Rubric Tasks Overview

Note: These are general guidelines for rubric writing that apply to all rubrics tasks.

A rubric task usually consists of a prompt and a rubric.

➡ The prompt is a question.
➡ The rubric provides objective structure to evaluate what makes a good response to this prompt.

What is a Rubric?

The rubric is a list of **specific, simple**, and **declarative** criteria that can be evaluated as **true** or **false** for **any** given response to the prompt.

The criteria, when taken as a whole, **should be able to evaluate any possible response** and rank different responses.

Rubric isn't a completely new concept; it's essentially an extension of RLHF preference ranking, where rubrics are added to provide a quantitative measure for the rankings.

Here is an example:

(please note that the example below is for illustrating the concept of rubric. This prompt and rubric is far too easy)

Task Prompt: What's a fun indoor activity for kids aged 3–5 on a rainy day? **Rubric**: ● [+5] The response suggests an indoor activity appropriate for 3–5 year-olds (e.g. building with blocks, indoor obstacle course, story time) ● [+3] The response mentions how the activity helps with development (e.g. fine motor skills, creativity, social interaction) ● [+1] The response acknowledges that the activity is suitable for a rainy day (e.g. "indoors," "rainy day," "when it's wet outside")	
Task Response 1: Try building a blanket fort together! Kids can help arrange cushions and	✅[+5] The response suggests an appropriate indoor activity (building a blanket fort and story time) ❎[+3] It doesn't mention how the activity helps with development Street ✅[+1] It implies suitability for a rainy day by mentioning "a cozy story time inside"

blankets, then enjoy a cozy story time inside.	Total: 6 points – A good answer with a strong suggestion and some context, but it could be more informative.
Task Response 2: Finger painting is a great activity for kids ages 3–5. It encourages creativity and helps develop fine motor skills.	✅[+5] Suggests an age-appropriate indoor activity (finger painting) ✅[+3] Mentions how it supports development (creativity, fine motor skills) ❎[+1] Doesn't mention anything about being indoors or rainy days Total: 8 points – A strong answer that hits key elements, but lacks the situational context.
Task Response 3: When it's raining outside, try setting up an indoor obstacle course using pillows, chairs, and boxes. It keeps kids 3–5 physically active and helps with balance and coordination.	✅[+5] Suggests an appropriate indoor activity (indoor obstacle course) ✅[+3] Mentions developmental benefits (balance and coordination) ✅[+1] Acknowledges the rainy day context ("when it's raining outside," "indoor obstacle course") Total: 9 points – The most complete and well-rounded response.

1. Writing Good Rubrics

General Principle

- **MECE: Mutually Exclusive, Collectively Exhaustive**
  - **Completeness:** Consider **all** the elements you would want to include to create a perfect response and put them into the rubric. This means including not only the facts and statements directly requested by the prompt, but also the supporting details that provide justification, reasoning, and logic for your response. Each of these elements should have a criterion because each criterion helps to develop the answer to the question from a slightly different angle.
  - **No overlapping**: the same error from a model shouldn't be punished multiple times.
- **Diversity**
  - The rubric items should include variable types of information. If all criteria are like "the response mentions A", "the response mentions B", then this is not a good rubric.
- **How many rubric items for each prompt**
  - As many as needed. There is no golden standard, and the desired number of rubrics varies by accounts and task types. 10-30 is a good range, but there is no strict limit. The principle here is to write rubrics that cover all aspects of an ideal response.

        - In general, tasks that can be fully evaluated with less than 10 rubric items are not complicated enough. In such cases, we should think about whether the prompt is difficult enough rather than blindly adding more rubrics.

- **How many rubric items to fail**
    - A good rule of thumb is that the model fails on 50% of rubrics items, otherwise the task might be too easy.

Atomicity / Non-stacked

- Each rubric criterion should evaluate exactly one distinct aspect. Avoid bundling multiple criteria into a single rubric. Most stacked criteria with the word “and” can be broken up into multiple pieces.

❎ Response identifies George Washington as the first U.S. president **and** mentions he served two terms.
✅ Response identifies George Washington as the first U.S. president. ✅ Response mentions that George Washington served two terms.

Specificity

- Criteria should be binary (true or false) and objective.
- Avoid vague descriptions (e.g., "the response must be accurate" is vague).
- Define precisely what is expected.
    - Example: "The response should list exactly three examples."

Self-contained

- Each criterion should contain all the information needed to evaluate a response, e.g.

❎ Mentions the capital city of Canada.
✅ Mentions the capital city of Canada is Ottawa.

- Criterion should be verifiable without requiring external search.

❎ Response names any of the Nobel Prize winners in Physics in 2023
✅ Response names any of the following Nobel Prize winners in Physics in 2023: Pierre Agostini, Ferenc Krausz, or Anne L’Huillier.

2. Classifying Rubrics

Objective vs Subjective

- **Objective Rubric**: Evaluation criteria directly link to explicit instructions or measurable conditions. Evaluators consistently arrive at the same conclusion.
    - Example: "The response must contain exactly five items."
- **Subjective Rubric**: Evaluation criteria require evaluator judgment, inference, or interpretation. Evaluations might vary slightly between evaluators.
    - Example: "The response tone should feel professional and polite."

Explicit vs Implicit

- **Explicit Rubric**: Criteria explicitly stated within the prompt or instructions.
    - Example: Prompt explicitly asks for a table, the rubric verifies table formatting.
- **Implicit Rubric**: Criteria inferred from cultural norms, language conventions, or general expectations, not directly stated in the prompt.
    - Example: Correct grammar and punctuation are typically implicit unless explicitly required otherwise.

3. Common Pitfalls and How to Avoid Them

Pitfall: Response ranking doesn't truly reflect response quality

- We derived the response ranking and ranking based on the rubrics outcome. Given that we haven’t figured out how to write good rubrics, and our rubrics do not fully capture what makes a good response, the rating and rankings derived from the rubrics are off.
- How to solve: rank the responses first, and then finalize rubric and rating with side by side view with the response raking to ensure alignment.

Pitfall: Response ranking doesn't match to the rubrics ratings

Source: Redacted302 EN rubrics quality assessment

- From the rubrics ratings it’s not obvious why the chosen response is the best response.
- How to solve:
    - side by side view of response ranking and the rubric ratings to ensure alignment.
    - Potentially capture the implicit weighing of rubrics that contributors use to choose the best response

Pitfall: Overly Complex or Ambiguous Criteria

- Keep each criterion simple, atomic, and verifiable.
- Provide examples for evaluators to minimize subjective interpretation.

Pitfall: Subjectivity Confusion

- Provide clear examples and clarifications for subjective rubrics.

- Consider breaking subjective criteria into explicit, objective sub-components when possible.

Pitfall: Too few or too many criteria

- Rubrics should be designed in such a way that a model response that satisfies all of the criteria is close to a perfect response. Ensuring an appropriate number of criteria is essential for this

More Examples of Rubrics

Note that these first two examples come from a different project, but they can still help you understand ***the extent of a good rubric coverage*** *for a specific prompt given to a model.*

Question / Prompt	Example Rubric
1. How is the implementation of `skipna` different in Python vs. Cython modules?	1. Does the response discuss that Python is more readable whereas Cython is more performant? 2. Does the response discuss that Cython is more performant because it avoids Python's interpreter overhead? 3. Does the response discuss how Cython is compiled to C which enables higher performance? 4. Does the response mention that proper implementation of the `skipna` parameter in aggregation functions requires editing both Python and Cython implementations? 5. Does the response note that the Python implementation of `skipna` uses conditional logic checks? 6. Does the response note the use of NumPy's masked array system for handling NA values in the Python implementation? 7. Does the response note that in Cython there are separate code paths for different data types (e.g., float64)? 8. Does the response direct memory access through typed memory views and pointer manipulation? 9. Does the response discuss how Cython

	implementations are stored in .pyx files? 10. Does the response discuss how Cython enables parallel multi-core processing through the @cython.parallel decorator? 11. Does the response discuss the use of vectorized operations with explicit NaN handling at the C level in the Cython implementation? 12. Does the response discuss how Cython implementations are used for performance critical operations whereas Pyton implementations are used for less common or performance critical operations?
2. What would be the approach to add the `skipna` parameter to GroupBy aggregation methods? Propose at least two approaches and describe their relevant performance tradeoffs.	1. Does the response propose at least two different approaches to implementing the `skipna` parameter for aggregation logic? 2. Does the response include a recommendation for the best implementation path with supporting reasoning? 3. Does the response include at least one pro and one con for each potential implementation path? 4. Does the response propose adding a `skipna` parameter to the relevant user facing files and then passing that parameter down to relevant implementation channels? 5. Does the response mention the need to update different implementations in both Python and Cython to accomodate the new functionality? 6. Does the response discuss aggregation logic in for the `GroupBy` class `pandas/core/groupby/generic.py` and how this is where relevant user facing functions aggregation functions (e.g., `mean`, `sum`) are stored? 7. Does the response discuss the need to update Numba code generators in `pandas/core/groupby/numba_.py` to handle the `skipna` parameter for JIT-compiled functions? 8. Does the response discuss how to use mask operations to handle NA values in a performant way? 9. Does the response's discussion of performance tradeoffs discuss implementation simplicity vs. speed and memory allocation? 10. Does the response discuss how Cython

	implementations of `groupby` functions are wrapped in code from `pandas/core/groupby/ops.py`? 11. Does the response note that for all implementation paths, the `skipna` parameter must be added to user facing functions in the `GroupBy` class?