Actividad Integradora

Bruno Yánez, Javier Lizárraga, Maximiliano Martínez, Pedro Escoboza

```
rm(list=ls());
options(stringAsFactors = FALSE);
```

```
library("gplots"); # heatmap.2()
```

```
##
## Attaching package: 'gplots'
```

```
## The following object is masked from 'package:stats':
##
##     lowess
```

```
library("NLP")
library("RISmed");
library("tm");
```

```
# Función para cálculo de diferencia de prueba t student.
t_student_diff <- function(df, index_list_a, index_list_b, col_names =  c("Tumor", "Normal", "Diff")) {
  res <- t(apply(df, 1,
                 function(x) {
                   m_1 <- mean(x[index_list_a], na.rm = TRUE);
                   m_2 <- mean(x[index_list_b], na.rm = TRUE);
                   m_diff <- abs(m_1 - m_2);
                   c(m_1, m_2, m_diff);
                 }));
  colnames(res) <- col_names;
  return(res);
};

# Función para cálculo de diferencia de prueba t student para dataframes con esquema de clases.
t_student_classes <- function(df, classes, cr_a, cr_b,
                              col_names = c("A", "B", "p_value", "fold_change")){
  samples_a <- which(classes == cr_a);
  samples_b <- which(classes == cr_b);
  t_res <- t(apply(df, 1,
                   function(x){
                     t_test <- t.test(x[samples_a], x[samples_b]);
                     c(t_test$estimate[1], t_test$estimate[2], t_test$p.value, t_test$estimate[1] - t_te
                   }));
  colnames(t_res) <- col_names;
  return(t_res);
};

# Regresa un dataframe con los primeros n resultados ordenados por la columna col.
get_top_n <- function(df, col, n, decreasing = FALSE){
  return(head(df[order(col, decreasing=decreasing),],n));
};

# Normalización de datos.
normalize <- function(x, min, max){
  return((x-min)/(max-min));
};

# División de datos en grupos por rangos de valores.
freq_groups <- function(vec, bounds){
  num_bounds <- length(bounds);
  freqs <- integer(num_bounds);
  for (i in 2:num_bounds){
    for (j in 1:length(vec)){
      if (vec[j] >= bounds[i-1] & vec[j] < bounds[i]){
        freqs[i] = freqs[i] + 1;
      }
    }
  }
  return(freqs);
};
```

Análisis de Multi__Cancer__Data

```
load("Multi_Cancer_Data.Rdata");
df <- multi_cancer_data;
rm(multi_cancer_data);
```

Diferenia entre muestras normales y muestras de cáncer color

```
# Selección de muestras normales.
normal_samples_indexes <- grep("Normal", colnames(df));
print(normal_samples_indexes);
```

```
##  [1] 191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209
## [20] 210 211 212 213 214 215 216 217 218 219 220 221 222 223 224 225 226 227 228
## [39] 229 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247
## [58] 248 249 250 251 252 253 254 255 256 257 258 259 260 261 262 263 264 265 266
## [77] 267 268 269 270 271 272 273 274 275 276 277 278 279 280
```

```
# Selección de muestras de cáncer colorrectal.
colorectal_cancer_indexes <- grep("Tumor__Colorectal", colnames(df));
print(colorectal_cancer_indexes);
```

```
##  [1] 33 34 35 36 37 38 39 40 41 42 43
```

```
# Prueba t student.
tstudent_normal_with_colorectal <- data.frame(t_student_diff(df, normal_samples_indexes, colorectal_canc

# Seleccionar 10 entradas con mayor diferencia.
tstudent_normal_with_colorectal <- get_top_n(tstudent_normal_with_colorectal, tstudent_normal_with_colo

print(tstudent_normal_with_colorectal);
```

```
##                                                                                    Tumor
## MMP12 Matrix metalloproteinase 12 (macrophage elastase)_L23808_at           -0.203500000
## CARCINOEMBRYONIC ANTIGEN PRECURSOR_M29540_at                                  0.036511111
## MMP1 Matrix metalloproteinase 1 (interstitial collagenase)_X54925_at        -0.143000000
## CDX1 Caudal type homeo box transcription factor 1_U51095_at                  0.078966667
## Transforming growth factor-beta induced gene product (BIGH3) mRNA_M77349_at -0.200255556
## TUMOR-ASSOCIATED ANTIGEN CO-029_M35252_at                                    0.095433333
## Homeobox protein Cdx2 mRNA_U51096_at                                        -0.293233333
## Gamma-glutamyl hydrolase (hGH) mRNA_U55206_at                               -0.089533333
## GC-Box binding protein BTEB2_D14520_at                                       0.007233333
## NF-E2-related factor 3_RC_AA132523_at                                       -0.088766667
##                                                                               Normal
## MMP12 Matrix metalloproteinase 12 (macrophage elastase)_L23808_at           2.307545
## CARCINOEMBRYONIC ANTIGEN PRECURSOR_M29540_at                                2.538545
## MMP1 Matrix metalloproteinase 1 (interstitial collagenase)_X54925_at        2.088818
## CDX1 Caudal type homeo box transcription factor 1_U51095_at                 2.144182
## Transforming growth factor-beta induced gene product (BIGH3) mRNA_M77349_at 1.732727
## TUMOR-ASSOCIATED ANTIGEN CO-029_M35252_at                                   2.008545
## Homeobox protein Cdx2 mRNA_U51096_at                                        1.581818
## Gamma-glutamyl hydrolase (hGH) mRNA_U55206_at                               1.710818
## GC-Box binding protein BTEB2_D14520_at                                      1.775000
## NF-E2-related factor 3_RC_AA132523_at                                       1.659273
##                                                                                 Diff
## MMP12 Matrix metalloproteinase 12 (macrophage elastase)_L23808_at           2.511045
## CARCINOEMBRYONIC ANTIGEN PRECURSOR_M29540_at                                2.502034
## MMP1 Matrix metalloproteinase 1 (interstitial collagenase)_X54925_at        2.231818
## CDX1 Caudal type homeo box transcription factor 1_U51095_at                 2.065215
## Transforming growth factor-beta induced gene product (BIGH3) mRNA_M77349_at 1.932983
## TUMOR-ASSOCIATED ANTIGEN CO-029_M35252_at                                   1.913112
## Homeobox protein Cdx2 mRNA_U51096_at                                        1.875052
## Gamma-glutamyl hydrolase (hGH) mRNA_U55206_at                               1.800352
## GC-Box binding protein BTEB2_D14520_at                                      1.767767
## NF-E2-related factor 3_RC_AA132523_at                                       1.748039
```

Análisis de TCGA_COADREAD_comp_data

```r
rm(list=setdiff(ls(), lsf.str()));
load("TCGA_COADREAD_comp_data.RData");
df <- tcga_coadread;
rm(tcga_coadread);
```

Diferencia entre jóvenes y adultos

```r
# Prueba de t student para TCGA COARDREAD por las clases Young  y Old.
tcga_t_test <- t_student_classes(df,tcga_coadread_class,"Young","Old",c("Young", "Old", "p_value", "Fol

# Filtración de datos para eliminar entradas no significativas.
tcga_t_test_filter <- apply(tcga_t_test[,1:2],1,function(x){all(x<1)});
tcga_t_test <- tcga_t_test[-which(tcga_t_test_filter),];

# Ordenar por diferencia.
tcga_t_test <- tcga_t_test[order(tcga_t_test[,4], decreasing=TRUE),];

# Genes con mayor diferencia de expresión entre jóvenes y ancianos.
```

```
print("# Genes con mayor diferencia de expresión entre jóvenes y ancianos:");
```

```
## [1] "# Genes con mayor diferencia de expresión entre jóvenes y ancianos:"
```

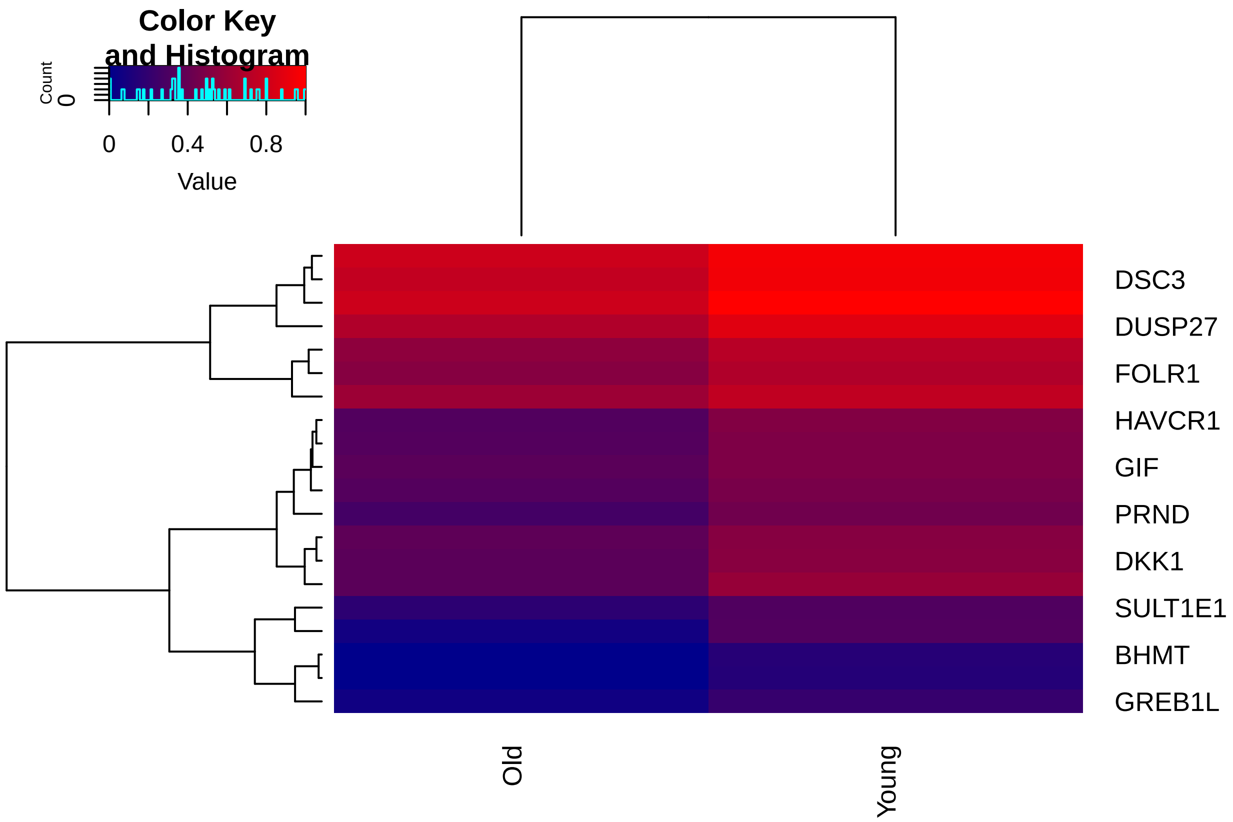
```
write.table(rownames(tcga_t_test[which(tcga_t_test[,4] > 0),])[1:20], sep='\t', quote=F, row.names=F, c
```

```
## GATA4
## PCSK1N
## XIST
## DUSP27
## HAVCR1
## DSC3
## DKK1
## PRND
## FOLR1
## CPS1
## GAL
## FZD9
## GLDC
## GREB1L
## SULT1E1
## BHMT
## GIF
## PEG10
## NKX2-1
## LEFTY2
```

```
# Generar matriz para mapa de calor.
hm_mat <- tcga_t_test[rownames(tcga_t_test)[1:20],];

# Remover columnas de p_value y fold_change.
hm_mat <- hm_mat[,-(3:4),drop=FALSE];
colnames(hm_mat) <- colnames(tcga_t_test)[1:2];

# Normalizar valores de expresión.
exp_values <- c(hm_mat[,1], hm_mat[,2]);
min_exp_values <- min(exp_values);
max_exp_values <- max(exp_values);

hm_mat[,1] <- normalize(hm_mat[,1], min_exp_values, max_exp_values);
hm_mat[,2] <- normalize(hm_mat[,2], min_exp_values, max_exp_values);


num_colors = 128;
```

Mapa de calor

```
# Construcción de mapa de calor.
colors_h <- colorRampPalette(c("darkblue","red"))(num_colors);
h_breaks <- seq(from=0, to=1, length=num_colors+1);

heatmap.2(hm_mat, col=colors_h, trace="none", breaks=h_breaks, cexCol=1);
```

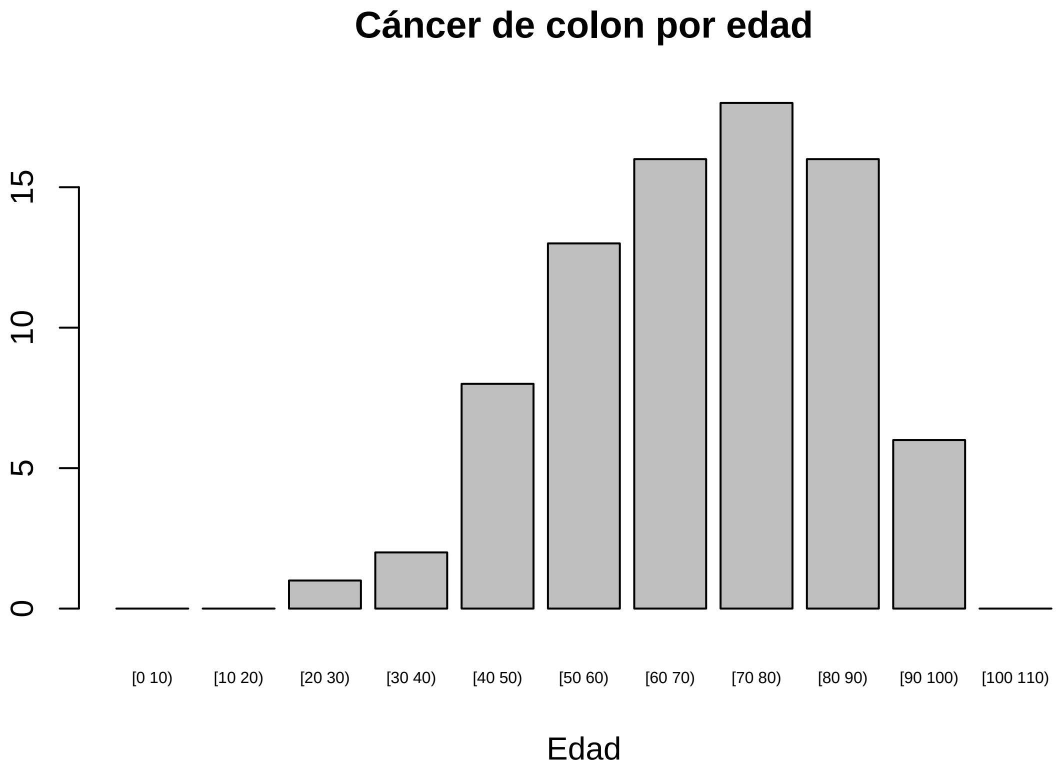


Análisis de 9_PACIENTES_DE_NUEVO_INGRESO.csv

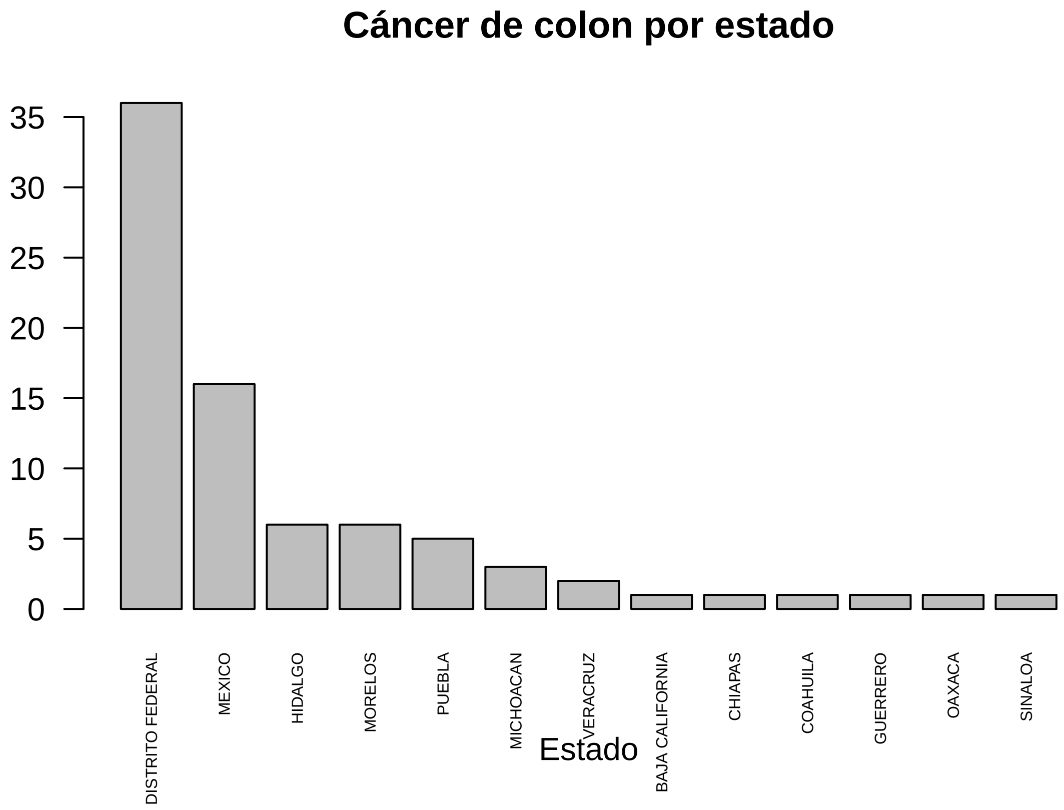
```
rm(list=setdiff(ls(), lsf.str()));
df <- read.csv("9_PACIENTES_DE_NUEVO_INGRESO.csv");

# Selección de entradas de tumores de colon.
colon_cancer <- df[grep("COLON", df$DESCRIPCION.DIAGNOSTICO),];
print(head(colon_cancer));
```

```
##     FOLIO EDAD      SEXO           ESTADO  MUNICIPIO DESCRIPCION.DIAGNOSTICO
## 85     85   76 Masculino          MORELOS XOCHITEPEC TUMOR MALIGNO DEL COLON
## 108   108   72 Masculino          HIDALGO    ACATLAN TUMOR MALIGNO DEL COLON
## 132   132   49  Femenino           MEXICO     CHALCO TUMOR MALIGNO DEL COLON
## 140   140   68  Femenino DISTRITO FEDERAL    TLALPAN TUMOR MALIGNO DEL COLON
## 145   145   54 Masculino           MEXICO   TULTEPEC TUMOR MALIGNO DEL COLON
## 180   180   35 Masculino DISTRITO FEDERAL XOCHIMILCO TUMOR MALIGNO DEL COLON
```

```
# Cáncer de colon por edad.
ranges <- c(0,10,20,30,40,50,60,70,80,90,100);
age_freq <- freq_groups(colon_cancer$EDAD, ranges);
label <- c("[0 10)", "[10 20)","[20 30)","[30 40)","[40 50)","[50 60)", "[60 70)", "[70 80)","[80 90)",
barplot(age_freq,  main="Cáncer de colon por edad", xlab="Edad", names.arg=label, cex.names=0.5);
```



```r
# Cáncer de colon por estado.
state_freq <- as.data.frame(table(colon_cancer$ESTADO));
state_freq <- state_freq[which(state_freq$Freq != 0),];
state_freq <- state_freq[order(state_freq$Freq, decreasing=TRUE),];
barplot(state_freq$Freq, main="Cáncer de colon por estado", xlab="Estado", names.arg=state_freq$Var1, c
```



Búsqueda de artículos relacionados en PubMed

```
# Correr la opción paraque no se lean los strings como factores.
options(stringsAsFactors = F)

# Creamos un query para buscar artículos en PUBMED desde R. Usando los operadores
# lógicos AND y OR y la opción TitleAbstract.

query_colon <- "\"colon\"[TIAB] AND \"cancer\"[TIAB] AND \"young\"[TIAB] AND
(\"mutation\"[TIAB] OR \"alteration\"[TIAB] OR \"treatment\"[TIAB] OR
\"hereditary\"[TIAB])"
#usamos la opción EUtilsSummary de RISmed

search_query <- EUtilsSummary(query_colon)
summary(search_query)
```

```
## Query:
## "colon"[TIAB] AND "cancer"[TIAB] AND "young"[TIAB] AND ("mutation"[TIAB] OR "alteration"[TIAB] OR "t
##
## Result count:  331
```

```
#Después, obtenemos un data frame con el título, abstract y ID de los artículos.
records <- EUtilsGet(search_query)
pubmed_data <- data.frame('Title' = ArticleTitle(records), 'Abstract' =
AbstractText(records), 'PID' = ArticleId(records))
pubmed_data [1:3,c("Title","PID")]
```

```
##                                                                                     Title
## 1                                                   Colorectal cancer statistics, 2020.
## 2                                      Colon Cancer: A Clinician's Perspective in 2019.
## 3 Effect of vitamin B17 on experimentally induced colon cancer in adult male albino rat.
##        PID
## 1 32133645
## 2 32095167
## 3 32073131
```

```
#Quitamos caracteres (. : , ; [ ]) del título y el abstract.
pubmed_data$Title <- gsub(pattern = "//.|:|,|;|//[|//]", replacement = "",
pubmed_data$Title)
pubmed_data$Abstract <- gsub(pattern = "//.|:|,|;|//[|//]", replacement = "",
pubmed_data$Abstract)

#Convertimos todo a minúsculas
pubmed_data$Title <- tolower(pubmed_data$Title)
pubmed_data$Abstract <- tolower(pubmed_data$Abstract)
pubmed_data$Title[1:3]
```

```
## [1] "colorectal cancer statistics 2020."
## [2] "colon cancer a clinician's perspective in 2019."
## [3] "effect of vitamin b17 on experimentally induced colon cancer in adult male albino rat."
```

```
#usamos la función strsplit y unlist para obtener las palabras contenidas en el abstract.
unlist(strsplit(pubmed_data$Abstract[1], " "))[1:10]
```

```
##  [1] "colorectal" "cancer"     "(crc)"      "is"         "the"
##  [6] "second"     "most"       "common"     "cause"      "of"
```

```
#Hay algunos artículos que pudieran no incluir el abstract
which(pubmed_data$Abstract == "")
```

```
##  [1]  11  18  31  35  37  41  42  46  62  65  91  99 150
```

```
#Creamos el vector sobre el cuál vamos a iterar:
word_list <- c()

#El bucle para todos los abstracts
for (i in 1:length(pubmed_data$Abstract)){
  #Obtenemos las palabras como vector
  aux_word <- unlist(strsplit(pubmed_data$Abstract[i], " "))

  #eliminamos abstracts vacíos con la condicionante "if"
  if (length(aux_word) > 0){
    #Concatenamos las palabras y el ID. Con c bind recuperamos en una columna los IDs en
    # donde se encuentran las palabras y los concatenamos con la columna aux_word.
    aux_list <- cbind(pubmed_data$PID[i], aux_word)
    #Pegamos este data frame en el vector inicial con row bind.
    word_list <- rbind(word_list, aux_list)

}

}

colnames(word_list) <- c("PID","Word")
ncol(word_list)
```

```
## [1] 2
```

```r
nrow(word_list)
```

```
## [1] 81936
```

```r
dim(word_list)
```

```
## [1] 81936     2
```

```r
word_list[1:5,]
```

```
##      PID        Word
## [1,] "32133645" "colorectal"
## [2,] "32133645" "cancer"
## [3,] "32133645" "(crc)"
## [4,] "32133645" "is"
## [5,] "32133645" "the"
```

```r
head(word_list)
```

```
##      PID        Word
## [1,] "32133645" "colorectal"
## [2,] "32133645" "cancer"
## [3,] "32133645" "(crc)"
## [4,] "32133645" "is"
## [5,] "32133645" "the"
## [6,] "32133645" "second"
```

```r
#Usamos la libreria tm para obtener la lista de "stopwords(palabras vacías)" (articulos,
#adverbios, pronombres, conjunciones)
library(tm)
stop_words <- stopwords(kind = "en")
stop_words
```

```
##   [1] "i"          "me"         "my"         "myself"     "we"
##   [6] "our"        "ours"       "ourselves"  "you"        "your"
##  [11] "yours"      "yourself"   "yourselves" "he"         "him"
##  [16] "his"        "himself"    "she"        "her"        "hers"
##  [21] "herself"    "it"         "its"        "itself"     "they"
##  [26] "them"       "their"      "theirs"     "themselves" "what"
##  [31] "which"      "who"        "whom"       "this"       "that"
##  [36] "these"      "those"      "am"         "is"         "are"
##  [41] "was"        "were"       "be"         "been"       "being"
##  [46] "have"       "has"        "had"        "having"     "do"
##  [51] "does"       "did"        "doing"      "would"      "should"
##  [56] "could"      "ought"      "i'm"        "you're"     "he's"
##  [61] "she's"      "it's"       "we're"      "they're"    "i've"
##  [66] "you've"     "we've"      "they've"    "i'd"        "you'd"
##  [71] "he'd"       "she'd"      "we'd"       "they'd"     "i'll"
##  [76] "you'll"     "he'll"      "she'll"     "we'll"      "they'll"
##  [81] "isn't"      "aren't"     "wasn't"     "weren't"    "hasn't"
##  [86] "haven't"    "hadn't"     "doesn't"    "don't"      "didn't"
##  [91] "won't"      "wouldn't"   "shan't"     "shouldn't"  "can't"
##  [96] "cannot"     "couldn't"   "mustn't"    "let's"      "that's"
## [101] "who's"      "what's"     "here's"     "there's"    "when's"
## [106] "where's"    "why's"      "how's"      "a"          "an"
## [111] "the"        "and"        "but"        "if"         "or"
## [116] "because"     "as"          "until"       "while"       "of"
## [121] "at"          "by"          "for"         "with"        "about"
## [126] "against"     "between"     "into"        "through"     "during"
## [131] "before"      "after"       "above"       "below"       "to"
## [136] "from"        "up"          "down"        "in"          "out"
## [141] "on"          "off"         "over"        "under"       "again"
## [146] "further"     "then"        "once"        "here"        "there"
## [151] "when"        "where"       "why"         "how"         "all"
## [156] "any"         "both"        "each"        "few"         "more"
## [161] "most"        "other"       "some"        "such"        "no"
## [166] "nor"         "not"         "only"        "own"         "same"
## [171] "so"          "than"        "too"         "very"
```

```r
#guardamos los índices de las palabras de nuestra lista que corresponden a stopwords y
#que deben ser removidas
index_stop_word <- which(word_list[,2] %in% stop_words)
length(index_stop_word)
```

```
## [1] 28444
```

```r
dim(word_list)
```

```
## [1] 81936     2
```

```r
word_list <- word_list[-index_stop_word,]
dim(word_list)
```

```
## [1] 53492     2
```

```r
head(word_list)
```

```
##      PID        Word
## [1,] "32133645" "colorectal"
## [2,] "32133645" "cancer"
## [3,] "32133645" "(crc)"
## [4,] "32133645" "second"
## [5,] "32133645" "common"
## [6,] "32133645" "cause"
```

```r
#Ahora podemos ver el top10 de las palabras mas frecuentes
sort(table(word_list[,2]), decreasing = T) [1:10]
```

```
##
##     cancer   patients      colon colorectal      young        age      years
##       1133       1037        619        465        462        401        268
##  treatment       risk      study
##        251        233        209
```

```r
word_df <- data.frame(PID=as.numeric(word_list[,1]), Word=word_list[,2],
PIDWord=as.character(apply(word_list, 1, paste, collapse="_")))
word_df[1:5,]
```

```
##        PID       Word             PIDWord
## 1 32133645 colorectal 32133645_colorectal
## 2 32133645     cancer     32133645_cancer
## 3 32133645      (crc)      32133645_(crc)
## 4 32133645     second     32133645_second
## 5 32133645     common     32133645_common
```

```r
dup_index <- duplicated(word_df$PIDWord)
word_df$PIDWord[1:30]
```

```
##  [1] "32133645_colorectal"       "32133645_cancer"
##  [3] "32133645_(crc)"            "32133645_second"
##  [5] "32133645_common"           "32133645_cause"
##  [7] "32133645_cancer"           "32133645_death"
##  [9] "32133645_united"           "32133645_states."
## [11] "32133645_every"            "32133645_3 years"
## [13] "32133645_american"         "32133645_cancer"
## [15] "32133645_society"          "32133645_provides"
## [17] "32133645_update"           "32133645_crc"
## [19] "32133645_occurrence"       "32133645_based"
## [21] "32133645_incidence"        "32133645_data"
## [23] "32133645_(available"       "32133645_2016)"
## [25] "32133645_population-based" "32133645_cancer"
## [27] "32133645_registries"       "32133645_mortality"
## [29] "32133645_data"             "32133645_(through"
```

```r
length(which(dup_index))
```

```
## [1] 17321
```

```r
dim(word_df)
```

```
## [1] 53492     3
```

```r
word_df <- word_df[-which(dup_index),]
dim(word_df)
```

```
## [1] 36171     3
```

```r
#volvemos a ver el top de las palabras mas frecuentes
sort(table(word_df[,2]), decreasing = T) [1:5]
```

```
## 
##     cancer      colon      young   patients colorectal 
##        282        278        278        217        178
```

```r
#ordenamos el data frame por ID en orden decreciente para tener los artículos más
#recientes
word_df <- word_df[order(word_df$PID, decreasing=T),]
print(word_df[1:40,]);
```

```
##         PID             Word                   PIDWord
## 1  32133645       colorectal       32133645_colorectal
## 2  32133645           cancer           32133645_cancer
## 3  32133645            (crc)            32133645_(crc)
## 4  32133645           second           32133645_second
## 5  32133645           common           32133645_common
## 6  32133645            cause            32133645_cause
## 8  32133645            death            32133645_death
## 9  32133645           united           32133645_united
## 10 32133645          states.          32133645_states.
## 11 32133645            every            32133645_every
## 12 32133645          3 years          32133645_3 years
## 13 32133645         american         32133645_american
## 15 32133645          society          32133645_society
## 16 32133645         provides         32133645_provides
## 17 32133645           update           32133645_update
## 18 32133645              crc              32133645_crc
## 19 32133645       occurrence       32133645_occurrence
## 20 32133645            based            32133645_based
## 21 32133645        incidence        32133645_incidence
## 22 32133645             data             32133645_data
## 23 32133645       (available       32133645_(available
## 24 32133645            2016)            32133645_2016)
## 25 32133645 population-based 32133645_population-based
## 27 32133645       registries       32133645_registries
## 28 32133645        mortality        32133645_mortality
## 30 32133645         (through         32133645_(through
## 31 32133645            2017)            32133645_2017)
## 32 32133645         national         32133645_national
## 33 32133645           center           32133645_center
## 34 32133645           health           32133645_health
## 35 32133645      statistics.      32133645_statistics.
## 36 32133645             2020             32133645_2020
## 37 32133645    approximately    32133645_approximately
## 38 32133645           147950           32133645_147950
## 39 32133645      individuals      32133645_individuals
## 40 32133645             will             32133645_will
## 41 32133645        diagnosed        32133645_diagnosed
## 43 32133645            53200            32133645_53200
## 45 32133645              die              32133645_die
## 46 32133645          disease          32133645_disease
```