maar: Tidy Inference under the 'Models as Approximations' Framework in R

Abstract

Linear regression using ordinary least squares (OLS) is a critical part of a statistician's toolkit. In R, this is elegantly implemented via the lm() and related functions. However, this functionality is based on the unrealistic assumption that the linear model is well-specified. A large body of econometric and statistical research has focused on performing OLS inference under more practical assumptions. There is a need for a comprehensive workflow in R for OLS under these practical assumptions. In this paper, we introduce our R package maar, to enable a *tidy* inferential workflow in the misspecified OLS setting. Under potential misspecification, we demonstrate how the maar package performs variance estimation, confidence interval computation, hypothesis testing, and diagnostics. The outputs of maar functions are tailored for further exploration using the tidyverse R packages.

1 Introduction

Multiple linear regression is one of the most commonly used data analytic tools. Although the general notion of "all models are approximations" is well received in terms of inference under model misspecification, most statistical software (including R) does not yet have a comprehensive implementation of all such aspects. In detail, the elegant implementation of the least squares estimator through the lm() function is based on the classical and unrealistic assumptions of a linear model i.e. linearity, homoscedasticity, and normality. The lm() implementation does provide methods for diagnosing these assumptions. However, it has long been recognized that inference should be performed under more general assumptions, even if the estimator is constructed based on likelihood principles [1, 5].

Over the years, several R packages (such as car [3] and sandwich [15, 16]) have been developed to implement heteroscedasticity and non-linearity consistent estimators of the asymptotic variance. In this misspecfied setting, a tidy workflow is currently unavailable in R. By a *tidy workflow*, we mean a suite of functions for variance estimation, confidence interval computation, hypothesis testing, and diagnostics whose outputs are tailored for further exploration using the tidyverse R packages [13]. In this paper, we introduce our maar R package that provides such a tidy workflow for data analysis with ordinary least squares (OLS) linear regression. Our package is motivated, in large part, by the discussion of Buja et al. [1, 2]. Part of our motivation is also pedagogical, in making statisticians more aware of the consequences of model-based variance estimators for inference.

In the remaining part of the paper, we introduce the functions in our maar R package[1] and, when applicable, compare them with the lm() function. For illustrative purposes only, we use the *LA Homeless data* from [1] to demonstrate the maar package functionality.

As noted in [1], this dataset has 505 observations, each representing a sampled metropolitan LA County Census tract. It also has 7 numeric variables measuring different quantities of interest in each tract. For linear modeling purposes, the response variable (*StreetTotal*) is a count of the homeless people in each tract. There are six covariates for regression purposes. These include the Median Income of Households (*MedianInc ($1000)*), and the share of non-Caucasian residents (*PercMinority*). The remaining four covariates measure the proportion of the different types of lots in each tract (*PercCommercial*, *PercVacant*, *PercResidential* and *PercIndustrial*).

1.1 OLS under well-specification

Suppose we have regression data $(X_i, Y_i) \in \mathbb{R}^d \times \mathbb{R}$, $1 \le i \le n$. The well-specified (classical) linear model stipulates that (X_i, Y_i), $1 \le i \le n$ are independent and satisfy

$$Y_i = X_i^\top \beta_0 + \varepsilon_i, \tag{1}$$

where $\varepsilon_i | X_i \sim N(0, \sigma^2)$. This implies that $\mathbb{E}(Y_i|X_i) = X_i^\top \beta_0$ (linearity) and $\mathrm{Var}(Y_i|X_i) = \sigma^2$ (homoscedasticity). Usually one also assumes covariates to be non-stochastic. Under these assumptions, we get that the OLS estimator

$$\widehat{\beta} = \underset{\theta \in \mathbb{R}^d}{\arg\min} \sum_{i=1}^n (Y_i - X_i^\top \theta)^2,$$

has a normal distribution (conditional on $X_i, i \le n$):

$$\widehat{\beta} - \beta_0 \ \sim \ \mathcal{N}(0, \sigma^2 (\textstyle\sum_{i=1}^n X_i X_i^\top)^{-1}) \tag{2}$$

This distribution yields confidence intervals, hypothesis tests, and p-values for β_0. The R function lm() provides inference based on (2). For the LA Homeless data, the implementation of OLS along with its inference are as follows:

```
# import LA homeless data from source
lah_df <- readr::read_csv('http://www-stat.wharton
    .upenn.edu/~buja/STAT-961/Homeless_LA_by_
    Census_Tracts.csv')
# fit lm on LA Homeless data
mod_fit <- stats::lm(formula = StreetTotal ~ .,
                     data = lah_df)

# Summary diagnostics on fitted linear model
```

[1] The package name i.e. maar, is a derived abbreviation of *Models As Approximations in R*, based on the series of papers with a similar name [1, 2].

```
summary(mod_fit)

Call:
stats::lm(formula = StreetTotal ~ ., data = lah_df)

Residuals:
    Min      1Q  Median      3Q     Max
-148.10  -28.35  -10.87    8.18  870.85

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)
(Intercept)          0.76032   22.76688   0.033  0.97337
`MedianInc ($1000)` -0.18298    0.18735  -0.977  0.32921
PercVacant           4.62868    0.90060   5.140 3.96e-07 ***
PercMinority         0.12341    0.17603   0.701  0.48360
PercResidential     -0.05006    0.17130  -0.292  0.77021
PercCommercial       0.73727    0.27309   2.700  0.00718 **
PercIndustrial       0.90540    0.32131   2.818  0.00503 **
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 74.92 on 498 degrees of freedom
Multiple R-squared:  0.1331,    Adjusted R-squared:  0.1226
F-statistic: 12.74 on 6 and 498 DF,  p-value: 2.109e-13
```

As mentioned, the standard errors, t-scores, and the p-values reported in the summary above are probably invalid, because the linear model for LA Homeless data is not well-specified [1]. We will now discuss the properties of OLS under misspecification of the linear model. It should be mentioned here that we will only discuss misspecification of the linear model but require independence of the observations. Dependence will be part of our future work (see Section 7).

1.2 OLS under misspecification

If $(X_i, Y_i) \in \mathbb{R}^d \times \mathbb{R}, 1 \le i \le n$ are independent and identically distributed (i.i.d.), then the least squares estimator $\widehat{\beta}$ converges (in probability) to[2]

$$\beta^{\infty} := \underset{\theta \in \mathbb{R}^d}{\arg\min}\, \mathbb{E}\left((Y - X^{\top}\theta)^2\right)$$

where $(X, Y) \in \mathbb{R}^d \times \mathbb{R}$ is an independent copy of (X_i, Y_i). If the linear model (1) holds true, then $\beta^{\infty} = \beta_0$. Assuming only i.i.d. observations with finite moments, it is easy to obtain the normal limiting distribution:

$$\sqrt{n}(\widehat{\beta} - \beta^{\infty}) \xrightarrow{d} N(0, J^{-1}VJ^{-1})$$

where $J = \mathbb{E}(XX^{\top})$, and $V = \mathbb{E}\left(XX^{\top}(Y - X^{\top}\beta^{\infty})^2\right)$; see [1] for details. A similar limit theorem also holds when the observations are independent but not identically distributed (i.n.i.d.). The matrices J and V can be readily estimated by replacing the expectations by averages and β^{∞} by $\widehat{\beta}$ leading to the sandwich estimator $\widehat{J}^{-1}\widehat{V}\widehat{J}^{-1}$ [1]. Here

$$\widehat{J} = \frac{1}{n}\sum_{i=1}^{n} X_i X_i^{\top} \quad \text{and} \quad \widehat{V} = \frac{1}{n}\sum_{i=1}^{n} X_i X_i^{\top}(Y_i - X_i^{\top}\widehat{\beta})^2 \tag{3}$$

It can be shown that $\widehat{J}^{-1}\widehat{V}\widehat{J}^{-1}$ consistently estimates the asymptotic variance under i.i.d., and over-estimates the asymptotic variance under i.n.i.d. data [6]. It is noteworthy that the estimator $\widehat{J}^{-1}\widehat{V}\widehat{J}^{-1}$ converges to the model-based variance if the linear model (1) holds true.

Our implementation is aimed at appending the columns of the summary table from `lm()` by statistically valid quantities based on the sandwich variance estimator, the empirical bootstrap, and the multiplier bootstrap.

2 Guiding Design Principles: `maar`

Our goal is for the `maar` package to be a natural inferential `R` workflow for Ordinary Least Squares (OLS) modeling under model misspecification. To achieve this, we set ourselves the following four software design principles (**DP.1**–**DP.4**) when developing the `maar` package:

- **DP.1** Build `maar` model misspecification tools on top of existing `R` modeling objects used in a well-specified modeling workflow e.g. `lm()`.
- **DP.2** Use a vectorized and `tidyverse` `R` code style in both the `maar` package development and in the final inferential workflow.
- **DP.3** Write rigorously benchmarked and unit tested code.
- **DP.4** Use modern open-source best practices to ensure an inclusive package development environment.

We provide brief details on the design principles **DP.1**–**DP.4**, which also inform the structure of this paper. First, in **DP.1**, we note that OLS parameter estimation under both the well-specified and misspecified setting is identical. However, the key differences are the underlying assumptions driving estimation in each setting and the associated interpretability and inference of the estimated parameters. As such, we build our `maar` estimates using existing `R` statistical functions. For example, to perform OLS estimation and inference for linear models under model-misspecification, the user first creates an `lm()` object, thus maintaining the current well-specified workflow. This is emphasized with examples in Sections 3 and 4.

Second, in **DP.2** we emphasize the use of vectorized `R` and `tidyverse` coding style in developing the `maar` package, where possible. We believe this design principle allows `maar` package contribution to be more accessible to a larger number of users. Furthermore, almost all `maar` functions return `tibble` objects [8] for data, and all plots are `ggplot2` objects [12]. This ensures easy exploration using the `tidyverse` `R` pacakges e.g. `dplyr` [14]. For more details see Sections 3 and 4.

Third, in **DP.3** we seek to ensure that our `maar` functions are rigorously benchmarked and unit tested. A well defined benchmarking framework demonstrates how our code is optimized for computational efficiency. A detailed unit testing framework allows us to effectively perform statistical validation and error handling. See Section 5 for details.

Finally, in **DP.4** we note that the `maar` package is developed in an open-source manner that is inclusive towards new

[2] Note that β^{∞} ($\widehat{\beta}$ at $n = \infty$) is the population projection parameter.

users and contributors. To ensure this, and our efforts in making maar a long-term part of a statistician's computational toolkit, are described in Sections 5 and 6.

3 Variance estimation in maar

Under model misspecification, one of the main inferential tools in estimation of the model's parameters is accurate and efficient estimation of variance. Under the maar framework we have three options for this readily implemented as part of a tidy pipe (%>%) friendly workflow. These estimation methods are the sandwich estimator, and the empirical and multiplier bootstrap. We review the workflow and functionality behind each estimation method in turn in this section.

3.1 Sandwich Estimator

Under the 'models as approximations' framework, using only the i.i.d. and finite moments assumptions, the asymptotic variance of the OLS is given by the sandwich estimator [1, 2]: $\widehat{\text{Var}}(\widehat{\beta}) = n^{-1}\widehat{J}^{-1}\widehat{V}\widehat{J}^{-1}$, with $\widehat{J}$ and $\widehat{V}$ defined in (3). Then the standard error for the j^{th} coefficient is given by $\widehat{\text{SE}}_j(\widehat{\beta}) = n^{-1/2}(\widehat{J}^{-1}\widehat{V}\widehat{J}^{-1})_{jj}^{1/2}$. In the maar package, this and other inferential summary statistics can be readily computed through a one-line function as follows.

```
# summary based on sandwich std. errors
summary_sand <- mod_fit %>%
    maar::comp_sandwich_var()
```

The tidy output of this code as shown in Table 1 is in the form of a tibble[3] as typically achieved using the broom::tidy functions [9]. This makes it readily amenable to wrangling and visualization using the tidyverse packages, in line with **DP.2**.

Term	$\widehat{\beta}_j$	SE_{lin}	SE_{sand}	t_{sand}	p_{sand}
(Intercept)	0.76	22.77	15.91	0.05	0.96
MedianInc ($1000)	-0.18	0.19	0.10	-1.75	0.08
PercVacant	4.63	0.90	1.29	3.60	0.00
PercMinority	0.12	0.18	0.16	0.76	0.45
PercResidential	-0.05	0.17	0.11	-0.46	0.64
PercCommercial	0.74	0.27	0.39	1.90	0.06
PercIndustrial	0.91	0.32	0.58	1.56	0.12

Table 1. Sandwich Estimator Standard Errors

The asymptotic $(1-\alpha)$ confidence interval for the same coefficient is given by

$$\widehat{\text{CI}}_j = \left[\widehat{\beta}_j - z_{\alpha/2}\sqrt{\frac{(\widehat{J}^{-1}\widehat{V}\widehat{J}^{-1})_{jj}}{n}}, \widehat{\beta}_j + z_{\alpha/2}\sqrt{\frac{(\widehat{J}^{-1}\widehat{V}\widehat{J}^{-1})_{jj}}{n}}\right]$$

where $z_{\alpha/2}$ is the quantile of the standard normal. This can readily computed and plotted in the R framework. In Figure 1, we show the 95% confidence intervals for the OLS estimates based on the lm() and the sandwich standard errors.

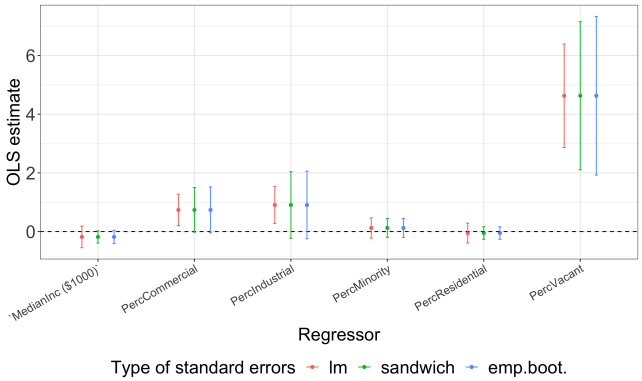

Figure 1. 95% confidence intervals: comparison of lm(), sandwich, and empirical bootstrap.

3.2 Empirical Bootstrap

Under the independence and finite moment assumptions, another method for the estimation of variance is the empirical bootstrap. The m-out-of-n empirical bootstrap consists of two consecutive steps. In the first step, for $1 \le b \le B$, we resample with replacement m observations from the original data set and obtain the OLS estimate $\widehat{\beta}_b^*$ on the bootstrapped data set. In the second step, we compute m times the sample covariance of $\widehat{\beta}_b^*, 1 \le b \le B$ as an estimate of $\widehat{J}^{-1}\widehat{V}\widehat{J}^{-1}$, the asymptotic variance of $\sqrt{n}(\widehat{\beta} - \beta^{\infty})$. The justification for this method follows from the fact that, conditional on the original data $\{(\mathbf{X}_i, Y_i)\}_{i=1}^n$, we have

$$\sqrt{m}(\widehat{\beta}_b^* - \widehat{\beta}) \xrightarrow{d} \mathcal{N}(0, \widehat{J}^{-1}\widehat{V}\widehat{J}^{-1}) \text{ as } m \to \infty$$

and thus we can show that

$$\frac{m}{B-1}\sum_{b=1}^{B}(\widehat{\beta}_b^* - \overline{\widehat{\beta}^*})(\widehat{\beta}_b^* - \overline{\widehat{\beta}^*})^T \xrightarrow{p} \widehat{J}^{-1}\widehat{V}\widehat{J}^{-1} \text{ as } B \to \infty.$$

Here $\overline{\widehat{\beta}^*} = B^{-1}\sum_{b=1}^{B}\widehat{\beta}_b^*$. Therefore, as B increases, the covariance of the OLS estimates on the bootstrapped data sets converges in probability to $\widehat{J}^{-1}\widehat{V}\widehat{J}^{-1}$. In maar, the n-out-of-n empirical bootstrap estimates for the LA Homeless data can be obtained as follows.

```
# obtain OLS estimates on bootstrapped data sets
coef_emp_boot <- mod_fit %>%
maar::comp_empirical_bootstrap(B = 1000, m = 505)
```

Using the distribution of the OLS estimates on the bootstrapped data sets, we can check whether sample size is large enough to justify the asymptotic regime. In maar, the user can compare the distribution of the estimates via a normal Q-Q plot as follows (with tidy output in Figure 2).

```
maar::qqnorm_bootstrap(coef_emp_boot)
```

[3] Throughout this paper the tidy tibble outputs from maar variance estimation functions are formatted as LaTeX tables for visual clarity.

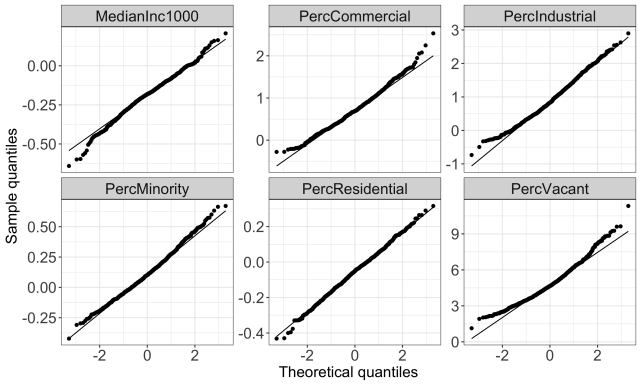

Figure 2. Normal Q-Q plot - Empirical bootstrap

We can obtain a `tidy` summary for the results of the bootstrap as follows:

```
# summary based on emp. bootstrap std. errors
se_emp <-  mod_fit %>% maar::comp_bootstrap_
    summary(boot_out = coef_emp_boot,
    boot_type = 'emp')
```

The `tidy` output of this code is displayed in Table 2.

Term	$\widehat{\beta}_j$	SE_{boot}	t_{boot}	p_{boot}
(Intercept)	0.76	16.22	0.05	0.04
MedianInc ($1000)	-0.18	0.11	-1.62	0.90
PercCommercial	0.74	0.37	2.00	0.95
PercIndustrial	0.91	0.57	1.59	0.89
PercMinority	0.12	0.17	0.74	0.55
PercResidential	-0.05	0.11	-0.45	0.34
PercVacant	4.63	1.36	3.42	0.99

Table 2. Empirical Bootstrap Standard Errors

Finally, we can obtain 95% confidence intervals for the coefficients via the percentile method as follows:

```
# 95% CI for OLS estimates based on emp. boot.
maar::comp_conf_int_bootstrap(coef_emp_boot,
     probs = c(0.025, 0.975))
```

The resulting confidence intervals are presented in Figure 1. This is to allow for easier comparison to the corresponding confidence intervals from `lm()`, and the sandwich estimator (per Section 3.1).

3.3 Multiplier Bootstrap

The main disadvantage of the empirical bootstrap is that it requires the computation of the OLS estimate B times. Furthermore, the bootstrapped design matrix might be singular. The multiplier bootstrap is an alternative method that helps overcome these difficulties. The method consists of estimating B times

$$\widehat{\beta}_b^* = \widehat{\beta} + \frac{1}{n}\sum_{i=1}^{n} w_{i,b}\widehat{J}^{-1}X_i(Y_i - X_i^\top\widehat{\beta}), \quad 1 \le b \le B. \tag{4}$$

The weights $\{w_{i,b}\}_{i=1}^n$ are independent with $\mathbb{E}(w_{i,b}) = 0$ and $\mathbf{Var}(w_{i,b}) = 1$, for each $1 \le i \le n$, $1 \le b \le B$. These can be computed as follows:

```
# obtain OLS estimates via multiplier bootstrap
coef_mult_boot <- mod_fit %>%
 maar::comp_multiplier_bootstrap_var(B = 1000,
        weights_type = "rademacher")
```

The distribution of the weights in (4) is specified through the `weights_type` option. Following the boottest package [10] in Stata, we provide the user with four prespecified values: `rademacher`, `mammen`, `webb`, `gaussian`. The `rademacher` weights are sampled from the two point Rademacher distribution which has takes values $\{-1, 1\}$, with equal probability. The `mammen` weights are sampled from the two point Mammen distribution which takes values $(\mp\sqrt{5}+1)/2$, with probabilities $(\sqrt{5}\pm 1)/(2\sqrt{5})$, respectively. The `webb` weights are sampled from the six point Webb distribution which takes values $\{\pm\sqrt{3/2}, \pm\sqrt{1/2}, \pm 1\}$, with equal probability. Finally, the `gaussian` weights are sampled from the Standard Gaussian distribution.

In `maar`, the output of the multiplier bootstrap in (4) can be analyzed through the same functions that were described in Section 3.2 for analyzing the regression coefficients obtained via empirical bootstrap.

3.4 Putting it All Together: Summary of Results

In Sections 3.1 and 3.3 we demonstrated how the `maar` package can be used to calculate `tidy` format summary statistics using the sandwich estimator, empirical bootstrap, and multiplier bootstrap. If the user requires a consolidated table of these summary statistics, they can call the wrapper function `maar::summary_var` as follows:

```
# Summary for sandwich and empirical bootstrap
se_comb <- mod_fit %>% maar::summary_var(
  se_boot_emp = TRUE, se_boot_mult = FALSE)
```

In this code example, the user simply passes in the fitted OLS model i.e. `mod_fit`. This produces the sandwich estimator output from Table 1 by default. By passing in `TRUE` for the empirical bootstrap the function would also append columns from Table 2 using default empirical bootstrap parameters. By passing in `FALSE` for the multiplier bootstrap ensures that this would not be run and appended, in this case.

4 Diagnostic Tools for Well-Specification

When the regression model is correctly specified, the regression coefficients do not depend on the distribution of $\mathbf{X}$ (see [2, Proposition 4]). Equivalently, variations in the OLS estimates under reweighting of $\mathbf{X}$ would suggest that the model is not well-specified. In `maar`, we have implemented the graphical model diagnostics tools through which the statistician can investigate whether the OLS assumptions are broken. These tools, which were proposed by [2], supplement of the model diagnostics returned by `lm()`.

The key idea behind these model diagnostics is to study how the OLS estimates vary under arbitrary reweighting of $\mathbf{X}$. It is hard visualize such a diagnostic with arbitrary reweighting and hence we restrict (as in [2]) to reweighting along certain regressors. Although restrictive, the resulting plots provide intuitive understanding of misspecification in data. For reweighting along the j-th regressor $X(j)$,[4] $1 \le j \le d$, we do the following two operations:

DT.1 Select a grid of K values (or centers) $c_{1,j}, \ldots, c_{K,j}$ on the support of $X(j)$.

DT.2 For each center $c_{\ell,j}, 1 \le \ell \le K$, compute the weighted least squares estimator

$$\widehat{\beta}_\ell = \arg\min_{\theta \in \mathbb{R}^d} \sum_{i=1}^{n} (Y_i - X_i^\top \theta)^2 e^{-(X_i(j) - c_{\ell,j})^2/(2\gamma^2)}.$$

Intuitively, this is the least squares estimator that gives more weight to observations that have j-th regressor values close to $c_{\ell,j}$.

The procedure consists of fitting K regressions. This will give us one curve and for comparison, we also run the same procedure on bootstrapped data leading to $\widehat{\beta}^*_\ell, 1 \le \ell \le K$. These steps can be run using the following function.

```
# Regressions based on reweighted coefficients
coef_rwgt <- mod_fit %>% maar::comp_coef_rwgt(
  B = 300, terms_to_rwgt = names(lah_df)[-1])
```

In the package, we have developed two types of grids for **DT.1**. The default grid of reweighting centers in **DT.1** is based on the deciles of $X(j)$ as in [2]. Alternatively, the user can select a grid of evenly spaced values between $X_{(1)}(j)$ and $X_{(n)}(j)$. For **DT.2**, `maar` uses $\gamma = (\sum_{i=1}^n (X_i(j) - \overline{X}(j))^2/(n-1))^{1/2}$. To assist the statistician in interpreting the results of the refitting procedure, [2] have proposed using the following three graphical model diagnostics tools.

- **focal slope**: Consider changes in coefficient of interest $\widehat{\beta}_\ell(k)$ to the reweighting of each of the regressors $X(j)$, $1 \le j \le d$; this provides insights into the interactions between regressor $X(k)$ and all other regressors $X(j)$.
- **nonlinearity detection**: Consider changes in $\widehat{\beta}_\ell(k)$ to the reweighting of its own regressor $X(k)$; this provides insights into marginal nonlinear behaviors of the response surface.
- **focal reweighting variable**: Consider changes in all coefficients $\widehat{\beta}_\ell(j)$ to the reweighting of a given regressor $X(k)$.

For further information and interpretations of these types of diagnostics, see [2, Section 5]. The three types of graphical diagnostics can be obtained in `maar` as follows.

```
# focal slope for 'PercVacant' regressor
mod_fit %>% maar::focal_slope(coef_rwgt, '
    PercVacant')
```

[4]For any vector $x \in \mathbb{R}^d$, we write $x(j)$ to denote its j-th coordinate.

```
# run nonlinearity detection
mod_fit %>% maar::nonlinearity_detect(coef_rwgt)
# focal reweighting for 'PercVacant' regressor
mod_fit %>% maar::focal_rwgt_var(coef_rwgt, '
    PercVacant')
```

A sample output of the focal slope diagnostics is shown in Figure 3. These plots show how the OLS estimate of the prevalence of vacant lots (*PercVacant*) in our running example for 300 bootstrapped data sets (gray lines) vary under reweighting of all regressors (plot panel titles). The changes observed in the means of the estimates (black lines in the middle of each panel) indicate that the model is unlikely to be well-specified. In particular, these plots suggest the presence of an interaction between *PercVacant* and other regressors such as *PercMinority*, *PercResidential*, and *MedianInc ($1000)*. In the plot in the bottom right corner, which depicts the estimate of *PercVacant* under reweighting of its own regressor, we observe that the reweighted estimates are far larger than the unweighted ones on the original data (blue dashed line).

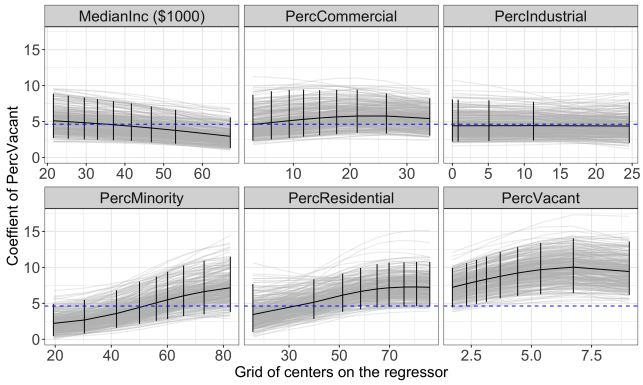

Figure 3. Focal slope: the x-axis represents the grid mentioned in **DT.1**

5 Benchmarking and Unit Testing

In developing the `maar` package, as noted in **DP.3**, we have used a comprehensive benchmarking and testing framework. We conduct high-precision benchmarking of the `maar` package functionality using the `bench` package [4].

Similarly, we test our statistical functionality and error handling using the `testthat` package [11]. We conduct unit testing to validate the statistical accuracy of each of our core function outputs against a differently coded alternative. We also test that our expected errors are handled gracefully. Moreover, testing is built into the development cycle through cross-platform continuous integration using Github Actions.

6 Community and Inclusion

Our goal in developing the `maar` package is for it to be the standard `R` workflow for doing inference under the model misspecification framework described in [1, 2]. To enable

active usage and open-source development we have implemented industry standard best practices to ensure an inclusive community for all users and contributors.

We use the `tidyverse` contributing guide and code of conduct to ensure that new contributors have ample guidance on our development practices. We use `roxygen2` for detailed function documentation. The documentation is automatically rendered into the official `maar` package website using Github pages. On our package website we use `vignettes` to illustrate the `maar` inferential functionality via case studies on simulated and real datasets.

7 Conclusion and Future Work

In this paper, we review the `maar` package functionality to perform OLS inference under model misspecification. The `maar` package is designed to implement the key inferential ideas from [1, 2] in a tidy R workflow.

The `maar` package is built on several core software design principles (see **DP.1**-**DP.4**). These design principles affect the day-to-day inferential workflow under model misspecification. They also influence the way in which users can contribute to the existing `maar` functionality. The main tools for statistical inference under model misspecification are based on different variance estimation methods. The three variance estimation methods in `maar` include the sandwich estimator, standard errors computed using the m-out-of-n empirical bootstrap, and the multiplier bootstrap (see Section 3). To help diagnose issues under the misspecified setting there are a number visualization tools readily accessible to the `maar` user. These currently include the focal slope graphs of regressor variables [2] and Q-Q plots.

To ensure that the `maar` package is used and contributed to by the wider open-source statistical community, we ensure that the code is rigorously benchmarked and unit tested. The package is developed using thorough and well cited documentation.

Our `maar` package is still in active development and the final version we envision includes more tools related to misspecification. In detail, future work includes other variants of heteroscedasticity consistent variance estimators, inference under dependent/time series data [15], bootstrap/resampling under dependence, and adding more pedagogical tools comparing model-based and model-free inferences. We intend to demonstrate new package functionality through more illustrative case studies (as `vignettes`). Finally, we want to extend the functions to GLMs, the Cox proportional hazards model, and inference after data exploration e.g. PoSI [7].

References

[1] Andreas Buja, Lawrence Brown, Richard Berk, Edward George, Emil Pitkin, Mikhail Traskin, Kai Zhang, and Linda Zhao. 2019. Models as approximations I: consequences illustrated with linear regression. *Statist. Sci.* 34, 4 (2019), 523–544.

[2] Andreas Buja, Lawrence Brown, Arun Kumar Kuchibhotla, Richard Berk, Edward George, and Linda Zhao. 2019. Models as approximations II: A model-free theory of parametric regression. *Statist. Sci.* 34, 4 (2019), 545–565.

[3] John Fox and Sanford Weisberg. 2019. *An R Companion to Applied Regression* (third ed.). Sage, Thousand Oaks CA.

[4] Jim Hester. 2020. *bench: High Precision Timing of R Expressions.* https://CRAN.R-project.org/package=bench R package version 1.1.1.

[5] Peter J Huber. 1967. The behavior of maximum likelihood estimates under nonstandard conditions. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, Vol. 1. University of California Press, 221–233.

[6] Arun K Kuchibhotla, Lawrence D Brown, and Andreas Buja. 2018. Model-free study of ordinary least squares linear regression. *arXiv preprint arXiv:1809.10538* (2018).

[7] Arun K. Kuchibhotla, Lawrence D. Brown, Andreas Buja, Junhui Cai, Edward I. George, and Linda H. Zhao. 2020. Valid post-selection inference in model-free linear regression. *Ann. Statist.* 48, 5 (2020), 2953–2981.

[8] Kirill Müller and Hadley Wickham. 2020. *tibble: Simple Data Frames.* https://CRAN.R-project.org/package=tibble R package version 3.0.4.

[9] David Robinson, Alex Hayes, and Simon Couch. 2020. *broom: Convert Statistical Objects into Tidy Tibbles.* https://CRAN.R-project.org/package=broom R package version 0.7.2.

[10] David Roodman, Morten Ørregaard Nielsen, James G MacKinnon, and Matthew D Webb. 2019. Fast and wild: Bootstrap inference in Stata using boottest. *The Stata Journal* 19, 1 (2019), 4–60.

[11] Hadley Wickham. 2011. testthat: Get Started with Testing. *The R Journal* 3 (2011), 5–10.

[12] Hadley Wickham. 2016. *ggplot2: Elegant Graphics for Data Analysis.* Springer-Verlag New York. https://ggplot2.tidyverse.org

[13] Hadley Wickham, Mara Averick, Jennifer Bryan, Winston Chang, Lucy D'Agostino McGowan, Romain François, Garrett Grolemund, Alex Hayes, Lionel Henry, Jim Hester, Max Kuhn, Thomas Lin Pedersen, Evan Miller, Stephan Milton Bache, Kirill Müller, Jeroen Ooms, David Robinson, Dana Paige Seidel, Vitalie Spinu, Kohske Takahashi, Davis Vaughan, Claus Wilke, Kara Woo, and Hiroaki Yutani. 2019. Welcome to the tidyverse. *Journal of Open Source Software* 4, 43 (2019), 1686.

[14] Hadley Wickham, Romain François, Lionel Henry, and Kirill Müller. 2020. *dplyr: A Grammar of Data Manipulation.* https://CRAN.R-project.org/package=dplyr R package version 1.0.2.

[15] Achim Zeileis. 2004. Econometric Computing with HC and HAC Covariance Matrix Estimators. *Journal of Statistical Software* 11, 10 (2004), 1–17.

[16] Achim Zeileis, Susanne Köll, and Nathaniel Graham. 2020. Various Versatile Variances: An Object-Oriented Implementation of Clustered Covariances in R. *Journal of Statistical Software* 95, 1 (2020), 1–36.