# **1) Vibe Code for Analysis Instructions**

**Choose your path:** **Quant** (country indicators) or **Qual** (text corpus).  
  
Choose your **mode:**

* **A: Meta-dialogue** (maximum freedom, most challenging)
* **B: Prompt template** (structured, intermediate challenge)
* **C: Script remix** — (plug-and-play, easiest)

**Copy code into your environment:**

* For **A/B**: paste generated code into **Colab**, **JDoodle**, or similar.
* For **C**: open the **provided** .ipynb/.py in Colab (no need to generate a new script at first).

**Run → debug → iterate once:** If stuck, reduce scope (fewer columns/topics) and re-prompt or tweak parameters.

**Optional: visualize:** If you get analysis outputs, use the **Generic Viz Meta-Prompt** to get options and code/HTML for a quick viz.

**Keep your takeaway:** Save any outputs and write a 1–2 sentence insight.

**Minimum data**

* **Quant CSV:** Country + 2–4 numeric measures (e.g., LifeSatisfaction2018, LifeExpectancy2018, GDPPerCapita2018).
* **Qual CSV:** country, text.

# 

# **2) Prompt Templates (Copy/Paste)**

## **Quantitative**

### **A) Meta-Dialogue (max freedom)**

# Part 1 — Meta

Given my construct [X], measures [Y1, Y2, …], and question [Z],

which procedure should I run (e.g., summary+correlation, regression, clustering, PCA)?

List 2–3 concrete steps and provide runnable Python code.

# Part 2 — Code

Write Python that loads a CSV (URL or uploaded file path).

Assume a 'Country' column plus the measures I listed.

Use pandas/numpy (and scikit-learn only if the method needs it).

Include clear placeholders for column names I can edit.

### **B) Prompt Template (structured request)**

# Quant Template

Context

- Construct: [YOUR CONSTRUCT]

- Country column: [Country]

- Measures (2–4): [M1, M2, M3, M4]

- Optional filter: [ALL or list of countries/regions]

- Method (pick one): [summary+correlation | regression | clustering | pca]

- Regression target (only if regression): [TARGET or N/A]

Task

1) Load my CSV (URL or uploaded path); print df.shape and df.head(3).

2) Run the selected method and print interpretable results:

- summary+correlation → summary table + correlation matrix

- regression → coefficients, intercept, R² + brief interpretation

- clustering → KMeans with a sensible k (e.g., 3–6) + brief interpretation

- pca → explained variance + loadings + brief interpretation

3) Prepare small, in-memory results for optional viz:

- Create DataFrame result\_scatter with columns: Country, x, y [, group]

• regression: choose two explanatory measures for x,y

• clustering: two features as x,y + cluster as group

• pca: PC1→x, PC2→y (also return loadings)

4) Print 1–2 key findings in plain English.

Please return runnable Python with light comments and clearly labeled edit points.

### **C) Script Template (remixing code)**

Find the quant script file on Google Drive or Github

## **Qualitative / NLP**

### **A) Meta-Dialogue (max freedom)**

# Part 1 — Meta

Given my construct [X], corpus type, and question [Z],

which procedure(s) should I run: word frequency, simple sentiment, or semantic/BERT modeling?

List 2–3 concrete steps and provide runnable Python code.

# Part 2 — Code

Write Python that loads a CSV (URL or uploaded file path) with columns doc\_id, text.

Use pandas (and sentence-transformers/scikit-learn if the method needs it).

Include clear placeholders I can edit.

### **B) Prompt Template (structured request)**

# Qual Template

Context

- Columns: doc\_id, text

- Corpus type (brief): [e.g., short descriptions, interviews]

- Goal/question: [YOUR QUESTION]

- Method (choose 1–2, in order):

[ word-frequency | simple-sentiment | semantic-embeddings ]

Task

1) Load the CSV; print df.shape and df.head(3).

2) If "word-frequency":

- Clean/tokenize, remove stopwords

- Produce top-N unigrams and bigrams (two small DataFrames)

3) If "simple-sentiment":

- Compute a quick lexicon-style polarity per document

- Return DataFrame with: doc\_id, sentiment

- Note one sentence on limitations

4) If "semantic-embeddings":

- Use a lightweight model (e.g., sentence-transformers "all-MiniLM-L6-v2"); if unavailable, fall back to TF-IDF + cosine

- Choose ONE exploration path and implement it cleanly:

• KMeans clusters with 1–3 representative snippets per cluster, or

• Nearest-neighbor search with 2–3 exemplar queries, or

• 2D projection (PCA/UMAP) producing a small table (doc\_id, x, y, label)

5) Prepare compact results (tables/lists) suitable for quick visualization.

6) Print a short interpretation connecting results to the goal/question.

Please return runnable Python with light comments and clearly labeled edit points.

### **C) Script Template (remixing code)**

Find the qual script file on Google Drive or Github

## **Generic Viz Meta-Prompt (open-ended; any path)**

Context

- Question: [YOUR QUESTION]

- Procedure I ran: [ANALYSIS PROCEDURE]

- What I have now (outputs): [names + key columns/fields of DataFrame(s)/objects]

Task

1) Propose 2–3 visualization concepts that clearly communicate my findings.

2) Pick the strongest concept and generate ONE implementation:

- Either a self-contained HTML I can save and open, OR

- A Python cell that runs in my environment with common libs.

3) Add brief usage notes: what to look for and how to adapt column names.

Keep dependencies light and include a quick fallback if a library is missing.

Return only the code and the short usage notes.

## **Troubleshooting & Debugging**

**Quick fixes**

* **Ask the LLM** → copy your error message into your conversation box. Conversely, screenshot an output (such as a sub-par chart) and describe what needs to be fixed.
* **Import/library errors** → Ask for a variant using only pandas/numpy (and add sentence-transformers/scikit-learn only if you chose semantic or advanced methods).
* **Data not loading** → Try a direct CSV URL vs. file upload; confirm column names with df.head().
* **Slow/heavy** → Use fewer columns, smaller k, or fewer sample rows/snippets.
* **Viz unclear** → Ask for axis labels, tooltips, and a small preview table of what’s plotted.