The Journal of Open Source Software

LazyModeler: An R package for automatic simplification, check, and visualization of regression models

Lara M. Kösters [1,2*] **and Kevin Karbstein** [1,2*]

1 Max-Planck-Institute for Biogeochemistry, Department of Biogeochemical Integration, Jena, Germany
2 Technical University Ilmenau, Data-Intensive Systems and Visualization Group (dAI.SY), Ilmenau, Germany ***** These authors contributed equally.

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor: Open Journals

Reviewers:

- @openjournals

Submitted: 01 January 1970
Published: unpublished

Summary

Setting up, simplifying, checking, and visualizing regression models continues to be a time-consuming task involving multiple, sometimes concurrent, workflows and software packages. This particularly applies to big data research where several models with different response variables and many explanatory variables need to be set up and optimized. To tackle this problem, we present `LazyModeler` - a statistical package for the programming language R that allows to easily perform regression modeling. It includes removal of autocorrelated variables, choice between several types of (non)linear regression models, standard stepwise model simplification, various model quality checks, plotting of coefficient estimates and relationships, and output generation. `LazyModeler` will significantly speed up regression modeling, enabling people to analyze and illustrate their data in a statistically reliable and standardized manner.

Statement of need

Statistical modeling describes the process of finding a mathematical function with specific statistical assumptions that best fits the observed data (Crawley, 2007, 2015; Henley et al., 2020). This process attempts, in practice, to find a (causal) relationship between a dependent response variable y and an independent predictor variable x for any postulated hypothesis. For statistical inference and graphics in science, the programming environment R (R Core Team, 2024) has become highly popular.

Linear models, as one of the most basic and powerful tools, have been frequently applied in this context (Crawley, 2007, 2015; Li, 2023; Schielzeth et al., 2020). Because of their flexibility, they also allow for non-normally distributed response variables (e.g., in the case of binomial, proportional, or count data), and any kind of transformation for numerical (e.g., polynomial or logarithmic) and categorical (e.g., centered or one-hot/fractional encoded) predictor variables, as well as interactions among them (Cai et al., 2023; Henley et al., 2020; Karbstein et al., 2019, 2020, 2021; Liaw et al., 2021; Römermann et al., 2016; Schielzeth, 2010). Regression models also provide the ability to control for random effects that may influence the variables of interest (e.g., Bauer & Albrecht, 2020; Schielzeth et al., 2020; Wicke et al., 2016). Although other statistical technologies can outperform them in highly complex, non-linear scenarios, regression models allow for detailed variable transformation and interaction, mathematical formula specification, calculation of effect sizes, determination of variable significance, and thus hypothesis testing and explanation (Benjamin et al., 2018; Bzdok & Ioannidis, 2019; Cai et al., 2023; Karbstein et al., 2023; Li, 2023; Schulz et al., 2020). Recent developments make regression models also applicable to nonlinear scenarios (e.g., Bates et al., 2024; Hastie, 2023). Consequently, they are of high practical value in finding and interpreting significant relationships.

In statistical modeling, and especially in real-world applications, multiple predictors are assumed for a given response variable. As a consequence, people strive to exclude the irrelevant from the relevant (statistically significant) information, which is called model simplification (Crawley, 2007, 2015; Forstmeier & Schielzeth, 2011). One of the most widely used optimization workflows is stepwise model simplification. For example, starting from a full/saturated model, the least significant variable ($p>0.05$) is excluded until the final minimal adequate model is attained [‘backward simplification’; Crawley (2007); Forstmeier & Schielzeth (2011); Crawley (2015)]. Each model simplification step will be justified with certain metrics (e.g., SSE, AIC, or BIC) (Henley et al., 2020). Given the number of models, variables of interest, and their data characteristics, this task can be extraordinarily time consuming. Currently, only AIC/BIC-based automated simplification is available (e.g., ‘stepAIC,’ Venables & Ripley, 2002). Nevertheless, model simplification continuous to be a rather manual process [on GoogleScholar, only ca. 5,000 “stepAIC” entries despite ca. 5,000,000 “linear regression model” studies (0.1%); e.g., Römermann et al. (2016); Karbstein et al. (2019); Henley et al. (2020); Karbstein et al. (2020); Cai et al. (2023); Li (2023)]. In addition, simplification and other aspects such as data cleaning, model comparison and quality control, and output visualization have not been automated. An easy-to-use, all-in-one function for the entire modeling process within a single software package is missing.

Our R package `LazyModeler` addresses these issues by automating variable selection, model optimization, and output illustration and generation. In detail, users will be enabled to automatically remove autocorrelated variables, choose between several types of (non)linear regression models (e.g., LM, GLM, LMER, GLMER, GAM, or NLMER), perform stepwise model simplification, check model quality, plot coefficient estimates and relationships, and generate the output of the final model.

Overview and major functions

`LazyModeler` automatizes all necessary steps needed for use of (non)linear regression models. It comprises three major functions that are included within the main function `optimize_model`.

The first major function `remove_autocorrelations` checks for any autocorrelations ($|r| > 0.7$) (Dormann et al., 2013) given a list of variables sorted by relevance. Automatic removal of these autocorrelations is possible through the use of a function parameter. Removal will follow the order of the list of variables, ensuring that the user’s expertise on the importance of features is respected. A named list is returned with a) a vector containing all removed predictors, and b) a dataframe listing autocorrelations and information on deleted variables.

The main function provides the model formula to the second major function `simplify_model`. If autocorrelations were detected, the formula is updated accordingly. The regression model is then calculated. Options for the models are: `lm`, `glm`, `lmer`, `glmer`, `gam`, or `nlmer`, with all possible distributions of the response variable being allowed. Stepwise backward simplification or forward model selection takes place using an iterative process where each time the metric(s) specified by the user are applied on the model to check whether further simplification/selection is needed. Main variables are kept when they are involved in interactions. Options for the metrics are: `aov`, `aic`, `aicc`, or `bic`. The final model is returned to the main function alongside its metadata as well as simplification history if requested by the user.

Using the third major function `plot_model_features`, the final model then undergoes multiple visualization steps. Plots to assess model quality are created using the standard plot function available through base R, or model check included in the `performance` R package (Lüdecke et al., 2021). Furthermore, the script produces regression, box, or violin plots for each numerical or categorical coefficient as well as plots depicting effects sizes and estimates. All generated plots are returned to the user within a named list. The main function additionally returns the output of both the model simplification/selection and autocorrelation functions as well as the summary of the final model.

LazyModeler makes use of the R package corrplot (Wei & Simko, 2021) to calculate correlations between variables, lme4 (Bates et al., 2024) and lmerTest (Kuznetsova et al., 2017) for regression modeling, tidyverse (Wickham et al., 2019) for data handling, and MuMIn (Bartoń, 2024) for calculation of AICc scores. For generation of plots visualizing regression, effect size, and estimates, the script further leverages tidyverse and color palettes included in the colorspace (Zeileis et al., 2020) and viridis (Garnier et al., 2024) R packages.

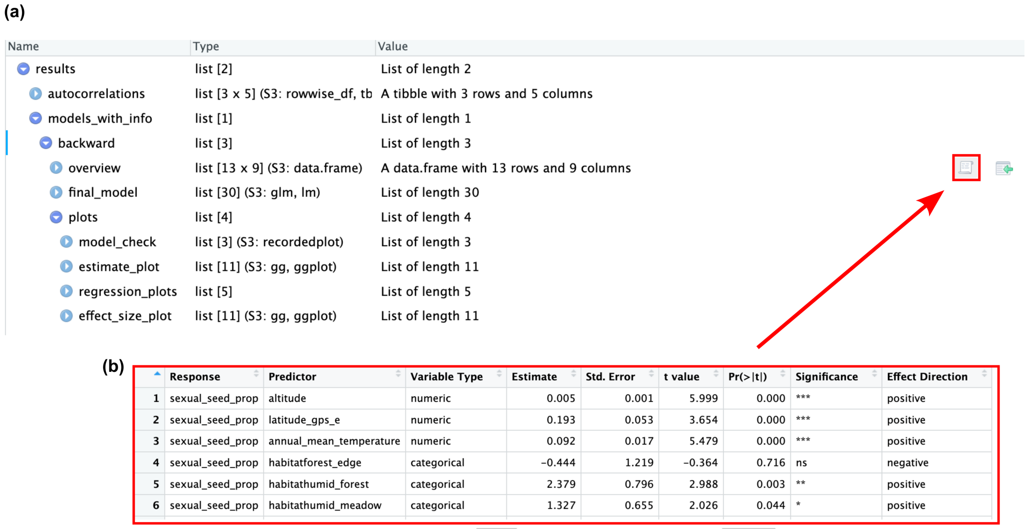
Example

```
# import example data
data(plants)

# check data structure
str(plants)
summary(plants)

# testing dataset (subset) based on Karbstein et al. 2021
# (https://onlinelibrary.wiley.com/doi/10.1111/mec.15919)

results_example <- optimize_model(plants,
    quote(sexual_seed_prop ~
    altitude + latitude_gps_n + longitude_gps_e +
    (solar_radiation + annual_mean_temperature +
    isothermality)^2 + I(isothermality^2) +
    habitat + ploidy),
    autocorrelation_cols = c("solar_radiation",
    "annual_mean_temperature", "isothermality", "altitude",
    "latitude_gps_n", "longitude_gps_e"),
    automatic_removal=TRUE,
    autocorrelation_threshold = 0.8,
    correlation_method="spearman",
    model_type = "glm",
    model_family = "quasibinomial",
    assessment_methods=c("anova"),
    simplification_direction="backward",
    omit.na="overall",
    scale_predictor=TRUE,
    plot_quality_assessment="performance",
    round_p=3,
    cor_use="complete.obs",
    plot_relationships=TRUE,
    jitter_plots=TRUE,
    plot_type="violinplot",
    stat_test="wilcox",
    backward_simplify_model=TRUE,
    trace=TRUE)
```



(a)

Name	Type	Value
results	list [2]	List of length 2
autocorrelations	list [3 x 5] (S3: rowwise_df, tb	A tibble with 3 rows and 5 columns
models_with_info	list [1]	List of length 1
backward	list [3]	List of length 3
overview	list [13 x 9] (S3: data.frame)	A data.frame with 13 rows and 9 columns
final_model	list [30] (S3: glm, lm)	List of length 30
plots	list [4]	List of length 4
model_check	list [3] (S3: recordedplot)	List of length 3
estimate_plot	list [11] (S3: gg, ggplot)	List of length 11
regression_plots	list [5]	List of length 5
effect_size_plot	list [11] (S3: gg, ggplot)	List of length 11

(b)

	Response	Predictor	Variable Type	Estimate	Std. Error	t value	Pr(>|t|)	Significance	Effect Direction
1	sexual_seed_prop	altitude	numeric	0.005	0.001	5.999	0.000	***	positive
2	sexual_seed_prop	latitude_gps_e	numeric	0.193	0.053	3.654	0.000	***	positive
3	sexual_seed_prop	annual_mean_temperature	numeric	0.092	0.017	5.479	0.000	***	positive
4	sexual_seed_prop	habitatforest_edge	categorical	−0.444	1.219	−0.364	0.716	ns	negative
5	sexual_seed_prop	habitathumid_forest	categorical	2.379	0.796	2.988	0.003	**	positive
6	sexual_seed_prop	habitathumid_meadow	categorical	1.327	0.655	2.026	0.044	*	positive

Figure 1: Navigating through the output. For example, (a) simply click on dataframe button highlighted with a red arrow to (b) illustrate the final model output.

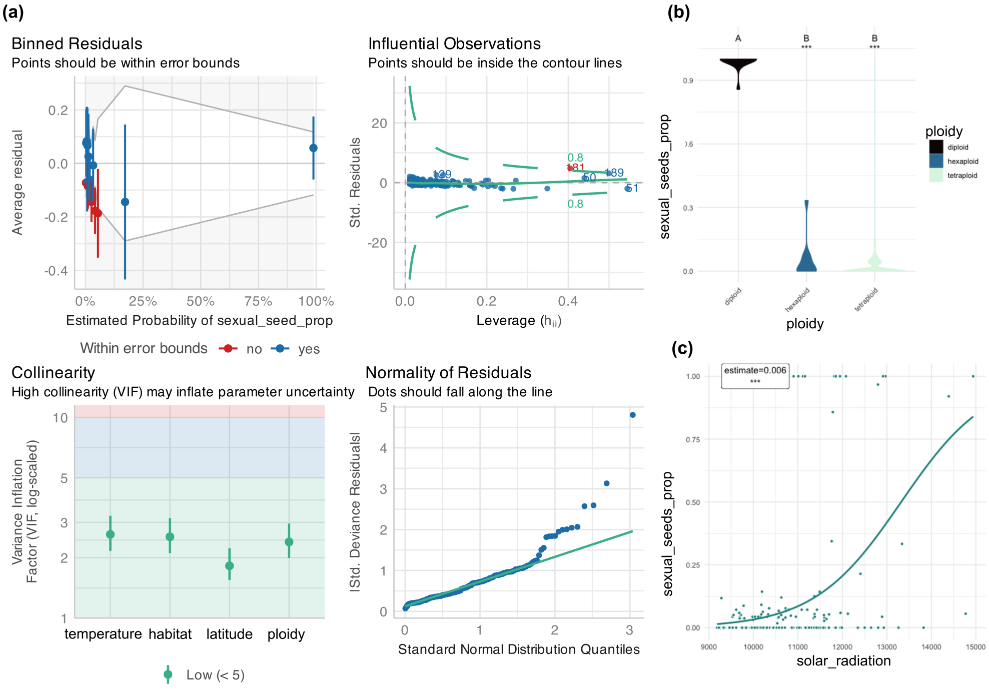


Figure 2: (a) Model quality check and (b,c) exemplary output plots of significant relationships.

Conclusions

In summary, `LazyModeler` streamlines the process of building, simplifying, and visualizing regression models in R. By automating key steps such as autocorrelation removal, model selection, quality assessment, and output generation, it significantly reduces manual effort. The package is especially valuable for researchers dealing with large and complex datasets who seek a reproducible and statistically sound regression modeling workflow. We anticipate that `LazyModeler` will serve as a practical and accessible tool for both novice and experienced users in the scientific community.

Important note

The model selection procedures implemented in LazyModeler are provided for convenience and exploratory analysis, and reflect practices recommended in widely used applied statistics literature (Crawley, 2007, 2015). Users should be aware, however, that statistical inference reported from a model chosen in a data-driven way may be anti-conservative (e.g., p-values may appear smaller than they truly are, confidence intervals narrower). This issue is known as post-selection inference (PSI). Specialized methods have been developed to address it, for instance (Lee et al., 2016), but they are not yet broadly applicable across the full range of model classes supported by LazyModeler. We have implemented PSI for (generalized) linear regression models based on the 'selcorr' R package (Cattaneo, 2021), but users are free to use the retained model from LazyModeler for more sophisticated PSI analyses.

Code availability

The code including basic documentation and an exemplary testing dataset will be made available upon publication on Github and on Comprehensive R Archive Network (CRAN).

Acknowledgements

We acknowledge financial support from the German Federal Ministry of Education and Research (BMBF) grant 01IS20062.

References

Bartoń, K. (2024). *MuMIn: Multi-model inference*. https://doi.org/10.32614/cran.package.mumin

Bates, D., Maechler, M., Bolker, B., & Walker, S. (2024). *lme4 - Linear mixed-effects models using 'Eigen' and S4*. https://github.com/lme4/lme4/

Bauer, M., & Albrecht, H. (2020). Vegetation monitoring in a 100-year-old calcareous grassland reserve in Germany. *Basic and Applied Ecology*, *42*, 15–26. https://doi.org/10.1016/j.baae.2019.11.003

Benjamin, A. S., Fernandes, H. L., Tomlinson, T., Ramkumar, P., VerSteeg, C., Chowdhury, R. H., Miller, L. E., & Kording, K. P. (2018). Modern machine learning as a benchmark for fitting neural responses. *Frontiers in Computational Neuroscience*, *12*(July), 1–13. https://doi.org/10.3389/fncom.2018.00056

Bzdok, D., & Ioannidis, J. P. A. (2019). Exploration, Inference, and Prediction in Neuroscience and Biomedicine. *Trends in Neurosciences*, *42*(4), 251–262. https://doi.org/10.1016/j.tins.2019.02.001

Cai, L., Kreft, H., Taylor, A., Denelle, P., Schrader, J., Essl, F., Kleunen, M. van, Pergl, J., Pyšek, P., Stein, A., Winter, M., Barcelona, J. F., Fuentes, N., Inderjit, Karger, D. N., Kartesz, J., Kuprijanov, A., Nishino, M., Nickrent, D., … Weigelt, P. (2023). Global models and predictions of plant diversity based on advanced machine learning techniques. *New Phytologist*, *237*(4), 1432–1445. https://doi.org/10.1111/nph.18533

Cattaneo, M. (2021). *Selcorr: Post-selection inference for generalized linear models*. https://CRAN.R-project.org/package=selcorr

Crawley, M. J. (2007). *The R Book* (p. 942). John Wiley & Sons, Ltd. https://doi.org/10.1002/9780470515075

Crawley, M. J. (2015). *Statistics: an introduction using R* (sec. ed., p. 339). John Wiley & Sons. ISBN: 1118448960

Dormann, C. F., Elith, J., Bacher, S., Buchmann, C., Carl, G., Carré, G., Marquéz, J. R. G., Gruber, B., Lafourcade, B., Leitão, P. J., Münkemüller, T., McClean, C., Osborne, P. E., Reineking, B., Schröder, B., Skidmore, A. K., Zurell, D., & Lautenbach, S. (2013). Collinearity: a review of methods to deal with it and a simulation study evaluating their performance. *Ecography*, *36*(1), 27–46. https://doi.org/10.1111/j.1600-0587.2012.07348.x

Forstmeier, W., & Schielzeth, H. (2011). Cryptic multiple hypotheses testing in linear models: overestimated effect sizes and the winner's curse. *Behavioral Ecology and Sociobiology*, *65*(1), 47–55. https://doi.org/10.1007/s00265-010-1038-5

Garnier, Simon, Ross, Noam, Rudis, Robert, Camargo, Pedro, A., Sciaini, Marco, Scherer, & Cédric. (2024). *viridis(Lite) - colorblind-friendly color maps for r*. https://doi.org/10.5281/zenodo.4679423

Hastie, T. (2023). *gam: Generalized Additive Models*. https://cran.r-project.org/web/packages/gam/index.html

Henley, S. S., Golden, R. M., & Kashner, T. M. (2020). Statistical modeling methods: challenges and strategies. *Biostatistics & Epidemiology*, *4*(1), 105–139. https://doi.org/10.1080/24709360.2019.1618653

Karbstein, K., Prinz, K., Hellwig, F., & Römermann, C. (2020). Plant intraspecific functional trait variation is related to within-habitat heterogeneity and genetic diversity in Trifolium montanum L. *Ecology and Evolution*, *10*(11), 5015–5033. https://doi.org/10.1002/ece3.6255

Karbstein, K., Römermann, C., Hellwig, F., & Prinz, K. (2023). Population size affected by environmental variability impacts genetics, traits, and plant performance in Trifolium montanum L. *Ecology and Evolution*, *13*(8), 1–19. https://doi.org/10.1002/ece3.10376

Karbstein, K., Tomasello, S., Hodač, L., Lorberg, E., Daubert, M., & Hörandl, E. (2021). Moving beyond assumptions: Polyploidy and environmental effects explain a geographical parthenogenesis scenario in European plants. *Molecular Ecology*, *30*(11), 2659–2675. https://doi.org/10.1111/mec.15919

Karbstein, K., Tomasello, S., & Prinz, K. (2019). Desert-like badlands and surrounding (semi-)dry grasslands of Central Germany promote small-scale phenotypic and genetic differentiation in *Thymus praecox*. *Ecology and Evolution*, *9*(24), 14066–14084. https://doi.org/10.1002/ece3.5844

Kuznetsova, A., Brockhoff, P. B., & Christensen, R. H. B. (2017). lmerTest package: Tests in linear mixed effects models. *Journal of Statistical Software*, *82*(13), 1–26. https://doi.org/10.18637/jss.v082.i13

Lee, J. D., Sun, D. L., Sun, Y., & Taylor, J. E. (2016). Exact post-selection inference, with application to the lasso. *Annals of Statistics*, *44*(3), 907–927. https://doi.org/10.1214/

15-AOS1371

Li, J. (2023). Overview of high dimensional linear regression models. *Theoretical and Natural Science*, *5*(1), 656–661. https://doi.org/10.54254/2753-8818/5/20230427

Liaw, K., Khomik, M., & Arain, M. A. (2021). Explaining the shortcomings of log-transforming the dependent variable in regression models and recommending a better alternative: Evidence from soil CO_2 emission studies. *Journal of Geophysical Research: Biogeosciences*, *126*(5), 1–18. https://doi.org/10.1029/2021JG006238

Lüdecke, D., Ben-Shachar, M. S., Patil, I., Waggoner, P., & Makowski, D. (2021). Performance: An r package for assessment, comparison and testing of statistical models. *Journal of Open Source Software*, *6*(60), 3139. https://doi.org/10.21105/joss.03139

R Core Team. (2024). *R: a language and environment for statistical computing*. R Foundation for Statistical Computing. http://www.r-project.org/

Römermann, C., Bucher, S. F., Hahn, M., & Bernhardt-Römermann, M. (2016). Plant functional traits – fixed facts or variable depending on the season? *Folia Geobotanica*, *51*(2), 143–159. https://doi.org/10.1007/s12224-016-9250-3

Schielzeth, H. (2010). Simple means to improve the interpretability of regression coefficients. *Methods in Ecology and Evolution*, *1*(2), 103–113. https://doi.org/10.1111/j.2041-210X.2010.00012.x

Schielzeth, H., Dingemanse, N. J., Nakagawa, S., Westneat, D. F., Allegue, H., Teplitsky, C., Réale, D., Dochtermann, N. A., Garamszegi, L. Z., & Araya-Ajoy, Y. G. (2020). Robustness of linear mixed-effects models to violations of distributional assumptions. *Methods in Ecology and Evolution*, *11*(9), 1141–1152. https://doi.org/10.1111/2041-210X.13434

Schulz, M.-A., Yeo, B. T. T., Vogelstein, J. T., Mourao-Miranada, J., Kather, J. N., Kording, K., Richards, B., & Bzdok, D. (2020). Different scaling of linear models and deep learning in UKBiobank brain images versus machine-learning datasets. *Nature Communications*, *11*(1), 4238. https://doi.org/10.1038/s41467-020-18037-z

Venables, W. N., & Ripley, B. D. (2002). *Modern Applied Statistics with S* (Fourth). Springer. ISBN: 0-387-95457-0

Wei, T., & Simko, V. (2021). *R package 'corrplot': Visualization of a correlation matrix*. https://github.com/taiyun/corrplot

Wicke, S., Müller, K. F., DePamphilis, C. W., Quandt, D., Bellot, S., & Schneeweiss, G. M. (2016). Mechanistic model of evolutionary rate variation en route to a nonphotosynthetic lifestyle in plants. *Proceedings of the National Academy of Sciences of the United States of America*, *113*(32), 9045–9050. https://doi.org/10.1073/pnas.1607576113

Wickham, H., Averick, M., Bryan, J., Chang, W., McGowan, L. D., François, R., Grolemund, G., Hayes, A., Henry, L., Hester, J., Kuhn, M., Pedersen, T. L., Miller, E., Bache, S. M., Müller, K., Ooms, J., Robinson, D., Seidel, D. P., Spinu, V., … Yutani, H. (2019). Welcome to the tidyverse. *Journal of Open Source Software*, *4*(43), 1686. https://doi.org/10.21105/joss.01686

Zeileis, A., Fisher, J. C., Hornik, K., Ihaka, R., McWhite, C. D., Murrell, P., Stauffer, R., & Wilke, C. O. (2020). colorspace: A toolbox for manipulating and assessing colors and palettes. *Journal of Statistical Software*, *96*(1), 1–49. https://doi.org/10.18637/jss.v096.i01