OrdinalRepresentation.R

hamze

2020-01-21

```r
############################################################
############################################################
######################ORDINAL ANLAYSIS######################
############################################################
############################################################

####################################################
#Installing the packages
#install.packages("plyr")
####################################################
#set workspace to this folder
setwd("D:/Work/IJGIS/R-scripts")
####################################################
library(plyr)
```

```
## Warning: package 'plyr' was built under R version 3.5.3
```

```r
###########################PREPROCESSING#########################
fun.to.int <- function(file_address, result_address) {
  raw <- read.table(file = file_address, sep = ",")
  processed = data.frame(V1 = c(raw))
  processed$woQ <- gsub("Q-", "", raw$V1)
  processed$woQA <- gsub("A-", "", processed$woQ)
  write.table(processed$woQA, file=result_address,
              quote = F, sep = " ", row.names = F, col.names = F)
}
fun.to.int("../sequences/prominence-nf-Q.txt",
           "../sequences/prominence-nf-Q-int.txt")
fun.to.int("../sequences/prominence-nf-A.txt",
           "../sequences/prominence-nf-A-int.txt")
fun.to.int("../sequences/scale-nf-Q.txt",
           "../sequences/scale-nf-Q-int.txt")
fun.to.int("../sequences/scale-nf-A.txt",
           "../sequences/scale-nf-A-int.txt")
###########################PROMINENNCE#########################

all_questions <- read.table("../sequences/prominence-nf-Q-int.txt",
                            header = FALSE, sep = " ",
                            col.names = paste0("V",seq_len(4)), fill = TRUE)
all_answers <- read.table("../sequences/prominence-nf-A-int.txt",
                          header = FALSE, sep = " ",
                          col.names = paste0("V",seq_len(13)), fill = TRUE)

differential_mat = matrix(0, nrow = 19, ncol = 1) #prominence

for (i in 1:length(all_questions$V1)) {
  id = all_questions[i, 1]

  num_vec = as.numeric(all_questions[i,2:4])
  num_vec = num_vec[!is.na(num_vec)]

  min_val = min(num_vec)

  ans_vec = all_answers[all_answers$V1 == id, 2:13]
  ans_vec = ans_vec[!is.na(ans_vec)]
  diff_ans = ans_vec - min_val

  for (j in 1:length(diff_ans)) {
    differential_mat[diff_ans[j] + 10, 1] = differential_mat[diff_ans[j] + 10, 1] + 1
  }
}
```

```
## Warning in min(num_vec): no non-missing arguments to min; returning Inf

## Warning in min(num_vec): no non-missing arguments to min; returning Inf
```

```
df_differrential = as.data.frame(differential_mat)
colnames(df_differrential) = c("frequency")
rownames(df_differrential) = as.character(-9:9)
df_differrential
```

```
##    frequency
## -9         0
## -8         0
## -7         0
## -6         9
## -5        60
## -4       125
## -3       226
## -2       524
## -1       771
## 0       1794
## 1       2103
## 2       1883
## 3       1561
## 4       1566
## 5       1031
## 6        180
## 7          0
## 8          0
## 9          0
```

```
write.csv(df_differential, "result/differential_prominence.csv") #prominence

###########################Scale#########################

all_questions <- read.table("../sequences/scale-nf-Q-int.txt",
                            header = FALSE, sep = " ",
                            col.names = paste0("V",seq_len(4)), fill = TRUE)
all_answers <- read.table("../sequences/scale-nf-A-int.txt",
                          header = FALSE, sep = " ",
                          col.names = paste0("V",seq_len(13)), fill = TRUE)


differential_mat = matrix(0, nrow = 21, ncol = 1) #scale

for (i in 1:length(all_questions$V1)) {
  id = all_questions[i, 1]

  num_vec = as.numeric(all_questions[i,2:4])
  num_vec = num_vec[!is.na(num_vec)]

  min_val = min(num_vec)

  ans_vec = all_answers[all_answers$V1 == id, 2:13]
  ans_vec = ans_vec[!is.na(ans_vec)]
  diff_ans = ans_vec - min_val

  for (j in 1:length(diff_ans)) {
    differential_mat[diff_ans[j] + 11, 1] = differential_mat[diff_ans[j] + 11, 1] + 1 #scale
  }
}




df_differential = as.data.frame(differential_mat)
colnames(df_differential) = c("frequency")
rownames(df_differential) = as.character(-10:10)
df_differential
```

```
##     frequency
## -10         0
## -9          0
## -8          0
## -7          0
## -6          0
## -5         18
## -4         33
## -3         71
## -2        167
## -1        515
## 0        1008
## 1        1949
## 2        1160
## 3        1353
## 4         392
## 5         101
## 6           7
## 7           0
## 8           0
## 9           0
## 10          0
```

```
write.csv(df_differrential, "result/differential_scale.csv") #scale
```