using two measures: 'Sales' and the newly created 'Sales Goal – Actual Sales'. Do this by first changing the mark type on your view to Pie. Now place 'Measure Names' on the Colors marks card and 'Measure Values' on the Angle marks card. Finally, filter the view on 'Measure Names' to only include 'Sales' and 'Sales Goal – Actual Sales'. At this point,

your view should look like this:

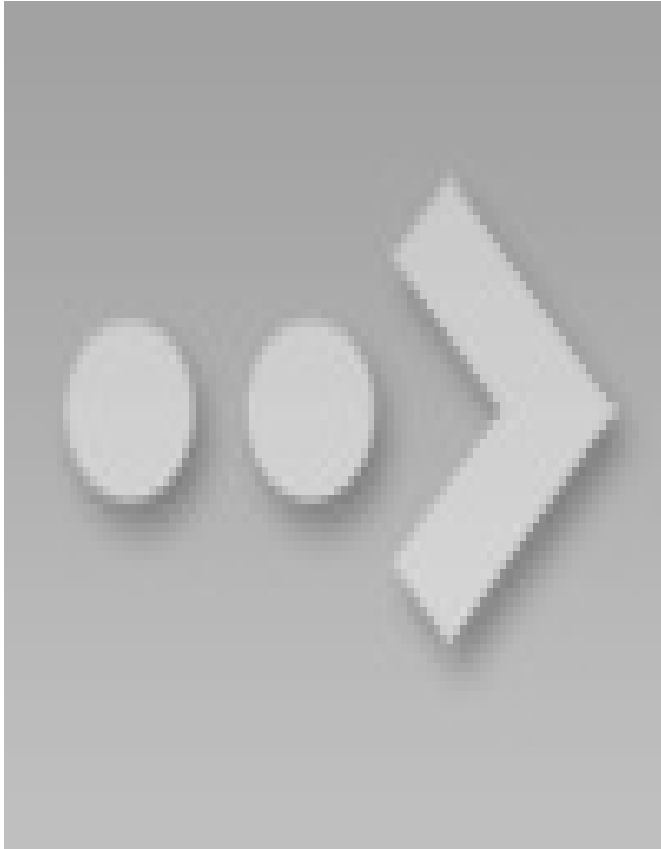
[Insert Image 64-2]



Step 3

Format the donut chart (currently a pie) by increasing the size, adding a border, and most importantly: coloring the ‘Sales’ measure to a color of your choice and the ‘Sales Goal – Actual Sales’ measure white. This will eventually create the filling gauge effect we are going for.

[Insert Image 64-3]



Step 4

The final step in creating a true donut chart is to 'poke a hole' in the chart we have created so far by adding a circle to the middle. I have two approaches for this final step.

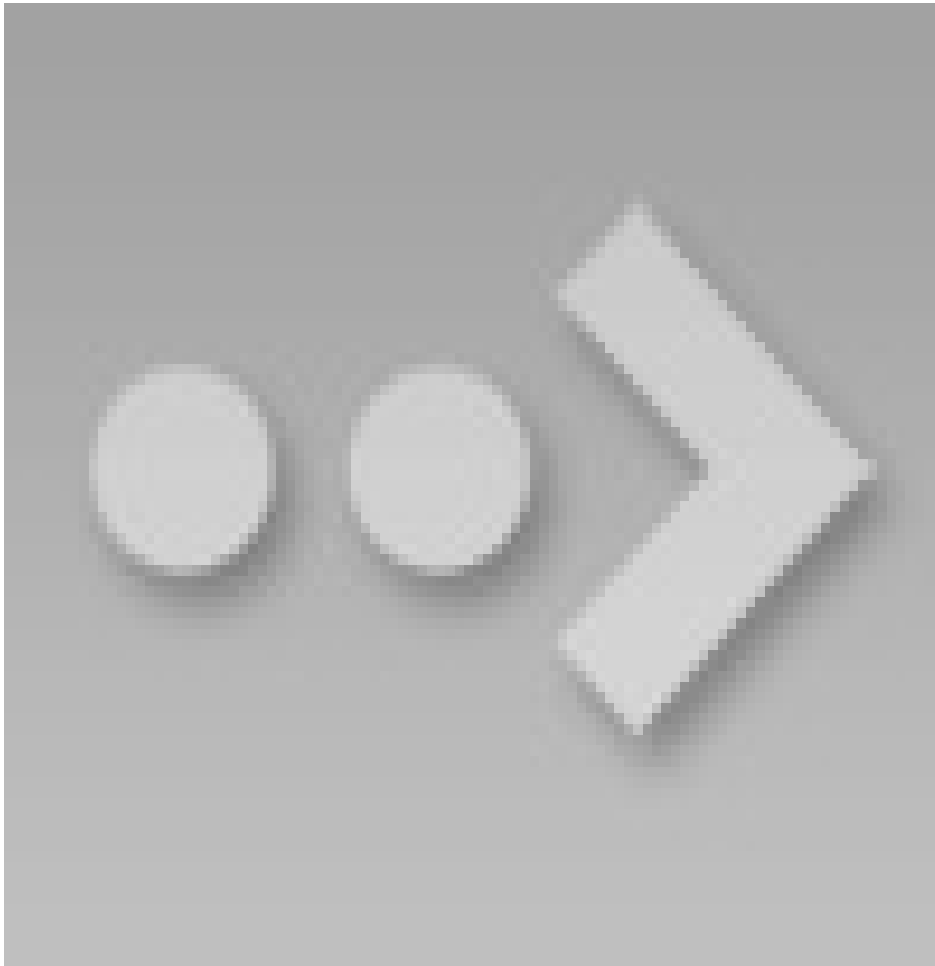
Approach 1: Single Sheet

For most situations, this first approach will be the most elegant solution.

- Start by creating a calculated field called "Placeholder" that has this formula: `MIN(0)`
- Create a Dual-Axis Chart by placing this new calculated field on the Rows Shelf, then adding it again to the opposite axis. At this point, my view looks

like this:

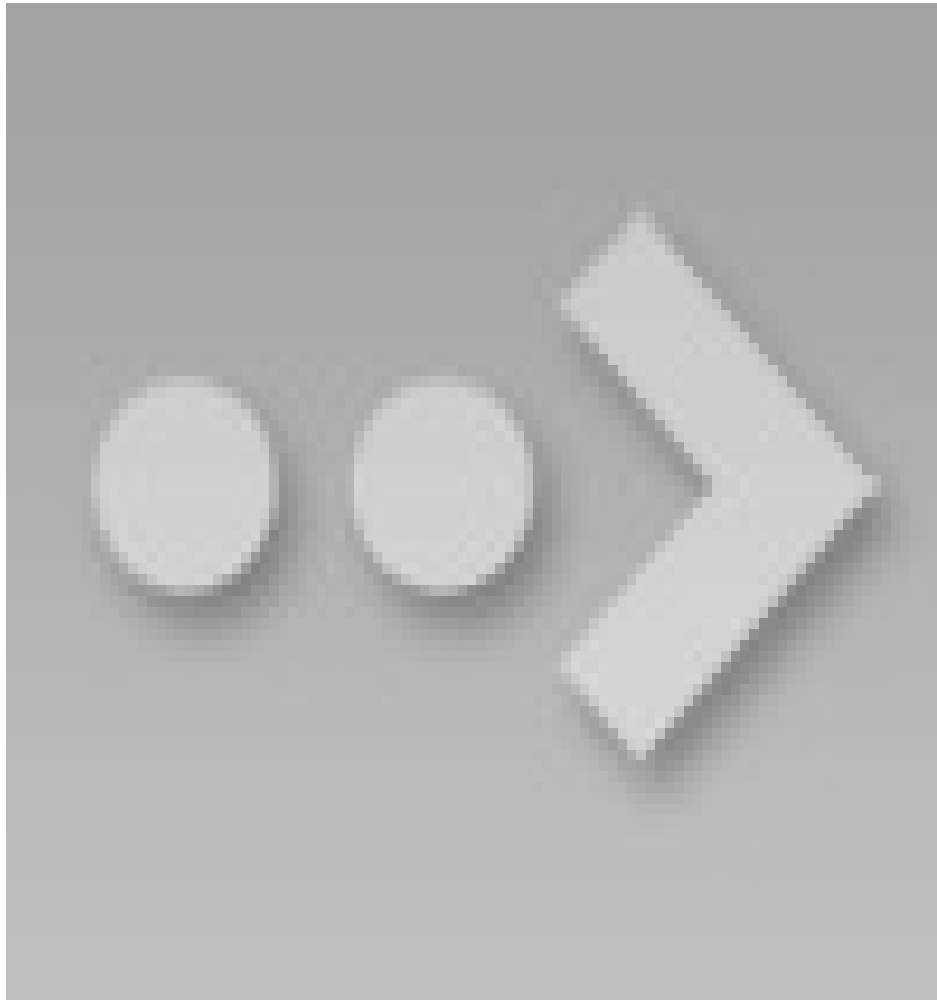
[Insert Image 64-4]



Notice that there are now two different Marks Shelves - one for each Placeholder field. That means that you can now edit the marks for each field independently of each other. On the Marks Shelf for the first placeholder, leave the settings as is with a mark type of pie - this is our original chart and we don't want to change it. On the Marks Shelf for the second placeholder field, change the mark type to circle, which will become the hole in the donut chart. Reduce the size and change the color of the circle as desired. You can also modify the marks for the second placeholder further by adding in a label, *etc.* At this point, my view looks

like this:

[Insert Image 64-5]

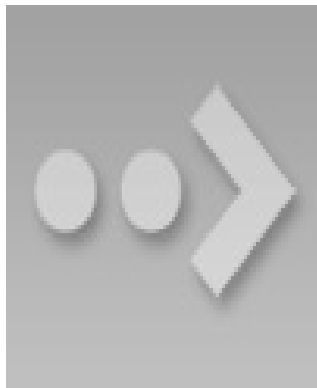


Finally, clean up the view by doing some simple formatting such as hiding zero lines and headers. You now have a donut chart that is comparing sales to sales goal. You can take this a step further by adding a dimension to the Columns shelf to create a small multiples view that compares performance across several different dimension members (i.e. Region, Segment, etc.).

Approach 2: Floating Sheets

If your visualization requires icons or additional graphic design, as is the case in my Your Salary vs. a MLB Player's Salary viz, I recommend you float a .png image over the chart. It is very important that you use a .png file so that your corners are transparent and you will be able to see the underlying chart.

[Insert Image 64-6]



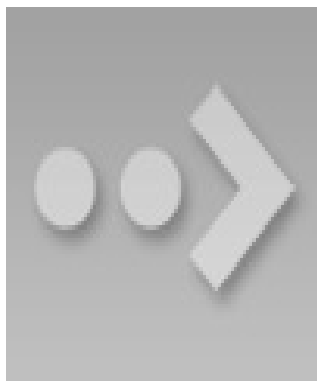
Finally, I mentioned earlier that I don't mind donut charts because the real estate available in the inner circle can be used to communicate additional context either through icons or text.

If you're using approach 1, you can create a calculated field that determines the percentage of sales goal obtained, and then add this calculated field to the label of the placeholder 2 marks card.

If you're using approach 2, you can enhance the .png image shared above with your own icons or other design elements. If you would like to use a calculated field to show progress to goal, you can create a sheet with that metric and float it over the chart and .png image.

In either approach, your final product will end up looking like this.

[Insert Image 64-7]



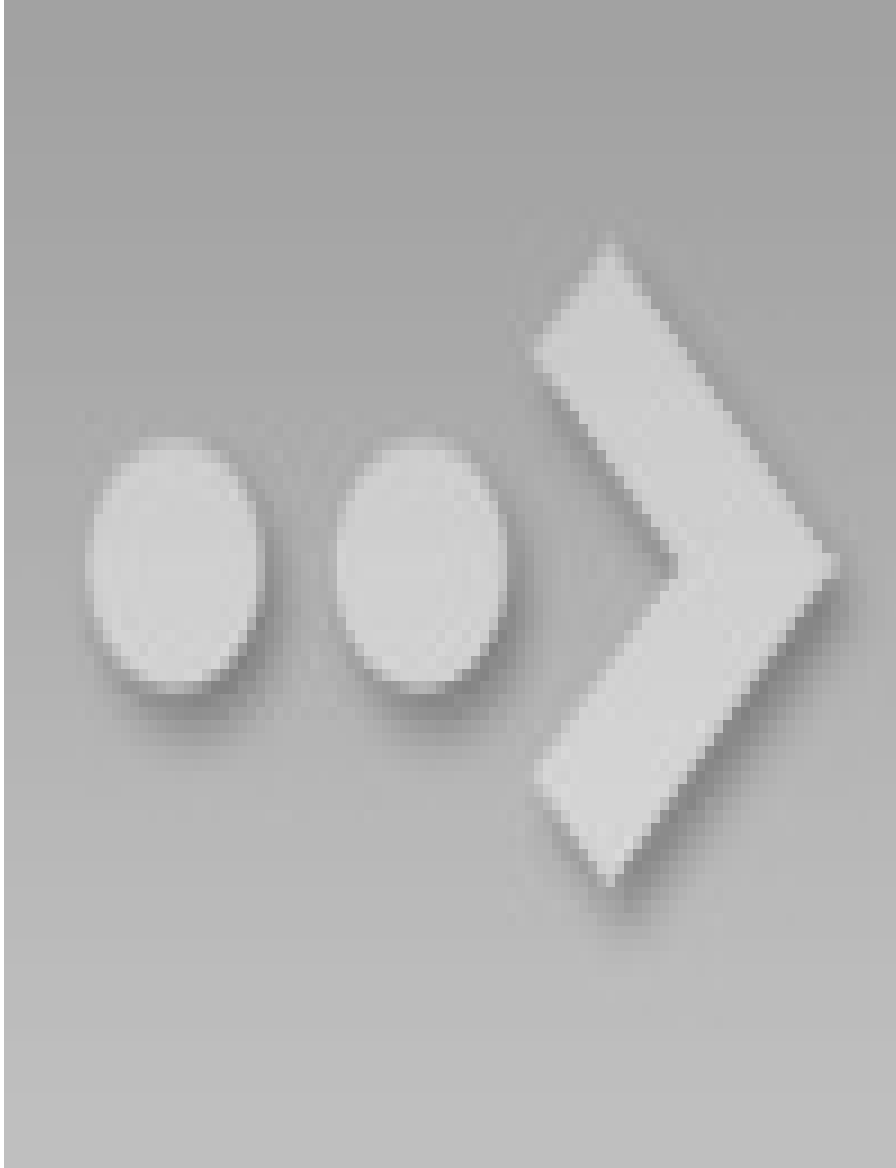
Chapter 65 – How to Make Funnel Charts

The Odds of Going Pro In Sports viz has generated more questions around how it was created than any other viz I have put together during my career with Tableau. With its one dominant funnel chart and icon-based navigation, the viz tells the story about the share of high school athletes progressing to the college and pro levels across several sports for each gender.

The most common question I receive: *That was made in Tableau?* I would be lying if I said that question doesn't make me want to stand up a little taller, but the secret is, the viz was one of the easiest dashboards I have ever put together. In fact, I put it together in a couple of hours on a Sunday afternoon. Funnel charts are one of the simplest chart types you can create, but they have proved to be incredibly effective in a corporate setting - think conversion rates and customer flows. Funnel charts are not one of the out-of-the-box “Show Me” options in Tableau, so this chapter will walk you through multiple approaches to creating funnel charts.

First, let's take a look at the full version of *What are the odds of going pro in sports?*

[Insert Image 65-1]

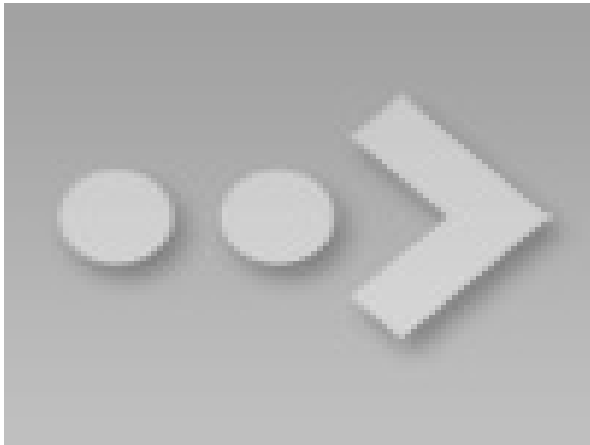


How to make funnel charts in Tableau

Option 1: The Step Dimension

Going into this project, I knew that I wanted the main view to be a funnel chart. For this reason, I was able to plan ahead and put the data in an optimal format for Tableau. Here is a sample of the underlying data:

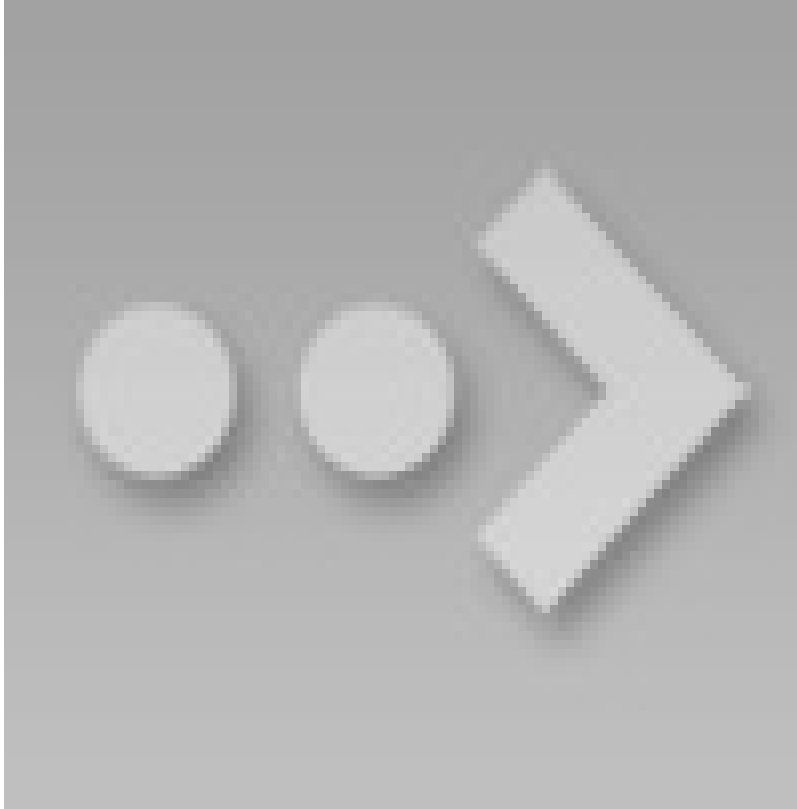
[Insert Image 65-2]



Note that there is a dimension of “Step”, which is just an arbitrary, intuitive name for each of the three levels of high school, college, and pro. The measure is then the corresponding number of players that reached each step. This is the easiest way to build a funnel chart; here are the steps:

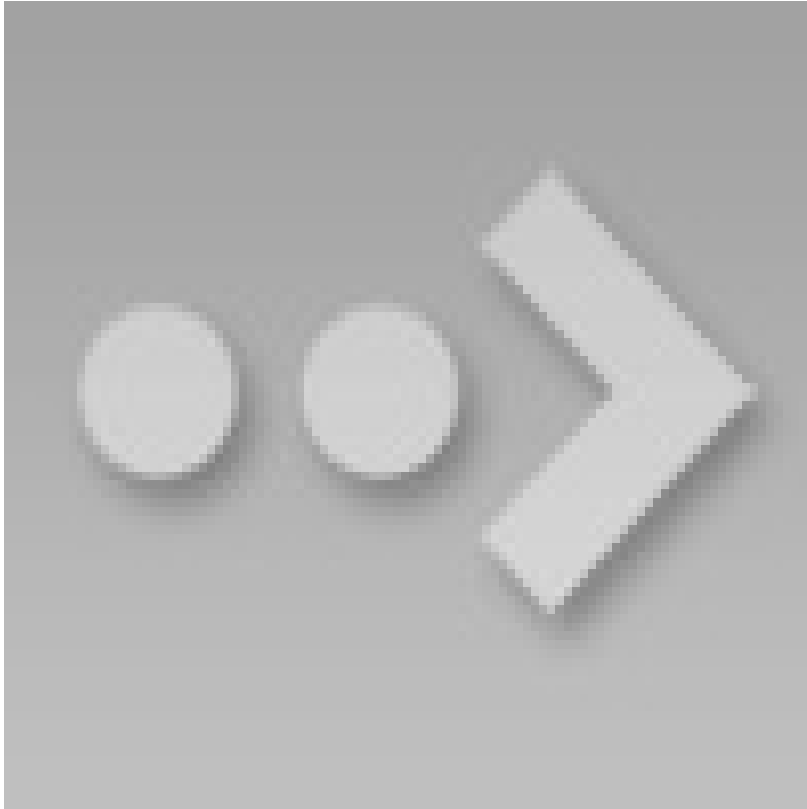
1. Create a horizontal bar chart by dragging your step dimension onto the rows shelf and your players measure to the rows shelf.

[Insert Image 65-3]



2. Step three is where the magic happens. Simply change the mark type from 'Automatic' (which is currently 'Bar') to 'Area'. You are left with the right side of the funnel.

[Insert Image 65-4]



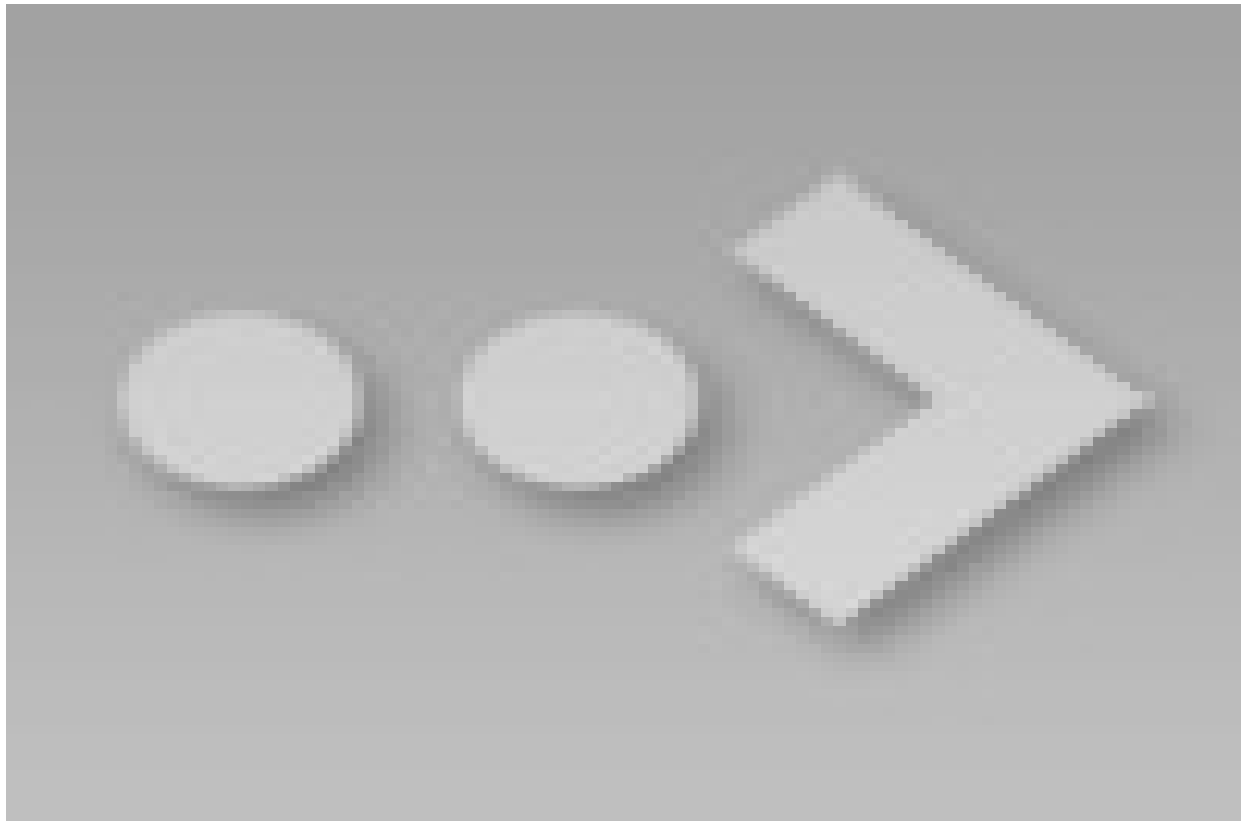
3. If your data is in a format outlined in the table above, there is a simple solution to mirroring the right side of the funnel onto the left side. To do this, you will create a calculated field by right-clicking on the 'Players' measure and selecting 'Create Calculated Field'. Name your calculated field 'Negative Players' and add a '-' in front of '[Players]'. Your entire formula will look like this:

[Insert Image 65-5]



4. The final step in creating a symmetrical funnel chart is to drag and drop your new 'Negative Players' measure in front of the 'Players' measure on the columns shelf.

[Insert Image 65-6]



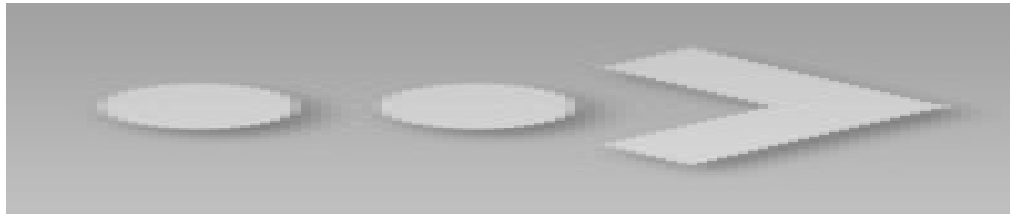
From here, your funnel chart can be used like any other chart in Tableau. Encode it by color (in my case, its colored by gender), add quick filters, and add tooltips *labels* annotations to make the data in the chart clear to the end user.

Option 2: Separate Measures

It is likely that your corporate data is not in the layout of the table above by default, and you may not be in a position to easily change this on your own. This solution is not as flexible or optimal for Tableau, but it is functional. Let's

Solution is not as flexible or optimal for Tableau, but it is functional. Let's assume that instead of the layout in the table above, your data looks more like this, with each 'Players' measure broken out by level, causing your layout to be more horizontal instead of vertical:

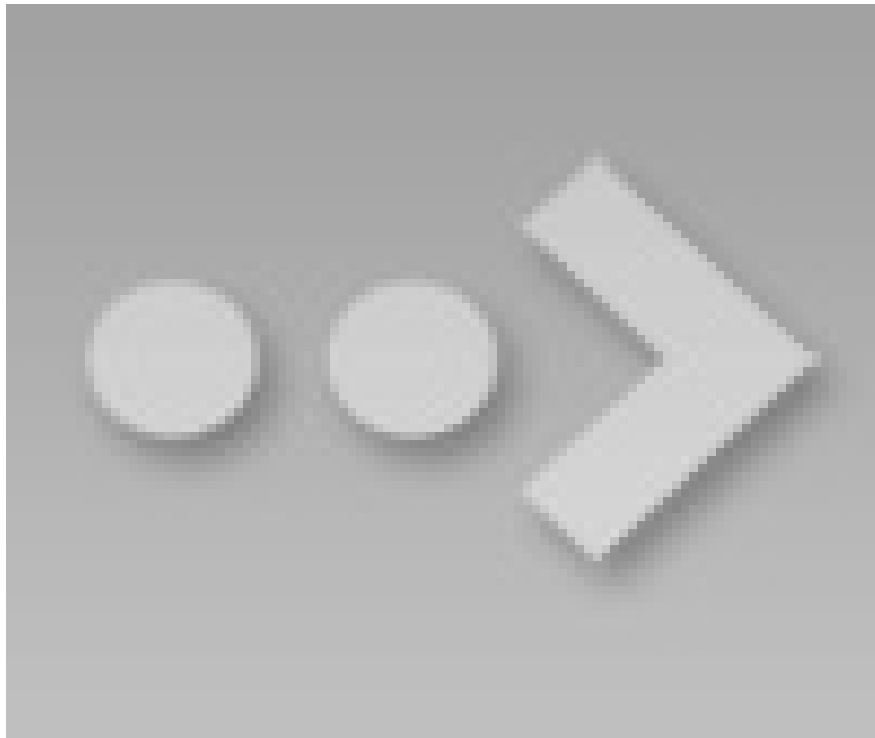
[Insert Image 65-7]



In this case, because you no longer have a dimension of step, the approach is slightly different:

1. To create the right side of the funnel, you will need to add the measures to the view individually. The easiest way to do this is to double click each of your three measures (Step 1 Players, Step 2 Players, and Step 3 Players) to add them to the view. Double-clicking each measure will create a small multiples chart, so you will have to use the 'Show Me' button to change the chart type to a bar chart. While the set-up is different, you should now see the beginnings of the right side of your funnel, just like in step one from the first approach above:

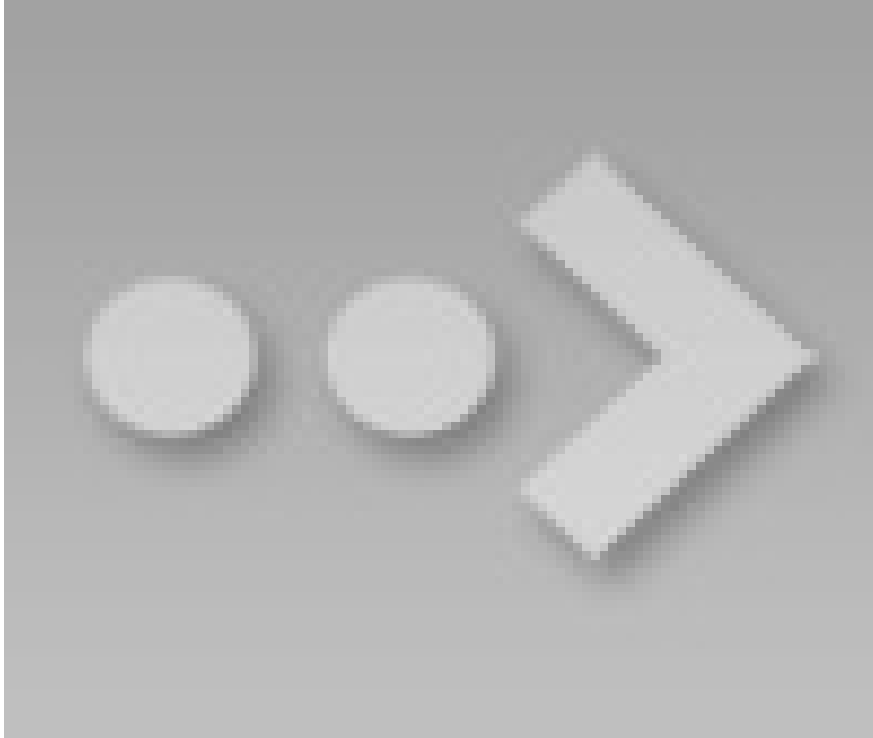
[Insert Image 65-8]



2. Follow Step 2 from the first approach, changing your mark type from 'Automatic' to 'Area'.

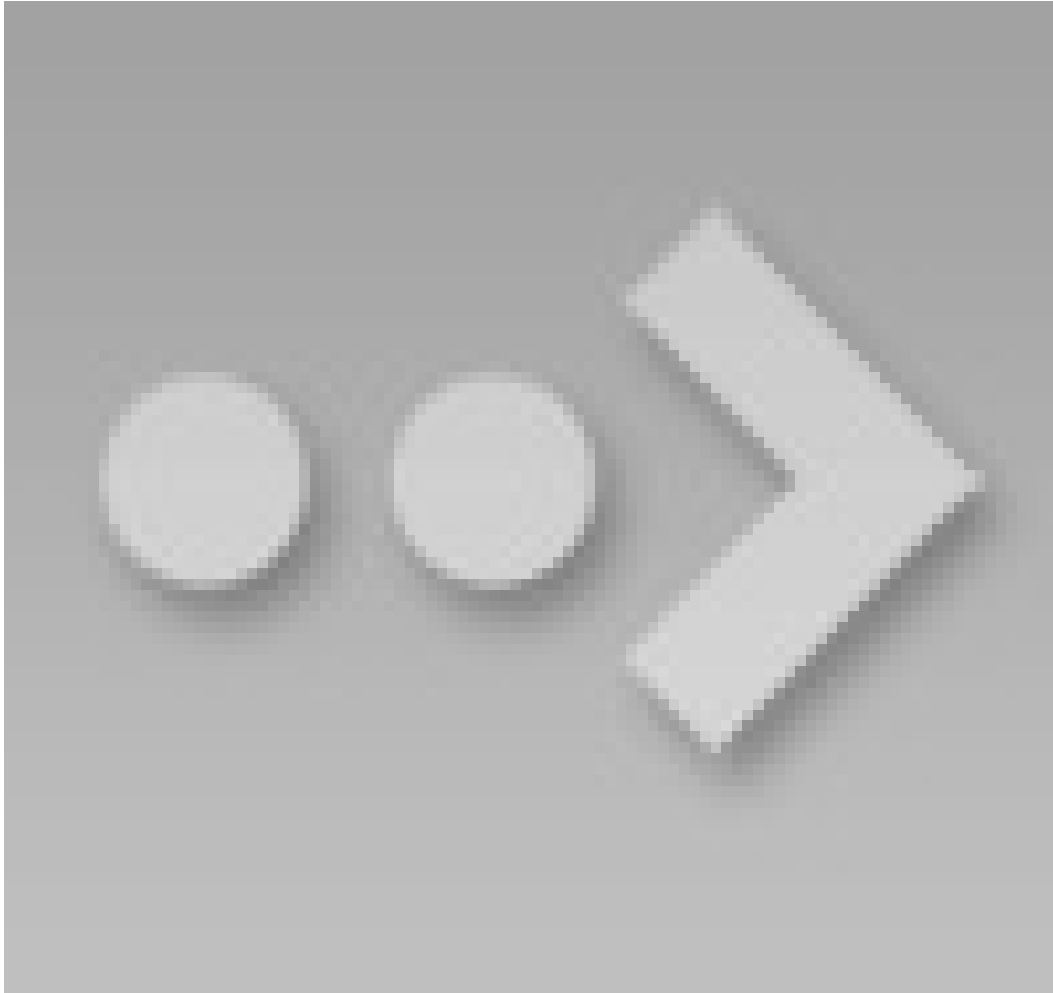
3. Step 3 on the second approach is where things get tricky. The first thing you need to do is create a negative version of each of your three measures, much like creating a negative version of the 'Players' measure in step four from the first approach. Once you have three new calculated fields, one negative version for each of your three measures (i.e. Negative Step 1 Players, Negative Step 2 Players, and Negative Step 3 Players), open a new workbook sheet and follow steps 1-3 above – this time using your newly created negative versions of each measure. You should end up with the left side of your funnel:

[Insert Image 65-9]



4. The final step to this approach is joining the left and right side of your funnel. This can be accomplished by floating each sheet on a dashboard and lining them up:

[Insert Image 65-10]



Admittedly, this second approach requires some hackish creativity, but it works with most data out of box without any reformatting.

So there you have it! Funnel charts in four simple steps, regardless of the layout of your funnel data.

Chapter 66 – Introducing Pace Charts in Tableau

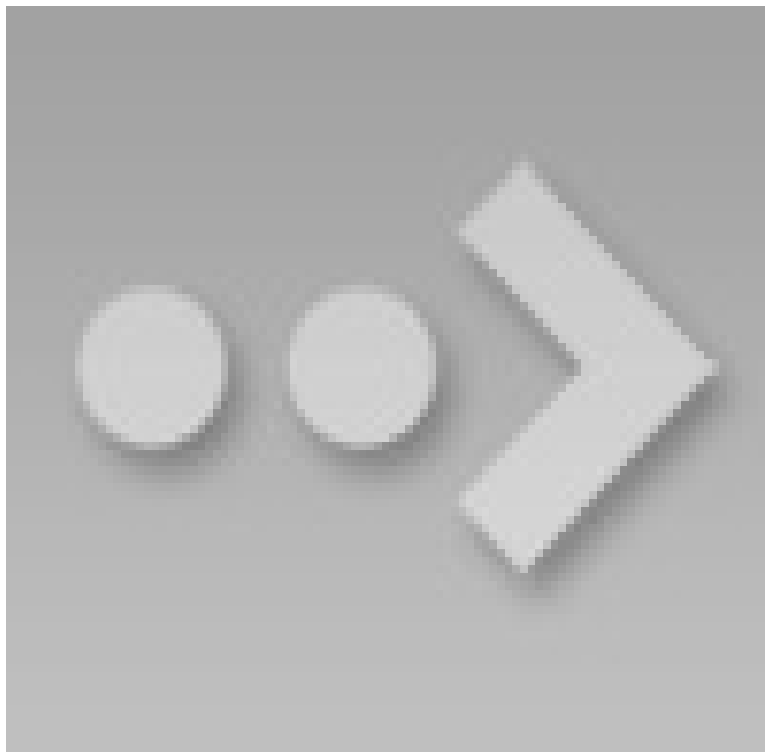
Pace charts are an alternative [bullet graph](#) design that normalize progress to goal visualizations across KPIs, even if the KPIs have different data formats, scales, and/or seasonal trends. They are useful for providing an ‘apples to apples’ pace to goal comparison in businesses that have KPIs that span different categories such as revenue, social media followers, attendance, *etc.*

With KPIs such as revenue, social media followers, and attendance, not only are the metrics in different formats, but they are often on very different scales and have varying seasonality. For example, you may gain social media followers throughout an entire year, but if you are an NFL team, your attendance won't start until August. Further, your growth on social media might be on a scale of thousands, while revenue may be on a scale of millions.

Pace charts normalize KPIs by comparing them all on an axis that ends at 100% (the goal). In addition to showing how much progress each KPI has made toward the goal, a linear or seasonal pace is displayed to illustrate whether progress to goal is on pace to reach goal. To enhance the illustration, the marks can be colored to show how current progress to goal for each respective KPI compares to its pace to goal.

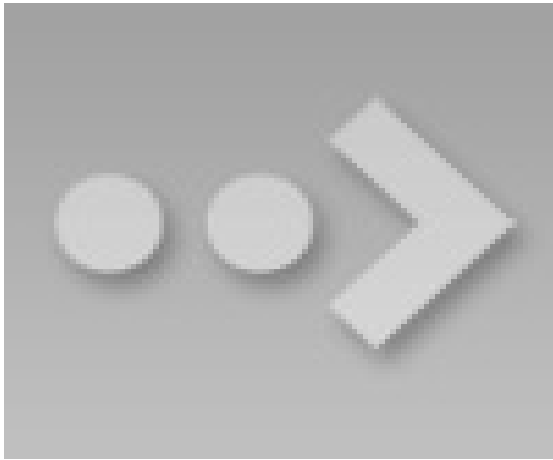
To illustrate how to create a pace chart in Tableau, I will start by recreating this pace chart showing a variety of KPIs that are on a linear pace (i.e. they should all be at the same progress to goal at this point in the year):

[Insert Image 66-1]



The underlying dataset used to create this pace chart looks like this:

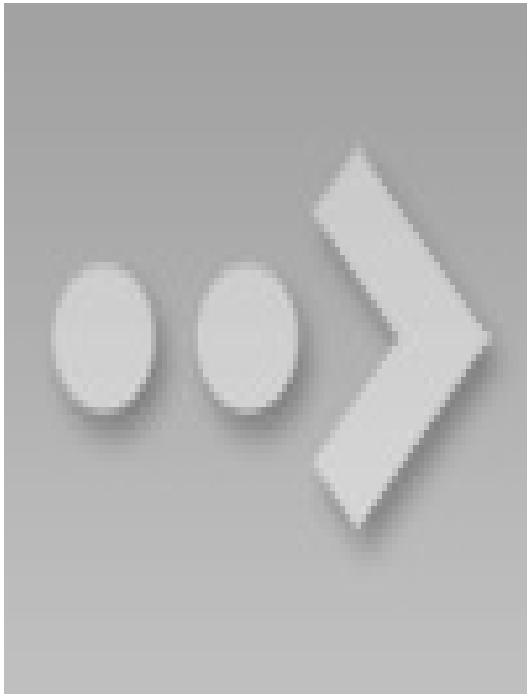
[Insert Image 66-2]



Even though the KPIs are on very different scales, it is easy to compare them to determine which are on pace, slightly behind pace, or behind pace.

To affirm this point, here is the same data using a traditional bullet graph:

[Insert Image 66-3]



With this visualization, it is impossible to compare the progress to goal of KPI 3, which is on a much smaller scale than the other KPIs. It is also challenging to determine the progress to goal of KPI 7, because it is on the same scale as KPI 1,

which has the largest goal and is extending the X-axis. While you could break this graph up into seven different parts to fix the scaling, there is a better way to normalize the data.

How to create a pace chart with a linear pace in Tableau

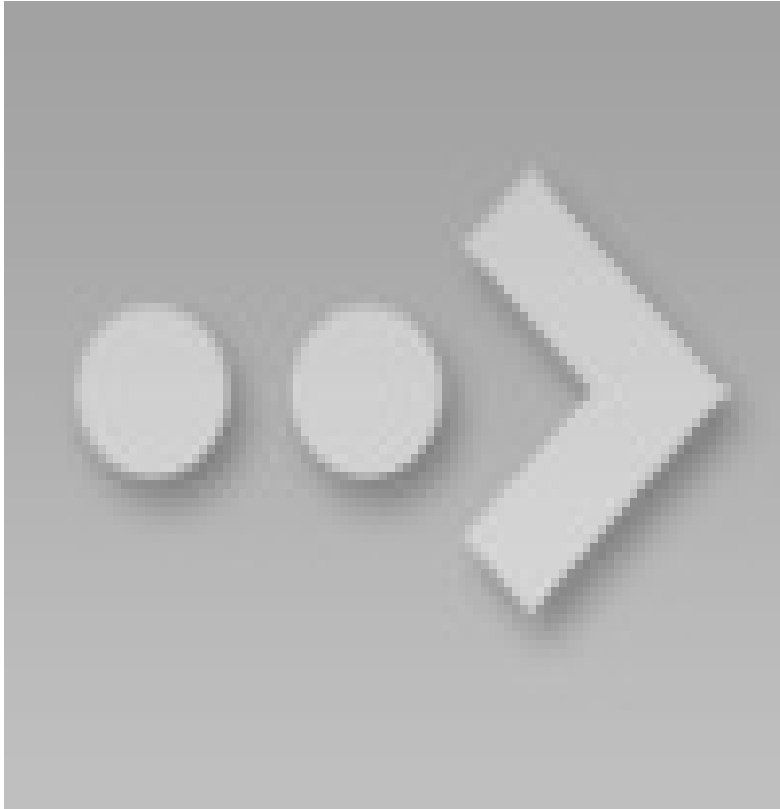
To normalize the bars in a pace chart, create a calculated field which calculates the progress to goal. This is used to represent the bars instead of the current values. The formula for this calculation is $\text{[Current Value]} / \text{[Goal]}$

[Insert Image 66-4]



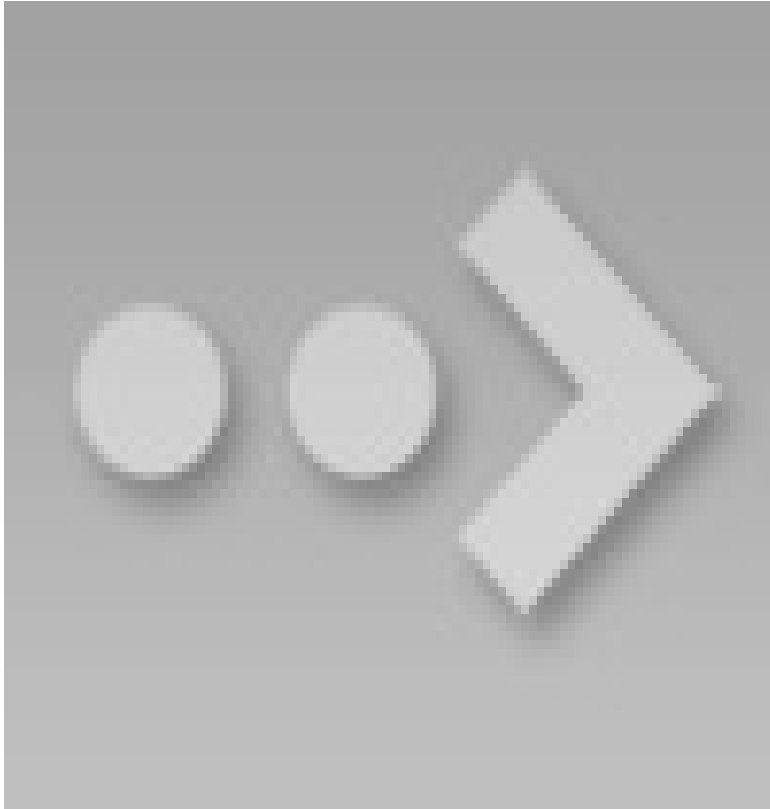
After you have created the Progress to Goal calculated field, create a horizontal bar chart showing the progress to goal for each KPI:

[Insert Image 66-5]



With a pace chart, the goal line is always normalized across every KPI at 100%. To add this reference to the visualization, simply add a reference line with a constant of 1 (which equals 100%):

[Insert Image 66-6]

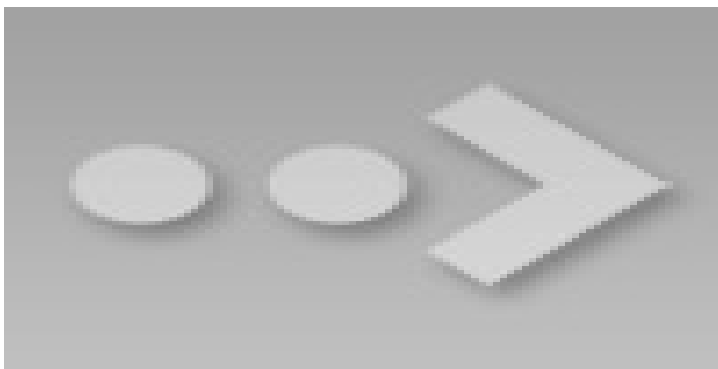


For this illustration, we will pretend that the pace to goal should be the same across all seven of our KPIs. When using a linear pace, a calculated field can be created to calculate how far to goal each KPI should be at the current point in the year. For example, if we are in week 42, the pace calculation would be:

$$(1/52) * 42$$

This calculation is dividing the year into 52 equal parts (i.e. weeks), then multiplying that fraction by the number of weeks that have passed in the year.

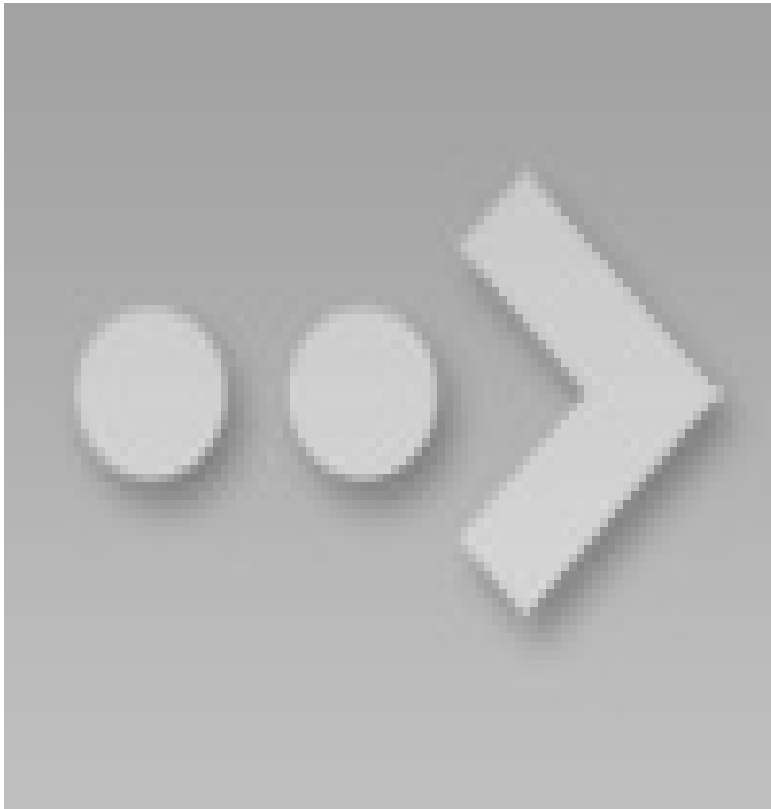
[Insert Image 66-7]



In Tableau, you have the option to replace the 42 with a [parameter](#) that allows the end user to change the multiplier.

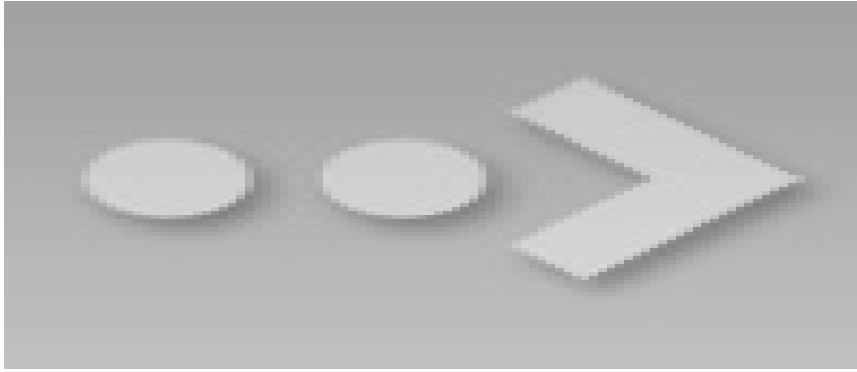
Once you have the Pace calculated field, add it to the Detail Marks Card so it can be used as a reference line. Then add a reference line that shows where the pace should be at this point in the year:

[Insert Image 66-8]



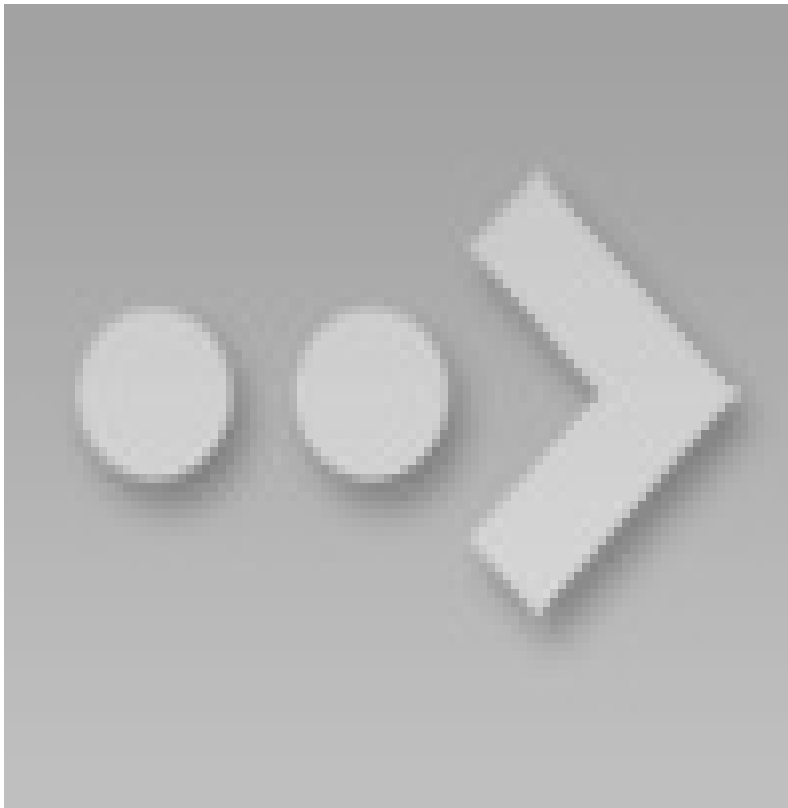
Lastly, to color the bars to illustrate whether each KPI is on pace, slightly behind pace, or behind pace, create a calculated field with the scoring logic. This will vary based on your own requirements, but as one example, I'll pretend that 100% or above is on pace, 90 – 99.99% is slightly behind pace, and anything less than 90% is behind pace:

[Insert Image 66-9]



This pace score is then dragged to the Color Marks Card to color each bar by its progress to goal classification:

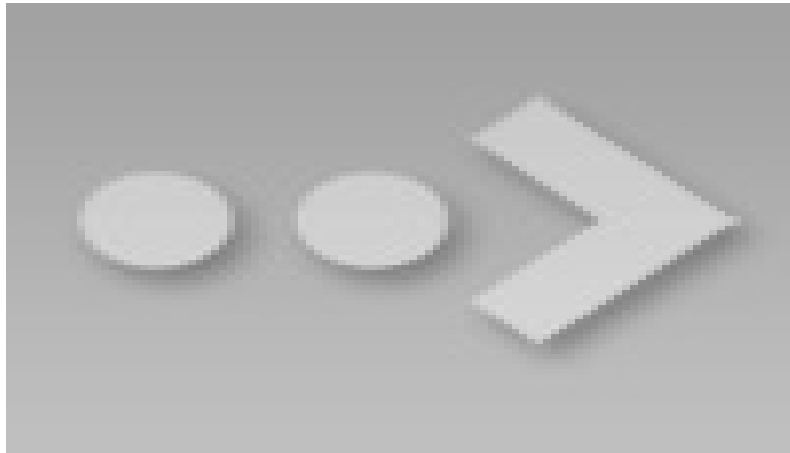
[Insert Image 66-10]



In this tutorial, we used a linear pace that was calculated by taking $\frac{1}{52}$ of the year multiplied by the current week of the year. However, this pace can be replaced with a different metric such as the value for each KPI at this point last year, or a goal for each respective KPI at this point in the year. Here is one more example using a different expected pace for each respective KPI. In this example, I have the expected pace at this point in the year as an additional field

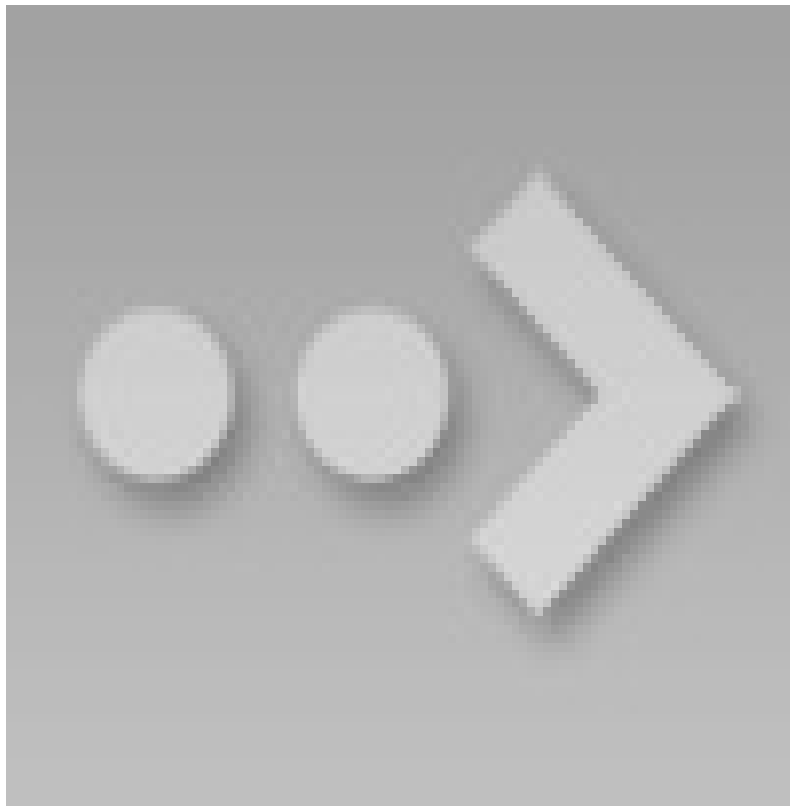
in my underlying data:

[Insert Image 66-11]



The seasonal pace chart using this data looks like this in Tableau:

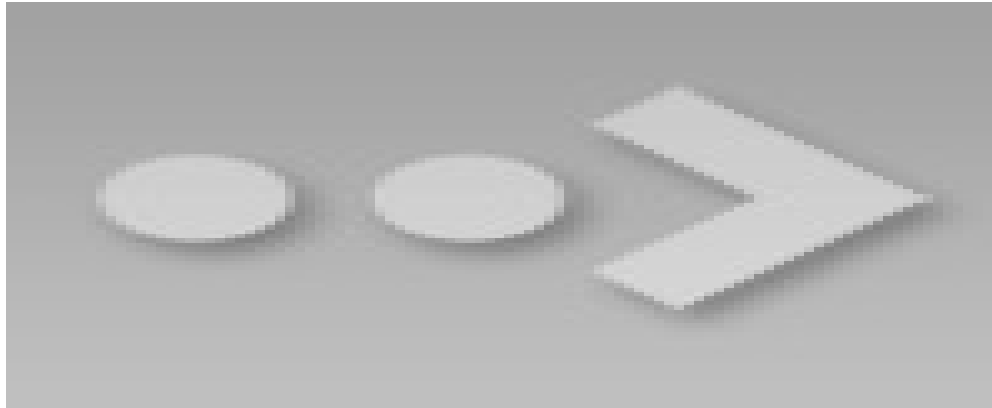
[Insert Image 66-12]



To create this version, I replaced the calculated linear pace reference line with a reference line for expected pace, which shows the expected pace for each

reference line for expected pace, which shows the expected pace for each respective KPI. I also replaced the linear pace calculation in the Pace Score calculated field with the Expected Pace measure from the underlying data:

[Insert Image 66-13]



Notice how this seasonal pace chart tells a different story regarding the progress to goal for each KPI than the pace chart with the linear pace. But in both cases, pace charts were used to normalize progress to goal calculations across KPIs to get a more effective visualization about the business.

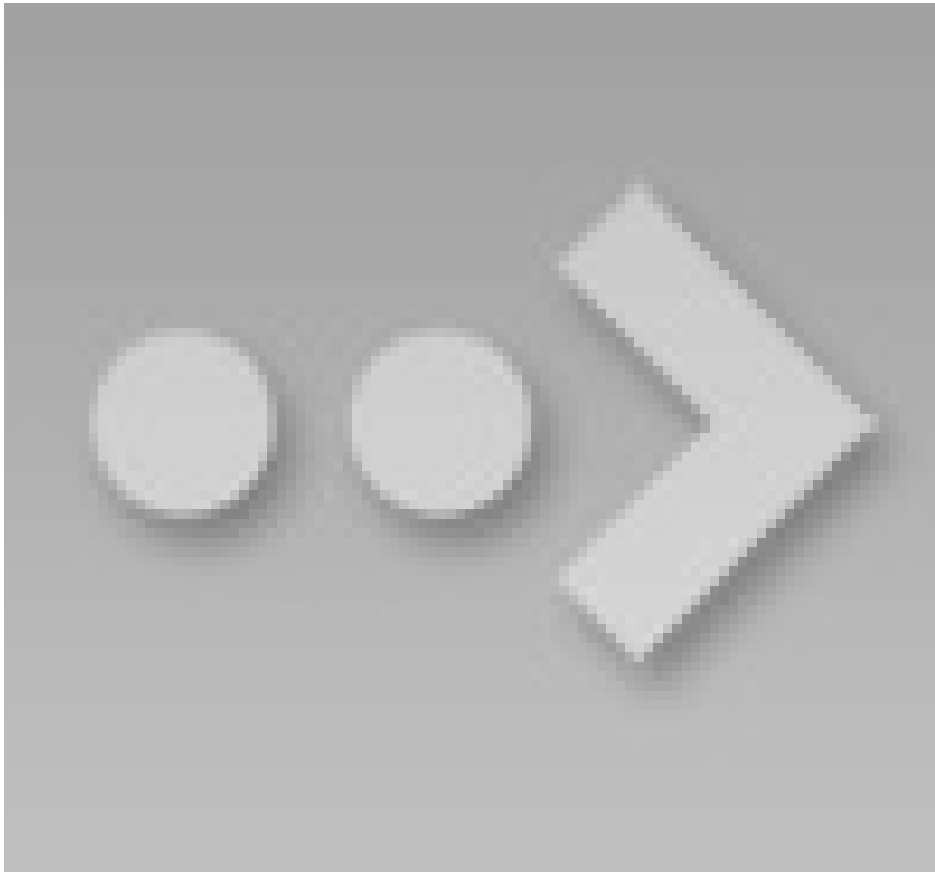
Chapter 67 – How to Make a Pareto Chart

A Pareto Chart, named for Vilfredo Pareto, is a dual-axis combination chart used to highlight dimension members that are having the biggest impact to the measure in question. On the primary axis, bars are used to show the raw quantities for each dimension member, sorted in descending order. On the secondary axis, a line graph is used to show the cumulative total in percent format. While this chart type can serve a variety of purposes, it is part of the seven basic tools of quality control, and is traditionally used to identify the biggest opportunities for improvement.

How to make a Pareto chart in Tableau

In sticking with a quality control scenario, this tutorial will use the Sample – Superstore dataset to look at which product sub-categories contribute the most returned items. The final chart will look like this:

[Insert Image 67-1]

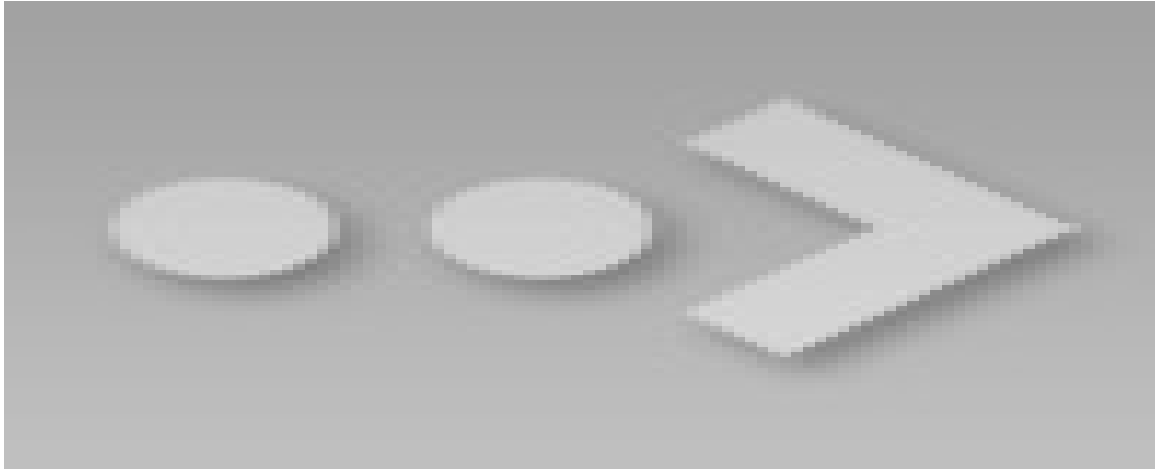


As with the rest of *Practical Tableau*, you can follow along using the Sample – Superstore data that comes with every download of Tableau. However, this tutorial required me to take two additional steps to prepare the data.

1. Left join the Returns table to the Orders table

In do this, right-click on the Sample Superstore data connection and choose “Edit data source...”. A new interface will appear where you can drag the Returns table next to the Orders table and set up a left join on Order ID.

[Insert Image 67-2]



2. *Create a calculated field to count the number of returns*

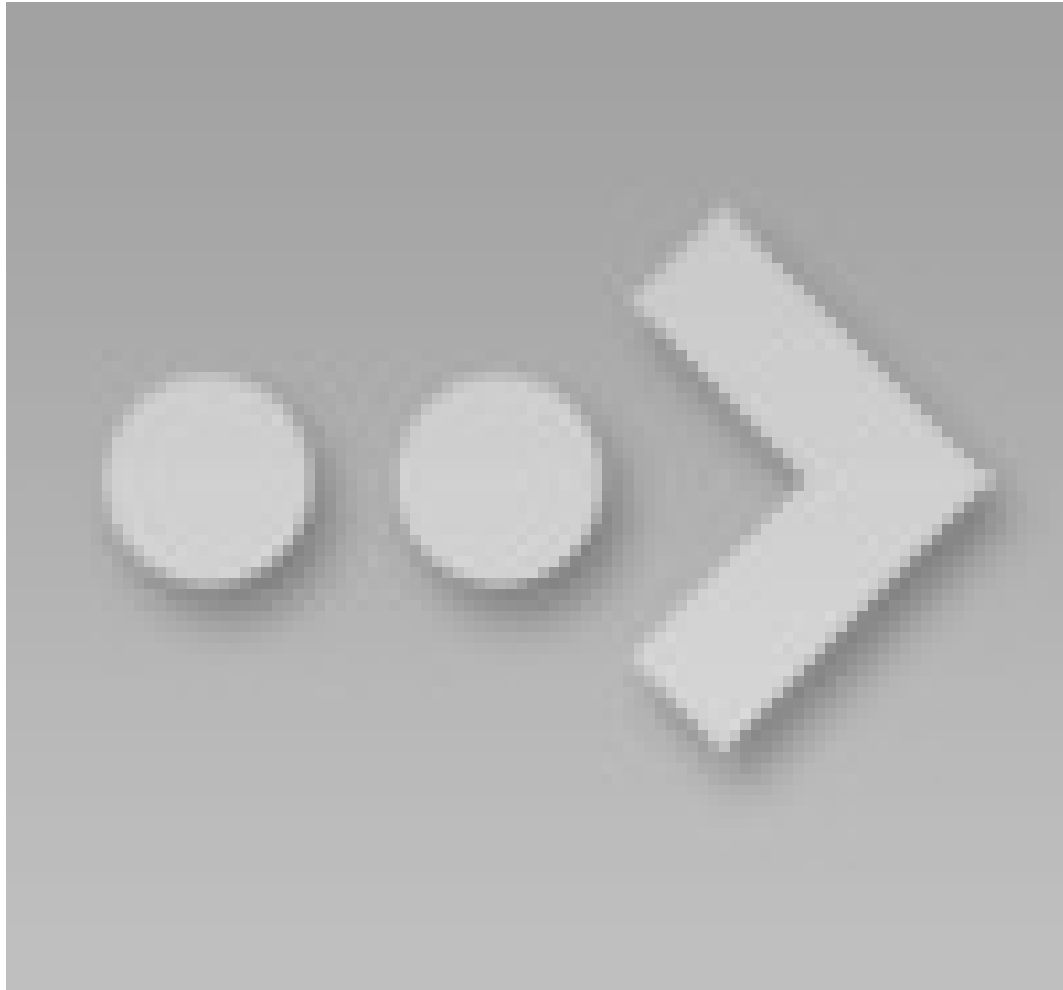
The formula for this calculation is `COUNT([Returned] = "Yes")`

[Insert Image 67-3]



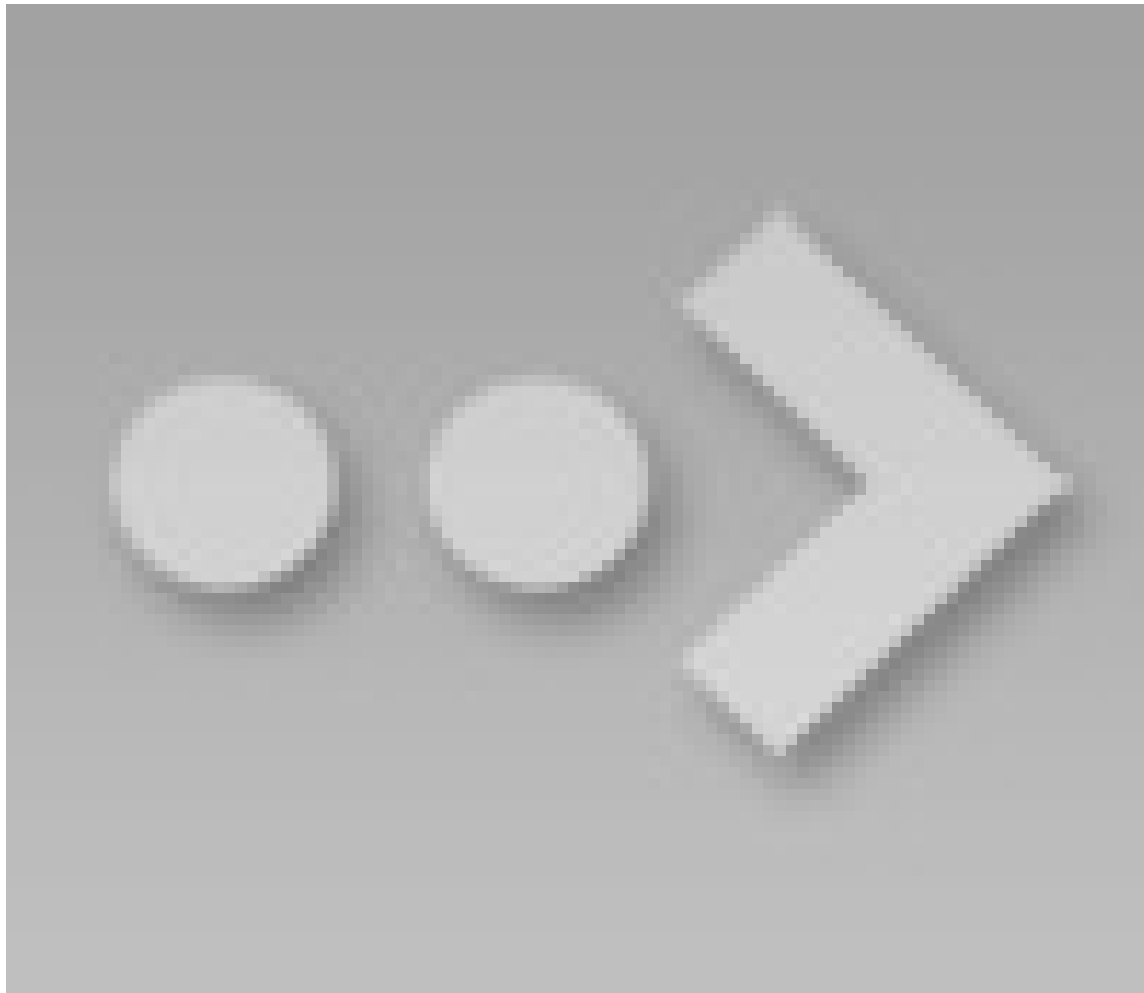
We are now ready to build out the Pareto chart. First, create a bar chart that looks at the number of returns per Product Sub-Category and sort it in descending order:

[Insert Image 67-4]



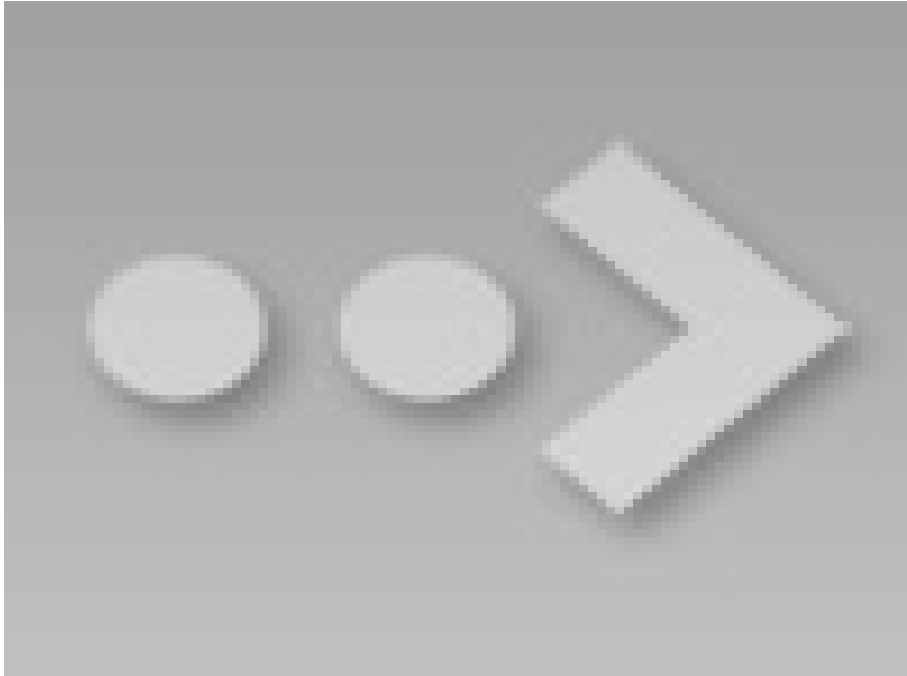
Next, create a dual-axis combination chart by dragging the Returns measure from the Measure Shelf to the right axis, changing the mark type on the primary axis back to bar, and changing the mark type for the secondary axis to line. For a refresher on creating this type of chart, see [How to Make a Dual-Axis Combo Chart in Tableau](#). At this point, the visualization will look similar to this:

[Insert Image 67-5]



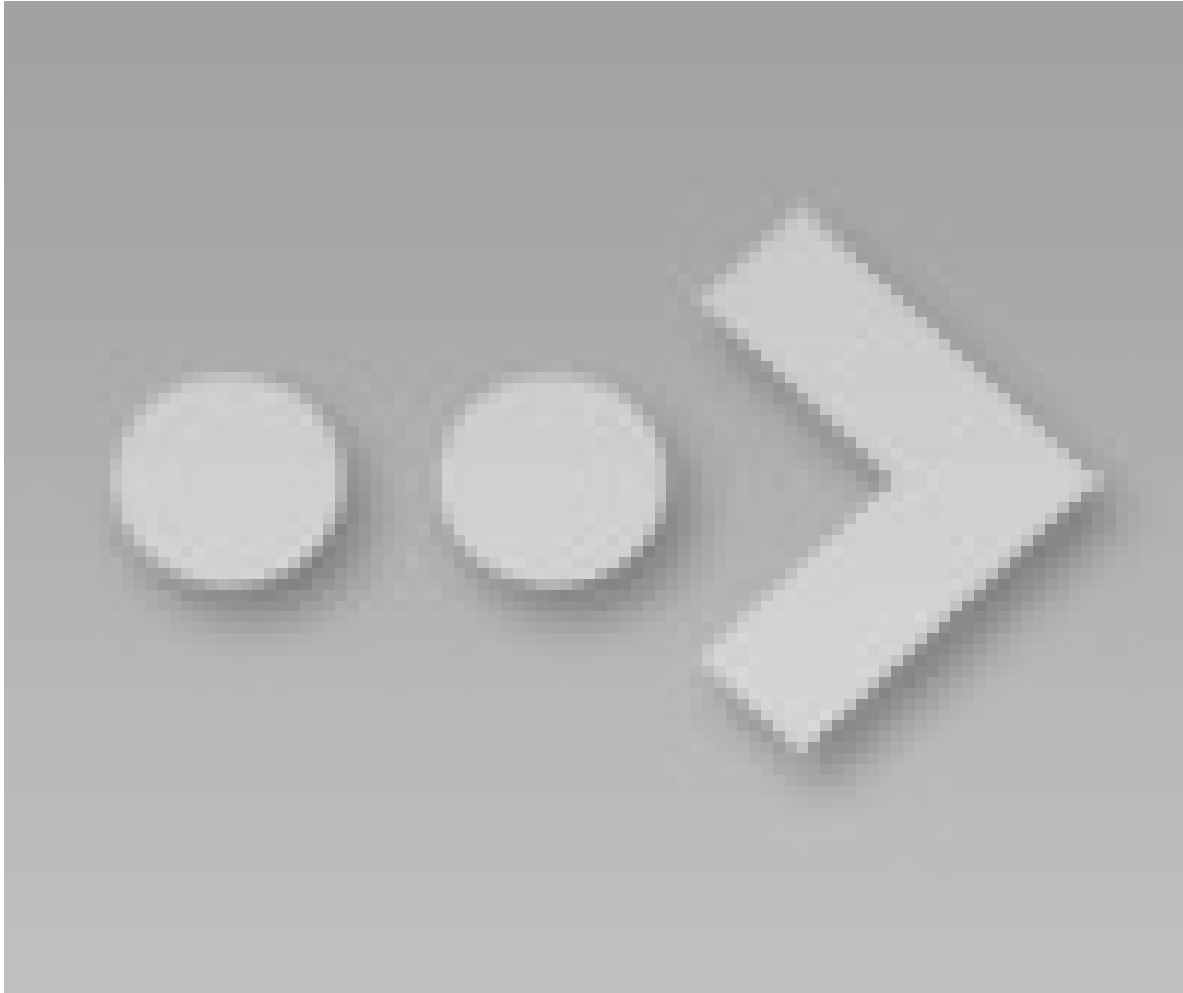
Now the step that makes this a Pareto chart. We will add a table calculation and a secondary table calculation to the second Returns pill to display the cumulative percent of returns across product sub-categories. To add the first table calculation, click on the second Returns pill, hover over “Quick table calculation”, and choose “Running total”. With the table calculation for running total, you can add a second table calculation on the result. This is how we can calculate the raw running total number at each product sub-category, then add a secondary calculation to determine the cumulative percent of total. To add a secondary table calculation, click the second Returns pill again, choose “Edit table calculation...”, and check the box at the bottom of the interface that says “Add Secondary Calculation”. Changing the Secondary Calculation Type to Percent of Total will display the cumulative percent of total.

[Insert Image 67-6]



After changing the axis tick marks for a cleaner look, I am left with this Pareto Chart:

[Insert Image 67-7]



This chart can now be used for insights such as, “The business’ three most returned product sub-categories are causing about 40% of the total returns.” Pareto charts are an effective way to quickly highlight opportunities for improvement and provide a scale for how urgently a quality control problem should be treated.

Chapter 68 – How to Make a Control Chart

Control charts, or Shewhart charts, were designed to evaluate whether production is in a state of statistical control. Along with Pareto charts, histograms, and scatter plots, they are one of the seven basic tools for quality control. Basic control charts show a time-series analysis with reference

lines that show the average performance along with an upper control limit, typically three standard deviations above the mean, and lower control limit, typically three standard deviations below the mean.

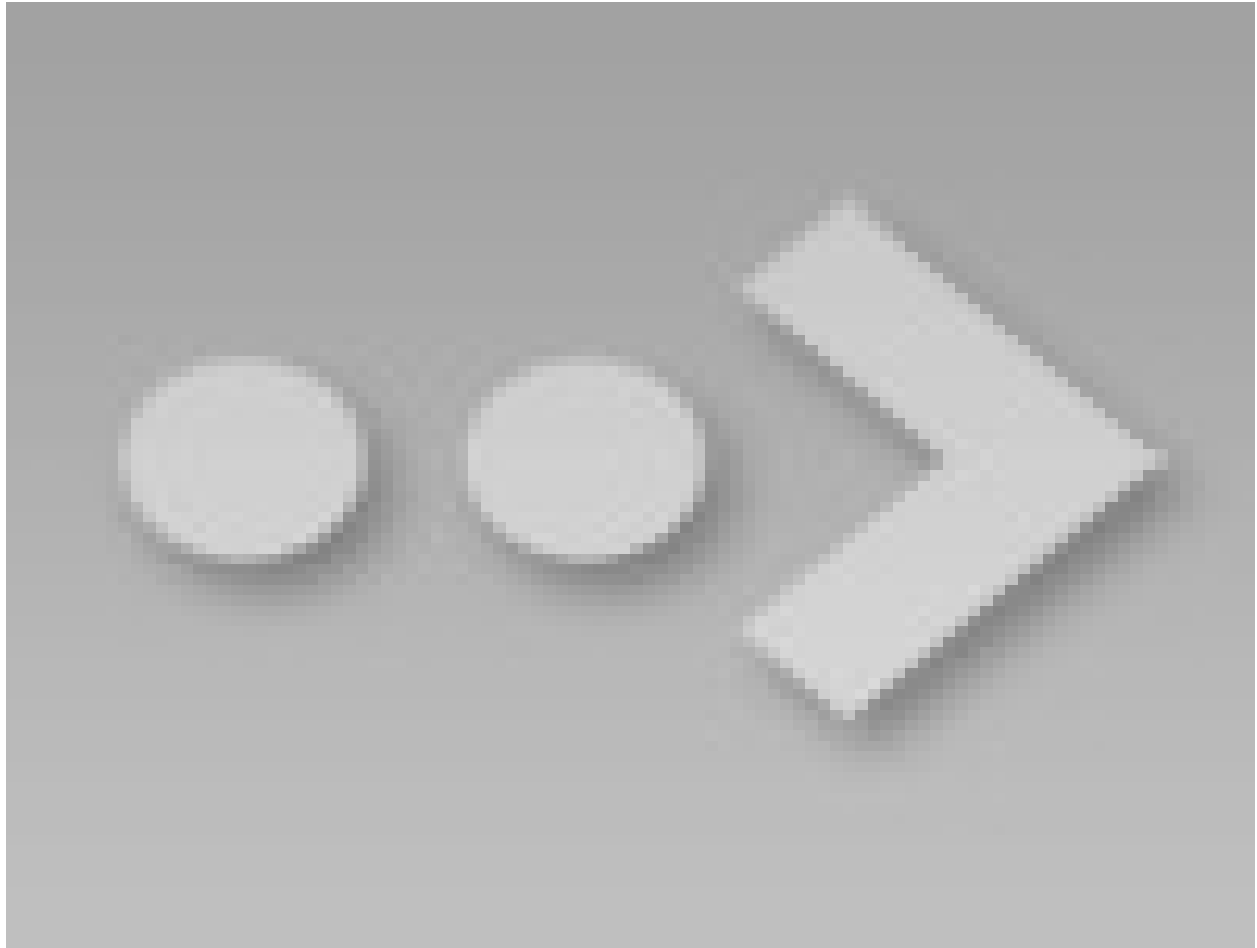
Control charts were originally called Shewhart charts, named for Walter Shewhart, who invented the visualization type while working on a way to improve the reliability of the telephony transmission systems at Bell Labs in the 1920's. Control charts were born in, and are still primarily used for, evaluating manufacturing processes. For example, a factory may use this type of analysis to predict production levels and have a statistical indicator for when corrective action needs to be taken.

How to make control charts in Tableau

While designed for manufacturing, I find control charts to be useful for any measure in which historical data can be used to predict performance and provide visual cues for when a statistically significant change is occurring in the business. In that vein, and so we can all follow along using the same data, this tutorial will create a basic control chart which evaluates the Profit Ratio measure in the Sample Superstore dataset.

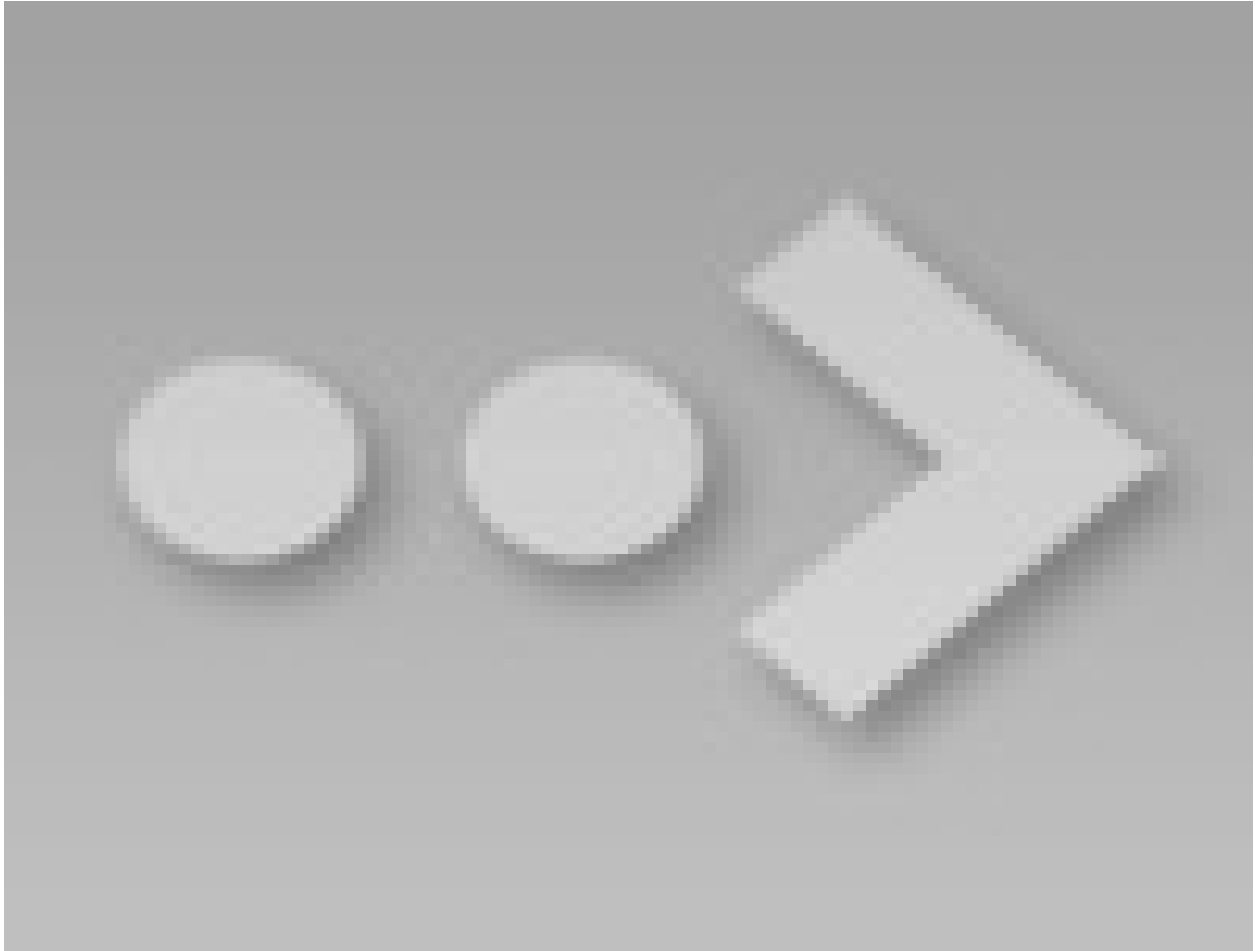
To create a control chart, start by creating a continuous line graph that looks at Profit Ratio by week:

[Insert Image 68-1]



It may be concerning that we see several weeks with a negative profit ratio, but based on our historical data, is it a *statistically significant* cause for concern? The answer comes through control charts. Control charts start with a line that shows the mean performance for all of the data points. This is easily added in Tableau as a reference line. To add the reference line, right-click on the axis for Profit Ratio, choose “Add reference line”, and click the “OK” button. Here’s how the view looks at this point after removing the label on the reference line and formatting the line for color and weight:

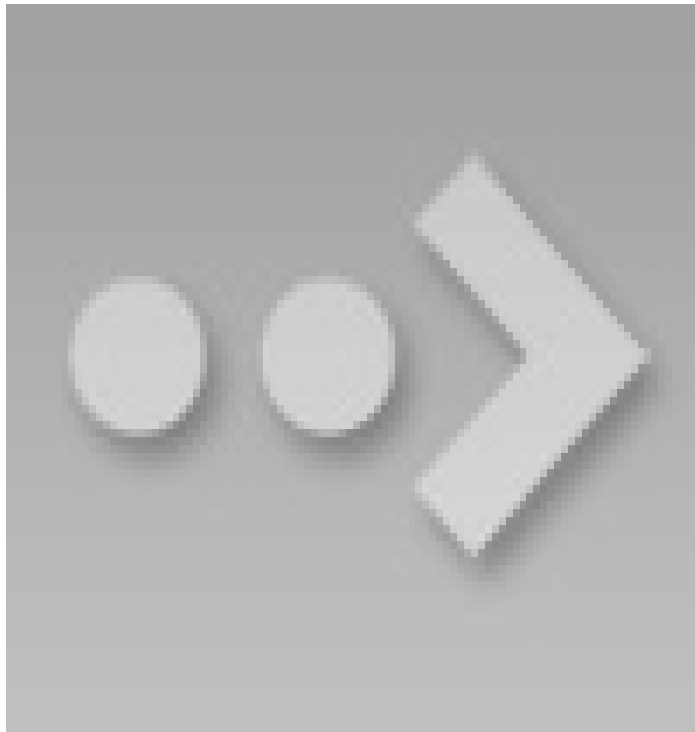
[Insert Image 68-2]



To finalize the control chart, we need to add the upper and lower control limits, which are typically three standard deviations from the mean. This step will also be accomplished through reference lines in Tableau. More specifically, a reference distribution will be used that draws two lines on the view, one for the upper control limit, and one for the lower control limit. To do this, follow these steps:

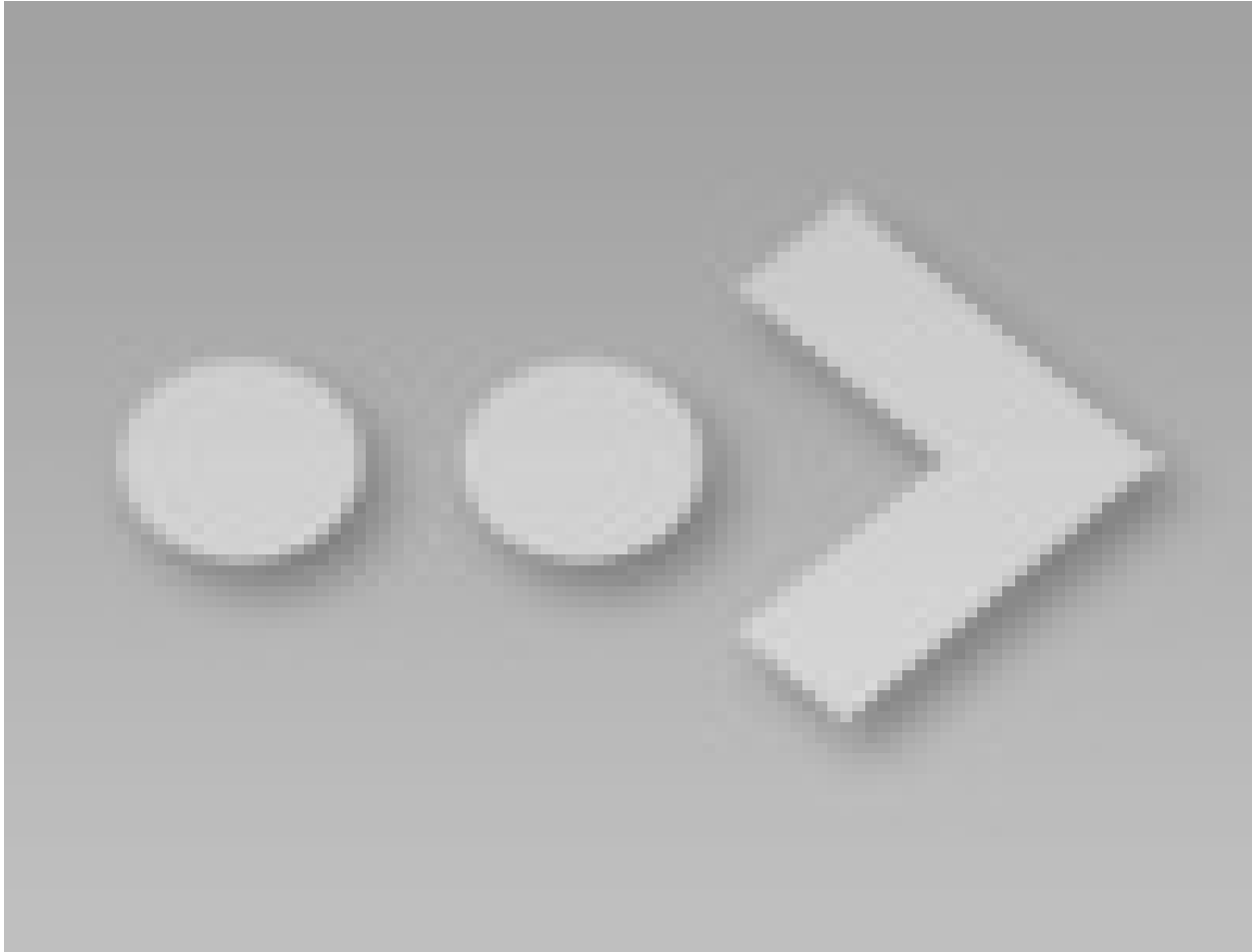
- Right-click on the axis for Profit Ratio and choose “Add reference line”
- In the reference line options listed at the top of the dialog box, choose “Distribution”
- In the dropdown for the distribution value, select the “Standard Deviation” option
- Change the factors to -3 and 3

[Insert Image 68-3]



Here's how my final view looks after removing the line labels and doing some formatting to the reference distribution:

[Insert Image 68-4]



This final control chart shows not only the average profit ratio during the weeks in our analysis, but that the business can expect profit ratio to fall between -25% and 48%. In the current example, there were four weeks that fell outside of the lower control limit and deserve further investigation.

For more on this chart type, see Ben Jones' Tableau Public post, [How to Make Control Charts in Tableau](#).

Chapter 69 – How to Make Dynamic Dual-Axis Bump Charts

Bump charts are an effective way to show how dimension members are ranking across different measures over time. For example, you may want to see how specific product categories have ranked in sales for your company from year to year. Or you may want to see how different months have ranked in terms of time to see if

year. Or maybe you want to use discrete months as your element of time to see if the rankings for product categories change based on seasonality. Or maybe you want to do something outside of work and see how your fantasy football players are ranking across different statistics from week to week!

Whatever the case may be, I've found bump charts to be an engaging and easy to interpret means for visualizing ranks over time. Fellow Tableau Zen Master, Matt Chambers, created one of the most popular Tableau Public visualizations of the year with a bump chart and provided his technique at his site, [Sir Viz-a-Lot](#).

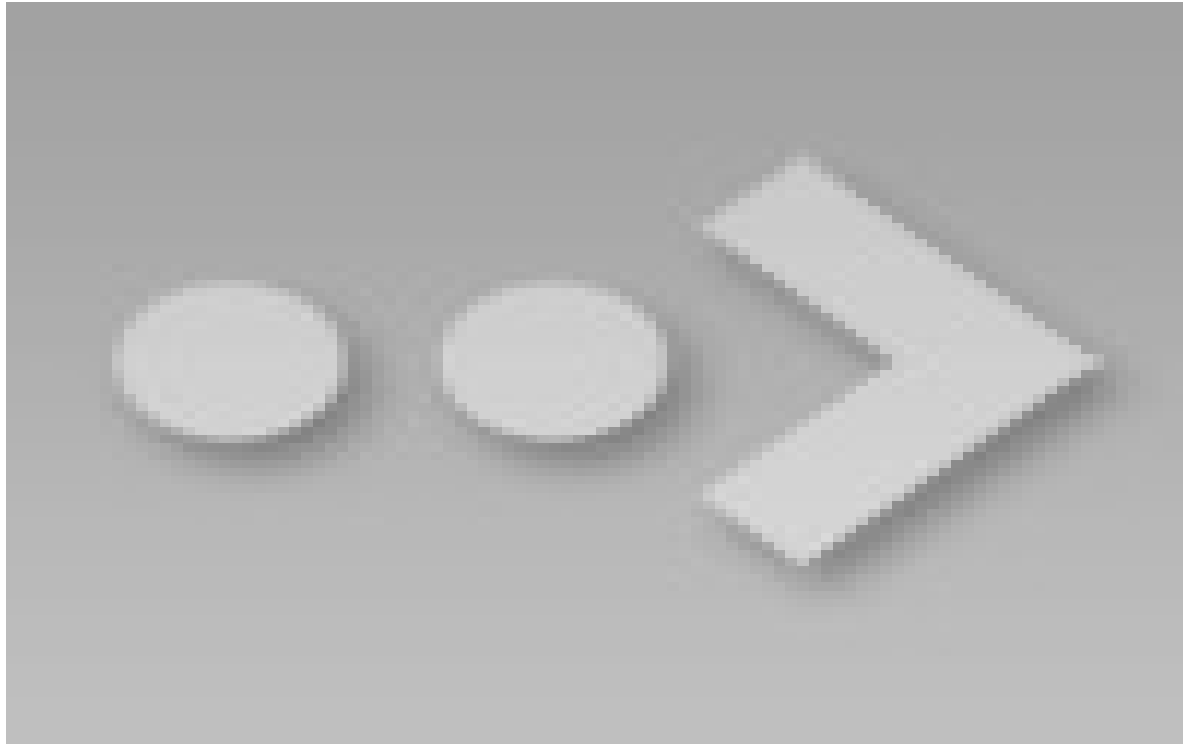
In addition to being an engaging and easy to understand visualization, this chart type is very easy to make in Tableau.

This tutorial will show you how to make bump charts in Tableau, but also (1) how to leverage a dual-axis to provide additional context on the view and (2) how to allow the end user to choose the measure being ranked and the dimension members being compared.

How to make bump charts in Tableau

For the first part of the tutorial, we will recreate this bump chart that shows us how the segments from the Sample – Superstore dataset rank between January and December for the SUM(Sales) measure.

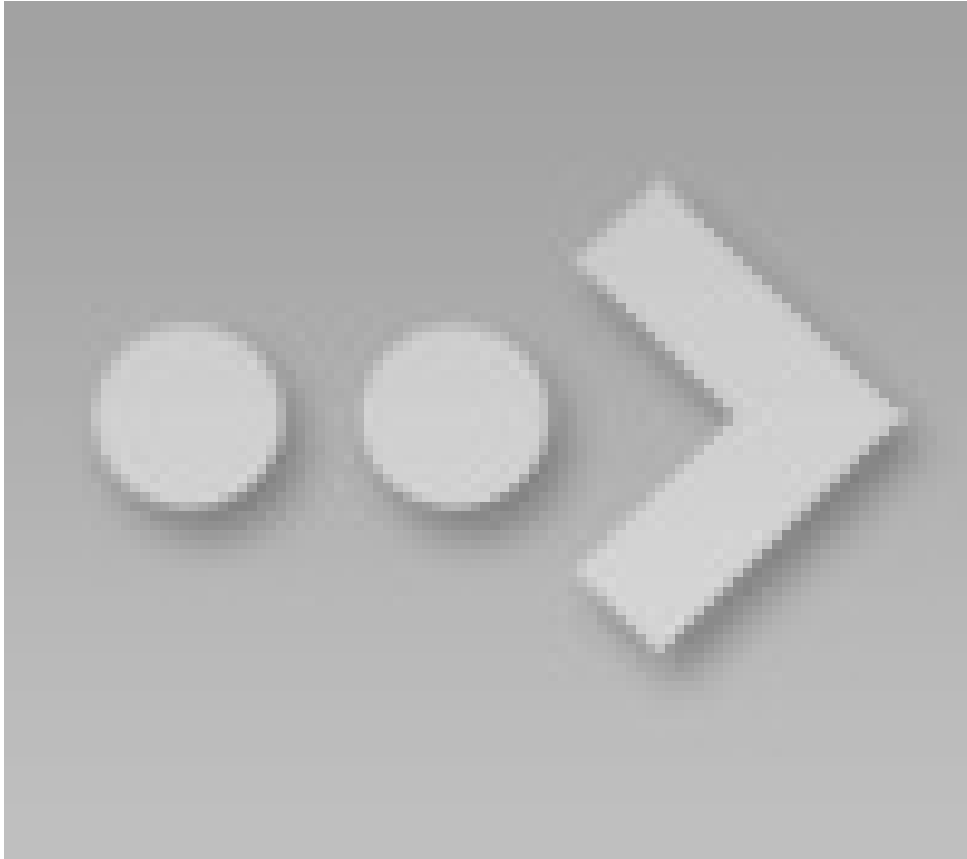
[Insert Image 69-1]



To create a bump chart, start by placing the element of time that you want to evaluate onto the Columns Shelf. To recreate the bump chart pictured, I placed discrete MONTH(Order Date) onto the Columns Shelf.

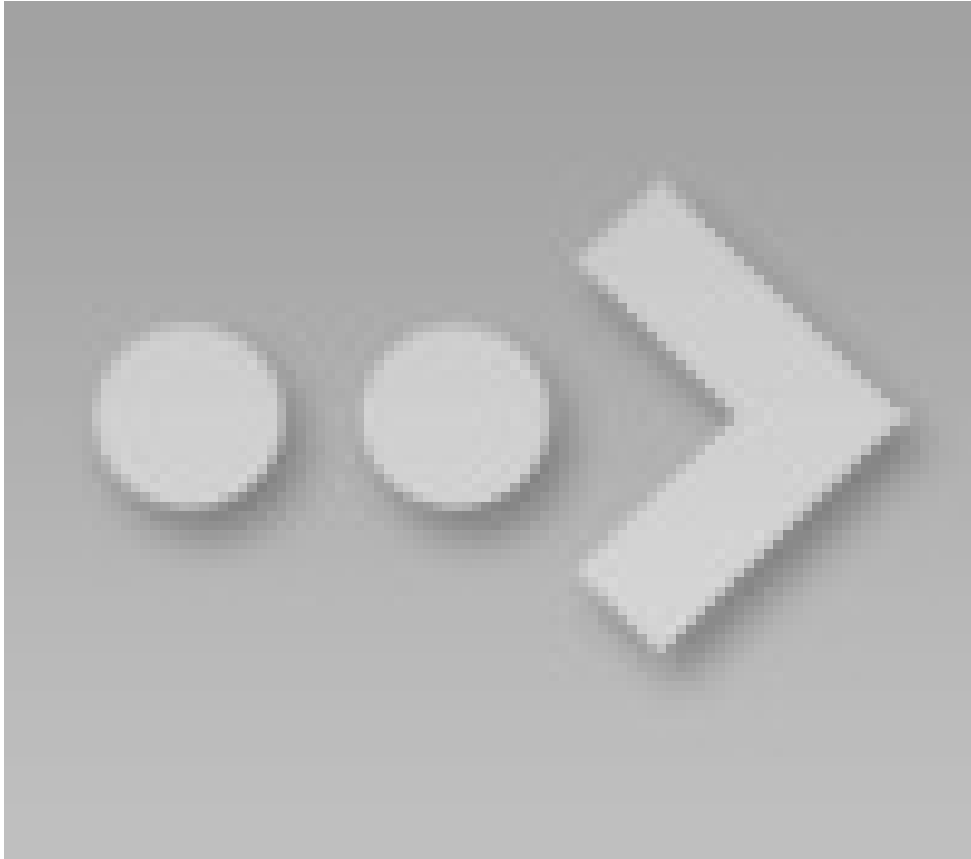
On the Measures Shelf, I will place a calculated field for the measure that we want to rank; in this case Sales. The formula for my calculated field is RANK(SUM(Sales)). At this point, the view looks like this:

[Insert Image 69-2]



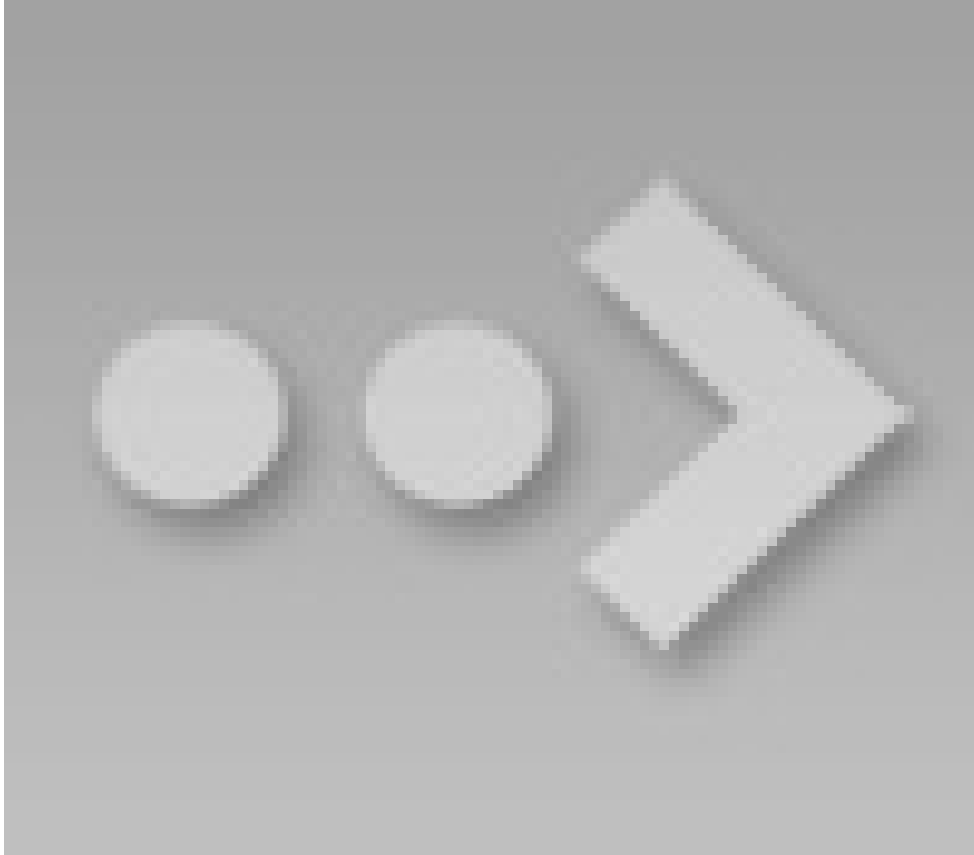
The RANK function acts as a table calculation in Tableau, so when it was added to the view, it was computed across the table from left to right. To rank the sales amounts by the Segment dimension, let's make the level of detail in the view more granular by dragging the Segment dimension to the Color Marks Card.

[Insert Image 69-3]



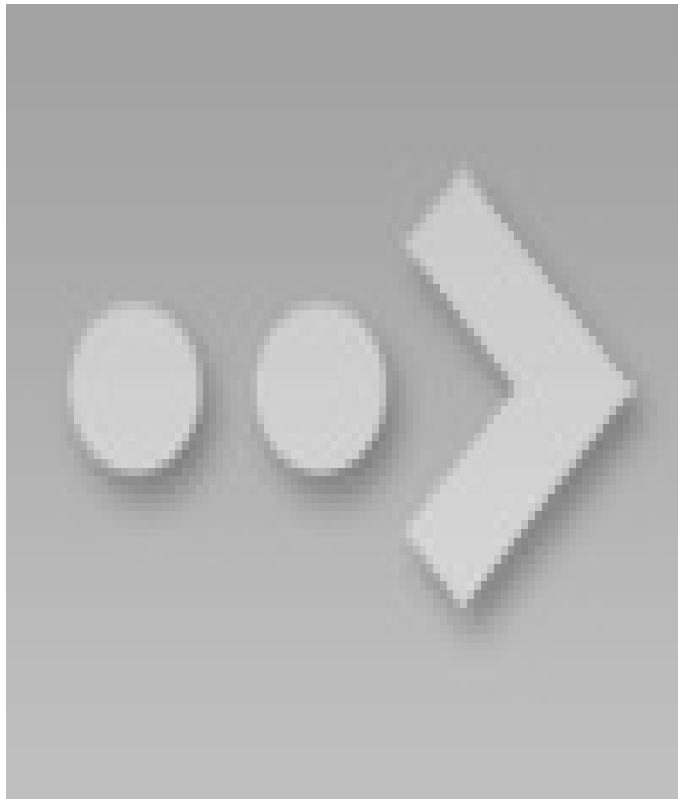
That got us closer, but we still need to change the RANK table calculation to compute for each segment. The Delta symbol on the Sales Rank measure tells us that a table calculation is happening. To edit how the table calculation is being computed, right-click on the Sales Rank measure, hover over “Compute Using”, and choose “Segment”.

[Insert Image 69-4]



At this point, we’ve technically got a bump chart in Tableau. It may be more intuitive for the best rank (i.e. 1) to be on top, so I will reverse the axis by right clicking on the Sales Rank axis, clicking “Edit Axis...”, and checking the box to reverse the scale.

[Insert Image 69-5]

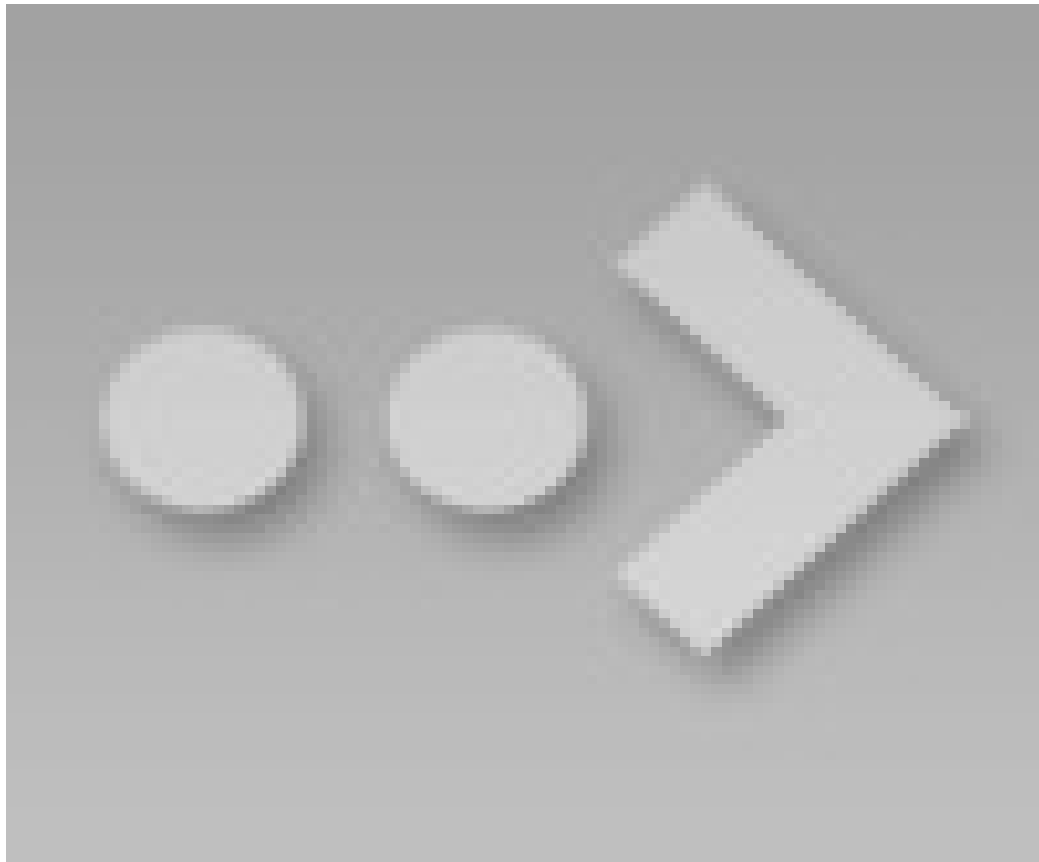


We have a bump chart, but wouldn't it be great to provide some additional value to the analysis? It's good to know that the consumer segment led the way for the first nine months of the year, but we don't know if the sales for that segment were going up or down throughout the year. To provide that context I will create a dual-axis combination chart. The first axis will be the bump chart that we just set up, and the second axis will be a dot plot that displays the percent change in month over month sales.

To make this bump chart dual-axis, duplicate the Sales Rank measure and place it on the opposite axis and right-click on the right axis to click "Synchronize Axis". If working with a second axis sounds new to you, be sure to read [How to Make a Dual-Axis Combination Chart in Tableau](#) before moving forward.

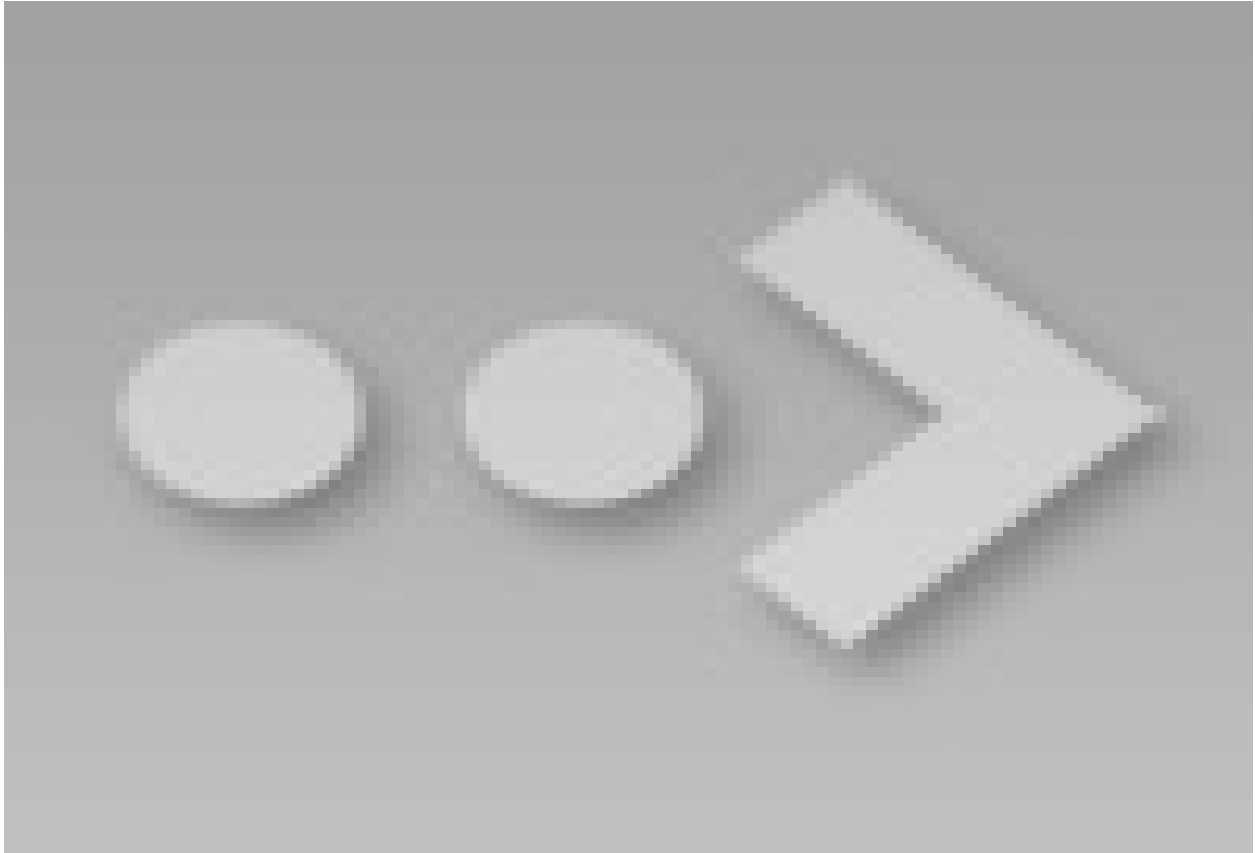
At this point, we have the same line graph twice, with one laying on top of the other. Now that we duplicated the Sales Rank measure, we have two separate sets of Marks Cards, and they can be edited independently. I will change the mark type on the second Marks Shelf to circle to make my bump chart look a little bit nicer.

[Insert Image 69-6]



I now have some real estate to add some additional context to the marks. One possibility is to display the rank (1 through 3) for the mark on each circle. That may be a good use if you're dealing with many separate rows of marks, but with only three, that seems like a waste of valuable real estate. Instead, I will add the Sales measure to the Label Marks Card (for the circles only), then add a table calculation that computes the month over month percent change in sales for each segment. I also hid the right axis to finalize my bump chart.

[Insert Image 69-7]

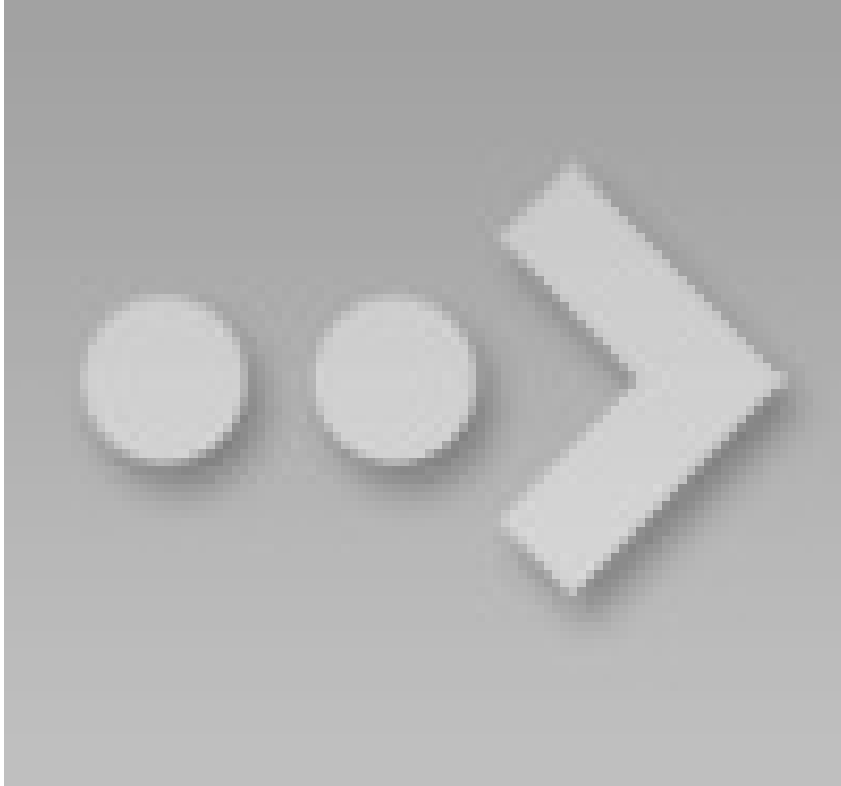


For more on table calculations, see the chapter *An Introduction to Tableau Table Calculations*.

Our Tableau bump chart is now showing the month over month rank per segment, but also the month over month percent change in sales for each segment. This is turning out to be a solid static analysis, but why don't we add even more value in Tableau by allowing our end user to choose what dimension is used in the sales ranking.

To accomplish this, I will use this trick to allow end users to choose measures and dimensions in Tableau. For this example, I will allow my end users to choose from the Segment, Ship Mode, or Category dimensions. The first step is to create a string parameter with the choices of Segment, Ship Mode, and Category.

[Insert Image 69-8]



Then I will create a calculated field to give Tableau instructions on what to display for each parameter choice.

[Insert Image 69-9]



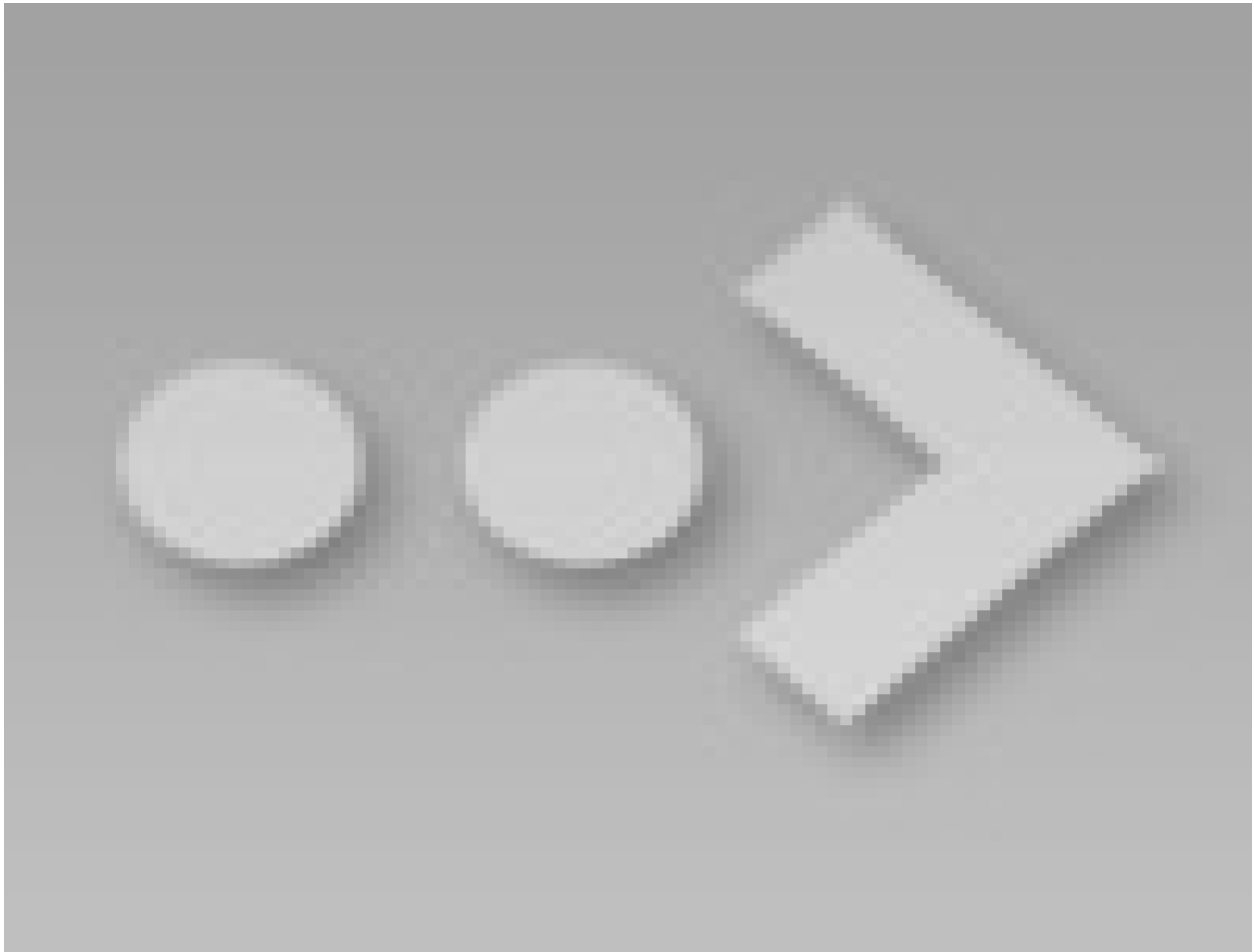
Now if I replace the Segment dimension on the color marks cards of my bump chart with my newly created Dimension Choice field, the lines and circles will be colored based on what my end user selects in the parameter.

In addition to placing the Dimension Choice dimension on the color marks card for both axes, there are two more small steps to get this working: 1. Ensure the table calculations on the Rows Shelf are set to compute on Dimension Choice

table calculations on the Rows shelf are now computing on Dimension Choice (they were previously computing on Segment) and 2. Show the parameter control for Dimension Choice so the end user can change the selection (to do this, right-click on the parameter and choose “Show Parameter Control”).

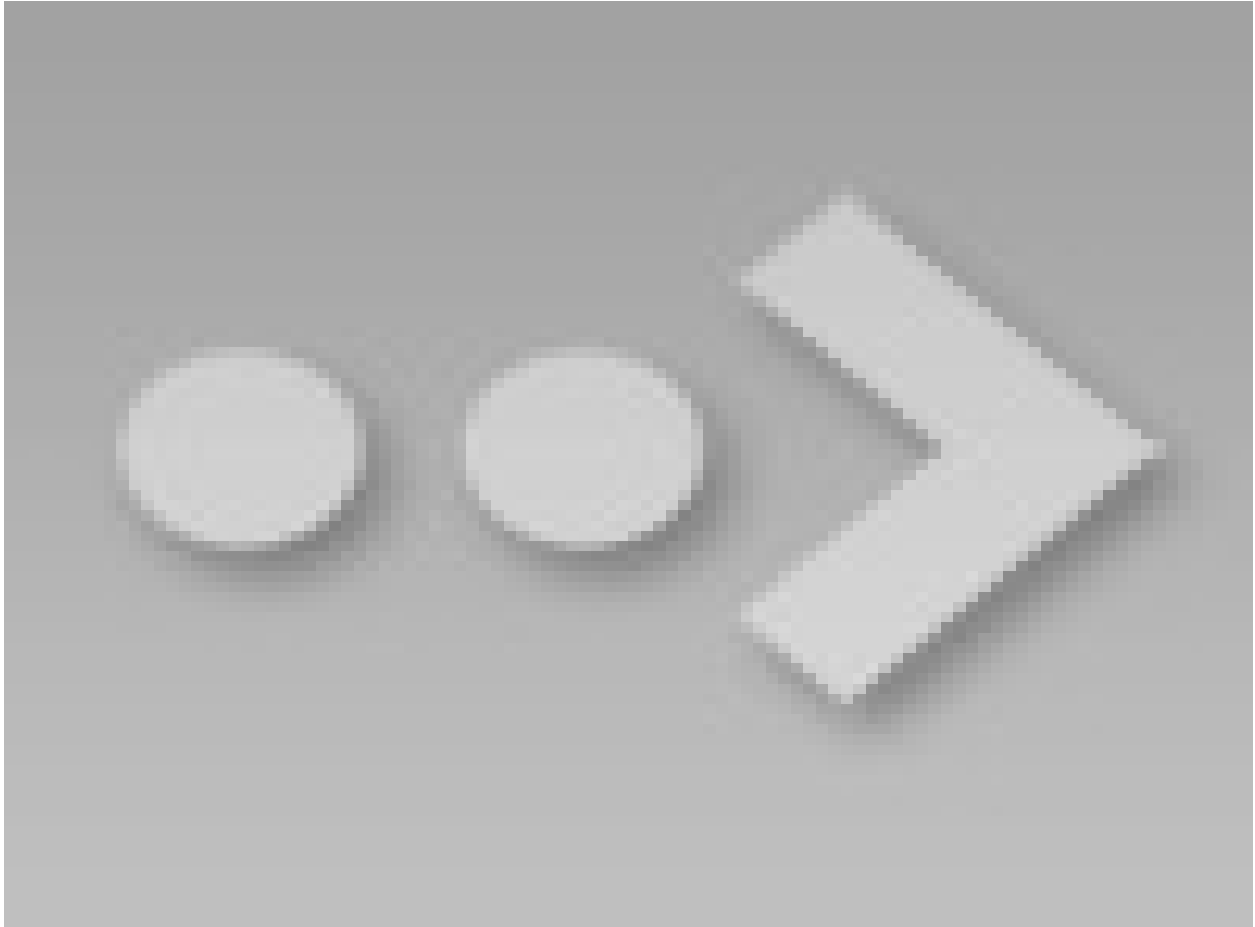
Now if I select Ship Mode in the parameter control, my bump chart rankings are based on Ship Mode.

[Insert Image 69-10]



If I choose Category, my bump chart rankings are based on Category.

[Insert Image 69-11]



A dynamic, dual-axis bump chart that is providing multiple layers of insight for our end users, and allows them to control the visualization. Your wheels may already be turning at the possibilities for your own analyses, and yes, you can also use the parameter selection approach to change the measure for the ranking!

Chapter 70 – How to Make Dumbbell Charts

Tableau dumbbell charts, also known as DNA charts, are an alternative visualization choice for illustrating the change between two data points. Dumbbell charts get their slang name from their appearance, which look similar to weights, and sometimes strands of DNA, when they are in a horizontal orientation.

I personally love that there is an outside-of-the-box chart type that isn't named after a delicious food such as donuts or waffles. You may even say that dumbbell charts are a healthy alternative...

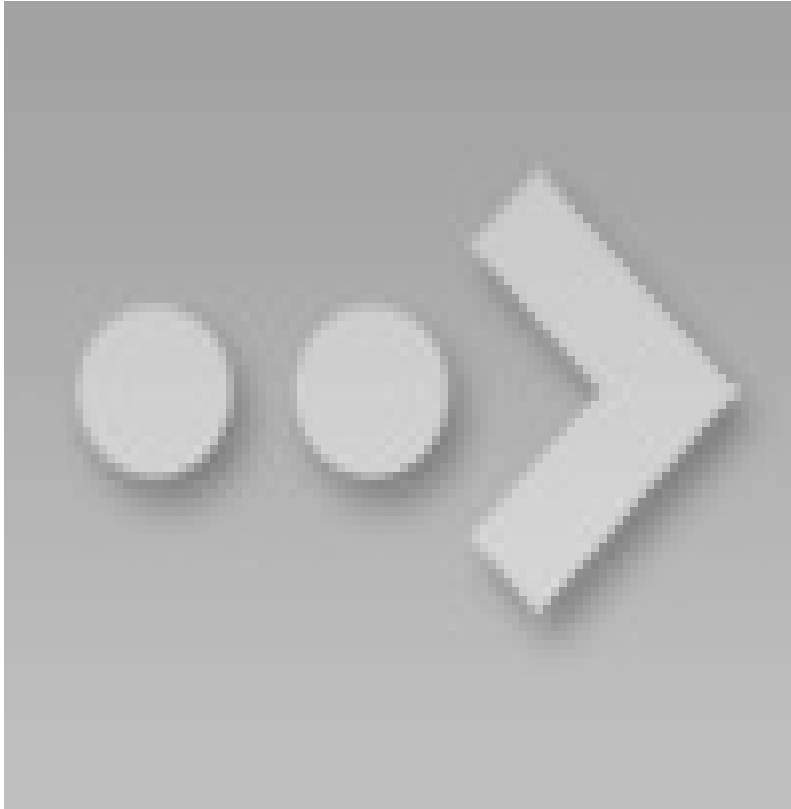
Names aside, this is another chart type that I find to be engaging, effective, and relatively easy to create in Tableau. This tutorial will illustrate how to make a dumbbell chart in Tableau in just a few steps.

Tableau dumbbell charts are actually dual-axis combination charts, where one of the axes have a mark type of circle and the other has a mark type of line. If you are unfamiliar with using two axes in Tableau, it may be helpful to read [How to Make a Dual-Axis Combination Chart in Tableau](#) before reading further.

How to make Tableau dumbbell charts

For this tutorial, we will recreate this visualization which compares the year over year sales per sub-category in the Sample-Superstore dataset.

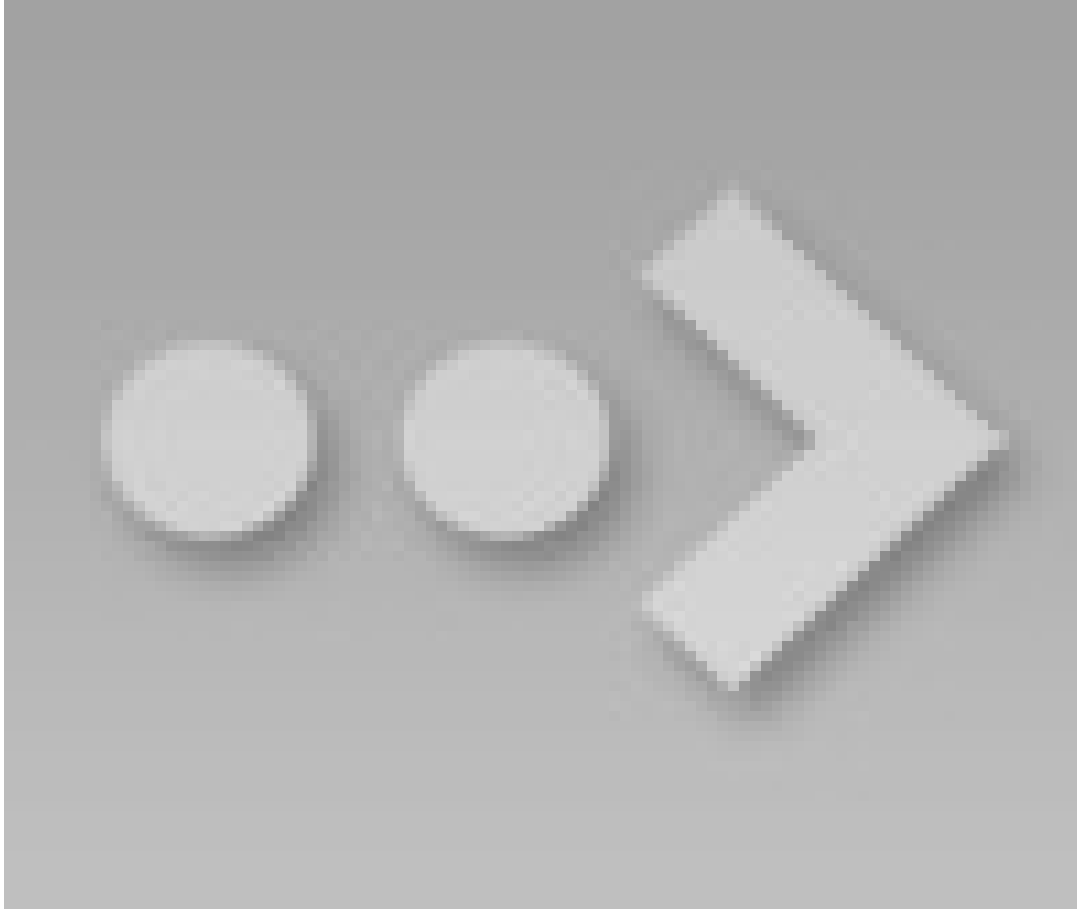
[Insert Image 70-1]



The first step to creating a dumbbell chart in Tableau is to create a dot plot with the measure and dimension you want to visualize. In this example, I will place the Sales measure from the Sample – Superstore dataset onto the Columns Shelf and the Sub-Category dimension onto the Rows Shelf. This creates a sales by sub-category bar chart that can easily be converted to a dot plot by changing the mark type from automatic to circle.

I will also filter the visualization to the last two years in the dataset so that I have only two comparison points per sub-category, and also color the marks by year to distinguish which year is which. At this point, my view looks like this:

[Insert Image 70-2]

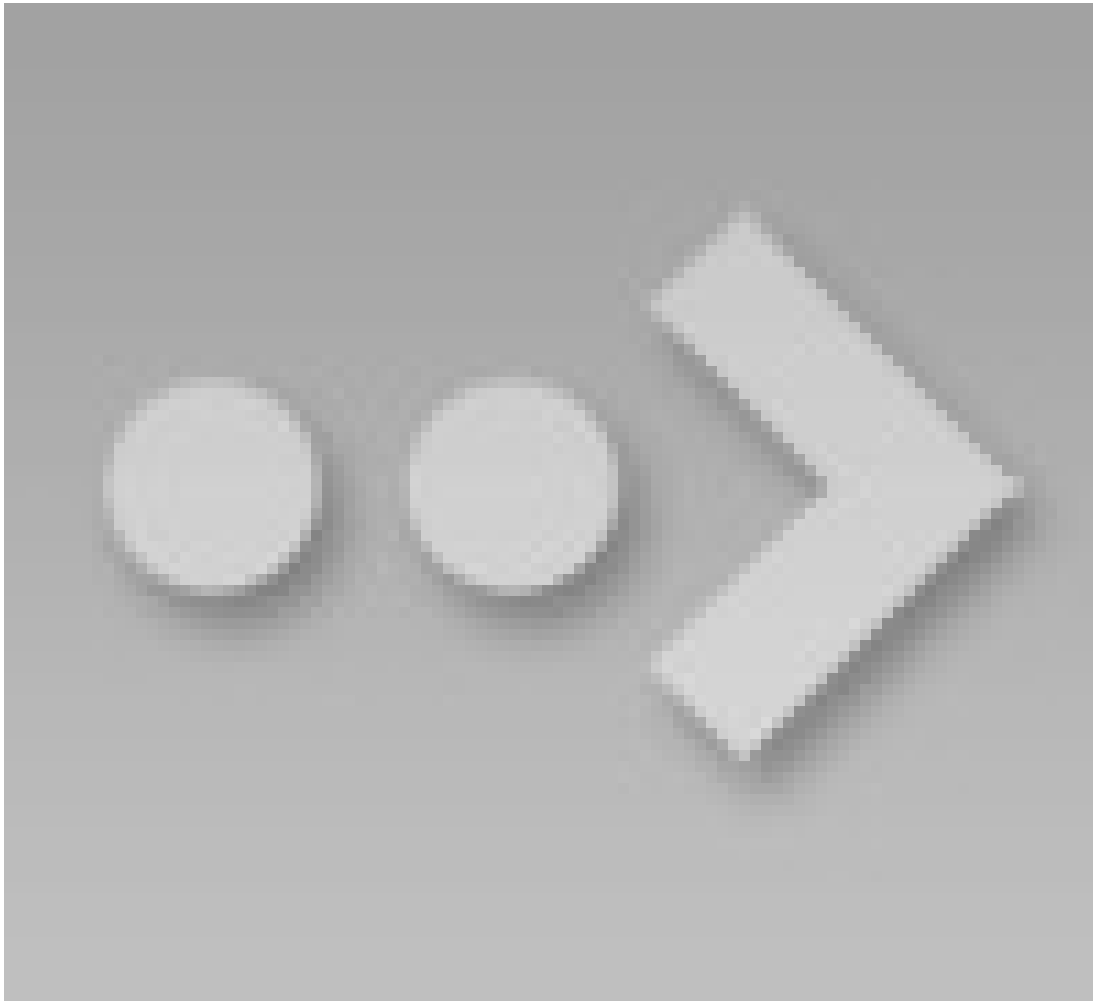


Note that this chart type can also be created with a vertical orientation by swapping the location of the fields on the rows and columns shelves, but we will stick with the horizontal orientation for this tutorial.

The second, and actually final, step is to create a second axis with the Sales measure and change its mark type to line. To turn this dot plot into a dual-axis combination chart, drag the Sales measure near the top of the chart, directly across from the sales axis on the bottom of the chart; when a dashed line appears, drop the measure on the view. Alternatively, you can place a second occurrence of the Sales measure onto the Columns Shelf, right-click on the second pill, and choose “Dual Axis”. Ensure the axes always line up by right-clicking on the top axis and choosing “Synchronize Axis”.

At this point, you have two sets of marks cards (one for each occurrence of the Sales measure), and they can be edited independently. Navigate to the second set of marks cards, change the mark type to line, then drag the YEAR(Order Date) dimension on the marks shelf to the Path Marks Card.

[Insert Image 70-3]



That’s all it takes to get to a nice looking dumbbell chart in Tableau. From here, you can format the size and color of the marks and hide the top axis by right-clicking on it and deselecting “Show Header”.

Chapter 71 – How and Why to Make Customizable Jitter Plots

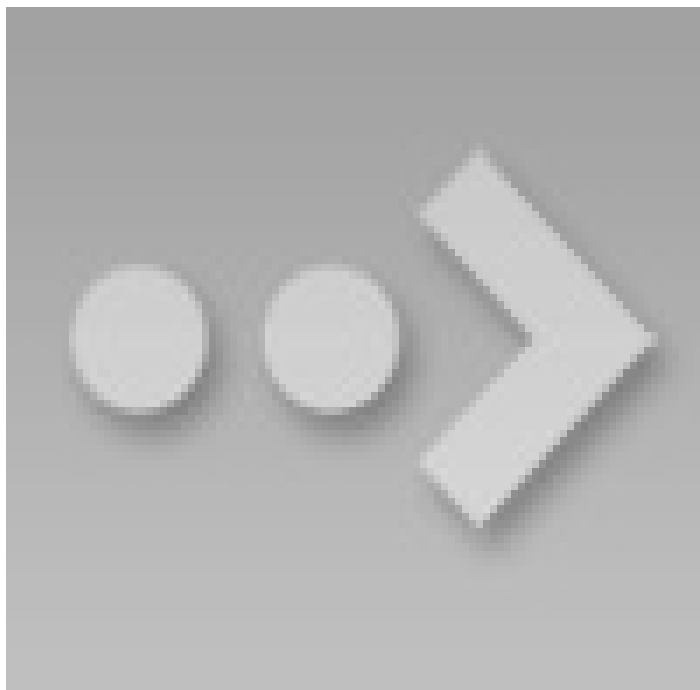
“Jittering” is a technique for separating overlapping marks on a view. By giving marks some extra room by separating them into different columns, hidden data is often revealed and it is easier to visualize how the data points are distributed. The bad news is that Tableau does not provide an out-of-the-box option to jitter

data points. The good news is that Tableau has an amazing community of very smart people who are willing to share their ideas. Jitter plots have been written about by at least three Tableau Zen Masters that I know of: [Steve Wexler](#), [Mark Jackson](#), and [Jeffrey Shaffer](#).

If it's good enough for them, it's good enough for me. This chapter shares my favorite technique for creating jitter plots in Tableau, and also shows you how to put the intensity of the jitter into the hands of your end users.

Consider the following box and whiskers plot, which shows the sales distribution of all of the customers in the Sample – Superstore data source:

[Insert Image 71-1]



While the box and whiskers themselves are providing some valuable information about the distribution of the customers, it is impossible to get a good sense of how many customers are at each sales value on the Y-axis. We are able to see a couple of outliers at the very top, but the view summary in the bottom left corner of the authoring interface is telling me that there are 793 marks (i.e. customers) on the view. I cannot tell what is happening with the other 791 customers because they are all overlapping on a single column in the chart.

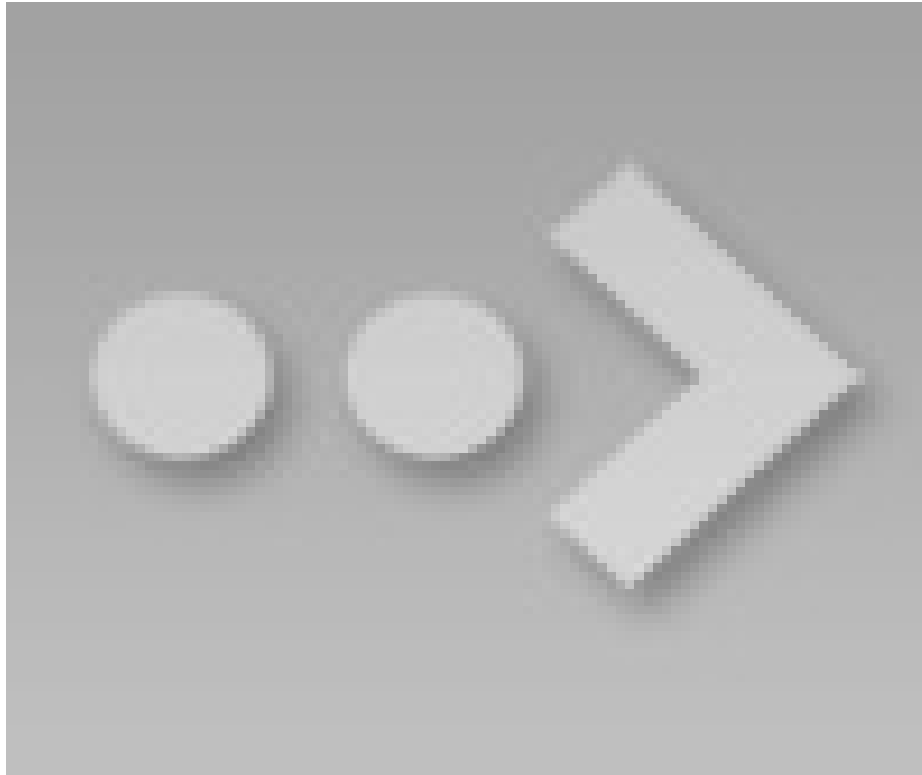
Jitter plots are created using the INDEX() function and changing the addressing to the most granular level of detail – which in our case is Customer Name. To get started with jitter plots, create a calculated field named Jitter that includes only the INDEX() function:

[Insert Image 71-2]



When this newly created calculated field is placed on the Columns Shelf, a new axis will appear, but all of the marks will still be lined up in one column. To apply the jitter, click on the calculated field, hover over “Compute using” and choose the most granular level of detail: Customer Name:

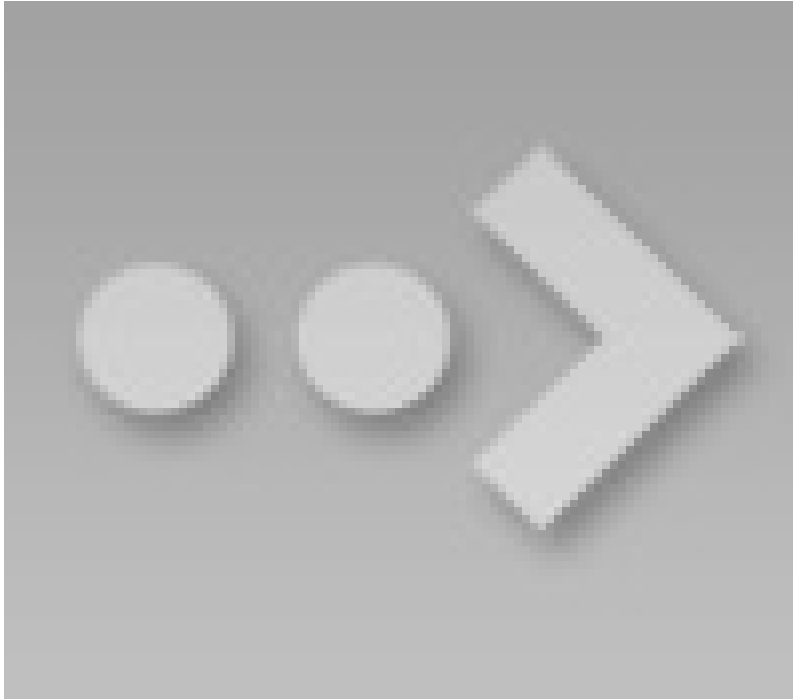
[Insert Image 71-3]



For more on table calculations and changing the addressing, see the chapter: [An Introduction to Tableau Table Calculations](#).

After changing the transparency and size of the marks, hiding the Jitter axis, and bringing the right side of the view in to make the chart skinnier, I am left with this jitter plot:

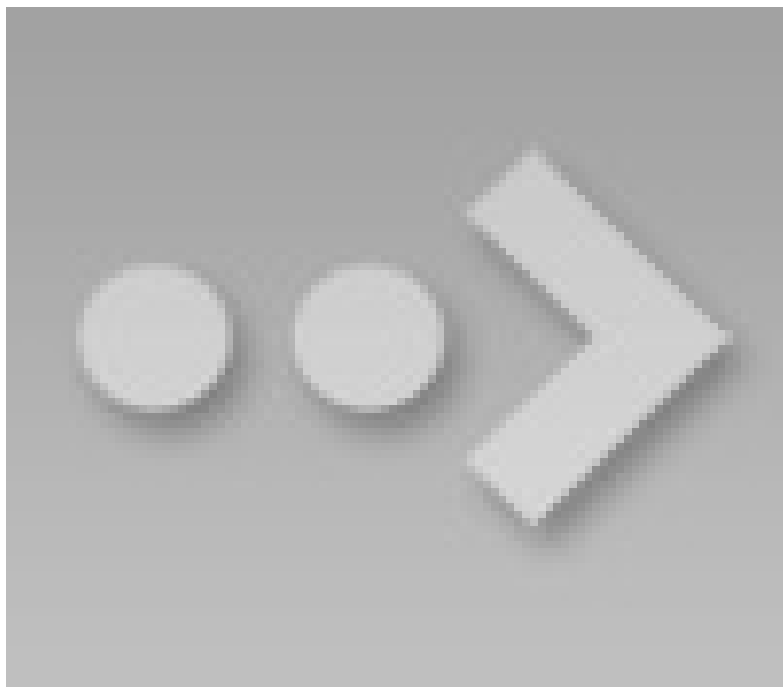
[Insert Image 71-4]



As you can see, it is now much easier to see how our customers are distributed on the Y-axis. The horizontal position of each mark type does not mean anything to the analysis, but it separates the marks so they are not all lying on top of each other.

This is a perfectly usable jitter plot, but there may be times where you want to control the intensity of the jitter, or how many columns there are on the X-axis. To hard code the number of columns used to jitter the marks, simply edit the Jitter calculated field and add %X, where X represents the number of desired columns, immediately after the INDEX() function. Here's how the same analysis from above looks when I edit the underlying formula to be "INDEX()%20":

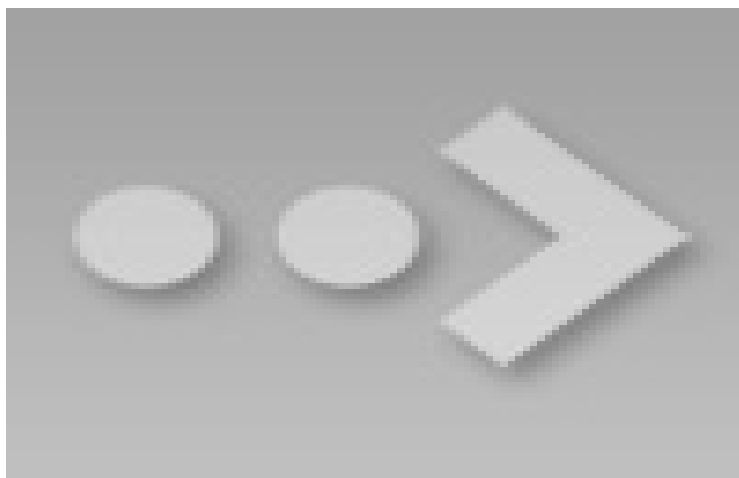
[Insert Image 71-5]



This technique works particularly well if additional dimensions are going to be added to the columns shelf for the analysis because all of the columns will have the same jitter intensity specified in the previous step.

Instead of picking just one number to specify the jitter intensity, you can even replace the number in the jitter calculated field with an integer parameter to allow the end user to quickly change it. Here's how my jitter calculated field looks after building a parameter and replacing the "20" with the new parameter:

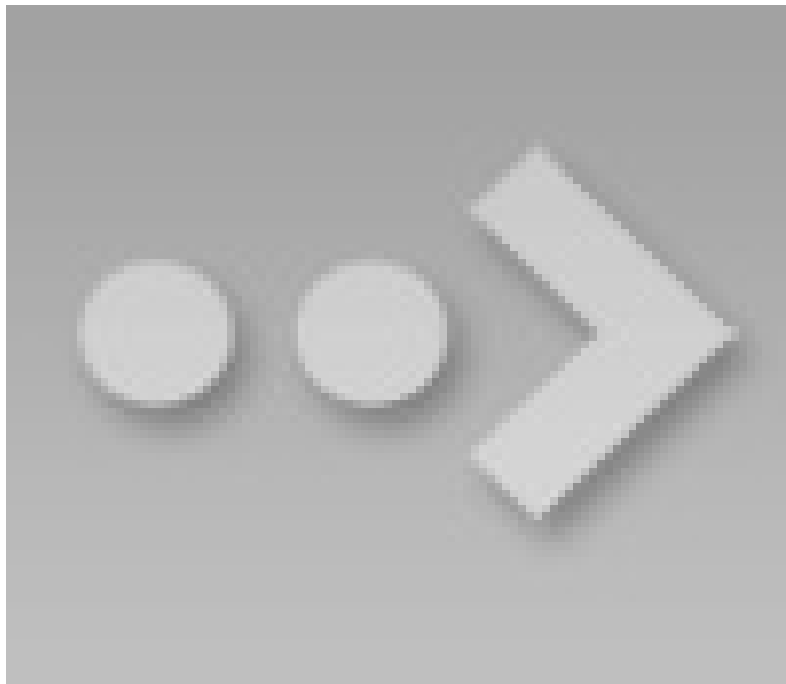
[Insert Image 71-6]



For this example, I set up my parameter to include integers from 10 to 100 with

For this example, I set up my parameter to include integers from 10 to 100 with a step size of 10. After showing the parameter control, my end users can change the intensity of the jitter plot, picking any multiple of 10 between 10 and 100. Here's how my final view looks after some formatting to remove the vertical zero line and changing the jitter intensity to 50:

[Insert Image 71-7]



For more on parameters, see the chapter: [An Introduction to Tableau Parameters](#).