Tableau for Slot Performance Analysis

Course Manual Version 1.1

William Dunn - Instructor

DUNN GAMING SOLUTIONS

Slotfocus® PERFORMANCE ANALYTICS

Contents

Tableau for Slot Performance Analysis

Section 1

Objectives

- Review data types
- Tableau environment overview
- Review of our slot data (dimensions and measures)
- Create a text table
- Calculate Measures: Average Bet, Theo Win Per Machine Day, and Occupancy
- Create logical calculations to scrub and filter data
- Use some of Tableau's Functions
- Discuss Statistical vs Taxable slot win and create variance calculations
- Learn about different types of data visualizations
- Join our slot data to a secondary data source

What is Tableau?

Tableau is a visual data analytics platform. There are various modules such as Prep, Data Management and Reader, but our focus will be on the core data analysis module. This can be either Tableau Desktop, Server, or Cloud. Desktop is software installed locally on your computer, whereas Server and Cloud are hosted in a server environment. Server and Cloud are essentially the same software, but Cloud is hosted and maintained by Tableau. Within Cloud and Server, there are three license types: Creator, Explorer, and Viewer. Tableau Desktop is the preferred environment for authoring and editing workbooks which can then be published to the server environment. Creators can author and edit in the server environment, but not all of the Desktop functionality is included.

Tableau also offers a free server environment known as Tableau Public. You can author workbooks with it, but saving, sharing, and security functionality is limited.

Data Types

There are six basic data types supported in Tableau:

Icon	Data type	Notes
Abc	Text (string) values	
	Date values	
	Date & Time values	
#	Numerical values	Can be INT (whole number) or FLOAT (decimal)
T|F	Boolean values (relational only)	True / False values
	Geographic values (used with maps)	

Discrete vs Continuous

Discrete data is the type of data that has clear spaces between values. Continuous data is data that falls in a constant sequence. Discrete data is countable while continuous is measurable.

Tableau Quirks

Up to at least Desktop version 2020.4, you may encounter issues, depending on your data source connection, if data types change in your source data (even a numeric changing from INT to FLOAT). You can resolve by going Data/Edit Data Source and clicking on the data type icon to respecify the data type.

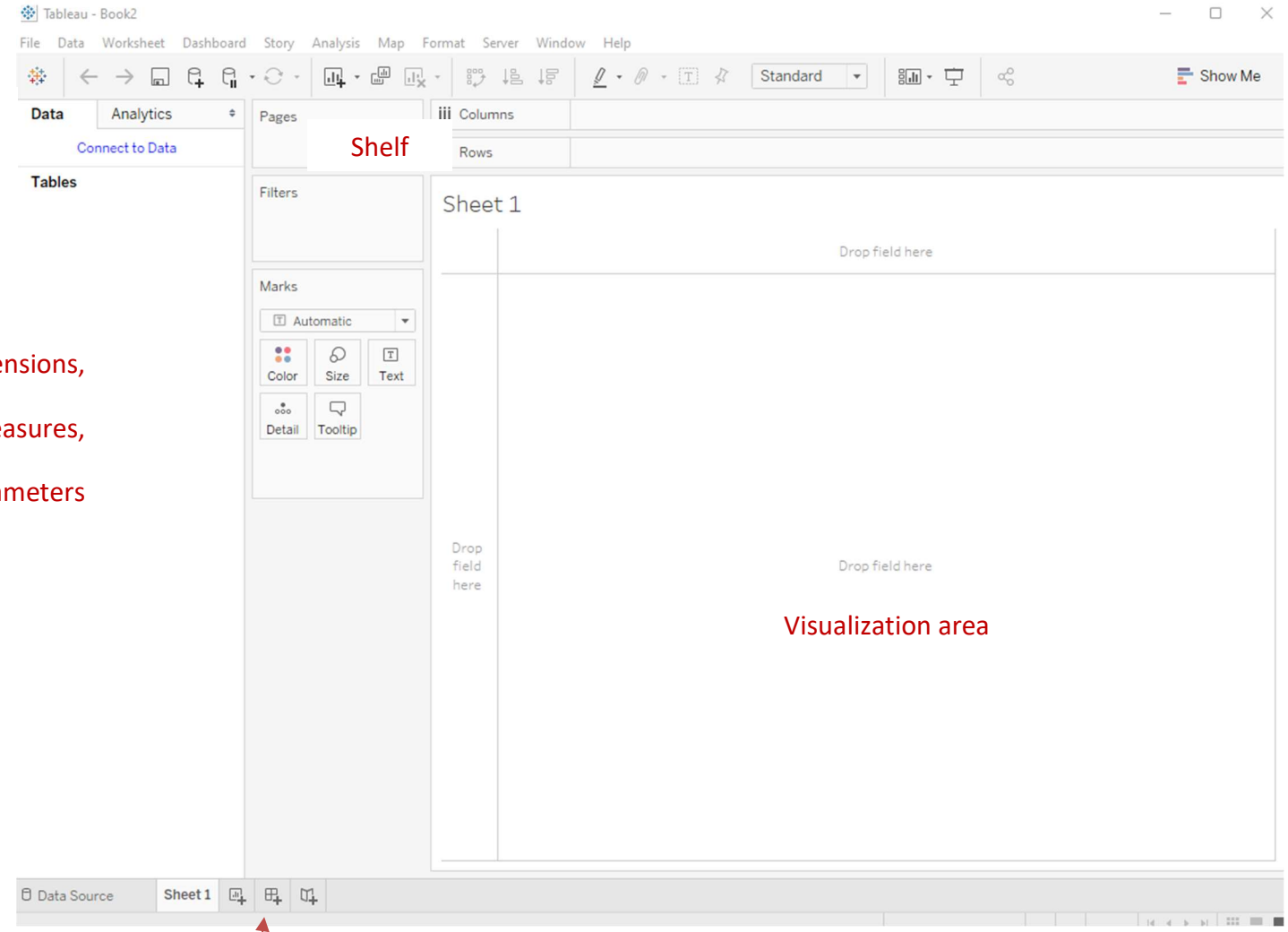
Not all calculations are supported for every data source type. For example, you can do a MEDIAN calculation with an Excel connection, but not with a SQL Server connection. Level of Detail calculations are only supported for certain data sources.

Tableau is **CASE SeNsiTiVe**! Please be mindful in naming Calculations and Parameters as many reference others.

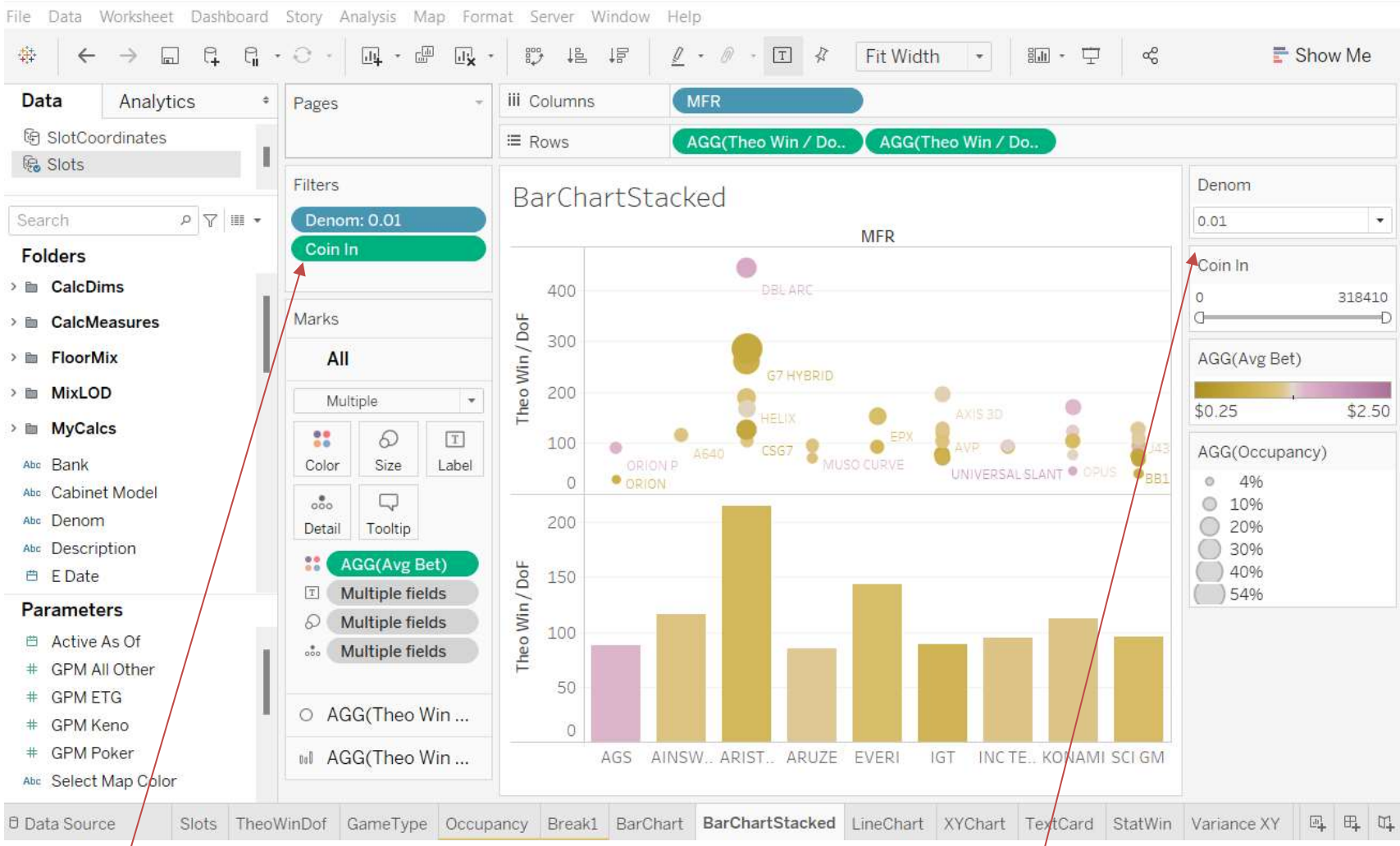
Tableau Environment

The Tableau workspace consists of menus, a toolbar, the Data pane, various shelves, and worksheets. Worksheets can be text tables or individual visualizations (such as a bar chart, line chart, scatter chart, geographic map, etc.). Individual worksheets can be combined on Dashboards.

Workspace Diagram



Worksheet with Data Filters



Filter “capsules” located on filter shelf establish data filters

Filter “cards” control the data filters

Connecting to Data Tableau Desktop (v 2020)

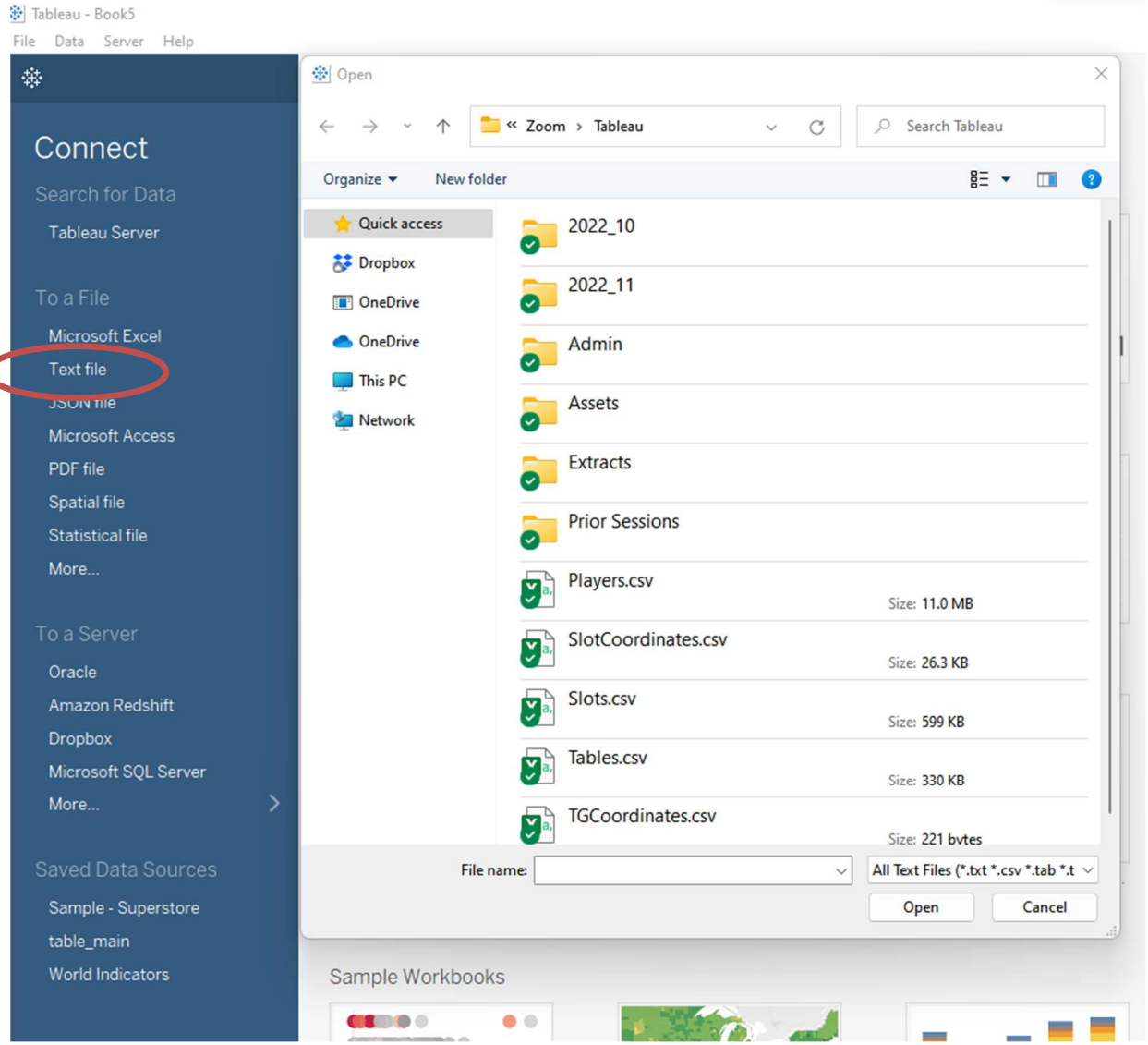
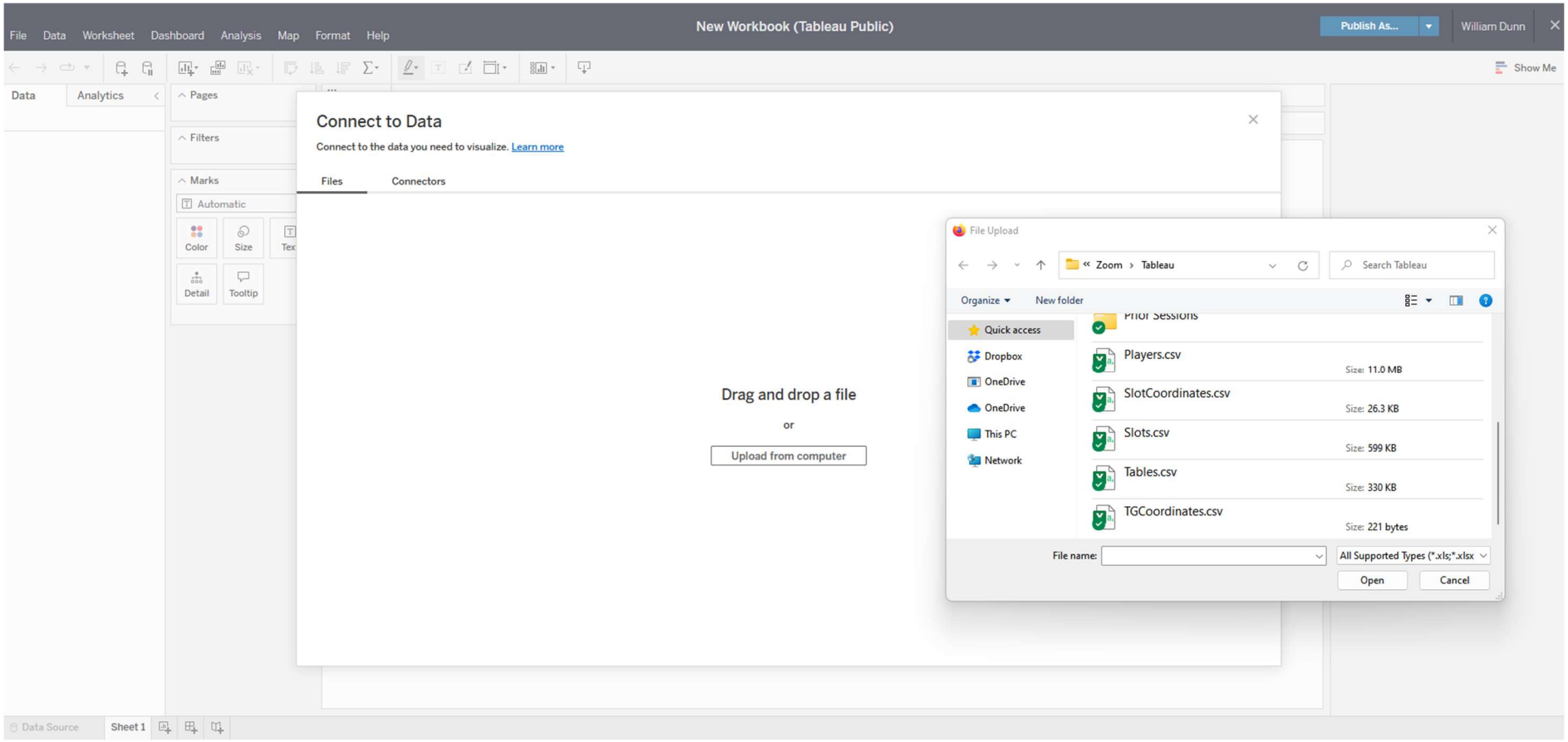
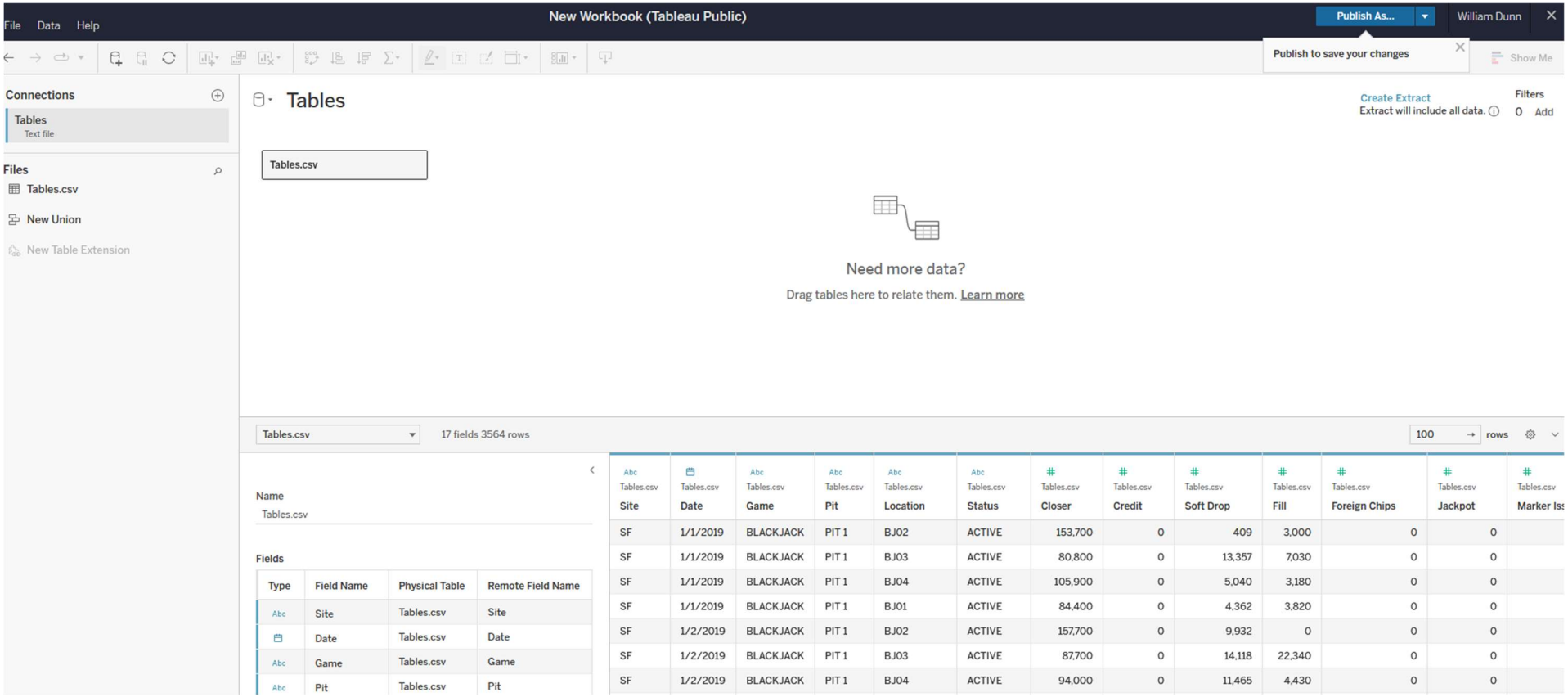


Tableau Public Online (v 2022)

Create / Web Authoring

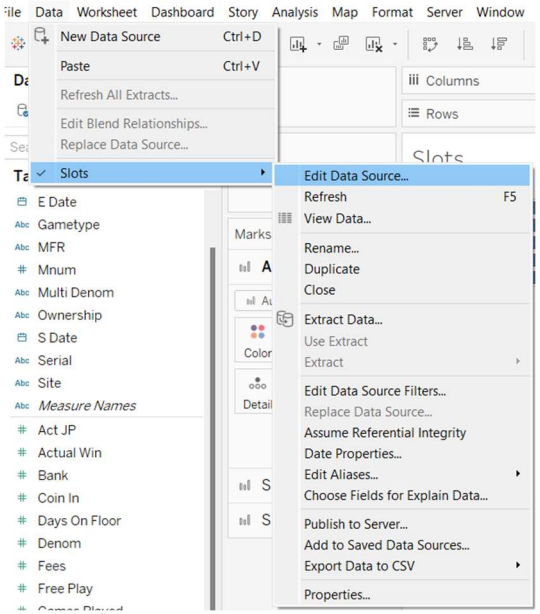


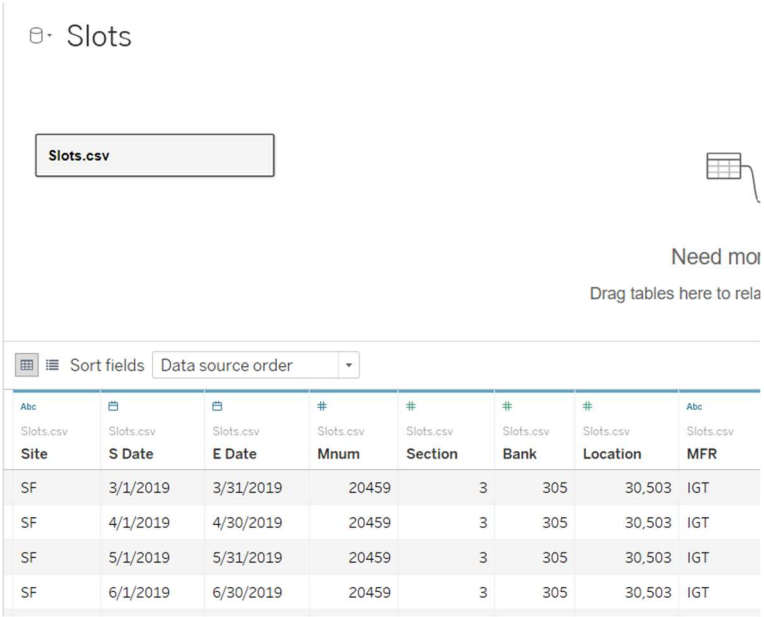
Then select “Update Now” for a preview of the data *(see screen shot next page)*.


File Data Help
New Workbook (Tableau Public)
Publish As...
William Dunn
Publish to save your changes
Show Me
Connections
Tables
Text file
Files
Tables.csv
New Union
New Table Extension
Tables
Tables.csv
Create Extract
Extract will include all data.
Filters
0 Add
Need more data?
Drag tables here to relate them. Learn more
Tables.csv
17 fields 3564 rows
100 rows
Name
Tables.csv
Fields
Type
Field Name
Physical Table
Remote Field Name
Site
Tables.csv
Site
Date
Tables.csv
Date
Game
Tables.csv
Game
Pit
Tables.csv
Pit
Site
Date
Game
Pit
Location
Status
Closer
Credit
Soft Drop
Fill
Foreign Chips
Jackpot
Marker Iss
SF 1/1/2019 BLACKJACK PIT 1 BJ02 ACTIVE 153,700 0 409 3,000 0 0
SF 1/1/2019 BLACKJACK PIT 1 BJ03 ACTIVE 80,800 0 13,357 7,030 0 0
SF 1/1/2019 BLACKJACK PIT 1 BJ04 ACTIVE 105,900 0 5,040 3,180 0 0
SF 1/1/2019 BLACKJACK PIT 1 BJ01 ACTIVE 84,400 0 4,362 3,820 0 0
SF 1/2/2019 BLACKJACK PIT 1 BJ02 ACTIVE 157,700 0 9,932 0 0 0
SF 1/2/2019 BLACKJACK PIT 1 BJ03 ACTIVE 87,700 0 14,118 22,340 0 0
SF 1/2/2019 BLACKJACK PIT 1 BJ04 ACTIVE 94,000 0 11,465 4,430 0 0

Exploring our Slot Data

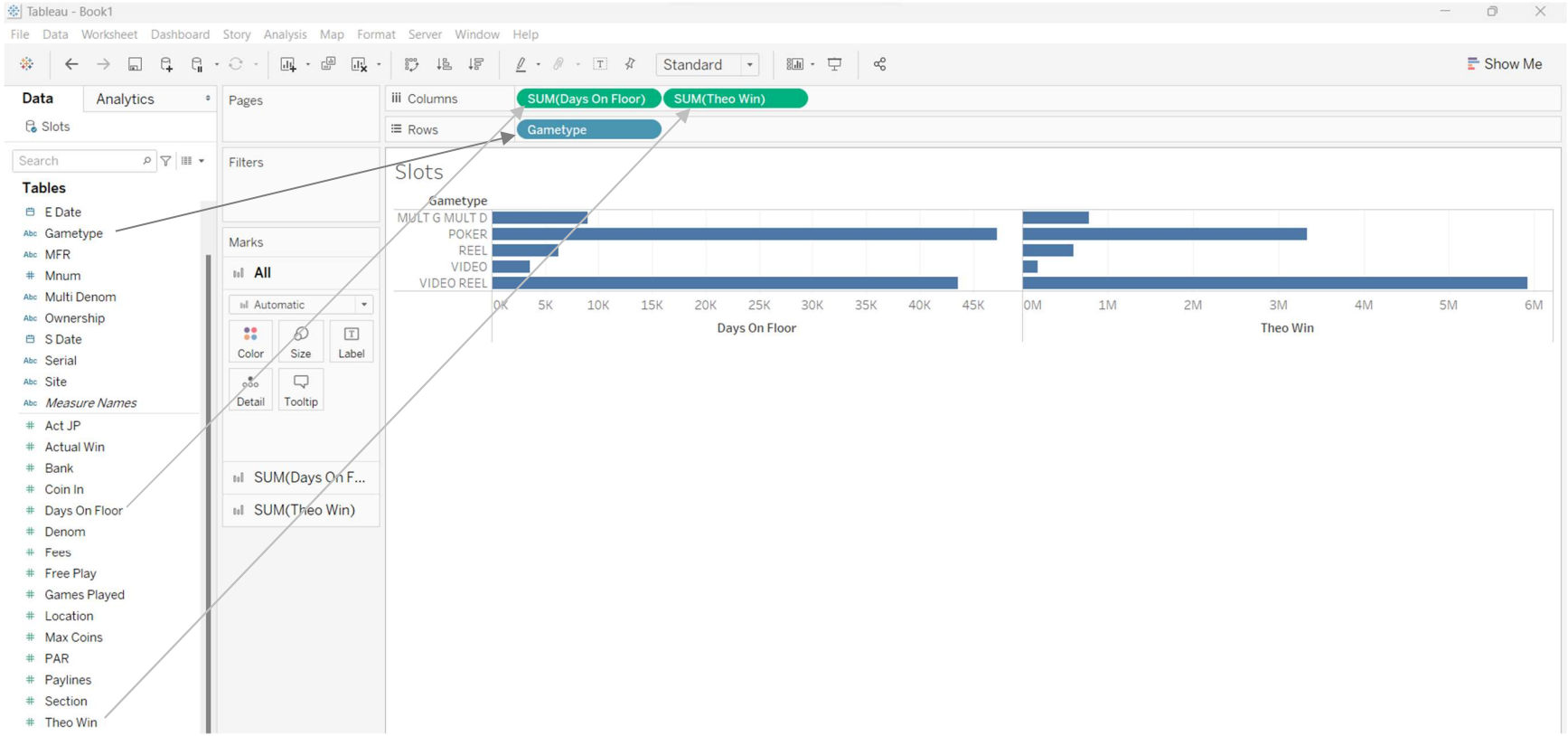
- Open your Slots workbook
- From the Data menu, select Slots, then Edit Data Source to review



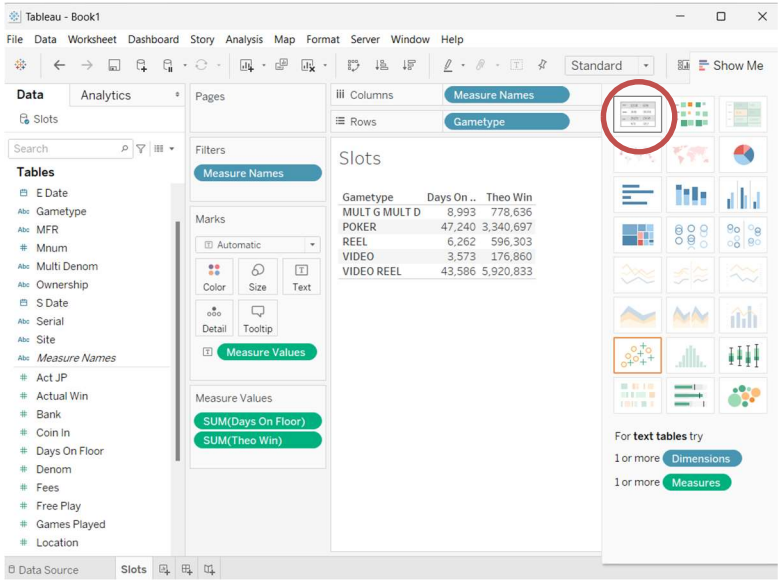


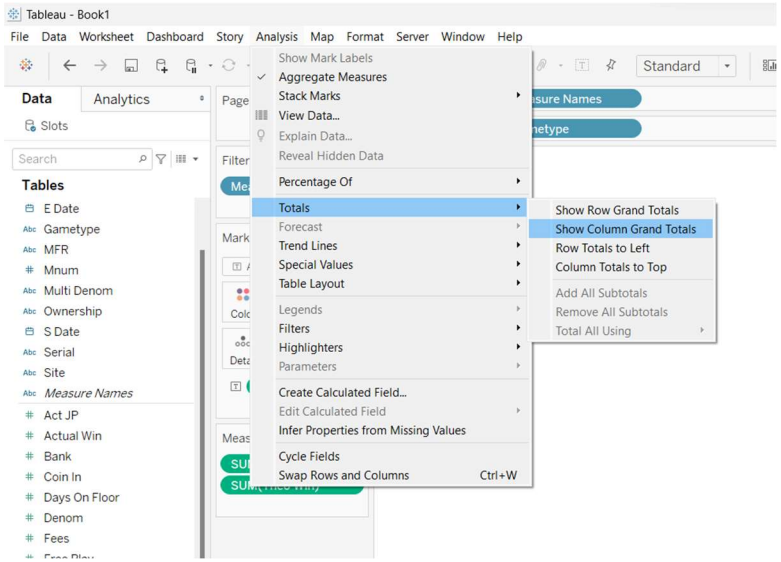
- In our data, Section, Bank, and Location were originally numeric and were converted to text. (this can be done in the Data Source editor or directly in the worksheet)
- In the Slots worksheet environment, Section, Bank, and Location were converted from Measures to Dimensions

- **Create a Text Table**
  - From the Slots worksheet, drag the Gametype dimension to the Rows shelf
  - Drag Days on Floor and Theo Win measures to the Columns shelf



- Click on Show Me and select Text Table (easiest to start with one Dimension and two Measures)
- From the Analysis menu, select Totals, Show Column Grand Totals





- Drag **E Date** to Rows and select and from the carrot on the capsule, select Month (with the Year) instead of the default Year. Note how Tableau changes the data from Discrete to Continuous. Remove **E Date**

Useful Slot Performance Calculations

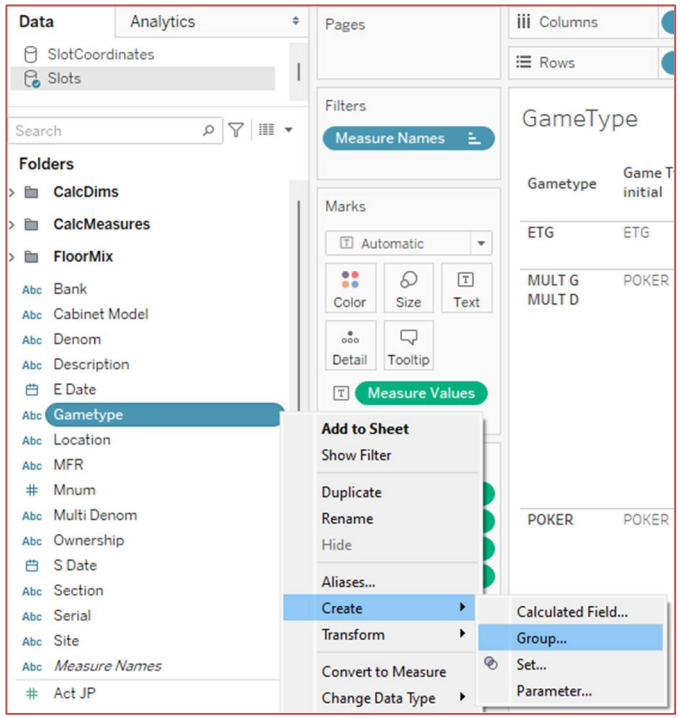
The dialogue box to Create Calculated Fields (for both measures and dimensions) and be accessed from the Analysis Menu, the dropdown carrot next to the Data search bar, or by right clicking a dimension or measure and following the "Create…" option.

- **Count Machines =** COUNTD([Mnum]) //This counts distinct machine numbers
- **Theo Win / DoF =** SUM([Theo Win])/SUM([Days On Floor])
- **Avg Days on Floor =** SUM([Days On Floor]) / [Count Machines]
  - Format as Number with one decimal place
- **Avg Bet =** SUM([Coin In]) / SUM([Games Played])
  - Format as Currency with two decimal places
  - Default Property Color Red-White-Green

Groups and Sets

Groups enable you to group attributes of dimensions together, whereas Sets are pre-defined groupings of filters. Sets are dynamic while groups are not. When your data changes the set will update with it while this is not an option with groups. Sets offer greater flexibility as you can link them to a condition.

We could create a group to combine Gametypes "VIDEO" and "VIDEO REEL".



IF / THEN / ELSE Expressions

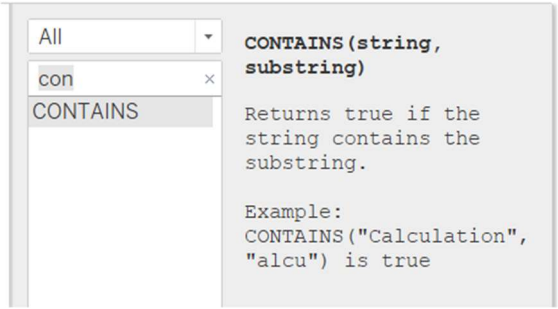
IF / THEN / ELSE is a logical test that evaluates a statement and executes a specific command if the statement is true. Otherwise, it executes an alternate command if the statement is false.

- **Game Type**

  IF [Gametype] = "MULT G MULT D" THEN "POKER"
  ELSEIF [Gametype] = "VIDEO REEL" THEN "VIDEO"
  ELSE [Gametype]
  END

- **The CONTAINS function**

  Using the CONTAINS function, we'll append the Game Type to create a category for **KENO**. Keep in mind that order of operation matters.



- **Occupancy** (estimate based on a rate of play of 10 games per minute for any video poker, 1.5 per minute for ETG and 6 games played per minute assumption all other; for 24 hours we use 60*24)

  IF [Game Type] = "POKER" THEN SUM([Games Played]) / (SUM([Days On Floor]) * (10*60*24))

  ELSEIF [Game Type] = "ETG" THEN SUM([Games Played]) / (SUM([Days On Floor]) * (1.5*60*24))

  ELSE SUM([Games Played]) / (SUM([Days On Floor]) * (6*60*24))
  END

  The above will give you an error about mixing aggregate and nonaggregate comparisons. The fix is to aggregate the Game Type dimension:

```
IF MAX([Game Type]) = "POKER"
THEN SUM([Games Played]) / (SUM([Days On Floor]) * (10*60*24))

IF MAX([Game Type]) = "ETG"
THEN SUM([Games Played]) / (SUM([Days On Floor]) * (1.5*60*24))

ELSE SUM([Games Played]) / (SUM([Days On Floor]) * (6*60*24))
END
```

- Format as Percentage with one decimal place

***Note**: it is not a best practice in coding to embed numbers (other than perhaps 0's and 1's for Booleans) such as our 10, 1.5, and 6 GPM assumptions. We will address this later by creating variables.*

In slot analysis, we usually want to segment machines owned by the casino (often described as Core games) from games with fees paid to the manufacturer (Premium or leased games). Our data has a **Fee Type** field populated with: "80/20", "Daily Fee", "Owned", and "WAP". We could use that as a filter, but to make it simpler, we can create a logical test named **Ownership** as:

```
IF [Fee Type] = "Owned" THEN "Owned"
ELSE "Lease/Premium"
END
```

Combining Dimensions

Dimensions can be combined simply by using the + function. For example, if we wanted to combine our Description and first the letters of the manufacturer dimensions in parenthesis to a dimension called **Game Theme**, we could utilize the + function and the LEFT(string, NumChars) function to write: [Description] + "(" + LEFT([MFR],3) + ")"

Break 1 Exercises

1. Go to the **Break1** worksheet; use **Slots** as the data source
2. Drag **MFR** to the Rows shelf
3. Drag **Actual Win** to the Columns shelf
4. Drag **Coin In** to the Columns shelf
5. Use the **Show Me** button to make your worksheet a Text Table
6. Show Column Grand Totals on your table (found in the Analysis / Totals menu)
7. Add Month/Year of **E Date** Filter (is asked "How do you want to filter on [E Date]" change the selection from "Range of Dates" to "Month/Year" and select "Use All" from the next dialogue box, then select Apply, OK)
8. Show the filter and filter the data to only January 2020
9. Compute SUM([Actual Win] – [Fees]) / SUM([Days on Floor]) named **Net Actual Win / DoF** and add to the Text Table
10. Convert **Max Coins** and **Paylines** to data type String and from convert from Measures to Dimensions

Bar, Line, and Scatter Charts and Text Marks

Bar charts, line (trend), and scatter (x/y) charts are three very useful data visualizations. Here in our online session, we will demonstrate creating and customizing these types of charts. Text marks can also be used to create informational data cards.

Statistical vs Taxable Slot Win

Statistical Win is the actual slot win value that is comparable to theoretical win. Any free play used contributes to these values. Slot theoretical win is Coin In (which includes any Free Play) * Game Theoretical Win %.

The actual win in our data is commonly referred to as Taxable Win (meaning and free play used is netted out of the amount). For example, if a machine's statistical win = $1,000 and $150 of Free Play was wagered, then the taxable win reported = $1,000 – $150 = $850.

- **Actual Win Statistical**
  SUM([Actual Win]) + SUM([Free Play])

- **Variance to Theo**
  [Actual Win Statistical] - SUM([Theo Win])

- **Variance % to Theo**
  [Variance to Theo] / SUM([Theo Win])

- **Actual Win %**
  [Actual Win Statistical] / SUM([Coin In])

- **Theo Win %**
  SUM([Theo Win])/ SUM([Coin In])

  Often, regulations require investigation of machines with large ***percentage point*** variances between Actual Win % and Theoretical Win %. We can therefore compute the variance and later learn how to apply a custom filter:

- **Pct Point Var to Theo**
  [Actual Win %] - [Theo Win %]

In Slot Performance Analysis, we usually want to evaluate based on contribution to the bottom line. For theoretical win, we can create a Net Theoretical measure to subtract any fees for premium games and any free play wagered:

- **Net Theoretical**
  SUM([Theo Win] - [Fees] - [Free Play])

  Having done this, we might want to modify our **Theo Win / DoF** computation to use **Net Theoretical**.

Coercing a Date to a YYYYMM Integer

- YEAR and MONTH functions return numeric value of year and month number on a date field
- STR coerces non-text to text while INT coerces non-numeric to numeric
- **YYYYMM**

```
INT(
  STR(YEAR([E Date])) +
  IF LEN(STR(MONTH([E Date]))) = 1 THEN "0" + STR(MONTH([E Date]))
  ELSE
  STR(MONTH([E Date]))
  END
  )
```

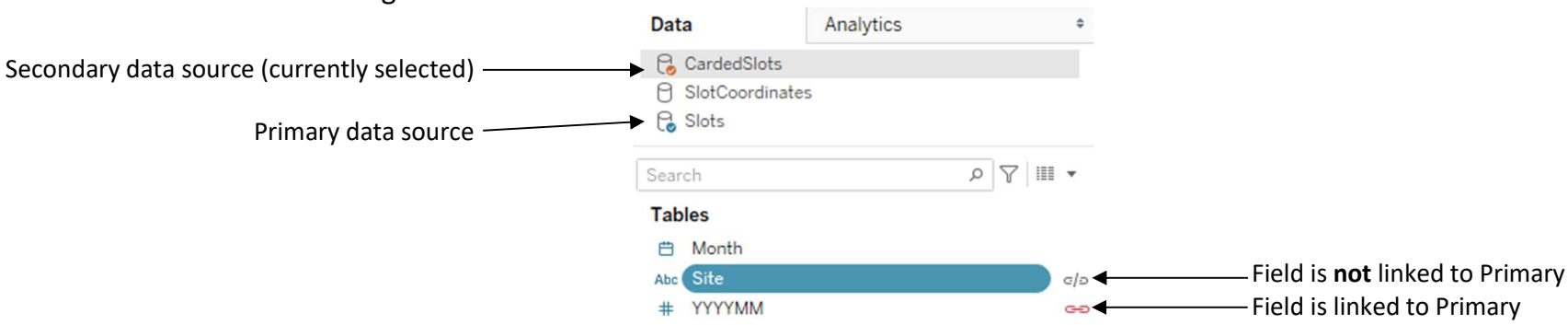
- Change YYYYMM from Measure to Dimension

Calculation Referencing Multiple Data Sources

- From the **Slots** data source, drag **Coin In** to columns and drag Month/Year of **E Date** to rows
- Click on the **CardedSlots** data source and link using YYYYMM and Site (***note*** *YYYYMM need to be a Dimension, not a Measure, in both data sources).*
- Drag **Carded Slot Coin In** to columns
- Click back on the **Slots** data source (this is the source that will contain our calculation)
- From the Slots data source, create **Carded Pct of Coin In**:

```
SUM([CardedSlots].[Carded Coin In]) / SUM([Coin In])
```

- Format as Percentage



Day 1 Assignments

1. Comple any unfinished Break1 assignments

Complete the following in the worksheet called **Day1:**

2. Edit the **Occupancy** calculation (located in CalcMeasures folder) to include Game Type = Keno with a rate of play assumption of 5 games played per minute (verify this change the KENO Occ % in G20 cabinet model).

3. Create a calculation to render “S 2000” as “S2000” from the Cabinet Model field (an IF THEN ELSE statement will work well for this) and name it **Cabinet**

4. Create a dimension that combines [MFR] + [Cabinet Model] (or, in place of Cabinet Model, use your Cabinet calculation from above that corrected “S 2000: to “S2000”); try adding a blank space to make more readable

5. Remove **Game Type** and **Cabinet Model** from Rows (also remove your *Cabinet* dimension if you dragged it in)

6. In Rows, replace **MFR** with **YYYYMM** (drag and drop; YYYYMM is located in the CaclDims folder)

7. Adjust the **MFR** filter from showing only IGT to show **All**

8. From the Data source selector, select **Carded Slots**; join your **Slots** data to secondary data source **CardedSlots** on the **Site** and **YYYYMM** dimensions

9. Go back to the **Slots** data source and compute SUM([CardedSlots].[Carded Slot Win Theo]) / SUM([Theo Win]) as **Carded Pct of Theo;** format as Percentage with one decimal and add to Measure Values on the Text Table

10. Create a Folder called **MyCalcs** and place all of the above calculations (and your **Net Actual Win / DoF** from the Break 1 assignment) in that folder; right clicking on any of your calculations will allow you to access the Folders dialogue

Tableau for Slot Performance Analysis

Section 2

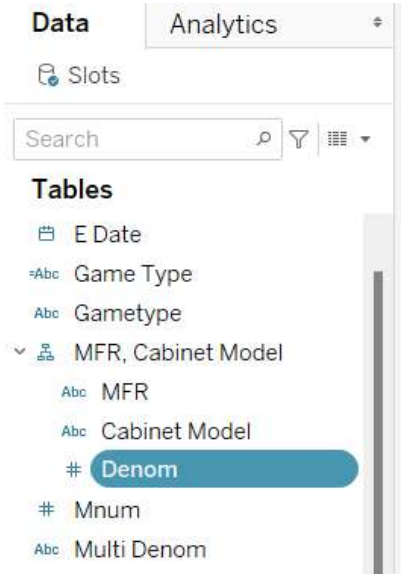
Objectives

- Create a Tableau Hierarchy
- Create Parameters to capture user input and calculate measures/dimensions that reference parameter values
- Learn how to customize Tooltips
- Create a Dashboard
- Create Dashboard Actions
- Learn how to use Tableau's Table Calculation feature
- Demonstrate Tableau A.I. Explain Data feature

Hierarchies

Tableau Hierarchies are used for nesting Dimensions. To create one, simply drag one Dimension on top of another and Tableau will create a Hierarchy (though it is a good practice to make copies of the Dimensions you wish to use, so you can maintain access to the separate individual dimensions). To add more, drag in other Dimensions. However, in my opinion, Tableau's hierarchies are not user intuitive. We'll demonstrate.

In this example, Cabinet Model was dragged onto the MFR (manufacturer) dimension, then Denom was dragged into the MFR, Cabinet Model hierarchy.

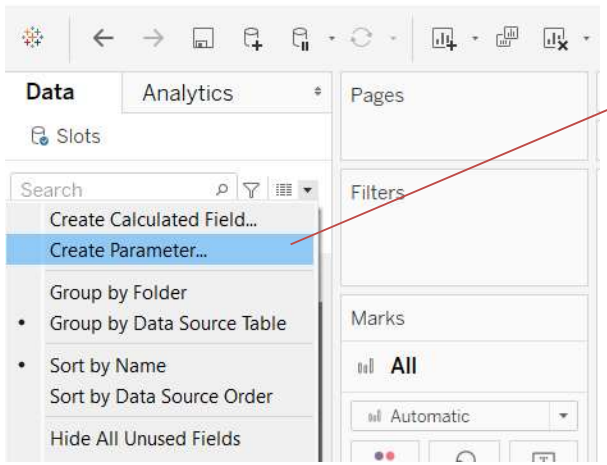


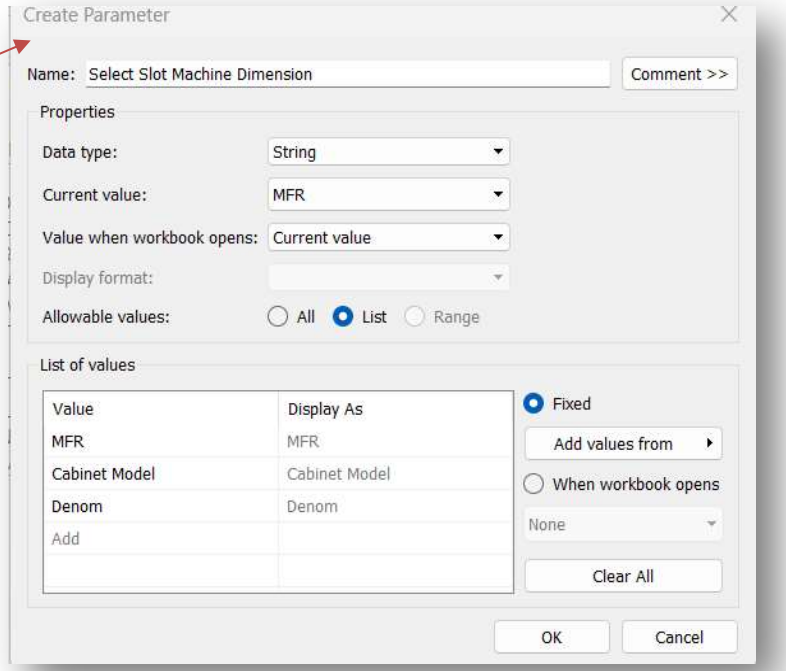
Parameters and CASE Statements

Parameters enable user input in Tableau. They can be used to change how data are viewed, to store variables for calculations, and even to change what worksheets are displayed on a dashboard. They can be numeric, string, or date/time data types.

- We have a simple parameter called "**Enter a Number**" and a measure called **Theo Win * Number Entered** as an example on the Param worksheet

- Create a **Parameter** named "**Select Slot Machine Dimension**" (from the dropdown carrot next to the Search bar on the Data pane) then right click on the parameter to Show Parameter





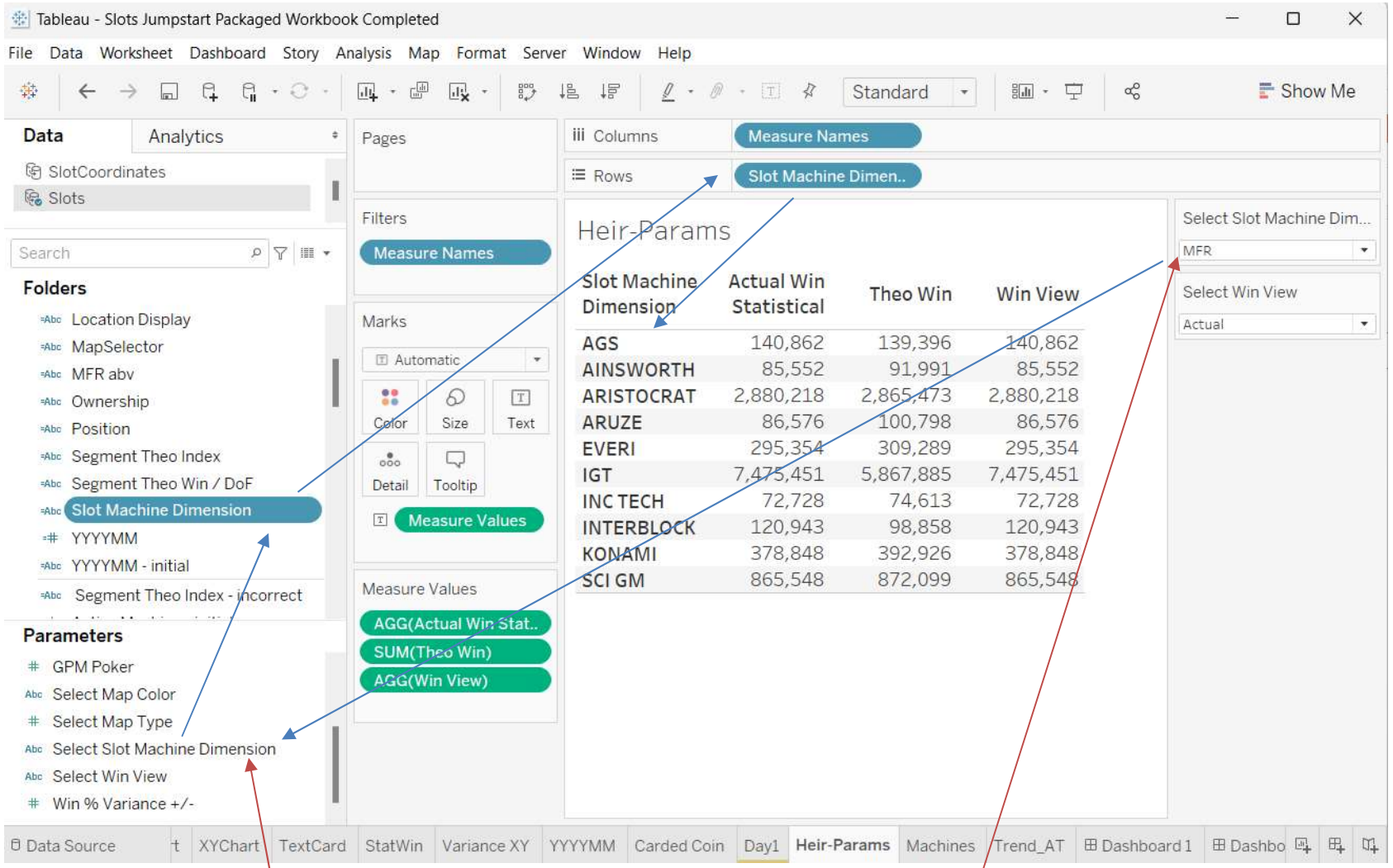
Next, using a CASE statement, we'll create a measure that references the parameter to return the dimension selected in the parameter:

- **Slot Dimension**

```
CASE [Select Slot Machine Dimension]
WHEN "MFR" THEN [MFR]
WHEN "Cabinet Model" THEN [Cabinet Model]
WHEN "Denom" THEN [Denom]
END
```

To summarize the process:

1. You create a parameter to store a value that can be manipulated by the user
2. The parameter contains values and, when specifying a list of values, you can define how the values are displayed to the user
3. You create a calculation that references the parameter value (in the example above, the parameter contains text values "MFR, Cabinet Model, and Denom")
4. You specify what you want your calculation to do based on the parameter's value (in our example, when the parameter value = "MFR", we have our calculation return the [MFR] dimension



1) Parameter is created and can modified here

2) Parameter value is selected on the parameter's card

3) Calculation references the parameter's value and, in this case, returns the MFR dimension

Parameters are also useful for allowing users to select a measure they want to view in a visualization.

Let's create a new Parameter named **Select Win View** (string, list values = Actual, Theoretical). Then, we'll create a Measure that references the Parameter and returns the selected measure.

- **Win View**

```
CASE [Select Win View]
WHEN "Actual" THEN [Actual Win Statistical]
WHEN "Theoretical" THEN SUM([Theo Win])
END
```

Numeric Parameters are useful for allowing users to filter data based on values. While this can often easily be accomplished by simply using measure values as filters, there are use cases where parameters can be used to achieve desired results. Say we want to find outliers (such as machines with Actual Win % outside of +/- 15%). We can go to the **Machines** worksheet to demonstrate.

- Create a Parameter named **Actual Win % +/-** (type = float, current value = 0, display format as %)

- Create a Calculated Field called **Actual Win % Filter**

```
IF [Actual Win %] <= - [Actual Win % +/-] THEN TRUE
ELSEIF [Actual Win %] >= [Actual Win % +/-] THEN TRUE
ELSE FALSE
END
```

- Add as a Filter = True

Date Parameters can be used to filter our data to only include machines that are currently active on the floor (our data is through 1/31/2020).

- Create a new Parameter named **Active As Of** (type = Date, current value = 1/31/2020)
- **Active Machine – initial** (note this one does not function perfectly in Tableau; we'll fix it later)

```
IF MAX([E Date]) = [Active As Of] THEN TRUE
ELSE FALSE
END
```

Finally, we'll use a Parameter as a variable for our **Occupancy** calculation's number of games played per minute assumption.

- Create a Parameter named **GPM Poker** (type = float, current value = 10) and edit the **Occupancy** calculation:

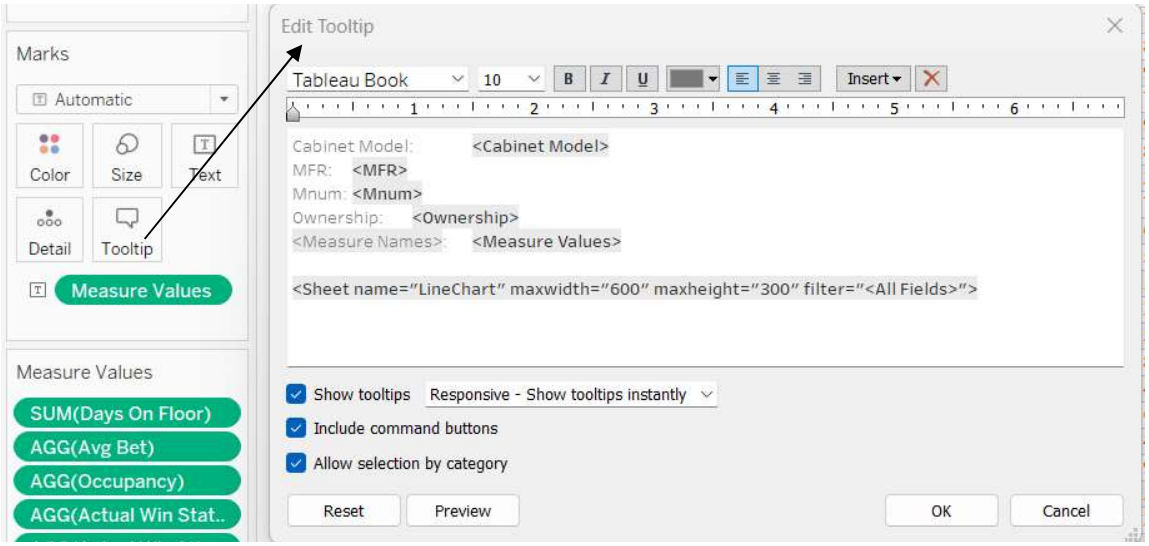
```
IF MAX([Game Type]) = "POKER"
THEN SUM([Games Played]) / (SUM([Days On Floor]) * ([GPM Poker]*60*24))
```

Tooltips

The Tooltip in tableau is a useful feature located on the Marks shelf which helps us understand the values and attributes in the dataset associated with any graph or visualization. A tooltip is anything that appears in the form of text labels when you hover over an element or focus on a particular object. You can also embed a worksheet data visualization. Tableau tooltips, therefore, help you to identify or locate elements on either the dashboard or under the workflow when they're activated. They help in producing cleaner and more appealing visuals.

Tableau tooltips are one of the best ways to add contextual information and data without taking up any space on your dashboard. Tooltips have other options to generate groups and sets, display marks that have the same value and view the underlying data. In short, tooltips generate additional data.

We'll demonstrate tooltips on our **Machines** worksheet.



Break 2 Assignment

1. In the **Occupancy** worksheet, create Float Parameters to store games played per minute ("GPM") assumptions for game types ETG (value = 1.5), Keno (value = 5), and All Other (value = 6). *Shortcut: you can duplicate and edit the GPM Poker Parameter*
2. Show each of these Parameters on the worksheet
3. Edit the **Occupancy** calculation to reference the values stored in these Parameters

Creating a Dashboard

Dashboards can contain worksheets, text boxes, images, webpages, and containers. Here in our online session, we will demonstrate Dashboard creation and formatting.

Dashboard Actions

Actions enable user interactivity with Worksheets and Dashboards. You can create actions to:

- Filter the worksheets on a Dashboard
- Highlight items
- **Pass a filter value and link to another Worksheet or Dashboard**
- Link to a Worksheet
- Link to a URL
- Change Parameters and set values

You can create automatic filters for worksheets on a dashboard by pressing the filter button on a given worksheet.

Filter enabled:

Filter disabled:

You can also create and edit Actions from the **Dashboards / Actions** menu path.

Let's create an Action called "**Highlight Bar Chart**" to our dashboard to have the Machines worksheet highlight the bar chart instead of filtering it. We'll also need to edit the automatically generated filter from our Machines worksheet to stop it from filtering our bar chart.

- "Run action on"
  - Hover (will happen when you mouse over)
  - Select (will happen when you click)
  - Menu (will appear as a link in a tooltip)
- "Clearing the selection will"
  - Keep the filtered values (no change to the Target Sheet)
  - Show all values (clears the filter from the Target Sheet)
  - Exclude all values (hides all data on the Target Sheet)

Finally, we'll create a filter action to link from the Machines text table on the dashboard to our Line Chart worksheet.

Table Calculations

Table calculations are a special type of calculated field that computes on the local data in Tableau. They are calculated based on what is currently in the visualization and do not consider any measures or dimensions that are filtered out of the visualization. You can use table calculations for a variety of purposes, including:

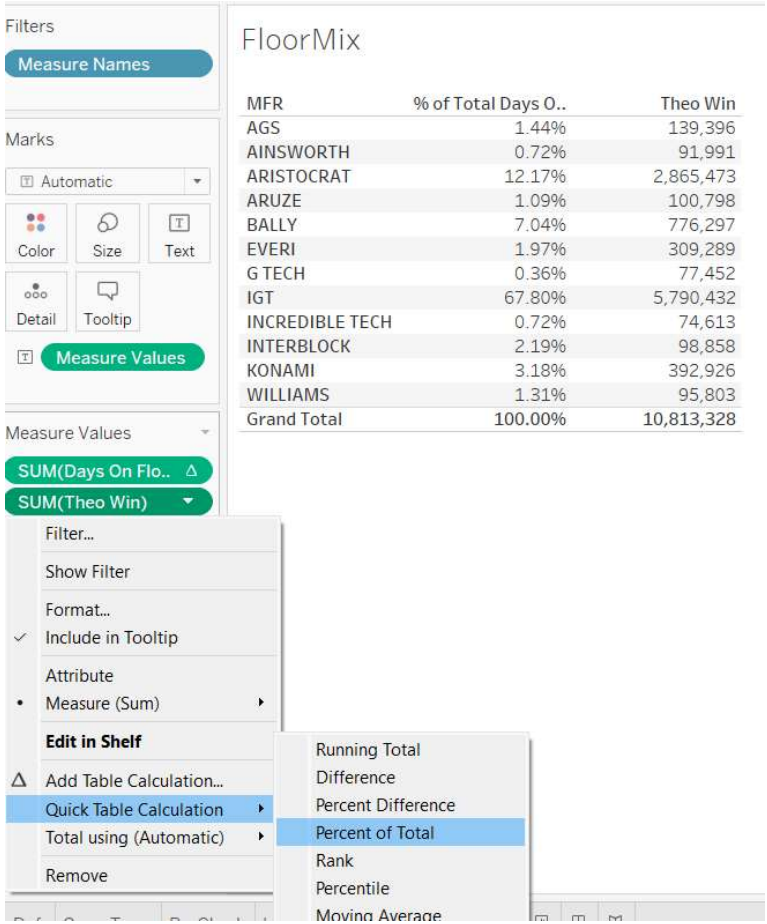
- Transforming values to rankings
- Transforming values to show running totals
- Transforming values to show percent of total
- Transforming values to differences

For any Tableau visualization, there is a virtual table that is determined by the dimensions in the view. This table is not the same as the tables in your data source. Specifically, the virtual table is determined by the dimensions within the "**level of detail**," which means the dimensions on the shelves and filters in a Tableau worksheet.

We'll create a table calculation to show our floor mix (percent of total days on floor and percent of total theoretical win) by manufacturer. There is also % of Total a shortcut in the **Analysis** menu.

We could also alter our **Line Chart** worksheet to a moving average calculation using a Table Calculation.

Finally, we have an interesting bar chart in the **TableCalc** worksheet we can modify to see the <u>difference %</u> in theo win.



FloorMix

MFR	% of Total Days O..	Theo Win
AGS	1.44%	139,396
AINSWORTH	0.72%	91,991
ARISTOCRAT	12.17%	2,865,473
ARUZE	1.09%	100,798
BALLY	7.04%	776,297
EVERI	1.97%	309,289
G TECH	0.36%	77,452
IGT	67.80%	5,790,432
INCREDIBLE TECH	0.72%	74,613
INTERBLOCK	2.19%	98,858
KONAMI	3.18%	392,926
WILLIAMS	1.31%	95,803
Grand Total	100.00%	10,813,328

Tableau AI: Explain Data

Explain Data in the Data Guide helps you to inspect, uncover, and dig deeper into the marks in a viz as you explore your data. You can use Explain Data to analyze dashboards, sheets, or selected marks for possible outliers and correlations in the underlying data. Explain Data builds statistical models and proposes possible explanations for individual marks in a viz, including potentially related data from the data source that isn't used in the current view.

Explain Data is:

- A tool and a workflow that leverages your domain expertise
- A tool that surfaces relationships in your data and recommends where to look next
- A tool and a workflow that helps expedite data analysis and make data analysis more accessible to a broader range of users

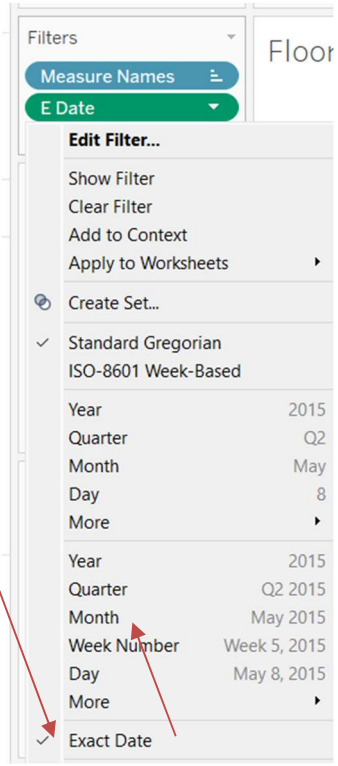
Explain Data is not:

- A statistical testing tool
- A tool to prove or disprove hypotheses
- A tool that is giving you an answer or telling you anything about causality in your data

We can demonstrate it on the **FloorMix** worksheet.

Tableau also offers **Generative AI** tools (Stories and Einstein) to analyze data and provide natural language summaries. However, it is currently only available on Tableau Cloud

Day 2 Assignments



1. On the **FloorMix** worksheet, add **E Date** as a filter, selecting "Range of Dates" as the filter style option; from the filter card, change the date option from Exact Date to Month (the one that also has the year; show the filter

2. Add **MFR** and **Ownership** as filters and show both filters on the worksheet

3. Change the **Net Theo** column to **Percent of Total Net Theo (along table down)** using a Table Calculation; do the same for the **Net Actual** column

4. Duplicate the **Utilization table calc Theo** calculation (located in the Floor Mix folder). Edit the copy's name to "**Utilization table calc Actual**" and edit the calculation to replace **Net Theo** with **Net Actual**

5. Show the **Select Win View** parameter on the worksheet

6. Edit the **Utilization table calc Selectable** measure to a CASE statement referencing the Select Win View parameter to be **Utilization table calc Theo** when the "Theoretical" is selected in the parameter and to be **Utilization table calc Actual** when the "Actual" is selected in the parameter

NOTE: after you edit **Utilization table calc (**which affects the **Optimal Change table calc),** they might turn to all **zero values** on your worksheet; to fix **remove both** measures from the Measure Values card **then drag them both back** to the Measure Values card.

7. Color the worksheet by the **Optimal Change table calc** (tip: change the Marks card option from Automatic to Square to make the color pop more and edit the color palette to your liking)

8. Modify the Tooltip to your personal liking

9. Create a new worksheet that is a single axis line chart of **Net Theo / DoF** and **Net Actual Win / DoF** (from Day 1 Assignments) on the Rows shelf and **Month / Year of E Date** on the Columns shelf

10. Create a dashboard with the Floor Mix and your newly created trend chart

11. Be sure the **MFR** and **Ownership** filters are applied to both worksheets on the dash

12. Add a Dashboard / Action to filter the trend chart by hoovering on the Floor Mix worksheet

Your dashboard should be similar to the sample Floor Mix dashboard found here:
https://public.tableau.com/app/profile/william.dunn7551/viz/SlotAnalyticsSamples/0?publish=yes

Section 3

Objectives

- Learn and create Level of Detail expressions
- Understand how Tableau's Order of Operations affects calculations
- Create Slot Performance Indexes
- Create Performance Categorizations
- Create a Slot Conversions tracking worksheet
- Create a Slot Floor Map visualization
- Complete your final assignment!

Level of Detail Expressions

Level of Detail expressions (also known as LOD expressions) allow you to compute values at the data source level and the visualization level.

Say we want to know the percentage of total Slot Coin In generated by each manufacturer. This computation will not work because it mixes aggregate and non-aggregate arguments and would SUM at the level of detail within the visualization:

[Slot Coin In] / SUM ([Slot Coin In])

We could create a Table Calculation, but if we want to filter the data in our table, we will lose access to the site total coin in. Instead, we'll create a Level of Detail (LOD) expression.

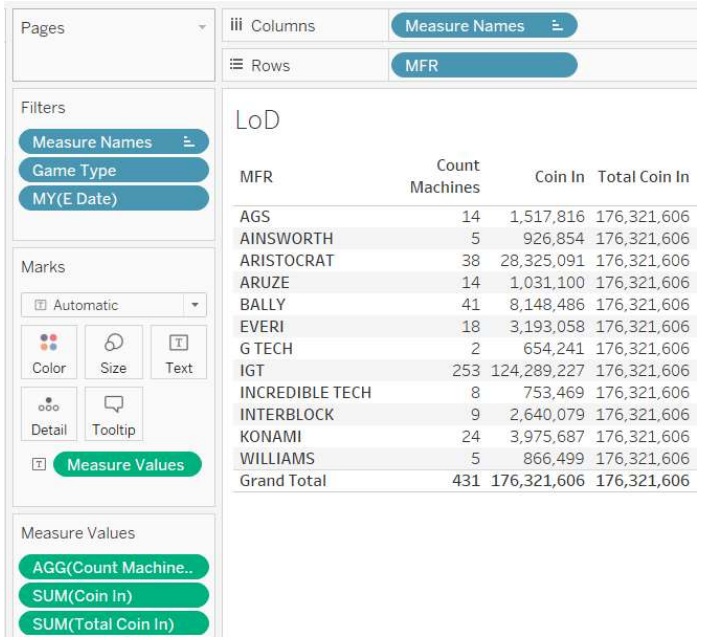
This calculation (named **Total Coin In**) will sum coin-in for the entire casino Site:

{ FIXED [Site] : SUM([Coin In]) }

Wherever this measure appears in a visualization, it will be **fixed** to the Site total. *Note: Site is optional in the above since our data contains only one casino site. So, for this data set you could use*:

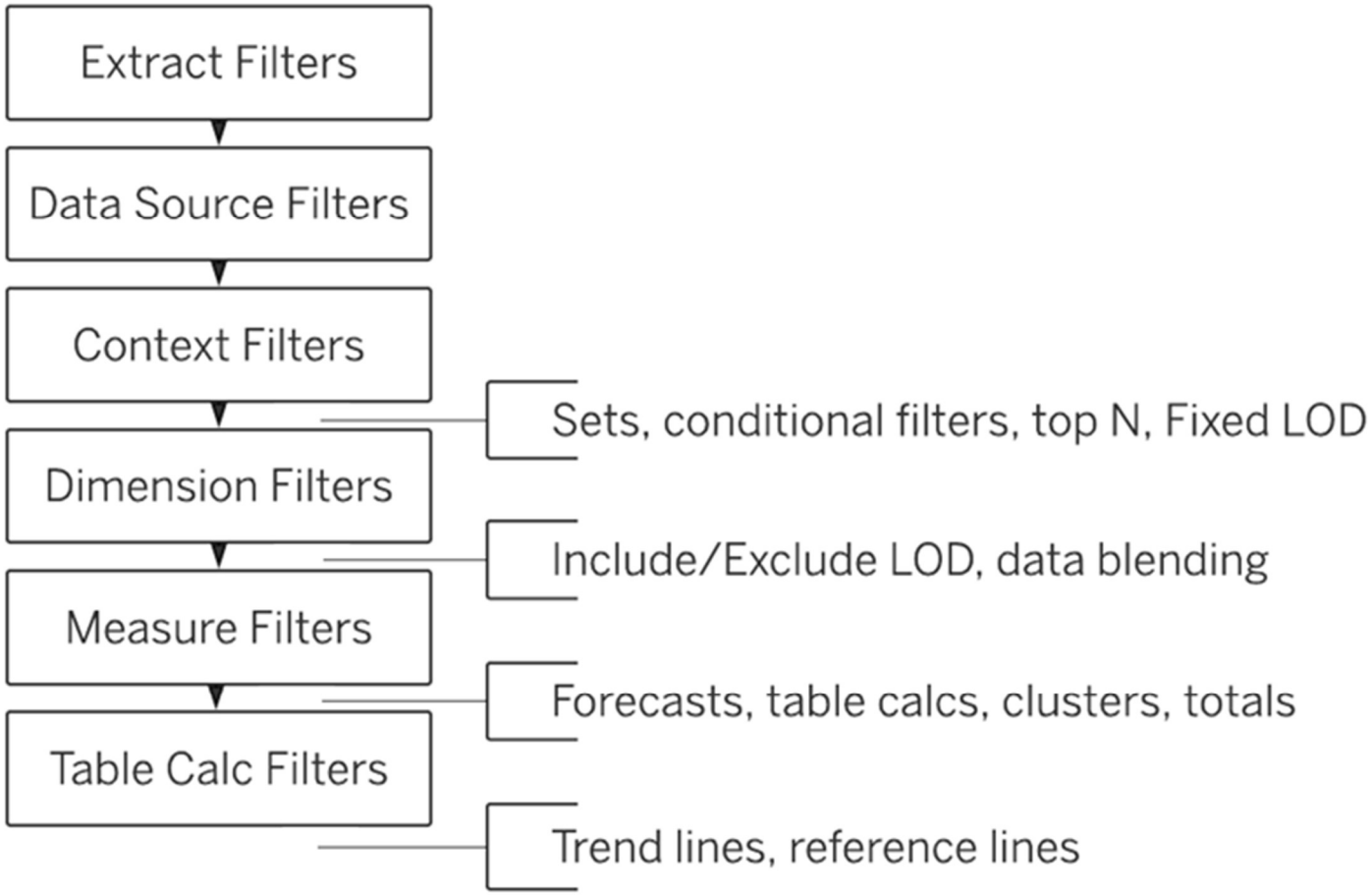
{ FIXED : SUM([Coin In]) } *to fix Coin In to the data-source total.*

When using LOD expressions, we need to understand Tableau's order of operations.



MFR	Count Machines	Coin In	Total Coin In
AGS	14	1,517,816	176,321,606
AINSWORTH	5	926,854	176,321,606
ARISTOCRAT	38	28,325,091	176,321,606
ARUZE	14	1,031,100	176,321,606
BALLY	41	8,148,486	176,321,606
EVERI	18	3,193,058	176,321,606
G TECH	2	654,241	176,321,606
IGT	253	124,289,227	176,321,606
INCREDIBLE TECH	8	753,469	176,321,606
INTERBLOCK	9	2,640,079	176,321,606
KONAMI	24	3,975,687	176,321,606
WILLIAMS	5	866,499	176,321,606
Grand Total	431	176,321,606	176,321,606

Tableau's Order of Operations



Creating Performance Indexes (more depth on LOD expressions)

An index is an individual value divided by either the average of the entire data series or some benchmark number. Indexes are used frequently in slot operations, especially in discussions about performance with the machine manufacturers. Indexes can be created in Tableau with Level of Detail expressions.

Technically, site overall Theo Win per Day on Floor would be:

```
{ FIXED [Site] : SUM([Theo Win]) } / { FIXED [Site] : SUM([Days On Floor]) }
```

Index to Site would be:

```
[Theo Win / DoF] /
MAX
(
{ FIXED [Site] : SUM([Theo Win]) } / { FIXED [Site] : SUM([Days On Floor]) }
)
```

But we can simply write **Theo Win / DoF Index** as:

```
[Theo Win / DoF] /  MAX ( { FIXED [Site] : [Theo Win / DoF] } )
```

Note we use the MAX function again to keep from mixing aggregate and non-aggregate arguments. Also, we can technically omit Site since there is only one site in the data source. { FIXED : [Theo Win / DoF] } *is the data source total.*

We can fix our expression to more than one dimension, such as Section in this **Index to Section Theo Win / DoF**:

```
[Theo Win / DoF] /  MAX ( { FIXED [Site], {Section] : [Theo Win / DoF] } )
```

Note we'll need to add our date filter to ***Context*** *due to Tableau's Order of Operations.*

Say we want to categorize and segment machines based on their performance indexes as:

- Above Average (index > 1.2)
- Below Average (index < 0.8)
- Near Average (index between 0.8 and 1.2)

We'll demonstrate why this **Index Segment** expression does not return the expected results:

```
IF [Theo Win / DoF Index] > 1.2 THEN "Above Average"
ELSEIF [Theo Win / DoF Index] < 0.8 THEN "Below Average"
ELSE "Near Average"
END
```

We would need to use an LOD expression that is fixed to the slot machine number to properly create this categorization:

- **Index Machine Theo Win / DoF**

```
{ FIXED [Mnum] : [Theo Win / DoF Index] }
```

We would then refactor our original **Index Segment** above to replace [Theo Win / DoF Index] with [Index Machine Theo Win / DoF].

Alternatively, you could reference **Theo Win / DoF** (instead of the index of it) to **Segment Theo Win / DoF** as:

```
IF { FIXED [Mnum] : [Theo Win / DoF] } > ({ FIXED : [Theo Win / DoF] } * 1.2) THEN "Above Average"
ELSEIF  { FIXED [Mnum] : [Theo Win / DoF] } < ({ FIXED : [Theo Win / DoF] } *0.8) THEN "Below Average"
ELSE "Near Average"
END
```

Though we would probably want to create calculations for each LOD used and refactor the expression.

As a reminder, if we want to filter our worksheet data, we need to be cognizant of Tableau's order of operations. If LOD expressions are used, we'll need to decide if we want to add the filters to Context.

Next, recall our **Active Machine** expression:

```
IF MAX([E Date]) = [Active As Of] THEN TRUE
ELSE FALSE
END
```

This can also be done with an LOD expression:

```
IF MAX([E Date]) = MAX({ FIXED [Site] : MAX([E Date]) }) THEN TRUE
ELSE FALSE
END
```

We have essentially replaced the Data Current Month with { FIXED [Site] : MAX([E Date]) }, but we need to put the MAX function on it, otherwise our expression would be mixing aggregate and non-aggregate arguments. This is cleaner because it does not require the user input parameter.

However, for this filter to function properly, we must also think about machines and locations:

```
IF { FIXED [Mnum], [Location] : MAX([E Date]) } = MAX({ FIXED [Site] : MAX([E Date]) }) THEN TRUE
ELSE FALSE
END
```

For our Floor Mix analysis, we can use LoD expressions to compute the overall **Total Days on Floor** as { FIXED : SUM([Days On Floor]) } along with **Total Theo Win**. We can create a **Pct Total DoF** measure as SUM([Days On Floor]) / MAX([Total Days on Floor]) along with **Pct Total Theo**.

Moreover, we can now create computations for the difference between % of theoretical and % of days on floor at the row level as **Utilization** = [Pct Total Net Theo] - [Pct Total DoF]. This enables us to compute the **Optimal Change** in machine count at the row level as [Utilization] * [Count Machines LOD], where Count Machines LOD is our data's total number of machines MAX({ FIXED : [Count Machines] }). You may recall Count Machines was COUNTD([Mnum]).

Slot Conversion Tracking: Complex Expressions

We'll create an LOD expression to identify locations that have had more than one game theme:

- **Multi Theme**

```
{ FIXED [Location] : COUNTD([Description]) }
```

Next, we'll create a Table Calculation to compute the % difference in Index to Theo Win / DoF along a row going down:

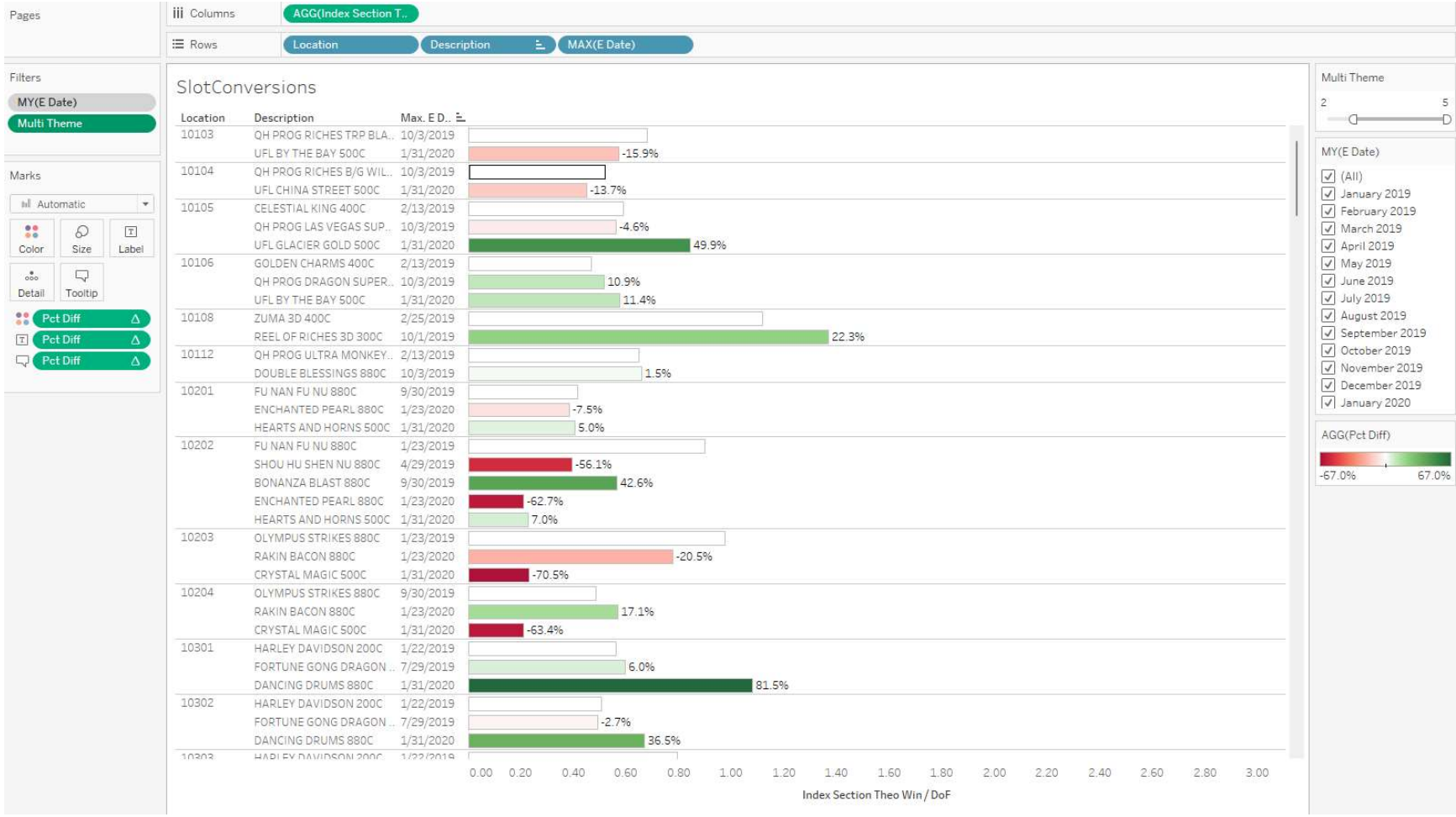
```
(ZN([Theo Win / DoF Index]) - LOOKUP(ZN([Theo Win / DoF Index]), -1))
/ ABS(LOOKUP(ZN([Theo Win / DoF Index]), -1))
```

Finally, we add IF THEN ELSE logic to compute the % difference only when the row is the same Location as the prior row. This suppresses the % difference value label in the first theme in each location.

- Location Pct Diff

```
IF MAX([Location]) = LOOKUP(MAX([Location), - 1) THEN
(ZN([Theo Win / DoF Index]) - LOOKUP(ZN([Theo Win / DoF Index]), -1))
/ ABS(LOOKUP(ZN([Theo Win / DoF Index]), -1))
ELSE NULL
END
```

Conversion Tracking Worksheet



Break Assignments

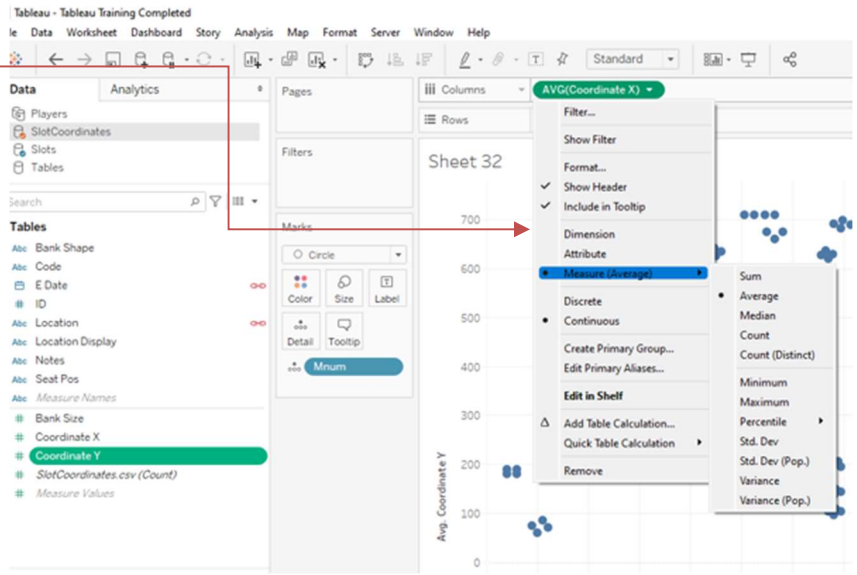
Complete the following on the Break 3 worksheet:

1. Duplicate the **Theo Win / DoF Index** in the CalcMeasures folder. Edit your copy to fix the index to **Ownership** and **Section**.
2. Create **Avg Bet Index** that is fixed to the Site overall Avg Bet
3. Create **Occupancy Index** that is fixed to the Site overall Occupancy
4. Add both of these index measure to the Measure Values shelf

Creating a Slot Floor Map

- Create a new Worksheet with Slots as the primary data source; make SlotCoordinates the secondary data source
- From the **Slots** data source, drag in Location and Mnum (machine number) to Detail in the Marks section
- From the **SlotCoordinates** data source, drag Coordinate X to Columns and Coordinate Y to Rows

- In the Columns and Rows, change Coordinate X and Coordinate Y from Sum to Average
- From **Slots** data source, drag Location to Label



- **Location for Display**

  LEFT([Location],3) + "-" + RIGHT([Location],2)

- Create a new Parameter called "**Select Slot Map Color**" (string, list values = None, Occupancy, Theo / DoF)

- **Slot Map Color**

  CASE[Select Slot Map Color]
  WHEN "None" THEN NULL
  WHEN "Occupancy" THEN [Occupancy]
  WHEN "Theo / Dof" THEN [Theo Win / DoF]
  END

Final Assignments:

1. Edit the Slot Map Color measure to:
   a. Change **Theo Win / DoF** to **Theo Win / DoF Index**
   b. Change **Occupancy** to **Occupancy Index**
   c. Add **Avg Bet** to the parameter and **Avg Bet Index** to the measure
   d. Edit the color scale on the slot map to make the Start value = 0, End value = 3, and Center value = 1

2. Create an interactive Dashboard

   Standard Option: **Slot Floor Map** (you can start on the MapToggle dashboard, but you will need to create a worksheet, name it "BankInfo", that is text table to show each game on a bank and a dashboard action to show the worksheet filtered for just the Bank on hover)

Bank	Position	Game Theme	Theo Win / DoF	Theo Win / DoF Index
110	01	BUFFALO GOLD 360C(ARI)	260	2.4
	02	BUFFALO GOLD 360C(ARI)	229	2.1
	03	MORE MORE CHILLI 800C(ARI)	62	0.6
	04	SHARKNADO 400C(ARI)	78	0.7
Grand Total			157	1.5

   Advanced Option: **Manufacturer Comparison**

   Alternate Advanced Option: **Design your own dashboard useful for slot analysis**

   Reference dashboards:
   https://public.tableau.com/app/profile/william.dunn7551/viz/SlotAnalyticsSamples/0?publish=yes

Tableau Knowledge Base: How to create doughnut charts

Environment
Tableau Desktop

Resolution

Option 1: Use Two Pie Charts
Step 1: Create a pie chart

1. In Tableau Desktop, connect to Superstore sample data.
2. Under **Marks**, select the **Pie** mark type.
3. Drag **Segment** to **Color**.
4. Drag **Sales** to **Angle.**
5. Drag a second copy of **Sales** to **Label**.
6. Resize the pie chart as desired.

Step 2: Switch to the dual-axis chart

1. Select Analysis > Create Calculated Field
2. In the Calculated Field dialog box that opens, do the following, and then click OK:
    1. Name the calculated field. In this example, the calculated field is named "Dummy Axis"
    2. In the formula field, enter the formula: **MIN(1)**
3. Drag **Dummy Axis** to **Rows**.
4. Drag **Dummy Axis** to **Rows** again.
5. On **Rows**, right-click the second instance of **Dummy Axis**, and then check **Dual Axis**.

Step 3: Change the second pie chart to a circle

1. At the bottom of the **Marks** card, click **AGG(Dummy Axis) (2)**.
2. Remove **Segment** from **Color**.
3. Remove **Sales** from **Angle**.
4. Click **Color**, and then choose the same color as the background. In this example, click white.
5. Click **Size**, and then drag the slider to the left to make the circle smaller.
6. Right-click on each of the axes and uncheck **Show Header**.

Option 2: Use One Pie Chart and an Image File
You can also create a pie chart as in Step 1 above, add it to a dashboard, and place a circular .png image over the middle.