scale-categorical-distribution.R

hamze

2020-01-22

```
############################################################
############################################################
#######################Scale 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.
```

```
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'
```

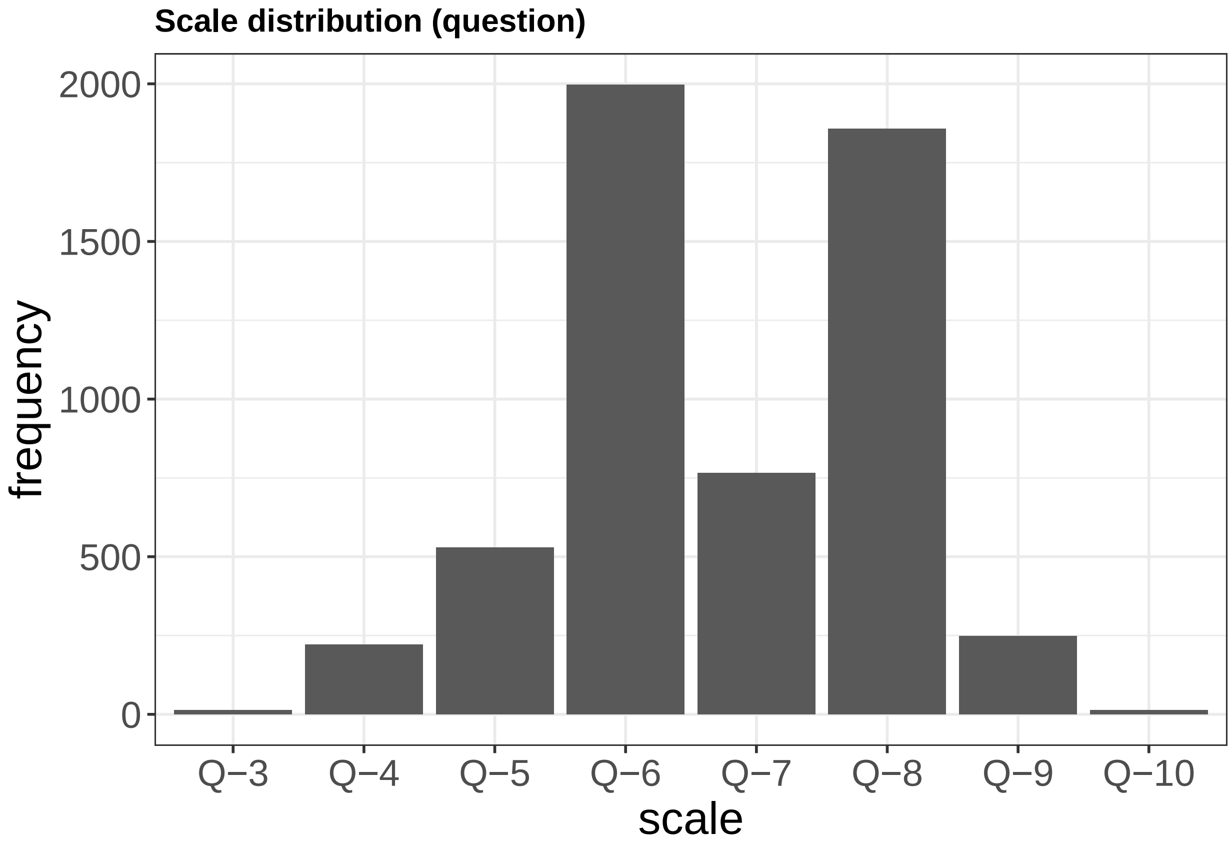
```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

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

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

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

all_questions <- read.table("../sequences/scale-nf-Q.txt",
                            header = FALSE, sep = " ",
                            col.names = paste0("V",seq_len(5)), fill = TRUE)
all_answers <- read.table("../sequences/scale-nf-A.txt",
                          header = FALSE, sep = " ",
                          col.names = paste0("V",seq_len(13)), fill = TRUE)
all_qas <- read.table("../sequences/scale-nf-all-s.txt"
                      , header = FALSE, sep = " ",
                      col.names = paste0("V",seq_len(17)), fill = TRUE)
scales_q = as.data.frame(table(all_answers$V2))

vector_a = all_answers$V3
for (i in 1:9) {
  vector_a = c(as.character(vector_a), as.character(all_answers[,i+3]))
}
```

```
scales = as.data.frame(table(vector_a))

write.csv(scales_q, file="result/scales_q.csv")
write.csv(scales, file="result/scales_a.csv")

aa =all_answers#[,2:6]
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("scale", "count")
  for (i in 2:length(df)) {
    temp = df %>% dplyr::group_by(df[,i]) %>% dplyr::summarize(count=dplyr::n())
    names(temp) <- c("scale", "count")
    result = rbind(result, temp)
  }
  result <- result %>% dplyr::group_by(scale) %>% dplyr::summarize(total=sum(count))
  result <- result[2:length(result$scale),]
  result <- as.data.frame(result[order(result$total,  decreasing = TRUE),])
  result <- result[order(as.character(result$scale)), ]
  return (result)
}

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


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.assess])[j]

    }
    for(d in 1:k) {
      cluster.sizes[d, i] <- unlist(
        cluster.stats(d = dist, clustering = cutree(tree, k = i))[clust.size])[d]
      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(9, "YlOrRd")) ###only for ordinal values
colourCount <- 9
gt <- getPalette(colourCount)

############question
agg_qq = fun.histogram(qq)

ggplot(agg_qq, aes(x = as.character(scale), y = total))+
  geom_bar(stat = "identity")+labs(title="Scale distribution (question)",x="scale", y = "frequency") +
  scale_x_discrete(limits=c("Q-3","Q-4","Q-5","Q-6","Q-7","Q-8","Q-9","Q-10")) +
  theme_bw() + theme(plot.title = element_text(color = "black", size = "12", face = "bold"),
                     text = element_text(color = "black", size=17))
```

```
## Warning: Removed 3768 rows containing missing values (position_stack).
```



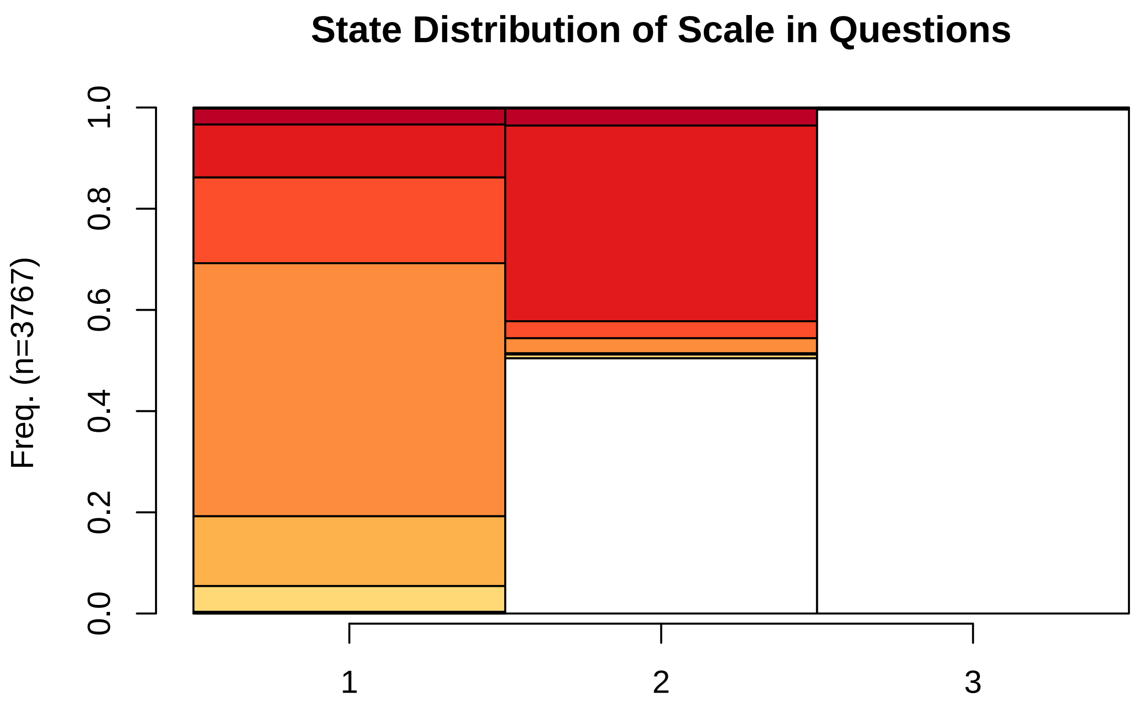
```
qq = all_questions[, 2:4]

qq = fun.naming(qq)

all_sequence <- seqdef(qq)
```

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

##  [>] 9 distinct states appear in the data:

##      1 =

##      2 = Q-10

##      3 = Q-3

##      4 = Q-4

##      5 = Q-5

##      6 = Q-6

##      7 = Q-7
##      8 = Q-8

##      9 = Q-9

##  [>] state coding:

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

##      1

##      2  Q-10        Q-10     Q-10

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

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

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

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

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

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

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

##  [>] 3767 sequences in the data set

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

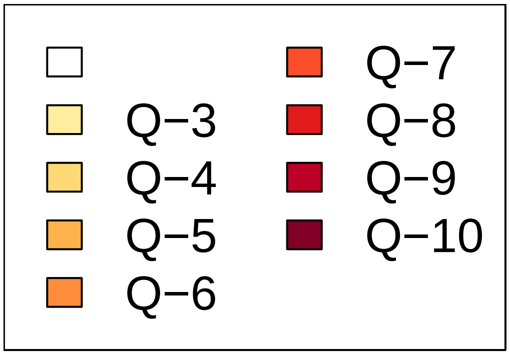
```
cpal(all_sequence)
```

```
## [1] "#8DD3C7" "#FFFFB3" "#BEBADA" "#FB8072" "#80B1D3" "#FDB462" "#B3DE69"
## [8] "#FCCDE5" "#D9D9D9"
```

```
attr(all_sequence, "labels") <- as.character(
  c("","Q-3","Q-4","Q-5", "Q-6", "Q-7", "Q-8", "Q-9", "Q-10"))
attr(all_sequence, "alphabet") <- as.character(
  c("","Q-3","Q-4","Q-5", "Q-6", "Q-7", "Q-8", "Q-9", "Q-10"))
gt[1] <- "#FFFFFF"
attr(all_sequence, "cpal") <- gt


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

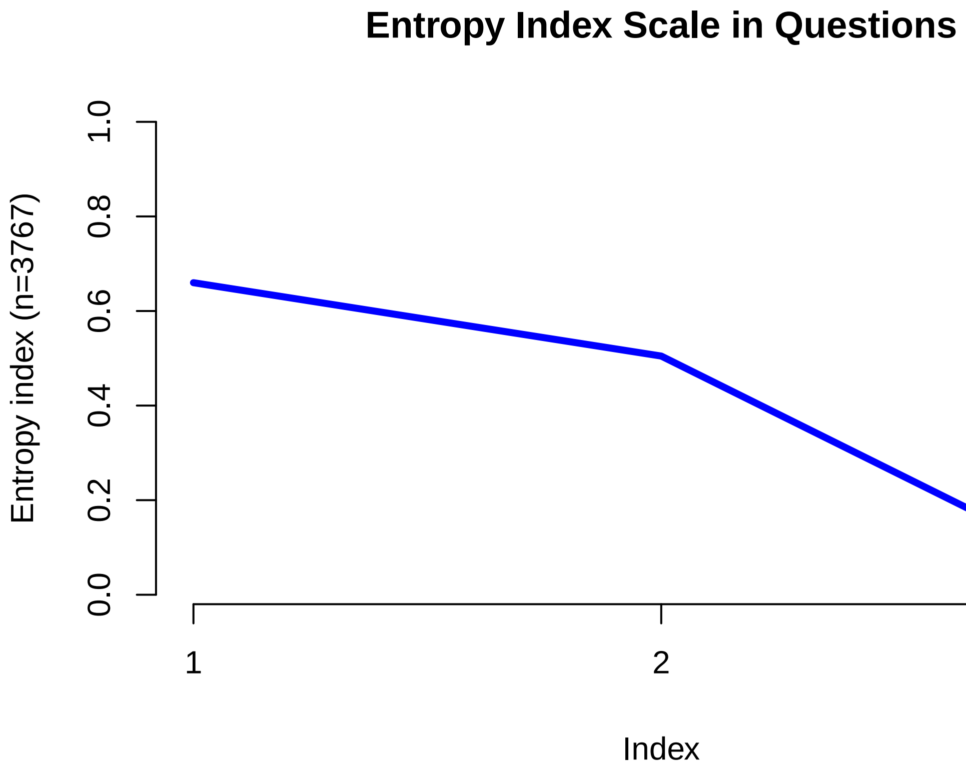


```
seqlegend(all_sequence, cex=1.5, ncol=2)
```



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

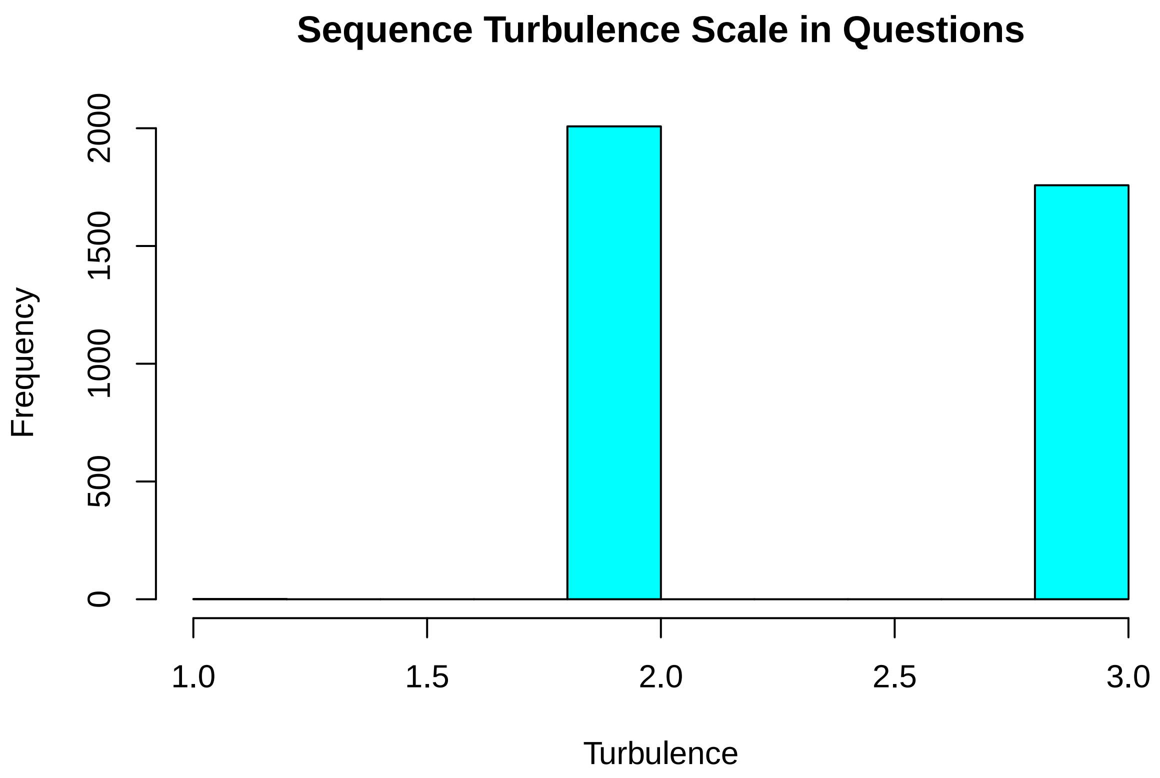
```
##  [!] In rmarkdown::render() : title is deprecated, use main instead.
```



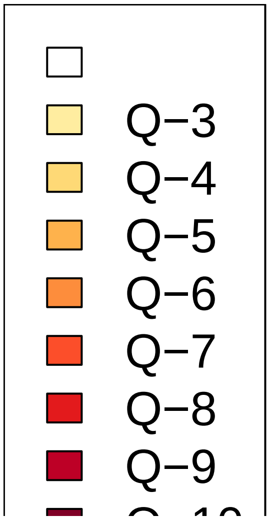
```
Turbulence <- seqST(all_sequence)
summary(Turbulence)
```

```
##    Turbulence
##  Min.   :1.000
##  1st Qu.:2.000
##  Median :2.000
##  Mean   :2.466
##  3rd Qu.:3.000
##  Max.   :3.000
```

```
hist(Turbulence, col = "cyan", main = "Sequence Turbulence Scale in Questions")
```

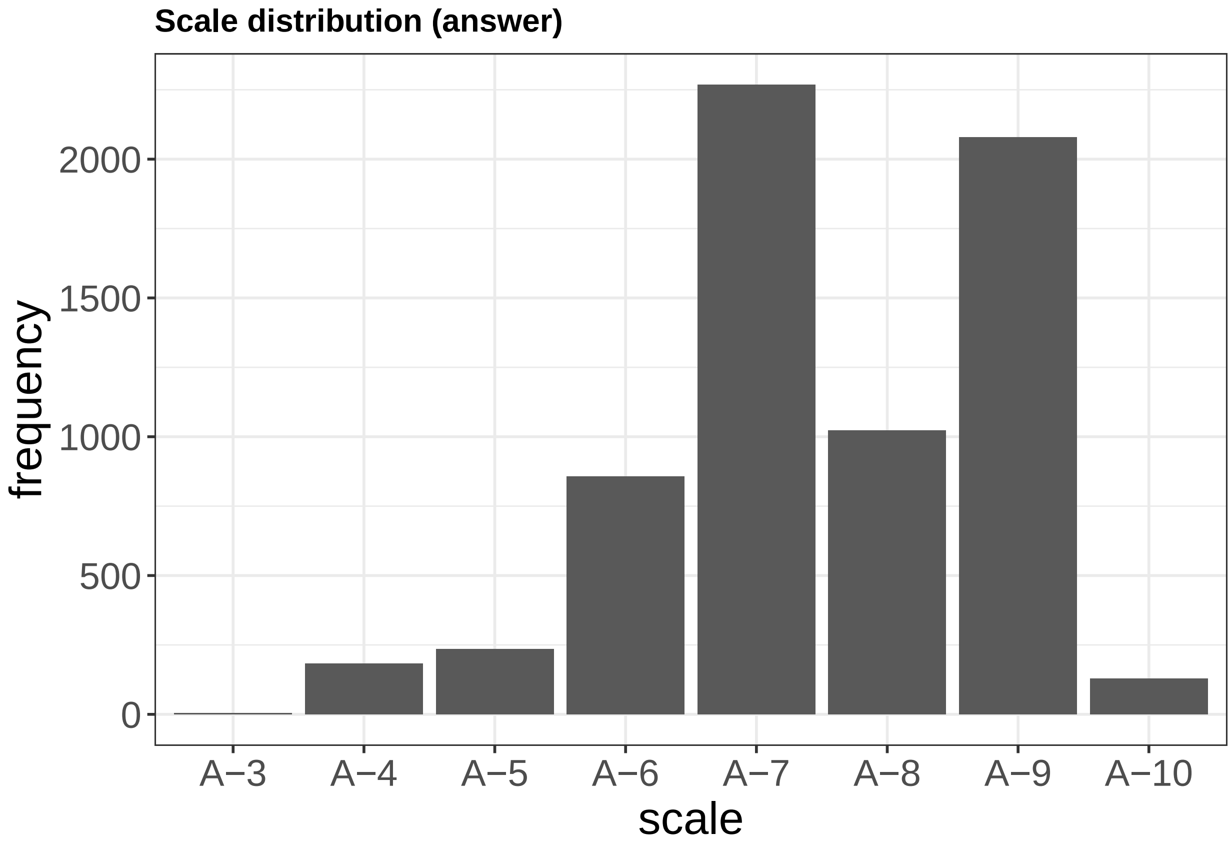


```r
seqlegend(all_sequence, cex=1.5, ncol = 1 )
```



```
############answers
agg_aa = fun.histogram(aa)

ggplot(agg_aa, aes(x = as.character(scale), y = total))+
  geom_bar(stat = "identity")+labs(title="Scale distribution (answer)",x="scale", y = "frequency") +
  scale_x_discrete(limits=c("A-3","A-4","A-5","A-6","A-7","A-8","A-9","A-10")) +
  theme_bw() + theme(plot.title = element_text(color = "black", size = "12", face = "bold"),
                       text = element_text(color = "black", size=17))
```

```
## Warning: Removed 3767 rows containing missing values (position_stack).
```



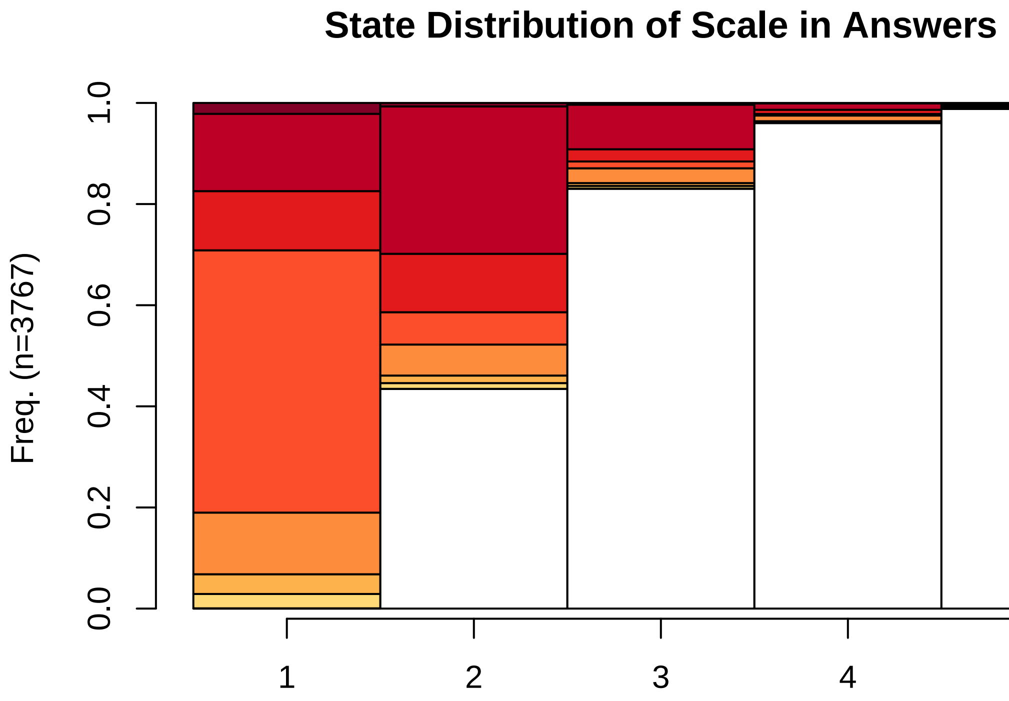
```
aa =all_answers[,2:6]
aa = fun.naming(aa)

all_sequence <- seqdef(aa)
```

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

##  [>] 9 distinct states appear in the data:

##      1 =

##      2 = A-10

##      3 = A-3

##      4 = A-4

##      5 = A-5

##      6 = A-6

##      7 = A-7
##      8 = A-8

##      9 = A-9

##  [>] state coding:

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

##      1

##      2  A-10        A-10     A-10

##      3  A-3         A-3      A-3

##      4  A-4         A-4      A-4

##      5  A-5         A-5      A-5

##      6  A-6         A-6      A-6

##      7  A-7         A-7      A-7

##      8  A-8         A-8      A-8

##      9  A-9         A-9      A-9

##  [>] 3767 sequences in the data set

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

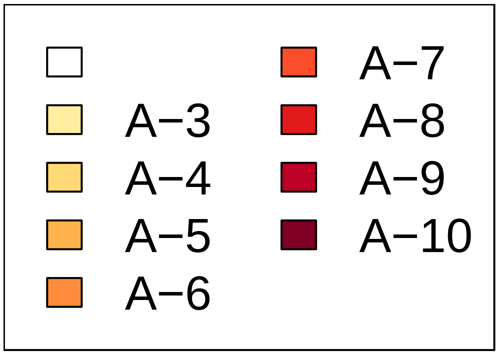
```
cpal(all_sequence)
```

```
## [1] "#8DD3C7" "#FFFFB3" "#BEBADA" "#FB8072" "#80B1D3" "#FDB462" "#B3DE69"
## [8] "#FCCDE5" "#D9D9D9"
```

```
gt[1] <- "#FFFFFF"

attr(all_sequence, "labels") <- as.character(
  c("","A-3","A-4","A-5", "A-6", "A-7", "A-8", "A-9", "A-10"))
attr(all_sequence, "alphabet") <- as.character(
  c("","A-3","A-4","A-5", "A-6", "A-7", "A-8", "A-9", "A-10"))

attr(all_sequence, "cpal") <- gt

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

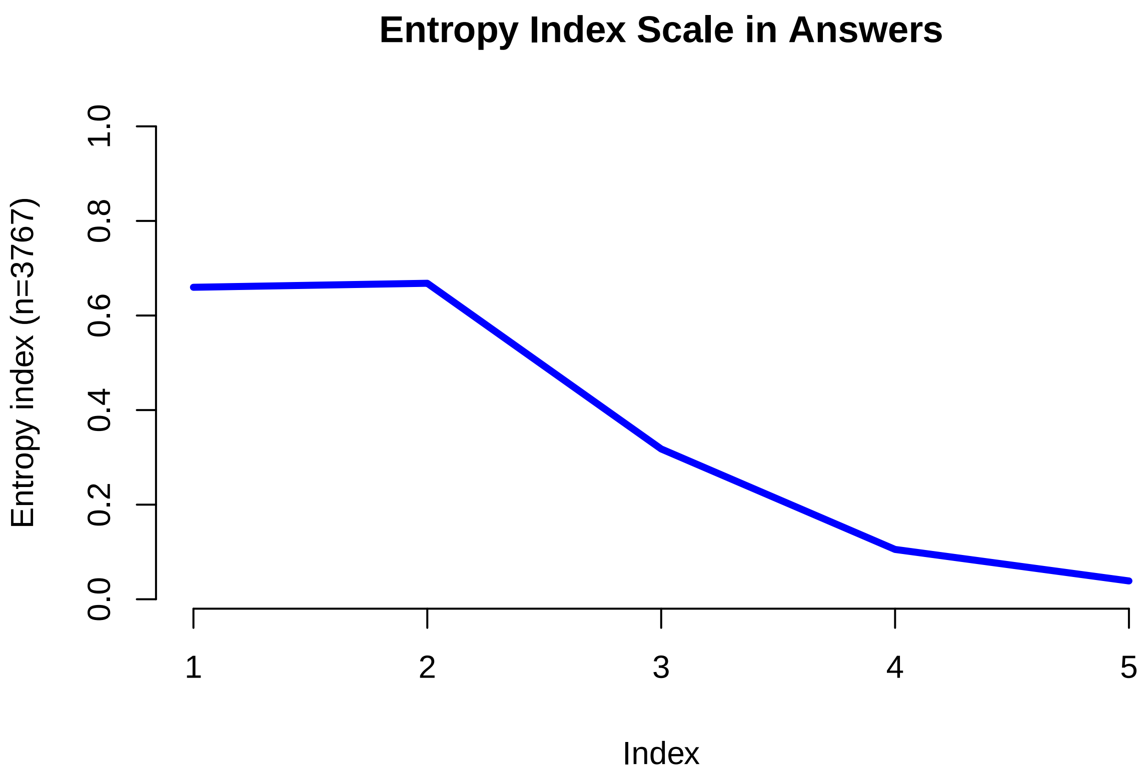


```
seqlegend(all_sequence, cex=1.5, ncol=2)
```



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

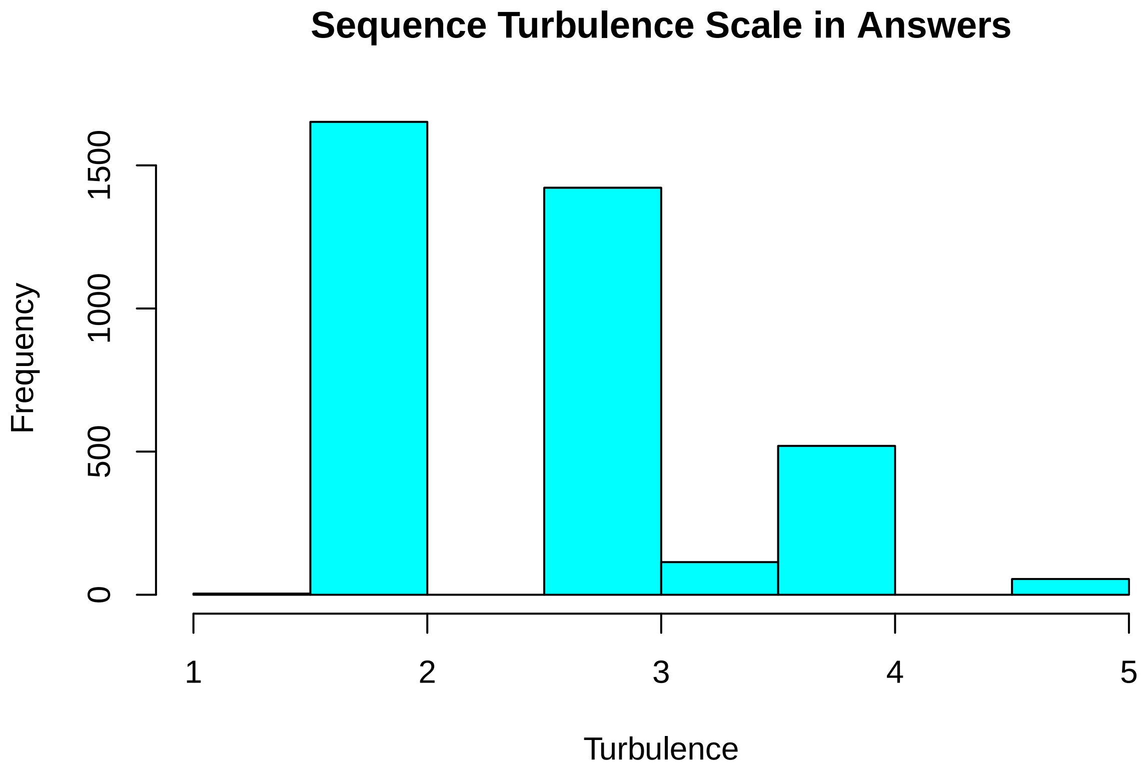
```
##  [!] In rmarkdown::render() : title is deprecated, use main instead.
```



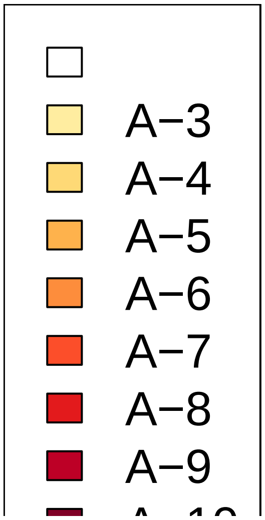
```
Turbulence <- seqST(all_sequence)
summary(Turbulence)
```

```
##    Turbulence
##  Min.   :1.000
##  1st Qu.:2.000
##  Median :3.000
##  Mean   :2.722
##  3rd Qu.:3.000
##  Max.   :5.000
```

```
hist(Turbulence, col = "cyan", main = "Sequence Turbulence Scale in Answers")
```

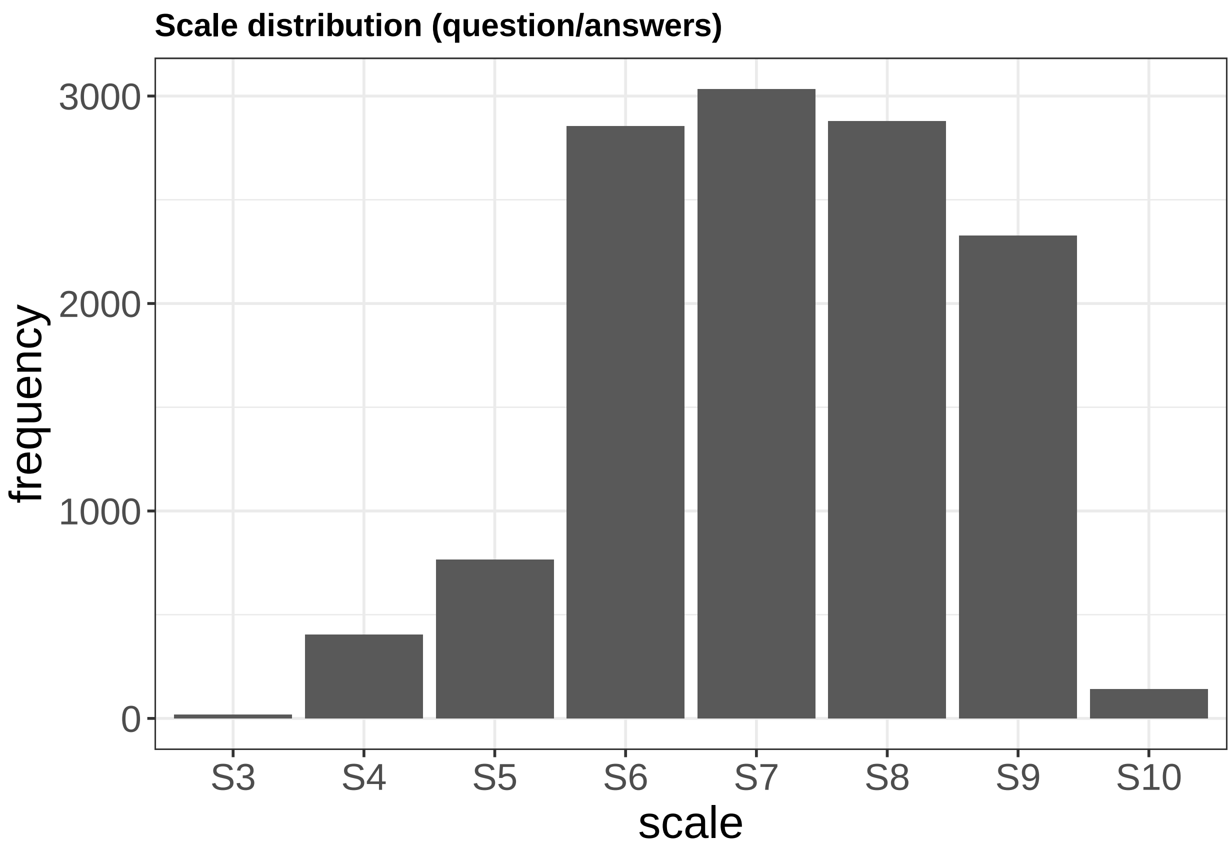


```r
seqlegend(all_sequence, cex=1.5, ncol = 1 )
```



```
############concatenated q-a
agg_aq = fun.histogram(aq)
ggplot(agg_aq, aes(x = as.character(scale), y = total))+
  geom_bar(stat = "identity")+labs(title="Scale distribution (question/answers)",
                                   x="scale", y = "frequency") +
  scale_x_discrete(limits=c("S3","S4","S5","S6","S7","S8","S9","S10")) +
  theme_bw() + theme(plot.title = element_text(color = "black", size = "12", face = "bold"),
                     text = element_text(color = "black", size=17))
```

```
## Warning: Removed 3768 rows containing missing values (position_stack).
```



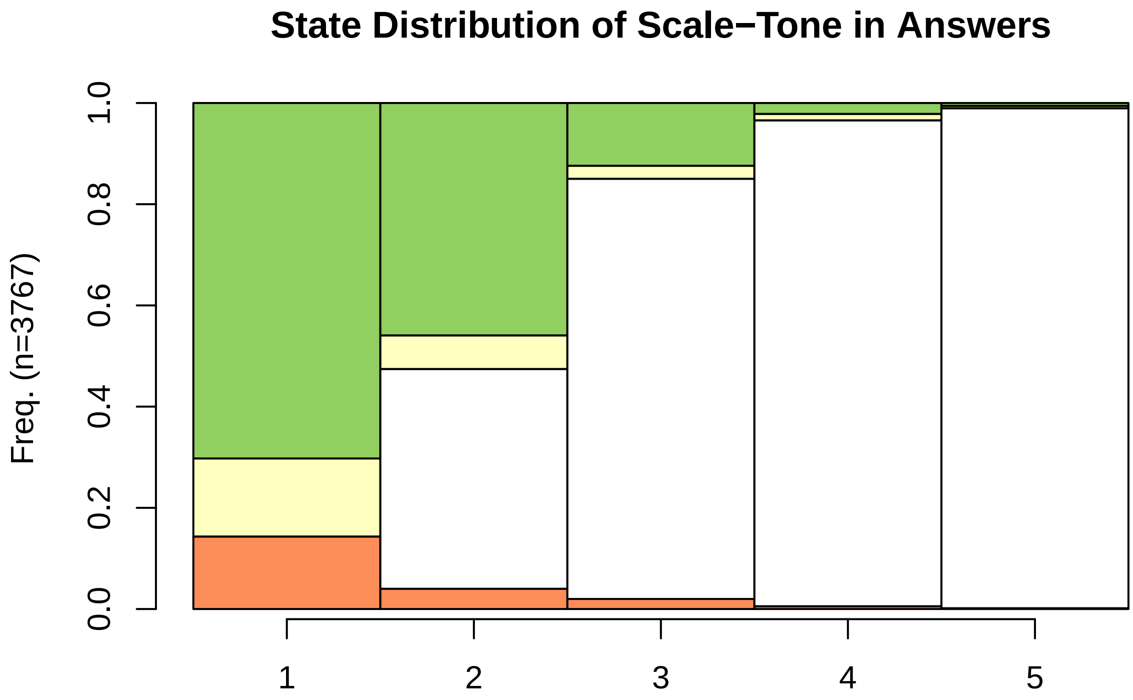
```r
############tone########
questions <- read.table("../sequences/scale-nf-Q-int.txt",
                        header = FALSE, sep = " ",
                        col.names = paste0("V",seq_len(5)), fill = TRUE)
answers <- read.table("../sequences/scale-nf-A-int.txt",
                      header = FALSE, sep = " ",
                      col.names = paste0("V",seq_len(13)), fill = TRUE)

answers_tone <- matrix(data = NA, nrow = length(answers[,1]), ncol = length(answers))
for (i in 1:length(answers[,1])) {
  qvec = questions[questions$V1 == answers[i, 1],]
  qvec = qvec[!is.na(qvec)]
  asked_scale = min(qvec)
  answer_tone = sign(answers[i, 2:13] - asked_scale)
  answers_tone[i, 1] = answers[i, 1]
  answers_tone[i, 2:13] = t(answer_tone)
}

answers_tone_wid = as.data.frame(answers_tone[, 2:6])
answers_tone_factor <- mapply(answers_tone_wid, FUN=as.character)
answers_tone_factor <- matrix(data=answers_tone_factor,
                              ncol=length(answers_tone_wid), nrow=length(answers_tone_wid[,1]))
for (i in 1:length(answers_tone_wid[,1])) {
  temp = answers_tone_factor[i,]
  temp[is.na(temp)] <- " "
  answers_tone_factor[i, ] <- t(temp)
}
answers_tone_factor_df <- as.data.frame(answers_tone_factor)
answers_tone_factor_df = fun.naming(answers_tone_factor_df)

all_sequence <- seqdef(as.data.frame(answers_tone_factor_df))
```

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

##  [>] 4 distinct states appear in the data:

##      1 = -1

##      2 =

##      3 = 0

##      4 = 1

##  [>] state coding:

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

##      1  -1          -1       -1

##      2

##      3  0           0        0

##      4  1           1        1

##  [>] 3767 sequences in the data set

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

```r
cpal(all_sequence)
```

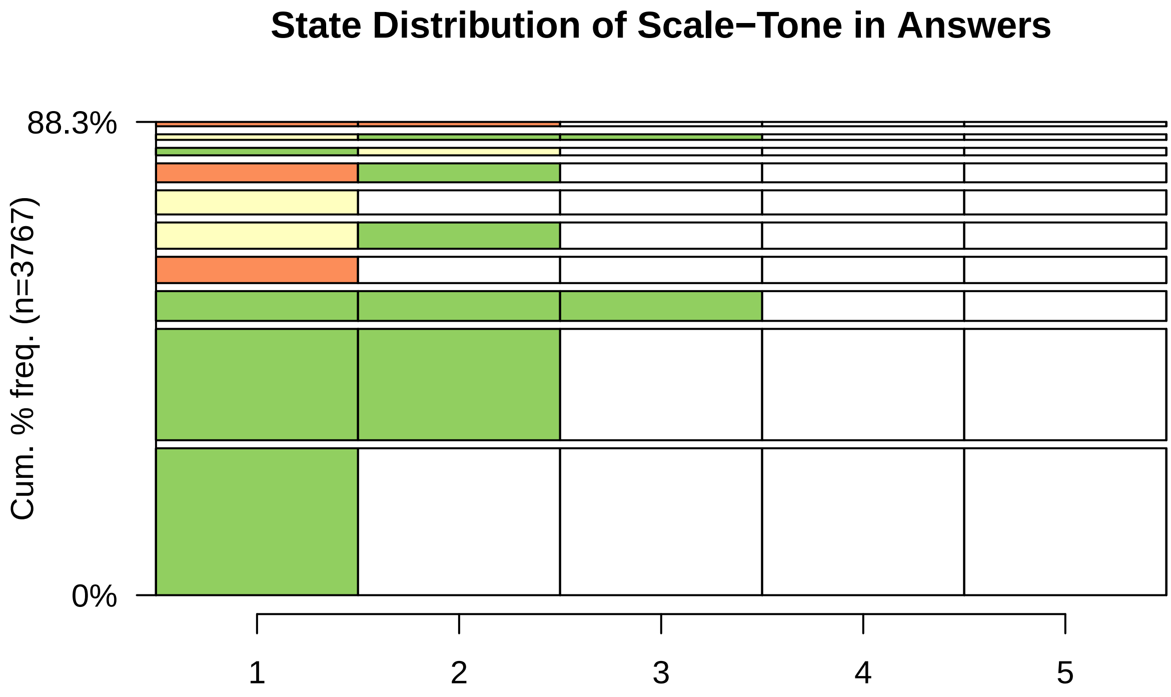
```
## [1] "#7FC97F" "#BEAED4" "#FDC086" "#FFFF99"
```

```r
getPalette = colorRampPalette(brewer.pal(3, "RdYlGn")) ###only for ordinal values
colourCount <- 3
gt <- getPalette(colourCount)
gt <- c(gt[1], "#FFFFFF", gt[2], gt[3])
attr(all_sequence, "cpal") <- gt

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

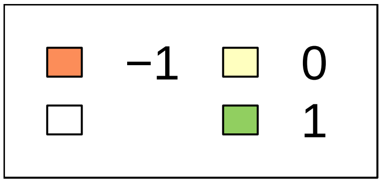


```r
seqfplot(all_sequence, with.legend = F, border = T,
         main = "State Distribution of Scale-Tone in Answers")
```

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



```
seqlegend(all_sequence, cex=1.5, ncol=2)
```

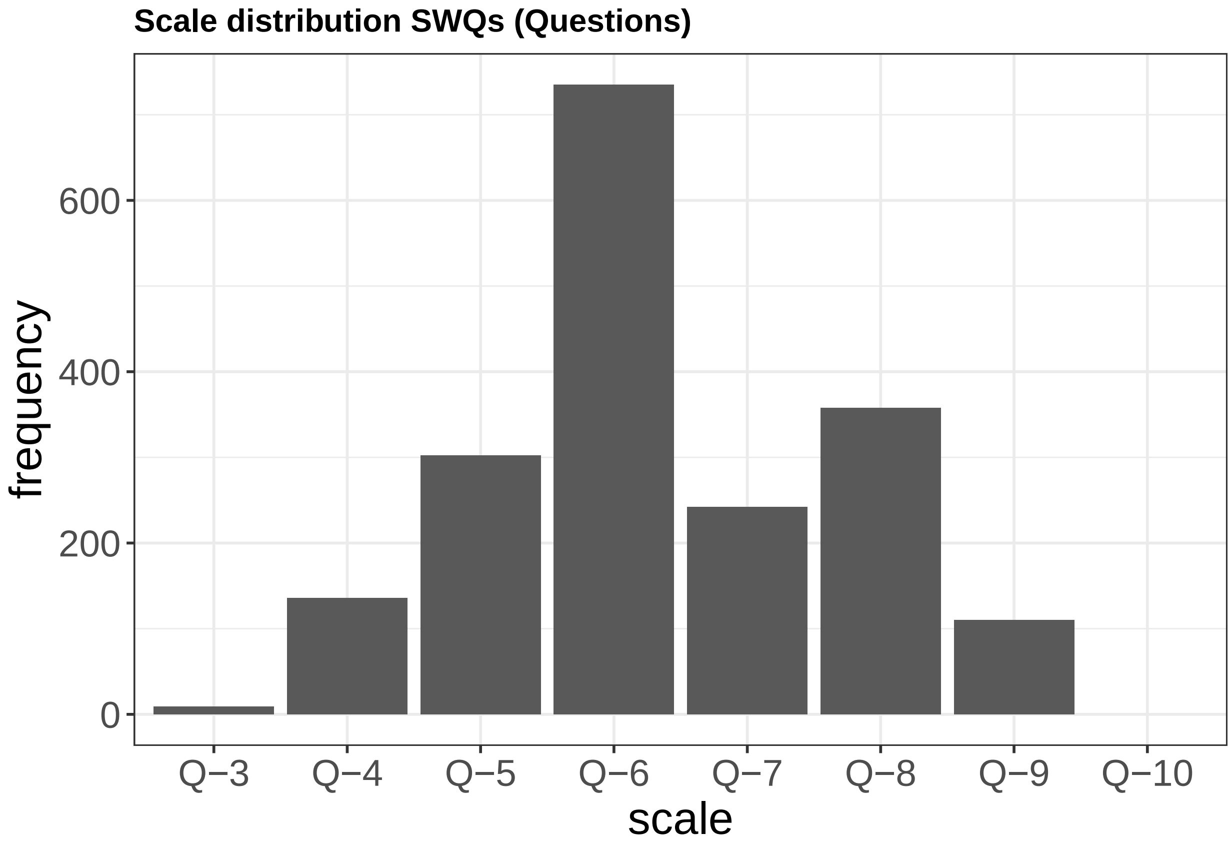


```
########################################SWQ######################################
all_qas <- read.table("../sequences/scale-nf-all-SWQ.txt", header = FALSE,
                      sep = " ", col.names = paste0("V",seq_len(20)), fill = TRUE)

agg_aq = fun.histogram(all_qas)
```

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

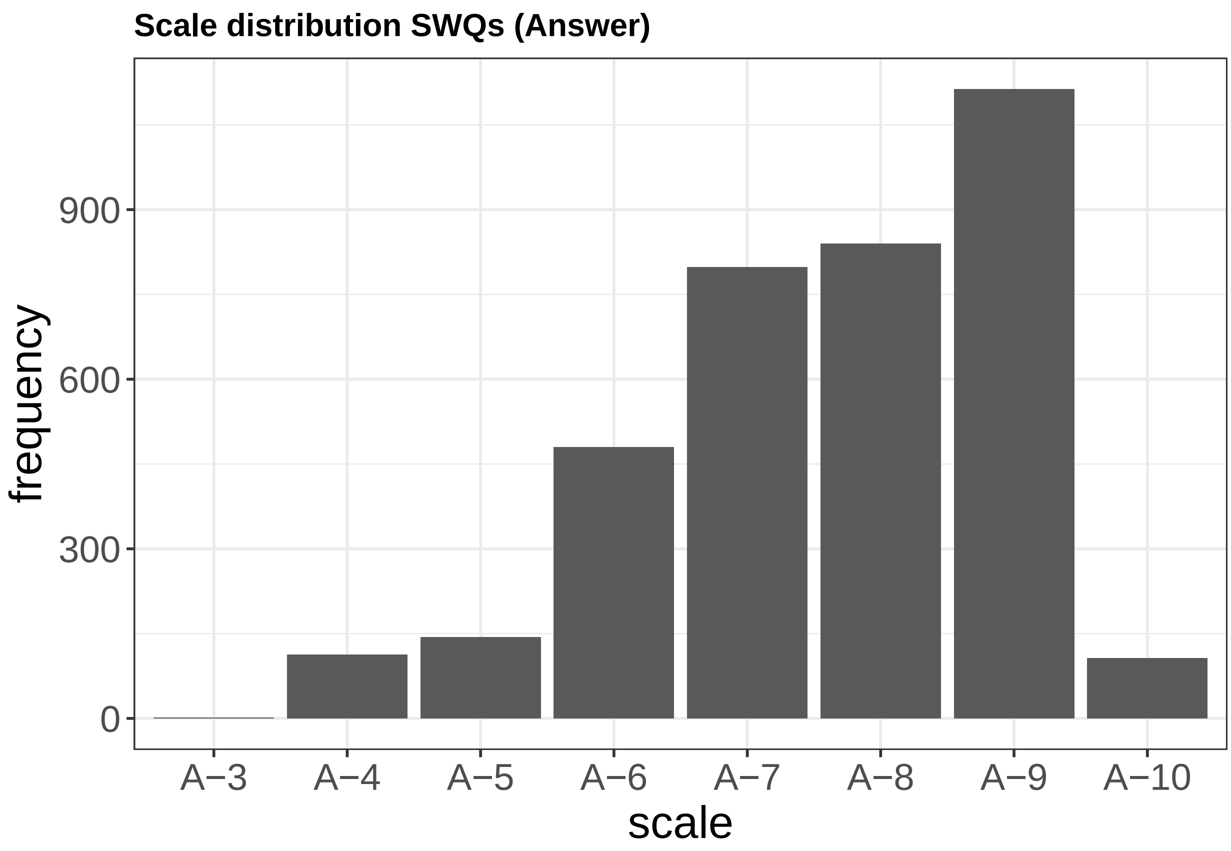
```
ggplot(agg_aq, aes(x = as.character(scale), y = total))+
  geom_bar(stat = "identity")+labs(title="Scale distribution SWQs (Questions)",
                                   x="scale", y = "frequency") +
  scale_x_discrete(limits=c("Q-3","Q-4","Q-5","Q-6","Q-7","Q-8","Q-9","Q-10")) +
  theme_bw() + theme(plot.title = element_text(color = "black", size = "12", face = "bold"),
                     text = element_text(color = "black", size=17))
```

```
## Warning: Removed 10 rows containing missing values (position_stack).
```



```
ggplot(agg_aq, aes(x = as.character(scale), y = total))+geom_bar(stat = "identity")+
  labs(title="Scale distribution SWQs (Answer)",x="scale", y = "frequency") +
  scale_x_discrete(limits=c("A-3","A-4","A-5","A-6","A-7","A-8","A-9","A-10")) +
  theme_bw() + theme(plot.title = element_text(color = "black", size = "12", face = "bold"),
                     text = element_text(color = "black", size=17))
```

```
## Warning: Removed 9 rows containing missing values (position_stack).
```



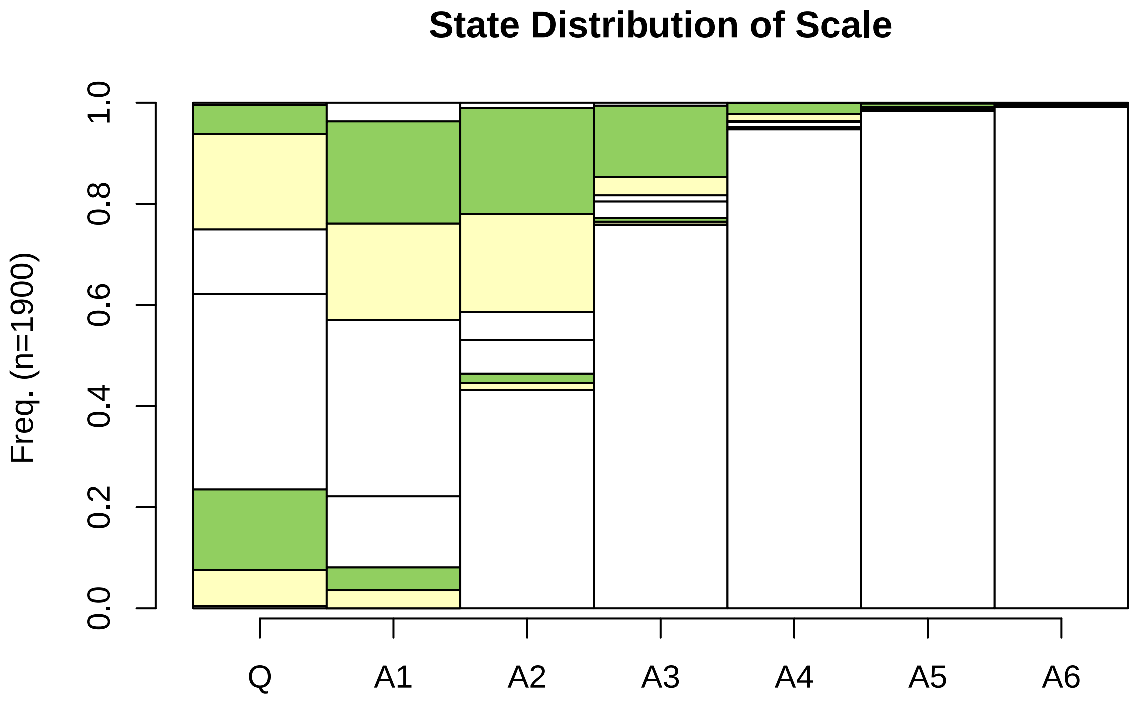
```r
fun.to.scale('../sequences/scale-nf-all-SWQ.txt', '../sequences/scale-nf-all-SWQ-s.txt')
all_swq_qas <- read.table("../sequences/scale-nf-all-SWQ-s.txt",
                          header = FALSE, sep = " ", col.names = paste0("V",seq_len(20)), fill = TRUE)

aa =all_swq_qas[,1:7]
aa = fun.naming(aa)
colnames(aa) <- c("Q", "A1", "A2", "A3", "A4", "A5", "A6")

all_sequence <- seqdef(aa)
```

```
##  [>] 9 distinct states appear in the data:

##      1 =

##      2 = S10

##      3 = S3

##      4 = S4

##      5 = S5

##      6 = S6
##      7 = S7

##      8 = S8

##      9 = S9

##  [>] state coding:

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

##      1

##      2  S10         S10      S10

##      3  S3          S3       S3

##      4  S4          S4       S4

##      5  S5          S5       S5

##      6  S6          S6       S6

##      7  S7          S7       S7

##      8  S8          S8       S8

##      9  S9          S9       S9

##  [>] 1900 sequences in the data set

##  [>] min/max sequence length: 7/7
```

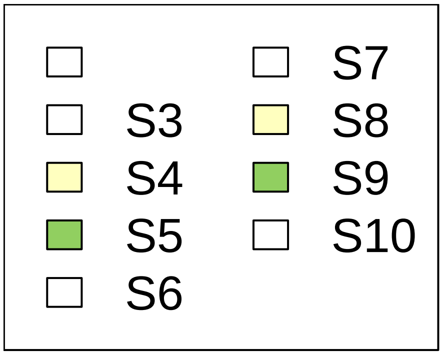
```
cpal(all_sequence)
```

```
## [1] "#8DD3C7" "#FFFFB3" "#BEBADA" "#FB8072" "#80B1D3" "#FDB462" "#B3DE69"
## [8] "#FCCDE5" "#D9D9D9"
```

```
gt[1] <- "#FFFFFF"

attr(all_sequence, "labels") <- as.character(c("","S3","S4","S5", "S6", "S7", "S8", "S9", "S10"))
attr(all_sequence, "alphabet") <- as.character(c("","S3","S4","S5", "S6", "S7", "S8", "S9", "S10"))

attr(all_sequence, "cpal") <- gt

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

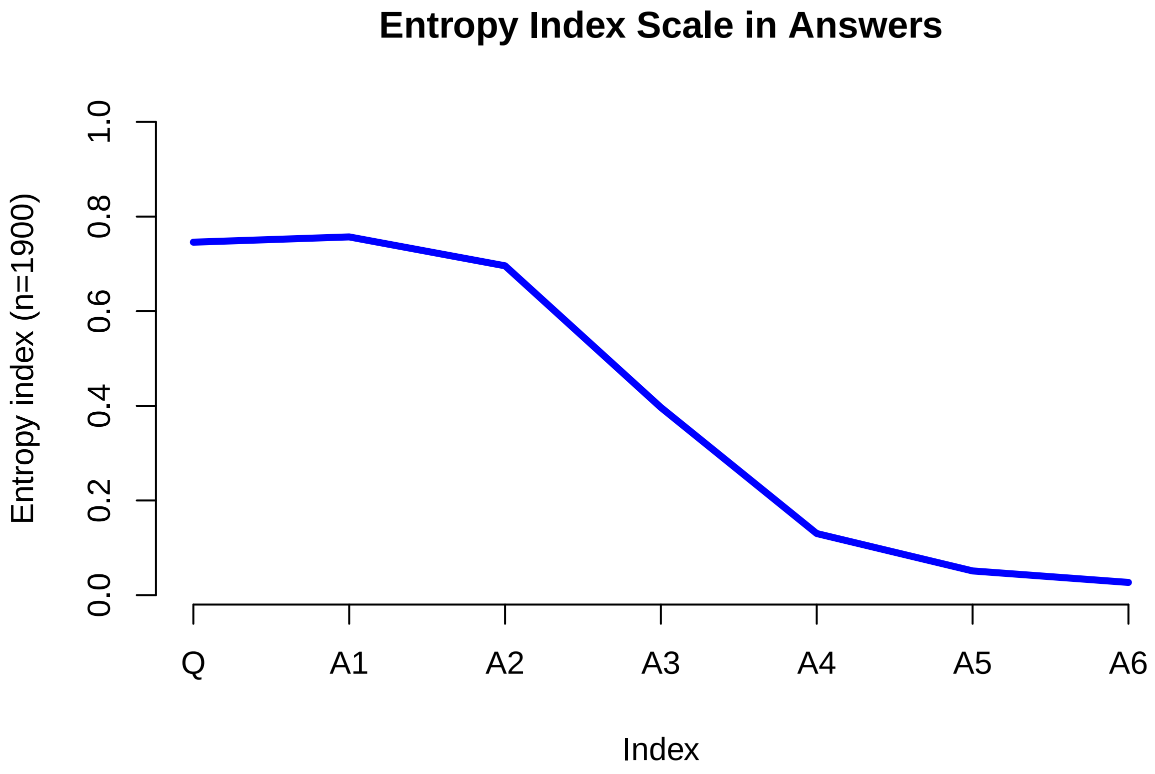


```
seqlegend(all_sequence, cex=1.5, ncol=2)
```



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

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

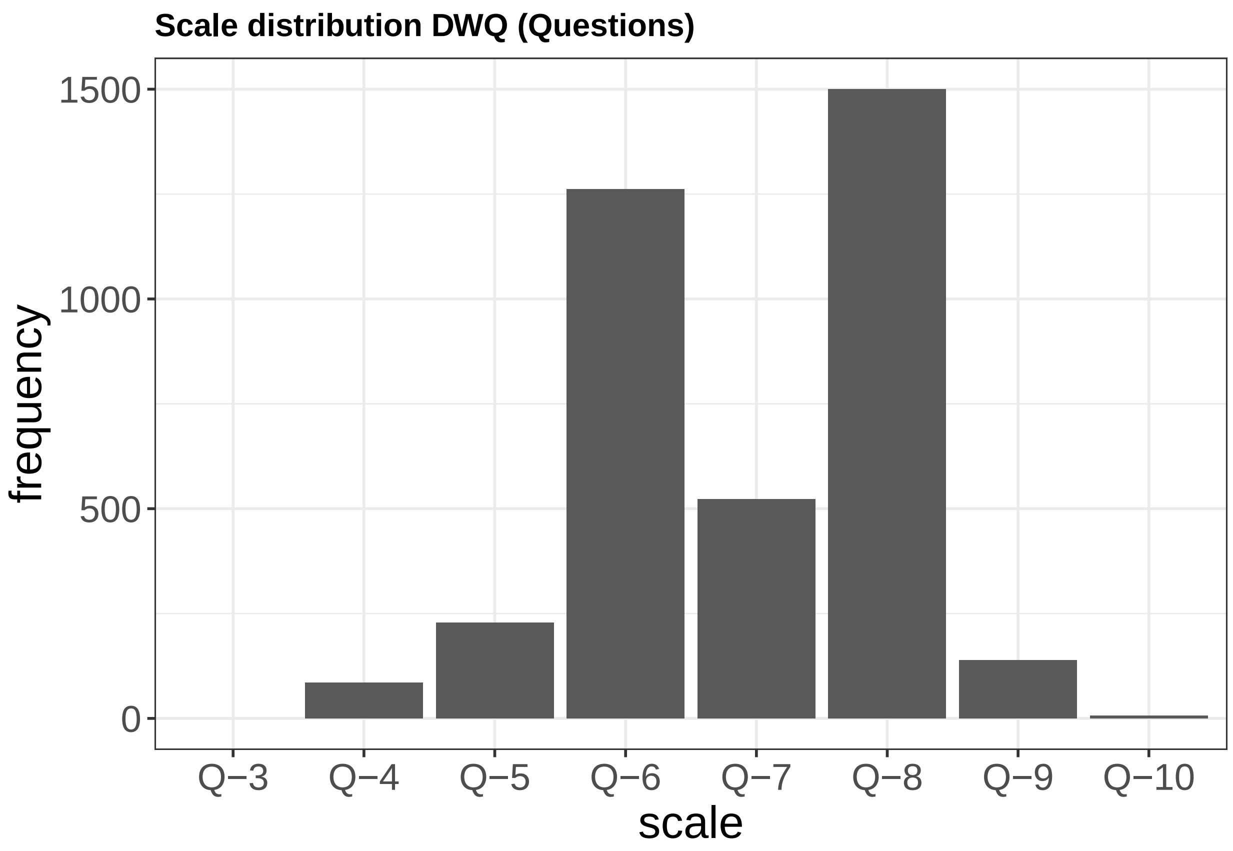


```
########################################RWQ####################################
all_qas <- read.table("../sequences/scale-nf-all-DWQ.txt",
                      header = FALSE, sep = " ",
                      col.names = paste0("V",seq_len(20)), fill = TRUE)

agg_aq = fun.histogram(all_qas)
```

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

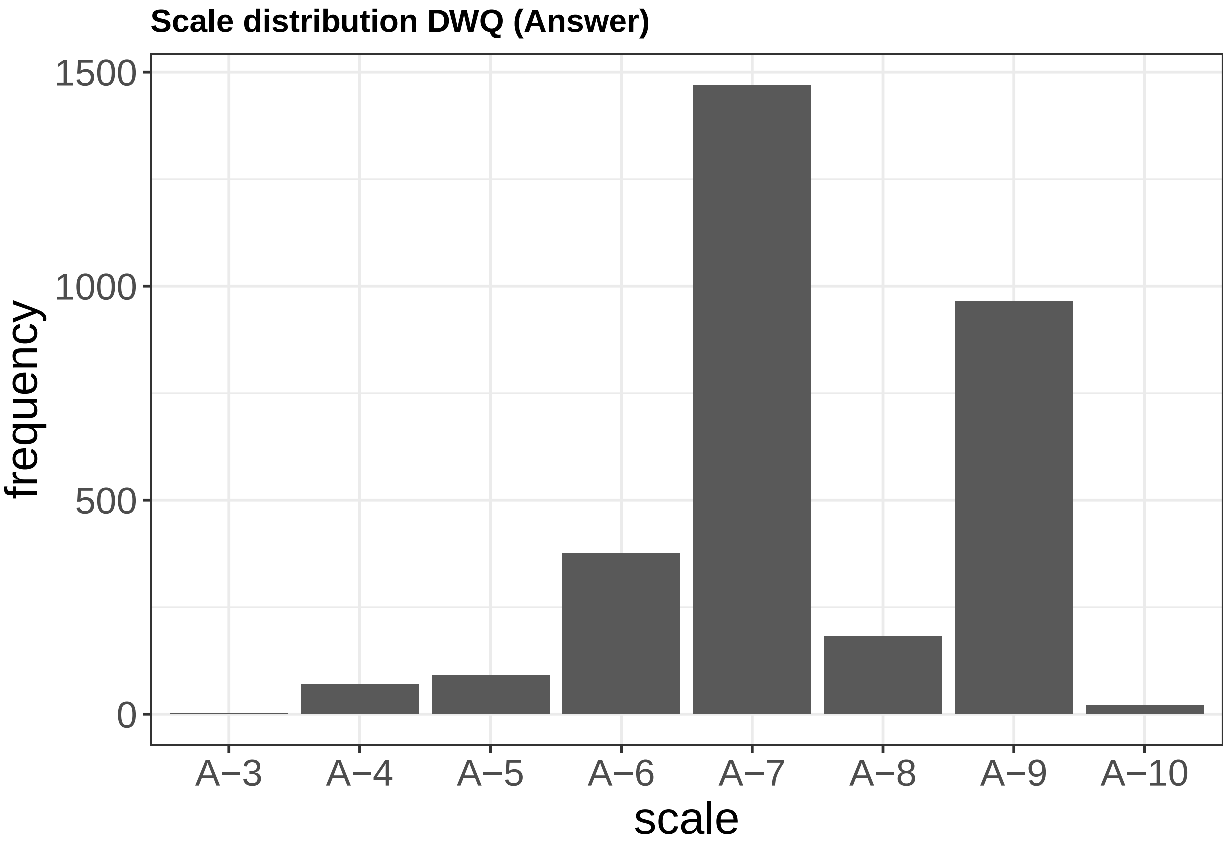
```
ggplot(agg_aq, aes(x = as.character(scale), y = total))+geom_bar(stat = "identity")+
  labs(title="Scale distribution DWQ (Questions)",x="scale", y = "frequency") +
  scale_x_discrete(limits=c("Q-3","Q-4","Q-5","Q-6","Q-7","Q-8","Q-9","Q-10")) +
  theme_bw() + theme(plot.title = element_text(color = "black", size = "12", face = "bold"),
                     text = element_text(color = "black", size=17))
```

```
## Warning: Removed 10 rows containing missing values (position_stack).
```



```
ggplot(agg_aq, aes(x = as.character(scale), y = total))+geom_bar(stat = "identity")+
  labs(title="Scale distribution DWQ (Answer)",x="scale", y = "frequency") +
  scale_x_discrete(limits=c("A-3","A-4","A-5","A-6","A-7","A-8","A-9","A-10")) +
  theme_bw() + theme(plot.title = element_text(color = "black", size = "12", face = "bold"),
                   text = element_text(color = "black", size=17))
```

```
## Warning: Removed 9 rows containing missing values (position_stack).
```



Scale distribution DWQ (Answer)