Do gridlines on a line graph correct for compatibility effects in quantitative estimates of change?

Misha Ash

5/3/2018

introduction

This experiment aims to elicit a cognitive bias that derives from a mental metaphor that maps the dimension of emotional valence onto a vertical spatial dimension (e.g., see Dolscheid & Casasanto, 2014 and 2015). A mental metaphor like this may be deployed by representing the values along a non-spatial dimension ranging from negative emotions like sadness to positive emotions like happiness in terms of positions along a vertical spatial dimension ranging from low to high. It is expected that quantitative estimates made from line graphs will be affected by this bias such that incompatible graphs—that is, graphs representing a decrease in a positively valenced phenomenon like happiness or an increase in a negatively valenced phenomenon like sadness—will inflate error rates relative to compatible graphs.

- *null hypothesis 1*: The mean error will not differ between compatible and incompatible conditions.
- *alternative hypothesis 1*: The mean error will be lower for compatible than incompatible conditions.

As reported by Heer and Bostock (2010), gridlines can improve accuracy when not spaced too tightly. To the extent that accuracy is affected by compatibility effects, then, this bias may be mitigated by gridlines.

- *null hypothesis 2*: The mean error will not differ between conditions with and without an additional gridline.
- *alternative hypothesis 2*: The mean error will be lower in the condition with additional gridlines.

design

The study is organized around a 2x2x2x3 design:

valence: positive (happiness), negative (sadness) direction: increasing, decreasing *grid lines: 0 (labels at 0 and 100) and 1 (mid-line added at 50)* city name: Oswa, Escor, Aflos (control variable)

City names are used to control for unintended valence effects. People vary in the valence and strength of their associations with concepts represented by words. For example, the name "Ludwig" might trigger strong positive feelings for someone who has fond memories of positive experiences with someone named Ludwig, which may affect how the happiness or sadness of Ludwig is interpreted. Alternatively, someone who has never known a Ludwig will likely be much more neutral in their affective response to this name, or their affective response may be colored by associations activated by resemblence with similar sounding words, such as "wig," or by their memories of the music of Ludwig van Beethoven or the arguments of Ludwig Wittgenstein. To mitigate and control for world knowledge effects on valence, the graphs will be presented as being about the happiness or sadness score of a fictional city. Three city names were generated on the Fantasy Name Generators website (each two syllables):

- *Oswa*
- *Escor*
- *Aflos*

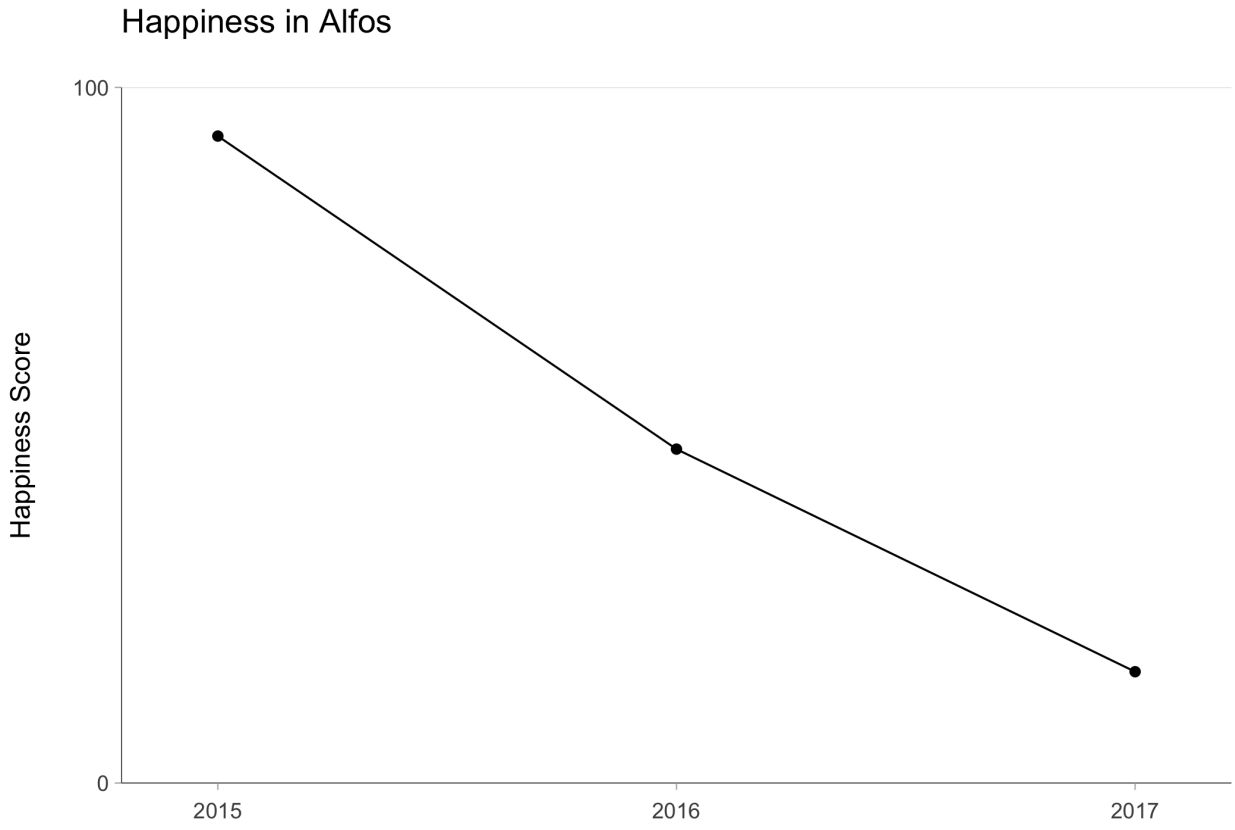
A within-subjects designs can increase power for a given sample size. However, the comparability (and power) gained would require either using the same values for stimuli across conditions, which is prone to order effects and would require extensive counter-balancing resulting in the need for a larger sample, or different values across conditions that vary between participants to converge on comparable means, which would require a

larger stimulus set and a larger sample. Given the exploratory nature of the current experiment, a one-shot, between-subjects design is used. The study will consist of one HIT with one of 8 possible conditions, randomly assigned and counter-balanced.

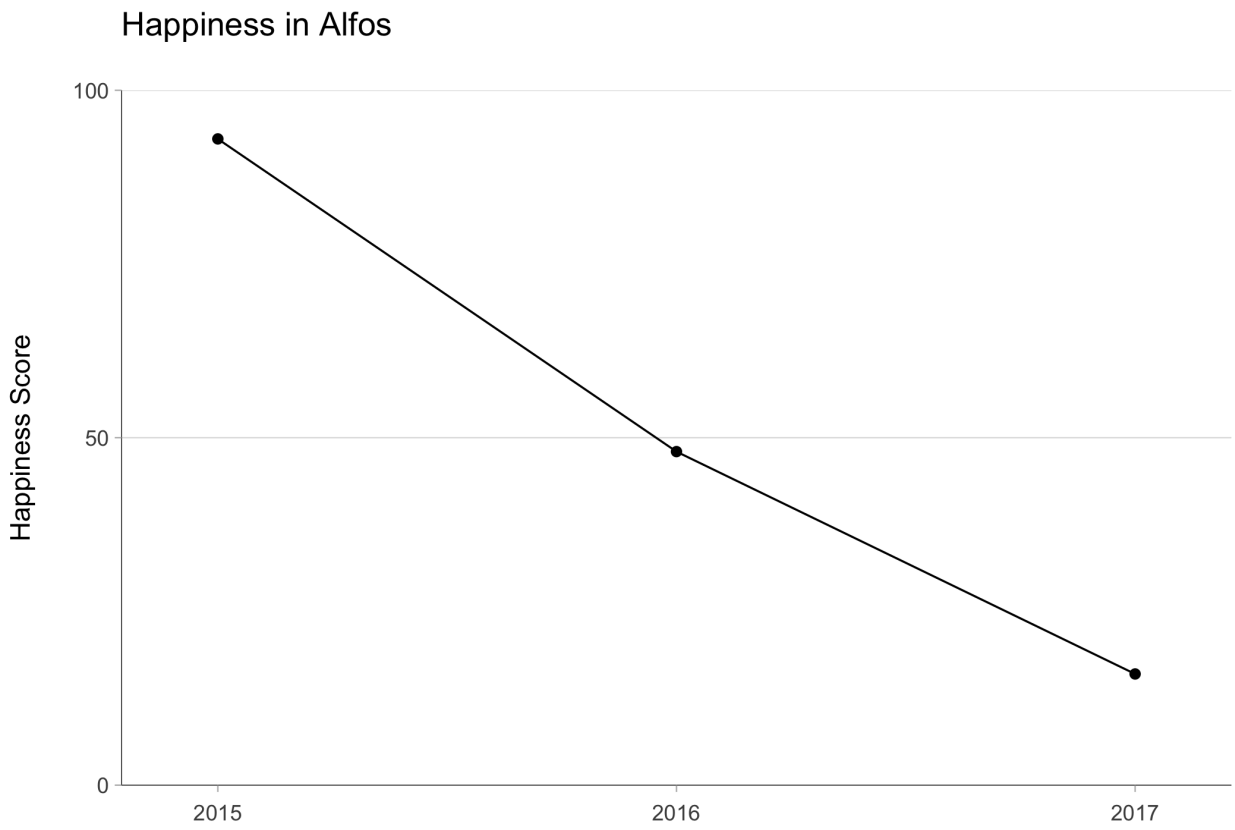
A set of 24 stimuli graphs is generated by crossing these factors (8 x 3 for each city):

Each visualization displays the same three data points in ascending or descending order. The data value triplet used across the graphs is randomly sampled from its respective third of the 0 - 100 range ±5 (to keep the value from being directly on a grid line and force estimation). The method used by the base R sampling function is analogous to Cleveland and McGill's use of a "uniform random number generator" (1984, p. 539).

example of stimulus graph (incompatible, no gridline):



example of stimulus graph (incompatible, with gridline):



method

The radomization logic of the stimulus conditions was implemented in Qualtrics (see survey flow in repository materials for more information). After being presented with the stimulus graph for 20 seconds, the graph disappears and an estimate is made by using a slider.

The framing in the title and y-axis labels of the graph are inten

results

```
tidy_qualtrics_batch_1 <- qualtrics_batch_1 %>%

  filter(Finished == 1,
         .[,18] == 2 | .[,71] == 2, # min value qualification for increasing (QID5, QID14)
         .[,19] == 4 | .[,72] == 4, # max value qualification for increasing (QID8, QID15)
         !is.na(.[,123]), # direction value not missing
         !is.na(.[,124]), # valence value not missing
         !is.na(.[,125]), # gridline value not missing
         !is.na(.[,126])) %>% # city value not missing

  select("Response ID",
         "User Language",
         "Duration (in seconds)",
```

```
          "Indicate how much the Happiness Score changed between 2015 and 2017. - Happiness Score Change
          "Indicate how much the Sadness Score changed between 2015 and 2017. - Happiness Score Change",
          "direction",
          "valence",
          "gridline",
          "city") %>%

  data.table::setnames(old = c("Response ID", "User Language", "Duration (in seconds)", "Indicate how mu
                       new = c("id", "language", "duration_sec", "happy_change", "sad_change")) %>%

  gather(key = score, value = est_change, happy_change, sad_change) %>%
  filter(!is.na(est_change)) %>%
  select(-score) %>%

  # add compatibility condition
  mutate(compatibility =
           ifelse((.$direction == "increasing" & .$valence == "happiness" |
             .$direction == "decreasing" & .$valence == "sadness"), 1, 0)) %>%

  # calculate numeric estimate errors
  mutate(act_change = ifelse(.$direction == "decreasing", -77, 77)) %>%
  mutate(diff_change = as.double(est_change) - act_change) %>%

  # calculate absolute value estimate errors
  mutate(abs_change = 77) %>%
  mutate(error = abs((abs(as.double(est_change)) - abs_change)))

#fix city names
tidy_qualtrics_batch_1 <- mutate(tidy_qualtrics_batch_1,
                                 city = str_replace(tidy_qualtrics_batch_1$city, "Alfos", "Aflos"))
```

```
error_se <- tidy_qualtrics_batch_1 %>%
  dplyr::select(error, compatibility, gridline, direction) %>%
  data.table()


tbl_err_compat_grid_dir <- error_se[, list(mean = mean(error),
                                sd = sd(error),
                                se = sqrt(var(error)/length(error)),
                                se_test = sd(error)/length(error),
                                # 95% confidence intervals of the mean (&plusmn;2se)
                                ci_min = mean(error) - 2*(sd(error)/length(error)),
                                ci_max = mean(error) + 2*(sd(error)/length(error))),
                            by=list(compatibility, gridline)]
```

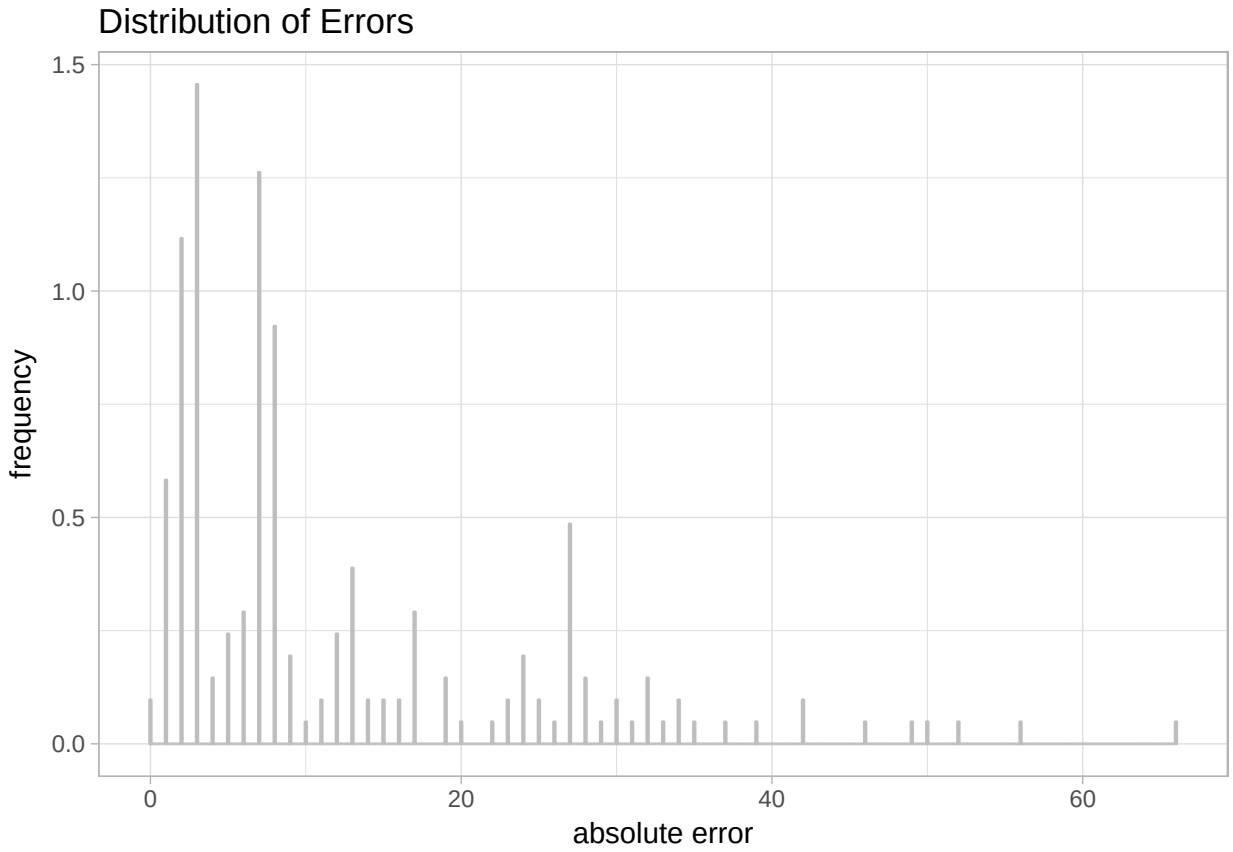
exploratory glances

Mean error does not differ by city, so this control can be set aside.

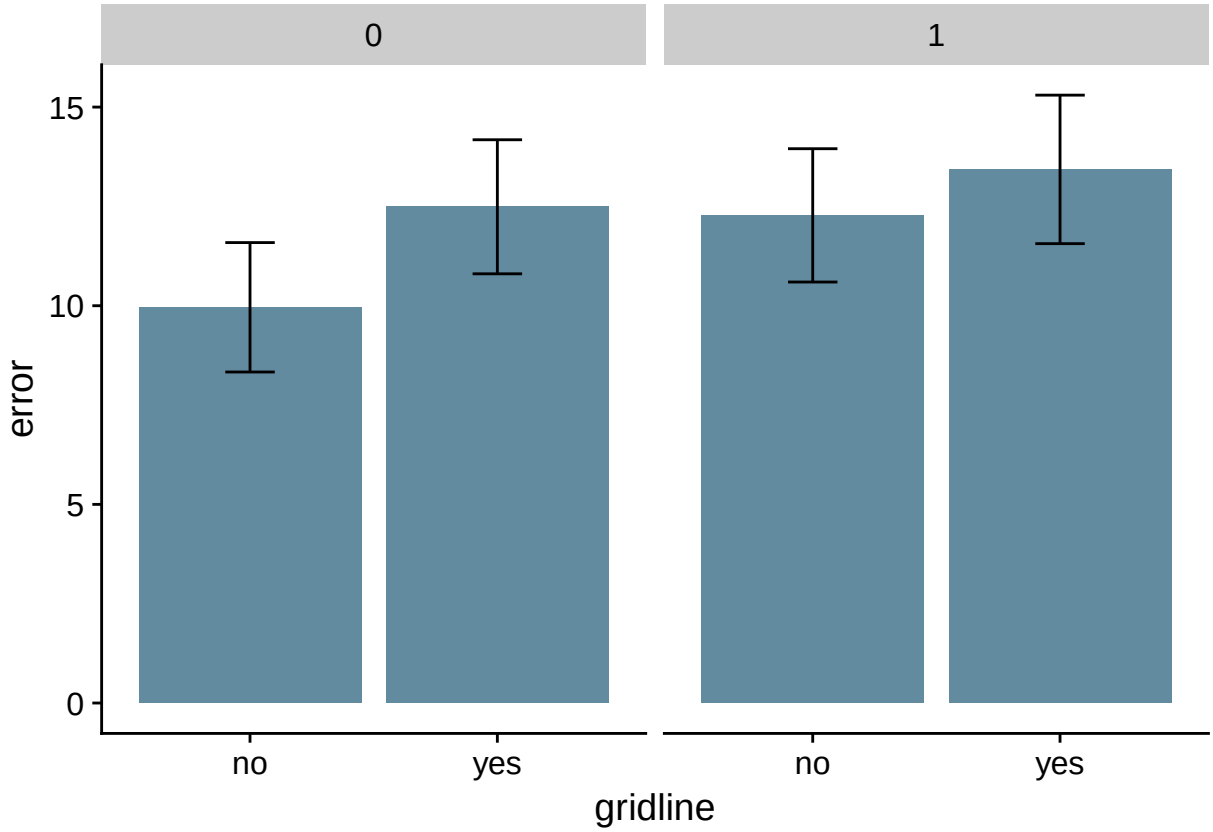
Distribution of errors

```
ggplot(tidy_qualtrics_batch_1, aes(error)) +
  geom_histogram(aes(y = ..density..), binwidth = .1, color="grey", fill="light blue") +
  theme_light() +
```

```
  ggtitle('Distribution of Errors') +
  xlab('absolute error') +
  ylab('frequency')
```




```
ggplot(tidy_qualtrics_batch_1, aes(gridline, error)) +
  stat_summary(fun.y = "mean", geom = "bar", fill =  "#628B9F") +
  stat_summary(fun.data = mean_se, geom = "errorbar", width = 0.2) +
  facet_wrap(~ compatibility)
```




```
#
mean_accuracy_compat_grid <- tidy_qualtrics_batch_1 %>%
  dplyr::select(error, compatibility, gridline) %>%
  dplyr::group_by(compatibility, gridline) %>%
  dplyr::summarize(mean_error = sum(error)/length(error), n = length(error))
```