

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
## v dplyr      1.1.4      v readr      2.1.5
## v forcats    1.0.0      v stringr   1.5.1
## v ggplot2    3.5.1      v tibble    3.2.1
## v lubridate  1.9.3      v tidyr     1.3.1
## v purrr      1.0.2
```

```
## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()     masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
data <- read_csv("measurements.csv")
```

```
## Rows: 766 Columns: 96
```

```
## -- Column specification -----
## Delimiter: ","
```

```
## chr  (9): fpath, KUK_ID, class, FileName, FolderPath, subcorpus, DocumentTit...
```

```
## dbl  (85): RuleAbstractNouns, RuleAmbiguousRegards, RuleAnaphoricReferences, ...
```

```
## lgl  (2): ClarityPursuit, SyllogismBased
```

```
##
```

```
## i Use `spec()` to retrieve the full column specification for this data.
```

```
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

The following snippet is taken (pretty much) from classifier.Rmd.

```
data_clean <- data %>%
```

```
  select(!c(
```

```
    fpath,
```

```
    KUK_ID,
```

```
    FileName,
```

```
    FolderPath,
```

```
    subcorpus,
```

```
    class,
```

```
    DocumentTitle,
```

```
    ClarityPursuit,
```

```
    Readability,
```

```
    SyllogismBased,
```

```
    SourceDB
```

```
  )) %>%
```

```
  # replace -1s in variation coefficients with NAs
```

```
  mutate(across(c(
```

```
    `RuleDoubleAdpos.max_allowable_distance.v`,
```

```
    `RuleTooManyNegations.max_negation_frac.v`,
```

```
    `RuleTooManyNegations.max_allowable_negations.v`,
```

```
    `RuleTooManyNominalConstructions.max_noun_frac.v`,
```

```
    `RuleTooManyNominalConstructions.max_allowable_nouns.v`,
```

```
    `RuleCaseRepetition.max_repetition_count.v`,
```

```
    `RuleCaseRepetition.max_repetition_frac.v`,
```

```
    `RulePredSubjDistance.max_distance.v`,
```

```
    `RulePredObjDistance.max_distance.v`,
```

```
    `RuleInfVerbDistance.max_distance.v`,
```

```
    `RuleMultiPartVerbs.max_distance.v`,
```

```
    `RuleLongSentences.max_length.v`,
```

```

`RulePredAtClauseBeginning.max_order.v`,
`mattr.v`,
`maentropy.v`
), ~ na_if(.x, -1))) %>%
# replace NAs with 0s
replace_na(list(
  RuleGPcoordovs = 0,
  RuleGPdeverbaddr = 0,
  RuleGPpatinstr = 0,
  RuleGPdeverbsubj = 0,
  RuleGPadjective = 0,
  RuleGPpatbenperson = 0,
  RuleGPwordorder = 0,
  RuleDoubleAdpos = 0,
  RuleDoubleAdpos.max_allowable_distance = 0,
  RuleDoubleAdpos.max_allowable_distance.v = 0,
  RuleAmbiguousRegards = 0,
  RuleReflexivePassWithAnimSubj = 0,
  RuleTooManyNegations = 0,
  RuleTooManyNegations.max_negation_frac = 0,
  RuleTooManyNegations.max_negation_frac.v = 0,
  RuleTooManyNegations.max_allowable_negations = 0,
  RuleTooManyNegations.max_allowable_negations.v = 0,
  RuleTooManyNominalConstructions.max_noun_frac.v = 0,
  RuleTooManyNominalConstructions.max_allowable_nouns.v = 0,
  RuleFunctionWordRepetition = 0,
  RuleCaseRepetition.max_repetition_count.v = 0,
  RuleCaseRepetition.max_repetition_frac.v = 0,
  RuleWeakMeaningWords = 0,
  RuleAbstractNouns = 0,
  RuleRelativisticExpressions = 0,
  RuleConfirmationExpressions = 0,
  RuleRedundantExpressions = 0,
  RuleTooLongExpressions = 0,
  RuleAnaphoricReferences = 0,
  RuleLiteraryStyle = 0,
  RulePassive = 0,
  RulePredSubjDistance = 0,
  RulePredSubjDistance.max_distance = 0,
  RulePredSubjDistance.max_distance.v = 0,
  RulePredObjDistance = 0,
  RulePredObjDistance.max_distance = 0,
  RulePredObjDistance.max_distance.v = 0,
  RuleInfVerbDistance = 0,
  RuleInfVerbDistance.max_distance = 0,
  RuleInfVerbDistance.max_distance.v = 0,
  RuleMultiPartVerbs = 0,
  RuleMultiPartVerbs.max_distance = 0,
  RuleMultiPartVerbs.max_distance.v = 0,
  RuleLongSentences.max_length.v = 0,
  RulePredAtClauseBeginning.max_order.v = 0,
  RuleVerbalNouns = 0,
  RuleDoubleComparison = 0,

```

```

    RuleWrongValencyCase = 0,
    RuleWrongVerbominalCase = 0,
    RuleIncompleteConjunction = 0
  ))

# norm data expected to correlate with text length
data_clean_normed <- data_clean %>%
  mutate(across(c(
    RuleGPcoordovs,
    RuleGPdeverbaddr,
    RuleGPpatinstr,
    RuleGPdeverbsubj,
    RuleGPadjective,
    RuleGPpatbenperson,
    RuleGPwordorder,
    RuleDoubleAdpos,
    RuleAmbiguousRegards,
    RuleFunctionWordRepetition,
    RuleWeakMeaningWords,
    RuleAbstractNouns,
    RuleRelativisticExpressions,
    RuleConfirmationExpressions,
    RuleRedundantExpressions,
    RuleTooLongExpressions,
    RuleAnaphoricReferences,
    RuleLiteraryStyle,
    RulePassive,
    RuleVerbalNouns,
    RuleDoubleComparison,
    RuleWrongValencyCase,
    RuleWrongVerbominalCase,
    RuleIncompleteConjunction,
    num_hapax,
    RuleReflexivePassWithAnimSubj,
    RuleTooManyNominalConstructions,
    RulePredSubjDistance,
    RuleMultiPartVerbs,
    RulePredAtClauseBeginning
  ), ~ .x / word_count)) %>%
  mutate(across(c(
    RuleTooFewVerbs,
    RuleTooManyNegations,
    RuleCaseRepetition,
    RuleLongSentences,
    RulePredObjDistance,
    RuleInfVerbDistance
  ), ~ .x / sent_count))

cor_to_graph <- function(data) {
  matrix <- cor(data) %>% as_tibble()
  cnames <- names(matrix)
  matrix %>%
    mutate(Source = cnames) %>%
    select(Source, everything()) %>%

```

```

    pivot_longer(!Source, names_to = "Target", values_to = "Weight") %>%
    mutate(across(Weight, ~ abs(.x))) %>%
    filter(Weight > .25 & Weight < 1)
}

export_to_gephi <- function(edges, ename, nname) {
  edges %>% write_csv(ename)
}

nodes <- names(data_clean)
tibble(Id = nodes, Label = nodes) %>% write_csv("gephi/nodes.csv")

edges <- cor_to_graph(data_clean)
edges_normed <- cor_to_graph(data_clean_normed)

export_to_gephi(edges, "gephi/edges.csv")
export_to_gephi(edges_normed, "gephi/edges_normed.csv")

```