

Data-to-Deck Macro Framework

This document outlines a complete pipeline for transforming raw tabular data (e.g., CSV or XLSX files) into presentation-ready slides, following a style similar to *The Economist* or other financial press publications.

0. Input recap & goal

Assumed setup

- **Data:** One or more tabular data files (CSV or XLSX format).
- **Tools:**
 - VS Code with Python (Jupyter notebooks available).
 - At least one LLM capable of running code on uploaded files (e.g., ChatGPT with Advanced Data Analysis, or Claude with code execution).
 - *Optional:* A charting/presentation tool such as Flourish, PowerPoint, Google Slides, or Gamma (for creating final visuals).

Goal

Design a manual AI pipeline in which a human orchestrates multiple AI agents and local code to transform:

Raw dataset → 1–2 validated insights → clear chart → styled chart → a single executive slide → editorial polish.

The focus of this framework is on **Step 1: Insight Discovery**, which is broken down into concrete sub-steps that can be delegated to team members.

1. Insight discovery (Step 1 AI)

1.1 Overview of Step 1 sub-steps

Run Step 1 in the following order:

- 1A. Project setup & analysis brief
- 1B. Data audit & cleaning
- 1C. Baseline EDA & hypothesis list
- 1D. Segmentation & effect-size scan
- 1E. Candidate insights drafting & validation

For each sub-step, this guide outlines:

- What to do in VS Code / in project files.

- What to send to ChatGPT or Claude (the prompt content).
- What output to carry forward into the next step.

Use **one continuous conversation thread** in your AI chat (ChatGPT or Claude) for all of Step 1 to ensure context is preserved throughout.

1A. Project setup & analysis brief

Objective

Set up a clean project repository and write a brief so the AI clearly understands the business question and the data structure.

What to do in VS Code / files

1. Create a project folder structure. For example:
2. `data/raw/` : original raw data files (CSV/XLSX).
3. `data/clean/` : cleaned data (to be created later).
4. `data/derived/` : derived tables (to be created later).
5. `notebooks/` : Jupyter notebooks.
6. `docs/` : documentation in Markdown.
7. Create a file `docs/business_brief.md` containing:
 8. The business question and decision to influence.
 9. The time period covered and data grain (what each row represents).
 10. Core KPIs (key performance indicators).
 11. Important segments (e.g., regions, customer segments, channels, products).
 12. Any constraints or context (e.g., "cannot cut prices in DE").
13. Create `docs/data_dictionary.md` with a definition for each column, including:
 14. `name`
 15. `type` (data type)
 16. `description`
 17. `unit` (if applicable)
 18. Example values

No code is required yet. The goal is to produce a clear written brief that any human analyst can understand.

What to send to ChatGPT/Claude

Start a new chat (this will be the Step 1 thread) and paste the following prompt:

PROMPT_1A_BRIEF – BUILD EDA PLAN

You are my senior data analyst.

Here is the BUSINESS BRIEF:
{{contents of business_brief.md}}

Here is the DATA DICTIONARY:
{{contents of data_dictionary.md}}

Task:

- 1) Restate the business problem in your own words.
- 2) List the core KPIs we should focus on.
- 3) Propose a structured EDA & insight-discovery PLAN with sections, in order, including:
 - Data audit & cleaning checks
 - Baseline univariate and time-series analysis
 - Segment comparisons and hypothesis testing
 - Any additional checks needed given this context
- 4) For each section, specify:
 - Questions it answers
 - Tables/plots it expects to produce
 - Which decision levers it informs (pricing, mix, targeting, etc.)

Output the plan as a numbered list. No code yet.

Output to carry forward

- `EDA_PLAN` = the numbered EDA plan returned by the AI. Save this as `docs/eda_plan.md`. This plan will serve as a checklist for sub-steps 1B-1D.

1B. Data audit & cleaning

Objective

Produce a reproducible data audit and a basic data-cleaning script. The AI will write most of the code, but you will run it and maintain ownership of the process.

What to do in VS Code

1. Create a new notebook: `notebooks/01_data_audit.ipynb`.
2. *(Optional)* Quickly open the raw data file to verify basic structure (e.g., number of rows, columns, and column names).

What to send to ChatGPT/Claude

Continue in the same chat thread and send:

PROMPT_1B_AUDIT – DATA AUDIT & CLEANING CODE

We will now implement the "Data audit & cleaning" part of this EDA PLAN:

{{EDA_PLAN section about audit & cleaning}}

Dataset info:

- File path: data/raw/{{your_file_name}}
- High-level structure: {{short recap of rows, columns, size}}

Write Python code (pandas) to:

- 1) Load the dataset.
- 2) Print:
 - head()
 - dtypes
 - number of rows and columns
 - missing value counts per column
 - basic stats for numeric columns
- 3) Check and report:
 - duplicated rows
 - obvious impossible values (based on data dictionary)
 - category level counts for key segment columns
- 4) Propose and IMPLEMENT simple, explicit cleaning steps:
 - Type conversions
 - Obvious outlier handling or flagging (do NOT silently drop large chunks).
 - Standardized category labels where inconsistent.

Important:

- Keep all cleaning steps clearly commented.
- Do NOT guess business logic; when unsure, only flag issues and suggest options in comments.

Output:

- A single, self-contained Python script for these tasks.

Then (locally):

- Copy the AI-generated code into your 01_data_audit.ipynb notebook and run it.
- Address any errors (e.g., incorrect file paths, library imports, typos) and rerun until the audit completes successfully.

How the AI sees the results

After running the notebook, copy the key results back to the chat:

Here are the key results from the audit:

- Shape: 1,200,000 rows, 42 columns
- Missing values (top 10 columns by missingness):
{{paste table}}
- Duplicates: {{count}}
- Notable issues found:
 - {{bullet list from your own notes}}

Then ask the AI:

Given these audit results and your earlier plan, summarize:

- 1) Main data quality risks.
- 2) Confirmed cleaning steps we should keep.
- 3) Any columns we should drop or derive for analysis.

Keep it concise; this will be documented in docs/data_quality_notes.md

Save this summary as docs/data_quality_notes.md.

Output to carry forward

- Cleaned dataset (e.g., data/clean/data_clean.parquet).
- docs/data_quality_notes.md – key data issues and decisions from the audit.

These outputs will serve as inputs to the baseline EDA in the next step.

1C. Baseline EDA & hypothesis list

Objective

Generate baseline exploratory analysis outputs (plots and tables) and compile a list of hypotheses about what might be driving the key KPIs.

What to do in VS Code

- Create a new notebook: notebooks/02_baseline_eda.ipynb.

What to send to ChatGPT/Claude

In the same chat thread, send:

PROMPT_1C_BASELINE – BASELINE EDA & HYPOTHESES

We now have a cleaned dataset at data/clean/data_clean.parquet.

Remember the BUSINESS BRIEF and EDA PLAN:

{{short reminder or paste relevant part of eda_plan.md}}

Here is a short summary of data quality notes:

{{paste key bullets from data_quality_notes.md}}

Task:

1) Propose Python code (pandas + matplotlib/plotly) for a BASELINE EDA that includes:

- Univariate distributions of key KPIs and main drivers.
- Time-series plots of KPIs at the overall level.
- Segment size plots: counts by region/segment/channel/etc.

- 2) Then, based on the expected outputs, propose 10 hypotheses about what might be driving variation in our core KPIs. For each hypothesis, specify:
- KPI metric (with formula if needed).
 - Segment(s) to compare.
 - Time window (if relevant).
 - Why it might matter for the business decision.

Output:

- Python code cells for baseline EDA.
- A table of 10 hypotheses (as text, not code).

Then (locally):

- Copy the AI-proposed code into `02_baseline_eda.ipynb` and run it.
- Export key summary outputs (e.g. segment size tables, overall KPI trends) as CSV files in `data/derived/` or as Markdown files in `docs/` for reference.

How the AI sees the EDA results

Back in the chat, provide a summary of the baseline findings:

Here are key baseline EDA outputs:

1) Overall KPI over time:

{{short textual summary or small table}}

2) Segment sizes:

{{paste condensed table}}

3) Any patterns I noticed:

- {{your bullets}}

Please refine your 10 hypotheses, marking the 5 most promising ones for deeper analysis based on both your earlier reasoning and these baseline patterns.

Save the refined list of hypotheses as `docs/hypotheses.md`.

Output to carry forward

- `docs/hypotheses.md` – approximately 10 hypotheses, with the top 5 flagged as “high priority” for further analysis.

1D. Segmentation & effect-size scan

Objective

Quantify the magnitude and reliability of differences across segments and over time for each high-priority hypothesis.

What to do in VS Code

- Create a new notebook: `notebooks/03_segment_scan.ipynb`.

What to send to ChatGPT/Claude

Continue in the same chat thread and send:

PROMPT_1D_SCAN – SEGMENTATION & EFFECT SIZE

We now want to test our high-priority hypotheses in a systematic way.

Here are the high-priority hypotheses:

{{paste the 5 marked hypotheses from hypotheses.md}}

We have a cleaned dataset at `data/clean/data_clean.parquet`.

Write Python code that:

1) For each high-priority hypothesis:

- Computes the KPI by the relevant segment(s) and overall.
- Creates summary tables with columns:
KPI_name, segment_dimension, segment_value, mean, median, N
- Computes differences and ratios between key segment pairs, producing a "comparison" table with columns:
KPI, segment_dimension, segment_A, segment_B, diff, ratio, N_A, N_B
- For time-based hypotheses, compares pre/post or different regime periods.

2) (Optional) For sufficiently large samples, compute simple statistical tests (e.g., t-test or non-parametric) and include a p-value column.

3) At the end, build a "scoreboard" DataFrame that ranks all segment differences by:

- Absolute difference (or relative ratio).
- Sample size.
- Statistical significance (if available).

Make the code modular and clearly commented.

Then (locally):

- Run the AI-generated code in `03_segment_scan.ipynb`.
- Export two key CSV files to `data/derived/`:
 - `segment_summary.csv` (mean/median values for each segment).
 - `segment_scoreboard.csv` (largest differences, including N and p-values if calculated).

How the AI sees the scan results

Back in the chat, share a summary of the results:

Here are the top rows of segment_scoreboard.csv:
{{paste the top ~20 rows as a markdown table}}

Here are a few observations from me on what looks interesting:
- {{your 3-5 bullet points of notable findings}}

Using only this scoreboard and the earlier business brief, please identify 6-8 candidate insights with:

- KPI metric & formula
- Segments compared
- Time window (if relevant)
- Approximate magnitude (direction and rough size)
- Any robustness concerns

Then assign each a relevance_score (1-5) based on:
effect size, sample size, stability (if observable), and business actionability.

Save the AI's answer as docs/candidate_insights_raw.md.

Output to carry forward

- data/derived/segment_scoreboard.csv – summary of segment differences.
- docs/candidate_insights_raw.md – list of 6-8 candidate insights with their relevance scores.

1E. Candidate insights drafting & validation

Objective

Refine and validate the candidate insights to produce 2-3 final insights suitable for an executive presentation.

What to do in VS Code

- Create a new notebook: notebooks/04_insight_validation.ipynb.
- For each candidate insight with relevance_score ≥ 4:
 - Recompute the KPI differences using clear, explicit code (to independently verify the insight).
 - Create a quick chart (line or bar) that illustrates the pattern or difference.
 - Add a short markdown note on sample sizes, any irregular time periods, and relevant caveats.

What to send to ChatGPT/Claude

In the same chat thread, send the following:

PROMPT_1E_SYNTHESIS – FINAL INSIGHT SYNTHESIS

I have validated the following candidate insights in my own notebook.

Here is a summary of the validated numbers and caveats:

{{for each candidate insight, paste a compact block:

- KPI definition
 - Segments compared
 - Time window
 - Numeric gap / ratio
 - N and any data issues
- }}

Business brief reminder:

{{short restatement of key decisions}}

Task:

From these validated candidates, produce 3-5 INSIGHT DRAFTS with the following structure:

- Insight_label: 1 clear sentence, decision-oriented.
- KPI_definition: formula and units.
- Segments_compared: explicit.
- Time_window: explicit.
- Evidence_summary: direction + magnitude with approximate numbers.
- Robustness_caveats: short and honest.
- Why_hypothesis: potential explanation, clearly marked as hypothesis.
- Relevance_score 1-5: based on effect size, sample size, stability, actionability.

Then, in a final section "Top_insights_to_pursue", pick the best 2-3 and explain why they are most suitable to be visualized and shown to executives.

Final deliverables of Step 1

- docs/final_insights.md – a cleaned-up writeup of the top 2-3 insights.

These will feed into **Step 2: Visual strategy**.

2. Visual strategy (Step 2 AI)

Goal

Determine the best way to visualize the chosen insight. This includes deciding on:

- A primary chart type most appropriate for the insight.
- A backup chart type to use if the data is sparse or messy.
- Any required data transformations (aggregation, indexing, smoothing, scaling) needed for the chart.

Recommended AI type

- A general-purpose LLM (e.g., ChatGPT or Claude), ideally the same session used in Step 1 to preserve context.

Expected INPUT

- `{{CHOSEN_INSIGHT_TEXT}}` (the text of the chosen insight from final_insights.md)
- `{{METRIC_DEFINITION}}`
- `{{SEGMENTS_AND_TIME}}`
- `{{DATA_STRUCTURE}}` (column names and data types)
- `{{AUDIENCE_DESCRIPTION}}`
- `{{DECISION_CONTEXT}}`

Expected OUTPUT

- Primary chart type
- Backup chart type
- Rationale for the recommendation
- Required data transformations

Mini-prompt template: `PROMPT_STEP_2`

PROMPT_STEP_2 – VISUAL STRATEGY FOR A BUSINESS INSIGHT

Context:

You are a data visualization strategist designing a chart for senior decision-makers.

INSIGHT (verbatim from previous step):

`{{CHOSEN_INSIGHT_TEXT}}`

METRIC DEFINITION:

`{{METRIC_DEFINITION}}`

SEGMENTS / TIME STRUCTURE:

`{{SEGMENTS_AND_TIME}}`

DATA STRUCTURE (columns and types):

`{{DATA_STRUCTURE}}`

AUDIENCE & CONTEXT:

Audience: `{{AUDIENCE_DESCRIPTION}}`

Decision: `{{DECISION_CONTEXT}}`

Instructions:

1. Briefly restate the insight, focusing on:
 - What is being compared (segments, channels, regions, tiers).
 - Over which time period (or in which cross-section).
2. Recommend **exactly**:
 - One PRIMARY chart type.
 - One BACKUP chart type (for sparse/messy data).
3. For the PRIMARY chart, specify:
 - Recommended visual (e.g. multi-series line, clustered bar, slope chart,

waterfall).

- X-axis: variable + unit.
- Y-axis: variable + unit.
- Series/color encoding: which variable defines each line/bar/point group.
- Data transformations: aggregations, indexing, normalization, smoothing, scales.
 - What should be visually emphasized (trend break, gap, top/bottom segment, etc.).

4. For the BACKUP chart:

- Type.
- When to prefer it.

Constraints:

- Favor simple, interpretable charts for a non-technical, time-pressed audience.
- Avoid dual axes unless essential and clearly justified.

3. Chart production (Step 3 AI)

Goal

Convert the visual strategy into a concrete chart specification and (optionally) into plotting code.

Recommended AI type

- An LLM with coding capabilities (e.g., ChatGPT with Advanced Data Analysis or Claude's coding mode), *or*
- An LLM that provides a chart specification for manual implementation (e.g., in Flourish or PowerPoint).

Expected INPUT

- `{{DATA_SCHEMA}}` (column names, data types, and example values)
- `{{PRIMARY_CHART_TYPE_AND_REASONING}}`
- `{{DATA_TRANSFORM_INSTRUCTIONS}}`
- `{{CHOSEN_INSIGHT_TEXT}}`
- *(Optional)* `{{PLOTTING_ENVIRONMENT}}` (if asking for code in a specific library)

Expected OUTPUT

- A human-readable chart specification (defining axes, series, filters, transforms, legend, annotations)
- *(Optional)* Plotting code

Mini-prompt template: `PROMPT_STEP_3`

PROMPT_STEP_3 – CHART SPEC + OPTIONAL CODE

Context:

You are a chart-production assistant. Your job is to turn a visual strategy into a precise, implementation-ready chart specification.

DATA SCHEMA:

{{DATA_SCHEMA}}

INSIGHT (this must be obvious in the chart):

{{CHOSEN_INSIGHT_TEXT}}

PRIMARY CHART TYPE AND RATIONALE:

{{PRIMARY_CHART_TYPE_AND_REASONING}}

DATA TRANSFORMATIONS FROM PRIOR STEP:

{{DATA_TRANSFORM_INSTRUCTIONS}}

PLOTTING ENVIRONMENT (optional; adjust if needed):

{{PLOTTING_ENVIRONMENT}}

Instructions:

1. Produce a CLEAR CHART SPECIFICATION as bullet points:

- Chart_title:
- Chart_subtitle (what/where/when):
- X_axis:
 - variable:
 - unit:
 - scale & ordering:
- Y_axis:
 - variable:
 - unit:
 - scale (include zero or not, and why):
- Series_encoding:
 - what defines each line/bar/point group:
 - initial color mapping (neutral):
- Filters:
 - time range:
 - segments included/excluded:
- Data_transformations:
 - aggregations:
 - indexing / normalization:
 - smoothing / rolling averages:
- Legend:
 - items:
 - wording:
- Annotations (at least ONE, initial draft):
 - what to highlight:
 - annotation text:
 - where it sits:

2. Then, optionally, output a GENERIC CODE SNIPPET:

- Use clear variable names consistent with the schema.
- Include comments for each major step (loading data, transforming, plotting, annotating).
- Avoid environment-specific hacks.

Styling guidelines:

- Keep styling neutral at this stage (muted palette, simple gridlines).
- Focus on correctness of data, axes, and annotation placement.

4. Chart styling & attention design (Step 4 AI)

Goal

At this stage, you have a structurally correct chart. This step focuses on:

- Choosing colors and emphasis to draw attention to the key insight.
- Simplifying and organizing the chart elements (gridlines, labels, ordering, background) for clarity and aesthetics.

Recommended AI type

- Continue with the same LLM for design guidance, and use a charting tool (Flourish, PowerPoint, Google Sheets, etc.) to implement the styling changes.

Expected INPUT

- `{{CHART_SPEC}}` (the chart specification from Step 3)
- *(Optional)* A screenshot or textual description of the current chart
- `{{CHOSEN_INSIGHT_TEXT}}`
- `{{AUDIENCE_DESCRIPTION}}`
- `{{CONTEXT_AND_DECISION}}`
- *(Optional)* `{{BRAND_COLORS}}`, `{{ACCESSIBILITY_REQUIREMENTS}}`

Expected OUTPUT

- A styling plan (color palette, what to highlight vs keep as context, labels/ticks, background, gridlines)
- 1–3 suggested visual artifacts (e.g., shaded region for context, callout boxes, highlighted data points)
- A short implementation checklist for applying the style

Mini-prompt template: `PROMPT_STEP_4`

PROMPT_STEP_4 – CHART STYLING & ATTENTION DESIGN

Context:

You are a data-visualization designer. A chart already exists with the correct data and axes. Your job is to guide attention to the core insight through color, contrast, and minimal design changes.

CHART SPEC (from previous step):
{{CHART_SPEC}}

INSIGHT TO HIGHLIGHT:
{{CHOSEN_INSIGHT_TEXT}}

AUDIENCE:
{{AUDIENCE_DESCRIPTION}}

MEETING CONTEXT & DECISION:
{{CONTEXT_AND_DECISION}}

BRAND / ACCESSIBILITY CONSTRAINTS (optional):
{{BRAND_COLORS}}
{{ACCESSIBILITY_REQUIREMENTS}}

Instructions:

1. Identify:

- Focus_series_or_region: which line/bar/segment or time window should stand out.
- Context_elements: which elements should be visible but subdued.

2. Propose a STYLING PLAN:

- Color_palette:
 - Highlight_color(s) for focus.
 - Neutral_colors for others.
 - Any colorblind-safe adjustments.
- Line_and_bar_styles:
 - Line weight, bar opacity, marker use for focus vs context.
- Background_and_grid:
 - Gridlines (on/off, light/dark).
 - Background (light/dark, simple).
- Labels_and_ticks:
 - Which values to label directly on the chart.
 - Tick density and formatting.

3. Suggest 1-3 VISUAL ARTIFACTS:

- Examples: shaded band for a crisis period, circle around an outlier, text box with a key number.
- For each artifact:
 - Location (which point/region).
 - Short text (≤ 15 words).

4. Provide a concise IMPLEMENTATION CHECKLIST that the human can apply in their charting tool (Flourish, PowerPoint, etc.).

Constraints:

- Minimize clutter: no more than 3 non-essential decorative elements.
- Everything added must support the insight.

5. Slide composition (Step 5 AI)

Goal

Design a single presentation slide around the finished chart, following a style similar to *The Economist* or other financial press.

Recommended AI type

- An LLM acting as a slide strategist (e.g., ChatGPT or Claude) to draft content.
- A slide tool (e.g., Gamma, PowerPoint, Google Slides) to implement the final layout.

Expected INPUT

- `{{FINAL_CHART_DESCRIPTION}}` (a description of the final styled chart in words)
- `{{CHOSEN_INSIGHT_TEXT}}`
- `{{AUDIENCE_DESCRIPTION}}`
- `{{CONTEXT_AND_DECISION}}`

Expected OUTPUT

- Slide title and subtitle
- 2–4 supporting bullet points
- Layout description (placement of chart and text)
- A "so-what" statement (key takeaway or implication)

Mini-prompt template: `PROMPT_STEP_5`

PROMPT_STEP_5 – SINGLE-SLIDE BLUEPRINT AROUND THE CHART

Context:

You are a slide strategist for an executive presentation (style inspired by The Economist / financial press).

INSIGHT (core message):

`{{CHOSEN_INSIGHT_TEXT}}`

FINAL CHART (describe content and styling, no code):

`{{FINAL_CHART_DESCRIPTION}}`

AUDIENCE:

`{{AUDIENCE_DESCRIPTION}}`

MEETING CONTEXT & DECISION:

`{{CONTEXT_AND_DECISION}}`

Instructions:

Design ONE slide as a blueprint.

1. Slide_title:

- Insight-first, declarative (not "Analysis of X").

2. Subtitle / context_line:
 - What is shown, where, when (e.g. "Revenue and margin by segment, 2020–2024, EMEA").
3. Supporting_bullets (2–4 bullets, each ≤ 20 words):
 - Short, factual.
 - Each bullet should quantify the pattern or add essential context.
4. Layout_description:
 - Chart_position (e.g. "right half", "full-width top").
 - Bullets_position.
 - Key_numbers / call-outs: where to place 1–2 prominent figures.
 - How to echo chart annotations with text boxes.
 - Footnote_position and suggested text (source, definitions).
5. So_what_statement (1–2 lines max):
 - Explicit link from the pattern to a decision or action.

Constraints:

- No text block longer than 30 words.
- Clear visual hierarchy: Title > chart & key number > bullets > footnote.

6. Presentation-ready refinement (Step 6 AI)

Goal

Polish all text on the slide for maximum clarity, brevity, and executive impact—without altering any facts.

Recommended AI type

- Any advanced editing-focused LLM (e.g., ChatGPT or Claude).

Expected INPUT

- The draft slide copy (title, subtitle, bullets, so-what statement, annotations)
- `{{AUDIENCE_DESCRIPTION}}` (e.g., board, operations team, regulators)

Expected OUTPUT

- Final refined title, subtitle, bullets, so-what, and annotations text
- Three alternative title options (neutral / stronger / provocative but fact-based)
- 1–2 suggestions on tone (style adjustments for the audience)

Mini-prompt template: `PROMPT_STEP_6`

`PROMPT_STEP_6` – EDITORIAL POLISH FOR EXECUTIVE SLIDE

Context:

You are an editor polishing slide text for senior decision-makers.

AUDIENCE:

{{AUDIENCE_DESCRIPTION}}

CURRENT DRAFT:

- Title: {{DRAFT_SLIDE_TITLE}}
- Subtitle: {{DRAFT_SUBTITLE}}
- Bullets:
{{DRAFT_BULLETS}}
- So_what_statement:
{{DRAFT_SO_WHAT}}
- Annotation_texts (if any):
{{DRAFT_ANNOTATIONS}}

Instructions:

1. Tighten all text:
 - Shorten without losing information.
 - Remove redundancy and vague wording.
 - Ensure numbers, directions, and qualifiers remain accurate.
2. Output:
 - Final_title:
 - Final_subtitle:
 - Final_bullets (2-4, each ≤ 18 words):
 - Final_so_what (max 2 lines):
 - Final_annotation_texts (concise and clear):
3. Provide 3 alternative title options:
 - Neutral_title:
 - Stronger_title:
 - Provocative_but_fact_based_title:
4. Tone adjustments (2 brief suggestions):
 - How to adapt for this audience (e.g. emphasize risk vs upside).

Constraints:

- Do NOT change underlying facts or the direction of insights.
- Do NOT invent any numbers.
- Preserve the core insight and its implication for decisions.

7. Pipeline summary tables

7.1 Main steps (1-6)

Step	Goal	Recommended AI type	Human INPUT needed	AI OUTPUT produced	Mini-prompt
1. Insight discovery	From raw tabular data → 1-2 non-obvious, segmented, actionable insights	ChatGPT (Advanced Data Analysis) or Claude with analysis & code execution	Dataset description, schema, sample rows or summary stats, business question, segment dimensions, time span, constraints	Candidate insights with metric definitions, segments, time windows, evidence, caveats, hypotheses; shortlist of top 2-3	PROMPT_1A - PROMPT_1E bundle
2. Visual strategy	Choose best chart type & transforms for a chosen insight and audience	Same LLM as Step 1	Chosen insight text, metric definition, segmentation, data structure, audience & decision context	Primary chart type + backup, rationale, required transforms	PROMPT_STEP_2
3. Chart production	Convert visual strategy into chart spec and optional code	LLM with code (ChatGPT (Advanced Data Analysis) / Claude) or spec for Flourish/PPT	Data schema, chart type choice, transform instructions, insight text, plotting environment	Detailed chart spec (axes, series, filters, transforms, legend, base annotations) plus optional plotting code	PROMPT_STEP_3
4. Chart styling & attention design	Use color, labels, and minimal artifacts to guide attention and improve aesthetics	Same LLM as Steps 1-3; applied in Flourish / PPT / Sheets	Chart spec or description, insight text, audience & context, brand colors / accessibility constraints	Styling plan (palette, emphasis vs context, labels, grid), 1-3 annotations/artifacts, implementation checklist	PROMPT_STEP_4

Step	Goal	Recommended AI type	Human INPUT needed	AI OUTPUT produced	Mini-prompt
5. Slide composition	Design a single slide around the finished chart	LLM slide strategist + Gamma / PPT for layout	Final chart description, insight text, audience description, meeting context & decision	Slide blueprint: title, subtitle, 2-4 bullets, layout guidance, key numbers, footnote text, so-what	PROMPT_STEP_5
6. Presentation-ready refinement	Polish all text for clarity, brevity, and executive impact	Any strong editing LLM	Draft slide content + audience description	Final polished copy, 3 alternative titles, tone-adjustment suggestions	PROMPT_STEP_6

7.2 Step 1 sub-steps (1A–1E)

Step 1 sub-step	Goal	Recommended AI usage	Human INPUT needed	AI OUTPUT produced	Mini-prompt
1A. Setup & brief	Define business problem, KPIs, segments, EDA plan	ChatGPT/ Claude as senior analyst (no code)	business_brief.md , data_dictionary.md	Restated problem, KPIs, numbered EDA_PLAN	PROMPT_1A
1B. Data audit & cleaning	Audit data & create explicit cleaning script	ChatGPT/ Claude writes pandas code; you run in VS Code	Raw data paths, EDA_PLAN section	Python script, list of issues, proposed cleaning steps, summary text	PROMPT_1B
1C. Baseline EDA & hypotheses	Generate base plots & 10 hypotheses	ChatGPT/ Claude writes EDA code & hypotheses	Cleaned data path, EDA_PLAN, data-quality notes	EDA code, 10 hypotheses with 5 marked high-priority	PROMPT_1C
1D. Segment scan & scoreboard	Quantify gaps across segments/ time	ChatGPT/ Claude writes segment/ effect-size code; you run	High-priority hypotheses, cleaned data path	segment_summary.csv , segment_scoreboard.csv , 6–8 candidate insights with scores	PROMPT_1D
1E. Insight synthesis & validation	Produce 2–3 validated, executive-ready insights	ChatGPT/ Claude summarizes; you validate in notebooks	Validated numbers & caveats per candidate insight	3–5 insight drafts; final 2–3 as top insights to pursue	PROMPT_1E

This refined macro framework is ready to be shared, versioned, and used as the reference playbook for your Data-to-Deck pipeline.
