



## Consistent and Clear Reporting of Results from Diverse Modeling Techniques: The A3 Method

Scott Fortmann-Roe

University of California, Berkeley

---

### Abstract

The measurement and reporting of model error is of basic importance when constructing models. Here, a general method and an R package, **A3**, are presented to support the assessment and communication of the quality of a model fit along with metrics of variable importance. The presented method is accurate, robust, and adaptable to a wide range of predictive modeling algorithms. The method is described along with case studies and a usage guide. It is shown how the method can be used to obtain more accurate models for prediction and how this may simultaneously lead to altered inferences and conclusions about the impact of potential drivers within a system.

*Keywords:* goodness of fit, variable importance, model selection.

---

## 1. Introduction

A range of metrics have been developed to assess model results for prediction or other inferences. These metrics include the classic coefficient of determination and  $p$  values, risk functions such as MSE, and newer approaches such as information theoretic techniques like AIC or BIC. Although many methods can be used to assess such error metrics, it can be argued that three basic categories of criteria should be employed to discuss the suitability of these metrics in practice:

**Accuracy:** Does the method accurately report error or goodness of fit?

**Accessibility:** Is the manner in which the results are reported understandable to end-users?

**Adaptability:** Can the method be applied to a variety of different modeling approaches?

The importance of accuracy is self-evident. If the method of reporting model quality contains significant biases itself, it may be of little value. The widely reported metric  $R^2$  is perhaps

the best example of a commonly used, yet potentially highly inaccurate measure of quality. The key issue with the coefficient of determination is, of course, that it will always increase as more variables are added to the model. The commonly used adjusted  $R^2$  measure attempts to account for this issue, but even that may be slightly biased when the model is overfit.

Accessibility is of equal importance as accuracy. A metric may objectively be highly accurate, but if practitioners “on the ground” are unable to apply it properly in practice, then it is flawed. An example of a widely discussed measure (Nickerson 2000; Cohen 1994; Loftus 1993; Falk and Greenbaum 1995 and others) that has repeatedly been shown to be inaccessible is the  $p$  value. Although the subject of basic statistics courses,  $p$  values have been demonstrated to not only be misunderstood by laypeople, but also to be misunderstood by those who absolutely should have a mastery of the topic. In one study of 30 university statistics instructors, 80% made at least one error when answering six basic true/false questions about the interpretation of  $p$  values. An example of one such question is, “[Given a  $p$  value] you know, if you decide to reject the null hypothesis, the probability that you are making the wrong decision.” (Haller and Krauss 2002).

Adaptability is of almost as much importance as accuracy and accessibility. This criterion indicates how flexible the error metric is in regards to being applied to different forms of model construction algorithms. Some metrics – such as AIC or BIC, for instance – require statistical models that generate likelihoods. Many types of models naturally generate likelihoods, but other types of model construction algorithms (such as CART, classification and regression trees; Breiman, Friedman, Olshen, and Stone 1984) may not lend themselves to generating likelihoods. By utilizing a metric that is limited to a certain subspace of model construction algorithms, we limit ourselves in our ability to compare results to other modeling techniques.

The A3 method (pronounced  $A^3$ ) and the **A3** package (Fortmann-Roe 2015) implementation of it for R (R Core Team 2015) targets these three criteria. The method is designed to be accessible, accurate and adaptable (it is from that acronym that the method is named) and the package is available from the Comprehensive R Archive Network (CRAN) at <http://CRAN.R-project.org/package=A3>.

In terms of accessibility, the method is designed to use familiar concepts. It is based around the  $R^2$  metric and  $p$  values. Because these metrics are familiar, most practitioners will not have to learn new concepts to understand the A3 output. The A3 method utilizes a derivative of  $R^2$  – the added  $R^2$  – in order to assess variable importance and the contribution of each feature to the success of the overall model. Unlike likelihood-based approaches – such as AIC-based approaches – or more application specific approaches – such as Gini importance for tree-base methods – this technique is quite general and so can be applied wherever A3 itself may be utilized.

The added  $R^2$  metric, analogous to semi partial correlation in linear regression, indicates how much a model improves when a given variable is added to the model. This change indicates the practical importance of the added variables which in practice may be more meaningful than statistical significance. The A3 method also provides a slope metric, analogous to the slope in linear regression, to indicate the effect of a variable on the outcome. In summary, the method calculates the following three items for each feature in a data set:

**Slope:** How a change in the feature affects the dependent variable. A distribution of values is calculated for each feature in addition to a single, summarizing average.

**Added  $R^2$ :** Predictive utility of the feature that is unique from all other features in the model.

**$p$  value:** The chance of seeing the observed level of predictive utility for a feature assuming the null hypothesis that the feature in fact has no predictive utility.

Muñoz and van der Laan (2012) introduced a parameter and derived asymptotically linear, semi-parametric estimators of that parameter to quantify the effect of displaced covariates on an outcome. If this parameter were to be normalized by the displacement amount, it would be similar to the slope calculated by the **A3** package. In the **A3** package, the slope parameter for a feature is estimated by approximating the slope at each data point using simple displacement and then averaging these results.

In terms of accuracy, the A3 method uses robust, resampling-based methods for the calculation of  $p$  values and  $R^2$  metrics.  $R^2$  is calculated using cross-validation which correctly accounts for overfitting and does well in matching the true  $R^2$  value. The term “ $R^2$ ” is defined here simply as the fraction of the squared error explained by a model compared to the null model. This is the definition used in the A3 method’s calculation of  $R^2$  and the definition used in the rest of this paper.  $p$  values are calculated using a randomization test. Full details about the methods are available in Appendix A. These methods may be computationally intensive for complex models, but they require no parametric assumptions other than independence between observations (a constraint which itself may be violated, see Section 5). The use of these methods make the results reported by the A3 method more robust to user misuse and abuse than do many standard parametric model results (where the parametric assumptions may frequently not be tested for in practice).

Last, the A3 method is highly adaptable to different modeling techniques. Technically, the A3 method is defined as a wrapper function that encapsulates an arbitrary predictive modeling algorithm. Thus the method can theoretically be applied to any modeling technique that generates predictions. The same principle is used in the **A3** package where the primary package functions can take an arbitrary predictive modeling function. Different modeling methods can be utilized by passing different functions (e.g., `lm` for linear regression models, `glm` for logistic regression models, `rpart` for CART models) to the package’s `a3` function. The A3 method can seamlessly encapsulate different techniques and generate a consistent output for them that facilitates the direct comparison of these different methods using the same criteria.

It should be noted that the A3 method is focused on inferential statistics and offers little in regards to descriptive statistics or visualizations. There are a wide range of sources addressing and advancing these topics including classics such as Cleveland’s *Visualizing Data* (Cleveland 1993) and Tukey’s *Exploratory Data Analysis* (Tukey 1976). R offers a variety of powerful packages to support descriptive analyses. Several R packages such as **Hmisc** and **pastecs** provide functions for quickly calculating descriptive statistics (Grosjean and Ibanez 2014; Harrell 2015). Other packages such as **data.table** or **dplyr**, in addition to the built-in `data.frame`, make it straightforward to subset and aggregate data by category (Dowle, Short, Lianoglou, and Srinivasan 2014; Wickham and François 2015). Lastly a number of packages such as the phenomenal **ggplot2** make it possible to rapidly create high-quality data visualizations (Wickham 2009). Rather than attempting to improve on these current capabilities in R, the **A3** package instead focuses on inferential statistics and prediction.

When the true data-generating process or model is unknown, the A3 method may be used for inference about the significance of features in the data set and for model selection to improve the predictive accuracy of a regression. If the true model were known (or there was strong evidence to justify the adoption of a parametric model), it would of course outperform the semi-parametric models A3 is generally applied to. The following sections of this work will first describe the **A3** package in general. Next, two applications of the method are developed for predictive and inferential tasks. Finally, a discussion of using the package to analyze correlated data will be presented followed by general conclusions. One important note should be made at this point. In general, I would make the claim that mathematical models are constructed for three purposes: prediction, inferences other than prediction, and conceptual/narrative applications (e.g., “telling a story about a system”). The **A3** package can be applied both to predictive and inferential usage cases. It is based on a predictive framework, but it can also assess the statistical significance of variables within this framework. The key for doing this is to make a small shift in our thinking about inference in order to reframe it in a predictive manner. As an example, take the question “Is there a relationship between  $X$  and  $Y$ ?” We can rephrase the same question in a predictive framework as “Does knowledge of  $X$  help us to predict  $Y$ ?” The second question is answered by the **A3** package allowing inferences based solely on the predictive accuracy of models.

## 2. Usage overview

The primary function provided by the **A3** package is `a3` which takes three principle arguments:

**formula:** A regression formula object.

**data:** A data frame containing the data for the regressions.

**model.fn:** A function that generates a regression model that has a corresponding `predict` method.

As example of the usage of the `a3` function, we may use R’s built-in `lm` function with the `a3` function to generate the **A3** results for a linear regression model of R’s built-in `attitude` data set. The output is an S3 object with a `print` method that displays an A3 results table. This table contains four columns: the features in the model, the median slope for each feature (analogous to the coefficient in a linear regression, and directly equivalent to it when used for a linear regression model), the cross-validated  $R^2$  for the whole model and the added  $R^2$  for each feature (a measure of the practical significance of that feature), and the  $p$  values for the model itself and for each feature.

```
R> a3(formula = rating ~ ., data = attitude, model.fn = lm)
```

	Average slope	CV	$R^2$	p value
-Full model-		47.8 %	< 0.01	
(Intercept)	10.78707639	-	8.5 %	0.97
complaints	0.61318761	+	10.5 %	< 0.01
privileges	-0.07305014	-	5.7 %	0.87
learning	0.32033212	+	7.8 %	0.05

raises	0.08173213	-	6.1 %	0.82
critical	0.03838145	-	8.3 %	0.98
advance	-0.21705668	-	3.6 %	0.59

Please note the  $p$  value estimates in this table and the rest of this paper are reported to only two decimals places. This is because the  $p$  values were estimated using 100 randomizations. More precise  $p$  value estimates can be obtained by increasing the number of random simulations in the calculations of the distribution of  $R^2$  values assuming the null hypothesis. For instance, 1,000 randomizations would give an estimate precision of 0.001 but would come at the cost of approximately ten times the computational effort. Refer to Section 4 and Appendix A for more details on the calculation of  $p$  values.

R comes with numerous predictive modeling algorithms in its core distribution and base packages. An even larger set of modeling techniques are available in the wider ecosystem of user contributed and maintained packages. The `a3` method can support most of these techniques (see Appendix B for details). For instance, the `e1071` package (Meyer, Dimitriadou, Hornik, Weingessel, and Leisch 2014) provides support vector machine regressions (Cortes and Vapnik 1995) using the function `svm`. We can use the `svm` function in place of `lm` to obtain the A3 results for support vector machines. Please note the use of “+0” in the formula object. This removes the constant term generated in the model matrix which is unnecessary for the `svm` function.

```
R> library("e1071")
R> a3(rating ~ . + 0, attitude, svm)

              Average slope   CV R^2 p value
-Full model-                42.3 % < 0.01
complaints      0.37018519 + 25.3 % < 0.01
privileges      0.079660425 -  3.5 %  0.44
learning        0.120305835 +  0.9 %  0.14
raises          0.07477541 +  4.0 %  0.11
critical       -0.138117725 - 11.5 %  0.92
advance        -0.1573689 +  5.4 %  0.06
```

As another example, we can use the `randomForest` package (Liaw and Wiener 2002) to generate random forest regressions (Breiman 2001). In order to speed calculation we can set the desired accuracy for the calculation of  $p$  values (`p.acc`) to a coarser value than the default 0.01. Setting `p.acc` to NULL would disable the calculation of  $p$  values completely.

```
R> library("randomForest")
R> out.rf <- a3(rating ~ . + 0, attitude, randomForest, p.acc = 0.05)
R> out.rf

              Average slope   CV R^2 p value
-Full model-                43.5 % < 0.05
complaints      0.033075 + 26.9 % < 0.05
privileges           0 +  2.2 %  1.00
learning        0.0199 + 13.1 %  0.05
```

	Average slope	CV $R^2$	$\Pr(> R^2)$
-Full model-		43.5%	< 0.05
complaints	0.033075	+26.9%	< 0.05
privileges	0	+2.2%	1.00
learning	0.0199	+13.1%	0.05
raises	0.036966665	+9.7%	0.25
critical	-0.012158335	-0.5%	1.00
advance	-0.008216665	+6.7%	0.65

Table 1: L<sup>A</sup>T<sub>E</sub>X formatted A3 output.

```
raises      0.036966665 + 9.7 % 0.25
critical    -0.012158335 - 0.5 % 1.00
advance     -0.008216665 + 6.7 % 0.65
```

Arguments passed to the `a3` function are not passed on to the function specified in argument `model.fn` (in this case `randomForest`). To pass additional arguments to the function specified in `model.fn`, you can pass a list of arguments to the argument `model.args`. For instance, the following code sets the `ntree` argument of `randomForest` to 1000.

```
R> a3(rating ~ . + 0, attitude, randomForest,
+     model.args = list(ntree = 1000), p.acc = 0.05)
```

```
                Average slope  CV R^2 p value
-Full model-          49.1 % < 0.05
complaints    0.042470835 + 30.1 % < 0.05
privileges           0 + 4.9 % 0.95
learning      0.039183335 + 17.7 % < 0.05
raises        0.039104165 + 9.9 % 0.55
critical      -0.00775 + 1.0 % 1.00
advance       -0.02085 + 6.7 % 0.60
```

In addition to the general `a3` function, we can also use the specialized `a3.lm` function specifically for linear models which removes the need for the argument `model.fn` (it is set automatically to `glm`). Both `a3` and `a3.lm` return an S3 ‘A3’ object. The `print` method for ‘A3’ objects prints an ASCII results table. You can also use the `xtable` package (Dahl 2014) to create a nicely formatted output table from this object (see Table 1).

```
R> xtable(out.rf, caption = "\\LaTeX \\, formatted A3 output.",
+         label = "xtableFormat")
```

Several plotting functions are included in the **A3** package to display results. `plotPredictions` is important in assessing the overall predictive accuracy of the model. For each observation in the data set, it plots the predicted and original values along with an optional line marking ideal results. In Figure 1 we can see that our random forest model for the attitude data appears to tend to overestimate ratings for low values and underestimate them for high ratings.

```
R> plotPredictions(out.rf)
```

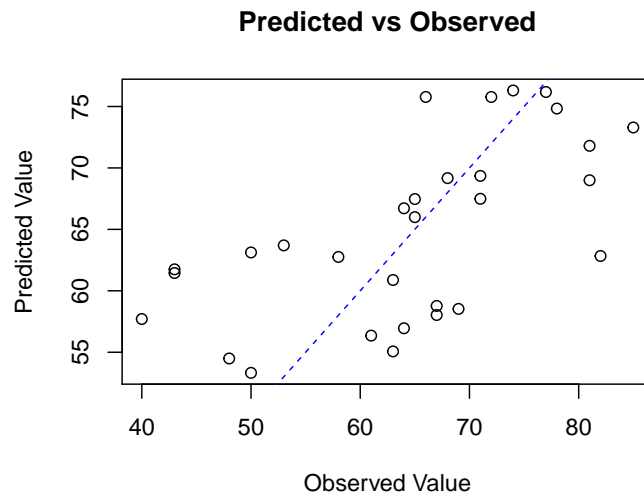


Figure 1: A plot of predicted versus observed values for the random forest model of the attitude data.

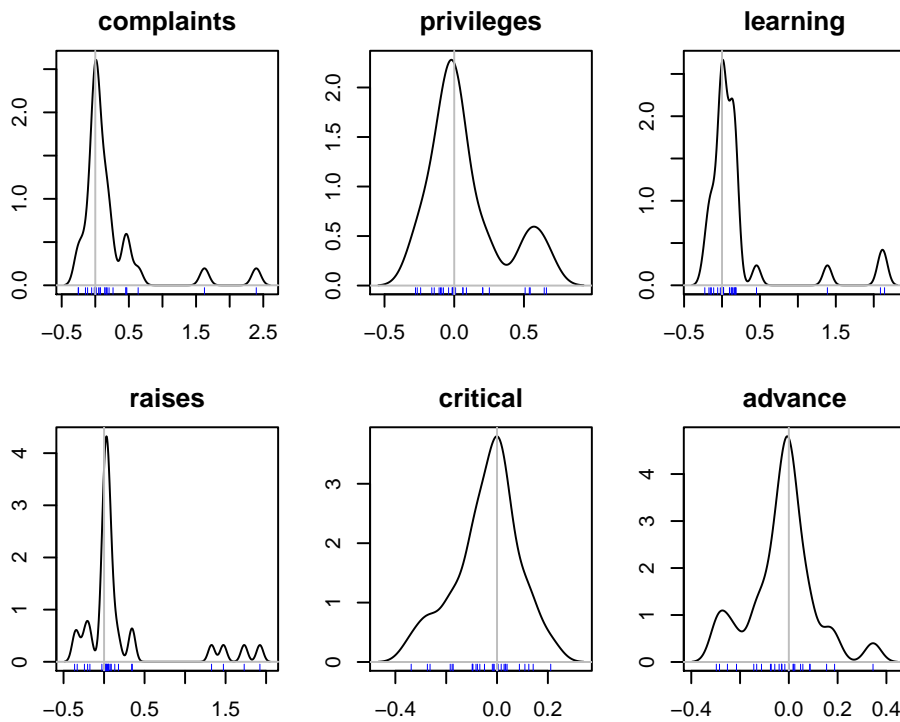


Figure 2: Slope distributions for the random forest model of the attitude data.

In the results table, the median of the slopes at each observation are reported. These slopes indicate how the prediction for an observation will change as an observation is displaced (or, put another way, whether a given feature has a positive or negative effect on the outcome

and how strong this effect is). A slope is calculated for each feature at each data point. The slope is approximated by calculating the value of the outcome at a pair of customizable, fixed displacements from each point. For linear regressions, this procedure will result in the same value as the regression coefficients. A fixed displacement is used rather than attempting to estimate the derivative directly at each point as methods such as random forests will often place a step function at data points leading to undefined derivatives at the points themselves. For some models, such as linear regressions, this slope will be constant between observations. However, for other models, such as random forests, the slope may change at each point as the model's behavior may differ between regions of the feature space. The `plotSlopes` function may be used to plot the distribution of slopes for each feature (see Figure 2).

```
R> plotSlopes(out.rf)
```

The `plot` method for 'A3' objects may be used to plot both the predictions and slopes at once.

### 3. Worked applications

The most effective way to illustrate the use of the **A3** package and its utility is through applied case studies. Two example applications will be used to illustrate the method. The first comes from an attempt to predict housing prices, the second is focused on drawing inferences in an ecological application.

#### 3.1. Housing application

This application is based on a data set that includes information on the prices of houses from the Boston area. It originally was developed by [Harrison and Rubinfeld \(1978\)](#) and the digital copy used in this analysis was provided by [Frank and Asuncion \(2010\)](#). The data set is included in the **A3** package as `housing`. The following are some of the key features in the data set (based on the summary of [Frank and Asuncion 2010](#)).

**NOX:** Nitrogen oxides pollutant concentration (parts per 10 million).

**ROOMS:** Average number of rooms per dwelling.

**AGE:** Proportion of owner-occupied units built prior to 1940.

**HIGHWAY:** Index of accessibility to radial highways.

**PUPIL.TEACHER:** Pupil-teacher ratio by town.

**MED.VALUE:** Median value of owner-occupied homes in \$1,000's.

A typical approach to analyzing this data set, either to build a predictive model or for inferences about the significances of the features, might be to apply a linear regression to the data set (it is important to note that the authors of the cited paper carry out a different form of analysis and this section is simply illustrative of a commonly applied approach). The results of a linear regression are shown in Table 2. Although these specific results are generated by R, the selection of displayed data is very similar to that of other packages. These results allow



	Estimate	Std. error	<i>t</i> value	Pr(>   <i>t</i>  )
(Intercept)	7.7674	4.9888	1.56	0.1201
AGE	-0.0151	0.0138	-1.10	0.2738
ROOMS	7.0056	0.4117	17.02	0.0000
NOX	-13.3142	3.9026	-3.41	0.0007
PUPIL.TEACHER	-1.1165	0.1480	-7.54	0.0000
HIGHWAY	-0.0249	0.0426	-0.58	0.5593

Table 2: Housing data linear regression model results ( $R^2 = 0.6037$ , Adjusted  $R^2 = 0.5997$ ).

us to clearly see that `ROOMS`, `NOX`, and `PUPIL.TEACHER` are statistically significant variables at the 5% level. However, they do not provide information on the practical importance of these variables in regards to prediction accuracy.

```
R> data("housing", package = "A3")
R> reg <- lm(MED.VALUE ~ AGE + ROOMS + NOX + PUPIL.TEACHER + HIGHWAY,
+   housing)
R> print.reg(reg, label = "HousingLinear",
+   caption = "Housing data linear regression model results")
```

In order to attempt to gain an understanding of the practical significance of the different variables, we can use the `A3` results. Table 3 contains the `A3` results for a linear regression of the housing data. The primary changes in the physical construction of this table is the removal of the standard error and *t* value columns and the addition of the cross-validated  $R^2$  column. This column first reports the  $R^2$  for the whole model, and then the added  $R^2$ 's for each feature. The cross-validated  $R^2$  is slightly lower, as will generally be the case for overfitting models, than the adjusted  $R^2$ .

From the added  $R^2$  measures reported in the cross-validated  $R^2$  column, it can be seen that the `ROOMS` variable explains 23% more of the squared error when it is added to the model while the `NOX` variable explains less than 1% of the squared error when it is added to the model. Although both these variables are highly statistically significant ( $p < 0.01$ ), the `A3` output makes it very clear that `ROOMS` is a much more important predictor of housing prices than `NOX`. In applications, this information may have great practical importance.

```
R> housing.lm <- a3.lm(MED.VALUE ~ AGE + ROOMS + NOX + PUPIL.TEACHER +
+   HIGHWAY, housing, p.acc = 0.01, n.folds = 50)
```

Furthermore, the `A3` package allows the straightforward comparison of results between different forms of predictive models. The `A3` method is a wrapper that can theoretically be used on any predictive model (see Appendix B for more details). Tables 4 and 5 show the results of the `A3` method applied to, respectively, a support vector machine model construction algorithm and a random forest model construction algorithm for the housing data set. There are two primary things to note from these results. The first is both of these algorithms were able to generate much better predictive models (10% to 15% higher  $R^2$  values for the full models) than the linear regression.

```
R> housing.svm <- a3(MED.VALUE ~ AGE + ROOMS + NOX + PUPIL.TEACHER +
+   HIGHWAY + 0, housing, svm, p.acc = 0.01, n.folds = 50)
```

	Average slope	CV $R^2$	Pr(> $R^2$ )
-Full model-		59.3%	< 0.01
(Intercept)	7.7674	-0.1%	0.26
AGE	-0.0151	-0.1%	0.25
ROOMS	7.0056	+23.2%	< 0.01
NOX	-13.3142	+0.8%	< 0.01
PUPIL.TEACHER	-1.1165	+4.5%	< 0.01
HIGHWAY	-0.0249	-0.2%	0.96

Table 3: Housing data A3 results for a linear regression model.

	Average slope	CV $R^2$	Pr(> $R^2$ )
-Full model-		70.9%	< 0.01
AGE	-0.069444	+0.9%	0.01
ROOMS	5.188636	+34.7%	< 0.01
NOX	-0.000581	+2.2%	< 0.01
PUPIL.TEACHER	-0.247277	+1.6%	< 0.01
HIGHWAY	0.042276	-0.3%	0.16

Table 4: Housing data A3 results for the support vector machine model.

	Average slope	CV $R^2$	Pr(> $R^2$ )
-Full model-		74.1%	< 0.01
AGE	-0.0346	-1.1%	< 0.01
ROOMS	4.5326	+20.3%	< 0.01
NOX	-1.4777	+6.1%	< 0.01
PUPIL.TEACHER	-0.7140	-1.1%	< 0.01
HIGHWAY	0.0000	-2.2%	0.02

Table 5: Housing data A3 results for the random forest model.

```
R> housing.rf <- a3(MED.VALUE ~ AGE + ROOMS + NOX + PUPIL.TEACHER +
+   HIGHWAY + 0, housing, randomForest, p.acc = 0.01, n.folds = 50)
```

The second item of note, and which is of even greater importance, is that the inferences drawn from the data have changed in these latter models. In the linear regression model, the **AGE** variable is not at all statistically significant. However in both the support vector machine and random forest models, it is significant ( $p = 0.01$ ). Thus we can see that the data does in fact support a relationship between the age of a house and its price. It is not a trivial linear relationship, but it does exist. If we constrained ourselves to only explore linear models, as is often done in practice, we would have failed to identify this relationship and the significance of the **AGE** variable. It should still be noted however, that if we were building a predictive model, it would still be best to exclude **AGE** from the model despite this statistical significance due to its low added  $R^2$ .

Lastly, it is beneficial to gain a fuller understanding of the meaning of the Average slope column in these results. Unlike the case for the linear regression model, the slopes in the support vector machine and random forest models may change between different regions of the data space. As such it can be useful to plot the distribution of slopes rather than simply

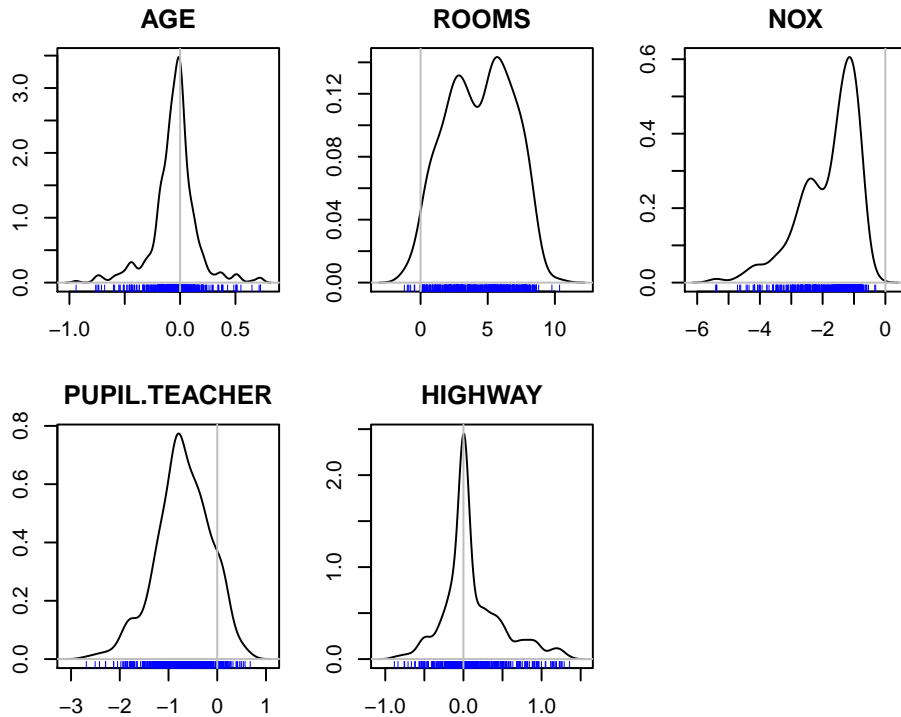


Figure 3: A plot of the slopes for the random forest model of the housing data.

relying on the median value as shown in the table. The results of this distribution are shown in Figure 3.

```
R> plotSlopes(housing.rf)
```

From this figure it can be seen that the variable NOX always has a negative effect while a variable such as AGE has differing effects – sometimes negative and sometimes positive – depending on where a specific observation fits.

### 3.2. Ecology application

This case study relates a dryland ecosystem’s multifunctionality (roughly speaking, the magnitude of different services performed by an ecosystem) to a range of environmental variables. It was collected by a large team of researchers and published in the journal *Science* along with a statistical analysis (Maestre *et al.* 2012).

A copy of the data set is provided by the **A3** package in the data variable `multifunctionality`. The following features are in the data set:

**ELE:** Elevation of the site.

**LAT and LONG:** Location of the site.

**SLO:** Site slope.

	Estimate	Std. error	<i>t</i> value	Pr(>   <i>t</i>  )
(Intercept)	1.0081	0.1747	5.77	0.0000
SR	0.0099	0.0042	2.35	0.0197
SLO	0.0176	0.0056	3.14	0.0019
SAC	-0.0174	0.0020	-8.52	0.0000
PCA_C1	-0.0209	0.0389	-0.54	0.5918
PCA_C2	-0.0677	0.0527	-1.28	0.2004
PCA_C3	0.0348	0.0355	0.98	0.3285
PCA_C4	-0.2663	0.0380	-7.00	0.0000
LAT	0.0024	0.0013	1.80	0.0737
LONG	-0.0019	0.0005	-3.47	0.0006
ELE	-0.0002	0.0001	-3.89	0.0001

Table 6: Multifunctionality data linear regression model results ( $R^2 = 0.5645$ , Adjusted  $R^2 = 0.5441$ ).

SAC: Soil sand content.

PCA\_C1, PCA\_C2, PCA\_C3, PCA\_C4: Principal components of a set of 21 climatic features.

SR: Species richness.

MUL: Multifunctionality.

The original data were analyzed using a multi-model inference approach after [Burnham and Anderson \(2002\)](#). Although the authors also looked at a simultaneous autoregression model, only linear regressions were considered in this multi-model inference and so will be the focus of this reanalysis. The full linear regression explored in the original work is shown in Table 6 with all other linear regressions explored in their work being subsets of this one. From the table, we can see that SR, SLO, SAC, PCA\_C4, LONG, and ELE are all statistically significant at the 5% level.

```
R> data("multifunctionality", package = "A3")
R> reg <- lm(MUL ~ SR + SLO + SAC + PCA_C1 + PCA_C2 + PCA_C3 + PCA_C4 +
+ LAT + LONG + ELE, multifunctionality)
R> print.reg(reg, label = "MFLlinear", caption =
+ "Multifunctionality data linear regression model results")
```

However, as with the housing application, we are again missing information on the importance of each of these variables in regards to the actual predictive accuracy of the model. This information is simply not available in the standard linear regression output table. The A3 method, however, makes it very clear. As shown in Table 7, SAC explains the greatest additional squared error when added to the model, followed by PCA\_C4. This basic conclusion agrees with the conclusions of the original researchers as they note, “By this criterion [the sum of Akaike weights across models], the two most important predictors of multifunctionality were annual mean temperature (reflected in ... [PCA\_C4]) and the sand content in the soil [SAC].”

	Average slope	CV $R^2$	Pr(> $R^2$ )
-Full model-		52.6%	< 0.01
(Intercept)	1.00810	+7.2%	< 0.01
SR	0.00989	+0.9%	< 0.01
SLO	0.01761	+1.8%	< 0.01
SAC	-0.01742	+16.5%	< 0.01
PCA_C1	-0.02087	-0.5%	0.92
PCA_C2	-0.06771	+0.1%	0.12
PCA_C3	0.03479	-0.2%	0.25
PCA_C4	-0.26630	+10.6%	< 0.01
LAT	0.00235	+0.2%	0.06
LONG	-0.00187	+2.3%	< 0.01
ELE	-0.00025	+2.8%	< 0.01

Table 7: Multifunctionality data A3 Results for a linear regression model.

	Average slope	CV $R^2$	Pr(> $R^2$ )
-Full model-		67.9%	< 0.01
SR	0.001592	+1.2%	< 0.01
SLO	0.002248	-1.4%	0.87
SAC	-0.004656	+3.4%	< 0.01
PCA_C1	0.072615	+1.5%	< 0.01
PCA_C2	0.027495	+0.3%	0.01
PCA_C3	0.015621	-0.1%	0.24
PCA_C4	-0.090020	+0.7%	< 0.01
LAT	0.012747	+0.9%	< 0.01
LONG	0.000476	+0.4%	0.04
ELE	0.000000	+0.4%	< 0.01

Table 8: Multifunctionality data A3 results for the random forest model.

```
R> mult.lm <- a3.lm(MUL ~ SR + SLO + SAC + PCA_C1 + PCA_C2 + PCA_C3 +
+   PCA_C4 + LAT + LONG + ELE, multifunctionality, p.acc = 0.01,
+   n.folds = 50)
```

However, if we only use a linear regression model we run the risk of missing important nonlinear relationships. Using the **A3** package, we can quickly explore different modeling techniques and compare them to the linear results. Table 8 shows the A3 results for applying a random forest model to the multifunctionality data.

```
R> mult.rf <- a3(MUL ~ SR + SLO + SAC + PCA_C1 + PCA_C2 + PCA_C3 +
+   PCA_C4 + LAT + LONG + ELE + 0, multifunctionality, randomForest,
+   p.acc = 0.01, n.folds = 50)
```

As with the housing data, there are two things to note. First, the random forest model has much higher predictive accuracy compared to the linear regression model ( $R^2$  of 0.679 compared to 0.526). Second, our inferences about the significances of the features change. For instance, PCA\_C1 was not significant when using a linear regression. When using the random

forest model, however, it became highly significant. As with the housing data, we can see that variables which do not appear statistically significant when using classical linear regression, may actually be highly significant; a fact that can sometimes be revealed when using more data adaptive modeling methods such as random forests.

We can also use the added  $R^2$ 's to go beyond simple statistical significance to determine the importance of the different features. From these, we can see that **SAC** and **PCA\_C1** are the two most important features in the random forest model which is different from the **SAC** and **PCA\_C4** features identified as the most important in the linear regression model. Such changing inferences may fundamentally alter the scientific results and conclusions that are drawn from a study. It should also be noted that in the random forest results, the added  $R^2$ 's are by and large much lower than those seen for the linear regression results. This is most likely due to correlation between the features. The random forest model, in this case, is better at exploiting the duplicated information leading to lower added  $R^2$ 's.

#### 4. Controlling A3 accuracy and computation time

For large data sets or complex model construction algorithms, the **A3** package may require significant computation effort. There are several parameters that can be used to fine-tune the behavior of the package and to either speed up computation or obtain increased precision in results.

**n.folds**: is the number of folds to use in the  $k$ -fold cross-validation. Increased number of folds leads to increased computation time. Generally speaking, a small number of folds will lead to an over-estimation of model error and the higher the number of folds, the more accurate the results will be. Since small numbers of folds lead to over-estimation of error, you may generally safely reduce the number of folds if computation is taking too long and you will obtain a conservative estimate of model accuracy. The maximum number of folds is the number of observations (leave-one-out cross-validation) and a value of 0 for **n.folds** is a shorthand to use this.

**p.acc**: controls the desired precision for the calculation of  $p$  values. A value of 0.01, for instance means  $p$  values will be calculated to the second decimal place. A value of 0.001, indicates that they will be calculated to the third decimal place. **p.acc** has a very important effect on performance. Roughly speaking, **a3** computation time is proportional to  $1/\text{p.acc}$ . A value of **NULL** for **p.acc** specifies that  $p$  values should not be calculated at all.

**features**: controls the calculation of results for each feature. By default,  $p$  values and added  $R^2$ 's are calculated for each of the features in the model. You can turn off the calculation of these by setting the **features** argument to **FALSE**. When **features** is **TRUE**, computation time is proportional to the number of features.

Generally speaking, given a model construction algorithm that takes time  $t$  to complete for a given data set, the **A3** package will, if **features** is **TRUE**, require  $T$  time as approximated in Equation 1.

$$T \approx \frac{(\text{n.features} + 1) \times \text{n.folds} \times t}{\text{p.acc}} \quad (1)$$

If `features` is `FALSE`,  $T$  may be approximated with Equation 2.

$$T \approx \frac{\text{n.folds} \times t}{\text{p.acc}} \quad (2)$$

Fortunately, the A3 method falls under the category of “embarrassingly parallelizable” algorithms that may theoretically be split between multiple machines in a trivial manner. Although the package does not yet contain built-in parallelization code, this is planned for a future version of the package. Given enough resources for parallel computing, it would be conceivable to achieve close to  $T \approx t$  in practice (assuming the model construction algorithm itself cannot take advantage of the parallel resources).

## 5. Dealing with correlation between observations

The basic A3 method makes no assumptions about the generation of the data except for one: that observations are independent and identically distributed. However, many prediction and inference tasks may in fact violate this assumption. For instance, some form of correlation between observations will often be the norm when dealing with temporal or geographic data. In fact, both of the illustrative case studies presented in Section 3, arguably exhibit a spatial correlation structure that was not addressed in the initial analyses.

When correlations between observations exist,  $p$  values will be biased. Fortunately, the A3 method contains a built-in way to directly correct for these biases. The method of calculating  $p$  values in the A3 method is based on generating random data with the same properties as the original data. As a consequence, this makes it generally straightforward to adjust for well-defined correlation structures or other issues: simply replicate that correlation structure in the randomly generated data.

As an illustrative example we can generate two first order auto-correlated data series: `x` and `y`. These series are independent of each other. The auto-correlated nature of the series, however, when not corrected for, creates artificially significant  $p$  values when attempting to use one to predict the other (Table 9).

```
R> set.seed(1)
R> createAutoCorrelatedSeries <- function(n, r) {
+   dat <- rnorm(n, 0, 1)
+   for(i in 2:n) dat[i] <- dat[i - 1] * r + dat[i] * (1 - r)
+   dat
+ }
R> sample <- data.frame(x = createAutoCorrelatedSeries(100, 0.95),
+   y = createAutoCorrelatedSeries(100, 0.95))
R> reg <- lm(y ~ x, sample)
R> print.reg(reg, label = "genAutoCor",
+   caption = "Biased $p$~values as a result of auto-correlation")
```

The method of calculating  $p$  values used by the A3 method can adjust for this by generating stochastic noise with the same correlation structure as the observed data. If this is not done, however, the method will also generate biased  $p$  values (Table 10).

	Estimate	Std. error	$t$ value	$\Pr(>  t )$
(Intercept)	-0.1398	0.0168	-8.32	0.0000
$x$	0.5513	0.0867	6.36	0.0000

Table 9: Biased  $p$  values as a result of auto-correlation ( $R^2 = 0.2922$ , Adjusted  $R^2 = 0.285$ ).

	Average slope	CV $R^2$	$\Pr(> R^2)$
-Full model-		27.0%	< 0.01
(Intercept)	-0.13983683	+48.7%	< 0.01
$x$	0.55127716	+27.0%	< 0.01

Table 10: Biased  $p$  values in the A3 method as a result of auto-correlation.

	Average slope	CV $R^2$	$\Pr(> R^2)$
-Full model-		26.9%	0.45
(Intercept)	-0.13983683	+46.9%	< 0.01
$x$	0.55127716	+26.9%	0.57

Table 11: A3 corrected  $p$  values accounting for auto-correlation.

```
R> out <- a3.lm(y ~ x, sample)
R> xtable(out, label = "A3AutoCor", caption =
+   "Biased  $p$  values in the A3 method as a result of auto-correlation.")
```

The `a3` and `a3.lm` functions contain an argument `data.generating.fn` which can be used to specify the method for generating random noise. This argument takes a list of functions one for each of the independent columns in the model matrix. By default, the `data.generating.fn` is primarily a resampling based method. However, this is not valid for auto-correlated data. Our simulated data can be correctly analyzed by setting the `data.generating.fn` argument to the `a3.gen.autocor` function which generates first-order auto-correlated data with the same properties as the original data. The results with corrected  $p$  values are shown in Table 11. Please note that in this table, the intercept has a higher added  $R^2$  than the  $R^2$  for the overall model. This simply indicates that without the intercept term, the model is actually worse than the null model. In fact, if we remove the independent variable from the model, we can see that the  $R^2$  value will be 0, as we are left with just the intercept which is the definition of the null model.

```
R> out <- a3.lm(y ~ x, sample,
+   data.generating.fn = list(a3.gen.default, a3.gen.autocor))
R> xtable(out, label = "A3AutoCorCorrect", caption =
+   "A3 corrected  $p$  values accounting for auto-correlation.")
```

## 6. Conclusions

As a thought experiment, we can use the metaphor of a mountain range to describe the space of all models. Each latitude and longitude in the range represents a different model and the height of the range at that point corresponds to the accuracy of that model given a prediction



task. Each peak in the range might consist of a single type of model. For instance, one peak might correspond to linear regressions, another peak to support vector machines, yet another peak to a set of mechanistic models, and so on. What is often done in practice when building models for prediction or other inferences is to explore only one peak in this range of mountains. Whether it be just linear regressions or some other technique, researchers and practitioners often only explore a single model family and then make broad conclusions based on the results of this limited analysis.

Conclusions drawn from such a narrow exploration of the model space cannot but be viewed with skepticism. In practice, it is almost impossible to say *a priori* for an unknown data generating process that, for instance, a linear regression model will do the best job of available algorithms in approximating it. It is imperative that practitioners attempt to further explore the model space beyond a single peak – beyond a single model family. The **A3** package facilitates this exploration by defining an adaptable algorithm and reporting format that allows the direct comparison of results between different predictive model families.

When predictions and statistical measures of significance drawn from models affect policy and scientific decisions, it is of great importance that the best suited modeling techniques available be used. The two worked applications in this paper have demonstrated the use of the **A3** package to facilitate the exploration of the model space. In both cases, it was a straightforward process to obtain significantly more predictive models (10%–15% additional explanation of the squared error) compared to the linear regression models simply by exploring one or two additional model families. More importantly, the more predictive models revealed altered conclusions about the significance of the drivers of the systems. Variables that were not statistically significant in the linear regression model became statistically significant in the more predictive random forest models, and vice versa. The ease at which our scientific conclusions can change when we apply more accurate models should be strong motivation to explore a greater part of this infinitely large range of models. The **A3** method is just one tool to facilitate this exploration.

## Computational details

The analyses in this article were conducted using the following software versions R 2.15.2, **A3** 0.9.2, **e1071** 1.6-1, **pbapply** 1.0-5, **randomForest** 4.6-7, and **xtable** 1.7-1. When using more recent versions of R and these packages, small differences in results as compared to the replication script may be observed due to the stochastic nature of the **A3** method. Despite these small differences, the results remain qualitatively equivalent.

## References

- Breiman L (2001). “Random Forests.” *Machine Learning*, **45**(1), 5–32.
- Breiman L, Friedman JH, Olshen RA, Stone CJ (1984). *Classification and Regression Trees*. Wadsworth, California.
- Burnham KP, Anderson DR (2002). *Model Selection and Multimodel Inference: A Practical Information-Theoretic Approach*. Springer-Verlag.

- Cleveland W (1993). *Visualizing Data*. Hobart Press, Summit.
- Cohen J (1994). “The Earth Is Round ( $p < .05$ ).” *American Psychologist*, **49**(12), 997–1003.
- Cortes C, Vapnik V (1995). “Support-Vector Networks.” *Machine Learning*, **20**(3), 273–297.
- Dahl DB (2014). *xtable: Export Tables to L<sup>A</sup>T<sub>E</sub>X or HTML*. R package version 1.7-4, URL <http://CRAN.R-project.org/package=xtable>.
- Dowle M, Short T, Lianoglou S, Srinivasan A (2014). *data.table: Extension of data.frame*. R package version 1.9.4; with contributions from R Saporta and E Antonyan, URL <http://CRAN.R-project.org/package=data.table>.
- Falk R, Greenbaum CW (1995). “Significance Tests Die Hard: The Amazing Persistence of a Probabilistic Misconception.” *Theory & Psychology*, **5**(1), 75–98.
- Fortmann-Roe S (2015). *A3: Accurate, Adaptable and Accessible Model Error Metrics*. R package version 1.0.0, URL <http://CRAN.R-project.org/package=A3>.
- Frank A, Asuncion A (2010). “UCI Machine Learning Repository.” University of California, Irvine, School of Information and Computer Sciences. URL <http://archive.ics.uci.edu/ml>.
- Grosjean P, Ibanez F (2014). *pastecs: Package for Analysis of Space-Time Ecological Series*. R package version 1.3-18, URL <http://CRAN.R-project.org/package=pastecs>.
- Haller H, Krauss S (2002). “Misinterpretations of Significance: A Problem Students Share with Their Teachers.” *Methods of Psychological Research Online*, **7**(1), 1–20.
- Harrell Jr FE (2015). *Hmisc: Harrell Miscellaneous*. R package version 3.16-0; with contributions from Charles Dupont and many others, URL <http://CRAN.R-project.org/package=Hmisc>.
- Harrison D, Rubinfeld DL (1978). “Hedonic Housing Prices and the Demand for Clean Air.” *Journal of Environmental Economics and Management*, **5**(1), 81–102.
- Liaw A, Wiener M (2002). “Classification and Regression by **randomForest**.” *R News*, **2**(3), 18–22.
- Loftus GR (1993). “A Picture is Worth a Thousand  $p$  Values: On the Irrelevance of Hypothesis Testing in the Microcomputer Age.” *Behavior Research Methods*, **25**(2), 250–256.
- Maestre FT, Quero JL, Gotelli NJ, Escudero A, Ochoa V, Delgado-Baquerizo M, Garcia-Gomez M, Bowker MA, Soliveres S, Escolar C, Garcia-Palacios P, Berdugo M, Valencia E, Gozalo B, Gallardo A, Aguilera L, Arredondo T, Blones J, Boeken B, Bran D, Conceicao AA, Cabrera O, Chaieb M, Derak M, Eldridge DJ, Espinosa CI, Florentino A, Gaitan J, Gatica MG, Ghiloufi W, Gomez-Gonzalez S, Gutierrez JR, Hernandez RM, Huang X, Huber-Sannwald E, Jankju M, Miriti M, Monerris J, Mau RL, Morici E, Naseri K, Ospina A, Polo V, Prina A, Pucheta E, Ramirez-Collantes DA, Romao R, Tighe M, Torres-Diaz C, Val J, Veiga JP, Wang D, Zaady E (2012). “Plant Species Richness and Ecosystem Multifunctionality in Global Drylands.” *Science*, **335**(6065), 214–218.

- Meyer D, Dimitriadou E, Hornik K, Weingessel A, Leisch F (2014). *e1071: Misc Functions of the Department of Statistics (e1071), TU Wien*. R package version 1.6-4, URL <http://CRAN.R-project.org/package=e1071>.
- Muñoz ID, van der Laan M (2012). “Population Intervention Causal Effects Based on Stochastic Interventions.” *Biometrics*, **68**(2), 541–549.
- Nickerson RS (2000). “Null Hypothesis Significance Testing: A Review of an Old and Continuing Controversy.” *Psychological Methods*, **5**(2), 241–301.
- R Core Team (2015). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL <http://www.R-project.org/>.
- Stone M (1974). “Cross-Validatory Choice and Assessment of Statistical Predictions.” *Journal of the Royal Statistical Society B*, **36**(2), 111–147.
- Tukey JW (1976). *Exploratory Data Analysis*. Addison-Wesley, Massachusetts.
- Wickham H (2009). *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag, New York.
- Wickham H, François R (2015). *dplyr: A Grammar of Data Manipulation*. R package version 0.4.1, URL <http://CRAN.R-project.org/package=dplyr>.

## A. Algorithm details

This appendix describes the primary algorithms used by the **A3** package. It provides brief narrative descriptions along with algorithmic outlines. For further details, the commented source code in the package itself should be referred to.

### A.1. Slopes

The **A3** package calculates a measure of slope for each feature that is approximately analogous to the regression coefficient in linear regressions (and in fact it reduces to the regression coefficient when applied to linear regression models). One common way coefficients are described when discussing linear regressions is that they represent how much the dependent variable changes for one unit of change in the independent variable. The slope as reported by the **A3** package is calculated in directly this way and is done so for each point in the data set.

The reason that this apparently crude measure approximation of slope is used, rather than attempting to actually estimate the derivative right at each point, is that many models generate non-smooth functions (CART, random forests, etc). For instance, in a random forest model, the exact slope at a point will either be 0 (if there is not a branch at that point), or infinite (if there is a branch). Thus the straightforward  $\pm n$  metric (where  $n$  is a user-adjustable value which may vary by feature) is used instead of attempting to precisely estimate the derivative at a point.

```
function Slopes(Features, Results, n)
  Data: The data includes Features, a matrix where each column is a feature and each
    row an observation, and Results, a vector where each element corresponds to a
    row in the Features matrix. A model construction algorithm FitModel is also
    assumed with the resulting model having a PredictResult function.
  Result: A vector of slopes for each feature at each observation.
  Slopes = [];
  Model = FitModel(Features);
  for Feature ∈ Columns(Features) do
    Slopes[Feature] = [];
    for Observation ∈ Rows(Features) do
      Lower = Clone(Features);
      Lower[Observation, Feature] = Lower[Observation, Feature] - n;
      LowerValue = Model.PredictResult(Lower);
      Upper = Clone(Features);
      Upper[Observation, Feature] = Upper[Observation, Feature] + n;
      UpperValue = Model.PredictResult(Upper);
      Slopes[Feature][Observation] = (UpperValue - LowerValue)/2;
    end
  end
  return Slopes
end
```

**Algorithm 1:** The calculation of slope distributions.

```

function CrossValidatedR2(Features, Results)
  Data: The data includes Features, a matrix where each column is a feature and each
    row an observation, and Results, a vector where each element corresponds to a
    row in the Features matrix. A model construction algorithm FitModel is also
    assumed with the resulting model having a PredictResult function.
  Result: The cross-validated  $R^2$  for the model construction algorithm and data set.
    This version of the algorithm uses leave-one-out cross-validation.
  SumSquaredErrorNull = 0;
  SumSquaredErrorModel = 0;
  for Observation  $\in$  Rows(Features) do
    SumSquaredErrorNull =
      SumSquaredErrorNull + (Mean(Results[-Observation]) - Results[Observation])2;
    Model = FitModel(Features[-Observation, AllColumns]);
    SumSquaredErrorModel = SumSquaredErrorModel +
      (Model.PredictResult(Features[Observation, AllColumns]) -
        Results[Observation])2;
  end
  RCrossValidated2 = 1 - SumSquaredErrorModel/SumSquaredErrorNull;
  return RCrossValidated2;
end

```

**Algorithm 2:** Calculation of cross-validated  $R^2$ .

## A.2. Cross-validated $R^2$

The calculation of cross-validated  $R^2$  is straightforward. Cross-validation is a widely used technique in which a data set is divided into smaller subsets and where the error for each subset is determined using a model developed without that subset (Stone 1974). The **A3** package supports  $k$ -fold cross-validation. Algorithm 2 details the calculation of the cross-validated  $R^2$  using leave-one-out cross-validation (when the  $k$  in  $k$ -fold cross-validation is the number of observations).

Algorithm 3 details the calculation of the added  $R^2$ 's in the model. These indicate how much the predictive accuracy of the model is increased when a given feature is added to the model. This is analogous to semi-partial correlation in linear regression.

## A.3. $p$ values

The calculation of  $p$  values is done using a randomization test where the accuracy of the added cross-validated  $R^2$  for a portion of the data is compared to that for randomly generated data. Assuming the properties of the randomized data are the same as for the real data, the distribution of  $R^2$  values obtained using the randomized data represents the distribution of  $R^2$  values were the null hypothesis is true. A  $p$  value is then estimated by finding the position of the observed  $R^2$  values within this distribution. A suitable method for generating the stochastic data must be chosen such that the simulated data has the same properties as the actual data. For independent and identically distributed data, a resampling method is used by default in the **A3**. For correlated data, more complex techniques must be used. The **A3** package comes with a function `a3.gen.autocor` to help generating stochastic data for

```

function AddedR2(Features, Results)
  Data: The data includes Features, a matrix where each column is a feature and each
    row an observation, and Results, a vector where each element corresponds to a
    row in the Features matrix.
  Result: A vector of added  $R^2$ 's corresponding to each of the features in the model.
   $R_{FullModel}^2 = CrossValidatedR^2(Features, Results);$ 
   $R_{Added}^2 = [];$ 
  for Feature  $\in$  Columns(Features) do
     $R_{Submodel}^2 = CrossValidatedR^2(Features[AllRows, -Feature], Results);$ 
     $R_{Added}^2[Feature] = R_{FullModel}^2 - R_{Submodel}^2;$ 
  end
  return  $R_{Added}^2;$ 
end

```

**Algorithm 3:** Calculation of added  $R^2$ 's.

```

function pValue(Features, Results, Accuracy)
  Data: The data includes Features, a matrix where each column is a feature and each
    row an observation, Results, a vector where each element corresponds to a row
    in the Features matrix, and Accuracy, the desired accuracy for  $p$  values. The
    existence is also assumed of a function GenerateRandomFeatures that creates
    stochastic noise with the same properties as the Features matrix.
  Result: The  $p$  value for the significance of the entire model.
   $Iterations = Ceiling(1/Accuracy);$ 
   $R_{Original}^2 = CrossValidatedR^2(Features, Results);$ 
   $R_{Stochastic}^2 = [];$ 
  for  $i \in [1..Iterations]$  do
     $DataVector = GenerateRandomFeatures(Features);$ 
     $R_{Stochastic}^2[i] = CrossValidatedR^2(DataVector, Results);$ 
  end
   $p = 1 - Quantile(R_{Original}^2, Sort(R_{Stochastic}^2));$ 
  return  $p$ 
end

```

**Algorithm 4:** Calculation of  $p$  values.

first-order auto-correlated data such as may sometimes be applicable to temporal data.

The basic algorithm is detailed in Algorithm 4. The same technique can also be applied to calculated the significance of individual features.

## B. Integration

The **a3** function, in effect, “wraps” an existing algorithm and generates its statistics based on the output of the method it is wrapping. As such, there is necessarily a level of abstraction when using the A3 method that may make it difficult to fully utilize the unique features of a given modeling technique that could be used if the modeling technique were applied directly.

The technique it uses to carry out this wrapping works for many existing R model construction functions. However, in some cases the implementation of the target model construction algorithm may make it fail. In these instances, custom code may be required to bridge the **A3** package and the target model construction algorithm.

The **a3** method assumes the following properties of a model function denoted  $f$ :

- $f$  accepts a **formula** argument that specifies a regression relationship.
- $f$  accepts a **data** argument that contains a data frame for the specified formula.
- $f$  returns a model object for which a **predict** method has been defined. This **predict** method's first argument should be the regression model and the second argument should be new data from which to generate predictions.

Many, built-in R functions conform to these three criteria as do many user contributed packages and functions. However, in cases where a function does not conform to this specification, custom code may be written to allow to use the function with the **a3** function. As an example, the following general framework could be used. In it, we create a wrapper for a model with an associated predict function. We then call **a3** using the wrapper function.

```
R> customFunction.wrapper <- function(formula, data){
+   x <- createModelWithCustomFunction(...)
+   class(x) <- "a3CustomFunction"
+   x
+ }
R> predict.a3CustomFunction <- function(regression, new.data){
+   predictWithCustomFunction(...)
+ }
R> a3(y ~ ., data, customFunction.wrapper)
```

### Affiliation:

Scott Fortmann-Roe  
 Department of Environmental Science, Policy and Management  
 130 Mulford Hall  
 University of California  
 Berkeley, CA 94720, United States of America  
 E-mail: [ScottFR@Berkeley.edu](mailto:ScottFR@Berkeley.edu)  
 URL: <http://scott.fortmann-roe.com/>