Automated Residual Plot Assessment with the R Package autovi and the Shiny App autovi.web

Response to reviewers

2025-05-14

We thank the reviewers for their comments (colored in purple below). We have addressed it as below (in black). We also submit the difference between the previously submitted document and the revised version produced via `latexdiff` (named `diff.pdf`).

Referee 1

1. The Abstract could be improved further. Certain sentences in the Abstract contain grammatical errors and are hard to read.

We have revised the abstract to improve its clarity and readability. The new version is as follows.

> *Visual assessment of residual plots is a common approach for diagnosing linear models, but it relies on manual evaluation, which does not scale well and can lead to inconsistent decisions across analysts. The lineup protocol, which embeds the observed plot among null plots, can reduce subjectivity but requires even more human effort. In today's data-driven world, such tasks are well-suited for automation. We present a new R package that uses a computer vision model to automate the evaluation of residual plots. An accompanying Shiny app is provided for ease of use. Given a sample of residuals, the model predicts a visual signal strength (VSS) and offers supporting information to help analysts assess the adequacy of their model fit.*

2. The "autovi.web" RShiny app (e.g., https://akdd4t-patrick-li.shinyapps.io/autovi_web/) lacks clear instructions regarding the format of CSV files it accepts. As a result, some users may upload the original dataset instead of a two-column CSV file with fitted values and residuals.

We have updated the app to include clearer instructions on the required input format. More specifically, the app now includes information for specific requirement of the input CSV and an information icon which shows another optional input column for multiple lineups on hover.

3. One suggestion is to consider integrating the widely-used regression model fitting functions (e.g., lm(), aov() or glm ()) into "autovi" to automate the model fitting and assessment of residuals.

Thank you for the suggestion. We agree that this is a valuable direction for future development. We have added a new Section 4.6 that discusses how the workflow could be extended to accommodate broader model classes. We note, however, that this extension is only applicable for certain class of models as the computer vision model is trained on a fitted-vs-residual plot, which is not the optimal plot for diagnostic of, say, logistic model. Future research would benefit from building computer vision models tailored for visual dianostics suitable for other class of models.

4. Finally, please check the English in the manuscript. Certain lengthy sentences can be hard to understand.

We have revised the manuscript to improve clarity and grammar. These are most easily seen via the document (`diff.pdf`) which shows the difference between the original and revised version. Notable changes include:

- Section 2
  - Original: "VSS is an estimate of the distance between the residual distribution of a fitted classical normal linear regression model and a reference distribution; more details can be found in Li et al. 2024b."
  - Revised: "VSS estimates the distance between the residual distribution of a fitted classical normal linear regression model and a reference distribution (see Li et al. 2024b, for65details)."
- Section 5.1
  - Original: "Due to the resource allocation policy of shinyapps.io, the server enters a sleep mode during periods of inactivity, resulting in the clearing of the local Python virtual environment. Consequently, when the application"wakes up" for a new user session, these libraries need to be reinstalled."
  - Revised: "Due to shinyapps.io's resource policy, inactive servers enter sleep mode, clearing the local Python environment. When reactivated for a new session, libraries must be reinstalled."
- Section 4.1
  - Original: "The third example is generated using the `surreal` package (Balamuta 2024) where structured residuals are hidden in data, to be revealed if the correct

model is specified. Here a quote based on Tukey is used as the residual structure 'visual summaries focus on unexpected values'."
  - Revised: "The third example is generated using the `surreal` package (Balamuta 2024), where structure residuals are embedded in the data. In this case, a quote inspired by Tukey 'visual summaries focus on unexpected values', is used to define the residual structure."

- Section 4.1
  - Original: "This structure is blindingly obvious visually,"
  - Revised: "Visually, the structure is strikingly clear,"

Referee 2

- some introduction on generating the null plot and bootstrap residual, which provides some background to understand the code summary in page 7 and Figure 3 on the comparison between VSS from the two scenarios.

We have added a new Section 3 (shown below) to introduce the definitions and generation procedures for null plots and bootstrapped residuals, providing necessary context for Figure 3.

> *In the subsequent sections, we will frequently refer to null residuals and bootstrapped residuals, so it is helpful to first define and explain how they are generated.*
>
> ***Null residuals*** *are used to generate null plots within the lineup protocol framework, serving as the foundation for the statistical testing in our automated residual plot assessment. Specifically, they represent residuals generated under the null hypothesis that the model is correctly specified. A common method for simulating null residuals in linear regression involves sampling from a normal distribution with mean zero and variance equal to the estimated variance of the error term. These simulated residuals and their corresponding plots depict what one would expect from a correctly specified model. If the true residual plot exhibits noticeable deviations from these null plots, it may suggest model misspecification.*
>
> *Our computer vision model is trained to assign lower VSS to null plots and higher VSS to plots that display distinct patterns. Accordingly, statistical testing is performed by computing the proportion of null plots whose VSS equals or exceeds that of the observed residual plot. This proportion serves as a p-value for a one-sided hypothesis test.*
>
> ***Bootstrapped residuals*** *are obtained by refitting the model on bootstrap samples, which are generated by sampling individual observations with replacement from the original dataset. The residual plots obtained from these refitted models are evaluated using the same computer vision model. The predicted VSS from the bootstrapped plots provide an empirical estimate of the variation in the VSS of the observed residual plot. By examining*

the proportion of bootstrapped plots that also exhibit significant violations, we can assess whether the original conclusion is robust to sampling variability.

- information on the type/ scope of regression models used to train the cv model in Section 2 to understand the applicability of the cv model.

We have expanded the final paragraph of Section 2 to describe the types of regression models used for training. This provides better insight into the model's scope and applicability.

The computer vision model approximates this mapping from a set of residuals to its corresponding distance measure. It is trained on a large number of synthetic regression models, each designed to simulate specific violations of classical linear regression assumptions. These models incorporate non-linearity through Hermite polynomial transformations of predictors, heteroskedasticity by making the error variance a predictor-dependent function, and non-normality by drawing residuals from distributions such as discrete, uniform, and lognormal. Both simple and multiple linear regression structures are used, with controlled parameters to generate diverse and complex residual patterns. Since the data-generating process is known, the distance measure D can be explicitly calculated, enabling supervised training. The computer vision model takes a residual plot as input and outputs the corresponding distance measure, learning to quantify model violations directly from visual patterns. Additional details are provided in Li et al. (2024).

- a brief description of each module in Figure 4, for example, what does auxiliary computation (in Figure 4) include?

We have extended the last paragraph of Section 4.5 to include brief descriptions of each module in Figure 4, including the tasks performed under "auxiliary computation".

The package takes regression models and a `Keras` model as primary inputs. Modules for VSS prediction and p-value computation are fixed but accessible via function arguments, using `TensorFlow` for inference and statistical testing. The image loading module is also fixed, using `PIL` to read and resize images based on the `Keras` model's input shape. The remaining seven modules are overridable, allowing users to adapt the workflow as needed. The data extraction module extracts a `data.frame` containing variables used in the regression model. The bootstrapping and refitting module resamples the data and refits the model. The fitted values and residuals extraction module returns these values as a `data.frame`. The auxiliary computation module calculates scagnostics such as monotonicity. The plotting module generates a `ggplot` in a standard format, and the plot saving module exports it at the same resolution as the training images. These modules are described in detail in the package documentation.

- the necessary changes need to make to extend the workflow to diagnose other models e.g. GLM, in Section 3.5.

This will require training and testing the computer vision model across a broader range of models, including those with different types of assumption departures. Future research could build on our prototype to support this broader class. In the meantime, we have added Section 4.6 to illustrate how the workflow could potentially be extended to other model types.

- the term "attention map" used in line 331 and Figure 7 caption is not introduced in the paper.

We have added a definition for "attention map" as follows:

> *The attention map is the gradient of the model output with respect to the grayscale image input, indicating the sensitivity of the output to each pixel.*

- The authors may consider moving Section 3.5 earlier in Section 3 to provide an overview of the package.

Thank you for the suggestion. We considered this structure but chose to prioritize a streamlined usage summary early in the manuscript, based on the assumption that many readers will be primarily interested in applying the tool.

- The infrastructure is built on the bandicoot OOP system, could the authors comment on this choice compared to the latest S7?

The `bandicoot` system is a custom object-oriented system tailored for our development needs, in a similar spirit to how `ggproto` is used within `ggplot2`. The `S7` system is quite new (first published on CRAN on 2024-11-07) and yet to be mature or have widespread usage.

- The word "be" in line 60 is redundant in the sentence.

Thank you. This has been corrected.