edarf: **E**xploratory **D**ata **A**nalysis using **R**andom Forests

Zachary M. Jones[1] and Fridolin J. Linder[1]

[1]Pennsylvania State University

6 October 2016

Paper DOI: http://dx.doi.org/10.21105/joss.00092
Software Repository: http://github.com/zmjones/edarf/
Software Archive: https://dx.doi.org/10.5281/zenodo.162238

Summary

This package contains functions useful for exploratory data analysis using random forests, which can be fit using the randomForest, randomForestSRC, or party packages (Liaw and Wiener 2002; Ishwaran and Kogalur 2013; Hothorn, Hornik, and Zeileis 2006). These functions can compute the partial dependence of covariates (individually or in combination) on the fitted forests' predictions, the permutation importance of covariates, as well as the distance between data points according to the fitted model.

Random forests are an attractive method for social scientists. Random forests have only a few important tuning parameters and can be adapted to do classification, regression, and clustering. Many research tasks require interpretable models, and, although methods for interpreting random forests exist, in the most popular packages for fitting random forests these methods are available inconsistently. For example although there are methods for computing permutation importance in the `randomForest`, `randomForestSRC` and `party` packages, only `randomForest` can compute local importance. None of the packages can compute permutation importance for groups of covariates. Similarly partial dependence is only implemented in `randomForest`, and it has limited functionality compared to the functions provided herein. This software has been used in Jones and Linder (2015); Jones and Lupu (2016).

Partial dependence, as described by Friedman (2001), estimates the marginal relationship between a subset of the covariates and the model's predictions by averaging over the marginal distribution of the compliment of this subset of the covariates. This approximation allows the display of the relationship between this subset of the covariates and the model's predictions even when there are many covariates which may interact. This functionality works with models fit by any of the aforementioned packages to any of the supported types of outcome variables. `partial_dependence` can be parallelized and also contains a number of additional parameters which allow the user to control this approximation. There is an associated plot function `plot_pd` which constructs plots for a wide variety of possible outputs from `partial_dependence` (e.g., when pairs of covariates are considered jointly, when each covariate is considered separately, when the outcome variable is categorical, etc).

Permutation importance estimates the importance of a covariate by randomly shuffling its values, breaking any dependence between said covariate and the outcome, and then computing the difference between the predictions made by the model with that covariate shuffled and the predictions made when the covariate was not shuffled. If the covariate was useful in generating predictions then the prediction errors will increase in expectation when the covariate is shuffled, whereas no such increase can be expected when the covariate has no influence. Although all three of the random forest packages provide at least one method of assessing variable importance, `variable_importance` provides a consistent way to compute permutation importance

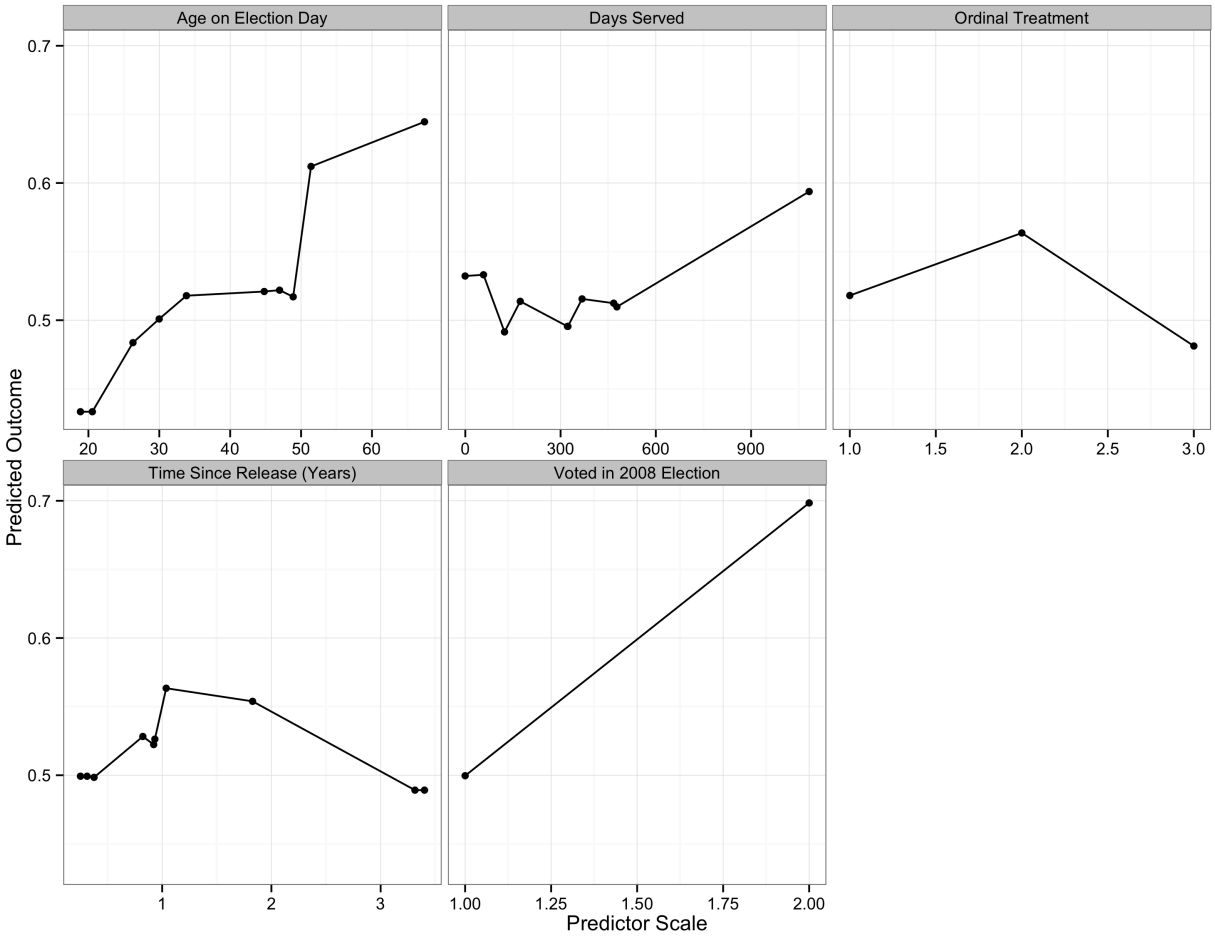

Figure 1: The partial dependence of several covariates (each considered separately) on the probability that a convict voted in the 2012 presidential election, given that they had registered to do so. Data is from Gerber et al. (2015).

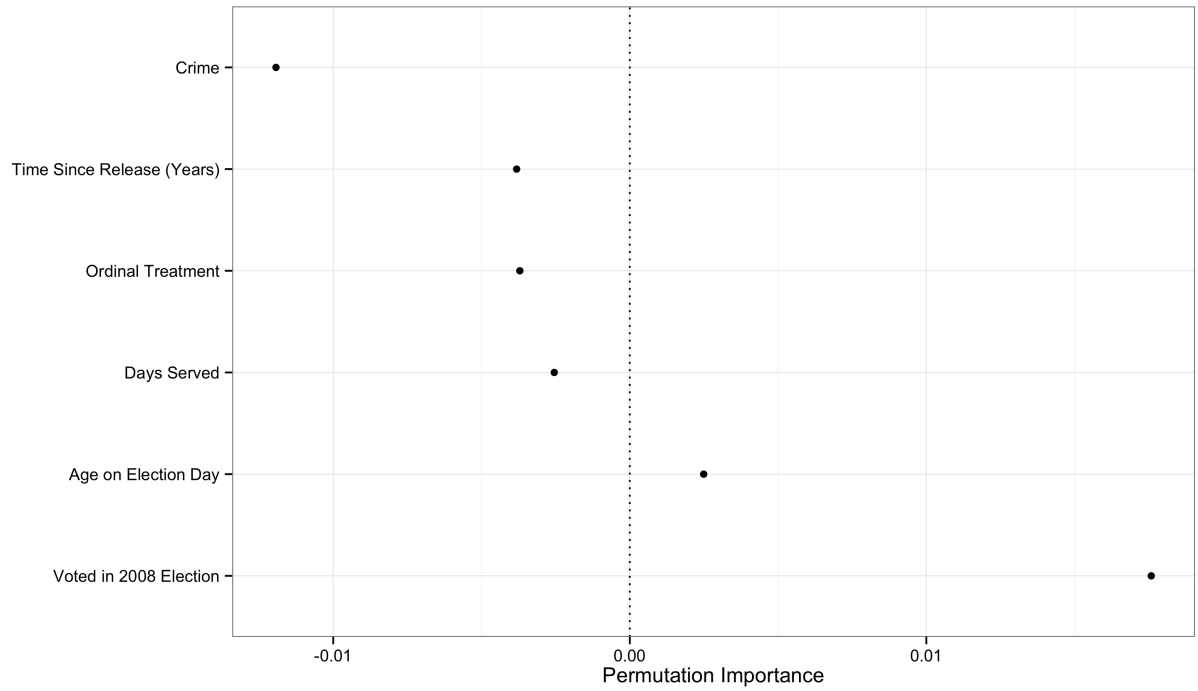

Figure 2: The permutation importance of the same covariates.

across all packages. `variable_importance` can also compute local importance. Rather than computing the average difference in prediction errors between the permuted and unpermuted data across all the training data (giving one number for each covariate), local importance computes the average change in the prediction error for each observation. This can be examined directly, or, in the case of a categorical outcome variable, can be aggregated to each class level. In the case of a continuous outcome variable the change in the prediction errors can be smoothed at different values of the outcome variable, giving a similar display. Lastly, `variable_importance` can operate on multiple variables simultaneously, giving the joint permutation importance of a set of covariates. This can be used to detect interactions by comparing the permutation importance of, for example, two covariates, by computing their joint permutation importance and comparing that to the sum of their individual permutation importance estimates. `plot_imp` provides a visualizations for all possible outputs.

Due to the tree-structure of a random forest, there is a natural way to define a distance between data points: the proportion of times that the data points were in the same terminal node. This is called "proximity" and can be used do model-based clustering when the random forest was unsupervised, and can be used to visualize the model in the supervised case. Making generic the compuation of the proxmity matrix would require a consistent API for accessing information in the individual trees in the random forest, which does not exist, however, `extract_proximity` can extract a proximity matrix computed by one of the supported packages. These matrices are too high dimensional to be visualized directly, so `plot_prox` supports the visualization of two of the principal components of this matrix, as estimated by `prcomp` (included in the base distribution of `R`). `plot_prox` provides arguments which additionally allow the user to change the color, shape, and size of points according to auxillary information, such as the observed class label for categorical outcomes, or a covariates, which may aid the aforementioned visualization tasks.

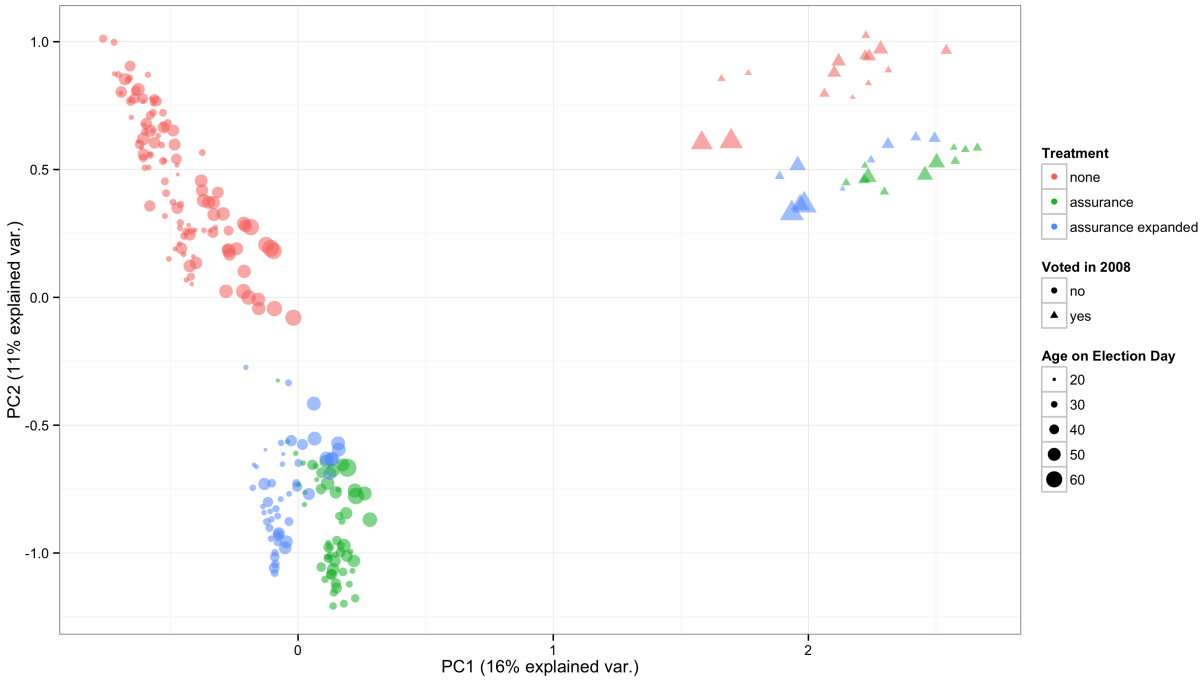

Figure 3: The proximity of the training data, colored according to the encouragement condition implemented by Gerber et al. (2015), with the shape of the points mapped to whether the individual had voted in the 2008 election and the size of the point mapped to the individual's age.

References

Friedman, Jerome H. 2001. "Greedy Function Approximation: A Gradient Boosting Machine." *The Annals of Statistics* 29 (5). The Institute of Mathematical Statistics: 1189–1232. doi:10.1214/aos/1013203451.

Gerber, Alan S., Gregory A. Huber, Marc Meredith, Daniel R. Biggers, and David J. Hendry. 2015. "Can Incarcerated Felons Be (Re)integrated into the Political System? Results from a Field Experiment." *American Journal of Political Science* 59 (4): 912–26. doi:10.1111/ajps.12166.

Hothorn, Torsten, Kurt Hornik, and Achim Zeileis. 2006. "Unbiased Recursive Partitioning: A Conditional Inference Framework." *Journal of Computational and Graphical Statistics* 15 (3). Taylor & Francis: 651–74. doi:10.1198/106186006X133933.

Ishwaran, H, and U Kogalur. 2013. "Random Forests for Survival, Regression and Classification (Rf-Src)." *URL Http://Cran.r-Project.org/Web/Packages/RandomForestSRC/. R Package Version* 1.

Jones, Zachary M., and Fridolin Linder. 2015. "Exploratory Data Analysis Using Random Forests." In *Proceedings of the 73rd Annual Mpsa Conference*. http://zmjones.com/static/papers/rfss_manuscript.pdf.

Jones, Zachary M., and Yonatan Lupu. 2016. "Is There More Violence in the Middle?"

Liaw, Andy, and Matthew Wiener. 2002. "Classification and Regression by RandomForest." *R News* 2 (3): 18–22. http://CRAN.R-project.org/doc/Rnews/.