Dark Market

Simon Delecourt & Edouard Donze

Contents

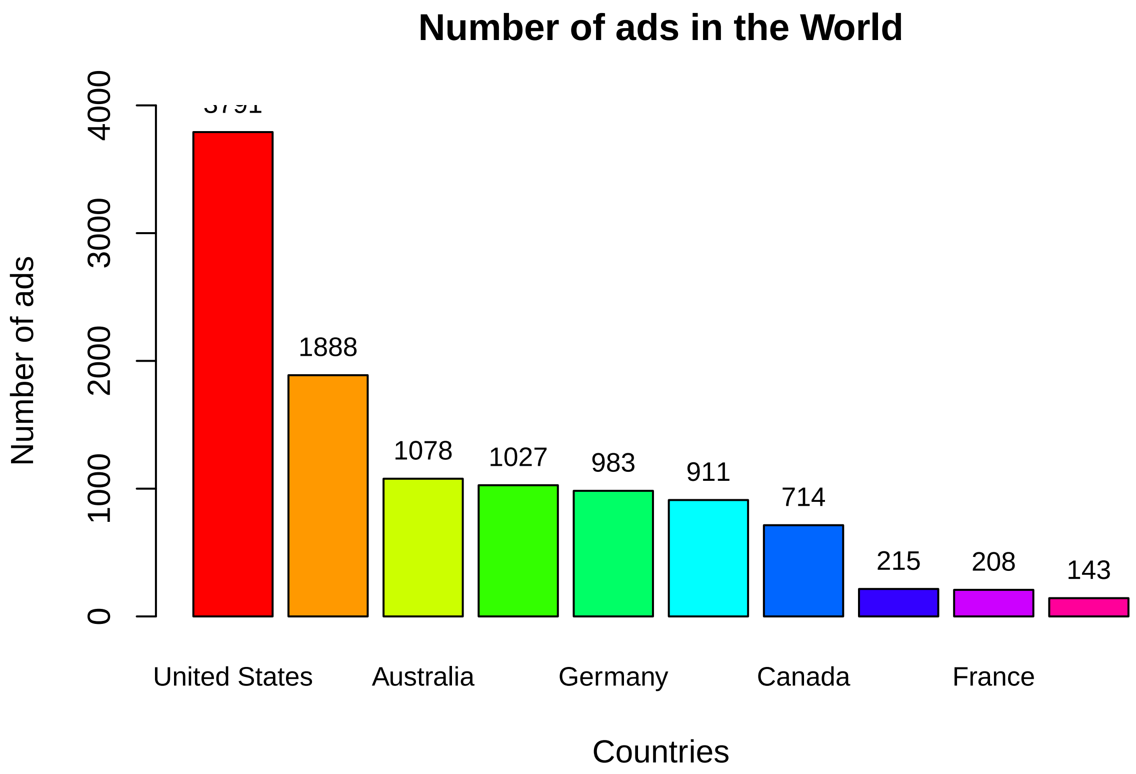
```
#-----------------------------------------------------------
#                       Library :
#-----------------------------------------------------------

#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 Function :
#-----------------------------------------------------------

cleaningData <- function(database) {
  for(i in 2:4)
    { database[,i] <- iconv(database[,i], from="UTF-8", to="latin9", sub=" ")
          # conversion UTF in ISO/IEC 8859-15
      database[,i] <- gsub(pattern="<.*?>|\n", replacement=" ", database[,i])
          # HTML tags and \n
      database[,i] <- tolower(database[,i])
          # put in lowercase
      database[,i] <- gsub(pattern="\\s{2,}", replacement=" ", database[,i])
          # remove spaces
    }

  return (database)

}

data <- cleaningData(database = data)

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

computerReadable <- function(database) {

  oz_conversion <- 28.3495

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

```r
#------------------------------

  # 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(database$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 u
# 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
```

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

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

for(i in 1:length(database$unit)) {
    if(!(is.na(database[i,"unit"]))) {
      if ((str_detect(database[i,"unit"],"g") | (str_detect(database[i,"unit"],"lb")))) {
        value <- set_units(database[i,"dose"],with(ud_units,database[i,"unit"]))
        database[i,"dose"] <- as.units(value, with(ud_units, g))
        database[i,"unit"] <- "g"
      }
      else if (str_detect(database[i,"unit"],"l"))  {
        value <- set_units(database[i,"dose"],with(ud_units,database[i,"unit"]))
        database[i,"dose"] <- as.units(value, with(ud_units, l))
        database[i,"unit"] <- "l"
      }
      else if (str_detect(database[i,"unit"],"oz")) {
        database[i,"dose"] <- database[i,"dose"] * oz_conversion
        database[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 :
database$quantity  <- str_extract(database$title,regex_extract_quantity)

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

  # 2- Conversion in numerical element

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

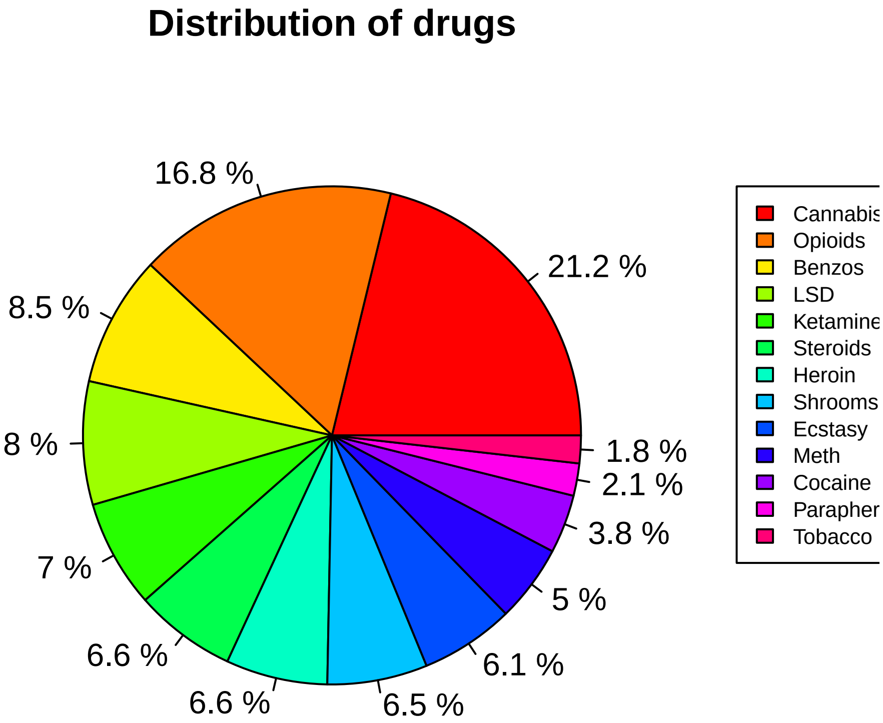
```
  # Conversion :
  database$quantity <- as.numeric(database$quantity)


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

    # 1- column price as numeric :

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

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

  # Conversion :
  database$price <- as.numeric(database$price)

    # 2- Price per unit :

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

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

  #Insertion in the data frame
  database$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(database$dose)) {
      if(is.na(database[i,"dose"])) {price_unit_dose[i] <- database[i,"priceUnit"]}
      else {price_unit_dose[i] <- database[i,"priceUnit"]/database[i,"dose"]}
  }

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


  return(database)
}

data <- computerReadable(database = data)

#-----------------------------------------------
#      Number of ads in the world
```

```
#-----------------------------------------------

#Get rid of unwanted orign like Worldwide and Null which are not relevant
matching_vector <- c(  !str_detect(data$origin, "Worldwide") & !str_detect(data$origin, "NULL"))

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

#Bar plot with the total number ofs ads in each country
barp <- barplot(sumup[1:10], main="Number of ads in the World", xlab="Countries", ylab="Number of ads",

barp <- text(x = barp, y = sumup[1:10], label = sumup[1:10], pos=3 , cex = 0.8, col= "black")
```

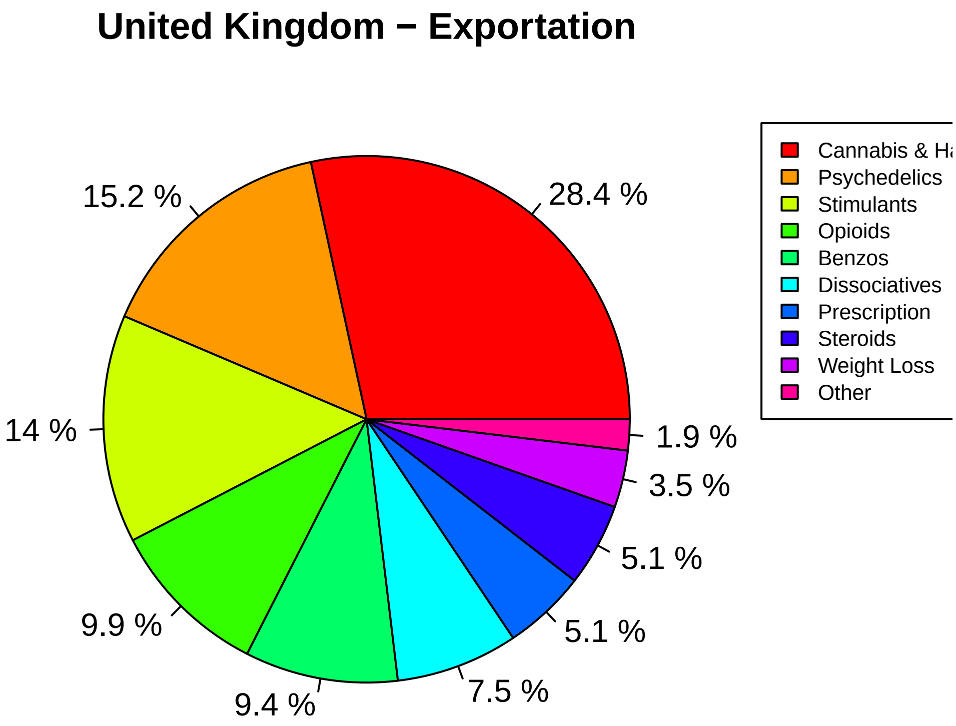


```
# 4- Margin :
par(mar=c(0,0,0,0))
#-----------------------------------------------
#      Distribution of Drugs in the market
#-----------------------------------------------

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

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

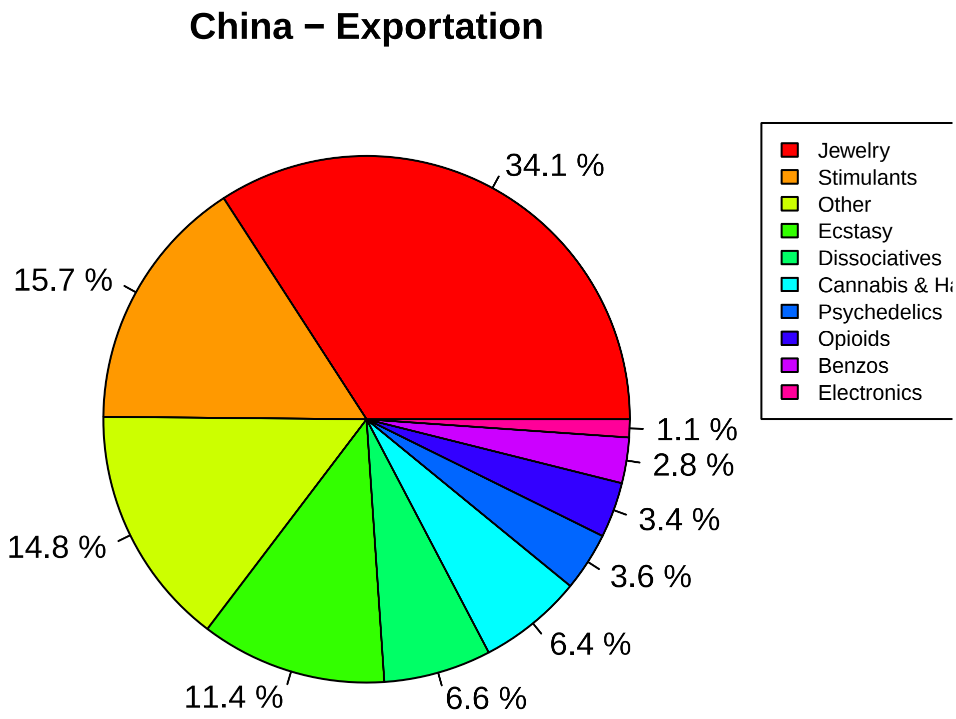
```
drugs <- c("Cocaine", "Meth", "LSD", "Opioids", "Cannabis", "Steroids", "Ecstasy", "Ketamine", "Heroin"

freq <- c()
for(i in 1:length(drugs)){
  matching_vector <- selectDrug(drugs[i]);
  sumup<-summary(matching_vector)
  freq[i] <- sumup[3]
}

freq <- as.numeric(freq)

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

#----------------------
#     Pie Chart
#----------------------

# 1- Labels :

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

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- Margin :
par(mar=c(1,4,4,0))

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

# 6- Legend :
legend(1.3,0.8,res$drugs, cex = 0.7, fill = c)
```



```
#-------------------------------------------------
#   Importation / Exportation of a country
#-------------------------------------------------

country_Import_Export <- function(country,num) {

  #-------------------
  #  Initialization
  #-------------------

  # Importation / Exportation :
  if (num == 0) {
    way <- "origin"
    txt <- "- Exportation"
  } else if (num == 1) {
    way <- "destination"
    txt <- "- Importation"
  }

  #------------------
  #    Analysis
  #------------------

  # Country as destination
  matching_vector <- str_detect(data[,way], country)
  # list of the categories (among the line that have "Country" as origin)
  # -> Products (categories) exporting by the country
  country_cat <- data[matching_vector,"category"]

  # Handling of this categories
  # Regular expression for spliting the categories
  regex <- "/(.*)/(.*)/(.*)"
  cat <- str_match(country_cat, regex)

  # Counting this categories
  tab <- table(cat[,3])   #cat[,3] : 2nd category
  tab <- sort(tab, decreasing = TRUE)  # Sorting (biggest in first)
  tab <- tab[1:10] # Taking only the most important


  #-----------------
  #    Pie Chart
  #-----------------

  # 1- Labels :

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

  lab <- c()

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

  # 2- Title :
  title <- paste(country, txt, sep=" ")

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

  # 4- Margin :
  par(mar=c(2,2,2,0))

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

  # 6- Legend :
  legend(1.2,0.9, names(piepercent), cex = 0.7, fill = c)

}

country_Import_Export("United Kingdom",0)
```



```
country_Import_Export("China",0)
```


China − Exportation
34.1 %
15.7 %
14.8 %
11.4 %
6.6 %
6.4 %
3.6 %
3.4 %
2.8 %
1.1 %
Jewelry
Stimulants
Other
Ecstasy
Dissociatives
Cannabis & Ha
Psychedelics
Opioids
Benzos
Electronics