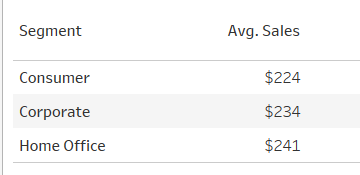
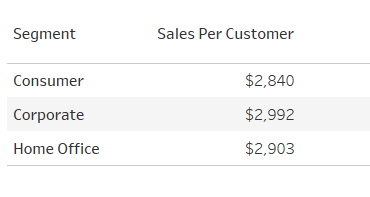
1. LOD Used Cases - <https://www.flerlagetwins.com/2020/02/lod-uses.html>
2. LOD Resources:
   1. <https://www.rigordatasolutions.com/post/how-to-use-exclude-level-of-detail-expression-in-tableau>
   2. <https://www.rigordatasolutions.com/post/how-to-use-include-level-of-detail-expression-in-tableau>
   3. <https://absentdata.com/level-of-detail-calculations/>
3. Tableau formatting - <https://help.tableau.com/current/pro/desktop/en-us/formatting.htm>
4. Tableau summary: <https://www.tutorialspoint.com/tableau/tableau_string_calculations.htm>
5. Tableau Date Functions: <https://www.rigordatasolutions.com/post/tableau-date-functions>

**Level of Detail**

1. There are three main types of LOD expressions.

* FIXED LOD This expression computes values using the specified dimensions without reference to any other dimensions in the view.
* INCLUDE LOD This level of detail expressions compute values using the specified dimensions in addition to whatever dimensions are in the view.
* EXCLUDE LOD These levels of detail expressions subtract dimensions from the view level of detail

**Level of Detail - Include**

1. “INCLUDE” level of detail expression compute aggregations considering dimensions which are specified in the calculation and also take into consideration those dimensions which are present in the view.
2. **Analyses:** Compare the average customer sales across different product segments
3. In this view, if we directly use **“Avg(Sales)” across “Segment”** then it just shows the normal average value of sales taking into consideration all the respective “Order Id’s”  
   
4. But, as the requirement is to show average customer sales across segments, in the view we are going to show only “Segment” but in calculations, we need to include “Customer Name” as well so that average customer sales can be calculated
5. Create calculated field **“Sales per customer” = AVG({INCLUDE [Customer Name] : SUM([Sales])})**
6. In the above formula, we are including the lower level of dimension i.e. Customer Name to get the aggregate (sum) values of Sales at that level and the level in the view (i.e segment) and then we are showing the average at higher level i.e. Segment. Below is the result of using the INCLUDE function  
   
7. You can also replicate the same thing without include function:
   1. **Drag sales to measures** (this will be sum of sales at the view level - in this case , segment level)
   2. Drag calculated field **COUNTD([Customer Name])** to measure. This will be the count of distinct customers who bought in this segment
   3. Drag **calculated field = sum([Sales])/COUNTD([Customer Name])** to measure. This will give the same value as include

**Level of Detail - Exclude**

1. “EXCLUDE” level of detail expression is used to omit specified dimensions from the aggregations. Using “EXCLUDE”, the user can omit the lower level granularity dimension which is present in the view and can directly calculate the value at higher granularity level.
2. “EXCLUDE” level of detail expression is majorly used to calculate ‘difference from overall average’ or ‘percent of total’
3. **Analysis (Percent of total: exclude eg):** Let’s consider the requirement is to compare total sales with the monthly sales of the East region
4. To get the monthly sales we need to have **“Region”, “Category” and “Sum of Sales” in the view.**
5. But, now to get the total sales of Region, we need to exclude “Category” in the calculation so that we can have total sales value for the region
   1. Create a **calculated field: SUM({EXCLUDE [Category] : SUM([Sales])})**
   2. In the above formula, we are excluding the lower level of dimension i.e. Category to get the aggregate (sum) values of Sales at a higher level i.e. Region.
6. **Analysis (difference from overall: exclude eg):** Compute the difference in sales of subcategory “Chairs” to other subcategories
7. Lets show sales by subcategory - **Drag subcategory to rows and sales to columns (bar chart)**
8. Lets compute the sales for sub-category Chairs using the formula below:
   1. Create a **calculated field “Chair Sales”: if [Sub-Category]= 'Chairs' THEN [Sales] END**
   2. Drag this calculated field also to columns
9. Create calculated field **“All chair sales”: { EXCLUDE [Sub-Category]: SUM([Chair sales])}** 
   1. Drag the above calculated field also to columns
10. Now you can **create a calculated field “Final subcatg sales” = ATTR([All chair sales]) - SUM([Sales])**

**Level of Detail - FIXED**

1. “FIXED” level of detail expression aggregates the value only at the dimensions which are specified by the user in the calculation. “FIXED” expression does not take into consideration those dimensions in the view.
2. **Drag region, state to rows and sum(Sales) to measures**
3. **Formula/Expression: SUM({FIXED [Region] : SUM([Sales])})**
4. The above calculation will show the sales value across the region. The objective of using FIXED is that even if a user puts another field say “State” in the view, the above calculation still calculates the value at region level only. It will disregard the other dimensions which are in the view i.e. State

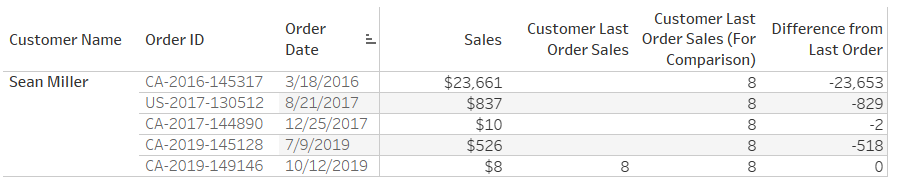
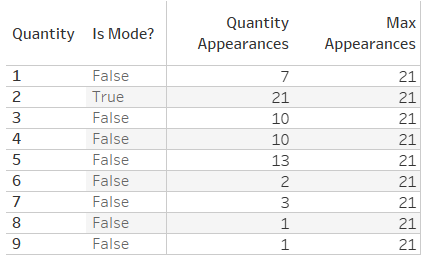
**Level of Detail - Used Cases**

1. **Dealing with duplicate records**
2. let’s say our Superstore People table had 2 salespeople for each region. When you join Orders to People, each order will then be duplicated (in our case, each order will have two records instead of one). If you aggregate a measure, that result will then be twice the value you actually want.
3. 
4. Create calculated field **“Sales LOD” = {FIXED [Row ID]: MAX([Sales])}**
5. 
6. **Get a single aggregate**
7. There are many scenarios where you may want to get an overall min or max value. Let’s say you want to find the maximum sales amount for any order.
8. **{FIXED : MAX([Sales])}**
9. We want to get the overall max across all orders, so we’re not fixing on any specific dimension.
10. **Isolate a specific value**
11. **Analysis:** Assume for each customer , you want the last date in which he/she bought something from technology category
12. **Drag customer name, category, order date to rows. Make order date as exact date and convert to dimension**
13. **Create calculated field “last tech order date” =   
    // Get last date on which customers purchased technology.**
14. **{FIXED [Customer Name]: MAX(**
15. **IF [Category]="Technology" THEN**
16. **[Order Date]**
17. **END**
18. **)}**
19. You can create a similar calculated field using IIF also   
    **{FIXED [Customer Name]: MAX(IIF( [Category]="Technology", [Order Date], NULL))}**
20. **Drag** **“last tech order date” to columns**
21. **IF and IIF**

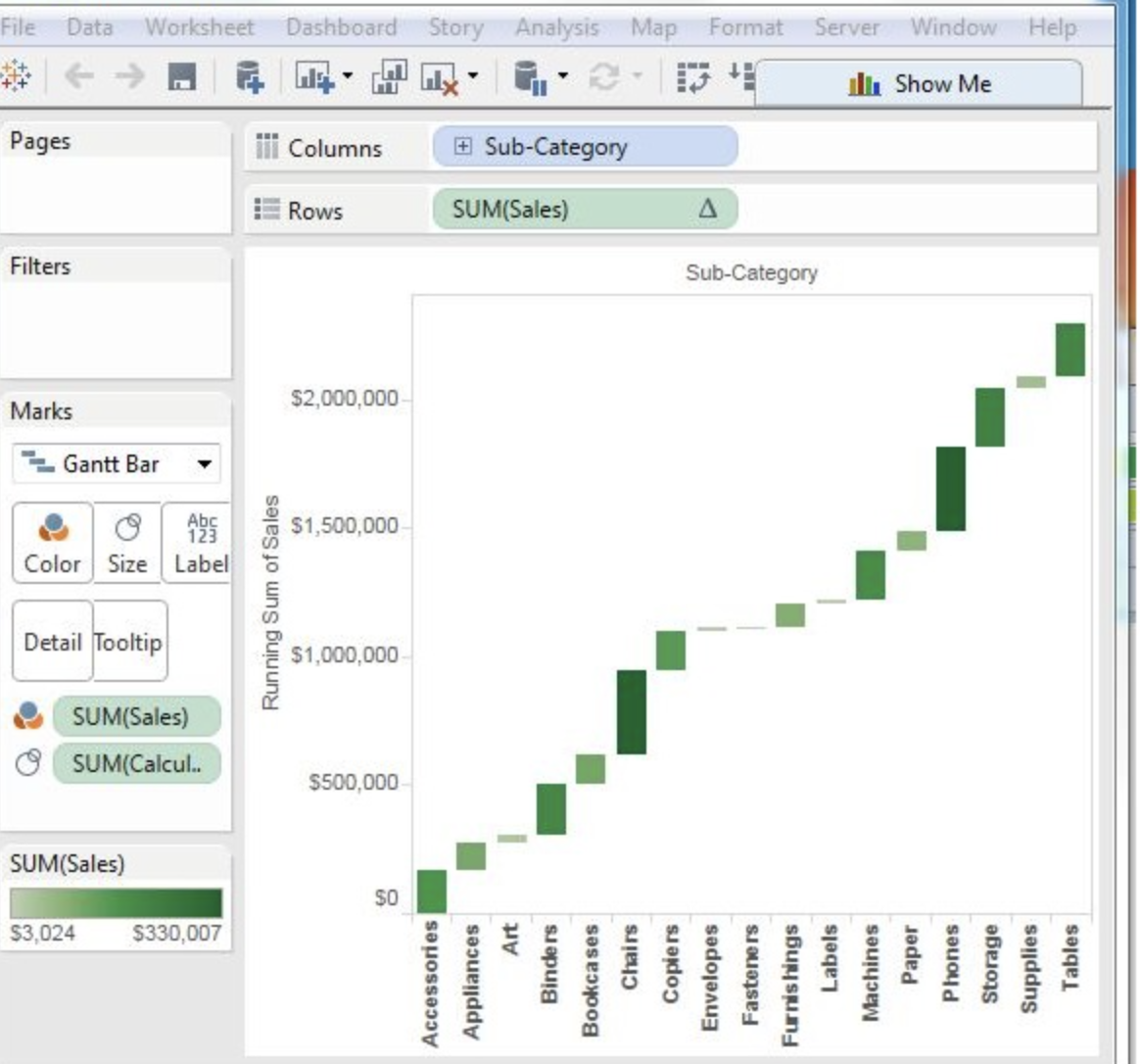
* Both the IF and IIF first check if the test is true; but the IIF then tests if the value is False. IF doesn't test for the False component – it treats everything not True in the same way. IIF handles those items that aren't True or False (or Unknown) differently to IF.
* Although these two logical functions are similar, there are differences. The key difference is the IIF statement handling unknown values.
* The simplest way to explain is by an example. The field [Sales] contains three values: 15; Null; 65. The test is if Sales > 50.
* IIF([Sales]>50,"High","Low","No value")
* The equivalent formula using an IF statement:
* IF [Sales]>50 THEN "High" ELSE "Low" END
* The two If statements return slightly different results:

| Test Value | IIF result | IF result |
| --- | --- | --- |
| 15 | Low | Low |
| Null | No value | Low |
| 65 | High | High |

* Notice the results of the Null sales value.

1. **Get point in time data**
2. **Analyses:** We want to find the sales of Sean Miller’s last order
3. **Drag customer name, order ID, order date (exact date) to rows**
   1. Create calculated field: **Customer Last Order Sales**
   2. **// Get the sales for the customer's last order.**
   3. **IF [Order Date] = {FIXED [Customer Name]: MAX([Order Date])} THEN**
   4. **[Sales]**
   5. **END**
4. The above calculated field will only show sales when order date for the customer is max date , for all other order dates of that customer it will be null
5. To compare each sales amount to the last order’s sales amount. To do this, we’d need to wrap another LOD around this LOD
   1. **Customer Last Order Sales (For Comparison)**
   2. **// Get the sales for the customer's last order and make available for comparisons to other sales orders.**
   3. **{FIXED [Customer Name]: SUM(**
   4. **IF [Order Date] = {FIXED [Customer Name]: MAX([Order Date])} THEN**
   5. **[Sales]**
   6. **END**
   7. **)}**
6. We could then create a simple calculated field to get the difference as shown below.
7. 
8. **Calculate the mode**
9. The mode of a set of numbers is the one that appears most frequently. For example, if you have values 1, 2, 2, 2, 4, 5, 5, 6, 7, then the mode is 2 because it appears more frequently than all the other numbers. A set of numbers can also have multiple modes.
10. **Analysis:**  let’s try to determine the most common quantity of copiers purchased in one order. Each qty should have a flag if its mode or not
11. **Start by filtering my view by the “Copiers” sub-category. I’ll also add this filter to context so that it computes before my LOD** 
    1. **Max Appearances**
    2. **// Get the max appearances of a given quantity.**
    3. **{FIXED : MAX({FIXED [Quantity]: SUM([Number of Records])})}**
12. Check if each quantity is mode or not:
    1. **Is Mode?**
    2. **// Is this number the mode?**
    3. **// Note: Could be multiple modes.**
    4. **IF {FIXED [Quantity]: SUM([Number of Records])} = [Max Appearances] THEN**
    5. **TRUE**
    6. **ELSE**
    7. **FALSE**
    8. **END**
13. 

**Waterfall Charts**

1. 
2. **Drag subcategory to columns -> sales to rows**
3. **Sort the data in ascending order of sales values -> use the sort option appearing in the middle of the vertical axis when you hover the mouse over it**
4. **Right-click on the SUM (Sales) value -> Select the running total from the table calculation option**
5. **Change the chart type to Gantt Bar.**
6. **Create a calculated field named “-sales” and mention the following formula for its value = -[Sales]**
7. **Drag the newly created calculated field (-sales) to the size shelf under Marks Card.**
8. **Drag the newly created calculated field (-sales) to color**

**Tableau Date Functions**

1. Tableau Date Functions: <https://www.rigordatasolutions.com/post/tableau-date-functions>

**Compare prior year data by month (Compare Jan 2021 to Jan 2020 for eg:) using table calculation**

1. **Create calculated field “prior sales” = lookup(sum([Sales]),-1)**
2. **Drag year(order date) to rows -> Drag month(order date) to rows**
3. **Drag “prior sales” to text measures**
4. **Drag sales to text measures**
5. **RC prior sales in measures under marks pane -> edit table calculation -> window opens**
6. **Under “compute using” -> click “specific dimensions”**
7. **Under “At the level” , click “year of order date”**
8. **In the box below “compute using” -> Drag “month of order date” before “year of order date”**
9. **Under “Restarting every” -> click “month of order date”**
10. Now the “prior sales” column gives the sales of the same month last year and “sales” column gives the sales for the month the same year

**Compare prior year data by month (Compare Jan 2021 to Jan 2020 for eg:) using LOD**

1. **Filter for year = 2021**
2. **Drag order date to columns and make it month(order date)**
3. **Difference 2021 to 2020**
   1. **// Difference in 2021 to 2020 sales by month.**
   2. **{FIXED MONTH([Order Date]): SUM(IIF(YEAR([Order Date])=2021, [Sales], NULL))}**
   3. **-**
   4. **{FIXED MONTH([Order Date]): SUM(IIF(YEAR([Order Date])=2020, [Sales], NULL))}**
4. **Drag the above calculated field to rows**
5. **Make it an area chart**

**Based on selected date , calculate MTD and YTD:** [**https://jyoti05iitd.medium.com/mtd-ytd-in-tableau-using-a-reference-date-parameter-50829230a475**](https://jyoti05iitd.medium.com/mtd-ytd-in-tableau-using-a-reference-date-parameter-50829230a475)