prominence-categorical-distribution.R

hamze

2020-01-22

```r
##############################################################
##############################################################
#####################Prominence 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)
```

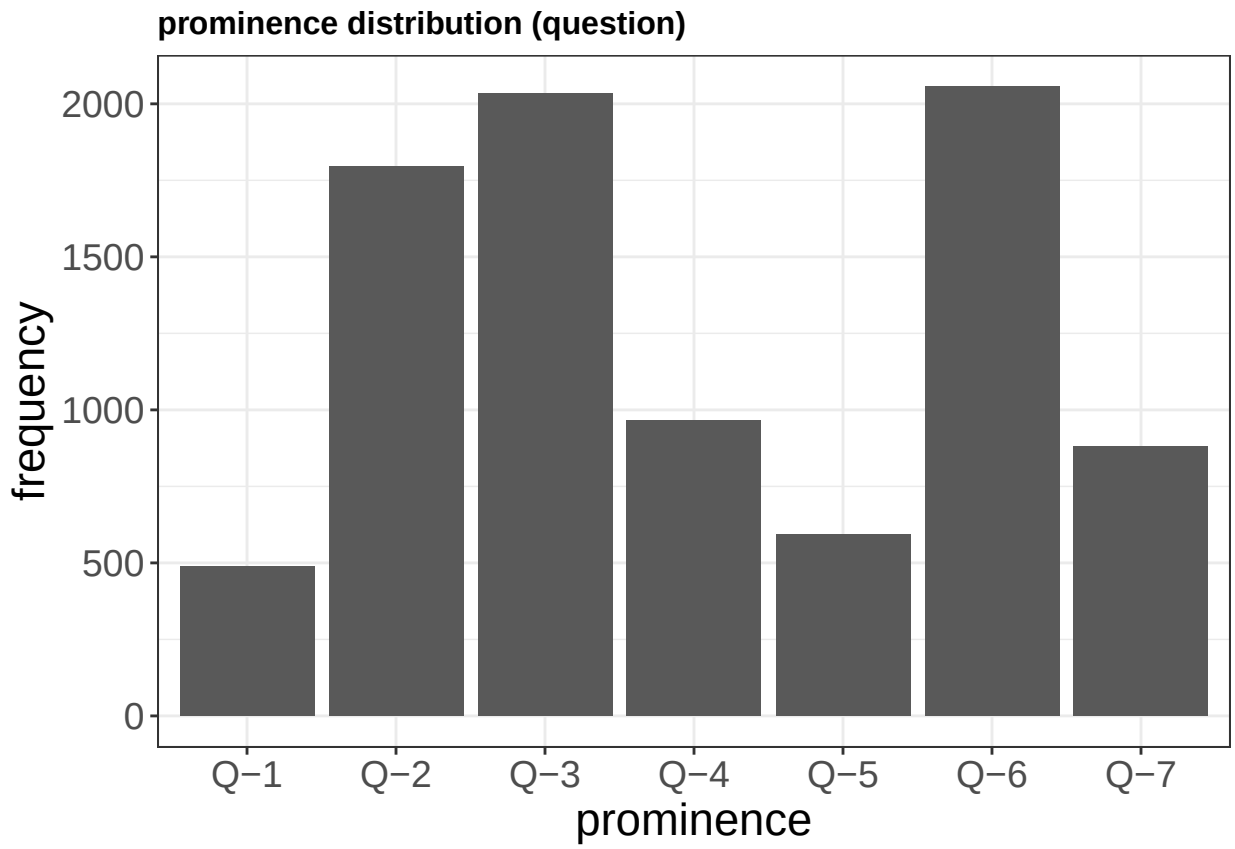
```
##
## Attaching package: 'dplyr'

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

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

prominences_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
prominences = as.data.frame(table(vector_a))

write.csv(prominences_q, file="result/prominences_q.csv")
write.csv(prominences, file="result/prominences_a.csv")

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

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

    }
```

```r
    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(8, "YlOrRd")) ###only for ordinal values
colourCount <- 8
gt <- getPalette(colourCount)

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

ggplot(agg_qq, aes(x = as.character(prominence), y = total))+
  geom_bar(stat = "identity")+labs(title="prominence distribution (question)",
                                   x="prominence", y = "frequency") +
  scale_x_discrete(limits=c("Q-1", "Q-2","Q-3","Q-4","Q-5","Q-6","Q-7")) +
  theme_bw() + theme(plot.title = element_text(
    color = "black", size = "12", face = "bold"), text = element_text(color = "black", size=17))
```

```
## Warning: Removed 1 rows containing missing values (position_stack).
```




```
qq = fun.naming(qq)

all_sequence <- seqdef(qq)
```

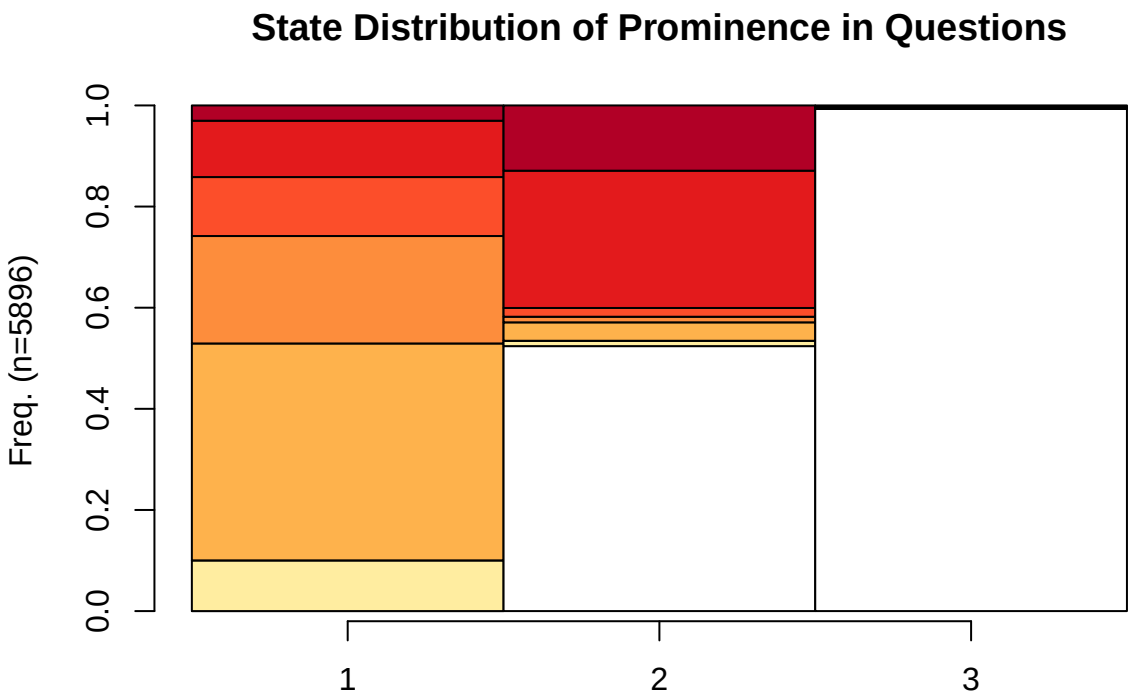
```
##  [!] found '-' character in state codes, not recommended

##  [>] 8 distinct states appear in the data:

##      1 =

##      2 = Q-1

##      3 = Q-2

##      4 = Q-3

##      5 = Q-4

##      6 = Q-5

##      7 = Q-6

##      8 = Q-7
```

```
##  [>] state coding:

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

##      1

##      2  Q-1         Q-1      Q-1

##      3  Q-2         Q-2      Q-2

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

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

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

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

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

##  [>] 5896 sequences in the data set

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

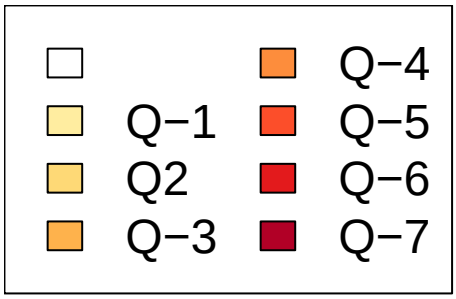
```r
cpal(all_sequence)
```

```
## [1] "#7FC97F" "#BEAED4" "#FDC086" "#FFFF99" "#386CB0" "#F0027F" "#BF5B17"
## [8] "#666666"
```

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

attr(all_sequence, "labels") <- as.character(
  c("", "Q-1", "Q2", "Q-3","Q-4","Q-5", "Q-6", "Q-7"))
attr(all_sequence, "alphabet") <- as.character(
  c("", "Q-1", "Q2", "Q-3","Q-4","Q-5", "Q-6", "Q-7"))

attr(all_sequence, "cpal") <- gt

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

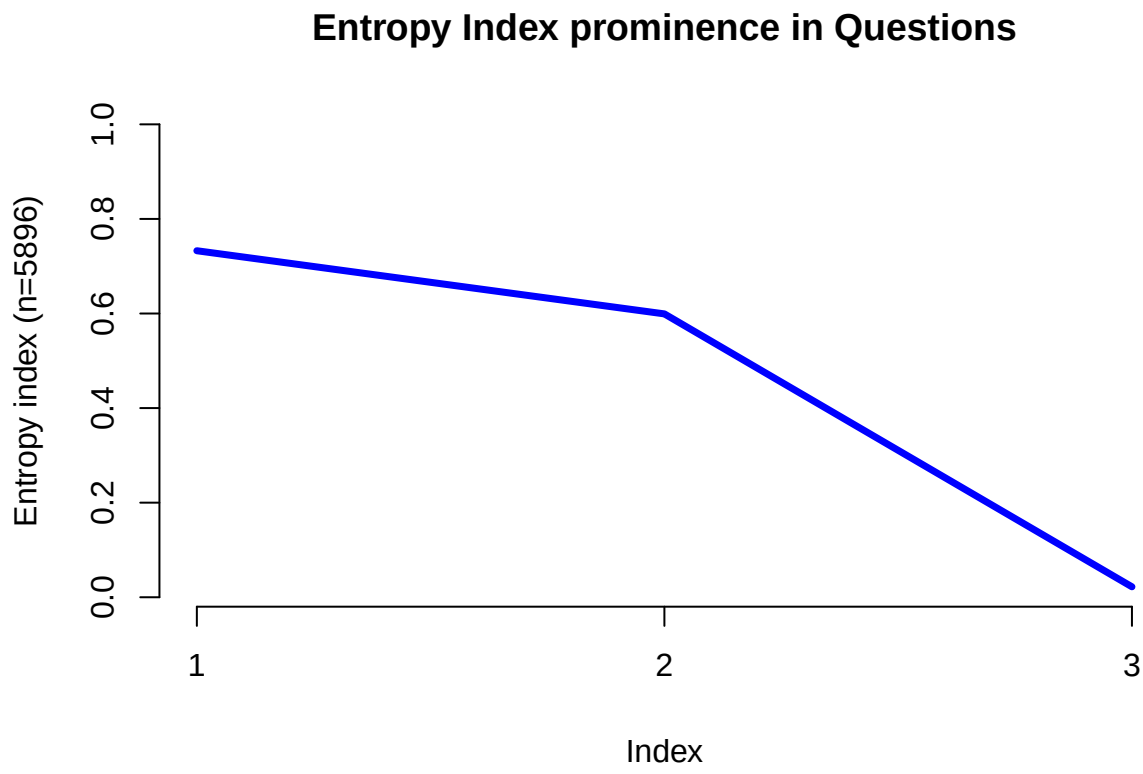



```
seqlegend(all_sequence, cex=1.5, ncol=2)
```




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

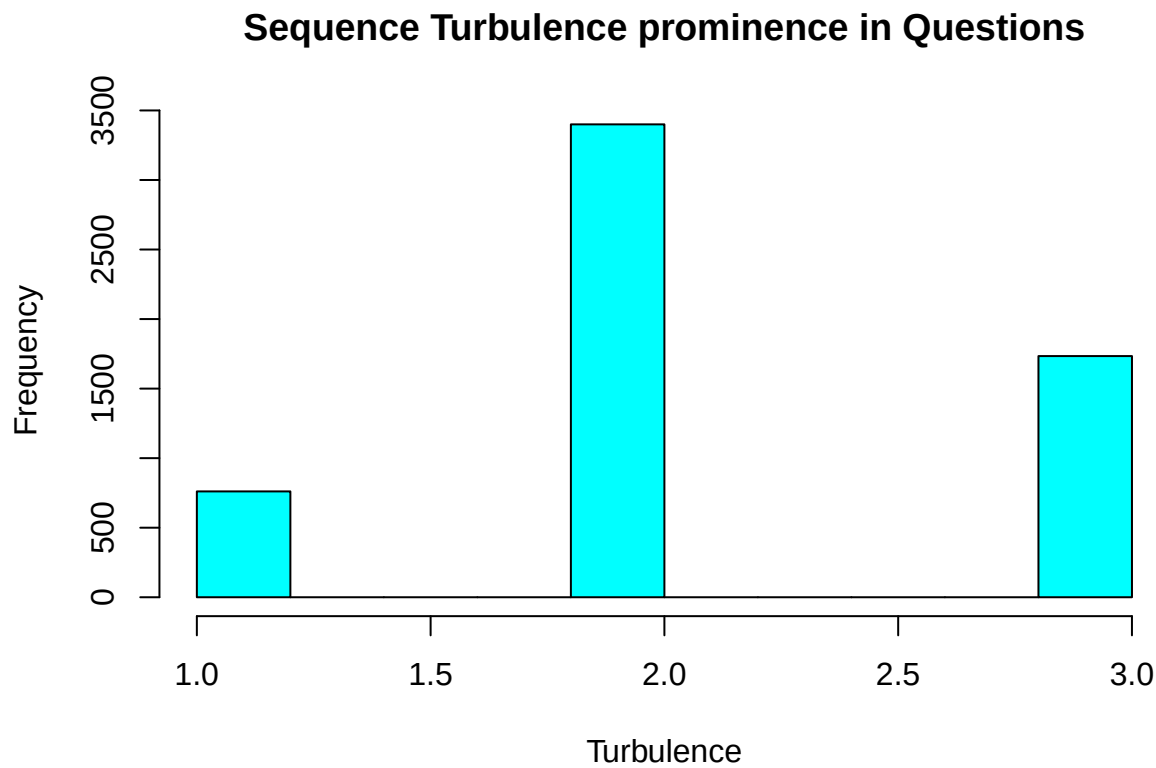
```
##  [!] In rmarkdown::render() : title is deprecated, use main instead.
```




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

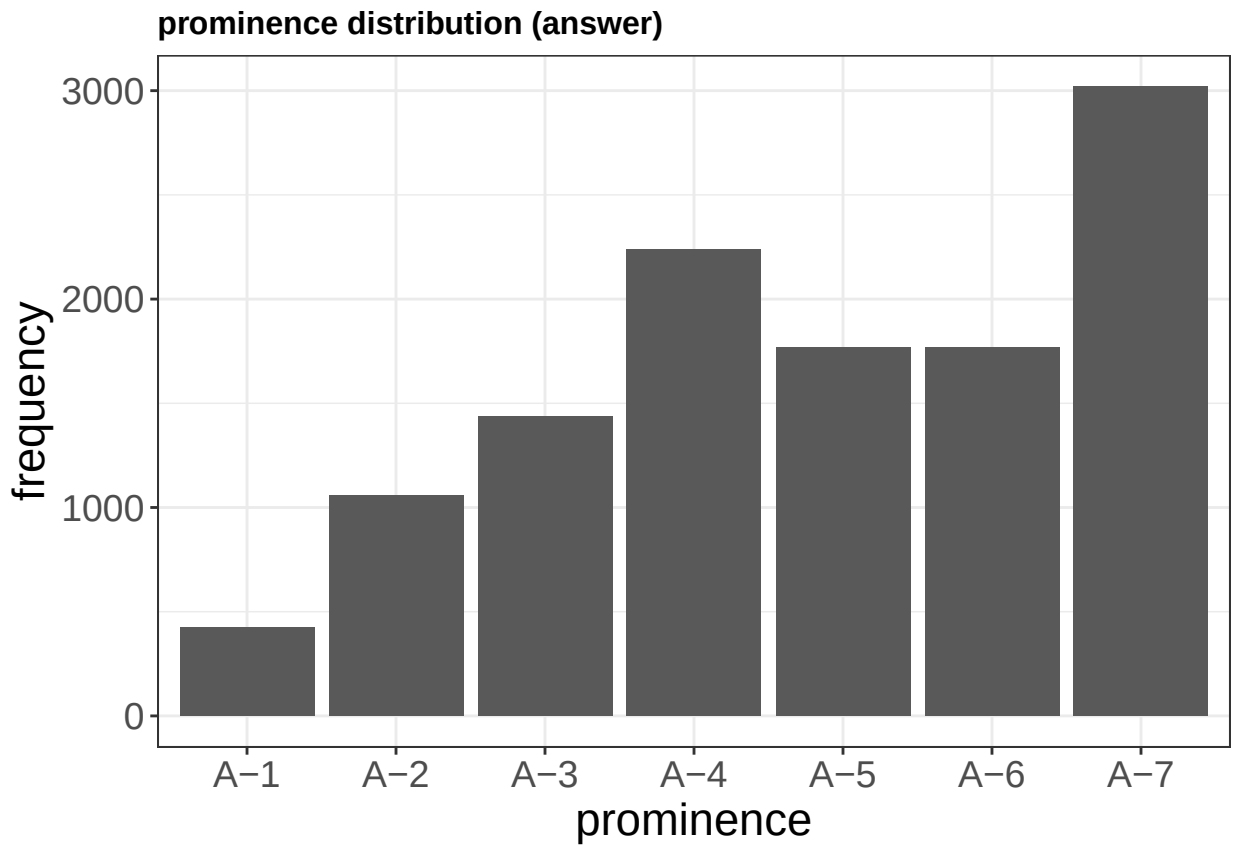
```
##    Turbulence
##  Min.   :1.000
##  1st Qu.:2.000
##  Median :2.000
##  Mean   :2.165
##  3rd Qu.:3.000
##  Max.   :3.000
##  NA's   :1
```

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




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

ggplot(agg_aa, aes(x = as.character(prominence), y = total))+
  geom_bar(stat = "identity")+labs(title="prominence distribution (answer)",x="prominence",
                                   y = "frequency") +
  scale_x_discrete(limits=c("A-1", "A-2","A-3","A-4","A-5","A-6","A-7")) +
  theme_bw() + theme(plot.title = element_text(
    color = "black", size = "12", face = "bold"), text = element_text(color = "black", size=17))
```

```
## Warning: Removed 1 rows containing missing values (position_stack).
```




```
aa = fun.naming(aa)

all_sequence <- seqdef(aa)
```

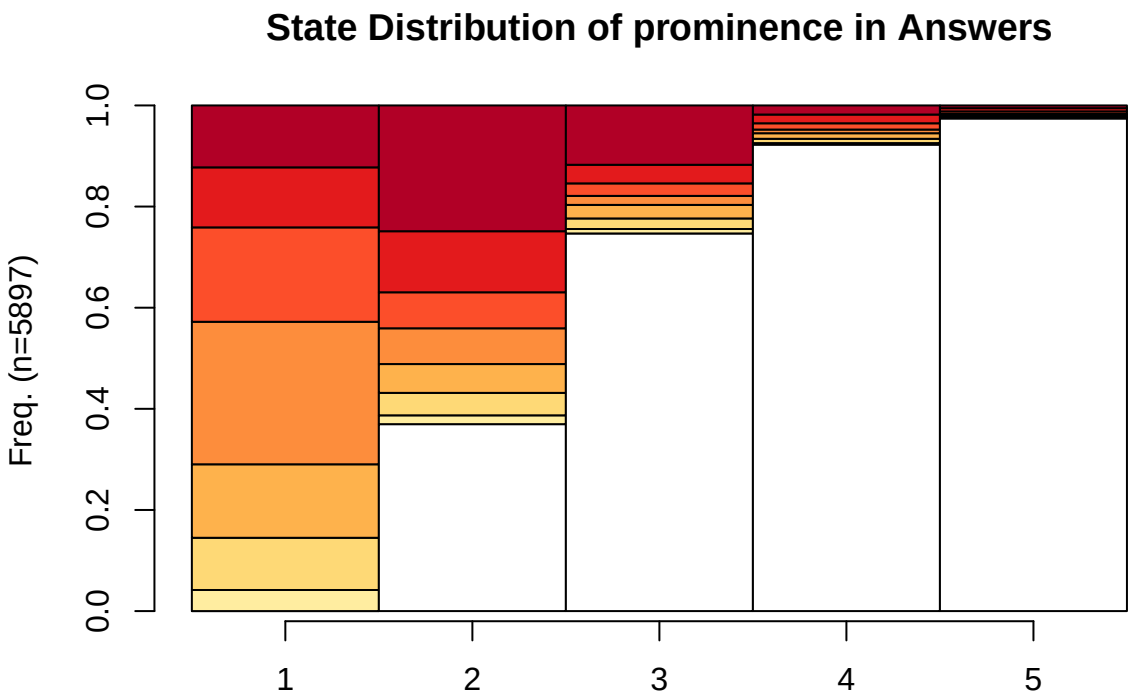
```
##  [!] found '-' character in state codes, not recommended

##  [>] 8 distinct states appear in the data:

##      1 =

##      2 = A-1

##      3 = A-2

##      4 = A-3

##      5 = A-4

##      6 = A-5

##      7 = A-6

##      8 = A-7
```

```
##  [>] state coding:

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

##      1

##      2  A-1         A-1      A-1

##      3  A-2         A-2      A-2

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

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

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

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

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

##  [>] 5897 sequences in the data set

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

```r
cpal(all_sequence)
```

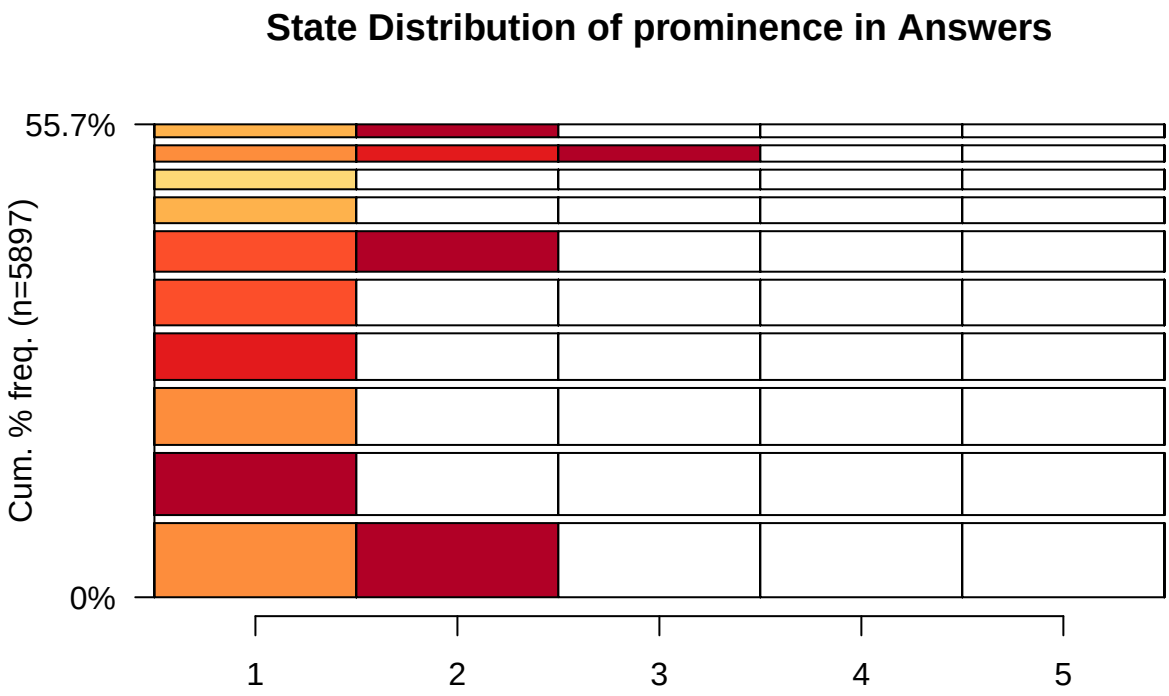
```
## [1] "#7FC97F" "#BEAED4" "#FDC086" "#FFFF99" "#386CB0" "#F0027F" "#BF5B17"
## [8] "#666666"
```

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

attr(all_sequence, "labels") <- as.character(c(
  "", "A-1", "A-2", "A-3","A-4","A-5", "A-6", "A-7"))
attr(all_sequence, "alphabet") <- as.character(c(
  "", "A-1", "A-2", "A-3","A-4","A-5", "A-6", "A-7"))

attr(all_sequence, "cpal") <- gt

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



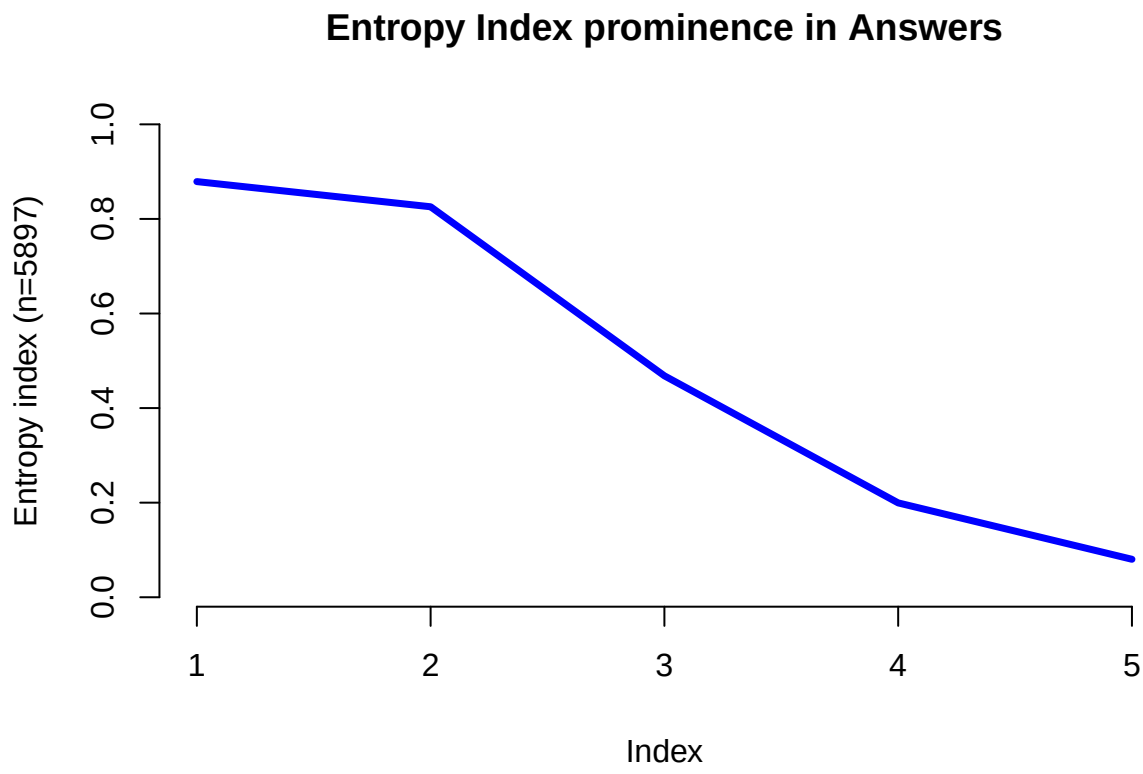

```r
seqfplot(all_sequence, with.legend = F, border = T, main =
           "State Distribution of prominence in Answers")
```

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

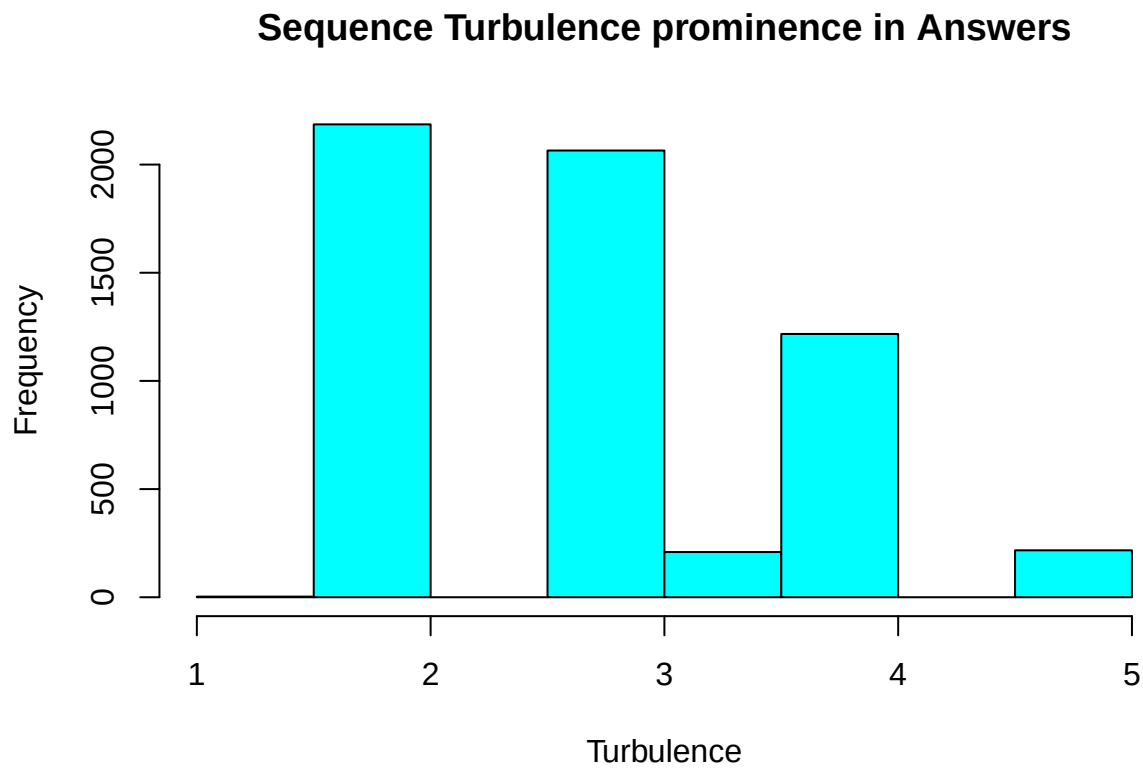
```
##  [!] In rmarkdown::render() : title is deprecated, use main instead.
```



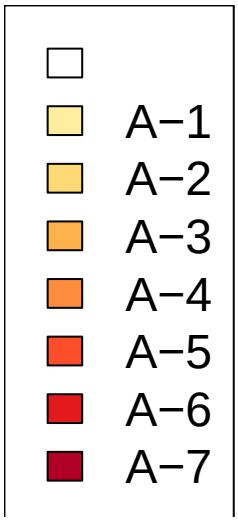

```
Turbulence <- seqST(all_sequence)
summary(Turbulence)
```

```
##    Turbulence
##  Min.   :1.000
##  1st Qu.:2.000
##  Median :3.000
##  Mean   :2.895
##  3rd Qu.:3.379
##  Max.   :5.000
```

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

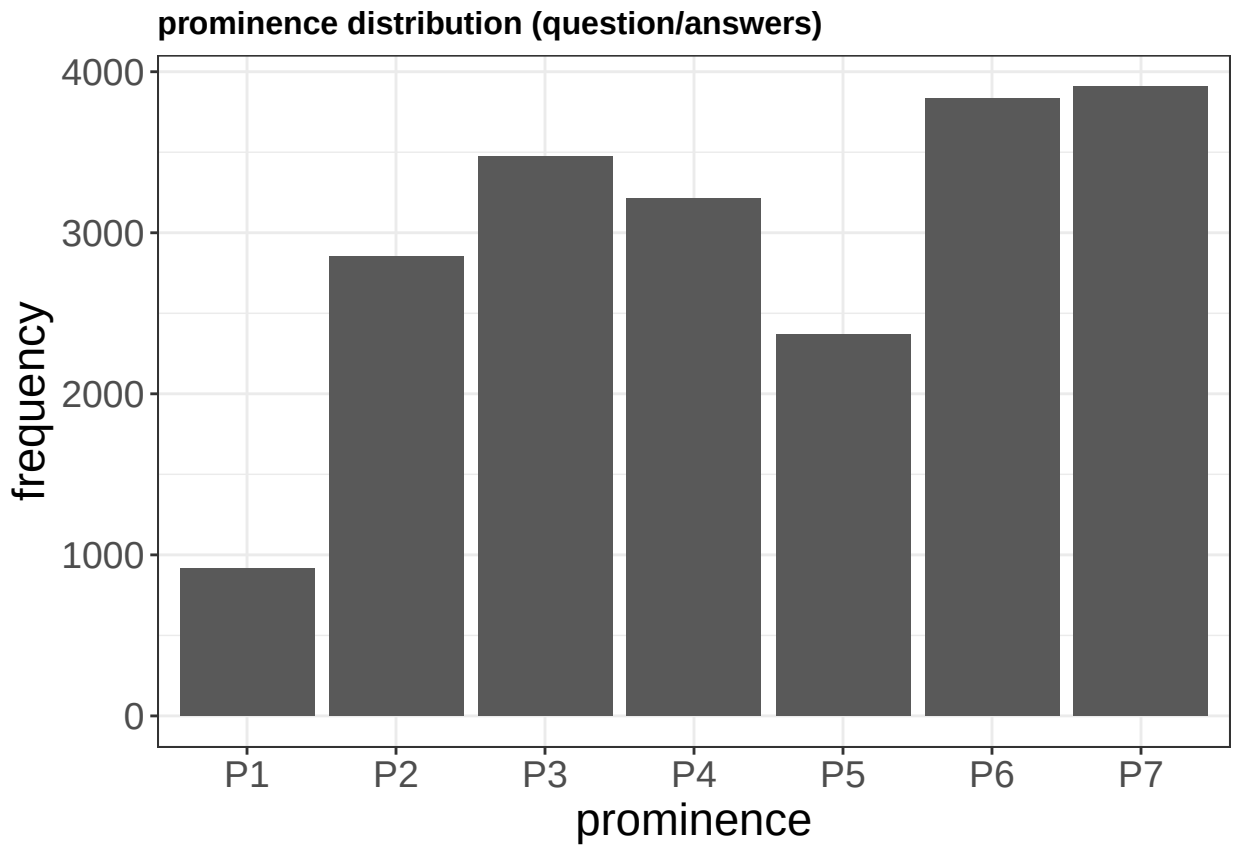



```
seqlegend(all_sequence, cex=1.5, ncol = 1 )
```




```
############concatenated q-a
agg_aq = fun.histogram(aq)
ggplot(agg_aq, aes(x = as.character(prominence), y = total))+geom_bar(stat = "identity")+
  labs(title="prominence distribution (question/answers)",x="prominence", y = "frequency") +
  scale_x_discrete(limits=c("P1", "P2","P3","P4","P5","P6","P7")) +
  theme_bw() + theme(plot.title = element_text(color = "black", size = "12", face = "bold"),
                     text = element_text(color = "black", size=17))
```

```
## Warning: Removed 1 rows containing missing values (position_stack).
```




```
##########################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")

#######################################TONE ANALYSIS################################
###TONE +/-/0  (3 elements)
###Assumption: if multiple values (toponyms) in the question;
#people wants to localize the less known one!
###The role of well-known places in the question can be related to several cases such as
#1- current state of knowledge of inquirer, or 2- disambiguation of the less known place.
questions <- read.table("../sequences/prominence-nf-Q-int.txt",
                        header = FALSE, sep = " ", col.names = paste0("V",seq_len(5)), fill = TRUE)
answers <- read.table("../sequences/prominence-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))
```

```r
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)
}
```

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

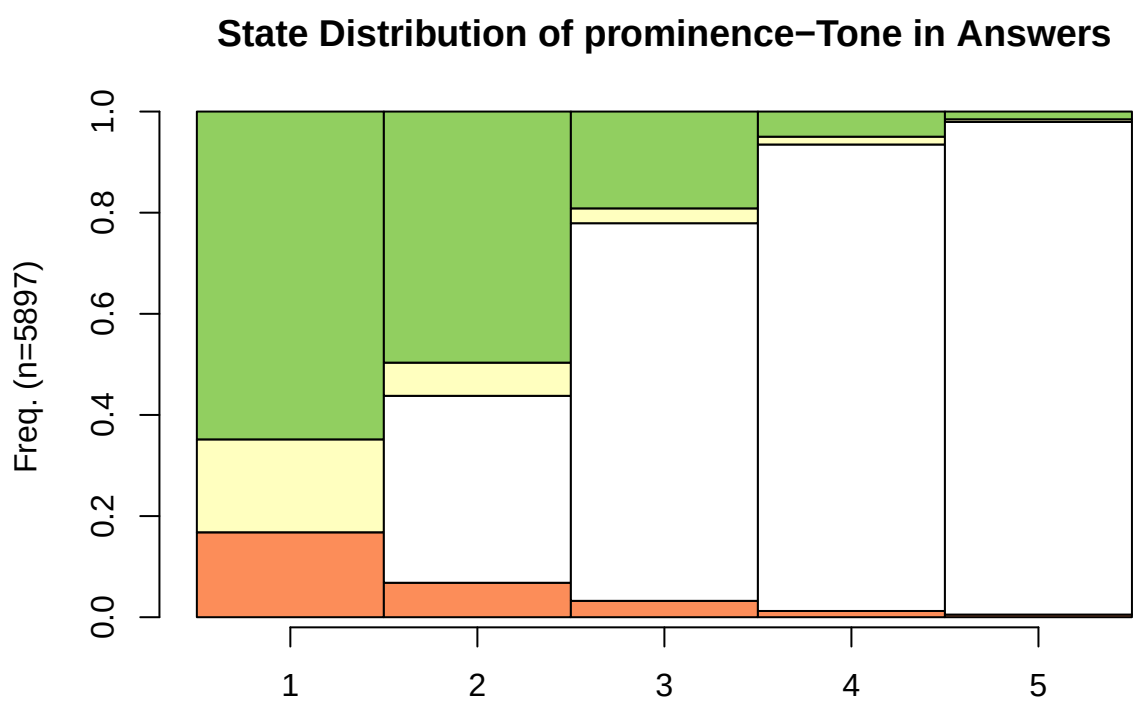
```r
answers_tone_wid = as.data.frame(answers_tone[, 2:6])
answers_tone_factor <- mapply(answers_tone_wid, FUN=as.character)
answers_tone_factor <- matrix(data=answers_tone_factor,
                              ncol=length(answers_tone_wid), nrow=length(answers_tone_wid[,1]))
for (i in 1:length(answers_tone_wid[,1])) {
  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

##  [>] 5897 sequences in the data set

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

```r
cpal(all_sequence)
```

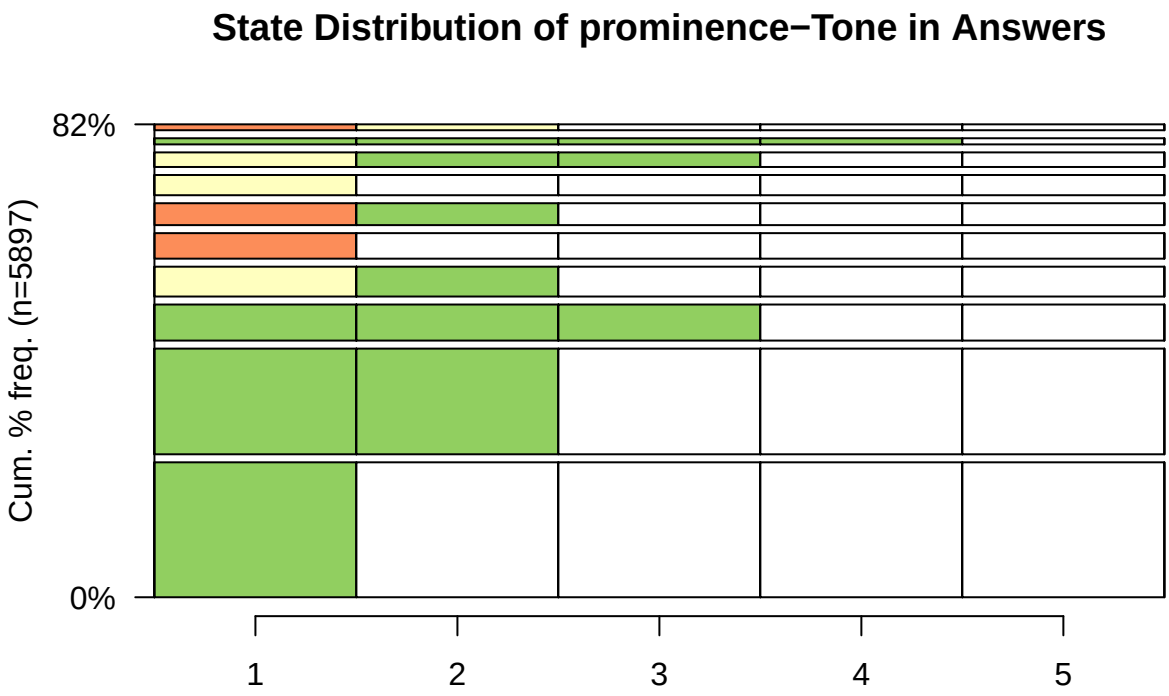
```
## [1] "#7FC97F" "#BEAED4" "#FDC086" "#FFFF99"
```

```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 prominence-Tone in Answers")
```

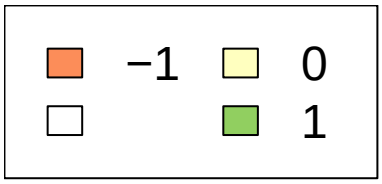



```r
seqfplot(all_sequence, with.legend = F, border = T,
         main = "State Distribution of prominence-Tone in Answers")
```

```
## 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)
```




```r
#######################################SWQ#######################################
all_qas <- read.table("../sequences/prominence-nf-all-SWQ.txt",
                      header = FALSE, sep = " ",
                      col.names = paste0("V",seq_len(19)), fill = TRUE)
head(all_qas)
```

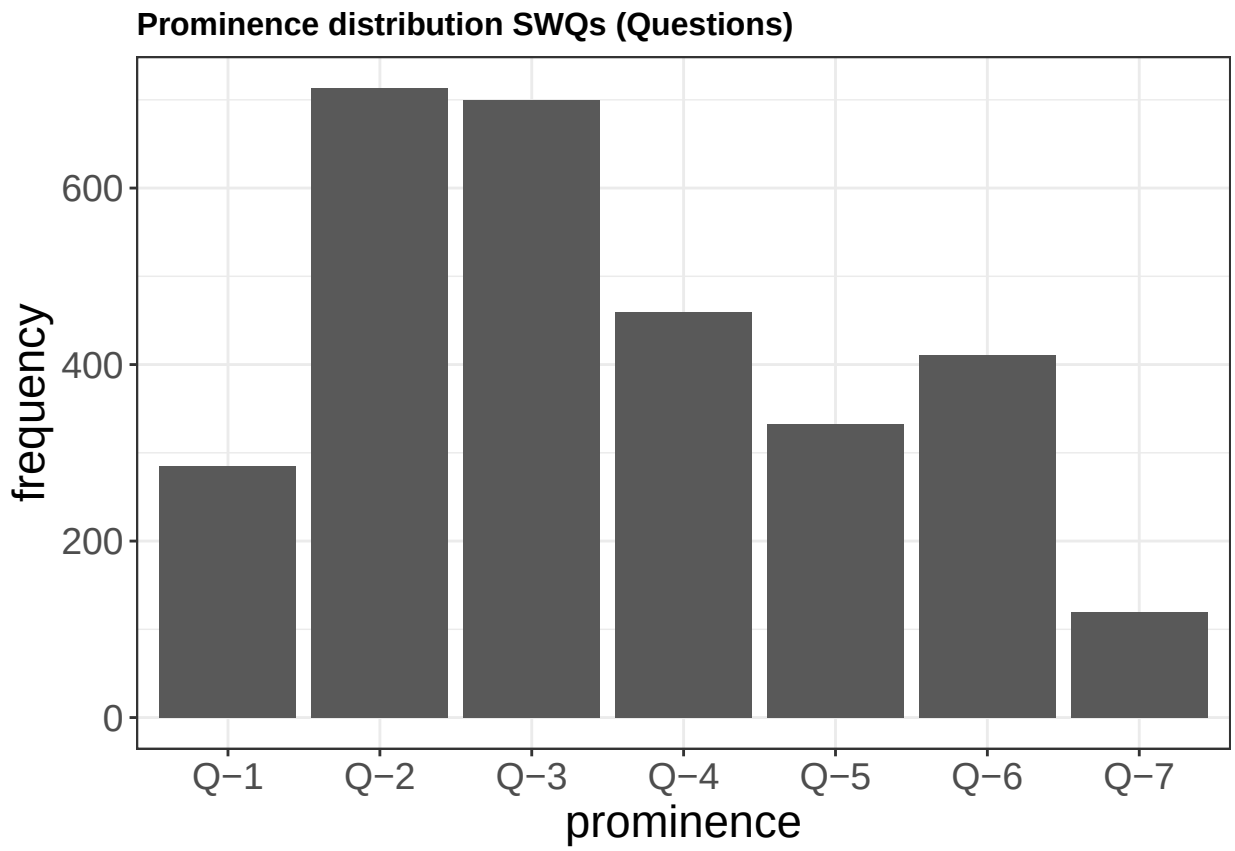
```
##    V1  V2  V3  V4  V5 V6 V7 V8 V9 V10 V11 V12 V13 V14 V15 V16 V17 V18 V19
## 1 Q-4 A-4 A-7 A-6 A-6                      NA  NA  NA  NA  NA  NA  NA  NA
## 2 Q-2 A-7                                  NA  NA  NA  NA  NA  NA  NA  NA
## 3 Q-3 A-2 A-6                              NA  NA  NA  NA  NA  NA  NA  NA
## 4 Q-4 A-4 A-3 A-7                          NA  NA  NA  NA  NA  NA  NA  NA
## 5 Q-5 A-6                                  NA  NA  NA  NA  NA  NA  NA  NA
## 6 Q-1 A-6 A-7                              NA  NA  NA  NA  NA  NA  NA  NA
```

```r
agg_aq = fun.histogram(all_qas)
```

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

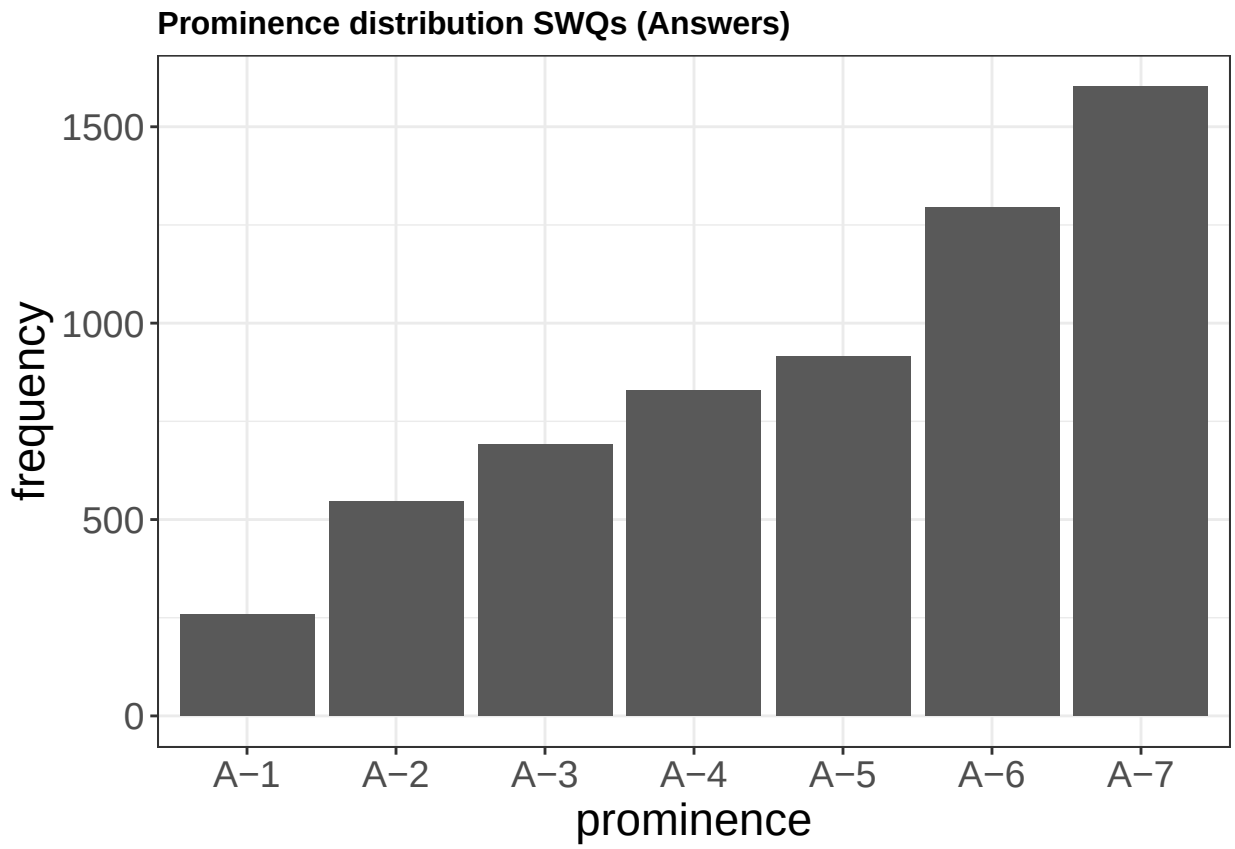
```r
ggplot(agg_aq, aes(x = as.character(prominence), y = total))+geom_bar(stat = "identity")+
  labs(title="Prominence distribution SWQs (Questions)",x="prominence", y = "frequency") +
  scale_x_discrete(limits=c("Q-1","Q-2","Q-3","Q-4","Q-5","Q-6","Q-7")) +
  theme_bw() + theme(plot.title = element_text(color = "black", size = "12", face = "bold"),
                     text = element_text(color = "black", size=17))
```

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




```
ggplot(agg_aq, aes(x = as.character(prominence), y = total))+
  geom_bar(stat = "identity")+labs(title="Prominence distribution SWQs (Answers)",
                                   x="prominence", y = "frequency") +
  scale_x_discrete(limits=c("A-1", "A-2","A-3","A-4","A-5","A-6","A-7")) +
  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).
```




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

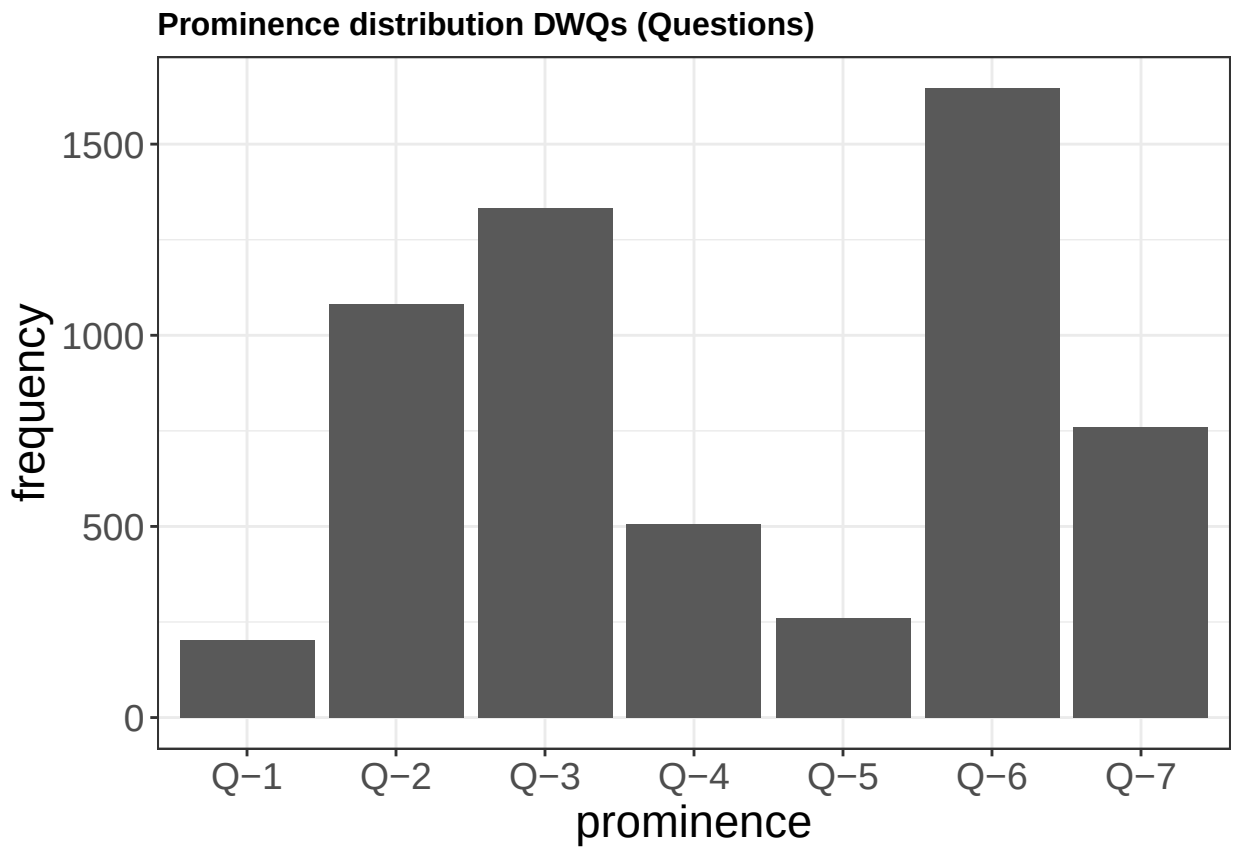
```
##    V1  V2  V3  V4  V5 V6 V7 V8 V9 V10 V11 V12 V13 V14 V15 V16 V17 V18 V19 V20
## 1 Q-3 Q-6 A-4                                                   NA  NA  NA  NA
## 2 Q-1 Q-6 A-6                                                   NA  NA  NA  NA
## 3 Q-3 Q-6 A-4 A-7                                               NA  NA  NA  NA
## 4 Q-2 Q-6 A-5                                                   NA  NA  NA  NA
## 5 Q-2 Q-6 A-2 A-4 A-7                                           NA  NA  NA  NA
## 6 Q-2 Q-7 A-4                                                   NA  NA  NA  NA
```

```
agg_aq = fun.histogram(all_qas)
```

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

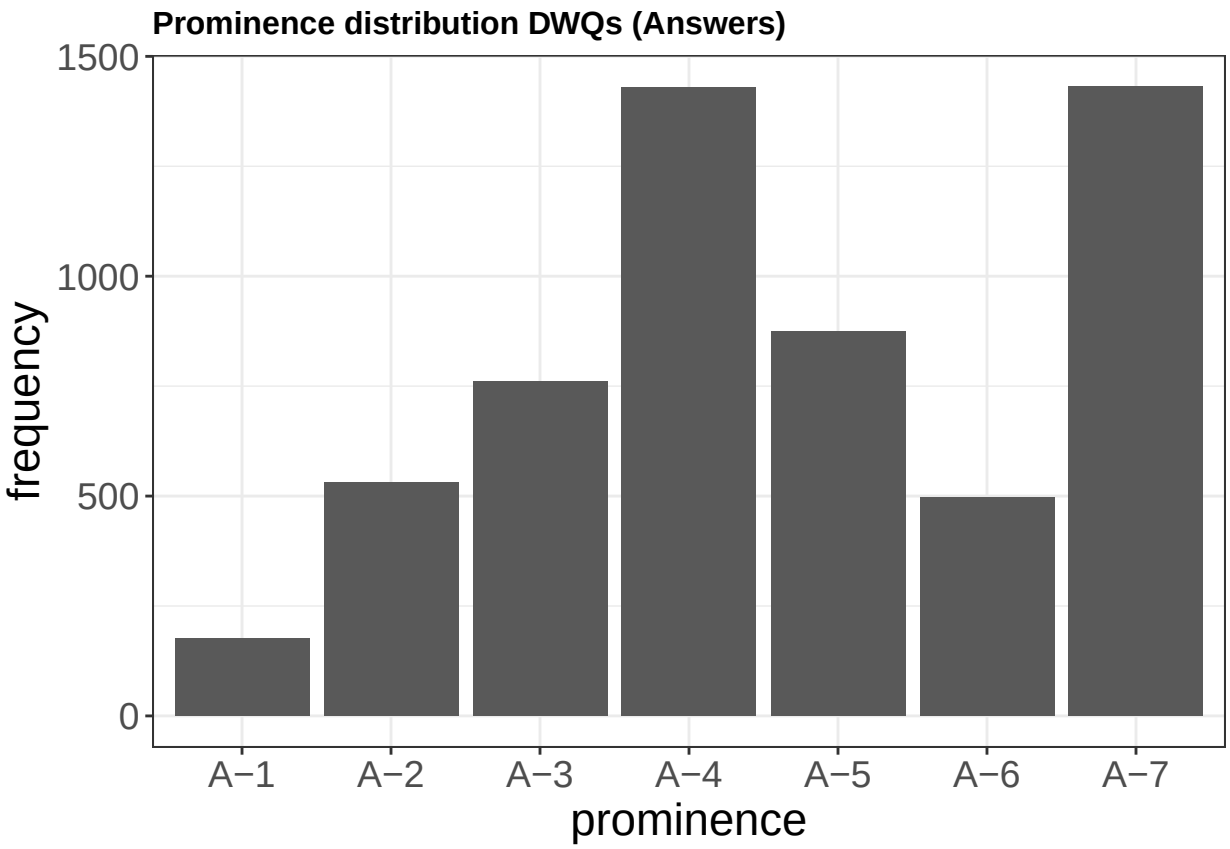
```
ggplot(agg_aq, aes(x = as.character(prominence), y = total))+geom_bar(stat = "identity")+
  labs(title="Prominence distribution DWQs (Questions)",x="prominence", y = "frequency") +
  scale_x_discrete(limits=c("Q-1","Q-2","Q-3","Q-4","Q-5","Q-6","Q-7")) +
  theme_bw() + theme(plot.title = element_text(color = "black", size = "12", face = "bold"),
                     text = element_text(color = "black", size=17))
```

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




```r
ggplot(agg_aq, aes(x = as.character(prominence), y = total))+
  geom_bar(stat = "identity")+labs(title="Prominence distribution DWQs (Answers)",
                                   x="prominence", y = "frequency") +
  scale_x_discrete(limits=c("A-1", "A-2","A-3","A-4","A-5","A-6","A-7")) +
  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).
```


Prominence distribution DWQs (Answers)
1500
1000
500
0
frequency
A−1
A−2
A−3
A−4
A−5
A−6
A−7
prominence