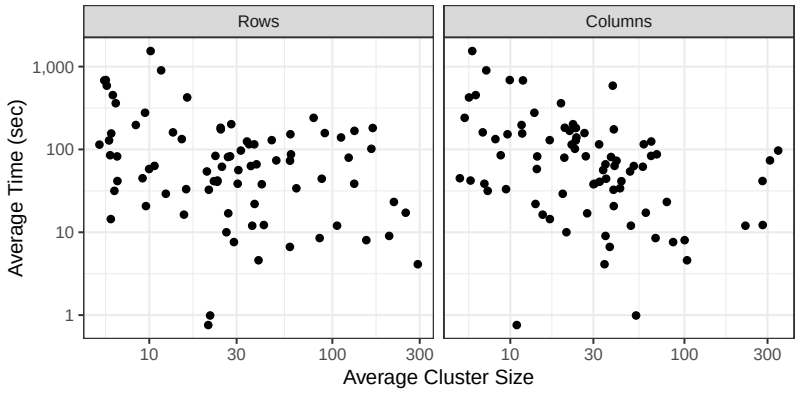


Figure 13: Relationship between average cluster sizes and elapsed time

Acknowledgments

This research was supported in part by Syngenta Seeds and by a Kingland Data Analytics Faculty Fellowship at Iowa State University.

Bibliography

S. Bergmann, J. Ihmels, and N. Barkai. Iterative signature algorithm for the analysis of large-scale gene expression data. *Physical Review E - Statistical Physics, Plasmas, Fluids, and Related Interdisciplinary Topics*, 2003. ISSN 1063651X. URL https://doi.org/10.1103/PhysRevE.67.031902. [p]

S. Busygin, O. Prokopyev, and P. M. Pardalos. Biclustering in data mining. *Computers and Operations Research*, 35(9), 2008. URL https://doi.org/10.1016/j.cor.2007.01.005. [p]

Y. Cheng and G. M. Church. Biclustering of expression data. *Proceedings International Conference on Intelligent Systems for Molecular Biology ; ISMB. International Conference on Intelligent Systems for Molecular Biology*, 8, 2000. [p]

D. T. Ewoud. *BiBitR: R Wrapper for Java Implementation of BiBit*, 2017. R package version 0.4.2. [p]

D. W. Goodall. A new similarity index based on probability. *Biometrics*, 22:882–907, 1966. [p]

D. Gusenleitner and A. Culhane. *iBBiG: Iterative Binary Biclustering of Genesets*, 2019. URL http://bcb.dfci.harvard.edu/~aedin/publications/. R package version 1.28.0. [p]

F. M. Harper and J. A. Konstan. The MovieLens datasets. *ACM Transactions on Interactive Intelligent Systems*, 2015. [p]

J. A. Hartigan. Direct clustering of a data matrix. *Journal of the American Statistical Association*, 67(337): 123–129, 1972. URL https://doi.org/10.1080/01621459.1972.10481214. [p]

L. Hubert and P. Arabie. Comparing partitions. *Journal of Classification*, 1985. ISSN 01764268. doi: 10.1007/BF01908075. [p]

S. Kaiser and F. Leisch. A toolbox for bicluster analysis in {R}. *Compstat 2008—Proceedings in Computational Statistics*, pages 201–208, 2008. [p]

T. Khamiakova. *superbiclust: Generating Robust Biclusters from a Bicluster Set (Ensemble Biclustering)*, 2014. URL https://CRAN.R-project.org/package=superbiclust. R package version 1.1. [p]

Y. Kluger, R. Basri, J. T. Chang, and M. Gerstein. Spectral biclustering of microarray data: Coclustering genes and conditions. *Genome Research*, 13(4):703–716, 2003. URL https://doi.org/10.1101/gr.648603. [p]

L. Lazzeroni and A. Owen. Plaid models for gene expression data. *Statistica Sinica*, 12:61–86, 2002. ISSN 10170405. URL https://doi.org/10.1017/CBO9781107415324.004. [p]

J. Li, J. Reisner, H. Pham, S. Olafsson, and S. Vardeman. Biclustering for missing data. *Information Sciences*, 510:304–316, 2020. URL https://doi.org/10.1016/j.ins.2019.09.047. [p]

M. Malosetti, J. M. Ribaut, and F. A. van Eeuwijk. The statistical analysis of multi-environment data: Modeling genotype-by-environment interaction and its genetic basis. *Frontiers in Physiology*, 4(44), 2013. URL https://doi.org/10.3389/fphys.2013.00044. [p]

A. Oghabian, S. Kilpinen, S. Hautaniemi, and E. Czeizler. Biclustering methods: Biological relevance and application in gene expression analysis. *PLOS ONE*, 9(3):1–10, 03 2014. doi: 10.1371/journal.pone.0090801. URL https://doi.org/10.1371/journal.pone.0090801. [p]

M. Rand. Objective criteria for the evaluation of methods clustering. *Journal of the American Statistical Association*, 66(336):846–850, 1971. URL https://doi.org/10.1080/01621459.1971.10482356. [p]

M. Sill and S. Kaiser. *s4vd: Biclustering via Sparse Singular Value Decomposition Incorporating Stability Selection*, 2015. URL https://CRAN.R-project.org/package=s4vd. R package version 1.1-1. [p]

K. M. Tan and D. M. Witten. Sparse Biclustering of Transposable Data. *Journal of Computational and Graphical Statistics*, 2014. ISSN 15372715. URL https://doi.org/10.1080/10618600.2013.852554. [p]

H. Wickham. *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York, 2009. ISBN 978-0-387-98140-6. URL http://ggplot2.org. [p]

H. Wickham. *tidyverse: Easily Install and Load 'Tidyverse' Packages.* 2016. URL https://cran.r-project.org/package=tidyverse. [p]

H. Wickham. *nycflights13: Flights that departed NYC in 2013*, 2017. URL https://CRAN.R-project.org/package=nycflights13. R package version 0.2.2. [p]

H. Wickham and L. Henry. *tidyr: Easily Tidy Data with 'spread()' and 'gather()' Functions*, 2019. URL https://CRAN.R-project.org/package=tidyr. R package version 0.8.3. [p]

Y. Zhang, J. Xie, J. Yang, A. Fennell, C. Zhang, and Q. Ma. QUBIC: a bioconductor package for qualitative biclustering analysis of gene co-expression data. *Bioinformatics*, 33(3):450–452, 2017. URL https://doi.org/10.1093/bioinformatics/btw635. [p]

John Reisner
Department of Statistics
Iowa State University
United States
johntreisner@gmail.com

Hieu Pham
Department of Industrial and Manufacturing Systems Engineering
Iowa State University
United States
htpham@iastate.edu

Sigurdur Olafsson
Department of Industrial and Manufacturing Systems Engineering
Iowa State University
United States
olafsson@iastate.edu

Stephen Vardeman
Department of Statistics
Department of Industrial and Manufacturing Systems Engineering
Iowa State University
United States
vardeman@iastate.edu

Jing Li
Boehringer Ingelheim Animal Health
St. Joseph, Missouri
United States
jingli2014cymail@gmail.com

auditor: an R Package for Model-Agnostic Visual Validation and Diagnostics

by Alicja Gosiewska and Przemysław Biecek

Abstract

Machine learning models have successfully been applied to challenges in applied in biology, medicine, finance, physics, and other fields. With modern software it is easy to train even a complex model that fits the training data and results in high accuracy on test set. However, problems often arise when models are confronted with the real-world data. This paper describes methodology and tools for model-agnostic auditing. It provides functinos for assessing and comparing the goodness of fit and performance of models. In addition, the package may be used for analysis of the similarity of residuals and for identification of outliers and influential observations. The examination is carried out by diagnostic scores and visual verification. The code presented in this paper are implemented in the **auditor** package. Its flexible and consistent grammar facilitates the validation models of a large class of models.

Introduction

Predictive modeling is a process using mathematical and computational methods to forecast outcomes. Many algorithms in this area have been developed and novel ones are continuously being proposed. Therefore, there are countless possible models to choose from and a lot of ways to train a new new complex model. A poorly- or over-fitted model usually will be of no use when confronted with future data. Its predictions will be misleading (Sheather, 2009) or harmful (O'Neil, 2016). That is why methods that support model diagnostics are important.

Diagnostics are often carried out only by checking model assumptions. However, they are often neglected for complex machine learning models and they may be used as if they were assumption-free. Still, there is a need to verify their quality. We strongly believe that a genuine diagnosis or an audit incorporates a broad approach to model exploration. The audit includes three objectives.

- **Objective 1:** Enrichment of information about model performance.
- **Objective 2:** Identification of outliers, influential and abnormal observations.
- **Objective 3:** Examination of other problems relating to a model by analyzing distributions of residuals, in particular, problems with bias, heteroscedasticity of variance and autocorrelation of residuals.

In this paper, we introduce the **auditor** package for R, which is a tool for diagnostics and visual verification. As it focuses on residuals[1] and does not require any additional model assumptions, most of the presented methods are model-agnostic. A consistent grammar across various tools reduces the amount of effort needed to create informative plots and makes the validation more convenient and available.

Diagnostic methods have been a subject of much research (Atkinson, 1985). Atkinson and Riani (2012) focus on graphical methods of diagnostics regression analysis. Liu et al. (2017) present an overview of interactive visual model validation. One of the most popular tools for verification are measures of the differences between the values predicted by a model and the observed values (Willmott et al., 1985). These tools include Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) (Hastie et al., 2001). Such measures are used for well-researched and easily interpretable linear model as well as for complex models such as random forests (Ho, 1995), gradient-boosted trees (Chen and Guestrin, 2016), or neural networks (Venables and Ripley, 2002).

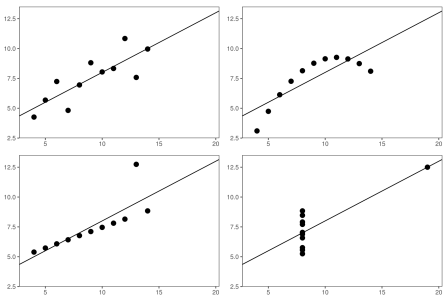


Figure 1: Anscombe Quartet data sets are identical when examined with he use of simple summary statistics. The difference is noticeable after plotting the data.

However, no matter which measure of model performance we use, it does not reflect all aspects of the model. For example, Breiman (2001) points out that a linear regression model validated only on the basis of R^2

[1]Residual of an observation is the difference between the observed value and the value predicted by a model.

may lead to many false conclusions. The best known example of this issue is the Anscombe Quartet (Anscombe, 1973). It contains four different data sets constructed to have nearly identical simple statistical properties such as mean, variance, correlation, etc. These measures directly correspond to the coefficients of the linear models. Therefore, by fitting a linear regression to the Anscombe Quartet we obtain four almost identical models (see Figure 1). However, residuals of these models are very different. The Anscombe Quartet is used to highlight that the numerical measures should be supplemented by graphical data visualizations.

The analysis of diagnostics is well-researched for linear and generalized linear models. The said analysis is typically done by extracting raw, studentized, deviance, or Pearson residuals and examining residual plots. Common problems with model fit and basic diagnostics methods are presented in Faraway (2002) and Harrell Jr. (2006)

Model validation may involve both checking the overall trend in residuals and looking at residual values of individual observations (Littell et al., 2007). Gałecki and Burzykowski (2013) discussed methods based on residuals for individual observation and groups of observations.

Diagnostics methods are commonly used for linear regression (Faraway, 2004). Complex models are treated as if they were assumption-free, which is why their diagnostics is often ignored. Considering the above, there is a need for more extensive methods and software dedicated for model auditing. Many of diagnostic tools, such as plots and statistics developed for linear models, are still useful for exploring machine learning models. Applying the same tools to all models facilitates their comparison.

The paper is organized as follows. Section 16.2 summarizes related work and state of the art. Section 16.3 contains an architecture of the **auditor** package. Section 16.4 provides the notation. Selected tools that help to validate models are presented in Section 16.5 and conclusions can be found in Section 16.6.

Related work

In this chapter, we overview common methods and tools for auditing and examining the validity of the models. There are several attempts to validate. They include diagnostics for predictor variables before and after model fit, methods dedicated to specific models, and model-agnostic approaches.

Data diagnostics before model fitting

The problem of data diagnostics is related to the Objective 2 presented in the Introduction, that is, the identification of problems with observations. There are several tools that address this issue. We review the most popular of them.

- One of the tools that supports the identification of errors in data is the **dataMaid** package (Petersen and Ekstrom, 2018). It creates a report that contains summaries and error checks for each variable in data. Package **lumberjack** (van der Loo, 2017) provides row-wise analysis. It allows for monitoring changes in data as they get processed. The **validatetools** (de Jonge and van der Loo, 2018) is a package for managing validation rules.
- The `datadist` function from **rms** package (Harrell Jr, 2018) computes distributional summaries for predictor variables. They include the overall range and certain quantiles for continuous variables, as well as distinct values for discrete variables. It automates the process of fitting and validating several models due to storing model summaries by `datadist` function.
- While above packages use pipeline approaches, there are also tools that focus on specific step of data diagnostic. The package **corrgram** (Wright, 2018) calculates a correlation of variables and displays corrgrams. Corrgrams (Friendly, 2002) are visualizations of correlation matrices, that help to identify the relationship between variables.

Diagnostics methods for linear models

As linear models have a very simple structure and do not require high computational power, they have been and still are used very frequently. Therefore, there are many tools that validate different aspects of linear models. Below, we overview the most widely known tools implemented in R packages.

- The **stats** package provides basic diagnostic plots for linear models. Function `plot` generates six types of charts for `"lm"` and `"glm"` objects, such as a plot of residuals against fitted values, a scale-location plot of $\sqrt{|residuals|}$ against fitted values and a normal quantile-quantile plot. These visual validation tools may be addressed to the Objective 3 of diagnostic, related to the

examination of model by analyzing the distribution of residuals. The other three plots, that include: a plot of Cook's distances, a plot of residuals against leverages, and a plot of Cook's distances against $\frac{leverage}{1-leverage}$ may be addressed to the identification of influential observations (Objective 1).

- Package **car** (Fox and Weisberg, 2011) extends the capabilities of **stats** by including more types of residuals, such as Pearson and deviance residuals. It is possible to plot against values of selected variables and to group residuals by levels of factor variables. What is more, **car** provides more diagnostic plots such as, among others, partial residual plot (`crPlot`), index plots of influence (`infIndexPlot`) and bubble plot of studentized residuals versus hat values (`influencePlot`). These plots allow for checking both the effect of observation and the distribution of residuals, what address to the Objective 2 and the Objective 3 respectively.
- A linear regression model is still one of the most popular tools for data analysis due to its simple structure. Therefore, there is a rich variety of methods for checking its assumptions, for example, the normality of residual distribution and the homoscedasticity of the variance.

  The package **nortest** (Gross and Ligges, 2015) provides five tests for normality: the Anderson-Darling (Anderson and Darling, 1952), the Cramer-von Mises (Cramer, 1928; Von Mises, 1928), the Kolmogorov-Smirnov (Stephens, 1974), the Pearson chi-square (F.R.S., 1900), and the Shapiro-Francia (Sanford Shapiro and S. Francia, 1972) tests. The **lmtest** package (Zeileis and Hothorn, 2002) also contains a collection of diagnostic tests: the Breusch-Pagan (Breusch and Pagan, 1979), the Goldfield-Quandt (Goldfeld and Quandt, 1965) and the Harrison-McCabe (Harrison and McCabe, 1979) tests for heteroscedasticity and the Harvey-Collier (Harvey and Collier, 1977), the Rainbow (Utts, 1982), and the RESET (Ramsey, 1969) tests for nonlinearity and misspecified functional form. A unified approach for examining, monitoring and dating structural changes in linear regression models is provided in **strucchange** package (Zeileis et al., 2002). It includes methods to fit, plot and test fluctuation processes and F-statistics. The **gvlma** implements the global procedure for testing the assumptions of the linear model (find more details in Peña and Slate (2006)).

  The Box-Cox power transformation introduced by Box and Cox (1964) is a way to transform the data to follow a normal distribution. For simple linear regression, it is often used to satisfy the assumptions of the model. Package **MASS** (Venables and Ripley, 2002) contains functions that compute and plot profile log-likelihoods for the parameter of the Box-Cox power transformation.
- The **broom** package (Robinson, 2018) provides summaries for about 30 classes of models. It produces results, such as coefficients and p-values for each variable, R^2, adjusted R^2, and residual standard error.

Other model-specific approaches

There are also several tools to generate validation plots for time series, principal component analysis, clustering, and others.

- Tang et al. (2016) introduced the **ggfortify** interface for visualizing many popular statistical results. Plots are generated with **ggplot2** (Wickham, 2009), what makes them easy to modify. With one function `autoplot` it is possible to generate validation plots for a wide range of models. It works for, among others, `lm`, `glm`, `ts`, `glmnet`, and `survfit` objects.

  The **autoplotly** (Tang, 2018) package is an extension of **ggfortify** and it provides functionalities that produce plots generated by **plotly** (Sievert et al., 2017). This allows for both modification and interaction with plots.

  However, **ggorftify** and **autoplotly** do not support some popular types of models, for instance, random forests from **randomForest** (Liaw and Wiener, 2002) and **ranger** (Wright and Ziegler, 2017) packages.
- The **hnp** package (Moral et al., 2017) provides half-normal plots with simulated envelopes. These charts evaluate the goodness of fit of any generalized linear model and its extensions. It is a graphical method for comparing two probability distributions by plotting their quantiles against each other. The package offers a possibility to extend the `hnp` for new model classes. However, this package provides only one tool for model diagnostic. In addition, plots are not based on **ggplot2**, what makes it difficult to modify them.

Model-agnostic approach

The tools presented above target specific model classes. The model-agnostic approach allows us to compare different models.

- The **DALEX** (Descriptive mAchine Learning EXplanations) (Biecek, 2018) is a methodology for exploration of black-box models. Main functionalities focus on understanding or proving how the input variables impact on final predictions. There are also two simple diagnostics: reversed empirical cumulative distribution function for absolute values of residuals and box plot of absolute values of residuals. As methods in the **DALEX** are model-agnostic, they allow for comparison of two or more models.
- The package **iml** (Molnar et al., 2018) also contains methods for structure-agnostic exploration of model. For example, a measure of a feature's importance by calculating the change of the model performance after permuting values of a variable.

Model-agnostic audit

In this paper, we present the **auditor** package for R, which fills out the part of model-agnostic validation. As it expands methods used for linear regression, it may be used to verify any predictive model.

Package Architecture

The **auditor** package works for any predictive model which returns a numeric value. It offers a consistent grammar of model validation, what is an efficient and convenient way to generate plots and diagnostic scores. A diagnostic score is a number that evaluates one of the properties of a model. That might be, for example, an accuracy of model, an independence of residuals or an influence of observation.

Figure 2 presents the architecture of the package. The **auditor** provides 2 pipelines for model validation. First of them consists of two steps. Function `audit` wraps up the model with meta-data, then the result is passed to the `plot` or `score` function. Second pipeline includes an additional step, which consists of calling one of the functions that generate computations for plots and scores. These functions are: `modelResiduals`, `modelEvaluation`, `modelFit`, `modelPerformance`, and `observationInfluence`. Further, we call them computational functions. Results of these functions are tidy data frames (Wickham, 2014).

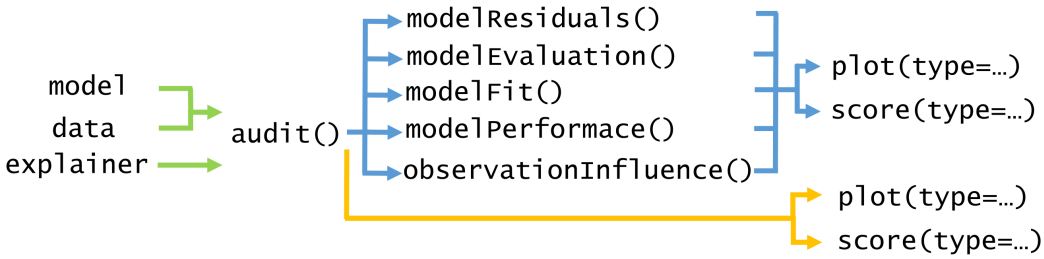


Figure 2: Architecture of the **auditor**. The blue color indicates the first pipeline, while orange indicates the second. Function `audit` takes model and data or `"explainer"` object created with the **DALEX** package.

Both pipelines for model audit are compared below.

1. **model %>% audit() %>% computational function %>% plot(type=...)**
   We recommend this pipeline. Function `audit` wraps up a model with meta-data used for modeling and creates a `"modelAudit"` object. One of the computational functions takes `"modelAudit"` object and computes the results of validation. Then, outputs may be printed or passed to functions `score` and `plot` with defined type. We describe types of plots in Chapter 16.5. This approach requires one additional function within the pipeline. However, once created output of the computational function contains all necessary calculations for related plots. Therefore, generating multiple plots is fast.
2. **model %>% audit() %>% plot(type=...)**
   This pipeline is shorter than the previous one. The only difference is that it does not include computational function. Calculations are carried out every time a generic `plot` function is called. Omitting one step might be convenient for ad-hoc model analyses.

Implemented types of plots are presented in Table 1. Scores are presented in Table 2. All plots are generated with **ggplot2**, what provides a convenient way to modify and combine plots.

Plot	Function	plot(type = ...)	Reg.	Class.
Autocorrelation Function	`modelResiduals`	`"ACF"`	+	+
Autocorrelation	`modelResiduals`	`"Autocorrelation"`	+	+
Cooks's Distances	`observationInfluence`	`"CooksDistance"`	+	+
Half-Normal	`modelFit`	`"HalfNormal"`	+	+
LIFT Chart	`modelEvaluation`	`"LIFT"`		+
Model Correlation	`modelResiduals`	`"ModelCorrelation"`	+	+
Model PCA	`modelResiduals`	`"ModelPCA"`	+	+
Model Ranking	`modelPerformance`	`"ModelRanking"`	+	+
Predicted Response	`modelPerformance`	`"ModelPerformance"`	+	+
REC Curve	`modelResiduals`	`"REC"`	+	+
Residuals	`modelResiduals`	`"Residual"`	+	+
Residual Boxplot	`modelResiduals`	`"ResidualBoxplot"`	+	+
Residual Density	`modelResiduals`	`"ResidualDensity"`	+	+
ROC Curve	`modelEvaluation`	`"ROC"`		+
RROC Curve	`modelResiduals`	`"RROC"`	+	+
Scale-Location	`modelResiduals`	`"ScaleLocation"`	+	+
Two-sided ECDF	`modelResiduals`	`"TwoSidedECDF"`	+	+

Table 1: Columns contain respectively: name of the plot, name of the computational function, value for `type` parameter of the function `plot`, indications whether the plot can be applied to regression and classification tasks.

Score	Function	score(type = ...)	Reg.	Class.
Cook's Distance	`observationInfluence`	`"CooksDistance"`	+	+
Durbin-Watson	`modelResiduals`	`"DW"`	+	+
Half-Normal	`modelFit`	`"HalfNormal"`	+	+
Mean Absolute Error	`modelResiduals`	`"MAE"`	+	+
Mean Squared Error	`modelResiduals`	`"MSE"`	+	+
Area Over the REC	`modelResiduals`	`"REC"`	+	+
Root Mean Squared Error	`modelResiduals`	`"RMSE"`	+	+
Area Under the ROC	`modelEvaluation`	`"ROC"`		+
Area Over the RROC	`modelResiduals`	`"RROC"`	+	+
Runs	`modelResiduals`	`"Runs"`	+	+
Peak	`modelResiduals`	`"Peak"`	+	+

Table 2: Columns contain respectively: name of a score, name of a computational function, value for `type` parameter of function the `score`, indications whether the score can be applied to regression and classification tasks.

Notation

Let us use the following notation: $x_i = (x_i^{(1)}, x_i^{(2)}, ..., x_i^{(p)}) \in \mathcal{X} \subset \mathcal{R}^p$ is a vector in space $\mathcal{X}$, $y_i \in \mathcal{R}$ is an observed response associated with x_i. A single observation we denote as a pair (y_i, x_i) and n is the number of observations.

Let us denote a model as a function $f : \mathcal{X} \to \mathcal{R}$. Predictions of the model f for particular observation we shall denote as

$$f(x_i) = \hat{y}_i. \tag{1}$$

The row residual, or simply the residual, is the difference between the observed value y_i and the predicted value $\hat{y}_i$. We shall denote residual of particular observation as

$$r_i = y_i - \hat{y}_i. \tag{2}$$

Illustrations

Diagnostics allows for evaluation of different properties of a model. They may be related to the following questions: Which model has better performance? Does the model fit data? Which observations are abnormal? These questions are directly related to the diagnostics objectives described in the Introduction. First of them refers to the evaluation of a model performance, which was proposed as the Objective 1. The second question concerns the examination of residuals distribution (Objective 3). The last one refers to outliers and influential observations (Objective 2).

In this Section we illustrate chosen validation tools that allow for exploration of the above issues. To demonstrate applications of the **auditor**, we use the data set `apartments` available in the **DALEX** package. First, we fit two models: simple linear regression and random forest.

```
library("auditor")
library("DALEX")
library("randomForest")

lm_model <- lm(m2.price ~ ., data = apartments)
set.seed(59)
rf_model <- randomForest(m2.price ~ ., data = apartments)
```

The next step creates `"modelAudit"` objects related to these two models.

```
lm_audit <- audit(lm_model, label = "lm",
              data = apartmentsTest, y = apartmentsTest$m2.price)
rf_audit <- audit(rf_model, label = "rf",
              data = apartmentsTest, y = apartmentsTest$m2.price)
```

Below, we create objects of class `"modelResidual"`, which are needed to generate plots. Parameter `variable` determines the order of residuals in the plot. When the `variable` argument is set to `"Fitted values"` residuals are sorted by values of predicted responses. Entering a name of a variable `"m2.price"` implies that residuals will be in order of this variable.

```
lm_res_fitted <- modelResiduals(lm_audit, variable = "Fitted values")
rf_res_fitted <- modelResiduals(rf_audit, variable = "Fitted values")

lm_res_observed <- modelResiduals(lm_audit, variable = "m2.price")
rf_res_observed <- modelResiduals(rf_audit, variable = "m2.price")
```

Model Ranking Plot

In this subsection, we propose a Model Ranking plot which compares models performance across multiple measures (see Figure 3). The implemented measures are listed in Table 2 in Chapter 16.3. The descriptions of all scores are in (Gosiewska and Biecek, 2018).

Model Ranking Radar plot consists of two parts. On the left side there is a radar plot. Colors correspond to models, edges to values of scores. Score values are inverted and rescaled to $[0, 1]$.

Let us use the following notation: $m_i \in \mathcal{M}$ is a model in a finite set of models $\mathcal{M}$, where $|\mathcal{M}| = k$, $score : \mathcal{M} \to \mathbb{R}$ is a loss function for the model under consideration. Higher values mean worse model performance. The $score(m_i)$ is a performance of model m_i.

Definition 16.5.1. We define the inverted score of model m_i as

$$invscore(m_i) = \frac{1}{score(m_i)} \min_{j=1...k} score(m_j). \tag{3}$$

Models with the larger *invscore* are closer to the centre. Therefore, the best model is located the farthest from the center of the plot. On the right side of the plot is a table with results of scoring. The third column contains scores scaled to one of the models.

Let $m_l \in \mathcal{M}$ where $l \in \{1, 2, ..., k\}$ be a model to which we scale.

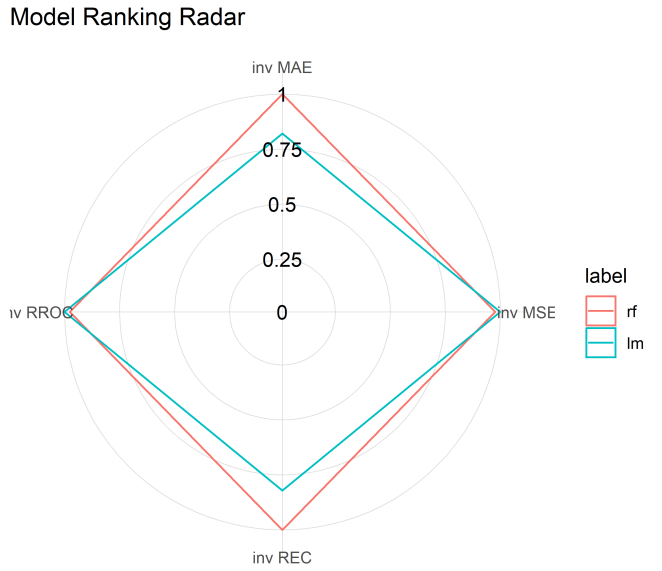
Definition 16.5.2. We define the scaled score of model m_i to model m_l as

$$scaled_l(m_i) = \frac{score(m_l)}{score(m_i)}. \tag{4}$$

As values of $scaled_l(m_l)$ are always between 0 and 1, comparison of models is easy, regardless of the ranges of scores.

The plot below is generated by `plot` function with parameter `type = "ModelRanking"` or by function `plotModelRanking`. The scores included in the plot may be specified by `scores` parameter.

```
rf_mp <- modelPerformance(rf_audit)
lm_mp <- modelPerformance(lm_audit)
plot(rf_mp, lm_mp, type = "ModelRanking")
```



name	label	values	scaled
MAE	rf	2.16e+02	1.000
MAE	lm	2.63e+02	0.821
MSE	rf	8.21e+04	1.000
MSE	lm	8.01e+04	1.025
REC	rf	2.16e+02	1.000
REC	lm	2.63e+02	0.821
RROC	rf	3.32e+12	1.000
RROC	lm	3.24e+12	1.024

Figure 3: Model Ranking Plot. Random forest (red) has better performance in aspect of MAE and REC scores, while linear model (blue) is better in aspect of MSE and RROC scores.

REC Curve Plot

Regression Error Characteristic (REC) curve (see Figure 4) is a generalization of Receiver Operating Characteristic (ROC) curve for binary classification (Swets, 1988).

REC curve estimates the Cumulative Distribution Function of the error. On the x axis of the plot there is an error tolerance. On the y axis there is an accuracy at the given tolerance level. Bi and Bennett (2003) define the accuracy at tolerance ϵ as a percentage of observations predicted within the tolerance ϵ. In other words, residuals larger than ϵ are considered as errors.

Let us consider pairs (y_i, x_i) introduced in the beginning of Chapter 16.5. Bi and Bennett (2003) define an accuracy as follows.

Definition 16.5.3. An accuracy at tolerance level ϵ is given by

$$acc(\epsilon) = \frac{|\{(x,y) : loss(f(x_i), y_i) \leq \epsilon, i = 1, ..., n\}|}{n}. \tag{5}$$

REC Curves implemented in the **auditor** are plotted for a special case of Definition 16.5.3 where the loss is defined as

$$loss(f(x_i), y_i) = |f(x_i) - y_i| = |r_i|. \tag{6}$$

The shape of the curve illustrates the behavior of errors. The quality of the model can be evaluated and compared for different tolerance levels. The stable growth of the accuracy does not indicate any problems with the model. A small increase of accuracy near 0 and the areas where the growth is fast signalize bias of the model predictions.

The plot below is generated by `plot` function with parameter `type = "REC"` or by `plotREC` function.

```
plot(rf_res_fitted, lm_res_fitted, type = "REC")
```

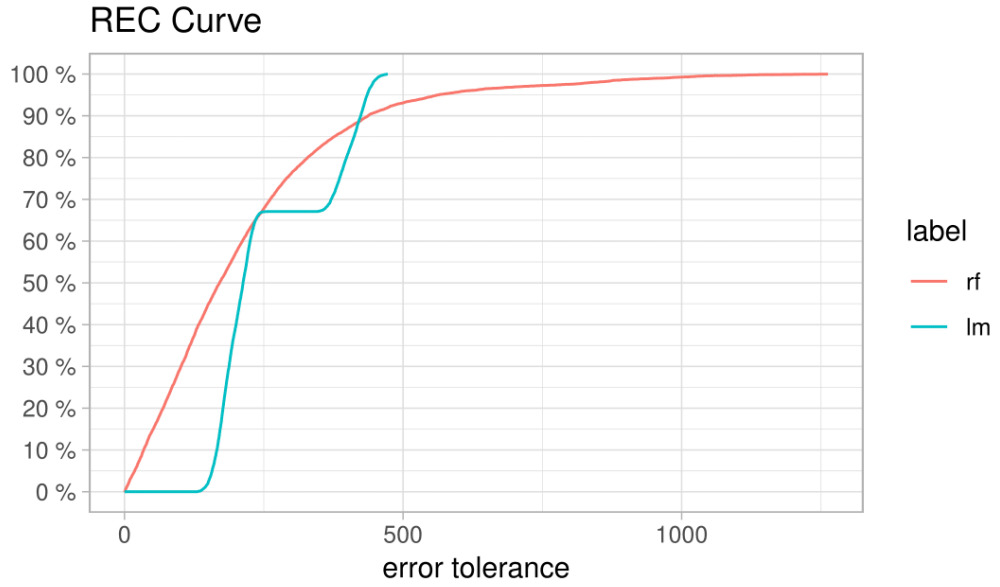


Figure 4: REC curve. Curve for linear model (blue) suggests that the model is biased. It displays poor accuracy when the tolerance ϵ is small. However, once ϵ exceeds the error tolerance 130, accuracy rapidly increases. The random forest (red) has a stable increase of accuracy when compared to the linear model. However, there is s fraction of large residuals.

As often it is difficult to compare models on the plot, there is an REC score implemented in the **auditor**. This score is the Area Over the REC Curve (AOC), which is a biased estimate of the expected error for a regression model. As Bi and Bennett (2003) proved, AOC provides a measure of the overall performance of regression model.

Scores may be obtained by `score` function with `type = "REC"` or `scoreREC` function.

```
scoreREC(lm_res_fitted)
scoreREC(rf_res_fitted)
```

Residual Boxplot Plot

Residual boxplot shows the distribution of the absolute values of residuals r_i. They may be used for analysis and comparison of residuals. Example plots are presented in Figure 5. Boxplots (Tukey, 1977) usually consist of five components. The box itself corresponds to the first quartile, median, and third quartile. The whiskers extend to the smallest and largest values, no further than 1.5 of Interquartile Range (IQR) from the first and third quartile respectively. Residual boxplots consists of a sixth component, namely a red dot which stands for Root Mean Square Error (RMSE). In case of an appropriate model, most of the residuals should lay near zero. A large spread of values indicates problems with a model.

The plot presented below is generated by `plotResidualBoxplot` or by `plot` function with parameter `type = 'ResidualBoxplot'` function.

```
plot(lm_res_fitted, rf_res_fitted, type = "ResidualBoxplot")
```

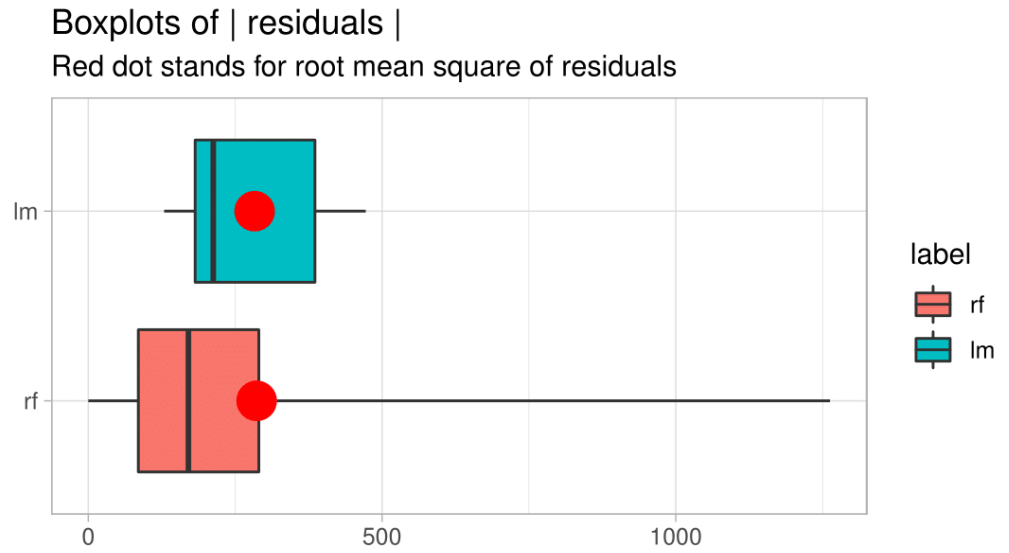


Figure 5: Boxplots of absolute values of residuals. Dots are in similar places, hence RMSE for both models is almost identical. However, the distribution of residuals of these two models is different. For the linear model (blue), most of the residuals are around the average. For the random forest (red), most residuals are small. Nevertheless, there is also a fraction of large residuals.

Residual Density Plot

Residual Density plot detects the incorrect behavior of residuals. An example is presented in Figure 6. On the plot, there are estimated densities of residuals. For some models, the expected shape of density derives from the model assumptions. For example, simple linear model residuals should be normally distributed. However, even if the model does not have an assumption about the distribution of residuals, such a plot may be informative. If most of the residuals are not concentrated around zero, it is likely that the model predictions are biased. Values of errors are displayed as marks along the x axis. That makes it possible to ascertain whether there are individual observations or groups of observations with residuals significantly larger than others.

The plot below is generated by `plotResidualDensity` function or by `plot` function with parameter `type = "ResidualDensity"`.

```
plot(rf_res_observed, lm_res_observed, type = "ResidualDensity")
```

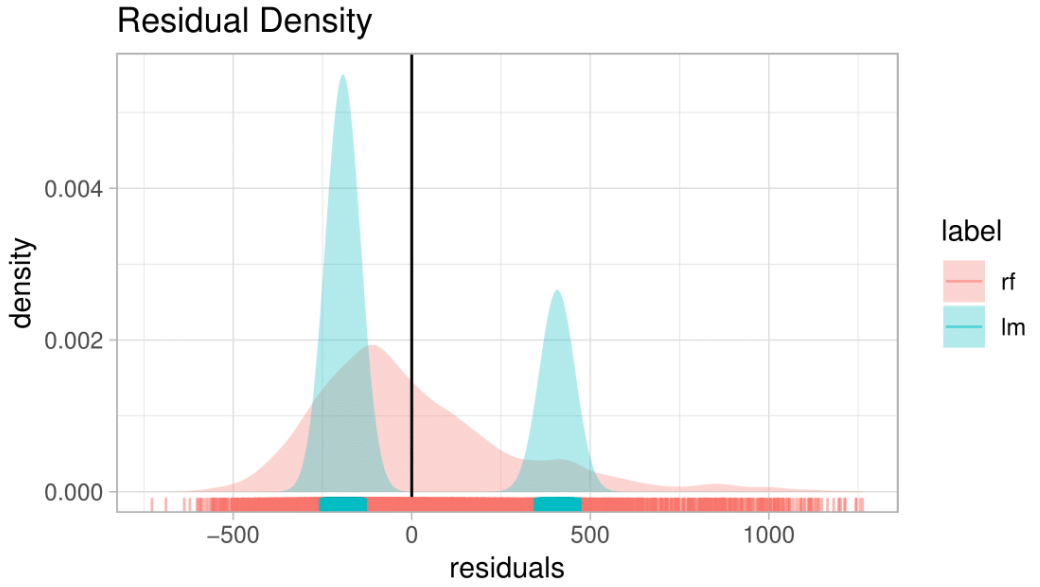


Figure 6: Residual Density Plot. The density of residuals for the linear model (blue) forms two peaks. There are no residuals with values around zero. Residuals do not follow the normal distribution, what is one of the assumptions of the simple linear regression. There is an asymmetry of residuals generated by random forest (red).

Two-sided ECDF Plot

Two-sided ECDF plot (see Figure 7) shows an Empirical Cumulative Distribution Functions (ECDF) for positive and negative values of residuals separately.

Let $x_1, ..., x_n$ be a random sample from a cumulative distribution function $F(t)$. The following definition comes from van der Vaart (2000).

Definition 16.5.4. The empirical cumulative distribution function is given by

$$F_n(t) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}\{x_i \leq t\}. \tag{7}$$

Empirical cumulative distribution function presents a fraction of observations that are less than or equal to t. It is an estimator for the cumulative distribution function $F(t)$.

On the positive side of the x-axis, there is the ECDF of positive values of residuals. On the negative side, there is a transformation of ECDF:

$$F_{rev}(t) = 1 - F(t). \tag{8}$$

Let n_N and n_P be numbers of negative and positive values of residuals respectively. Negative part of the plot is normalized by multiplying it by the ratio of the n_N over $n_N + n_P$. Similarly, positive part is normalized by multiplying it by the ratio of the n_P over $n_N + n_P$. Due to the applied scale, the ends of the curves add up to 100% in total. The plot shows the distribution of residuals divided into groups with positive and negative values. It helps to identify the asymmetry of the residuals. Points represent individual error values, what makes it possible to identify 'outliers'.

The plot below is generated by `plotTwoSidedECDF` function or by `plot` function with parameter `type = "TwoSidedECDF"`.

```
plot(rf_res_fitted, lm_res_fitted, type = "TwoSidedECDF")
```

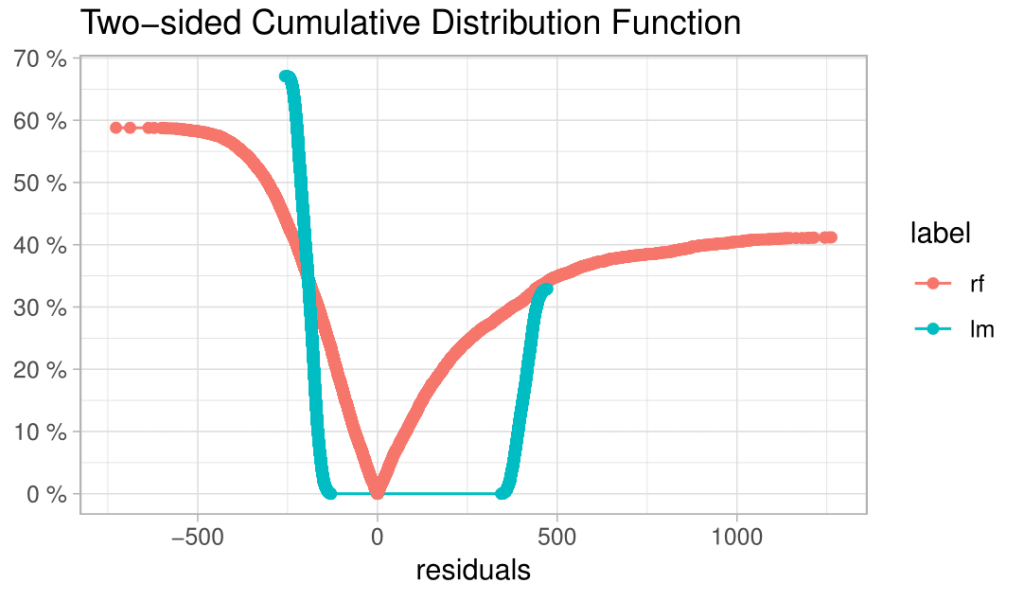


Figure 7: Two-sided ECDF plot. The plot shows that majority of residuals for the random forest (red) is smaller than residuals for the linear model (blue). However, random forest has also fractions of large residuals.

Conclusion and future work

In this article, we presented the **auditor** package and selected diagnostic scores and plots. We discussed the existing methods of model validation and proposed new visual approaches. We also specified three objectives of model audit (see Section 16.1), proposed relevant verification tools, and demonstrated their usage. Model Ranking Plot and REC Curve enrich the information about model performance (Objective 1). Residual Boxplot, Residual Density, and Two-Sided ECDF Plots expand the knowledge about the distribution of residuals (Objective 3). What is more, the latter two tools allow for identification of outliers (Objective 2). Finally, we proposed two new plots, the Model Ranking Plot and the Two-Sided ECDF Plot.

We implemented all the presented scores and plots in the **auditor** package for R. The included functions are based on a uniform grammar introduced in Figure 16.3. Documentation and examples are available at `https://mi2datalab.github.io/auditor/`. The stable version of the package is on CRAN, the development version is on GitHub (`https://github.com/MI2DataLab/auditor`). A more detailed description of methodology is available in the extended version of this paper on arXiv: `https://arxiv.org/abs/1809.07763` (Gosiewska and Biecek, 2018).

There are many potential areas for future work that we would like to explore, including more extensions of model-specific diagnostics to model-agnostic methods and residual-based methods for investigating interactions. Another potential aim would be to develop methods for local audit based on the diagnostics of a model around a single observation or a group of observations.

Acknowledgements

We would like to acknowledge and thank Aleksandra Grudziąż and Mateusz Staniak for valuable discussions. Also, we wish to thank Dr. Rafael De Andrade Moral for his assistance and help related to the **hnp** package.

The work was supported by NCN Opus grant 2016/21/B/ST6/02176.

Bibliography

T. W. Anderson and D. A. Darling. Asymptotic theory of certain goodness of fit criteria based on stochastic processes. *Ann. Math. Statist.*, 23(2):193–212, 1952. URL `https://doi.org/10.1214/aoms/1177729437`. [p]

F. J. Anscombe. Graphs in statistical analysis. *The American Statistician*, 27(1):17–21, 1973. URL https://doi.org/10.1080/00031305.1973.10478966. [p]

A. Atkinson and M. Riani. *Robust Diagnostic Regression Analysis*. Springer Series in Statistics. Springer-Verlag, 2012. ISBN 9781461211600. URL https://books.google.pl/books?id=sZ3SBwAAQBAJ. [p]

A. C. Atkinson. *Plots, Transformations, and Regression: An Introduction to Graphical Methods of Diagnostic Regression Analysis*. Oxford statistical science series. Clarendon Press, 1985. URL https://books.google.pl/books?id=oFjgnQEACAAJ. [p]

J. Bi and K. P. Bennett. Regression error characteristic curves. In *ICML*, 2003. [p]

P. Biecek. DALEX: Explainers for Complex Predictive Models. *ArXiv e-prints*, 2018. [p]

G. E. P. Box and D. R. Cox. An analysis of transformations. *Journal of the Royal Statistical Society B*, pages 211–252, 1964. [p]

L. Breiman. Statistical modeling: The two cultures (with comments and a rejoinder by the author). *Statist. Sci.*, 16(3):199–231, 2001. URL https://doi.org/10.1214/ss/1009213726. [p]

T. S. Breusch and A. R. Pagan. A simple test for heteroscedasticity and random coefficient variation. *Econometrica*, 47(5):1287–1294, 1979. ISSN 00129682, 14680262. URL http://www.jstor.org/stable/1911963. [p]

T. Chen and C. Guestrin. Xgboost: A scalable tree boosting system. *CoRR*, abs/1603.02754, 2016. URL http://arxiv.org/abs/1603.02754. [p]

H. Cramer. On the composition of elementary errors: Second paper: Statistical applications. *Scandinavian Actuarial Journal*, 1928(1):141–180, 1928. [p]

E. de Jonge and M. van der Loo. *Validatetools: Checking and Simplifying Validation Rule Sets*, 2018. URL https://CRAN.R-project.org/package=validatetools. R package version 0.4.3. [p]

J. J. Faraway. *Practical Regression and Anova Using R*. University of Bath, 2002. URL https://books.google.pl/books?id=UjhBnwEACAAJ. [p]

J. J. Faraway. *Linear Models with R*. Chapman & Hall/CRC Texts in Statistical Science. Taylor & Francis, 2004. ISBN 9780203507278. URL https://books.google.pl/books?id=fvenzpofkagC. [p]

J. Fox and S. Weisberg. *An R Companion to Applied Regression*. Sage, Thousand Oaks CA, 2nd edition, 2011. URL http://socserv.socsci.mcmaster.ca/jfox/Books/Companion. [p]

M. Friendly. Corrgrams: Exploratory displays for correlation matrices. *The American Statistician*, 56(4): 316–324, 2002. [p]

K. P. F.R.S. *X. On the Criterion That a Given System of Deviations from the Probable in the Case of a Correlated System of Variables Is Such That It Can Be Reasonably Supposed to Have Arisen from Random Sampling*, volume 50. Taylor & Francis, 1900. URL https://doi.org/10.1080/14786440009463897. [p]

A. Gałecki and T. Burzykowski. *Linear Mixed-Effects Models Using R: A Step-by-Step Approach*. Springer Texts in Statistics. Springer-Verlag, 2013. ISBN 9781461439004. URL https://books.google.pl/books?id=rbk_AAAAQBAJ. [p]

S. M. Goldfeld and R. E. Quandt. Some tests for homoscedasticity. *Journal of the American Statistical Association*, 60(310):539–547, 1965. URL https://doi.org/10.1080/01621459.1965.10480811. [p]

A. Gosiewska and P. Biecek. auditor: An R Package for Model-Agnostic Visual Validation and Diagnostic. *ArXiv e-prints*, 2018. [p]

J. Gross and U. Ligges. *Nortest: Tests for Normality*, 2015. URL https://CRAN.R-project.org/package=nortest. R package version 1.0-4. [p]

F. E. Harrell Jr. *Regression Modeling Strategies*. Springer-Verlag, Berlin, Heidelberg, 2006. ISBN 0387952322. [p]

F. E. Harrell Jr. *Rms: Regression Modeling Strategies*, 2018. URL https://CRAN.R-project.org/package=rms. R package version 5.1-2. [p]

M. J. Harrison and B. P. M. McCabe. A test for heteroscedasticity based on ordinary least squares residuals. *Journal of the American Statistical Association*, 74(366):494–499, 1979. ISSN 01621459. URL http://www.jstor.org/stable/2286361. [p]