Cluster Robust Inference: Bias Reduced Linearization, Effective Degrees of Freedom, and Two-Way Clustered Standard Errors

by

Sean Elliott

An essay submitted to the Department of Economics
in partial fulfillment of the requirements for
the degree of Master of Arts

Queen's University
Kingston, Ontario, Canada
August 2016

Abstract

This paper will provide an overview of some recent developments in the area of cluster robust inference and test these methods under relatively more stringent conditions than has been undertaken in previous analyses. The methods under consideration are, the conventional CRVE, Bell and McCaffrey (2002) bias reduced linearization methods, the wild cluster bootstrap, and Young (2016) effective degrees of freedom corrections. Certain known issues regarding computational drawbacks of these methods will be discussed as well, highlighting certain properties which researchers should be aware of. Specifically, these estimators will be tested when the number of clusters is small and when there is a significant variation in cluster sizes. Finally, a generalization in the topic of two-way cluster robust estimation is tested, where Bell and McCaffrey (2002) residual adjustments will be applied to the standard two-way clustering approach.

Acknowledgments

I would first like to express gratitude to my supervisor James G. MacKinnon for his expert advice and meaningful feedback. Additionally, I would like to thank Mark Babcock and Sergei Shibaev for their technical assistance. Lastly, I would like to acknowledge my family and friends for their unwavering support. All errors within are my own.

Table of Contents

1 Introduction

The need for cluster robust methods arises when the model disturbances are correlated within some level of the data. An example of this would be within-group correlation with groups being Canadian provinces or American states. The primary issue is that, with the presence of within-cluster correlation, the usual ordinary least squares (OLS) standard errors tend to be biased downwards resulting in the rejection of true null hypotheses. The most obvious approach one could take in order to correct for this would be to add group level fixed-effects. However, Bertrand, Duflo, and Mulanaithan (2004) provide evidence that adding group fixed-effects may not entirely account for the intra-group correlation that causes the bias. Additionally, Kloek (1981), one of the original motivators for this idea, also notes that if a variable does not vary within a cluster, these fixed effects cannot be used. Hence, when the researcher is faced with this scenario, the utilization of cluster robust standard errors is required.

The most common approach is the cluster robust covariance matrix estimator (CRVE), which is implemented in Stata and was proposed by Liang and Zeiger (1986).[1] This covariance matrix estimator can be shown to be consistent given three assumptions. These assumptions are that, the number of clusters tends to infinity, the within-cluster error correlations are equal for each cluster, and the number of observations within each cluster is the same for all clusters. However, these relatively strong assumptions are not likely to hold in practice. For example, one may encounter what is considered to be a 'small' number of clusters if they wish to cluster the data according to Canadian provinces. Cameron, Gelbach, and Miller (2008) provide evidence that the standard CRVE over-rejects quite severely in the case where there are only ten clusters. A large variation in cluster sizes, such as fifty state-sized clusters

[1] Also referred to as CR_1.

also results in CR_1 performing poorly, as shown in MacKinnon and Webb (2016).

Various methods have been proposed to correct for these known issues. Specifically, a cluster robust version of the wild bootstrap seems to be an improvement over the conventional CRVE when encountering varying cluster sizes. For the problem of a small number of clusters, a method proposed by Bell and McCaffrey (2002) modifies the residuals in a way similar to that of the heteroskedasticity robust HC_2 and HC_3 corrections in MacKinnon and White (1985). Due to their resemblance to their heteroskedasticity robust counter-parts, these methods are typically referred to as CR_2 and CR_3. More recently Young (2016) has extended these methods by proposing modifications to CR_1, CR_2, and CR_3. Additionally he suggests using what he terms effective degrees of freedom corrections. These methods will be explored in detail later.

As given in MacKinnon and Webb (2016), the basic set up is a linear model which is estimated by OLS,

$$\boldsymbol{y} = \begin{bmatrix} \boldsymbol{y_1} \\ \boldsymbol{y_2} \\ \vdots \\ \boldsymbol{y_G} \end{bmatrix} = \boldsymbol{X\beta} + \boldsymbol{u} = \begin{bmatrix} \boldsymbol{X_1} \\ \boldsymbol{X_2} \\ \vdots \\ \boldsymbol{X_G} \end{bmatrix} \boldsymbol{\beta} + \begin{bmatrix} \boldsymbol{u_1} \\ \boldsymbol{u_2} \\ \vdots \\ \boldsymbol{u_G} \end{bmatrix} \tag{1}$$

where the gth cluster has N_g observations. In general, the matrix $\boldsymbol{X}$ and vectors $\boldsymbol{y}$ and $\boldsymbol{u}$ have $N = \sum_{g=1}^{G} N_g$ rows. Additionally, $\boldsymbol{X}$ has dimension $N \times k$ and the parameter vector $\boldsymbol{\beta}$ is $k \times 1$. As the disturbances are assumed to be uncorrelated across clusters, the covariance matrix of the vector of disturbances $\boldsymbol{u}$ is block diagonal with

each $N_g \times N_g$ block given by

$$E(\boldsymbol{u}_g \boldsymbol{u}_g') = \boldsymbol{\Omega}_g, \quad g = 1, ..., G \tag{2}$$

where $\boldsymbol{\Omega}_g$ is unknown for each cluster g.

Moulton (1990) highlights why clustering is an important issue by providing an example which illustrates the difference in magnitude between the true covariance matrix and the standard OLS covariance matrix. Suppose the simplest regression model with a constant and one independent variable, with coefficient β_2. Denote the conventional OLS covariance matrix as $\text{var}_c(\hat{\beta}) = \sigma^2(\boldsymbol{X}'\boldsymbol{X})^{-1}$ and suppose that $\text{var}(\hat{\beta})$ is the true covariance matrix.[2] If it is supposed first that all intra-cluster correlations are constant and given by ρ_g, then it can be shown that, given N_g equal for each cluster,

$$\frac{\text{var}(\hat{\beta}_2)}{\text{var}_c(\hat{\beta}_2)} = 1 + (N_g - 1)\rho_g, \tag{3}$$

where the square root of the right-hand side is called the Moulton factor.[3] From this one can observe that, even if within-cluster correlations are small, the conventional OLS standard error may be significantly larger than the correct standard error if N_g is large. Additionally, this factor is increasing in ρ_g implying that the larger the intra-cluster correlation, the more biased the conventional OLS standard errors will be.

In this paper I will compare recent results in the literature with methods proposed previously, where I will be testing these newer results under more stringent conditions

[2]It is worthwhile to note that the variance of the estimate of the parameter β_2 is simply the second diagonal element of these covariance matrices.

[3]The details of this are given in the appendix.

and environments than done previously. Additionally, certain extensions of the wild cluster bootstrap using modified residuals will be tested as well. These methods will be tested when the number of clusters is small and when there is a large variation in the number of observations per cluster. Lastly, a small extension of two-way clustering will also be tested whereby CR_1 will be replaced by other standard error estimates. Here I find a significant improvement in performance over the conventional two-way standard error estimates.

The contribution of this analysis is twofold. Firstly, it is to scrutinize certain methods which have not been analyzed under situations where they are likely to fail. Monte Carlo simulations will be used extensively to draw conclusions regarding the performance of the most commonly used cluster robust estimation techniques and how they compare with recently proposed extensions. The second contribution will be to discuss the methods of two-way clustering, propose a small extension, and analyze the performance of these extensions using Monte Carlo experiments. This is necessary due to the less than ideal performance of the conventional method for two-way cluster robust estimation, which shows favourable performance in only a few scenarios.

The outline for this paper is as follows. Section 2 will briefly cover the basics of all cluster robust estimation methods which will be tested in this analysis. This section will also include a brief discussion on when these methods are known to perform poorly. Section 3 will discuss the issues and limitations associated with computing cluster robust estimators. Section 4 will contain the first set of Monte Carlo experiments testing the recently proposed estimators and comparing their performance to conventional methods. Finally, Section 5 will discuss how two-way clustering can be extended and Monte Carlo evidence will be provided to show how these extensions can improve the performance of two-way cluster robust estimation.

2 Cluster Robust Inference: An Overview

Here I provide a general summary of which methods will be tested with some of their basic properties and known issues or strengths. This discussion will be an overview and one should appeal to the original publications on these methods for more detail and asymptotic theory.

2.1 Conventional CRVE

This method was originally proposed by Liang and Zeiger (1986) and is the robust covariance matrix estimator employed by Stata's *cluster* command. The form of this covariance matrix estimator is given by

$$\text{CR}_1 = \frac{G(N-1)}{(G-1)(N-k)}(\boldsymbol{X}'\boldsymbol{X})^{-1}\Big(\sum_{g=1}^{G}\boldsymbol{X}_g'\hat{\boldsymbol{u}}_g\hat{\boldsymbol{u}}_g'\boldsymbol{X}_g\Big)(\boldsymbol{X}'\boldsymbol{X})^{-1} \tag{4}$$

where $\hat{\boldsymbol{u}}_g$ are the residuals corresponding to cluster $g \in (1, 2, \cdots, G)$ and $\boldsymbol{X}_g$ is the $N_g \times k$ portion of the matrix of observations $\boldsymbol{X}$. Alternatively, one could write this in a more compact fashion by using $\{\cdot\}$ to denote a block diagonal matrix and $C_{\text{CR}_1} = \frac{G(N-1)}{(G-1)(N-k)}$. Then (4) could be written as,

$$\text{CR}_1 = C_{\text{CR}_1}(\boldsymbol{X'X})^{-1}\boldsymbol{X'}\{\hat{\boldsymbol{u}}_g\hat{\boldsymbol{u}}_g'\}\boldsymbol{X}(\boldsymbol{X'X})^{-1}. \tag{5}$$

This method controls for error heteroskedasticity across clusters, correlation within clusters of a general form, and heteroskedasticity within clusters. However, this is only under the assumption that the number of clusters G becomes arbitrarily large. Additionally, the degrees of freedom adjustment C_{CR_1} is asymptotically negligible as

it tends to $\frac{N}{N-k} \approx 1$ as $G \to \infty$.

One benefit of CR_1 is that it is computationally cheap, relative to other methods, as it does not require the inversion of any large matrices.[4] For example, CR_2 and CR_3 require inverting $N_g \times N_g$ matrices which, depending on cluster size, may become computationally impractical or even unfeasible. A drawback, however, as pointed out in Cameron, Gelbach, and Miller (2008), is that $E(\hat{\boldsymbol{u}}_g \hat{\boldsymbol{u}}_g{}') \neq \Omega_g = E(\boldsymbol{u}_g \boldsymbol{u}_g')$ implying that it is a biased estimator. While (4) may be biased, it is also known that it is typically biased downward.[5] Due to the bias present in CR_1, Bell and McCaffrey (2002), proposed corrections where they used (4) and replaced $\hat{\boldsymbol{u}}_g$ with modified residuals. These modifications, in the cluster robust case, are analogous to the heteroskedasticity robust residual modifications proposed by MacKinnon and White (1985).

Initially, the proposed hypothesis test using this estimator was utilizing critical values based on the Student's t distribution with $N-k$ degrees of freedom. However, Donald and Lang (2007) and Bester, Conley, and Hansen (2011) suggest, and provide evidence that, a $t(G-1)$ distribution is much more appropriate. In fact, this is the method which Stata uses and will be the distribution used for simulation experiments involving CR_1 in this analysis. Other methods as suggested by Carter, Schnepel, and Steigerwald (2015) suggest using what is termed effective number of clusters, G^*, which depends on the $\boldsymbol{X}_g$ matrices. These methods will not be discussed further. However, one may refer to MacKinnon and Webb (2016) for simulation evidence involving these distributions.

[4]Despite the computation of $(\boldsymbol{X}'\boldsymbol{X})^{-1}$ being unavoidable, this matrix is only $k \times k$, which for all practical purposes is a non-issue.

[5]When using $\hat{\boldsymbol{u}}_g$ instead of the true disturbances Kézdi (2004) provides simulation evidence suggesting that the bias is downward and between 9% to 16%.

2.2 Bias Reduced Linearization

Initially proposed by Bell and McCaffrey (2002), these methods involve a rescaling of the residuals in order to deal with the bias associated with (4). This is due to the fact that when the number of clusters is small, the standard error estimates have a strong tendency to be downwards biased. They propose using

$$\dot{\boldsymbol{u}}_g = (\boldsymbol{I} - \boldsymbol{P}_{gg})^{-1/2}\hat{\boldsymbol{u}}_g \qquad g = 1, \cdots, G \tag{6}$$

as a transformation of the standard OLS residual. Here $\boldsymbol{P}_{gg} \equiv \boldsymbol{X}_g(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}_g'$, which is the $N_g \times N_g$ block of the projection matrix $\boldsymbol{P}_{\boldsymbol{X}} = \boldsymbol{X}(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'$ corresponding to cluster g. Since this method is a generalization of the heteroskedasticity robust method HC_2, it is commonly referred to as CR_2. Additionally they use an alternative method whereby the symmetric square root is replaced by the inverse. This again is a cluster robust generalization of the heteroskedasticity robust case proposed by MacKinnon and White (1985), HC_3, hence it is referred to as CR_3. Hence, one may write,

$$\mathrm{CR}_2 = (\boldsymbol{X}'\boldsymbol{X})^{-1}\Big(\sum_{g=1}^{G}\boldsymbol{X}_g'((\boldsymbol{I} - \boldsymbol{P}_{gg})^{-1/2}\hat{\boldsymbol{u}}_g)((\boldsymbol{I} - \boldsymbol{P}_{gg})^{-1/2}\hat{\boldsymbol{u}}_g)'\boldsymbol{X}_g\Big)(\boldsymbol{X}'\boldsymbol{X})^{-1} \tag{7}$$

and, CR_3 simply replaces $(\cdot)^{-1/2}$ with $(\cdot)^{-1}$. If we let $\boldsymbol{M}_{gg} = \boldsymbol{I} - \boldsymbol{P}_{gg}$ (7) may be written more compactly as

$$\mathrm{CR}_2 = (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\{\boldsymbol{M}_{gg}^{-1/2}\hat{\boldsymbol{u}}_g\hat{\boldsymbol{u}}_g'\boldsymbol{M}_{gg}^{-1/2}\}\boldsymbol{X}(\boldsymbol{X}'\boldsymbol{X})^{-1/2}. \tag{8}$$

Results presented by Cameron, Gelbach, and Miller (2008) suggest that CR_3 adjustment performs better than the CR_1 adjustment when using the $t(N-k)$ distribution. However, they did not compare the rejection rates of CR_3 to CR_1 when using what is

now the preferred method employing a $t(G-1)$ distribution. While the performance of the estimators CR_2 and CR_3 may be better than that of the conventional alternative CR_1, it should be noted that these methods are computationally expensive. This is due to the unavoidable inversion of the $N_g \times N_g$ matrices which appear in the residual rescalings.

2.3 Effective Degrees of Freedom Corrections

Bell and McCaffrey (2002) proposed this idea initially. They suggest using degrees of freedom based on the Satterthwaite (1946) approximation. This idea was motivated by the fact that the standard errors of OLS coefficients are only χ^2-distributed under a particular set of assumptions. Additionally, they show that the alternative distribution, using $G-1$ degrees of freedom, was only true if $\boldsymbol{X}_g'\boldsymbol{X}(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{l}_p$ is constant for all clusters $g \in (1, 2, ..., G)$, where $\boldsymbol{l}_p$ is a $k \times 1$ vector which has the pth element equal to one, and zero otherwise. However, in general, this condition does not hold. Instead they suggest using an alternative t-distribution based on a different degrees of freedom calculation. The following exposition is based on a simplified version, which is useful for our scenario and can be attributed to Imbens and Kolesar (2012).

Formally, they define the $N \times G$ matrix Z where the gth column is given by

$$Z_g = (\boldsymbol{I}_N - \boldsymbol{P}_{\boldsymbol{X}})_g'(\boldsymbol{I}_{N_g} - \boldsymbol{P}_{gg})^{-1/2}\boldsymbol{X}_g(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{l}_p, \tag{9}$$

where $(\boldsymbol{I}_N - \boldsymbol{P}_{\boldsymbol{X}})_g$ refers to the N_g rows of the $N \times N$ projection matrix $\boldsymbol{M}_{\boldsymbol{X}} = \boldsymbol{I} - \boldsymbol{P}_{\boldsymbol{X}}$. Then the relevant degrees of freedom are given by

$$\upsilon_Z = \frac{\left(\sum_{i=1}^{G} \lambda_i\right)^2}{\sum_{i=1}^{G} \lambda_i^2}, \tag{10}$$

where the λ_i are the eigenvalues of the square matrix $\boldsymbol{Z'Z}$. Ideally one would be using the eigenvalues of $\boldsymbol{Z'\Omega Z}$; however, $\boldsymbol{\Omega}$ is difficult to estimate correctly. Imbens and Kolesar (2012) propose a structure for $\boldsymbol{\Omega}$ which leads to a slightly different degrees of freedom correction. However, this will not be explored in this paper. Additionally, it is worthwhile to note that as opposed to the scenario where one uses $G-1$ degrees of freedom, this method results in different degrees of freedom depending on which coefficient one is testing.

Tipton (2015) and Pustejovsky and Tipton (2016) endorse using CR_2 with the degrees of freedom correction given by (10). Recently, Young (2016) has proposed a method which is similar to that of Bell and McCaffrey (2002), using a similar degrees of freedom correction also based on Satterthwaite (1946). The method proposed involves testing whether or not a linear combination of the coefficient vector $\boldsymbol{w'\beta}$ is significantly different than a specified null value w_0. Using $\boldsymbol{V}_i$ as the estimated covariance matrix of $\boldsymbol{\beta}$, he rewrites the usual test statistic, denoted by $\tilde{t}_i$ as

$$\tilde{t}_i = \frac{\boldsymbol{w'\beta} - w_0}{\sqrt{\boldsymbol{w'V}_i\boldsymbol{w}}} = \frac{\frac{\boldsymbol{w'\beta}-w_0}{\sqrt{\sigma^2\boldsymbol{w'(X'X)^{-1}w}}}}{\sqrt{\frac{\boldsymbol{w'V}_i\boldsymbol{w}}{\sigma^2\boldsymbol{w'(X'X)^{-1}w}}}} = \frac{\frac{\boldsymbol{w'\beta}-w_0}{\sqrt{\sigma^2\boldsymbol{w'(X'X)^{-1}w}}}}{\sqrt{\frac{\boldsymbol{u'}}{\sigma}\boldsymbol{B}_i\frac{\boldsymbol{u}}{\sigma}}}. \tag{11}$$

where he assumes that $\boldsymbol{V}_i$ can be re-expressed as a standard normal quadratic form with the matrix $\boldsymbol{B}_i$.[6] Next, he notes that if $\boldsymbol{u}$ is iid normal, then this quadratic form has mean $\mu_i = \text{trace}(\boldsymbol{B}_i)$ and variance $v_i = 2[\text{trace}(\boldsymbol{B}_i\boldsymbol{B}_i)]$.[7] Additionally, if $\boldsymbol{B}_i$ is an idempotent matrix, this quadratic form is a χ^2 random variable with μ_i degrees of freedom.

[6]Here i refers to a specific covariance matrix estimator, that is, $i = \text{CR}_1,\ \text{CR}_2,\ \text{CR}_3$.

[7]The mean comes from the property of iid as, in general with quadratic forms, $E(\boldsymbol{X'AX}) = \text{trace}(\boldsymbol{A\Sigma}) + \boldsymbol{\mu'A\mu}$. Since it is assumed that $\boldsymbol{\Sigma} = I$ and $\boldsymbol{\mu} = \boldsymbol{0}$, we find that $E(\frac{\boldsymbol{u'}}{\sigma}\boldsymbol{B}_i\frac{\boldsymbol{u}}{\sigma}) = \text{trace}(\boldsymbol{B}_i)$. The variance comes from the fact that in general, if $\boldsymbol{a} \sim N(0, \sigma^2 I)$ and $\boldsymbol{M}$ is a symmetric idempotent matrix of rank m, then $\frac{\boldsymbol{a}'}{\sigma}\boldsymbol{M}\frac{\boldsymbol{a}}{\sigma} \sim \chi^2(\text{trace}(\boldsymbol{M}))$.

However, if $\boldsymbol{B}_i$ is not idempotent, corrections can be made such that the resulting expression will mimic the moment conditions of the idempotent form. This correction involves multiplying the expression in the denominator of (11) by $2\mu/v$ so that it will have mean $2\mu^2/v$ and variance $4\mu^2/v$. The next step is to mimic the conventional t-statistic by dividing the denominator by the degrees of freedom of the approximately χ^2 distributed random variable. Then (11) becomes

$$\tilde{t}_i = \frac{\boldsymbol{w'\beta} - w_0}{\sqrt{\boldsymbol{w}'\boldsymbol{V}_i\boldsymbol{w}}} = \frac{\frac{\boldsymbol{w'\beta}-w_0}{\sqrt{\sigma^2\boldsymbol{w'(X'X)^{-1}w}}}}{\sqrt{\frac{1}{\mu}\frac{\boldsymbol{w}'\boldsymbol{V}_i\boldsymbol{w}}{\sigma^2\boldsymbol{w'(X'X)^{-1}w}}}} = \frac{\frac{\boldsymbol{w'\beta}-w_0}{\sqrt{\sigma^2\boldsymbol{w'(X'X)^{-1}w}}}}{\sqrt{\frac{1}{2\mu^2/v}\frac{\boldsymbol{u}'}{\sigma}(\boldsymbol{B}_i\frac{2\mu}{v})\frac{\boldsymbol{u}}{\sigma}}} \tag{12}$$

which is approximately t-distributed with $2\mu/v$ effective degrees of freedom. Here the effective degrees of freedom calculation is given by (10) where now the eigenvalues come from the matrix $\boldsymbol{B}_i$.

Next, to calculate the $\boldsymbol{B}_i$ matrices, first recall the expression for CR_1 given in (4) and the corresponding CR_2 and CR_3 adjustments. In order to write this more compactly, where $\{\cdot\}$ represents a block diagonal matrix we define the following,

$$\begin{aligned} \boldsymbol{z}' = \boldsymbol{z}'_{\mathrm{CR}_1} &= \boldsymbol{w}'(\boldsymbol{X'X})^{-1}\boldsymbol{X}' \\ \boldsymbol{z}'_{\mathrm{CR}_2} &= \boldsymbol{w}'(\boldsymbol{X'X})^{-1}\boldsymbol{X}\{\boldsymbol{M}_{gg}^{-1/2}\} \\ \boldsymbol{z}'_{\mathrm{CR}_3} &= \boldsymbol{w}'(\boldsymbol{X'X})^{-1}\boldsymbol{X}'\{\boldsymbol{M}_{gg}^{-1}\} \end{aligned} \tag{13}$$

and (11) implies that,

$$\frac{\boldsymbol{w}'\boldsymbol{V}_i\boldsymbol{w}}{\sigma^2\boldsymbol{w}'(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{w}} = \frac{\boldsymbol{u}'}{\sigma}\boldsymbol{B}_i\frac{\boldsymbol{u}}{\sigma}. \tag{14}$$

Then the relevant $\boldsymbol{B}_i$ matrices can be written as

$$\boldsymbol{B}_i = \frac{C_i}{\boldsymbol{z}'\boldsymbol{z}}\boldsymbol{M}_{\boldsymbol{X}}\{\boldsymbol{z}_{i,g}\boldsymbol{z}'_{i,g}\}\boldsymbol{M}_{\boldsymbol{X}} \qquad i = C_{\mathrm{CR}_1},\ C_{\mathrm{CR}_2},\ C_{\mathrm{CR}_3} \tag{15}$$

with $C_{\mathrm{CR}_2} = C_{\mathrm{CR}_3} = 1$, and C_{CR_1} being the CR_1 degrees of freedom adjustment present in (4). As mentioned before, now $\tilde{t}_i$ will be approximately t-distributed with $2\mu_i/v_i$ effective degrees of freedom.

2.4 Wild Cluster Bootstrap Methods

The wild cluster bootstrap was originally introduced by Cameron, Gelbach, and Miller (2008) as a technique to deal with the conventional CRVE performing poorly when the number of clusters is small. One noteworthy feature of this method is that the within-cluster error correlations are preserved through the use of an auxiliary probability distribution, which is applied to the residuals. This is different than the wild bootstrap used to control for heteroskedasticity, where the auxiliary random variable is applied to each observation. Suppose we wish to use this method to test the hypothesis that a specific coefficient is zero. Just for illustration purposes, suppose this coefficient is the last element of the vector $\boldsymbol{\beta}$, β_k. First, we illustrate this method for the CR_1 estimator. However, it may be generalized. The procedure is as follows:

1. Estimate (1) by OLS.

2. Calculate the t-statistic for the hypothesis that $\beta_k = 0$ using the square root of the k^{th} diagonal element of the covariance matrix CR_1 as the standard error.

3. Re-estimate the original model with the restriction $\beta_k = 0$ imposed. This is done to obtain restricted residuals $\tilde{u}$ and restricted parameter estimates $\tilde{\boldsymbol{\beta}}$.

4. In each of the $b \in B$ bootstrap replications, generate a new set of bootstrap dependent variables using the bootstrap DGP

$$\boldsymbol{y}_{ig}^{*b} = \boldsymbol{X}_{ig}\tilde{\boldsymbol{\beta}} + \tilde{\boldsymbol{u}}_{ig}v_g^{*b} \tag{16}$$

where v_g is a Rademacher random variable specific to each cluster, which takes on the value 1 or -1 with equal probability.

5. For each $b \in B$, estimate (1) using $\boldsymbol{y}^{*b}$ as the dependent variable and calculate the bootstrap t-statistic t_k^{*b} for the hypothesis that $\beta_k = 0$ using the bootstrap residuals.

6. Calculate the bootstrap P value

$$\hat{p}^* = \frac{1}{B} \sum_{b=1}^{B} I(|t_k^{*b}| > |t_k|) \tag{17}$$

where $I(\cdot)$ is the indicator function.

However, this is not the only variation of the wild cluster bootstrap, as one may adjust the residuals in the fashion suggested by Bell and McCaffrey (2002) and perform the same procedure. As suggested by MacKinnon (2015), one may replace $\tilde{\boldsymbol{u}}_{ig}$ in (16) with $\dot{\boldsymbol{u}}_{ig} = \boldsymbol{M}_{gg}^{-1/2}\tilde{\boldsymbol{u}}_{ig}$ or $\ddot{\boldsymbol{u}}_{ig} = \boldsymbol{M}_{gg}^{-1}\tilde{\boldsymbol{u}}_{ig}$.

Upon first inspection, one may expect that the residual transformations would be far too computationally expensive as one would be needing to compute an inverse or a symmetric square root inverse on each bootstrap replication. However, this is not the case. Rather, as mentioned in MacKinnon (2015), this computation only needs to be made once and then it can be reused in each bootstrap replication. Of course, this does not negate the fact that if the $N_g \times N_g$ matrices are large enough, this method becomes computationally impractical or even unfeasible.

Additionally, Webb (2014) outlines the issues associated with this procedure when the number of clusters is small. When G is small, so too is the number of unique bootstrap samples. What this then implies is that the number of unique samples

depends on the auxiliary distribution being applied to the residuals. Additionally, he points out that if one uses the Rademacher distribution, there are only 2^G unique bootstrap samples and only 2^{G-1} unique t-statistics in absolute value. Hence, if G is small, then it will be the case that any given t-statistic can appear multiple times. This is problematic as the procedure supposes that if there are B bootstrap replications, there will be B unique t-statistics. As a result of this, he suggests a 6-point distribution which has moments that resemble those of the Rademacher distribution. This random variable takes on the values

$$v_g = \left(-\sqrt{\frac{3}{2}}, -\sqrt{\frac{2}{2}}, -\sqrt{\frac{1}{2}}, \sqrt{\frac{1}{2}}, \sqrt{\frac{2}{2}}, \sqrt{\frac{3}{2}} \right), \tag{18}$$

each occurring with equal probability. Such a distribution increases the number of bootstrap samples from 2^G to 6^G.

2.5 Multi-Way Clustering

The need for multi-way clustering arises when there is more than one variable where the disturbances display intra-cluster correlation. An example of this could be the need to cluster on both province and time variables. The most straight-forward method to deal with this problem is to add fixed effects corresponding to the variables which exhibit within-cluster correlation. However, as mentioned previously, this is not always feasible. Alternatively, it may be that adding fixed effects does not fully control for the intra-cluster correlation. The most well known procedure was proposed by Cameron, Gelbach, and Miller (2011). However, as their simulation evidence shows, these methods are very sensitive to certain properties in the data.

This analysis will be focused solely on two-way clustering. However, generalizing

this method for n-way clustering is a straightforward application of the two-way case. Essentially, this process is done by clustering on each variable individually and then clustering on the intersection of these groups, summing the two former and subtracting the latter. The setting differs slightly from (1) and is as follows. Consider the case where each observation belongs to more than one dimension of the data. That is, suppose that each individual belongs to a group $g \in (1, 2, ..., G)$ and also a group $h \in (1, 2, ..., H)$. Then one may rewrite the model in (1) as

$$\boldsymbol{y}_{igh} = \boldsymbol{x}'_{igh}\boldsymbol{\beta} + \boldsymbol{u}, \tag{19}$$

where it is assumed that for $i \neq j$, $E(\boldsymbol{u}_{igh}\boldsymbol{u}_{ig'h'}|\boldsymbol{x}_{igh}, \boldsymbol{x}_{ig'h'}) = 0$ unless $g = g'$ or $h = h'$. Then the idea is that the covariance matrix for $\hat{\boldsymbol{\beta}}$ can be computed as the sum of the one-way cluster covariance matrices.

The quantity we are interested in is $\hat{\text{var}}(\hat{\boldsymbol{\beta}})$, and we denote $\hat{\text{var}}^G(\hat{\boldsymbol{\beta}})$ as the one-way cluster robust estimate (CR_1) corresponding to the group of clusters $g \in (1, 2, ..., G)$. Cameron, Gelbach, and Miller (2011) suggest the following,

$$\hat{\text{var}}(\hat{\boldsymbol{\beta}}) = \hat{\text{var}}^G(\hat{\boldsymbol{\beta}}) + \hat{\text{var}}^H(\hat{\boldsymbol{\beta}}) - \hat{\text{var}}^{G\cap H}(\hat{\boldsymbol{\beta}}) \tag{20}$$

where $G \cap H$ must be subtracted to avoid double-counting. However, one must be aware that there are certain practical limitations to this method. That is, there is the possibility that some of the diagonal elements of (20) are negative, which would imply negative variance and that the covariance matrix is not positive definite. Of course, this is quite problematic, and Cameron, Gelbach, and Miller (2011) propose the following way to correct this issue. First, perform an eigenvalue decomposition on the matrix given in (18) to obtain $\hat{\text{var}}(\hat{\boldsymbol{\beta}}) = \boldsymbol{D}\boldsymbol{\Lambda}\boldsymbol{D}'$, where $\boldsymbol{D}$ is the matrix where the columns are the eigenvectors of (20), and $\boldsymbol{\Lambda} = \{\lambda_i\}$, the eigenvalues of (20). Next,

replace $\boldsymbol{\Lambda}$ with $\boldsymbol{\Lambda}^+ = \{\lambda_i^+\}$ where $\lambda_i^+ = \max\{0, \lambda_i\}$. Then finally, replace $\hat{\text{var}}(\hat{\boldsymbol{\beta}})$ with $\hat{\text{var}}^+(\hat{\boldsymbol{\beta}}) = \boldsymbol{D}\boldsymbol{\Lambda}^+\boldsymbol{D}'$. Additionally, the authors reported that, when the eigenvalues were negative, typically they were not large in absolute value.

2.6 Known Issues

As stated previously, the asymptotic theory for cluster robust estimation is predicated on the number of clusters becoming arbitrarily large. Angrist and Pischke (2009) proposed what they called the 'rule of 42', which essentially says that 42 is a large enough number of clusters to assume the asymptotic properties hold in finite samples. However, this is not generally true, and can only be relied upon in certain scenarios. There are a few well known cases where this 'rule' tends to fail. One such setting where this rule is violated is when there is large variation in the number of observations per cluster. MacKinnon and Webb (2016) find that given 50 clusters proportional to the populations of US states, the conventional CRVE does not perform well. However, they also show that the wild cluster bootstrap does alleviate most of this issue, and that its performance also depends on how much within-cluster correlation is present in the data.

Additionally, MacKinnon and Webb (2016) highlight the issues associated with the wild cluster bootstrap and the conventional CRVE when the clustered data incorporates treatment effects. They show that when using dummy variables for treatment, regardless of whether or not cluster size is equal, there can be severe over-rejection when the number of treated (or untreated) clusters is small. Additionally, it is shown that this is the case in a few different settings. These scenarios include, when one is using cluster-level treatments, and when using difference-in-difference regressions with and without fixed-effects. Furthermore, it is the case that the situation is least

favourable when some or all of the treated clusters are small.

In terms of two-way clustering, the problem appears to be quite sensitive to how many clusters there are in either of the groups upon which one wishes to cluster. In Cameron, Gelbach, and Miller (2011) the best results were achieved when there were an equal number of clusters per group and the number of clusters is large. That is, the best case was when $G = H = 100$. Aside from this case, they obtained fairly severe over-rejection when either, or both groups, had a small number of clusters. However, they did obtain an improvement on these rejection rates when they also included group specific fixed-effects.

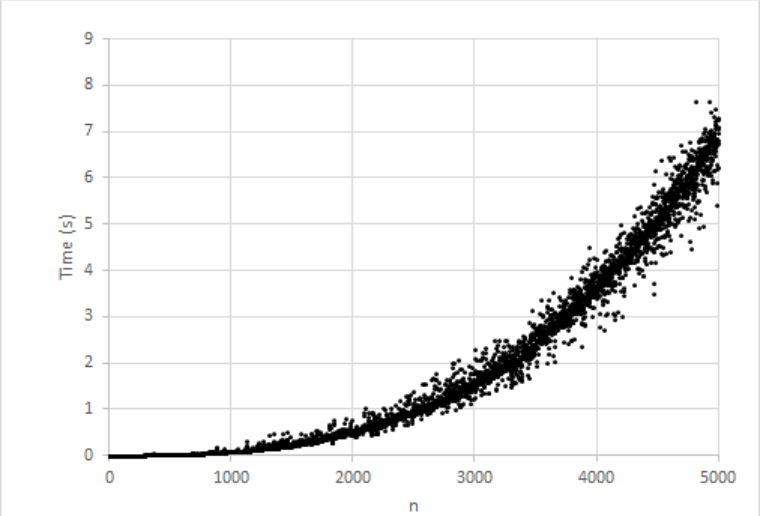
Finally, as will be outlined in the next section, a large number of these methods are not computationally trivial. That is, it may be the method that performs best is impractical in certain data sets. This can occur when the number of observations for any cluster is extraordinarily large. In fact, as shown in the next section, these methods actually become unfeasible in certain scenarios. As a result of this, one must be aware of the size of the clusters they are using, the amount of memory available on the system which one is using, and make the choice of which method to apply taking these into consideration.

3 Computational Properties of Cluster Robust Estimation

As mentioned previously, one potential drawback of certain cluster robust methods is that they possess an unavoidable matrix inversion, where the matrix needing to be inverted can be relatively large. The need for matrix inversion typically arises when solving a system of linear equations. However, in practice, computationally there are

much more efficient ways of solving such a system without the need for expensive matrix inversions. For example, performing an LU-decomposition or Gaussian elimination are ways to avoid computing a matrix inverse. In MATLAB when solving a system of the form $\boldsymbol{Ax} = \boldsymbol{b}$, the famous '\' command executes a series of algorithms which have been optimized to solve linear systems as efficiently as possible. This algorithm works by utilizing different approaches depending on whether the matrix is square, triangular, hermitian, or a number of other characteristics. By using this method, one can be certain that the system of linear equations is being solved in the manner which is best for the problem at hand and with no matrix inversion necessary.

Figure 1: Run time of matrix inversion in MATLAB



However, when the computation of an inverse of a matrix is unavoidable, operations tend to become computationally expensive quite quickly. In fact, as shown in the appendix, the approach MATLAB, Stata, and R use to compute the inverse of a matrix has a computational complexity of $\mathcal{O}(n^3)$. This fact implies that the run-time of such an algorithm increases cubically as n increases. In the case of the methods requiring such an inverse, CR_2, CR_3, and Young's corresponding methods, n here would be the number of observations for a given cluster. Figure 1 shows the

run-time for the matrix inversion algorithm implemented in MATLAB, as the size of the matrix increases.[8] The figure here was created by constructing an $n \times n$ matrix with each entry generated from a $N(0, 1)$ distribution, and then inverting that matrix. However, only the run-time corresponding to the matrix inversion itself was measured here. It is also worthwhile to note that this is an ideal situation for matrix inversion as the matrix is what one would term well-conditioned. That is, it is possible to have a 'near-singular' matrix which, while an inverse may exist analytically, it may be difficult or impossible to invert it numerically.

Figure 1 clearly shows the rate of increase of the computational run-time and it's nonlinear nature. Even using a value as low as $n = 5000$, we see that a matrix inversion will take nearly 10 seconds to be computed. If one were to have many large clusters, computing CR_2 or CR_3 could take upwards of several minutes, which is not highly problematic, but relatively inconvenient. Additionally, if one considers the fact that some of these matrices may not be well-conditioned, the run-time of these inversion algorithms can increase significantly. While this has the potential to be time consuming, an even more serious issue is how to actually store these $n \times n$ matrices.

When using a regression package such as Stata, the matrices themselves will be stored in RAM. As one can see in (6) the matrix which is being inverted and stored is $N_g \times N_g$. Suppose one had a data set with 100,000 observations where the number of observations per Canadian province is proportional to the province's population. In this case, the $\boldsymbol{M}_{gg} = \boldsymbol{I} - \boldsymbol{P}_{gg}$ matrix corresponding to observations from Ontario would be 38500×38500. If the matrix is double-precision, that is, each floating-point

[8]The specifications of the machine one is using plays a significant role in the computation time of these algorithms. In the case of this experiment the processor was a 3.4 Ghz. i7 2600 (4 cores, 8 GB) and the operating system was 64-bit Linux.

Table 1: RAM Required to Store $n \times n$ Matrices

	$n = 500$	$n = 2500$	$n = 10000$	$n = 25000$	$n = 40000$	$n = 50000$
Memory Required (GBs)	0.00186	0.00745	0.74505	4.65661	11.92093	18.62645

number occupies 8 bytes, then an $n \times n$ matrix will take up $n^2 \times 8/1024^3$ gigabytes of memory. Table 1 illustrates the different amounts of memory required to store double-precision matrices of different sizes.

As we can see, storing the M_{gg} matrix is clearly unfeasible if a given cluster is large enough. While of course there are systems with enough RAM to support such a calculation, this needs to be taken into account when one is considering which cluster robust method to employ. That is, if a specific data set possesses such a large cluster, then a high-powered computer, beyond the typical desktop, would be required. If one does not have such a system at their disposal, a compromise may have to be made whereby the method used is not necessarily the one best fit for the analysis, but the one which is computationally feasible. However, in the future as computation and memory become cheaper, this is a problem which should ultimately disappear. That being said, presently, it is certainly something which anyone applying cluster robust estimation needs to be aware of.

Finally, another potential issue, which arises in all of the methods discussed, is that of numerical accuracy. Numerical accuracy is a problem which arises any time one is performing arithmetic with floating-point values. Since floating-point numbers ultimately have a finite number of digits, rounding-off may be done at every stage of a numerical routine. When a numerical routine involves a significant number of steps, any round-off error has the potential to accumulate throughout the algorithm producing large errors which have compiled over time. For example, even something as seemingly innocuous as defining a variable in MATLAB as `x = 0.1` involves round-off

error, as 1/10 is an infinite series in binary. The result of this is that our variable `x` will be very close to 1/10, but not exact.

Currently, MATLAB, Stata, and R all use the IEEE standard 64-bit double (or 32-bit single if specified) format.[9] A simple example given by Golub and Van Loan (2013) shows, how using this format, one can encounter quite different results despite having mathematically identical expressions. If one considers the quadratic equation $x^2 - 2px - q = 0$, solving this using the quadratic formula will give one of the roots of this equation to be

$$r_1 = p - \sqrt{p^2 + q} \tag{21}$$

However, if one were to multiply (21) by $\frac{p+\sqrt{p^2+q}}{p+\sqrt{p^2+q}}$, then we would get a mathematically identical expression given by

$$r_2 = \frac{-q}{p + \sqrt{p^2 + q}}. \tag{22}$$

Then, using IEEE double floating-point arithmetic we would obtain, which one can check using the program of their choice, the following values:

$$\begin{aligned} r_1 &= -4.097819328308106 \times 10^{-8} \\ r_2 &= -4.050000332100021 \times 10^{-8} \end{aligned} \tag{23}$$

where the second is correct.[10]

While numerical accuracy is not something someone may necessarily be able to account for when applying any of the cluster-robust methods discussed in this analysis, it is useful to be aware of it. The most important take-away is that not all

[9] The IEEE is a computing standard which was developed in 1985 and is used almost unanimously in computing software.

[10] To obtain this one needs to set $q = 1$ and $p = 12345678$.

mathematically equivalent expressions will produce equivalent results numerically. Finite precision implies that round-off error is ubiquitous in the numerical routines which software packages make use of to compute these estimators. Not all routines are alike numerically, and some may perform more accurately than others. It may be the case that the idiosyncrasies in these routines produce significantly different results.

4 Monte Carlo Experiments

In this section, a series of Monte Carlo experiments are run which are meant to test and compare these estimators under different scenarios. While some of these estimators have been tested under these circumstances, many have not, nor have they been put in direct comparison with one another. Young (2016) compares the method he is proposing with the conventional CR_1, CR_2, and CR_3 estimators. However, the smallest number of clusters in any given experiment is $G = 11$, and the average number of clusters is $G = 211$. With G this large, it is hard to infer how well these estimators perform in the case where there is a small number of clusters. Cameron, Gelbach, and Miller (2008) performed similar experiments comparing various bootstrap methods, including the wild cluster bootstrap. One drawback of their simulation results is that they are based on a mere 1000 replications. With such a small number of replications, it is unlikely that such results would be reliable due to the amount of randomness associated with such a small sample size. To illustrate this, certain experiments were run multiple times with only 1000 replications and the results varied wildly. Examples of this are given in the appendix.

4.1 Number of Clusters

The first set of experiments will be testing all methods discussed previously for different numbers of clusters. Specifically, the number of clusters will be $G = 5$, $G = 10$, $G = 20$, and $G = 30$. The first two sets of tests will be using $R = 25000$ replications and equal cluster sizes of $N_g = 30$. However, the third set of experiments, with unequal cluster sizes, will be using $R = 10000$ replications as these tests are computationally intensive due to the relatively large cluster sizes.

4.1.1 Homoskedastic Disturbances

Following the experiment suggested by Cameron, Gelbach, and Miller (2008), the data were generated according to

$$y_{ig} = \beta_0 + \beta_1 x_{ig} + u_{ig} = \beta_0 + \beta_1(z_g + z_{ig}) + (\varepsilon_g + \varepsilon_{ig}). \tag{24}$$

Here z_g, z_{ig}, u_g, and u_{ig} are independent $N(0, 1)$ draws and the parameters are set as $\beta_0 = 0$ and $\beta_1 = 1$. The z_g and u_g are cluster specific standard normal random variables which induce the within-cluster correlation present in each generated sample. The results of this experiment are given below in Table 2, where WB_2 and WB_3 are the wild cluster bootstrap with the residual modifications corresponding to CR_2 and CR_3, respectively. YCR_1, YCR_2, and YCR_3 refer to the adjustments on the CR_1, CR_2, and CR_3 estimators proposed by Young (2016) and CR_2^{df} is CR_2 with the Bell and McCaffrey (2002) degrees of freedom correction.

Here we are performing the hypothesis test that $\beta_1 = 1$ which has been imposed in the DGP. Since the null hypothesis $\beta_1 = 1$ is true, what one would expect is that if the test has appropriate size, it will reject the null hypothesis exactly 5% of the

Table 2: Rejection Rates: Tests of Nominal Size 0.05 (Homoskedasticity)

		Clusters			
		$G = 5$	$G = 10$	$G = 20$	$G = 30$
Method	CR_1	0.0992	0.0896	0.0750	0.0672
	CR_2	0.0640	0.0692	0.0636	0.0602
	CR_3	0.0270	0.0438	0.0496	0.0497
	WB_1	0.0742	0.0567	0.0510	0.0513
	WB_2	0.0310	0.0483	0.0479	0.0492
	WB_3	0.0489	0.0523	0.0503	0.0508
	YCR_1	0.0678	0.0667	0.0615	0.0590
	YCR_2	0.0482	0.0556	0.0563	0.0558
	YCR_3	0.0317	0.0458	0.0511	0.0519
	CR_2^{df}	0.0934	0.0806	0.0678	0.0624

time. Of course, judging by the table, especially when the number of cluster is small, this is not the case. What can be seen is that, given the smallest number of clusters $G = 5$, the conventional CR_1 estimator under performs in comparison to every other estimator, rejecting almost 10% of the time. However, we can see that as the number of clusters increases, the rejection rate approaches 5%. This is consistent with the asymptotic theory and previous simulation experiments performed by Cameron, Gelbach, and Miller (2008).

Additionally, we notice that the wild cluster bootstrap methods improves upon the performance of CR_1, CR_2, and CR_3. In fact, when $G = 5$ we can see that WB_3 outperforms all other estimators rejecting 4.89% of the time. Additionally, we see that the Young (2016) corrections also improve the performance of the conventional standard errors. However, the difference in performance between Young's methods and the wild cluster bootstrap appears to be negligible when G is small. Additionally, once G increases it appears that the bootstrap methods produce the best results. This is especially the case when $G = 30$, as we can see that WB_1, WB_2, and WB_3 reject at a rate which is very close to the desired 5% level. CR_2 also very nearly rejects at exactly 5% in the cases where $G = 20$ and $G = 30$.

Considering how it is always possible that computing a CR_2 or CR_3 residual correction is numerically impractical, it could be that WB_1 is the best option given a small number of clusters. This is because once $G = 10$, WB_1 seems to perform reasonably well, and realistically, $G = 5$ is an extreme case which one is not necessarily likely to encounter in practice anyway. Also, in this analysis, the auxiliary distribution being used in the wild bootstrap procedure is the Rademacher distribution, which Webb (2014) has shown is not necessarily the best choice when G is quite small. Hence, this means that one may see better results using a different auxiliary random variable, such as the one given in (18).

4.1.2 Heteroskedastic Disturbances

Now we consider disturbances which are correlated within clusters and heteroskedastic. In this case the data are generated by

$$y_{ig} = \beta_0 + \beta_1 x_{ig} + u_{ig} = \beta_0 + \beta_1(z_g + z_{ig}) + (\varepsilon_g + \varepsilon_{ig}) \tag{25}$$

with z_g, z_{ig}, ε_g being independent $N(0,1)$ draws, but instead $\varepsilon_{ig} \sim N(0, 9\times(z_g+z_{ig})^2)$, and again $N_g = 30$, and there are $R = 25000$ replications. This specification introduces additional complications by adding heteroskedasticity along with the within-cluster correlation. In a scenario such as this, the traditional OLS covariance matrix estimator assuming iid disturbances performs quite poorly, as shown in Cameron, Gelbach, and Miller (2008), rejecting nearly 30% of the time regardless of the number of clusters. Table 3 shows the result of this experiment.

As we see in the table, the result with heteroskedastic disturbances is similar to that of the case of homoskedasticity, however, with a slight loss in performance. When

Table 3: Rejection Rates: Tests of Nominal Size 0.05 (Heteroskedasticity)

		Clusters			
		$G = 5$	$G = 10$	$G = 20$	$G = 30$
Method	CR_1	0.0901	0.0841	0.0757	0.0665
	CR_2	0.0744	0.0751	0.0647	0.0610
	CR_3	0.0450	0.0512	0.0505	0.0522
	WB_1	0.0628	0.0520	0.0515	0.0487
	WB_2	0.0256	0.0442	0.0486	0.0465
	WB_3	0.0555	0.0547	0.0502	0.0490
	YCR_1	0.0598	0.0637	0.0629	0.0582
	YCR_2	0.0557	0.0585	0.0571	0.0558
	YCR_3	0.0503	0.0540	0.0524	0.0541
	CR_2^{df}	0.1059	0.0841	0.0693	0.0642

the number of cluster is smallest, that is $G = 5$, YCR_3 seems to perform the best. However, as the number of clusters increases, the wild cluster bootstrap methods perform the best in comparison to the other estimators. Due to the computational complexity of Young's methods and the residual modifications, they have the possibility of being quite expensive to compute depending on one's data set. As a result of this, the wild bootstrap with no residual modifications appears to be the best option for $G \geq 10$. However, if the data set being used does not have any clusters with an exceptionally large number of observations, this is not a major concern.

Overall, as we would expect, the results do not change all that much once heteroskedasticity is introduced. The reason for this is, as mentioned before, that these methods control for disturbance heteroskedasticity across clusters and within-cluster correlation. The one exception to this, as Young (2016) outlines, is that the bias and effective degrees of freedom calculations are being done under the assumption of iid normal disturbances. However, as he explains, even in the case of non-iid disturbances, it can still improve inference in comparison to the CR_1, CR_2, and CR_3 estimators. As a result of this, despite the fact that the disturbances are heteroskedastic and correlated within-clusters, Young's adjustments do reject at a rate relatively

close to what one would desire.

4.2 Unequal Cluster Sizes

In this scenario we consider data generated by the process given in (25). However, now the cluster sizes are no longer equal. In this experiment we look at three different scenarios, one with $G = 13$ and cluster sizes proportional to Canadian provinces and territories, one with $G = 10$ with cluster sizes proportional to only the Canadian provinces, and one with $G = 50$ with cluster sizes proportional to US states. Each experiment is done with a total of $N = 2000$ observations. This implies that, in the case of Canada including the territories, the smallest clusters only have two observations and the largest cluster, corresponding to Ontario, has 385 observations. Similarly for the United States, the smallest cluster only has 4 observations while the largest has 244.

The case where cluster sizes are proportional to Canadian provinces imposes two potential issues for these methods, as it is a relatively small number of clusters and the cluster sizes vary significantly. The case of $G = 50$, corresponding to the US, is interesting because of Angrist and Pischke's 'rule of 42', which states that $G = 42$ is sufficient to consider the number of clusters large. However, as shown in MacKinnon and Webb (2016), this fails to hold for the wild bootstrap and the conventional CRVE. Here we wish to test how this variation in cluster sizes affects the performance of the methods under consideration in this analysis.

Table 4 shows the results of these experiments where the starred Canadian provinces entry refers to the case of $G = 10$, where the territories have been removed. At first glance, what we see is that removing the clusters corresponding to the territories slightly improves the performance of a few estimators, and worsens it for others.

Table 4: Rejection Rates: Tests of Nominal Size 0.05 (Unequal Cluster Sizes)

		Cluster Size Proportional To:		
		Canadian Provinces	Canadian Provinces*	American States
Method	CR_1	0.2259	0.2004	0.0982
	CR_2	0.1172	0.0995	0.0770
	CR_3	0.0379	0.0271	0.0582
	WB_1	0.0738	0.0691	0.0549
	WB_2	0.0561	0.0582	0.0550
	WB_3	0.0270	0.0244	0.0476
	YCR_1	0.1020	0.0912	0.0748
	YCR_2	0.0443	0.0396	0.0628
	YCR_3	0.0348	0.0182	0.0538
	CR_2^{df}	0.0921	0.0850	0.0740

Specifically, we see that CR_1, CR_2, WB_1, and YCR_1 perform slightly better once the smallest clusters, representing the territories, have been removed. Generally, however, the performance of all estimators suffers compared to the case where cluster sizes are equal. However, with that being said, it would appear that WB_2 provides the most reliable inference in the case of Canadian provinces. However, in the case of cluster sizes proportional to American states, WB_3 and YCR_3 perform the best.

Additionally, we see that the Young (2016) adjustments tend to perform considerably worse than in the constant cluster size case. This is particularly noticeable in the Canadian provinces case where YCR_1 rejects roughly 10% of the time. Furthermore, we see that, for the case of the American states, while the number of clusters $G = 50$ could be considered large, almost every estimator performs worse than in the case where $G = 30$ and the number of observations per cluster was equal. These results are consistent with what was found in MacKinnon and Webb (2016), whereby the variation in cluster sizes produced relatively poor results despite the number of clusters being large. That being said, WB_2 appears to perform the best in the context of Canadian provinces, and WB_3 or YCR_3 performs the best when $G = 50$. However,

in the case of cluster sizes proportional to American states WB_1 performs relatively well, and may be preferred due to its computational efficiency.

5 Two-Way Clustering

When Cameron, Gelbach, and Miller (2011) initially proposed the two-way clustering procedure as given in (20), the one-way cluster robust estimator being used was CR_1. What they propose does not indicate that using CR_1 is the only covariance matrix estimator which would be valid in this procedure. As they show, an alternative way to write out the two-way cluster covariance matrix estimator given by (20) is to use an $N \times N$ indicator matrix $\boldsymbol{S}^G$. This matrix, $\boldsymbol{S}^G$, is such that the ij^{th} entry will be equal to one if the ith and jth observation belong to the same cluster and will be equal to zero otherwise. Using this idea implies we can redefine the original CRVE given in (4) by the following way

$$\begin{aligned} \text{CR}_1 &= C_{\text{CR}_1}(\boldsymbol{X}'\boldsymbol{X})^{-1}\Big(\sum_{g=1}^{G}\boldsymbol{X}_g'\hat{\boldsymbol{u}}_g\hat{\boldsymbol{u}}_g'\boldsymbol{X}_g\Big)(\boldsymbol{X}'\boldsymbol{X})^{-1} \\ &= C_{\text{CR}_1}(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}(\hat{\boldsymbol{u}}\hat{\boldsymbol{u}}'.\times\boldsymbol{S}^G)\boldsymbol{X}(\boldsymbol{X}'\boldsymbol{X})^{-1}, \end{aligned} \tag{26}$$

where $.\times$ is element-by-element multiplication. Using this interpretation will now be useful for the two-way case. Here we will go into slightly more detail than what Cameron, Gelbach, and Miller (2011) provided in order to clearly illustrate the process.

Suppose instead, if we have two levels of the data upon which we wish to cluster, then we define the indicator matrix $\boldsymbol{S}^{GH}$ corresponding to both groups. This clearly illustrates how the two-way covariance matrix estimator may be written as a sum of each of its one-way cluster robust components, as one can observe that $\boldsymbol{S}^{GH} =$

$\boldsymbol{S}^G + \boldsymbol{S}^H - \boldsymbol{S}^{G\cap H}$. Hence, one may rewrite (26) as

$$\begin{aligned}
\mathrm{CR}_1{}^{\text{two-way}} &= C_{\mathrm{CR}_1}(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'(\hat{\boldsymbol{u}}\hat{\boldsymbol{u}}'.\times \boldsymbol{S}^{GH})\boldsymbol{X}(\boldsymbol{X}'\boldsymbol{X})^{-1} \\
&= C_{\mathrm{CR}_1}(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'(\hat{\boldsymbol{u}}\hat{\boldsymbol{u}}'.\times (\boldsymbol{S}^{G}+\boldsymbol{S}^{H}-\boldsymbol{S}^{G\cap H}))\boldsymbol{X}(\boldsymbol{X}'\boldsymbol{X})^{-1} \\
&= C_{\mathrm{CR}_1}[(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'(\hat{\boldsymbol{u}}\hat{\boldsymbol{u}}'.\times \boldsymbol{S}^{G})\boldsymbol{X}(\boldsymbol{X}'\boldsymbol{X})^{-1}+ \\
&\qquad (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'(\hat{\boldsymbol{u}}\hat{\boldsymbol{u}}'.\times \boldsymbol{S}^{H})\boldsymbol{X}(\boldsymbol{X}'\boldsymbol{X})^{-1} \\
&\qquad -(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'(\hat{\boldsymbol{u}}\hat{\boldsymbol{u}}'.\times \boldsymbol{S}^{G\cap H})\boldsymbol{X}(\boldsymbol{X}'\boldsymbol{X})^{-1}].
\end{aligned} \tag{27}$$

For the degrees of freedom adjustment, C_{CR_1}, Cameron, Gelbach, and Miller (2011) propose using three different adjustments, one corresponding to each group of clusters. That is, one would have $C^G_{\mathrm{CR}_1} = \frac{G}{G-1}\frac{N-1}{N-k}$, $C^H_{\mathrm{CR}_1} = \frac{H}{H-1}\frac{N-1}{N-k}$, and $C^I_{\mathrm{CR}_1} = \frac{I}{I-1}\frac{N-1}{N-k}$ where I is the number of clusters formed from the intersection of the two groups. Hence, instead of C_{CR_1} being applied to all three terms in (27) each term would be multiplied by its respective degrees of freedom adjustment. An alternative calculation which has been proposed would be to use only one adjustment, $C_{\mathrm{CR}_1} = \frac{J}{J-1}\frac{N-k}{N-1}$, where $J = \min(G, H)$. However, this will not be used in this analysis.

As shown in the previous section, CR_1 is outperformed by every other estimator in the case where G is small. In fact, for the extreme cases such as $G = 5$ and $G = 10$, CR_1 performs quite poorly, rejecting nearly 10% of the time. Since the cases where the two-way cluster robust estimator fails to perform well is typically where either G or H is small, we will apply similar adjustments as those proposed by Bell and McCaffrey (2002) to the two-way case. The reason for doing this is that these corrections are meant to adjust for the bias present in the one-way CR_1 estimator, which is highly problematic when there is a small number of clusters.

These adjustments will be made by replacing $\hat{\boldsymbol{u}}$ in (27) by $\dot{\boldsymbol{u}} = \boldsymbol{M}_{\boldsymbol{X}}^{-1/2}\boldsymbol{u}$ in the

CR_2 case and $\ddot{\boldsymbol{u}} = \boldsymbol{M}_{\boldsymbol{X}}^{-1}\boldsymbol{u}$ in the CR_3 case. As before, the degrees of freedom adjustment is no longer present in the CR_2 and CR_3 estimators. The intuition for why these adjustments improve inference is the same as in the one-way case, where the original CR_1 estimator over-rejects when the number of clusters is small. The CR_2 and CR_3 adjustments systematically change the size of the standard errors by rescaling the residuals and hence reduce the rejection frequency. Of course, as mentioned previously, the computational limitations of these methods will still be present as one is still required to perform an unavoidable matrix inversion.

5.1 Two-Way Monte Carlo Experiments

This exercise is a two-way random effects model for the disturbances which also includes a heteroskedastic component. Following Cameron, Gelbach, and Miller (2011), the data were generated according to

$$y_{igh} = \beta_0 + \beta_1 x_{1igh} + \beta_2 x_{2igh} + u_{igh}, \tag{28}$$

with $\beta_0 = \beta_1 = \beta_2 = 1$. Using a rectangular design, this implies there are $G \times H$ observations implying the i subscript is not necessary in this scenario. Here $x_{1gh} = z_g + z_{1gh}$ and $x_{2gh} = z_h + z_{2gh}$ where each of the z_{1gh} and z_{2gh} are an iid $N(0,1)$ draw with z_g and z_h being a cluster specific $N(0,1)$ draw. Additionally, $u_{gh} = \varepsilon_g + \varepsilon_h + \varepsilon_{gh}$ where ε_g and ε_h are $N(0,1)$ and ε_{gh} induces conditional heteroskedasticity as $\varepsilon_{gh} \sim N(0, |x_{1gh} \times x_{2gh}|)$.

The first case being considered will be where $G = H$ and the number of clusters will range from 10 to 50 in each group. Next, we will consider the case where the number of clusters per group is not equal with $G < H$ in each case. Since this

Table 5: Rejection Rates: Tests of Nominal Size 0.05 (Equal Number of Clusters)

		Cluster Size									
		$G = H = 10$		$G = H = 20$		$G = H = 30$		$G = H = 40$		$G = H = 50$	
Method	CR_1	0.1628	0.1533	0.1066	0.1038	0.0846	0.0892	0.0743	0.0801	0.0743	0.0739
	CR_1^{df}	0.1230	0.1157	0.0882	0.0858	0.0736	0.0780	0.0668	0.0713	0.0671	0.0668
	CR_2	0.1387	0.1294	0.0937	0.0915	0.0770	0.0811	0.0682	0.0741	0.0687	0.0684
	CR_2^{df}	0.1031	0.0950	0.0758	0.0757	0.0666	0.0686	0.0613	0.0661	0.0622	0.0628
	CR_3	0.0862	0.0790	0.0708	0.0710	0.0636	0.0655	0.0589	0.0646	0.0607	0.0615
	CR_3^{df}	0.0603	0.0530	0.0569	0.0560	0.0542	0.0561	0.0518	0.0575	0.0544	0.0557

experiment has two right hand side independent variables, we will be considering inference based on the two coefficients $\hat{\beta}_1$ and $\hat{\beta}_2$ testing whether or not $\beta_1 = 1$ and $\beta_2 = 1$. Additionally, two critical values will be used, the first are from the $t(N-k)$ distribution and the second are from the $t(\min(G,H)-1)$ distribution. What one would expect is that the latter will outperform the former, and this is indeed the result we see in the next section.

5.1.1 Equal Number of Clusters

The first set of experiments was performed for the case where $G = H$, and the results of the experiment can be found in Table 5. The results are based on $R = 25000$ replications and the tests have a nominal size of 5%. This implies that since the null hypothesis is true, a test with proper size should reject exactly 5% of the time. In the table, the superscript *df* refers to the critical values based on Student's t-distribution with $\min(G,H)-1$ degrees of freedom, and no subscript simply refers to the one with $N-k$ degrees of freedom. Additionally, the first value refers to the rejection rate for $\hat{\beta}_1$, while the second refers to the value for $\hat{\beta}_2$.

Immediately we notice, as one would expect, that the case using $\min(G,H)-1$ t-distribution outperforms that of the $N-k$ degrees of freedom distribution in every scenario. Additionally, as is consistent with what was found in Cameron, Gelbach,

and Miller (2011), the performance of the CR_1 two-way estimator depends significantly on the size of G and H. That is, CR_1 severely over-rejects when G and H are small. However, it approaches 5% as the number of clusters becomes large. In fact, in the case where $G = H = 10$ essentially all estimators being considered do not perform well. The only exception to this is CR_3^{df} which rejects 10% less often than CR_1, rejecting relatively close to 5% of the time.

As G and H increase in size, we see that essentially all of the estimators begin to reject less frequently. However, even when $G = H = 50$, we find that CR_1 is still slightly over-rejecting and as a result is outperformed by CR_2 and CR_3, although by not nearly as much as the case when G and H were small. Perhaps again what is most interesting, and has been seen in the previous section, is that the rejection rate of CR_3, especially with the $\min(G, H) - 1$ degrees of freedom, seems to depend very little on the number of clusters. However, with that being said, it seems quite clearly that inference could be improved quite significantly by opting to perform the rescaling of the residuals.

5.1.2 Unequal Number of Clusters

In these experiments the same data generating process was used as in the previous subsection with the slight difference being that in each case $G < H$. A rectangular design is still used implying that the total number of observations is still $N = G \times H$ with exactly one observation corresponding to each (g, h) pair. What we should expect from these experiments is that inference on $\hat{\beta}_2$ should be more reliable than that of $\hat{\beta}_1$. This is because in the DGP, the independent variable which has cluster g specific random components, and thus a smaller number of clusters, is the one corresponding to $\hat{\beta}_1$. Table 6 which shows the results of these experiments confirms

Table 6: Rejection Rates: Tests of Nominal Size 0.05 (Unequal Number of Clusters)

		Cluster Size							
		$G = 10$	$H = 50$	$G = 20$	$H = 50$	$G = 30$	$H = 50$	$G = 40$	$H = 50$
Method	CR_1	0.1411	0.0939	0.1018	0.0780	0.0870	0.0778	0.0785	0.0741
	CR_1^{df}	0.0971	0.0580	0.0828	0.0612	0.0736	0.0659	0.0712	0.0665
	CR_2	0.1149	0.0879	0.0886	0.0730	0.0775	0.0717	0.0728	0.0698
	CR_2^{df}	0.0757	0.0528	0.0717	0.0563	0.0658	0.0610	0.0649	0.0621
	CR_3	0.0750	0.0711	0.0703	0.0628	0.0640	0.0629	0.0628	0.0623
	CR_3^{df}	0.0480	0.0412	0.0556	0.0484	0.0539	0.0535	0.0552	0.0539

this expectation. Additionally, as in the previous case these results are based on $R = 25000$ replications.

In this scenario, the results differ somewhat significantly from the case where $G = H$. The rejection rate for the hypothesis test that $\beta_1 = 1$ is larger than that of the rejection rate for the hypothesis that $\beta_2 = 1$, which is what was expected. As mentioned previously, since $G < H$ in each set of experiments and the performance of these estimators depends quite heavily on how large G and H are, this result is not surprising. This is because if we recall that $x_{1gh} = z_{1gh} + z_g$, this implies that the independent variable x_{1gh} with corresponding parameter β_1 has a cluster specific element for each cluster $g \in (1, 2, \cdots, G)$. However, aside from this, the result is quite similar to the case with $G = H$.

As before, we see that when G is small, CR_1 heavily over-rejects and CR_2 improves somewhat on this over-rejection, but CR_3^{df} is by far and away the best. Similarly, we see that using $\min(G, H) - 1$ degrees of freedom produces much better results than using $N-k$ degrees of freedom. One interesting result is that, in the case where $G = 10$ and $H = 50$, we see that CR_3^{df} actually slightly under-rejects for both parameters, which was never the case when $G = H$. Again, as before, G increasing seems to have little effect on the performance of CR_3^{df} relative to the improvement seen in CR_1 and even CR_2. However, with that being said, it appears still that the best choice is in

fact CR_3^{df} when the number of clusters per group is unequal.

5.2 Drawbacks and Further Extensions

Here I will highlight a few issues associated with two-way clustering which are mainly logistical. Additionally I will discuss other ways one could approach two-way clustering by computing two-way wild cluster bootstrap standard errors.

5.2.1 Drawbacks

As highlighted previously, we know the computational complications which can arise when computing the residual adjustments $\dot{\boldsymbol{u}}_g = (\boldsymbol{M}_{gg})^{-1/2}\hat{\boldsymbol{u}}$ and $\ddot{\boldsymbol{u}}_g = (\boldsymbol{M}_{gg})^{-1}\hat{\boldsymbol{u}}$ if any given cluster is too large. However, this problem may only be exacerbated by the fact that we are required to cluster on $G \cap H$ as well, implying that one may be required to perform considerably more matrix inversions than in the case of the one-way clustering. While the simulation evidence suggests that CR_3^{df} is the best candidate for two-way clustering, these computational limitations need to be kept in mind when looking to employ this method. That is, if one knows that any group $g \in (1, 2, \cdots, G)$, or $h \in (1, 2, \cdots, H)$, or their intersection is particularly large, CR_1^{df} may be the only appropriate way to approach such a problem. However, that being said, as Cameron, Gelbach, and Miller (2011) show, the performance of CR_1 can be improved by adding group specific fixed-effects as well.

Another issue is that presently no major regression package supports two-way clustering with CR_2 or CR_3. The experiments performed in this analysis were done using code which I have written in MATLAB specifically for the purpose of these Monte Carlo experiments.[11] However MATLAB is not necessarily the most user-

[11]This was done by modifying a MATLAB script originally written by Daniel Taylor, which

friendly way, nor the most popular way, to perform OLS due to the lack of general implementation. In terms of ease of use, a program like Stata, or even R, is much more straightforward when dealing with data and running regressions. However, due to the fact that there is already Stata implementation for two-way standard errors using CR_1, it should be relatively straightforward to adapt this procedure to allow for the residual adjustments as well.

5.2.2 Extensions

A natural extension of the methods discussed in this section would be to ask how one would employ a two-way wild cluster bootstrap. Here it is useful to now make the distinction between the bootstrap-t and the bootstrap-se procedure as outlined in Cameron, Gelbach, and Miller (2008). In a bootstrap-se procedure one computes a bootstrap estimate of the standard error which would replace the basic OLS standard error in the denominator of the t-statistic or the Wald statistic. Here we assume again that we have a linear model as given in (1), and the disturbances are correlated within-clusters. We also assume, without a loss of generality, that we are interested in testing whether or not the last coefficient, β_k, is significantly different from a null value β_k^0. The following is an example of how one may compute a wild cluster bootstrap-se procedure.

1. Estimate (1) by OLS.

2. Re-estimate the original model with the restriction $\beta_k = 0$ imposed. This is done to obtain restricted residuals $\tilde{\boldsymbol{u}}$ and restricted parameter estimates $\tilde{\boldsymbol{\beta}}$.

3. In each of the $b \in B$ bootstrap replications generate a new set of bootstrap

computed two-way cluster robust standard errors using CR_1. The original two-way cluster robust estimation using CR_1 script can be found on his homepage.

dependent variables using the bootstrap DGP

$$\boldsymbol{y}_{ig}^{*b} = \boldsymbol{X}_{ig}\tilde{\boldsymbol{\beta}} + \tilde{\boldsymbol{u}}_{ig} v_g^{*b} \tag{29}$$

where v_g is a Rademacher random variable specific to each cluster which takes 1 or -1 with equal probability.

4. For each $b \in B$ estimate (1) using $\boldsymbol{y}^{*b}$ as the dependent variable to obtain the parameter estimate $\hat{\boldsymbol{\beta}}^{*b}$.

5. Form the test statistic $t_{bse} = \frac{\hat{\beta}_k - \beta_k^0}{se_{\beta_k,B}}$ where

$$se_{\beta_k,B} = \left(\frac{1}{B-1} \sum_{b=1}^{B} (\hat{\beta}_k^{*b} - \bar{\hat{\beta}}_k^{*})^2 \right)^{1/2} \tag{30}$$

and

$$\bar{\hat{\beta}}_k^{*} = \frac{1}{B} \sum_{b=1}^{B} \hat{\beta}_k^{*b}. \tag{31}$$

This procedure differs from the bootstrap-t procedure at the fourth step where one is no longer required to compute a bootstrap t-statistic on every replication, but rather, just obtain the parameter estimate $\hat{\boldsymbol{\beta}}^{*b}$. As Cameron, Gelbach, and Miller (2008) outline, the wild cluster bootstrap-t is preferred to the wild cluster bootstrap-se for one primary reason. Both methods are asymptotically valid. However, bootstrap-se procedures do not provide asymptotic refinement which causes issues when the number of clusters is small. While the bootstrap-t procedure may be preferred for this reason, it is not clear how this method could be employed in the two-way cluster case.

One could quite easily adapt the bootstrap-se procedure to the two-way case by simply calculating three sets of bootstrap standard errors. That is, suppose we have

two groups, G and H. We could then calculate the one-way standard errors $se^{G}_{\beta_k,B}$, $se^{H}_{\beta_k,B}$, and $se^{G\cap H}_{\beta_k,B}$, and then form the two-way cluster bootstrap standard error

$$se^{2-way}_{\beta_k,B} = se^{G}_{\beta_k,B} + se^{H}_{\beta_k,B} - se^{G\cap H}_{\beta_k,B}. \tag{32}$$

This method would provide two-way wild cluster bootstrap standard errors using the less preferred bootstrap-se procedure. While this procedure is relatively straightforward, such is not the case when trying to extend the wild cluster bootstrap-t procedure to the two-way case.

The problem arises due to the fact that we are no longer simply calculating a parameter estimate on each bootstrap replication in order to construct a standard error. Rather, there is a standard error being constructed on each bootstrap replication in the denominator of the bootstrap t-statistic. What this implies is that, if we first clustered on G, we would obtain a set of bootstrap t-statistics where the bootstrap DGP was generated by applying an auxiliary random variable to the restricted residuals from each cluster $g \in (1, 2, \cdots, G)$. Similarly, one would compute a set of bootstrap t-statistics corresponding to H and $G \cap H$. This implies that one would have three sets of t-statistics and three bootstrap p-values. Essentially, there is no way to compute three separate standard errors in order to employ the two-way cluster standard error calculation. Despite this, a future analysis could involve using the two-way bootstrap-se procedure and comparing its performance to the two-way CR_1, CR_2, and CR_3 methods.

6 Conclusion

Cluster robust estimation has been present in econometric literature for over three decades. During this period many different approaches have been taken to improve inference when the errors exhibit within-cluster correlation. However, one major drawback of some of the methods which have been proposed is that they are quite computationally intensive. In this analysis, these issues have been discussed in detail highlighting some of the primary problems one encounters numerically when using these estimation techniques. Specifically, the problem with the inversion of large matrices has been discussed in detail where it is clear that, depending on the data set one is using, certain estimators become computationally impractical. In addition to this, not all numerical routines are equal and due to the nature of floating-point operations, they may produce different results despite being mathematically equivalent.

The main result from the first set of Monte Carlo simulations is that, in the extreme case of $G < 10$, WB_3 provides the most reliable inference in the case of homoskedasticity, and YCR_3 in the case of heteroskedasticity. Additionally, the wild cluster bootstrap procedure without the residual adjustments (WB_1) seems to be a relatively good alternative if the computational limitation of inverting the M_{gg} matrices is present. Once $G > 10$, the performance of WB_1 seems to be superior, especially when considering how computationally cheap it is relative to the residual adjustments and effective degrees of freedom corrections. However, applying the residual corrections to the wild cluster bootstrap procedure does seem to improve performance over WB_1. This result is especially apparent when $G = 5$, where WB_3 appears to correct for the over-rejection present in WB_1. The case where there is variation in the size of the clusters severely impacts the performance of almost every estimator being considered. This is especially true when G is small.

The extension of two-way clustering, where the Bell and McCaffrey adjustments was applied to the two-way case provided favorable results. Aside from the computational limitations, it appears that applying CR_2 or CR_3 to the two-way standard error procedure is always an improvement over CR_1. Specifically, the simulations provide evidence that two-way CR_3, and using critical values from a t-distribution with $\min(G, H) - 1$ degrees of freedom, produces the best results. Additionally, a bootstrap-se procedure was discussed for the case of two-way cluster robust estimation. That is, a way that one could compute two-way wild cluster bootstrap standard errors to be used in t or Wald statistics.

In order to obtain reliable inference when one is faced with the problem of having disturbances correlated within-clusters, one is required to choose the method which is most appropriate for the problem at hand. While it is not always clear which approach one should take, this analysis should serve to assist anyone who requires cluster robust estimation in the one-way or two-way case. As we have seen, the most commonly used estimator CR_1, which is the default option in Stata, under-performs in every scenario when compared to other options. This fact is most apparent in the two-way case where CR_2 and CR_3 adjustments vastly improve the performance of the two-way cluster standard error estimates.

7 Bibliography

[1] Angrist, Joshua D., and Jorn-Steffen Pischke (2008) *Mostly Harmless Econometrics: An Empiricist's Companion* (Princeton: Princeton University Press)

[2] Bell, Robert M., and Daniel F. McCaffrey (2002) 'Bias reduction in standard errors for linear regression with multi-stage samples.' *Survey Methodology* 28(2), 169-181

[3] Bertrand, Marianne, Esther Duflo, and Sendhil Mullainathan (2004) 'How much should we trust differences-in-differences estimates?' *The Quarterly Journal of Economics* 119(1), 249-275

[4] Bester, C. Alan, Timothy G. Conley, and Christian B. Hansen (2011) 'Inference with dependent data using cluster covariance estimators.' *Journal of Econometrics* 165(2), 137-151

[5] Cameron, A. Colin, Jonah B. Gelbach, and Douglas L. Miller (2008) 'Bootstrap-based improvements for inference with clustered errors.' *The Review of Economics and Statistics* 90(3), 414-427

[6] Cameron, A. Colin, Jonah B. Gelbach, and Douglas L. Miller (2011) 'Robust inference with multiway clustering.' *Journal of Business & Economic Statistics* 29(2), 238-249

[7] Cameron, A. Colin, and Douglas L. Miller (2015) 'A practitioner's guide to cluster robust inference.' *Journal of Human Resources* 50, 312-372

[8] Carter, Andrew V., Kevin T. Schnepel, and Douglas G. Steigerwald (2015) 'Asymptotic behavior of a t test robust to cluster heterogeneity.' Technical Report, University of California, Santa Barbara.

[9] Conley, Timothy G., and Christopher R. Taber (2011) 'Inference with "difference in differences" with a small number of policy changes' *The Review of Economics and Statistics* 93(1), 113-125

[10] Cormen, Thomas H., Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein (2001) *Introduction to Algorithms* (Cambridge: MIT Press)

[11] Donald, Stephen G., and Kevin Lang (2007) 'Inference with difference-in-differences and other panel data.' *The Review of Economics and Statistics* 89(2), 221,233

[12] Golub, Gene H., and Charles F. Van Loan (2013) *Matrix Computations (4th Edition)* (Baltimore: The Johns Hopkins University Press)

[13] Gow, Ian D., Gaizka Ormazabal, and Daniel J. Taylor (2010) 'Correcting for cross-sectional and time-series dependence in accounting research.' *The Accounting Review* 85(2), 483-512

[14] Imbens, Guido W., and Michal Kolesar (2012) 'Robust standard errors in small samples: Some practical advice.' Working Paper 18487, National Bureau of Economic Research, October

[15] Kézdi, Gábor (2004) 'Robust standard error estimation in fixed-effect panel models.' *Hungarian Statistical Review* 9, 96-116

[16] Kloek, T. (1981) 'OLS estimation in a model where a microvariable is explained by aggregates and contemporaneous disturbances are equicorrelated.' *Econometrica* 49(1), 205-207

[17] Liang, Kung-Yee and Scott L. Zeiger (1986) 'Longitudinal data analysis using generalized linear models.' *Biometrika* 73(1), 13-22

[18] MacKinnon, James G. (2015) 'Wild cluster bootstrap confidence intervals.' *L'Actualité Économique* 91(1), 11-33

[19] MacKinnon, James G., and Matthew D. Webb (2016) 'Wild bootstrap inference for wildly different cluster sizes.' *Journal of Applied Econometrics*, forthcoming.

[20] MacKinnon, James G., and Halbert White (1985) 'Some heteroskedasticity consistent covariance matrix estimators with improved finite sample properties.' *Journal of Econometrics* 29(3), 305-325

[21] Moulton, Brent R. (1990) 'An Illustration of a pitfall in estimating the effects of aggregate variables on micro unites.' *Review of Economics & Statistics* 72(2), 334-338

[22] Pustejovsky, James E., and Elizabeth Tipton (2016) 'Small sample methods for cluster-robust variance estimation and hypothesis testing in fixed effects models.' Working Paper, January 2016

[23] Satterthwaite, F. (1946) 'An approximate distribution of estimates of variance components' *Biometrics* 2, 110-114

[24] Tipton, Elizabeth (2015) 'Small sample adjustments for robust variance estimation with meta-regression.' *Psychological Methods* 20(3), 375-393

[25] Webb, Matthew D. (2014) 'Reworking wild bootstrap based inference for clustered errors.' Working Paper 1315, Queen's University, Department of Economics, August

[26] Young, Alwyn (2016) 'Improved, nearly exact, statistical inference with robust and clustered covariance matrices using effective degrees of freedom corrections.' Technical Report, London School of Economics, November

8 Appendix

A.1 The Moulton Factor

Here we follow Moulton (1990) and fill in some of the details of the algebra to derive the previously mention Moulton factor. First consider the following model

$$\begin{aligned} \boldsymbol{y} &= \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{u} \\ E(\boldsymbol{u}) &= 0 \\ E(\boldsymbol{u}\boldsymbol{u}') &= \sigma^2\boldsymbol{V} = \sigma^2[(1-\rho)\boldsymbol{I}_N + \rho\boldsymbol{S}\boldsymbol{S}'] \end{aligned} \tag{A1}$$

where $\boldsymbol{S}$ is a $N \times G$ indicator matrix for membership of a given cluster $g \in (1, 2, \cdots, G)$ and ρ is the within-cluster correlation. Additionally we are assuming that $\boldsymbol{\beta}$ is $k \times 1$ and $\boldsymbol{X}$ is $N \times k$. Here we wish to know how the true covariance matrix corresponding to this model compares to that of the standard OLS covariance matrix. Estimating this model by OLS, we know that the estimate of the parameter vector is given by

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{y}. \tag{A2}$$

Inserting $\boldsymbol{y}$ from (A1) implies that

$$\hat{\boldsymbol{\beta}} - \boldsymbol{\beta} = (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{u} \tag{A3}$$

and since we are assuming here that $\hat{\boldsymbol{\beta}}$ is unbiased, var$(\hat{\boldsymbol{\beta}})$ is simply the expectation of

$$(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})' = (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{u}\boldsymbol{u}'\boldsymbol{X}(\boldsymbol{X}'\boldsymbol{X})^{-1} \tag{A4}$$

conditional on $\boldsymbol{X}$. Since $\boldsymbol{u}$ is the only stochastic quantity on the right-hand side of the previous expression and we know the form of $E(\boldsymbol{u}\boldsymbol{u}')$ from (A1), this expectation

is given by,

$$(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'E(\boldsymbol{u}\boldsymbol{u}')\boldsymbol{X}(\boldsymbol{X}'\boldsymbol{X})^{-1} = \sigma^2(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{V}\boldsymbol{X}(\boldsymbol{X}'\boldsymbol{X})^{-1}$$

$$\text{var}(\hat{\boldsymbol{\beta}}) = \sigma^2(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'[(1-\rho)\boldsymbol{I}_N + \rho\boldsymbol{S}\boldsymbol{S}']\boldsymbol{X}(\boldsymbol{X}'\boldsymbol{X})^{-1}. \tag{A5}$$

However, one may simplify this expression further by defining $\boldsymbol{N} = \boldsymbol{X}'\boldsymbol{S}\boldsymbol{S}'\boldsymbol{X}(\boldsymbol{X}'\boldsymbol{X})^{-1}$ as one can now rewrite (A5) as

$$\begin{aligned}
&\sigma^2(\boldsymbol{X}'\boldsymbol{X})^{-1}[\boldsymbol{X}'(1-\rho)\boldsymbol{I}_N\boldsymbol{X} + \rho\boldsymbol{X}'\boldsymbol{S}\boldsymbol{S}'\boldsymbol{X}](\boldsymbol{X}'\boldsymbol{X})^{-1} \\
&= \sigma^2(\boldsymbol{X}'\boldsymbol{X})^{-1}[(1-\rho)\boldsymbol{X}'\boldsymbol{X} + \rho\boldsymbol{X}'\boldsymbol{S}\boldsymbol{S}'\boldsymbol{X}](\boldsymbol{X}'\boldsymbol{X})^{-1} \\
&= \sigma^2(\boldsymbol{X}'\boldsymbol{X})^{-1}[(1-\rho) + \rho\underbrace{\boldsymbol{X}'\boldsymbol{S}\boldsymbol{S}'\boldsymbol{X}(\boldsymbol{X}'\boldsymbol{X})^{-1}}_{\boldsymbol{N}}] \\
&= \sigma^2(\boldsymbol{X}'\boldsymbol{X})^{-1}[1 + \rho(\boldsymbol{N}-1)].
\end{aligned} \tag{A6}$$

Next, if we consider the case where the number of observations per cluster is N_g for each cluster, and all of the regressors are fixed within clusters, then (A6) can be rewritten as

$$\text{var}(\hat{\boldsymbol{\beta}}) = \sigma^2(\boldsymbol{X}'\boldsymbol{X})^{-1}[1 + \rho(N_g - 1)]. \tag{A7}$$

Finally, since the standard OLS covariance matrix is $\text{var}_c(\hat{\boldsymbol{\beta}}) = \sigma^2(\boldsymbol{X}'\boldsymbol{X})^{-1}$, we may rewrite (A7). If we suppose, without a loss of generality, that we are interested in the standard error of the last element of the parameter vector, β_k, then the Moulton factor is given by

$$\sqrt{\frac{\text{var}(\hat{\beta}_k)}{\text{var}_c(\hat{\beta}_k)}} = \sqrt{1 + \rho(N_g - 1)}. \tag{A8}$$

■

A.2 Matrix Inversion by LUP-Decomposition

Here we wish to show that the algorithm, which most common software packages such as MATLAB, R, and Stata make use of to compute the matrix inverse, has computational complexity $\mathcal{O}(n^3)$. This exposition is based on Cormen et. al (2011), and simply shows the triple-nested structure of the algorithm, with n repetitions at each step. This explanation is three-fold, first showing how one may compute an LUP-decomposition, secondly how to compute a matrix inverse from an LUP-decomposition, and lastly showing the algorithm used to perform this calculation. Suppose we have a non-singular matrix $\boldsymbol{A}$ we wish to invert. To do this we need to find a lower triangular matrix $\boldsymbol{L}$, an upper triangular matrix $\boldsymbol{U}$, and a permutation matrix $\boldsymbol{P}$ so that $\boldsymbol{PA} = \boldsymbol{LU}$. A permutation matrix $\boldsymbol{P}$ is simply a matrix which is constructed by rearranging the rows or columns of the identity matrix; its purpose is to rearrange the columns or rows of the matrix you apply it to.

One key feature of an LUP-decomposition, as opposed to an LU-decomposition, is that it involves pivoting. Pivoting is done to improve the numerical stability of the LU-decomposition algorithm where the pivot is chosen to be the element with the largest magnitude among the pivot candidates. The first step is to move a nonzero element a_{k1} from the first column to the (1,1) position of the matrix. For the purpose of numerical stability, a_{k1} is the number with the largest magnitude in the first column.[1] This would be the same thing as multiplying $\boldsymbol{A}$ by the permutation matrix $\boldsymbol{Q}$, where $\boldsymbol{Q}$ is simply the identity matrix with the first and kth row interchanged.

[1] Since $\boldsymbol{A}$ is non-singular we know at least one element in every column must be nonzero. However, by choosing the largest value this helps to avoid using numbers which are close to zero in floating-point format ('numerically' zero).

Taking this into account, one may rewrite the product of $\boldsymbol{Q}$ and $\boldsymbol{A}$ as,

$$\boldsymbol{QA} = \begin{pmatrix} a_{k1} & \boldsymbol{w}' \\ \boldsymbol{v} & \tilde{\boldsymbol{A}} \end{pmatrix} \tag{A9}$$

where the column vector $\boldsymbol{v} = (a_{21}, a_{31}, \cdots, a_{n1})'$ except a_{k1} in $\boldsymbol{v}$ is replaced by a_{11}, the row vector $\boldsymbol{w}' = (a_{k2}, a_{k3}, \cdots, a_{kn})$, and $\tilde{\boldsymbol{A}}$ is the rest of the matrix $\boldsymbol{A}$, which is now $(n-1) \times (n-1)$. Now since it is guaranteed that $a_{k1} \neq 0$ we can avoid dividing by zero and rewrite $\boldsymbol{QA}$ as,

$$\boldsymbol{QA} = \begin{pmatrix} a_{k1} & \boldsymbol{w}' \\ \boldsymbol{v} & \tilde{\boldsymbol{A}} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ \boldsymbol{v}/a_{k1} & \boldsymbol{I}_{n-1} \end{pmatrix} \times \begin{pmatrix} a_{k1} & \boldsymbol{w}' \\ 0 & \tilde{\boldsymbol{A}} - \boldsymbol{v}\boldsymbol{w}'/a_{k1} \end{pmatrix}. \tag{A10}$$

Next, we note that $\tilde{\boldsymbol{A}} - \boldsymbol{v}\boldsymbol{w}'/a_{k1}$ is non-singular which can be proved by contradiction. Suppose to the contrary that the $(n-1)\times(n-1)$ matrix $\tilde{\boldsymbol{A}} - \boldsymbol{v}\boldsymbol{w}'/a_{k1}$ is singular, this implies that $\mathbf{rank}(\tilde{\boldsymbol{A}} - \boldsymbol{v}\boldsymbol{w}'/a_{k1}) < n-1$ which implies that the lower $n-1$ rows of the matrix $\begin{pmatrix} a_{k1} & \boldsymbol{w}' \\ 0 & \tilde{\boldsymbol{A}} - \boldsymbol{v}\boldsymbol{w}'/a_{k1} \end{pmatrix}$ has row rank less than $n-1$. However, what this then implies is that the entire matrix must have rank less than n which contradicts the fact that $\boldsymbol{A}$ is non-singular, hence $\tilde{\boldsymbol{A}} - \boldsymbol{v}\boldsymbol{w}'/a_{k1}$ is also non-singular. However, since it can be shown that any non-singular matrix possesses an LUP-decomposition (see Cormen et al. 2001), then the matrix $\tilde{\boldsymbol{A}} - \boldsymbol{v}\boldsymbol{w}'/a_{k1}$ must have an LUP-decomposition which may be found recursively. That is, there is a permutation matrix $\tilde{\boldsymbol{P}}$, a unit lower-triangular matrix $\tilde{\boldsymbol{L}}$, and an upper-triangular matrix $\tilde{\boldsymbol{U}}$ so that,

$$\tilde{\boldsymbol{P}}(\tilde{\boldsymbol{A}} - \boldsymbol{v}\boldsymbol{w}'/a_{k1}) = \tilde{\boldsymbol{L}}\tilde{\boldsymbol{U}}. \tag{A11}$$

Then, if one defines

$$\boldsymbol{P} = \begin{pmatrix} 1 & \mathbf{0} \\ \mathbf{0} & \tilde{\boldsymbol{P}} \end{pmatrix} \boldsymbol{Q} \tag{A12}$$

this is also a permutation matrix as it is the product of two permutation matrices. To sketch a proof this suppose in general you have two $n \times n$ permutation matrices $\boldsymbol{B}$ and $\boldsymbol{C}$ and you apply the permutation $\boldsymbol{B} \times \boldsymbol{C}$. The resulting matrix will simply be a reordering of the rows and columns $\boldsymbol{C}$ which will then still be row or column reordering of the identity matrix, and hence a permutation matrix. Now we may find the LUP-decomposition as,

$$\begin{aligned}
\boldsymbol{PA} &= \begin{pmatrix} 1 & \mathbf{0} \\ \mathbf{0} & \tilde{\boldsymbol{P}} \end{pmatrix} \boldsymbol{QA} \\
&= \begin{pmatrix} 1 & \mathbf{0} \\ \mathbf{0} & \tilde{\boldsymbol{P}} \end{pmatrix} \times \begin{pmatrix} 1 & \mathbf{0} \\ \boldsymbol{v}/a_{k1} & \boldsymbol{I}_{n-1} \end{pmatrix} \times \begin{pmatrix} a_{k1} & \boldsymbol{w}' \\ 0 & \tilde{\boldsymbol{A}} - \boldsymbol{v}\boldsymbol{w}'/a_{k1} \end{pmatrix} \\
&= \begin{pmatrix} 1 & \mathbf{0} \\ \tilde{\boldsymbol{P}}\boldsymbol{v}/a_{k1} & \tilde{\boldsymbol{P}} \end{pmatrix} \times \begin{pmatrix} a_{k1} & \boldsymbol{w}' \\ 0 & \tilde{\boldsymbol{A}} - \boldsymbol{v}\boldsymbol{w}'/a_{k1} \end{pmatrix} \\
&= \begin{pmatrix} 1 & \mathbf{0} \\ \tilde{\boldsymbol{P}}\boldsymbol{v}/a_{k1} & \boldsymbol{I}_{n-1} \end{pmatrix} \times \begin{pmatrix} a_{k1} & \boldsymbol{w}' \\ 0 & \tilde{\boldsymbol{P}}(\tilde{\boldsymbol{A}} - \boldsymbol{v}\boldsymbol{w}'/a_{k1}) \end{pmatrix} \\
&= \begin{pmatrix} 1 & \mathbf{0} \\ \tilde{\boldsymbol{P}}\boldsymbol{v}/a_{k1} & \boldsymbol{I}_{n-1} \end{pmatrix} \times \begin{pmatrix} a_{k1} & \boldsymbol{w}' \\ 0 & \tilde{\boldsymbol{L}}\tilde{\boldsymbol{U}} \end{pmatrix} \\
&= \begin{pmatrix} 1 & \mathbf{0} \\ \tilde{\boldsymbol{P}}\boldsymbol{v}/a_{k1} & \tilde{\boldsymbol{L}} \end{pmatrix} \times \begin{pmatrix} a_{k1} & \boldsymbol{w}' \\ 0 & \tilde{\boldsymbol{U}} \end{pmatrix} \\
&= \boldsymbol{LU}.
\end{aligned} \tag{A13}$$

Where we see that since $\tilde{\boldsymbol{L}}$ is lower-triangular so is $\boldsymbol{L}$ and $\tilde{\boldsymbol{U}}$ is upper-triangular hence

so is $\boldsymbol{U}$. Next we show the steps of the algorithm used to compute this decomposition as presented in Cormen et al. (2001).

1. $n =$ # of rows of $\boldsymbol{A}$
2. define $\pi[1, \cdots, n]$ as a vector
3. **for** $i = 1$ **to** n
4. $\quad \pi[i] = i$
5. **for** $k = 1$ **to** n
6. $\quad p = 0$
7. $\quad$ **for** $i = k$ **to** n
8. $\quad\quad$ **if** $|a_{ik}| > p$
9. $\quad\quad\quad p = |a_{ik}|$
10. $\quad\quad\quad k' = i$
11. $\quad$ **if** $p == k$
12. $\quad\quad$ **error** "singular matrix"
13. $\quad$ exchange $\pi[k]$ with $\pi[k']$
14. $\quad$ **for** $i = 1$ **to** n
15. $\quad\quad$ exchange a_{ki} with $a_{k'i}$
16. $\quad$ **for** $i = k + 1$ **to** n
17. $\quad\quad a_{ik} = a_{ik}/a_{kk}$
18. $\quad\quad$ **for** $j = k + 1$ **to** n
19. $\quad\quad\quad a_{ij} = a_{ij} - a_{ik}a_{kj}$

Since this algorithm has a triply nested looping structure, it has a run time of $\mathcal{O}(n^3)$.

Next we wish to know how to compute a matrix inverse from an LUP-decomposition. After performing the previously explained procedure one would have a decomposition of the $n \times n$ matrix $\boldsymbol{A}$ so that $\boldsymbol{PA} = \boldsymbol{LU}$. In order to do make use of this we construct the set of linear equations

$$\boldsymbol{AX} = \boldsymbol{I}_n \tag{A14}$$

where the inverse of $\boldsymbol{A}$ given by $\boldsymbol{X}$ is a set of n distinct equations of the form $\boldsymbol{Ax} = \boldsymbol{b}$. One way to write this is to let $\boldsymbol{X}_i$ denote the ith column of $\boldsymbol{X}$ and using the notation from Section 2 let $\boldsymbol{l}_i$ be the unit basis vector with one in the ith position and zeros elsewhere. Then the equation given in (A14) can be solved for $\boldsymbol{X}$ using our LUP-decomposition to solve each of the equations

$$\boldsymbol{AX}_i = \boldsymbol{l}_i \qquad i = 1, \cdots, n. \tag{A15}$$

What this then implies is that the inverse $\boldsymbol{A}^{-1}$ can be computed in time $\mathcal{O}(n^3)$.

A.3 Variation in Experiment Results when $R = 1000$

Here in order to illustrate the variation in results when using a relatively small number of replications, as in Cameron, Gelbach, and Miller (2008), several separate Monte Carlo simulations were run with $R = 1000$. Table A1 shows the results of these experiments, where the DGP is the same as that of the heteroskedastic case from Section 4 and $G = 10$. Here we see that the results vary significantly between each replication, with the difference between the highest and lowest value being greater than 1% for almost every method. One of the largest differences is in WB_3 where the in one experiment it over-rejects at 6.2% and then in the next it under-rejects at 4.6%.

Table A1: Rejection Rates: Repeated Experiment

		Trial				
		1	2	3	4	5
Method	CR_1	0.0850	0.0970	0.0880	0.0860	0.0980
	CR_2	0.0750	0.0840	0.0750	0.0770	0.0890
	CR_3	0.0590	0.0570	0.0580	0.0470	0.0590
	WB_1	0.0520	0.0590	0.0590	0.0540	0.0620
	WB_2	0.0410	0.0480	0.0470	0.0380	0.0520
	WB_3	0.0520	0.0610	0.0620	0.0460	0.0620
	YCR_1	0.0660	0.0730	0.0660	0.0650	0.0770
	YCR_2	0.0620	0.0680	0.0630	0.0570	0.0730
	YCR_3	0.0580	0.0570	0.0630	0.0510	0.0670
	CR_2^{df}	0.0840	0.0940	0.0870	0.0900	0.0960

One way to think about why there is so much variation between experiments is to think of a binomial random variable with probability p and n trials. Since we are interested in how many times a success, being a rejection of the null, occurs over the number of replications, really we want to use this information about the binomial random variable to think about the sample proportion. Suppose we think of the variable as being $X \sim b(n, p)$ where n is the number of replications and p is the likelihood of a rejection. A true null hypotheses tested at the 5% level, should reject exactly 5% of the time if the test has proper size and hence $p = 0.05$. Then the sample proportion is just the number of successes (rejections of the null) divided by n. Then, the variance of X/n is just the variance of X divided by n^2, as we know that given a constant a, $\text{var}(aX) = a^2\text{var}(X)$. Since a binomial random variable has variance $p(1-p)$ this implies that the variance of the sample proportion is $p(1-p)/n$. Hence we see that the variation of the rejection rate is decreasing in n. That is, more replications will decrease the variance.