

Example Application: A Text-As-Data Approach for Using Open-Ended Responses as Manipulation Checks

Jeffrey Ziegler

Introduction, Installing and Loading Package

The example comes from a study that investigated rhetorical responsiveness in the Catholic Church and the motivations for the Pope to be responsive. Participants of the online survey experiments were limited to self-identified Catholics. The survey was carried out among a nationally representative quota sample from each Brazil and Mexico ($N \approx 2,500$).

Respondents were presented with three selected news headlines on the same topic outlining recent statements made by the Pope (conflict, human rights, socio-political issues, economy, and control/religious issues). These messages were meant to represent the typical language content and phrasing used in the media regarding the Pope's statements. Respondents were randomly assigned to receive news stories pertaining to either (1) a topic that they believed is most important (the "responsive" treatment), or (2) one of the four other issue areas ("non-responsive"). Within those respondents that received "non-responsive" messages, there was an even probability of assignment to each topic.

Prior to the outcome questions, but after the textual treatment, participants were asked to recall the stories they read on the previous page in an open-ended response manipulation check. Afterward, respondents then expressed the degree to which they thought the Church is responsive, the degree to which they trusted the Church, and the degree to which they anticipated increasing their organizational participation.

As a visual reference, the distributions for the n -gram similarity measures (Jaccard and cosine) in each country are shown in Figure 1 and 2. To create these figures, we first need to download and install the library from my [GitHub](#) webpage, which researchers can do by executing the code below into their R console. All of the documentation for the functions and arguments included in the R package can be found on the [GitHub](#) webpage.

```
1 # install and load package
2 devtools::install_github('jeffreyziegler/openEnded', force=T, build_
  vignettes=T)
3 library(openEnded)
```

Next, users can load in their data and specify which vector within their dataframe contains the prompts and which vector contains the responses to calculate their similarity measures. I import the data for the survey experiments in Brazil and Mexico, which are also located on the [GitHub](#) webpage. The shortcut for the link to the data is provided below.

```

1
2 # load data from GitHub
3 replication_complete.cases <- read.csv("http://bit.ly/repData",

```

The vector of responses to the open-ended manipulation check are stored in `replication_complete.cases$validityCheck` and the treatments that respondents read are stored in `replication_complete.cases$textViewed`. To create our various n -gram similarity measures, such as the Jaccard and the cosine of the angle between the vectors, we can execute the function `similarityMeasures` as seen below. We continue to assign $n=3$ as we did in the manuscript.

```

1 replication_complete.cases$Concordant <- relevel(as.factor(replication_complete.cases$Concordant),
2                                               ref = "0")
3 replication_complete.cases$AttendanceBin <- relevel(as.factor(replication_complete.cases$AttendanceBin),
4                                                  ref = "Never/yearly")
5
6 # create jaccard and cosine similarity measures
7 replication_complete.cases <- similarityMeasures(dataframe=replication_complete.cases,
8                                                  similarity_measures_to_calculate=c("jaccard", "cosine", "jw", "dl"),
9                                                  prompt="textViewed", response="validityCheck", ngrams=3)

```

With our similarity measures in hand, we can plot the distribution of all respondents with the function `plotMeasures`. Figure 1, specifically, shows the plotted output from the code below. The default `plotMeasures` does not currently include the ability to label select responses, this feature will be made available in future versions of the package.

```

1 # plot Jaccard distances for just Brazil
2 brazil_rows <- which(replication_complete.cases$Country=="Brazil")
3 plotSimilarity(dataframe=replication_complete.cases[brazil_rows,],
4               measure="jaccardDist", xlab="Jaccard Similarity")

```

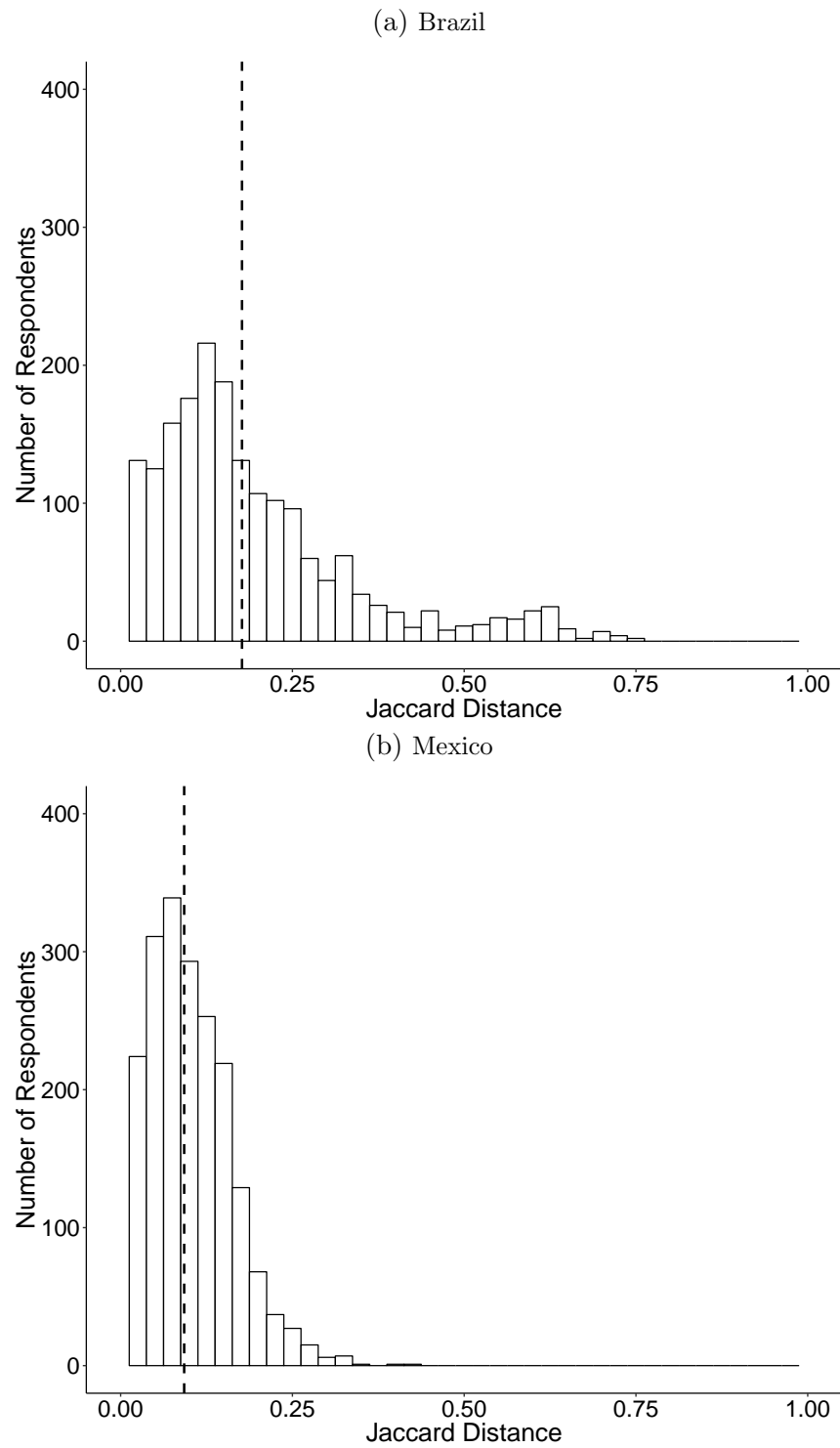
The distributions, especially in Mexico, are more highly skewed to the left than the data presented in the manuscript from Kane (2019), which means that more respondents will be down-weighted with lower levels of k . Nevertheless, the Jaccard and cosine measures are high correlated, as seen in Figure 3, which can be created with the function `plotSimilarityCorr`.

```

1 # plot correlation between similarity measures
2 plotSimilarityCorr(dataframe=replication_complete.cases,
3                   measures=c("cosineDist", "jaccardDist",
4                             "jwDist", "dlDist"),
5                   labels=c("Cosine", "Jaccard", "Jaro",
6                             "Damerau-Levenshtein"))

```

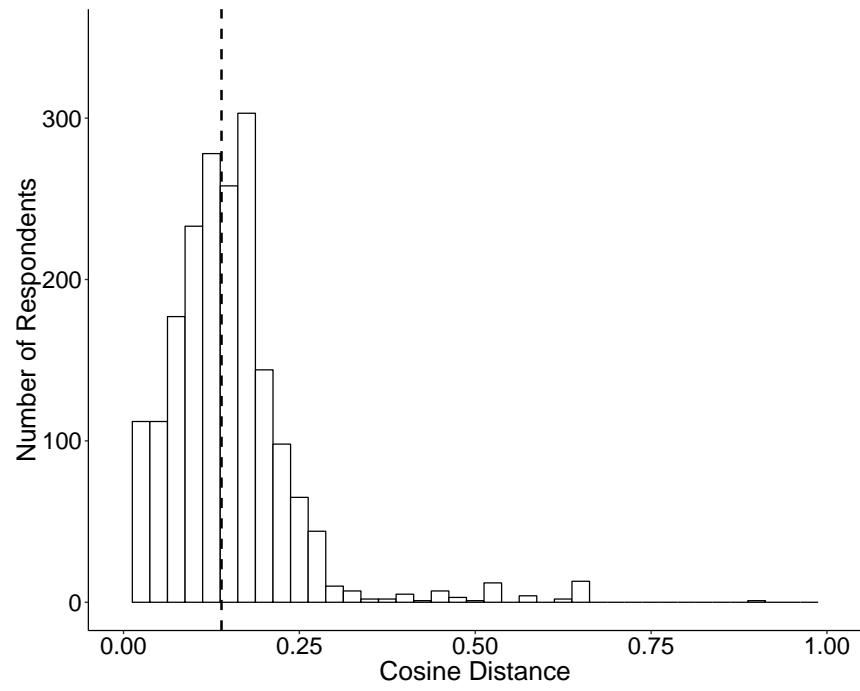
Figure 1: Distribution of raw Jaccard similarity measures for respondents in Brazil and Mexico.



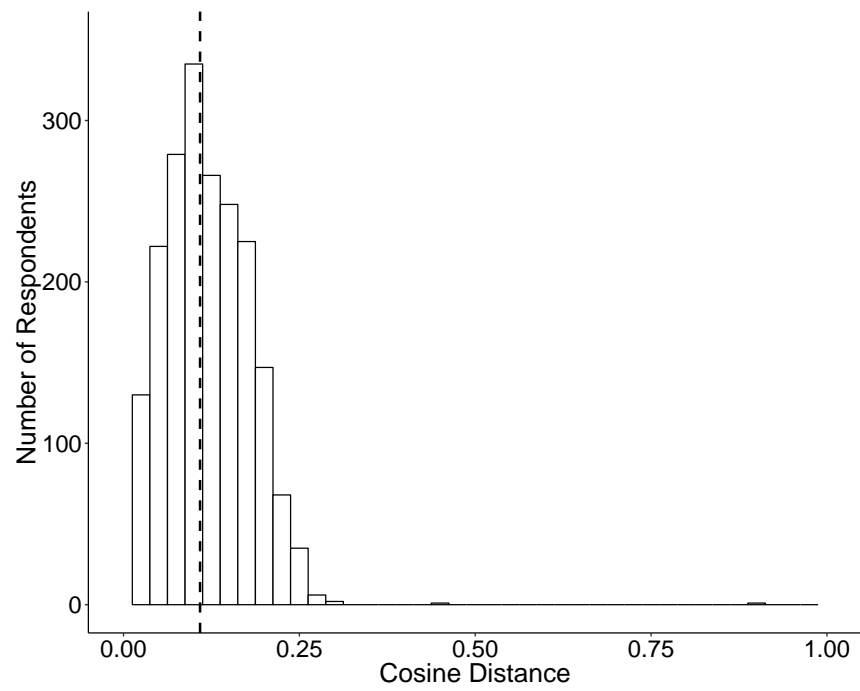
Notes: The mean distance for each country is represented by the vertical dotted-line.

Figure 2: Distribution of raw cosine of angles for respondents in Brazil and Mexico.

(a) Brazil

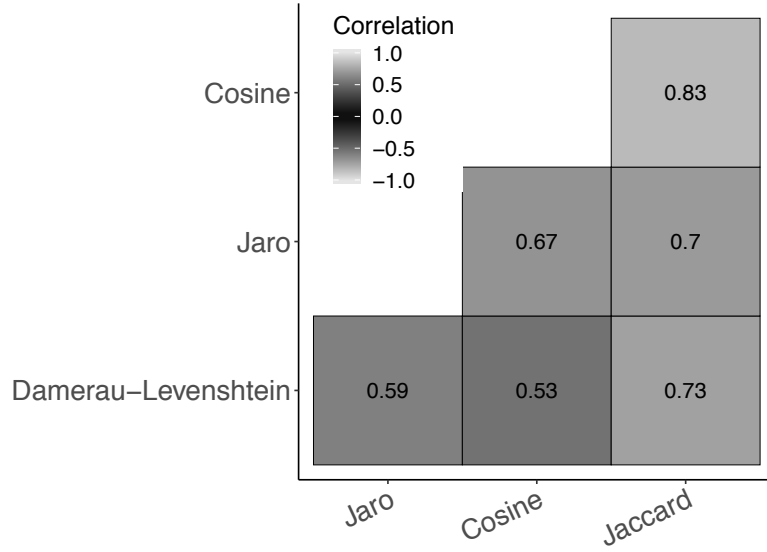


(b) Mexico



Notes: The mean distance for each country is represented by the vertical dotted-line.

Figure 3: Correlation between distance measures for respondents in Brazil and Mexico.



Now, I present the regression results from models estimated with (1) the full sample irrespective of attention, (2) a reduced sample using list-wise deletion based on an arbitrary threshold set for participants that "passed" (those respondents with weights ≥ 0.1), and (3) a weighted least squares model based on the weighted average of the Jaccard and cosine similarity measures.

To execute the three regressions, we can run the function `regressionComparison`, which estimates the three separate regression models. You do not need to calculate the average similarity, the function computes this for you, you only need to define a value for k and which similarity measures to include in the averaged measure. The output of the regression models from this function will be automatically loaded into your global environment (for instance, labeled as name of outcome + "_baseModel") and can be used as typical regression objects in R, so we can get the estimated coefficients to reproduce Table 1.

```

1 # execute regressions to compare overall results for 5 models
2 # (1) Trust
3 # (2) Responsiveness
4 # (3) Future Attendance
5 # (4) Future Volunteerism
6 # (5) Interest in signing political petition
7 trustLM <- formula(trustChurch_postTreat ~ Concordant*attendanceBin)
8 responsiveLM <- update(trustLM, responsiveness_postTreat ~ .)
9 volunteerLM <- update(trustLM, futureVolunteer_postTreat ~ .)
10 attendanceLM <- update(trustLM, futureAttendance_postTreat ~ .)
11 petitionLM <- update(trustLM, petitionInterest_postTreat ~ .)
12
13 for(current_model in c(trustLM, responsiveLM, volunteerLM, attendanceLM, petitionLM)){
14   regressionComparison(formula=current_model,
15     dataframe=replication_complete.cases, k=3,
16     similarity_measures=c("jaccardDist", "cosineDist"),
17     model_type="ls")
18 }

```

Table 1: Estimated coefficients from (1) regression with all observations, (2) weighted regression based on attentiveness, (3) regression on subsetted sample based on attentiveness.

	Outcome:														
	(1) Trust	(2) Trust	(3) Trust	(4) Responsive	(5) Responsive	(6) Responsive	(7) Volunteer	(8) Volunteer	(9) Volunteer	(10) Attendance	(11) Attendance	(12) Attendance	(13) Petition	(14) Petition	(15) Petition
Responsive papal messaging	-0.27 (0.13)	-0.29 (0.13)	-0.30 (0.14)	-0.06 (0.14)	-0.06 (0.14)	-0.04 (0.14)	-0.54 (0.14)	-0.67 (0.14)	-0.63 (0.14)	-0.36 (0.13)	-0.42 (0.13)	-0.40 (0.13)	-0.15 (0.13)	-0.15 (0.13)	-0.18 (0.14)
Attendance (Monthly)	0.86 (0.12)	0.93 (0.13)	0.92 (0.13)	0.89 (0.13)	0.89 (0.13)	0.91 (0.14)	1.31 (0.13)	1.31 (0.14)	1.34 (0.14)	1.49 (0.12)	1.52 (0.13)	1.57 (0.13)	0.59 (0.12)	0.59 (0.13)	0.64 (0.13)
Attendance (Weekly)	1.83 (0.12)	1.90 (0.12)	1.91 (0.13)	1.70 (0.13)	1.70 (0.13)	1.73 (0.13)	2.39 (0.12)	2.41 (0.13)	2.46 (0.13)	2.28 (0.12)	2.34 (0.12)	2.35 (0.12)	0.94 (0.12)	0.95 (0.12)	0.99 (0.12)
Responsiveness*Attendance (Monthly)	0.42 (0.18)	0.43 (0.18)	0.48 (0.19)	0.14 (0.19)	0.14 (0.19)	0.19 (0.19)	0.65 (0.19)	0.72 (0.19)	0.73 (0.20)	0.48 (0.17)	0.50 (0.18)	0.48 (0.18)	0.09 (0.18)	0.11 (0.18)	0.14 (0.19)
Responsiveness*Attendance (Weekly)	0.46 (0.17)	0.48 (0.17)	0.49 (0.18)	0.38 (0.18)	0.38 (0.18)	0.40 (0.18)	0.72 (0.17)	0.83 (0.18)	0.78 (0.19)	0.63 (0.16)	0.67 (0.17)	0.65 (0.17)	0.28 (0.17)	0.24 (0.17)	0.26 (0.18)
Weights	✓			✓			✓			✓			✓		
R ²	0.13	0.14	0.14	0.11	0.11	0.11	0.19	0.20	0.21	0.20	0.21	0.21	0.04	0.04	0.04
Adj. R ²	0.13	0.14	0.14	0.11	0.11	0.11	0.19	0.20	0.21	0.20	0.21	0.21	0.04	0.04	0.04
N	4206	3936	3698	3936	3936	3698	4206	3936	3698	4206	3936	3698	4206	3936	3698

Notes: Standard errors are presented in the parentheses.

Once we have estimated our three regression models, we can estimate and plot the average marginal effects with the function `plotMarginalEffect`. Users must only insert the relevant regression models they have estimated with `regressionComparison` previously. The output of `plotMarginalEffect` is a plot of the marginal effects given the user-defined formula that was used to estimate the regression models, such as Figure 4.

```

1 # plot overall marginal effects for three regression models
2 # replicate Figure A.10 on A22
3 plotMarginalEffect(regression_models=list(c(baseModel_trustChurch_postTreat, listwiseModel_
4     trustChurch_postTreat,
5     weightedModel_trustChurch_postTreat),
6     c(baseModel_responsiveness_postTreat,
7     listwiseModel_responsiveness_postTreat,
8     weightedModel_responsiveness_postTreat),
9     c(baseModel_futureAttendance_postTreat,
10    listwiseModel_futureAttendance_postTreat,
11    weightedModel_futureAttendance_postTreat),
12    c(baseModel_futureVolunteer_postTreat,
13    listwiseModel_futureVolunteer_postTreat,
14    weightedModel_futureVolunteer_postTreat),
15    c(baseModel_petitionInterest_postTreat,
16    listwiseModel_petitionInterest_postTreat,
17    weightedModel_petitionInterest_postTreat)))

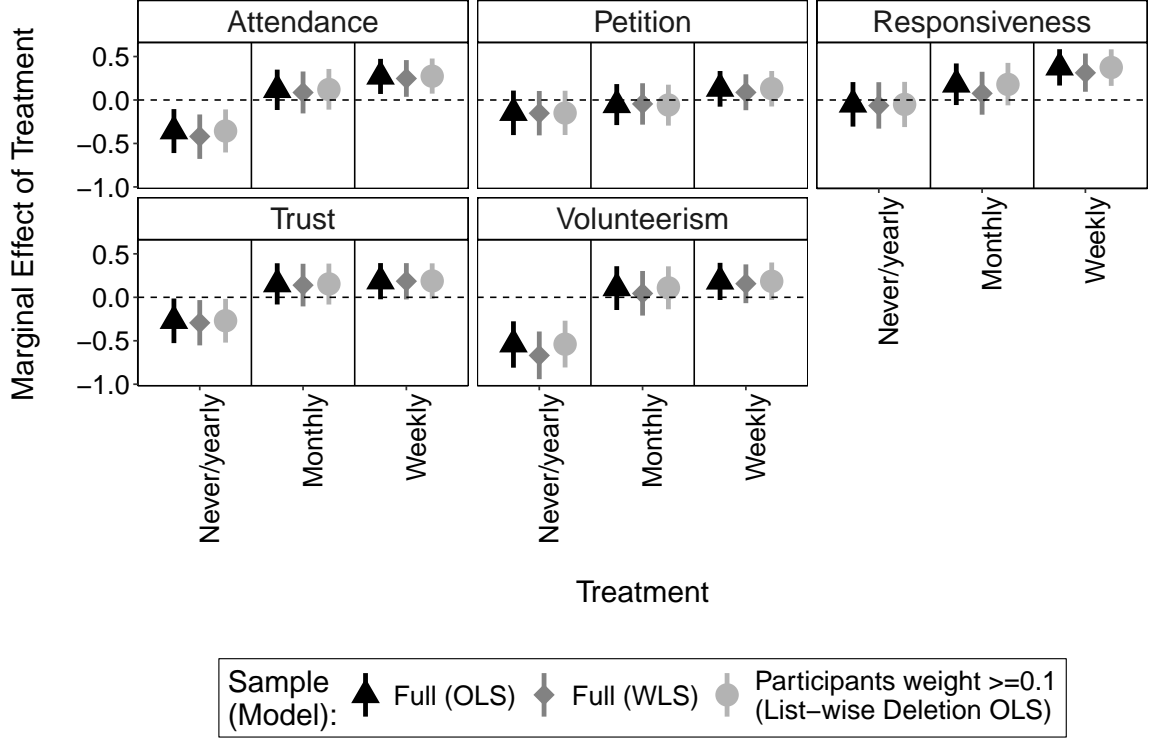
```

The main findings, presented in Figure 4, indicate that dedicated members (those that attend church weekly) were more likely to increase their anticipated future attendance of Church services. Again, the full estimated coefficients are shown in Table 1. When asked how strongly respondents agree with the statement, "I plan to attend more church services in the future", members that attend church services weekly were more likely to increase their support if they receive responsiveness.

The estimated average treatment effect of receiving papal responsiveness for weekly attendees was associated with about a 0.3 point increase in the strength of their anticipated attendance of church services. These findings suggest that respondents' were more willing to view the Church as responsive, and more willing to participate in the Church, when they receive responsive papal statements. The results do not change substantively or statistically when the full sample is used versus samples that exclude or weight respondents based on attention. This signals that inattentive participants and attentive participants do not respond to the outcomes systematically different, or at least not enough to alter the overall treatment effects. To double-check, I simulate the average treatment effect for those that would pass and fail the manipulation check by randomly varying the threshold of document similarity for passing.

To investigate whether attentive and inattentive participants respond differently in a systematic manner, which may explain some of the null estimates of the overall ATEs in Figure 4, I simulate the distribution of ATE for compliers and non-compliers. We can achieve this by executing `plotComplierATE`, and the output is Figure 5. The user merely inputs the cutoff threshold which represents that maximum value of attention at which a participant would be considered a non-complier, while n references how many simulations the user wishes to perform (the default is 100 which matches the application in the manuscript).

Figure 4: Marginal treatment effects by church attendance and sample.



Notes: The figure plots marginal effect of the treatment measured by the change in the predicted level of support among the outcome categories. The mean marginal effects is represented by the solid point, while the 2.5%-97.5% percentiles of the sampling distributions are designated by the vertical lines. The marginal effects of each country are generated from 10,000 simulations that use asymptotic normal approximation to the log-likelihood to estimate the first difference for each category of attendance.

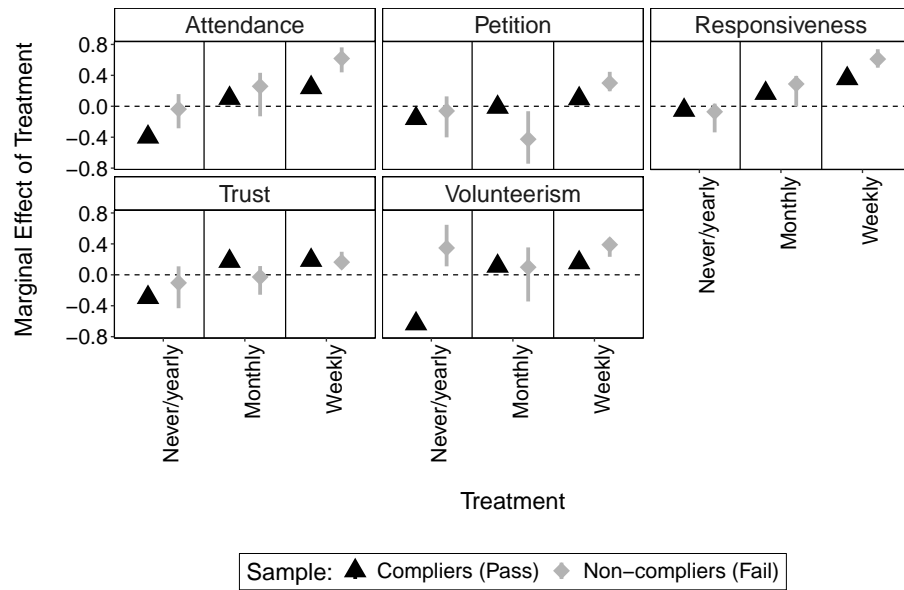
```

1 # plot marginal effects for "complier" and "non-complier"
2 plotComplierATE(dataframe=replication_complete.cases, similarity_measures=c("jaccardDist", "
  cosineDist"),
3                   bounds=c(0.1, 0.2), n=100, seed=12345, k=3, model_type="ls",
4                   formula=list(trustLM, responsiveLM, volunteerLM, attendanceLM, petitionLM))

```

Figure 5 plots the median treatment effect for 100 simulations of the ATE for participants above (those that "passed") and below (those that "failed") a randomly selected weight threshold. Beginning with those participants that would pass the manipulation check, we can see that the ATE typically increases as respondents' church attendance increases. Moreover, the distribution is tightly compact showing little variation in the ATE of compliers. Non-compliers do not consistently differ from compliers, with the exception of a few outcomes. Rather, non-compliers appear to add more uncertainty and heterogeneity into the average treatment effect, which may explain the lack of precision for the ATEs in Figure 4.

Figure 5: Distribution of average marginal treatment effects by church attendance for respondents that likely absorbed the treatment and those that did not.



Notes: The figure plots the median marginal effects of respondents that "passed" the manipulation check. The vertical lines represent the 2.5%-97.5% percentiles of the sampling distribution of the average marginal effect for compliers and non-compliers. Some distributions are so narrow that they are subsumed by the point estimate. Each distribution consists of $N = 100$.