Journal of Data Science xx (xx), 1–20 DOI: xx.xxxx/xxxx_xxxx
July 2023

# My wonderful paper

**Abstract**

Unexpected results in data analysis can prompt researchers to examine data quality and analysis steps. While some researchers can typically diagnose issues effectively with a few checks, others may struggle to identify appropriate checks to diagnose the problem. [an example of check] These checks are often informal and difficult to trace and discuss in publications, resulting in others questioning the trustworthiness of the analysis. To address this, we formalize the informal checks into an *analysis plan* that encompasses the analysis steps and (the unit tests): one test for whether the result meets expectations and multiple tests for checking the analysis. We then present a procedure to assess the quality of these unit tests based on their accuracy and redundancy on simulated versions of the original data. The accuracy is assessed using binary classification metrics, *i.e.*, precision and recall, while redundancy is calculated using mutual information. This procedure can be used to conduct a sensitivity analysis, compare different analysis plans, and to identify the optimal cutoff point for the unit tests.

## Table of contents

## 1 Introduction

- TODO: emphasize trustworthy data science since it is the theme of the special issue

In data analysis, experienced researchers often rely on their prior knowledge or domain expertise to quickly assess whether results align with their expectations. When outcomes deviate, they can quickly iterate through the analysis cycle to identify potential issues, refine methods, or revise assumptions. However, this iterative process they employ is rarely explicit, documented, or discussed in publications. As a result, readers are typically presented with the final outcomes of the analysis cycle where the results and expectations are aligned – achieved either by refining the analysis or updating the expectations based on statistical evidence. These missing pieces of information provides little guidance for diagnosing issues in the analysis when the same methodology is applied to a new dataset that produces different outcomes. Similarly, when researchers with different background knowledge it becomes unclear whether variations in results arise from differences in the assumptions or in the analysis itself.

In this paper, we frame these expectations as checks within an *analysis plan*, which allows us to examine these implicit assumptions made about the data during analysis and to compare different checks for diagnosing unexpected outcomes. We then introduce a procedure to evaluate the quality of these checks, either individually or in combination through logic regression, based on their accuracy and redundancy across simulation of the original data. Accuracy is assessed using binary classification metrics, precision and recall, while redundancy is measured using mutual information. The proposed workflow offers a numerical guarantee, based on simulation, that the analysis will produce the expected results, assuming the data generating mechanism holds. [something about trustworthy data science]

The rest of the paper is organized as follows: Section 2 reviews the concept of diagnosing unexpected outcomes and data quality checks. Section 3 introduces the concept of analysis plan, illustrated with a toy example on fitness step count. Section 4 describes the procedure for selecting optimal check combinations to predict unexpected outcomes. Section 5.1 applies this selection procedure to a linear regression example examining the effect of PM10 on mortality. Section 6 discusses a few key considerations and Section 7 concludes the paper.

## 2 Literature review

### 2.1 Diagnosing unexpected outcomes in data analysis

(Peng et al., 2021) describes three pedagogical exercises of introducing diagnosing unexpected outcome into a classroom setting.

TODO: what if the expectation is “wrong”

### 2.2 Data analysis checks

A substantial body of literature has addressed the definition of data quality (Cichy and Rass, 2019, more) and developed frameworks, that includes dimensions, attributes, and measures to evaluate and improve data quality (Cai and Zhu, 2015; Wang and Strong, 1996; Sidi et al., 2012; Woodall et al., 2014). These frameworks are often used information system and database management and support business decision-making in the industry. For research purposes, high-quality data ensures the credibility of scientific findings and supports reproducibility and reusability in future studies [ref]. With the growing prevalence of open data in scientific research, the consumers or users of the data typically are no longer the data producers or collectors who have the full knowledge of data in hand, prompting more interest towards the data quality checks in the data analysis process. In R, there are some packages, like`skimr` (Waring et al., 2022) and `dataMaid` (Petersen and Ekstrøm, 2019), provides basic data screening and reporting, while another class of packages, e.g. `assertr` (Fischetti, 2023), `validate` (van der Loo and de Jonge, 2021), and `pointblank` (Iannone et al., 2024), focuses on providing data validation tools, allowing users to define customized data quality checks based on the applications.

## 3 Analysis validation checks

### 3.1 Framing expectations as checks

Q: Whether we should formulate these concept with math notation?? A: only if it helps

While data validation focuses on verifying or subsetting data based on predefined rules (Di Zio et al., 2016), data analysis checks specify assumptions made on the data required for the analysis process to obtain the reported results. These checks often involve evaluating aspects such as data distribution and outliers, relationships between variables, and the validity of model assumptions.

However, there are other checks that reflect the mental process of of how data analysts diagnose the unexpected output that are not documented in this process. For example, when an unexpected output occurs, an analyst may check on whether a column in the data frame is between two values for data quality. However, it is not documented what motivates the analyst to conduct this check – whether it also applies to other researchers analyzing new data in the same contexts, whether it is a common practice in the field, or whether it is a reaction to this particular data or scenario. Currently, most of these assumptions are not discussed in the publication or captured by tools themselves. While one might be able to infer some of the mental process of the analysts from external sources, such as talking to them or watching screencast videos produced by the analysts e.g. TidyTuesday screencast videos, these insights are not systematically documented or made machine-readable. A question to consider here is whether it is possible to deliver this type of quality checks as structured data in a data analysis. This gap highlights the need for tools that provide higher level documentation of reasoning behind the checks, facilitating a more transparent and interpretable analysis process.

An analysis plan is a set of analysis steps combined with expectations. Expectations represent our belief about certain aspects of the analysis, independent of the analysis itself. It can be divided into two types: *outcome expectation* and *plan expectation*. Outcome expectation refers to what we anticipate from the main result of the analysis based on prior knowledge. They shape how we interpret the results and assess whether they are consistent with existing knowledge or indicate the need for updates (Grolemund and Wickham, 2014). For example, in public health, prior research shows the average increase in mortality rate per unit increase in PM10 is about 0.50% (Liu et al., 2019). This serves as an expectation for similar future studies. Plan expectations concern the intermediate steps within the analysis rather than the final outcome. They serve as checkpoints to detect deflection in the analysis process For example, we may expect temporal data to be ordered with no gaps and duplicates, or expect that temperature will be a significant covariate in the linear regression model of PM10 on mortality.

Experienced analysts often have implicit expectation about the outcome and rely on a few "directional signs" to check when the outcome deviate from those expectation. However, these expectations are rarely made explicit within the analysis workflow. This makes it challenging for consumers of the analysis to evaluate the results, since it becomes difficult to disentangle whether discrepancies arise from differing expectations or from the use of statistical technique, without running the analysis themselves. Non-expert analysts, lacking prior knowledge or instinct, may not have clear expectations of the results. This can lead to reduced confidence of the analysis and makes it more difficult and time-consuming to diagnose the cause of the deviation when the results don't align with expectations. By explicitly formulating these expectations, an analysis plan can guide the analysis process, facilitate the communication and evaluate the validity of the results.

Yet, making them explicit is crucial for

1) helping junior researchers interpret results and diagnose the unexpected,
2) providing checkpoints that support independent replication or application of methods to new data, and
3) aligning assumption across researchers from different backgrounds.

## 3.2 A toy example

Consider a 30-day step count goal. Suppose you resolve to walk at least 8,000 steps each day, using an app to record your daily step count. Your target average is 9,000 steps per day, with some "lows" day, where you walk around 4,000 steps and "high" days where you reach about 12,000 steps. After 30 days, you check how many times your step count fall below 8,000 and aim for no more than five days under this threshold.

To simulation this data, three normal distributions with different means are used for the daily step counts: $\mathcal{N}(4000, 200)$ for low days, $\mathcal{N}(12000, 200)$ for high days, and $\mathcal{N}(9000, 300)$ for typical days. The number of low and high days can be simulated from a Poisson distribution with $\lambda = 4$. Figure 1 displays the number of days with fewer than 8,000 steps across 300 simulated 30-day periods.

In this scenario, the outcome expectation is that the number of days with a step count below 8,000 will be no more than five. To diagnose potential reasons why this outcome expectation might fail, we can establish a few plan expectations. For example, if the average step count is too low, this may suggest there are too many low days, potentially lead to an unexpected outcome. Similarly, we can also check the quantile of the step count, if more than a third of the days fall below 8,000, this could indicate an excess of low-count days. Additionally, we may may expect the standard deviation of the step count not to be overly large.

TODO: add one or two scenarios on why step count could go up or down: 1) you take off your watch in the middle of the day, 2) have an workout

TODO: phrase it as Researcher measuring someone's step count rather than measuring your own step count

These considerations yield the following three unit tests as plan expectations:

- test1: the test fails if the mean step count is below 8,200
- test2: the test fails if the 30th percentile of the step counts is below 8,200
- test3: the test fails if the standard deviation of the step countsexceeds 2,500.

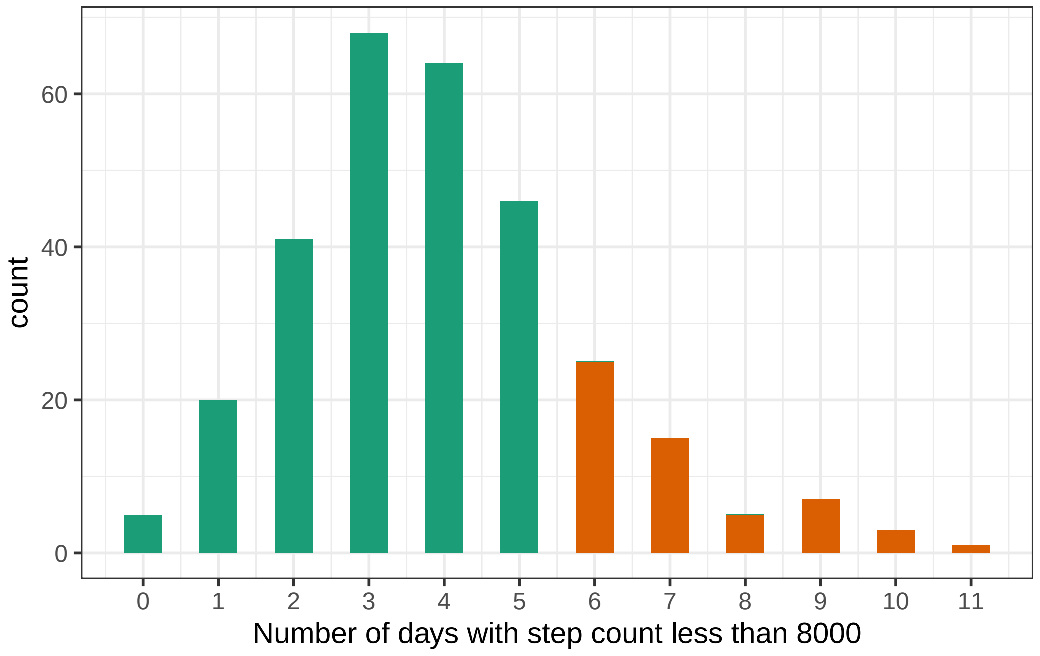

Figure 1: Number of days with fewer than 8,000 steps across 300 simulated 30-day periods. The orange bars indicate instances where the count exeeds five days, representing an unexpected outcome in this scenario.

# 4 Method

## 4.1 A workflow to assess the quality of unit tests

In real-world applications, it is rare to create a set of unit tests can fully guarantee expected results. On one hand, there is the cost of effort involved in manually developing all these tests;

on the other, there is the inherent complexity of the problem. (can we - if we're thriving for detecting 95% of the cause?). However, the quality of unit tests can be evaluated by simulated data. One set of tests is considered better than another if a small set of tests can reliably detect unexpected outcomes, which brings two criteria in the evaluation metric: accuracy and parsimony. Figure 2 illustrates the workflow for calculating the metrics.

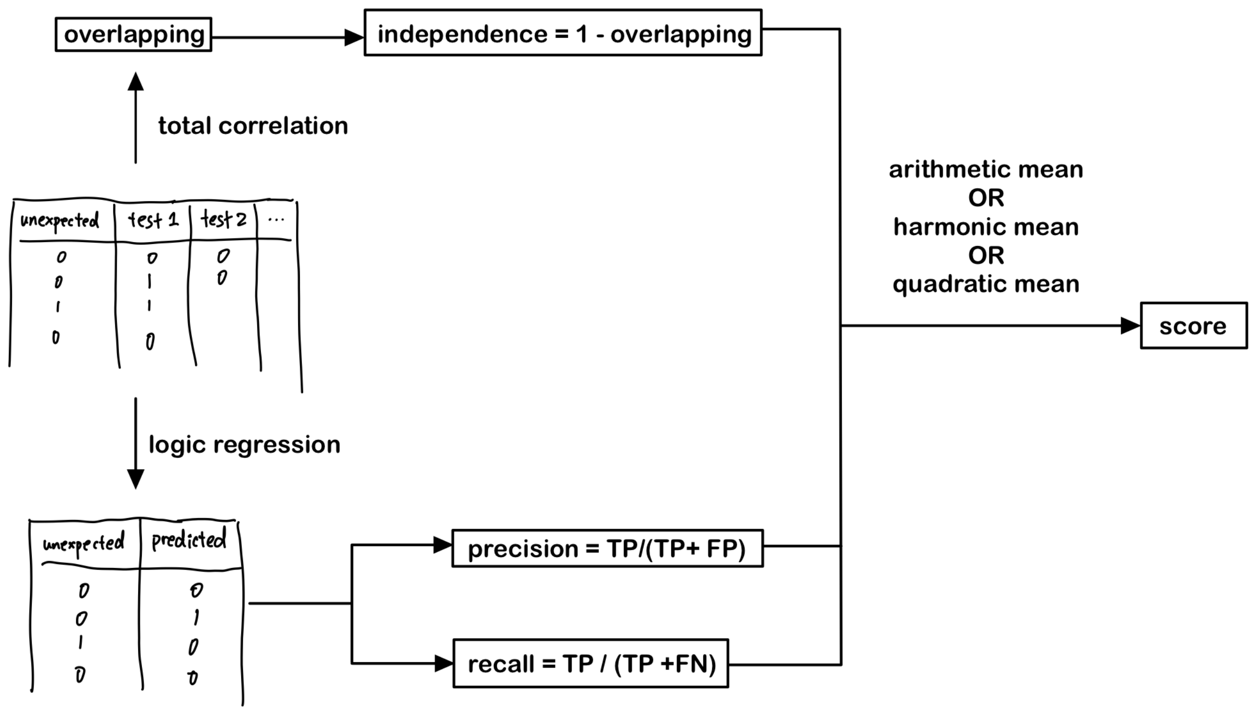

Figure 2: this is the cap

Accuracy refers to a set of tests' ability to accurately detect unexpected outcomes while minimizing false positives and false negatives. A false positive can indicate (caution or skepticism on checking the data), whereas a false negative suggests the tests may lack sensitivity to unexpected outcomes. To model the relationship between the plan and outcome expectation (binary-binary), a logic regression model is used (Ruczinski et al., 2003). Originally developed for SNP microarray data, logic regression constructs Boolean combinations of binary variables to solve regression problems [more introduction on logic regression]. Compared to other tree-based methods or machine learning methods for binary-binary prediction, logic regression produces Boolean combinations, or meta rules, that combines unit tests to solve the prediction problem. [need rewrite here]. The performance of the tests can be evaluated using precision and recall metrics derived from the confusion matrix of the logic regression prediction

- precision: the proportion of correctly identified unexpected results (true positives) out of all the predicted unexpected results (true positives + false positives)
- recall: the proportion of correctly identified unexpected results (true positives) out of all the actual unexpected results (true positives + false negatives)

The second criteria is parsimony in the tests. While tests may score high on accuracy, they may be less effective at explaining the reasons behind unexpected results. This could happen if a set of tests are all tangentially related to the cause of the unexpected results, but none addressing the root cause. It may also occur if the tests are correlated with one another, leading to redundancy.

To quantify redundancy, the concept of mutual information is used. Mutual information $I(x, y)$ measures the amount of information shared between two random variables and is defined as the KL-distance $D(p \parallel q)$ between the joint distribution of the two variables and the product of the marginal distributions:

$$I(x, y) = D(p(x, y) \parallel p(x)p(y)) = \sum_{x} \sum_{y} p(x, y) \log \frac{p(x, y)}{p(x)p(y)}$$

This concept extends naturally to multiple variables through total correlation [ref], $C(X_1, X_2, \cdots, X_n)$, which captures redundancy across a set of $n$ variables:

$$C(X_1, X_2, \cdots, X_n) = \sum_{x_1} \sum_{x_2} \cdots \sum_{x_n} p(x_1, x_2, \cdots, x_n) \log \frac{p(x_1, x_2, \cdots, x_n)}{p(x_1)p(x_2) \cdots p(x_n)}$$

A high mutual information value indicates redundancy among the tests, while a low value suggests that the tests are independent and provide unique information to diagnose the unexpected outcome. To standardize this measure, the total correlation *per observation* is calculated, and an independence score, ranging between 0 and 1, is defined as 1 - mutual information.

To combine precision, recall, and independence into a single metric, various mathematical means, such as arithmetic mean, harmonic mean, and quadratic mean, can be used. The differences among these means are minimal when the three metrics are similar. However, as the differences among the metrics increases, the harmonic mean tends to produce the smallest overall score, as it penalizes low values, while the quadratic mean tends to produce the largest score by rewarding higher values more. For simple interpretation of the score, the arithmetic mean is preferred, while in applications where the difference between precision, recall, and independence need to be penalized or rewarded more, the harmonic and quadratic mean should be considered.

## 4.2 Toy example revisited

In the step count example, we can use the logic regression model to evaluate the quality of the unit tests. The logic regression model is fitted to the three unit tests (test1, test2, test3) and the outcome expectation (unexpect) as the response variable. The model is then used to predict the outcome expectation based on the unit tests. The prediction is then compared to the actual outcome expectation to calculate the precision and recall of the tests. The independence of the tests is also calculated to assess the redundancy of the tests. Figure 3 presents the suggested logic regression model as a combination of test 1 and test 3 with an OR operator.

Table 1 presents the calculated precision, recall, and independence for the three individual tests and the combined test rule (test1 OR test3) from the logic regression. The harmonic and arithmetic means are included to evaluate the quality of the unit tests in diagnosing unexpected step counts. The results show that the combined test rule (test1 OR test3) has the highest precision, recall, and independence, suggesting that it is the most effective test for diagnosing unexpected step counts. We also include the metric calculated from fitting a regression tree model to the data to compare the performance of the logic regression model. The regression tree produces a similar model of first split on test1 and then split on test3, and results in the same accuracy and overall score as the logic regression model.

top-down and bottom up (regression tree + logic tree): more naturally useful summary of the say it is organized. (put down the plot)

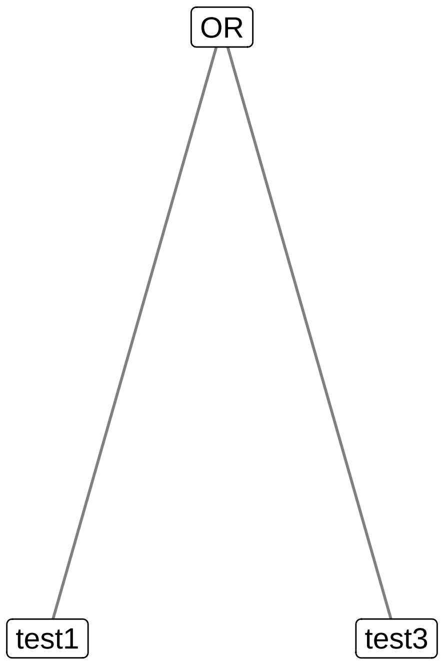

Figure 3: Logic regression model fitted to the three unit tests (test1, test2, test3) and the outcome expectation (unexpect) as the response variable. The model suggests using an OR rule to combine test1 and test3 to predict the outcome expectation.

Table 1: Accuracy (precision and recall) and parsimony (independence) metrics for each individual unit test and for the combined test rule (test1 OR test3) derived from the logic regression model. The harmonic and arithmetic means of the three metrics are included to evaluate the quality of the unit tests in diagnosing unexpected step counts (more than five days with fewer than 8,000 steps).

| tests | precision | recall | independence | harmonic | arithmetic |
|---|---|---|---|---|---|
| test1 | 0.482 | 0.964 | 1.000 | 0.730 | 0.815 |
| test2 | 0.214 | 1.000 | 1.000 | 0.450 | 0.738 |
| test3 | 0.589 | 0.805 | 1.000 | 0.762 | 0.798 |
| test1 OR test3 | 0.821 | 0.836 | 0.999 | 0.879 | 0.886 |
| regression tree | 0.821 | 0.836 | 1.000 | 0.879 | 0.886 |

## 5 Applications

A regression example is produced to illustrate the test selection process for analyzing the effect of PM10 on mortality. The example demonstrates how the test process can be used to select cutoff values in the unit tests and how the test can be used to diagnose an unexpected PM10 coefficient from the generalized linear model.

### 5.1 Effect of PM10 on mortality

Consider a generalized linear model (GLM) to study the effect of PM10 on mortality [TODO: provide more context of the mortality-PM10 study]. Analysts may expect a PM10 coefficient between [0, 0.005] after considering the temperature confounding [TODO: reference]. This expectation can be framed into a check that fails, labelled as 1, if the PM10 coefficient is outside the range [0, 0.005]. Multiple factors can impact the PM10 coefficient, such as the sample size, the strength of the correlation between mortality and PM10, and the strength of the correlation between mortality and temperature. Analysts may expect a reasonable sample size to ensure the reliability of the coefficient estimate. Outliers in the three variables can also leverage the coefficient. While these are possible factors that could affect the analysis result, it is not clear the cutoff values for these checks to determine a failure. Here a list of checks are created in Table 2.

Table 2: A list of checks considered for the generalized linear model of mortality on PM10 and temperature. The checks are based on the sample size, correlation between mortality and PM10, correlation between mortality and temperature, and univariate outlier detection. Multiple cutoff values are specified for each check to determine a failure.

| the check fails if ... |
|---|
| Sample size less than or equal to 200 |
| Sample size less than or equal to 400 |
| Sample size less than or equal to 600 |
| Sample size less than or equal to 800 |
| Mortality-PM10 correlation less than -0.03 |
| Mortality-PM10 correlation less than -0.04 |
| Mortality-PM10 correlation less than -0.05 |
| Mortality-PM10 correlation less than -0.06 |
| Mortality-temperature correlation greater than -0.3 |
| Mortality-temperature correlation greater than -0.35 |
| Mortality-temperature correlation greater than -0.4 |
| Mortality-temperature correlation greater than -0.45 |
| Outlier(s) are presented in the variable PM10 |
| Outlier(s) are presented in the variable mortality |

To generate replicate of the data, we first simulate the correlation matrix of the three variables in a grid and then use a Gaussian copula to generate a multivariate normal distribution based on the specified correlation matrix and sample size. The multivariate normal distribution is transformed using the normal CDF before the inverse CDF of the assumed distributions of the three variables is applied. To determine the appropriate distribution of each variable, various distributions are fitted and compared. This includes poisson and negative binomial for mortality; gamma, log-normal, exponential, weibull, and normal for pm10 and temperature; and beta for pm10 after rescaling the data to $[0-1]$. AIC is used to determine the best distribution fit for each variable with qq plot presented in Figure 4 to evaluate the fit. AIC suggests a negative binomial distributio nfor mortality, a beta distribution for PM10 (multiple by 100 to obtain the original scale), and a Weibull distribution for temperature. To include the effect of outlier, we add a single outlier to the data for mortality and PM10 [more details].

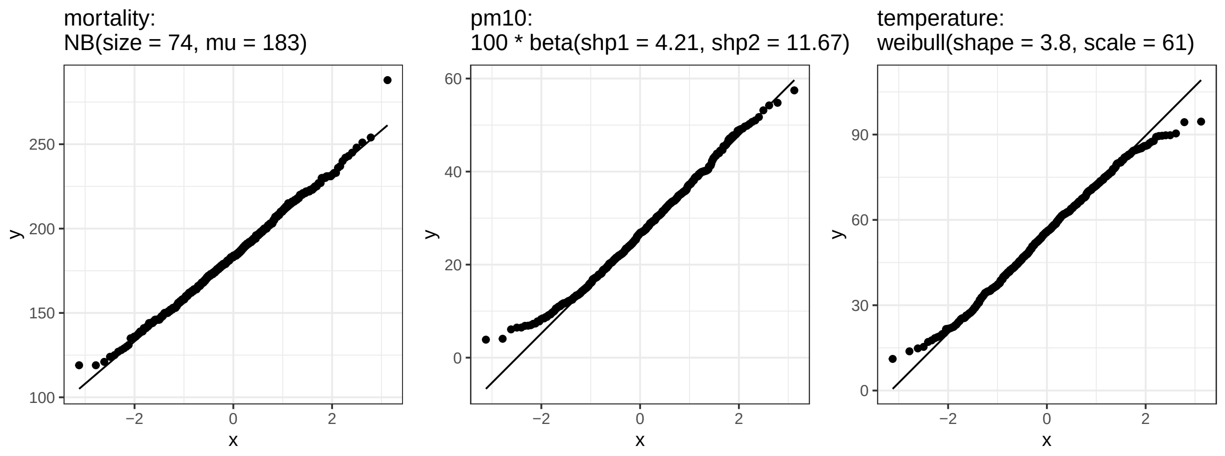

Figure 4: QQ-plot of the distribution fit for mortality, PM10, and temperature based on the fitted distribution from the original data. The fitted distribution is compared to the observed data to assess the fit.

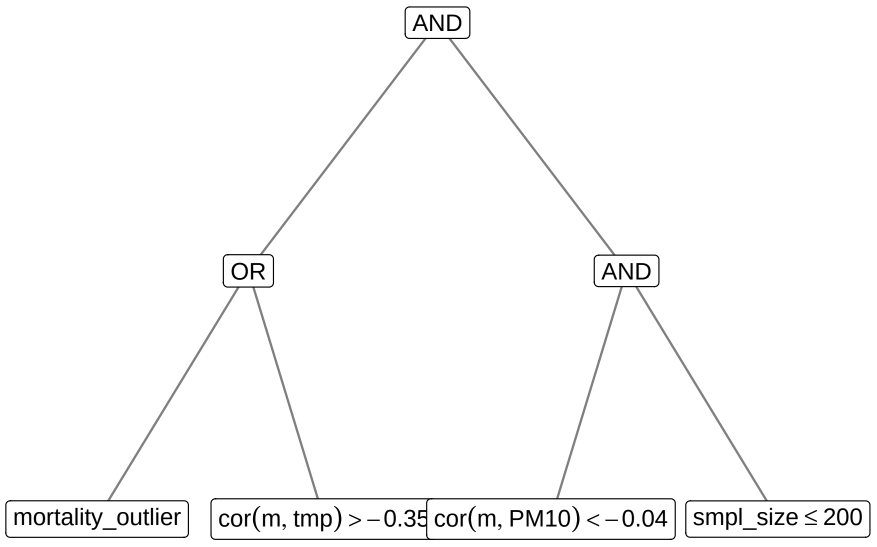

Figure 5: Logic regression model fitted to the twelve unit tests and the outcome expectation (unexpect) as the response variable. The model suggests the relationship: (cor(m, tmp) $>$ $-0.35$ AND smpl_size $\leq 600$) OR cor(m, PM10) $< -0.05$

Table 3: Accuracy (precision and recall) and parsimony (independence) metrics derived from the logic regression model, along with harmonic and arithmetic means, for individual unit tests (1: sample size, 2: mortality-PM10 correlation, 3: mortality-temperature correlation), and the combined test rule 4: (sample size AND mortality-temperature correlation) OR mortality-PM10 correlation.

| tests | precision | recall | overlapping | independence | harmonic | arithmetic |
|---|---|---|---|---|---|---|
| 1 | 0.087 | 0.215 | 0 | 1 | 0.175 | 0.434 |
| 2 | 0.988 | 0.610 | 0 | 1 | 0.822 | 0.866 |
| 3 | 0.392 | 0.581 | 0 | 1 | 0.569 | 0.658 |
| 4 | 0.649 | 0.641 | 0 | 1 | 0.732 | 0.763 |
| 5 | 0.760 | 0.880 | 0 | 1 | 0.869 | 0.880 |

A logic regression is fitted to predict whether the PM10 coefficient is unexpected (outside the range of [0, 0.005]) using the checks listed in Table 2. Precision, recall, and independence score, along with their harmonic and arithmetic mean are calculated. Figure 5 shows the logic regression tree from the fitted model and Table 3 prints the numerical summary of four selected single test and their combined test found by the logic regression. The logic regression model picks up the following cutoff value for each type of check:

- sample size *larger than* 200
- mortality-temperature correlation greater than -0.35
- mortality-PM10 correlation less than -0.04
- mortality contain outliers that are detected by the univariate outlier detection

The tree structure suggests checking mortality-PM10 correlation and a sample size larger than 200 with an additional check of either outlier on mortality or correlation between mortality and temperature. This combined check rule generates a 0.76 precision and a 0.88 recall for predicting the unexpected PM10 coefficient. The single check, `cor(m, PM10) < -0.03`, is also powerful with a high precision of 0.988, but the low recall value of 0.61 suggests its high false positive rate, as compared to the combined rule suggested by the logic regression.

## 6 Discussion

- how to systematically simulate data is still unknown, sensitivity of the simulation to the results
- plotting is a critical way to check data and they can still be frame into a unit test. it is a open problem to how to encode the visualization into the unit tests. Maybe a procedure like confirm plot (this looks alright to you) and then press the button to continue
- currently no automated way to generate unit tests. It is interesting to see the automation of generating unit tests, although it requires the inputs from experts across a wide array of common scenarios.
- There are cost and benefit on setting expectation on different granularity. At the lowest level, one may have a plan for each data entry and every data handling steps. This

requires more work from the analysts and may not be practical in practice. For more complex analyses, analysts may divide the analysis into sections and set expectations for each. They can then focus on the specific sections flagged by the tests and sub-divide the sections to set expectation and diagnose the analysis in a hierarchical manner.

- (only mention it) Software testing relies on "oracles" to provide the expected output necessary for verifying test results. For example, to test whether the program correctly calculates 1 + 1, one need to supply the correct answer, 2. However, establishing this "correct" output can sometimes be challenging, where obtaining a solution may be difficult without the program itself. This situation leads to the oracle problem (Barr et al., 2014). In data analysis, the similar oracle problem can happen, as the "truth" of an outcome, the expectation, depends on the underlying theory or interpretation. For example, in a linear regression model, the significance of a coefficient may be expected or unexpected based on the theory, making it challenging for researcher with a different theory to assess the results and the analysis.

## 7 Conclusion

## References

Earl T Barr, Mark Harman, Phil McMinn, Muzammil Shahbaz, and Shin Yoo. The oracle problem in software testing: A survey. *IEEE transactions on software engineering*, 41(5): 507–525, 2014.

Li Cai and Yangyong Zhu. The challenges of data quality and data quality assessment in the big data era. *Data science journal*, 14:2–2, 2015.

Corinna Cichy and Stefan Rass. An overview of data quality frameworks. *IEEE Access*, 7: 24634–24648, 2019. doi: 10.1109/ACCESS.2019.2899751.

M. Di Zio, N. Fursova, T. Gelsema, S. Giessing, U. Guarnera, J. Petrauskiene, L.Q.

von Kalben, M. Scanu, K. ten Bosch, M. Van der Loo, and K. Walsdorfer. Methodology for data validation 1.0. Technical report, ESSNet on Validation, 2016. URL https://ec.europa.eu/eurostat/documents/7755309/7769541/Methodology_for_data_validation_V1.0_REV-2016-06_FINAL.pdf/ed7abb5d-6f1f-4e92-9a72-40d8a7fe0b85.

Tony Fischetti. *assertr: Assertive Programming for R Analysis Pipelines*, 2023. URL https://CRAN.R-project.org/package=assertr. R package version 3.0.1.

Garrett Grolemund and Hadley Wickham. A Cognitive Interpretation of Data Analysis. *International Statistical Review*, 82(2):184–204, 08 2014. doi: 10.1111/insr.12028. URL https://onlinelibrary.wiley.com/doi/10.1111/insr.12028.

Richard Iannone, Mauricio Vargas, and June Choe. *pointblank: Data Validation and Organization of Metadata for Local and Remote Tables*, 2024. URL https://CRAN.R-project.org/package=pointblank. R package version 0.12.2.

Cong Liu, Renjie Chen, Francesco Sera, Ana M Vicedo-Cabrera, Yuming Guo, Shilu Tong, Micheline SZS Coelho, Paulo HN Saldiva, Eric Lavigne, Patricia Matus, et al. Ambient particulate air pollution and daily mortality in 652 cities. *New England Journal of Medicine*, 381(8):705–715, 2019.

Roger D. Peng, Athena Chen, Eric Bridgeford, Jeffrey T. Leek, and Stephanie C. Hicks. Diagnosing Data Analytic Problems in the Classroom. *Journal of Statistics and Data Science Education*, 29(3):267–276, September 2021. doi: 10.1080/26939169.2021.1971586. URL https://doi.org/10.1080/26939169.2021.1971586.

Anne Helby Petersen and Claus Thorn Ekstrøm. datamaid: Your assistant for documenting supervised data quality screening in r. *Journal of Statistical Software*, 90:1–38, 2019.

Ingo Ruczinski, Charles Kooperberg, and Michael LeBlanc. Logic Regression. *Journal of Computational and Graphical Statistics*, 12(3), September 2003. ISSN 1061-8600. doi: 10.1198/1061860032238. URL https://doi.org/10.1198/1061860032238. Publisher: ASA Website pages = 475–511,.

Fatimah Sidi, Payam Hassany Shariat Panahy, Lilly Suriani Affendey, Marzanah A. Jabar, Hamidah Ibrahim, and Aida Mustapha. Data quality: A survey of data quality dimensions. In *2012 International Conference on Information Retrieval & Knowledge Management*, pages 300–304, 2012. doi: 10.1109/InfRKM.2012.6204995.

Mark PJ van der Loo and Edwin de Jonge. Data validation infrastructure for r. *Journal of Statistical Software*, 97:1–33, 2021. doi: 10.18637/jss.v097.i10. URL https://www.jstatsoft.org/article/view/v097i10.

Richard Y Wang and Diane M Strong. Beyond accuracy: What data quality means to data consumers. *Journal of management information systems*, 12(4):5–33, 1996.

Elin Waring, Michael Quinn, Amelia McNamara, Eduardo Arino de la Rubia, Hao Zhu, and Shannon Ellis. *skimr: Compact and Flexible Summaries of Data*, 2022. URL https://CRAN.R-project.org/package=skimr. R package version 2.1.5.

Philip Woodall, Martin Oberhofer, and Alexander Borek. A classification of data quality assessment and improvement methods. *International Journal of Information Quality 16*, 3(4):298–321, 2014.