Behav Res
DOI 10.3758/s13428-016-0771-8

Using propensity score matching to construct experimental stimuli

Stefan Huber[1] · **Julia F. Dietrich**[1,2] · **Benjamin Nagengast**[3,4] · **Korbinian Moeller**[1,2,4]

© Psychonomic Society, Inc. 2016

Abstract Propensity score matching is widely used in various fields of research, including psychology, medicine, education, and sociology. It is usually applied to find a matched control group for a treatment group. In the present article, we suggest that propensity score matching might also be used to construct item sets matched for different parameters. We constructed stimuli to illustrate the use of propensity score matching in item construction for the exemplary cases of numerical cognition research and reading research. In particular, we provide a step-by-step approach, using the statistics software R, for how to apply propensity score matching for constructing matched stimuli. This approach involves deciding on a population of stimuli, determining and calculating the covariates, and finally applying the propensity-matching method to find a set of items matched to another predefined set. Thereby, we were able to construct well-matched item sets for both examples. Hence, we conclude that the propensity-score-matching method is useful for constructing matched stimuli. Further cases of application are discussed.

Keywords Propensity score · Numerical cognition · Fraction · Reading · Stimuli construction

✉ Stefan Huber
s.huber@iwm-tuebingen.de

[1] Leibniz-Institut für Wissensmedien, Schleichstrasse 6, 72076 Tübingen, Germany

[2] Department of Psychology, Eberhard Karls University Tübingen, Tübingen, Germany

[3] Hector Research Institute of Education Sciences and Psychology, Eberhard Karls University Tübingen, Tübingen, Germany

[4] LEAD Graduate School, Eberhard Karls University Tübingen, Tübingen, Germany

In experimental research, the effects of manipulating independent variables on one or more dependent variables are usually evaluated while controlling for several covariates. Controlling covariates across independent variables is of special importance to be able to conclude that a potential effect can be linked to the variable under manipulation. For instance, in numerical cognition an important research topic is the processing of the magnitude of symbolic numbers. The most common task to study the magnitude processing of symbolic number is the *magnitude comparison task*, in which the larger of two numbers has to be identified. An important predictor of the performance in magnitude comparison is the numerical distance between the numbers (Moyer & Landauer, 1967). Generally, participants' responses get longer and more error-prone as the distance between the numbers decreases (e.g., 1 vs. 9, distance = 8, as compared to 4 vs. 5, distance = 1). This effect has to be controlled for when, for instance, studying whether two-digit numbers are processed holistically (i.e., as an integrated entity) or componentially (i.e., separated into units, tens, hundreds, etc.; e.g., Nuerk, Weger, & Willmes, 2001).

To study the processing of two-digit numbers, Nuerk et al. (2001) contrasted two different kinds of two-digit number pairs: unit–decade compatible pairs, for which the separate comparisons of tens and units yielded the same decision biases (e.g., 42 vs. 57, in which both $4 < 5$ and $2 < 7$), and incompatible number pairs, for which separate comparisons of the tens and units yielded opposing decision biases (e.g., 47 vs. 62, in which $4 < 6$ but $7 < 2$). To guarantee the validity of the hypothesized unit–decade compatibility effect, it was important to match the incompatible and compatible number pairs for several covariates. In sum, the authors controlled for 11 covariates (among them, overall distance), to ensure that the response time differences between compatible and incompatible number pairs could not be attributed to the covariates under control, but rather to the specific manipulation of unit–decade compatibility.

Published online: 15 July 2016

The necessity of such control for covariates is not restricted to the study of two-digit numbers, but generalizes to other tasks in numerical cognition research (e.g., mental arithmetic or the comparison of nonsymbolic quantities) and beyond the number domain (e.g., word frequency in reading research). However, often controlling the covariates across independent variables can be challenging, time-consuming, and daunting. In the present article, we show how to use propensity score matching (see, e.g., Stuart, 2010) for matching covariates, as an alternative to matching them by hand.

Propensity score matching is most commonly applied to match treatment groups and control groups according to several covariates in quasi-experimental settings (e.g., Guo & Fraser, 2010). For instance, treatments had been implemented before conducting the study. A particular problem when using quasi-experimental settings, however, is that the treatment effect might not be unbiased. Hence, treatment and control groups might differ according to several covariates. One solution to this problem is to balance nonequivalent groups using propensity scores, as was suggested by Rosenbaum and Rubin (1983). Propensity scores are most commonly obtained by running logistic regression. When using logistic regression, the dependent variable is the treatment-versus-control group dichotomous variable, and the independent variables are the covariates according to which the groups should be balanced. By running the logistic regression analysis, the predicted probability of each participant falling in either of the groups can be calculated. These probabilities are the propensity scores. Hence, the propensity score is the conditional probability of a participant receiving the treatment given the covariates. Propensity scores can be used as a measure of the distance between two individuals. These distance measures are needed to determine whether an individual is a good match for another.

The four primary distance measures are the exact distance, Mahalanobis distance, propensity score, and linear propensity score (Stuart, 2010). The exact distance is 0 if the vectors of covariates of two individuals are equal; otherwise, it is infinite. The Mahalanobis distance is similar to the Euclidean distance between two vectors (i.e., the sum of the squared differences in the covariates), but also considers the correlation structure via the inverse variance–covariance matrix. The difference between the propensity score and linear propensity score is that, in the first method, the raw probability is used to calculate the distance, whereas for the linear propensity score, the probability is logit-transformed before calculating the distance.

Various matching methods use any of these distance measures, or a combination of them (Guo & Fraser, 2010; Stuart, 2010). Among them are exact matching (Rosenbaum & Rubin, 1983), the nearest neighbor (e.g., Rubin, 1973), optimal matching (e.g., Gu & Rosenbaum, 1993), and subclassification (e.g., Cochran, 1968). The *exact* matching method matches each participant with another that has exactly the same values on each covariate. In the simplest form of the nearest-neighbor matching method, *1:1 nearest-neighbor matching*, for each treated participant another participant from the control group is identified, so that the distance between the participant becomes minimal. The *optimal* matching method tries to minimize the mean absolute distances across all matched pairs. The *subclassification* method creates subgroups that are similar with respect to a criterion—for instance, according to the quintiles of the propensity score distribution.

Propensity score analysis has been applied to various fields, including psychology (e.g., Jones, D'Agostino, Gondolf, & Heckert, 2004), medicine (e.g., Gum, Thamilarasan, Watanabe, Blackstone, & Lauer, 2001), education (e.g., Adelson, 2013), and sociology (e.g., Smith, 1997). However, to date we are not aware of any such study in numerical cognition, suggesting that such propensity-score-matching methods might be feasible for constructing matched sets of stimuli. Therefore, in the present study we provide an example of how to apply propensity score matching to construct matched stimuli for fraction magnitude comparison tasks. Importantly, however, propensity score matching is not restricted to numerical cognition research. Instead, it may be applied whenever stimulus sets have to be matched on the parameter values of several covariates. To substantiate that this approach is also applicable to other domains of research, we provide another example, from reading research, of how to create matched sets of words considering several covariates (e.g., word length and rated familiarity).

In the course of these two examples, we will use the R statistics software (R Development Core Team, 2015) and the R package MatchIt (Ho, Imai, King, & Stuart, 2011), which implements various methods for matching (for a step-by-step guide to MatchIt, see Randolph, Falbe, Manuel, & Balloun, 2014). In both examples, we will use the linear propensity score as the distance measure and the nearest-neighbor method for finding matched pairs. Moreover, we will use the following procedure to construct the stimuli: (1) deciding on a population of stimuli and determining covariates, and (2) applying the propensity-matching method to find one or two matched sets for a base set.

Constructing matched fraction pairs

Research on the mental representation of fractions has gained increasing interest recently (Siegler, Fazio, Bailey, & Zhou, 2013). The representation of fractions is mostly studied using magnitude comparison tasks, in which participants are presented two fractions and have to decide which of them is the numerically larger one (e.g., Bonato, Fabbri, Umiltà, & Zorzi, 2007; Dewolf, Grounds, Bassok, & Holyoak, 2014; Faulkenberry & Pierce, 2011; Ganor-Stern, Karasik-Rivkin,

& Tzelgov, 2011; Meert, Grégoire, & Noël, 2009, 2010). Studies employing this task have revealed that participants use component-based comparison strategies, focusing on either the numerators or the denominators, when they can easily be applied—for instance, when two fractions share common components (e.g., 3/7 vs. 4/7; see, e.g., Huber, Moeller, & Nuerk, 2014; Ischebeck, Weilharter, & Körner, 2016; Meert et al., 2009). In contrast, participants seem to rely on the overall magnitude of the whole fractions when decision-relevant components cannot be identified easily (Meert et al., 2010; Schneider & Siegler, 2010; Sprute & Temple, 2011).

However, even when the overall magnitude of the whole fractions is the most important predictor of response times and error rates, the components nevertheless exert an influence on the comparison process (e.g., DeWolf & Vosniadou, 2015; Meert et al., 2010; Obersteiner, Van Dooren, Van Hoof, & Verschaffel, 2013). For instance, Meert et al. found that the distance between numerators was also a significant predictor of response times, in addition to the overall distance of the whole fractions. Moreover, Obersteiner et al. showed that response times and error rates differed depending on whether separate comparisons of the components were congruent or incongruent with the comparison of the overall magnitude of the whole fractions. In this context, fraction pairs without common components can be subdivided into three groups: (1) separate comparisons of the components are congruent with the comparison of the fractions (e.g., $3/8 < 7/9$, with $3 < 7$ and $8 < 9$; a *congruent, congruent pair: CC*), (2) the comparison of the numerators is congruent, but the comparison of the denominators is incongruent with the comparison of the fractions (e.g., $3/8 < 4/5$, with $3 < 4$ and $8 > 5$; a *congruent, incongruent pair: CI*), and (3) the separate comparisons of both components are incongruent with the comparison of the fractions (e.g., $4/7 < 3/5$, with $4 > 3$ and $7 > 5$; an *incongruent, incongruent pair: II*).[1] To be able to evaluate whether the comparison of components indeed influences the comparison of the overall fraction magnitudes, as indicated by different response times and error rates for the fraction pair types (CC, CI, and II), it is necessary to ensure that in the item sets employed, these fraction pairs do not differ with regard to their overall distance between the whole fractions. In the following sections, we will outline how an item set for these different fraction pair types (CC, CI, and II) with matched overall mean distances between the whole fractions can be created by relying on the propensity-score-matching method.

[1] Note that fraction pairs with incongruent numerators and congruent denominators do not exist. This can be easily shown. Without loss of generality, we set the relationship of two fractions a/b and c/d to $a/b < c/d$, with a, b, c, and d being natural numbers. Accordingly, incongruent numerators would mean that $a > c$, and congruent denominators that $b < d$ or $d > b$. This would imply that $a \times d > c \times d$, which is a contradiction to $a/b < c/d$, because $a/b < c/d$ is equivalent to $a \times d < c \times d$. Therefore, it is not possible to construct fraction pairs with incongruent numerators and congruent denominators.

Item population

The first step in creating an item set for fraction comparison is to decide on the population of fractions that should be used in the study. At large, fractions can be divided into proper and improper fractions. A fraction is called *proper* when the numerator is smaller than the denominator (e.g., 3/7 with $3 < 7$), and *improper* otherwise. Ischebeck, Schocke, and Delazer (2009) found differences in response times and error rates between proper and improper fractions even when distances were matched. Hence, when creating an item set for investigating fraction processing, researchers should decide whether to include both proper and improper fractions and matching fraction pair types according to this property, or whether to focus on either proper or improper fractions. We chose to include only proper fractions in this example. Moreover, usually only irreducible fractions are used such that participants are not able to apply the strategy of simplifying fractions before comparing them.

Another relevant property is the number of digits in the numerators and denominators. Studies have employed either only single-digit fractions (Bonato et al., 2007; Huber et al., 2014; Meert et al., 2009, 2010) or a combination of single-digit and multidigit fractions (DeWolf & Vosniadou, 2015; Obersteiner et al., 2013; Schneider & Siegler, 2010). Ischebeck et al. (2016) investigated explicitly whether single-digit and two-digit fractions are processed differently in a magnitude comparison task, and they found evidence that the results were similar for single-digit and two-digit fractions. Thus, it seems safe to include both single-digit and two-digit fractions in the same item set.

Taken together, we used irreducible and proper single-digit and two-digit fractions to construct an item set discriminating the three fraction pair types: CC, CI, and II. In this process, we controlled for the overall distance between the whole fractions, the numerator distance, as well as denominator distance (for an implementation of the procedure using R, please see Appendix A). However, apart from relying on fraction, numerator, or denominator distance, participants may employ other strategies when comparing two fractions. Accordingly, it would be desirable to also match the applicability of these other strategies across fraction pair types (see Faulkenberry & Pierce, 2011). For instance, fraction magnitudes can be compared on the basis of the cross-product (i.e., multiplying the numerator/denominator of one fraction with the denominator/numerator of the other fraction and comparing the results), or of the numerical difference between the numerators and denominators of the respective fractions, or participants might employ a benchmarking strategy (i.e., comparing fractions on the basis of whether one of the fractions is smaller than one

half, while the other is larger than one half). Although it would be favorable to match fraction pair types according to all of these criteria, it would not have been possible to find sufficiently matched item sets for the three fraction pair types in the present study considering all of these covariates. Hence, we limited the number of covariates to the aforementioned ones, to describe the application of the propensity score approach exemplarily.

In the first step, we generated all possible irreducible and proper single-digit and two-digit fractions by combining each single- and two-digit number with all other possible single- and two-digit numbers, with the restrictions that the first number (i.e., the numerator) was smaller than the second number (i.e., the denominator) and that the largest common divisor of both numbers was 1. This revealed that 3,003 irreducible and proper single-digit and two-digit fractions meeting these criteria exist. Hence, there are $(3{,}003 \times 3{,}002)/2 = 4{,}507{,}503$ possible combinations of fractions (excluding fraction pairs that can be created by simply reversing the order of fractions). In the next step, we removed all fraction pairs with the same numerators and denominators, leaving 4,370,958 fraction pairs. Then we categorized the remaining fraction pairs according to the congruity of the separate comparisons of the magnitudes of the numerators and denominators with the comparison of the overall magnitude of the whole fraction. The set contained 2,219,805 CC fraction pairs, 1,435,302 CI fraction pairs, and 715,851 II fraction pairs.

These were too many fraction pairs to allow us to run the matching procedure. Hence, in the next step we reduced the fraction pairs by preselecting specific fraction pairs. To do so, we first computed the covariates (i.e., the overall fraction distance, numerator distance, and denominator distance). Then we divided all fraction pairs into 443,634 groups based on the three covariates, by rounding the fraction distances to hundredths and the numerator and denominator distances to ones and assigning each fraction pair into a particular group on the basis of the specific combination of values for the three covariates. In the next step, we removed all groups from the item sets that did not contain at least one fraction pair from all three fraction pair types. This reduced the number of fraction pairs to 128,423, which we then used in the matching procedure (21,661 CC fraction pairs, 63,281 CI fraction pairs, and 43,481 II fraction pairs).

Matching

Finally, we created item sets comprising 30 items of each fraction pair type and matched them regarding fraction distance, numerator distance, and denominator distance. To do so, we drew 30 fraction pairs randomly from the set of II pairs and tried to find 30 fraction pairs from the other two pair types until the following constraints to our item set were met: The overall fraction distance should not differ by more than 0.01 between the fraction pair types, and the maximum difference allowed for both the numerator and denominator distances was set to 0.5. To find matched fraction pairs, we called the matchit function from the R package MatchIt twice. In the matchit function, we specified the covariates, overall fraction distance, numerator distance, and denominator distance. Moreover, for matching we used the nearest-neighbor method, and propensity scores as a distance measure (i.e., the default distance measure). Using this procedure, we found reasonable well-matched fraction pairs (see Table 1). Thus, we showed that propensity score matching can be applied to construct matched items sets for numerical stimuli. However, the approach is not restricted to numerical stimuli, but can also be applied for matching sets of target words according to several covariates, as is necessary in reading research. This will be exemplified in the next section.

Constructing matched word sets

Regarding content domains, propensity score matching is of course not restricted to numerical cognition research. Instead, it may be applied whenever stimulus sets have to be matched on the parameter values of several covariates—as is often necessary in reading research. For instance, Paizi, Zoccolotti, and Burani (2010) investigated the interaction between word frequency and stress dominance. The word frequency effect indicates that the reading times for words decrease as their frequencies in a given language increase (e.g., "about" is more frequent than "above" and thus is read faster; see, e.g., Brysbaert et al., 2011). *Word stress* denotes the emphasis that is given to the syllables of a word. It is typically indicated by increased loudness and vowel length. In Italian, the position of stress varies for three- or more-than-three-syllable words. The stress can be either dominant, in the case of words that are stressed on the penultimate syllable, or nondominant, in the case of words that are stressed on the antepenultimate syllable. Like for the word frequency effect, stress dominance influences the reading of words: Words with dominant stress are

Table 1 Mean fraction distances, numerator distances, denominator distances, and numbers of digits in the numerators and denominators of the final item set

	Fraction Pair Type		
Covariate	CC	CI	II
Overall fraction distance	0.10	0.09	0.10
Numerator distance	5.27	5.20	5.33
Denominator distance	16.30	16.47	16.73

CC = congruent numerator and denominator, CI = congruent numerator and incongruent denominator, II = incongruent numerator and denominator

read faster and more accurately than words with nondominant stress (Colombo, 1992). However, several other covariates influence the reading of words.

Therefore, Paizi, Zoccolotti, and Burani (2010) created four matched item sets for the four conditions of their study (high frequency–dominant stress, high frequency–nondominant stress, low frequency–dominant stress, and low frequency–nondominant stress), considering the following covariates: subjective age of acquisition (the age at which raters indicate they first learned a word in either spoken or written form; Juhasz, 2005), rated familiarity (the rated frequency of a word in daily life; Barca, Burani, & Arduino, 2002), imageability (the ease and rapidity with which a word evokes a mental image), orthographic neighborhood size (i.e., the number of words that differ by one letter from the target word), length (in letters and syllables), bigram frequency (the frequency of two letters occurring in sequence), orthographic complexity (the complexity of translating written words into the correct sequence of phonems; see Burani, Barca, & Ellis, 2006), and initial phoneme. Most of these variables (except orthographic complexity) are available via the LEXVAR database (available online at www.istc.cnr.it/material/database/; Barca et al., 2002). The database provides parameters for both children and adults. As in Paizi, Zoccolotti, and Burani's study, we used the values for children (Marconi, Ott, Pesenti, Ratti, & Tavella, 1993). In the following section, we outline how four matched item sets based on the above-mentioned covariates can be created by relying on the propensity-score-matching method.

Matching

In contrast to generating items for fraction magnitude comparison, we did not have to generate the population of items and calculate the covariates, because they could be downloaded in an Excel file from the LEXVAR database. Hence, in the first step, we divided the set of words available in the database into four categories, according to word frequency and stress dominance. Thus, we subdivided the words into low-frequency and high-frequency words on the basis of a median split, and also categorized them into stress-dominant and stress-nondominant words. We determined the numbers of words falling in the crossed categories. There were 274 high-frequency, stress-dominant words; 30 high-frequency, stress-nondominant words; 256 low-frequency, stress-dominant words; and 66 low-frequency, stress-nondominant words. Thus, the number of words was smallest for the high-frequency, stress-nondominant word category, and hence we used the parameter values of this category as the basis for selecting words for the other categories.

As with the fraction comparison task, we drew a random subset of words from the category of high-frequency, stress-nondominant words containing 20 words (for an implementation of the procedure using R, please see Appendix B). We did not use the maximal number of words (i.e., 30) because using a smaller subset increased the likelihood of finding well-matched word categories. Then we ran the matchit function from the R package MatchIt three times. In the matchit function, the high-frequency, stress-nondominant word category always served as the treatment group, whereas one of the other categories served as the control group. Moreover, we included the following covariates: subjective age of acquisition, rated familiarity, imageability, orthographic neighborhood size, bigram frequency, and length according to numbers of letters and syllables. We did not include initial phoneme as a covariate because the LEXVAR database has different classification schemes for initial phonemes, and we were not sure which of them was used by Paizi, Zoccolotti, and Burani (2010). Furthermore, in contrast to the fraction magnitude comparison task, we did not try to optimize the matching by rerunning the matchit functions until the difference between the covariate means fell below a specific threshold, because the word categories were matched well according to the covariates after running the matchit function for each item category once (see Table 2). Accordingly, we observed that propensity score matching can also be used to construct matched item sets for word stimuli.

Conclusion

In the present article, we showed how to apply the propensity-score-matching method to construct matched item sets for numerical cognition and reading research. Using such procedures, we were able to successfully create matched stimulus sets for a symbolic fraction magnitude comparison task as well as a study on word reading. Therein, we adhered to the following steps. When constructing matched stimulus sets for a symbolic fraction magnitude comparison task, we decided on the population of stimuli under investigation and determined the possible covariates. Afterward, we divided the stimuli into two sets (comparable to a treatment and a control group). The one set (i.e., the treatment group) served as the basis for which we tried to find a matched other set of items (i.e., the control group) considering the relevant covariates. Finally, we used the matchit function. Following this approach, we were able to generate matched item groups for fraction pairs and different categories of target words. However, it is important to note that the approach is not restricted to these two applications, but can be generalized easily in different ways regarding both the matching method used and the content domain under study.

As regards the matching method, a further application might be to construct matched stimuli for another task used in numerical cognition research, such as addition. For this task, it would be important to select problems with matched

Table 2 Mean subjective ages of acquisition, rated familiarities, imageabilities, orthographic neighborhood sizes, bigram frequencies, and lengths according to number of letters and syllables of the four word categories

	Word Category			
	High-Frequency		Low-Frequency	
Covariate	Dominant	Nondominant	Dominant	Nondominant
Age of acquisition	3.33	3.35	3.35	3.59
Rated familiarity	6.37	6.46	6.39	6.28
Imageability	5.06	5.12	5.36	5.24
Orthographic neighborhood size	0.25	0.35	0.15	0.30
Bigram frequency	10.87	10.91	10.82	10.81
Length letters	7.20	7.30	6.90	7.10
Length syllables	3.20	3.25	3.15	3.15

item parameters for addition with and without a carryover. Similar to the examples described in the present article, the first step would be to decide on the population of addition problems under study. Then the relevant covariates would have to be determined and computed (e.g., the problem size, which reflects the numerical magnitude of the involved numbers, for which it is known that difficulty increases with increases in problem size; see, e.g., Zbrodoff & Logan, 2005, for a review). The next step would be to select a random subsample of addition problems with carryover that would serve as the treatment group, before finding a matched set of addition problems without carryover in the last step. However, instead of using the nearest-neighbor method (as in the example of fractions), which resulted in reasonably well-matched covariates on average, another matching method might be more desirable—that is, exact matching. Exact matching might be preferred to match addition problems with and without carryover because it would allow for identifying matched addition problems with the exact same parameter values of the covariate(s). For instance, the problems 23 + 14 and 18 + 19 both result in a sum (i.e., problem size) of 37. However, only the latter requires the execution of a carry procedure.

Moreover, regarding the content domain, propensity score matching is of course not restricted to numerical cognition research or reading research, as described in our examples. Instead, it may be applied whenever stimulus sets have to be matched on the parameter values of several covariates. A further case of application might be research on visual perception or emotional pictures. For instance, Murphy, Hill, Ramponi, Calder, and Barnard (2010) investigated the influence of the impact of emotion on attention to negative emotional images. To do so, they matched high-impact, low-impact, and neutral images according to several covariates, including valence, arousal, distinctiveness, visual complexity, and tendency to approach or avoid. The approach using propensity scores suggested in the present study might also be applied for finding matched item sets in this context.

Finally, we will point out that (1) it may not always be possible to match item sets according to all possible covariates. Generally, it can be said that the more covariates are included in the matching procedure, the more likely it is that the item sets will not be sufficiently well-matched according to one or more of the covariates. Additionally, (2) it may even be the case that covariates cannot be matched across item sets by definition. For example, fraction comparison items can be either congruent or incongruent with respect to their respective numerical differences between the numerators and denominators; this means that the overall larger fraction might have the larger but also the smaller numerator–denominator difference [e.g., congruent: 13/42 > 3/31, and 29 (i.e., 42 – 13) > 28 (i.e., 31 – 3); incongruent: 13/28 > 5/22, but 15 < 17]. Importantly, however, items of types CI and II are by definition incongruent regarding this covariate, and thus it is impossible to find matched sets according to this covariate.[2]

Taken together, we conclude that the propensity-score-matching method can be used as a general approach for constructing matched stimuli in different domains of cognitive experimental research. This approach comes with several benefits, as compared to matching covariates by hand. Most importantly, it is less labor-intensive, and thus less time-consuming, as well as less expensive. Additionally, it avoids human errors. Therefore, propensity score matching might be a valuable tool to corroborate the validity of psychological research, because it can help ensure that the effect of interest is indeed driven by the variable being manipulated rather than by potential covariates.

Author note S.H. was supported by the Leibniz-Competition Fund, providing funding to Elise Klein (Grant No. SAW-2014-IWM-4). J.F.D. was supported by the German Research Foundation (DFG), providing funding to K.M. and Elise Klein (Grant No. MO 2525/2-1).

[2] We thank an anonymous reviewer for suggesting this comment.

Appendix A

To create the item set for the fraction magnitude comparison task, we first wrote a function in R that calculates the greatest common divisor of any two integers, a and b, using Euclid's algorithm:

```
gcd <- function(a, b){
  if(b == 0){
    return(a)
  } else {
    return(gcd(b, a %% b))
  }
}
```

Next, we generated all possible irreducible and proper single-digit and two-digit fractions using a for loop. In this for loop, we combined each single- and two-digit number with all other possible single- and two-digit number with the restrictions that the first number (i.e., the numerator) is smaller than the second number (i.e., the denominator) and that the largest common divisor of them is 1. When these criteria were met, we stored them in the data frame fractions:

```
fractions <- data.frame(id=-1,n=-1,d=-1)
cur <- 1

for (i in 1:99){ # i = numerator
  for (j in 1:99){ # j = denomintaor
    if (i < j & gcd(i,j) == 1){
      fractions[cur,] <- c(cur,i,j)
      cur <- cur + 1
    }
  }
}
```

In the next step, we generated all possible combinations of fractions and excluded fraction pairs that can be created by simply reversing the order of two fractions and fraction pairs with same numerators and denominators:

```
fraction.pairs <- apply(fractions, 1, function(f){
  cur <- cbind(rep(f[1], nrow(fractions)-1),
               rep(f[2], nrow(fractions)-1),
               rep(f[3], nrow(fractions)-1))
  rownames(cur) <- seq(1, nrow(cur))
  curFractionPairs <- cbind(cur,
     fractions[fractions$id != as.numeric(f[1]),])
  return(curFractionPairs)
})

fraction.pairs <- do.call(mapply, c(cbind, fraction.pairs))
fraction.pairs <- data.frame(fraction.pairs)
colnames(fraction.pairs) <- c("id1", "n1", "d1", "id2", "n2", "d2")
fraction.pairs <-
  fraction.pairs[fraction.pairs$id1 < fraction.pairs$id2,]
fraction.pairs <-
  fraction.pairs[fraction.pairs$n1 != fraction.pairs$n2,]
fraction.pairs <-
  fraction.pairs[fraction.pairs$d1 != fraction.pairs$d2,]
```

Then we categorized the fraction pairs according to the congruity of the separate comparisons of the magnitudes of numerators and denominators with the comparison of the overall magnitude of the whole fractions:

```
fraction.pairs$f1 <- fraction.pairs$n1/fraction.pairs$d1
fraction.pairs$f2 <- fraction.pairs$n2/fraction.pairs$d2
fraction.pairs[,"type"] <- "CC"
fraction.pairs[fraction.pairs$f1 < fraction.pairs$f2 &
               fraction.pairs$n1 < fraction.pairs$n2 &
               fraction.pairs$d1 > fraction.pairs$d2,]$type <- "CI"
fraction.pairs[fraction.pairs$f1 > fraction.pairs$f2 &
               fraction.pairs$n1 < fraction.pairs$n2 &
               fraction.pairs$d1 < fraction.pairs$d2,]$type <- "II"
```

Next, we computed the covariates (i.e., overall fraction distance, numerator distance and denominator distance):

```
fraction.pairs$distFrac <-
  abs(fraction.pairs$f1 - fraction.pairs$f2)
fraction.pairs$distNum <-
  abs(fraction.pairs$n1 - fraction.pairs$n2)
fraction.pairs$distDenom <-
  abs(fraction.pairs$d1 - fraction.pairs$d2)
```

We reduced the fraction pairs by pre-selecting specific fraction pairs. Then, we divided all fraction pairs in 443,634 groups based on the three covariates, by rounding the fraction distance to hundredths and the numerator and denominator distances to ones, and assigning each fraction pair into a particular group based on the specific combination of values for the three covariates. In the next step, we selected only the groups that contained at least one fraction pair from the three fraction pair types:

```
fraction.pairs$distFracGroup <- round(fraction.pairs$distFrac, 2)
fraction.pairs$distNumGroup <- round(fraction.pairs$distNum, 0)
fraction.pairs$distDenomGroup <- round(fraction.pairs$distDenom, 0)

fraction.pairs$combinedGroup <-
  paste(fraction.pairs$distFracGroup,
        fraction.pairs$distNumGroup,
        fraction.pairs$distDenomGroup,sep="_")

fractionsGroupAllTyps <-
  tapply(fraction.pairs$type,
         fraction.pairs$combinedGroup,
         function(type) {
           any(type == "CC") &
           any(type == "CI") &
           any(type == "II")})

fraction.pairs <- subset(
                fraction.pairs,
                combinedGroup %in%
                names(fractionsGroupAllTyps[fractionsGroupAllTyps]))
```

Finally, we created an item set comprising 30 items of each fraction pair type and matched them regarding fraction distance, numerator distance, and denominator distance. Therefore, we first stored the respective fraction pair types in three different data frames:

```
fraction.pairs.CC <- fraction.pairs[fraction.pairs$type == "CC",]
fraction.pairs.CI <- fraction.pairs[fraction.pairs$type == "CI",]
fraction.pairs.II <- fraction.pairs[fraction.pairs$type == "II",]
```

Then, we randomly drew 30 fraction pairs from the data frame fraction.pairs.II and tried to find 30 fraction pairs from the other data frames, whereby we set the following constraints to our item set: overall fraction distance should not differ by more than 0.01 between the fraction pair types and the maximum difference allowed for numerator distance and denominator distance was set to 0.5. The initial 30 fraction pairs from the data frame fraction.pairs.II might be biased such that our criteria might not be met. Therefore, we ran a while loop until our criteria were met:

```
  distMatch <- 10
  distNumMatch <- 10
  distDenomMatch <- 10
  nDigitsNumMatch <- 10
  nDigitsDenomMatch <- 10

  while (distMatch > 0.01 |
         distNumMatch > 0.5 |
         distDenomMatch > 0.5)
  {

    fraction.items.II <-
      fraction.pairs.II[sample(1:nrow(fraction.pairs.II),
  30,replace=FALSE),]
    fractions.match1 <-
      rbind(fraction.items.II,fraction.pairs.CI)
    fractions.match1$type.cat <-
      ifelse(fractions.match1$type == "CI",0,1)
    fractions.match2 <-
      rbind(fraction.items.II,fraction.pairs.CC)
    fractions.match2$type.cat <-
      ifelse(fractions.match2$type == "CC",0,1)

require(MatchIt)
m.out1 <- matchit(type.cat ~ distFrac + distNum + distDenom ,
                  data = fractions.match1, method = "nearest")
distMatch <- abs(summary(m.out1)$sum.matched[2,4])
distNumMatch <- abs(summary(m.out1)$sum.matched[3,4])
distDenomMatch <- abs(summary(m.out1)$sum.matched[4,4])

if (!(distMatch > 0.01 |
  distNumMatch > 0.5 |
  distDenomMatch > 0.5))
{
  m.out2 <- matchit(type.cat ~ distFrac + distNum + distDenom ,
                    data = fractions.match2, method = "nearest")
  distMatch <- abs(summary(m.out2)$sum.matched[2,4])
  distNumMatch <- abs(summary(m.out2)$sum.matched[3,4])
  distDenomMatch <- abs(summary(m.out2)$sum.matched[4,4])

  print(paste(distMatch,distNumMatch,distDenomMatch))
}
}
```

In this R code, we first set our criteria variables to 10 (i.e., values that do not meet the criteria, such that the while loop is entered). In the while loop, we first drew a random subset of 30 items of the data frame fraction.items.II. Then we merged this data frame with the data frame for fraction pairs CC and fraction pairs CI, and created a column that was 1 for fraction pairs of type II and otherwise 0. In the next step, we called the matchit function from the R package MatchIt twice. In the matchit function, we specified the covariates, overall fraction distance, numerator distance, and denominator distance. Moreover, for matching we used the nearest-neighbor method and propensity scores as a distance measure (i.e., the default distance measure). Afterward, we stored the maximum of the achieved mean differences in the criteria variables. Finally, the matched item sets could be retrieved by calling the function match.data:

```
df1 <- match.data(m.out1)
df2 <- match.data(m.out2)
```

Appendix B

To create matched word categories, we first read in the information about Italian words form the database LEXVAR.XLS (available at www.istc.cnr.it/grouppage/lexvar). To be able to read the Excel file, we had to rename the header line. Moreover, we removed columns that we did not need for creating matched word categories. Finally, we saved the modified Excel file as a tab-separated text file and read it into the workspace of R:

```
dfWords <-
  read.table("wordsMatchingItemSet.txt", header=TRUE, sep="\t")
```

Then, we categorized the variable containing the frequency counts for children applying a median split:

```
dfWords$frequencyCat <-
  ifelse(dfWords$totalFrequencyChildren <=
           median(dfWords$totalFrequencyChildren), "low", "high")
```

In the next step, we created the four item categories: high-frequency, stress-dominant words (Category 1), high-frequency, stress-nondominant words (Category 2), low-frequency, stress-dominant words (Category 3), and low-frequency, stress-nondominant words (Category 4).

```
getCondition <- function(frequencyCat, stress){
  if(is.na(frequencyCat) | is.na(stress))
    return(NA)
  if(frequencyCat == "high" & stress == "p")
    return("group1")
  if(frequencyCat == "high" & stress == "ap")
    return("group2")
  if(frequencyCat == "low" & stress == "p")
    return("group3")
  if(frequencyCat == "low" & stress == "ap")
    return("group4")
}
  dfWords$group <- apply(dfWords, 1,
            function(x){getCondition(x["frequencyCat"],x["Stress"])})
```

Finally, we ran the matching procedure. Therefore, we first drew 20 words from the smallest category of high-frequency, stress-nondominant words (i.e., Category 2). Then, we combined these words with the other categories into three separate data frames and added an additional column with a binary label indicating whether a particular word belonged to Category 2 or to one of the other categories. In the last step, we called the matchit function from the R package MatchIt three times. In the matchit function, we specified the covariates, subjective age of acquisition, rated familiarity, imageability, orthographic neighborhood size, bigram frequency, and length according to number of letters and syllables. Moreover, for matching we again used the nearest neighbor method and propensity scores as a distance measure (i.e., the default distance measure):

```
words.match1 <-
  rbind(dfGroup2Sample,dfWords[dfWords$group == "group1",])
words.match3 <-
  rbind(dfGroup2Sample,dfWords[dfWords$group == "group3",])
words.match4 <-
  rbind(dfGroup2Sample,dfWords[dfWords$group == "group4",])
words.match1$type.cat <- ifelse(words.match1$group == "group1",0,1)
words.match3$type.cat <- ifelse(words.match3$group == "group3",0,1)
words.match4$type.cat <- ifelse(words.match4$group == "group4",0,1)

m.out1 <- matchit(type.cat ~ ageOfAcquisition + familiarity +
imageability + orthographicNeighborhood + lengthSyllables +
lengthLetters + bigramFrequency,
data = words.match1, method = "nearest")
m.out2 <- matchit(type.cat ~ ageOfAcquisition + familiarity +
imageability + orthographicNeighborhood + lengthSyllables +
lengthLetters + bigramFrequency,
data = words.match3, method = "nearest")
m.out3 <- matchit(type.cat ~ ageOfAcquisition + familiarity +
     imageability + orthographicNeighborhood + lengthSyllables +
     lengthLetters + bigramFrequency,
data = words.match4, method = "nearest")
```

Finally, the matched item sets can be retrieved by calling the function match.data:

```
df1 <- match.data(m.out1)
df2 <- match.data(m.out2)
df3 <- match.data(m.out3)
```

References

Adelson, J. L. (2013). Educational research with real-world data: Reducing selection bias with propensity scores. *Practical Assessment, Research & Evaluation, 18*(2), 1–11.

Barca, L., Burani, C., & Arduino, L. S. (2002). Word naming times and psycholinguistic norms for Italian nouns. *Behavior Research Methods, Instruments, & Computers, 34,* 424–434. doi:10.3758/BF03195471

Bonato, M., Fabbri, S., Umiltà, C., & Zorzi, M. (2007). The mental representation of numerical fractions: Real or integer? *Journal of Experimental Psychology: Human Perception and Performance, 33,* 1410–1419. doi:10.1037/0096-1523.33.6.1410

Brysbaert, M., Buchmeier, M., Conrad, M., Jacobs, A. M., Bölte, J., & Böhl, A. (2011). The word frequency effect. *Experimental Psychology, 58,* 412–424. doi:10.1027/1618-3169/a000123

Burani, C., Barca, L., & Ellis, A. W. (2006). Orthographic complexity and word naming in Italian: Some words are more transparent than others. *Psychonomic Bulletin & Review, 13,* 346–352. doi:10.3758/BF03193855

Cochran, W. G. (1968). The effectiveness of adjustment by subclassification in removing bias in observational studies. *Biometrics, 24,* 295–313. doi:10.2307/2528036

Colombo, L. (1992). Lexical stress effect and its interaction with frequency in word pronunciation. *Journal of Experimental Psychology: Human Perception and Performance, 18,* 987–1003. doi:10.1037/0096-1523.18.4.987

Dewolf, M., Grounds, M. A., Bassok, M., & Holyoak, K. J. (2014). Magnitude comparison with different types of rational numbers. *Journal of Experimental Psychology: Human Perception and Performance, 40,* 71–82. doi:10.1037/a0032916

DeWolf, M., & Vosniadou, S. (2015). The representation of fraction magnitudes and the whole number bias reconsidered. *Learning and Instruction, 37,* 39–49. doi:10.1016/j.learninstruc.2014.07.002

Faulkenberry, T. J., & Pierce, B. H. (2011). Mental representations in fraction comparison. *Experimental Psychology, 58,* 480–489.

Ganor-Stern, D., Karasik-Rivkin, I., & Tzelgov, J. (2011). Holistic representation of unit fractions. *Experimental Psychology, 58,* 201–206.

Gu, X. S., & Rosenbaum, P. R. (1993). Comparison of multivariate matching methods: Structures, distances, and algorithms. *Journal of Computational and Graphical Statistics, 2,* 405–420. doi:10.2307/1390693

Gum, P. A., Thamilarasan, M., Watanabe, J., Blackstone, E. H., & Lauer, M. S. (2001). Aspirin use and all-cause mortality among patients being evaluated for known or suspected coronary artery disease: A propensity analysis. *JAMA, 286,* 1187–1194. doi:10.1001/jama.286.10.1187

Guo, S., & Fraser, M. W. (2010). *Propensity score analysis: Statistical methods and applications. Advanced quantitative techniques in the social sciences series (Vol. 11)*. Thousand Oaks, CA: Sage.

Ho, D. E., Imai, K., King, G., & Stuart, E. A. (2011). MatchIt: Nonparametric preprocessing for parametric causal inference. *Journal of Statistical Software, 42*(8), 1–28.

Huber, S., Moeller, K., & Nuerk, H. C. (2014). Adaptive processing of fractions—Evidence from eye-tracking. *Acta Psychologica, 148,* 37–48. doi:10.1016/j.actpsy.2013.12.010

Ischebeck, A., Schocke, M., & Delazer, M. (2009). The processing and representation of fractions within the brain: An fMRI investigation. *NeuroImage, 47,* 403–413.

Ischebeck, A., Weilharter, M., & Körner, C. (2016). Eye movements reflect and shape strategies in fraction comparison. *Quarterly Journal of Experimental Psychology, 69,* 713–727. doi:10.1080/17470218.2015.1046464

Jones, A. S., D'Agostino, R. B., Gondolf, E. W., & Heckert, A. (2004). Assessing the effect of batterer program completion on reassault using propensity scores. *Journal of Interpersonal Violence, 19,* 1002–1020. doi:10.1177/0886260504268005

Juhasz, B. J. (2005). Age-of-acquisition effects in word and picture identification. *Psychological Bulletin, 131,* 684–712. doi:10.1037/0033-2909.131.5.684

Marconi, L., Ott, M., Pesenti, E., Ratti, D., & Tavella, M. (1993). *Lessico elementare: Dati statistici sull'italiano letto e scritto dai bambini delle elementari*. Bologna, Italy: Zanichelli.

Meert, G., Grégoire, J., & Noël, M.-P. (2009). Rational numbers: Componential versus holistic representation of fractions in a magnitude comparison task. *Quarterly Journal of Experimental Psychology, 62,* 1598–1616. doi:10.1080/17470210802511162

Meert, G., Grégoire, J., & Noël, M.-P. (2010). Comparing 5/7 and 2/9: Adults can do it by accessing the magnitude of the whole fractions. *Acta Psychologica, 135,* 284–292.

Moyer, R. S., & Landauer, T. K. (1967). Time required for judgements of numerical inequality. *Nature, 215,* 1519–1520. doi:10.1038/2151519a0

Murphy, F. C., Hill, E. L., Ramponi, C., Calder, A. J., & Barnard, P. J. (2010). Paying attention to emotional images with impact. *Emotion, 10,* 605–614. doi:10.1037/a0019681

Nuerk, H.-C., Weger, U., & Willmes, K. (2001). Decade breaks in the mental number line? Putting the tens and units back in different bins. *Cognition, 82,* B25–B33. doi:10.1016/S0010-0277(01)00142-1

Obersteiner, A., Van Dooren, W., Van Hoof, J., & Verschaffel, L. (2013). The natural number bias and magnitude representation in fraction comparison by expert mathematicians. *Learning and Instruction, 28,* 64–72. doi:10.1016/j.learninstruc.2013.05.003

Paizi, D., Zoccolotti, P., & Burani, C. (2010). Lexical stress assignment in Italian developmental dyslexia. *Reading and Writing, 24,* 443–461. doi:10.1007/s11145-010-9236-0

R Development Core Team. (2015). R: A Language and Environment for Statistical Computing. Vienna, Austria: R Foundation for Statistical Computing. Retrieved from www.r-project.org/

Randolph, J. J., Falbe, K., Manuel, A. K., & Balloun, J. L. (2014). A step-by-step guide to propensity score matching in R. *Practical Assessment, Research & Evaluation, 19*(18), 1–6.

Rosenbaum, P. R., & Rubin, D. B. (1983). The central role of the propensity score in observational studies for causal effects. *Biometrika, 70,* 41–55. doi:10.2307/2335942

Rubin, D. B. (1973). Matching to remove bias in observational studies. *Biometrics, 29,* 159–183. doi:10.2307/2529684

Schneider, M., & Siegler, R. S. (2010). Representations of the magnitudes of fractions. *Journal of Experimental Psychology: Human Perception and Performance, 36,* 1227–1238.

Siegler, R. S., Fazio, L. K., Bailey, D. H., & Zhou, X. (2013). Fractions: The new frontier for theories of numerical development. *Trends in Cognitive Sciences, 17,* 13–19. doi:10.1016/j.tics.2012.11.004

Smith, H. L. (1997). Matching with multiple controls to estimate treatment effects in observational studies. *Sociological Methodology, 27,* 325–353. doi:10.1111/1467-9531.271030

Sprute, L., & Temple, E. (2011). Representations of fractions: Evidence for accessing the whole magnitude in adults. *Mind, Brain, and Education, 5,* 42–47.

Stuart, E. A. (2010). Matching methods for causal inference: A review and a look forward. *Statistical Science : A Review Journal of the Institute of Mathematical Statistics, 25,* 1–21. doi:10.1214/09-STS313

Zbrodoff, N. J., & Logan, G. D. (2005). What everyone finds: The problem-size effect. In *Handbook of mathematical cognition* (pp. 331–345). New York, NY: Psychology Press.