rcme: Recounting Crime with Measurement Error - workshop

British Society of Criminology: 2022-06-29

Introduction

In this workshop you will get practical experience of using our new package - **rcme**: Recounting Crime with Measurment Error to assess the sensitivity of regression results using recorded crime data to the presence of measurement error. No prior knowledge of R (and RStudio) is required as the workshop proceeds in a step-by-step fashion. However, we have assumed a working knowledge of regression methods.

This package is still a work-in-progress so we advise you to proceed with some caution. You can follow updates on the likely timeline for full publication and read a pre-print paper about the package at https://recountingcrime.com .

Before we begin: Introducing R and RStudio

R is an open source software package that can be used for a very wide range of statistical analyses. You can obtain and install it for free, with versions available for PCs, Macs and Linux. To find out what is available, go to the Comprehensive R Archive Network (CRAN) at http://cran.r-project.org/. Being free is not necessarily a good reason to use R. However, R is also a well developed, documented and supported (by an extensive user community) data analysis software. It is widely used in research, both academic and commercial.

RStudio is a free user-interface for R that makes basic data analysis much more straightforward and user-friendly. In particular, it allows you to see your data, outputs and user commands simultaneously. It can be downloaded from https://rstudio.com/products/rstudio/.

RStudio is not required for this workshop, however it can make data analysis easier. You can find out more about how to use it at https://education.rstudio.com/learn/beginner/.

R is command-line driven. That is, the user types a command that the software interprets and responds to. This may mean that R initially feels a bit daunting, however once you know the commands it is usually much faster to type them than to work through a series of menu options. A log or script of the commands can also be saved for use on another occasion or for sharing with others. All of the R code in the worksheet will be identified in separate boxes that look like this:

```
Example of R code
```

When you see code in one of these boxes you should type the code into the R Console (or a script file). You can also copy the text directly from your browser to save typing!

All the R output that you will get in the R console will be identified in boxes that look like this:

```
## [1] "Example of R output"
```

Downloading and installing rcme

R **packages** are user written programs that vastly increase the capabilities of R, enabling you to conduct almost any form of statistical analysis, as well as create interactive webpages, draw maps, and scrape websites. We have designed **rcme**: Recounting Crime with Measurement Error as an R Package that can be downloaded directly from github. R packages need to first be downloaded onto your computer and saved with R. R refers to this process as installing the package. Because our package is a work in progress, we also need to install `devtools`.

```
install.packages("devtools")
devtools::install_github("alex-cernat/rcme")
```

Installing a package makes it available for use by R. However, in order to actually use the package it must also be loaded into the current workspace. This is done with the `library()` command.

```
library(rcme)
```

As a reminder, the first time you use a package it must first be downloaded and installed into R using the `install.packages()` command. The downloaded package must then be loaded into the current workspace using the command `library()`. If you come back to RStudio at a later date you do not need to re-install the packages, but you ALWAYS need to load them in using `library()`.

IMPORTANT: It is easy to forget to correctly load packages, particularly if R restarts for some reason. Usually, the main cause of errors when trying to complete these workshops is that one or more packages have not correctly loaded, so it is often a good idea to check this first when things go wrong. You can quickly see which packages you have loaded into your current R session by consulting the Packages window in the bottom right quadrant of RStudio. Loaded packages will have a tick next to them.

Example 1: Violent crime and disorder across Local Authorities

In our first example we will examine the effect of violent crime on disorder across Local Authorities in England and Wales. The data is from a sample of 250 Local Authorities with the included variables simulated to broadly match the data reported in Pina-Sanchez et al. (2022b). The data is included with the **rcme** package and we can view the top few rows by typing:

```
head(crime_disorder)
```

```
##   violent_crime white_british unemployment median_age    disorder log_violent_crime
## 1     17.954613     -1.875242  -0.74880766 -0.4089145  0.64666711          2.887847
## 2     12.177885      2.201486   0.03518811  2.1158555 -0.07387513          2.499622
## 3      8.141439      1.610787  -0.14918213  1.8627676  1.21494115          2.096967
## 4     25.822089     -1.074690  -0.32459869 -1.3888129 -0.21920971          3.251230
## 5     22.655406     -2.114620   0.48229187 -2.2554911  0.92906244          3.120399
## 6     16.560196     -0.521898  -0.41359409 -1.2104559 -0.69439539          2.807002
```

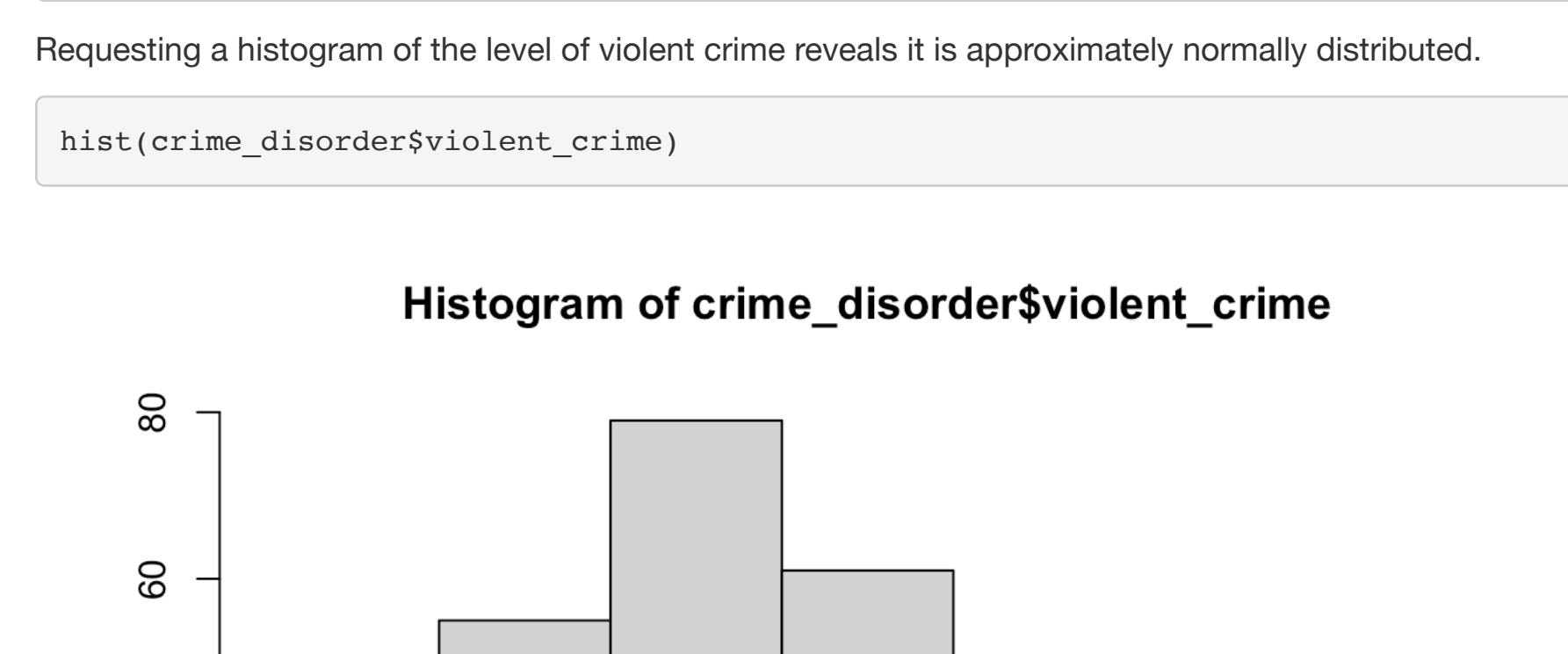
Let's start by taking a summary look at the raw data. Here we can see a mean violent crime rate of 14.5 violent crimes per 1,000 in each Local Authority with a minimum of 4 and maximum of 33.3. The remaining variables have all been standardised with a mean of 0 and standard deviation of 1. These detail the size of the white british population in each LA, the level of unemployment, median age and extent of disorder. Disorder is measured as the area weighted average score from local resident's assessments of the extent of disorderly behaviour with higher scores corresponding to areas with higher levels of disorder.

```
summary(crime_disorder)
```

```
##  violent_crime    white_british       unemployment        median_age          disorder        log_violent_crime
##  Min.   : 4.216   Min.   :-2.85200   Min.   :-2.21630   Min.   :-2.48795   Min.   :-3.1834   Min.   :1.439
##  1st Qu.: 9.814   1st Qu.:-0.68914   1st Qu.:-0.68375   1st Qu.:-0.71623   1st Qu.:-0.6609   1st Qu.:2.284
##  Median :13.765   Median : 0.07608   Median :-0.05838   Median : 0.02947   Median : 0.0817   Median :2.622
##  Mean   :14.529   Mean   : 0.00000   Mean   : 0.00000   Mean   : 0.00000   Mean   : 0.0000   Mean   :2.581
##  3rd Qu.:18.706   3rd Qu.: 0.69612   3rd Qu.: 0.67639   3rd Qu.: 0.69951   3rd Qu.: 0.6679   3rd Qu.:2.929
##  Max.   :33.320   Max.   : 2.65133   Max.   : 2.62725   Max.   : 3.09363   Max.   : 3.0735   Max.   :3.506
```

Requesting a histogram of the level of violent crime reveals it is approximately normally distributed.

```
hist(crime_disorder$violent_crime)
```



We start our analysis by estimating a linear regression model exploring the effect of violent crime on levels of neighbourhood disorder, whilst also controlling for levels of unemployment, the percentage of residents that are White British, and the median age. We will save the model results in the object `naive.1` and request the full model output using `summary()`. Pina-Sanchez et al (2022a) recommend logging the crime variable in most situations to mitigate some of the more adverse impacts of the multiplicative measurement error form that affects crime rates, which we achieve including the wrapper `log()` around the crime variable.

```
naive.1 <- lm(disorder ~ log(violent_crime) + white_british + unemployment + median_age,  data = crime_disorder)
summary(naive.1)
```

```
## 
## Call:
## lm(formula = disorder ~ log(violent_crime) + white_british + 
##     unemployment + median_age, data = crime_disorder)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -2.66107 -0.58347 -0.00198  0.52401  2.19310 
## 
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)   
## (Intercept)        -1.02838    0.40467  -2.541  0.01166 * 
## log(violent_crime)  0.39849    0.15539   2.564  0.01093 * 
## white_british      -0.08915    0.08241  -1.082  0.28044   
## unemployment        0.21015    0.06697   3.138  0.00191 **
## median_age         -0.17004    0.09018  -1.886  0.06054 . 
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.8584 on 245 degrees of freedom
## Multiple R-squared:  0.275,  Adjusted R-squared:  0.2632 
## F-statistic: 23.23 on 4 and 245 DF,  p-value: 2.705e-16
```

Here we observe the expected positive effect of violent crime on disorder. The crime rate is logged so a 10% increase in the crime rate is associated with a statistically significant yet modest (.39 standard deviations) increase in disorder. Areas of higher unemployment are also identified as having higher disorder, while the older and whiter the area, the lower the level of disorder (although these latter effects do not reach standard levels of statistical significance).

To examine whether the observed effect of violent crime on disorder is robust to the measurement error mechanisms affecting police recorded rates of violent crime (e.g. when violent crime is an independent variable), we use the function `rcme_ind()`. Users must specify the model formula followed by details of the dataset being used. *Note that we must exclude the wrapper log() from the formula, we will tell rcme to log the crime variable in a separate statement*. Next we identify the crime variable as our key predictor of interest (`key_predictor`). Users then have the option to include values for the levels of systematic (`S`) and random error (`R_sd`), as well as (optionally) the correlation between the measurement error and key predictor (`D`). For now we will ignore `D`, but we will return to this in our second example. Finally, we must tell **rcme** that the crime variable should be logged by adding `log_var = T`.

```
me.1ex <- rcme_ind(  #Change when function changed - e.g. logging etc.
  formula = "disorder ~ violent_crime + white_british + unemployment + median_age",
  data = crime_disorder,
  key_predictor = "violent_crime",
  S = c(0.31, 0.67, 1.0),
  R_sd = c(0.08, 0.10, 0.12),
  log_var = T)
```

In this example we have selected systematic error rates (`S`) that broadly match official estimates for the proportion of reported incidents of violence recorded by the police (0.67), and the proportion of all crimes experienced that are recorded (an estimated 46% of violent crimes are reported meaning 31% of all violent incidents are recorded - since 0.67*0.46 = .31). The final value (1.0) assumes no systematic under counting, allowing us to also examine the unique effect of random errors (Her Majesty Inspectorate of Constabulary, 2014:64). We use the same source to generate plausible values for the random error, `R_sd`. Importantly, here we must provide an estimated SD across all areas, with HMIC data indicating an SD for undercounting of 0.10 across Police Forces. We include two further values (0.08 and 0.12 to explore the sensitivity of our assumption).

rcme prints out the measurement error adjusted estimate of the key predictor on the dependent variable for all possible combinations of the systematic, random and differential errors specified by the user. In this example we only included systematic and random errors so these can be straightforwardly viewed in tabular form.

```
me.1ex
```

```
## $sim_result
##      S R_sd D log_var key_predictor    SE
## 1 0.31 0.08 0    TRUE         0.245 0.120
## 2 0.67 0.08 0    TRUE         0.359 0.147
## 3 1.00 0.08 0    TRUE         0.390 0.154
## 4 0.31 0.10 0    TRUE         0.167 0.101
## 5 0.67 0.10 0    TRUE         0.333 0.143
## 6 1.00 0.10 0    TRUE         0.387 0.153
## 7 0.31 0.12 0    TRUE         0.099 0.078
## 8 0.67 0.12 0    TRUE         0.309 0.138
## 9 1.00 0.12 0    TRUE         0.379 0.152
## 
## $naive
## 
## Call:
## lm(formula = paste0(outcome, " ~ ", paste0(c(paste0("log(", key_predictor, 
##     ")", collapse = ""), predictors[!predictors %in% key_predictor]), 
##     collapse = " + ")), data = data)
## 
## Coefficients:
##        (Intercept)  log(violent_crime)       white_british        unemployment  
##           -1.02838             0.39849            -0.08915             0.21015  
##         median_age  
##           -0.17004  
## 
## 
## $key_predictor
## [1] "violent_crime"
```

The results included in the columns headed `key_predictor` and `SE` are the error adjusted estimates of the effect of violent crime on disorder for each of the different combinations of systematic (`S`) and random (`R_sd`) error. These results imply that when considering systematic and random measurement error mechanisms simultaneously, in the presence of substantial undercounting (where S is 0.31, combining under-reporting and under-recording), the magnitude of the effect of crime on disorder is severely attenuated and clearly non-significant. This is true for all three levels of random error, with the attenuation effect proportional to the magnitude of the random error. When the focus is solely on under-recording (where S is 0.67), the attenuating effect is still present (albeit smaller in magnitude), but the results remain statistically significant unless in the presence of substantial random error. Therefore, we cannot conclusively say that violent crime leads to higher disorder, at least such effect should not be considered as strong as it would be otherwise reported.

Example 2: Criminal Damage across London

In our second example, we examine the effect of collective efficacy on levels of crime across London. The data is from a sample of 250 Middle Layer Super Output Areas (MSOA) in London with the included variables simulated to broadly match the data reported in Pina-Sanchez et al. (2022b)

```
head(crime_damage)
```

```
##   collective_efficacy unemployment median_age white_british damage_crime log_damage_crime
## 1         -1.28442575   0.05379451 -0.4016502    -0.2027723    3.9094352        1.3633929
## 2         -1.03925828   1.18204531 -1.1662788    -1.2573142    3.0694682        1.1215043
## 3          0.51861659  -0.81635704  2.0705378     0.8522283    0.4778192       -0.7385229
## 4          0.67044665  -0.98987457  1.5649438     0.3077032    2.2011857        0.7889962
## 5          0.58813673   0.18282236 -0.1707982    -0.2032584    2.8591986        1.0505414
## 6         -0.03851433  -0.47728529  1.1711697    -0.5604057    1.2604280        0.2314514
```

Examining the raw data we can see a mean criminal damage rate of 2.9 violent crimes per 1,000 in each MSOA. Collective efficacy (measured using Metropolitan Police Public Attitudes Survey data) is a combination of social cohesion and neighbourhood informal social control derived by aggregating residents; assessments to the area level, with higher scores representing areas where residents would be more likely to intervene in the presence of disorder and crime. The remaining variables are the same as in the first example (albeit measured in each MSOA rather than Local Authority) and have again been standardised with a mean of 0.

```
summary(crime_damage)
```

```
##  collective_efficacy  unemployment        median_age        white_british       damage_crime    log_damage_crim
e
##  Min.   :-3.10657    Min.   :-2.26315   Min.   :-2.47902   Min.   :-2.59402   Min.   :0.4778   Min.   :-0.7385  
##  1st Qu.:-0.61599    1st Qu.:-0.71732   1st Qu.:-0.69115   1st Qu.:-0.73960   1st Qu.:2.0265   1st Qu.: 0.7063  
##  Median : 0.01054    Median :-0.07106   Median :-0.02687   Median : 0.08378   Median :2.8164   Median : 1.0355  
##  Mean   : 0.00000    Mean   : 0.00000   Mean   : 0.00000   Mean   : 0.00000   Mean   :2.8682   Mean   : 0.9546  
##  3rd Qu.: 0.69989    3rd Qu.: 0.69868   3rd Qu.: 0.62289   3rd Qu.: 0.72843   3rd Qu.:3.6483   3rd Qu.: 1.2943  
##  Max.   : 2.40057    Max.   : 2.55476   Max.   : 2.49421   Max.   : 2.51350   Max.   :6.4906   Max.   : 1.8704
```

As before, we start by estimating our model of interest. And once again, we log the crime variable as this is expected to mitigate some of the more adverse impacts of the multiplicative measurement error form present in crime rates.

```
naive.2 <- lm(log(damage_crime) ~ collective_efficacy + unemployment + white_british + median_age, data = crime_d
amage)
summary(naive.2)
```

```
## 
## Call:
## lm(formula = log(damage_crime) ~ collective_efficacy + unemployment + 
##     white_british + median_age, data = crime_damage)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -1.47667 -0.19247  0.05156  0.29677  0.69409 
## 
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)    
## (Intercept)          0.95460    0.02606  36.637  < 2e-16 ***
## collective_efficacy -0.11004    0.03493  -3.150 0.001835 ** 
## unemployment         0.08025    0.04052   1.980 0.048786 *  
## white_british        0.13076    0.03666   3.567 0.000434 ***
## median_age          -0.17582    0.03746  -4.693 4.47e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.412 on 245 degrees of freedom
## Multiple R-squared:  0.2694, Adjusted R-squared:  0.2575 
## F-statistic: 22.59 on 4 and 245 DF,  p-value: 6.758e-16
```

Our focus is on the effect of collective efficacy on police recorded criminal damage, where we observe the expected negative association. Areas that are higher in collective efficacy generally experience lower levels of criminal damage. The effect of collective efficacy is statistically significant, although its size is relatively modest.

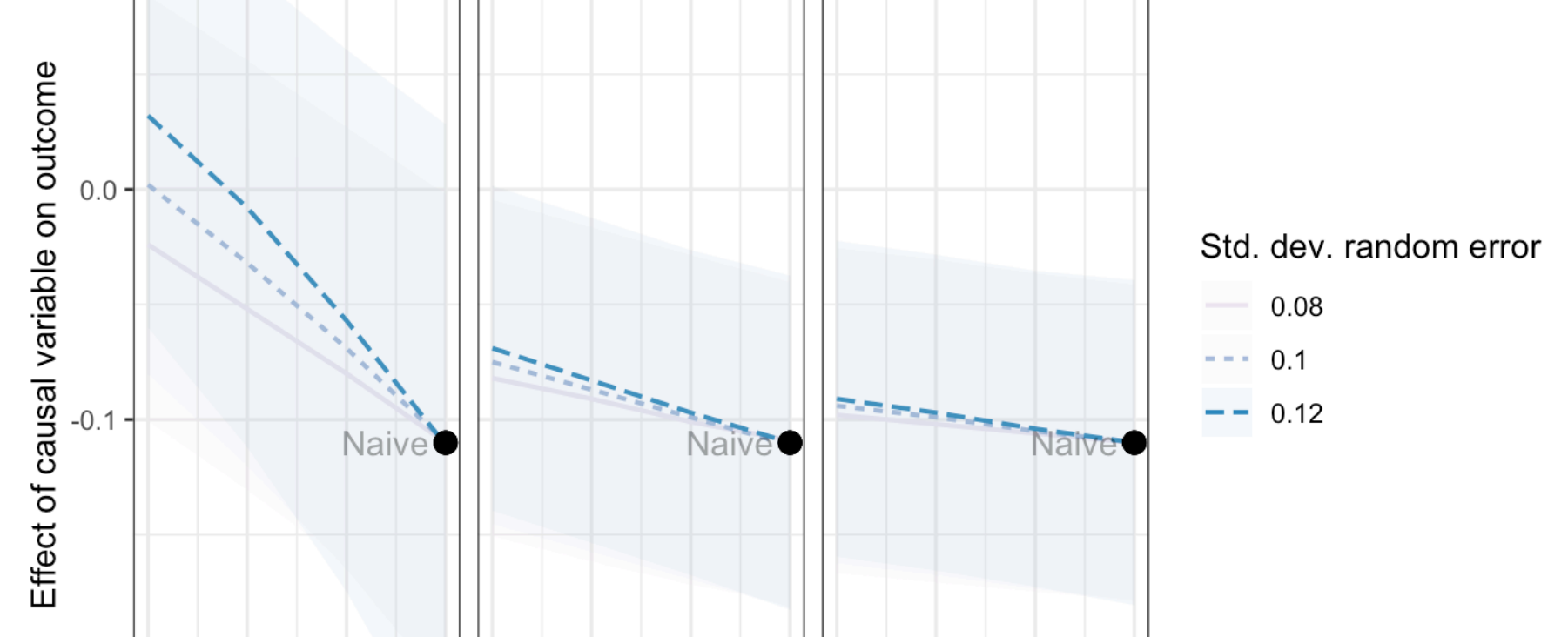
When crime is the response variable we use the functon `rcme_out()`. This time we specify `collective_efficacy` as the `key_predictor` since we are interested in understanding the extent that the estimated relationship with recorded crime may be biased as a result of measurement error in crime. For the case of criminal damage, reporting (34%) and recording (86%) rates differ markedly from violent crime (ONS, 2020; Her Majesty Inspectorate of Constabulary, 2014: 65). However, the overall counting rate, after considering the share of crimes reported that are recorded, is remarkably similar at 28.9% (0.86*0.34). We therefore retain a similar minimum counting rate, `S`, of 0.29 , selecting additional values of 0.86 (ignoring those crimes not reported to the police) and 1.0 (representing the absence of systematic errors). There is no good reason to expect that there will be more (or less) random error when considering the case of criminal damage, therefore we retain values of 0.08, 0.10 and 0.12 for the random error, `R_sd`.

```
me.2ex <- rcme_out( #Update when package finished - logging etc.
  formula = "damage_crime ~ collective_efficacy + unemployment + white_british + median_age",
  data = crime_damage,
  key_predictor = "collective_efficacy",
  S = c(0.29, 0.85, 1.0),
  R_sd = c(0.08, 0.10, 0.12),
  D = c(-0.3, -0.2, -0.1, 0),
  log_var = T)
```

To incorporate differential errors, we also need to provide a plausible range of estimates for the correlation, `D`, between the systematic error in police recorded violent crime rates and our key predictor (collective efficacy). Pina-Sanchez et al. (2022b) sets out one plausible strategy for estimating this association using a combination of survey data and census statistics, with an overall association of -0.3. We include three additional values (-0.2, -0.1, 0) to examine the sensitivity of our results to this differential error form.

We could explore the raw data in tabular format like the first example. However, with the added complexity of including differential errors it is generally easier to examine the measurement error effect visually. This can be done with the command `rc`me_sim_plot()`.

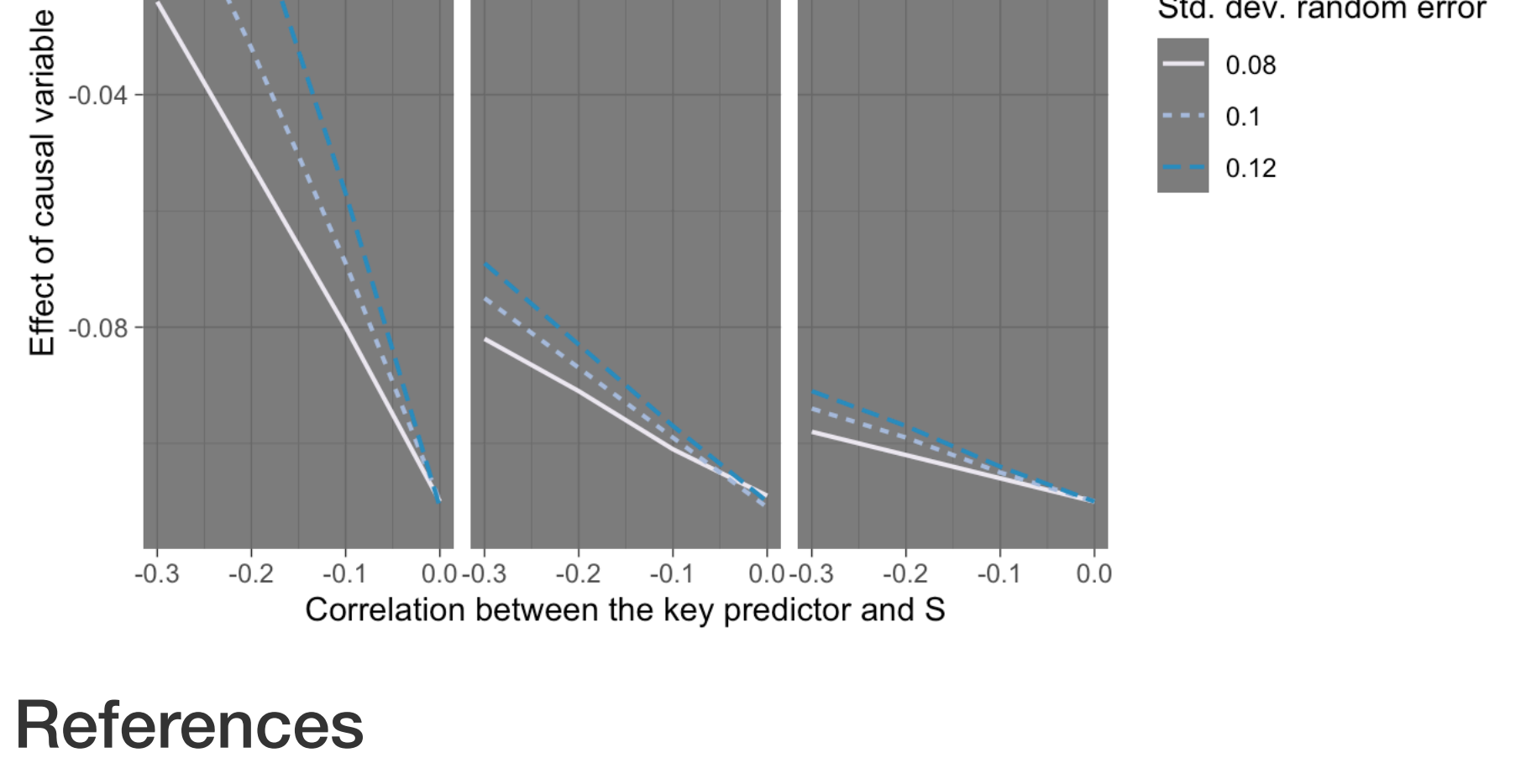
```
rcme_sim_plot(me.2ex)
```



The three panels show the range of possible values for the association between collective efficacy and criminal damage (represented on the vertical axis), as the negative correlation between collective efficacy and measurement error gets stronger (horizontal axis), for the three counting rates considered (0.29, 0.86 and 1.00). The plots also include different values for the random error component (represented by the colour-graded lines). We find that the effect of collective efficacy in reducing criminal damage has likely been overestimated. However, the extent of the bias appears to be relatively negligible unless in the presence of large levels of under-counting (S = 0.29), that is both variable across areas and moderately associated with collective efficacy. In particular, when there is substantial under-counting but it is not correlated with collective efficacy (the rightmost part of each plot), the estimated effect of collective efficacy is largely unchanged.

It is also straightforward to edit the plot like a standard ggplot object. For example:

```
rcme_sim_plot(me.2ex, ci = F, naive = F) +
  ggplot2::theme_dark()
```



References

Pina-Sánchez, J., Buil-Gil, D., Brunton-Smith, I., and Cernat, A. (2022a) 'The Impact of Measurement Error in Models Using Police Recorded Crime Rates'. Journal of Quantitative Criminology. Pre-print available at: https://osf.io/preprints/socarxiv/ydf4b/

Pina-Sánchez, J., Brunton-Smith, I., Buil-Gil, D., and Cernat, A. (2022b) 'RecountingCrime: A Sensitivity Analysis Tool to Explore the Impact of Measurement Error in Police Recorded Crime Rates'. Working paper available at: https://osf.io/preprints/socarxiv/sbc8w/