# Exercise: RegEx

*Peter Meißner*

*28 Februar 2016*

## 1) Character Encoding

```
readLines("text1.txt")
```

```
## [1] "Iñtërnâtiônàlizætiøn"
```

```
readLines("text2.txt")
```

```
## [1] "I\xf1t\xebrn\xe2ti\xf4n\xe0liz\xe6ti\xf8n"
```

a) use the encoding option of readLines() to read show files correctly

```
readLines("text1.txt", encoding = "UTF-8")
```

```
## [1] "Iñtërnâtiônàlizætiøn"
```

```
readLines("text2.txt", encoding = "latin1")
```

```
## [1] "Iñtërnâtiônàlizætiøn"
```

b) use Encoding to show the files correctly

```
text1 <- readLines("text1.txt")
Encoding(text1) <- "UTF-8"
text1
```

```
## [1] "Iñtërnâtiônàlizætiøn"
```

```
text2 <- readLines("text2.txt")
Encoding(text2) <- "latin1"
text2
```

```
## [1] "Iñtërnâtiônàlizætiøn"
```

c) use iconv to solve the problem

```
text1 <- readLines("text1.txt")
text1 <- iconv(text1, "UTF-8", "UTF-8")
text1
```

```
## [1] "Iñtërnâtiônàlizætiøn"
```

```
text2 <- readLines("text2.txt")
text2 <- iconv(text2, "latin1", "UTF-8")
text2
```

```
## [1] "Iñtërnâtiônàlizætiøn"
```

d) use stringi::stri_enc_detect() to guess Encoding of the files

```
library(stringi)

text1 <- readLines("text1.txt")
stri_enc_detect(text1)
```

```
## [[1]]
## [[1]]$Encoding
## [1] "UTF-8"      "ISO-8859-1" "UTF-16BE"   "UTF-16LE"   "Shift_JIS"
## [6] "GB18030"    "Big5"       "IBM420_ltr"
##
## [[1]]$Language
## [1] ""   "da" ""   ""   "ja" "zh" "zh" "ar"
##
## [[1]]$Confidence
## [1] 1.00 0.21 0.10 0.10 0.10 0.10 0.10 0.10
```

```
stri_enc_detect2(text1)
```

```
## [[1]]
## [[1]]$Encoding
## [1] "UTF-8"
##
## [[1]]$Language
## [1] NA
##
## [[1]]$Confidence
## [1] 1
```

```
text2 <- readLines("text2.txt")
stri_enc_detect(text2)
```

```
## [[1]]
## [[1]]$Encoding
## [1] "UTF-16BE"  "UTF-16LE"  "Shift_JIS" "GB18030"   "Big5"
##
## [[1]]$Language
## [1] ""   ""   "ja" "zh" "zh"
##
## [[1]]$Confidence
## [1] 0.1 0.1 0.1 0.1 0.1
```

```
stri_enc_detect2(text2)
```

```
## [[1]]
## [[1]]$Encoding
##  [1] "macintosh"             "x-mac-turkish"
##  [3] "x-roman8"              "x-mac-centraleurroman"
##  [5] "ISO-8859-1"            "ISO-8859-2"
##  [7] "ISO-8859-3"            "ISO-8859-4"
##  [9] "ISO-8859-9"            "ISO-8859-10"
## [11] "ISO-8859-13"           "iso-8859_14-1998"
## [13] "ISO-8859-15"           "ibm-901_P100-1999"
## [15] "ibm-902_P100-1999"     "cp922"
## [17] "windows-1250"          "windows-1252"
## [19] "windows-1254"          "windows-1257"
## [21] "windows-1258"          "ibm-1250_P100-1995"
## [23] "ibm-1252_P100-2000"    "ibm-1254_P100-1995"
## [25] "ibm-1257_P100-1995"    "ibm-5353_P100-1998"
## [27] "ibm-1258_P100-1997"    "x-mac-greek"
## [29] "ibm-1129_P100-1997"
##
## [[1]]$Language
##  [1] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA
## [24] NA NA NA NA NA NA
##
## [[1]]$Confidence
##  [1] 0.7142857 0.7142857 0.6428571 0.5714286 0.5000000 0.5000000 0.5000000
##  [8] 0.5000000 0.5000000 0.5000000 0.5000000 0.5000000 0.5000000 0.5000000
## [15] 0.5000000 0.5000000 0.5000000 0.5000000 0.5000000 0.5000000 0.5000000
## [22] 0.5000000 0.5000000 0.5000000 0.5000000 0.5000000 0.5000000 0.5000000
## [29] 0.5000000
```

e) write your name with \u0000 expressions (http://unicode-table.com/en/), e.g.:

```
name <- "\u0055\u0064\u006f"
name
```

```
## [1] "Udo"
```

f) read in 17814-0.txt and get the encoding right

```
readLines("17814-0.txt", encoding = "UTF-8")[1:10]
```

```
##  [1] "The Project Gutenberg EBook of Lysistrata, by Aristophanes"
##  [2] ""
##  [3] "This eBook is for the use of anyone anywhere at no cost and with"
##  [4] "almost no restrictions whatsoever.  You may copy it, give it away or"
##  [5] "re-use it under the terms of the Project Gutenberg License included"
##  [6] "with this eBook or online at www.gutenberg.org"
##  [7] ""
##  [8] ""
##  [9] "Title: Lysistrata"
## [10] ""
```

## 2) Information Extraction

a) read in pg345.txt

- count how many times the word “blood” or “Blood” is used throughout the book
- use grep() to get an index of lines containing the word
- use hist(. . . ,n=100000) to make a “zebra?”-chart
- what might be other interesting words?
- can you add them to your RegEx?

```
library(stringr)
text <- readLines("pg345.txt")

sum(str_count(text, "[bB]lood"))
```

```
## [1] 121
```

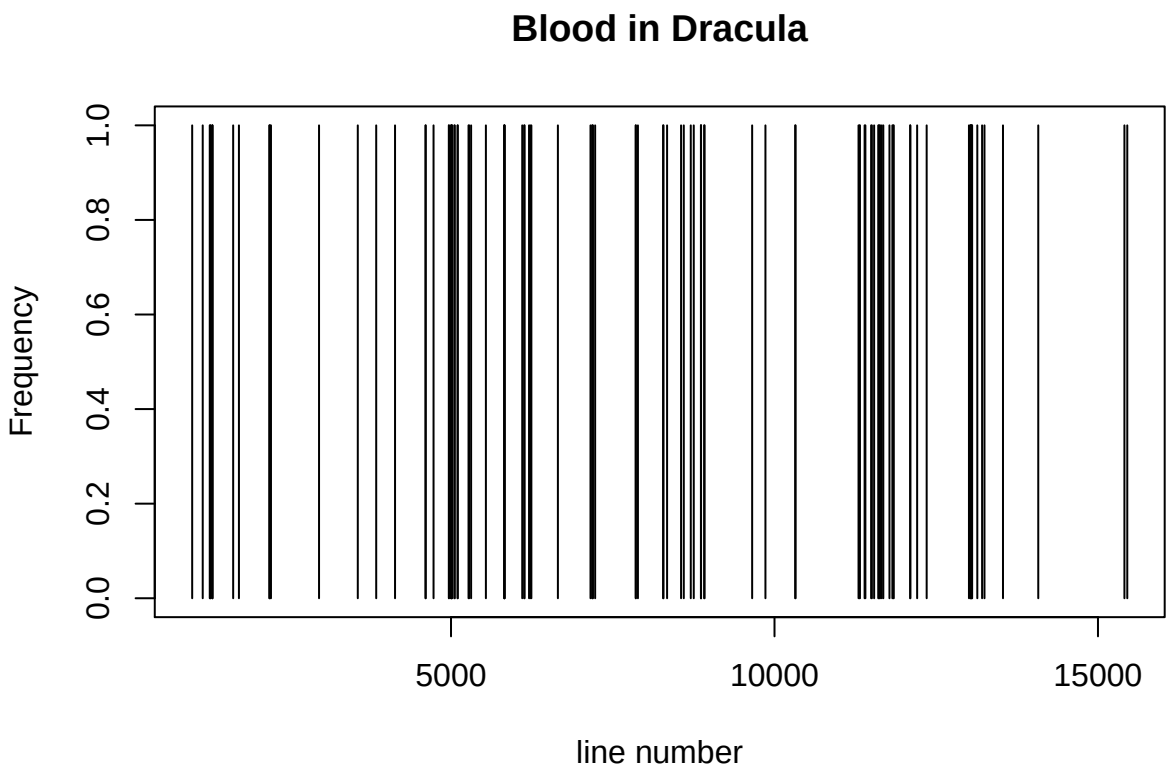
```
sum(str_count(text, "\\b[bB]lood\\b"))
```

```
## [1] 113
```

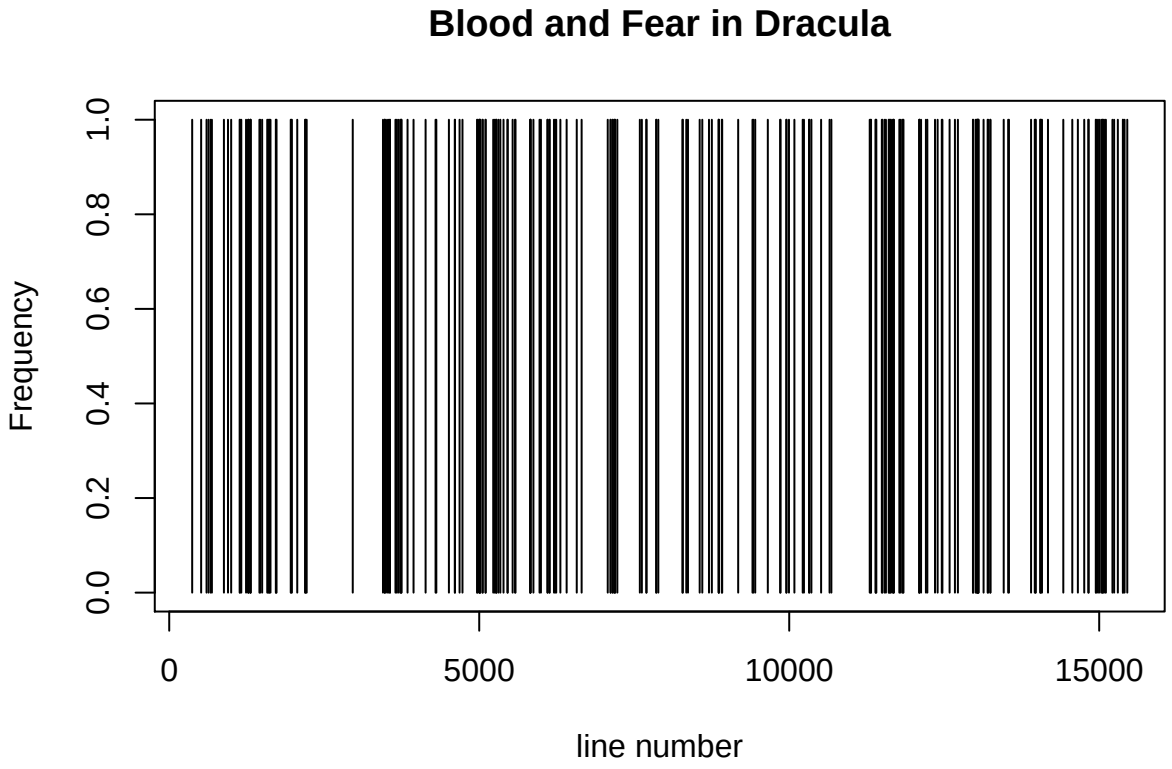
```
grep("\\b[bB]lood\\b", text)
```

```
##   [1]   999  1162  1271  1279  1282  1309  1311  1314  1633  1722  2192
##  [12]  2196  2204  2217  2960  3559  3844  4135  4606  4609  4730  4967
##  [23]  4968  4996  5008  5010  5011  5015  5051  5058  5102  5104  5271
##  [34]  5281  5311  5539  5823  5825  5826  5828  6101  6131  6137  6206
##  [45]  6224  6242  6652  7156  7182  7186  7194  7228  7855  7857  7889
##  [56]  8278  8284  8340  8556  8598  8706  8752  8863  8864  8865  8913
##  [67]  8919  9655  9860 10320 10326 11299 11315 11320 11393 11405 11494
##  [78] 11505 11540 11605 11612 11628 11630 11658 11682 11777 11822 11835
##  [89] 11843 12098 12099 12205 12352 13006 13027 13028 13035 13040 13045
## [100] 13050 13055 13135 13209 13246 13532 14077 15409 15453
```

```
hist(grep("\\b[bB]lood\\b", text),n=100000, main="Blood in Dracula", xlab="line number")
box()
```




```
hist(grep("\\b[bB]lood\\b|\\b[fF]ear\\b", text),n=100000, main="Blood and Fear in Dracula", xlab="line n
box()
```




b) read in pg345.txt

- use `paste(..., collapse="\n")` to combine the text into one single string
- use str_split() to split this string into words
- use tabulate() and sort to get the 10 most frequent words as well as the 10 least frequent words

c) for all files: 17814-0.txt, pg11.txt, pg1661.txt, pg174.txt, pg2600.txt, pg345.txt, pg34901.txt

- find a way to extract:
  - title
  - author
  - posting date
- find a way to drop information added by Projct Gutenberg

```
txt <- readLines("pg11.txt")
txt[1:10]
```

```
##  [1] "Project Gutenberg's Alice's Adventures in Wonderland, by Lewis Carroll"
##  [2] ""
##  [3] "This eBook is for the use of anyone anywhere at no cost and with"
##  [4] "almost no restrictions whatsoever.  You may copy it, give it away or"
##  [5] "re-use it under the terms of the Project Gutenberg License included"
##  [6] "with this eBook or online at www.gutenberg.org"
##  [7] ""
##  [8] ""
##  [9] "Title: Alice's Adventures in Wonderland"
## [10] ""
```

```
grep("^Title", txt, value = TRUE)
```

```
## [1] "Title: Alice's Adventures in Wonderland"
```

```
textfiles <- c("17814-0.txt", "pg11.txt", "pg1661.txt", "pg174.txt", "pg2600.txt", "pg345.txt")
TXT <- list()
for( i in seq_along(textfiles) ){
  TXT[[i]] <- readLines(textfiles[i])
}

grep("^Title", TXT[[1]], value = TRUE)[1]
```

```
## [1] "Title: Lysistrata"
```

```
get_gutenberg_title <- function(fname){
  txt <- readLines(fname)
  tmp <- grep("^Title", txt, value = TRUE)[1]
  str_replace(tmp, "^Title: ", "")
}

get_gutenberg_author <- function(fname){
  txt <- readLines(fname)
  tmp <- grep("by", txt, value = TRUE)[1]
  str_replace(tmp, "^.*by ","")
}
```

```r
get_gutenberg_posting_date <- function(fname){
  txt <- readLines(fname)
  tmp <- grep("posting date|release date", txt, value = TRUE, ignore.case = TRUE)[1]
  str_replace_all(tmp, "^.*: | \\[.*$","")
}

lapply(textfiles, get_gutenberg_title)
```

```
## [[1]]
## [1] "Lysistrata"
##
## [[2]]
## [1] "Alice's Adventures in Wonderland"
##
## [[3]]
## [1] "The Adventures of Sherlock Holmes"
##
## [[4]]
## [1] "The Picture of Dorian Gray"
##
## [[5]]
## [1] "War and Peace"
##
## [[6]]
## [1] "Dracula"
```

```r
lapply(textfiles, get_gutenberg_author)
```

```
## [[1]]
## [1] "Aristophanes"
##
## [[2]]
## [1] "Lewis Carroll"
##
## [[3]]
## [1] "Arthur Conan Doyle"
##
## [[4]]
## [1] "Oscar Wilde"
##
## [[5]]
## [1] "Leo Tolstoy"
##
## [[6]]
## [1] "Bram Stoker"
```

```r
lapply(textfiles, get_gutenberg_posting_date)
```

```
## [[1]]
## [1] "February 21, 2006"
##
```

```
## [[2]]
## [1] "June 25, 2008"
##
## [[3]]
## [1] "April 18, 2011"
##
## [[4]]
## [1] "June 9, 2008"
##
## [[5]]
## [1] "January 10, 2009"
##
## [[6]]
## [1] "August 16, 2013"
```

d) read in 2012.txt and build a data.frame containing the following information

- name
- number of reviews
- institution

```
txt <- readLines("2012.txt", warn = FALSE)[-c(1:52)]
txt <- paste0(txt, collapse="")

txt <-
  str_split(txt, "\\)") %>%
  unlist()

txt <-
  txt %>%
  str_replace_all("\t|  \f"," ") %>%
  str_trim() %>%
  str_replace_all("  ","")

name <-
  txt %>%
  str_extract("^.*\\.") %>%
  str_replace("\\.","")

reviews <-
  txt %>%
  str_extract("\\d") %>%
  as.numeric()

institution <-
  txt %>%
  str_extract("\\..*\\(") %>%
  str_replace_all("\\.|\\(","") %>%
  str_trim()
```