Additional_Stats

Silvie Cinkova, 2017-02-07, TextLink Training School UFAL Prague

NB – The output at the end has been truncated in this document.

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 3.2.5
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(tidyr)
```

```
## Warning: package 'tidyr' was built under R version 3.2.5
```

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.2.5
```

```
library(stringr)
```

```
## Warning: package 'stringr' was built under R version 3.2.5
```

More about Documents - Do They Differ Genrewise?

```
#src_path <- file.path("edu", "r", "textlink", "src_data")
src_path <- file.path("C:", "Seafile", "Silvie_R", "textlink", "src_data")
words_path <- paste(src_path, "PDT_document_word_counts.csv", sep = "/")
readLines(words_path, n = 4)
```

```
## [1] "cmpr9410_001,644" "cmpr9410_002,629" "cmpr9410_003,202"
## [4] "cmpr9410_004,597"
```

CSV with no header. Do not read first line as header and give names to columns.

```
words <- read.csv(words_path, header = FALSE, col.names = c("document_id", "w
ord_count"))
head(words)
```

```
##    document_id word_count
## 1 cmpr9410_001        644
## 2 cmpr9410_002        629
## 3 cmpr9410_003        202
## 4 cmpr9410_004        597
## 5 cmpr9410_005        382
## 6 cmpr9410_006        132
```

Read PDT30 again and merge the two data frames by the document id. It was smart to call the columns identically in both data frames!

```
pdt30 <- read.csv(paste(src_path, "PDT_30.csv", sep = "/"))
pdt30_words <- merge(pdt30, words, by = "document_id")
```

have a look at selected columns

```
dplyr::distinct(.data = pdt30_words, document_id,.keep_all = TRUE) %>% select
(one_of(c("document_id", "word_count", "number_of_sentences", "genre"))) %>%
head()
```

```
##    document_id word_count number_of_sentences       genre
## 1 cmpr9410_001        644                  33     comment
## 2 cmpr9410_002        629                  41 description
## 3 cmpr9410_003        202                  16      advice
## 4 cmpr9410_004        597                  37 description
## 5 cmpr9410_005        382                  29      advice
## 6 cmpr9410_006        132                   7        news
```

Normalized Deviation of Proportions for Different Discourse Classes

Deviation of Proportions, introduced by Gries, 2008 and Gries & Lijffijt 2012. The implementation presented here draws on Levshina, 2015, p.82-84. A value between 0 and 1. The closer to 0, the more evenly is the given word (here discourse class) distributed across the defined groups (can be documents, registers, here genres). The core idea behind this metrics is the difference between observed and expected proportions like in Chisq, but the result applies to a single word (or word-like phenomenon of choice) and the values are convenient to compare. The normalized Deviation of Proportions (norm_DP) is defined as `norm_DP_word <- (sum(abs(obs_word_prop - exp_word_prop))/2)/(1-min(exp_word_prop))`.

We need to know the total word count per genre and the total count of each discourse class per genre.

```
genre_tab <- dplyr::summarize(group_by(pdt30_words, genre), sum(word_count))
colnames(genre_tab) <- c("genre", "total_word_count")
```

Compute expected proportions of genres. They represent the probability of a discourse class to accur in a particular genre by mere chance.

```
genre_tab <- dplyr::mutate(genre_tab, prop_genre = total_word_count/sum(total
_word_count))
head(genre_tab)
```

```
## # A tibble: 6 × 3
##         genre total_word_count   prop_genre
##        <fctr>            <int>        <dbl>
## 1      advice           487577 0.0371396122
## 2     caption             1677 0.0001277401
## 3  collection           227240 0.0173092773
## 4     comment           881896 0.0671755958
## 5 description          1678383 0.1278454353
## 6       essay          3337380 0.2542142042
```

Expected proportions are stored in the `prop_genre` column of `genre_tab`.

```
exp_prop <- genre_tab$prop_genre
names(exp_prop) <- as.character(genre_tab$genre)
```

Absolute counts of EXPANSION in each genre:

```
discl_expansion_abs <- filter(pdt30_words, discourse_class == "EXPANSION") %>
% .$genre %>% table()
discl_expansion_abs
```

```
## .
##        advice       caption    collection       comment   description
##           293            20           226           656          1034
##         essay    invitation        letter          news         other
##          1569           128           111          1979           250
##      overview person_interv          plot       program        review
##            78           306            10            14           482
##         sport        survey  topic_interv       weather
##          1095            68           518            14
```

Proportions (relative counts) of EXPANSION in each genre:

```
discl_expansion_prop <- prop.table(discl_expansion_abs)
discl_expansion_prop
```

```
## .
##        advice       caption    collection       comment   description
##   0.033103604   0.002259632   0.025533838   0.074115919   0.116822958
##         essay    invitation        letter          news         other
##   0.177268105   0.014461643   0.012540956   0.223590555   0.028245396
##      overview person_interv          plot       program        review
##   0.008812564   0.034572365   0.001129816   0.001581742   0.054457123
##         sport        survey  topic_interv       weather
##   0.123714834   0.007682748   0.058524461   0.001581742
```

Now compute the DP of EXPANSION.

```
norm_DP_EXPANSION <- (sum(abs(discl_expansion_prop - exp_prop))/2)/(1- min(ex
p_prop))
norm_DP_EXPANSION

## [1] 0.2072511
```

Write a function that computes normalized_DP_expansion from expected and observed proportions.

```
norm_DP <- function(exp_word_prop, obs_word_prop) {
  norm_DP <- (sum(abs(obs_word_prop - exp_word_prop))/2)/(1-min(exp_word_prop
))
  return(norm_DP)
}
```

Get observed proportions of CONTRAST. (Do it like for EXPANSION above, merging two chunks into one by additional pipe and getting directly proportions instead of absolute counts)

```
discl_contrast_prop <- filter(pdt30_words, discourse_class == "EXPANSION") %>
% .$genre %>% table() %>% prop.table()
discl_contrast_prop

## .
##        advice       caption    collection       comment   description
##   0.033103604   0.002259632   0.025533838   0.074115919   0.116822958
##         essay    invitation        letter          news         other
##   0.177268105   0.014461643   0.012540956   0.223590555   0.028245396
##      overview person_interv          plot       program        review
##   0.008812564   0.034572365   0.001129816   0.001581742   0.054457123
##         sport        survey  topic_interv       weather
##   0.123714834   0.007682748   0.058524461   0.001581742
```

Now use the function

```
norm_DP_CONTRAST <- norm_DP(exp_word_prop = exp_prop, obs_word_prop = discl_c
ontrast_prop)
norm_DP_CONTRAST

## [1] 0.2072511
```

Extract observed proportions for CONTINGENCY, do it like with CONTRAST above

```
discl_contingency_prop <- filter(pdt30_words, discourse_class == "CONTINGENCY
") %>% .$genre %>% table() %>% prop.table()
discl_contingency_prop

## .
##        advice       caption    collection       comment   description
##  0.0514784088  0.0017017656  0.0244628802  0.1027440970  0.1106147628
##         essay    invitation        letter          news         other
##  0.1942139970  0.0065943416  0.0191448628  0.2091044459  0.0227611147
```

```
##      overview person_interv          plot       program        review
##  0.0038289726  0.0368006807  0.0010636035  0.0002127207  0.0444586258
##         sport        survey  topic_interv       weather
##  0.0987024037  0.0070197830  0.0650925335  0.0000000000
```

```
norm_DP_CONTINGENCY <- norm_DP(exp_word_prop = exp_prop, obs_word_prop = disc
l_contingency_prop)
norm_DP_CONTINGENCY
```

```
## [1] 0.1906186
```

Now the same for TEMPORAL.

```
discl_temporal_prop <- filter(pdt30_words, discourse_class == "TEMPORAL") %>%
.$genre %>% table() %>% prop.table()
discl_temporal_prop
```

```
## .
##        advice       caption    collection       comment   description
##  0.0225140713  0.0009380863  0.0347091932  0.0647279550  0.0938086304
##         essay    invitation        letter          news         other
##  0.1716697936  0.0065666041  0.0075046904  0.2429643527  0.0450281426
##      overview person_interv          plot       program        review
##  0.0028142589  0.0572232645  0.0018761726  0.0037523452  0.0328330206
##         sport        survey  topic_interv       weather
##  0.1613508443  0.0037523452  0.0459662289  0.0000000000
```

```
norm_DP_TEMPORAL <- norm_DP(exp_word_prop = exp_prop, obs_word_prop = discl_t
emporal_prop)
norm_DP_TEMPORAL
```

```
## [1] 0.2348267
```

All discourse classes appear to have very similar and rather low genre bias. This does not mean that either normalized deviation of proportions or chisq is wrong. It rather means that the differences in the distribution of discourse classes do not occur by chance (i.e. this would be a common result in a very large number of experiments with different data but identical setup), but that the effect size is rather low, still. The implication would be - the distribution of discourse classes is probably not a good predictor of genre in an unknown text.

Comparing Genres according to a Quantitative Variable - Connectedness

Let's think of a variable that we call *connectedness of a document*. It is rendered as the proportion of connectives and the total amount of words in the document. We will observe the distribution of connectedness values in each genre and test whether two genres differ in the connectedness of their documents.

Data wrangling: row = document, variables = genre and connectedness. To compute connectedness, we need total word count (available) and total number of connectives in the document. Let's compute this first.

```
connectives_in_doc <- summarize(group_by(pdt30_words, document_id), total_con
nectives = length(discourse_class))
head(connectives_in_doc, 2)

## # A tibble: 2 × 2
##    document_id total_connectives
##         <fctr>             <int>
## 1 cmpr9410_001                12
## 2 cmpr9410_002                10

table(pdt30_words$document_id)[1:2] #check

##
## cmpr9410_001 cmpr9410_002
##           12           10
```

We will merge it with `pdt30` into `pdt30_docs`.

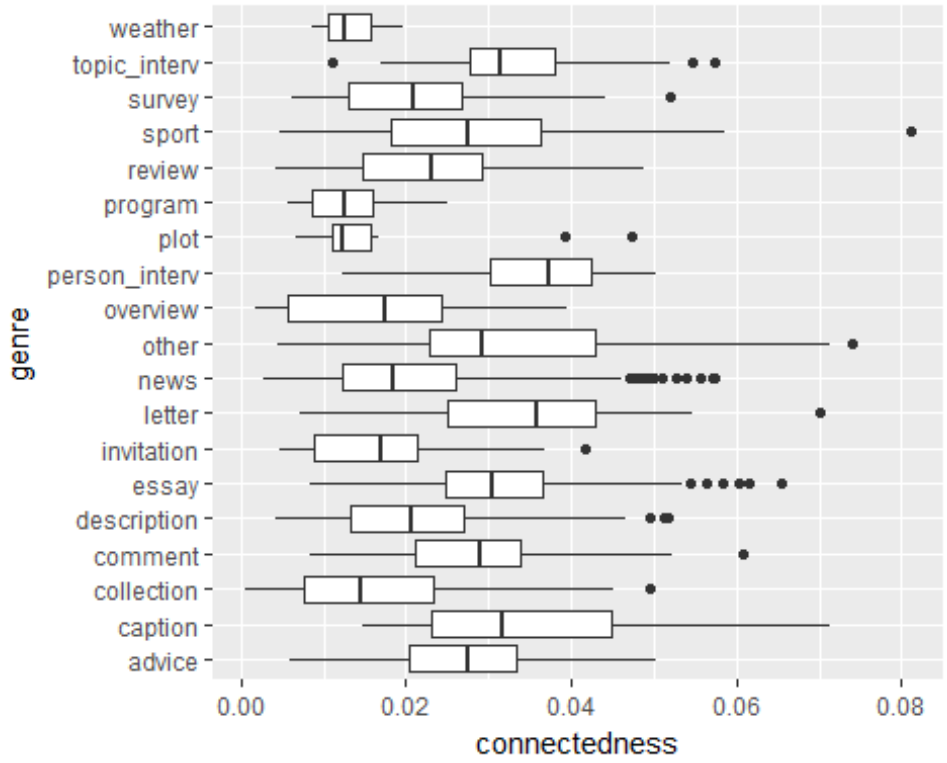
```
pdt30_docs <- merge(pdt30_words, connectives_in_doc, by = "document_id") %>%
distinct(document_id, .keep_all = TRUE) %>% select(-(starts_with("discourse")
)) %>% select(-starts_with("sentence"))
```

We add a column with computed connectedness.

```
pdt30_docs <- mutate(pdt30_docs, connectedness = total_connectives/word_count
)
```

Let's compare boxplots

```
ggplot(pdt30_docs, aes(x = genre, y = connectedness)) + geom_boxplot() + coor
d_flip()
```

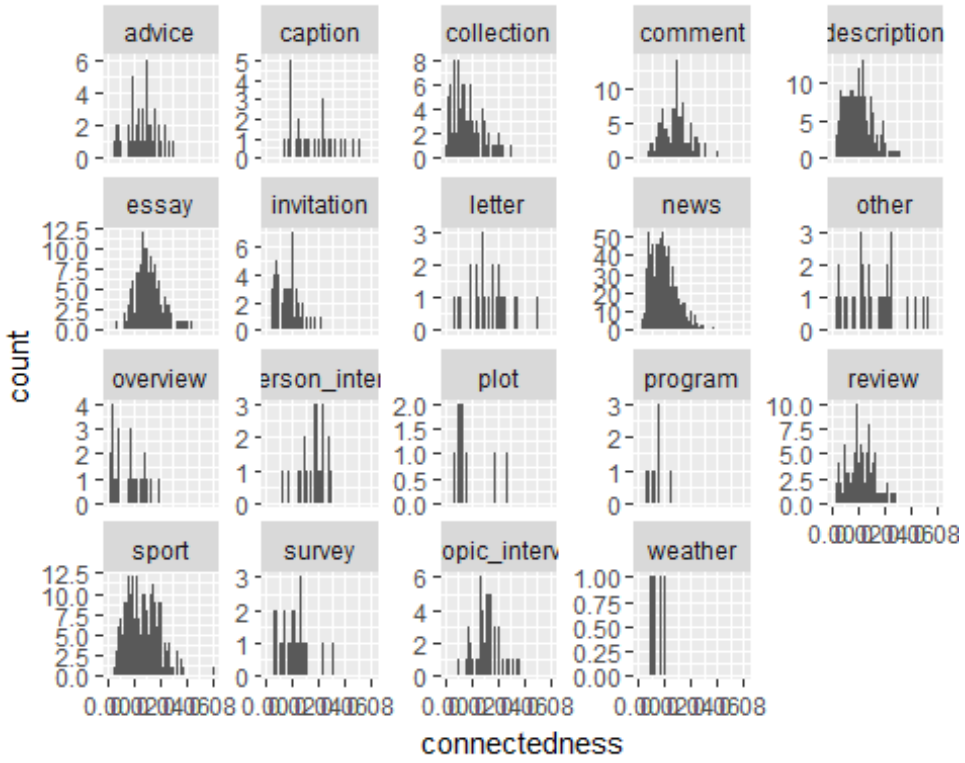



```
max(pdt30_docs$connectedness)
```

```
## [1] 0.08108108
```

Let's compare histograms

```
ggplot(pdt30_docs, aes(x = connectedness,  group = genre)) + geom_histogram(b
inwidth = 0.001) + facet_wrap(~ genre, scales = "free_y")
```



They do not look similar. Let's get rid of data-sparse genres: table number of documents to identify them

```
docsums <- as.data.frame(table(pdt30_docs$genre))
colnames(docsums) = c("genre", "number_of_documents")
docsums
```

```
##            genre number_of_documents
## 1         advice                  56
## 2        caption                  30
## 3     collection                 115
## 4        comment                 150
## 5    description                 233
## 6          essay                 186
## 7     invitation                  61
## 8         letter                  27
## 9           news                1111
## 10         other                  34
## 11      overview                  27
## 12 person_interv                  24
## 13          plot                  10
## 14       program                  10
```

```
## 15         review                 117
## 16          sport                 296
## 17         survey                  26
## 18   topic_interv                  61
## 19        weather                   6
```

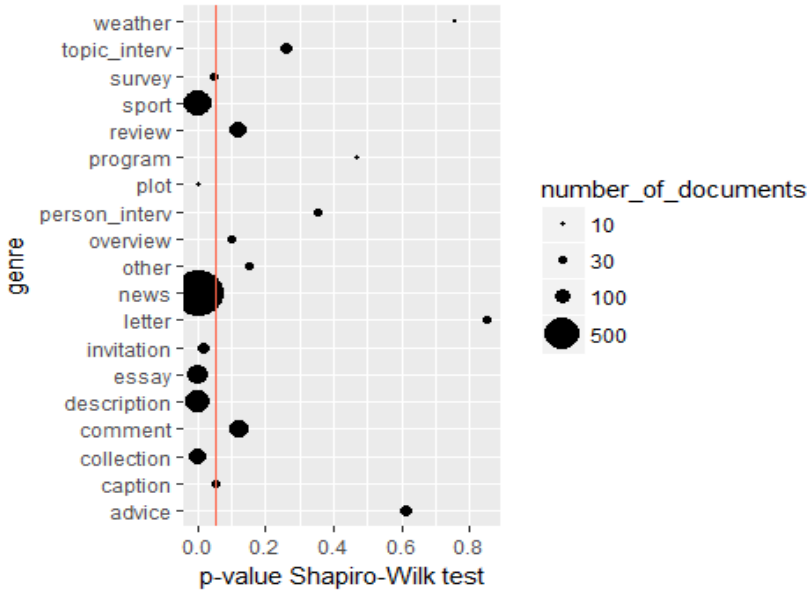
An efficient test to check whether two samples differ according to a distribution of a quantitative variable is t-test. But it assumes (among others) that each sample contains at least 30 observations and/or the observations are normally distributed. Tabulation reveals that some samples are smaller. If our samples are not normally distributed on top of that, we cannot use t-test and must resort to its non-parametric counterpart. Let's test the distributions with Shapiro-Wilk test. Do they significantly differ from normal distribution? We will get p-values for the H0 that they do not differ.

```
shapiro.test(pdt30_docs$connectedness[pdt30_docs$genre == "news"])

##
##  Shapiro-Wilk normality test
##
## data:  pdt30_docs$connectedness[pdt30_docs$genre == "news"]
## W = 0.95392, p-value < 2.2e-16

shap_genre <- pdt30_docs %>%
    group_by(genre) %>%
    summarise(stest = shapiro.test(connectedness)$p.value)
shap_genre <- merge(shap_genre, docsums, by = "genre")

ggplot(shap_genre, aes(x = genre, y = stest, size = number_of_documents)) + g
eom_point() + coord_flip() + geom_hline(yintercept = 0.05, color = "tomato")
+ scale_size(range = c(0, 10), breaks = c(10, 30, 100, 500)) + ylab("p-value
Shapiro-Wilk test")
```



The plot shows that quite many even differ from the normal distribution (left of the red line at 0.05). We could possibly achieve normal distribution in some by removing the outlier document from collection (the one with one connective per 160 sentences) and by merging caption and plot, but we would anyway still have most larger genres not normally distributed. Let's use the Wilcox test (aka Mann–Whitney U test, Mann–Whitney–Wilcoxon (MWW), Wilcoxon rank-sum test, or Wilcoxon–Mann–Whitney test). It should be nearly as efficient as the parametric t-test. Mind that both t-test and U-test have two variants: for independent vs. paired samples. Our samples are independent.

Now we would have to compare the genres pairwise.

We could write a script to do it bulk-wise for all pairs of genres, but let's run a few tests manually:

Connectedness in news vs. description

```
news_connectedness <- filter(pdt30_docs, genre == "news")
description_connectedness <- filter(pdt30_docs, genre == "description")
wilcox.test(news_connectedness$connectedness, description_connectedness$conne
ctedness, paired = FALSE, conf.int = TRUE, conf.level = 0.95)

##
##  Wilcoxon rank sum test with continuity correction
##
## data:  news_connectedness$connectedness and description_connectedness$conn
ectedness
## W = 120050, p-value = 0.08155
## alternative hypothesis: true location shift is not equal to 0
## 95 percent confidence interval:
##  -0.0026296363  0.0001297716
## sample estimates:
## difference in location
##           -0.001229996
```

They are not different according to connectedness. The difference between their medians can lie between -0.0026296363 0.0001297716, i.e. can also be zero. This is the imagined outcome of 95% of a huge number of imagined repeated experiments with the same setup but of course with different texts. (this is said by *Confidence interval at 95%*).

Bulk Computation for All Genre Pairs

Get all possible pairs

```
genre_pairs <- combn(as.character(docsums$genre), 2, simplify = FALSE)
head(genre_pairs)

## [[1]]
## [1] "advice"  "caption"
##
## [[2]]
```

```
## [1] "advice"     "collection"
##
## [[3]]
## [1] "advice"  "comment"
##
## [[4]]
## [1] "advice"      "description"
##
## [[5]]
## [1] "advice" "essay"
##
## [[6]]
## [1] "advice"     "invitation"
```

```
for (i in 1:length(genre_pairs)) {
  x_connectedness <- filter(pdt30_docs, genre == genre_pairs[[i]][[1]])
y_connectedness <- filter(pdt30_docs, genre == genre_pairs[[i]][[2]])
cat(genre_pairs[[i]][1], "vs.", genre_pairs[[i]][2], "\n", sep = " ")
result <- wilcox.test(x_connectedness$connectedness, y_connectedness$connecte
dness, paired = FALSE, conf.int = TRUE, conf.level = 0.95)
str(result)
#cat("statistic:", result$statistic, "\n", "p.value: ", result$p.value, "\n",
"alternative: ", result$alternative, "\n", "method", result$method, "\n", "co
nfidence interval: ", result$conf.int, "\n", "estimate: ", result$estimate, "
\n", sep = " ")
cat("\n---------------------\n")

}
```

```
## advice vs. caption
## List of 9
##  $ statistic  : Named num 624
##   ..- attr(*, "names")= chr "W"
##  $ parameter  : NULL
##  $ p.value    : num 0.0514
##  $ null.value : Named num 0
##   ..- attr(*, "names")= chr "location shift"
##  $ alternative: chr "two.sided"
##  $ method     : chr "Wilcoxon rank sum test with continuity correction"
##  $ data.name  : chr "x_connectedness$connectedness and y_connectedness$con
nectedness"
##  $ conf.int   : atomic [1:2] -0.013205 0.000021
##   ..- attr(*, "conf.level")= num 0.95
##  $ estimate   : Named num -0.00672
##   ..- attr(*, "names")= chr "difference in location"
##  - attr(*, "class")= chr "htest"
##
## ---------------------
## advice vs. collection
## List of 9
```

```
##  $ statistic  : Named num 4918
##   ..- attr(*, "names")= chr "W"
##  $ parameter  : NULL
##  $ p.value    : num 2.28e-08
##  $ null.value : Named num 0
##   ..- attr(*, "names")= chr "location shift"
##  $ alternative: chr "two.sided"
##  $ method     : chr "Wilcoxon rank sum test with continuity correction"
##  $ data.name  : chr "x_connectedness$connectedness and y_connectedness$con
nectedness"
##  $ conf.int   : atomic [1:2] 0.00812 0.01542
##   ..- attr(*, "conf.level")= num 0.95
##  $ estimate   : Named num 0.0121
##   ..- attr(*, "names")= chr "difference in location"
##  - attr(*, "class")= chr "htest"
##
## ---------------------
## advice vs. comment
## List of 9
##  $ statistic  : Named num 4010
##   ..- attr(*, "names")= chr "W"
##  $ parameter  : NULL
##  $ p.value    : num 0.618
##  $ null.value : Named num 0
##   ..- attr(*, "names")= chr "location shift"
##  $ alternative: chr "two.sided"
##  $ method     : chr "Wilcoxon rank sum test with continuity correction"
##  $ data.name  : chr "x_connectedness$connectedness and y_connectedness$con
nectedness"
##  $ conf.int   : atomic [1:2] -0.00392 0.00225
##   ..- attr(*, "conf.level")= num 0.95
##  $ estimate   : Named num -0.000801
##   ..- attr(*, "names")= chr "difference in location"
##  - attr(*, "class")= chr "htest"
##
## ---------------------
## advice vs. description
## List of 9
##  $ statistic  : Named num 8834
##   ..- attr(*, "names")= chr "W"
##  $ parameter  : NULL
##  $ p.value    : num 3.91e-05
##  $ null.value : Named num 0
##   ..- attr(*, "names")= chr "location shift"
##  $ alternative: chr "two.sided"
##  $ method     : chr "Wilcoxon rank sum test with continuity correction"
##  $ data.name  : chr "x_connectedness$connectedness and y_connectedness$con
nectedness"
##  $ conf.int   : atomic [1:2] 0.00369 0.00997
##   ..- attr(*, "conf.level")= num 0.95
```

```
##  $ estimate   : Named num 0.00686
##   ..- attr(*, "names")= chr "difference in location"
##  - attr(*, "class")= chr "htest"
##
## ---------------------
## advice vs. essay
## List of 9
##  $ statistic  : Named num 4160
##   ..- attr(*, "names")= chr "W"
##  $ parameter  : NULL
##  $ p.value    : num 0.0225
##  $ null.value : Named num 0
##   ..- attr(*, "names")= chr "location shift"
##  $ alternative: chr "two.sided"
##  $ method     : chr "Wilcoxon rank sum test with continuity correction"
##  $ data.name  : chr "x_connectedness$connectedness and y_connectedness$con
nectedness"
##  $ conf.int   : atomic [1:2] -0.006613 -0.000484
##   ..- attr(*, "conf.level")= num 0.95
##  $ estimate   : Named num -0.00354
##   ..- attr(*, "names")= chr "difference in location"
##  - attr(*, "class")= chr "htest"
##
## ---------------------
## advice vs. invitation
## List of 9
##  $ statistic  : Named num 2680
##   ..- attr(*, "names")= chr "W"
##  $ parameter  : NULL
##  $ p.value    : num 1.14e-07
##  $ null.value : Named num 0
##   ..- attr(*, "names")= chr "location shift"
##  $ alternative: chr "two.sided"
##  $ method     : chr "Wilcoxon rank sum test with continuity correction"
##  $ data.name  : chr "x_connectedness$connectedness and y_connectedness$con
nectedness"
##  $ conf.int   : atomic [1:2] 0.00716 0.01433
##   ..- attr(*, "conf.level")= num 0.95
##  $ estimate   : Named num 0.0108
##   ..- attr(*, "names")= chr "difference in location"
##  - attr(*, "class")= chr "htest"
##
## ---------------------
## advice vs. letter
## List of 9
##  $ statistic  : Named num 538
##   ..- attr(*, "names")= chr "W"
##  $ parameter  : NULL
##  $ p.value    : num 0.0341
##  $ null.value : Named num 0
```