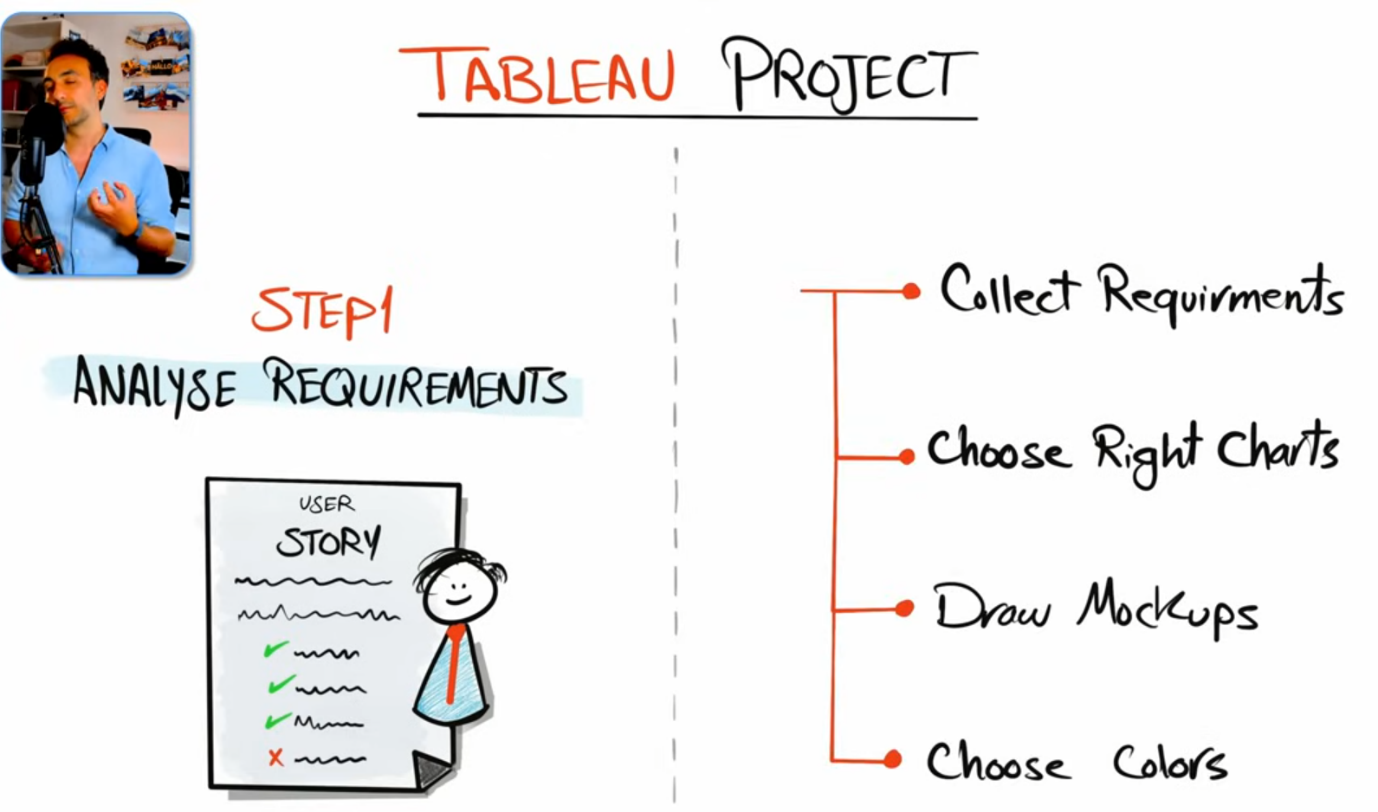
**Data with Baraa**

**Tableau Complete Project End-to-End (Sales Dashboard)**



A diagram of a project

AI-generated content may be incorrect.

A diagram of a graph and chart of a diagram

AI-generated content may be incorrect.

A white board with a diagram

AI-generated content may be incorrect.

A diagram of a project

AI-generated content may be incorrect.

**Step 1: Analyze Requirements**

**Tableau User Story | Sales Performance**

**Introduction**

This user story outlines the specifications for building **two dashboards** using tableau to help stakeholders, including sales managers and executives to analyze **sales performance** and **customers**.

**Sales Dashboard | Requirements**

**Dashboard Purpose**

The purpose of sales dashboard is to present an overview of the sales metrics and trends to analyze year-over-year sales performance and understand sales trends.

**Key Requirements**

KPI Overview

Display a summary of total sales, profits and quantity for the current year and the previous year.

Let’s decide which type of charts that we must present for these requirements, we can go with the BANs. **BANs** are very useful to show the main metrics like the total profit quantity and other metrics, so for this requirement we're going to go and create BANs for it.

Sales Trends

 – Present the data for each KPI (Sales, Profit, Quantity) on a monthly basis for both the current year and the previous year.

 – Identify months with highest and lowest sales and make them easy to recognize.

Here we are talking about change over time right for both the current year and compared to the previous year and as well here they want us to identify the months with the highest and the lowest sales. So we must choose a chart that presents **change over time** and for this you can go with the line charts and precisely we're going to go and use the **Sparkline charts** to highlight the Max and Min values.

Product Subcategory Comparison

 – Compare sales performance by different product subcategories for the current year and the previous year.

 – Include a comparison of sales with profit.

Here we are comparing the members of the subcategories and for that we can use the bar charts and since we have two values the current year and the previous year we can use for example **bar in bar charts** and then for the second point to compare the sales with the profit we can present as well another **bar chart** side by side to the sales in order to show The Profit information.

Weekly Trends for Sales & Profit

 – Present weekly sales and profit data for the current year.

 – Display the average weekly values.

 – Highlight weeks that are above and below the average to draw attention to sales & profit performance.

Here we are talking about change over time because we have the time aspect and we must display the average weekly values as well we must highlight the weeks that are above and below the average to understand the trends in our charts. So here again we are talking about change over time but on the weekly basis, so here we can go as well with the **line chart** to compare the sales and profits.

**Customer Dashboard | Requirements**

**Dashboard Purpose**

The customer dashboard aims to provide an overview of customer data, trends and behaviors. It will help marketing teams and management to understand customer segments and improve customer satisfaction.

**Key Requirements**

KPI Overview

Display a summary of total number of customers, total sales per customer and total number of orders for the current year and the previous year.

Customer Trends

 – Present the data for each KPI on a monthly basis for both the current year and the previous year.

 – Identify months with highest and lowest sales and make them easy to recognize.

The above 3 requirements can be satisfied by using **BANs** and **Sparkline** charts.

Customer Distribution by Number of Orders

Represent the distribution of customers based on the number of orders they have placed to provide insights into customer behavior, loyalty and engagement.

We can use **Histogram** for this requirement.

Top 10 Customers By Profit

 – Present the top 10 customers who have generated the highest profits for the company.

 – Show additional information like rank, number of orders, current sales, current profit and the last order date.

We can use simple table to tabulate this data.

**Design & Interactivity Requirements**

Dashboard Dynamic

 – The Dashboard should allow users to check historical data by offering them the flexibility to select any desired year.

 – Provide users with the ability to navigate between the dashboards easily.

 – Make the charts and graphs interactive, enabling users to filter data using the charts.

The first requirement states the dashboard should be dynamic where the user select the year that they want to compare it with the previous year so it should not be always the last current year and for that we can use **parameters** to solve this task.

The second requirement requires ease of navigation. For that, we usually use **buttons** inside our dashboards to switch back and forth between the dashboards.

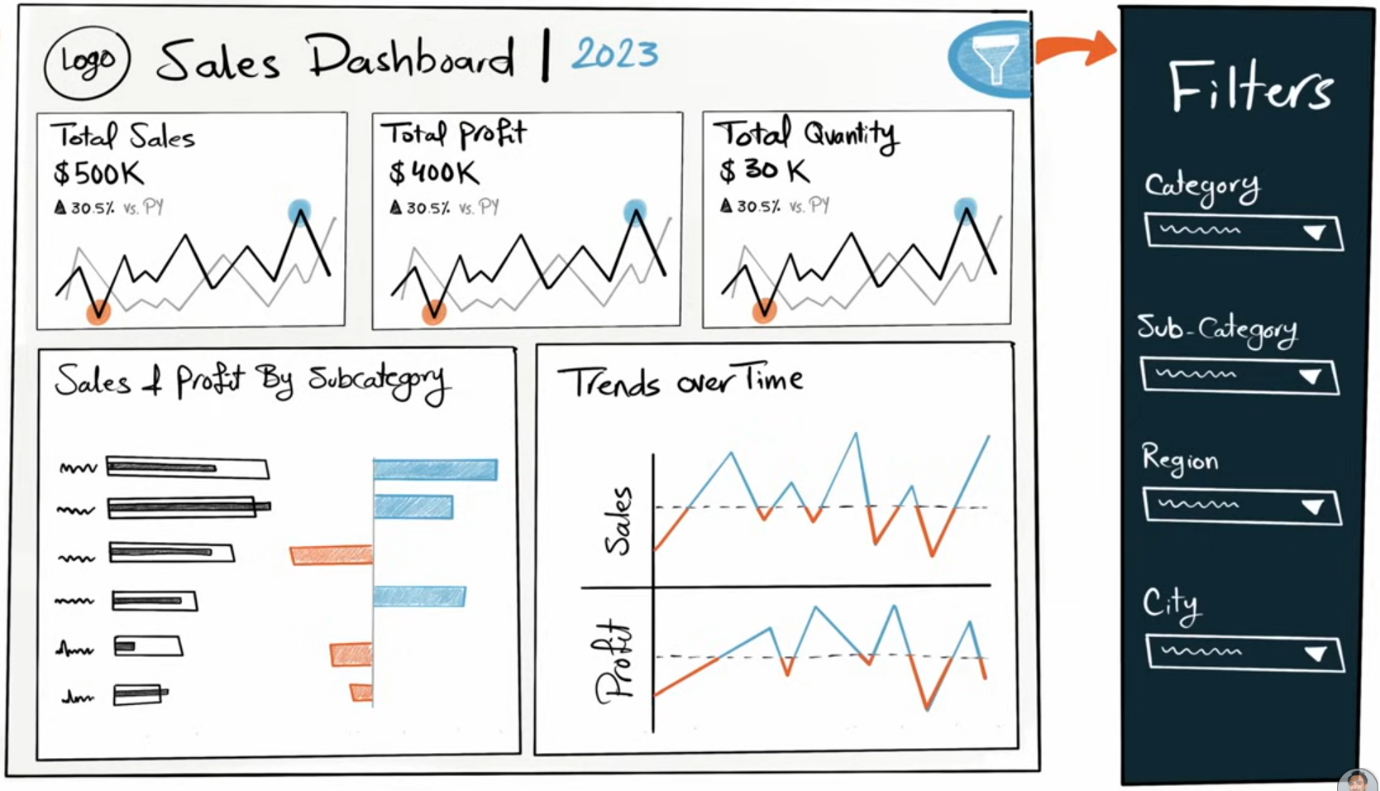
For the third requirement, we can use dashboard **filters**.

Data Filters

Allow users to filter data by product information like category and subcategory and by location information like region, state and city.

This requirement is about data filters. We have to provide all these **filters** inside our dashboard as well.

**Step 1B: Draw Mockups & Choose Colours**



This is how we usually draw a mock-up for a dashboard in Tableau.

**✅ 1. Title Section**

* **Dashboard Title**: *Sales Dashboard*
* **Dynamic Year Display**: e.g., “Current Year: 2023”

**📊 2. KPI Section (Top Pane)**

**Three KPI Blocks**:

* **Total Sales**
* **Total Profit**
* **Total Quantity**

Each block includes:

1. **Main Metric**: Big number (e.g., total sales amount)
2. **YoY Change (%)**: Compared to previous year
3. **Performance Icon**: Up/Down arrow to show positive/negative trend

**📈 3. Monthly Trend Section (Sparklines)**

* Sparkline charts for:
  + **Sales**
  + **Profit**
  + **Quantity**
* Each sparkline includes:
  + **Two lines**: One for current year, one for previous year
  + **Highlighted Min & Max values**: Using circles

**📊 4. Subcategory Comparison Section**

* **Bar-in-Bar Chart** for **Sales**:
  + **Background bar**: Previous year
  + **Foreground bar**: Current year
* **Profit Comparison**:
  + Shown **side-by-side with Sales**
  + Bar chart with **positive and negative values**

**📆 5. Weekly Trend Section**

* Two separate **line charts**:
  + **Sales**
  + **Profit**
* Each includes:
  + **Reference Line**: Showing average value
  + **Color-coded highlights**: Data above/below average

**⚙️ 6. Filter Section**

* **Collapsible Filter Panel** (shown/hidden via icon button)
* Filters include:
  + **Product filters**
  + **Location filters**
  + Other relevant parameters

**🎨 Dashboard Colouring Strategy**

1. **Importance of Early Decision**:
   * Decide on the **colour scheme at the start** of the project.
   * Prevents major rework later.
   * Do it **collaboratively with users** during mockup phase.
2. **Max 4 Colours Recommended**:
   * **2 Basic Colors** (based on Tableau's background):
     + If background is **white**:
       - Use **dark gray** (#303030) and **light gray** (#b3b3b3) for text, borders, and visuals.
   * **2 Custom Colors**:
     + Chosen based on **user preference** or **brand/logo colors**.
3. **Integrated in Mockup**:
   * Coloring is part of the mockup - not just chart types and layout, but also **visual identity**.
4. **Navigation Buttons**:

* Two buttons in the header: **Sales** and **Customers**
* Clicking buttons switches between:
  + **Sales Dashboard**
  + **Customer Dashboard**

**Step 2: Build Data Source**

**A**. From home page, select Connect To a File > Text file > Select the data source you are going to work with.

**B**. To **create data model**, identify **Facts** & **Dimensions**. Always start with the FACT table to build the data model.

**Dimensions**

Definition: Qualitative fields that describe the "who", "what", "where", or "when" of a business process.

Examples:

* Customer Name
* Product Category
* Region
* Order Date

Characteristics:

* Often used for **grouping, filtering, and labeling**
* Typically **textual or categorical**

**Facts**: Facts are event tables which contain IDs, dates and measures

Definition: Quantitative fields that contain measurable, numeric data — the "how much".

Examples:

* Sales Amount
* Profit
* Quantity Ordered
* Discount

Characteristics:

* Typically aggregated (sum, average, etc.)
* Linked to dimensions via keys

Facts are **event-based records** - they capture things that **happen**, like a sale, a shipment, or a customer support call.

**So, a Fact Table typically contains:**

1. **IDs (Foreign Keys)**:
   * Link to dimension tables (e.g., CustomerID, ProductID, DateID)
2. **Dates**:
   * To record when the event happened (e.g., Order Date)
3. **Measures**:
   * Numeric data that you want to analyze (e.g., Sales, Profit, Quantity)

📦 **Example (Fact\_Sales table)**:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **OrderID** | **CustomerID** | **ProductID** | **DateID** | **Sales** | **Quantity** |
| 101 | C123 | P456 | 2023-01-15 | 500.00 | 3 |

* CustomerID, ProductID, DateID → Foreign Keys to Dimensions
* Sales, Quantity → Measures
* Each row = **one sales event**

**🤝 How It Ties to Our Original Definition:**

Both are true:

* Facts = **measures about business events**
* Fact Tables = **tables where each row = an event**, and it includes:
  + **IDs** → Link to dimensions
  + **Dates** → When event occurred
  + **Measures** → What was measured

**🧠 So Why Start with the Fact Table?**

Because it:

* Defines the **business event** you're analyzing
* Tells you **which dimensions** you’ll need to describe the event
* Anchors your **schema and relationships**

The first step is to **drag and drop the fact table** into the data model workspace. This table represents the central business event - for example, sales or transactions.

Once the fact table is in place, we then start **bringing in all the related dimension tables**. Tableau automatically **creates relationships** between the fact and dimension tables based on matching keys.

With this we have our data model where we have one fact and all the dimensions are connected to this fact.

A screenshot of a computer

AI-generated content may be incorrect.

**C**. Rename the table to remove .csv extension.

**D**. Next, we're going to check the data types for the fields whether they are correct or not. Incorrect data types can result in inaccurate visualizations. Make sure that dates and measures in FACTS are not in String data type.

**1. Date Fields Check**

* Example fields: Order Date, Shipping Date
* These fields **must have the data type: *Date***.
* ❗In many real-world datasets, **dates may appear as strings** due to corrupted or inconsistent data.
  + This leads to problems when using date filters, hierarchies, or time-based aggregations.

**🔢 2. Numeric Fields Check**

* All quantitative fields (like Sales, Profit, Quantity) should have the **data type: *Number***.
* Why this matters:
  + Tableau treats **strings as dimensions**, not measures.
  + If a numeric field is a string:
    - You **cannot perform aggregations** like SUM or AVG.
    - Tableau **won’t recognize it as a continuous field**, breaking charts and calculations.

E. Explore and Understand the Data (Before Building Visuals)

**🎯 Why This Step Matters**

Understanding the **structure, relationships, and content** of your data tables helps you:

* Build accurate visuals
* Avoid mistakes in interpretation
* Facilitate better discussions with stakeholders

**✅ Key Actions to Take**

**1. Switch to the Worksheet and Explore Fields**

* **Drag and drop fields** (like Category, Sub-Category, Segment) to view their unique values.
* Helps in identifying:
  + Value ranges
  + Hierarchies
  + Data anomalies or duplicates

**2. Identify Hierarchies and Relationships**

* Example: Category → Sub-Category shows a hierarchy
* If you drag unrelated dimensions together and see duplication or incorrect aggregation, it may **indicate no direct relationship**

**3. Detect Data Quality Issues**

* Example: Seeing duplicate values for a field like Segment when combined with unrelated fields
* Conclusion: Segment might be **related to customers**, not products

**4. Understand Geography**

* Drag fields like Country, Region, State
  + Learn geographic coverage (e.g., only USA)
  + Ensures correct use of maps and filters later

**🧠 Pro Tip:**

**“Browsing the data” is a conversation starter with your data.**  
Use it to gain context, clean expectations, and align your dashboards with the business questions being asked.

**Step 3: Build Charts**

**1.** Build Charts for **each KPI**

**A**. Build Charts

Now, we're going to start with the first chart where we're going to build **BANs**. So the requirement says display a summary of total sales, profits and quantity for the current year and the previous year and let's not forget the requirement that it says the dashboard should allow users to check historical data by offering them the option to select the desired year to be the current year.

**1. Calculated Fields**

* **What it is**: Custom fields created using formulas.
* **Use Cases**:
  + **Conditional KPIs** (e.g., IF [Sales] > [Target] THEN "Above Target"...)
  + For example, for the field **CY Sales**:

IF YEAR([Order Date]) = [Select Year] THEN [Sales]

END

* + **PY Sales**:

IF YEAR([Order Date]) = [Select Year] - 1 THEN [Sales]

END

* + **Date logic** (e.g., YEAR ([Order Date]))
* **Order Date (Year)**:

YEAR([Order Date])

* + **Ratio metrics** (e.g., Profit Ratio = [Profit]/[Sales])
    - **% Diff Sales**:

(SUM([CY Sales]) - SUM([PY Sales]))/(SUM([PY Sales]))

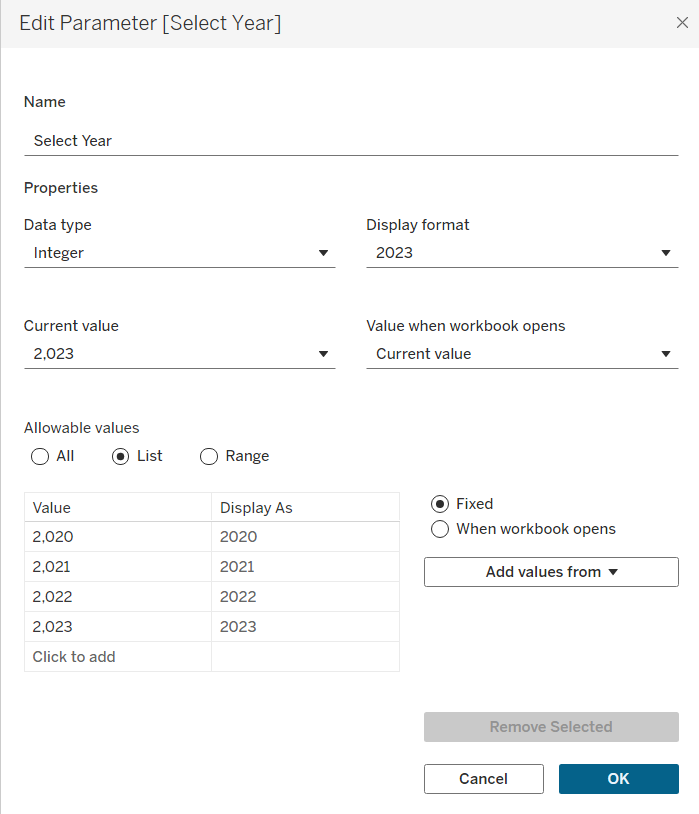
Change the Number Format to “Percentage”.

* **Functions to Know**:
  + IF, CASE, DATEDIFF, DATEPART, ZNULL, aggregation functions (SUM, AVG), etc.

**2. Parameters**

* **What it is**: User-controlled input to dynamically change views.
* **Used for**:
  + Year selection (e.g., Choose 2023 or 2024 to compare KPIs)
  + Dynamic measures/dimensions (switch between Profit/Sales/Quantity)
  + Custom filter or threshold (e.g., Top N customers)

**Select Year**:



**3. KPI Tiles**

* **Design Tips**:
  + Use big numbers with color-coded indicators (green for growth, red for decline).
  + Add icons/arrows for trend direction.
* **Metrics to Show**:
  + Total Sales
  + Profit Ratio
  + YoY % Change
  + Order Count
  + Quantity Sold
* **How to Build**:
  + Drop the fields required for creating BANs in Marks > Details. (For ex, CY Sales and % Diff Sales)
  + BANs: Click on the title and enter name of the measure corresponding BANs and do the required formatting. For eg,

Total Sales

**<SUM({SUM([CY Sales])})>**

<SUM({[% Diff Sales]})> vs. PY

Change the CY Sales Number Format for better readability. Similarly change the % difference format to custom format, to display up/down arrows: ▲ 0.0%; ▼ -0.0%;

* + Click on Show Parameters and verify the BANs for selected year.

**4. Sparklines**

* **What it is**: Tiny line graphs to show trend over time (e.g., monthly sales trend).
* **How to Build**:
  + Add Order Date to Columns and switch it to Months. This gives us the X-axis. Add CY Sales to Rows and switch the MONTH(Order Date) to continuous. Now drop PY Sales to the Y-axis, since we have to compare CY and PY Sales.
  + Create a new Calculated Field for **Min/Max Sales**:

IF SUM([CY Sales]) = WINDOW\_MAX(SUM([CY Sales]))

THEN SUM([CY Sales])

ELSEIF SUM([CY Sales]) = WINDOW\_MIN(SUM([CY Sales]))

THEN SUM([CY Sales])

END

* + **TIP**: Whenever you're building something complex in Tableau, it is recommended to start with a separate table to test and validate the numbers. It's much easier to spot and correct issues in a new table before switching to visual elements like circles or lines.
  + Now drop the Min/Max Sales Field to rows. Set Min/Max to Circle and the first chart to Line. Right click Min/Max dropdown > Select Dual axis > Hide Axis on right side.
  + Continuous measures in VIZ leads to a range of values in title. Change the Measure Values (add {} besides the fields) and update the title correspondingly.
* **Best Practices**:
  + Use tooltips for extra info.
  + Match color to KPI tile (green/red trend).

**B**. Format

1. **Remove Lines & Grids**: Right Click the chart > Format > Format Lines> set Zero lines and Grid Lines to None in Row, Col, Sheet > Set all Sheet Format Borders to None in Sheet.

2. **Clear Axis & Headers**: Remove the axis by deselecting Show Header. Right click Month Axis>Edit Axis>Remove Title. Month Axis>Right click>Format>Abbreviation. Right click Month Axis>Edit Axis>Tick Marks>Fixed.

In dashboards, adding too much information can distract users and prevent them from focusing on what truly matters - identifying trends within the view. To maintain clarity, it's essential to minimize clutter and display only the most relevant information. A minimalist design approach is key.

3. **Colouring**: Add colours by going to Measure Values > Colours > Edit Colours and add colours for CY Sales (#212121) and PY Sales (#cecece) and add them to Custom Colours

4. Now go to Min/Max Sales and drop Min/Max Sales from Rows to Colours of Min/Max Sales by pressing Ctrl > Edit Colours > Custom Diverging > 2 Steps > Max Colour (#1da2d0) & Min Colour (#ff5500). Also add these colours to Custom Colours. Reduce the Opacity of the circles (to 70%).

5. **Tooltips**: Now select Tooltip from All, this opens the Edit Tooltip box. Before moving forward,

1. Create Calculated Fields for “Current Year” and “Previous Year” and change them to dimensions.

2. Drag and drop Current Year, Previous Year, CY Sales, PY Sales, %Diff Sales to All>Tooltip.

Add the following in Tooltip, choose the colours:

Sales of <MONTH(Order Date)>,<ATTR(Current Year)>: <SUM(CY Sales)>

Sales of <MONTH(Order Date)>,<ATTR(Previous Year)>: <SUM(PY Sales)>

Sales Differences: <AGG(% Diff Sales)>

Highest/Lowest Sales: <AGG(Min/Max Sales)>

Set the Number formats of each attribute/field and add tabs to ensure uniformity and readability.

Build BANs for Profits & Quantities following the same steps as we’ve used to build Sales BAN.

**2.** Build the charts about the **subcategories**.

For the subcategory comparison, we’ll compare the current year and the previous year of sales by that subcategory and we'll add the profit information as well. We have decided in the mockup to use **bar in bar charts** for the sales of the current and previous year and as well another separated bar chart for the profits.

**A.** Build Charts

Step 1: Ask ourselves do we have all the data that we need for this charts or do we have to create any calculated Fields.

For these charts, we are safe, we have almost everything. So this is what usually happens in the projects, for the first chart, you're going to end up creating a lot of calculated fields, but as you are creating new views and charts, you will have less need of creating new calculated Fields.

Drop Subcategory (from Products Table) in rows. Drop CY Sales and PY Sales to Columns.

Format the CY Sales Bar Chart. Decrease the size, change the colour to (#212121) and change the type to Bar.

Similarly format the PY Sales Bar Chart.

Now right click CY Sales in columns and select Dual Axis. Synchronise and Hide Axis.

Drop CY Profit to columns.

Step 2: Format

Right click on the chart>Format>Set Format Borders to None & Set zero lines & grid lines to None.

Right click on Axis>Deselect Show headers. Right click Sub category on chart>Hide Field Label for Rows.

Right click any subcategory>Format>Change Font, Size, colour. Increase Bar size of PY Sales and decrease the size of CY Sales, to see the effect of bar in bar chart better. To edit colour of CY Profit, drag and drop CY profit to Colours. Change the colour to 2 step colour as before.

Step 3:

Now, we will add a KPI to indicate where we have an issue, i.e., we’re going to show an icon when the current year is less than the previous year. So we’ll create a KPI. Create a new calculated field, **KPI CY Less PY**:

IF SUM([CY Sales]) < SUM([PY Sales]) THEN '⬤'

ELSE ''

END

Now just add this new calculated field in rows.

Now we’ll sort the data for rank. Right click on sub category in rows>Sort>Field>CY Sales>Descending.

Drag and drop required fields for tooltip and create tooltip.

3. Create a **line chart** to show the **weekly trends for sales and profit** and show the average line for both sales and profits and highlight what is above the line and below the line as well.

Step 1: Do we have all the data that we need to build the chart? Well yes, we have the current year data for the profit and for the sales.

Step 2: Build Charts

Drop Order Date to Columns>Change to Weeks & Continuous.

Drop CY Sales and CY Profits to Rows.

Add reference line (Average line) for both Sales & Profit. Right Click Y axis>Add Reference Line>Label>Custom> “Avg. <Value>”.

You might wonder why we don’t simply merge the two charts using a dual axis. We would consider that if we weren’t using average lines. But in this case, combining them would result in two average lines on the same chart, which would make it difficult to interpret. Since our focus is on the average values, it makes more sense to keep the charts separate to maintain clarity.

Format Lines but not borders (because there are two separate charts). Remove X axis title. Format the reference line (use dashed, opacity 40%. Now format the lines. Go to All>Path>Steps.

In SUM(CY Sales), drop CY Sales to Colours. Create a new calculated field, **KPI Sales Avg**, to check values above and below the average value:

IF SUM([CY Sales]) > WINDOW\_AVG(SUM([CY Sales]))

THEN 'above'

ELSE 'below'

END

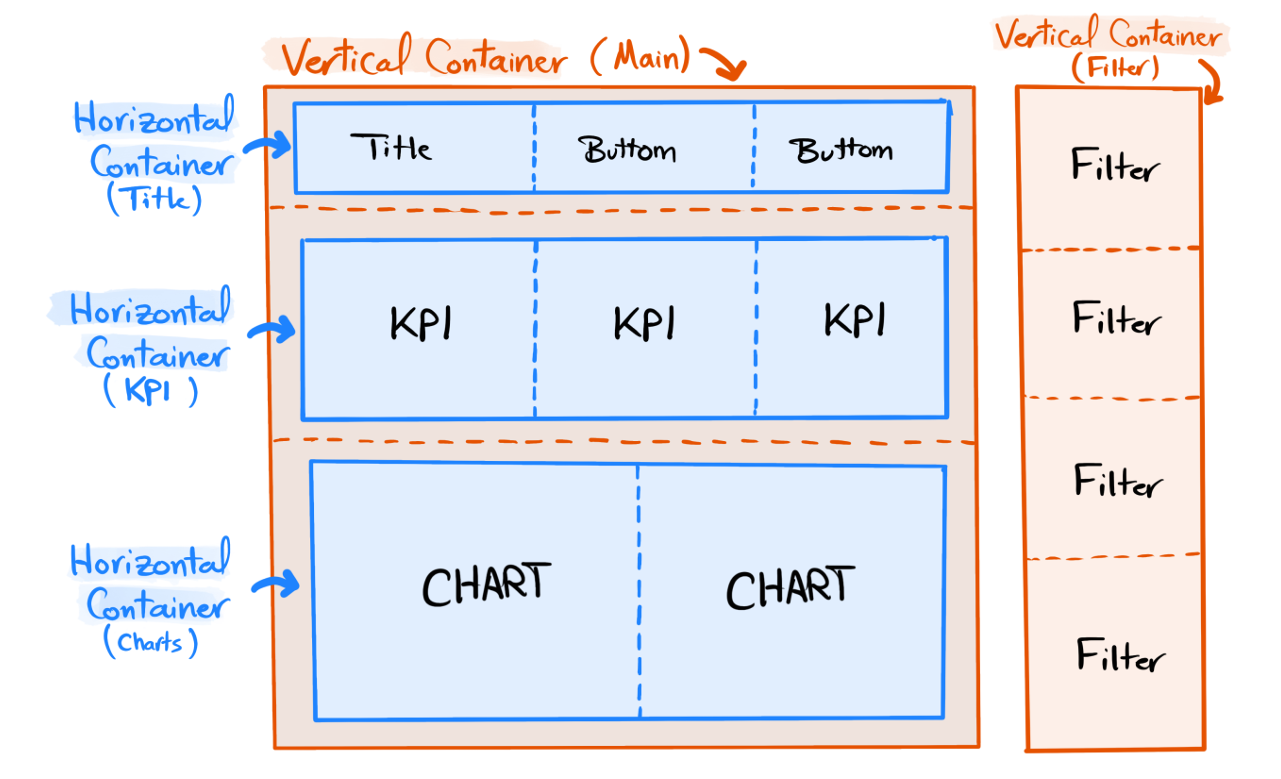
Drop this field to Colours of SUM(CY Sales) and edit colour accordingly.

Similarly, repeat the same for CY Profit.

Create tooltips.

**Step 4: Build Dashboard**

A. Draw Mockup for Container



B.

Create a new **Dashboard**. Fix the size>Fixed Size>1200 x 800 px.

Always start by creating the ‘Floating’ containers before ‘Tiled’ containers.

To start with the floating Container:

1. Take any sheet (like KPI Sales), drag and drop it in the middle. Tableau automatically creates a vertical container on the right side containing parameters, filters, legends, etc. This is the container we’re going to use for the filters. Convert this container to floating element by holding shift and dragging the container by its handle icon. Now remove the chart.

2. Select the container>Layout>Border = line>Select colour, line style, etc. Select background colour.

3. Give a name to the container by editing the name in Item hierarchy.

Now, let’s build the main container for the dashboard.

1. Go to dashboard>drop Vertical Container to the dashboard> Add border and background. Rename it.

2. Add a Blank for placeholder for elements inside the container. For the title bar, add a horizontal container. Set border and background. Rename the container Title. Now to add content, add Text. Add two vertical navigation panes in title for navigation.

3. Drop another horizontal container for KPIs. Add content inside it using Blank. Add 3 Blanks for 3 KPIs.

4. Drop a horizontal container for charts. Set border, background, name. Add 2 Blanks for 2 charts.

Take time designing the containers to create clean and organized dashboards.

Put all charts together.

C. Formatting Dashboard

1. Select the container through layout>Click Distribute contents evenly.

2. Change Title Font/Size. Put the two Navigation buttons in a separate horizontal container and distribute contents evenly. Edit the navigation buttons and navigate to other dashboards accordingly.

3. Click on the Filter container>add Show/Hide button>Hide.

4. Make sure all elements fit “Entire View”.

5. Insert Legend Subcategory and Subcategory Chart in a vertical container. Insert Legend for Profits as text.

6. Add Spaces between the charts (Padding). Go to Dashboard options>Format>Add background lightest shade of grey. For each KPI chart, add top padding = 0, rest all padding = 10. Add inner padding of 5 for each element.

D. Add filters & Dynamic

Go to any Sheet (say, KPI Sales) and Right click Category>Show Filter. Add requested filters: Category, Sub Category, Region, State, City. Right click each filter and select Apply to Worksheets>All using this data source.

Now we will construct the filter on the dashboard. First show the filter. Click on any of the charts>Filter>Select required filters. Select Multiple Values (dropdown) for clarity.

The dashboard dynamic requirement mentions the graphs, enabling users to filter data using the charts. So, select Sales and profit by subcategory and Trends over time and click Use as filter.

Add icons to Filter button. Click and select Edit button. When icon is hidden and shown, choose appropriate icons. Verify functionality by going in Presentation mode. Add the company icon in the title by selecting the image icon and adding the image. Add a floating blank, to test if padding has to be added to any element. To convert Sales Dashboard and Customer Dashboard as buttons to icons, select edit buttons. Choose button style as image and choose the image.

**Making the Customer Dashboard**

Step 1B: In Tableau projects, you should always follow the same mockup and defined design standards.

Create 3 KPIs namely, Customers, Sales per Cust and Orders.

**Customer Distribution by no of orders (Histogram):**

1. Create a Calculated Field, **Orders per Cust**, and convert it to Dimension:

{ FIXED [CY Customers]: COUNTD([CY Orders])}

2. Drop it to Columns and create a new row field, COUNTD([CY Customers]). Convert the marks section to Bars. Drag this field to Colours. Edit the colours, add the borders, drop this field to Labels.

Create a new sheet Top Customers to display top 10 customers.

1. Drop Customer Name to Rows. Drop CY Profit to the table. Drop Customer Name to filters>Top> by field. Format the CY Profit to display complete numbers.

2. Sort the table by profit. Right click Customer Name in Rows>Sort>Field>Desc>CY Profit. Now enter a field ‘INDEX’ to Rows and switch it to discrete. This creates the ranking of customers.

3. Drag and drop CY Sales to the table. Now add COUNTD([CY Orders]) to Measure Values.

4. Drag and drop Order Date to the Rows and select Measures>Maximum, to get the last order date.