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

Rohan Alexander, Lindsay Katz, Callandra Moore, Zane Schwartz

September 30, 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: https://github.com/RohanAlexander/evaluating_decency_and_consistency. Contributions: RA had the initial idea. RA and CM 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. RA wrote 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 finalizing 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 15 variables. Every day new observations are added, based on newly released donations records made available by provincial and federal elections agencies. This release cycle is variable, with periods of inactivity followed by bursts of multiple releases. 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 a simulation; ask for expectations given 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 experiments 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 one- and few-shot learning. We also find that including detailed descriptions of the expected dataset can help improve the quality of the tests that are produced, even more than including an example of the dataset. Surprisingly, there was not much of a difference found between GPT-3.5 and GPT-4.

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 errors in the data cleaning and preparation pipeline. Data validation tests help assuage these concerns, and using LLMs to establish an initial suite of tests can encourage their use.

The remainder of this paper is structured as follows: Section 2 provides a brief overview of the underlying dataset about which we are writing tests, as well as LLMs. Section 3 details the human coded LLM responses, which is the analysis dataset used in this paper. Section 4 specifies the analysis model used and Section 5 details the estimates. Finally Section 6 discusses some of the implications of this study, and details a few weaknesses.

2 Background

2.1 The political donations dataset

The Investigative Journalism Foundation (IJF) is a nonprofit news media outlet that is centered around public interest journalism. Their mandate is to help rebuild trust in Canadian democracy and hold the powerful accountable through data-driven investigative reporting. A core component of the IJF's work is building and actively maintaining eight publicly available databases with information on political donations, registered charities, and political lobbying in Canada. While the information that these databases are built upon are ostensibly public, they are not maintained or available in a way that is widely accessible or conducive to analysis. Further, the format and accessibility of these data vary greatly over time and across jurisdictions, making it incredibly difficult to look at temporal or regional differences in the data. In turn, the IJFs collation of these databases and high-quality journalism informed by these data serves as a crucial contribution to rebuilding trust and transparency in Canadian democracy.

One of the IJFs eight databases, and the focus of this work, is the political donations database. 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 (The 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 (The 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 (The 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"

(The 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" (The 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. With over 9.2 million rows in this database, it would be a massive undertaking for the IJF to manually identify all errors and inconsistencies present in their data - whether that is as minor as an incorrectly formatted date, or as major as a reported donation amount thousands of dollars above the legal limit. To address this challenge, the IJF has been exploring the use of computational tools to assess data quality (Katz and Moore 2023). The team has developed a suite of comprehensive data tests for all eight databases using Python's `Great Expectations` library, which contains a number of pre-defined functions to test that particular characteristics expected of a dataset hold true in the data at large.

As described by Katz and Moore (2023), the process of developing this test suite was iterative in nature, and significant domain knowledge was necessary to develop tests which were both accurate and valuable. For the political donations database, Katz and Moore (2023) developed a comprehensive suite of tests pertaining to missingness, formatting, and value expectations in the data. For instance, they set the expectation that for donations made in British Columbia, Ontario, or Federally, the `donation_date` entry should not be missing, because those are the only three jurisdictions which collect that variable (Katz and Moore 2023). They also test that donations data from 2022 align with the 2022 legal donation limits for each jurisdiction, and that, where applicable, the sum of reported monetary and non-monetary contribution amounts add up to the total reported donation amount. Katz and Moore (2023) provides complete details on the development and implementation of these tests.

2.2 Large Language Models

Large Language Models (LLMs) are customized and refined through in-context learning using prompting (Ouyang et al. 2023). A prompt is a set of natural language instructions which define the parameters and context for the desired output and may include one or more input-output examples. By using this approach, LLMs can apply their existing knowledge gained from training on various datasets to adapt to new contexts specified by the prompt. The outcomes generated by LLMs are highly sensitive to the specific phrasing and structure of these natural language instructions. Consequently, there is ongoing work to establish effective prompt patterns (White et al. 2023).

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

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

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

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

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

- 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("Contributi
        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()
                                                     condition_parser = 'pandas')
    """
```

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

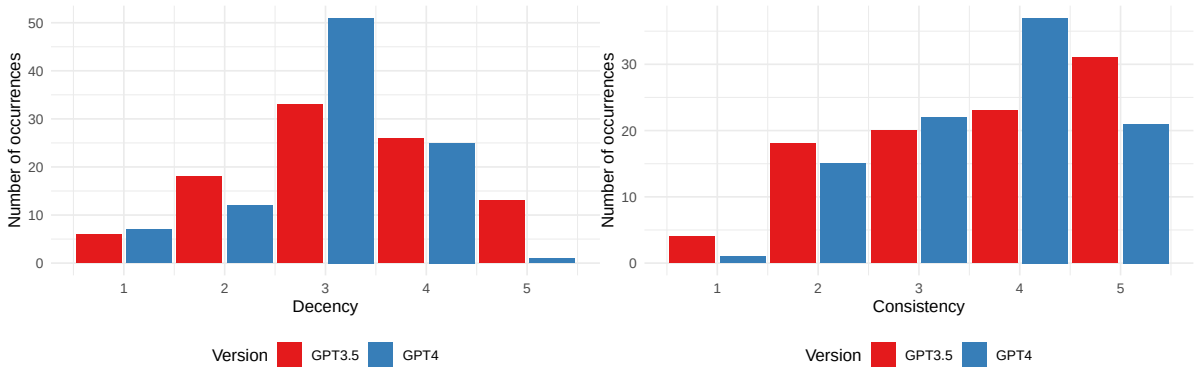
```
- You are a helpful assistant.
- You are a highly-trained, experienced, data scientist who is an expert at writing
```

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, and 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 contains, 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 for decency, 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

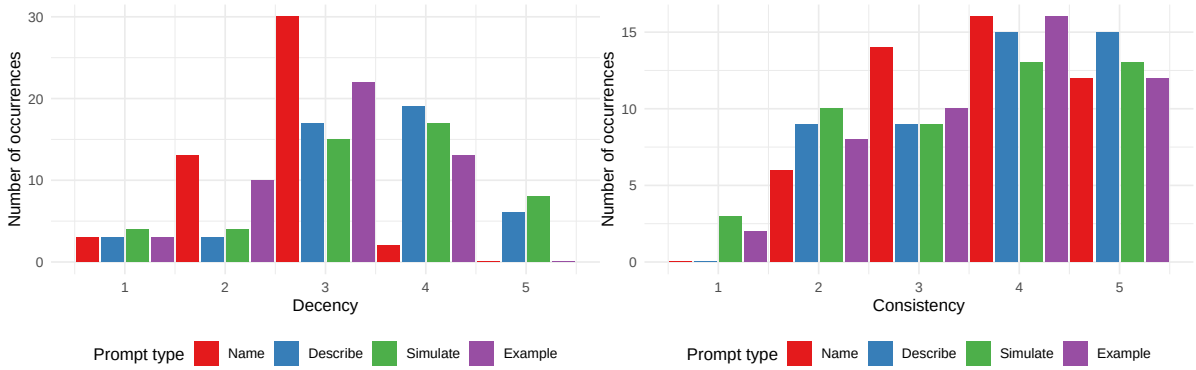
Figure 1: How decency and consistency change based on whether GPT-3.5 or GPT-4 is used

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 the names of the columns that are expected to be in the dataset. 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, consisting 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

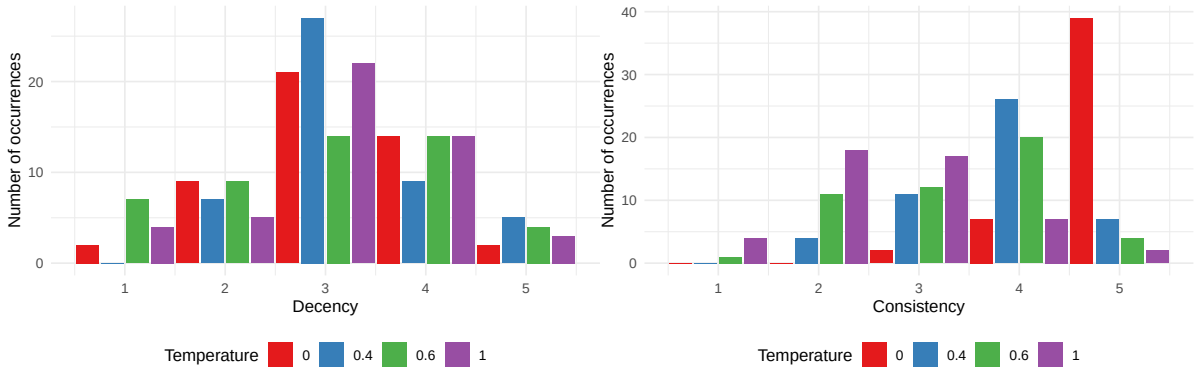
(b) Consistency

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.

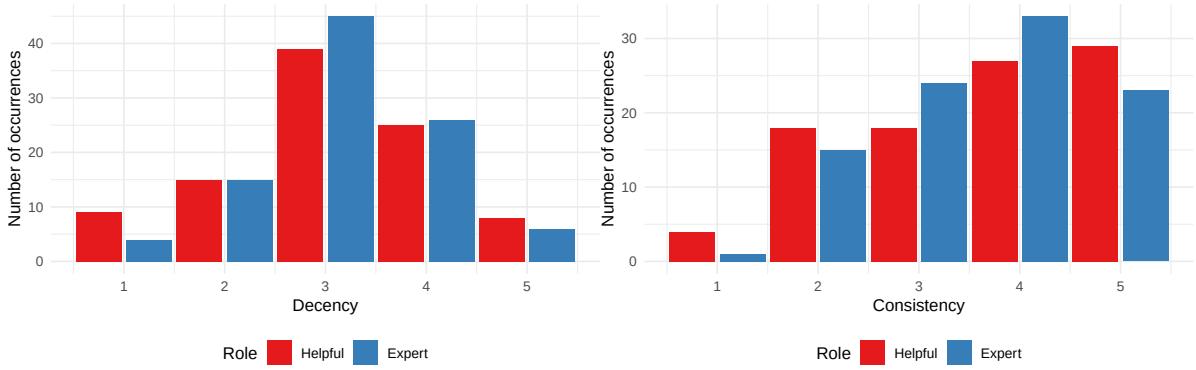
In contrast, as expected temperature has an effect on consistency. Temperature values of 0 are very clearly associated with ratings of less consistency, and higher temperatures are clearly associated with higher consistency (Figure 3b). Their means differ considerably, with 4.77, 3.75, 3.31, and 2.69 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 not differ by much in either case.



(a) Decency (b) Consistency

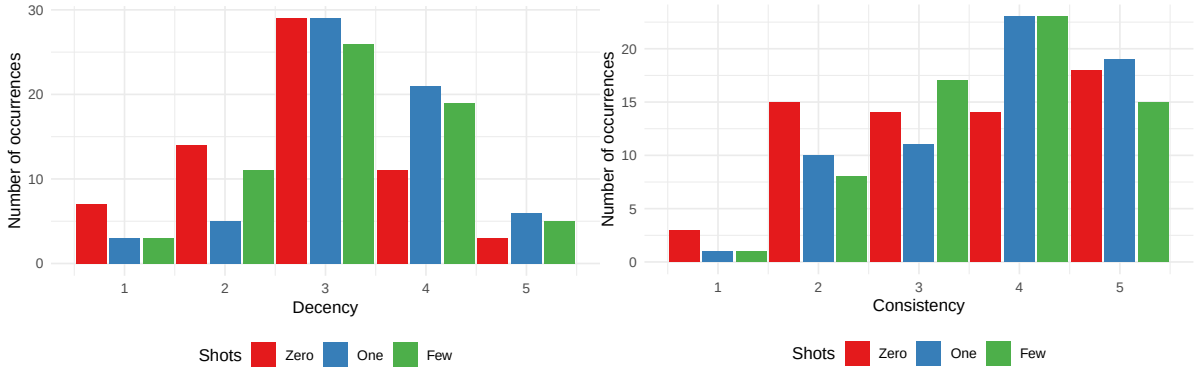
Figure 4: How decency and consistency change based on whether role is 'Helpful' or 'Expert'

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-, one-, and few-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 to use an ordered-logit model. Such a model works by estimating the probability that the outcome, R_i, is less than or equal to some specific category, which in this case is the coded rank of that response (i.e. 1, 2, 3, 4, 5). Models for consistency and decency are estimated separately.

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

In terms of our predictors, version, is a binary variable as to whether we are using GPT-3.5 or GPT-4. We expect that GPT-4 will do better in terms of both decency and consistency, than GPT-3.5 Prompt can be one of four values: "Name", "Describe", "Simulate", and "Example". We expect that "Simulate" and "Example" will be associated with higher decency than "Name" and "Describe". However, we expect the opposite relationship with consistency. This is because

we expect that what will increase decency will be the more specific guidance provided by the "Simulate" and "Example" prompts, which should also increase the consistency. Temperature can take one of four values: 0, 0.4, 0.6 and 1. We expect that higher temperature values will be associated with less consistency. But it is difficult to anticipate how it should be related to decency. Role can be one of two values: "Helpful", or "Expert", while shots can be one of three: "zero", "one", or "few". In the case of both role and shots, we expect that the "Expert" role, and one- and few-shot learning will be associated with higher decency, and consistency.

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

5 Results

The estimates from our models are shown in Table 1 and Figure 6. The intercepts, 1|2, 2|3, 3|4, 4|5, shown in Table 1 reflect cutpoints, where a rating goes from one category to the next (e.g. from 1 to 2, etc).

In general, the models do not identify a substantial difference between GPT-3.5 and GPT-4 in terms of consistency, and, surprisingly, even a somewhat negative association in terms of decency. Our models do not identify much difference between the different prompt types in terms of consistency. However, they do find that "Describe" and "Simulate" are associated with increased decency, as is "Example", although surprisingly to a lesser extent. As expected increases in temperature are associated with less consistency. However they are unable to identify much of a relationship between temperature and decency. The models do not find any association between which role is used, in terms of either consistency or decency. And finally, one- and few-shot learning is associated with increased decency and slightly increased consistency.

6 Discussion

6.1 On the importance and challenges of writing validation tests

Much important work has been done to illustrate the importance of reproducibility and reliability in data science across domains. In particular reproducibility promotes transparency, trustworthiness and credibility, and a more thorough understanding of the data. The value of data validation is not limited to particular domains—it is a fundamental component of a reliable data science workflow, regardless of the data at hand.

There are a number of challenges that accompany the development and implementation of data validation tests. First, implementing automated tests requires the ability to develop

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

	Consistency	Decency
VersionGPT4	−0.03	−0.34
	(0.26)	(0.25)
Prompt_nDescribe	0.14	1.51
	(0.37)	(0.37)
Prompt_nSimulate	−0.35	1.47
	(0.36)	(0.37)
Prompt_nExample	−0.19	0.53
	(0.39)	(0.34)
Temperature0.4	−2.56	0.15
	(0.44)	(0.35)
Temperature0.6	−3.31	−0.13
	(0.49)	(0.34)
Temperature1	−4.38	0.07
	(0.51)	(0.35)
Role_nExpert	0.03	0.11
	(0.27)	(0.24)
Shot_nOne	0.63	0.89
	(0.33)	(0.30)
Shot_nFew	0.39	0.64
	(0.32)	(0.30)
1|2	−6.90	−1.64
	(0.72)	(0.44)
2|3	−4.40	−0.19
	(0.56)	(0.41)
3|4	−2.97	2.00
	(0.53)	(0.44)
4|5	−0.83	4.08
	(0.47)	(0.53)
Num.Obs.	192	192
ELPD	−226.4	−255.4
ELPD s.e.	9.8	9.6
LOOIC	452.7	510.8
LOOIC s.e.	19.5	19.2
WAIC	452.6	510.7

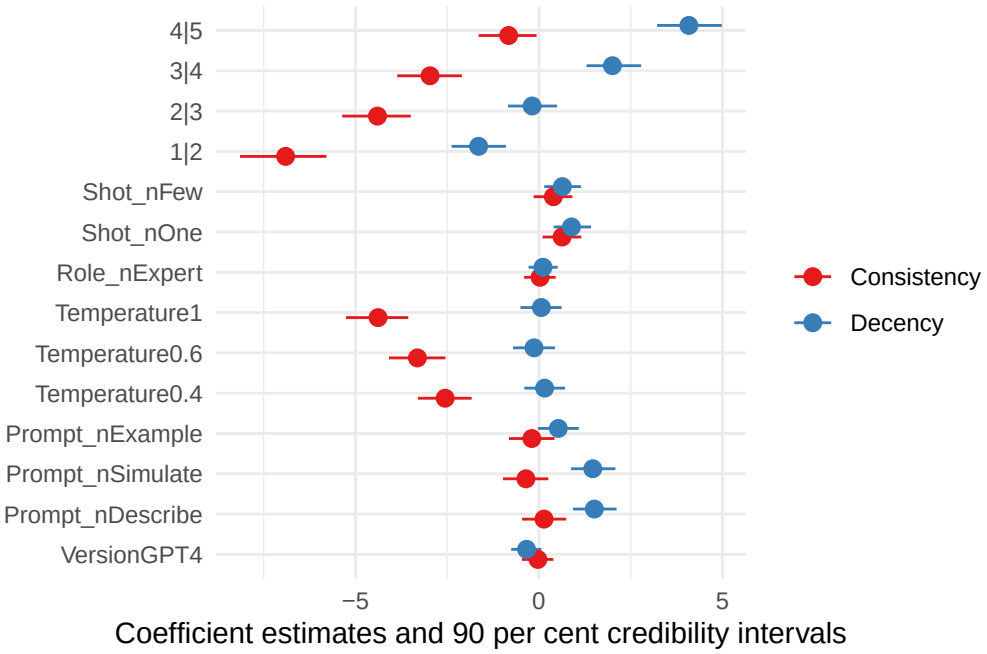


Figure 6: Exploring the relationships that consistency and decency have with model, prompt, temperature, role, and number of shots

code (Zhang et al. 2023). While there are a number of accessible, well-documented testing-centered libraries in different programming languages (e.g., `Great Expectations` in Python, and `pointblank` in R), the implementation of a test suite may require significant data processing work. One of the main takeaways from Katz and Moore (2023) on their data validation efforts for the IJF is that data tests should not be compromised just to be compatible with pre-defined test functions that exist in a software package. In order to develop the most valuable suite of tests, it may be necessary to create new variables in a dataset or to merge multiple datasets, for example. This presents a challenge for individuals who are not trained in or comfortable with computer programming, but wish to programmatically test their data. In addition, the development of a comprehensive, bespoke, and accurate data test suite can rarely be done by one data scientist alone. As advocated by Katz and Moore (2023), domain expertise is fundamental for the creation and development of high-quality data tests. For instance, without incorporating the knowledge that donation date is only collected in three jurisdictions (British Columbia, Ontario and Federal) into their code, Katz and Moore (2023)'s test surrounding missingness for that variable would have produced inflated failure rates. This is knowledge that was gained through collaborative efforts with IJF team members who were involved in dataset construction. For an individual working on an unfamiliar dataset, seeking out domain knowledge and gaining a thorough understanding of one's data is a non-trivial task in and of itself. Finally, the process of designing, developing, and deploying data validation tests is iterative in nature and takes a great deal of time. For researchers or journalists who

are eager to publish their findings for instance, there may not be enough incentive or resources for them to want to do this time-consuming validation work. This challenge is only amplified if an individual was not involved in the dataset construction process and does not have access to vital domain expertise, or if an individual does not have programming experience.

Evidently, while data validation is incredibly valuable for the production of transparent and trustworthy research, it comes with many challenges which hinder its accessibility. This work presents an opportunity to promote data validation through the use of LLMs. While there has been a great deal of work done to develop more accessible programming tools for data validation, these tools do not entirely mitigate the challenges which accompany data validation, namely the need for programming knowledge, domain expertise, and a great deal of time. Our findings illustrate the ability of LLMs to rapidly produce usable data validation tests, written in code, with limited information.

6.2 On the use of LLMs in data science

The contributions of these experiments and this paper are twofold. Firstly, this work adds to the emerging area of inquiry of prompt engineering for large language models. In doing so, our prompts may serve as examples – both good and bad – for other scientists hoping to implement LLMs into their research and data workflows. Secondly, this work contributes to the academic study of LLMs for the development of data science methodologies. As these tools become more powerful and the precision in our ability to specify their outputs improves, they will become tools to reduce the time and resource burden of writing software and tests.

Our work is consistent with previous literature demonstrating that LLM performance on complex, user-defined tasks is sensitive to prompt engineering. Prompting is a brittle process in which minor alterations can lead to large fluctuations in performance Zhao et al. (2021), however with OpenAI's LLMs in particular, GPT-4 (Giorgi et al. 2023) seems less susceptible to this than ChatGPT (GPT-3.5). In general, previous research has demonstrated that when using ChatGPT, prompts should provide context Clavié et al. (2023), be specific Shieh (n.d.), define a persona or role Clavié et al. (2023), break complex reasoning into smaller steps Henrickson and Meroño-Peñuela (2023), and provide example responses Shieh (n.d.) to improve performance on a variety of tasks, including classifying job types (Clavié et al. 2023), generating images for construction defect detection (Yong et al. 2023), and generating hermeneutically valuable text (Henrickson and Meroño-Peñuela 2023).

These observations are predominantly borne out in our experiments. For instance, the more specified prompt ("Describe") and the prompt eliciting step-by-step thinking ("Simulate") both show improvements over the simplest prompt ("Name"). However, improvements made by specificity break down when example data is provided ("Example"). We hypothesize that this is due to ChatGPT and GPT-4's poor performance on numerical and structured data in comparison to more domain-specific LLMs such as BloombergGPT which has been trained on numeric, structured financial data and utilizes a tokenizer more suited to these data (Wu et al.

2023). We do not see significant changes in performance when two different roles are specified to the LLM, which contradicts previous literature Clavié et al. (2023). Finally, our significant improvements moving from zero- to one-shot and marginal to no improvements from one- to few-shot align with existing literature (Giorgi et al. 2023). Some studies have demonstrated ChatGPT and GPT-4's difficulties in responding to queries in the desired format during zero-shot structured tasks, including named entity recognition (Hu et al. 2023) and multiple choice question (Clavié et al. 2023). Giorgi et al. (2023) thus attribute performance improvements in one- and few-shot prompts to the in-context examples' guidance as to the desired output structure. Upon inspection of the outputs from zero-, one-, and few-shot prompts, we believe that this same effect exists for our prompts.

Our work additionally contributes to the growing body of work using LLMs to automate portions of the data science workflow. LLMs have already been integrated into software tools such as GitHub's Co-Pilot ("Your AI Pair Programmer," n.d.) and AutoGPT ("AutoGPT," n.d.) and IDEs such as IntelliJ (Krochmalski 2104) and VSCode (Dias 2023) to produce code in a variety of languages. These tools are and will continue to assist data scientists to accelerate, resolve bugs in, document, and optimize their code in all parts of the data science workflow. LLMs have also demonstrated potential to automate other onerous or tedious tasks in data science pipelines. These include generating synthetic text data Yu et al. (2023), improving search-based software testing (Lemieux et al. 2023), generating unit tests from natural language in a variety of programming languages and contexts Schafer et al. (2023), generating property-based tests from API documentation (Vikram, Lemieux, and Padhye 2023), and conducting exploratory data analysis (Ma et al. 2023). As LLM prompting and fine-tuning methods improve, they will reduce technical barriers to conducting transparent, accurate, reproducible analysis, enabling data scientists to be more productive and put their energy towards analysis. higher-order reasoning, and sophisticated inference. The work described in this paper towards automating data validation using LLMs has the capacity to further this evolution.

6.3 Limitations

There are a variety of limitations of this study. The most notable is that there was only one coder of the LLMs responses. While this should ensure internal consistency, it is possible that some of the coding, especially that for decency, would have benefited from an additional coder, and their responses could then have been averaged. The study also only considers one, reasonably specific, setting. Expanding our approach to consider a variety of settings could be especially beneficial.

A limitation of our work on prompt engineering (and indeed of prompt engineering generally) is that we are unable to provide fully reliable explanations for improvements in performance. Though we can make convincing speculations, these are ultimately educated guesses. Previous studies have shown that in addition to being brittle to precise semantics Zhao et al. (2021), prompts show improved performance from additional text which provides no new information

(Henrickson and Meroño-Peñuela 2023) such as naming the model (e.g., "You are Frederick, a helpful AI"), emulating the chatbot saying that it understands the instructions, asking it to output the "right conclusion", and providing positive feedback (Clavié et al. 2023). Despite the opacity of the emergent properties of LLMs, we may be able to develop greater understanding of the impacts of prompting choices by testing these prompts on a greater diversity of datasets from across disciplines and contexts.

Our current experiments have been run in a very limited context — namely one Canadian political donations dataset. Testing these prompts in a variety of contexts will provide evidence as to whether our prompting strategies are generalizable across datasets in different formats, contexts, and technical regimes. Prompts can be conceptualized as knowledge transfer methods analogous to software patterns which are intended to provide reusable solutions to common problems (White et al. 2023). This aligns with the goal of automated data validation to reduce the need for domain expertise and resources dedicated to data validation. Thus, in order to improve the explainability of our methods and to ensure our method is generalizable, we must validate that the prompts that are effective on this dataset show similar performance on other datasets from different contexts and in different formats.

7 Appendix

7.1 Model diagnostics

7.1.1 Consistency model

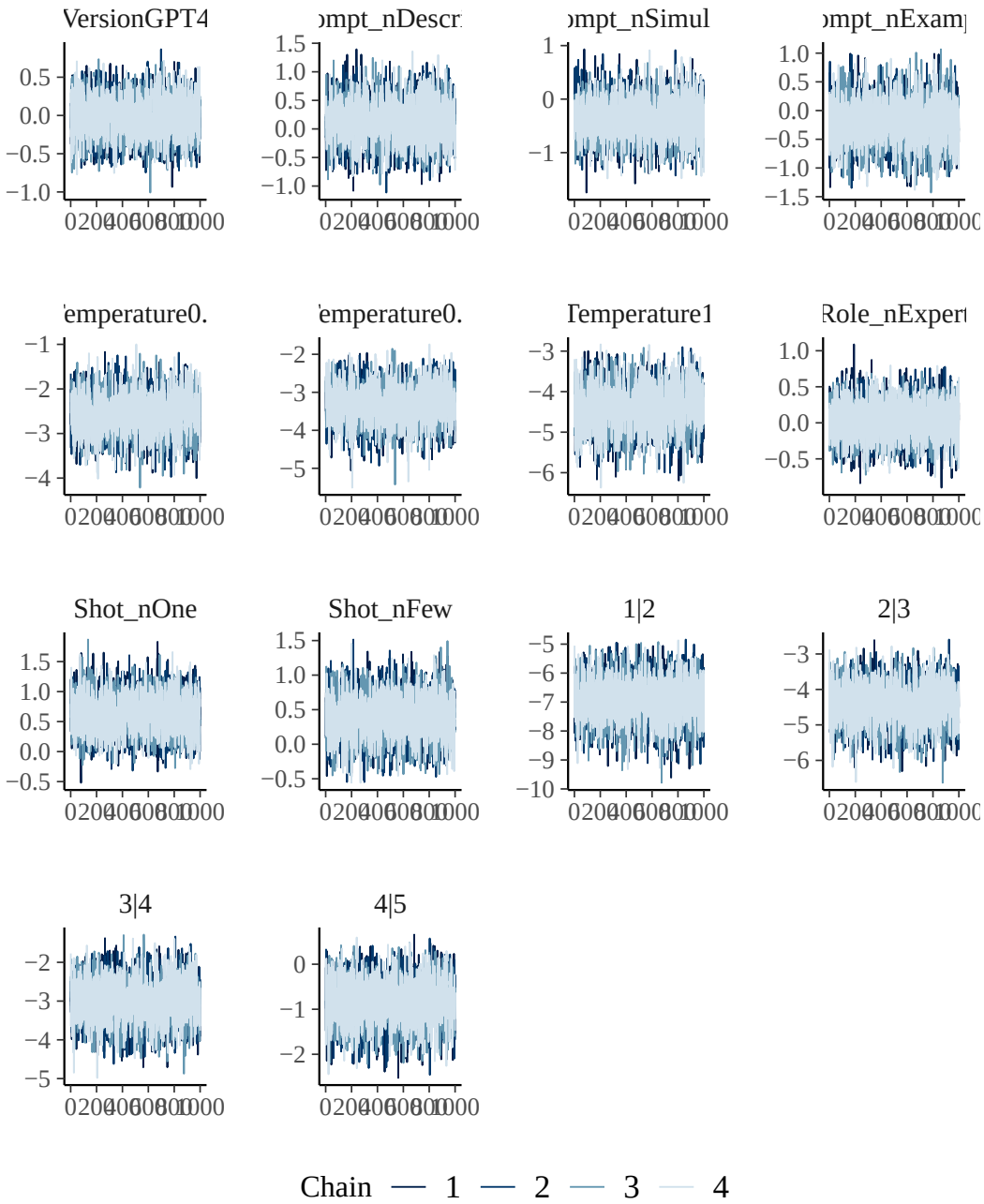


Figure 7: Consistency model trace plot

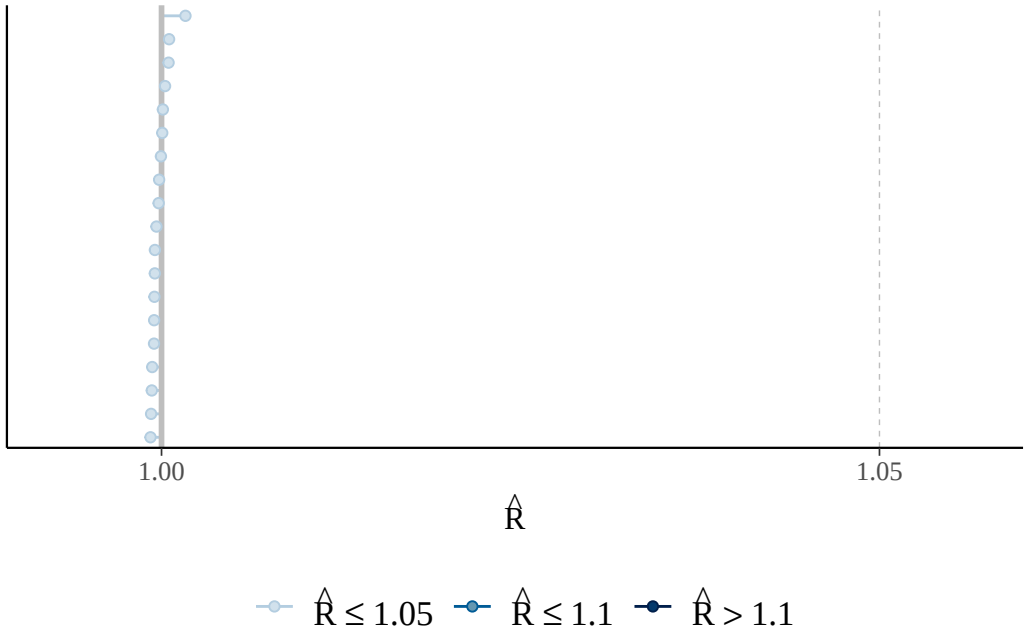


Figure 8: Consistency model rhat plot

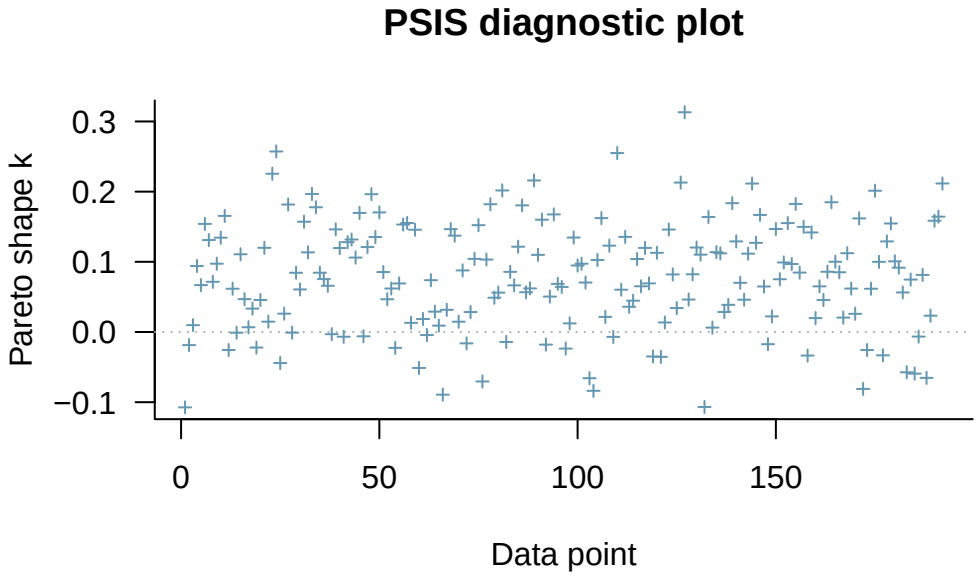


Figure 9: Consistency model loo plot

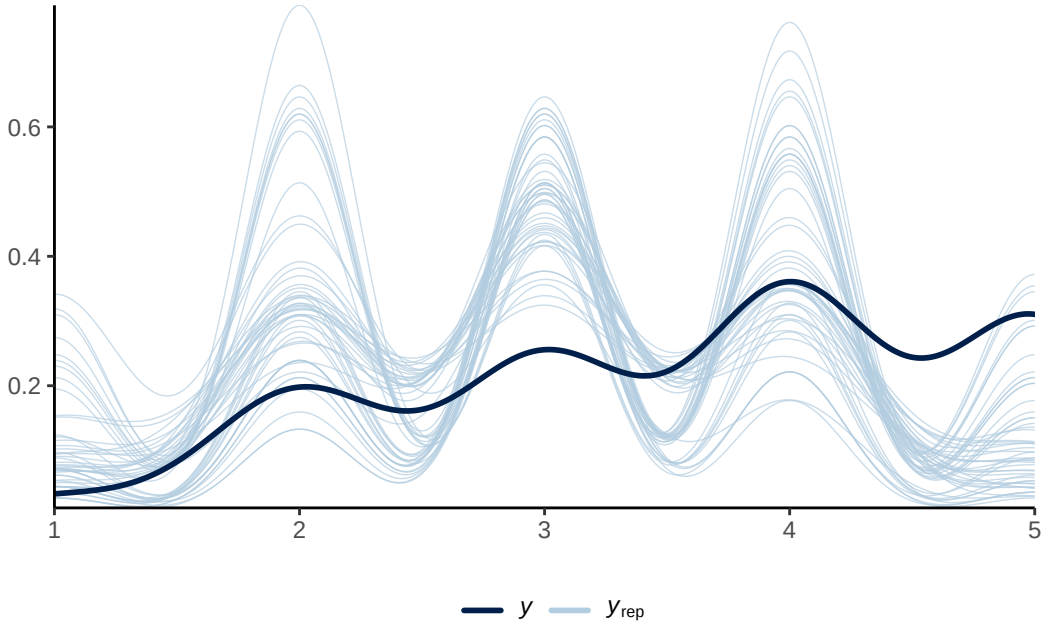


Figure 10: Consistency model posterior predictive check

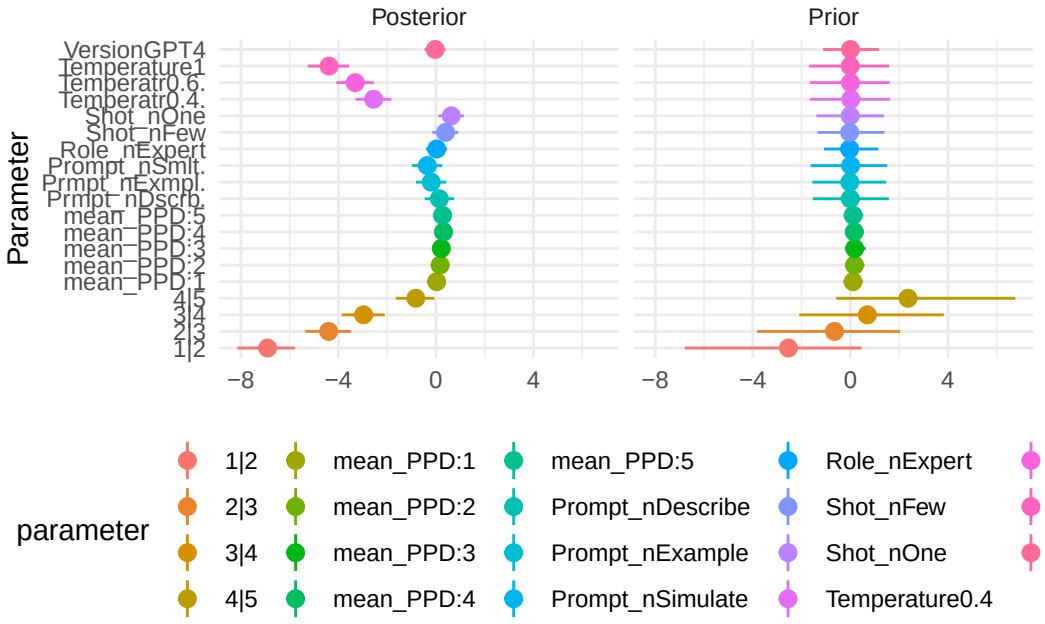


Figure 11: Consistency model posterior vs prior

7.1.2 Decency model

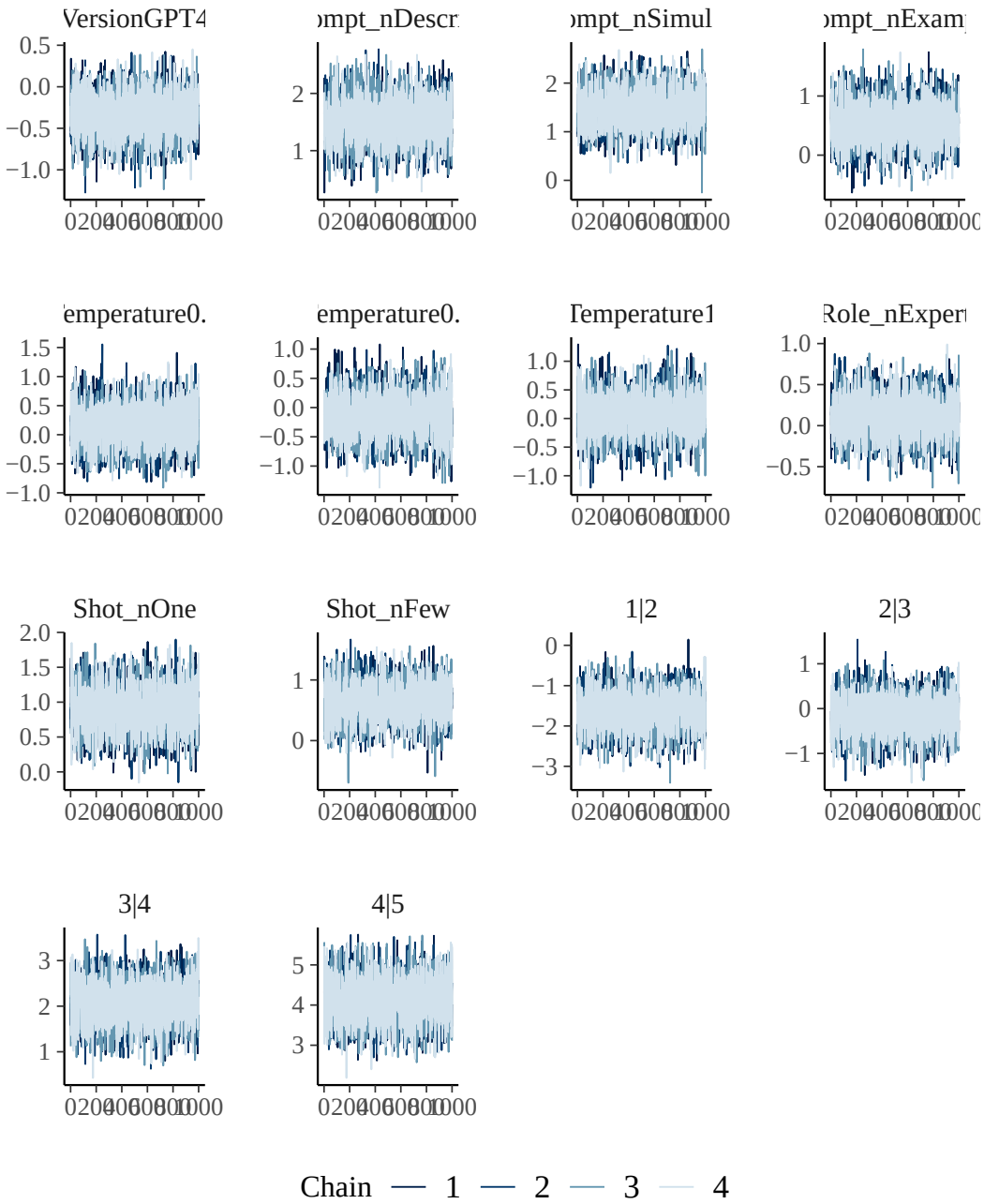


Figure 12: Decency model trace plot

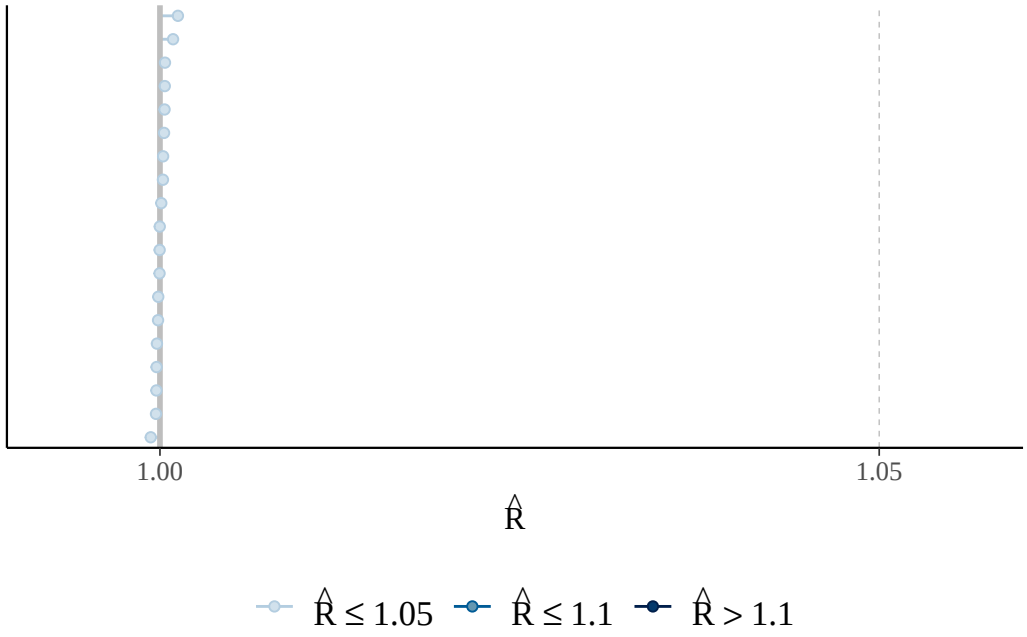


Figure 13: Decency model rhat plot

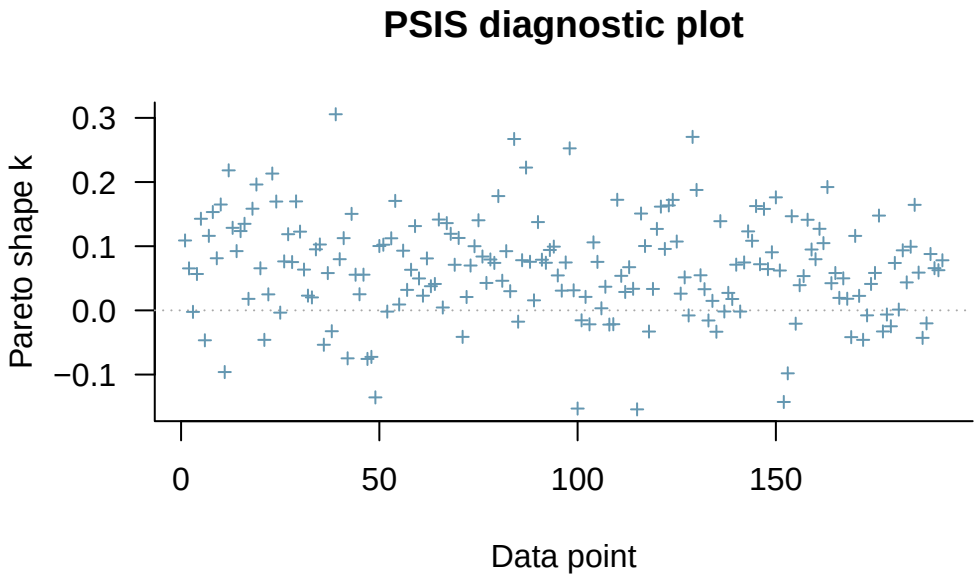


Figure 14: Decency model loo plot

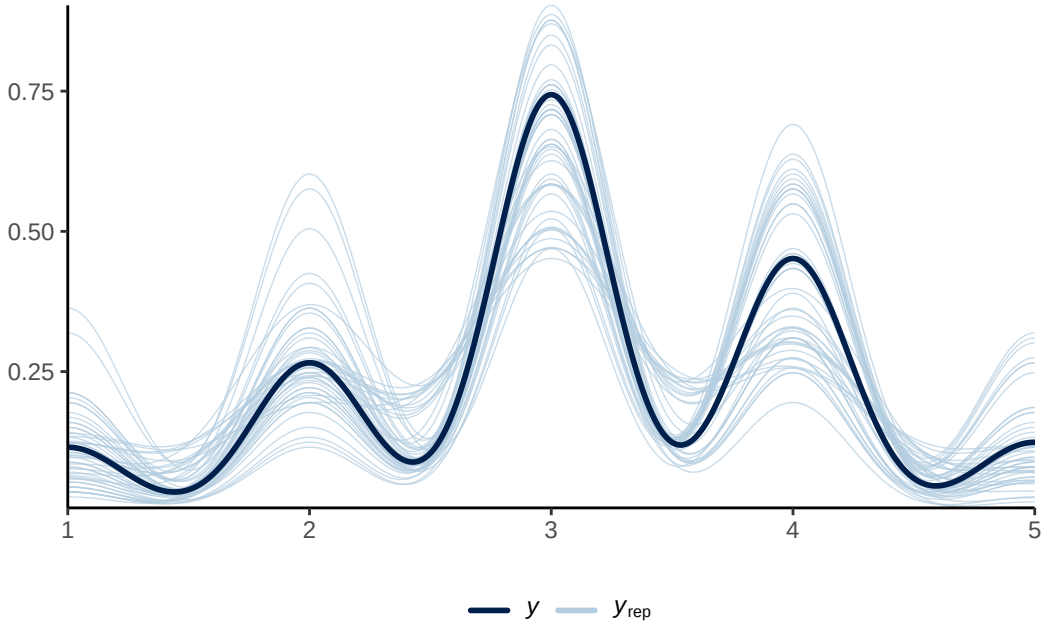


Figure 15: Decency model posterior predictive check

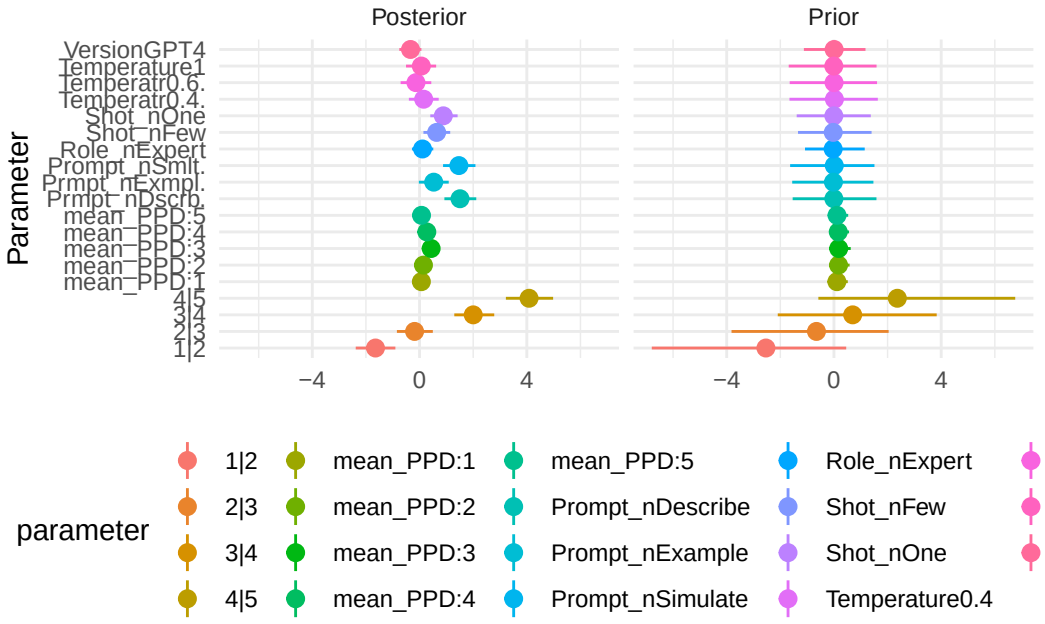


Figure 16: Decency model posterior vs prior

References

Arora, Simran, Avanika Narayan, Mayee F. Chen, Laurel Orr, Neel Guha, Kush Bhatia, Ines Chami, Frederic Sala, and Christopher Ré. 2022. "Ask Me Anything: A Simple Strategy for Prompting Language Models." https://arxiv.org/abs/2210.02441.

"AutoGPT." n.d. https://autogpt.net/; AutoGPT.

Brown, Tom B., Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, et al. 2020. "Language Models Are Few-Shot Learners." *CoRR* abs/2005.14165. https://arxiv.org/abs/2005.14165.

Chung, John, Ece Kamar, and Saleema Amershi. 2023. "Increasing Diversity While Maintaining Accuracy: Text Data Generation with Large Language Models and Human Interventions." In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 575–93. Toronto, Canada: Association for Computational Linguistics. https://doi.org/10.18653/v1/2023.acl-long.34.

Clavié, Benjamin, Alexandru Ciceu, Frederick Naylor, Guillaume Soulié, and Thomas Brightwell. 2023. "Large Language Models in the Workplace: A Case Study on Prompt Engineering for Job Type Classification." In *Natural Language Processing and Information Systems*, edited by Elisabeth Métais, Farid Meziane, Vijayan Sugumaran, Warren Manning, and Stephan Reiff-Marganiec, 3–17. Cham: Springer Nature Switzerland.

Dias, Chris. 2023. "Visual Studio Code and GitHub Copilot." https://code.visualstudio.com/blogs/2023/03/30/vscode-copilot; Microsoft.

Giorgi, John, Augustin Toma, Ronald Xie, Sondra Chen, Kevin An, Grace Zheng, and Bo Wang. 2023. "WangLab at MEDIQA-Chat 2023: Clinical Note Generation from Doctor-Patient Conversations Using Large Language Models." In *Proceedings of the 5th Clinical Natural Language Processing Workshop*, 323–34. Toronto, Canada: Association for Computational Linguistics. https://doi.org/10.18653/v1/2023.clinicalnlp-1.36.

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

Henrickson, Leah, and Albert Meroño-Peñuela. 2023. "Prompting Meaning: A Hermeneutic Approach to Optimising Prompt Engineering with ChatGPT." *AI & Society*, September. https://doi.org/10.1007/s00146-023-01752-8.

Hu, Yan, Iqra Ameer, Xu Zuo, Xueqing Peng, Yujia Zhou, Zehan Li, Yiming Li, Jianfu Li, Xiaoqian Jiang, and Hua Xu. 2023. "Zero-Shot Clinical Entity Recognition Using ChatGPT." https://arxiv.org/abs/2303.16416.

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

Krochmalski, Jaroslaw. 2104. *IntelliJ IDEA Essentials*. Packt Publishing Ltd.

Lahiri, Shuvendu K., Aaditya Naik, Georgios Sakkas, Piali Choudhury, Curtis von Veh, Madanlal Musuvathi, Jeevana Priya Inala, Chenglong Wang, and Jianfeng Gao. 2022. "Interactive Code Generation via Test-Driven User-Intent Formalization." *arXiv*.

Lemieux, Caroline, Jeevana Priya Inala, Shuvendu K Lahiri, and Siddhartha Sen. 2023. "Codamosa: Escaping Coverage Plateaus in Test Generation with Pre-Trained Large Language Models." In *45th International Conference on Software Engineering*. ICSE.

Ma, Pingchuan, Rui Ding, Shuai Wang, Shi Han, and Dongmei Zhang. 2023. "Demonstration of InsightPilot: An LLM-Empowered Automated Data Exploration System." https://arxiv.org/abs/2304.00477.

Ouyang, Long, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, et al. 2023. "Training Language Models to Follow Instructions with Human Feedback." NeurIPS.

Schafer, Max, Sarah Nadi, Aryaz Eghbali, and Frank Tip. 2023. "Adaptive Test Generation Using a Large Language Model." https://arxiv.org/abs/2302.06527v3.

Shieh, Jessica. n.d. "Best Practices for Prompt Engineering with OpenAI API." https://help.openai.com/en/articles/6654000-best-practices-for-prompt-engineering-with-openai-api; OpenAI.

The IJF. 2023. "Donations Methodology." https://theijf.org/donations-methodology.

Vikram, Vasudev, Caroline Lemieux, and Rohan Padhye. 2023. "Can Large Language Models Write Good Property-Based Tests?" https://arxiv.org/abs/2307.04346.

White, Jules, Quchen Fu, Sam Hays, Michael Sandborn, Carlos Olea, Henry Gilbert, Ashraf Elnashar, Jesse Spencer-Smith, and Douglas C. Schmidt. 2023. "A Prompt Pattern Catalog to Enhance Prompt Engineering with ChatGPT." https://arxiv.org/abs/2302.11382.

Wu, Shijie, Ozan Irsoy, Steven Lu, Vadim Dabravolski, Mark Dredze, Sebastian Gehrmann, Prabhanjan Kambadur, David Rosenberg, and Gideon Mann. 2023. "BloombergGPT: A Large Language Model for Finance." https://arxiv.org/abs/2303.17564.

Yong, Gunwoo, Kahyun Jeon, Daeyoung Gil, and Ghang Lee. 2023. "Prompt Engineering for Zero-Shot and Few-Shot Defect Detection and Classification Using a Visual-Language Pretrained Model." *Computer-Aided Civil and Infrastructure Engineering* 38 (11): 1536–54. https://doi.org/https://doi.org/10.1111/mice.12954.

"Your AI Pair Programmer." n.d. https://github.com/features/copilot; GitHub.

Yu, Yue, Yuchen Zhuang, Jieyu Zhang, Yu Meng, Alexander Ratner, Ranjay Krishna, Jiaming Shen, and Chao Zhang. 2023. "Large Language Model as Attributed Training Data Generator: A Tale of Diversity and Bias." https://arxiv.org/abs/2306.15895.

Zhang, Lei, Sean Howard, Tom Montpool, Jessica Moore, Krittika Mahajan, and Andriy Miranskyy. 2023. "Automated Data Validation: An Industrial Experience Report." *The Journal of Systems & Software* 197: 111573. https://doi.org/https://doi.org/10.1016/j.jss.2022.111573.

Zhao, Tony Z., Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. "Calibrate Before Use: Improving Few-Shot Performance of Language Models." https://arxiv.org/abs/2102.09690.