Cognitive Bias and Symptom Importance, Main Analyses

Chris Beevers

8/9/2018

This report includes code snippets used to generate results for all the analyses reported in Beevers et al., *Association between negative cognitive bias and depression: A symptom-level approach.*

R packages used in this analysis. Note that the beset package is not available on CRAN, but can be found here: https://github.com/jashu/beset

```
library(knitr); library(haven)
library(tidyverse)
library(beset)
library(gridExtra); library(ggpubr)
library(irr)
```

Demographic data

Summarize sample. Table 1.

```
#age
kable(demo %>%
  dplyr::summarize(N = n_distinct(id),
            age_mean = mean(age, na.rm = TRUE),
            age_sd = sd(age, na.rm = TRUE)))
```

N	age_mean	age_sd
218	23.29358	4.877868

```
#gender
kable(demo %>%
  group_by(female) %>%
  summarise(n = n()) %>%
  mutate(freq = (n / sum(n))*100) %>%
  ungroup())
```

female	n	freq
0	74	33.94495
1	144	66.05505

```
#hispanic
kable(demo %>%
  group_by(hispanic) %>%
  summarise(n = n()) %>%
  mutate(freq = (n / sum(n))*100) %>%
```

```
  ungroup())
```

hispanic	n	freq
0	151	69.26605
1	67	30.73394

```
#white (2 individuals indicated an additional race)
kable(demo %>%
  group_by(white) %>%
  summarise(n = n()) %>%
  mutate(freq = (n / sum(n))*100) %>%
  ungroup())
```

white	n	freq
0	2	0.9174312
1	216	99.0825688

```
#never married
kable(demo %>%
  group_by(never_marry_yn) %>%
  summarise(n = n()) %>%
  mutate(freq = (n / sum(n))*100) %>%
  ungroup())
```

never_marry_yn	n	freq
0	33	15.13761
1	185	84.86239

```
#years in school
kable(demo %>%
  dplyr::summarize(
    yis_mean = mean(years_of_school, na.rm = TRUE),
    yis_sd = sd(years_of_school, na.rm = TRUE)))
```

yis_mean	yis_sd
14.42661	2.387878

```
#private health insurance
kable(demo %>%
  group_by(priv_ins_yn) %>%
  summarise(n = n()) %>%
  mutate(freq = (n / sum(n))*100) %>%
  ungroup())
```

priv_ins_yn	n	freq
0	91	41.7431193
1	126	57.7981651

priv_ins_yn	n	freq
NA	1	0.4587156

```r
#household income
kable(demo %>%
  group_by(financial_support) %>%
  summarise(n = n()) %>%
  mutate(freq = (n / sum(n))*100) %>%
  ungroup())
```

financial_support	n	freq
1	84	38.532110
2	45	20.642202
3	26	11.926606
4	19	8.715596
5	44	20.183486

```r
#mdd
kable(demo %>%
  group_by(mdd_current_meets) %>%
  summarise(n = n()) %>%
  mutate(freq = (n / sum(n))*100) %>%
  ungroup())
```

mdd_current_meets	n	freq
0	156	71.5596330
1	60	27.5229358
NA	2	0.9174312

```r
#lifetime mdd
kable(demo %>%
  group_by(mdd_past_meets) %>%
  summarise(n = n()) %>%
  mutate(freq = (n / sum(n))*100) %>%
  ungroup())
```

mdd_past_meets	n	freq
0	83	38.073394
1	132	60.550459
NA	3	1.376147

```r
#any current anxiety disorder
kable(demo %>%
  group_by(tot_cur_anxiety_yn) %>%
  summarise(n = n()) %>%
  mutate(freq = (n / sum(n))*100) %>%
  ungroup())
```

tot_cur_anxiety_yn	n	freq
0	154	70.6422018
1	62	28.4403670
NA	2	0.9174312

```r
#any current disorder
kable(demo %>%
  group_by(tot_cur_diag_yn) %>%
  summarise(n = n()) %>%
  mutate(freq = (n / sum(n))*100) %>%
  ungroup())
```

tot_cur_diag_yn	n	freq
0	100	45.871560
1	115	52.752294
NA	3	1.376147

```r
#BDI
kable(demo %>%
  dplyr::summarize(
    bdi_mean = mean(bdi_total, na.rm = TRUE),
    bdi_sd = sd(bdi_total, na.rm = TRUE)))
```

bdi_mean	bdi_sd
17.98165	11.55231

```r
cor.test(demo$mdd_current_meets, demo$tot_cur_anxiety_yn)
```

```
##
##  Pearson's product-moment correlation
##
## data:  demo$mdd_current_meets and demo$tot_cur_anxiety_yn
## t = 5.6538, df = 214, p-value = 4.982e-08
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.2384841 0.4713116
## sample estimates:
##       cor
## 0.3605002
```

Last paragraph of the participants section.

```r
#comorbidity between current MDD and anxiety disorder
kable(demo %>%
  group_by(mdd_anx_comor) %>%
  summarise(n = n()) %>%
  mutate(freq = (n / sum(n))*100) %>%
  ungroup())
```

mdd_anx_comor	n	freq
0	183	83.9449541
1	33	15.1376147
NA	2	0.9174312

```
#of those with current MDD, how many have an anxiety disorder
kable(demo %>%
  filter(mdd_current_meets==1) %>%
  group_by(tot_cur_anxiety_yn) %>%
  summarise(n = n()) %>%
  mutate(freq = (n / sum(n))*100) %>%
  ungroup())
```

tot_cur_anxiety_yn	n	freq
0	27	45
1	33	55

Inter-rater reliability for MINI interviews

```
rater_test <- filter(rater, field == "MDD Current")
kappam.fleiss(rater_test[,c(2,3,4)])
```

```
##  Fleiss' Kappa for m Raters
##
##  Subjects = 22
##    Raters = 3
##     Kappa = 1
##
##         z = 8.12
##   p-value = 4.44e-16
```

```
rater_test <- filter(rater, field == "MDD Lifetime")
kappam.fleiss(rater_test[,c(2,3,4)])
```

```
##  Fleiss' Kappa for m Raters
##
##  Subjects = 22
##    Raters = 3
##     Kappa = 0.812
##
##         z = 6.6
##   p-value = 4.21e-11
```

```
rater_test <- filter(rater, field == "MDD Recurrent")
kappam.fleiss(rater_test[,c(2,3,4)])
```

```
##  Fleiss' Kappa for m Raters
##
##  Subjects = 22
##    Raters = 3
```

```
##     Kappa = 0.939
##
##         z = 7.63
##   p-value = 2.33e-14
```

```r
rater_test <- filter(rater, field == "Panic Disorder Current")
kappam.fleiss(rater_test[,c(2,3,4)])
```

```
##  Fleiss' Kappa for m Raters
##
##  Subjects = 22
##    Raters = 3
##     Kappa = 1
##
##         z = 8.12
##   p-value = 4.44e-16
```

```r
rater_test <- filter(rater, field == "Panic Disorder Lifetime")
kappam.fleiss(rater_test[,c(2,3,4)])
```

```
##  Fleiss' Kappa for m Raters
##
##  Subjects = 22
##    Raters = 3
##     Kappa = 1
##
##         z = 8.12
##   p-value = 4.44e-16
```

```r
rater_test <- filter(rater, field == "GAD Current")
kappam.fleiss(rater_test[,c(2,3,4)])
```

```
##  Fleiss' Kappa for m Raters
##
##  Subjects = 21
##    Raters = 3
##     Kappa = 1
##
##         z = 7.94
##   p-value = 2e-15
```

```r
rater_test <- filter(rater, field == "Alcohol Use Disorder Current" )
kappam.fleiss(rater_test[,c(2,3,4)])
```

```
##  Fleiss' Kappa for m Raters
##
##  Subjects = 22
##    Raters = 3
##     Kappa = 0.858
##
##         z = 6.97
##   p-value = 3.2e-12
```

```r
rater_test <- filter(rater, field == "OCD Current" )
kappam.fleiss(rater_test[,c(2,3,4)])
```

```
##  Fleiss' Kappa for m Raters
##
```

```
##  Subjects = 22
##    Raters = 3
##     Kappa = 0.784
##
##         z = 6.37
##   p-value = 1.94e-10
```

Split-half reliability for SRET metrics

Number of negative words endorsed.

```r
library(boot)
cor.mu <- function(df, n) {
     df = df[n,]
     x <- df$even
     y <- df$odd
     return(cor(x, y, use = "complete.obs", method = "spearman"))
 }

sret_bias <- sret_summary %>%
  select(id, trial_type, num.neg.words) %>%
  spread(trial_type, num.neg.words)

boot.cor.mu <- boot(sret_bias, cor.mu, R = 10000)
boot.cor.mu
```

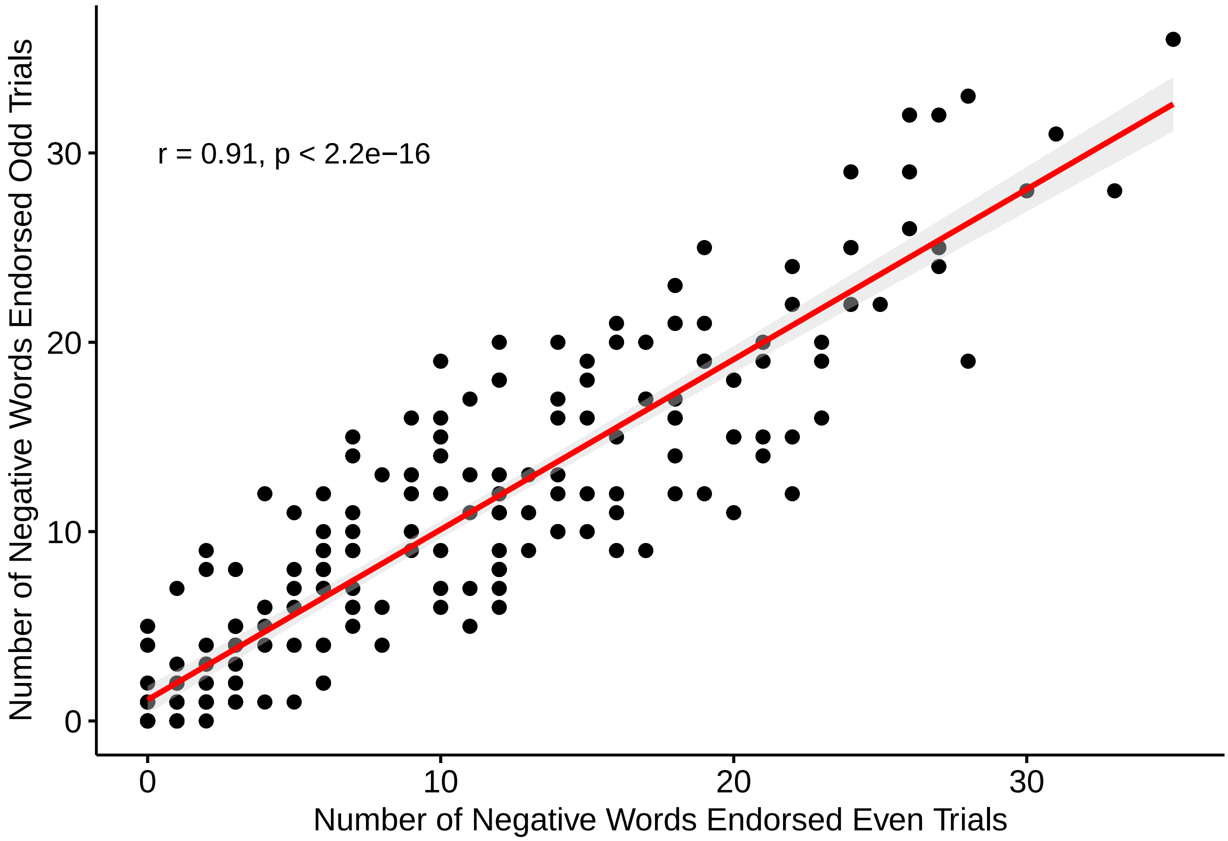
```
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = sret_bias, statistic = cor.mu, R = 10000)
##
##
## Bootstrap Statistics :
##      original       bias    std. error
## t1* 0.9100278 -0.002026475  0.01178537
```

```r
boot.ci(boot.cor.mu, type = "norm")
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 10000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = boot.cor.mu, type = "norm")
##
## Intervals :
## Level      Normal
## 95%   ( 0.8890,  0.9352 )
## Calculations and Intervals on Original Scale
```

```r
ggscatter(sret_bias, x = "even", y = "odd",
   add = "reg.line",
   add.params = list(color = "red", fill = "lightgray"),
```

```
  conf.int = TRUE) +
  stat_cor(method = "spearman", label.x = 5, label.y = 30) +
  labs(x = "Number of Negative Words Endorsed Even Trials",
       y = "Number of Negative Words Endorsed Odd Trials")
```



The spearman rho for number of negative words endorsed using 10,000 bootstrap samples is excellent, *rho* = 0.91, 95% CI [.89, .94]. Note that for these analyses only we used a simple count and didn't require endorsements on two of three repititions because for these analyses we split the data into even and odd trials.

Drift rate.

```
drift_bias <- drift %>%
  select(id, trial_type, v_negative) %>%
  spread(trial_type, v_negative) %>%
  mutate(even = as.numeric(even),
         odd = as.numeric(odd))

boot.cor.mu <- boot(drift_bias, cor.mu, R = 10000)
boot.cor.mu
```

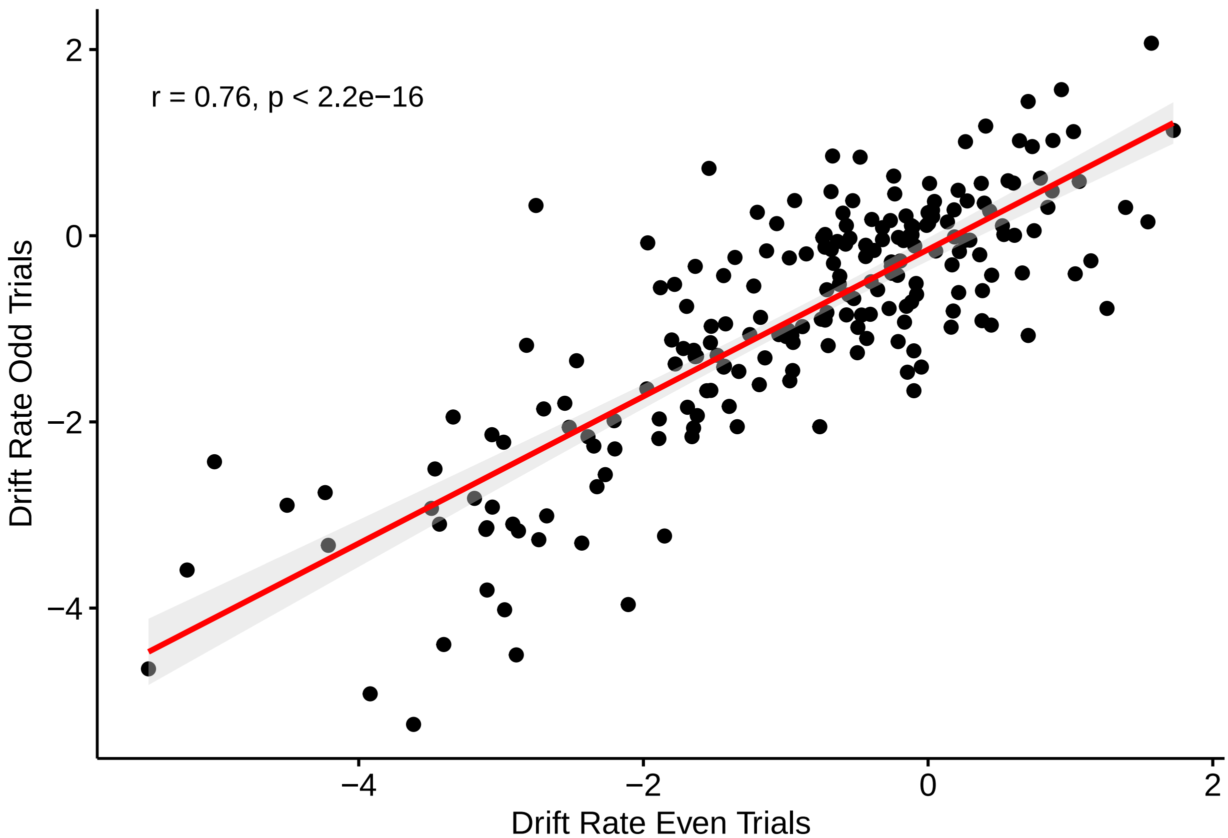
```
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = drift_bias, statistic = cor.mu, R = 10000)
##
##
## Bootstrap Statistics :
```

```
##      original      bias    std. error
## t1* 0.7639311 -0.00254327  0.03810449
```

```
boot.ci(boot.cor.mu, type = "norm")
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 10000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = boot.cor.mu, type = "norm")
##
## Intervals :
## Level      Normal
## 95%   ( 0.6918,  0.8412 )
## Calculations and Intervals on Original Scale
```

```
ggscatter(drift_bias, x = "even", y = "odd",
   add = "reg.line",
   add.params = list(color = "red", fill = "lightgray"),
   conf.int = TRUE) +
  stat_cor(method = "spearman", label.x = -4.5, label.y = 1.5) +
  labs(x = "Drift Rate Even Trials",
       y = "Drift Rate Odd Trials")
```



The spearman rho for drift rate for negative words using 10,000 bootstrap samples is good, $rho = 0.76$, 95% CI [.69, .84]. Reliability is good despite using a technique (diffusion model) that typically requires a larger number of trials than what we used in the current study, especially given that for reliabiltiy analyses we split analyses into even and odd trials.

Number of trials dropped because of wrong response or outlier

```
(96-mean(bias_missing$n_dp_valid))/96*100
```

```
## [1] 9.84805
```

There were 96 trials involving sad stimuli. Thus, the percentage of trials missing due to incorrect response or deemed an outlier was 9.84%.

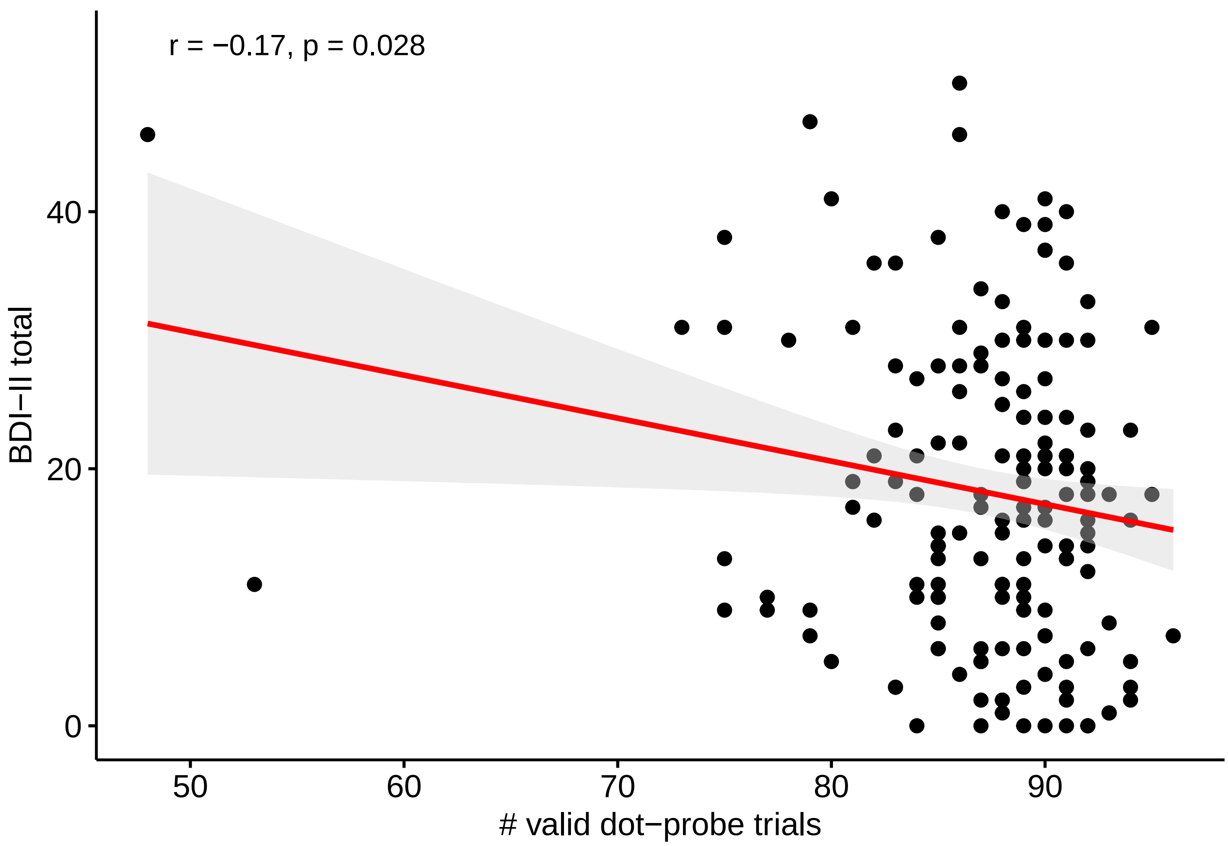
```
load(file="~/Box/Active/Projects/R56/papers/symptom importance/data/r56_dp_drop_session1.RData")

demo <- left_join(demo, bias_missing)
```

```
## Joining, by = "id"
```

```
demo <- left_join(demo, dp_drop)
```

```
## Joining, by = "id"
```

```
ggscatter(demo[which(demo$dp_keep==1),], x = "n_dp_valid", y = "bdi_total",
   add = "reg.line",
   add.params = list(color = "red", fill = "lightgray"),
   conf.int = TRUE) +
  stat_cor(method = "pearson", label.x = 55, label.y = 53) +
  labs(x = "# valid dot-probe trials",
       y = "BDI-II total") +
  theme(legend.position= "none")
```



Among those retained for dot-probe analyses, there was a small correlation between number of valid trials and depression severity (r = -0.17). People with higher depression severity had fewer valid trials. However, all retained participants had at least 70 trials with good data, except for two participants who had 50 and 53 good trials, respectively. Thus, we believe the amount of analyzed data was adequate, regardless of depression severity. Perhaps not surprisingly, people with higher depression severity were more likely to make mistakes or respond too slowly or quickly.

Split-half reliability for dot-probe metrics

Spearman rho for traditional RT.

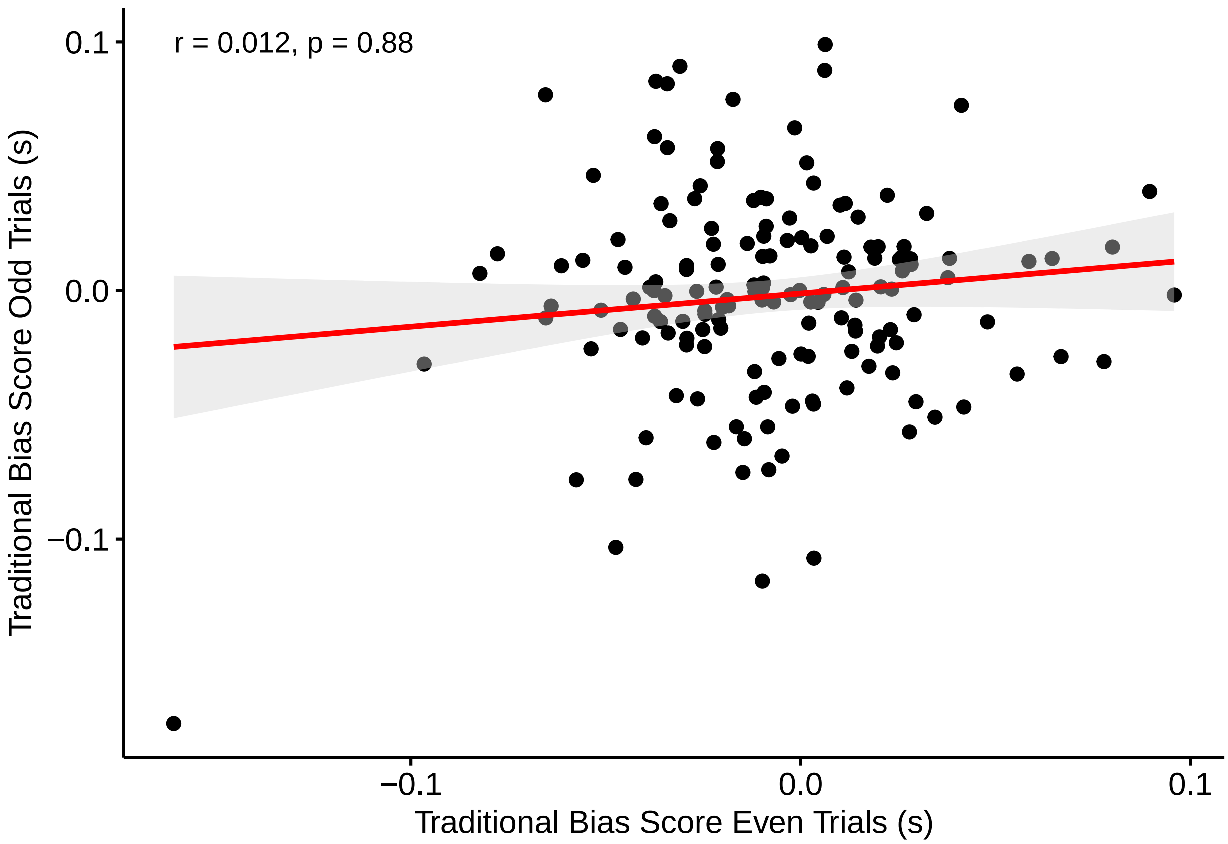
```
trad_bias <- bias_summary %>%
  select(id, trial_type, dp_bias) %>%
  spread(trial_type, dp_bias)

boot.cor.mu <- boot(trad_bias, cor.mu, R = 10000)
boot.cor.mu
```

```
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = trad_bias, statistic = cor.mu, R = 10000)
##
##
## Bootstrap Statistics :
##      original        bias    std. error
## t1* 0.0122489 -0.002136211  0.08000153
```

```
boot.ci(boot.cor.mu, type = "norm")
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 10000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = boot.cor.mu, type = "norm")
##
## Intervals :
## Level      Normal
## 95%   (-0.1424,  0.1712 )
## Calculations and Intervals on Original Scale
```

```
ggscatter(trad_bias, x = "even", y = "odd",
   add = "reg.line",
   add.params = list(color = "red", fill = "lightgray"),
   conf.int = TRUE) +
  stat_cor(method = "spearman", label.x = -.13, label.y = .1) +
  labs(x = "Traditional Bias Score Even Trials (s)",
       y = "Traditional Bias Score Odd Trials (s)")
```



The spearman rho for the traditional bias score measured with reaction time is very poor, $rho = 0.012$, 95% CI [-.14, .17].

Spearman rho for TLBS towards sad stimuli.

```
tlbs_bias <- bias_summary %>%
  select(id, trial_type, mean_dp_toward) %>%
  spread(trial_type, mean_dp_toward)

boot.cor.mu <- boot(tlbs_bias, cor.mu, R = 10000)
boot.cor.mu
```

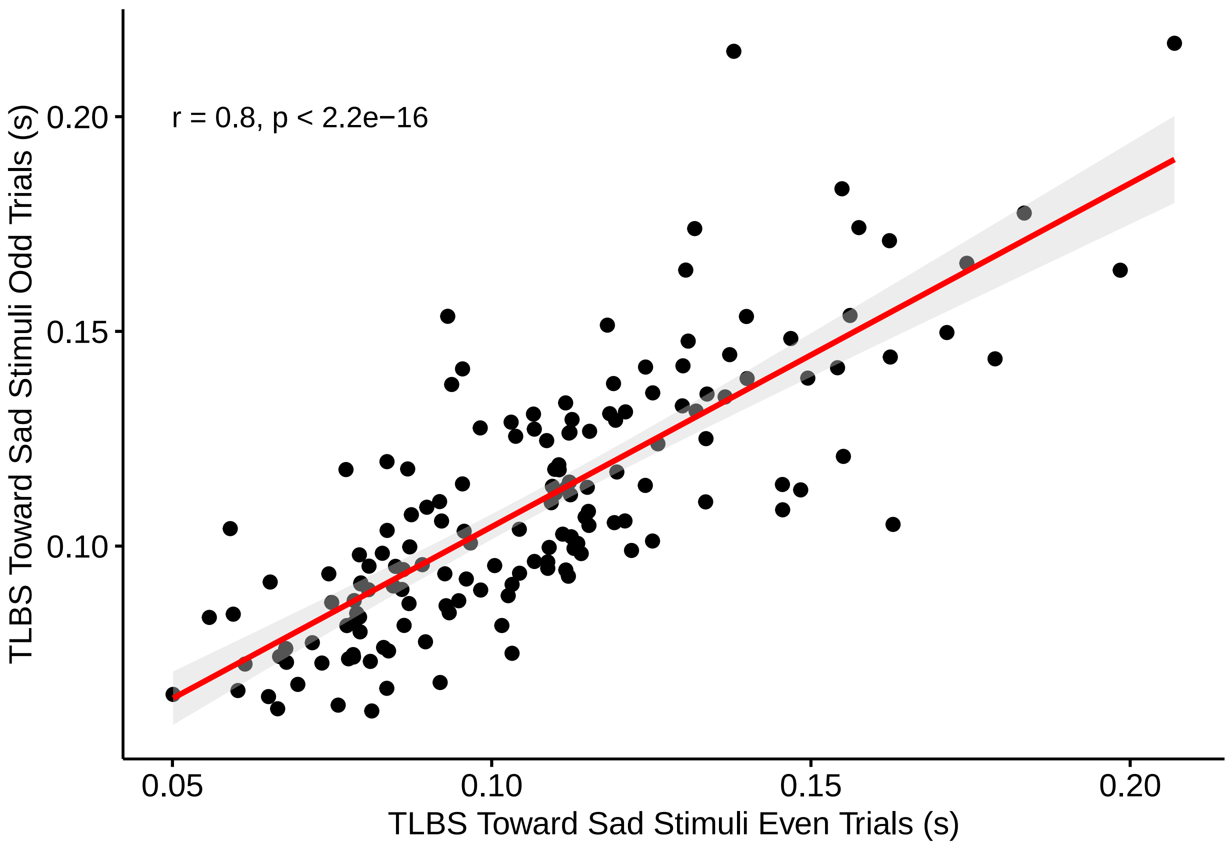
```
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = tlbs_bias, statistic = cor.mu, R = 10000)
##
##
## Bootstrap Statistics :
##      original       bias    std. error
## t1* 0.8029416 -0.004289624  0.03210412
```

```
boot.ci(boot.cor.mu, type = "norm")
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 10000 bootstrap replicates
##
## CALL :
```

```
## boot.ci(boot.out = boot.cor.mu, type = "norm")
##
## Intervals :
## Level      Normal
## 95%   ( 0.7443,  0.8702 )
## Calculations and Intervals on Original Scale
```

```
ggscatter(tlbs_bias, x = "even", y = "odd",
   add = "reg.line",
   add.params = list(color = "red", fill = "lightgray"),
   conf.int = TRUE) +
  stat_cor(method = "spearman", label.x = .07, label.y = .2) +
  labs(x = "TLBS Toward Sad Stimuli Even Trials (s)",
       y = "TLBS Toward Sad Stimuli Odd Trials (s)")
```



The spearman rho for the tlbs towards sad stimuli measured with reaction time is very good, $rho = 0.80$, 95% CI [.74, .87].

Spearman rho for eye gaze bias score for sad stimuli.

```
gz_bias <- bias_summary %>%
  select(id, trial_type, gaze_bias) %>%
  spread(trial_type, gaze_bias)

boot.cor.mu <- boot(gz_bias, cor.mu, R = 10000)
boot.cor.mu
```

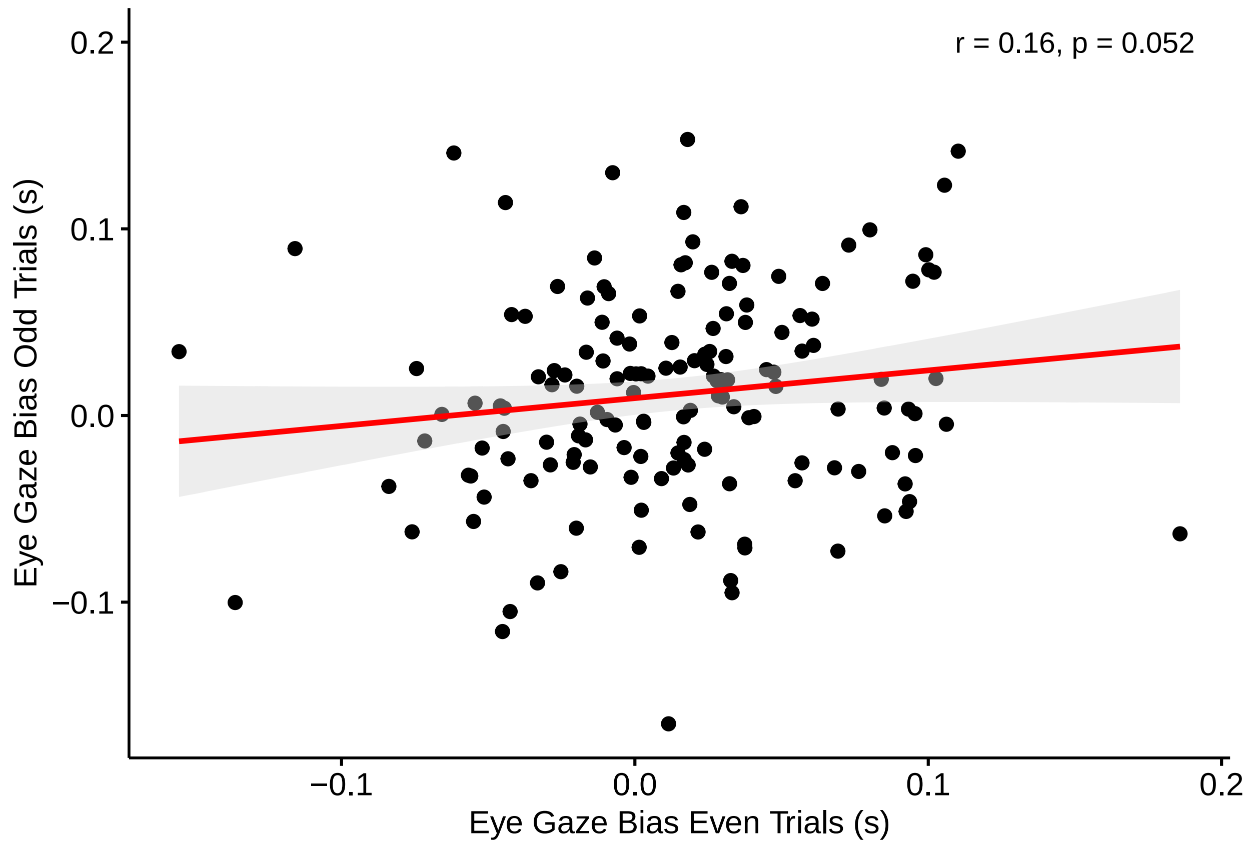
```
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
```

```
##
## Call:
## boot(data = gz_bias, statistic = cor.mu, R = 10000)
##
##
## Bootstrap Statistics :
##      original        bias    std. error
## t1* 0.1556757 -0.0003830148  0.08370235
```

```
boot.ci(boot.cor.mu, type = "norm")
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 10000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = boot.cor.mu, type = "norm")
##
## Intervals :
## Level      Normal
## 95%   (-0.0080,  0.3201 )
## Calculations and Intervals on Original Scale
```

```
ggscatter(gz_bias, x = "even", y = "odd",
   add = "reg.line",
   add.params = list(color = "red", fill = "lightgray"),
   conf.int = TRUE) +
  stat_cor(method = "spearman", label.x = .15, label.y = .2) +
  labs(x = "Eye Gaze Bias Even Trials (s)",
       y = "Eye Gaze Bias Odd Trials (s)")
```



The spearman rho for eye gaze bias is also very low, $rho = 0.16$, 95% CI [-.01, .32].

Spearman rho for percentage of trials where total fixation is greater for sad than neutral stimuli.

```
gz_pct_bias <- bias_summary %>%
  select(id, trial_type, pct_gaze_toward) %>%
  spread(trial_type, pct_gaze_toward)

boot.cor.mu <- boot(gz_pct_bias, cor.mu, R = 10000)
boot.cor.mu
```

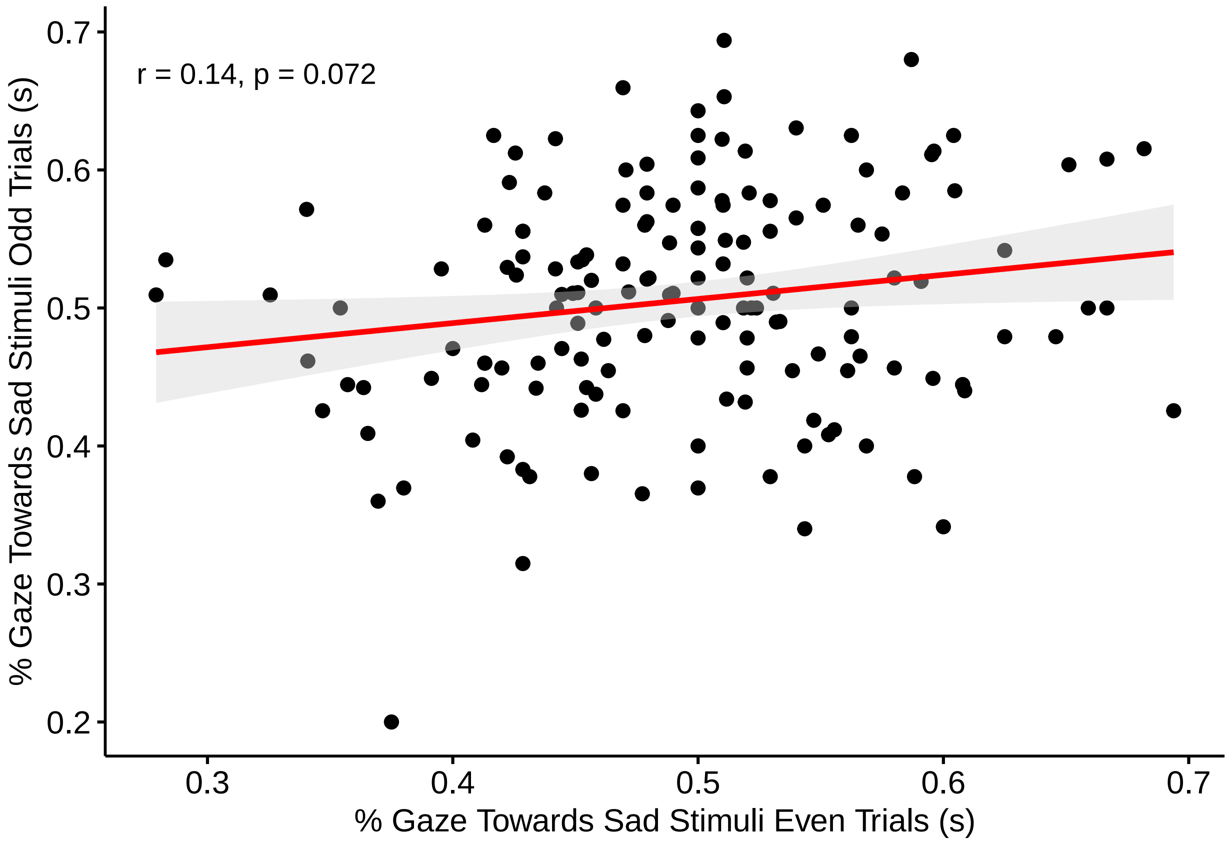
```
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = gz_pct_bias, statistic = cor.mu, R = 10000)
##
##
## Bootstrap Statistics :
##      original        bias    std. error
## t1* 0.1439954 -0.0007671209  0.08244898
```

```
boot.ci(boot.cor.mu, type = "norm")
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 10000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = boot.cor.mu, type = "norm")
```

```
##
## Intervals :
## Level      Normal
## 95%   (-0.0168,  0.3064 )
## Calculations and Intervals on Original Scale
```

```r
ggscatter(gz_pct_bias, x = "even", y = "odd",
   add = "reg.line",
   add.params = list(color = "red", fill = "lightgray"),
   conf.int = TRUE) +
  stat_cor(method = "spearman", label.x = .32, label.y = .67) +
  labs(x = "% Gaze Towards Sad Stimuli Even Trials (s)",
       y = "% Gaze Towards Sad Stimuli Odd Trials (s)")
```



The spearman rho for percentage of trials towards sad stimuli is also low, *rho* = 0.14, 95% CI [-.02, .30]. Thus, the TLBS towards sad stimuli is the only dot-probe index with acceptable internal reliablity. Nevertheless, the traditional scoring of the dot-probe is used so often that we will retain those metrics for analyses, bearing in mind their poor psychometric characteristics.

SRET: Drift rate

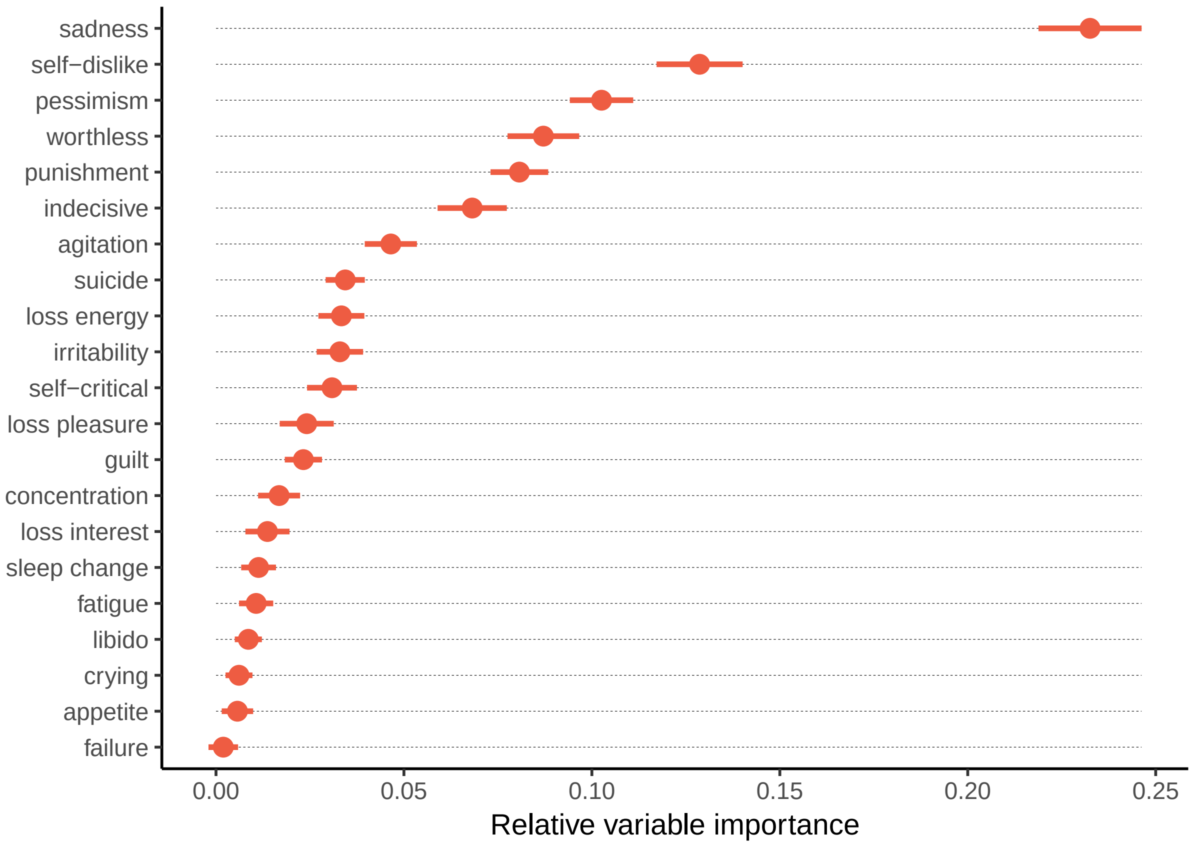
Random forest model for drift rate.

```r
drift_rf <- beset_rf(v_negative ~ ., data=bdi[,c(2:23)])
summary(drift_rf)
```

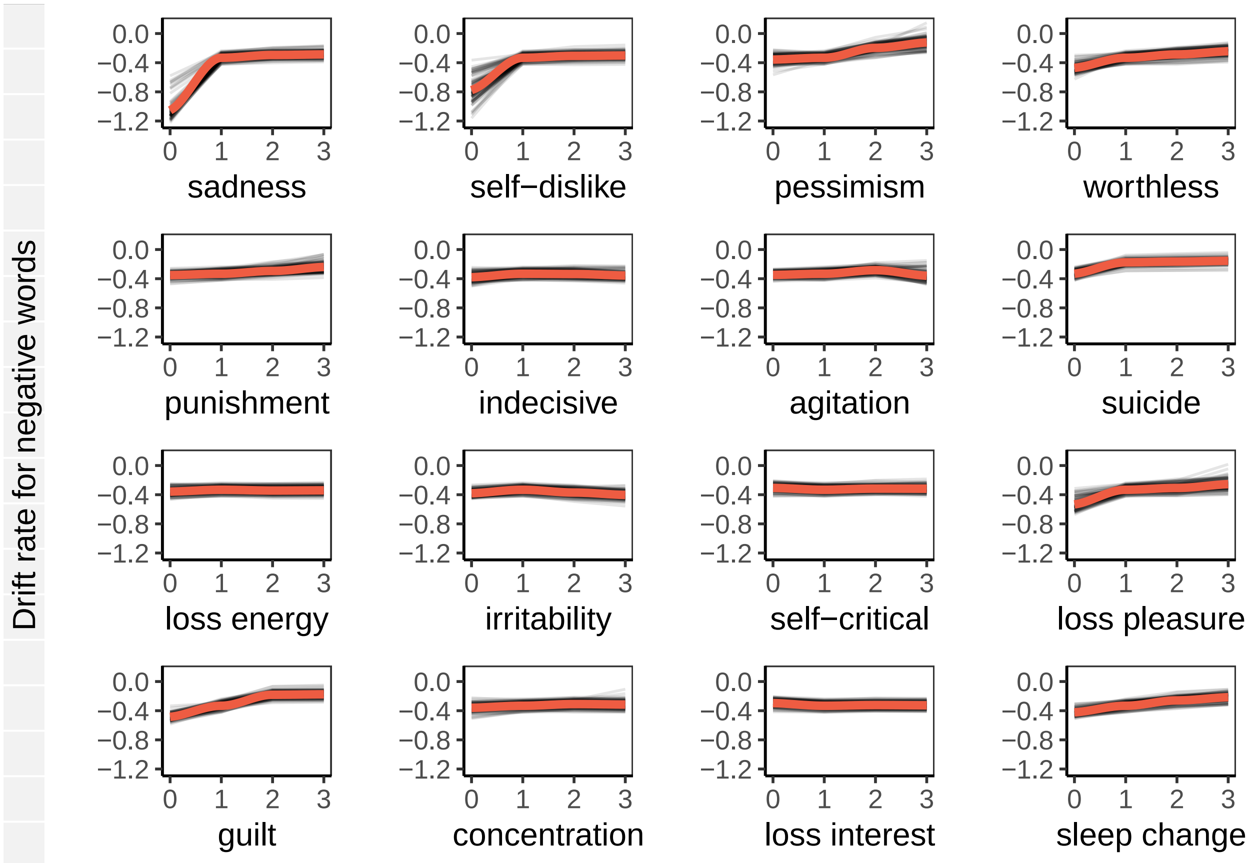
```
## Type of random forest: regression
```

```
## Number of trees: 500
## No. of variables tried at each split: 7
## ========================================================
## OOB estimate of % Var explained: 39.81
## CV estimate of % Var explained: 40.74
##
##            Importance    Min   Max
## pre_bdi_1       0.233  0.219 0.246
## pre_bdi_7       0.129  0.117 0.140
## pre_bdi_2       0.103  0.094 0.111
## pre_bdi_14      0.087  0.078 0.097
## pre_bdi_6       0.081  0.073 0.088
## pre_bdi_13      0.068  0.059 0.077
## pre_bdi_11      0.047  0.040 0.053
## pre_bdi_9       0.034  0.029 0.040
## pre_bdi_15      0.033  0.027 0.039
## pre_bdi_17      0.033  0.027 0.039
## pre_bdi_8       0.031  0.024 0.037
## pre_bdi_4       0.024  0.017 0.031
## pre_bdi_5       0.023  0.018 0.028
## pre_bdi_19      0.017  0.011 0.022
## pre_bdi_12      0.014  0.008 0.020
## pre_bdi_16      0.011  0.007 0.016
## pre_bdi_20      0.011  0.006 0.015
## pre_bdi_21      0.009  0.005 0.012
## pre_bdi_10      0.006  0.003 0.010
## pre_bdi_18      0.006  0.002 0.010
## pre_bdi_3       0.002 -0.002 0.006
##
##
## Prediction Metrics
## (Results of 10-fold cross-validation repeated 10 times)
##                      Mean   S.E.   Min   Max
## Mean Absolute Error 0.749 0.0376 0.732 0.759
## Mean Cross Entropy  1.430 0.0540 1.410 1.440
## Mean Squared Error  1.020 0.1070 0.986 1.040
## Variance Explained  0.402 0.0539 0.392 0.423
## ========================================================
```

```
#importance plot
imp_drift_rf <- imp_drift_rf <- beset::importance(drift_rf)
fig_imp_drift_rf <- plot(imp_drift_rf, p_max = 21, labels = names_bdi)
fig_imp_drift_rf
```




```
#dependence plots
dep_drift_rf <- beset::dependence(drift_rf,
                        y_lab = "Drift rate for negative words",
                        x_lab = c("sadness", "pessimism", "failure",
                                  "loss pleasure", "guilt", "punishment",
                                  "self-dislike", "self-critical",
                                  "suicide", "crying", "agitation",
                                  "loss interest", "indecisive",
                                  "worthless", "loss energy", "sleep change",
                                  "irritability", "appetite", "concentration",
                                  "fatigue", "libido"))
fig_dep_drift_rf <- plot(dep_drift_rf, order = "import")
```




```
fig_dep_drift_rf
```

```
## TableGrob (4 x 5) "arrange": 17 grobs
##                z     cells    name                grob
## sadness        1 (1-1,2-2) arrange      gtable[layout]
## self-dislike   2 (1-1,3-3) arrange      gtable[layout]
## pessimism      3 (1-1,4-4) arrange      gtable[layout]
## worthless      4 (1-1,5-5) arrange      gtable[layout]
## punishment     5 (2-2,2-2) arrange      gtable[layout]
## indecisive     6 (2-2,3-3) arrange      gtable[layout]
## agitation      7 (2-2,4-4) arrange      gtable[layout]
## suicide        8 (2-2,5-5) arrange      gtable[layout]
## loss energy    9 (3-3,2-2) arrange      gtable[layout]
## irritability  10 (3-3,3-3) arrange      gtable[layout]
## self-critical 11 (3-3,4-4) arrange      gtable[layout]
## loss pleasure 12 (3-3,5-5) arrange      gtable[layout]
## guilt         13 (4-4,2-2) arrange      gtable[layout]
## concentration 14 (4-4,3-3) arrange      gtable[layout]
## loss interest 15 (4-4,4-4) arrange      gtable[layout]
## sleep change  16 (4-4,5-5) arrange      gtable[layout]
##               17 (1-4,1-1) arrange text[GRID.text.1118]
```

```
## Saving 6.5 x 4.5 in image
```

Random forest with reduced number of symptoms.

```
drift_rf_red <- beset_rf(v_negative ~ ., data=bdi[,c(2, 3, 9, 4, 16, 8, 15, 13)])
summary(drift_rf_red)
```

```
## Type of random forest: regression
## Number of trees: 500
## No. of variables tried at each split: 2
## =========================================================
## OOB estimate of % Var explained: 38.58
## CV estimate of % Var explained: 40.32
##
##            Importance   Min   Max
## pre_bdi_1       0.268 0.254 0.282
## pre_bdi_6       0.173 0.162 0.185
## pre_bdi_7       0.154 0.142 0.166
## pre_bdi_14      0.127 0.116 0.138
## pre_bdi_13      0.120 0.110 0.131
## pre_bdi_2       0.082 0.073 0.091
## pre_bdi_11      0.075 0.064 0.086
##
##
## Prediction Metrics
## (Results of 10-fold cross-validation repeated 10 times)
##                      Mean   S.E.   Min   Max
## Mean Absolute Error 0.756 0.0377 0.746 0.772
## Mean Cross Entropy  1.440 0.0558 1.430 1.460
## Mean Squared Error  1.040 0.1090 1.020 1.080
## Variance Explained  0.391 0.0460 0.370 0.406
## =========================================================
```

Compare full and reduced models.

```
#drift rate
drift_rf_full <- validate(drift_rf)
drift_rf_redu <- validate(drift_rf_red)

compare(drift_rf_full, drift_rf_redu)
```

```
##                     Model1 Model2    Delta 95% CI for Delta
## Mean Absolute Error  0.749  0.756  0.00709   [-0.035, 0.05]
## Mean Cross Entropy   1.430  1.440  0.00962  [-0.044, 0.063]
## Mean Squared Error   1.022  1.042  0.01988   [-0.089, 0.13]
## Variance Explained   0.402  0.391 -0.01162  [-0.075, 0.054]
```

Alternate models, elastic net.

```
#nested cross-validation to tune and then test
drift_elnet <- beset_elnet(v_negative ~ ., data=bdi[,c(2:23)], nest_cv = TRUE)
summary(drift_elnet)
```

```
##
## Results of nested 10-fold cross-validation repeated 10 times
## =========================================================
## Most conservative tuning parameters within
## 1 SE of best cross-validation Mean Squared Error:
##         Mean  S.E.        Range
## alpha  0.990 0.000   0.99 - 0.99
## lambda 0.247 0.013 0.239 - 0.256
##
##
## Non-zero coefficients ranked in order of importance:
```

```
##             Stnd.Coef.  S.E.   Min   Max
## pre_bdi_1       0.163 0.007 0.161 0.164
## pre_bdi_7       0.116 0.007 0.114 0.120
## pre_bdi_6       0.065 0.007 0.062 0.068
## pre_bdi_2       0.059 0.008 0.058 0.063
## pre_bdi_5       0.046 0.005 0.043 0.048
## pre_bdi_13      0.046 0.006 0.043 0.050
## pre_bdi_4       0.045 0.007 0.043 0.046
## pre_bdi_8       0.010 0.005 0.006 0.016
## pre_bdi_16      0.006 0.004 0.003 0.009
## pre_bdi_9       0.003 0.003 0.000 0.006
## pre_bdi_19      0.003 0.003 0.001 0.004
## pre_bdi_14      0.002 0.002 0.000 0.005
## pre_bdi_12      0.001 0.001 0.000 0.004
##
##
## Prediction Metrics:
##                  Variance Explained   S.E.   Min   Max
## Train Sample                  0.383 0.0082 0.379 0.390
## CV-Tune Holdout               0.337 0.0418 0.330 0.343
## CV-Test Holdout               0.341 0.0269 0.322 0.353
## ==========================================================
```

Compare elastic net and random forest.

```
sum_drift_en <- validate(drift_elnet)
sum_drift_rf <- validate(drift_rf)

compare(sum_drift_en, sum_drift_rf)
```

```
##                      Model1 Model2   Delta 95% CI for Delta
## Mean Absolute Error   0.803  0.749 -0.0540  [-0.11, 0.0035]
## Mean Cross Entropy    1.479  1.430 -0.0487    [-0.12, 0.02]
## Mean Squared Error    1.127  1.022 -0.1046    [-0.27, 0.04]
## Variance Explained    0.341  0.402  0.0612   [-0.027, 0.14]
```

SRET: Endorsements of negative words as self-descriptive.

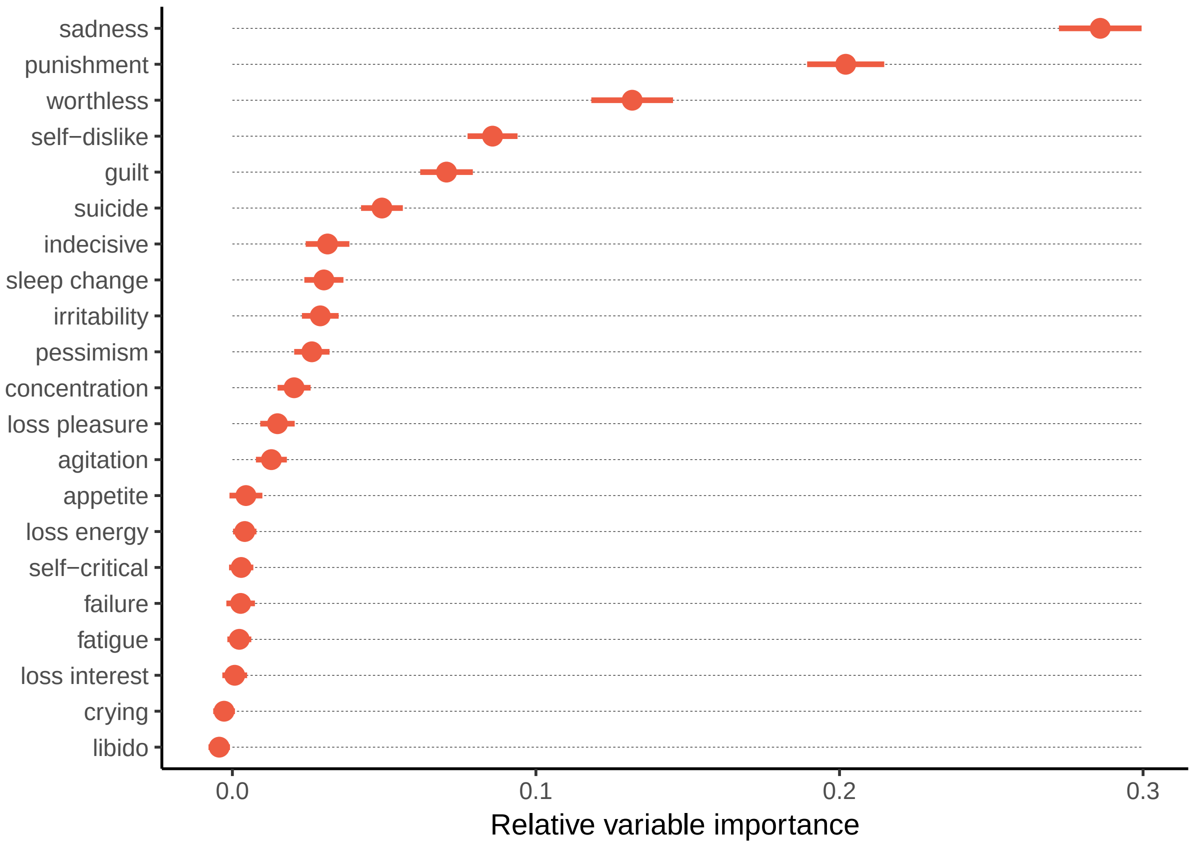
Random forest.

```
neg_end_rf <- beset_rf(num.neg.endorsed ~ ., data=bdi[,c(1, 3:23)])
summary(neg_end_rf)
```

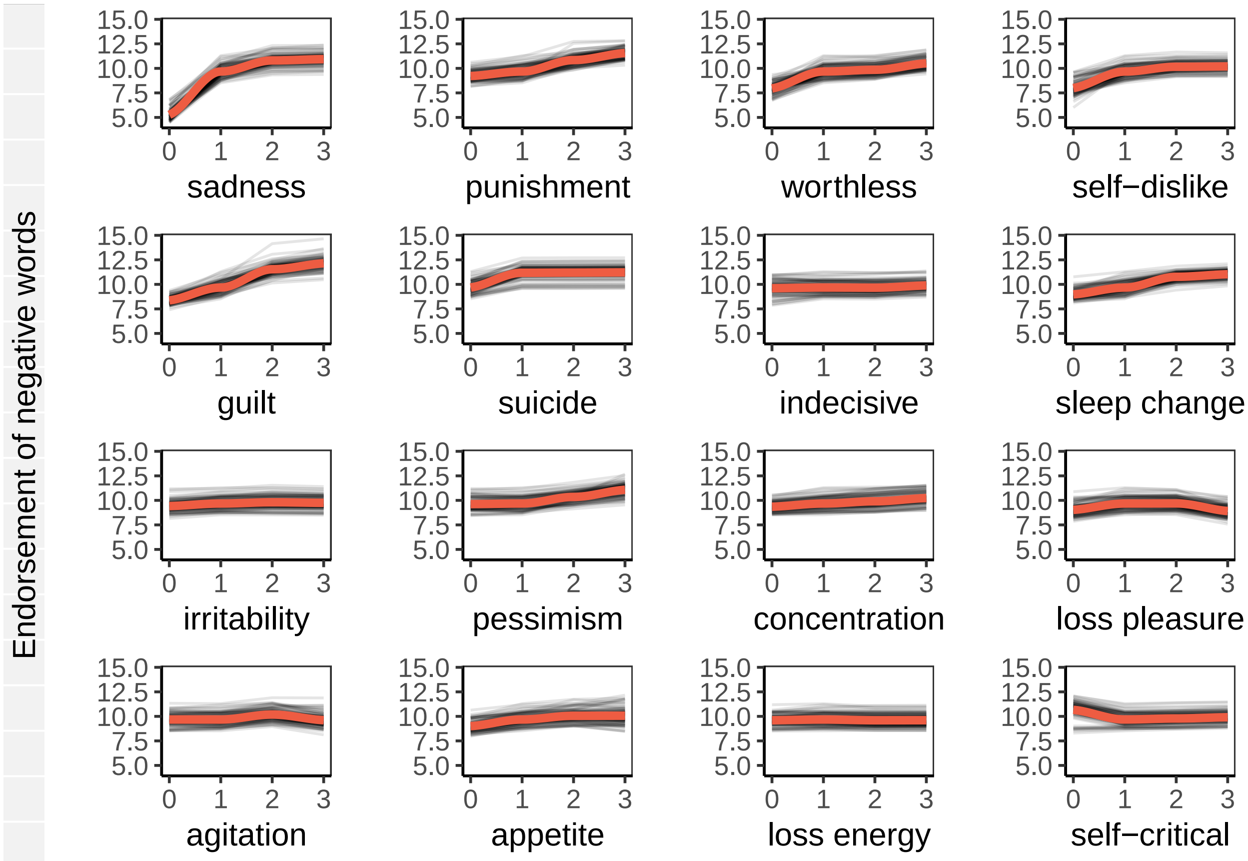
```
## Type of random forest: regression
## Number of trees: 500
## No. of variables tried at each split: 7
## ==========================================================
## OOB estimate of % Var explained: 44.56
## CV estimate of % Var explained: 45.68
##
##             Importance    Min    Max
## pre_bdi_1        0.286  0.272  0.300
## pre_bdi_6        0.202  0.189  0.215
## pre_bdi_14       0.132  0.118  0.145
```

```
## pre_bdi_7       0.086  0.077  0.094
## pre_bdi_5       0.071  0.062  0.079
## pre_bdi_9       0.049  0.042  0.056
## pre_bdi_13      0.031  0.024  0.039
## pre_bdi_16      0.030  0.024  0.037
## pre_bdi_17      0.029  0.023  0.035
## pre_bdi_2       0.026  0.020  0.032
## pre_bdi_19      0.020  0.015  0.026
## pre_bdi_4       0.015  0.009  0.021
## pre_bdi_11      0.013  0.008  0.018
## pre_bdi_18      0.004 -0.001  0.010
## pre_bdi_15      0.004  0.000  0.008
## pre_bdi_8       0.003 -0.001  0.007
## pre_bdi_3       0.003 -0.002  0.007
## pre_bdi_20      0.002 -0.002  0.006
## pre_bdi_12      0.001 -0.003  0.005
## pre_bdi_10     -0.003 -0.006  0.001
## pre_bdi_21     -0.004 -0.008 -0.001
##
##
## Prediction Metrics
## (Results of 10-fold cross-validation repeated 10 times)
##                       Mean   S.E.    Min   Max
## Mean Absolute Error  3.450 0.1930  3.380  3.51
## Mean Cross Entropy   2.930 0.0559  2.910  2.95
## Mean Squared Error  20.600 2.2600 19.900 21.30
## Variance Explained   0.453 0.0560  0.433  0.47
## =======================================================
```

```r
#importance plot
imp_end_rf <- beset::importance(neg_end_rf)
fig_imp_end_rf <- plot(imp_end_rf, p_max = 21, labels = names_bdi)
fig_imp_end_rf
```




```r
#dependence plots
dep_end_rf <- beset::dependence(neg_end_rf,
                               y_lab = "Endorsement of negative words",
                               x_lab = c("sadness", "pessimism", "failure",
                                         "loss pleasure", "guilt", "punishment",
                                         "self-dislike", "self-critical",
                                         "suicide", "crying", "agitation",
                                         "loss interest", "indecisive",
                                         "worthless", "loss energy", "sleep change",
                                         "irritability", "appetite", "concentration",
                                         "fatigue", "libido"))
fig_dep_end_rf <- plot(dep_end_rf, order = "import")
```




```
fig_dep_end_rf
```

```
## TableGrob (4 x 5) "arrange": 17 grobs
##                z     cells    name               grob
## sadness        1 (1-1,2-2) arrange      gtable[layout]
## punishment     2 (1-1,3-3) arrange      gtable[layout]
## worthless      3 (1-1,4-4) arrange      gtable[layout]
## self-dislike   4 (1-1,5-5) arrange      gtable[layout]
## guilt          5 (2-2,2-2) arrange      gtable[layout]
## suicide        6 (2-2,3-3) arrange      gtable[layout]
## indecisive     7 (2-2,4-4) arrange      gtable[layout]
## sleep change   8 (2-2,5-5) arrange      gtable[layout]
## irritability   9 (3-3,2-2) arrange      gtable[layout]
## pessimism     10 (3-3,3-3) arrange      gtable[layout]
## concentration 11 (3-3,4-4) arrange      gtable[layout]
## loss pleasure 12 (3-3,5-5) arrange      gtable[layout]
## agitation     13 (4-4,2-2) arrange      gtable[layout]
## appetite      14 (4-4,3-3) arrange      gtable[layout]
## loss energy   15 (4-4,4-4) arrange      gtable[layout]
## self-critical 16 (4-4,5-5) arrange      gtable[layout]
##               17 (1-4,1-1) arrange text[GRID.text.1967]
```

```
## Saving 6.5 x 4.5 in image
```

Random forest with reduced number of symptoms.

```
neg_end_rf_red <- beset_rf(num.neg.endorsed ~ ., data=bdi[,c(1, 3, 8, 16, 9, 7, 11)])
summary(neg_end_rf_red)
```

```
## Type of random forest: regression
## Number of trees: 500
## No. of variables tried at each split: 2
## =========================================================
## OOB estimate of % Var explained: 46.34
## CV estimate of % Var explained: 47.19
##
##            Importance   Min   Max
## pre_bdi_1       0.408 0.396 0.420
## pre_bdi_6       0.247 0.235 0.260
## pre_bdi_7       0.100 0.092 0.108
## pre_bdi_5       0.095 0.087 0.103
## pre_bdi_14      0.094 0.085 0.102
## pre_bdi_9       0.056 0.049 0.063
##
##
## Prediction Metrics
## (Results of 10-fold cross-validation repeated 10 times)
##                       Mean   S.E.    Min   Max
## Mean Absolute Error  3.470 0.1900  3.420  3.55
## Mean Cross Entropy   2.920 0.0551  2.900  2.94
## Mean Squared Error  19.900 2.1600 19.200 20.80
## Variance Explained   0.469 0.0554  0.447  0.49
## =========================================================
```

Compare full and reduced models.

```
end_rf_full <- validate(neg_end_rf)
end_rf_redu <- validate(neg_end_rf_red)

compare(end_rf_full, end_rf_redu)
```

```
##                     Model1 Model2   Delta 95% CI for Delta
## Mean Absolute Error  3.451  3.469  0.0181    [-0.17, 0.21]
## Mean Cross Entropy   2.931  2.915 -0.0153  [-0.061, 0.029]
## Mean Squared Error  20.560 19.944 -0.6167      [-2.5, 1.2]
## Variance Explained   0.453  0.469  0.0164  [-0.032, 0.066]
```

Alternative model, best subsets regression with negative binomial distribution.

```
#this models takes a while to run, so will not run it here
sret_negbin <- beset_glm(num.neg.endorsed ~ ., data=bdi[,c(1, 3:23)],
                         p_max = 10, family = "negbin")

saveRDS(sret_negbin, file = rds_file)
```

```
sret_negbin <- readRDS(file = rds_file)

summary(sret_negbin, oneSE = FALSE)
```

```
##
## =========================================================
## Best Model:
##   ~ pre_bdi_1 + pre_bdi_5 + pre_bdi_6 + pre_bdi_7 + pre_bdi_9 + pre_bdi_13 + pre_bdi_16
##
## 82 Nearly Equivalent Models:
##   ~ pre_bdi_1 + pre_bdi_2 + pre_bdi_4 + pre_bdi_5 + pre_bdi_6 + pre_bdi_9 + pre_bdi_16
```

```
##   ~ pre_bdi_1 + pre_bdi_2 + pre_bdi_5 + pre_bdi_6 + pre_bdi_7 + pre_bdi_9 + pre_bdi_11
##   ~ pre_bdi_1 + pre_bdi_2 + pre_bdi_5 + pre_bdi_6 + pre_bdi_7 + pre_bdi_9 + pre_bdi_13
##   ~ pre_bdi_1 + pre_bdi_2 + pre_bdi_5 + pre_bdi_6 + pre_bdi_7 + pre_bdi_9 + pre_bdi_16
##   ~ pre_bdi_1 + pre_bdi_2 + pre_bdi_5 + pre_bdi_6 + pre_bdi_7 + pre_bdi_9 + pre_bdi_19
##   ...
##    + 77 more
##   ...
##
## Coefficients:
##             Estimate
## (Intercept)  1.04300
## pre_bdi_1    0.33390
## pre_bdi_5    0.11770
## pre_bdi_6    0.15900
## pre_bdi_7    0.10090
## pre_bdi_9    0.20490
## pre_bdi_13   0.07407
## pre_bdi_16   0.13160
##
## (Dispersion parameter for Negative Binomial(3.3748) family taken to be 1)
##
## Log-likelihood: -609.6 on 9 Df
## AIC: 1237.1
##
## Number of Fisher Scoring iterations: 1
##
## Train-sample R-squared = 0.39
## Cross-validated R-squared =  0.35
## =========================================================
```

```r
summary(sret_negbin)
```

```
##
## =========================================================
## Best Model:
##   ~ pre_bdi_1 + pre_bdi_6
##
## Nearly Equivalent Model:
##   ~ pre_bdi_1 + pre_bdi_5
##
## Coefficients:
##             Estimate
## (Intercept)   1.3040
## pre_bdi_1     0.5963
## pre_bdi_6     0.2510
##
## (Dispersion parameter for Negative Binomial(2.5903) family taken to be 1)
##
## Log-likelihood: -624.8 on 4 Df
## AIC: 1257.6
##
## Number of Fisher Scoring iterations: 1
##
## Train-sample R-squared = 0.31
## Cross-validated R-squared =  0.29
```

```
## ========================================================
```

Dot-Probe: Reaction Time Metrics

Random forest for traditional dot-probe metric.

```
rt_bias_rf <- beset_rf(dp_bias ~ ., data=dp_bias[,c(1:21, 24)])
summary(rt_bias_rf)
```

```
## Type of random forest: regression
## Number of trees: 500
## No. of variables tried at each split: 7
## ========================================================
## OOB estimate of % Var explained: -13.25
## CV estimate of % Var explained: -9.47
##
##            Importance    Min    Max
## pre_bdi_12      0.246  0.209  0.283
## pre_bdi_18      0.117  0.089  0.145
## pre_bdi_1       0.091  0.062  0.119
## pre_bdi_5       0.085  0.060  0.111
## pre_bdi_17      0.077  0.052  0.102
## pre_bdi_4       0.064  0.037  0.092
## pre_bdi_8       0.052  0.033  0.071
## pre_bdi_19      0.044  0.024  0.065
## pre_bdi_6       0.041  0.021  0.060
## pre_bdi_14      0.038  0.018  0.058
## pre_bdi_2       0.035  0.023  0.047
## pre_bdi_11      0.030  0.013  0.047
## pre_bdi_15      0.028  0.012  0.043
## pre_bdi_20      0.020  0.008  0.033
## pre_bdi_21      0.018 -0.001  0.038
## pre_bdi_7       0.016  0.000  0.032
## pre_bdi_3       0.011 -0.007  0.029
## pre_bdi_16      0.009 -0.008  0.026
## pre_bdi_13      0.002 -0.017  0.022
## pre_bdi_10     -0.009 -0.025  0.008
## pre_bdi_9      -0.015 -0.027 -0.004
##
##
## Prediction Metrics
## (Results of 10-fold cross-validation repeated 10 times)
##                          Mean     S.E.       Min       Max
## Mean Absolute Error  0.022100 0.001310  0.021600  0.022700
## Mean Cross Entropy  -2.090000 0.075100 -2.100000 -2.060000
## Mean Squared Error   0.000901 0.000158  0.000878  0.000942
## Variance Explained  -0.124000 0.063900 -0.174000 -0.094600
## ========================================================
```

Random forest for TLBS bias towards sad stimuli.

```
tlbs_bias_rf <- beset_rf(mean_dp_toward ~ ., data=dp_bias[,c(1:21, 27)])
summary(tlbs_bias_rf)
```

```
## Type of random forest: regression
## Number of trees: 500
## No. of variables tried at each split: 7
## =========================================================
## OOB estimate of % Var explained: -6.45
## CV estimate of % Var explained: -0.37
##
##            Importance    Min    Max
## pre_bdi_13      0.265  0.237  0.292
## pre_bdi_15      0.127  0.104  0.150
## pre_bdi_8       0.110  0.090  0.131
## pre_bdi_19      0.086  0.067  0.105
## pre_bdi_1       0.081  0.062  0.100
## pre_bdi_20      0.064  0.040  0.088
## pre_bdi_14      0.063  0.044  0.082
## pre_bdi_11      0.048  0.032  0.063
## pre_bdi_3       0.043  0.024  0.063
## pre_bdi_12      0.036  0.021  0.052
## pre_bdi_10      0.031  0.017  0.045
## pre_bdi_17      0.028  0.012  0.043
## pre_bdi_7       0.024  0.007  0.041
## pre_bdi_2       0.014 -0.002  0.031
## pre_bdi_4       0.012 -0.002  0.026
## pre_bdi_16      0.007 -0.012  0.025
## pre_bdi_6       0.003 -0.012  0.017
## pre_bdi_9      -0.004 -0.012  0.005
## pre_bdi_18     -0.004 -0.020  0.012
## pre_bdi_5      -0.007 -0.028  0.013
## pre_bdi_21     -0.027 -0.042 -0.012
##
##
## Prediction Metrics
## (Results of 10-fold cross-validation repeated 10 times)
##                           Mean     S.E.       Min      Max
## Mean Absolute Error  0.022500 1.08e-03  0.022000  0.02290
## Mean Cross Entropy  -2.120000 5.08e-02 -2.130000 -2.09000
## Mean Squared Error   0.000852 7.93e-05  0.000819  0.00090
## Variance Explained  -0.033200 7.72e-02 -0.090700  0.00757
## =========================================================
```

Random forest for eye gaze bias.

```
gz_bias_rf <- beset_rf(gaze_bias ~ ., data=dp_bias[,c(1:21, 30)])
summary(gz_bias_rf)
```

```
## Type of random forest: regression
## Number of trees: 500
## No. of variables tried at each split: 7
## =========================================================
## OOB estimate of % Var explained: -17.83
## CV estimate of % Var explained: -11.91
##
##            Importance    Min    Max
## pre_bdi_12      0.206  0.178  0.235
## pre_bdi_15      0.155  0.129  0.182
```

```
## pre_bdi_4       0.142  0.117  0.166
## pre_bdi_1       0.106  0.085  0.126
## pre_bdi_7       0.078  0.056  0.100
## pre_bdi_18      0.072  0.052  0.091
## pre_bdi_10      0.070  0.052  0.088
## pre_bdi_17      0.062  0.046  0.077
## pre_bdi_20      0.056  0.039  0.072
## pre_bdi_16      0.055  0.039  0.071
## pre_bdi_19      0.039  0.018  0.059
## pre_bdi_2       0.037  0.023  0.051
## pre_bdi_3       0.019  0.003  0.035
## pre_bdi_11      0.018  0.003  0.033
## pre_bdi_14      0.006 -0.008  0.019
## pre_bdi_8      -0.009 -0.025  0.008
## pre_bdi_5      -0.011 -0.025  0.003
## pre_bdi_6      -0.016 -0.028 -0.005
## pre_bdi_13     -0.017 -0.031 -0.003
## pre_bdi_9      -0.023 -0.033 -0.013
## pre_bdi_21     -0.042 -0.057 -0.026
##
##
## Prediction Metrics
## (Results of 10-fold cross-validation repeated 10 times)
##                           Mean     S.E.      Min      Max
## Mean Absolute Error  0.03310 0.001400  0.03270  0.03390
## Mean Cross Entropy  -1.75000 0.046100 -1.76000 -1.73000
## Mean Squared Error   0.00178 0.000163  0.00174  0.00183
## Variance Explained  -0.15200 0.051500 -0.18300 -0.12500
## =========================================================
```

Random forest for percentage of trials towards sad stimuli.

```r
pct_gz_bias <- beset_rf(pct_gaze_toward ~ ., data=dp_bias[,c(1:21, 34)])
summary(pct_gz_bias)
```

```
## Type of random forest: regression
## Number of trees: 500
## No. of variables tried at each split: 7
## =========================================================
## OOB estimate of % Var explained: -11.86
## CV estimate of % Var explained: -7.84
##
##             Importance    Min    Max
## pre_bdi_1       0.174  0.154  0.193
## pre_bdi_4       0.167  0.139  0.195
## pre_bdi_18      0.091  0.063  0.119
## pre_bdi_16      0.072  0.051  0.093
## pre_bdi_2       0.071  0.055  0.087
## pre_bdi_10      0.068  0.050  0.087
## pre_bdi_15      0.059  0.037  0.082
## pre_bdi_3       0.048  0.030  0.067
## pre_bdi_13      0.043  0.021  0.065
## pre_bdi_7       0.037  0.019  0.055
## pre_bdi_14      0.037  0.018  0.055
## pre_bdi_20      0.034  0.014  0.054
```

```
## pre_bdi_11      0.032  0.015  0.048
## pre_bdi_12      0.025  0.004  0.046
## pre_bdi_21      0.024  0.006  0.042
## pre_bdi_19      0.023  0.004  0.042
## pre_bdi_17      0.022  0.007  0.037
## pre_bdi_8       0.018  0.000  0.036
## pre_bdi_9      -0.002 -0.013  0.009
## pre_bdi_6      -0.015 -0.028 -0.001
## pre_bdi_5      -0.030 -0.045 -0.015
##
##
## Prediction Metrics
## (Results of 10-fold cross-validation repeated 10 times)
##                         Mean    S.E.      Min      Max
## Mean Absolute Error  0.05230 0.00181  0.05150  0.05350
## Mean Cross Entropy  -1.31000 0.03700 -1.32000 -1.29000
## Mean Squared Error   0.00431 0.00032  0.00419  0.00448
## Variance Explained  -0.11400 0.05800 -0.15900 -0.08500
## =======================================================
```

Bivariate correlations.

```
library(apaTables)
apa.cor.table(dp_bias[,c(24, 27, 30, 34, 40)],
              filename=table2,
              table.number=2,
              landscape = FALSE)
```

```
##
##
## Table 2
##
## Means, standard deviations, and correlations with confidence intervals
##
##
##   Variable           M     SD    1           2           3
##   1. dp_bias         -0.00 0.03
##
##   2. mean_dp_toward  0.11  0.03  .31**
##                                  [.17, .44]
##
##   3. gaze_bias       0.01  0.04  .32**       .10
##                                  [.18, .45]  [-.05, .25]
##
##   4. pct_gaze_toward 0.50  0.06  .16*        .03         .72**
##                                  [.01, .31]  [-.12, .18] [.64, .79]
##
##   5. bdi_total       18.21 11.73 -.00        .22**       -.04
##                                  [-.15, .15] [.07, .36]  [-.19, .11]
##
##   4
##
##
##
##
```

```
##
##
##
##
##
##
##
##   -.11
##   [-.26, .04]
##
##
## Note. M and SD are used to represent mean and standard deviation, respectively.
## Values in square brackets indicate the 95% confidence interval.
## The confidence interval is a plausible range of population correlations
## that could have caused the sample correlation (Cumming, 2014).
## * indicates p < .05. ** indicates p < .01.
##
```

Cross-validation for the linear model testing the association between TLBS bias toward sad stimuli and depression symptom total score.

```r
bdi_cv <- lm(mean_dp_toward ~ bdi_total, data = dp_bias)
summary(bdi_cv)
```

```
##
## Call:
## lm(formula = mean_dp_toward ~ bdi_total, data = dp_bias)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.052833 -0.020716 -0.002109  0.016192  0.093779
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.0990968  0.0040607   24.40  < 2e-16 ***
## bdi_total   0.0005403  0.0001876    2.88  0.00451 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.02819 on 163 degrees of freedom
## Multiple R-squared:  0.04842,    Adjusted R-squared:  0.04258
## F-statistic: 8.294 on 1 and 163 DF,  p-value: 0.004513
```

```r
validate(bdi_cv)
```

```
## Mean predictive performance under 10-fold cross-validation
## (with min-max range over 10 repetitions)
##                          Mean     S.E.       Min       Max
## Mean Absolute Error  0.022200 8.94e-04  0.022100  0.022300
## Mean Cross Entropy  -2.150000 4.66e-02 -2.150000 -2.140000
## Mean Squared Error   0.000801 7.32e-05  0.000792  0.000817
## Variance Explained   0.028900 3.91e-02  0.009390  0.040100
```