Supplement to: When (Not) to Rely on the Reliable Change Index: A Critical Appraisal and Alternatives to Consider in Clinical Psychology

ANONYMOUS FOR REVIEW

2023-11-30

Contents

This document provides additional details not included in the main body of the article "When (Not) to Rely on the Reliable Change Index: A Critical Appraisal and Alternatives to Consider in Clinical Psychology."

The supplement should be available as both an .html file and a .pdf file. In the .html file, analysis code can be hidden or revealed using the buttons labeled "Code" and "Hide". The .pdf is slightly harder to read.

First, we load some packages and set a theme.

```
# libraries:
require(here)
require(janitor)
require(rmarkdown)
require(kableExtra)
require(tidyverse)
require(patchwork)

# setting a theme for plots:
theme_set(theme_bw() +
```

```
            theme(panel.grid.major = element_blank(),
                  panel.grid.minor = element_blank(),
                  panel.background = element_blank()))

# creating data is used and described later on.
# It is created here first to allow the sections to be moved
# independently and computations to be centralized.
set.seed(20231129)
symData <- data.frame(true_x = seq(from = -4,
             to = 4, by = .01)) %>%
mutate(correct = case_when(true_x < 0 ~ pnorm(-1.96,
                                              mean = true_x),
                           true_x == 0 ~ .95,
                           true_x > 0 ~ pnorm(1.96, mean =
                                                true_x,
                                              lower.tail = FALSE)))
symData_obs <- symData %>%
  mutate(correct_obs = case_when(true_x <= 0 ~ pnorm(0, mean = true_x),
                                 true_x > 0 ~ pnorm(0, mean = true_x,
                                                    lower.tail = FALSE)))
nppl <- 50000L
simdata <- data.frame(true_score = rnorm(nppl,
                                         mean = 50,
                                         sd = 10)) %>%
  mutate(error1 = rnorm(nppl,
                        sd = 5),
         error2 = rnorm(nppl,
                        sd = 5),
         t1_obs = true_score + error1,
         t2_obs = true_score + error2)
Sdiff.sim <- sqrt(2 * (sd(simdata$t1_obs) *
                         sqrt(1 - cor(simdata$t1_obs,
                                      simdata$t2_obs)))^2)
simdata2 <- simdata %>%
  mutate(t2_true = true_score + rnorm(nrow(simdata), -5, sd = 10),
         t2_obs = t2_true + rnorm(nrow(simdata), sd = 5),
         obs_diff = t2_obs - t1_obs,
         true_diff = t2_true - true_score,
         error_diff = obs_diff - true_diff,
         RCI = 1.96 * Sdiff.sim,
         Category = case_when(obs_diff > RCI ~ "Reliable Det",
                              obs_diff < -RCI ~ "Reliable Imp",
                              TRUE ~ "Not Reliable"))
simdataTrueChange <- data.frame(true_score = rnorm(20000,
                                                   mean = 50,
                                                   sd = 10)) %>%
  mutate(t1_obs = true_score + rnorm(20000, sd = 5),
         t2_obs = true_score + rnorm(20000, sd = 5)) %>%
  mutate(error.rep = t2_obs - t1_obs)
simdataTrueChange2 <- mutate(simdataTrueChange,
      true_score2 = true_score +  rnorm(20000, sd = 5),  # the true change component
      t2_obs2 =     true_score2 + rnorm(20000, sd = 5))  # the error component
Sdiff.sim.2 <- sqrt(2 * (sd(simdataTrueChange2$t1_obs) *
```

```
                            sqrt(1 - cor(simdataTrueChange2$t1_obs,
                                         simdataTrueChange2$t2_obs2)))^2)
```

1 Mac's height

In the main text we introduced Mac, who was observed to be 98cm at Time 1 and 99cm at Time 2. Here are the observations we found:

```
mac.data <- tibble("Time" = c(1, 2),
                   "Height" = c(98, 99))
knitr::kable(mac.data) %>%
  kableExtra::kable_styling(full_width = F,
                            latex_options = "HOLD_position")
```

Time	Height
1	98
2	99

This is all of our data. Obviously, Mac may have grown, or not.

The benefit of measuring Mac more than two times would be that changes can more easily be detected. In the text, the example is given that we could have observed Mac several time on each day, instead of only once, we might be more confident in change. Here is that data:

```
mac.multiple.data <- tibble("Time" = c(1, 1, 1, 1, 1, 2, 2, 2, 2, 2),
                            "Height" = c(97, 97, 98, 98, 97, 99, 100, 100, 99, 100))
knitr::kable(mac.multiple.data) %>%
  kableExtra::kable_styling(full_width = F,
                            latex_options = "HOLD_position")
```

Time	Height
1	97
1	97
1	98
1	98
1	97
2	99
2	100
2	100
2	99
2	100

How confident could we be about Mac's growth in this case?

```
mac.multiple.data %>%
  group_by(Time) %>%
  summarise("Mean Height" = mean(Height),
            "SD" = sd(Height)) %>%
  knitr::kable(digits = 2) %>%
  kableExtra::kable_styling(full_width = F,
                            latex_options = "HOLD_position")
```

Time	Mean Height	SD
1	97.4	0.55
2	99.6	0.55

There are several ways to evaluate the differences between times in this data. I will start with an approach that is analytically very similar to the RCI's approach, and then show the results of a two-sample t-test as well.

The SEM is

$SEM = \frac{SD}{\sqrt{N}}$

Or in this case:

$SEM = \frac{0.55}{\sqrt{5}} = 0.25.$

The S_{diff} is equal to $SEM * \sqrt{2}$, so in this case S_{diff} is equal to 0.3535534.

The RCI is `1.96 * Sdiff`, or 0.6929646.

Any observed difference in these averages that is greater than 0.69cm would exceed the $p < .05$ threshold.

Our observed difference is 2.2cm in this case, which is 6.29 times greater than S_{diff}. If these values were distributed Normally, the chance of observing a difference score so large, given the null hypothesis of no growth, would be $1.5873299 \times 10^{-10}$.

Even the chance of observing a 1cm difference in average heights with five observations at each timepoint would be 0.0021374.

However, these values, since they are based on relatively few observations, are better approximated by a student's t distribution. Therefore, we might want to use a student's t-test.

A two-sample t-test would be appropriate if we are interested in the mean difference between these two timepoints. We do not really need to use a paired-samples t-test since the samples themselves are not paired.

We can compute the t-test value as follows:

```
t.test(mac.multiple.data %>% filter(Time == 1) %>% pull(Height),
       mac.multiple.data %>% filter(Time == 2) %>% pull(Height),
       var.equal = TRUE)
```

```
##
##  Two Sample t-test
##
## data:  mac.multiple.data %>% filter(Time == 1) %>% pull(Height) and mac.multiple.data %>% filter(Tim
## t = -6.3509, df = 8, p-value = 0.0002204
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -2.998823 -1.401177
## sample estimates:
## mean of x mean of y
##      97.4      99.6
```

The t-test is significant, $t(8) = -6.35$, $p = .0002$, 95% CI = [-3.00, -1.40]. This suggests there is an extremely high likelihood that some change has occurred, even though the chance is not so astronomically high. There is still a .0002% chance of seeing this difference in null data with small sample sizes.

If we had a smaller mean change with the same variability in measurements, the t-test would be this:

```r
t.test(mac.multiple.data %>% filter(Time == 1) %>% pull(Height) + 1,
       mac.multiple.data %>% filter(Time == 2) %>% pull(Height),
       var.equal = TRUE)
```

```
##
##  Two Sample t-test
##
## data:  mac.multiple.data %>% filter(Time == 1) %>% pull(Height) + 1 and mac.multiple.data %>% filter
## t = -3.4641, df = 8, p-value = 0.008516
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -1.9988233 -0.4011767
## sample estimates:
## mean of x mean of y
##      98.4      99.6
```

Again, this suggests there would be a very low chance (0.8%) measurement error was to blame for the different observations.

Instead of collecting more data from Mac, however, we decided to use the RCI by collecting data from Mac's classmates.

A representation of Mac's classmates might look like this for a specific class of 8 other children, each measured twice on the same day:

```r
class.data <- tibble("id" = 1:8,
                     "Time1" = c(90, 104, 104, 110, 99, 93, 101, 102),
                     "Time2" = c(90, 105, 104, 109, 99, 94, 101, 101))
knitr::kable(class.data) %>%
  kableExtra::kable_styling(full_width = F,
                            latex_options = "HOLD_position")
```

id	Time1	Time2
1	90	90
2	104	105
3	104	104
4	110	109
5	99	99
6	93	94
7	101	101
8	102	101

From this data we can compute the RCI.

The SEM in psychometric tests would usually be computed with the standard deviation and reliability estimate. Here, we can estimate our reliability as the correlation between the two timepoints, which is a good test-retest reliability estimate. We will adopt the SD of the first time point as the SD to use here. Note that because the sample size is so small, the resulting value may not be a particularly good estimate.

```r
SEM1 <- sd(class.data$Time1) * sqrt(1 - cor(class.data$Time1, class.data$Time2))
```

This value, 0.5032313, is an imperfect estimate of the true value, given the small sample. It is, however, very close to the true value, as can be seen by computing the value of SEM analytically. This is a somewhat different formula for SEM than readers may be familiar with, but is analytically correct. It highlights how the SEM is conceptually an average distance away from the true value we expect to be, for any given measurement. It is a "typical" error.

```r
sem.func <- function(data,
                     .cols = c(2:3)){
  mean(apply(data[, .cols], 1, var))^(1/2)
}

SEm1.rel <- sem.func(class.data)
SEm1.rel
```

```
## [1] 0.5
```

S_{diff}, then is calculated as follows:

```r
Sdiff <- sqrt(2 * SEm1.rel^2)  # equivalent to sqrt(2)*SEm
```

S_{diff} is 0.71, or more fully, 0.7071068. This is also coincidentally the square root of SE_m (0.5) in this case, but that is not generally true.

The RCI for our instrument is:

```r
RCI <- 1.96 * Sdiff
RCI
```

```
## [1] 1.385929
```

We would need to observe a difference score greater than 1.39cm in order to conclude that any difference score represented "reliable" change. Mac's observed difference score, 1cm, is not greater than the instrument's RCI, therefore we cannot reject the null hypothesis that he did not change.

Referring back to Equation 1 in the main text, the *RC index* for Mac's observed difference score is:

```r
RCindex <- (99 - 98) / Sdiff
```

The RC index, 1.4142136, is not greater than 1.96, and therefore we cannot reject the null hypothesis of no change for Mac. The RC Index is interpretable as a z-score, but contains exactly the same information as the RCI comparison above, using the same assumptions and data.

2 How precise does a measure need to be for a change score to be sufficient on its own?

As stated in the text, it would be nice if, like in some other areas of science, psychometric measurement tools could be so precise that their error could be ignored in an application. Why is this not possible?

The level of precision required for this is simply implausible for self-report scales. No matter how precise our measurement tool is, we will always have errors that impact observed scores, and thus, any observed score should be treated as uncertain.

Measurement error in the instrument can be ignored when all expectable errors are smaller than the observed increments of the scale. That is, the observed scores are so precise and the errors are so small that any given error will not change the observed value. When this is true, we could use a simple difference score on its own, rather than considering uncertainty. But how precise, exactly, does a score need to be? To begin, we should define "expectable" errors.

For simplicity, let's call a 5% error rate acceptable. This is ludicrously large in physical measurement (atomic clocks can have error rates of less than 0.0000000000001%) but conventional in psychology. Even 5% will illustrate how far psychometric measures are from being interpretable without considering uncertainty.

Using the previously created data set, here is the minimum real change, in S_{diff} units, that must be detected with a simple difference score:

```r
symData_obs %>%
  filter(correct_obs >= .95 & true_x > 0) %>%
  pull(true_x) %>%
  min()
```

```
## [1] 1.65
```

To achieve a 5% error rate with an observed difference score, the real difference must be 1.65 times larger than the S_{diff}. In order to be able to ignore measurement error completely, our scale must detect every real change that large or larger. No real changes smaller than this value can exist.

As a consequence, the increments of the scale must be at at least this large to ensure that these changes will be detected. We will assume we are using unit weights and sum scoring, so that an increase or decrease in a single response option on one item is equal to one observed point. In this case, the psychometric scale must have the property that one observed point is less than or equal to 1.65 * S_{diff}. Equivalently, the S_{diff} would need to be less than two-thirds of a scale point (`1 / 1.65 = 0.61`).

On a unit-weighted sum-scored self-report instrument (e.g., PHQ-9, BDI-II), this means that the RCI would need to be `0.61 * 1.96`, or 1.1956 scale points or smaller to be precise enough to ignore the measurement error.

Already, this value is very small for an RCI value on most psychometric scales. We can further translate these values into reliability coefficients. This will give us a sense of how plausible it is to create a scale with this property.

We achieve this by substituting S_{diff} is equal to `sqrt(2) * SEM`, and our S_{diff} is 0.61,

`sqrt(2) x SEM` = `0.61`

SEM = `0.61 / sqrt(2)`

This means that the SEM is - at most - 0.43.

Using the standard formula for SEM, `SEM = SD*sqrt(1 - rxx)`, We can make a formula to show the maximum allowable SD for a given reliability. The SD and reliability coefficients are related, because we assume that difference scores should have very little variability if the measurement error is small, but as error gets bigger, the SD will grow due to the error variance.

The formula is `SD = SEM / sqrt(1 - rxx)`, here substituting `SEM = 0.43`.

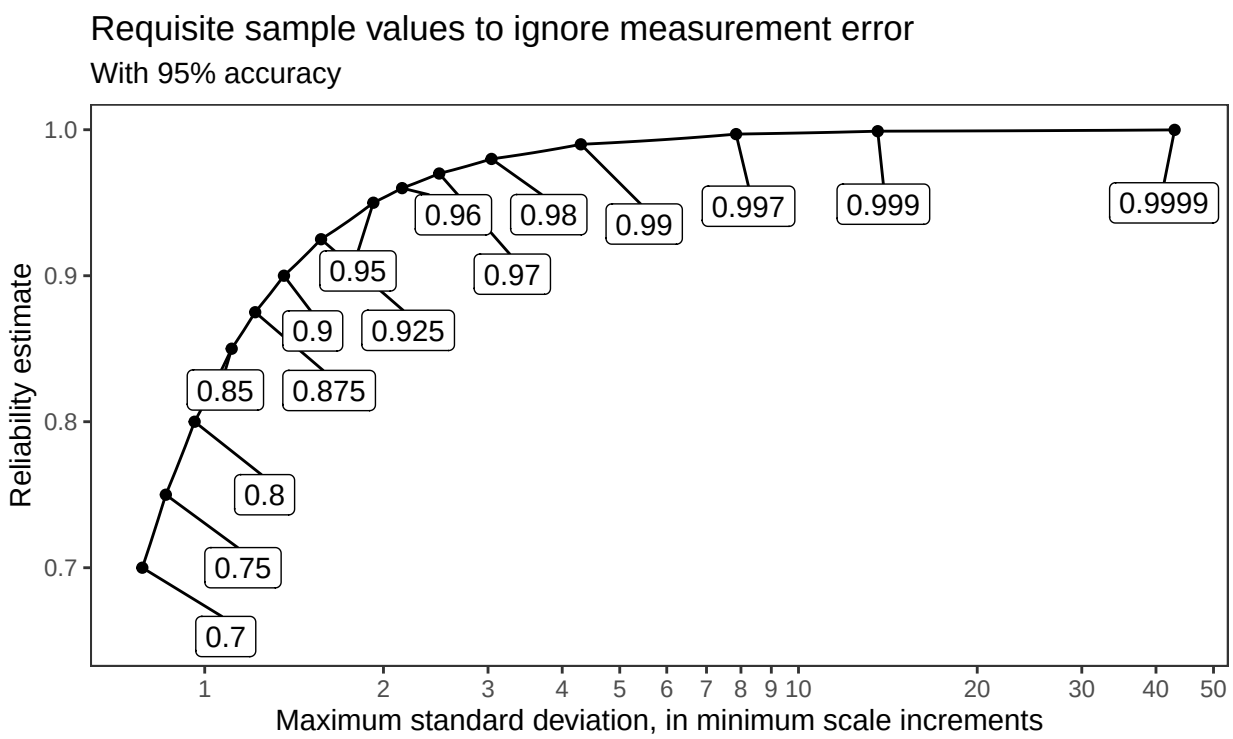
```r
max_SD_05 <- function(rel_val){
  0.43 / sqrt(1 - rel_val)
}
```

We can then plot the values that satisfy this equality:

```r
rho <- c(.7, .75, .8, .85, .875, .9, .925, .95, .96, .97, .98, .99, .997, .999, .9999)
max_SD_allowed <- max_SD_05(rho)

ggplot(data = data.frame(rho, max_SD_allowed),
  aes(y= rho,
      x = max_SD_allowed)) +
  geom_line() +
  geom_point() +
  coord_trans(x = 'log10') +
  ggrepel::geom_label_repel(aes(label = rho),
              nudge_x = .3,
              nudge_y = -.05) +
  scale_x_continuous(breaks = c(1:10, 20, 30, 40, 50)) +
  labs(title = "Requisite sample values to ignore measurement error",
```

```
        subtitle = "With 95% accuracy",
        y = "Reliability estimate",
        x = "Maximum standard deviation, in minimum scale increments",
        caption = "For a given SD value, the reliability must be no lower than shown.
        For a given Reliability value, the SD must be no greater than shown.
        The SD is expressed in the smallest possible increment of the scale,
        one point for sum-scored instruments.")
```



This shows that in the range of typical psychometric reliabilities (roughly .7 to .9), the maximum allowable SD of a sample of measurements would be less than 1.5 scale points to achieve a reasonably small SEM value.

If a sample SD is over about 2 scale points, the required reliability is so high (.95 or greater) that it would be extremely unusual. Very few scales that are of interest have sample SD values close to this. In mental health samples, the PHQ-9 has SDs between roughly 8-15 points. That would require reliability of about .997 or higher.

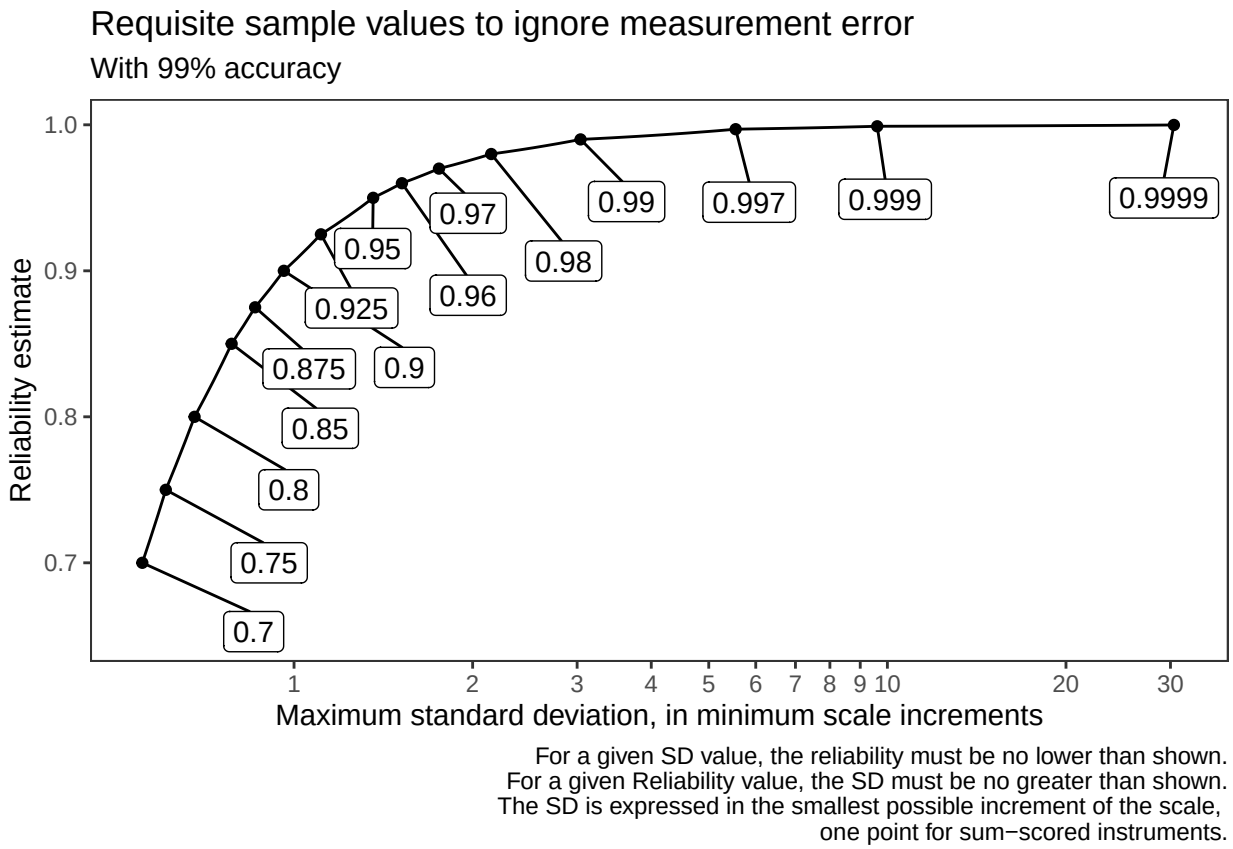
Again, this assumes that a 5% error rate is acceptable in using a measurement. This is not necessarily OK, since it maintains the same error rate many people would expect using the RCI. If we wanted to be 99% confident that any difference score observation would not be affected by measurement error, we would see the following:

```
symData_obs %>%
  filter(correct_obs >= .99 & true_x > 0) %>%
  pull(true_x) %>%
  min()
```

```
## [1] 2.33
```

```r
max_SD_01 <- function(rel_val){
  0.304 / sqrt(1 - rel_val)
}
rho <- c(.7, .75, .8, .85, .875, .9, .925, .95, .96, .97, .98, .99, .997, .999, .9999)
max_SD_allowed <- max_SD_01(rho)

ggplot(data = data.frame(rho, max_SD_allowed),
  aes(y= rho,
      x = max_SD_allowed)) +
  geom_line() +
  geom_point() +
  coord_trans(x = 'log10') +
  ggrepel::geom_label_repel(aes(label = rho),
             nudge_x = .3,
             nudge_y = -.05) +  scale_x_continuous(breaks = c(1:10, 20, 30, 40, 50)) +
  labs(title = "Requisite sample values to ignore measurement error",
       subtitle = "With 99% accuracy",
       y = "Reliability estimate",
       x = "Maximum standard deviation, in minimum scale increments",
       caption = "For a given SD value, the reliability must be no lower than shown.
       For a given Reliability value, the SD must be no greater than shown.
       The SD is expressed in the smallest possible increment of the scale,
       one point for sum-scored instruments.")
```



In this case, any SD over 1 scale point requires a reliability estimate of roughly 0.9 or higher. If the SD is 2 points, we need a reliability of .97 or higher. Again, the sample SD values for scales of interest are regularly much larger than this in samples that have relevant issues, so achieving this level of precision is simply not

going to happen.

Measurement precision is highly helpful and should be pursued. However, no psychometric measure is precise enough to completely ignore measurement error when considering a change score. None are even close.

This is yet another reason that change scores themselves are insufficient: They must be supplemented with additional data, either by collecting more observations of the changing process or by estimating the magnitude of measurement error, like the RCI or other methods discussed in the manuscript. But even using these methods there is so much uncertainty in the estimates that difference scores should never be considered definitely changed or definitely unchanged: there will always be a (high) chance that measurement error affects the observation.

3 Type S and M errors with the RCI

Gelman & Carlin (2014) provide the `retrodesign()` function:

```
retrodesign <- function(A, s, alpha=.05, df=Inf, n.sims=10000){
  z <- qt(1-alpha/2, df)
  p.hi <- 1 - pt(z-A/s, df)
  p.lo <- pt(-z-A/s, df)
  power <- p.hi + p.lo
  typeS <- p.lo/power
  estimate <- A + s*rt(n.sims,df)
  significant <- abs(estimate) > s*z
  exaggeration <- mean(abs(estimate)[significant])/A
  return(list(power=power, typeS=typeS, exaggeration=exaggeration))
}
```

To examine the design implications, let us suppose we have a true difference score equal to `Sdiff.sim`, 7.0832677. That is, a person actually changed about as much as a "typical" error in a difference score. The appropriate spread variable is also `Sdiff` in the case of the RCI.

Here is what `retrodesign()` returns:

```
retrodesign(A = Sdiff.sim, s = Sdiff.sim)
```

```
## $power
## [1] 0.170075
##
## $typeS
## [1] 0.009045272
##
## $exaggeration
## [1] 2.490672
```

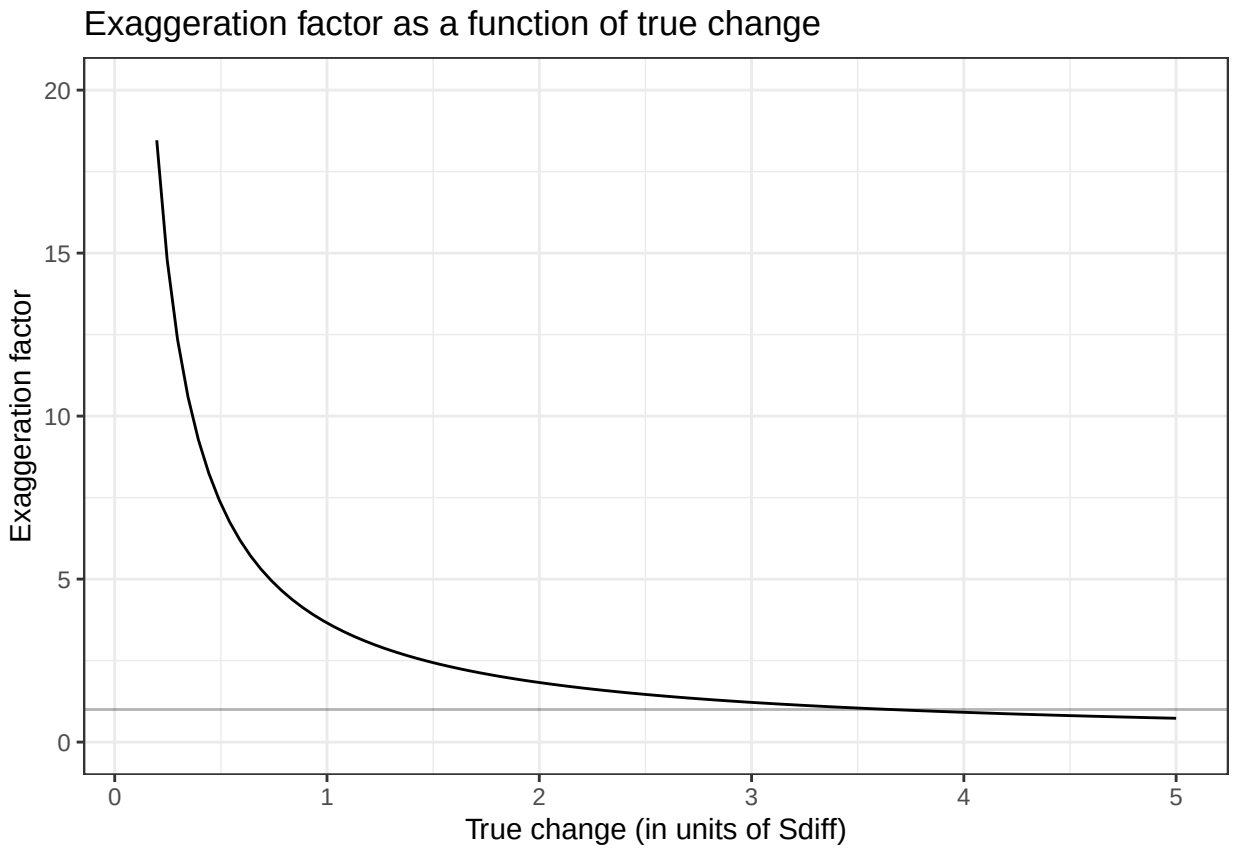
From that, we can see that we have power of 0.17 to detect the real change. The Type S error, the chance that a "significant" result would reflect the wrong sign, is given by `typeS`: `0.009`, or roughly 1%. This is a small value, so may not be concerning, but may still be much higher than expected. The `exaggeration` factor can be interpreted as how much larger, on average, a "significant" result would be than the true value that generated it. With this example, the exaggeration factor is `2.5`, showing that if we were "lucky" enough to correctly detect this true change, our actual measurement would be roughly 2.5 times greater than the true difference score, on average. So our measurements would be highly misleading. In other words, the opposite of reliable.

It is worth visualizing the exaggeration factor as a function of true change.

```
# version of retrodesign() for plotting:
# I am setting s=1 by default for ease, this is the spread variable.
```

```r
exagg <- function(A, s=1, alpha=.05, df=Inf, n.sims=10000){
  z <- qt(1-alpha/2, df)
  p.hi <- 1 - pt(z-A/s, df)
  p.lo <- pt(-z-A/s, df)
  power <- p.hi + p.lo
  typeS <- p.lo/power
  estimate <- A + s*rt(n.sims,df)
  significant <- abs(estimate) > s*z
  exaggeration <- mean(abs(estimate)[significant])/A
  return(exaggeration)  # this line is simpler
}


exaggPlot <- ggplot() +
  ylim(0, 20) +
  xlim(0.1,5)
exaggPlot +
  geom_hline(yintercept = 1,
             alpha = .3) +
  geom_function(fun = exagg) +
  labs(title = "Exaggeration factor as a function of true change",
       x = "True change (in units of Sdiff)",
       y = "Exaggeration factor") +
  theme_bw()
```



In general, when true changes are small (relative to the noise defined by S_{diff}), the exaggeration effect will be extremely large. Only when the *true* effect is about three times as large as the S_{diff} does the exaggeration

factor come close to 1 (the grey line). Observing such large actual changes should be uncommon if there is no true effect, and they are so much larger than the expectable error that we hardly need a statistical test to tell us they exist.

Why does this matter? If a patient has a large, reliable change and we infer they have improved meaningfully in treatment, many of these observations will be due to chance and exaggerated effect. The next observation of the patient should, on average show significant backsliding: the patient will appear to have lost improvement. This could be demoralizing and confusing to both patients and providers, who may seek to identify problems in care, when instead they should simply understand that the observations are noisy and the large "reliable" change was faulty.

This should discourage us from using dichotomies in general when we expect to combine noisy measures with possibly small effects. Even the correct decisions will be wrong and misleading.

3.1 Exaggeration of Mac's height

For the example of Mac's height, if he truly grew 1cm and our S_{diff} estimate is .71, only observations greater than 1.39 (which, again, is 1.96 * S_{diff}) are considered reliably changed. We might run `retrodesign()` on Mac's data:

```
Sdiff <- 0.71
retrodesign(A = 1, s = Sdiff)
```

```
## $power
## [1] 0.2910189
##
## $typeS
## [1] 0.001298915
##
## $exaggeration
## [1] 1.833214
```

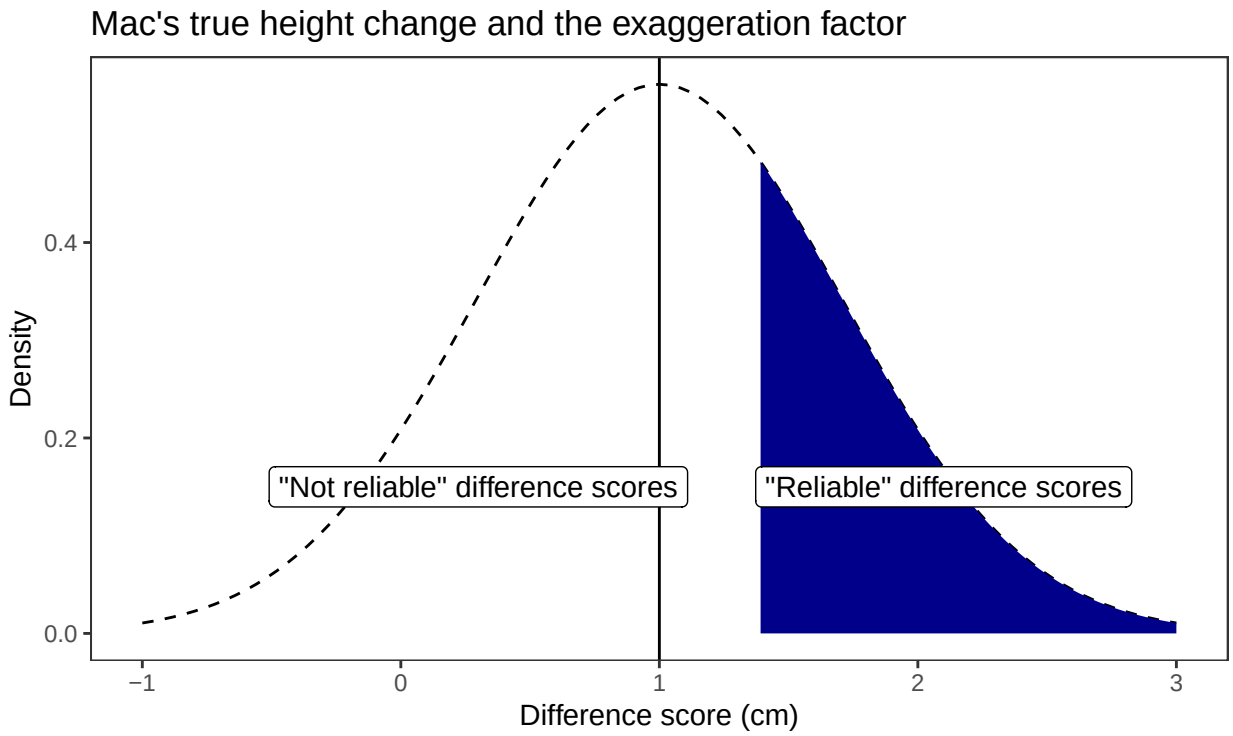
If we knew that he really grew 1cm in a month, we would only have a 28% chance of observing a change greater than the RCI with our measurement instrument. This shows how inadequate our data is, considering the research question we are asking. To properly measure this phenomenon, we need more precise data in the form of more observations or much more precise tools, not statistics.

We can also see that observations of Mac that are "reliable" will be overestimated on average by a factor of 1.8, and a very small percent of observations (0.1%) will conclude Mac reliably got shorter.

To visualize this part of Mac's example: If he truly grew 1cm, but we have a noisy measure, sometimes we would think he grew more than he really did, and other times less. The percent of the plausible measurements above (1.96 * S_{diff}) is the portion of his measurements that would be considered "reliable," but these are a biased sample of his measures.

```
cmPlot <- ggplot() + xlim(-1, 3)
cmPlot +
  geom_function(fun = dnorm,
                args = list(mean = 1,
                            sd = Sdiff),
                linetype = "dashed") +
  labs(title = "Mac's true height change and the exaggeration factor",
       x = "Difference score (cm)",
       y = "Density",
       caption = "If Mac's true growth is 1cm, the plausible observed differences could be anywhere nea
  geom_vline(xintercept = 1) +
  stat_function(fun = dnorm,
                args = list(mean = 1,
```

```
                          sd = Sdiff),
              geom = "area",
              fill = "darkblue",
              xlim = c(1.96*Sdiff, 3))  +
  geom_label(aes(label = c("\"Reliable\" difference scores"),
                 x = 2.1,
                 y = .15)) +
  geom_label(aes(label = c("\"Not reliable\" difference scores"),
                 x = 0.3,
                 y = .15))
```



If Mac's true growth is 1cm, the plausible observed differences could be anywhere near 1cm. The shaded area represents the portion of observations of Mac that would be considered 'reliable'. Note that they are all mismeasurements of true growth, and there is a large chance of an observed difference that would not be considered reliable.

As we can see, even if Mac truly grew exactly the 1cm we observed in this example, and we are "lucky" enough to find a reliable change in height, our measurement of his height will be very inaccurate and biased away from his actual change in height. This will lead to perceived instability in future observations, because it is driven by measurement error, even though it will be considered "reliable" by the RCI. That will likely result in a much smaller ("unreliable") increase or even an observed decrease in height the next time Mac is measured.

The small number of Type S errors are not shown in that plot but they exist to the far left.

Again, this is a reason to use more data points than 2: the question being asked of the data is more complex than the quality of the data, and the RCI does not solve this problem.

4 RCI total accuracy

We can visualize the likelihood of a "correct" classification using the RCI as a function of the true score change. The true score change in units of S_{diff} is equivalent to the signal-to-noise ratio (SNR). The term

"correct" here is really only theoretical for individuals, since we will never know the true score change for any real individual. At the population level, however, these are the chances of making a determination in the correct direction.

Given the uncertainty inherent in measurements, being "correct" is probably not the most productive framing: all observations are uncertain and may misrepresent the theoretical true state even when they are correct most of the time. It would be preferable to acknowledge that all observations are uncertain, though some are more likely to be correct than others.

Type I error is only applicable if the true score change is exactly 0. In this instance, the RCI will be correct 95% of the time.

Type II errors apply any other time, and the power function above provides the likelihood of correct classification under those circumstances. In this context, the RCI failing to detect a reliable change is considered an inaccurate decision.

The accuracy of the RCI when the true change is not 0 is equal to the power: the expected percent of observations that would reject the null hypothesis of 0 true effect.

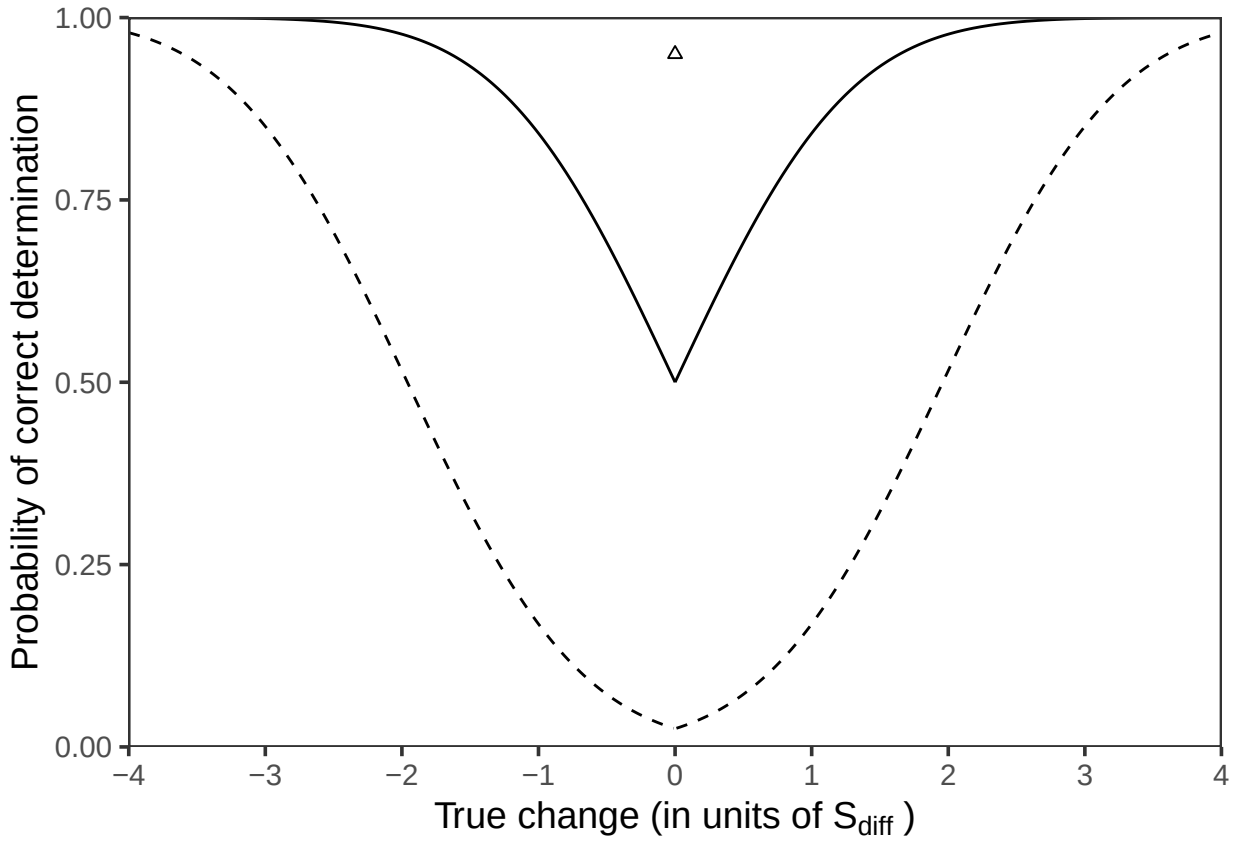
In the Figure below, the dashed line and the triangle represent the RCI, while the solid line is the naïve cut point of 0 (that is, if the observed change is positive, we would infer a positive true change).

```
symData <- data.frame(true_x = seq(from = -4,
             to = 4, by = .01)) %>%
mutate(correct = case_when(true_x < 0 ~ pnorm(-1.96,
                                              mean = true_x),
                           true_x == 0 ~ .95,
                           true_x > 0 ~ pnorm(1.96, mean =
                                                true_x,
                                              lower.tail = FALSE)))
symData_obs <- symData %>%
  mutate(correct_obs = case_when(true_x <= 0 ~ pnorm(0, mean = true_x),
                                 true_x > 0 ~ pnorm(0, mean = true_x,
                                                    lower.tail = FALSE)))
```

```
accPlot_combined <- ggplot(symData_obs,
                   aes(x = true_x,
                       y = correct_obs))  +
  # difference score lines
  geom_line(data = symData_obs %>%
              filter(true_x <= 0)) +
  geom_line(data = symData_obs %>%
              filter(true_x >= 0)) +
  # RCI lines
  geom_line(data = symData %>%
              filter(true_x < 0),
            aes(y = correct),
            linetype = "dashed") +
  geom_line(data = symData %>%
              filter(true_x > 0),
            aes(y = correct),
            linetype = "dashed") +
  geom_point(data = symData %>%
              filter(true_x == 0),
            aes(y = correct),
            shape = 2) +
  scale_x_continuous(breaks = -4:4,
```

```
                     expand = c(0, 0)) +
  scale_y_continuous(limits = c(0, 1),
                     expand = c(0, 0))

# for in-paper version take out the title and caption (they will appear in the text)
accPlot_combined +
  theme_bw(base_size = 14) +
  theme(panel.grid.major = element_blank(),
  panel.grid.minor = element_blank()) +
  labs(x = expression("True change (in units of "*S[diff]*" )"),
       y = "Probability of correct determination")
```



This plot is interpreted in the main article text, but shows that in every situation except a true null effect, the naïve method is more accurate. Additionally, for real changes to be detected at a high likelihood, they must be nearly 4 times as large as the measurement error.

5 PHQ-9 and the RCI, given a true change

What if we knew exactly how many scale points on the PHQ-9 had changed for a patient? How would the RCI perform?

Gyani et al. (2013) report RCI = 5.20, so Sdiff = `5.20 / 1.96`, or 2.65 scale points.

Therefore each scale point is `1 / 2.65`, 0.38 Sdiff units

```
# define a function to use in the lapply call, changing default SD to 1
retrodesign2 <- function(A){retrodesign(A = A, s = 1)}
```

To get the power, Type S, and Type M estimates:

```
# define a new accuracy function
accuracy_func <- function(true_x, cutpoint){
  pnorm(cutpoint, mean = true_x, lower.tail = FALSE)
}
# function to find chance of an observed 0 when real change is present
null_func <- function(true_x, cutpoint){
  1 -
    pnorm(-cutpoint, mean = true_x, lower.tail = TRUE) -
    pnorm(cutpoint, mean = true_x, lower.tail = FALSE)
}
# rci_acc_func <- function(true_x){
#   pnorm(1.96, mean = true_x, lower.tail = FALSE)
# }
lapply(list(0.38 * 1:10), retrodesign2) %>%
  data.frame() %>%
  mutate(diffscore = 1:10,
         .before = 1) %>%
  mutate(accuracy = accuracy_func(diffscore * 0.38,
                                  cutpoint = 0.38/2),
         diff_t2 = null_func(true_x = diffscore * .38,
                             cutpoint = .38/2),
         diff_sign = pnorm(0,
                           mean = diffscore * 0.38)) %>%
  knitr::kable(col.names = c("True difference score",
                             "RCI Power",
                             "RCI Type S Error",
                             "RCI Type M Error",
                             "Difference score accuracy",
                             "Difference score Type II error",
                             "Difference score Type S Error"),
               digits = 3,
               align = "c") %>%
  kableExtra::column_spec(1, width = "5em") %>%
  kableExtra::column_spec(3:7, width = "6em") %>%
  kableExtra::kable_styling(full_width = FALSE,
                            latex_options = "HOLD_position")
```

True difference score	RCI Power	RCI Type S Error	RCI Type M Error	Difference score accuracy	Difference score Type II error	Difference score Type S Error
1	0.067	0.145	8.499	0.575	0.140	0.352
2	0.118	0.028	4.250	0.716	0.113	0.224
3	0.207	0.005	2.833	0.829	0.079	0.127
4	0.330	0.001	2.125	0.908	0.048	0.064
5	0.476	0.000	1.700	0.956	0.025	0.029
6	0.626	0.000	1.417	0.982	0.012	0.011
7	0.758	0.000	1.214	0.993	0.005	0.004
8	0.860	0.000	1.062	0.998	0.002	0.001
9	0.928	0.000	0.944	0.999	0.000	0.000
10	0.967	0.000	0.850	1.000	0.000	0.000

6 PHQ-9 and the RCI, given an observed change

When we have observed a change score, we can compare it to the RCI and other metrics, like:

- MID estimates
- RC Index
- The probability this is the correct direction (pD)

RCI is simply greater than 5.2 is reliable MID is generally greater than 2 is important RC Index is simply the difference score divided by Sdiff, which is 2.65. pD is a call to `pnorm()`

```
observed_table <- data.frame(
  "diffscore" = 0:10) %>%
  mutate("RCI" = ifelse(diffscore > 5.2,
                        "Reliable",
                        "Not reliable"),
         "MID" = ifelse(diffscore > 2,
                        "Likely Important",
                        "Not likely important"),
         "Likely68" = ifelse(diffscore > .997 * 2.65,
                             "Likely changed",
                             "Not likely changed"),
         "RC_Index" = diffscore / 2.65,
         "pD" = pnorm(0,
                      mean = diffscore,
                      sd = 2.65,
                      lower.tail = FALSE))

knitr::kable(observed_table,
             col.names = c("Observed difference score",
                           "RCI category",
                           "MID category",
                           "Likely change",
                           "RC Index",
                           "Probability of direction"),
             caption = "Observed PHQ-9 difference scores and possible
             interpretations. The RCI is taken to be 5.2, from Gyani et al., 2013.
             The MID is taken to be 2, a simplification of recommendations by Kounali et al., 2022.
             Likely change indicates 68% confidence per Peipert et al., 2022.
             Probability of direction indicates the likelihood of the true
             score being greater than 0, using Sdiff as the error term.",
             digits = 3,
             align = "c") %>%
  kableExtra::column_spec(1:6, width = "7em") %>%
  # kableExtra::column_spec(5, width = "8em") %>%
  kableExtra::kable_styling(full_width = FALSE,
                            latex_options = "HOLD_position")
```

Table 1: Observed PHQ-9 difference scores and possible interpretations. The RCI is taken to be 5.2, from Gyani et al., 2013. The MID is taken to be 2, a simplification of recommendations by Kounali et al., 2022. Likely change indicates 68Probability of direction indicates the likelihood of the true score being greater than 0, using Sdiff as the error term.

Observed difference score	RCI category	MID category	Likely change	RC Index	Probability of direction
0	Not reliable	Not likely important	Not likely changed	0.000	0.500
1	Not reliable	Not likely important	Not likely changed	0.377	0.647
2	Not reliable	Not likely important	Not likely changed	0.755	0.775
3	Not reliable	Likely Important	Likely changed	1.132	0.871
4	Not reliable	Likely Important	Likely changed	1.509	0.934
5	Not reliable	Likely Important	Likely changed	1.887	0.970
6	Reliable	Likely Important	Likely changed	2.264	0.988
7	Reliable	Likely Important	Likely changed	2.642	0.996
8	Reliable	Likely Important	Likely changed	3.019	0.999
9	Reliable	Likely Important	Likely changed	3.396	1.000
10	Reliable	Likely Important	Likely changed	3.774	1.000

The pD statistic is particularly revealing. With only an observed difference score of 2, there is a 77% chance that the real change is in the direction observed (assuming the Sdiff is appropriate). The RC index is just a linear transformation of the difference score. The MID is more sensitive to small changes than the RCI, but, like the RCI, still needs to be interpreted probabalistically.

7 RCI and the statistical significance filter

Using any cut point on a noisy continuous variable will induce biased errors within the extreme categories. Much of this work comes from metascience studies, in which the sampling error inherent in a single study is the concerning error, but the algebra works exactly the same for a single estimate's measurement error.

The easiest way to demonstrate this fact is with simulated data.

First, we simulate data from a theoretical instrument with true mean of 50 and SD of 10. For now we will use an error component that allows for a true test-retest coefficient of about .80, which would be considered adequate in many contexts.

```
nppl <- 50000L
simdata <- data.frame(true_score = rnorm(nppl,
                                         mean = 50,
                                         sd = 10)) %>%
  mutate(error1 = rnorm(nppl,
                        sd = 5),
```

```
         error2 = rnorm(nppl,
                        sd = 5),
         t1_obs = true_score + error1,
         t2_obs = true_score + error2)
```

This code generates 50000 rows of data with these variables:

- `true_score`: The actual values we are interested in testing, but are unfortunately hidden from direct observation. In the code above, 10,000 random numbers are drawn from a normal distribution with Mean of 50 and SD of 10.

- `error1`: The random noise (measurement error) at the first observation. The SD of the random error component is half of the SD of the true variance. The mean of the measurement error is 0, which is the default of `rnorm()`.

- `error2`: The random noise (measurement error) at the second observation. The SD of the random error component is half of the SD of the true variance. The mean of the measurement error is 0, which is the default of `rnorm()`.
- `t1_obs`: The first observed score, which is close to `true_score` but has a random measurement error component added as well.

- `t2_obs`: The second observed score, which is close to `true_score` but has a separate random measurement error component, completely unrelated to the measurement error from `t1_obs` except that, on average across all people, they are the same magnitude.

The correlation between the observed scores at the two times is:

```
cor(simdata$t1_obs, simdata$t2_obs)
```

```
## [1] 0.8010762
```

Our r_{xx} value is 0.8.

Using the formula for the RCI, our S_{diff} is calculated here:

```
Sdiff.sim <- sqrt(2 * (sd(simdata$t1_obs) * sqrt(1 - cor(simdata$t1_obs, simdata$t2_obs)))^2)
```

The estimated S_{diff} for this data is **7.03**. This will be off from the true value by a very small amount due to sampling error (which is quite small with this N).

7.1 Adding real changes

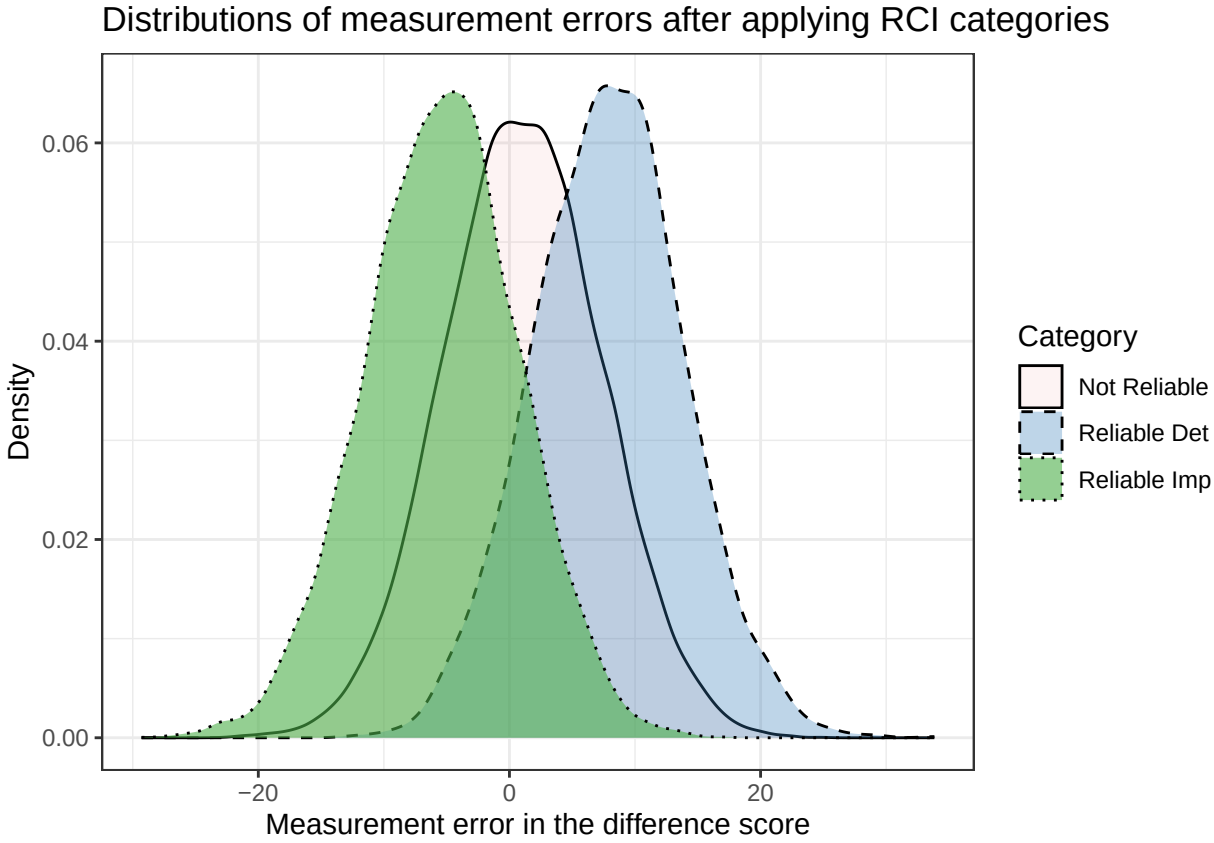
Now we will add a small amount of real change to the second administration. The error component will stay the same size as it was, but here we compute `t2_true` as a function of the first true score and some randomly-varying real change. On average the changes will be more likely to be negative than positive, though there is a wide range allowed: the SD of real changes is twice the mean value. That means some individuals will have positive true change scores as well. We also compute the RCI at this stage.

```
simdata2 <- simdata %>%
  mutate(t2_true = true_score + rnorm(nrow(simdata), -5, sd = 10),
         t2_obs = t2_true + rnorm(nrow(simdata), sd = 5),
         obs_diff = t2_obs - t1_obs,
         true_diff = t2_true - true_score,
         error_diff = obs_diff - true_diff,
         RCI = 1.96 * Sdiff.sim,
         Category = case_when(obs_diff > RCI ~ "Reliable Det",
                              obs_diff < -RCI ~ "Reliable Imp",
                              TRUE ~ "Not Reliable"))
```

The within-person effect size for this variable is -0.44, a small-to-medium effect on average, and small in the context of within-person changes on psychological interventions.

Increasing the average effect size will decrease bias that emerges with using a cutpoint. It is important to note that the following features depend on signal-to-noise ratio very directly.

```
errorDistRCI <- simdata2 %>%
  ggplot(aes(x = error_diff,
             fill = Category,
             linetype = Category)) +
  geom_density(aes(alpha = Category,
                   linetype = Category)) +
  scale_linetype_manual(values = c("solid", "dashed", "dotted")) +
  scale_alpha_discrete(range = c(0.05, 0.6))+
  scale_fill_brewer(palette = "Set1")+
  # geom_density(aes(fill = Category),
  #              alpha = .4) +
  # scale_fill_brewer(palette = "Set1") +
  theme_bw() +
  labs(y = "Density",
       x = "Measurement error in the difference score")
errorDistRCI +
  labs(title = "Distributions of measurement errors after applying RCI categories")
```



The average absolute error in difference scores and the bias (the average error) show these differences. For reference, in the full simulated data set, the absolute error is about 5 and there is no appreciable bias:

```
simdata2 %>%
  summarise(AvgAbsError = mean(abs(error_diff)),
```

```
            AbsError025 = quantile(abs(error_diff), .025),
            AbsError975 = quantile(abs(error_diff), .975),
            Bias = mean(obs_diff - true_diff),
            Bias025 = quantile(error_diff, .025),
            Bias975 = quantile(error_diff, .975)) %>%
  knitr::kable(digits = 2,
               col.names = c("Abs. Err.",
                             "Abs. Err. 95% LB",
                             "Abs. Err. 95% UB",
                             "Bias",
                             "Bias 95% LB",
                             "Bias 95% UB")) %>%
  kableExtra::kable_styling(full_width = FALSE,
                            latex_options = "HOLD_position")
```

Abs. Err.	Abs. Err. 95% LB	Abs. Err. 95% UB	Bias	Bias 95% LB	Bias 95% UB
5.64	0.23	15.79	0.02	-13.88	13.78

From this, we see that the typical magnitude of absolute error in a difference score is about 5.6, and the mean error (bias) is close to 0, which it should be. The 95% CIs for both parameters are large, showing that there are some quite large errors. All of this is what should happen given the simulation parameters.

```
simdata2 %>%
  group_by(Category) %>%
  summarise(AvgAbsError = mean(abs(error_diff)),
            AbsError025 = quantile(abs(error_diff), .025),
            AbsError975 = quantile(abs(error_diff), .975),
            Bias = mean(obs_diff - true_diff),
            Bias025 = quantile(error_diff, .025),
            Bias975 = quantile(error_diff, .975)) %>%
  knitr::kable(digits = 2,
               col.names = c("Category",
                             "Abs. Err.",
                             "Abs. Err. 95% LB",
                             "Abs. Err. 95% UB",
                             "Bias",
                             "Bias 95% LB",
                             "Bias 95% UB")) %>%
  kableExtra::kable_styling(full_width = FALSE,
                            latex_options = "HOLD_position")
```

Category	Abs. Err.	Abs. Err. 95% LB	Abs. Err. 95% UB	Bias	Bias 95% LB	Bias 95% UB
Not Reliable	5.06	0.20	14.20	1.08	-11.08	13.36
Reliable Det	8.40	0.52	19.86	7.89	-3.79	19.86
Reliable Imp	6.61	0.31	17.46	-5.38	-17.46	6.33

Dividing the same data into groups shows that the different "Reliable" groups have very different error profiles. Reliable deteriorators had very large errors which were almost entirely positively biased, reliable improvers had large errors which were almost all negatively biased, and those who were not categorized as reliable actually had the smallest absolute and least biased errors.

That is, *the people who would be considered not reliably changed in this data had the most accurately-measured difference scores*, while those that were reliably changed were associated with substantially larger and more

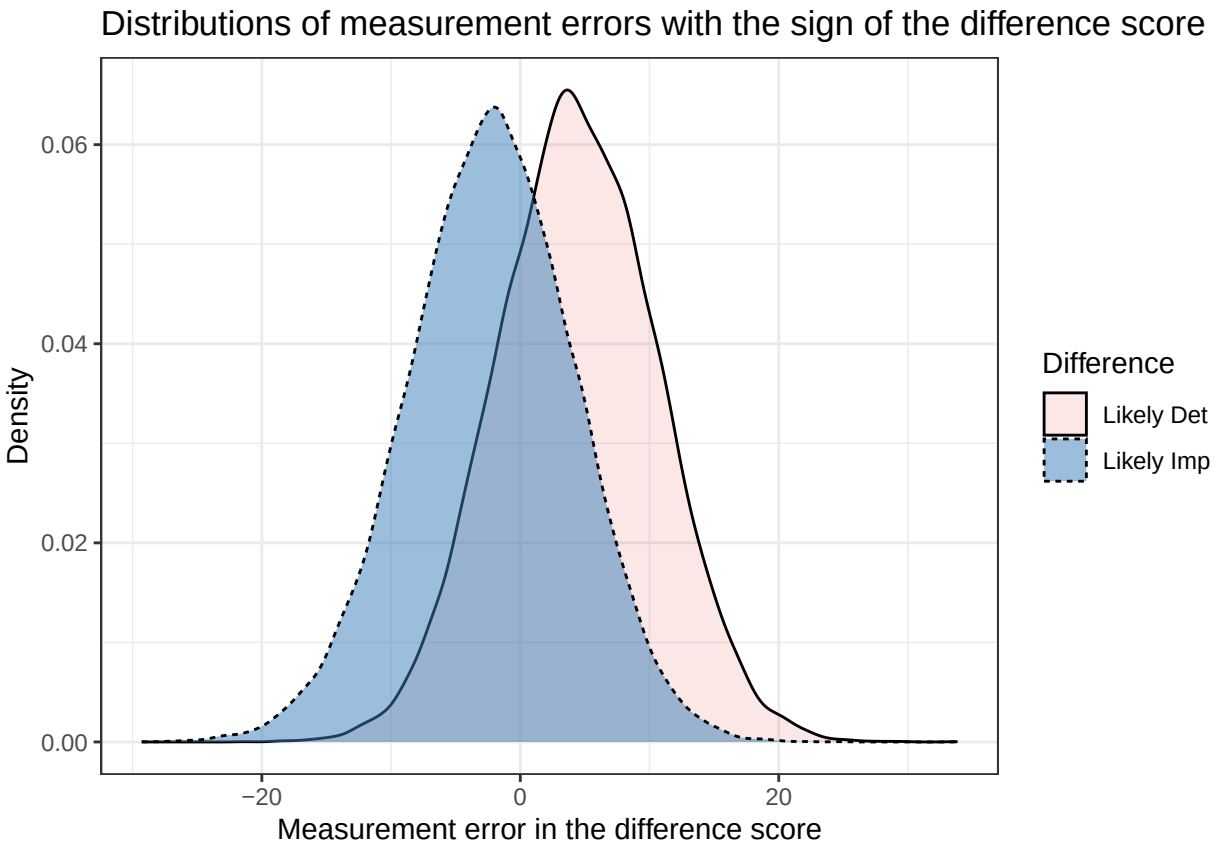
biased errors. In this data, we should be more confident in smaller magnitude difference scores, not less.

This is the statistical significance filter.

In contrast, here is what happens if we just use 0 as the cutpoint. In this case, every positive difference score is considered a "Likely Deteriorator" and every negative difference score is considered a "Likely Improver." Is the bias the same?

```
simdata3 <- simdata2 %>%
  mutate(Difference = case_when(obs_diff > 0 ~ "Likely Det",
                                obs_diff < 0 ~ "Likely Imp",
                                TRUE ~ "No obs change"))

errorDistSign <- simdata3 %>%
  ggplot(aes(x = error_diff,
             fill = Difference)) +
  geom_density(aes(alpha = Difference,
                   linetype = Difference)) +
  # scale_linetype_discrete(name = "RCI correct?",
  #                         labels = c("Correct", "Incorrect")) +
  scale_alpha_discrete(range = c(0.1, 0.5))+
  scale_fill_brewer(palette = "Set1")+
  # geom_density(aes(fill = Difference),
  #              alpha = .4) +
  # scale_fill_brewer(palette = "Set1") +
  theme_bw() +
  labs(y = "Density",
       x = "Measurement error in the difference score")
errorDistSign +
  labs(title = "Distributions of measurement errors with the sign of the difference score")
```



Clearly, there is still some bias occurring here. This is unavoidable in the context of error when using a cutpoint: this is one of the issues with categorizing a continuous variable. But there is considerably more overlap between these two groups than the reliably changed groups in the previous plot, indicating that their errors are more similar to each other.

```
simdata3 %>%
  group_by(Difference) %>%
  summarise(AvgAbsError = mean(abs(error_diff)),
            AbsError025 = quantile(abs(error_diff), .025),
            AbsError975 = quantile(abs(error_diff), .975),
            Bias = mean(obs_diff - true_diff),
            Bias025 = quantile(error_diff, .025),
            Bias975 = quantile(error_diff, .975)) %>%
  knitr::kable(digits = 2,
               col.names = c("Category",
                             "Abs. Err.",
                             "Abs. Err. 95% LB",
                             "Abs. Err. 95% UB",
                             "Bias",
                             "Bias 95% LB",
                             "Bias 95% UB")) %>%
  kableExtra::kable_styling(full_width = FALSE,
                            latex_options = "HOLD_position")
```

Category	Abs. Err.	Abs. Err. 95% LB	Abs. Err. 95% UB	Bias	Bias 95% LB	Bias 95% UB
Likely Det	6.09	0.26	16.47	4.39	-7.44	16.43
Likely Imp	5.40	0.22	15.39	-2.28	-15.08	9.99

While still displaying some bias, these groups have roughly similar errors as the sample as a whole, and they have less bias than the reliable change groups. This is the best possible situation for dichotomization of an error-prone variable, and is still not perfect.

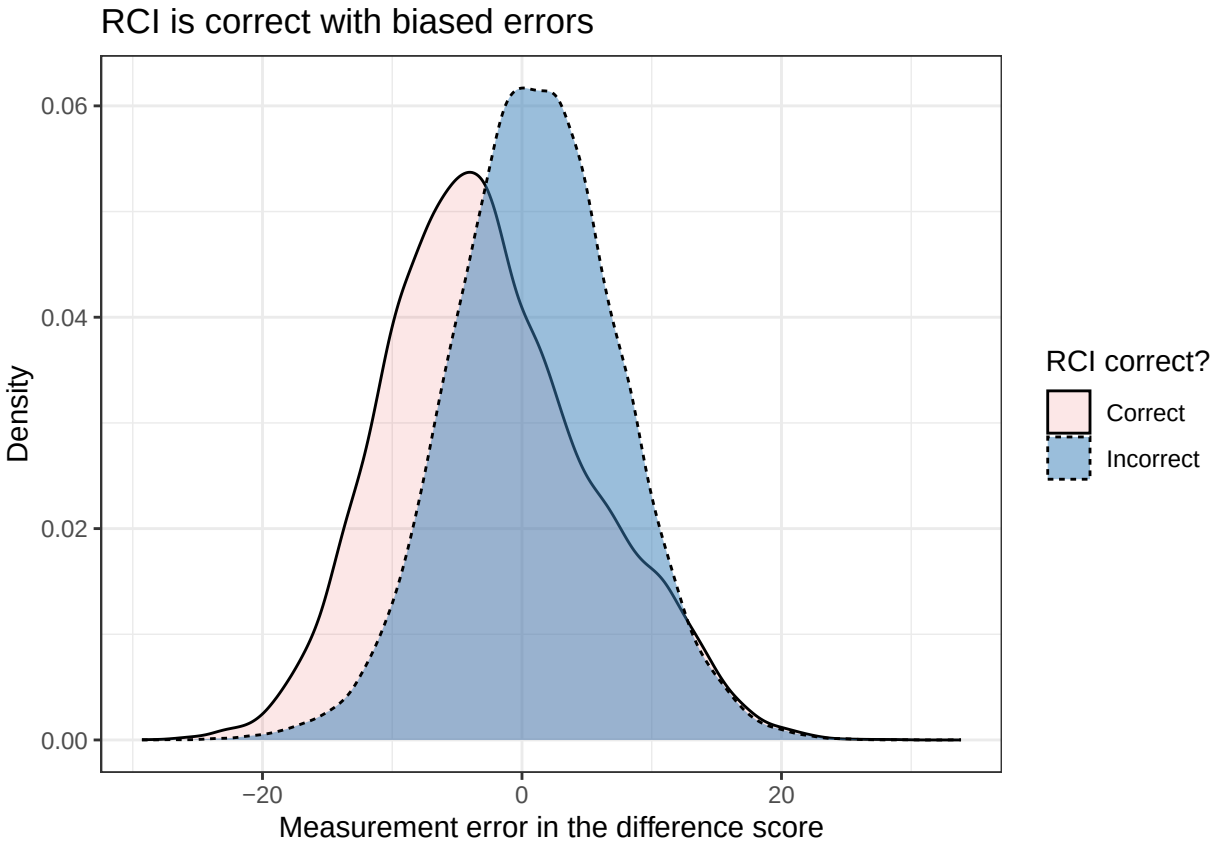
While this is not illustrated here, the magnitude of the bias decreases with higher SNR. So having more precise measurements will reduce these issues.

7.2 When the RCI is correct, it is still biased

An important point of this analysis is that when the RCI is correct, it will still be mistaken in a biased direction. Reproducing the plots above while only considering the RCI correct or incorrect based on the direction of the true change score, we can see the bias emerge:

```
simdata4 <- simdata3 %>%
  mutate(TrueChange = ifelse(true_diff > 0,
                             "TrueDet",
                             ifelse(true_diff < 0,
                                    "TrueImp",
                                    "TrueNo")),
         wrongRCI = case_when(Category == "Reliable Det" &
                                TrueChange == "TrueDet" ~ "RCI_correct",
                              Category == "Reliable Imp" &
                                TrueChange == "TrueImp" ~ "RCI_correct",
                              TRUE ~ "RCI_incorrect"),
         wrongSign = case_when(Difference == "Likely Det" &
                                 TrueChange == "TrueDet" ~ "Sign_correct",
                               Difference == "Likely Imp" &
                                 TrueChange == "TrueImp" ~ "Sign_correct",
                               TRUE ~ "Sign_incorrect"))

errorCorrectRCI <- simdata4 %>%
  ggplot(aes(x = error_diff,
             group = wrongRCI,
             fill = wrongRCI)) +
  geom_density(aes(alpha = wrongRCI,
                   linetype = wrongRCI)) +
  scale_linetype_discrete(name = "RCI correct?",
                          labels = c("Correct", "Incorrect")) +
  scale_alpha_discrete(range = c(0.1, 0.5),
                       name = "RCI correct?",
                       labels = c("Correct", "Incorrect"))+
  scale_fill_brewer(palette = "Set1",
                    name = "RCI correct?",
                    labels = c("Correct", "Incorrect"))+
  theme_bw() +
  labs(y = "Density",
       x = "Measurement error in the difference score")
errorCorrectRCI +
  labs(title = "RCI is correct with biased errors")
```

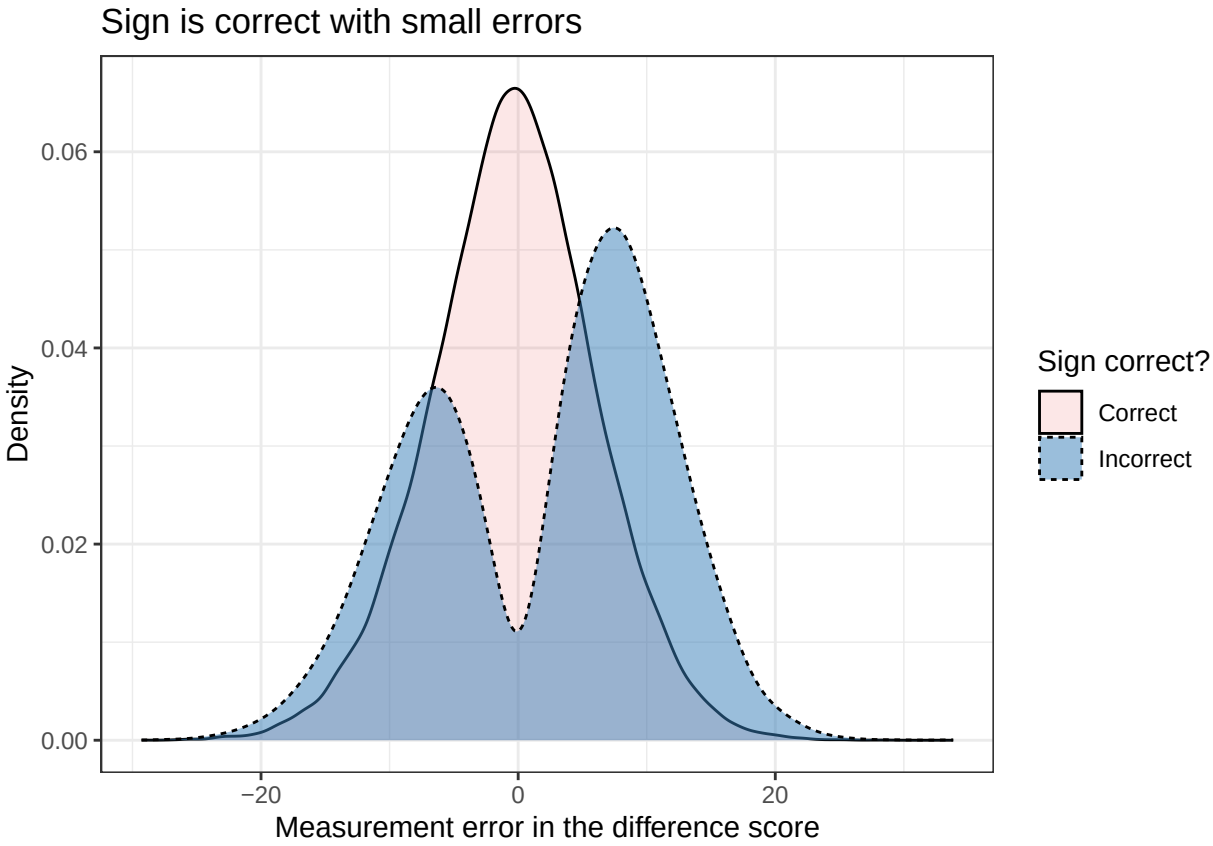


In that figure, the people whose RCI results were correct in either direction (improving or deteriorating) had right-skewed errors, while those whose RCI determinations were incorrect had less biased errors. The largest errors (those whose absolute value were in the 10-20 range) were much more common among the correct group, indicating that these determinations were correct largely by chance. The smallest errors (those between -5 and 5) were disproportionately incorrectly classified by the RCI.

In contrast, the sign of the difference score is most correct with smallest measurement errors.

```
errorCorrectSign <- simdata4 %>%
  ggplot(aes(x = error_diff,
             group = wrongSign,
             fill = wrongSign)) +
  geom_density(aes(alpha = wrongSign,
                   linetype = wrongSign)) +
  scale_linetype_discrete(name = "Sign correct?",
                          labels = c("Correct", "Incorrect")) +
  scale_alpha_discrete(range = c(0.1, 0.5),
                       name = "Sign correct?",
                       labels = c("Correct", "Incorrect"))+
  scale_fill_brewer(palette = "Set1",
                    name = "Sign correct?",
                    labels = c("Correct", "Incorrect"))+
  # scale_color_brewer(palette = "Set1") +
  # scale_fill_brewer(palette = "Set1",
  #                   name = "Sign correct?",
  #                   labels = c("Correct", "Incorrect"))+
  theme_bw() +
  labs(y = "Density",
```

```
       x = "Measurement error in the difference score")
errorCorrectSign +
  labs(title = "Sign is correct with small errors")
```



In theory, the accuracy of observations should be the important factor for whether you wish to use a measure to inform individual treatment decisions. If this is the case, using the RCI means that the noisiest estimates will be frequently in the "reliable" change groups, so their changes will be illusory.

8 True score changes during test-retest intervals

What if there is some true change in the test-retest interval? How will the RCI be affected? Of course, this is an easy question to answer: The RCI will be less sensitive than it should be, because the reliability estimate will be artificially deflated. But how much of an effect does this have in practice?

We will simulated data to illustrate this issue. Here `simdataTrueChange` is created with Mean of 50 and SD = 10, then we add some measurement error variability. This is basically a repeat of `simdata` from earlier in the supplement.

```
simdataTrueChange <- data.frame(true_score = rnorm(20000, mean = 50, sd = 10)) %>%
  mutate(t1_obs = true_score + rnorm(20000, sd = 5),
         t2_obs = true_score + rnorm(20000, sd = 5)) %>%
  mutate(error.rep = t2_obs - t1_obs)
```

In that data, both `t1_obs` and `t2_obs` are derived from `true_score`, meaning that there is no real change between T1 and T2.

What if, instead of just being a function of the true score plus error, the time 2 observation is truly different from time 1? Not by much, and no change on average, but each person might change with an SD of 5 points.

That is, people have fluctuations in their scores not only due to error, but because everyone experiences the test-retest interval differently and their symptoms will change naturally over time.

In this section, we simulate the effects of this true-score variation during the test-retest interval.

8.1 First simulation: Moderate change to noise ratio

In this simulation, the magnitude of the true change during test-retest is variable (some people will have a lot and others will have nearly 0), but is on average roughly as large as the effect of measurement error at each observation. I am calling this moderate change, though there is no clear criterion on which to judge the magnitude. The proper scale on which to judge this change is the signal to noise ratio, which is the amount of true change relative to the amount of error. In this simulation, they are the same average magnitude (though some people will have large errors, large true changes, both, or neither).

```
# t2_obs2 will have a random measurement error component
# and a random true score change component.
simdataTrueChange2 <- mutate(simdataTrueChange,
      true_score2 = true_score +  rnorm(20000, sd = 5),  # the true change component
      t2_obs2 =     true_score2 + rnorm(20000, sd = 5))  # the error component
```

In our new data, `true_score2` is the true score at time 2, while `t2_obs2` is the observed score at time 2. A test-retest coefficient that uses `t2_obs2` instead of `t2_obs` will not be appropriate because it has a small amount of true score variability included.

We already know how to compute the **true S_{diff}** value, `sqrt(2 * (sd(simdataTrueChange2$t1_obs) * sqrt(1 - cor(simdataTrueChange2$t1_obs, simdataTrueChange2$t2_obs)))^2)` 7.0344916, from our previous work with this simulated data. What if, instead of having the true test-retest value as we already had, we used the imperfect test-retest interval? What are the effects on Reliability, S_{diff}, and RCI?

Reliability:

```
cor(simdataTrueChange2$t1_obs, simdataTrueChange2$t2_obs2)
```

```
## [1] 0.7234591
```

The estimate of r_{xx} would be 0.723 where it should be 0.799 in truth.

It seems to have only dropped the reliability a few points. However, those few points are meaningful.

We can calculate our newly (mis-)estimated S_{diff} value:

```
Sdiff.sim.2 <- sqrt(2 * (sd(simdataTrueChange2$t1_obs) *
                           sqrt(1 - cor(simdataTrueChange2$t1_obs,
                                        simdataTrueChange2$t2_obs2)))^2)
```

S_{diff} would now be estimated to be 8.249, while it is truly 7.034.

This leads to the RCI value of 16.169 rather than the true value of 13.787.

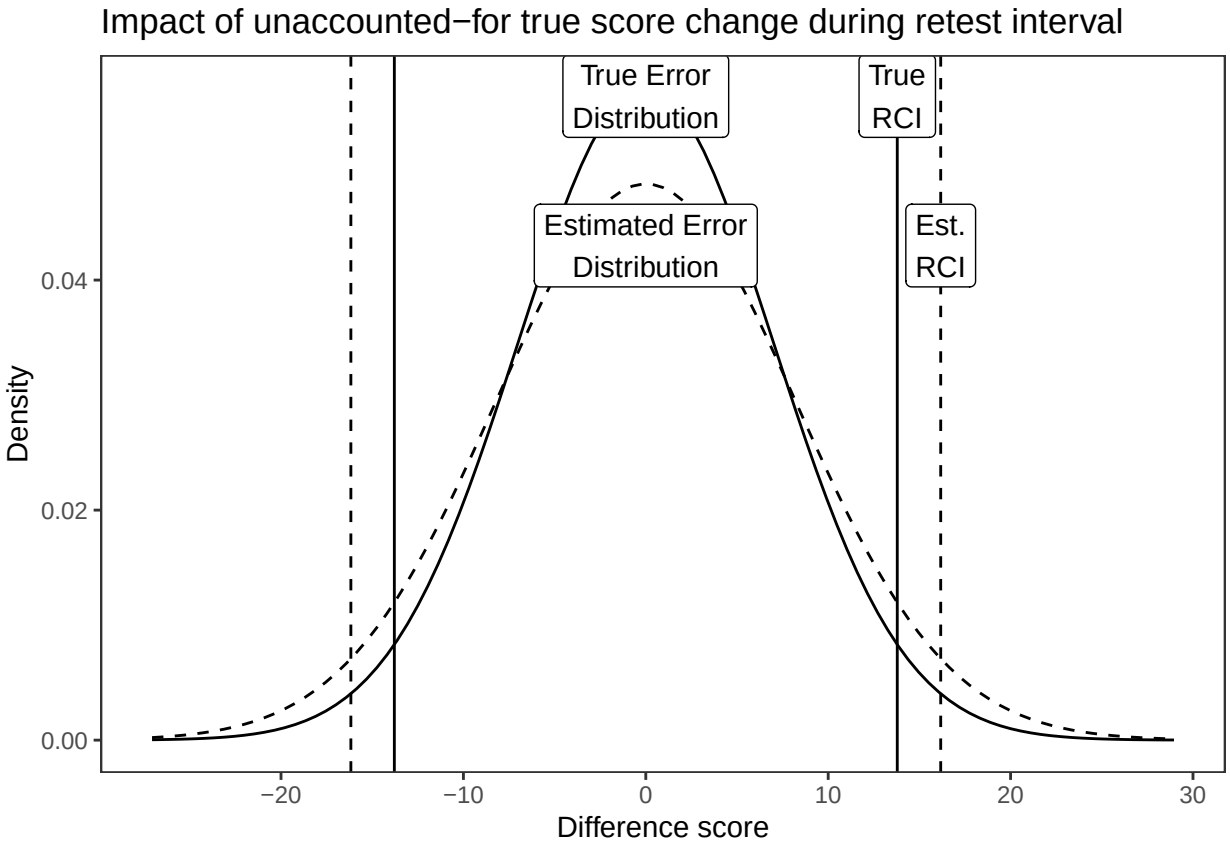
The small amount of true score change - which had a mean of 0 in the population - has increased our RCI by more than two points.
In this data for context, that is $d = 0.2$, or a 17.28% change. So the RCI loses sensitivity by an appreciable margin.

Here is a visualization:

```
simdataTrueChange2 <- simdataTrueChange2 %>%
  mutate(obs_diff = t2_obs - t1_obs,
         obs_diff2 = t2_obs2 - t1_obs)

ggplot(simdataTrueChange2,
```

```
     aes(x = obs_diff)) +
geom_function(fun = dnorm,
              args = list(mean = 0, sd = Sdiff.sim)) +
geom_function(fun = dnorm,
              args = list(mean = 0, sd = Sdiff.sim.2),
              linetype = "dashed") +
geom_vline(xintercept = 1.96 * c(Sdiff.sim, Sdiff.sim.2),
           linetype = c("solid", "dashed")) +
geom_vline(xintercept = -1.96 * c(Sdiff.sim, Sdiff.sim.2),
           linetype = c("solid", "dashed")) +
geom_label(data = tibble(label = "True Error\nDistribution"),
           aes(x = 0,
               y = 0.056,
               label = label)) +
geom_label(data = tibble(label = "Estimated Error\nDistribution"),
           aes(x = 0,
               y = 0.043,
               label = label)) +
geom_label(data = tibble(label = "True\nRCI"),
           aes(x = 1.96 * Sdiff.sim,
               y = 0.056,
               label = label)) +
geom_label(data = tibble(label = "Est.\nRCI"),
           aes(x = 1.96 * Sdiff.sim.2 ,
               y = .043,
               label = label)) +
labs(title = "Impact of unaccounted-for true score change during retest interval",
     y = "Density",
     x = "Difference score")
```



The impact of including even a relatively small amount of true change in the retest interval is to create a misleadingly large RCI value. This will be unduly insensitive.

8.2 Second simulation: Same changes, higher reliability

Since that simulation assumed a fairly unreliable ($r_{xx} = .8$) measure to begin with, and a highly dependable measure will also be subject to the same between-person true score changes, we might want to change some of the parameters of the simulation.

Specifically, we will make the measure more accurate at each time point without changing the scale of true score change (reduce the error variance relative to the true score variance). This would be similar to comparing a more reliable instrument to a less reliable one, but both have erroneously included true score changes in their computation of the test-retest (and therefore RCI).

```
simdataTrueChange3 <- data.frame(true_score = rnorm(20000, mean = 50, sd = 10)) %>%
  mutate(t1_obs = true_score + rnorm(20000, sd = 2),
         t2_obs = true_score + rnorm(20000, sd = 2),
         t2_obs2 = true_score + rnorm(20000, sd = 2) +
           rnorm(20000, sd = 5))
```

Now, our **true reliability** is much higher than in the previous simulation: $r_{xx} = 0.961$ where it was 0.799 in the previous simulation.

The estimated reliability, however, would be: 0.863. This is much lower.

S_{diff} would now be estimated to be 5.32, but its true value is 2.817. Again, much too large.

```
Sdiff.sim.est <- sqrt(2 * (sd(simdataTrueChange3$t1_obs) *
                             sqrt(1 - cor(simdataTrueChange3$t1_obs,
                                          simdataTrueChange3$t2_obs2)))^2)
```

```
Sdiff.sim.true <- sqrt(2 * (sd(simdataTrueChange3$t1_obs) *
                             sqrt(1 - cor(simdataTrueChange3$t1_obs,
                                          simdataTrueChange3$t2_obs)))^2)
```

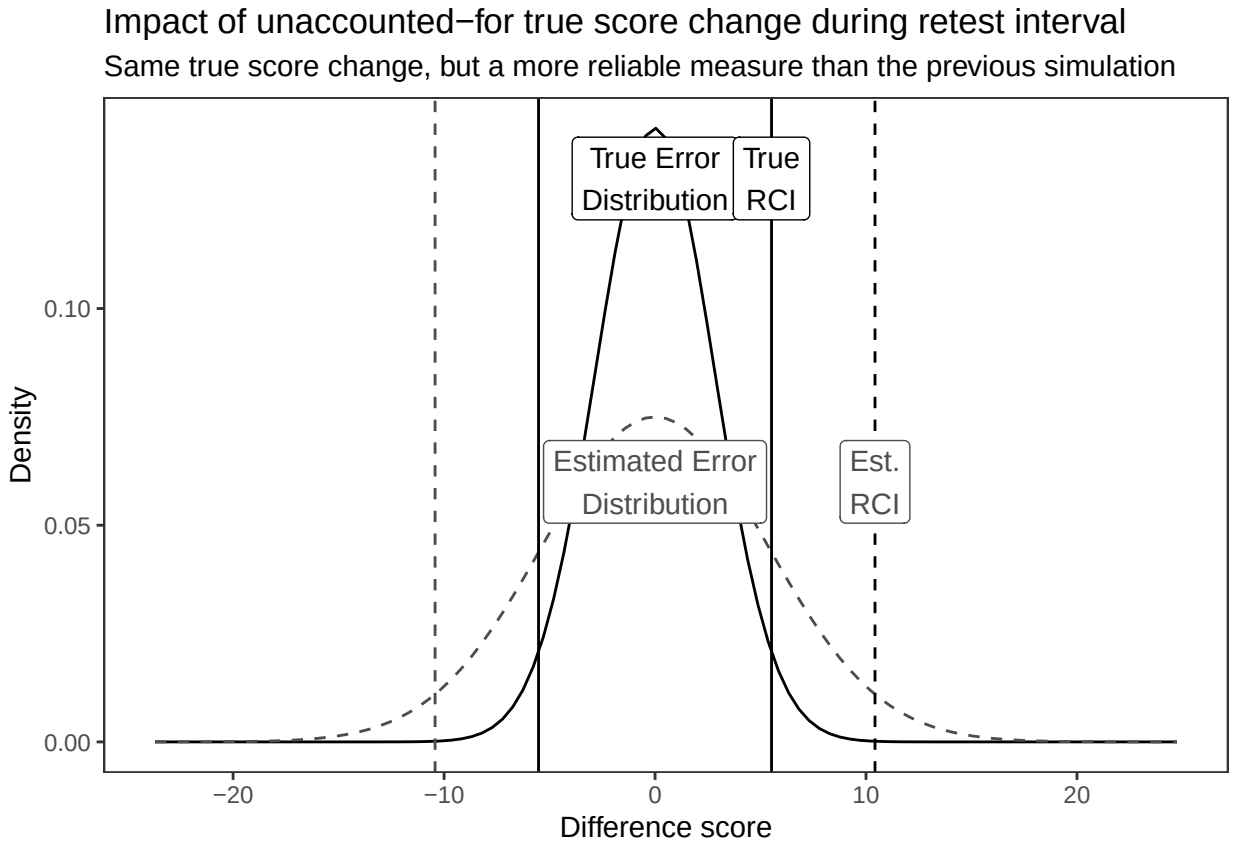
The RCI we would estimate from using this more accurate measure but including true score changes in the retest interval is therefore 10.426, when it should be 5.521.

Because our measure is actually more accurate than in the previous simulation, the estimated RCI is 88.83% larger than it should be.

Including true score variability in the retest interval more strongly affects more accurate measures.

```
simdataTrueChange3 <- simdataTrueChange3 %>%
  mutate(obs_diff = t2_obs - t1_obs,
         obs_diff2 = t2_obs2 - t1_obs)

ggplot(simdataTrueChange3,
       aes(x = obs_diff2)) +
  geom_function(fun = dnorm,
                args = list(mean = 0, sd = Sdiff.sim.true)) +
  geom_function(fun = dnorm,
                args = list(mean = 0, sd = Sdiff.sim.est),
                linetype = "dashed",
                color = "gray30") +
  geom_vline(xintercept = c(1.96* Sdiff.sim.true,
                            1.96* Sdiff.sim.est),
             linetype = c("solid", "dashed")) +
  geom_vline(xintercept = -1.96 * c(Sdiff.sim.true, Sdiff.sim.est),
             linetype = c("solid", "dashed"),
             color = c("black", "gray30")) +
  geom_label(data = tibble(label = "True Error\nDistribution"),
             aes(x = 0,
                 y = .13,
                 label = label)) +
  geom_label(data = tibble(label = "Estimated Error\nDistribution"),
             aes(x = 0,
                 y = .06,
                 label = label),
             color = "gray30") +
  geom_label(data = tibble(label = "True\nRCI"),
             aes(x = 1.96 * Sdiff.sim.true,
                 y = .13,
                 label = label)) +
  geom_label(data = tibble(label = "Est.\nRCI"),
             aes(x = 1.96 * Sdiff.sim.est ,
                 y = .06,
                 label = label),
             color = "gray30") +
  labs(title = "Impact of unaccounted-for true score change during retest interval",
       y = "Density",
       x = "Difference score",
       subtitle = "Same true score change, but a more reliable measure than the previous simulation")
```



The RCI is almost twice as large as its true value - meaning **much less sensitive** - because of this true score change that occurred during test-retest. Again, almost everyone changed less than the SD of the population, so if they were in a "normal" range at time 1 they were likely to stay there at time 2. This is well within the range of variability that could occur on many psychological variables over weeks to months, even in a non-clinical sample.

This is the danger of including any true score variability in the test-retest interval. It can dramatically decrease the ability of the RCI to detect change, and punishes more accurate measures very heavily, diminishing their apparent reliability.

8.3 Estimating over-estimation of S_{diff}

Another way to think about this is to hypothesize about how much common instruments' RCIs overestimate S_{diff}.

For instance, Delgadillo and colleagues (2011) report a 4-week test-retest value for the PHQ-9 of $r = .78$, which has been used at least once (Beard, C., Hsu, K. J., Rifkin, L. S., Busch, A. B., & Björgvinsson, T. (2016). Validation of the PHQ-9 in a psychiatric sample. *Journal of Affective Disorders*, 193, 267-273.) to calculate the RCI for this instrument. That study reports only 30% of patients made "reliable" changes during treatment according to their RCI.

The question that must be answered is this: Over four weeks, what is the expected absolute value of change in *true* PHQ-9 score for individuals? I have no firm answer for this, but we could imagine that most people will not change more than a few points on this 9-item measure. A conservative estimate might then be a SD of 1 point, so that almost every true value would be within 2 points at both times. However, spontaneous remission and sudden losses of functioning do occur, so some small percent might dramatically change over any given 4 weeks. Therefore, a more realistic estimate might be an SD of 3, such that almost everyone will be within 5-6 points of their first true value at the second time, but a few people can have very significant changes of 10 points or so.

If we take the very conservative approach that the true change has SD = 1, we can work out the true RCI:

```
RCIBeard <- 8.6
rxxBeard <- .78
SdiffBeard <- (RCIBeard/1.96)
SEmBeard <- SdiffBeard / sqrt(2)
SDBeard <- SEmBeard / sqrt(1 - rxxBeard)
# RCI returns here:
SDBeard * sqrt(1 - rxxBeard) * sqrt(2) * 1.96
```

```
## [1] 8.6
```

In the Beard and colleagues study, the standard error for measurement errors in the PHQ-9 (called `SEmBeard` here) is therefore 3.1026114.

Since we also know what the total observed variation is, the true score variation therefore, has a known SD.

```
SDBeardTrue <- sqrt(SDBeard^2 - SEmBeard^2)
```

`sqrt(SDBeard^2 - SEmBeard^2)`, or 5.8420241, is the true score variance of the scores in their study.

We can now simulate true and observed difference scores from this study.

```
simBeard <- tibble(true_phq = rnorm(55555, mean = 14.5, sd = SDBeardTrue),
                   obs_phq = true_phq + rnorm(55555, sd = SEmBeard))
```

We simulated the data to demonstrate that we know what the implied ratio of true score to random score change is in this data. The observed variables have the same SD as the reported data:

```
sd(simBeard$obs_phq)
```

```
## [1] 6.638562
```

Compare that value to `SDBeard`, which is 6.6147897. Very close, considering simulation and rounding errors.

Again, assuming that there is *very little* true score change during one month (almost no one has a true score change greater than 3 points on this scale within one month, ever), we can simulate data without this component. We need to first work out what the true SEm should be, which is variance subtraction.

The true SEM is:

```
SEmBeardTrue <- sqrt(SEmBeard^2 - 1^2)
```

SEmBeardTrue.

If the true scores change with SD = 1 point over the course of a month, then the accurate error of PHQ-9 scores would have a SD of `sqrt(SEmBeard^2 - 1^2)` if we could observe them, because this is the observed variance minus the true score variance. We will call this `SEmBeardTrue`.

```
simBeard <- mutate(simBeard,
                   obs_phq2 = true_phq - (14.5 - 9.89) +
                     rnorm(55555, sd = SEmBeard),
                   obs_phq2_clean = true_phq - (14.5 - 9.89) +
                     rnorm(55555, sd = SEmBeardTrue))
```

The resulting reliabilities are different but not that different.

The reliability they used:

```
cor(simBeard$obs_phq, simBeard$obs_phq2)  # the reliability they used
```

```
## [1] 0.7830052
```

While this was the reliability they should have used, under the assumed true-score change during the interval:

```r
cor(simBeard$obs_phq, simBeard$obs_phq2_clean)
```

```
## [1] 0.7934933
```

```r
# the reliability filtering out some true change during retest interval
```

And the appropriate RCI would be similar as well. They used this as the RCI:

```r
RCIBeard
```

```
## [1] 8.6
```

But we would use this:

```r
RCIBeard2 <- 1.96 * SEmBeardTrue * sqrt(2)
RCIBeard2
```

```
## [1] 8.141056
```

How much more "sensitive to change" is this?

```r
simBeard <- simBeard %>%
  mutate(relImp = case_when(abs(obs_phq2 - obs_phq) > RCIBeard ~ "Imp",
                            TRUE ~ "No"),
         relImpClean = case_when(abs(obs_phq2 - obs_phq) > RCIBeard2 ~ "Imp",
                            TRUE ~ "No"))
```

In this simulation, a smaller proportion of people were reliably changed using their RCI, compared to the 30% of people they report as reliably changed (likely due to nonnormality and variable treatment effects in the data which is not reported in the text). We assume that this is acceptable for the present purposes; the comparison of interest is within the simulation rather than compared to true data. If we were to re-analyze the raw data for a study, greater precision would be possible.

Here are the reliably improved versus no-change group frequencies:

```r
knitr::kable(table(simBeard$relImp),
             col.names = c("Reliable change", "Frequency")) %>%
  kableExtra::kable_styling(full_width = FALSE,
                            latex_options = "HOLD_position")
```

Reliable change	Frequency
Imp	9989
No	45566

The percent with "reliable" change is 18% in this simulation, using the authors (potentially erroneous) assumption that no true score change occurred in the retest interval.

However, using the *very slightly* different version of the RCI, which assumes people might have changed a very small amount during the test-retest period, we see that more Improvement is detected:

```r
knitr::kable(table(simBeard$relImpClean),
             col.names = c("Reliable change", "Frequency")) %>%
  kableExtra::kable_styling(full_width = FALSE,
                            latex_options = "HOLD_position")
```

Reliable change	Frequency
Imp	11595
No	43960

This results in 20.9% of people considered reliably changed.

This may not sound like much but it represents a **16.078% increase** in the number of patients detected as changed.

What if we believe that there could be more true score change than would occur with SD = 1? After all, some people completely remit from depression during a given month. Perhaps we could use an SD of 2, which would mean 96% of people would change less than 4 points, but maybe one in 750 people could truly change 6 points in a month (that is roughly how frequently a 3 SD difference would occur). This amount is close to the RCI of the PHQ-9, in many studies, suggesting that changes this large should be very rare in a non-treatment population, but could occur nonetheless.

Here is the simulation code:

```
SEmBeardTrue <- sqrt(SEmBeard^2 - 2^2)
simBeard <- mutate(simBeard,
                   obs_phq2 = true_phq - (14.5 - 9.89) + rnorm(55555, sd = SEmBeard),
                   obs_phq2_clean = true_phq - (14.5 - 9.89) + rnorm(55555, sd = SEmBeardTrue))
```

The reliability they used, again:

```
cor(simBeard$obs_phq, simBeard$obs_phq2)  # the reliability they used
```

```
## [1] 0.7824068
```

Versus the reliability they should have used under this assumption:

```
cor(simBeard$obs_phq, simBeard$obs_phq2_clean)  # the reliability filtering out some true change
```

```
## [1] 0.8217237
```

```
RCIBeard3 <- 1.96 * SEmBeardTrue * sqrt(2)
simBeard <- simBeard %>%
  mutate(relImpClean2 = case_when(abs(obs_phq2 - obs_phq) > RCIBeard3 ~ "Imp",
                                  TRUE ~ "No"))
```

This assumption results in 33.1% of people considered reliably changed.

This represents a **84.072% increase** in the number of patients detected as changed.

If, during the course of one month, there is some true change that is quite small on average - 96% of people stay within 4 points of their original scores, and 68% stay within 2 points - the RCI used in this study is almost half as sensitive as it should be, detecting almost half of the people it should detect if the retest interval were perfectly appropriate.

There is no obvious way to accurately estimate exactly how much true change there is either on average for a sample or for any individual person. All we have is observations. However, for symptoms of depression, it seems reasonable to assume that there is *some* true change for most people across one month, even if most people are almost exactly at the same level they were before and the average change is 0.

Put another way, it is not reasonable to attribute all instability in depression score over four weeks to error alone. The consequences of choices like this one can affect treatments and services provided to individuals and at large scale, because of the way the RCI is typically used.

Even if the change is extremely small, it will cause the RCI to be considerably less sensitive than it should be, simply by using an inappropriate test-retest interval.

To address this, shorter test-retest intervals are necessary. Even if all the other challenges to interpreting and using the RCI are not an issue in the particular case, using an inappropriate retest interval will still cause substantial insensitivity.

It should be noted that all of these issues are additive: the RCI is too insensitive because it is computed with inappropriate retest intervals, but also insensitive because it prioritized Type I errors over Type II errors, is not applicable in cases of floor effects, etc. To justify the RCI, all of the criteria should be met, not just some. This seems hard to do.