Bank of England

Revealing economic facts: LLMs know more than they say

Staff Working Paper No. 1,150
October 2025

Marcus Buckmann, Quynh Anh Nguyen and Ed Hill

Bank of England

Staff Working Paper No. 1,150

Revealing economic facts: LLMs know more than they say

Marcus Buckmann, (1) Quynh Anh Nguyen (2) and Ed Hill(3)

Abstract

We investigate whether hidden states of large language models (LLMs) can be used to estimate and impute economic and financial statistics. Focusing on county-level (eg unemployment) and firm-level (eg total assets) variables, we show that a linear regression trained on the hidden states of open-source LLMs outperforms the models' own text outputs. This indicates that internal representations encode richer economic information than is revealed directly in generated responses. A learning curve analysis shows that, in many cases, only a few dozen labelled examples suffice for training. We further propose a transfer learning method that improves estimation accuracy without requiring any labelled data for the target variable. Finally, we demonstrate the practical utility of hidden states in data imputation and super-resolution tasks.

Key words: Large language models, embeddings, economic statistics, data imputation.

JEL classification: C45, C81, C21.

---

(1) Bank of England. Email: marcus.buckmann@bankofengland.co.uk
(2) Bank of England. Email: quynhanh.nguyen@bankofengland.co.uk
(3) Bank of England. Email: ed.hill@bankofengland.co.uk

The views expressed in this paper are those of the authors, and not necessarily those of the Bank of England or its committees. We thank Max Bartolo, Philippe Bracke, David van Dijcke, Caspar Kaiser, Aleksandr Mattal, Paul Robinson, Arzu Ulu, and the participants of the CEBRA Annual Meeting 2024 and the ECONDAT 2025 spring meeting.

The Bank's working paper series can be found at www.bankofengland.co.uk/working-paper/staff-working-papers

Bank of England, Threadneedle Street, London, EC2R 8AH
Email: enquiries@bankofengland.co.uk

ISSN 1749-9135 (on-line)

1 Introduction

During training, generative large language models (LLMs) are exposed to vast amounts of information, including data relevant to economic modelling, such as geospatial statistics and firm-level financial metrics. If LLMs can effectively retrieve and utilise this knowledge, they could reduce dependence on external data sources that are time-consuming to access, clean, and merge, or that incur financial costs. Moreover, if LLMs accurately represent data, they could support downstream tasks like data imputation and outlier detection. In this study, we evaluate whether and how LLMs can be used for typical economic data processes.

Not all knowledge within an LLM may be explicit and retrievable in natural language by prompting the model. Instead, we hypothesise that LLMs have latent knowledge about entities such as firms or regions, which is only retrievable with access to the hidden states – also referred to as *embeddings* – of the LLM. For example, suppose we prompt an LLM to estimate the proportion of mortgages in arrears in Orange County, California. If this exact data point was not present in the training corpus, the model may be unable to provide an accurate written answer, despite having a generalised economic understanding of US counties that could inform an estimate. Furthermore, it is plausible that extensive post-training aimed at reducing hallucinations may have diminished a model's inclination or ability to make educated guesses. We tap into the LLM's generalised knowledge using its hidden states and find that these allow us to produce better estimates of economic and financial variables than the text output. While proprietary LLMs typically do not expose their hidden states, they are accessible in open-source models.

We test the usefulness of LLMs for data tasks with a large array of empirical analyses on regional economic data of the US, UK, EU, and Germany and financial data on US listed firms. First, we show that a regularised linear model learned on the hidden states often provides substantially better estimates of the economic and financial variables than the text output of the model, particularly for less common statistics. This result holds across open-source LLMs of varying sizes (1–70 billion parameters). A learning curve analysis finds that small samples of labelled data usually suffice to learn an accurate linear model on the embeddings. We also suggest a simple transfer learning algorithm that estimates a statistical variable without *any* labelled data for that variable. We achieve this by both learning from other labelled variables and using the text output of the LLMs as noisy labels on the variable of interest.

Another way to leverage an LLM's economic or financial knowledge is by using a reasoning paradigm that allows the model to break down the estimation task into steps, reflect on its knowledge and reasoning process, and refine its approach before arriving at an answer. While we find that a reasoning LLM outperforms the direct text output of a comparable model not tuned for reasoning, our approach—applying a linear model to the prompt's embeddings—remains superior. It not only delivers better results but is also orders of magnitude more computationally efficient.

Finally, we investigate the practical relevance of our findings for two data processes. Firstly, we show that we can exploit the embeddings to consistently improve the imputation of missing values in economic and financial variables, a common challenge in both research and industrial data pipelines. Secondly, we demonstrate that the hidden states of LLMs can be used for super-resolution tasks, inferring lower-level geographical statistics from high-level data, which can enable the estimation of statistics below the official reporting level. In both applications, we do not assume the LLM has seen the target statistics during training and can recall them by prompting; instead, we leverage the generalised, entity-level knowledge encoded in the model's hidden states.

Our findings support the value of LLMs in data processing tasks, which warrants attention, given that tasks such as identifying data sources, extracting numeric information, merging data series, and handling missing values are time-consuming and error-prone. This is evident from the numerous data providers that sell packaged, cleaned, and well-structured datasets, even when a large proportion of the underlying data is publicly accessible online.

This study is structured as follows. Section 1.1 discusses the relevant literature, Section 2.1 describes the datasets and Section 2 outlines our empirical approach. The main results on the accuracy of our approach to estimate economic variables from embeddings are shown in Section 3. Section 4 presents the methodology and results for two data processing tasks using embeddings: imputation and super-resolution. Section 5 concludes.

1.1 Literature review

Our study is most closely related to papers that investigate how well LLMs can recall geospatial statistics. Manvi et al. (2023) show that both GPT3.5 and fine-tuned Llama 2 can be prompted to reproduce geospatial information, such as population density, mean income, education, or house prices. The LLMs recall these variables more accurately than supervised machine learning methods, which inferred the information from other data sources such as satellite images of luminosity at night. Note that the authors only consider the text output of the LLM and do not exploit the hidden states as we do.

Similarly, Li et al. (2024) test how well LLMs can reproduce a large set of variables about cities and regions such as population density, life expectancy, average travel time, and number of patents. As in our study, this paper tests the accuracy of both the text output and a linear model on the hidden layer activations to estimate the statistics. However, they focus on a third approach: they ask the LLM to identify relevant features that can help to predict a variable of interest and then ask the LLM to generate values for these variables. After feeding these variables into a supervised ML model to predict the variable of interest, this approach performs on par with a linear model on the hidden states in 5-fold cross-validation. In contrast to our work, this paper focuses purely on how well the LLMs can recall statistics and does not consider transfer learning or downstream tasks.

Our main empirical approach of learning a linear model on hidden states of a language model is known as *linear probing* in the literature (Belinkov, 2022; Alain and Bengio, 2016) and has been used to understand the inner workings and capabilities of LLMs. Our study builds in particular on the findings by Gurnee and Tegmark (2023), showing that hidden states of LLMs linearly represent space and time. Concretely, a linear model trained on the hidden states of Llama 2 accurately infers the death of historical figures, the release year of artworks, or the geographical location of landmarks. Godey et al. (2024) investigate the scaling laws of these results, showing that larger models represent space and time more accurately. Further, Chen et al. (2023) perturb activations in the LLM to show that there is a causal connection between the representations and the text output.

Zhu et al. (2024) use linear probing to show that LLM embeddings represent the results of simple addition problems. While the exact number cannot be reconstructed using linear probes, the correlation between the actual number output and the inferred number (via the probes) is high. Linear probing is also used to gauge the truthfulness of statements (Marks and Tegmark, 2023; Orgad et al., 2024; Bürger et al., 2024) and Liu et al. (2024) show that hidden states can also be leveraged to estimate uncertainty of LLM responses in different tasks.

Our work also relates to studies that use hidden states of LLMs for downstream tasks such as classification. Particularly, Buckmann and Hill (2025) showed how a linear model trained on hidden states of small LLMs after task-specific prompting can outperform GPT-4 when trained on only a few dozen samples per class (see also Cho et al., 2023). Hidden states of LLMs have also proven to perform well as general text embeddings (Lee et al., 2024; Rui Meng, 2024) as evidenced by their competitive performance on the Massive Text Embedding Benchmark (MTEB) (Muennighoff et al., 2022), which includes tasks such as clustering, re-ranking, retrieval and classification.

Our transfer learning strategy draws on the literature on deep learning from noisy labels (see Song et al., 2022, for a review). Consistent with prior findings that deep networks can generalise beyond noisy supervision and achieve test accuracy above the quality of the training labels (Rolnick et al., 2017; Oyen et al., 2022), we observe that our model outperforms the noisy labels derived from LLM outputs. Moreover, following evidence that neural networks learn clean signal early and only later memorise noise, we employ early stopping to curb overfitting (Liu et al., 2020).

The literature on the use of LLMs for data imputation and super-resolution tasks – the two practical applications of our methodology that we explore in this study – is scarce. Hayat and Hasan (2024) fine-tuned Llama 2 to impute missing values given a textual description of the instance's features. Ding et al. (2024) use a similar approach for data imputation in recommender systems. By contrast, we do not rely on text output but use the embeddings of the LLM as additional features when employing standard imputation methods.

While there exists a rich statistical and machine learning literature on the estimation

Name	Observations	Variables	Regional level	Grouping variable	Year
US counties	3142	9	counties	51 states	2019
German districts*	401	13	districts	38 gov. districts	2019
UK districts	374	8	districts (LADs)	12 regions	2019
EU regions (NUTS2)	244	7	NUTS-2	27 countries	2019
US listed firms	1986	9	N/A	N/A	2022

*In German, the regional level is referred to as *Kreise und kreisfreie Städte* and the grouping level as *Regierungsbezirke*.

Table 1: Datasets. Appendix A provides full documentation of datasets and sources.

of geospatial statistics at a granular level (Tzavidis et al., 2018; Chi et al., 2022; Aiken et al., 2022; Viljanen et al., 2022), we are not aware of any study that explores the use of LLMs for this purpose.

2 Methodology

2.1 Datasets

We use four public regional datasets covering US counties, EU regions, UK districts and German districts. Additionally, we use a dataset containing public financial information about US listed firms. The datasets are listed in Table 1. We assembled the datasets using different public sources, which are described in detail in Appendix A. All datasets are cross-sectional and are based on statistics from 2019, with the exception of the US firms dataset for which we obtain data from 2022.[1] Intentionally, all datasets predate the 2023 knowledge cut-offs of the LLMs we tested so that we can assess whether the LLMs can exactly recall statistics.

Key variables that we use in the regional datasets are population, GDP per capita, unemployment rate, a measure of the income per capita, and a measure of life expectancy. Key variables we observe in the US firms dataset are total assets, market capitalisation, profitability metrics such as return on equity and return on assets. The full list of variables in all datasets are shown in Table AIII. For the super-resolution analysis in Section 4.2 we use additional data that we describe in that Section. We provide more extensive results for our main datasets, US counties and US listed firms, which contain a larger number of observations than the other datasets.

2.2 Experimental set-up

We exclusively use open-source LLMs, as, to the best of our knowledge, none of the providers of the leading proprietary models provide access to the hidden states of their generative models. Furthermore, we focus on smaller models because these are easier and cheaper to use. The models we test are listed in Table 2. In the main experiments,

[1]The free tier of the Yahoo Finance API limits access to earlier history.

Name	Source	Quantisation
Main models		
Llama 3.2 1B-Instruct	huggingface.co/meta-llama	no
Phi-3-Mini-4K-Instruct (3.8B)	huggingface.co/microsoft	no
Llama 3 8B-Instruct	huggingface.co/meta-llama	no
Llama 3 70B-Instruct	Text: replicate.com/meta	no
	Embeddings: huggingface.co/bartowski	IQ2_M.gguf
Reasoning experiments		
Qwen QwQ-32B	Text: groq.com: qwen-qwq-32b	
Qwen 2.5 32B	Text: groq.com: qwen-2.5-32b	
	Embeddings: huggingface.co/Qwen	q4_k_m.gguf

Table 2: LLMs used in this study.

we consider three models of the Llama 3 family (1B, 8B and 70B parameters) and the Phi-3-mini (3.8B parameters) model. Apart from Llama 3 70B, we accessed the models through the `transformers` library (Wolf et al., 2020) on an NVIDIA Tesla V100 GPU. Due to its large size, we use a quantised version of Llama 3 70B on 2 NVIDIA T4 GPUs and obtain the embeddings with the `llama-cpp-python` library. To obtain this model's text outputs, we use the non-quantised model hosted on `replicate.com`. More details on the computational requirements can be found in Appendix B.1.

We also consider a reasoning model: Qwen QwQ (32B parameters). To assess whether the reasoning paradigm improves the estimation of regional and financial statistics, we compare the model directly against Qwen 2.5 32B, which is the general-purpose LLM on which Qwen QwQ is based. To obtain the text output of both models, we use the instances hosted on `groq.com`. We obtain the embeddings of Qwen 2.5 32B from a quantised version using `llama-cpp-python`.

We prompt the models using a *completion prompt* as follows:

The {`variable`} *in* {`region`} *in* {`year`} *was*

For example, we use the prompt: "*The population in Orange County, California in 2019 was*". We also tested several other prompting strategies including a question-answering prompt, a few-shot prompt, and a chain-of-thought prompt. Appendix B.2 shows how the completion prompt delivered the most accurate results, on average.

When training a linear model on the embeddings, we also test a *generic prompt*, which does not vary with the variable of interest but just embeds the name of the entity and the year (e.g. *"Orange County, California in 2019"*).

2.3 Linear model on embeddings (LME)

We fit a ridge regression model on the hidden states of the prompt's last token. Our baseline model has 32 layers, with a 4096-dimensional embedding vector. If not stated otherwise, we use the embeddings of the 25^{th} layer because LME performs better on this

layer than on others (Figure AV in the appendix).

To estimate the performance of LME, we employ *grouped* cross-validation on the regional datasets. Specifically, we ensure that the training and test sets do not share any values of the grouping variable. The grouping variable for each dataset is shown in Table 1. For US counties, for example, the grouping variable is US states. Our baseline results are based on 25 repeats of repeated 5-fold cross-validation but we also report learning curves, where, adhering to the grouping, we sample training sets of increasing size.[2]

We either train the linear model on the embeddings directly or on the first k PCA components, with $k \in \{5, 10, 25, 50, 100, 200\}$. We fit the PCA to all observations and not only on the training set. The regularisation parameter α is tuned over 50 logarithmically spaced candidates between 10^{-5} and 10^{5}. The selected α minimises the mean squared error under 5-fold grouped cross-validation within the training set.

We denote the ground-truth values of the variable of interest by y, the embedding matrix by E, and the LLM's textual output by $\overset{txt}{y}$.

2.4 Data transformations and performance metrics

To avoid a strong influence of outliers on the performance metrics and on the estimation of LME, we transform some of the dependent variables. Specifically, across all datasets, we applied a log transformation to non-negative variables with a skewed distribution. For skewed variables with negative values, we use a cubic transformation ($f(x) = sgn(x) \times |x|^{\frac{1}{3}}$). The transformations applied to each variable are reported in Table AIII.

When assessing the performance of the text output and LME, we test both models on exactly the same observations. The text outputs of the LLMs sometimes fall outside the range of expected values (e.g. unemployment rate $> 50\%$). In this case, we decide to remove the observations. Parsing numeric values from LLMs' textual answers is challenging, due to the inconsistent format and inconsistent use of units. We made considerable effort to correctly parse the text output and conducted comprehensive manual checks on all variables to ensure the accuracy of our parsing approach. Our parsing approach is described in detail in Appendix B.3.

Despite the use of data transformations and the removal of obvious outliers, we still observe some outliers in the values parsed from the text output. To minimise the influence of these outliers on performance, we employ Spearman correlation as our main performance metric, measuring the rank correlation between the ground truth and estimated values of the variable of interest. We also report our key results for Pearson correlation and do not observe a qualitative difference in our conclusions.

[2]We repeat the learning curve sampling procedure between 30 and 300 times, depending on the dataset size and training sample size.

3 Using LLM embeddings to infer statistics

We start our analyses with a cross-validation exercise comparing the performance of LME and text output. Next, we run a learning curve analysis to understand how many training samples are required for LME to perform well. Finally, we test different transfer learning approaches, where we try to estimate a variable without access to any labelled data on that variable. In this section, we focus on the performance of the *baseline LME*, a ridge regression model trained on all 4096 embedding dimensions (i.e. without PCA) of the 25^{th} layer of Llama 3 8B.

3.1 Cross-validation

Figure 1 compares the performance (Spearman correlation) of the text output to the baseline LME learned on the embeddings of the completion prompt and the generic prompt. The variables are ordered by decreasing performance of the text output.[3] LME based on the completion prompt performs better than LME on the generic prompt in all datasets except the US firms data, where both perform equally well on most variables. We therefore focus on the completion prompt in the following analyses.

LME beats the text output most of the time. The performance advantage of LME increases with decreasing performance of the text output. On more common variables, such as population, or unemployment rate, the text output performs as well as, or better than, LME. On less common statistics, however, such as the proportion of mortgages that are at least 90 days delinquent (US counties) or the number of government employees (German districts), LME performs substantially better. The results are similar when choosing Pearson correlation as the performance metric, as shown in Table AIII and Figure AII in the Appendix.

Figure 2 plots the actual values against both the inferred values of LME (completion prompt, top row) and the text output (bottom row) on a few variables of the US counties dataset. The text output exactly recalls the population in a large number of counties, particularly for those counties with a large population. For the other variables, however, the text output is clustered on a few distinct values.[4,5] By contrast, the distribution of LME predictions does not cluster but also does not reproduce the exact values.

For the US states dataset, we also evaluate the performance of the text output and LME *within* states. To obtain stable estimates we exclude states that have less than 50 counties. Figure AVI in the appendix compares the Spearman correlation between the ground truth and the text output and LME, respectively. Except for the population

[3] See Table AIII for a tabular representation of the results including a measure of uncertainty, which we omitted from these charts to improve legibility.

[4] Clustering of values in the text outputs has also been observed by Li et al. (2024) in their analysis of LLMs' knowledge of regional statistics.

[5] Note that the values are not just clustered by US state. We typically observe several unique values within states as shown by Figure AVII which depicts the number of unique ground-truth and text output values for each state.

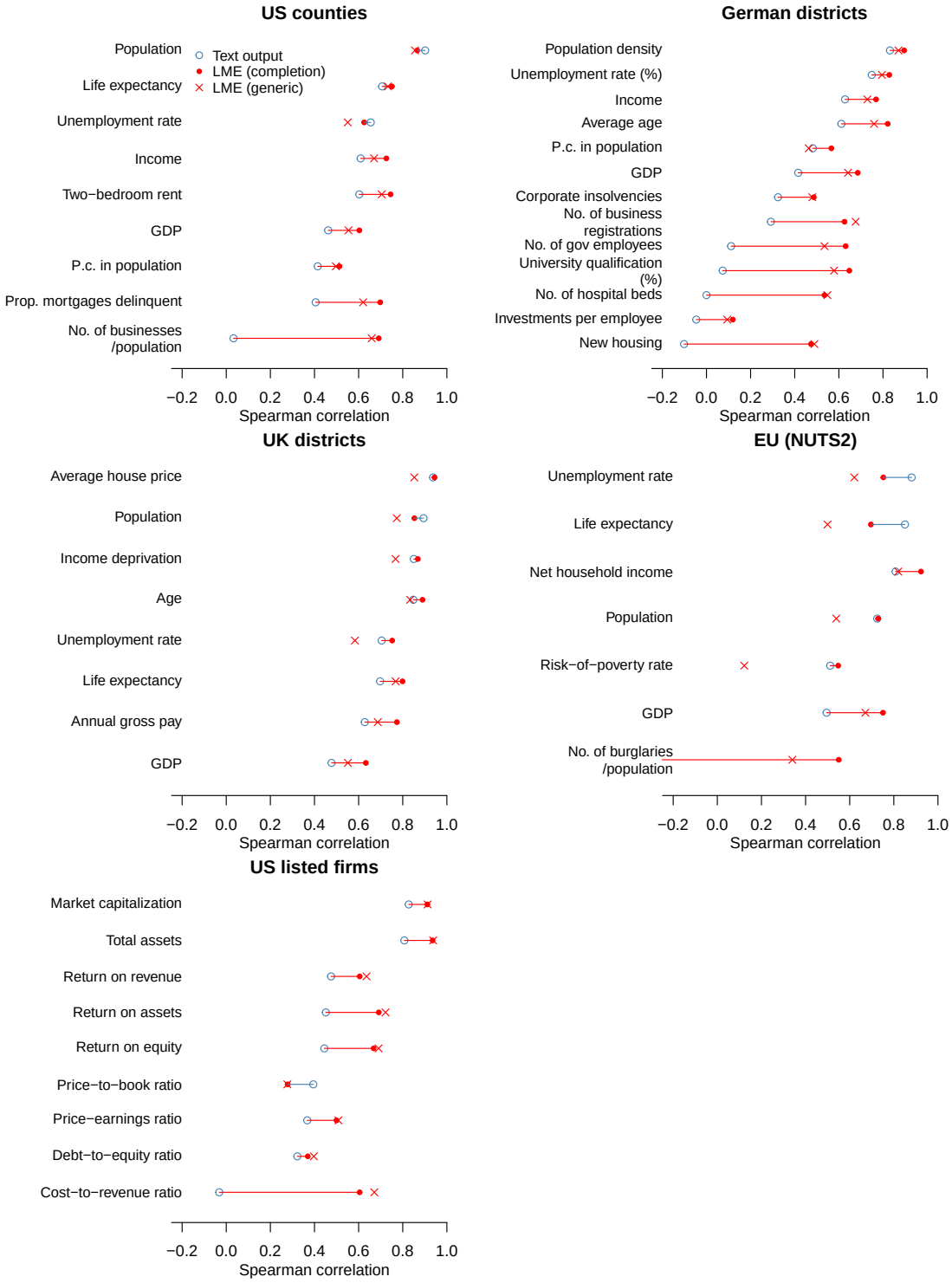

Figure 1: Cross-validation performance. The variable labels are shortened. The full variable names used when querying the LLM are shown in Table AIII.

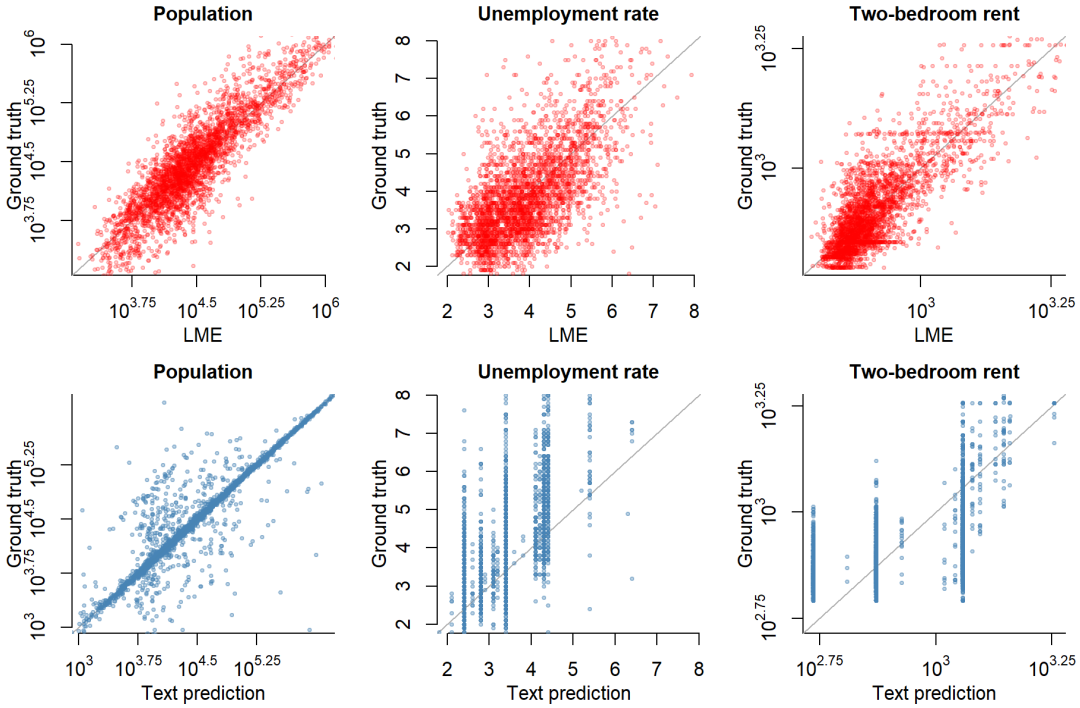

Figure 2: Comparing actual values (ground truth) to the predicted values by LME (top row) and the text output (bottom row).

variable, we observe a performance advantage of LME over the text output in most states.[6]

3.2 Robustness

We test whether we can improve the performance of LME by reducing the dimensionality of the embeddings using a PCA before training LME. For our two key datasets, US counties and US firms, Figures AIII and AIV in the appendix respectively show that LME usually performs best without applying PCA.

We also compare the performance by layer and see that across the key datasets, embeddings from the 25^{th} layer perform better on average than those from earlier or later layers (Figure AV in the appendix). However, the differences in performance are small with the exception of the lower performance when using layer 5, the earliest layer we examine.

[6]We cannot show the variable *proportion of mortgages being delinquent* because it has a low coverage in the raw data, which is why we do not observe at least 50 non-missing observations in any state.

3.2.1 Comparing base models

To test whether our finding that LME tends to beat the text output generalises across model families and model sizes, we run the cross-validation analysis on our two largest datasets (US counties and US listed firms) for all four LLMs.

Figure 3 (left panel) shows how the performance of LME improves consistently with increasing model size.[7] The middle panel of the figure shows that the accuracy of the text output also increases with larger models, but less consistently. The right panel shows the difference in Spearman correlation of the two approaches (LME - text output). While the mean difference (in red) is larger for the small models, it is still large – on average 16 percentage points – for the 8B and 70B model.

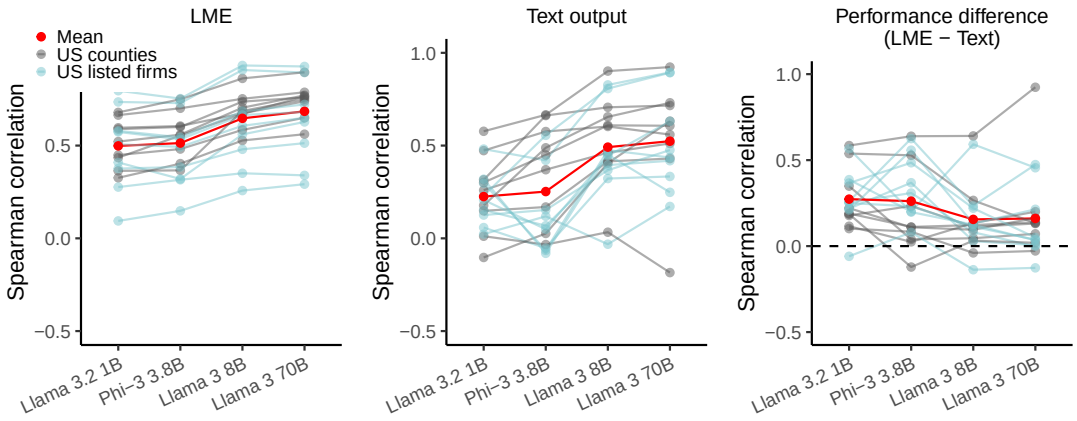

Figure 3: Performance of LLMs of different sizes.

3.2.2 LME vs. a reasoning model

Since 2024, it has emerged that the performance of language models does not only scale with training-time compute (i.e. larger models and more training data) but also test-time compute. Allowing an LLM more time to generate responses, break down tasks into multiple steps, and reflect on its reasoning process significantly enhances its performance on reasoning tasks. We test whether this reasoning paradigm is also useful for our specific task: Can an LLM accurately reason about its internal knowledge of regions and firms, integrate this information, and ultimately provide precise estimates of relevant statistics?

We use the open-source reasoning model Qwen QwQ-32B (Qwen Team, 2025). It is based on Qwen 2.5 32B (Yang et al., 2024), an LLM not tuned to reason using test-time

[7] As the models differ in their number of layers, we train LME on the embeddings of the last layer of each model. Furthermore, while LME on the embeddings of the Llama models work best when no dimensionality reduction is applied, Phi generally performs better when training LME on 25 PCA components. Thus, we applied the dimensionality reduction for Phi.

compute. By comparing these two models, we can directly measure the effect of the reasoning paradigm on accuracy.[8] We pit the text output of the two models against LME trained on the embeddings of the final token of the prompt of the Qwen base model (Qwen 2.5 32B). Due to the high computational costs of the reasoning model, we only test it on a subsample of 500 entities per variable.

A closer look at the reasoning process reveals that the LLM tries to reconcile different information to provide the best estimates based on its knowledge.[9] However, despite the efforts, we do not observe a consistent increase in performance.

Figure 4 shows the results for our two main datasets. Overall, we do not see a consistent improvement in performance when using the reasoning model to generate the text output. However, the reasoning model performs substantially better on the three US county variables on which the non-reasoning model performs worst. LME beats the text output of both models on most variables. The difference in computational costs is significant. The reasoning model uses a median of 1767 output tokens, whereas Qwen 2.5 answers the question with a median of 8 tokens.

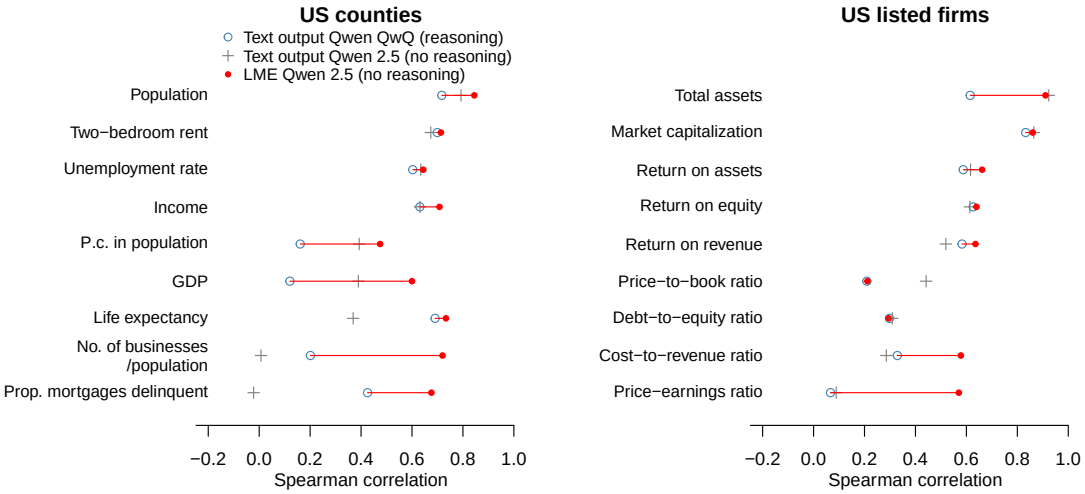

Figure 4: Cross-validation performance comparing reasoning model Qwen QwQ (text output) to Qwen 2.5 (text output + LME).

[8] In contrast to the previous experiments where we use a completion prompt, we use the question-answering prompt (see Section B.2) for the two Qwen models in order to elicit reasoning behaviour.

[9] For example, when asked for the proportion of delinquent mortgages in Atlantic County, New Jersey in 2019 one of many points the reasoning model made was "Wait, I think that in 2019, the national delinquency rate was around 3-4%, but that's the national average. Atlantic County might be different, especially if it's a coastal area that might have been affected by Hurricane Sandy in 2012, but that's a few years prior. Maybe the recovery from that could affect 2019 numbers? Not sure." Another sentence from the same answer: "Hmm, I'm not entirely sure, but I think the national average was around 1.8%, and New Jersey might be a bit higher. Since Atlantic County is a county with some economic challenges, maybe it's a bit higher than the state average."

3.3 Learning curves

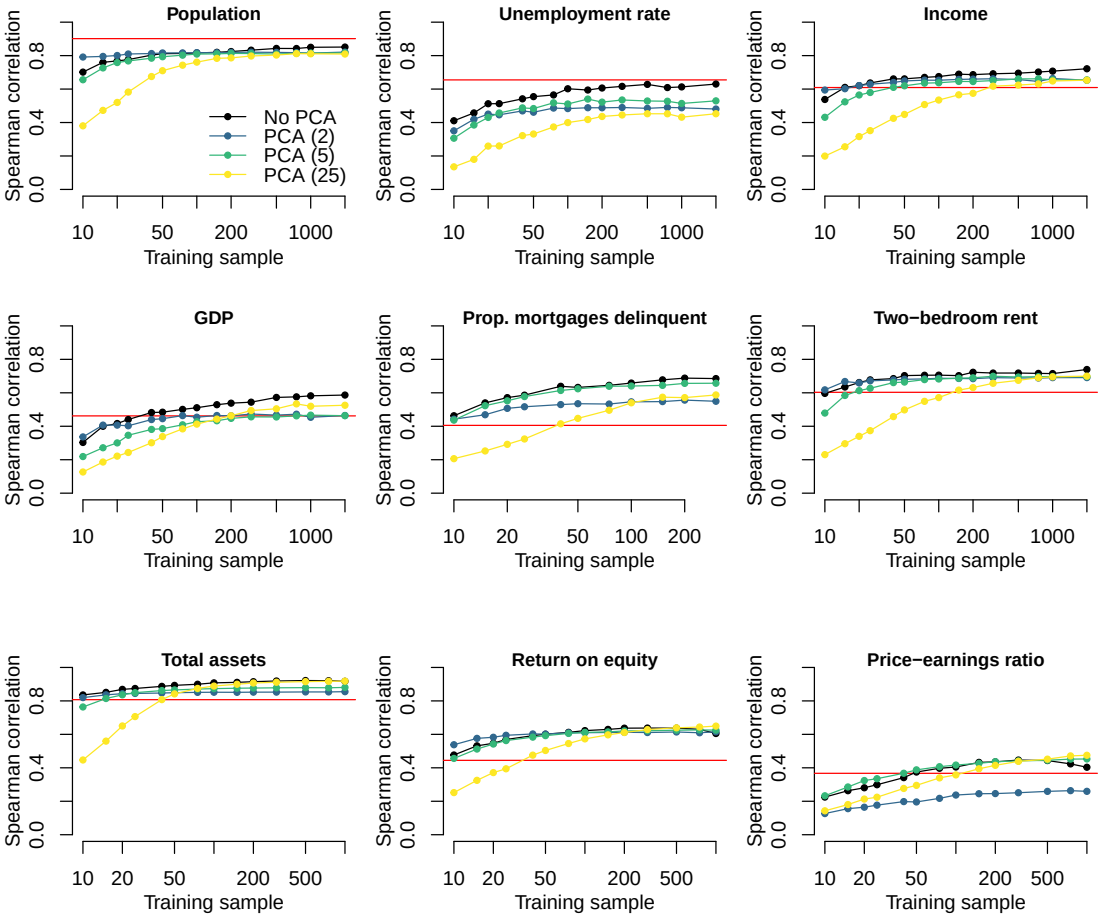

Figure 5: Learning curve analysis.

The cross-validation results show that we can infer statistical variables more accurately from the embeddings than from the text output. However, the 5-fold cross-validation setting assumes that 80% of the values on the variable we are predicting are available to the modeller. If LME can reach a high accuracy with only a small number of labelled data points, its relevance for data processing tasks will be much higher. To test this, we conduct a learning curve analysis, estimating model performance as a function of the number of training samples. Figure 5 compares the text output (red line) with LME trained on progressively larger training sets from the US counties and US firms datasets. LME's performance often converges with only a small number of training samples. Dimensionality reduction using PCA improves performance on some variables for very small sample sizes and when restricted to two components. Figure 6 summarises the learning-curve results across all datasets, comparing the text output with the baseline LME at training sample sizes of 25, 50, and 100 observations. With only 25 samples,

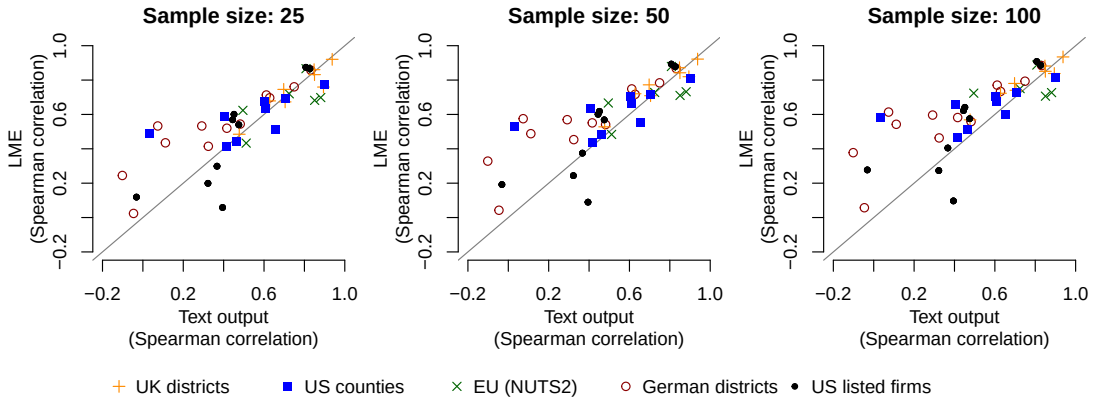

Figure 6: Comparison between text output and baseline LME with different training samples.

LME outperforms the text output for most variables, showing a mean Spearman correlation 5 percentage points higher.

3.4 Transfer learning

3.4.1 Learning from embeddings of other variables

The previous analyses assume that we have at least some labelled instances of a variable to estimate its values on other instances using the embeddings. Here, we test whether we can reliably estimate a regional or firm variable without access to any values of this variable. If the embeddings encode numbers in a consistent way across different variables, we should be able to predict variable v_{test} using a model learned from other variables. More formally, given all variables $\mathbb{V}$ of a dataset with n entities, we train a regression model on embeddings E_i and outcome values y_i of entities i, and test the models on variable v_{test}. Training observations are described as $\{(y_{vi}, E_{vi}) : v \in \mathbb{V} \setminus \{v_{test}\}, i = 1, ..., n\}$ and test observations as $\{(y_{vi}, E_{vi}) : v = v_{test}, i = 1, ..., n\}$. Note that $E_v \neq E_w, \forall v \neq w$, as we use the completion prompt to obtain the embeddings, which also contains the name of the variables.

To ensure our model is flexible enough to learn generalised embeddings, we train a neural network with two layers (128 neurons and 32 neurons, Leaky ReLU activation function) as our transfer learning model. We use the Adam optimiser with an initial learning rate of 10^{-5}. To avoid overfitting, we only learn for 20 epochs and apply a strong dropout of 0.5 on the first layer. We test this approach on all five datasets. All variables are transformed as described in Section 2.1 before they are normalised (mean of zero, standard deviation of one).

The left panel of Figure 7 compares the performance (Spearman correlation) of this approach to the text output. Each dot reflects one variable, for which a separate model

was trained on all other variables of the same dataset. This approach to transfer learning does not beat the text output. While transfer learning is superior in some cases, it falls behind the text output most of the time. We obtain similar results when using a linear model instead of the neural network. We also test another simple transfer learning approach, where we predict variable y in Dataset A from the same variable of different datasets. For example, we train a model on the unemployment rate of the UK and Europe to predict the unemployment rate in the US. However, this transfer learning approach is also inferior to the text output on average, and sometimes by a large margin.

Collectively, these transfer learning results show that embeddings do not generalise reliably to other variables.

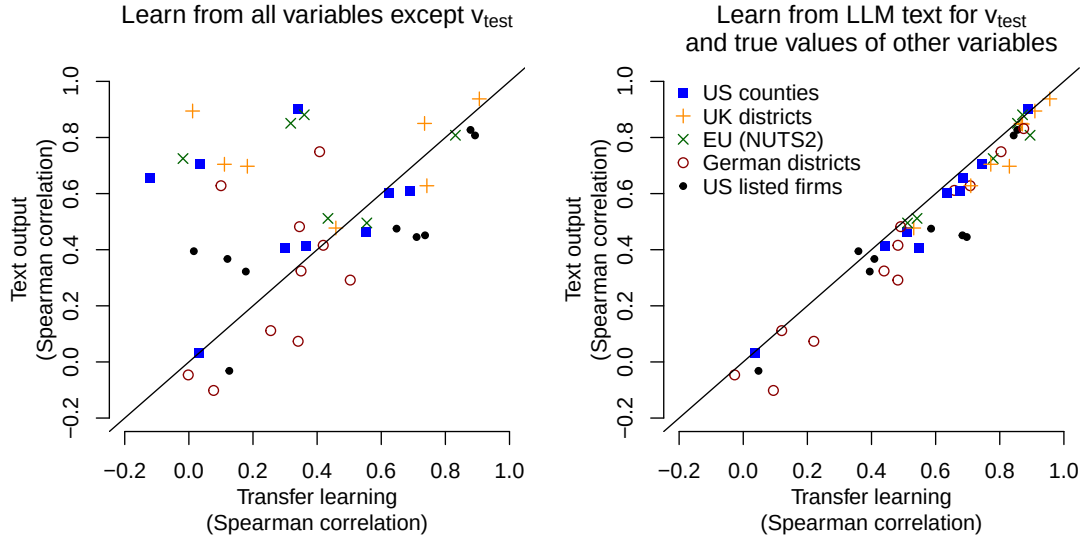

Figure 7: Transfer learning analysis.

3.4.2 Learning from text output

Given the poor performance of the simple transfer learning approaches, we refine our method to exploit information about the variable we predict (v_{test}), even without access to ground truth values. Concretely, we use the text outputs of the LLM as noisy or biased labels of the variable of interest in the training set ($\overset{txt}{y}_{v_{test}}$). For the remaining variables we use the ground truth labels, as above. Formally, let $y^{\dagger}_{vi} := \begin{cases} y_{vi}, & v \neq v_{test} \\ \overset{txt}{y}_{vi}, & v = v_{test} \end{cases}$ and the training set be $\{ (x_{vi}, y^{\dagger}_{vi}) : v \in \mathbb{V},\ i = 1, \ldots, n_v \}$. The test set, as before, is denoted as $\{(y_{vi}, E_{vi}) : v = v_{test}, i = 1, ..., n\}$. Even though the text labels are often inaccurate, they might provide enough information for the model to learn from the embeddings of the specific prompt of v_{test}.

We use the same neural network set-up as above. The right panel of Figure 7 compares

this performance of this transfer learning approach to the text output. This transfer learning approach almost always outperforms the text output even when the text output itself is accurate, on average by 7.2 percentage points. This is an important finding. We can extract knowledge about a specific economic or financial variable from LLM embeddings *without any* labelled instances of that variable. The result is robust to the choice of the hyperparameters of the neural network and we observe qualitatively similar, but slightly inferior performance, when using a linear model instead of the neural network.

4 Leveraging LLM embeddings for imputation and super-resolution

4.1 Imputation

Our analyses have established that LLMs have rich knowledge about entities such as regions and firms. Here we test whether this knowledge can be exploited to improve the imputation of missing values, a common challenge in economic modelling. As in Section 3, our approach does not assume the model has memorised the exact statistic to be imputed or that it can be recovered by prompting. Instead, we use the model's embeddings as features, leveraging their generalised knowledge of entities. This makes the method applicable when the ground-truth statistic has not previously been produced or reported for certain entities and therefore can only be estimated.

To test whether embeddings can improve imputation accuracy, we compare two imputation strategies on our five datasets. The *baseline* strategy estimates imputed values using only the K numeric features (e.g. population and unemployment rate) in the dataset. The *embedding* approach extends the feature matrix by appending the first c PCA components of the embeddings.

To obtain embeddings for the entities, we use generic prompts (see Section 2.2), which only contain the name of the entity (e.g. region or firm) and the year. Although some of our datasets contain missing values, we do not impute them for evaluation because the true values are unknown, precluding accuracy assessment. Instead, we simulate missingness by randomly masking observed entries.

Our approach to introduce missing values is as follows: Given a dataset with N observations and K variables, all variables are transformed as described in Section 2.1 before they are normalised (mean of zero, standard deviation of 1). First, we sub-sample k features, for each of which we randomly mask $p\%$ of its values, creating the incomplete data D_{miss}. We then impute the missing values in D_{miss} using imputation algorithm m.

We test three imputation methods m: using a low-rank Gaussian copula model (Zhao and Udell, 2020) and iterative multivariate imputation using (i) Bayesian Ridge regression and (ii) random forest (Wilson, 2020). We assess the accuracy of the imputation with the mean absolute error (MAE) between the ground truth and the reconstructed

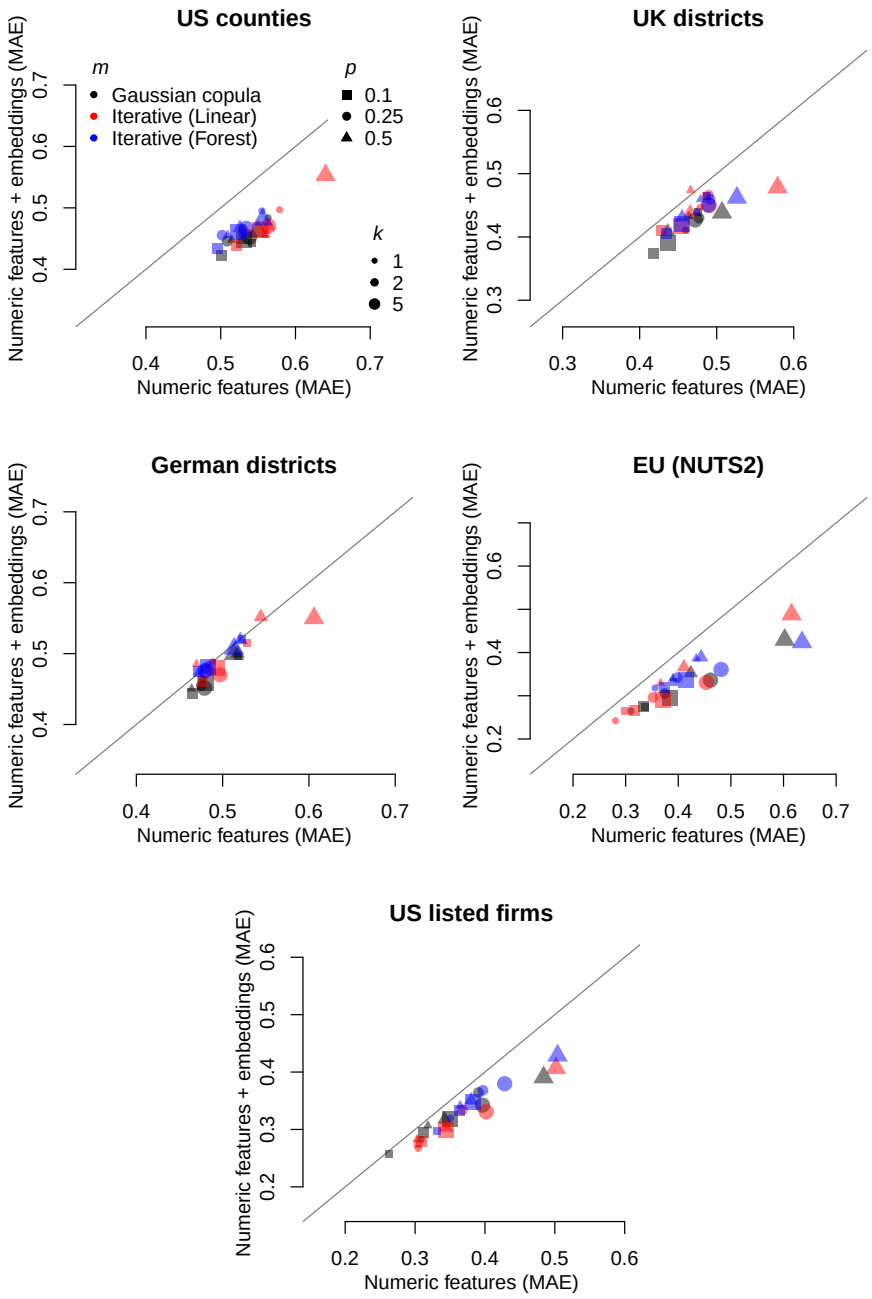

Figure 8: Imputation error when including or excluding the embedding vectors as features in the imputation method.

values.

	Regional level	
Region	lower	higher
US	3142 counties	51 states
EU (NUTS3)	1165 NUTS-3	244 EU NUTS-2
Germany	401 districts	38 gov. districts

Table 3: Regional levels of super-resolution datasets

For each imputation method, we conduct a large array of experiments to ensure robustness of our results. Specifically, we consider all possible combinations of the parameter settings $k \in \{1, 2, 5\}, p \in \{0.1, 0.25, 0.5\}$. To obtain stable results, we replicate the experiments for each parameter combination 25 times, randomly choosing the instances and features which are imputed. As embedding features, we use the first $c = 25$ PCA components of the 25^{th} layer of Llama 3 8B-Instruct.

Figure 8 presents the results across all five datasets and parameter settings m, k, and p. The embedding strategy consistently outperforms the baseline strategy. The improvement is particularly notable in some datasets. For instance, in the US counties dataset, the embedding strategy reduces the mean error by 14% on average across parameter settings.

4.2 Super-resolution

We define super-resolution as the estimation of statistics at a lower geographical level using information available at a higher level when data for the lower level are unavailable or unreliable. Like imputation, this task has high practical relevance. Many economic indicators are published only at higher administrative levels, despite systematic subregional heterogeneity that we may wish to estimate. LME, exploiting an LLM's generalised knowledge about regions, could provide accurate estimates in this situation. For evaluation, however, lower-level ground truth is required, which is why we test our super-resolution approach only on data for which lower-level statistics are reported.

We approach the super-resolution task by training LME on a higher regional level (e.g. unemployment rate of US states) to obtain estimates at the lower regional level (e.g. unemployment rate of US counties). We use the completion prompt for both geographical levels. We test this approach on three of our datasets (US, EU, Germany) for which we were able to find data on two regional levels, as shown in Table 3.

We compare this super-resolution approach with the text output at the lower level and with a *naive* method, which projects higher-level region values onto the lower level. For example, under the naive method, the unemployment rate of all counties in Alabama is set equal to the unemployment rate reported for the state of Alabama. Figure 9 presents the results, ordering the variables by decreasing performance of the text output.

The super-resolution approach performs well, surpassing the text output on most vari-

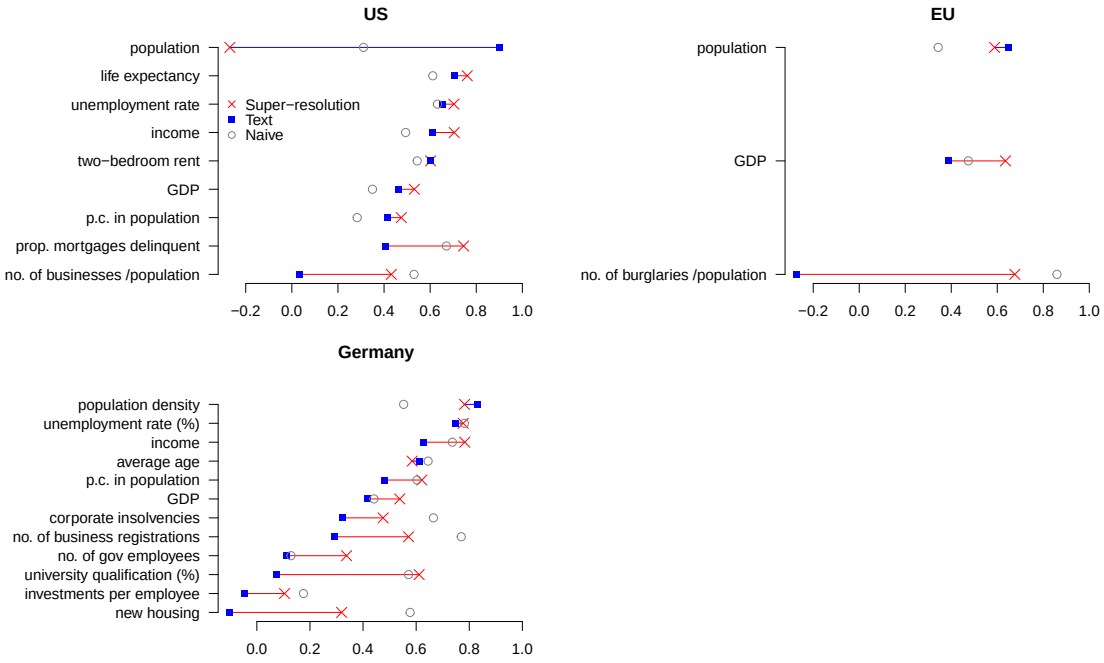

Figure 9: Super-resolution analysis.

ables. A notable outlier is population, where the super-resolution performs worse than random. The naive approach performs well on many of the variables in Germany but falls behind the other two approaches on the US data. Given the small training sample size, in particular for the US and Germany (51 and 38 observations at the higher geographical level, respectively), the performance of the super-resolution approach is impressive. Super-resolution is a particular transfer learning approach and thus supports our previous finding (Section 3.4) that the embeddings can be used for accurate estimation outside the domain of the training set.

5 Conclusion

We have shown that LLMs know more than they say. A linear model trained on the LLM's hidden states of only a few dozen labelled samples often outperforms the LLM's text output in the estimation of statistics – in particular when the target variable is a less common statistic. A key contribution of our work is the transfer learning approach which shows that we can estimate target variables more accurately than the text output without any labelled data on that target variable. Finally, we provide robust evidence that the hidden states can be used for data processing tasks. In particular, adding a low-rank representation of the hidden states to the feature matrix consistently improves the performance of standard imputation approaches in all of our datasets. Our super-resolution results show that, in most cases, we can beat the text output in esti-

mating statistics on a lower geographical level by learning from the higher geographical level.

We considered several LLMs with 1 to 70 billion parameters and show that our result that LLMs know more than they say holds across all of them, including a reasoning model that uses extensive test-time compute. We leave it for future work to test whether this finding also holds for the largest and most accurate open-source models such as Llama 3 405B or DeepSeek V3 with 671 billion parameters. In this study we exclusively worked with public data sources; it would be interesting to examine the extent to which our results hold for proprietary datasets, which are less likely to be observed in the training corpus of open-source LLMs. Finally, future research could investigate how useful the hidden states representing firms or geographic entities can be for other data processing tasks, such as outlier detection.

References

Aiken, E., S. Bellue, D. Karlan, C. Udry, and J. E. Blumenstock (2022). Machine learning and phone data can improve targeting of humanitarian aid. *Nature 603*(7903), 864–870.

Alain, G. and Y. Bengio (2016). Understanding intermediate layers using linear classifier probes. *arXiv preprint arXiv:1610.01644*.

Belinkov, Y. (2022). Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics 48*(1), 207–219.

Buckmann, M. and E. Hill (2025). Improving text classification: Logistic regression makes small LLMs strong and explainable'tens-of-shot'classifiers. *Bank of England Staff Working Paper Series* (1127).

Bürger, L., F. A. Hamprecht, and B. Nadler (2024). Truth is universal: Robust detection of lies in LLMs. *arXiv preprint arXiv:2407.12831*.

Chen, Y., Y. Gan, S. Li, L. Yao, and X. Zhao (2023). More than correlation: Do large language models learn causal representations of space? *arXiv preprint arXiv:2312.16257*.

Chi, G., H. Fang, S. Chatterjee, and J. E. Blumenstock (2022). Microestimates of wealth for all low- and middle-income countries. *Proceedings of the National Academy of Sciences 119*(3), e2113658119.

Cho, H., H. J. Kim, J. Kim, S.-W. Lee, S.-g. Lee, K. M. Yoo, and T. Kim (2023). Prompt-augmented linear probing: Scaling beyond the limit of few-shot in-context learners. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Volume 37, pp. 12709–12718.

Ding, Z., J. Tian, Z. Wang, J. Zhao, and S. Li (2024). Semantic understanding and data imputation using large language model to accelerate recommendation system. *arXiv preprint arXiv:2407.10078*.

Godey, N., É. de la Clergerie, and B. Sagot (2024). On the scaling laws of geographical representation in language models. *arXiv preprint arXiv:2402.19406*.

Gurnee, W. and M. Tegmark (2023). Language models represent space and time. *arXiv preprint arXiv:2310.02207*.

Hayat, A. and M. R. Hasan (2024). Claim your data: Enhancing imputation accuracy with contextual large language models. *arXiv preprint arXiv:2405.17712*.

Lahti, L., J. Huovari, M. Kainu, and P. Biecek (2017). Retrieval and analysis of Eurostat open data with the eurostat package. *The R Journal 9*(1), 385–392.

Lahti, L., J. Huovari, M. Kainu, P. Biecek, D. Hernangomez, D. Antal, and P. Kantanen (2023). eurostat: Tools for Eurostat open data. R package version 4.0.0.

Lee, C., R. Roy, M. Xu, J. Raiman, M. Shoeybi, B. Catanzaro, and W. Ping (2024). NV-Embed: Improved techniques for training LLMs as generalist embedding models. *arXiv preprint arXiv:2405.17428*.

Li, Z., Y. Wang, Z. Song, Y. Huang, R. Bao, G. Zheng, and Z. J. Li (2024). What can LLM tell us about cities? *arXiv preprint arXiv:2411.16791*.

Liu, L., Y. Pan, X. Li, and G. Chen (2024). Uncertainty estimation and quantification for LLMs: A simple supervised approach. *arXiv preprint arXiv:2404.15993*.

Liu, S., J. Niles-Weed, N. Razavian, and C. Fernandez-Granda (2020). Early-learning regularization prevents memorization of noisy labels. *Advances in Neural Information Processing Systems 33*, 20331–20342.

Manvi, R., S. Khanna, G. Mai, M. Burke, D. Lobell, and S. Ermon (2023). GeoLLM: Extracting geospatial knowledge from large language models. *arXiv preprint arXiv:2310.06213*.

Marks, S. and M. Tegmark (2023). The geometry of truth: Emergent linear structure in large language model representations of true/false datasets. *arXiv preprint arXiv:2310.06824*.

Muennighoff, N., N. Tazi, L. Magne, and N. Reimers (2022). MTEB: Massive text embedding benchmark. *arXiv preprint arXiv:2210.07316*.

Orgad, H., M. Toker, Z. Gekhman, R. Reichart, I. Szpektor, H. Kotek, and Y. Belinkov (2024). LLMs know more than they show: On the intrinsic representation of LLM hallucinations. *arXiv preprint arXiv:2410.02707*.

Oyen, D., M. Kucer, N. Hengartner, and H. S. Singh (2022). Robustness to label noise depends on the shape of the noise distribution. *Advances in Neural Information Processing Systems 35*, 35645–35656.

Qwen Team (2025, March). QwQ-32B: Embracing the power of reinforcement learning. https://qwenlm.github.io/blog/qwq-32b/.

Rolnick, D., A. Veit, S. Belongie, and N. Shavit (2017). Deep learning is robust to massive label noise. arxiv 2017. *arXiv preprint arXiv:1705.10694*.

Rui Meng, Ye Liu, S. R. J. C. X. Y. Z. S. Y. (2024). SFR-Embedding-Mistral: Enhance text retrieval with transfer learning. Salesforce AI Research Blog.

Song, H., M. Kim, D. Park, Y. Shin, and J.-G. Lee (2022). Learning from noisy labels with deep neural networks: A survey. *IEEE Transactions on Neural Networks and Learning Systems 34*(11), 8135–8153.

Tzavidis, N., L.-C. Zhang, A. Luna, T. Schmid, and N. Rojas-Perilla (2018). From start to finish: a framework for the production of small area official statistics. *Journal of the Royal Statistical Society Series A: Statistics in Society 181*(4), 927–979.

Vasilopoulos, K. (2023). *onsr: Client for the 'ONS' API.* R package version 1.0.2.

Viljanen, M., L. Meijerink, L. Zwakhals, and J. Van de Kassteele (2022). A machine learning approach to small area estimation: predicting the health, housing and well-being of the population of Netherlands. *International Journal of Health Geographics 21*(1), 4.

Wilson, S. (2020). Miceforest: Fast, memory efficient imputation with LightGBM. *Github https://github. com/AnotherSamWilson/miceforest*.

Wolf, T., L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, et al. (2020). Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 38–45.

Yang, A., B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Li, D. Liu, F. Huang, H. Wei, H. Lin, J. Yang, J. Tu, J. Zhang, J. Yang, J. Yang, J. Zhou, J. Lin, K. Dang, K. Lu, K. Bao, K. Yang, L. Yu, M. Li, M. Xue, P. Zhang, Q. Zhu, R. Men, R. Lin, T. Li, T. Tang, T. Xia, X. Ren, X. Ren, Y. Fan, Y. Su, Y. Zhang, Y. Wan, Y. Liu, Z. Cui, Z. Zhang, and Z. Qiu (2024). Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*.

Zhao, Y. and M. Udell (2020). Matrix completion with quantified uncertainty through low rank gaussian copula. *Advances in Neural Information Processing Systems 33*, 20977–20988.

Zhu, F., D. Dai, and Z. Sui (2024). Language models know the value of numbers. *arXiv preprint arXiv2401.03735*.

Çetinkaya Rundel, M., D. Diez, and L. Dorazio (2021). *usdata: Data on the States and Counties of the United States.* R package version 0.2.0.

Appendix

A Datasets

A.1 US counties

The data on US counties for the year 2019 is one of our two main datasets and we used a variety of sources to assemble it. We obtained income and GDP data from the website of the Bureau of Economic Analysis[10]. The unemployment data was sourced from the Economic Research Service of the US Department of Agriculture[11]. Life expectancy data is provided by the Institute for Health Metrics and Evaluation in the Global Health Data Exchange database. We obtained mortgage delinquency rates from the Consumer Financial Protection Bureau.[12] Data on the number of businesses was downloaded from the County Business Patterns Tables from the United States Census Bureau.[13] Information on the median rent is available from the website of the Office of Policy Development and Research.[14] The remaining variables we use come from the R package `usdata` (Çetinkaya Rundel et al., 2021) which lists the US Census and the American Community Survey as sources.

A.2 UK regions

We gathered data for the year 2019 for the 374 local authority districts (LADs) in the United Kingdom. Most variables were sourced from the Office for National Statistics (ONS) using API access via the R package `onsr` (Vasilopoulos, 2023). Some series are not directly accessible via API and were downloaded from the official websites of the ONS, Northern Ireland Statistics and Research Agency (NISRA) and the Scottish and Welsh government websites.

A.3 EU regions

All the EU series for the year 2019 are collected from Eurostat, an official database of the European Union. We access the series at the NUTS2 and NUTS3 levels via API calls using the R package `eurostat` (Lahti et al., 2017, 2023). Note that fewer series are available on the NUTS3 level.

A.4 German districts

All series for the year 2019 are downloaded directly from the Regionalatlas Deutschland[15] by the Federal Statistical Office of Germany.

[10] `https://www.bea.gov/data`

[11] https://www.ers.usda.gov/data-products/county-level-data-sets

[12] `https://www.consumerfinance.gov/data-research/mortgage-performance-trends/download-the-data`

[13] `https://www.census.gov/programs-surveys/cbp/data/tables.html`

[14] `https://www.huduser.gov/portal/datasets/50per.html`

[15] `https://regionalatlas.statistikportal.de/`

A.5 US listed firms

We collected financial information as of the end of 2022[16], including balance sheet, income and share price data, of US listed firms using the Yahoo Finance[17] free API access. We excluded values on the variables *return on revenue* and *return on equity* when firms have negative values on stockholders' equity or total revenue, respectively.

B Methodology

B.1 Computational requirements

We used an NVIDIA Tesla V100 with 16 GB of video memory to generate the LLM's embeddings and generate text. With 16-bit (fp16) parameters, this memory was sufficient for all models except Llama 70B variant. Using low-priority virtual machines, compute costs were under US $0.80 per hour for the GPU. Our baseline Llama 3 8B produced embeddings for a query in 0.1 seconds. We also tested the speed of a quantised version of Llama 3 8B running locally on an AMD Ryzen 7 3700X CPU. This required 0.9 seconds per query.

B.2 Prompting

In addition to the baseline completion prompting strategy (see Section 2.2), we also tested three alternative strategies. First, we apply the standard prompting template that is provided via the `apply_chat_template` function in the Transformers package (Wolf et al., 2020), referred to as question-answer prompting in this paper. For example, *What was the unemployment rate in Orange County, California in 2019?* This led to a lower response rate for US listed firms, with the model occasionally refusing to give an answer. Second, we test 5-shot prompting, where we provide 5 randomly chosen Q&A pairs to the model to learn from. Finally, we use chain-of-thought prompting, adding the phrase "think carefully before giving an answer" to the system prompt. The system prompts for the different prompting strategies are shown in Table AI.

Table AII indicates the response rates and Figure AI compares the text output's performance of different prompting strategies. On average, the completion strategy has the highest response rate and performs best.

Variable	Completion	Chain-of-thought	Few-shot	Q&A
US counties				
GDP per capita (in $)	**1.00**	**1.00**	0.99	**1.00**
life expectancy at birth	**1.00**	**1.00**	**1.00**	**1.00**
median monthly two-bedroom rent (in $)	**1.00**	**1.00**	0.98	**1.00**
personal income per capita (in $)	**1.00**	**1.00**	0.99	**1.00**

[16]We could not access data from 2019, as we do for the other datasets, because the free API access is limited to the most recent years.

[17]`https://finance.yahoo.com/`

population	**1.00**	**1.00**	**1.00**	**1.00**
unemployment rate	**1.00**	0.99	0.99	**1.00**
number of business establishments per 100,000 people	**1.00**	**1.00**	**1.00**	**1.00**
percentage change in population compared to the previous year	**1.00**	**1.00**	0.98	**1.00**
proportion of mortgages being 90 or more days delinquent	**1.00**	0.99	0.20	**1.00**
US listed firms				
cost-to-revenue ratio	**1.00**	0.97	0.69	0.83
debt-to-equity ratio	**1.00**	0.96	0.99	0.79
market capitalization	**1.00**	0.94	0.99	0.97
price-earnings ratio	**1.00**	0.88	0.99	0.91
price-to-book ratio	**1.00**	0.95	0.99	0.92
return on assets	**1.00**	0.88	0.99	0.96
return on equity	**1.00**	0.92	0.82	0.88
return on revenue	**1.00**	0.98	0.84	0.91
total assets	**1.00**	0.98	0.99	0.96

Table AII: Response rate when using different prompting strategies

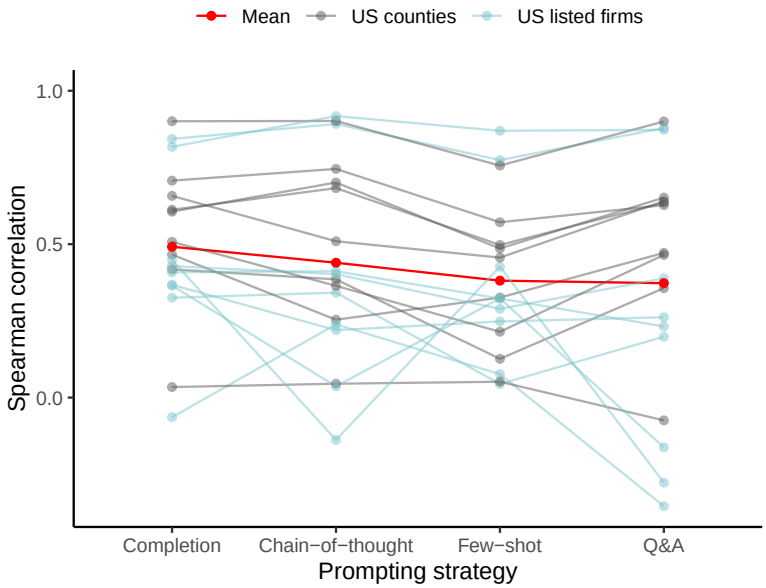

Figure AI: Performance of text output when using different prompting strategies.

B.3 Extraction of numeric estimates from text output

Extracting numeric values from text outputs of smaller, open-source LLMs is challenging due to inconsistencies in the response structure and the numerical representation. We develop a rich extraction function with regular expressions. Firstly, we remove date

Prompt type	System prompt
Completion prompt	You are a helpful assistant.
Question-answer prompt	You are a helpful assistant. If you do not know the answer to the question, provide your best estimate. Answer shortly like this. 'My answer: {number}
5-shot prompt	You are a helpful assistant. Answer the question. Use the format provided in the examples.
Chain-of-thought	You are a helpful assistant. If you do not know the answer to the question, provide your best estimate. Think carefully before giving an answer. Once you have the answer just state it like this. 'My final answer: {number}'.

Table AI: System prompts

strings (as these also contain numbers) and format numeric values by removing digit group separators before we convert answers into consistent units for the different variables. Additionally, with Q&A prompting, LLMs typically provide the answer first and then offer an explanation. In contrast, for other strategies, LLMs tend to explain first and return the answer at the end. Therefore, based on the prompting strategy, we apply an additional conditional statement to select either the first or the last numeric value as the final prediction.

C Additional results

Variable	Transformation	Spearman LME	Spearman Text	Pearson LME	Pearson Text
US firms					
return on assets	cubic	**0.69** (0.69-0.70)	0.45	**0.76** (0.75-0.76)	0.55
return on revenue	cubic	**0.60** (0.60-0.61)	0.48	**0.66** (0.66-0.67)	0.54
market capitalization	log	**0.91** (0.91-0.91)	0.83	**0.91** (0.91-0.91)	0.50
return on equity	cubic	**0.67** (0.66-0.67)	0.44	**0.68** (0.68-0.68)	0.50
total assets	log	**0.94** (0.93-0.94)	0.81	**0.92** (0.92-0.92)	0.48
price-earnings ratio	cubic	**0.50** (0.49-0.51)	0.37	**0.46** (0.45-0.47)	0.21
price-to-book ratio	cubic	0.28 (0.26-0.29)	**0.39**	0.14 (0.11-0.15)	**0.12**
debt-to-equity ratio	cubic	**0.37** (0.36-0.38)	0.32	**0.18** (0.15-0.19)	0.08
cost-to-revenue ratio	log	**0.60** (0.59-0.62)	-0.03	**0.48** (0.46-0.49)	0.02
US counties					
population	log	0.87 (0.86-0.87)	**0.90**	0.89 (0.88-0.89)	**0.91**
life expectancy at birth	no	**0.75** (0.74-0.76)	0.71	**0.74** (0.73-0.74)	0.69
unemployment rate	no	0.63 (0.60-0.67)	**0.65**	0.63 (0.60-0.67)	**0.64**
median monthly two-bedroom rent (in $)	log	**0.75** (0.74-0.75)	0.60	**0.85** (0.84-0.85)	0.63
personal income per capita (in $)	no	**0.73** (0.70-0.73)	0.61	**0.72** (0.71-0.73)	0.63
GDP per capita (in $)	log	**0.60** (0.59-0.62)	0.46	**0.56** (0.53-0.57)	0.42
proportion of mortgages being 90 or more days delinquent	no	**0.70** (0.66-0.72)	0.41	**0.69** (0.65-0.71)	0.41
percentage change in population compared to the previous year	no	**0.51** (0.50-0.52)	0.41	**0.48** (0.47-0.49)	0.38
number of business establishments per 100,000 people	log	**0.69** (0.68-0.70)	0.03	**0.68** (0.67-0.69)	0.04
UK districts					
average house price (in £)	no	**0.94** (0.91-0.96)	**0.94**	**0.89** (0.86-0.92)	**0.93**
population	log	0.85 (0.84-0.86)	**0.89**	0.85 (0.81-0.86)	**0.91**
median age	no	**0.89** (0.88-0.91)	0.85	**0.89** (0.88-0.91)	0.82
percentage of the population that live in income deprivation	no	**0.87** (0.85-0.88)	0.85	**0.86** (0.84-0.88)	0.81
median annual gross pay (in £)	no	**0.77** (0.76-0.80)	0.63	**0.83** (0.81-0.84)	0.72
life expectancy for females at birth	no	**0.80** (0.76-0.82)	0.70	**0.76** (0.70-0.81)	0.70
unemployment rate	no	**0.75** (0.70-0.78)	0.70	**0.73** (0.67-0.76)	0.68
GDP per capita (in £)	log	**0.63** (0.59-0.69)	0.48	**0.64** (0.58-0.69)	0.66
German districts					
number of hospital beds per 1,000 inhabitants	no	**0.53** (0.51-0.55)		**0.52** (0.49-0.55)	0.00[18]
population density (per square km)	log	**0.90** (0.88-0.90)	0.83	**0.91** (0.90-0.92)	0.82
unemployment rate (%)	no	**0.83** (0.82-0.84)	0.75	**0.81** (0.79-0.83)	0.74
disposable income per capita (in €)	no	**0.77** (0.76-0.78)	0.63	**0.76** (0.74-0.78)	0.60
average age of the population	no	**0.82** (0.81-0.84)	0.61	**0.84** (0.83-0.85)	0.57
GDP per capita (in €)	log	**0.69** (0.66-0.70)	0.42	**0.73** (0.70-0.74)	0.43
percentage change in population compared to the previous year	no	**0.57** (0.54-0.60)	0.48	**0.57** (0.52-0.61)	0.42
number of business registrations per 10,000 inhabitants	no	**0.63** (0.57-0.66)	0.29	**0.66** (0.59-0.69)	0.36

Continued on next page

[18] With constant responses across districts, the sample correlation is not defined. We record performance as 0.

Variable	Transformation	Spearman LME	Spearman Text	Pearson LME	Pearson Text
number of filed corporate insolvencies per 10,000 taxable companies	no	**0.49** (0.45-0.52)	0.32	**0.59** (0.56-0.62)	0.34
percentage of school leavers with a general university entrance qualification (%)	no	**0.65** (0.63-0.67)	0.07	**0.63** (0.60-0.65)	0.13
number of government employees per 1,000 inhabitants	no	**0.63** (0.62-0.65)	0.11	**0.63** (0.61-0.65)	0.08
number of completed housing units per 1,000 inhabitants	no	**0.47** (0.41-0.52)	-0.10	**0.42** (0.35-0.46)	0.04
total investments per employee (in €)	no	**0.12** (0.04-0.18)	-0.05	**0.07** (-0.01-0.13)	-0.09
EU (NUTS2)					
life expectancy at birth	no	0.70 (0.54-0.79)	**0.85**	0.86 (0.82-0.89)	**0.91**
unemployment rate	no	0.75 (0.52-0.82)	**0.88**	0.78 (0.48-0.85)	**0.87**
net disposable household income (in €)	no	**0.92** (0.89-0.94)	0.81	**0.94** (0.91-0.95)	0.82
population	log	**0.73** (0.69-0.77)	0.72	**0.75** (0.71-0.81)	0.68
risk-of-poverty rate	no	**0.55** (0.35-0.61)	0.51	**0.47** (0.34-0.59)	0.50
GDP per capita (in €)	log	**0.75** (0.65-0.79)	0.50	**0.76** (0.67-0.80)	0.46
number of burglaries per hundred thousand inhabitants	no	**0.55** (0.27-0.62)	-0.43	**0.47** (0.28-0.56)	-0.01

Table AIII: Cross-validation performance of the baseline LLM comparing LME with text output. For each performance metric, the best-performing approach is shown in bold. LME performance is based on 25 iterations of 5-fold cross-validation. Numbers before the parentheses indicate the mean; numbers in parentheses show the minimum and maximum performance across the 25 iterations.

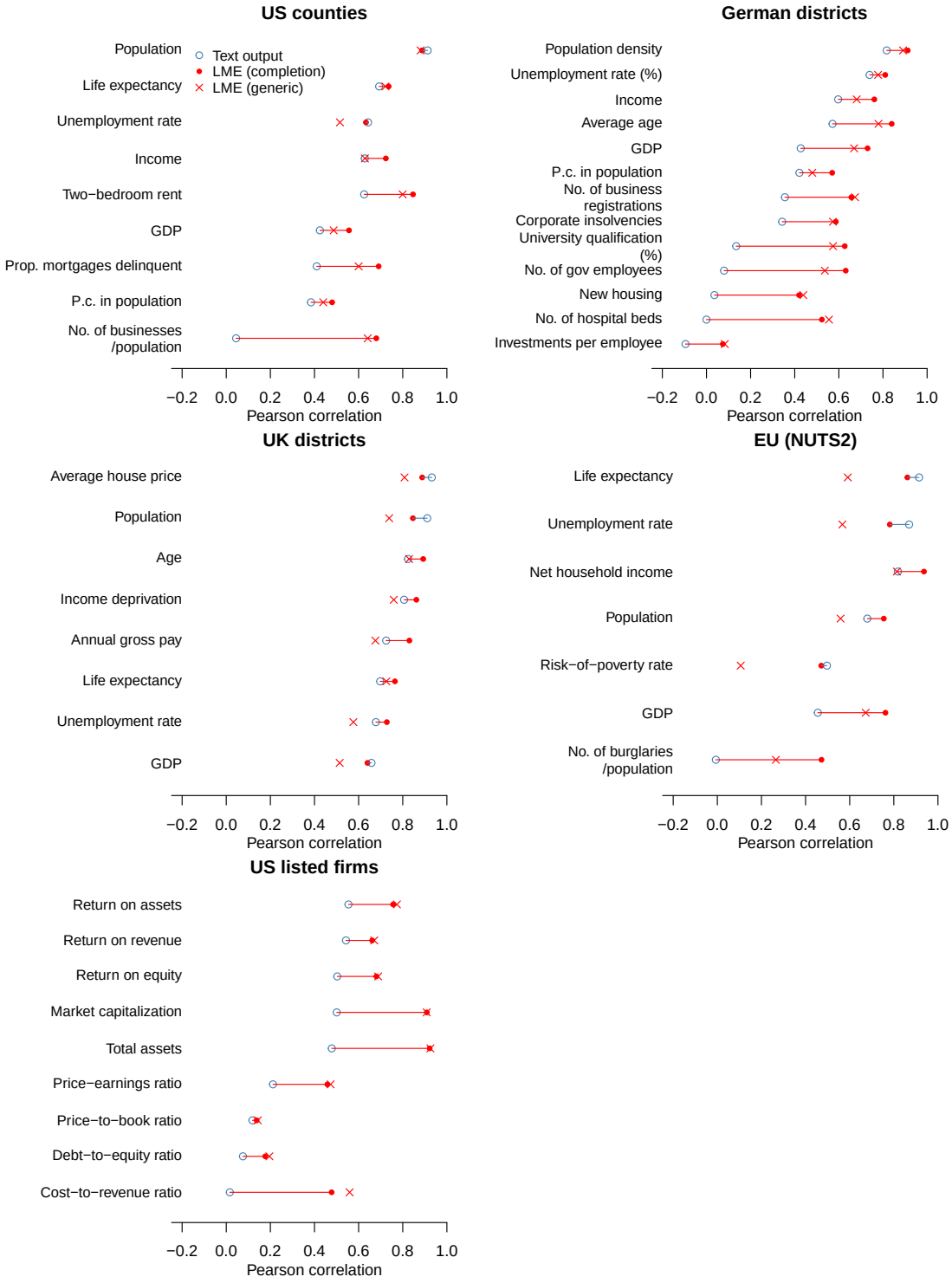

Figure AII: Cross-validation performance using Pearson correlation as the performance metric.

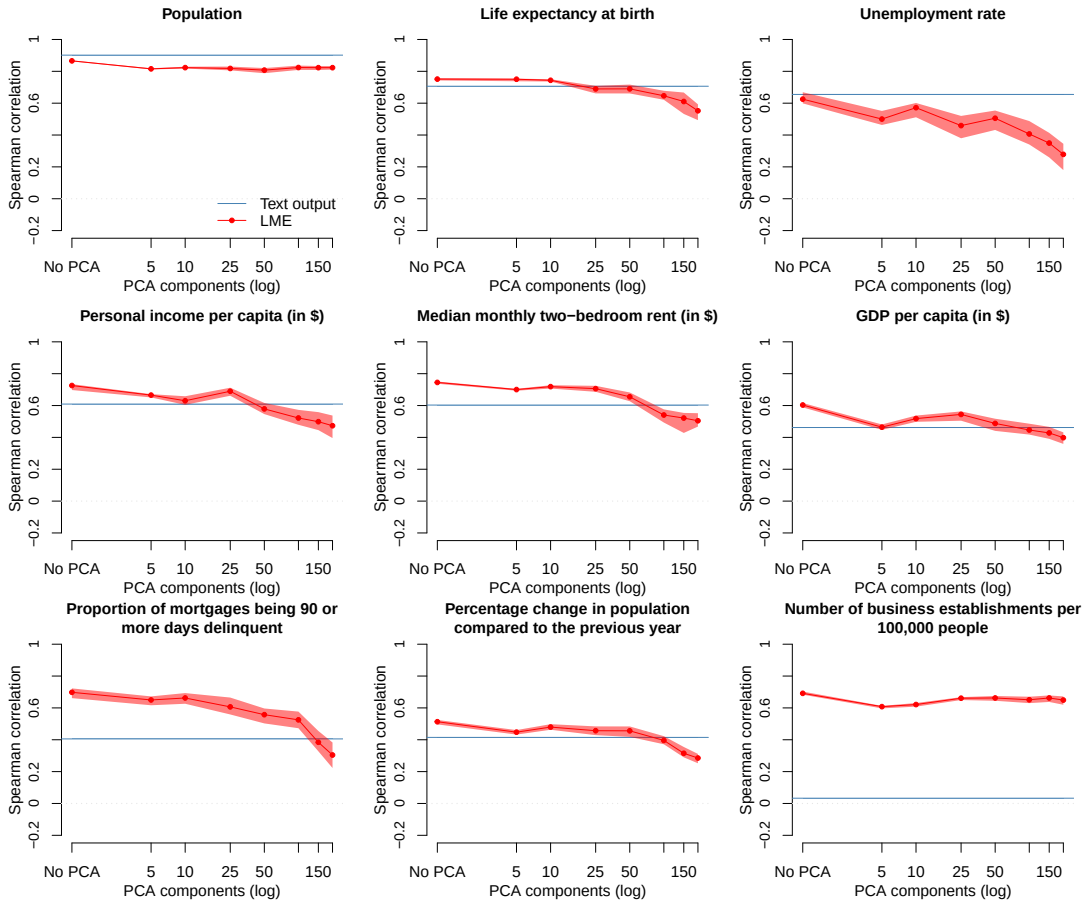

Figure AIII: Performance of the base model on the US counties dataset as a function of the number of PCA components. LME's mean performance across 25 cross-validation iterations is shown as a line; the shaded band indicates the minimum–maximum range across iterations.

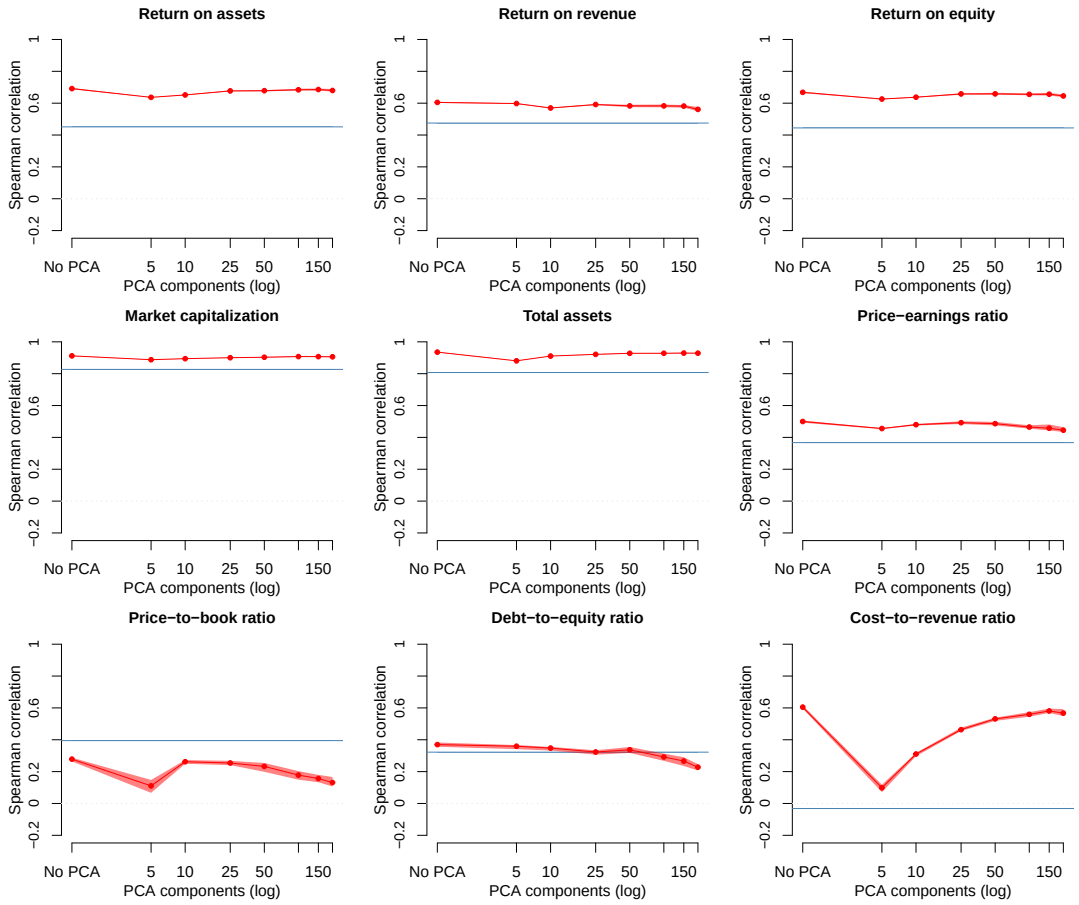

Figure AIV: Performance of the base model on the US-firms dataset as a function of the number of PCA components. LME's mean performance across 25 cross-validation iterations is shown as a line; the shaded band indicates the minimum–maximum range across iterations.

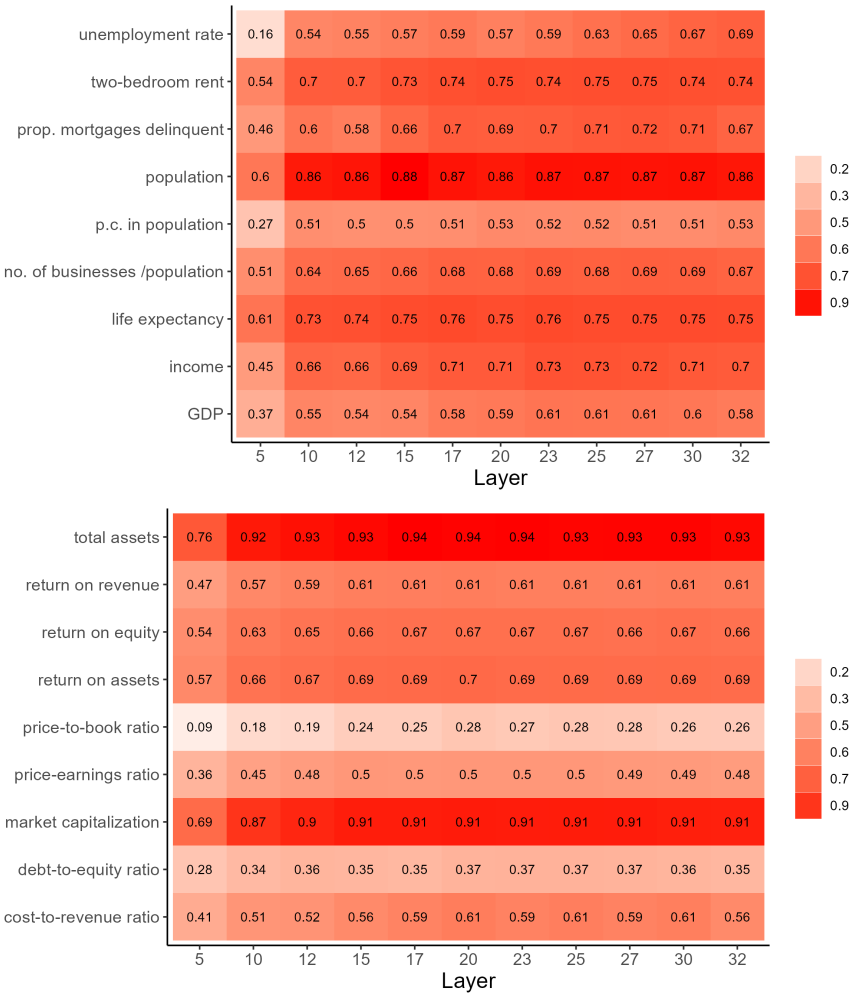

Figure AV: Comparing LME performance when trained on different layers.

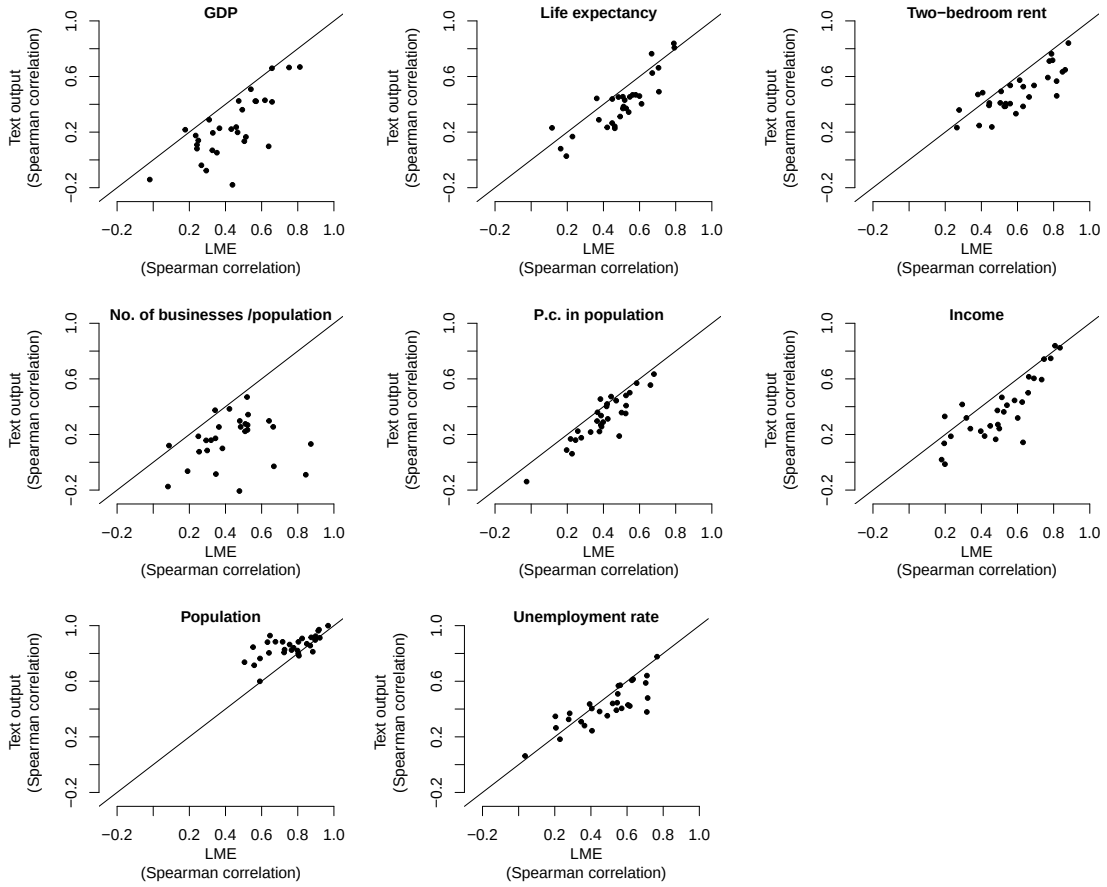

Figure AVI: Comparison of performance between LME and text output within states with at least 50 counties.

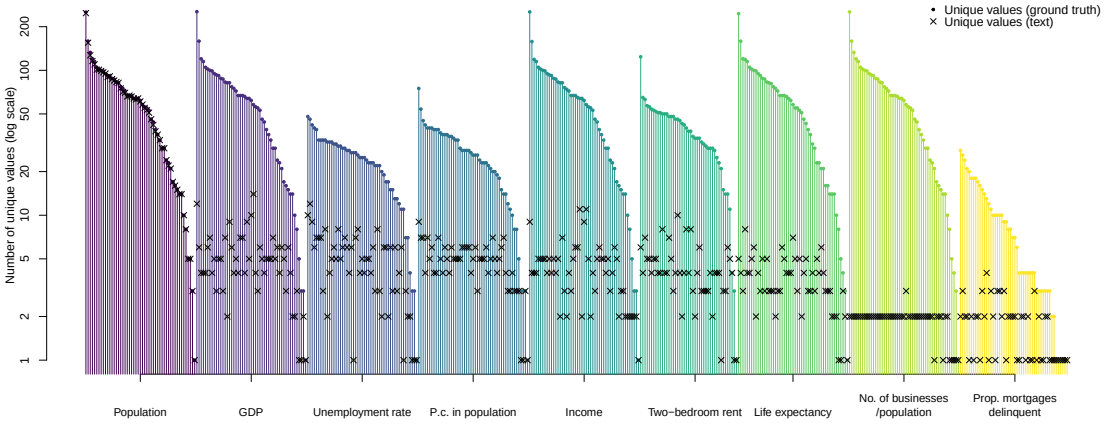

Figure AVII: Number of unique ground truth and text output values for each state. Each vertical lines represents one state; states are ordered by the decreasing number of unique values observed.