type-categorical-distribution.R

hamze

2020-01-22

```r
##########################################################
##########################################################
#########################Type Analysis#####################
##########################################################
##########################################################

#####################################################
#Installing the packages
#install.packages("TraMineR")
#install.packages("TraMineRextras")
#install.packages("dplyr")
#install.packages("ggplot2")
#install.packages("RColorBrewer")
#install.packages("fpc")
#####################################################
#set workspace to this folder
setwd("D:/Work/IJGIS/R-scripts")
#####################################################
#####################Libraries##########################
library(TraMineR)
```

```
##
## TraMineR stable version 2.0-14 (Built: 2020-01-19)

## Website: http://traminer.unige.ch

## Please type 'citation("TraMineR")' for citation information.
```

```r
library(TraMineRextras)
```

```
## TraMineRextras stable version 0.4.6 (Built: 2020-01-19)

## Functions provided by this package are still in test

##     and subject to changes in future releases.

##
## Attaching package: 'TraMineRextras'
```

```
## The following objects are masked from 'package:TraMineR':
##
##     seqprecarity, seqprecorr, seqprecstart
```

```r
library(dplyr)
```

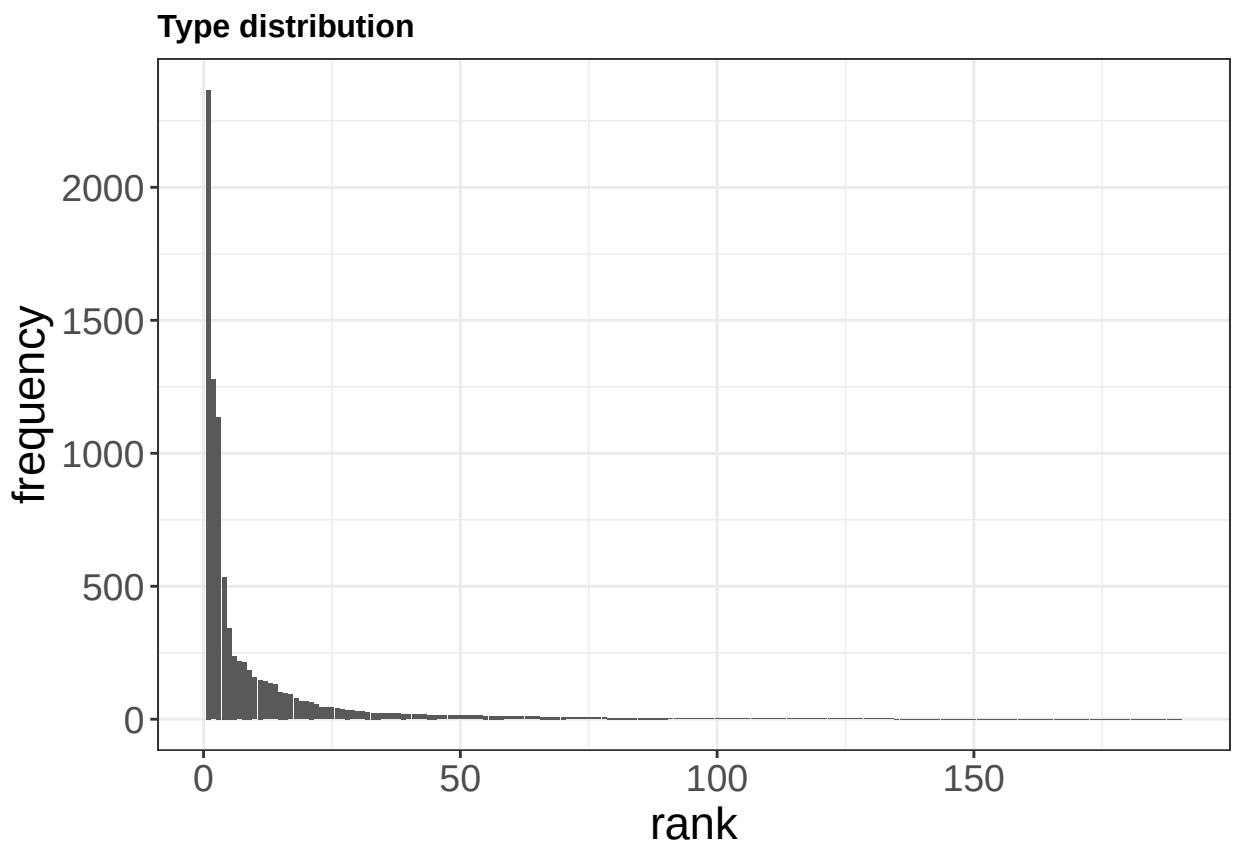
```
##
## Attaching package: 'dplyr'

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

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

```r
library(ggplot2)
library(RColorBrewer)

library(cluster)
library(fpc)
#########################################################
fun.to.type <- function(file_address, result_address) {
  raw <- read.table(file = file_address, sep = ",")
  processed = data.frame(V1 = c(raw))
  processed$woQ <- gsub("Q-", "T", raw$V1)
  processed$woQA <- gsub("A-", "T", processed$woQ)
  write.table(processed$woQA, file=result_address,
              quote = F, sep = " ", row.names = F, col.names = F)
}
#########################################################
#########READING FILES#########

all_questions <- read.table("../sequences/type-nf-Q.txt",
                            header = FALSE, sep = " ",
                            col.names = paste0("V",seq_len(5)), fill = TRUE)
all_answers <- read.table("../sequences/type-nf-A.txt",
                          header = FALSE, sep = " ",
                          col.names = paste0("V",seq_len(13)), fill = TRUE)
fun.to.type("../sequences/type-nf-all.txt",
            "../sequences/type-nf-all-t.txt")
all_qas <- read.table("../sequences/type-nf-all-t.txt",
                      header = FALSE, sep = " ",
                      col.names = paste0("V",seq_len(13)), fill = TRUE)

types_q = as.data.frame(table(all_answers$V2))

vector_a = all_answers$V3
for (i in 1:10) {
  vector_a = c(as.character(vector_a), as.character(all_answers[,i+3]))
}
```

```r
types = as.data.frame(table(vector_a))

write.csv(types_q, file="result/types_q.csv")
write.csv(types, file="result/types_a.csv")

aaID = all_answers[,1:6]
aa =all_answers[,2:6]
qqID = all_answers[,1:4]
qq = all_questions[, 2:4]
aq = all_qas[, 2:8]

#########FUNCTIONS#########
fun.histogram = function (df) {
  result = df %>% dplyr::group_by(df[,1]) %>% dplyr::summarize(count=dplyr::n())
  names(result) <- c("type", "count")
  if (length(df[1,]) > 1) {
  for (i in 2:length(df)) {
    temp = df %>% dplyr::group_by(df[,i]) %>% dplyr::summarize(count=dplyr::n())
    names(temp) <- c("type", "count")
    result = rbind(result, temp)
  }
  result <- result %>% dplyr::group_by(type) %>% dplyr::summarize(total=sum(count))
  #result <- result[2:length(result$type),]
  result <- as.data.frame(result[order(result$total,  decreasing = TRUE),])
  result <- result[order(as.character(result$type)), ]
  }
  return (result)
}

fun.naming = function(df) {
  for (i in 1:length(df)) {
    names(df)[i] = as.character(i)
  }
  return (df)
}

fun.change.to.other = function (df, number) {
  result <- data.frame(lapply(df, as.character), stringsAsFactors=FALSE)
  all_types_total = fun.histogram(result)
  all_types_total = as.data.frame(all_types_total[order(all_types_total$total,  decreasing = TRUE),])
  selected <- c(as.character(all_types_total[0:number,]$type)) #"Q-" issue
  for (i in 1:length(df)) {
    temp = result[, i]
    temp[!temp %in% selected] = "OTHER"
    temp[temp %in% c("Q-")] = "OTHER"
    result[, i] = temp
  }
  return (data.frame(lapply(result, as.factor), stringsAsFactors=FALSE))
}


cstats.table <- function(dist, tree, k) {
  clust.assess <- c("cluster.number","n","within.cluster.ss","average.within","average.between",
```

```
                    "wb.ratio","dunn2","avg.silwidth")
  clust.size <- c("cluster.size")
  stats.names <- c()
  row.clust <- c()
  output.stats <- matrix(ncol = k, nrow = length(clust.assess))
  cluster.sizes <- matrix(ncol = k, nrow = k)
  for(i in c(1:k)){
    row.clust[i] <- paste("Cluster-", i, " size")
  }
  for(i in c(2:k)){
    stats.names[i] <- paste("Test", i-1)

    for(j in seq_along(clust.assess)){
      output.stats[j, i] <- unlist(cluster.stats(d = dist, clustering = cutree(tree, k = i))[clust.asses

    }

    for(d in 1:k) {
      cluster.sizes[d, i] <- unlist(cluster.stats(d = dist, clustering = cutree(tree, k = i))[clust.size
      dim(cluster.sizes[d, i]) <- c(length(cluster.sizes[i]), 1)
      cluster.sizes[d, i]

    }
  }
  output.stats.df <- data.frame(output.stats)
  cluster.sizes <- data.frame(cluster.sizes)
  cluster.sizes[is.na(cluster.sizes)] <- 0
  rows.all <- c(clust.assess, row.clust)
  # rownames(output.stats.df) <- clust.assess
  output <- rbind(output.stats.df, cluster.sizes)[ ,-1]
  colnames(output) <- stats.names[2:k]
  rownames(output) <- rows.all
  is.num <- sapply(output, is.numeric)
  output[is.num] <- lapply(output[is.num], round, 2)
  output
}
############setting########

getPalette = colorRampPalette(brewer.pal(12, "Paired")) ###only for categories
colourCount <- 21

gt <- getPalette(colourCount)

###Shuffeling!!
gtA <- c(gt)
gtX <- list(gtA)
gt <- gtX[[1]][sample(1:length(gtA))]
gt = gt[sample(1:length(gt))]

############question
agg_qs = fun.histogram(all_questions[,2:5])
write.csv(file = "result/types_in_question_agg.csv", x = agg_qs)
agg_qq = fun.histogram(qq)
```

```
agg_qq = as.data.frame(agg_qq[order(agg_qq$total,  decreasing = TRUE),])
agg_qq = agg_qq[2:length(agg_qq[,1]),]
agg_qq$order = c(1:length(agg_qq$type))
ggplot(agg_qq, aes(x = order, y = total))+geom_bar(stat = "identity")+labs(title="Type distribution",x=
  theme_bw() + theme(plot.title = element_text(color = "black", size = "12", face = "bold"), text = ele
```




```
qq = fun.change.to.other(qq, 21)
qq = fun.naming(qq)

all_sequence <- seqdef(qq)
```

```
##  [!] found '-' character in state codes, not recommended

##  [>] 21 distinct states appear in the data:

##      1 =

##      2 = OTHER

##      3 = Q-ADM1

##      4 = Q-ADM2
```

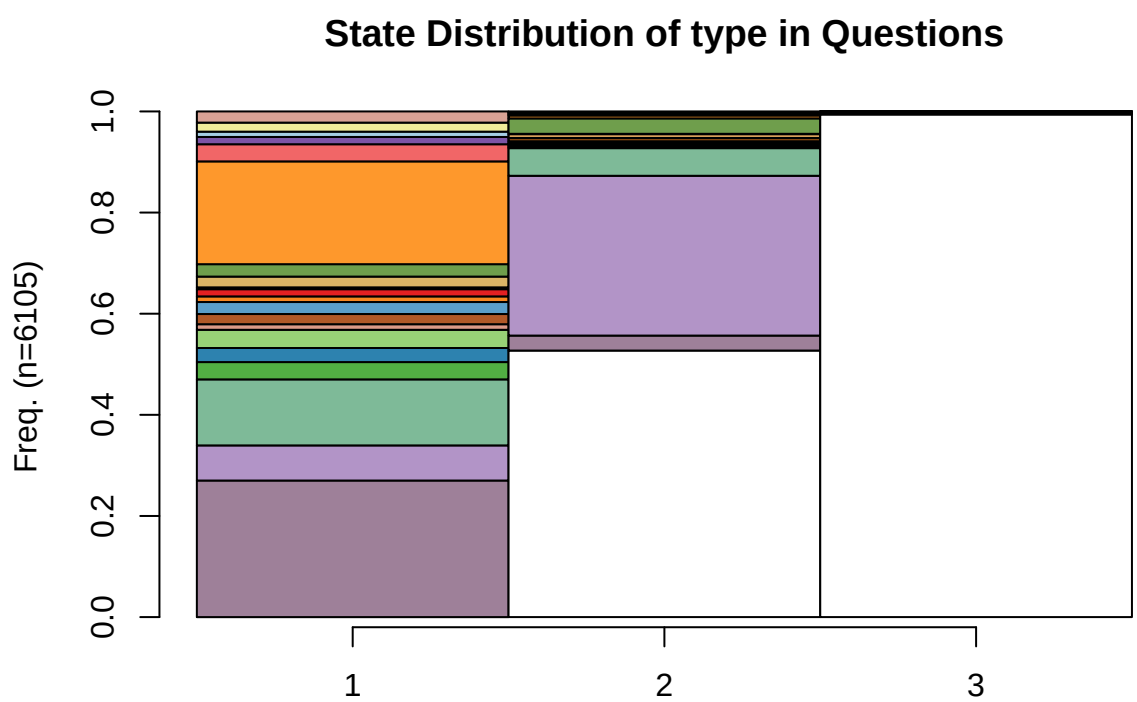
```
##      5 = Q-ADM3

##      6 = Q-ADM4

##      7 = Q-AREA

##      8 = Q-FRM

##      9 = Q-HTL

##      10 = Q-ISL

##      11 = Q-LCTY

##      12 = Q-LK

##       ...

## Warning:  [!] No automatic color palette assigned because number of states > 12.
##
##      Use 'cpal' argument to assign one.

##  [>] state coding:

##        [alphabet]  [label]  [long label]

##      1

##      2  OTHER       OTHER    OTHER

##      3  Q-ADM1      Q-ADM1   Q-ADM1

##      4  Q-ADM2      Q-ADM2   Q-ADM2

##      5  Q-ADM3      Q-ADM3   Q-ADM3

##      6  Q-ADM4      Q-ADM4   Q-ADM4

##      7  Q-AREA      Q-AREA   Q-AREA

##      8  Q-FRM       Q-FRM    Q-FRM

##      9  Q-HTL       Q-HTL    Q-HTL

##      10  Q-ISL       Q-ISL    Q-ISL

##      11  Q-LCTY      Q-LCTY   Q-LCTY
```

```
##      12  Q-LK        Q-LK     Q-LK

##       ... (21 states)

##  [>] no color palette attributed, provide one to use graphical functions

##  [>] 6105 sequences in the data set

##  [>] min/max sequence length: 3/3
```

```r
cpal(all_sequence)
```

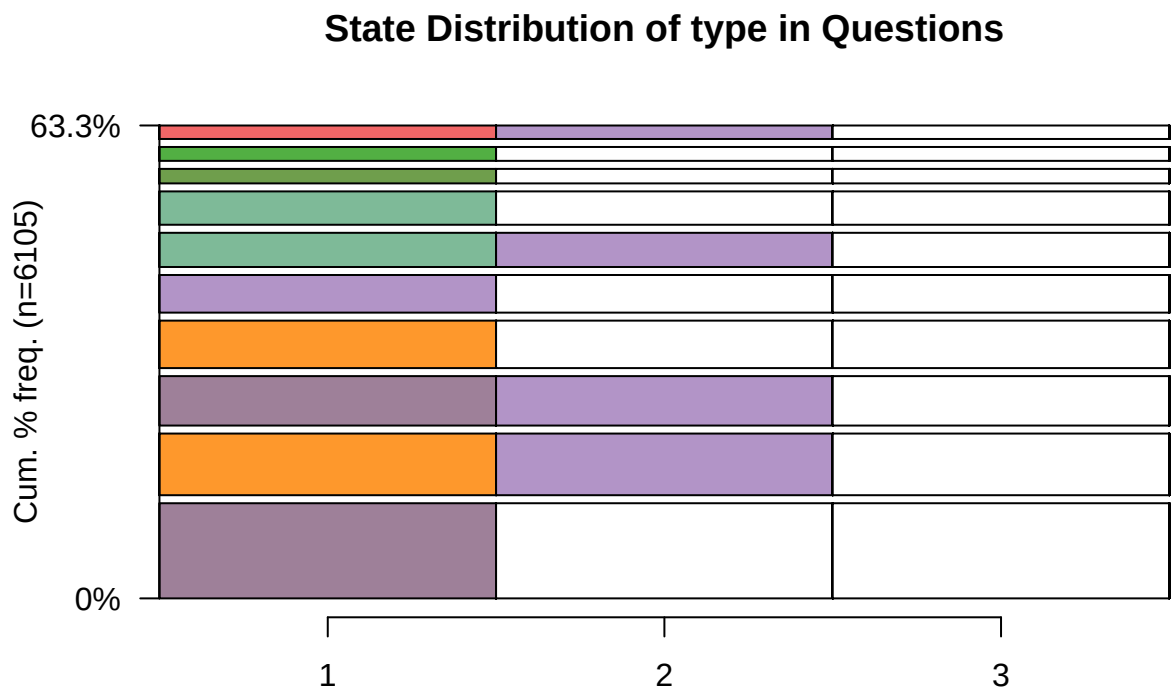
```
## NULL
```

```r
#temp = gt[1]
gt[1] <- "#FFFFFF"
allAts = attributes(all_sequence)
write.csv(x = allAts$cpal, file = "result/PERSIST_type_q_cpal")
write.csv(x = allAts$alphabet, file = "result/PERSIST_type_q_alphabet")
write.csv(x = allAts$labels, file = "result/PERSIST_type_q_labels")

attr(all_sequence, "cpal") <- gt


seqdplot(all_sequence, with.legend = F, border = T,
         main = "State Distribution of type in Questions")
```

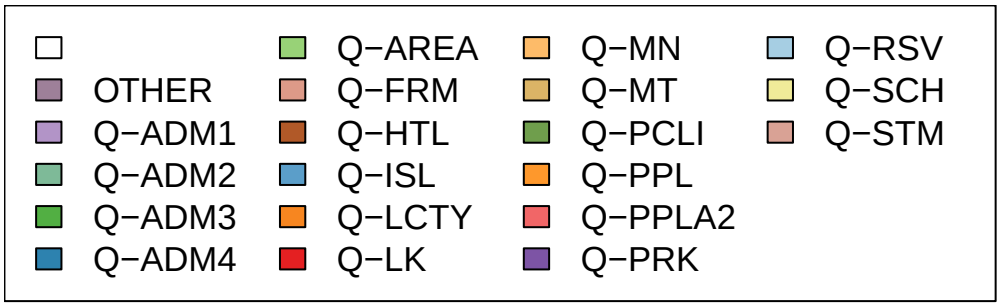


```r
seqfplot(all_sequence, with.legend = F, border = T,
         main = "State Distribution of type in Questions")
```

```
## Warning in (function (seqdata, idxs = 1:10, weighted = TRUE, format = "SPS", :
## '-' character in states codes may cause invalid results
```

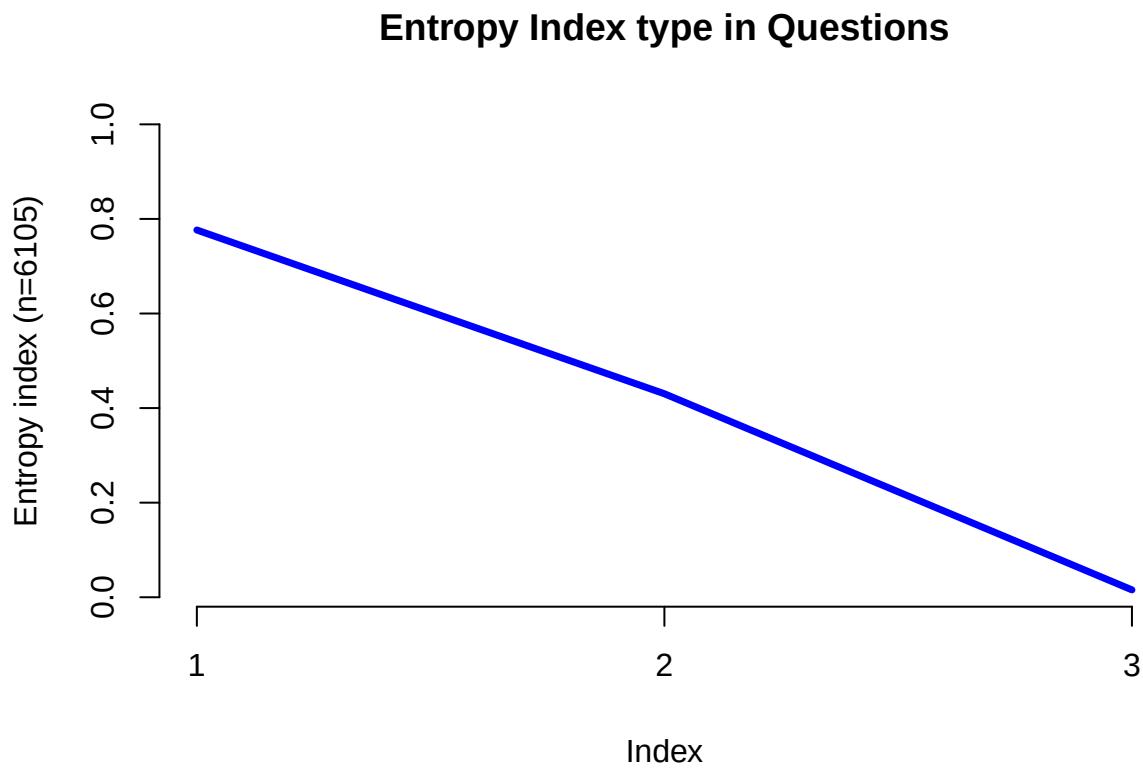



```r
seqlegend(all_sequence, cex=1.1, ncol=4)
```

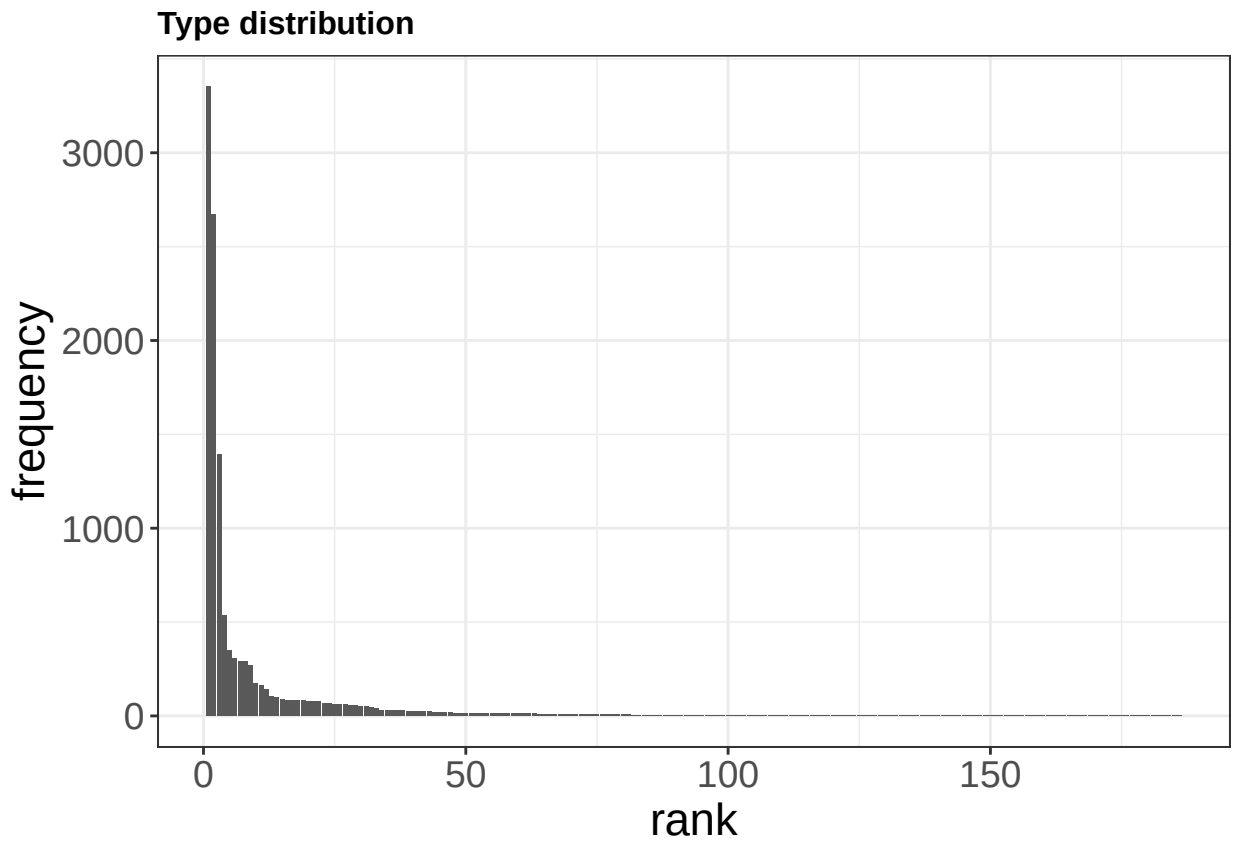



```r
seqHtplot(all_sequence, title = "Entropy Index type in Questions")
```

```
##  [!] In rmarkdown::render() : title is deprecated, use main instead.
```




```
############answers
agg_as = fun.histogram(all_answers[,2:13])
write.csv(file = "result/types_in_answers_agg.csv", x = agg_as)

agg_aa = fun.histogram(aa)
agg_aa = as.data.frame(agg_aa[order(agg_aa$total,  decreasing = TRUE),])
agg_aa = agg_aa[2:length(agg_aa[,1]),]
agg_aa$order = c(1:length(agg_aa$type))
ggplot(agg_aa, aes(x = order, y = total))+geom_bar(stat = "identity")+
  labs(title="Type distribution",x="rank", y = "frequency") +
  theme_bw() + theme(
    plot.title = element_text(color = "black", size = "12", face = "bold"),
    text = element_text(color = "black", size=17))
```




```
aa = fun.change.to.other(aa, 21)
aa = fun.naming(aa)

all_sequence <- seqdef(aa)
```

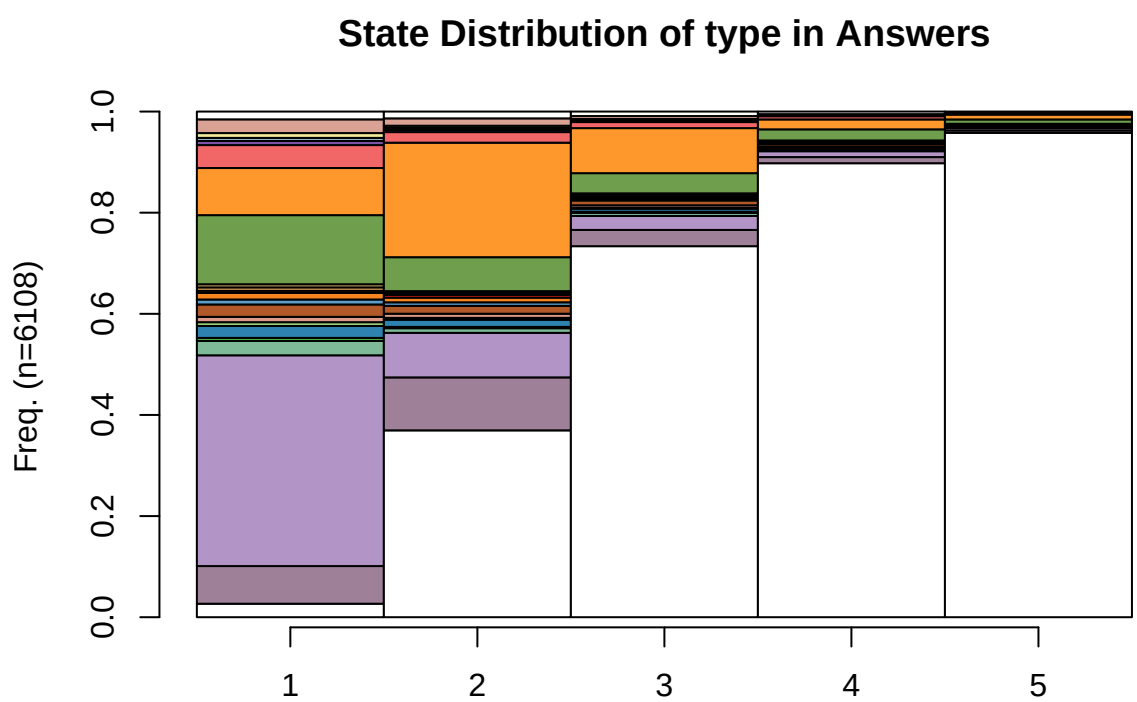
```
##  [>] 22 distinct states appear in the data:

##      1 =

##      2 = ADM1

##      3 = ADM2

##      4 = ADM3

##      5 = ADM4

##      6 = AREA

##      7 = CONT

##      8 = FRM
```

```
##      9 = HTL

##      10 = ISL

##      11 = LCTY

##      12 = MN

##       ...

## Warning:  [!] No automatic color palette assigned because number of states > 12.
##
##      Use 'cpal' argument to assign one.

##  [>] state coding:

##        [alphabet]  [label]  [long label]

##      1

##      2  ADM1        ADM1     ADM1

##      3  ADM2        ADM2     ADM2

##      4  ADM3        ADM3     ADM3

##      5  ADM4        ADM4     ADM4

##      6  AREA        AREA     AREA

##      7  CONT        CONT     CONT

##      8  FRM         FRM      FRM

##      9  HTL         HTL      HTL

##      10  ISL         ISL      ISL

##      11  LCTY        LCTY     LCTY

##      12  MN          MN       MN

##       ... (22 states)

##  [>] no color palette attributed, provide one to use graphical functions

##  [>] 6108 sequences in the data set

##  [>] min/max sequence length: 5/5
```

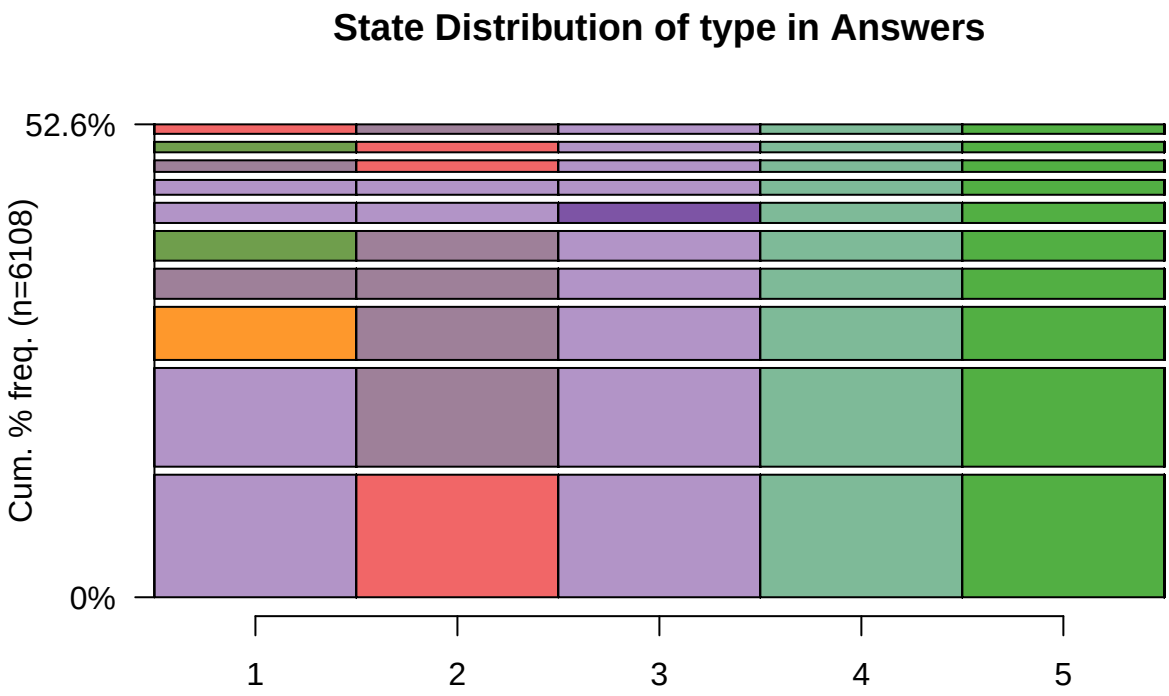
```r
cpal(all_sequence)
```

```
## NULL
```

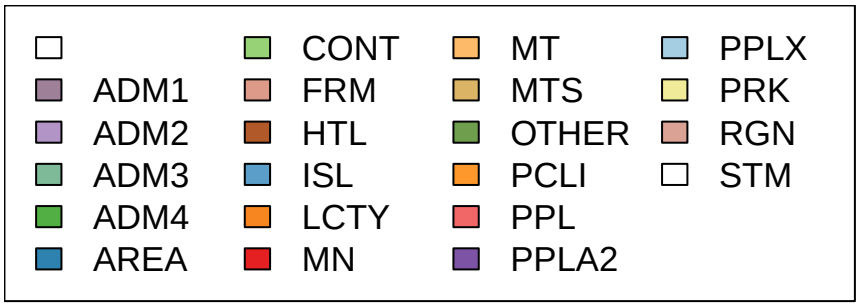
```r
attr(all_sequence, "cpal") <- gt

allAts = attributes(all_sequence)
write.csv(x = allAts$alphabet, file = "result/PERSIST_type_a_alphabet")
write.csv(x = allAts$labels, file = "result/PERSIST_type_a_labels")

seqdplot(all_sequence, with.legend = F, border = T,
         main = "State Distribution of type in Answers")
```




```r
seqfplot(all_sequence, with.legend = F, border = T,
         main = "State Distribution of type in Answers")
```

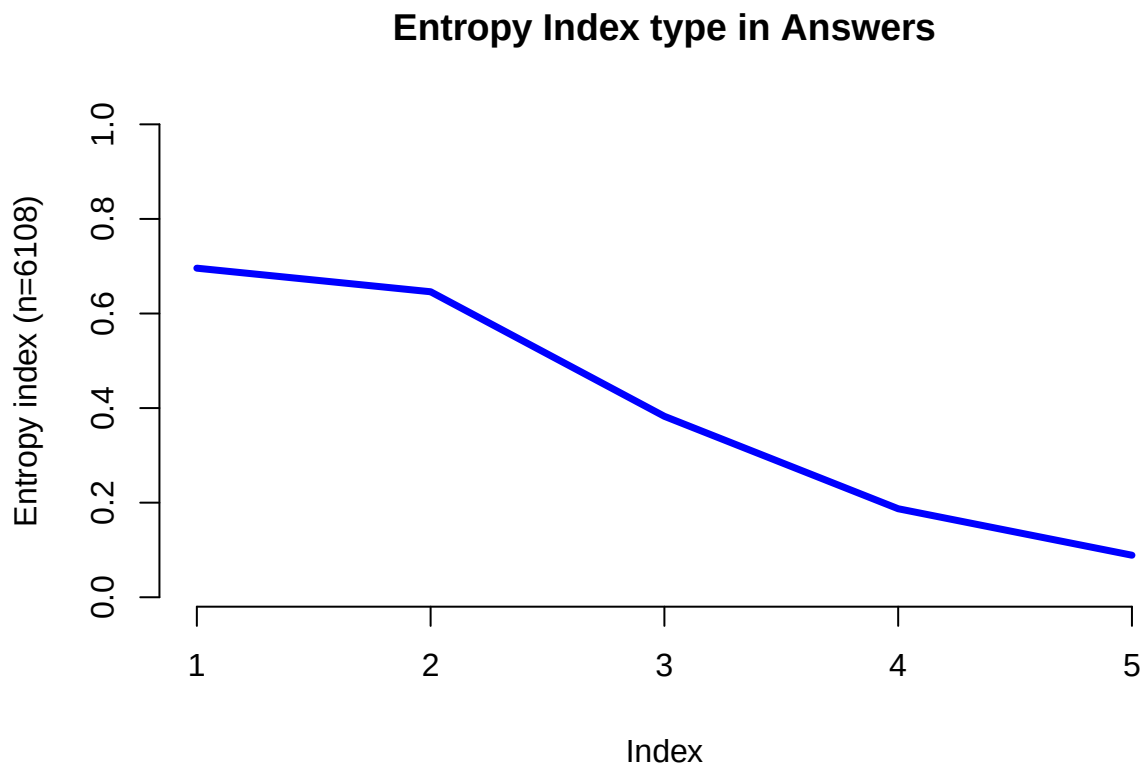



```r
seqlegend(all_sequence, cex=1.1, ncol=4)
```

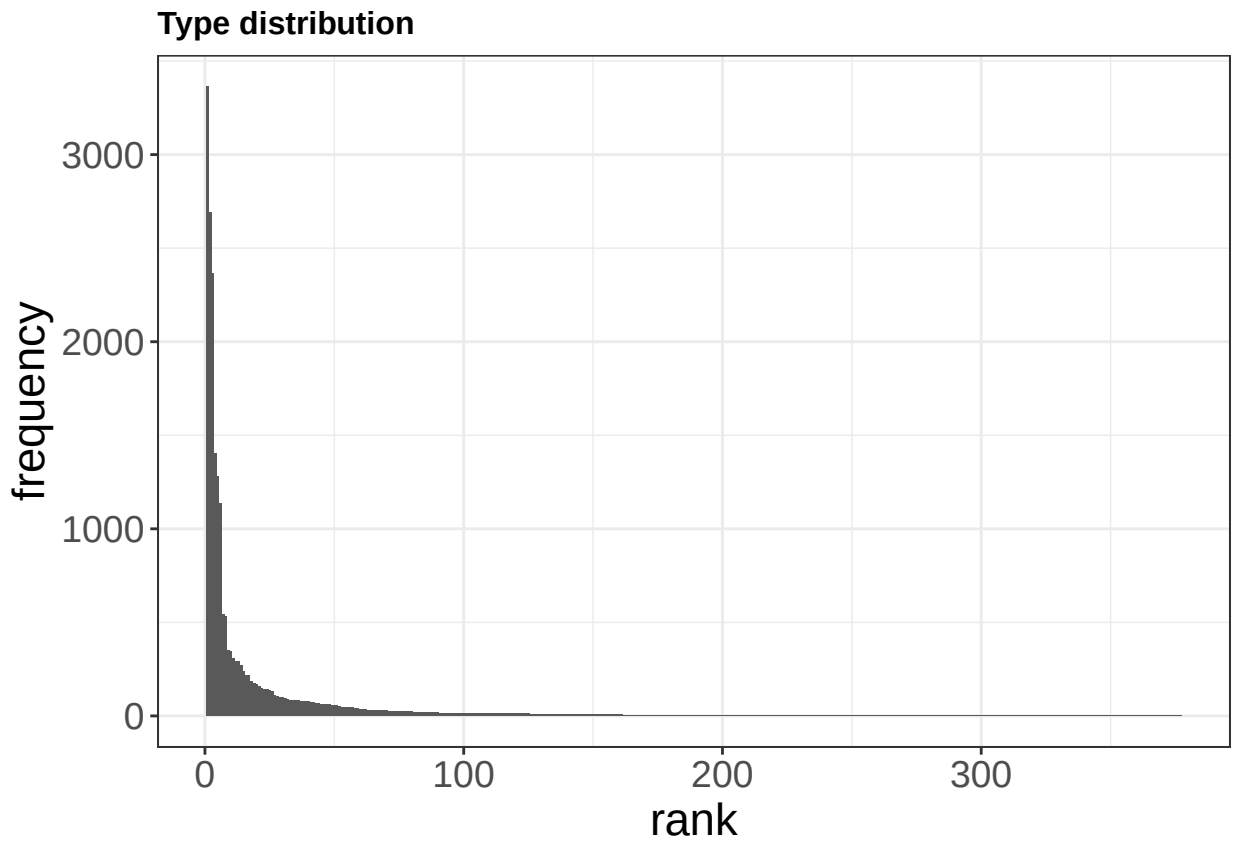



```r
seqHtplot(all_sequence, title = "Entropy Index type in Answers")
```

```
##  [!] In rmarkdown::render() : title is deprecated, use main instead.
```




```
############concatenated q-a
agg_aq = fun.histogram(aq)
agg_aq = as.data.frame(agg_aq[order(agg_aq$total,  decreasing = TRUE),])
agg_aq = agg_aq[2:length(agg_aq[,1]),]
agg_aq$order = c(1:length(agg_aq$type))
ggplot(agg_aq, aes(x = order, y = total))+geom_bar(stat = "identity")+
  labs(title="Type distribution",x="rank", y = "frequency") +
  theme_bw() + theme(
    plot.title = element_text(color = "black", size = "12", face = "bold"),
    text = element_text(color = "black", size=17))
```




```
#####################################################################
##########################SWQ VS DWQ#############################
#####################################################################

fun.extract.ncomplex.ids = function(questions, n) {
  validIds = c()
  counter = 0
  for (i in 1:length(questions[,1])) {
    qVals = questions[i, 2:5]
    if (length(qVals[qVals!=""]) == n) {
      counter= counter + 1
      validIds[counter] = questions[i, 1]
    }
  }
  return (validIds)
}


#########################################SWQ#######################################
Q_swq <- read.table("../sequences/type-nf-Q-SWQ.txt",
                    header = FALSE, sep = " ",
                    col.names = paste0("V",seq_len(5)), fill = TRUE)
Q_swq = as.data.frame(Q_swq[, 1])
A_swq <- read.table("../sequences/type-nf-A-SWQ.txt",
                    header = FALSE, sep = " ",
                    col.names = paste0("V",seq_len(20)), fill = TRUE)
```

```r
agg_swq_q = fun.histogram(Q_swq)
agg_swq_a = fun.histogram(A_swq)
```

```
## Warning: Factor `type` contains implicit NA, consider using
## `forcats::fct_explicit_na`
```

```r
write.csv(file = "result/types_in_question_agg_swq.csv", x = agg_swq_q)
write.csv(file = "result/types_in_answers_agg_swq.csv", x = agg_swq_a)

QA_swq <- A_swq
QA_swq[,1] <- Q_swq
QA_swq[, 2:7] <- A_swq[,1:6]
colnames(QA_swq) <- c("Q", "A1", "A2", "A3", "A4", "A5", "A6")

aa = QA_swq
aa = fun.change.to.other(aa, 16)
aa = fun.naming(aa)

all_sequence <- seqdef(aa)
```

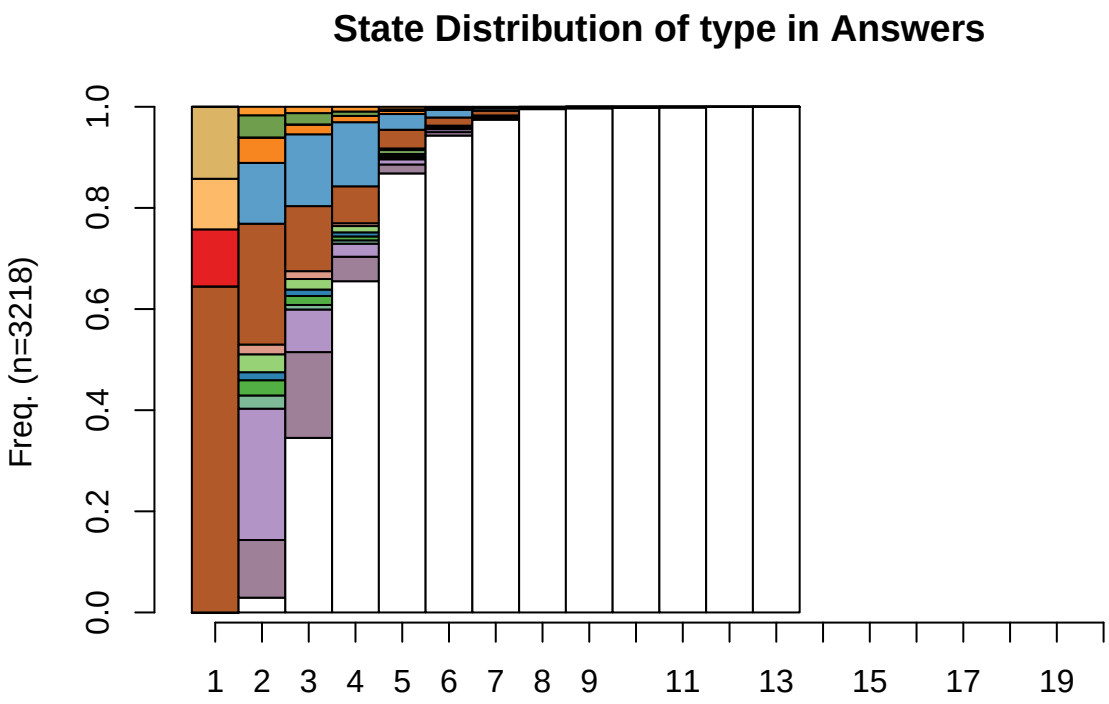
```
##  [!] found '-' character in state codes, not recommended

##  [>] found missing values ('NA') in sequence data

##  [>] preparing 3218 sequences

##  [>] coding void elements with '%' and missing values with '*'

##  [>] 16 distinct states appear in the data:

##      1 =

##      2 = ADM1

##      3 = ADM2

##      4 = ADM3

##      5 = AREA

##      6 = FRM

##      7 = HTL

##      8 = LCTY

##      9 = OTHER
```

```
##      10 = PCLI

##      11 = PPL

##      12 = Q-ADM1

##       ...

## Warning:  [!] No automatic color palette assigned because number of states > 12.
##
##      Use 'cpal' argument to assign one.

##  [>] state coding:

##        [alphabet]  [label]  [long label]

##      1

##      2  ADM1        ADM1     ADM1

##      3  ADM2        ADM2     ADM2

##      4  ADM3        ADM3     ADM3

##      5  AREA        AREA     AREA

##      6  FRM         FRM      FRM

##      7  HTL         HTL      HTL

##      8  LCTY        LCTY     LCTY

##      9  OTHER       OTHER    OTHER

##      10  PCLI        PCLI     PCLI

##      11  PPL         PPL      PPL

##      12  Q-ADM1      Q-ADM1   Q-ADM1

##       ... (16 states)

##  [>] no color palette attributed, provide one to use graphical functions

##  [>] 3218 sequences in the data set

##  [>] min/max sequence length: 13/13
```

```r
cpal(all_sequence)
```

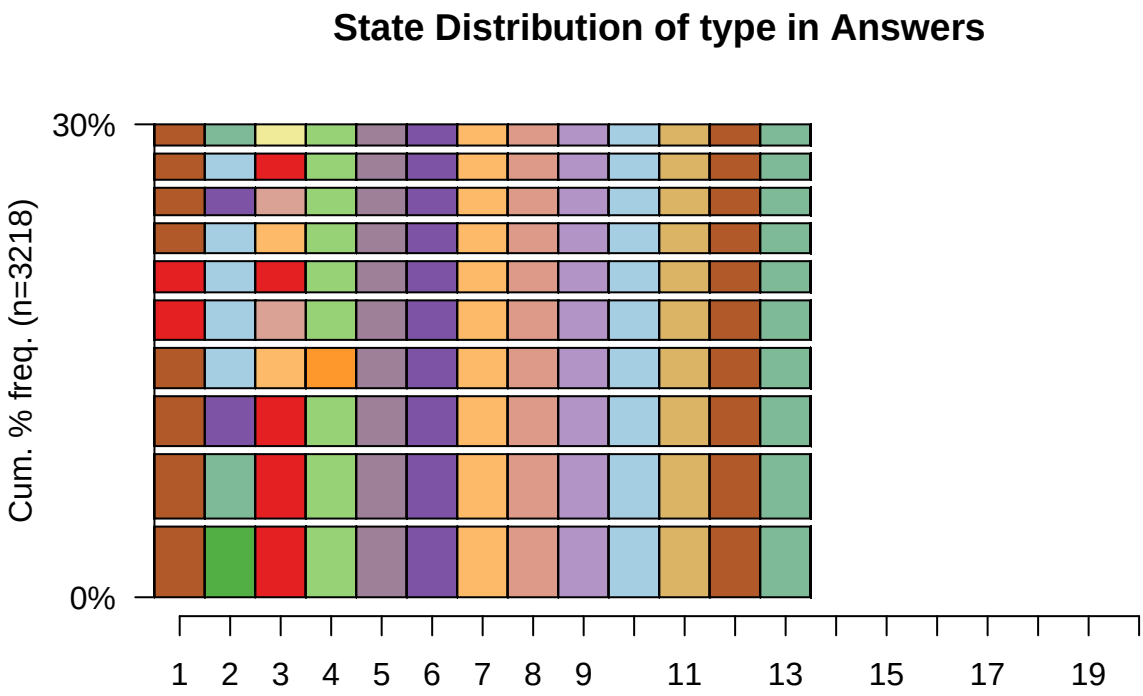
```
## NULL
```

```r
attr(all_sequence, "cpal") <- gt

allAts = attributes(all_sequence)

seqdplot(all_sequence, with.legend = F, border = T,
         main = "State Distribution of type in Answers")
```

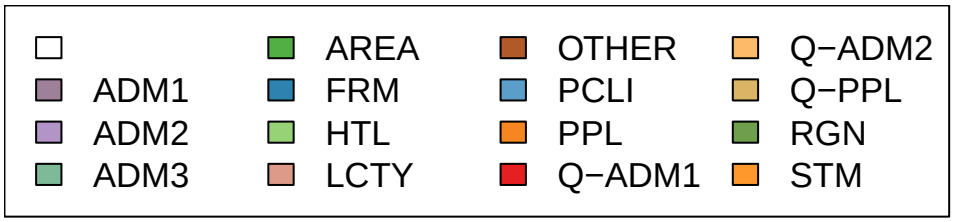



```r
seqfplot(all_sequence, with.legend = F, border = T,
         main = "State Distribution of type in Answers")
```

```
## Warning in (function (seqdata, idxs = 1:10, weighted = TRUE, format = "SPS", :
## '-' character in states codes may cause invalid results
```

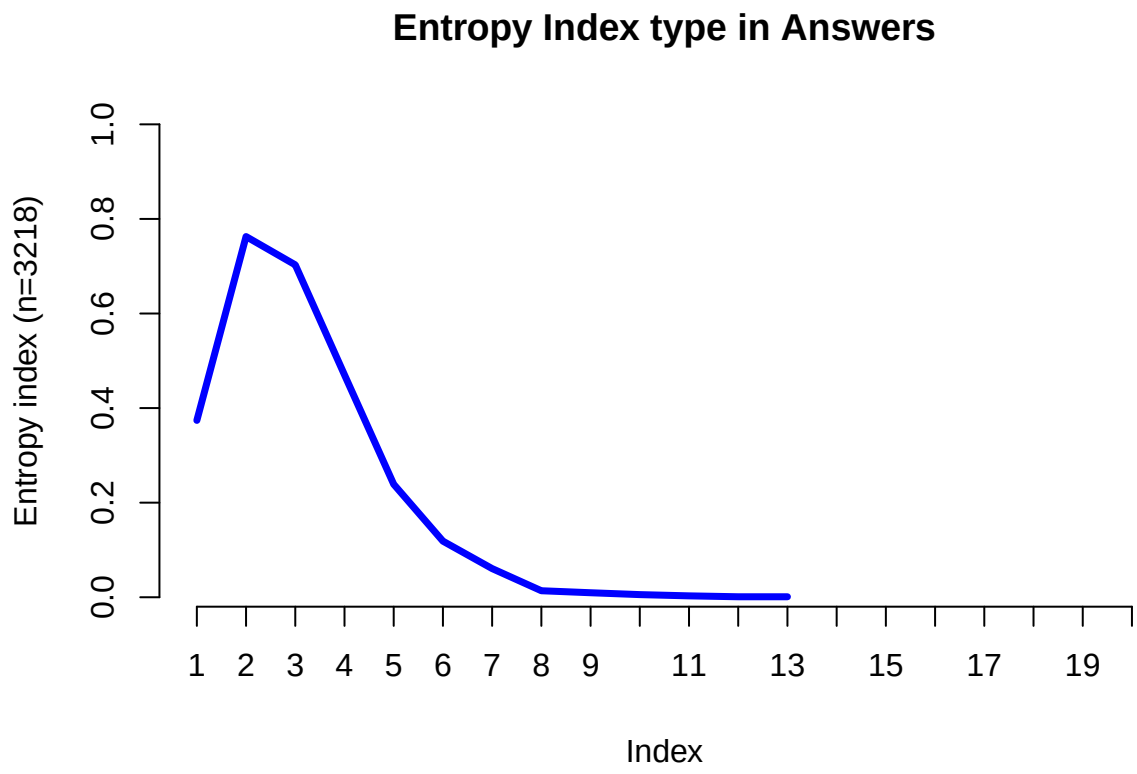



```
seqlegend(all_sequence, cex=1.1, ncol=4)
```




```r
seqHtplot(all_sequence, title = "Entropy Index type in Answers")
```

```
##  [!] In rmarkdown::render() : title is deprecated, use main instead.
```




```
#######################################DWQ#######################################
Q_rwq <- read.table("../sequences/type-nf-Q-DWQ.txt", header = FALSE,
                    sep = " ", col.names = paste0("V",seq_len(4)), fill = TRUE)
A_rwq <- read.table("../sequences/type-nf-A-DWQ.txt", header = FALSE,
                    sep = " ", col.names = paste0("V",seq_len(14)), fill = TRUE)

agg_rwq_q = fun.histogram(Q_rwq)
agg_rwq_a = fun.histogram(A_rwq)
```