Guided Statistical Workflows with Interactive Explanations and Assumption Checking

Category: Research

Paper Type: system

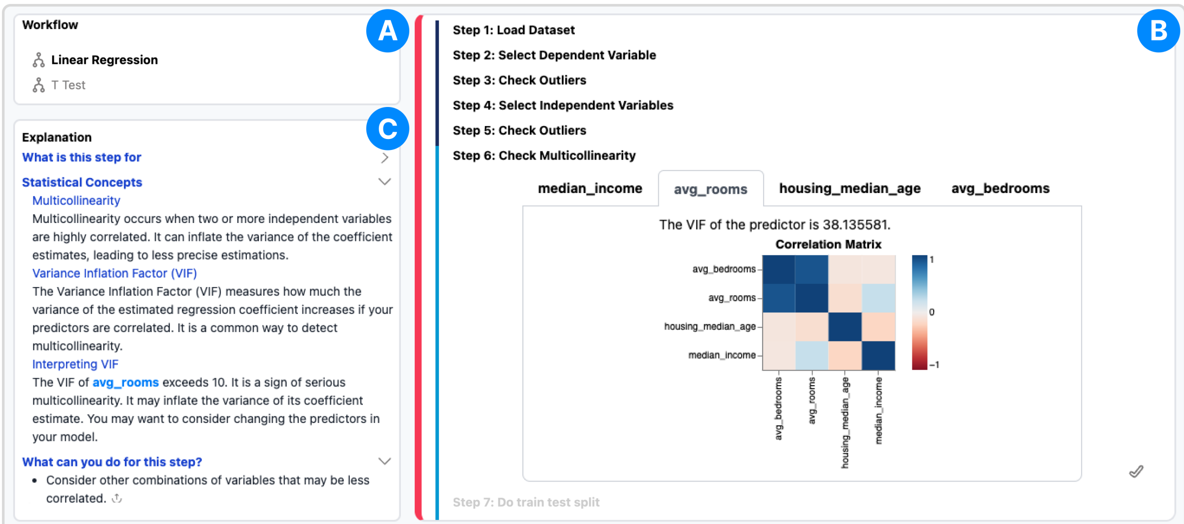

Fig. 1: *GuidedStats* helps users perform statistical analyses with guided workflows. (A) A user selects to run the linear regression workflow. (B) *GuidedStats* guides the user through the 9 steps of this workflow. Each step either solicits user input or helps users verify assumptions with visualizations and tests of the data. (C) Explanations are shown for the current step in the workflow. The user is currently on an *assumption checking step*; the explanation shows more information about the statistical concepts in the current step.

Abstract— Statistical practices such as building regression models or running hypothesis tests rely on following rigorous procedures of steps and verifying assumptions on data to produce valid results. However, common statistical tools do not verify users' decision choices and provide low-level statistical functions without instructions on the whole analysis practice. Users can easily misuse analysis methods, potentially decreasing the validity of results. To address this problem, we introduce *GuidedStats*, an interactive interface within computational notebooks that encapsulates guidance, models, visualization, and exportable results into interactive workflows. It breaks down typical analysis processes, such as linear regression and two-sample T-tests, into interactive steps supplemented with automatic visualizations and explanations for step-wise evaluation. Users can iterate on input choices to refine their models, while recommended actions and exports allow the user to continue their analysis in code. Case studies show how *GuidedStats* offers valuable instructions for conducting fluid statistical analyses while finding possible assumption violations in the underlying data, supporting flexible and accurate statistical analyses.

Index Terms—Data science, computational notebooks, guidance

1 Introduction

Statistical analysis is widely used to guide decision making and make sense of experimental results. However, producing *valid* statistical estimates requires following particular statistical procedures with specific data assumptions [1,20]. In practice, even basic analysis methods like null hypothesis testing or linear regression models are misused, leading to potentially incorrect analysis results [4,8,21,22].

One of the reasons statistical methods are misused is that assumptions are not properly verified [21]. All statistical models rely on assumptions about the input variables. If these assumptions are heavily violated, the statistical estimates may be biased and inconsistent, potentially leading to incorrect inferences and misguided decision making [4,13]. For example, during a two-sample t-test, analysts may neglect to check if the variances of the two samples are equal. The t statistic is calculated differently depending on whether variances are equal or not, and an incorrect t statistic might lead to a false rejection of the null hypothesis.

Correctly using statistical methods is difficult in part because the resources for *how* to do the analysis are separate from *where* the analysis takes place. Numerous textbooks and frameworks exist describing the assumptions that must be met when using statistical tools like hypothesis tests or linear regressions [9]. However, many analysts perform their analyses in coding environments like computational notebooks, where they can manipulate, visualize, and run statistics on their data using various packages. These code-based analysis environments are preferred over GUI-based tools since programming affords more flexibility in the analysis [16,18]. Despite their flexibility, these statistical analysis packages can be easily misused since they offer little **guidance** during the analysis. Guidance might take the form of explanations and instructions to assist users in verifying assumptions and transforming their data if assumptions are violated. Without guidance, the proper assumptions might not be checked and met, leading to the misuse of statistical methods.

In this paper, we introduce *GuidedStats*, an interactive system within computational notebooks to guide analysts through statistical workflows. *GuidedStats* contains different workflows for common statistical procedures that walk a user through steps to specify inputs for their analysis and check necessary assumptions. *GuidedStats* offers guidance in two primary ways: by walking a user through the proper steps in a workflow and explaining how to interpret each step. Step explanations include the step's objective, statistical concepts, and how to interpret the results. Suggestions are offered after checking assumptions to provide users with possible decisions based on the assumption results. Each suggestion includes a recommended action to take if the assumption is violated. Actions can populate inputs in later input steps or export code to transform the data to try to meet the assumption.

As a Jupyter Notebook Widget, *GuidedStats* allows fluid exchanges between the user interface and the coding environment. *GuidedStats* takes in a dataset for analysis; once the statistical workflow is complete, users can export the results back to code. During the analysis, users can explore or change the dataset used by *GuidedStats* with code.

We present two use cases where *GuidedStats* helps users find assumption violations and possible actions for the assumption checking results on linear regression and two sample t-test workflows. *GuidedStats* is open sourced and released at *URL removed for anonymity*.

In summary, our paper makes the following contributions.

1. A framework for guided statistical analysis with automatic assumption checking and explanations. Assumption checks help users ensure their data meets expected assumptions, while explanations help users understand the concepts and interpret the results at each step. Assumption violations can be corrected through recommended actions on the data.
2. *GuidedStats*, a computational notebook extension that incorporates these guidance elements into an interactive system for guided statistical workflows and supports hand-off between the interface and code.

2 Related Work

Commonly used statistical analysis tools such as SPSS, SAS, Stata, and R, provide UIs for statistical analysis but have ambiguous explanations and documentation [7]. This lack of support limits analysts in understanding the underlying concepts behind the methods they use [24]. Furthermore, these tools may not verify the assumptions of the statistical models the analysts choose, which leads to unreliable statistical estimates.

Researchers have developed various statistical tools that offer step-by-step guidance and verify assumptions during analysis. *StatHand* is a web-based application to assist students in identifying appropriate statistical tests [2]. It provides guidance on running tests in statistical tools and explanation of statistical concepts. However, *StatHand* can not perform statistical analysis in its application. Therefore, it is incapable of providing guidance on the decision choices made during the analysis. *Statsplorer* allows users to import data and helps them with learning and performing statistical tests on the data in its web-based application [23]. Except for explanation on statistical concepts, to assure validity, it verifies necessary test assumptions and visualizes the verification results. These tools explain statistical concepts and interpret model results. But they do not support the decision choices made during statistical analyses, such as how to manage the assumption checking results. In contrast, *GuidedStats* provides guidance specific to the current dataset and recommends actionable next steps for assumption violations.

Tea and *Tisane* provide a high-level declarative language to author statistical analyses such as Null Hypothesis Significance Test(NHST) or Generalized Linear Model(GLM) [10, 11]. They instruct users to explicitly specify their hypotheses, study designs, or conceptual relationships among variables. These systems then automatically verify assumptions and infer a valid Null Hypothesis Significance Test(NHST) or Generalized Linear Model(GLM). Besides, *Tisane* provides suggestions for possible solutions to the results of assumption verification. However, they still focus on explaining statistical concepts, with few instructions on how to proceed with the statistical analysis. Additionally, their suggestions for the assumptions are not linked to specific actions, requiring users to implement these suggestions by themselves. This may introduce potential misuse of statistical functions.

GuidedStats, in contrast, provides guidance for the entire statistical analysis process. The guidance includes explaining the objectives of each step and offering suggestions for interpreting the results of assumption verification. Analysts can refer to the provided suggestions whenever they have no clear idea about further actions. These suggestions are directly linked to actions in the user interface or can be exported as code. Besides, *GuidedStats* supports iterative model development by allowing direct changes on decision choices of previous steps, tightening the feedback loop of different decision choices.

Notably, current interactive statistical tools exist apart from where many analyses actually take place: computational notebooks. Computational notebooks integrate text, code, and visualization into a single document, supporting flexible and iterative analyses [16, 19]. Prior systems extend computational notebooks to better support revisiting history in analysis and data profiling [3, 12, 25]). Likewise, *GuidedStats* is a computational notebook extension to support easily jumping between GUI and code. A user provides an initial dataset to *GuidedStats*, which can be later updated if the data needs transformation after an assumption violation is discovered. At the end of an analysis, the results can be exported back to code.

3 *GuidedStats*: Step-by-step Statistical Workflows

GuidedStats is a computational notebook extension that has statistical **workflows** comprised of **steps** and **explanations**.

3.1 Statistical workflows

To start using *GuidedStats*, a user selects which statistical **workflow** they want to use. Each workflow represents a statistical analysis and guides users through the steps in the workflow. *GuidedStats* currently contains two workflows: Linear Regression and T-Tests, but can be extended in the future to other workflows. It proceeds from one step to the next only after users have set and confirmed all required decision choices at the current step. Workflows also support the iterative refinement of decision choices. Every time a previous step is edited, the following steps are rerun with the new parameters, enabling quick iteration when testing different decision choices.

Additionally, a workflow manages imports and exports for the analysis. As input, the workflow ingests the initial dataset. Once the analysis is complete, a user can export the created model and the report of the results. During the analysis, workflows can export code for each ongoing step and the dataset. If a dataset is transformed in the notebook, it can be re-imported back to the workflow, which will re-run the analysis up to the current step.

3.2 Steps in a workflow

In *GuidedStats*, workflows are comprised of individual **steps**. Each step represents a sub-task in the analysis. For instance, as shown in Figure 1, the linear regression workflow includes steps for identifying independent variables (Step 4) and verifying the assumption of outliers (Step 3 and 6). The assumptions used in *GuidedStats* are based on standard statistics textbooks [1, 6, 20]. The steps are modeled after examples from these textbooks [1, 6] and further refined using the documentation from statistical analysis packages like scikit-learn [15] There are three kinds of steps: User Inputs Steps, Result Presentation Steps, and Assumption Checking Steps (Table 1). User Input Steps enable users to set parameters in the analysis, such as selecting independent variables. Result Presentation Steps are always the last step in an analysis and present the results of a workflow. Finally, Assumption Checking Steps verify the assumptions of the current workflow. For example, in a linear regression workflow, this includes checking for outliers and multicollinearity.

The User Input Steps primarily affect accuracy through different inputs of decision choices. But **Assumption Checking Steps** mostly influence the validity of statistical models by addressing assumption violations. For instance, in a two-sample t-test, if the data fails to meet the assumption of normality, the user might transform the data or choose non-parametric tests instead. However, users may sometimes ignore minor violations in specific scenarios to preserve the original data. For example, with large sample sizes, the robustness of the t-test allows users to overlook non-normal distributions of two groups. Thus, instead of limiting actions for handling violations, the Assumption Checking

Step Type	Purpose	Explanation Components
User Input Step	Require user inputs from the interface to specify their decisions	1. *Objective:* purpose of the step 2. *Statistical Concepts & Interpretation:* explanations of important statistical concepts used in the step
Result Presentation Step	Presents the outcomes of the statistical analysis	1. *Objective:* purpose of the step 2. *Statistical Concepts & Interpretation:* interpretation of the modeling results
Assumption Checking Step	Verify the assumptions of models to ensure the validity of statistical estimates	1. *Objective:* purpose of the step 2. *Statistical Concepts & Interpretation:* the interpretation of the results of test used to verify the assumptions 3. *Suggested actions:* possible actions if the assumption is violated

Table 1: The types, purpose, and explanation components of steps. See § 3.3 for the example of each.

Step provides **Suggested Actions** in its explanation to recommend common actions for managing the results.

3.3 Step Explanations

Explanations are offered for each step in *GuidedStats* and have three components: the objective of this step, the statistical concepts & interpretation, and suggested actions. As shown in Table 1, only Assumption Checking Steps have suggested actions.

All steps have an objective explanation and a statistical concept explanation. The objective of the step clarifies why the step is included in the workflow and its importance to the overall statistical analysis. For instance, the step for stating the hypothesis in the two sample t-test has the objective "Formulate the null and alternative hypotheses for the t-test and define the Type I Error (alpha)".

The statistical concepts and interpretation explanation help users interpret the results in each step. This type of explanation is included in each step but is particularly important for the result presentation step. For example, in the result presentation for a linear regression, the explanation shows how to interpret the effect of the coefficients and the metrics for measuring the fitness of model, such as R^2.

Suggested actions are provided only for Assumption Checking Steps. These actions offer suggestions based on the assumption verification results. Actions take two forms: setting up the inputs for parameters in future steps or generating code to fix an assumption violation, if possible. For example, when checking the linear regression assumption of multicollinearity, a suggestion is presented as "consider other combinations of variables that are less correlated". Clicking on that action will produce a code cell to drop the variables with the highest correlation.

4 Use cases

In this section, we describe two example statistical analyses using *GuidedStats* with real-world datasets: a multivariate linear regression and a two-sample T-test. These analyses illustrate how analysts benefit from the design and guidance of *GuidedStats* during the analyses.

4.1 Linear Regression

In our first use case, we discuss an analyst who wants to understand how housing characteristics such as housing age and neighborhood income affect house value. For their analysis, they use the California Housing Prices Dataset, which contains 20,640 observations of the median house value for different blocks of houses in California [14]. They want to use a linear regression model to help them estimate the impact of different features on the median house value, so turn to *GuidedStats* to help them create a model.

First, the analyst imports their dataset and chooses the Linear Regression Workflow. This workflow contains 9 steps in total, including 3 assumption checking steps to check the core assumptions of linear regression: that large outliers are unlikely and there is no perfect multicollinearity [20]. After loading the data, the second step of this workflow is a variable selection step. In this step, the analyst selects `median_house_value` as the dependent variable from the list of columns presented. *GuidedStats* proceeds to Step 3 to help

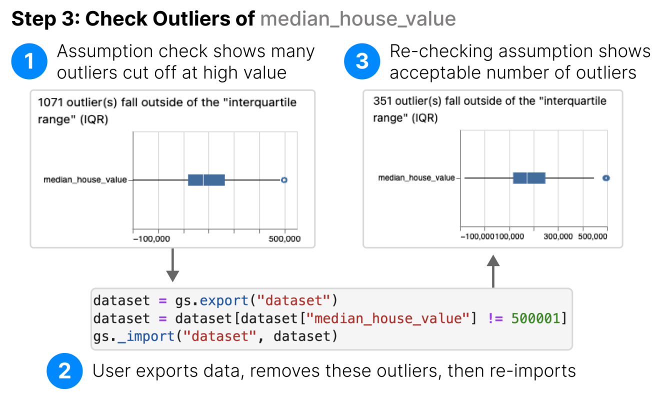

Fig. 2: *GuidedStats* supports an interactive loop of assumption checking, editing the data, then re-verifying assumptions.

the analyst verify the first assumption of a linear regression: that no large outliers exist in the target variable. As an assumption checking step, this step plots the distribution of `median_house_value` and tells the user that 1,071 outlier points lie outside the interquartile range (a common method for checking outliers). The explanation block for this step displays potential actions to deal with these outliers, such as removing the outlier points or transforming the column with a data transformation like a log transform. The analyst follows the suggestion to investigate the column before any manipulation and clicks on the action button for this suggestion. *GuidedStats* exports the example code for descriptive analysis on `median_house_value` to a new cell below. Its output shows that the most frequent value is 500,001, where `median_house_value` was manually clipped. Therefore, the analyst removes rows with `median_house_value = 500,001`, and imports the transformed dataset back to *GuidedStats* (Figure 2).

Next, in Step 4 (Variable Selection Step), the analyst selects `median_income`, `total_rooms`, `total_bedrooms`, `housing_median_age` as independent variables to see how they impact the median housing price. In the next step, *GuidedStats* once again helps the analyst check the assumption that no large outliers exist in the independent variables either. Inspecting the output from this step shows that some selected variables, `total_rooms` and `total_bedrooms`, have a large amount of outliers and thus might jeopardize the validity of the regression model if used. After reviewing the dataset description, the analyst decides to create more meaningful variables `avg_rooms` and`avg_bedrooms` as the average of the rooms and bedrooms for all houses in the block and use these as independent variables.

After ensuring no large outliers, *GuidedStats* proceeds to another assumption checking step to check for multicollinearity by calculating each independent variable's Variance Inflation Factor (VIF). This metric measures the correlation among the independent variables and its effect on accuracy. *GuidedStats* shows that the VIF of `avg_rooms` and `avg_bedrooms` are 38.14 and 30.57, meaning these two variables are of high concern for multicollinearity but may still be valid choices in

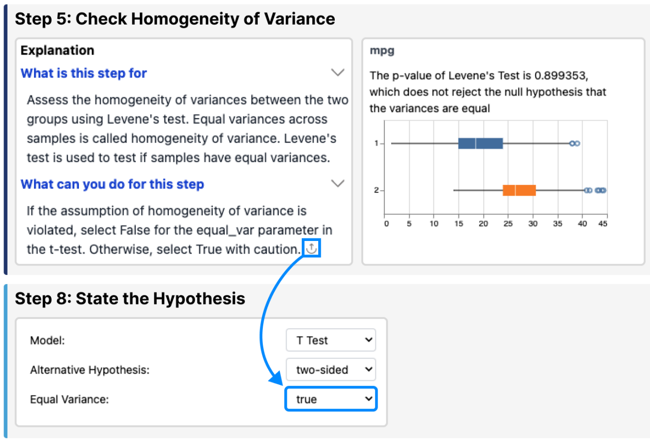

Fig. 3: For assumption checking steps, like the homogeneity of variance in the T-test workflow, *GuidedStats* recommends potential actions based on the results of assumption checks. In this example, the check suggests the variances are equal and the user can select the **action** to pre-set this parameter to True in the later model specification step.

context (Figure 1).

The analyst holds that concern and moves to the next two user input steps: data splitting and model hyperparameter selection. In the model specification step, they choose to use a simple linear regression. The analyst then moves on to the last step in the workflow for model evaluation. *GuidedStats* displays the coefficients and measures of goodness of fit for the trained model, including the R^2 value. The R^2 value, at 49.4%, indicates that approximately half of the variance in `median_house_value` is explained by selected independent variables. The positive coefficient for `avg_bedrooms` suggests that the median house price increases as the average number of bedrooms increases. Conversely, a negative coefficient for `avg_rooms` indicates as the average number of rooms increases, the median house price decreases. This somewhat contradictory result is likely due to the multicollinearity the analyst observed earlier. Such an issue undermines the model's interpretive power, which is a major concern. It highlights the necessity for further refinement of the model.

The analyst continues iterating on different combinations of independent variables in Step 4 (Variable Selection Step) while keeping other decision choices unchanged. They finalize a model with independent variables `median_income`, `avg_rooms`, `housing_median_age` and `households`. The model produced with these features has a slightly lower R^2 (0.47 compared to 0.49) than before. However, the analyst would rather use this newer model since there is no multicollinearity between the features.

4.2 Two Sample T-Test

In our second use case, we describe an analysis of car data to test if the miles per gallon (mpg) of cars differ between regions. This analysis uses the cars dataset, which contains 398 observations with 8 attributes about the cars [17]. Our analyst decides to test the null hypothesis that the cars have no difference in `mpg` between the US and Europe.

The analyst first imports the dataset into *GuidedStats* and selects the two sample t-test workflow. This workflow has 9 steps, with 4 of them to check the 3 core assumptions of a t-test. The workflow begins by asking the analyst for the variable where they select `mpg`. Next, *GuidedStats* guides the analyst to check the first assumption of a T-test: lack of outliers. Here, *GuidedStats* displays the same distribution overview used in the prior use case, showing only 1 outlier in `mpg`. Finding this acceptable, the analyst moves on to the next step, which is to select the two groups to compare in the t-test. They first select the column `origin`, then the values of US and Europe as the groups to compare.

GuidedStats continues to the next step to check the next assumption for t-tests: the homogeneity of variance between groups. Since the analyst is new to T-tests, they are unsure what this assumption is checking, they turn to the explanation block to learn more. *GuidedStats* explains that homogeneity of variance assesses whether the comparing groups have similar variances and can be evaluated using Levene's test [5] (Figure 3 top). *GuidedStats* plots the distributions of the two groups along with Levene's test result. The p-value of Levene's test is 0.90 and doesn't support rejecting that variances are equal. As suggested, the analyst clicks on its action button to pre-select the parameter for "equal variance" of the T-test as True in the later model specification step (Figure 3 bottom).

In steps 6 and 7 (Assumption Checking Steps), *GuidedStats* helps the analyst check the final assumption of a t-test on whether the two groups are normally distributed. The analyst initially observes that the distributions of the two groups are right-skewed in the density plot and is concerned that this assumption may be violated. However, *GuidedStats* includes a suggestion: "If the sample size is large enough (greater than 30 as a rule of thumb), the t-test is robust to violations of normality". When the analyst clicks on the action button following this suggestion, *GuidedStats* shows them a notice stating that the sizes of the two groups are both sufficiently large to be treated as normal. Therefore, the analyst bypasses the assumption and proceeds to Step 8 (Model Step). Here they select "two-sided" as the alternative hypothesis and retain the pre-selected "equal variance" parameter set to True from the earlier action. In the last step for evaluation, *GuidedStats* plots the mean difference of two groups and shows a t-statistic of -8.9147 with p-value close to zero. Consequently, the analyst rejects the null hypothesis and concludes that the cars have different `mpg` between the US and Europe.

In these analyses, *GuidedStats* helps the analysts identify and deal with outliers and multicollinearity in the linear regression model and verifies the assumptions of equal variance and normality in the two-sample t-test. The interpretation and suggested actions, such as interpreting high VIF scores and presetting the "equal variance" parameter, would have been difficult and time-consuming to manage without the support of *GuidedStats*.

5 Discussion and Future Work

In statistical analysis, verifying assumptions is important because it ensures the validity and reliability of results. *GuidedStats* provides a framework incorporating verification of assumptions with actionable explanations in the analysis workflow. It ensures that the assumptions of statistical models are well-supported by test results, visualizations, and explanations with suggested actions.

However, some assumptions can not be tested numerically as the assumptions lie beyond the dataset itself. For example, linear regression models assume variables are drawn randomly from the same distribution (i.i.d.) [20]. To verify this, users must know how data was collected before using it in their experiments. *GuidedStats* verifies assumptions mostly with only the dataset and thus does not cover the assumptions that are not numerically testable. It can provide suggestions on examining the randomness of data collection or export exploratory data analysis code to make sense of the dataset. However, such suggestions are not meaningful as they can not provide useful information from the dataset or the research design to assist in verifying the assumption. Moreover, adding steps with indirect suggestions could increase the ambiguity of statistical analyses to users. It may introduce new statistical concepts without interpretation or actions specific to the dataset or the research design. Therefore, in *GuidedStats* users have to verify numerically untestable assumptions themselves, which may still leave a threat to validity. Future work could expand to supporting assumptions that are difficult to verify by numerical calculation, which is the basis of current statistical tools.

As a computational notebook extension, *GuidedStats* supports moving from code-based analysis, into an interface, and then back to code. However, the guidance and explanations of *GuidedStats* are currently limited to the interface. Future work might explore other methods for augmenting statistical programming, such as monitoring the code that users write and notifying them of problems. Furthermore, other interfaces such as natural language chat-based tools can be used to permit users to ask questions about the workflow or their data.

References

[1] A. Agresti. *Statistical methods for the social sciences*. Pearson Prentice Hall, Upper Saddle River, N.J., array ed., 2009. 1, 2

[2] P. J. Allen, L. D. Roberts, F. D. Baughman, N. J. Loxton, D. V. Rooy, A. J. Rock, and J. Finlay. Introducing stathand: A cross-platform mobile application to support students' statistical decision making. *Frontiers in Psychology*, 7, 2016. 2

[3] W. Epperson, V. Gorantla, D. Moritz, and A. Perer. Dead or alive: Continuous data profiling for interactive data science. *IEEE Transactions on Visualization and Computer Graphics*, 30:197–207, 2023. 2

[4] A. F. Ernst and C. J. Albers. Regression assumptions in clinical psychology research practice—a systematic review of common misconceptions. *PeerJ*, 5, 2017. 1

[5] J. L. Gastwirth, Y. R. Gel, and W. Miao. The impact of levene's test of equality of variances on statistical theory and practice. *Quality Engineering*, 24:343, 2009. 4

[6] C. Hanck, M. Arnold, A. Gerber, and M. Schmelzer. *Introduction to Econometrics with R*. Universität Duisburg-Essen, 2021. 2

[7] C. B. Hodges, B. M. Stone, P. K. Johnson, J. H. Carter, C. K. Sawyers, P. R. Roby, and H. M. Lindsey. Researcher degrees of freedom in statistical software contribute to unreliable results: A comparison of nonparametric analyses conducted in spss, sas, stata, and r. *Behavior Research Methods*, 55:2813 – 2837, 2022. 2

[8] R. Hoekstra, H. A. L. Kiers, and A. Johnson. Are assumptions of well-known statistical techniques checked, and why (not)? *Frontiers in Psychology*, 3, 2012. 1

[9] E. Jun, M. K. Birchfield, N. de Moura, J. Heer, and R. Just. Hypothesis formalization: Empirical findings, software limitations, and design implications. *ArXiv*, abs/2104.02712, 2021. 1

[10] E. Jun, M. Daum, J. Roesch, S. E. Chasins, E. Berger, R. Just, and K. Reinecke. Tea: A high-level language and runtime system for automating statistical analysis. *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*, 2019. 2

[11] E. Jun, A. Seo, J. Heer, and R. Just. Tisane: Authoring statistical models via formal reasoning from conceptual and data relationships. *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, 2022. 2

[12] M. B. Kery, D. Ren, F. Hohman, D. Moritz, K. Wongsuphasawat, and K. Patel. mage: Fluid moves between code and graphical work in computational notebooks. *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*, 2020. 2

[13] J. W. Osbourne and E. Waters. Four assumptions of multiple regression that researchers should always test. *Practical Assessment, Research and Evaluation*, 8:2, 2002. 1

[14] R. K. Pace and R. P. Barry. Sparse spatial autoregressions. *Statistics & Probability Letters*, 33:291–297, 1997. 3

[15] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011. 2

[16] J. M. Perkel. Why jupyter is data scientists' computational notebook of choice, Oct 2018. 1, 2

[17] R. Quinlan. Auto MPG. UCI Machine Learning Repository, 1993. DOI: https://doi.org/10.24432/C5859H. 4

[18] A. Rule, A. Tabard, and J. D. Hollan. Exploration and explanation in computational notebooks. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, p. 1–12. Association for Computing Machinery, New York, NY, USA, 2018. doi: 10.1145/3173574.3173606 1

[19] A. Rule, A. Tabard, and J. D. Hollan. Exploration and explanation in computational notebooks. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, p. 1–12. Association for Computing Machinery, New York, NY, USA, 2018. doi: 10.1145/3173574.3173606 2

[20] J. H. Stock and M. W. Watson. *Introduction to econometrics*. Pearson, 2020. 1, 2, 3, 4

[21] M. S. Thiese, Z. C. Arnold, and S. D. Walker. The misuse and abuse of statistics in biomedical research. *Biochemia medica*, 25(1):5–11, 2015. 1

[22] J. B. Vornhagen, A. Tyack, and E. D. Mekler. Statistical significance testing at chi play: Challenges and opportunities for more transparency. *Proceedings of the Annual Symposium on Computer-Human Interaction in Play*, 2020. 1

[23] C. Wacharamanotham, K. Subramanian, S. T. Völkel, and J. Borchers. Statsplorer: Guiding novices in statistical analysis. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, p. 2693–2702. Association for Computing Machinery, New York, NY, USA, 2015. 2

[24] C. Wild and M. Pfannkuch. Statistical thinking in empirical enquiry. *International Statistical Review*, 67, 1999. 2

[25] Y. Wu, J. M. Hellerstein, and A. Satyanarayan. B2: Bridging code and interactive visualization in computational notebooks. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*, UIST '20, p. 152–165. Association for Computing Machinery, New York, NY, USA, 2020. 2