BLin_Assign6

Bin Lin

2017-3-10

1. Problem Set 1

1. When you roll a fair die 3 times, how many possible outcomes are there?

```
6 ^ 3
```

```
## [1] 216
```

2. What is the probability of getting a sum total of 3 when you roll a die two times?

There are two possibilities, (1, 2) and (2, 1). Since when we roll a die two times, the sum has total 6 * 6 possibilities, therefore the probability of getting a sum of 3 is 3 / 36 = 0.0833

3. Assume a room of 25 strangers. What is the probability that two of them have the same birthday? Assume that all birthdays are equally likely and equal to 1=365 each. What happens to this probability when there are 50 people in the room?

```
#Two of them have the same birthday is equal to the complement of event that not one has same bi
rthday.

#1 - (factorial(365)/factorial(365-25)) / (365 ^ 25) value out of range in 'gammafn'
1 - prod(365:(365-25+1))/(365^25)
```

```
## [1] 0.5686997
```

```
#When there are 50 peopl in the room?
1 - prod(365:(365-50+1))/(365^50)
```

```
## [1] 0.9703736
```

2. Problem Set 2 Write a program to take a document in English and print out the estimated probabilities for each of the words that occur in that document. Your program should take in a file containing a large document and write out the probabilities of each of the words that appear in that document. Please remove all punctuation (quotes, commas, hyphens etc) and convert the words to lower case before you perform your calculations.

```
library(stringr)

words_probability <- function (text)
{
  # I am removing all the punctuations and change all strings to the lower case.
  document <- str_replace_all(text,"[[:punct:]]","")
  #Or gsub("[[:punct:]]", "", raw_data)
  document <- str_to_lower(document)


  # I am using the str_split function to seperate the large string into individual words. The un
list function is going to return the result as vectors. Then the str_extract can remove words wh
ich contain numbers and dollar signs.
  words <- unlist(str_split(document, "[ ]"))
  words <- str_extract(words, "[A-z]+")



  # I am creating the probability table and change the result to a data frame. The last step is
 just simply order the data frame in descending order of the probability.
  words_table <- table(words)
  prop.table(words_table)

  words_table <- data.frame(prop.table(words_table))
  words_table <- words_table[order(-words_table$Freq),]
  return (words_table)
}
```

```
raw_data <- readLines("C:/Users/blin261/Desktop/DATA605/assign6/assign6.sample.txt", encoding =
"UTF-8")
```

```
## Warning in readLines("C:/Users/blin261/Desktop/DATA605/assign6/
## assign6.sample.txt", : incomplete final line found on 'C:/Users/blin261/
## Desktop/DATA605/assign6/assign6.sample.txt'
```

```
raw_data <- paste(raw_data, collapse = " ")

words_table <- words_probability(raw_data)
head(words_table)
```

```
##     words       Freq
## 489   the 0.05697151
## 1       a 0.03373313
## 24    and 0.02848576
## 175   for 0.02323838
## 222    in 0.02098951
## 328    of 0.02098951
```

Extend your program to calculate the probability of two words occurring adjacent to each other. It should take in a document, and two words (say the and for) and compute the probability of each of the words occurring in the document and the joint probability of both of them occurring together. The order of the two words is not important.

```
two_words <- function(text, word1, word2)
{
  #I am getting the probability of each words using the function for one-word probability.
  words_table <- words_probability(text)
  word1_p <- words_table$Freq[words_table$words == word1]
  word2_p <- words_table$Freq[words_table$words == word2]

  #The following steps are very similar to previsous step. It is all about tidying and transform
ation.
  document <- str_replace_all(raw_data,"[[:punct:]]","")
  #Or gsub("[[:punct:]]", "", raw_data)
  document <- str_to_lower(document)
  words <- unlist(str_split(document, "[ ]"))
  words <- str_extract(words, "[A-z]+")

  #It is necessary to get the frequency of the combination of two words. The total number of two
words combination is just the total number of single words substract 1. Then I am able to calcul
ate the joint probability.
  count1 <- str_count(document, paste(word1, word2))
  count2 <- str_count(document, paste(word2, word1))
  total_count <- length(words) - 1
  words_p <- (count1 + count2) / total_count


  print (paste(c(word1, "probability is", toString(word1_p)), collapse = " "))
  print (paste(c(word2, "probability is", toString(word2_p)), collapse = " "))
  print (paste(c("Joint probability is", toString(words_p)), collapse = " "))
}
```

```
raw_data <- readLines("C:/Users/blin261/Desktop/DATA605/assign6/assign6.sample.txt", encoding =
"UTF-8")
```

```
## Warning in readLines("C:/Users/blin261/Desktop/DATA605/assign6/
## assign6.sample.txt", : incomplete final line found on 'C:/Users/blin261/
## Desktop/DATA605/assign6/assign6.sample.txt'
```

```
raw_data <- paste(raw_data, collapse = " ")
two_words(raw_data, "for", "the")
```

```
## [1] "for probability is 0.0232383808095952"
## [1] "the probability is 0.0569715142428786"
## [1] "Joint probability is 0.00287150035893754"
```