

# How to use the NFreq package

CSQ Siew

2017-10-06

## Vignette Info

The NFreq package was built to help the user calculate the neighborhood density/degree and neighborhood frequency of a given list of words. The definition of 'neighbors' is based on the Levenshtein edit distance of 1. Therefore words that differ from each other by the substitution, deletion, or addition of 1 character (phoneme/letter) are considered to be neighbors.

This could be useful to any researcher who is working with linguistic data and wants to calculate the neighborhood density or frequency of words (or nonwords) based on character similarity (i.e., this would work for phonological transcriptions or orthographic representations), and based on a given corpus of data (which contains a set of phonological/orthographic transcriptions and the corresponding frequencies for each word).

This vignette will demonstrate how to use the functions in this package.

## First things first

This package is not hosted on CRAN, so the easiest way to download it is by installing the devtools package and downloading the NFreq package from my github page. Here's how to do it:

```
install.packages('devtools')  
library(devtools)  
devtools::install_github("csqsiew/NFreq")  
library(NFreq)
```

Viola!

## Set up

You will need to upload two sets of data: A list of words that you want to calculate these measures for, and a reference data set (i.e., some linguistic corpus).

Note that it is **very important** that words and data are set up correctly in order for the functions to work (an error message will be returned otherwise). words should be a vector of character type. data should be a dataframe that minimally contains two columns, Phono (e.g., phonological transcriptions) and Frequency (e.g., your favorite log frequency measures). You can double check using the following:

```

words <- c('hWs', 'rod')
class(words)      # character

## [1] "character"

is.vector(words)   # should be TRUE

## [1] TRUE

data <- read.csv('NFreq_data.csv')
head(data)

##   X   Phono Frequency
## 1 1      x     5.3662
## 2 2 ardvar  1.0000
## 3 3   xb@k   1.3010
## 4 4 @bxxks   1.0000
## 5 5   xb@ft   1.0000
## 6 6 @bxloni   1.0000

is.data.frame(data) # should be TRUE

## [1] TRUE

```

It should also be noted that words could consist of nonwords and Phono could be orthographic representations instead (e.g., 'house' instead of /hWs/)--since it is all based on character similarity anyway :)

## The get\_degree function

The get\_degree function calculates the degree or neighborhood density of a word based on Levenshtein edit distance of 1. It returns a dataframe with the words and their corresponding degree.

```

words.degree <- NFreq::get_degree(stimuli = words, database = data)
words.degree

##   Stimuli Degree
## 1     hWs      7
## 2     rod     29

# and then you can output the data if you wish
# write.csv(words.degree, file='word.degree.csv')

```

## The get\_neighbors function

The get\_neighbors function outputs a list of 1-edit distance neighbors of each word. It returns a dataframe with the words and their neighbors. I doubt this function would be used very often, but it might be useful to examine the internal contents of a word's neighborhood.

```
words.neighbors <- NFreq:::get_neighbors(stimuli = words, database = data)
# words.neighbors - did not output data as it is messy
# and then you can output the data if you wish
# write.csv(words.neighbors, file='word.neighbors.csv')
```

## The get\_nfreq function

The get\_nfreq function calculates the neighborhood frequency of a word, which is the average frequency of its neighbors. It returns a dataframe with the words and their corresponding neighborhood frequencies. For this function, your data *must* contain a Frequency column with numeric values.

```
words.nfreq <- NFreq:::get_nfreq(stimuli = words, database = data)
words.nfreq

##   Stimuli NeighborFreq
## 1     hws      1.757343
## 2     rod      1.957445

# and then you can output the data if you wish
# write.csv(words.nfreq, file='word.nfreq.csv')
```

Note that if a word does not have any neighbors, it will have an undefined NeighborFreq value of NaN.

## Comments, suggestions, bugs?

Email me at [cynsiewsq at gmail dot com](mailto:cynsiewsq@gmail.com) - I would love to hear from you! :)