Basic Network Analysis of Wordbank Data

Build Networks

List of libraries

```
library(purrr)
library(readr)
library(ggplot2)
library(langcog)
library(boot)
library(dplyr)
library(tidyr)
library(wordbankr)
library(directlabels)
library(stringr)
library(lmtest)
library(rwebppl)
library(jsonlite)
library(nlme)
library(feather)
library(broom)
library(HDInterval)
library(BBmisc)
library(igraph)
library(knitr)
library(xtable)
```

Import helper functions

```
source(paste(getwd(),"/helpers/all_helper.r",sep = ""), chdir = T)
```

Create pairs

```
#Import data from wordbank in a format that will be useful for us (especially when we will do developme

#The format: for each age (in months, it starts with 16, 17,..) list all words that have NOT yet been a

wb_data <- make_aoa_dataframe(lang="English (American)", lang_form = "WS", lex_class = "nouns")


#extract the first age (it is 16 month in English)
first_age<- wb_data$age[1]

#Extract the list of all uni_lemmas (like Hills, we will start with the analysis of all words first)
lemma_list<- wb_data %>%
  trim_all_unilemma() %>% #We do naive triming at this point, this means we are ignoring homophone/poly
  filter(age==first_age) %>% #Since the first month in our format is the month when all words are stil
  select(item, uni_lemma)

    # list of definitions (we don't need them at this point)
    def_list<- wb_data %>%
      trim_all_definition() %>%
      filter(age==first_age) %>%
      select(item, definition)
```

```r
#Make list of pairs for associative data
#The output: all pairs of words (first is named "item"" and second is named "pair"), link =0 (no link),
assoc_pairs <- make_assoc_pairs(lemma_list = lemma_list)

#Make list of pairs for MacRae features
#The output: same as above, but here instead of link, we have "shared" which specify the number of shar
feature_pairs <- make_feature_pairs(lemma_list = lemma_list)
```

Build networks

```r
assoc_links <- assoc_pairs %>%
  filter(link==1) %>%
  select(item, pair, item.definition, pair.definition)

feature_links <- feature_pairs %>%
  filter(shared > 0) %>% # arbitrary, what does the # of shared links represent?
  select(item, pair, item.definition, pair.definition)

assoc_network <- graph_from_data_frame(assoc_links, directed=FALSE, vertices=lemma_list) %>%
  simplify()

feature_network <- graph_from_data_frame(feature_links, directed=FALSE, vertices=lemma_list) %>%
  simplify()

networks <- list(assoc_network, feature_network)
```

Network Analysis

Large-Scale Structure of Networks

a la Steyvers 2005

```r
network_properties <- tibble(
  vertices = map_int(networks, vcount),
  edges = map_dbl(networks, ecount) %>%
    as.integer(), # for some reason igraph ecount returns double
  avg_degree = map(networks, degree) %>%
    map_dbl(mean),
  avg_shortest_path = map_dbl(networks, mean_distance),
  diameter = map_dbl(networks, diameter) %>%
    as.integer(),
  clustering_coefficient = map_dbl(networks, transitivity),
  avg_shortest_path_random = 0, # either estimate or calculate values for random nets
  diameter_random = 0,
  clustering_coefficient_random = 0
)

xtable(network_properties)
```

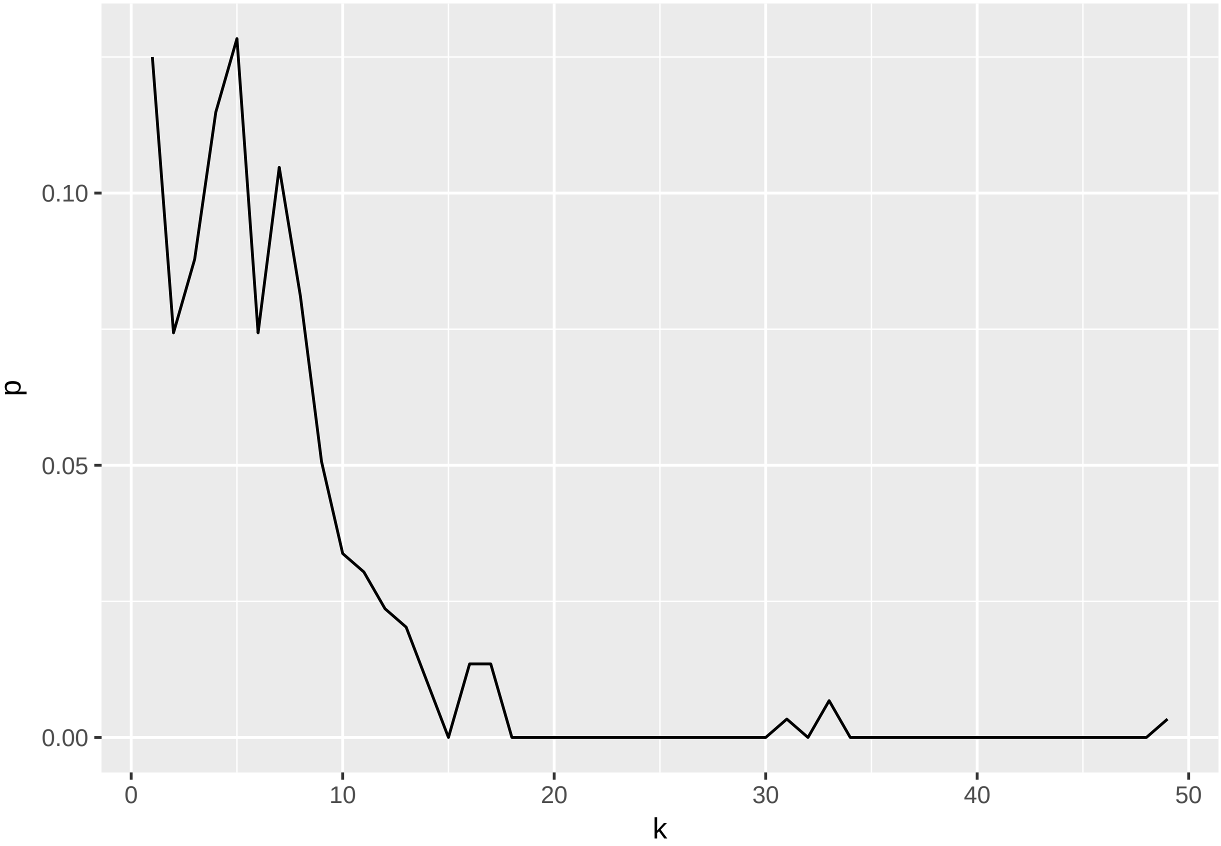
% latex table generated in R 3.4.3 by xtable 1.8-2 package % Sun May 13 00:48:11 2018

Degree Distribution

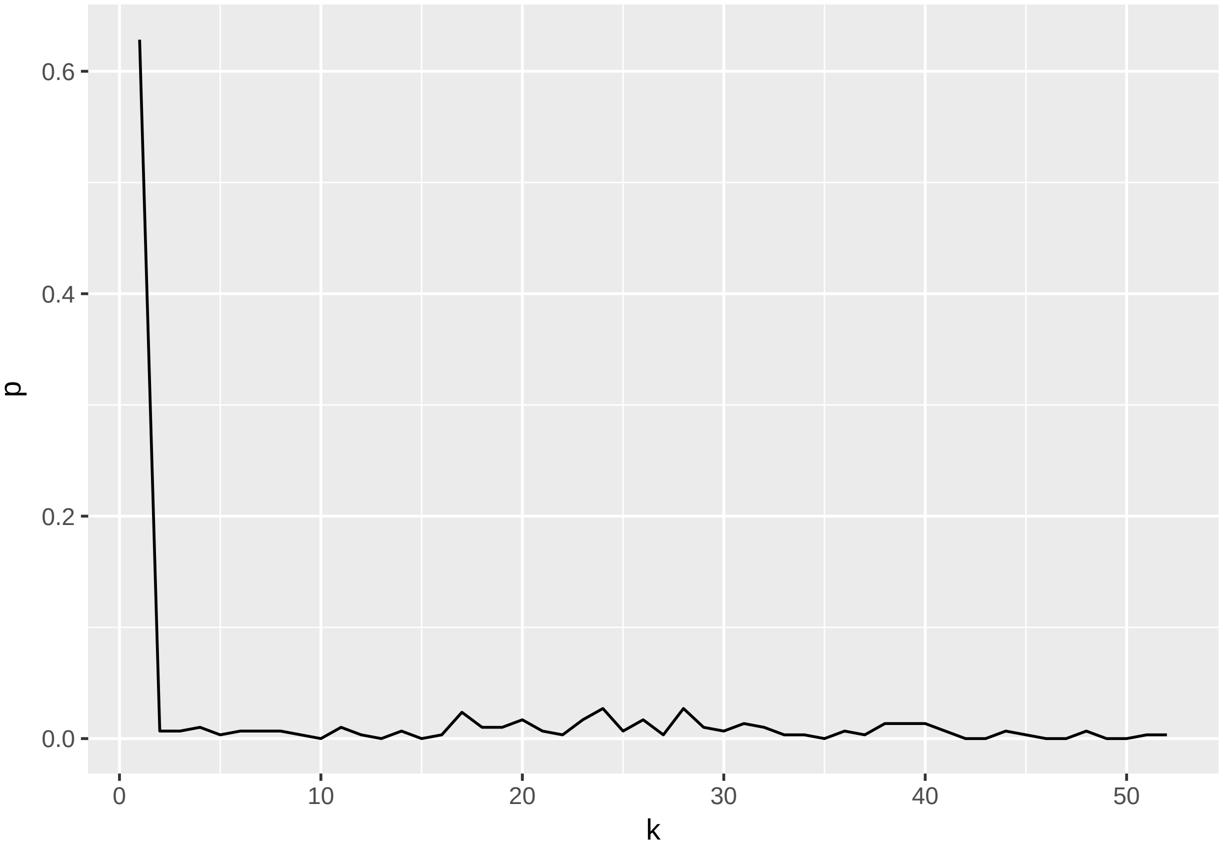
	vertices	edges	avg_degree	avg_shortest_path	diameter	clustering_coefficient	avg_shortest_path_random
1	296	773	5.22	3.61	9	0.17	0.00
2	296	1307	8.83	2.00	4	0.61	0.00

```
assoc_degree <- tibble(k = 1:length(degree_distribution(assoc_network)),
                       p = degree_distribution(assoc_network))
feature_degree <- tibble(k = 1:length(degree_distribution(feature_network)),
                         p = degree_distribution(feature_network))

ggplot(data = assoc_degree, aes(x = k, y = p)) + geom_line()
```



```
ggplot(data = feature_degree, aes(x = k, y = p)) + geom_line()
```


0.6
0.4
0.2
0.0
p
0
10
20
30
40
50
k