# Codebook explaining all variables

For Kenneth Benoit, Kevin Munger, and Arthur Spirling, “Measuring and Explaining Political Sophistication Through Textual Complexity”, *American Journal of Political Science*.

Please address any questions to kbenoit@lse.ac.uk.

## 1. Crowdflower input files

**Files:** `CF_input_921916.csv/tab`, `CF_input_952737.csv/tab`, and `CF_input_999866.csv/tab`

**Guide:** These are the snippet comparisons that were uploaded to Crowdflower to be coded. Each line contains two snippets to be compared, drawn from the State of the Union address corpus. The first CSV was a pilot. The second CSV is a larger version of the pilot, and contained snippets dating back to the first SOTU. The third CSV refined our comparisons to only snippets of very similar length and from after WWII.

All of the .csv files are encoded in UTF-8.

**Variables:**

- `docID1` - The document ID for the text from which first snippet was taken, e.g. “Bush-2005”
- `snippetID1` - A unique numeric snippet identifier for snippet 1
- `text1` - the plain text of snippet 1
- `docID2` - The document ID for the text from which first snippet was taken, e.g. “Bush-2005”
- `snippetID2` - A unique numeric snippet identifier for snippet 2
- `text2` - the plain text of snippet 2
- `X_golden` - a logical value (`TRUE` or `FALSE`) indicating whether the snippet pair was a “gold” question for which we supplied an answer
- `easier_gold` - a value of 1 or 2 indicating which of the gold pairing was the correct answer of “easier” (for gold questions only)
- `easier_gold_reason` - a plain text explanation for the gold question correct answer, e.g. “Text A is”easier" to read because it contains some combination of shorter sentences, more commonly used and more easily understood terms, and is generally less complicated and easier to read and grasp its point."
- `screener` - a logical value `TRUE` indicating whether the question was a “screener” (a special gold question with embedded instructions to the coder as to how to answer the task) or blank if the question was not a screener.

## 2. Crowdflower output files

**Files:** `CF_output_f921916.csv/tab`, `CF_output_f952737.csv/tab`, and `CF_output_f999866.csv/tab`

**Guide:** These are the coded snippet comparisons that were downloaded from Crowdflower after the crowd-sourced job was completed.

**Variables:**

- `_unit_id` - a unique numeric identifier for the unit, assigned by Crowdflower (e.g. 979545534)
- `_created_at` - a POSIX date/time stamp, e.g. “6/17/2016 19:28:11” indicating when the unit was created
- `_golden` - a Boolean (`true` or `false`) indicating whether the unit was a gold question
- `_id` - a unique numeric identifier for the answer, assigned by Crowdflower (e.g. 2025513496 )
- `_missed` - a Boolean (`true` or blank) indicating whether the crowd worker missed a gold question
- `_started_at` - a POSIX date/time stamp, e.g. “6/17/2016 19:28:11” indicating when the task was started
- `_tainted` - a Boolean (`true` or `false`) indicating whether the answer was “tainted” because the worker missed too many screening questions. Because we excluded tainted answers, in our data, all values of this variable are `false`.
- `_channel` - The crowd-sourcing channel (website) on which the task was answered.

- `_trust` - The “trust” score for the respondent, as computed by Crowdflower. Our minimum was 0.60 and our mean answer had a trust value of around 0.86.
- `_worker_id` - a unique numeric identifier for the worker, e.g. 34616922
- `_country` - a three-digit code indicating the country in which the worker answered the question
- `_region` - regional identifier, if available, for the worker
- `_city` - city identifier for the worker
- `_ip` - IP address assigned from the worker’s Internet Service Provider (ISP), used to identify the region and city above. Revealing this IP address was agreed by each respondent when they agreed to the terms and conditions of the task, through the Crowdflower platform.
- `easier` - 1 or 2, indicating which snippet was answered as being easier by the worker. This is the core data we used in scoring the snippets.
- `orig__golden` - `TRUE` if the pair was a gold question, or blank if not.
- `docid1`, `docid2`, `easier_gold`, `easier_gold_reason`, `screener`, `snippetid1`, `snippetid2`, `text1`, `text2` - these are the same as in the input .csv files above.

## 3. Sentence-level covariates for rated sentences

**Files:** `job999866covars.rda`, `job999866covars_chameleons.rda`

**Guide:** These files contain the covariate information for the snippets that have been labeled. The information in the two files are the same, but the ‘chameleons’ version is in the format used by the **BradleyTerry2** R package.

**Variables:**

- `snippetid` - (in `job999866covars`) numeric identifier for the snippet
- `text` - (in `job999866covars`) text of the snippet
- `doc_id` - (in `job999866covars_chameleons`) document identifier from which the snippet is taken, corresponding to `docID1` and `docID2` in the Crowdfloer input files above
- `meanSentenceLength` - Mean sentence length in words in the text
- `meanWordSyllables` - Mean syllables per word in the text
- `Dale.Chall.old` - The original Dale-Chall Readability score (see Dale, E. and Chall, J.S. 1948. “A Formula for Predicting Readability: Instructions.” *Educational Research Bulletin*, 37-54.)
- `Flesch` - Flesch Reading Ease score for the snippet (see Flesch, R. 1948. “A New Readability Yardstick”. *Journal of Applied Psychology*, 32(3), 221.)
- `Dale.Chall` - the New Dale-Chall Readability score (see Chall, J.S. and Dale, E. 1995. *Readability Revisited: The New Dale-Chall Readability Formula.* Brookline Books.)
- `FOG` - Gunning’s Fog Index (see Gunning, R. 1952. *The Technique of Clear Writing*. New York: McGraw-Hill.)
- `SMOG` - Simple Measure of Gobbledygook (SMOG) (see McLaughlin, G.H. 1969. “SMOG Grading: A New Readability Formula.” *Journal of Reading*, 12(8), 639-646.).
- `Spache` - Spache’s (1952) Readability Measure (see Spache, G. 1953. “A new readability formula for primary-grade reading materials.” *The Elementary School Journal*, 53, 410–413.)
- `Coleman.Liau` - the Coleman-Liau Estimated Cloze Percent (ECP) (see Coleman, M. & Liau, T.L. 1975. “A Computer Readability Formula Designed for Machine Scoring”. *Journal of Applied Psychology*, 60(2), 283.).
- `W` - number of words in the snippet
- `St` - number of sentences in the snippet (computed by **quanteda**)
- `C` - number of characters in the snippet
- `Sy` - number of syllables in the snippet
- `W3Sy` - Words in the text with at least 3 syllables
- `W2Sy` - Words in the text with 2 syllables
- `W_1Sy` - Words in the text with 1 syllable
- `W6C` - Words in the text with at least 6 characters
- `W7C` - Words in the text with at least 7 characters
- `Wlt3Sy` - Words in the text with fewer than 3 syllables

- W_wl.Dale.Chall - Words in the text matching the Dale-Chall list (see Dale, Edgar and Jeanne Chall. 1948. “A Formula for Predicting Readability.” *Educational Research Bulletin* 27(1): 11–20).
- meanWordChars - Mean characters per word in the text
- meanSentenceChars - Mean characters per sentence in the text
- meanSentenceSyllables - Mean sentence length in syllables in the text
- brown_mean - Brown corpus baseline usage, average of the words in the text
- brown_min - Brown corpus baseline usage, minimum of the words in the text
- google_mean_2000 - Google Books baseline usage for the year 2000, average of the words in the text
- google_min_2000 - Google Books baseline usage for the year 2000, minimum of the words in the text
- pr_sentence - Number of sentences per character in the text
- pr_noun - Proportion of nouns in the text
- pr_verb - Proportion of verbs in the text
- pr_adjective - Proportion of adjectives in the text
- pr_adverb - Proportion of adverbs in the text
- pr_clause - Average number of subordinate clauses in the text
- n_namedentities - number of named entities (as parsed by spaCy) in the snippet
- n_sentence - same as St (but computed by spaCy)
- n_noun - number of nouns in the snippet
- n_verb - number of verbs in the snippet
- n_adjective - number of adjectives in the snippet
- n_adverb - number of adverbs in the snippet
- n_clause - number of subordinate clauses in the snippet
- ntoken - total tokens in the snippet, same as W but computed by spaCy

## 4. Unstructured Bradley-Terry model estimates

**Files:** BT_unstructured_brT_abilities.rda, BT_unstructured_brF_abilities.rda

**Guide:** These files are model objects fitted by the **BradleyTerry2** package, and contain rankings of each snippet from easiest to hardest. The brT file was run with bias reduction, while brF was not. These are intermediate objects, and are best analyzed with code farther down the analysis pipeline.

## 5. Fitted random forest files

**Files:** rf_model_bias_reduced.rda, rf_model_non_bias_reduced.rda

**Guide:** These files are output by the **randomForest** package, and contain estimates of the predictive ability of each of the above listed covariates on the difficulty ratings of the snippets, producing a list of the most predictive variables that we then include in the structured Bradley-Terry model. These are intermediate objects, and are best analyzed with code farther down the analysis pipeline.

```
library("randomForest")
```

```
## randomForest 4.6-14
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
load("rf_model_bias_reduced.rda")
print(mod_bias_reduced)
```

```
##
## Call:
##  randomForest(x = X, y = yy, ntree = 1000)
##                Type of random forest: regression
##                      Number of trees: 1000
## No. of variables tried at each split: 7
##
##           Mean of squared residuals: 18.0962
##                     % Var explained: -0.71
```

```r
load("rf_model_non_bias_reduced.rda")
print(mod_non_bias_reduced)
```

```
##
## Call:
##  randomForest(x = X2, y = yy2, ntree = 1000)
##                Type of random forest: regression
##                      Number of trees: 1000
## No. of variables tried at each split: 7
##
##           Mean of squared residuals: 1535.454
##                     % Var explained: -0.97
```

## 6. Predictors for new texts

**Files:** `sotu_covars.rda`

**Guide:** This file contains the covariate information for the average covariate values of paragraphs that span the entirety of each SOTU address under study. The variables included in the analysis are defined and explained in Table 1 of the manuscript, and match those from `job999866covars.rda` explained above.

## 7. New data on paragraphs from the SOTU corpus

**Files:** `data_corpus_sotuparagraphs.rda`

**Guide:** This file consists of the SOTU addresses parsed into paragraphs, with document-level information fitted using the `predict()` methods from the **BradleyTerry2** package for the structured model.

```r
library("quanteda")
```

```
## Package version: 1.4.0

## Parallel computing: 2 of 12 threads used.

## See https://quanteda.io for tutorials and examples.

##
## Attaching package: 'quanteda'

## The following object is masked from 'package:utils':
##
##     View
```

```r
load("data_corpus_sotuparagraphs.rda")
summary(docvars(data_corpus_sotuparagraphs))
```

```
##   FirstName          President              Date
##  Length:22345       Length:22345       Min.   :1790-01-08
##  Class :character   Class :character   1st Qu.:1880-12-06
##  Mode  :character   Mode  :character   Median :1927-12-06
##                                        Mean   :1922-07-13
##                                        3rd Qu.:1974-01-30
##                                        Max.   :2018-01-30
##     delivery        type                        party
##  spoken : 8058   other:  434   Democratic           :10595
##  written:14287   SOTU :21911   Democratic-Republican: 1061
##                                Federalist           :   85
##                                Independent          :  219
##                                Republican           : 9900
##                                Whig                 :  485
##      Flesch         lambda_2000        prob_2000       scaled_2000
##  Min.   :-229.96   Min.   :-19.154   Min.   :0.0000   Min.   :-905.107
##  1st Qu.:  23.45   1st Qu.: -3.980   1st Qu.:0.1414   1st Qu.:  -6.796
##  Median :  37.13   Median : -3.259   Median :0.2529   Median :  35.876
##  Mean   :  36.21   Mean   : -3.447   Mean   :0.2688   Mean   :  24.788
##  3rd Qu.:  50.81   3rd Qu.: -2.670   3rd Qu.:0.3790   3rd Qu.:  70.779
##  Max.   : 120.20   Max.   :  9.897   Max.   :1.0000   Max.   : 814.773
##  lambda_lo_2000    lambda_hi_2000     prob_lo_2000      prob_hi_2000
##  Min.   :-27.049   Min.   :-19.154   Min.   :0.00000   Min.   :0.0000
##  1st Qu.: -4.616   1st Qu.: -3.394   1st Qu.:0.08024   1st Qu.:0.2281
##  Median : -3.710   Median : -2.706   Median :0.17761   Median :0.3703
##  Mean   : -3.944   Mean   :    Inf   Mean   :0.20921   Mean   :0.3782
##  3rd Qu.: -3.001   3rd Qu.: -2.130   3rd Qu.:0.30484   3rd Qu.:0.5090
##  Max.   :  9.897   Max.   :    Inf   Max.   :0.99999   Max.   :1.0000
##  scaled_lo_2000      scaled_hi_2000     lambda_local       prob_local
##  Min.   :-1372.530   Min.   :-905.11   Min.   :-19.155   Min.   :0.0000
##  1st Qu.:  -44.408   1st Qu.:  27.92   1st Qu.: -3.981   1st Qu.:0.1413
##  Median :    9.233   Median :  68.66   Median : -3.259   Median :0.2529
##  Mean   :   -4.672   Mean   :    Inf   Mean   : -3.448   Mean   :0.2689
##  3rd Qu.:   51.195   3rd Qu.: 102.77   3rd Qu.: -2.668   3rd Qu.:0.3794
##  Max.   :  814.773   Max.   :    Inf   Max.   :  7.204   Max.   :0.9999
##   scaled_local      lambda_lo_local   lambda_hi_local   prob_lo_local
##  Min.   :-905.185   Min.   :-26.700   Min.   :-19.155   Min.   :0.00000
##  1st Qu.:  -6.856   1st Qu.: -4.615   1st Qu.: -3.395   1st Qu.:0.08033
##  Median :  35.882   Median : -3.708   Median : -2.699   Median :0.17771
##  Mean   :  24.735   Mean   : -3.945   Mean   :    Inf   Mean   :0.20924
##  3rd Qu.:  70.870   3rd Qu.: -2.998   3rd Qu.: -2.121   3rd Qu.:0.30544
##  Max.   : 655.341   Max.   :  7.204   Max.   :    Inf   Max.   :0.99992
##  prob_hi_local    scaled_lo_local     scaled_hi_local
##  Min.   :0.0000   Min.   :-1351.876   Min.   :-905.18
##  1st Qu.:0.2282   1st Qu.:  -44.378   1st Qu.:  27.88
##  Median :0.3718   Median :    9.301   Median :  69.09
##  Mean   :0.3779   Mean   :   -4.679   Mean   :    Inf
##  3rd Qu.:0.5118   3rd Qu.:   51.365   3rd Qu.: 103.25
##  Max.   :1.0000   Max.   :  655.341   Max.   :    Inf
```

## 8. Model results predicted on new texts

**Files:** `bootstrap_results_AJPSR2.rda`, `Best_Model_Results_list.rda`, `BT_best.rda`

**Guide:** These files contain the results of the structured Bradley-Terry models.

**Variables:**

- `Best_Model_Results_list.rda` is a list object containing four variations:
  - `BT_basic_Flesch` is a straightforward replication of the original Flesch model.

  - `BT_optimal_Flesch` uses the same covariates as the original Flesch model but with optimized covariate weights.
  - `BT_basic_RF` contains only the variables estimated to be the most predictive by the randomForest model.
  - `BT_best` (also saved as `BT_best.rda`) is the model with the best performance, which is the same as `BT_basic_RF` with the addition of the theoretically-relevant 'meanWordChars' covariate.
- `bootstrap_results_AJPSR2.rda` contains the 500-iteration bootstrap estimates for each of these four models, allowing us to construct confidence intervals.

```
load("Best_Model_Results_list.rda")
lapply(model_results, summary)
```

```
## $BT_basic_Flesch
## 
## Call:
## BTm(player1 = easier, player2 = harder, formula = ~Flesch[ID], 
##     id = "ID", data = job999866covars_chameleons)
## 
## Deviance Residuals: 
##    Min      1Q  Median      3Q     Max  
## 0.6265  1.0440  1.1496  1.2559  1.7732  
## 
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)    
## Flesch[ID] 0.0202365  0.0008018   25.24   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for binomial family taken to be 1)
## 
##     Null deviance: 26936  on 19430  degrees of freedom
## Residual deviance: 26266  on 19429  degrees of freedom
## AIC: 26268
## 
## Number of Fisher Scoring iterations: 4
## 
## 
## $BT_basic_RF
## 
## Call:
## BTm(player1 = easier, player2 = harder, formula = ~google_min_2000[ID] + 
##     meanSentenceChars[ID] + pr_noun[ID], id = "ID", data = job999866covars_chameleons)
## 
## Deviance Residuals: 
##    Min      1Q  Median      3Q     Max  
## 0.3105  1.1143  1.1660  1.2043  2.4589  
## 
## Coefficients:
##                         Estimate Std. Error z value Pr(>|z|)    
## google_min_2000[ID]    1.298e+03  1.531e+02   8.481   <2e-16 ***
## meanSentenceChars[ID] -1.476e-02  5.613e-04 -26.298   <2e-16 ***
## pr_noun[ID]            4.281e-01  1.663e-01   2.575     0.01 *  
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for binomial family taken to be 1)
```

```
##
##     Null deviance: 26936  on 19430  degrees of freedom
## Residual deviance: 25909  on 19427  degrees of freedom
## AIC: 25915
##
## Number of Fisher Scoring iterations: 3
##
##
## $BT_optimal_Flesch
##
## Call:
## BTm(player1 = easier, player2 = harder, formula = ~meanSentenceLength[ID] +
##     meanWordSyllables[ID], id = "ID", data = job999866covars_chameleons)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
##  0.414   1.048   1.148   1.242   2.175
##
## Coefficients:
##                         Estimate Std. Error z value Pr(>|z|)
## meanSentenceLength[ID] -0.062778   0.002535  -24.77   <2e-16 ***
## meanWordSyllables[ID]  -1.788921   0.069718  -25.66   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 26936  on 19430  degrees of freedom
## Residual deviance: 25906  on 19428  degrees of freedom
## AIC: 25910
##
## Number of Fisher Scoring iterations: 4
##
##
## $BT_best
##
## Call:
## BTm(player1 = easier, player2 = harder, formula = ~google_min_2000[ID] +
##     meanSentenceChars[ID] + pr_noun[ID] + meanWordChars[ID],
##     id = "ID", data = job999866covars_chameleons)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
##  0.259   1.041   1.146   1.239   2.609
##
## Coefficients:
##                         Estimate Std. Error z value Pr(>|z|)
## google_min_2000[ID]    1.319e+03  1.556e+02   8.472   <2e-16 ***
## meanSentenceChars[ID] -1.384e-02  5.565e-04 -24.866   <2e-16 ***
## pr_noun[ID]            3.117e-01  1.676e-01   1.860   0.0629 .
## meanWordChars[ID]     -3.126e-01  2.367e-02 -13.205   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 26936  on 19430  degrees of freedom
## Residual deviance: 25732  on 19426  degrees of freedom
## AIC: 25740
##
## Number of Fisher Scoring iterations: 4
```

### 9. Supplementary fitted model results

**Files:** `Classic_Model_Results_list.rda`

**Guide:** This file contains the results of the structured Bradley-Terry models that replicate classic measures of textual complexity. The models are: Flesch, Dale-Chall, FOG, SMOG, Spache, and Coleman-Liau, each defined in the manuscript.

```
load("Classic_Model_Results_list.rda")
lapply(model_results_classic, summary)
```

```
## $FRE
##
## Call:
## BTm(player1 = easier, player2 = harder, formula = ~Flesch[ID],
##     id = "ID", data = dat)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## 0.6265  1.0440  1.1496  1.2559  1.7732
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## Flesch[ID] 0.0202365  0.0008018   25.24   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 26936  on 19430  degrees of freedom
## Residual deviance: 26266  on 19429  degrees of freedom
## AIC: 26268
##
## Number of Fisher Scoring iterations: 4
##
##
## $`Dale-Chall`
##
## Call:
## BTm(player1 = easier, player2 = harder, formula = ~Dale.Chall.old[ID],
##     id = "ID", data = job999866covars_chameleons)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## 0.4892  1.0477  1.1480  1.2538  2.0892
##
```

```
## Coefficients:
##                     Estimate Std. Error z value Pr(>|z|)
## Dale.Chall.old[ID] -0.198467   0.007952  -24.96   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 26936  on 19430  degrees of freedom
## Residual deviance: 26276  on 19429  degrees of freedom
## AIC: 26278
##
## Number of Fisher Scoring iterations: 4
##
##
## $FOG
##
## Call:
## BTm(player1 = easier, player2 = harder, formula = ~FOG[ID], id = "ID",
##     data = job999866covars_chameleons)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## 0.4457  1.0447  1.1491  1.2491  2.0300
##
## Coefficients:
##         Estimate Std. Error z value Pr(>|z|)
## FOG[ID] -0.11049    0.00397  -27.83   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 26936  on 19430  degrees of freedom
## Residual deviance: 26079  on 19429  degrees of freedom
## AIC: 26081
##
## Number of Fisher Scoring iterations: 4
##
##
## $SMOG
##
## Call:
## BTm(player1 = easier, player2 = harder, formula = ~SMOG[ID],
##     id = "ID", data = job999866covars_chameleons)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## 0.4803  1.0393  1.1468  1.2539  2.0831
##
## Coefficients:
##          Estimate Std. Error z value Pr(>|z|)
## SMOG[ID] -0.116338   0.004403  -26.42   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 26936  on 19430  degrees of freedom
## Residual deviance: 26186  on 19429  degrees of freedom
## AIC: 26188
##
## Number of Fisher Scoring iterations: 4
##
##
## $Spache
##
## Call:
## BTm(player1 = easier, player2 = harder, formula = ~Spache[ID],
##     id = "ID", data = job999866covars_chameleons)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## 0.4123  1.0309  1.1434  1.2518  2.1663
##
## Coefficients:
##            Estimate Std. Error z value Pr(>|z|)
## Spache[ID] -0.44382    0.01471  -30.18   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 26936  on 19430  degrees of freedom
## Residual deviance: 25904  on 19429  degrees of freedom
## AIC: 25906
##
## Number of Fisher Scoring iterations: 4
##
##
## $`Coleman-Liau`
##
## Call:
## BTm(player1 = easier, player2 = harder, formula = ~Coleman.Liau[ID],
##     id = "ID", data = job999866covars_chameleons)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## 0.7539  1.0829  1.1625  1.2413  1.6142
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## Coleman.Liau[ID] 0.020857   0.001105   18.88   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
```

```
##     Null deviance: 26936  on 19430  degrees of freedom
## Residual deviance: 26570  on 19429  degrees of freedom
## AIC: 26572
##
## Number of Fisher Scoring iterations: 4
```