**Documentation: Generating Data with ChatGPT**

This documentation details the process of using ChatGPT to generate, manipulate, and export structured spreadsheet data. The examples provided highlight ChatGPT’s ability to handle iterative instructions, complex constraints, and technical challenges. Below is a structured breakdown of key functionalities, workflows, and lessons learned from nine conversational examples.

**Table of Contents**

1. **Overview of ChatGPT’s Data Generation Capabilities**
2. **Key Functionalities Demonstrated**
3. **Workflow Breakdown**
4. **Handling Technical Challenges**
5. **Best Practices**
6. **Lessons Learned**

**1. Overview of ChatGPT’s Data Generation Capabilities**

ChatGPT excels at automating spreadsheet tasks, including:

* **Dynamic Data Generation:** Creating columns with custom logic (e.g., probabilities, formulas, or text concatenation).
* **File Management:** Splitting large datasets into multiple files based on row limits (e.g., 100,000 rows per file).
* **Data Restructuring:** Adjusting data formats (e.g., moving multi-goal cells to individual rows).
* **Relational Data Handling:** Generating unique combinations of IDs across tables.
* **Error Detection:** Correcting inconsistencies in user-provided probabilities or constraints.

**2. Key Functionalities Demonstrated**

**A. Generating Columns with Custom Logic**

**Example Tasks:**

* **Course Goals:** Adding a "Course Goals" column with 3 goals per course, later split into individual rows.
* **Payment Methods:** Assigning payment types (e.g., "CREDIT\_CARD: 21%") with probabilistic distribution.
* **Quiz Questions:** Creating question types (True/False, Multiple Choice) with answer columns conditionally populated.

**Process:**

1. **User Instruction:** Define column names, data types, and rules (e.g., "VideoUrl if Type=Video").
2. **ChatGPT Execution:** Generates data programmatically, validates constraints, and confirms completion.
3. **Output:** Downloadable Excel/CSV file with structured data.

**B. File Size Management**

**Example:** Splitting datasets exceeding 100,000 rows into multiple files (e.g., output\_part\_1.xlsx, output\_part\_2.xlsx).

**Workflow:**

1. **User Constraint:** Specify a row limit (e.g., "if records exceed 100K, create another file").
2. **ChatGPT Action:** Generates data, monitors row count, and splits files automatically.
3. **Output:** Multiple downloadable links for segmented datasets.

**C. Iterative Data Refinement**

**Example:** Adjusting from multi-goal cells to individual rows.

1. **Initial Output:** Goals stored in one cell (e.g., "Goal1, Goal2, Goal3").
2. **User Refinement:** "Make every goal in a row."
3. **ChatGPT Adjustment:** Expands rows to separate goals, ensuring unique combinations.

**D. Relational Data Generation**

**Example:** Linking "Order Id" and "Course Id" from separate tables.

1. **User Instruction:** Repeat "Order Id" based on "TotalAmount" and pair with unique "Course Id."
2. **ChatGPT Execution:** Combines data from multiple tables, enforces unique combinations, and splits files if needed.

**E. Error Detection & Correction**

**Example:** Social media probability inconsistency (probabilities summed to 142%).

1. **ChatGPT Detection:** Flags invalid probability totals.
2. **Auto-Correction:** Normalizes probabilities to 100% before generating data.

**3. Workflow Breakdown**

**Step 1: Upload & Context Setup**

* **Upload File:** Provide a spreadsheet (e.g., cr7.xlsx) to establish context.
* **Define Instructions:** Specify columns, rules, and constraints (e.g., "Status column: 85% Completed").

**Step 2: ChatGPT Processing**

* **File Inspection:** ChatGPT analyzes columns (e.g., "Id, Title, Duration").
* **Logic Confirmation:** Restates instructions to ensure alignment (e.g., "Duration divided by lessons").
* **Execution:** Generates data programmatically; handles errors proactively.

**Step 3: Output & Iteration**

* **Download Links:** Generated files provided as clickable links (e.g., courses\_with\_goals.xlsx).
* **User Feedback:** Refine instructions (e.g., "Separate goals into rows").
* **Regeneration:** ChatGPT adjusts data structure and provides updated files.

**4. Handling Technical Challenges**

**A. Environment Resets**

* **Issue:** Uploaded files lost due to session timeouts or resets.
* **Mitigation:**
  + Re-upload files when prompted.
  + ChatGPT regenerates data from scratch using prior instructions.

**B. Connection Issues**

* **Example:** "Analysis paused" due to connectivity problems.
* **ChatGPT Response:** Retries data generation automatically.

**C. Large Datasets**

* **Risk:** Timeouts or performance lag.
* **Solution:** Split outputs into smaller files using row limits.

**5. Best Practices**

1. **Specificity:** Clearly define column names, data types, and constraints.
   * *Example:* "Discount: 75% values = 38; others = random(33-79)."
2. **Anticipate Splits:** Set row limits upfront for large datasets.
3. **Validate Outputs:** Check generated files for accuracy, especially after iterations.
4. **Use Probabilities Carefully:** Ensure totals do not exceed 100%.

**6. Lessons Learned**

* **Iterative Design:** Start with a small dataset, refine instructions, then scale.
* **ChatGPT’s Strengths:**
  + Rapid prototyping of structured data.
  + Adapting to feedback (e.g., restructuring rows).
  + Error checking (e.g., probability normalization).
* **Limitations:**
  + File uploads may require re-submission after resets.
  + Complex tasks may require multiple iterations.

**Conclusion**

ChatGPT is a powerful tool for automating spreadsheet tasks, from simple column generation to managing relational data with complex constraints. By leveraging its ability to process iterative feedback and detect errors, users can efficiently generate, refine, and export high-quality datasets. The examples provided demonstrate both its versatility and the importance of clear communication in achieving desired outcomes.

**Example:**

