Working with Sets & Advanced Visuals

1 — What is a Set?

Definition (Theory):
A set is a custom collection (subset) of rows from your data defined either manually (you pick members) or dynamically (by a rule/condition). Sets are binary: a row is either in the set or not in the set. Sets are powerful because you can use them as dimensions (color, filters, dividing axes), compare set members vs non-members, and drive interactivity.

Why use sets?

- Compare a subset (e.g., top performers) vs the rest
- Create focused analyses (e.g., VIP customers, flagged invoices)
- Make dashboards interactive (click to add/remove members)

Real-time example — Students

- Dataset: Student, RegNo, Tamil, English, Maths, Science, Social
- Fixed set (manual): select students who attended a special coaching program
- Conditional set (dynamic): all students with TotalMarks > 400 automatically belong to the set

How to implement (Tableau Public Web):
If Tableau Web lacks the conditional tab, create a calculated-field flag:
Use this field as color / filter. (On Desktop you can use Create → Set → Condition.)

2 — Conditional Sets (Dynamic Sets)

Definition (Theory):
A conditional set updates automatically: it contains rows that satisfy a logical condition (e.g., Sales > 10000, or TotalMarks > 400). Whenever data changes, the set membership recalculates.

When to use:

- For thresholds (top customers, failing students, low-stock items)
- For monitoring KPIs that change over time

Real-time example — Hotel billing:
Condition: menu items where QuantitySold * Price >= 1000 → mark as "High revenue items"
Use that set to highlight menu items, allocate staff, or promote combos

How to implement (Tableau Public Web):
Create calculated field HighRevenue = `IF [Quantity]*[Price] >= 1000 THEN "High" ELSE "Other" END`
Use this field to color charts or filter to show only high-revenue items

3 — Set Actions (Interactive Sets)

Definition (Theory):
Set actions let end-users modify the content of a set by interacting with a viz (click, select). The set changes dynamically and other views that depend on the set update immediately. This turns sets into a powerful interactive selection mechanism.

Why it's powerful:

- Click points on a map to define a hotspot set
- Click bars to add those bars to a "selected" set and drive downstream visuals

Real-time example — Students:
Dashboard: click a student bar to add them to the "Comparative Group" set; the right-side panel shows only students in that set for detailed drill-down (marks, attendance)

How to implement (Tableau Public Web — limited):
Tableau Public Web supports set actions via Dashboard → Actions → Change Set Values (if available). If not, mimic with parameter or calculated-field toggles:

- Create a set (or boolean field as above)
- Create a dashboard action that changes set membership on Select (if available)
- If Web doesn't allow set actions, use selection to filter or highlight and a parameter to capture selection

4 — Filters

Definition (Theory):
A filter restricts which rows appear in a view. Filters exist at different levels: data source (global), context, worksheet, or dashboard-level filter actions.

Types of filters:

- Dimension filters (categorical selections) — e.g., RoadType, Category
- Measure filters (numeric thresholds) — e.g., TotalMarks >= 400
- Date filters, Top N filters, Context filters (for performance)

Best practices:

- Use context filters to pre-filter big data before expensive calculations
- Use parameter-driven filters when you want a single control to adjust behavior across views
- Keep filters visible on dashboards for user control (Show Filter)

Real-time example — Hotel billing:

- Category filter: checkbox-style category filters (Dosa, Idli, Drinks) let user include/exclude menu categories from the bill
- Price filter: a slider or numeric parameter to show only items under/above a price threshold (e.g., budget items)

How to implement (Tableau Public Web):

- Drag field to Filters shelf → choose values → right-click filter → Show Filter (appears on right)
- On dashboard, click filter dropdown → Apply to Worksheets → All Using This Data Source to make it global

5 — Parameters

Definition (Theory):
A parameter is a single-value control (number, string, date) that you define and can use inside calculated fields, reference lines, or filters. Parameters are not bound to data; they are user-driven inputs.

Common uses:

- Implement Top N (as N), thresholds, price multipliers
- Change measure (toggle between Sales/Profit) using a parameter-driven calculated field
- Scenario simulation (what-if with discount %, tax rate)

Difference from filter: filters choose rows from data, parameters hold a value that you plug into calculations

Real-time example — Students:

- Parameter: Top N integer control. Use it to set how many top students to show (Web may need workaround; see below)
- Parameter: PassingMark = 35. Create calculated field PassFail = `IF [TotalMarks] >= [PassingMark] THEN 'Pass' ELSE 'Fail' END`

How to implement (Tableau Public Web):

- Data pane → Create Parameter → set type & range → right-click → Show Parameter Control
- Create a calculated field that references the parameter and use it in filters/labels

Note: Ranking table calculations (INDEX/RANK) have limited behavior in Web; use parameters with thresholds or manual sort/Keep Only for Top N

6 — Advanced Visuals (Practical Patterns)

Common advanced visuals and when to use them:

- Dual-axis charts (compare two measures, e.g., quantity vs revenue). Use when scales differ
- Small multiples (pane repeat views) — compare same chart across categories
- Bullet charts — compare value to target (e.g., student total vs target 400)
- Heatmaps / density maps — show spatial intensity (accident hotspots, order density)
- Box plots — show distribution and outliers (marks distribution per subject)
- Waterfall chart — show stepwise contributions (components of bill total)

Real-time examples:

- Students: box plot for each subject to spot outliers; bullet chart to show each student's total vs class average
- Hotel billing: dual-axis chart: quantity sold (bars) and revenue (line) per item; treemap for category contribution; donut/pie for category share

Implementation notes:

- Use Marks card to control color, size, label, tooltip
- For spatial density, convert lat/long to geographic role and use Marks → Density or Map Layers
- For small multiples, use Columns and Rows with the same measure split by a dimension (e.g., subject)

Tableau Stories & Dashboards

1 — Dashboards (Theory & Best Practices)

Definition (Theory):
A dashboard is a canvas of multiple sheets assembled to show a coherent view — charts, filters, KPIs, and text. It's an interactive control surface for users.

Design principles:

- Purpose-first: Each dashboard should answer a small set of questions
- Visual hierarchy: Place the most important KPI at top-left or center
- Consistent filters: Use global filters carefully (Apply to Worksheets → All Using This Data Source)

- Interactivity: Use filter actions, highlight actions, URL actions, and set actions as appropriate
- Clarity: Limit 5–7 visual elements; use whitespace and titles
- Performance: Use extracts or context filters for large data. Minimize complex table calcs across many marks

Real-time example — Student Dashboard (layout idea):
Top row: KPIs — Total students, Avg marks, Pass percentage
Left: Bar chart (Total Marks per Student)
Right: Detail table (selected student marks)
Bottom: Subject comparison small multiples and a filter panel

How to implement (Tableau Public Web):

- Create sheets first
- Click New Dashboard
- Drag sheets to the canvas
- From each sheet, open sheet dropdown → Show Filter → the filter appears in dashboard
- Set filter behavior: filter dropdown → Apply to Worksheets → All Using This Data Source
- Add actions: Dashboard → Actions → Add Filter / Highlight / Change Set Values

2 — Story Points (Definition and Use)

Definition (Theory):
A Story in Tableau is a sequence of "story points" — each point is a particular view (sheet or dashboard) with a caption. Stories are used to narrate insights step-by-step (like slides), but remain interactive.

Use cases:

- Present an analysis narrative: Overview → Drill into problem → Recommendation
- Walk stakeholders through decision points
- Provide guided analytics (e.g., show class summary, then focus on failing students, then suggested remedial plan)

Real-time example — Student Story:

- Story Point 1: "Class Overview" — dashboard with KPIs and totals
- Story Point 2: "Top Students" — view showing conditional set of top performers
- Story Point 3: "Subject Weakness" — small multiples revealing one subject's low performance
- Story Point 4: "Action Plan" — charts plus text recommendations

How to implement (Tableau Public Web):

- Click New Story (bottom tab)
- Drag a dashboard or worksheet into the story point

- Add descriptive text in the caption area
- Create next story point → add next view
- Save and publish

3 — Layouts & Responsiveness

Theory:
Dashboards must be designed for the intended consumption device: wide desktop, tablet, or mobile. Tableau supports layouts and device designer (Desktop). In Web, design fixed or tiled layouts carefully.

Guidelines:

- Use tiled layout when you want consistent placement
- Use floating layout for overlays (e.g., KPI boxes)
- Use containers to keep related components together and to maintain responsive resizing

Real-time example — Hotel Billing:

- Desktop: wide layout with billing table and charts side-by-side
- Tablet/mobile: stack elements vertically: KPIs → table → charts

How to implement (Tableau Public Web):

- On the Dashboard canvas, choose Size → Fixed size or Automatic
- Use Objects → Horizontal/Vertical Container to group tiles
- Drag sheets / text into containers
- Use floating objects for sticky controls (e.g., a floating parameter slider)

Practical Mini-Workflows — Quick Recipes (Implementable in Tableau Public Web)

A — Create a Top-Performers Highlight (works in Web)

1. Create Total Marks calculated field
2. Create Top Performer calculated field: `IF [Total Marks] > 400 THEN "Top" ELSE "Other" END`
3. Sheet: Student → Rows, Total Marks → Columns
4. Drag Top Performer → Color
5. Add tooltip and format

6. Dashboard: add sheet; add Top Performer filter (Show Filter). Set Apply to Worksheets → All Using This Data Source

B — Make a parameter-driven Pass Threshold

1. Create parameter PassingMark (Integer, default 35). Show parameter control
2. Calculated field: `PassFail = IF [Total Marks] >= [PassingMark] THEN "Pass" ELSE "Fail" END`
3. Use PassFail as color or as filter. Users can change PassingMark to simulate different passing rules

C — Use Set Action (if available) to build selection-driven detail

1. Create a set on Student (even manual)
2. Dashboard → Actions → Add → Change Set Values. Choose Source sheet and set. On Select, add/remove
3. Make another sheet that shows details where Student is in Set. Add to dashboard. Clicking on bars toggles the detail

D — Build a Story

1. Prepare 3 dashboards: Overview, Deep-dive, Recommendations
2. New Story → drag Overview → add caption. Add New Story Point → drag Deep-dive → caption. Repeat. Publish

Common Pitfalls & How to Avoid Them (Practical)

- Thinking parameter = filter. Parameters are values; use them inside calculations or filters
- Expecting RANK()/INDEX() to be filterable in Web. Use thresholds or manual sorting; desktop has fuller support
- Hiding important filters. Always show key filters and label controls for users
- Overloading dashboards. Keep the canvas focused; use story points for sequential narrative instead of cramming everything

Short Checklist to Practice (Do These in Order)

1. Build Total Marks calculation
2. Create Top Performer flag (calculated field). Color students by it
3. Create a parameter PassingMark and a PassFail calculation; show parameter control
4. Build KPI cards (Total students, Avg marks, # Pass)
5. Build a dashboard; add sheets; show filters; set Apply to Worksheets
6. Add a set and (if available) a Set Action to make selection-driven detail
7. Create a Story with 3 points: Overview → Detail → Recommendation