Dark Market

Simon Delecourt & Edouard Donze

Contents

```
#-------------------------------------------------------------
#                        Library :
#-------------------------------------------------------------

#install.packages("stringr")
#install.packages("units")

library(stringr)
library(units)
```

```
#-------------------------------------------------------------
#               Importation of the data :
#-------------------------------------------------------------

data <- as.data.frame(read.csv("C:/BDP/Doc/alphaspider.csv"))
```

```
#-------------------------------------------------------------
#                       Cleaning :
#-------------------------------------------------------------

for(i in 2:4)
    { data[,i] <- iconv(data[,i], from="UTF-8", to="latin9", sub=" ")   # conversion UTF in ISO/IEC 885
      data[,i] <- gsub(pattern="<.*?>|\n", replacement=" ", data[,i])   # HTML tags and \n
      data[,i] <- tolower(data[,i])                                     # put in lowercase
      data[,i] <- gsub(pattern="\\s{2,}", replacement=" ", data[,i])    # remove spaces
    }
```

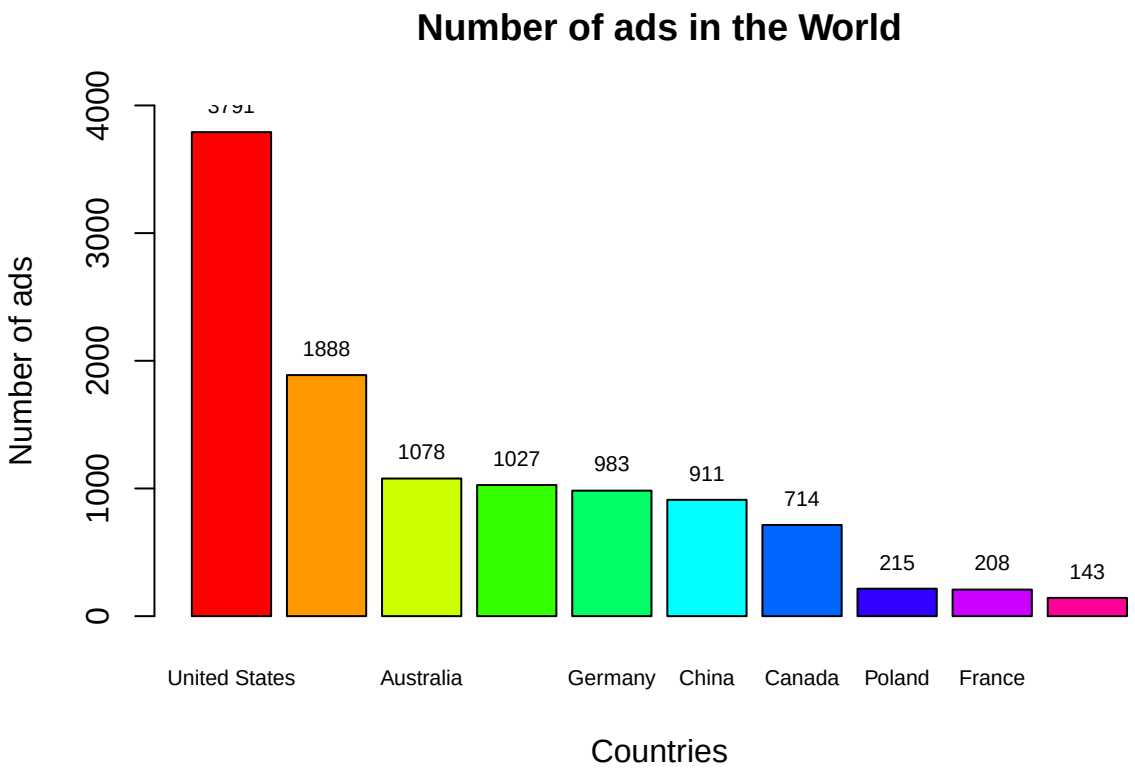
```
#-------------------------------------------------------------
#      Making data readable in a computering way :
#-------------------------------------------------------------

oz_conversion <- 28.3495

#----------------------------
#  Handling : Dose and unit
#----------------------------

  # 1- Extraction of characters matching with the dose and unit in the title

# Vector with all the unit that are allowed (add unit if needed)
unit_allowed <- c("mg","kg", "ug","lb","oz","ounce","g\\s","gr","gram")

# Construct a regular expression matching with digits + units allowed
regex_unit <- str_c("([0-9]+\\.?[0-9]*)((?:(\\s|)((",unit_allowed[1],")")
for(i in 2:length(unit_allowed)){
  regex_unit <- str_c(regex_unit,"|(",unit_allowed[i],")")
}
```

```
regex_unit <- str_c(regex_unit,")))")
# regex_unit = regular expression for dose and unit

# Extraction from the title :
dose_unit <- str_extract(data$title, regex_unit)

  # 2- Spliting the value and the unit

# Construct a regular expression
regex_extrac_unit <- str_c("(.*?)(",unit_allowed[1])
for(i in 1:length(unit_allowed)){
  regex_extrac_unit <- str_c(regex_extrac_unit,"|",unit_allowed[i])
}
regex_extrac_unit <- str_c(regex_extrac_unit,")")

# Spliting thanks to the regular expresion (regex_extrac_unit)
dose_unit <- str_match(dose_unit, regex_extrac_unit)

# amelioration of the string (removing blank)
dose_unit <- trimws(dose_unit)

  # 3- Conversion of units in SI (in order to use a library)

# Vector of conversion : first element of the vector is unit in SI, other elements are non standard uni
# Add your vector if needed
g <- c("g","gr","gram")
oz <- c("oz","ounce")

for(i in 2 : length(g)){
  dose_unit[,3] <- gsub(pattern=g[i], replacement=g[1],dose_unit[,3])
}
for(i in 2 : length(oz)){
  dose_unit[,3] <- gsub(pattern=oz[i], replacement=oz[1],dose_unit[,3])
}
#add loop for your vector if needed

  # 4- Insertion in the data frame

data$dose <- as.numeric(dose_unit[,2])    # Numerical conversion
data$unit <- dose_unit[,3]

  # 5- Conversion to SI units : 1g and 1l

for(i in 1:length(data$unit)) {
    if(!(is.na(data[i,"unit"]))) {
      if ((str_detect(data[i,"unit"],"g") | (str_detect(data[i,"unit"],"lb")))) {
        value <- set_units(data[i,"dose"],with(ud_units,data[i,"unit"]))
        data[i,"dose"] <- as.units(value, with(ud_units, g))
        data[i,"unit"] <- "g"
      }
      else if (str_detect(data[i,"unit"],"l"))  {
        value <- set_units(data[i,"dose"],with(ud_units,data[i,"unit"]))
        data[i,"dose"] <- as.units(value, with(ud_units, l))
```

```
        data[i,"unit"] <- "l"
      }
      else if (str_detect(data[i,"unit"],"oz")) {
        data[i,"dose"] <- data[i,"dose"] * oz_conversion
        data[i,"unit"] <- "g"
      }
    }
}

#-----------------------
#  Handling : Quantity
#-----------------------

  # 1- Extraction of characters matching with the quantity in the title

# (ex : 20 packs, 20x, x20, 20 tabs)
# add key words here if needed
key_words_quantity <- c("x","pack", "tab", "pill", "pcs", "piece")

# Particular treatment for "x" because it can be 20x or x20"
regex_extract_quantity <- str_c("(",key_words_quantity[1],"(\\s|)(\\d+,?\\d+)|(\\d+,?\\d+)(?:([-\\s]|)(

for(i in 2 : length(key_words_quantity)){
  regex_extract_quantity <- str_c(regex_extract_quantity,"|",key_words_quantity[i])
}
regex_extract_quantity <- str_c(regex_extract_quantity,")))")

# Extraction from the title + insertion in the data frame :
data$quantity  <- str_extract(data$title,regex_extract_quantity)

# Keeping only digits
data$quantity  <- str_extract(data$quantity , "(\\d+,?\\d+)")

  # 2- Conversion in numerical element

# English numbers to Standard numbers (problem with the comma)
data$quantity <- gsub(pattern=",", replacement="", data$quantity)

# Conversion :
data$quantity <- as.numeric(data$quantity)


#-------------------------
#    Handling : Price
#-------------------------

  # 1- column price as numeric :

# Keeping only digits (without "USD")
data$price <- str_extract(data$price, "(\\d+,?\\.?\\d+)")

# English numbers to Standard numbers (problem with the comma)
data$price <- gsub(pattern=",", replacement="", data$price)
```

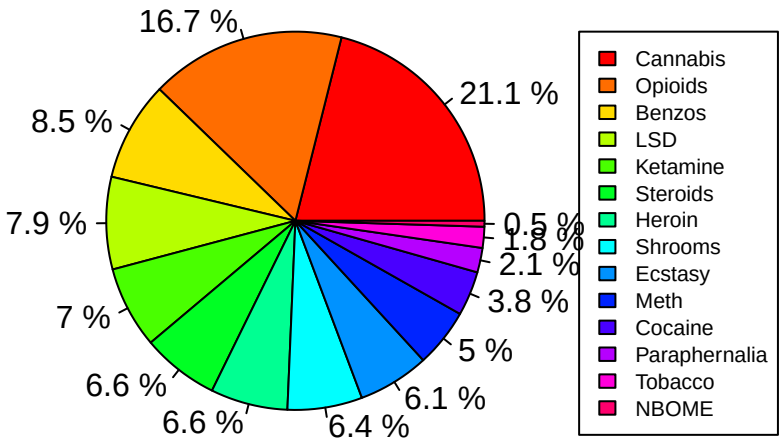
```r
# Conversion :
data$price <- as.numeric(data$price)

  # 2- Price per unit :

# Creation of a new vector with the price per unit
price_per_unit <- c()

for(i in 1:length(data$quantity)) {
  if(is.na(data[i,"quantity"])) {price_per_unit[i] <- data[i,"price"]}
  else {price_per_unit[i] <- data[i,"price"]/data[i,"quantity"]}
}

#Insertion in the data frame
data$priceUnit <- price_per_unit

  # 3- Price per unit per dose :

# Creation of a new vector with the price per unit per dose
price_unit_dose <- c()

for(i in 1:length(data$dose)) {
    if(is.na(data[i,"dose"])) {price_unit_dose[i] <- data[i,"priceUnit"]}
    else {price_unit_dose[i] <- data[i,"priceUnit"]/data[i,"dose"]}
}

#Insertion in the data frame
data$priceUnitDose <- price_unit_dose
```

```r
matching_vector <- c(!str_detect(data$origin, "Worldwide") & !str_detect(data$origin, "NULL"))

sumup <- sort(summary(data[matching_vector, "origin"]), decreasing=TRUE)


barp <- barplot(sumup[1:10], main="Number of ads in the World", xlab="Countries", ylab="Number of ads",y
barp <- text(x = barp, y = sumup[1:10], label = sumup[1:10], pos=3 , cex = 0.7, col= "black" )
```




```
#-----------------------------------------------
#      Distribution of Drugs in the market
#-----------------------------------------------

#----------------------------
#   The most common drugs
#----------------------------

selectDrug <- function(drugName){
  matching_vector <- c( (str_detect(data$category, drugName)))
  return(matching_vector)
}


drugs <- c("Cocaine", "Meth", "LSD", "Opioids", "Cannabis", "Steroids", "Ecstasy", "Ketamine", "Heroin"

freq <- c()
for(i in 1:length(drugs)){
  matching_vector <- selectDrug(drugs[i]);
  sumup<-summary(matching_vector)
  freq[i] <- sumup[3]
  #med[i] <- median((data[matching_vector, "priceUnitDose"]))
}

freq <- as.numeric(freq)#/length(rownames(data))
```

```r
res <- data.frame(drugs, freq)
res <- res[order(res$freq, decreasing = TRUE),]

# Calculation in percentage
piepercent<- round(100*res$freq/sum(res$freq), 1)
# round(a,1) : one digit after the comma

#---------------------------------------
#               PIE CHART
#---------------------------------------

  # 1- Labels :
  lab <- c()

  for(i in 1:length(piepercent)) {
    lab[i] <- paste(piepercent[[i]], "%", sep=" ")
  }

  # 2- Title :
  title <- "Distribution of drugs"

  # 3- Colors :
  c <- rainbow(length(piepercent))

  # 4- Plot :
  pie(piepercent,labels = lab, main = title ,col=c)

  # 5- Legend :
  legend(1.2,0.8,res$drugs, cex = 0.7, fill = c)
```


Distribution of drugs
16.7 %
21.1 %
8.5 %
7.9 %
0.5 %
1.8 %
2.1 %
3.8 %
5 %
7 %
6.6 %
6.1 %
6.6 %
6.4 %
Cannabis
Opioids
Benzos
LSD
Ketamine
Steroids
Heroin
Shrooms
Ecstasy
Meth
Cocaine
Paraphernalia
Tobacco
NBOME