A User’s Guide to Approximate Randomization Tests
with a Small Number of Clusters*

Yong Cai
Department of Economics
Northwestern University
yongcai2023@u.northwestern.edu

Ivan A. Canay
Department of Economics
Northwestern University
icanay@northwestern.edu

Deborah Kim
Department of Economics
Northwestern University
deborahkim@u.northwestern.edu

Azeem M. Shaikh
Department of Economics
University of Chicago
amshaikh@uchicago.edu

February 17, 2021

Abstract

This paper provides a user’s guide to the general theory of approximate ran-
domization tests developed in Canay et al. (2017a) when specialized to linear re-
gressions with clustered data. Such regressions include settings in which the data is
naturally grouped into clusters, such as villages or repeated observations over time
on individual units, as well as settings with weak temporal dependence, in which
pseudo-clusters may be formed using blocks of consecutive observations. An impor-
tant feature of the methodology is that it applies to settings in which the number
of clusters is small – even as small as five. We provide a step-by-step algorithmic
description of how to implement the test and construct confidence intervals for the
quantity of interest. We additionally articulate the main requirements underlying
the test, emphasizing in particular common pitfalls that researchers may encounter.
Finally, we illustrate the use of the methodology with two applications that fur-
ther elucidate these points: one to a linear regression with clustered data based on
Meng et al. (2015) and a second to a linear regression with temporally dependent
data based on Munyo and Rossi (2015). The required software to replicate these
empirical exercises and to aid researchers wishing to employ the methods elsewhere
is provided in both R and Stata.

KEYWORDS: Randomization tests, linear regression, clustered data, time series

JEL classification codes: C12, C14

*The research of the fourth author is supported by NSF Grant SES-1530661.
1 Introduction

This paper provides a user’s guide to the general theory of approximate randomization tests (ARTs) developed in Canay et al. (2017a) when specialized to linear regressions with clustered data. Here, clustered data refers to data that may be grouped so that there may be dependence within each cluster, but distinct clusters are approximately independent in a way to be made precise below. Such data is remarkably common, including not only data that are naturally grouped into clusters, such as villages or repeated observations over time on individual units, but also data with weak temporal dependence, in which pseudo-clusters may be formed using blocks of consecutive observations. An important feature of the methodology is that it applies to commonly encountered settings in which the number of clusters is small – even as small as five. In this respect, the proposed methodology contrasts sharply and meaningfully with many commonly employed methods for inference in such settings. We briefly elaborate on this point in our discussion of related literature below.

A principal goal of this paper is to make the general theory developed in Canay et al. (2017a) more accessible by providing a step-by-step algorithmic description of how to implement the test and construct confidence intervals for the quantity of interest in these types of settings. While we focus in Section 2 below on what we view as the most natural implementation of the test, we further show that it is numerically equivalent to an alternative implementation. This equivalence yields further insights into the main requirements underlying the test, which are clarified in Section 3. These requirements essentially demand that the quantity of interest is suitably estimable cluster-by-cluster. As discussed further in Section 3, when this is not satisfied, a researcher need not conclude that it is not possible to exploit the results in Canay et al. (2017a). Instead, several remedies are possible, including clustering more coarsely or changing the specification to ensure that this requirement is satisfied. We provide two applications that further elucidate these points: one to a linear regression with clustered data based on Meng et al. (2015) and a second to a linear regression with temporally dependent data based on Munyo and Rossi (2015). The required software to replicate these empirical exercises and to aid researchers wishing to employ the methods elsewhere is provided in both R and Stata.¹

The methodology described in this paper is part of a large and active literature on inference with clustered data. Following Bertrand et al. (2004), researchers are acutely aware of the need to adjust inferences appropriately to account for this sort of dependence. Many of the most commonly employed methods for doing so, however, are inadequate for the unusually common situation in which the number of clusters is small. Conventional wisdom suggests that the number of clusters is small when it is

¹The Stata and R packages ARTs can be downloaded from http://sites.northwestern.edu/iac879/software/.
less than forty. For example, the method described in Liang and Zeger (1986), which has enjoyed considerable popularity due to its availability in software packages such as Stata, is widely acknowledged to perform poorly when this rule-of-thumb is not satisfied. Similarly, the cluster wild bootstrap described in Cameron et al. (2008) requires either a sufficiently large number of clusters or, as shown by Canay et al. (2020), stringent homogeneity across clusters, to perform reliably. As explained further in Section 3, the methods developed in Canay et al. (2017a) and described in this paper, require neither a large number of clusters nor such homogeneity across clusters. We note that the methods by Ibragimov and Müller (2010, 2016), which are closely related to the ones described here, also do not require such restrictions, but are generally less powerful and permit testing a less rich variety of hypotheses. See Canay et al. (2017a) for further discussion of these points as well as Conley et al. (2018) for an insightful and thorough review of the related literature more broadly.

The remainder of this paper is organized as follows. In Section 2, we first formalize the setting and establish some notation. We then describe the implementation of approximate randomization tests (ARTs) in an algorithmic fashion as well as how to invert these tests to construct confidence intervals for the quantity of interest. In Section 3, we articulate the main requirements underlying the tests and discuss remedies for cases where these requirements are not satisfied. Our two empirical applications are contained in Section 4. Finally, we provide some concluding remarks in Section 5.

2 Review of ARTs in regression models

We start by reviewing the inference approach proposed by Canay et al. (2017a) in the context of a linear regression model with clustered data. In order to do so, we index clusters by $j \in J \equiv \{1, \ldots, q\}$ and units in the $j$th cluster by $i \in I_{n,j} \equiv \{1, \ldots, n_j\}$. We also denote by $n = \sum_{j=1}^{q} n_j$ the total number of observations. The observed data consists of an outcome of interest, $Y_{i,j}$, and a vector of covariates, $Z_{i,j} \in \mathbb{R}^{d_z}$, that are related through the equation

$$Y_{i,j} = Z_{i,j}' \beta + \epsilon_{i,j}, \quad (1)$$

where $\beta \in \mathbb{R}^{d_z}$ are unknown parameters and our requirements on $\epsilon_{i,j}$ are explained below in Section 3. Our goal is to test

$$H_0 : c' \beta = \lambda \quad \text{vs.} \quad H_1 : c' \beta \neq \lambda, \quad (2)$$

for given values of $c \in \mathbb{R}^{d_z}$ and $\lambda \in \mathbb{R}$, at level $\alpha \in (0, 1)$. An important special case of this framework is a test of the null hypothesis that a particular component of $\beta$ equals a given value, i.e.,

$$H_0 : \beta_\ell = \lambda \quad \text{vs.} \quad H_1 : \beta_\ell \neq \lambda,$$
for some $\ell \in \{1, \ldots, d_z\}$, by simply setting $c$ to be a standard unit vector with a one in the $\ell$th component and zeros otherwise. More generally, the approach we describe below extends immediately to the case where the hypothesis of interest involves multiple elements of $\beta$, in which case the test becomes

$$H_0 : R\beta = \Lambda \quad \text{vs.} \quad H_1 : R\beta \neq \Lambda ,$$

for a given $p \times d_z$-dimensional matrix $R$ and $p$-dimensional vector $\Lambda$, at level $\alpha \in (0, 1)$.

ARTs were developed more generally in Canay et al. (2017a) and admit a variety of different applications that go beyond the linear model considered here. For example, the method accommodates non-linear models, non-linear hypotheses, or even applications that go beyond inference with a small number of clusters (e.g., Canay and Kamat (2018) develop a variation that applies to inference in the regression discontinuity design). Here, we abstract away from the generality of the method and focus on the steps needed to use ARTs to test the null hypothesis in (2) in the context of the model in (1).

### 2.1 How to implement ARTs

The most straightforward way to test the hypotheses in (2) via ARTs is by following the steps described in Algorithm 2.1 below.

**Algorithm 2.1 (ARTs via within-cluster estimates).** This implementation of ARTs involves the following steps:

**Step 1:** For each cluster $j \in J$, run an ordinary least squares regression of $Y_{i,j}$ on $Z_{i,j}$ using the $n_j$ observations in cluster $j$. Denote the corresponding estimators of $\beta$ by

$$\{\hat{\beta}_{n,j} : j \in J\} .$$

**Step 2:** For each $j \in J$, define the random variables

$$S_{n,j} \equiv \sqrt{n_j}(c^\prime \hat{\beta}_{n,j} - \lambda) .$$

**Step 3:** Compute the sample mean and sample standard deviation of $\{S_{n,j} : j \in J\}$,

$$\bar{S}_n \equiv \frac{1}{q} \sum_{j=1}^{q} S_{n,j} \quad \text{and} \quad \hat{\sigma}_S \equiv \sqrt{\frac{1}{q-1} \sum_{j=1}^{q} (S_{n,j} - \bar{S}_n)^2} ,$$

and then construct the test statistic

$$T_n = \frac{\bar{S}_n}{\hat{\sigma}_S / \sqrt{q}} .$$
Step 4: Let $G = \{1, -1\}^q$, so $g = (g_1, \ldots, g_q) \in G$ is simply a $q$-dimensional vector with elements $g_j$ being either 1 or $-1$. For any element $g \in G$ define

$$S_n^*(g) = \frac{1}{q} \sum_{j=1}^q g_j S_{n,j} \quad \text{and} \quad \hat{\sigma}_g^2(g) = \sqrt{\frac{1}{q-1} \sum_{j=1}^q (g_j S_{n,j} - S_n^*(g))^2},$$

and then construct the test statistic

$$T_n^*(g) = \frac{|S_n^*(g)|}{\hat{\sigma}_g(g) / \sqrt{q}}.$$  

(8)

Step 5: Compute the $1 - \alpha$ quantile of $\{T_n^*(g) : g \in G\}$ as

$$\hat{c}_n(1 - \alpha) \equiv \inf \left\{ u \in \mathbb{R} : \frac{1}{|G|} \sum_{g \in G} I\{T_n^*(g) \leq u\} \geq 1 - \alpha \right\}.$$ 

(9)

Step 6: Compute the test as

$$\phi_n \equiv I\{T_n > \hat{c}_n(1 - \alpha)\},$$

(10)

where $T_n$ is as in (6) and $\hat{c}_n(1 - \alpha)$ is as in (9). The associated p-value is

$$\hat{p}_n \equiv \frac{1}{|G|} \sum_{g \in G} I\{T_n^*(g) \geq T_n\},$$

(11)

where $T_n^*(g)$ is as in (8).

Algorithm 2.1 involves six steps that are easy to implement from a computational standpoint, but some of the steps deserve some clarification. Step 1 involves $q$ within-cluster regressions that lead to $q$ estimates of $\beta$. This essentially demands that the parameter $\beta$ is identified cluster-by-cluster, and may fail to hold if some of the variables in the vector $Z_{i,j}$ are constant within cluster. We discuss possible remedies for this problem in Section 3 and illustrate their use in one of the applications in Section 4. An important feature of the method is that from Step 2 onwards, the original data is no longer needed as all the calculations only involve the $q$ estimators of the parameter $\beta$ obtained in Step 1. Step 3 defines a type of $t$-statistic that is appropriate for the null hypothesis in (2). If the null hypothesis of interest is the one in (3), then a Wald-type test statistic could be used instead, i.e.,

$$T_{n}^{\text{wald}} \equiv q S_n' \hat{\Sigma}^{-1} S_n,$$ 

(12)
where
\[ S_{n,j} \equiv \sqrt{n}(R\hat{\beta}_{n,j} - \Lambda) \quad \text{and} \quad \Sigma_S \equiv \frac{1}{q} \sum_{j=1}^{q} S_{n,j}S_{n,j}' . \]

Notably, Step 4 does not require one to recompute the estimates of \( \beta \). It rather uses the \( q \) estimates from Step 1 and applies sign changes to the \( q \)-dimensional vector \( \{S_{n,j} : j \in J\} \).

Since the cardinality of \( G \) is \( |G| = 2^q \), it exceeds 2,000 when \( q > 10 \) and in such cases it may be convenient to use a stochastic approximation without affecting the properties of the test (see Canay et al., 2017a, Remark 2.2). Formally, in this case we let
\[ \hat{G} \equiv \{g^1, \ldots, g^B\} , \]
where \( g^1 = (1, \ldots, 1) \) is the identity vector and \( g^b = (g^b_1, \ldots, g^b_q) \), for \( b = 2, \ldots, B \), are i.i.d. Rademacher random variables; i.e., each \( g^b_j \) equals \( \pm 1 \) with equal probability. We then implement Algorithm 2.1 with \( \hat{G} \) replacing \( G \) everywhere and set \( B = 1,000 \) (or any other reasonably large number chosen by the analyst). Step 5 requires computing the \( 1 - \alpha \) quantile of \( \{T_n(g) : g \in G\} \), which can be typically obtained by sorting the values of \( \{T_n(g) : g \in G\} \) and then taking the \( \lceil |G|(1 - \alpha) \rceil \)th highest element in the ordered list. Thus, if we denote the ordered values of \( \{T_n(g) : g \in G\} \) by
\[ T_n^{(1)} \leq T_n^{(2)} \leq \cdots \leq T_n^{(B)} , \]
then we may define \( \hat{c}_n(1 - \alpha) \) in (9) as \( \hat{c}_n(1 - \alpha) = T([|G|(1 - \alpha)]) \). This suggests that the test may have no power for very low values of \( q \). For example, when \( \alpha = 10\% \), this problem arises if \( q \leq 4 \). For \( q = 5 \) the test already has non-trivial power and is only slightly conservative under the null. Similarly, when \( \alpha = 5\% \) the test has non-trivial power for any \( q \geq 6 \). Step 6 is straightforward and it provides both the test \( \phi_n \) and the \( p \)-value \( \hat{p}_n \).

### 2.2 How to compute confidence intervals

We now discuss how to compute confidence sets for the parameter \( c'\beta \) by test inversion. As before, a particularly important case is when \( c \) selects the \( \ell \)th component of \( \beta \) and then the confidence set is simply a confidence interval for \( \beta_{\ell} \). The idea behind test inversion is straightforward: form confidence sets by collecting all values of \( c'\beta \) that cannot be rejected by a test at level \( \alpha \). Formally, for the test \( \phi_n \) in (10) we define
\[ \hat{C}_n = \{\lambda : \phi_n = 0 \quad \text{when testing} \quad H_0 : c'\beta = \lambda\} . \]

In an asymptotic framework where \( n \to \infty \) while \( q \) remains fixed, Canay et al. (2017a) show that \( \phi_n \) is asymptotically level \( \alpha \) under \( H_0 \). It follows from that result that, by construction, \( \hat{C}_n \) covers \( c'\beta \) with probability at least equal to \( 1 - \alpha \) asymptotically. The
set \( \hat{C}_n \) requires implementing Algorithm 2.1 for all possible values of \( \lambda \in \mathbb{R} \). This may be cumbersome from a computational point of view and may be not particularly easy to report due to spurious rejections under \( H_0 \) (that happen with probability \( \alpha \)). We therefore recommend reporting the convex hull of \( \hat{C}_n \), denoted below by \( C_n \), which is easier to compute and interpret. Let \( L_n \) be the smallest value of \( \lambda \) that cannot be rejected by \( \phi_n \) and \( U_n \) the highest value of \( \lambda \) that cannot be rejected by \( \phi_n \). We then define \( C_n \) as

\[
C_n = [L_n, U_n].
\]  

(15)

By construction the value \( \hat{\lambda} = \frac{1}{q} \sum_{j=1}^{q} c' \hat{\beta}_{n,j} \), where \( \{ \hat{\beta}_{n,j} : j \in J \} \) is defined in Step 1 of Algorithm 2.1, is never rejected. Finding \( L_n \) and \( U_n \) is then typically straightforward by means of a bisection algorithm once the analyst determines a small enough and a large enough value of \( \lambda \) for which the null in (2) is rejected. In the applications of Section 4 we report \( C_n \) with \( L_n \) and \( U_n \) computed via the bisection algorithm described below in Algorithm 2.2.

Algorithm 2.2 (ART-based confidence intervals for \( c' \beta \)). For \( \{ \hat{\beta}_{n,j} : j \in J \} \) as defined in Step 1 of Algorithm 2.1, the construction of the confidence interval involves the following steps:

**Step 1:** Find the minimum value of \( \tau \in \{1, 2, 3, \ldots \} \) such that \( \phi_n \) in (10) rejects the null \( H_0 \) in (2) with \( \lambda = \hat{\lambda} - \tau s_{\hat{\lambda}} \) where

\[
\hat{\lambda} = \frac{1}{q} \sum_{j=1}^{q} c' \hat{\beta}_{n,j} \quad \text{and} \quad s_{\hat{\lambda}} = \sqrt{\frac{1}{q} \sum_{j=1}^{q} (c' \hat{\beta}_{n,j} - \hat{\lambda})^2}. \]

(16)

Denote this value of \( \lambda \) by \( \lambda^{(0),r}_i \). Repeat for \( \lambda = \hat{\lambda} + \tau s_{\hat{\lambda}} \) and denote this value of \( \lambda \) by \( \lambda^{(0),r}_u \). Finally, set \( \lambda^{(0),a}_i = \lambda^{(0),a}_u = \hat{\lambda} \).

**Step 2:** In a while loop at iteration \( k \geq 1 \), let

\[
\lambda^{(k)}_i = \frac{1}{2} \left( \lambda^{(k-1),r}_i + \lambda^{(k-1),a}_i \right).
\]

Test \( H_0 \) in (2) for \( \lambda = \lambda^{(k)}_i \) and set

\[
\lambda^{(k),r}_i = \lambda^{(k)}_i \quad \text{and} \quad \lambda^{(k),a}_i = \lambda^{(k-1),a}_i \quad \text{when} \quad \phi_n = 1
\]

or

\[
\lambda^{(k),r}_i = \lambda^{(k-1),r}_i \quad \text{and} \quad \lambda^{(k),a}_i = \lambda^{(k)}_i \quad \text{when} \quad \phi_n = 0.
\]

For a set tolerance level \( \text{tol} \) and maximum number of iterations \( \text{max} \), if

\[
\frac{1}{2} \left| \lambda^{(k-1),r}_i - \lambda^{(k-1),a}_i \right| < \text{tol} \quad \text{or} \quad k = \text{max},
\]

then stop.
stop the algorithm and set $L_n = \lambda_1^{(k)_a}$. Otherwise, move to iteration $k + 1$.

**Step 3:** Repeat Step 2 for the upper bound $U_n$ using $\lambda_u^{(k)}$, $\lambda_u^{(k-1)_x}$, and $\lambda_u^{(k-1)_a}$ in place of $\lambda_0^{(k)}$, $\lambda_0^{(k-1)_x}$, and $\lambda_0^{(k-1)_a}$, respectively.

### 2.3 Alternative implementation via weighted scores

Before we move to the empirical applications we discuss an interesting feature of ARTs as described in Algorithm 2.1. It turns out that ARTs can be implemented by a slightly different algorithm that does not involve estimating the parameter $\beta$ within each cluster. This alternative algorithm involves replacing Steps 1 and 2 in Algorithm 2.1 by the two alternative steps described in Algorithm 2.3 below, while keeping Steps 3 to 6 unaffected.

**Algorithm 2.3** (ARTs via within-cluster weighted scores). *This implementation of ARTs involves the following steps:*

**Step 1′:** Run a full-sample least squares regression of $Y_{i,j}$ on $Z_{i,j}$ subject to the restriction imposed by the null hypothesis, i.e., $c'\beta = \lambda$. Denote by $\hat{c}_{i,j}$ the restricted residuals from this regression and by $\hat{\beta}_n$ the restricted LS estimator of $\beta$.

**Step 2′:** For each cluster $j \in J$, define

$$S_{n,j} \equiv c'\hat{\Omega}_{n,j}^{-1}\frac{1}{\sqrt{n_j}} \sum_{i \in I_{n,j}} Z_{i,j}c_{i,j}^r,$$  \hspace{1cm} (17)

where

$$\hat{\Omega}_{n,j} \equiv \frac{1}{n_j} \sum_{i \in I_{n,j}} Z_{i,j}Z_{i,j}' \hspace{1cm} (18)$$

is a $d_z \times d_z$ matrix that is assumed to be full rank with inverse $\hat{\Omega}_{n,j}^{-1}$.

**Steps 3-6:** Same as in Algorithm 2.1.

Note that Steps 3-6 remain unchanged given the alternative definition of $S_{n,j}$ in Step 2′. When it comes to Steps 1 and 2, there are two differences worth discussing. The first difference is that Step 1′ requires a single full-sample restricted least squares estimator of $\beta$ as opposed to the $q$ cluster-by-cluster estimators in Step 1 of Algorithm 2.1. The second difference is that Step 2′ is based on within-cluster weighted scores as opposed to the centered within-cluster estimates of $\beta$ in Step 2 of Algorithm 2.1. Interestingly, these two implementations are numerically equivalent and so implementing ARTs via Algorithm 2.1 or Algorithm 2.3 leads to identical results. To see this formally, it is enough to show that $S_{n,j}$ as defined in (4) and (17) are the same using the following
argument. For each \( j \in J \),

\[
S_{n,j} \equiv c' \hat{\Omega}^{-1}_{n,j} \frac{1}{\sqrt{n_j}} \sum_{i \in I_{n,j}} Z_{i,j} \epsilon_{i,j}
\]

\[
= c' \hat{\Omega}^{-1}_{n,j} \frac{1}{\sqrt{n_j}} \sum_{i \in I_{n,j}} Z_{i,j} (Y_{i,j} - Z_{i,j} \hat{\beta}_n)
\]

\[
= c' \hat{\Omega}^{-1}_{n,j} \frac{1}{\sqrt{n_j}} \sum_{i \in I_{n,j}} Z_{i,j} Y_{i,j} - c' \hat{\Omega}^{-1}_{n,j} \frac{1}{\sqrt{n_j}} \sum_{i \in I_{n,j}} Z_{i,j} Z_{i,j} \hat{\beta}_n
\]

\[
= \sqrt{n_j} (c' \hat{\beta}_{n,j} - c' \beta) - \sqrt{n_j} (c' \hat{\beta}_n - c' \beta)
\]

\[
= \sqrt{n_j} (c' \hat{\beta}_{n,j} - c' \beta) - \sqrt{n_j} (c' \hat{\beta}_n - c' \beta) - \sqrt{n_j} (c' \hat{\beta}_n - \lambda)
\],

where the fourth equality follows by adding and subtracting \( \sqrt{n_j} c' \beta \) and the last equality holds because \( c' \hat{\beta}_n = c' \beta = \lambda \) under the null hypothesis in (2). It thus follows that \( S_{n,j} \) in (4) and in (17) are identical and so ARTs can be alternatively implemented via Algorithm 2.1 or 2.3.

3 What we need for ARTs to work

The main requirement underlying ARTs is Assumption 3.1 in Canay et al. (2017a). This assumption guarantees that the test delivers rejection probabilities under the null hypothesis that are close to the nominal level \( \alpha \) in an asymptotic framework where \( n \to \infty \) and \( q \) remains fixed. In the context of the linear model in (1), this translates into the following two conditions. The first condition is that the cluster-by-cluster estimators of \( \beta \) jointly converge in distribution at some (possibly unknown) rate; i.e.,

\[
\left( \begin{array}{c}
  a_{n,1} (\hat{\beta}_{n,1} - \beta) \\
  \vdots \\
  a_{n,q} (\hat{\beta}_{n,q} - \beta)
\end{array} \right) \xrightarrow{d} \left( \begin{array}{c}
  S_1 \\
  \vdots \\
  S_q
\end{array} \right)
\]  

for a sequences \( a_{n,j} \to \infty \) and random variables \((S_1, \ldots, S_q)'\). The second condition requires that the limiting random variables are invariant to sign changes, i.e.,

\[
(g_1 S_1, \ldots, g_q S_q) \xrightarrow{d} (S_1, \ldots, S_q),
\]

for any \( g \) in \( G \), where \( G \) is defined in Step 4 of Algorithm 2.1.

Condition (19) holds, for example, when \( Z_{i,j} \) and \( \epsilon_{i,j} \) are uncorrelated and the analyst assumes some form of weak dependence within clusters that permits the application of an appropriate central limit theorem. In such a case, (19) typically holds with \( a_{n,j} = \sqrt{n_j} \) and each \( S_j \) being a mean-zero normal random variable. In fact, under the commonly used assumption of independent clusters, it also follows that \( S_j \perp \perp S_{j'} \) for any...
In this case the normally distributed random variables may not be identically distributed but are indeed independent. Condition (20), in turn, requires each $S_j$ to be symmetrically distributed around zero and independent of each other. This is immediately satisfied when each $S_j$ is a mean-zero normal random variable and clusters are independent. Importantly, these assumptions allow for the normally distributed random variables to have different variances across clusters; a type of heterogeneity not allowed by the cluster wild bootstrap approach popularized by Cameron et al. (2008) and later studied formally by Canay et al. (2020).

**Remark 3.1.** We focus our exposition on the case where $Z_{i,j}$ is exogenous but we emphasize that the conditions in (19) and (20) typically hold in instrumental variable (IV) models. Accommodating IV to ARTs then only requires modifying Step 1 in Algorithm 2.1 so that the least squares regression is replaced with the appropriate IV regression. Steps 2-6 remain unaffected.

An implicit requirement behind ARTs that deserves further comments lies in Step 1 of Algorithm 2.1, which requires that the analyst runs cluster-by-cluster regressions. This step implicitly assumes that the parameter $\beta$ is identified within each cluster. In practice, this means that the matrix $\hat{\Omega}_{n,j}$ in (18) must be invertible and hence the same requirement applies to Algorithm 2.3. This restriction may be substantially important in some applications and so here we discuss common ways in which the problem may manifest and two alternative remedies.

One case in which running least squares cluster-by-cluster is not feasible is when the coefficient of interest is associated with a variable that only varies across clusters. For example, consider the model in (1) and partition $Z_{i,j}$ into a constant term, a scalar variable that only varies across clusters, $Z_{j(1)}$, and another variable that varies across and within clusters, $Z_{i,j(2)}$. That is,

$$Y_{i,j} = \beta_0 + Z_{j(1)} \beta_1 + Z_{i,j(2)} \beta_2 + \epsilon_{i,j}, \quad (21)$$

where the analysts’ interest lies in the coefficient $\beta_1$, i.e., $c' \beta = \beta_1$. Clearly, the regression in Step 1 of Algorithm 2.1 would not separately identify $\beta_0$ and $\beta_1$ as $Z_{j(1)}$ is perfectly colinear with the constant term. The matrix $\hat{\Omega}_{n,j}$ in (18) is simply singular. This situation arises, for example, in the empirical application considered by Canay et al. (2017b) where $j \in J$ indexes schools and the variable of interest is a treatment indicator at the school level. A natural remedy in a situation like this is clustering more coarsely (e.g., by combining clusters) to obtain variation within the re-defined clusters. This is possible for ARTs since the validity of the method does not rely on having a large number of clusters and thus it can afford to work with coarser clustering. In fact, in certain settings combining clusters may be quite natural. For example, Canay et al. (2017b) re-defined clusters as “pairs” of schools (as opposed to just schools) given that...
the treatment assignment mechanism of the experiment was a matched pairs design and so the pairs used at the randomization stage represented natural groupings. In other settings where it is less clear how to group clusters, any grouping that satisfies the requisite identification condition leads to a valid test, but it may be further desirable to combine such tests to limit concerns about “data snooping” across groupings. To this end, results in DiCiccio et al. (2020) on combining tests may be relevant.

Remark 3.2. A quick inspection of (21) may lead the analyst to believe there is a workaround that does not involve combining clusters if one instead uses some estimator of $\hat{\beta}_0$ from a full sample regression. For example, the full sample least squares estimator $\hat{\beta}_{n,0}$ from the regression in (1). Then, assuming for simplicity that $Z_{j}^{(1)} \neq 0$ for all $j \in J$, one may consider modifying Step 1 in Algorithm 2.1 by running a regression of $Y_{i,j}$ on an intercept and $Z_{i,j}^{(2)}$ (not including $Z_{j}^{(1)}$) and then redefining $\{\hat{\beta}_{n,j} : j \in J\}$ as the difference between the within cluster intercept estimates, $\hat{\beta}_{j,0}$ and the full sample estimate $\hat{\beta}_{n,0}$, i.e., $\hat{\beta}_{n,j} = \hat{\beta}_{j,0} - \hat{\beta}_{n,0}$. Such strategies unfortunately introduce dependence between the $q$ estimators of $\beta$ (as they all depend on $\hat{\beta}_{n,0}$) and thus end up violating one of the two main conditions needed for ARTs to be asymptotically valid; mainly condition (20).

Another case where the lack of identification within cluster may manifest is when the variable of interest actually varies within clusters but the model specification involves other variables that are collinear with some other variable (including the variable of interest or the constant term) within clusters. For example, consider the model in (1) where instead of individuals indexed by $i \in I_{n,j}$, units within cluster are indexed over time $t \in T$. Partition $Z_{j,t}$ into the variable of interest, $Z_{j}^{(1)}$, and time fixed effects $\delta_t$. That is,

$$Y_{j,t} = Z_{j,t}^{(1)} \beta_1 + \sum_{t \in T} I\{\hat{\delta}_t = t\} \delta_{\hat{t}} + \epsilon_{j,t}.$$  

(22)

It then follows that, within each cluster $j \in J$, the time fixed effect $\delta_t$ absorbs all the variation in $Z_{j,t}^{(1)}$ and so $\beta_1$ is not identified. In cases like this the analyst could again combine clusters to obtain variation within the re-defined clusters. An alternative remedy is to change the specification by, for example, replacing the time fixed effect with a cluster-specific time trend. Such specification is more restrictive than the time fixed effect in the sense that it imposes a linear trend but, at the same time, is more general as it allows for heterogeneity across clusters in the linear trend. We illustrate this approach in the application we consider in Section 4.1. The need to identify $\beta$ within each cluster is in our view the main limitation of ARTs, but a limitation that needs to be dealt with in certain settings. One may then wonder why not simply use some other inference method that is valid when the number of clusters is small and that does not rely on estimating $\beta$ cluster-by-cluster. Perhaps the most popular approach in that category is the cluster wild bootstrap popularized.
by Cameron et al. (2008) and recently studied formally by Canay et al. (2020). While not having to estimate $\beta$ within each cluster represents an advantage over ARTs, this additional flexibility comes at a cost in terms of the degree of heterogeneity that the model can deal with. In particular, the results in Canay et al. (2020) show that the cluster wild bootstrap is expected to work well in settings with a small number of clusters as long as the clusters are “homogeneous,” in a sense made precise in Canay et al. (2020). Intuitively, it is required that the variance covariance matrix $\Omega_{n,j}$ defined in (18) is the same across clusters (up to scalar multiplication). Such stringent homogeneity condition is not required for ARTs to work well, as the method allows clusters to be arbitrarily heterogeneous as long as $\Omega_{n,j}$ is invertible for $j \in J$.

**Remark 3.3.** For ease of exposition, we have written the requirement in (19) in terms of the differences $\hat{\beta}_{n,j} - \beta$, but it is possible to replace it with the differences $c'\hat{\beta}_{n,j} - c'\beta$ (or $R\hat{\beta}_{n,j} - R\beta$, depending on the null hypotheses of interest). In most cases, re-writing the condition in this way is not useful, but it is in cases where $c'\beta$ is identified within each cluster while $\beta$ is not. For example, consider the model in (21) when the coefficient of interest is $\beta_2$ as opposed to $\beta_1$, i.e., $c'\beta = \beta_2$. In that case the entire term $\beta_0 + Z_j^{(1)}\beta_1$ may be absorbed into a cluster-specific intercept without affecting the identification and estimation of $c'\beta = \beta_2$ within each cluster.

4 Empirical applications

In this section we apply ARTs as described in Algorithm 2.1 and ART-based confidence intervals as described in Algorithm 2.2 in the context of two distinct empirical applications. The R and Stata packages and codes required to replicate the results in this section are available as part of the online supplemental material.

4.1 Meng, Qian and Yared (2015)

Meng et al. (2015, MQY) argue that China’s Great Famine, from 1959 to 1961, was the result of an inflexible food procurement policy by the central government. To make this point, they show that food production and mortality become positively correlated during the time of famine, when this coefficient is otherwise negative or not significantly different from 0 in normal times.

MQY consider the following regression,

$$ Y_{j,t+1} = Z_{j,t}(1)\beta_1 + Z_{j,t}(2)\beta_2 + \delta_t + \epsilon_{j,t} $$

where $j$ indexes provinces (ranging from 1 to 19) and $t$ indexes years (ranging from 1953
to 1982). Here,

\[ Y_{j,t+1} = \log(\text{number of deaths in province } j \text{ during year } t + 1) \]
\[ Z_{j,t}^{(1)} = \log(\text{predicted grain production in province } j \text{ during year } t) \]
\[ \times I\{t \text{ is a famine year}\} \]
\[ Z_{j,t}^{(2)} = \log(\text{predicted grain production in province } j \text{ during year } t) \]
\[ \delta_t = \text{time fixed effects}. \]

In this application the level of clustering is a province, and so in order to apply ARTs as described in Section 2.1, one needs to estimate \( \beta = (\beta_1, \beta_2)' \) and \( \delta_t \) province-by-province. This illustrates one of the situations where including time fixed effects province-by-province is infeasible for the implementation of ARTs, given that the only source of remaining variation within a province is indeed time. The second identification problem described in Section 3 then arises. As we discussed in that section, one way to deal with this issue consists of replacing the time fixed effects with a cluster-specific time trend, i.e., in Step 1 of Algorithm 2.1 estimate

\[ Y_{j,t+1} = Z_{j,t}^{(1)} \beta_1 + Z_{j,t}^{(2)} \beta_2 + \gamma_j t + \epsilon_{j,t}. \]

We will refer to this as Analysis #1. In addition, we also consider the following alternative specifications studied by MQY:

- Analysis #2: Repeating Analysis #1 using only data between 1953 and 1965.
- Analysis #3: Repeating Analysis #1 using four additional autonomous provinces.
- Analysis #4: Repeating Analysis #2 using four additional autonomous provinces.
- Analysis #5: Repeating Analysis #1 using actual rather than constructed grain production.
- Analysis #6: Repeating Analysis #2 using actual rather than constructed grain production.

As with Analysis #1, the above analyses differ from their MQY counterparts only in that a linear time trend \( \gamma_j t \) replaces time fixed effects \( \delta_t \). Table 1 summarizes the number of clusters and the number of observations for each of these analyses.

Meng et al. (2015) consider the following two null hypotheses of interest,

\[ H_0^{(1)} : \beta_1 = 0 \quad \text{and} \quad H_0^{(2)} : \beta_1 + \beta_2 = 0. \]

(24)
Table 1: Cluster Information. ‘Min. Size’, ‘Med. Size’, ‘Max. Size’ denote the minimum, the median, and the maximum size of clusters.

<table>
<thead>
<tr>
<th>Analysis</th>
<th># of Clusters</th>
<th>Min. Size</th>
<th>Med. Size</th>
<th>Max. Size</th>
<th>Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>#1, #5</td>
<td>19</td>
<td>29</td>
<td>30</td>
<td>30</td>
<td>29.95</td>
</tr>
<tr>
<td>#2, #6</td>
<td>19</td>
<td>12</td>
<td>13</td>
<td>13</td>
<td>12.95</td>
</tr>
<tr>
<td>#3</td>
<td>23</td>
<td>29</td>
<td>30</td>
<td>30</td>
<td>29.96</td>
</tr>
<tr>
<td>#4</td>
<td>23</td>
<td>12</td>
<td>13</td>
<td>13</td>
<td>12.96</td>
</tr>
</tbody>
</table>

Table 2: Results for Analyses #1-6, comparable to those in Table 2 of Meng, Qian and Yared (2015). ‘LS Estimate’ denotes the full sample OLS estimate for $\beta_1$. CCE refers to cluster-robust standard errors. ART $p$-values are obtained using Algorithm 2.1. ART-based 95% confidence intervals are obtained using Algorithm 2.2.

In Table 2 we replicate the main table in Meng et al. (2015) using cluster robust standard errors (CCE) and also include the results associated with ARTs for both $H_0^{(1)}$ and $H_0^{(2)}$ in (24). For $H_0^{(1)}$ we report $p$-values and 95% confidence intervals, while for $H_0^{(2)}$ we just report $p$-values following MQY. The authors note in footnote 33 that using the cluster wild bootstrap led to similar results as those presented in their main table so we do not include cluster wild bootstrap results here either.

We comment on the following main features of Table 2:

1. For the null hypothesis $H_0^{(1)}$ associated with the parameter $\beta_1$, the ART $p$-values are of comparable magnitude to traditional CCE $p$-values. Similarly, ART-based confidence intervals are of roughly the same length as those obtained based on CCE although the ART-based confidence intervals do not contain the LS estimates. This is because ART-based confidence intervals are centered around the mean of the province-by-province estimates, which may not necessarily be equal to the full
sample LS estimate of $\beta_1$.

2. For the null hypothesis $H_0^{(2)}$ associated with the parameter $\beta_1 + \beta_2$, the ART $p$-value is sometimes higher and sometimes lower than the CCE $p$-value depending on the specification. Given the relatively small number of clusters in this application, the ART $p$-values are likely to be more reliable than those associated with CCE as CCE is known to perform poorly when the number of clusters is not sufficiently large.

4.2 Munyo and Rossi (2015)

Munyo and Rossi (2015) study criminal recidivism of former prisoners by looking at the relationship between the number of inmates released from incarceration on a given day and the number of offenses committed on the same day. They claim that the liquidity constraints that inmates face on the day of release increase the likelihood of recidivism on the same day. Using data of 2631 days between January 1st 2004 and March 15 2011 collected from the criminal incidents reports in Montevideo in Uruguay, they estimate the following linear model by least squares

$$Y_t = Z_t' \beta + \epsilon_t$$

where $t$ indexes days and

- $Y_t = \text{the total number of offenses on day } t$
- $Z_t = \text{the total number of inmates released, temperature, rainfall, hours of sunshine on day } t$, a dummy for holidays, a dummy for December 31st and a yearly trend.

We refer to this as Analysis #1. Munyo and Rossi (2015) additionally consider the following four analyses:

- Analysis #2: $Z_t$ includes a daily trend in place of a yearly trend.
- Analysis #3: $Z_t$ includes a monthly trend in place of a yearly trend.
- Analysis #4: $Z_t$ includes an intra-month daily trend, month- and year- level fixed effects and their interactions in place of a yearly trend.
- Analysis #5: $Z_t$ includes month- and year- level fixed effects and their interactions in place of a yearly trend.

Analysis #5 is their preferred specification. Munyo and Rossi (2015) report the results of these analyses in Table 2 in their paper. They report least squares estimates of $\beta$ with Newey-West heteroskedasticity-autocorrelation-consistent (HAC) standard errors.
addition, they report ART p-values as described in Algorithm 2.1 for the null hypothesis that $H_0 : c'\beta = 0$ as in (2), where $c$ selects the coefficient on the total number of inmates released on day $t$.

In this application the level of clustering is not naturally determined by the data, but pseudo-clusters may be formed using blocks of consecutive observations under the assumption of weak temporal dependence. In order to apply ARTs as described in Algorithm 2.1 we then form $q$ pseudo-clusters by dividing the data into $q$ consecutive blocks of size $b_n = \lfloor n/q \rfloor$ where $n = 2631$ is the number of total observations. More concretely, we define the $j$th pseudo-cluster as

$$X_{j}^{(n)} = \{(Y_t, Z_t') : t = (j - 1)b_n + 1, \cdots, jb_n\} \quad \text{where} \quad j = 1, \cdots, q - 1,$$

and let the last $q$th pseudo-cluster contain all the remaining $n - b_n(q - 1)$ observations. Note that in this application the number of pseudo-clusters $q$ is a tuning parameter that the analyst must specify. Munyo and Rossi (2015) set $q = 10$. We repeat their analyses with alternative values of $q$ and investigate how sensitive the results are to this choice. The relevant cluster information is given in Table 3.

<table>
<thead>
<tr>
<th># of Clusters (q)</th>
<th>Cluster Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>328</td>
</tr>
<tr>
<td>10</td>
<td>219</td>
</tr>
<tr>
<td>16</td>
<td>164</td>
</tr>
</tbody>
</table>

Table 3: Pseudo-cluster size for different values of $q$.

Table 4 shows LS estimates of $\beta$, p-values for the hypothesis in (2), and 95% confidence intervals for each analysis. Following Munyo and Rossi (2015), we report results based on HAC standard errors. The table also shows ART p-values as described in Algorithm 2.1 and ART-based 95% confidence intervals as described in Algorithm 2.2 for $q = 8$, $q = 10$, and $q = 16$.

We summarize the main findings of the results in Table 4 as follows:

1. The choice of $q$ is important for the results of ARTs but currently there is no theory developed to choose this tuning parameter according to some data dependent criteria. The smaller $q$ is, the more observations are available within each cluster. Having more observations per cluster is important for one of the requirements behind ARTs, mainly (19). A small value of $q$, however, tends to affect the power of ARTs despite not really affecting the control of the rejection probability under the null hypothesis. This feature can be seen in Table 4, where ARTs p-values are decreasing in $q$ across different specifications. In this application, where there are
<table>
<thead>
<tr>
<th>Specification</th>
<th>#1</th>
<th>#2</th>
<th>#3</th>
<th>#4</th>
<th>#5</th>
</tr>
</thead>
<tbody>
<tr>
<td>LS Estimate</td>
<td>0.225</td>
<td>0.260</td>
<td>0.259</td>
<td>0.225</td>
<td>0.234</td>
</tr>
<tr>
<td>HAC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>se</td>
<td>0.124</td>
<td>0.123</td>
<td>0.123</td>
<td>0.096</td>
<td>0.096</td>
</tr>
<tr>
<td>p-value</td>
<td>0.068</td>
<td>0.034</td>
<td>0.034</td>
<td>0.019</td>
<td>0.015</td>
</tr>
<tr>
<td>95% CI</td>
<td>[-0.017, 0.468]</td>
<td>[0.02, 0.5]</td>
<td>[0.019, 0.5]</td>
<td>[0.038, 0.413]</td>
<td>[0.046, 0.421]</td>
</tr>
<tr>
<td>CRS: q=8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>estimate</td>
<td>0.277</td>
<td>0.220</td>
<td>0.217</td>
<td>0.177</td>
<td>0.182</td>
</tr>
<tr>
<td>p-value</td>
<td>0.008</td>
<td>0.023</td>
<td>0.023</td>
<td>0.102</td>
<td>0.102</td>
</tr>
<tr>
<td>95% CI</td>
<td>[0.124, 0.429]</td>
<td>[0.035, 0.391]</td>
<td>[0.035, 0.391]</td>
<td>[-0.07, 0.397]</td>
<td>[-0.067, 0.418]</td>
</tr>
<tr>
<td>CRS: q=10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>estimate</td>
<td>0.354</td>
<td>0.252</td>
<td>0.257</td>
<td>0.204</td>
<td>0.225</td>
</tr>
<tr>
<td>p-value</td>
<td>0.002</td>
<td>0.014</td>
<td>0.014</td>
<td>0.063</td>
<td>0.053</td>
</tr>
<tr>
<td>95% CI</td>
<td>[0.141, 0.603]</td>
<td>[0.068, 0.446]</td>
<td>[0.068, 0.458]</td>
<td>[-0.023, 0.431]</td>
<td>[-0.003, 0.452]</td>
</tr>
<tr>
<td>CRS: q=16</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>estimate</td>
<td>0.286</td>
<td>0.229</td>
<td>0.226</td>
<td>0.172</td>
<td>0.207</td>
</tr>
<tr>
<td>p-value</td>
<td>0.002</td>
<td>0.006</td>
<td>0.006</td>
<td>0.027</td>
<td>0.010</td>
</tr>
<tr>
<td>95% CI</td>
<td>[0.131, 0.444]</td>
<td>[0.097, 0.369]</td>
<td>[0.087, 0.371]</td>
<td>[0.02, 0.324]</td>
<td>[0.056, 0.367]</td>
</tr>
</tbody>
</table>

Table 4: Results for Analyses #1-5, comparable to those in Table 2 of Munyo and Rossi (2015). ‘LS Estimate’ denotes the full sample LS estimate of $\beta$. HAC refers to the heteroskedasticity and autocorrelation consistent standard error. ART p-values are obtained using Algorithm 2.1. ART-based 95% confidence intervals are obtained using Algorithm 2.2.
still over a hundred observations when \( q = 16 \), a larger value of \( q \) like \( q = 10 \) or \( q = 16 \) may be preferable to smaller values, like \( q = 8 \), based on power considerations. Note, however, that except in Analyses \#4–5, where the choice of \( q \) determines whether the null hypothesis is rejected at a given significance level, the results for Analyses \#1–3 are in all agreement at a 5% level.

2. Overall, the test results based on standard \( t \)-test with HAC standard errors are consistent to those of ARTs when \( q = 16 \). Both methods reject the null hypothesis \( H_0 : c' \beta = 0 \) at a 10% nominal level across different specifications. The results support the authors’ argument that the release of inmates from incarceration increase the chance of re-offenses on the day of release.

5 Concluding remarks

The goal of this paper is to make the general theory developed in Canay et al. (2017a) more accessible by providing a step-by-step algorithmic description of how to implement the test and construct confidence intervals in linear regression models with clustered data, as well as clarifying the main requirements and limitations of the approach. The main two takeaways are the following. First, implementation of ARTs in linear regression models is as straightforward as other more conventional inference methods. Algorithms 2.1 and 2.2 provide a clear explanation of how to take ARTs to the data, and the companion Stata and R packages available as part of the supplemental material are intended to facilitate adoption of ARTs. Second, our discussion on the main requirements behind ARTs hopefully show that understanding the trade-offs between ARTs and other popular alternatives for inference with a small number of clusters, like the clustered wild bootstrap, is fundamental for practitioners to choose a method that aligns well with the features of their application. In particular, while ARTs essentially demand that the parameter of interest is suitably estimable cluster-by-cluster without imposing restrictions on the degree of heterogeneity across clusters, the cluster wild bootstrap requires the clusters to be sufficiently homogeneous (see Canay et al., 2020) without demanding identification of the parameter of interest cluster-by-cluster.

References


Cameron, A. C., Gelbach, J. B. and Miller, D. L. (2008). Bootstrap-based im-


