```
install.packages("twitteR")
install.packages("ROAuth")
install.packages("tidytext")
install.packages("tm")
install.packages("wordcloud")
install.packages("igraph")
install.packages("glue")
install.packages("networkD3")
install.packages("rtweet")
install.packages("plyr")
install.packages("stringr")
install.packages("ggplot2")
install.packages("ggeasy")
install.packages("plotly")
install.packages("dplyr")
install.packages("hms")
install.packages("lubridate")
install.packages("magrittr")
install.packages("tidyverse")
install.packages("janeaustenr")
install.packages("widyr")
```

```
Installing package into ‘/usr/local/lib/R/site-library’
(as ‘lib’ is unspecified)

also installing the dependency ‘rjson’


Installing package into ‘/usr/local/lib/R/site-library’
(as ‘lib’ is unspecified)

also installing the dependencies ‘bitops’, ‘RCurl’
```

```
library(twitteR)
library(ROAuth)
library(hms)
library(lubridate)
library(tidytext)
library(tm)
library(wordcloud)
library(igraph)
library(glue)
library(networkD3)
library(rtweet)
library(plyr)
library(stringr)
library(ggplot2)
library(ggeasy)
library(plotly)
library(dplyr)
library(hms)
library(lubridate)
library(magrittr)
library(tidyverse)
library(janeaustenr)
library(widyr)
```

```
Attaching package: ‘twitteR’


The following objects are masked from ‘package:dplyr’:

    id, location


The following object is masked from ‘package:plyr’:

    id


The following object is masked from ‘package:rtweet’:

    lookup_statuses
```

```
api_key <- "xaR1HcoIMOpWxObXkQtzNZzJj"
api_secret <- "i7MNkwPuomoDe8pcj2IN3nUznKzdPCshycHskdr69sKwvJfUho"
access_token <- "1103974251491663873-WXe4Eg9fNoKVqu44N0hs2y5AWnyc8b"
access_token_secret <- "Z7M7bp8ZrMPEdh7jJnwBLql8Twq8zPI9jRAu98qXYtFo2"
#Note: This will ask us permission for direct authentication, type '1' for yes:
setup_twitter_oauth(api_key,api_secret,access_token,access_token_secret)
```

```
[1] "Using direct authentication"
```

```
# extracting 10000 tweets related to Mahsa_Amini
tweets <- searchTwitter("#Mahsa_Amini", n=10000, lang="en")
n.tweet <- length(tweets)

# convert tweets to a data frame
tweets.df <- twListToDF(tweets)
```

```
tweets.txt <- sapply(tweets, function(t)t$getText())
# Ignore graphical Parameters to avoid input errors
tweets.txt <- str_replace_all(tweets.txt,"[^[:graph:]]", " ")
tweets.txt <- str_replace_all(tweets.txt, "[^[:alnum:]]", " ")

## pre-processing text:
```

```
clean.text = function(x)
{
  # convert to lower case
  x = tolower(x)
  # remove rt
  x = gsub("rt", "", x)
  # remove at
  x = gsub("@\\w+", "", x)
  # remove punctuation
  x = gsub("[[:punct:]]", "", x)
  # remove numbers
  x = gsub("[[:digit:]]", "", x)
  # remove links http
  x = gsub("http\\w+", "", x)
  # remove tabs
  x = gsub("[ |\t]{2,}", "", x)
  # remove blank spaces at the beginning
  x = gsub("^ ", "", x)
  # remove blank spaces at the end
  x = gsub(" $", "", x)
  # some other cleaning text
  x = gsub('https://','',x)
  x = gsub('http://','',x)
  x = gsub('[^[:graph:]]', ' ',x)
  x = gsub('[[:punct:]]', '', x)
  x = gsub('[[:cntrl:]]', '', x)
  x = gsub('\\d+', '', x)
  x = str_replace_all(x,"[^[:graph:]]", " ")
  return(x)
}

cleanText <- clean.text(tweets.txt)
# remove empty results (if any)
idx <- which(cleanText == " ")
cleanText <- cleanText[cleanText != " "]
```

```
tweets.df %<>%
  mutate(
    created = created %>%
      # Remove zeros.
      str_remove_all(pattern = '\\+0000') %>%
      # Parse date.
      parse_date_time(orders = '%y-%m-%d %H%M%S')
  )

tweets.df %<>%
  mutate(Created_At_Round = created%>% round(units = 'hours') %>% as.POSIXct())

tweets.df %>% pull(created) %>% min()
```

```
    [1] "2022-12-09 08:03:47 UTC"
```

```
tweets.df %>% pull(created) %>% max()
```

```
  [1] "2022-12-10 04:05:23 UTC"
```

```
# URL <- "https://www.dropbox.com/s/j74otamjln4a5qw/opinion-lexicon-English.zip"
# download.file(URL, destfile = "opinion-lexicon-English.zip", method="curl")
```

Sentimental Analysis

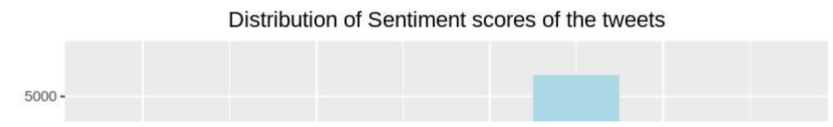
```
positive = scan('/content/positive-words.txt', what = 'character', comment.char = ';')
negative = scan('/content/negative-words.txt', what = 'character', comment.char = ';')
# add your list of words below as you wish if missing in above read lists
pos.words = c(positive,'upgrade','Congrats','prizes','prize','thanks','thnx',
              'Grt','gr8','plz','trending','recovering','brainstorm','leader')
neg.words = c(negative,'wtf','wait','waiting','epicfail','Fight','fighting',
              'arrest','no','not')
```

```
  score.sentiment = function(sentences, pos.words, neg.words, .progress='none')
  {
    require(plyr)
    require(stringr)

    # we are giving vector of sentences as input.
    # plyr will handle a list or a vector as an "l" for us
    # we want a simple array of scores back, so we use "l" + "a" + "ply" = laply:
    scores = laply(sentences, function(sentence, pos.words, neg.words) {

      # clean up sentences with R's regex-driven global substitute, gsub() function:
      sentence = gsub('https://','',sentence)
      sentence = gsub('http://','',sentence)
      sentence = gsub('[^[:graph:]]', ' ',sentence)
      sentence = gsub('[[:punct:]]', '', sentence)
      sentence = gsub('[[:cntrl:]]', '', sentence)
      sentence = gsub('\\d+', '', sentence)
      sentence = str_replace_all(sentence,"[^[:graph:]]", " ")
      # and convert to lower case:
      sentence = tolower(sentence)

      # split into words. str_split is in the stringr package
      word.list = str_split(sentence, '\\s+')
      # sometimes a list() is one level of hierarchy too much
      words = unlist(word.list)

      # compare our words to the dictionaries of positive & negative terms
      pos.matches = match(words, pos.words)
      neg.matches = match(words, neg.words)

    # match() returns the position of the matched term or NA
    # we just want a TRUE/FALSE:
    pos.matches = !is.na(pos.matches)
    neg.matches = !is.na(neg.matches)

    # TRUE/FALSE will be treated as 1/0 by sum():
    score = sum(pos.matches) - sum(neg.matches)

    return(score)
  }, pos.words, neg.words, .progress=.progress )

  scores.df = data.frame(score=scores, text=sentences)
  return(scores.df)
}
```

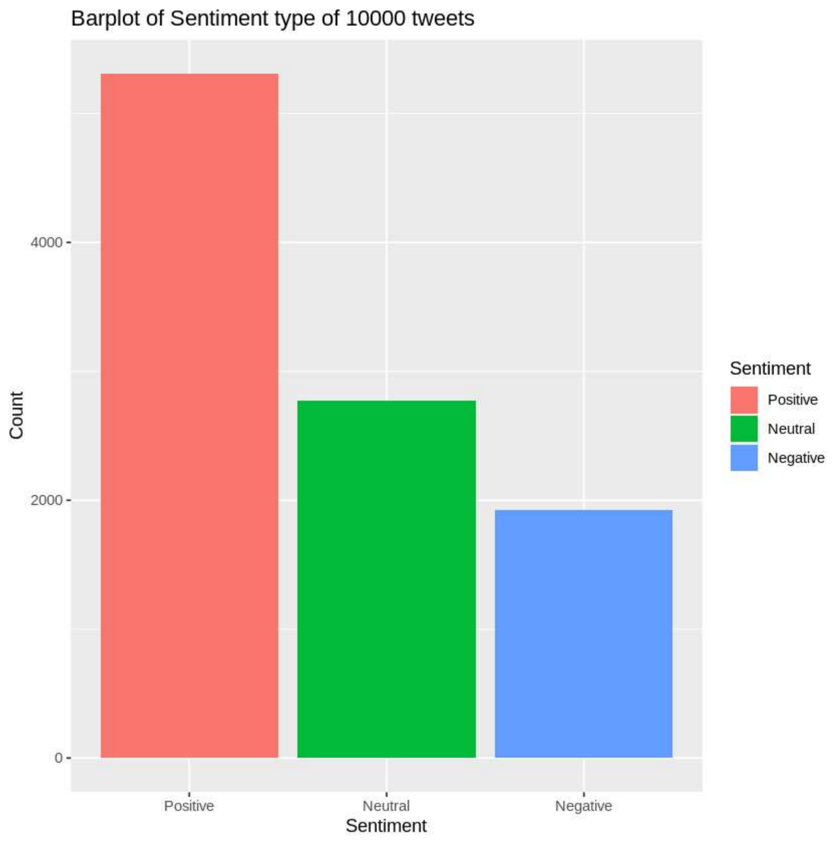
```
analysis <- score.sentiment(cleanText, pos.words, neg.words)
# sentiment score frequency table
table(analysis$score)
```

```
  -4   -3   -2   -1    0    1    2    3 
   1  120  646 1156 2771 5159  139    8 
```

```
analysis %>%
  ggplot(aes(x=score)) +
  geom_histogram(binwidth = 1, fill = "lightblue")+
  ylab("Frequency") +
  xlab("sentiment score") +
  ggtitle("Distribution of Sentiment scores of the tweets") +
  ggeasy::easy_center_title()
```

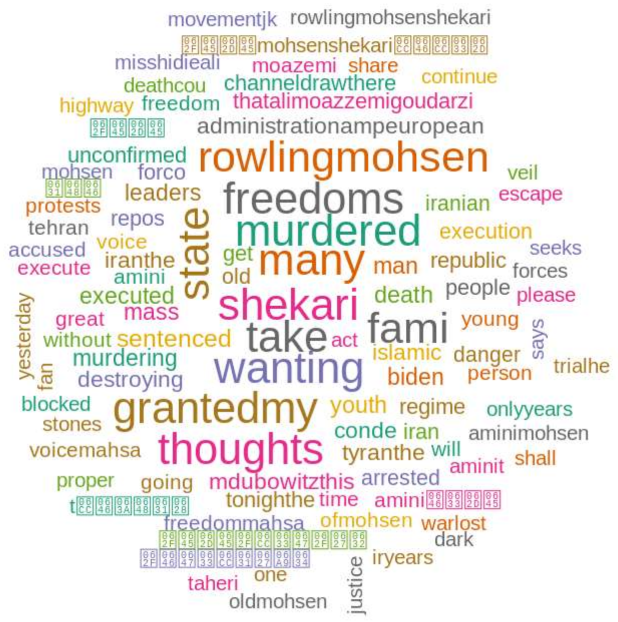


```r
neutral <- length(which(analysis$score == 0))
positive <- length(which(analysis$score > 0))
negative <- length(which(analysis$score < 0))
Sentiment <- c("Positive","Neutral","Negative")
Count <- c(positive,neutral,negative)
output <- data.frame(Sentiment,Count)
output$Sentiment<-factor(output$Sentiment,levels=Sentiment)
ggplot(output, aes(x=Sentiment,y=Count))+
  geom_bar(stat = "identity", aes(fill = Sentiment))+
  ggtitle("Barplot of Sentiment type of 10000 tweets")
```

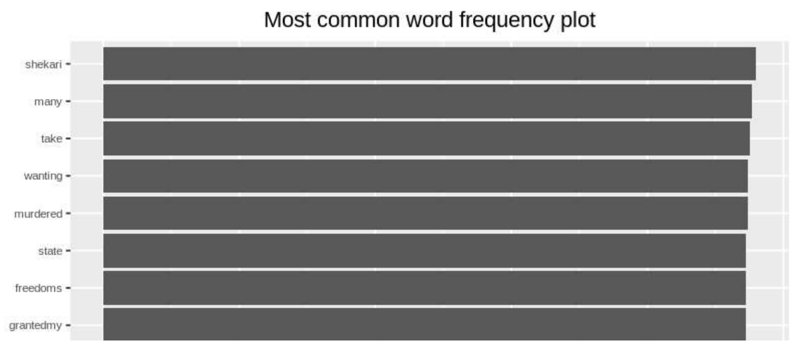


```r
text_corpus <- Corpus(VectorSource(cleanText))
text_corpus <- tm_map(text_corpus, content_transformer(tolower))
text_corpus <- tm_map(text_corpus, function(x)removeWords(x,stopwords("english")))
text_corpus <- tm_map(text_corpus, removeWords, c("global","globalwarming"))
tdm <- TermDocumentMatrix(text_corpus)
tdm <- as.matrix(tdm)
tdm <- sort(rowSums(tdm), decreasing = TRUE)
tdm <- data.frame(word = names(tdm), freq = tdm)
set.seed(123)
wordcloud(text_corpus, min.freq = 1, max.words = 100, scale = c(2.2,1),
          colors=brewer.pal(8, "Dark2"), random.color = T, random.order = F)
```

```
Warning message in tm_map.SimpleCorpus(text_corpus, content_transformer(tolower)):
“transformation drops documents”
Warning message in tm_map.SimpleCorpus(text_corpus, function(x) removeWords(x, stopword
“transformation drops documents”
Warning message in tm_map.SimpleCorpus(text_corpus, removeWords, c("global", "globalwar
“transformation drops documents”
```



```
ggplot(tdm[1:20,], aes(x=reorder(word, freq), y=freq)) +
  geom_bar(stat="identity") +
  xlab("Terms") +
  ylab("Count") +
  coord_flip() +
  theme(axis.text=element_text(size=7)) +
  ggtitle('Most common word frequency plot') +
  ggeasy::easy_center_title()
```



```
#bigram
bi.gram.words <- tweets.df %>%
  unnest_tokens(
    input = text,
    output = bigram,
    token = 'ngrams',
    n = 2
  ) %>%
  filter(! is.na(bigram))

bi.gram.words %>%
  select(bigram) %>%
  head(10)
```

A data.frame: 10 × 1

	bigram <chr>
1	rt carmen_ocean_
2	carmen_ocean_ alimoazemi
3	alimoazemi is
4	is sentenced
5	sentenced to
6	to death
7	death and
8	and is
9	is transferred
10	transferred to

```
extra.stop.words <- c('https')
stopwords.df <- tibble(
  word = c(stopwords(kind = 'es'),
           stopwords(kind = 'en'),
          extra.stop.words)
```

```
bi.gram.words %<>%
  separate(col = bigram, into = c('word1', 'word2'), sep = ' ') %>%
  filter(! word1 %in% stopwords.df$word) %>%
  filter(! word2 %in% stopwords.df$word) %>%
  filter(! is.na(word1)) %>%
  filter(! is.na(word2))
```

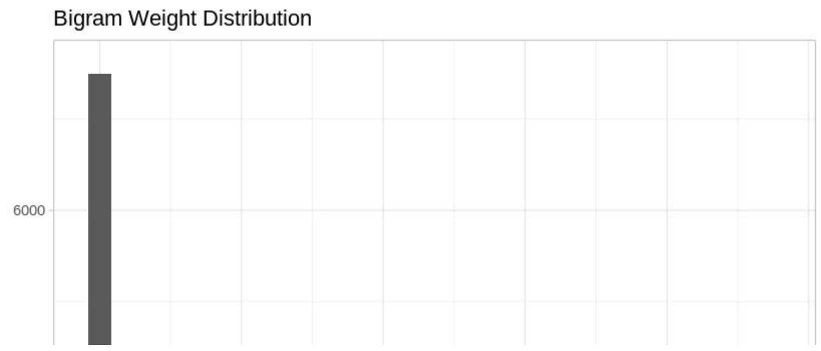
```
bi.gram.count <- bi.gram.words %>%
  dplyr::count(word1, word2, sort = TRUE) %>%
  dplyr::rename(weight = n)

bi.gram.count %>% head()
```

A data.frame: 6 × 3

	word1	word2	weight
	<chr>	<chr>	<int>
1	wanting	freedoms	4725
2	us	take	4724
3	mohsen	shekari	4679
4	jk_rowling	mohsen	4616
5	rt	jk_rowling	4613
6	rt	mdubowitz	689

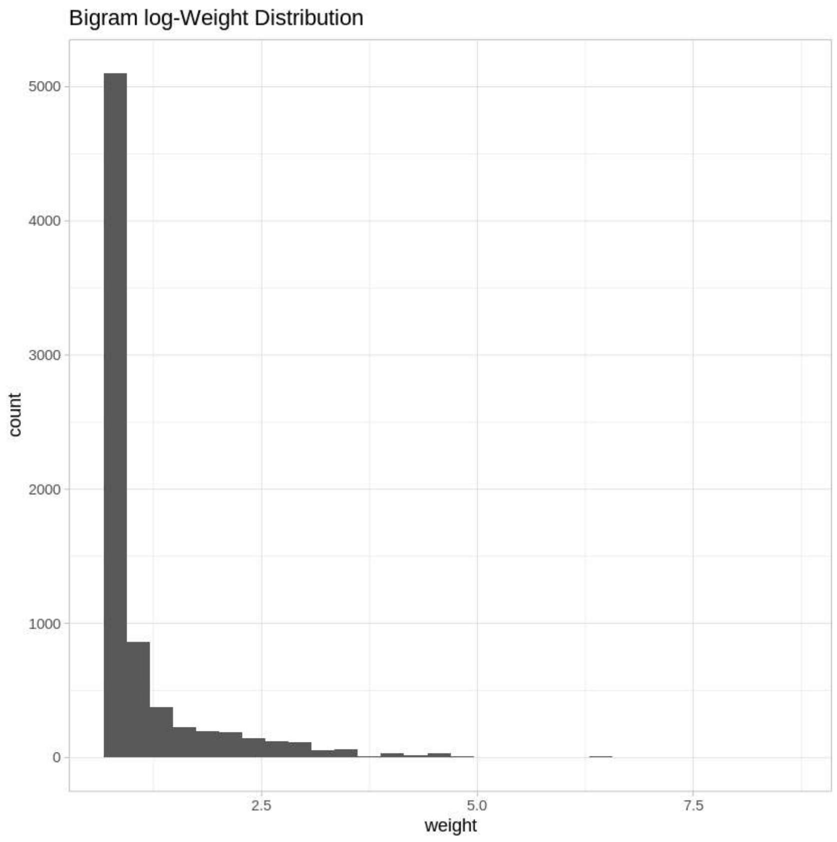
```
bi.gram.count %>%
  ggplot(mapping = aes(x = weight)) +
  theme_light() +
  geom_histogram() +
  labs(title = "Bigram Weight Distribution")
```

```
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

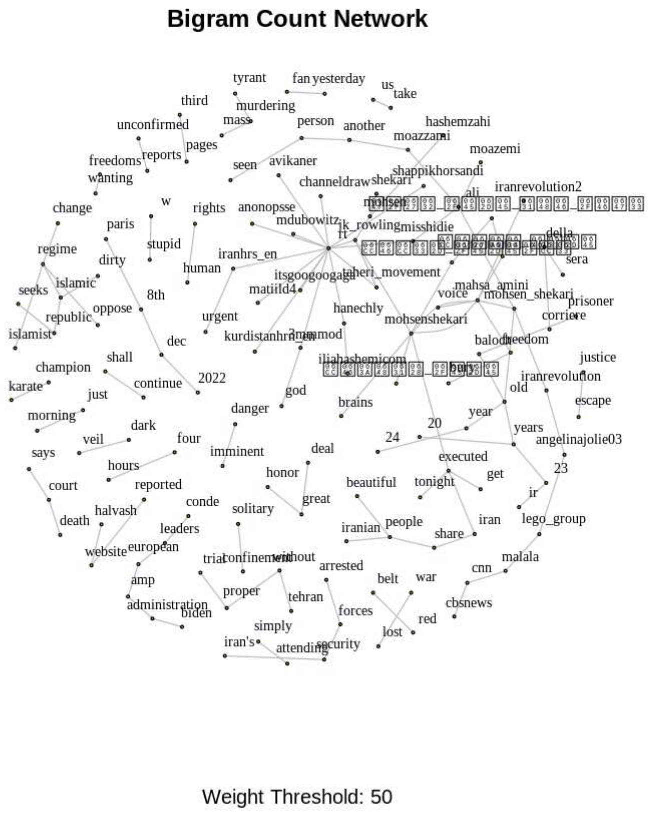


```
bi.gram.count %>%
  mutate(weight = log(weight + 1)) %>%
  ggplot(mapping = aes(x = weight)) +
  theme_light() +
  geom_histogram() +
  labs(title = "Bigram log-Weight Distribution")
```

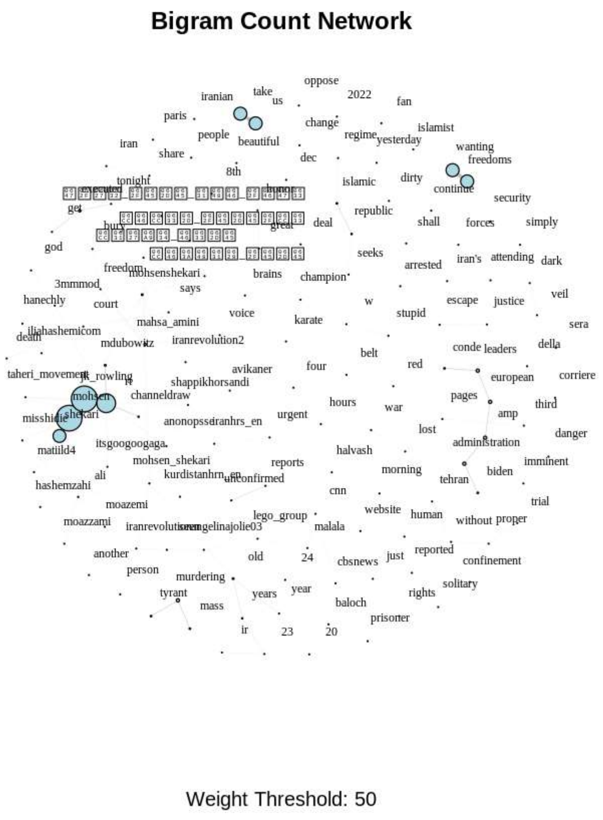
```
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```
threshold <- 50

# For visualization purposes we scale by a global factor.
ScaleWeight <- function(x, lambda) {
  x / lambda
}

network <-  bi.gram.count %>%
  filter(weight > threshold) %>%
  mutate(weight = ScaleWeight(x = weight, lambda = 2E3)) %>%
  graph_from_data_frame(directed = FALSE)

plot(
  network,
  vertex.size = 1,
  vertex.label.color = 'black',
  vertex.label.cex = 0.7,
  vertex.label.dist = 1,
  edge.color = 'gray',
  main = 'Bigram Count Network',
  sub = glue('Weight Threshold: {threshold}'),
  alpha = 50
)
```



```
V(network)$degree <- strength(graph = network)

# Compute the weight shares.
E(network)$width <- E(network)$weight/max(E(network)$weight)

plot(
  network,
  vertex.color = 'lightblue',
  # Scale node size by degree.
  vertex.size = 2*V(network)$degree,
  vertex.label.color = 'black',
  vertex.label.cex = 0.6,
vertex.label.dist = 1.6,
edge.color = 'gray',
# Set edge width proportional to the weight relative value.
edge.width = 3*E(network)$width ,
main = 'Bigram Count Network',
sub = glue('Weight Threshold: {threshold}'),
alpha = 50
)
```


Bigram Count Network
oppose
iranian
take
us
fan
paris
change
islamist
regime
yesterday
people
beautiful
iran
share
wanting
dec
freedoms
8th
tonight
executed
islamic
dirty
continue
security
republic
get
great
deal
shall
forces
simply
bury
god
seeks
arrested
iran's
attending
dark
freedom
mohsenshekari
brains
champion
3mmmod
says
veil
hanechly
court
voice
w
escape
justice
mahsa_amini
karate
stupid
sera
death
iliahashemicom
mdubowitz
iranrevolution2
conde
leaders
della
belt
taheri_movement
jk_rowling
avikaner
four
red
european
corriere
shappikhorsandi
mohsen
channeldraw
pages
third
hours
war
shekari
amp
misshidie
anonopssd
iranhrs_en
urgent
lost
danger
matiild4
administration
itsgoogoogaga
halvash
mohsen_shekari
imminent
reports
ali
kurdistanhrn_en
morning
biden
unconfirmed
tehran
hashemzahi
cnn
trial
moazemi
website
human
lego_group
without
proper
moazzami
iranrevolution
angelinajolie03
malala
another
reported
old
cbsnews
just
confinement
person
murdering
solitary
year
rights
tyrant
years
baloch
mass
prisoner
ir
Weight Threshold: 50