# Evaluating the Decency and Consistency of Data Validation Tests Generated by LLMs*

Lindsay Katz, Callandra Moore, Zane Schwartz, Rohan Alexander

September 28, 2023

We investigated the potential of Large Language Models (LLMs) in developing dataset validation tests. We carried out 96 experiments each for both GPT-3.5 and GPT-4, examining different prompt scenarios, learning modes, temperature settings, and roles. The prompt scenarios were: 1) Asking for expectations, 2) Asking for expectations with a given context, 3) Asking for expectations after requesting a simulation, and 4) Asking for expectations with a provided data sample. For learning modes, we tested: 1) zero-shot, 2) one-shot, and 3) few-shot learning. We also tested four temperature settings: 0, 0.4, 0.6, and 1. Furthermore, two distinct roles were considered: 1) "helpful assistant", 2) "expert data scientist". To gauge consistency, every setup was tested five times. The LLM-generated responses were benchmarked against a gold standard suite, created by an experienced data scientist knowledgeable about the data in question. We find there are considerable returns to the use of few-shot learning, and that the more explicit the data setting can be the better. The results of the best combinations complement, but do not replace, those of the gold standard. The best LLM configurations complement, rather than substitute, the gold standard results. This study underscores the value LLMs can bring to the data cleaning and preparation stages of the data science workflow.

*Contact: rohan.alexander@utoronto.ca. Code and data are available at: LINK. Contributions: RA had the initial idea and developed the experimental set-up. ZS established the political donations datasets. LK developed the initial suite of dataset validation tests that we compare the LLMs against. CM advised about the experimental set-up and contributed critical improvements. RA developed the code to interact with the LLMs, evaluated the LLM outputs, and did the modelling. All authors contributed to writing the initial draft as well as improving and finalising the paper.

# 1 Introduction

The Investigative Journalism Foundation (IJF) created and maintains a dataset related to political donations in Canada. As of September 2023, the dataset comprises 9,204,112 observations and 14 variables. Every day new observations are added, based on newly released donations records in the past time period that are made available by the provincial and federal elections agencies. This release is variable and a source may not post records for a few weeks and then many are made available. Katz and Moore (2023) manually construct an extensive suite of automated tests for this dataset. These impose certain minimum standards on the dataset, including: that constituent aspects add to match any total, class is appropriate, and null values are where expected. This suite allows researchers to use the dataset with confidence and ensures that new additions are fit for purpose.

We revisit this suite of tests to determine whether we can use Large Language Models (LLMs) to mimic this suite of validation tests. We consider a variety of prompts, roles, learning and temperature settings, resulting in 96 total experiments. In particular, we consider four prompt variations: ask for expectations; ask for expectations given described context; ask for expectations having first asked for simulation; ask for expectations giving a sample of data. We also consider zero-, one-, and few-shot learning; four temperature values: 0, 0.4, 0.6, and 1; and two roles: helpful assistant, and expert data scientist. For every combination we obtain five responses from the LLM. We run all this separately for both GPT-3.5 and GPT-4.

A human coder judges these responses produced by the LLMs, rating their decency (1-5, where 1 is the worst and 5 is the best) and their consistency (1-5, where 1 means they are very different and 5 means they are essentially identical). We then build an ordinal regression model with `rstanarm` to explore the relationships that decency and consistency have with prompts, roles, learning and temperature settings.

We find there are considerable returns to few-shot learning, and that the more explicit the data setting can be the better. [**Add more**]

Our results demonstrate one use for LLMs in a data analysis workflow, outside of just analysis. In particular, we are often concerned that our results may be an artifact of some

The remainder of this paper is structured as follows: Section 2 …. Section 3 …. Section 4 …. Section 5 …. Finally Section 6 ….

# 2 Background

**Schwartz - can you please add one paragraph of background about the IJF and why high-quality datasets are important for Canadian democracy/journalism/whatever.**

Canadian legislation requires political parties and candidates to disclose records of financial contributions they receive. These records are maintained by elections agencies across provinces and territories, and at the Federal level. The frequency and scope of these disclosures vary across jurisdictions. For instance, in British Columbia, parties, candidates, constituency associations and leadership and nomination contestants can receive political donations, while in Newfoundland and Labrador, donation recipients are limited to only parties and candidates (IJF, 2023).

The IJF's political donations database is a compilation of these political financing records across all Canadian jurisdictions, with data spanning from 1993 to the present day. The database contains 14 variables including the donor's name, the political party and entity to which the donation was made, the amount donated, as well as the region and year of the donation.

While the IJF's database is available in a clean, user-friendly format, the original records upon which it was created were not all accessible in this way. The format of donation records varies across jurisdiction and time. While some are available in readily downloadable spreadsheets, others are available as PDF and HTML files — the former necessitating the use of optical character recognition (OCR) technology (IJF, 2023). To prepare their database for publication, the IJF team performed significant manual cleaning. The majority of this work resulted from the conversion of PDF donation records to rectangular CSV format using OCR which is prone to scanning-related errors, such as the number 0 being scanned as the letter O. The IJF manually corrected these errors wherever they were identified by carefully comparing the machine-legible OCR output to the original static PDF donations record (IJF, 2023).

Additional cleaning was done for the purpose of data legibility. For instance, the IJF standardized donation dates to match the YYYY-MM-DD format, and they standardized donor names which were in the format "Surname, first name" to be in the form of "First name surname" (IJF, 2023). Party names and donor types were also standardized for consistency, and donation records with an abbreviated party name were supplied with the complete name for that party. Finally, in rows where the donor type was null but only individuals were legally allowed to make donations in that jurisdiction and year, the IJF changed that null entry to be "Individual" (IJF, 2023).

Despite the thoughtful and thorough cleaning performed by the IJF team, the magnitude of these data coupled with their self-reported nature makes them prone to both human error, and computational error arising from the parsing process.

the use of computational tools to assess data quality.

With over 9.2 million rows in this database, it would be a massive undertaking for the IJF to manually

**Katz - You've already added the paragraphs above, but please add whatever more you think relevant.**

## 3 Data

We are interested in the extent to which the LLMs can develop a suite of data validation tests that is similar to a suite developed by an experienced expert data scientist who is familiar with the dataset. To test this, we establish and run a series of experiements where we consider different specifications and then compare the LLM output. In particular, the variables that we consider are:

- Four prompts:

```
- The Investigative Journalism Foundation (IJF) created and maintains a CSV dataset relat
- The Investigative Journalism Foundation (IJF) created and maintains a CSV dataset relat
    - "amount" is a monetary value that cannot be less than $0. An example observation is
    - "amount" should be equal to the sum of "amount_monetary" and "amount_non_monetary"
    - "region" can be one of the following values: "Federal", "Quebec", "British Columbia
    - "donor_full_name" is a string. It cannot be NA. It is usually a first and last name
    - "donation_date" should be a date in the following format: YYYY-MM-DD. It could be N
    - "donation_year" should match the year of "donation_date" if "donation_date" is not
    - "political_party" cannot be NA. It should be a factor that is equal to one of: "New

    Please write a series of expectations using the Python package great_expectations for
- The Investigative Journalism Foundation (IJF) created and maintains a CSV dataset relat
    - "amount" is a monetary value that cannot be less than $0. An example observation is
    - "amount" should be equal to the sum of "amount_monetary" and "amount_non_monetary"
    - "region" can be one of the following values: "Federal", "Quebec", "British Columbia
    - "donor_full_name" is a string. It cannot be NA. It is usually a first and last name
    - "donation_date" should be a date in the following format: YYYY-MM-DD. It could be N
    - "donation_year" should match the year of "donation_date" if "donation_date" is not
    - "political_party" cannot be NA. It should be a factor that is equal to one of: "New

    Please simulate an example dataset of 1000 observations. Based on that simulation ple
- The Investigative Journalism Foundation (IJF) created and maintains a CSV dataset relat
    An example of a dataset is:
    index,amount,donor_location,donation_date,donor_full_name,donor_type,political_entity
    5279105,$20.00,"Granton, N0M1V0",2014-08-15,Shelley Reynolds,Individual,Party,New Dem
    2187800,$200.00,,,Robert Toupin,Individual,Party,Coalition Avenir Québec - l'Équipe M
    3165665,$50.00,,,Geneviève Dussault,Individual,Party,Québec Solidaire  (Avant Fusion)
    8803473,$250.00,"Nan, Nan",,Roger Anderson,Individual,Party,Reform Party Of Canada,Re
    2000776,"$1,425.00","Calgary, T3H5K2",2018-10-30,Melinda Parker,Individual,Registered
    9321613,$75.00,,2022-06-17,Jeffrey Andrus,Individual,Party,Bc Ndp,Bc Ndp,British Colu
    2426288,$50.00,"Stony Plain, T7Z1L5",2018-07-24,Phillip L Poulin,Individual,Party,Con
    4428629,$100.00,"Calgary, T2Y4K1",2015-07-30,Barry Hollowell,Individual,Party,New Dem
```

```
1010544,$20.00,"Langley, V1M1P2",2020-05-31,Carole Sundin,Individual,Party,New Democ
4254927,$500.00,"Welshpool, E5E1Z1",2015-10-10,Melville E Young,Individual,Party,Cons
8001740,$90.00,"Deleau, R0M0L0",2004-11-15,Clarke Robson,Individual,Party,New Democra

Based on this sample please write a series of expectations using the Python package g
```

- Zero-, one-, and few-shot learning:

```
- The following text in quotes is an example of an expectation for this dataset:
    """
    # Check that there is nothing null in any column of donations details
    donations_mv.expect_column_values_to_not_be_null(column='donor_full_name')
    """
- The following text in quotes is an example of three expectations for this dataset:
    """
    # Check that there is nothing null in any column of donations details
    donations_mv.expect_column_values_to_not_be_null(column='donor_full_name')
    # Check that the federal donation does not exceed the maximum
    donations_mv.expect_column_values_to_be_between(
        column = 'amount',
        max_value = 1675,
        row_condition = 'region=="Federal" & donor_full_name.str.contains("Contributions
        condition_parser = 'pandas'
    )
    # Check that the date matches an appropriate regex format
    donations_mv.expect_column_values_to_match_regex(column = 'donation_date',
                                                     regex = '\\d{4}-\\d{2}-\\d{2}',
                                                     row_condition = "donation_date.isna()==Fa
                                                     condition_parser = 'pandas')
    """
```

- Four temperature values: 0, 0.4, 0.6, and 1.
- Two roles:

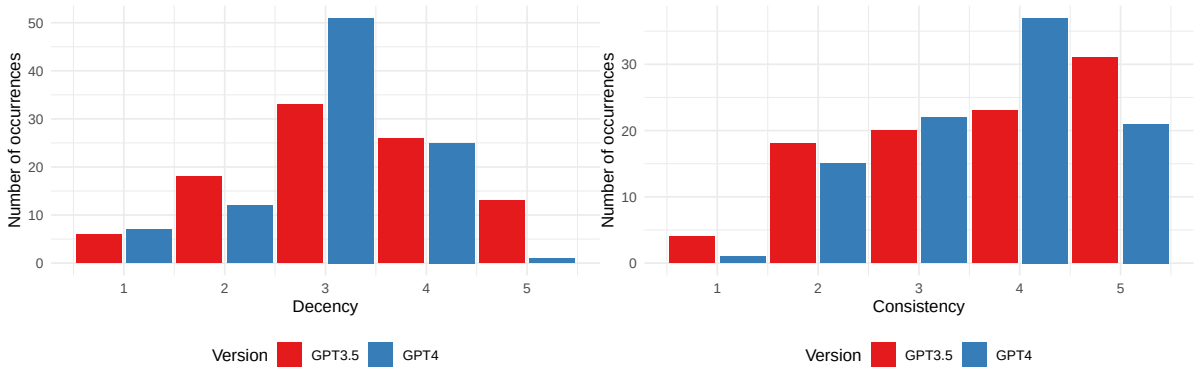
```
- You are a helpful assistant.
- You are a highly-trained, experienced, data scientist who is an expert at writing reada
```

This combination of variables and options, means that in total there are 96 different situations. We run these through both GPT-3.5 and GPT-4, using the API. For every combination we ask for five responses to understand variation.

This results in a dataset of responses. The option that gave rise to each response was blinded, and the order randomized, and then they were ranked by one experienced human coder on

two metrics. The first, "consistency", was a ranking 1-5, of how different each of the five responses was for that particular combination of variables. 1 means that Responses 1-5 were wildly different. 5 means that Responses 1-5 are entirely or essentially the same. The human coder then ranked the "decency" of the first response for each combination of variables. This is a measure of how different the LLM responses were, compared with the code written by the experienced data scientist who wrote the original suite of tests. The LLM responses are not expected to have the full context of the code, so we do not expect an exact match, but it should actually write code, import relevant libraries, add comments, deal with class, write a variety of relevant expectations. 1 means that the code is unusable, 2 means that it is not unusable but would need a lot of work and would be disappointing from a human, 3 means that it is fine but would need some fixing, 4 means it is broadly equivalent to what the gold standard suite, and 5 means it is in no worse and is better in some way than the gold standard validation suite.

Figure 1 examines how decency and consistency differ based on whether GPT-3.5 or GPT-4 is used. Unexpectedly, GPT-4 has fewer responses rated 5/5, compared with GPT-3.5 (Figure 1a). GPT-3.5 has fewer responses rated 2/5, but overall the mean decency response for GPT-3.5 is 3.23, while the mean decency of the responses generated by GPT-4 is 3.01. The consistency is not too different between the two versions, with GPT-3.5 having an average of 3.61, while GPT-4 has an average of 3.65. GPT-4 appears to have fewer responses that are completely identical (Figure 1b).




(a) Decency (b) Consistency

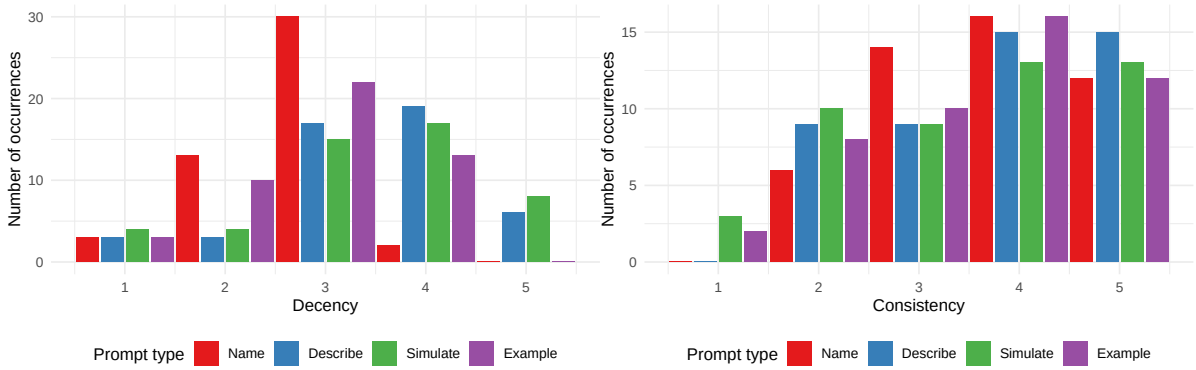
Figure 1: How decency and consistency change based on whether GPT-3.5 or GPT-4 is used

Figure 2 examines how decency and consistency differ based on which prompt is used. The prompts differ by how much information is provided. The least informative prompt, "Name", essentially consists of just providing the LLM with thee names of the columns that are expected. The next most informative prompt, "Describe", adds a detailed description of what we expect of the observations. "Simulate" adds that we expect the LLM to first simulate a dataset based on that description, before generating the expectations. And finally, the most informative prompt, "Example", provides a snapshot of the dataset, consisteing of the relevant variables

and ten observations.

There appears to be considerable difference in terms of the how the prompts are associated with decency. In particular, "Name" is never associated with a 5/5 rating (Figure 2a). Surprisingly, however, the most informative prompt, "Example", is also never associated with a 5/5 rating. Instead it is "Describe" and "Simulate" that tend to be associated with better decency ratings. This is reflected in the averages, which are 2.65, 3.46, 3.44, and 2.94 for "Name", "Describe", "Simulate", and "Example", respectively.

The pattern is not as clear when it comes to consistency (Figure 2b). All four have similar averages, at 3.71, 3.75, 3.48, and 3.58 for "Name", "Describe", "Simulate", and "Example", respectively. That said, it is clear that a wider variety of responses (as denoted by lower consistency ratings), are rarely seen for "Name" and "Describe".




(a) Decency

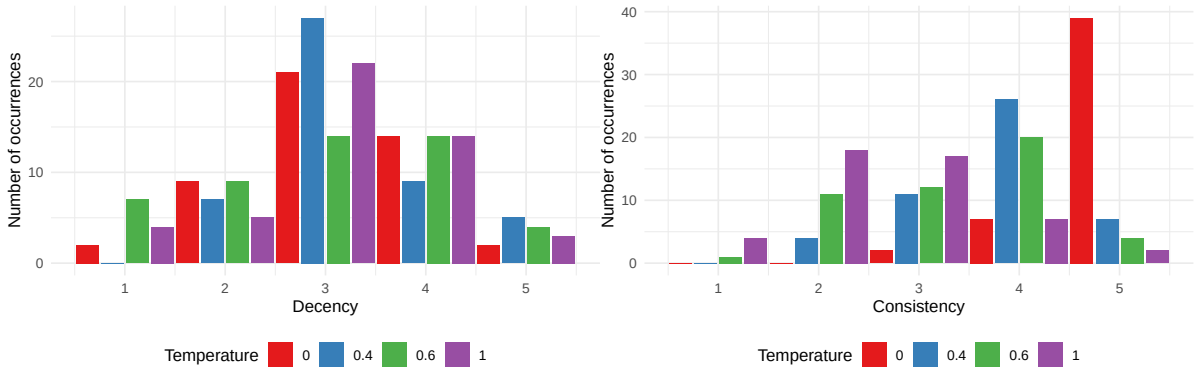
(b) Consistency

Figure 2: How decency and consistency change based on the type of prompt used

Temperature is a parameter that varies from 0 to 1, that we can use to manipulate how random the LLM is. At high temperatures, the LLM will produce a wider variety of responses. At lower temperatures it will focus on the single most likely response. Higher temperature should be associated with a wider variety of LLM responses.

Figure 3 examines how decency and consistency differ based on which of the four temperature values we consider—0, 0.4, 0.6, 1—is used. There appears to be limited difference in terms of how different temperature values are associated with decency (Figure 3a). They all have similar mean values at 3.10, 3.25, 2.98, and 3.15 for temperature values of 0, 0.4, 0.6, and 1, respectively.

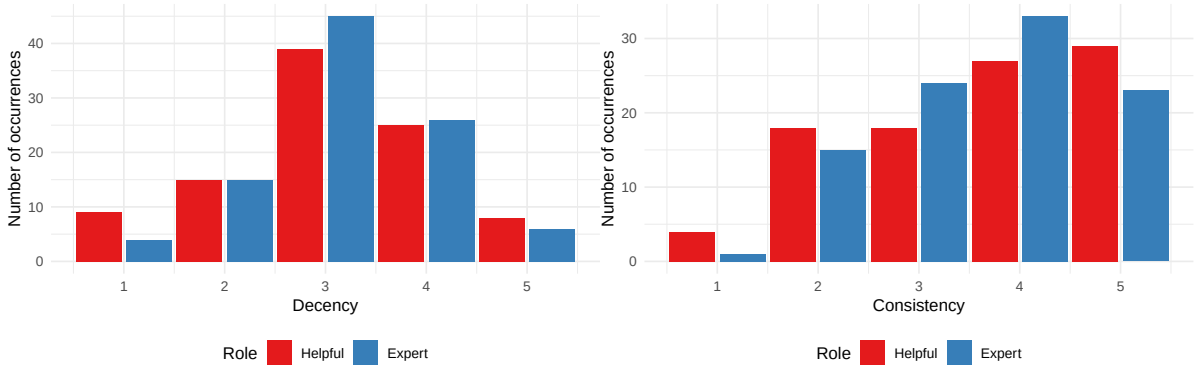
In contrast, as expected temperature has an effect on consistency. Temperature values of 0 are very clearly associated with ratings of less consistency, and higher temperatures are clearly associated with higher consistency (Figure 3b). Their means differ considerably, with 4.77, 3.75, 3.31, and 2.69 for temperature values of 0, 0.4, 0.6, and 1, respectively.




(a) Decency (b) Consistency

Figure 3: How decency and consistency change based on temperature

Role is an aspect of a prompt that is provided to the LLM before the main prompt content. We used two different roles, one that positioned the LLM as a helpful assistant, and the other that positioned the LLM as an experienced data scientist (Figure 4). We were expecting that the expert role would result in better code, but there was no obvious difference in terms of decency (Figure 4a). There was also no obvious difference between the consistency (Figure 4b). Their means did no differ by much in either case.




(a) Decency (b) Consistency

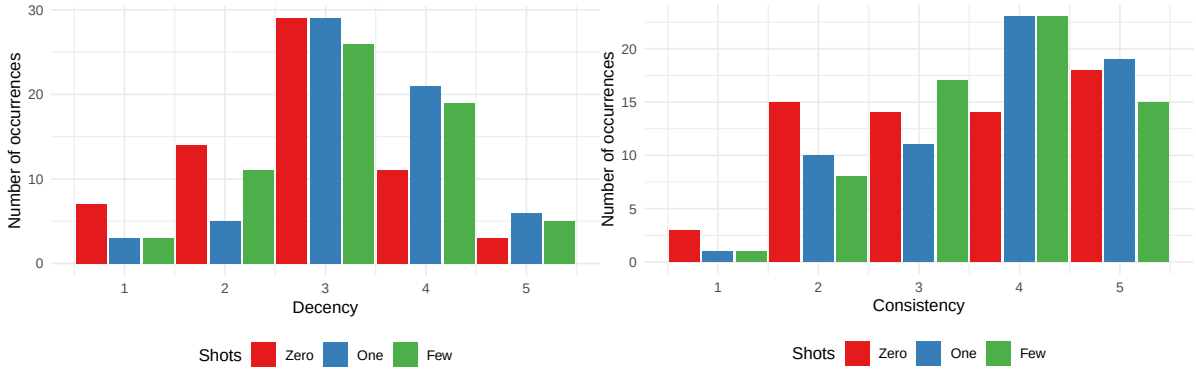
Figure 4: How decency and consistency change based on role

Shots refers to the number of examples provided to the LLM as part of the prompt. Zero-shot means that no examples are provided, while one-shot and few-shot refer to one- and a few-examples being provided, respectively. Although the advantage of LLMs such as GPT-3.5 and GPT-4 is that they typically do well with zero-shot learning, we would expect that they will do better with one-shot and few-shot.

We find substantial differences, especially when moving away from zero-shot learning (Figure 5). In particular, we see that decency of 1/5 is dominated by zero-shot, while zero-shot is under-

represented in 5/5 (Figure 5a). This is also reflected in the mean decency which for zero-shot is 2.83, while for one- and few-shot learning is 3.34 and 3.19, respectively.

We see this pattern in consistency as well. For instance, zero-shot learning is over-represented in the least consistent responses, both 1/5 and 2/5 (Figure 5b). And the mean level of consistency is lower for zero-shot learning, at 3.45, compared with single- and few-shot, at 3.77 and 3.67, respectively.




(a) Decency (b) Consistency

Figure 5: How decency and consistency change based on zero- and one-shot learning

## 4 Model

We are interested in exploring the relationships that consistency and decency have with model, prompt, temperature, role, and number of shots. consistency and decency, as dependent variables, are ordered, categorical, outcomes, which motivates our choice of model.

$$
\begin{aligned}
R_i &\sim \text{Ordered-logit}(\phi_i, \kappa) \\
\phi_i &= \beta_0 + \alpha^{\text{version}}_{g[i]} + \alpha^{\text{prompt}}_{a[i]} + \alpha^{\text{temp}}_{s[i]} + \alpha^{\text{role}}_{e[i]} + \alpha^{\text{shot}}_{e[i]} \\
\kappa_k &\sim \text{Normal}(0, 1.5)
\end{aligned}
$$

We estimate our models separately, for each of consistency and decency, using `rstanarm` (Goodrich et al. 2023).

## 5 Results

Our model estimates are shown in Table 1 and Figure 6. In general, they show that consistency increases with model X compared with model Y, as the prompt changes to be more/less specific,

as temperature increases/decreases, for role X compared with role Y, and as the number of shots increases/decreases. Similarly, decency increases with model X compared with model Y, as the prompt changes to be more/less specific, as temperature increases/decreases, for role X compared with role Y, and as the number of shots increases/decreases.

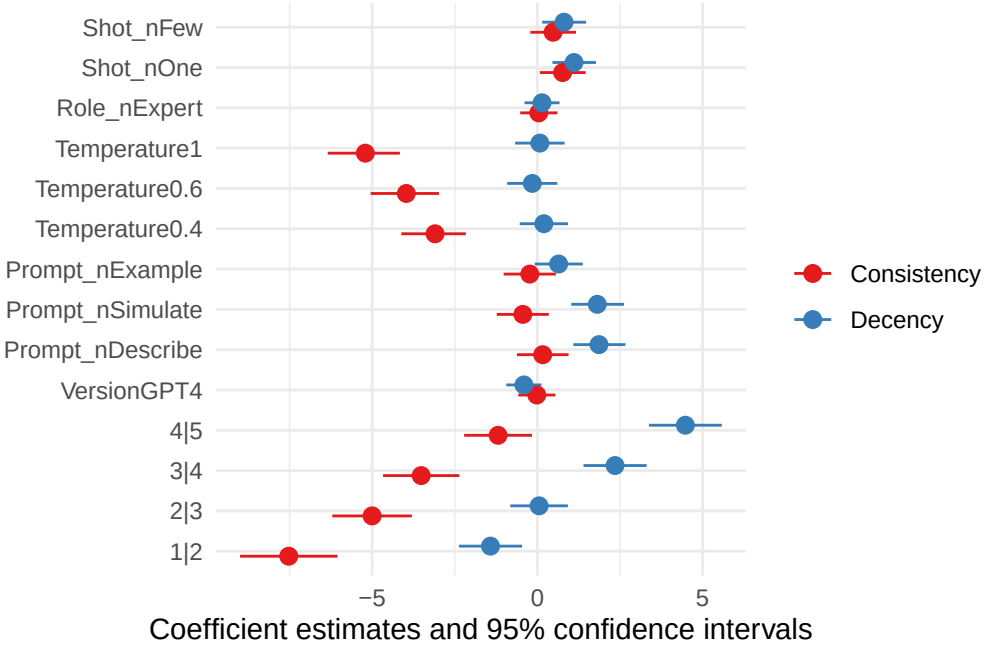



Figure 6: Exploring the relationships that consistency and decency have with model, prompt, temperature, role, and number of shots

# 6 Discussion

## 6.1 Why validation tests are important and some of the challenges of writing them

**Katz - can you please add a page or two that interacts with the literature and situates this contribution? And whatever else you think. Feel free to change the sub-heading**

## 6.2 On the use of LLMs in data science

**Moore - can you please add a page that interacts with the literature and situates this contribution? And/or whatever else you think. Feel free to change the sub-heading to what you want to write about.**

Table 1: Exploring the relationships that consistency and decency have with model, prompt, temperature, role, and number of shots

| | Consistency | Decency |
|---|---|---|
| 1\|2 | −7.53 | −1.42 |
| | (0.75) | (0.48) |
| 2\|3 | −5.00 | 0.05 |
| | (0.61) | (0.44) |
| 3\|4 | −3.52 | 2.35 |
| | (0.58) | (0.48) |
| 4\|5 | −1.19 | 4.48 |
| | (0.52) | (0.56) |
| VersionGPT4 | −0.02 | −0.41 |
| | (0.29) | (0.27) |
| Prompt_nDescribe | 0.16 | 1.86 |
| | (0.40) | (0.40) |
| Prompt_nSimulate | −0.44 | 1.81 |
| | (0.40) | (0.41) |
| Prompt_nExample | −0.23 | 0.64 |
| | (0.40) | (0.37) |
| Temperature0.4 | −3.10 | 0.19 |
| | (0.50) | (0.37) |
| Temperature0.6 | −3.97 | −0.15 |
| | (0.53) | (0.39) |
| Temperature1 | −5.21 | 0.07 |
| | (0.56) | (0.38) |
| Role_nExpert | 0.04 | 0.14 |
| | (0.29) | (0.27) |
| Shot_nOne | 0.76 | 1.11 |
| | (0.35) | (0.34) |
| Shot_nFew | 0.47 | 0.81 |
| | (0.35) | (0.34) |
| Num.Obs. | 192 | 192 |
| AIC | 451.8 | 512.2 |
| BIC | 497.4 | 557.8 |
| RMSE | 3.62 | 3.09 |

### 6.3 On the importance of experimentation in understanding frontier LLM tasks

### 6.4 Limitations

One coder.

One setting.

## References

Goodrich, Ben, Jonah Gabry, Imad Ali, and Sam Brilleman. 2023. "Rstanarm: Bayesian Applied Regression Modeling via Stan." https://mc-stan.org/rstanarm/.

Katz, Lindsay, and Callandra Moore. 2023. "Implementing Automated Data Validation for Canadian Political Datasets." https://doi.org/10.48550/arXiv.2309.12886.