Data-to-Deck Macro Framework

This document describes a full pipeline to turn raw tabular data (CSV/XLSX) into presentation-ready slides in the style of The Economist / financial media.

---

0. Input recap & goal

Assumed setup

- Data: 1+ CSV/XLSX tables.
- Tools:
- VS Code with Python / Jupyter.
- At least one LLM that can run code with file uploads (e.g. ChatGPT ADA, Claude with code).
- Optional: Flourish / PowerPoint / Google Slides / Gamma for charts and slides.

Goal

Design a manual AI pipeline where a human chains several AIs and local code to go from:

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

The focus is on **Step 1: Insight discovery**, which is broken down into concrete sub-steps that can be handed off to a team.

---

1. Insight discovery (Step 1 AI)

1.1 Overview of Step 1 sub-steps

Run Step 1 in this 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 we specify:

- What to do in VS Code / files.
- What to send to ChatGPT/Claude.
- What output to carry forward.

Use **one continuous conversation thread** in ChatGPT/Claude for all of Step 1 so context is preserved.

1A. Project setup & analysis brief

Objective
Create a clean repo and a written brief so the AI understands the business question and the data structure.

What to do in VS Code / files

1. Create a project structure, for example:
2. `data/raw/` → original CSV/XLSX.
3. `data/clean/` → cleaned data (later).
4. `data/derived/` → derived tables (later).
5. `notebooks/` → Jupyter notebooks.
6. `docs/` → markdown docs.
7. Create `docs/business_brief.md` with:
8. Business question and decision to influence.
9. Time period and grain (row = ?).
10. Core KPIs.
11. Important segments (region, segment, channel, product, etc.).
12. Constraints (e.g. "cannot cut prices in DE").
13. Create `docs/data_dictionary.md` with, for each column:
14. `name`
15. `type`
16. `description`
17. `unit`
18. example values

No code is required yet. The goal is a clear written brief a human analyst would understand.

What to send to ChatGPT/Claude

Open a new chat (this will be the Step 1 thread) and paste:

```
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 roadmap the AI returns.
Save it as `docs/eda_plan.md`. This is the checklist for 1B–1D.

---

### 1B. Data audit & cleaning

**Objective**
Obtain a reproducible data audit and basic cleaning script. The AI writes
most of the code; you run and own it.

**What to do in VS Code**

1. Create `notebooks/01_data_audit.ipynb`.
2. Optionally load the raw file once to confirm basic sanity (shape,
columns).

**What to send to ChatGPT/Claude**

In the same chat as 1A, paste:

```text
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).
   - Standardised 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 generated code into `01_data_audit.ipynb` and run it.
- Fix any errors (paths, typos, library imports).

How the AI sees the results

After running the notebook, copy key outputs back into 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:

```
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` with key issues and decisions.

These are inputs to baseline EDA.

1C. Baseline EDA & hypothesis list

Objective
Produce a standard set of plots/tables plus a hypothesis list about what might drive the KPIs.

What to do in VS Code

- Create `notebooks/02_baseline_eda.ipynb`.

What to send to ChatGPT/Claude

In the same chat, paste:

```
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 code to `02_baseline_eda.ipynb` and run it.
- Export key summary tables (e.g. segment sizes, KPI over time) as CSV or Markdown in `data/derived/` or `docs/`.

How the AI sees the EDA results

Back in the chat, send:

```
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 hypotheses as `docs/hypotheses.md`.

Output to carry forward

- `docs/hypotheses.md` with ~10 hypotheses, 5 flagged as “high priority”.

These feed into the systematic scan in 1D.

1D. Segmentation & effect-size scan

Objective
Quantify how large and reliable the differences are across segments and time for the high-priority hypotheses.

What to do in VS Code

- Create `notebooks/03_segment_scan.ipynb`.

What to send to ChatGPT/Claude

In the same chat, paste:

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

2) (Optional) For reasonably 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.
   - If available, statistical significance.

Make code modular and clearly commented.
```

Then (locally):

- Run the generated code in `03_segment_scan.ipynb`.
- Export two key CSVs to `data/derived/`:
- `segment_summary.csv` (means/medians per segment).
- `segment_scoreboard.csv` (largest differences, with N and p-values if used).

How the AI sees the scan results

Back in the chat, send condensed versions:

```
Here are the top rows of segment_scoreboard.csv:
{{paste top 20 rows in markdown table}}

Here is a short note from me on what looks interesting:
- {{your 3–5 bullets}}
```

```
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 + rough size)
- Any robustness concerns

Then rank them with a relevance_score 1–5 based on:
effect size, sample size, stability (if visible), and business actionability.
```

Save the AI's output as `docs/candidate_insights_raw.md`.

Output to carry forward

- `data/derived/segment_scoreboard.csv`.
- `docs/candidate_insights_raw.md` (6–8 candidate insights with relevance scores).

1E. Candidate insights drafting & validation

Objective
Turn the candidate insights into 2–3 final, validated insights suitable for executive presentation.

What to do in VS Code

- Create `notebooks/04_insight_validation.ipynb`.
- For each candidate insight with `relevance_score ≥ 4`:
- Recompute the KPI differences with explicit, clean code.
- Create a quick, simple chart showing the pattern (line/bar).
- Add short markdown notes on sample sizes, odd periods, and caveats.

What to send to ChatGPT/Claude

In the same chat, say:

```
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 visualised and shown to
executives.
```

Final deliverables of Step 1

- `docs/final_insights.md` (cleaned-up version of the 2–3 top insights).

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

---

2. Visual strategy (Step 2 AI)

Goal
Given one chosen insight, decide:

- The primary chart type.
- A backup chart type for messy data.
- Required data transformations.

Recommended AI type

- General LLM (ChatGPT / Claude), ideally the same one used in Step 1.

Expected INPUT

- `{{CHOSEN_INSIGHT_TEXT}}` (verbatim from final_insights.md).
- `{{METRIC_DEFINITION}}` .
- `{{SEGMENTS_AND_TIME}}` .
- `{{DATA_STRUCTURE}}` (column names + types).
- `{{AUDIENCE_DESCRIPTION}}` .
- `{{DECISION_CONTEXT}}` .

Expected OUTPUT

- Primary chart type.
- Backup chart type.
- Rationale.
- Required transforms (aggregation, indexing, smoothing, scales).

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 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-poor audience.
- Avoid dual axes unless essential and clearly justified.
```

3. Chart production (Step 3 AI)

Goal
Turn the visual strategy into a precise chart specification and, optionally, plotting code.

Recommended AI type

- LLM with code execution (ChatGPT ADA / Claude with code), or
- LLM that outputs a spec you implement in Flourish / PPT.

Expected INPUT

- `{{DATA_SCHEMA}}` (names, types, sample values).
- `{{PRIMARY_CHART_TYPE_AND_REASONING}}`.
- `{{DATA_TRANSFORM_INSTRUCTIONS}}`.
- `{{CHOSEN_INSIGHT_TEXT}}`.
- Optional: `{{PLOTTING_ENVIRONMENT}}`.

Expected OUTPUT

- Human-readable chart spec (axes, series, filters, transforms, legend, annotation).
- 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 (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:
   - Clear variable names consistent with the schema.
   - Comments for each major step (load, transform, plot, annotate).
   - 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
You already have a structurally correct chart. This step:

- Chooses colors and emphasis to direct attention to the insight.
- Simplifies and organizes the chart for clarity and aesthetics (gridlines, labels, ordering, background).

Recommended AI type

- Same LLM as before for style guidance, plus
- Chart tool (Flourish / PPT / Sheets / etc.) where you implement the styling.

Expected INPUT

- `{{CHART_SPEC}}` from Step 3.
- Optional: screenshot or textual description of current chart.
- `{{CHOSEN_INSIGHT_TEXT}}`.
- `{{AUDIENCE_DESCRIPTION}}`.
- `{{CONTEXT_AND_DECISION}}`.
- Optional: `{{BRAND_COLORS}}`, `{{ACCESSIBILITY_REQUIREMENTS}}`.

Expected OUTPUT

- Styling plan: palette, emphasis vs context, labels, background, gridlines.
- 1–3 visual artifacts: shaded bands, callout boxes, highlighted points.
- Short implementation checklist.

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 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 slide around the finished chart in a The Economist / financial-press style.

Recommended AI type

- LLM slide strategist (ChatGPT / Claude) for content.
- Gamma or PowerPoint/Slides for final layout implementation.

Expected INPUT

- `{{FINAL_CHART_DESCRIPTION}}` (post-styling chart, described in words).
- `{{CHOSEN_INSIGHT_TEXT}}`.
- `{{AUDIENCE_DESCRIPTION}}`.
- `{{CONTEXT_AND_DECISION}}`.

Expected OUTPUT

- Slide title and subtitle.
- 2–4 supporting bullets.
- Layout description (chart and text placement).
- So-what statement.

Mini-prompt template: `PROMPT_STEP_5`

```
PROMPT_STEP_5 — SINGLE-SLIDE BLUEPRINT AROUND THE CHART

Context:
You are a slide strategist for an executive deck (The Economist / financial
press style).

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 slide text for clarity, brevity and executive impact without changing facts.

Recommended AI type

- Any strong editing LLM (ChatGPT / Claude).

Expected INPUT

- Draft slide copy (title, subtitle, bullets, so-what, annotations).
- `{{AUDIENCE_DESCRIPTION}}` (board vs ops vs regulators).

Expected OUTPUT

- Final title, subtitle, bullets, so-what, annotations.
- Three alternative title options (neutral / stronger / more provocative but factual).
- 1–2 suggestions on tone.

Mini-prompt template: `PROMPT_STEP_6`

```
PROMPT_STEP_6 — EDITORIAL POLISH FOR EXECUTIVE SLIDE

Context:
You are an editorial editor improving slide copy 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.
   - Keep numbers, directions, and qualifiers 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. emphasise risk vs upside).

Constraints:
- Do NOT change underlying facts or directions of effects.
- Do NOT invent numbers.
- Preserve the core insight and decision implication.
```

---

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 ADA or Claude with analysis & code execution	Dataset description, schema, sample rows or summary stats, business question, segment dims, time span, constraints	Candidate insights with metric definitions, segments, time windows, evidence, caveats, hypotheses; shortlist of top 2–3	`PROMPT_1A` – `PROMPT_1E` bundle

Step	Goal	Recommended AI type	Human INPUT needed	AI OUTPUT produced	Mini-prompt
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 ADA / 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`
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-pror
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_`
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_`
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_`
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_`
1E. Insight synthesis & validation	Produce 2–3 validated, executive-ready insights	ChatGPT/ Claude summarises; you validate in notebooks	Validated numbers & caveats per candidate insight	3–5 insight drafts; final 2–3 as `Top_insights_to_pursue`	`PROMPT_`

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