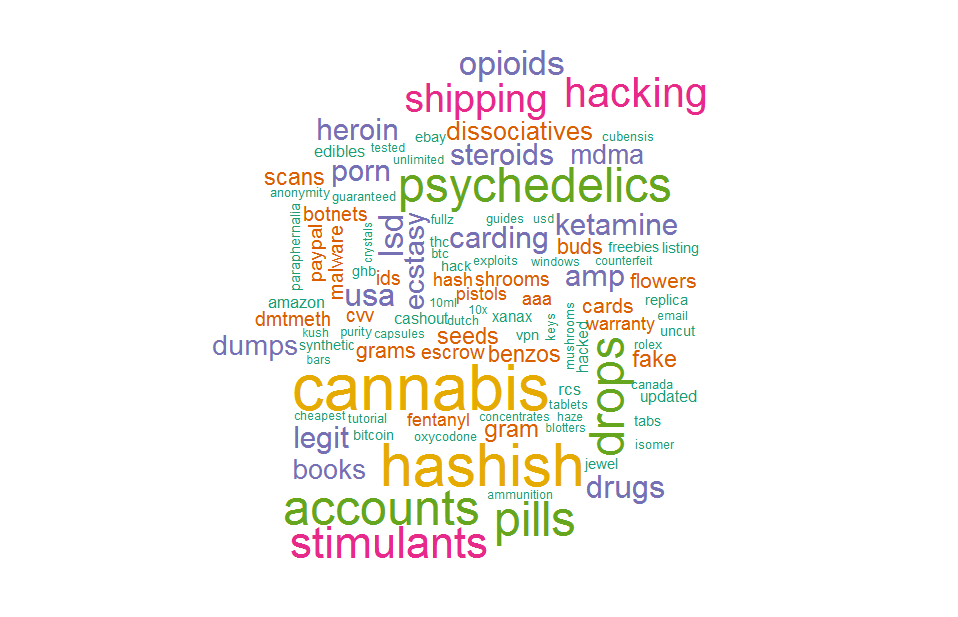
Data Mining - Dark Web Market

Simon Delecourt & Edouard Donze

Table of Contents



# - Introduction

For the past few years, a lot of *web market* has been developed and you can now easily buy items of any kinds. However, plenty of those marketplaces are growing up on *Dark Web*. Even if government attempts to fight against this kind of illegal market, new ones resurface or re-migrate frequently. On top of that, currently, we don't really know in detail how these websites operate.For more information on *Deep web* and *Dark Web marketplaces* please read the article: "Mining the Dark Web" [1]. You will also find in this paper an analysis of *Agora Market*.

This paper presents a research carried out on one of the largest marketplace (specially for drugs) on Internet, the ***AlphaBay Dark Web Market***. This web Market has caugth the attention of governemental agencies since two teenagers aged of 13 and 18 died after overdosing on a powerful synthetic opioid. Therefore, during our research, AlphaBay and Hansa, another dark web market, have been shut down on July, 2017 as a part of a law enforcement operation by the Federal Bureau of Investigation, the Drugs Enforcement Administration and European law enforcement agencies acting through Europol.[2] [3]

According to US Attorney General Jeff Sessions the aim of this action was to caution criminals from thinking that they could evade prosecution by using the dark web. However, it is widely believed that other web markets will take the place of *AlphaBay*. By the way, the popularity of AlphaBay can be explained by the shut down of *Silk Road 2.0* on 2013 since it has been launched on september 2014.

The understanding of such illegal market is crucial to fight it. Information gathered in those websites, allow to identify which are the most wanted ads for the consumer and where they come from. Therefore it might be possible to detect the footprint of each seller and, thus, help governmental agencies to identify recurrent sellers with various hidden identities.

Thus AlphaBay market will be analysed. Its nature, its different countries of origin, its main sellers, its predominance of items and so on will be investigated. During a first phase "Basic Statistics" will be carried out on the *Database*, in order to discover the marketplace and to point out its trends. Then, experimental results of *data mining techniques* will be discussed.

#----------------------------------------------------------  
# Library :  
#----------------------------------------------------------  
  
#install.packages("stringr")  
#install.packages("units")  
#install.packages("ggmap")  
#install.packages("plotrix")  
#install.packages("rattle")  
#install.packages("rpart")  
#install.packages("rpart.plot")  
#install.packages("RColorBrewer")  
#install.packages("arules")  
#install.packages("arulesViz")  
#install.packages("e1071")  
#install.packages("bnlearn")  
#install.packages("lubridate")  
#install.packages("dygraphs")  
#install.packages("zoo")  
#install.packages("forecast")  
#install.packages("RTextTools")  
#install.packages("tm")  
  
# Mamipulation string  
library(stringr)  
  
# Using unit  
library(units)  
  
# Mamipulation Date  
library(lubridate)  
library(zoo)  
library(forecast)  
  
# dynamic Plot  
library(dygraphs)  
  
# Plot a map  
library(ggmap)  
library(plotrix)  
  
# Decision Tree  
library(rattle)  
library(rpart)  
library(rpart.plot)  
library(RColorBrewer)  
  
# Association rules  
library(arules)  
library(arulesViz)  
  
# Bayesian classification, naive algorithm  
library(e1071)  
  
# Bayesian Network  
library(bnlearn)  
  
#Text mining   
library(RTextTools)  
library(tm)

#----------------------------------------------------------  
# Importation of the data :  
#----------------------------------------------------------  
  
data <- as.data.frame(read.csv("../../alphaClean.csv"))

# - Technical Implementation

## - Code Repository

All the code is publicly available in the Github project *"Data Mining - Dark Web Market"*. The repository is accessible from the following link: <https://github.com/SimonDele/Data-Ming-Dark-Web-Market>

## - Chosen Technologies

There is a number of technologies and programming languages that can be used for *Data Analysis*. Here are the ones that have been used for this project.

### - R / RStudio

The 2 main progamming languages for this kind of research are *Python and R*. Python is the most used. But since R has obtained more attention for the last few years, that is the one that will be used here. Moreover R, which is a language dedicated for statistical and graphical presentation, will suit our needs.

Furthermore, R has a powerful dedicated environment (IDE) which is *RStudio* and a lot of *open-source libraries*. Here is a list of the main libraries used for this project.

**stringr :** String manipulation library.

**units :** Unit library including solution for convertion.

**rpart :** Package that contains a wide library for decision tree method.

**arules :** Used for association rules.

**e1071 :** Bayesian Naive implementation library.

**bnlearn :** library including solution for bayesian network creation and visualisation.

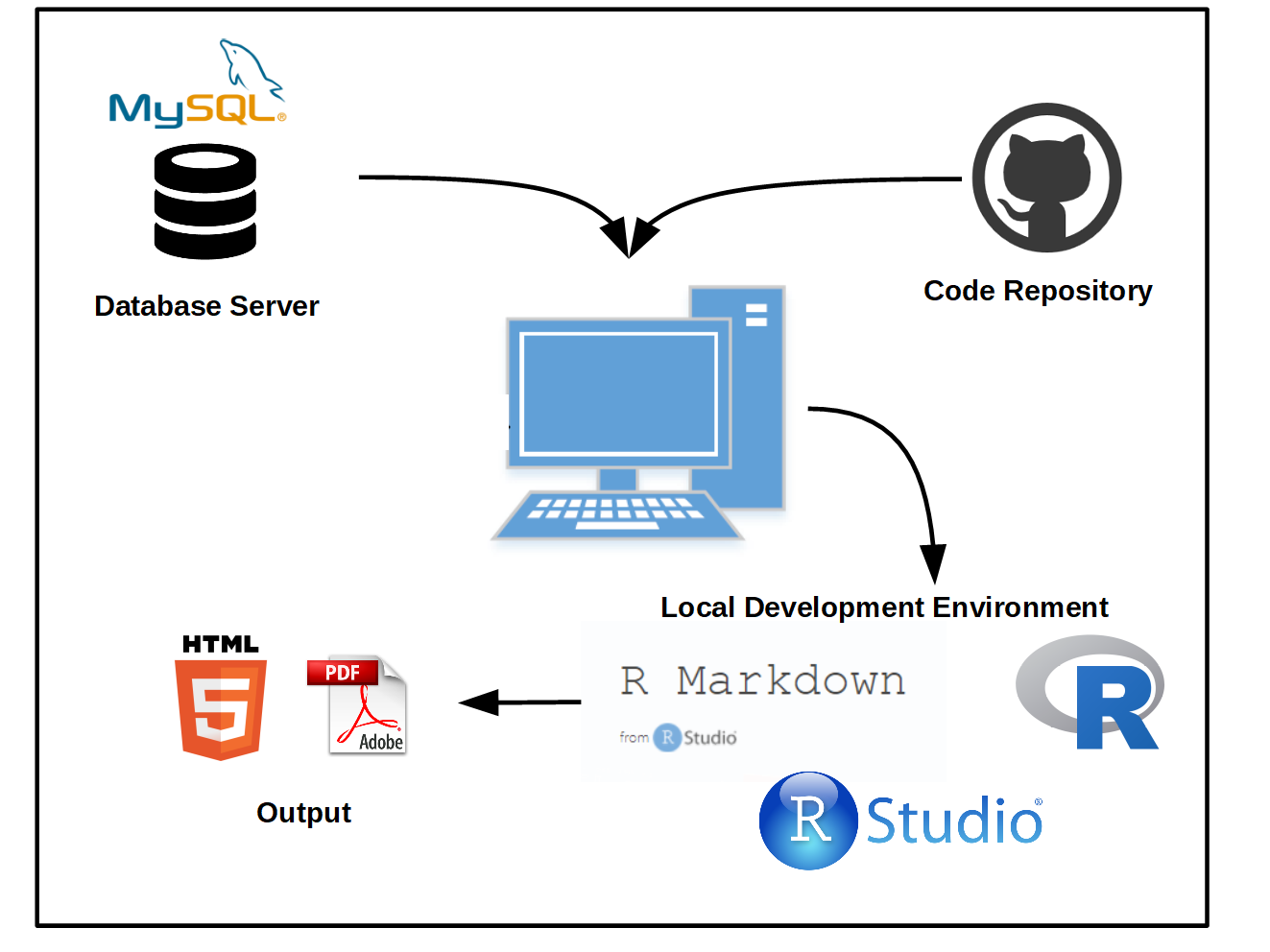
You can find the whole list of packages used in the *GitHub* repository.

### - R Markdown Notebooks

This document has been produced by *R Notebooks*. R Notebooks is an *R Markdown* document with chunks that can be executed independently and interactively. It enables to produce the document in various formats such as *HTML, PDF and Word*. In addition, it can be used directly into the RStudio IDE.

## - Detailed Diagram

This is the representation of the technical implementation taking place during this project:



## Figure 1 - Technical implementation

# - Data Retrieval

Thanks to Sin Wee Lee and Andres Baravalle data have been collected on the *AlphaBay Dark Web Market*. Each row of the data represents an ad and all the significant information can be find in the different columns : title, description, price (in USD), url link, seller, payment, origin, destination, category, timestamp that is to say the date when the ad was collected, creation date of the ad, number of product sold since this date and a link to the image.

The Data represents approximately *1/10* of the Web Market, but gives a pretty good representation since the uploaded ads were fairly distributed.

Thus, the first step was to clean the data (remove special characters, switch in lowercase ... ) and makes it readable in a computer way. That it's to say, to find in the title or description of the ads the amount (number and mass) of the product they are selling. Indeed, at the beginning these information were not given distinctly. Therefore an important work has been done on it in order to make analyses easier. With these information, the price of one unit of one dose (1 gram) of the product has been calculated and added.

Here is what the *Database* looks like :

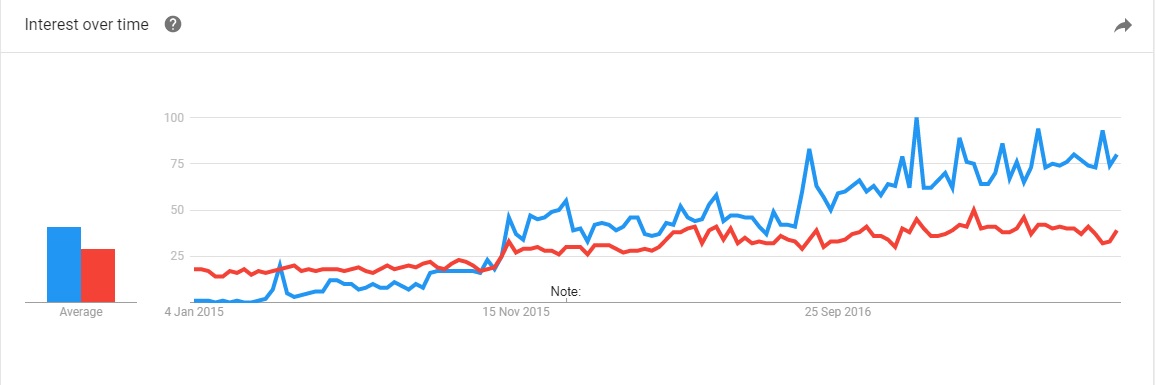
#----------------------------------------------------------  
# Display of the data :  
#----------------------------------------------------------  
  
printdata <- function() {  
 print.data <- subset (data[19409:19411,], select=-X)  
 for(i in 2:4){  
 print.data[,i] <- as.character(print.data[,i])  
 }  
 print(print.data)  
}  
  
printdata()

## id title  
## 19409 226445 ritalin 10 mg 90 tablets brand name from novartis  
## 19410 226448 tramadol 100mg 200 tablets  
## 19411 226449 panam::fe 1g of amsterdam's ketamine \*\*shipped from usa\*\*  
## brief  
## 19409 i am selling ritalin 10 mg 90 tablets from novartis,no generic or imitation from another countries,round white pill, ciba on one side, scored on the other side with a on the left of the score mark and b on the right (like "a | b")only mail in usa,thanks.  
## 19410 i am selling tramadol 100mg, 200 tablets,order's receive before 11am pacific time will go the same day, only mail in usa,thanks.  
## 19411 straight from the best god damn lab in amsterdam to you. panam's policy: i do not and refuse to cut any of my products! i would rather build up customers over time selling the best stuff then sell cheap shit for a quick buck. escrow listing: /listing.php?id=284146  
## ad  
## 19409 product description i am selling ritalin 10 mg 90 tablets from novartis,no generic or imitation from another countries,round white pill, ciba on one side, scored on the other side with a on the left of the score mark and b on the right (like "a | b")only mail in usa,thanks. ritalin 10 mg novartis   
## 19410 product description i am selling tramadol 100mg, 200 tablets,order's receive before 11am pacific time will go the same day, only mail in usa,thanks. tramadol 100mg 200 tablets   
## 19411 product description straight from the best god damn lab in amsterdam to you. panam's policy: i do not and refuse to cut any of my products! i would rather build up customers over time selling the best stuff then sell cheap shit for a quick buck. escrow listing: /listing.php?id=284146 ketamine special k kat   
## price url seller  
## 19409 250 http://pwoah7foa6au2pul.onion/listing.php?id=226445 milo8490  
## 19410 250 http://pwoah7foa6au2pul.onion/listing.php?id=226448 milo8490  
## 19411 60 http://pwoah7foa6au2pul.onion/listing.php?id=226449 Panam  
## payment origin destination  
## 19409 Escrow United States United States  
## 19410 Escrow United States United States  
## 19411 FE Listing 100% United States United States  
## category timestamp  
## 19409 /Drugs & Chemicals/Stimulants/Other 2017-06-14 19:21:52  
## 19410 /Drugs & Chemicals/Opioids/Pills 2017-06-15 10:57:36  
## 19411 /Drugs & Chemicals/Dissociatives/Ketamine 2017-06-09 16:38:09  
## sold\_since products\_sold  
## 19409 2016-10-13 28  
## 19410 2016-10-13 12  
## 19411 2016-10-13 NULL  
## image  
## 19409 NULL  
## 19410 http://pwoah7foa6au2pul.onion/images/uploads/2016/10/13/ff392dc8  
## 19411 NULL  
## dose unit quantity priceUnit priceUnitDose  
## 19409 0.01 g 90 2.777778 277.7778  
## 19410 0.10 g 200 1.250000 12.5000  
## 19411 1.00 g NA 60.000000 60.0000

## Table 1 - Database Sample

# - AlphaBay Market

As it has been said in the [Introduction](#introduction), *AlphaBay*, due to its popularity, drew the governmental attentions. As a matter of fact its reputation can be reflected by looking at *Google statistics* [4].



## Figure 2 - Evolution of AlphaBay and Dream Market Google researches

On this graph, *AlphaBay* is in blue and *Dream Market* is in red, which is an other Dark Web Market still operating. This is showing the evolution of *Google* researches that are related to these two Web Markets from 2015 until June 2017. It should be noted that *AlphaBay* has become more and more popular for the last tree years, and that just before being shut down by the authority in july 2017, it was one of the most popular marketplaces.

Let's now try to look at the evolution of the market with the collected data on *AlphaBay*. Here you can see the number of ads posted per month from *January 2015* until *June 2017*.

#------------------------------------------------------------  
# Evolution of the market  
#------------------------------------------------------------  
  
Evolution <- function() {  
  
 #-----------------  
 # New Data   
 #-----------------  
   
 # Select the column of the data that are interesting   
 Evo.data <- subset(data, select=c(category,sold\_since))  
 # Subset : choose the colunm that you want  
   
 # Remove ads with no informations  
 Evo.data <- Evo.data[which(Evo.data$sold\_since != "NULL"),]  
   
 # Formatting  
 Evo.data$sold\_since <- as.Date(Evo.data$sold\_since)  
   
 # Remove ads from 2014 (only 10)  
 Evo.data <- Evo.data[which(year(Evo.data$sold\_since) > 2014),]  
   
 # Counting the Ads  
 ad2015 <- nrow(Evo.data[which(year(Evo.data$sold\_since) == 2015),])  
 ad2016 <- nrow(Evo.data[which(year(Evo.data$sold\_since) == 2016),])  
 ad2017 <- nrow(Evo.data[which(year(Evo.data$sold\_since) == 2017),])  
 Number\_of\_Ads <- c(ad2015,ad2016,ad2017)  
   
 # Month  
 Evo.data$sold\_since <- as.yearmon(Evo.data$sold\_since)  
   
 # Calculating the number od ads per Month  
 tab\_Evo <- table(Evo.data$sold\_since)  
 Evo.data <- as.data.frame(tab\_Evo)  
 colnames(Evo.data) <- c("Time","Ads")  
 Evo.data <- Evo.data[1:(length(Evo.data$Time)-1),]  
 Evo.data$Time <- as.yearmon(Evo.data$Time)  
   
 # Time series object :  
 # start: from the first date that we have   
 # frequency=12 i.e. monthly observations ((1=annual, 4=quartly)  
 Evo <- ts(Evo.data$Ads, start=Evo.data$Time[1], frequency=12)  
   
 # Prediction to Dec 2017 :  
 # Analyse   
 fit <- HoltWinters(Evo)  
 # Prediction   
 # n.ahead : predict next six month with a confidence of 0.95  
 # Prediction.interval : up and low prediction  
 pred <- predict(fit, n.ahead=6,prediction.interval=TRUE,level=0.75)  
   
 # Merge   
 all <- cbind(Evo,pred)  
   
 # Ploting  
 plot <- dygraph(all, main = "Evolution of the Market", ylab = "Number of Ads") %>%  
 dySeries("Evo", label = "Number of Ads") %>%  
 dySeries(c("pred.lwr", "pred.fit", "pred.upr"), label = "Prediction") %>%  
 dyOptions(drawPoints=TRUE, pointSize=2) %>%  
 dyHighlight(highlightCircleSize=5) %>%  
 dyEvent("2017-06-01", "Shut-Down", labelLoc = "bottom") %>%  
 dyRangeSelector  
   
 return(plot)  
   
}  
  
plot <- Evolution()

## Figure 3 - Evolution of AlphaBay Web Market

The overall appearance and the growing popularity can be again pointed out with this graph. Between 2015 and 2016, there was a significant jump, the amount of ads rose from 7712 up to 14161. Nevertheless the most surprising thing is that the number of ads that have been posted during the six first months of 2017 (before the closing) is 12878 which is almost the same that in the whole 2016.

In order to see how the market would have looked like in the end of 2017 a prediction also has been added on this graph. Therefore, according to prognoses, the amount of ads would have reached a pick of 5,000 ads by the end of year.

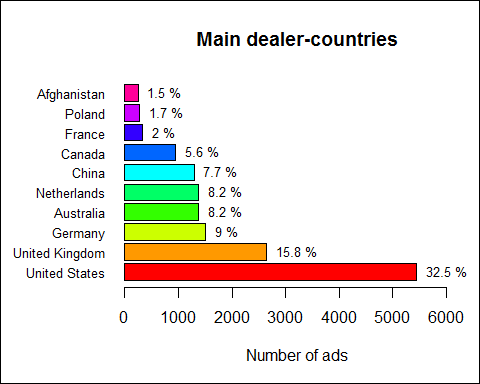
# - Basic Statistics

## - General distribution

As it has been said before, basic statistics have been first realized. Let's see the general distribution and trend of the market.

**1. Global view of ads distribution**

#-----------------------------------------------  
# Number of ads in the world  
#-----------------------------------------------  
  
NumberOfAds <- function() {  
  
 # 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(table(data[matching\_vector, "origin"]), decreasing=TRUE)  
   
 # Bar plot with the total number ofs ads in each country  
   
 par(las=1)#display yaxis horizontally  
 par(mar=c(5,6.5,4,0.5)) #give space for yaxis  
   
 barp <- barplot(sumup[1:10], main="Main dealer-countries", xlim= c(0,max(sumup[1:10])+1000), xlab="Number of ads",horiz = TRUE, col = rainbow(10), cex.names = 0.8)  
   
 # Labels  
 # Calculation in percentage  
 sumuppercent<- round(100\*(sumup/sum(sumup)), 1)  
 # round(a,1) : one digit after the comma  
   
 lab <- c()  
   
 for(i in 1:length(sumuppercent)) {  
 lab[i] <- paste(sumuppercent[[i]], "%", sep=" ")  
 }  
   
 barp <- text(y = barp, x = sumup[1:10], label = lab[1:10], pos=4 , cex = 0.8, col= "black")  
   
 # Frame  
 box(which = "outer", lty = "solid")  
  
}  
   
NumberOfAds()

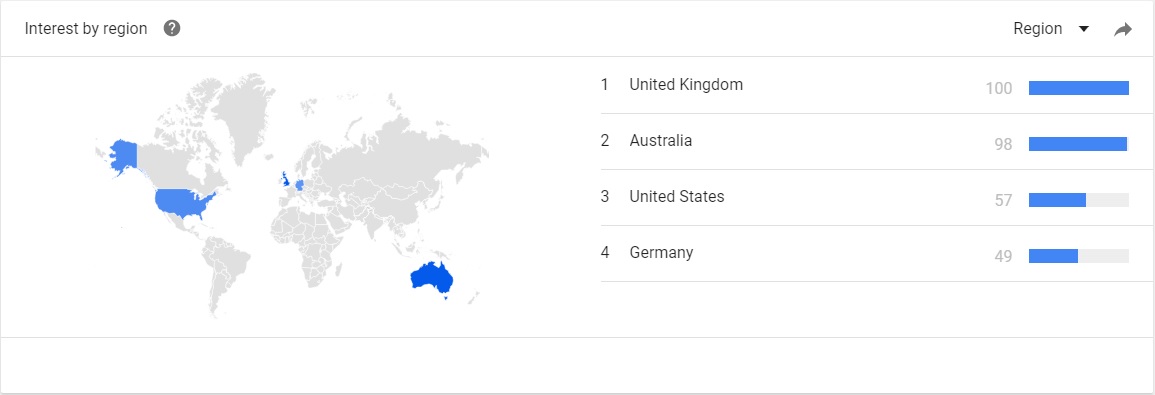


## Figure 4 - Main Dealer Countries

This bar-chart represents the 10 main countries in the world regarding the number of ads. As we can see, *United States* are the biggest dealer far ahead of the rest. Their number of ads is more than twice as the number of the second one, which is *United Kingdom*.

Moreover, it is noticeable that most of these countries are economically powerful. For instance on these ten main countries, five belong to the Group of Seven (G7), only *Japan* and *Italy* are not present. And other ones are also located in powerful areas where a lot of trade are made with other countries.

Furthermore an interesting thing to point out is that the first four countries are exactly the one where the word *"AlphaBay"* is the most researched on *Google* [5].

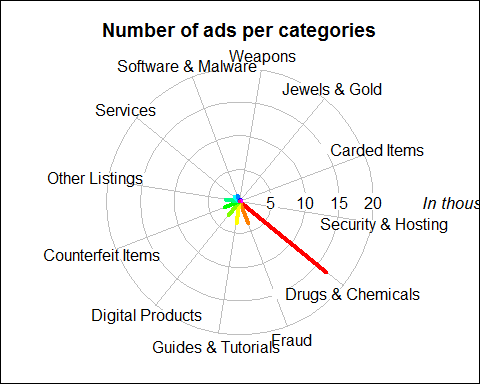


## Figure 5 - AlphaBay world Google researches

**2. Now let's have a look at the distribution of ads per category**

selectDrug <- function(drugName){  
 matching\_vector <- c( (str\_detect(data$category, drugName)))  
 return(matching\_vector)  
 }

#-----------------------------------------------  
# Number of ads per categories  
#-----------------------------------------------  
  
categ <- function(){  
   
 cat <- c()  
  
 for(i in 1:length(data$category)) {  
 cat[i] <- unlist(strsplit(as.character(data$category[i]), "/"))[2]  
 if(is.na(cat[i])) {cat[i] <- "Other Listings"}  
 }  
   
 tab\_cat <- table(cat)  
 tab\_cat <- sort(tab\_cat, decreasing=TRUE)   
 cat.data <- as.data.frame(tab\_cat)  
   
 radial.plot(cat.data$Freq,labels=cat.data$cat,label.prop=1.1,rp.type="r",start=5.6,clockwise=TRUE,lwd=4,line.col=rainbow(length(tab\_cat)),main="Number of ads per categories",radial.labels=c(5,10,15,20))  
  
 mtext("In thousands ads",side = 4,line=2,las=1,cex=1.08,font=3)  
   
 # Frame  
 box(which = "inner", lty = "solid")  
   
}  
  
categ()



## Figure 6 - Distribution of the Market

RateDrugAd <- function() {  
  
 rate <- c()  
   
 NbAd <- nrow(data)  
   
 # Select all "Drugs & Chemicals" ads  
 matching\_vector <- c( str\_detect(data$category, "Drugs & Chemicals"))  
 drug.data <- data[matching\_vector,]  
   
 NbAdDrug <- nrow(drug.data)  
   
 rate[1] <- round((NbAdDrug/NbAd)\*100,2)  
   
 # Select all "Fraud" ads  
 matching\_vector <- c( str\_detect(data$category, "Fraud"))  
 fraud.data <- data[matching\_vector,]  
   
 NbAdFraud <- nrow(fraud.data)  
   
 rate[2] <- round((NbAdFraud/NbAd)\*100,2)  
   
 return(rate)  
}  
  
rate <- RateDrugAd()

To begin with, it is observable that there are 12 main categories in this web marketplace. It appears that *"Drugs and Chemicals"* group is the largest one. By the way, it represents 45.64 % of the global market.

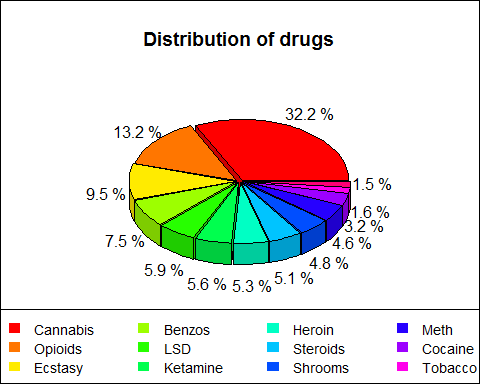
It is also worth noting that the second most popular category is *"Fraud"*, that is to say all the ads regarding impersonation, deception papers and accounts. It represents 13.5 % of the market. Eventually, all other items (digital product, weapons, jewelry ...) represent a small rate of the marketplace.

## - Drug Market

*AlphaBay Web Market* is a well known place for dealing drugs, this last chart has proved that. Thus, let's focus on the drug market.

**1. Distribution of drugs**

#-----------------------------------------------  
# Distribution of Drugs in the market  
#-----------------------------------------------  
  
DistributionDrugs <- function() {  
  
 #----------------------------  
 # The most common drugs  
 #----------------------------  
   
 drugs <- c("Cocaine", "Meth", "LSD", "Opioids", "Cannabis", "Steroids", "Ecstasy", "Ketamine", "Heroin", "Shrooms", "Tobacco", "Benzos", "Paraphernalia")  
   
 freq <- c()  
 for(i in 1:length(drugs)){  
 matching\_vector <- selectDrug(drugName=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- Plot :  
 pie3D(piepercent,labels = lab,labelcex = 1, main = title ,col=c, theta = 0.9, explode = 0.04)  
   
 # 5- Legend :  
 legend(x=-2.3,y=-1.1,res$drugs, cex = 0.9, fill = c,ncol=5,border=NA, xpd=NA)  
   
 # Frame  
 box(which = "inner", lty = "solid")  
}  
  
DistributionDrugs()

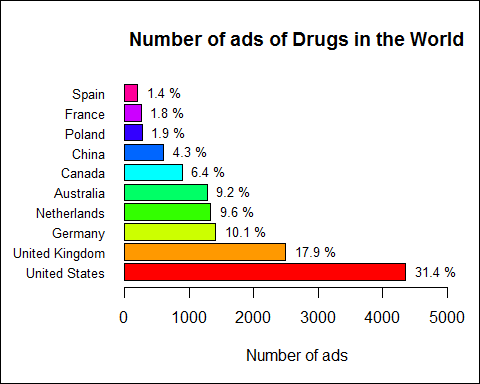


## Figure 7 - Drug Distribution

A large range of drugs categories can be discerned. Nevertheless, *Cannabis*, *Opioids* and *Ecstasy* cover *more than 50 %* of the market. This is not surprising since Cannabis and Opioids can be *easily found* in some countries. Thus, it is easy for them to sell these drugs for the remaining countries where they are very requested. As a matter of fact, Cannabis is *legalised* in some countries and some Opioids can be obtained thanks to *prescription* by a doctor. The massive presence of this kind of drugs on the market raises the issue of how some of them are prescripted. The *remaining* of the market is splited by all other drugs.

**2. World distribution of drugs**

#---------------------------------------------  
# Number of ads of Drugs in the world  
#---------------------------------------------  
  
NumberOfAdsDrugs <- function(){  
  
 # Get rid of unwanted orign like Worldwide and Null which are not relevant  
 matching\_vector <- c( str\_detect(data$category, "Drugs") & !str\_detect(data$origin, "Worldwide") & !str\_detect(data$origin, "NULL"))  
   
 sumup <- sort(table(data[matching\_vector, "origin"]), decreasing=TRUE)  
   
 # Bar plot with the total number of ads of Drugs in each country  
 par(las=1)#display yaxis horizontally  
 par(mar=c(5,6.5,4,0.5)) #give space for yaxis  
   
 barp <- barplot(sumup[1:10], main="Number of ads of Drugs in the World", xlab="Number of ads",xlim = c(0,max(sumup[1:10]+1000)), col = rainbow(10), cex.names = 0.8, horiz = TRUE)  
   
 # Calculation in percentage  
 sumuppercent<- round(100\*(sumup/sum(sumup)), 1)  
 # round(a,1) : one digit after the comma  
   
 lab <- c()  
   
 for(i in 1:length(sumuppercent)) {  
 lab[i] <- paste(sumuppercent[[i]], "%", sep=" ")  
 }  
   
 barp <- text(y = barp, x = sumup[1:10], label = lab[1:10], pos=4 , cex = 0.8, col= "black")  
   
 # Frame  
 box(which = "outer", lty = "solid")  
  
}  
  
NumberOfAdsDrugs()



## Figure 8 - Main drug dealer countries

Once more, the 10 main dealer-countries in the world are plotted but this time regarding the number of drugs ads. At first sight, the chart looks like the first one. This is coherent, indeed, by comparing the ratio between drugs ads and the total number of ads, it is intelligible that they are mainly dealing drugs.

This also matches with the second chart that shows that drugs are the main item in the market. There are few exceptions such as *Afghanistan* which has been replaced by *Spain* and *Canada* has been reversed with *China*.

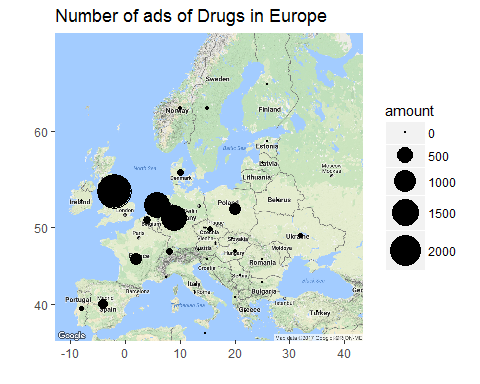
But so far it is possible to conclude that the market of drugs is gathered in Europe and the north of America.

**3. Take a global view of the drugs market in Europe with the following map**

MapEurope <- function() {  
  
 # Select the ads about drugs and get rid of the irrelevant orign Worlwide  
 matching\_vector <- c( (str\_detect(data$category, "Drugs") ) & !str\_detect(data$origin, "Worldwide"))  
   
 sumup <- sort(summary(data[matching\_vector, "origin"]), decreasing=TRUE)  
   
 # Read a file containing the latitude and longitude of the "center" of each country  
 data\_country <- read.csv("../Stats/lat\_long.csv")  
 lat\_long <- data.frame(Country = data\_country$Country , long= data\_country$Longitude..average., lat= data\_country$Latitude..average.)  
   
 # Create a data.frame with the name of the country and its nb of ads  
 v <- data.frame(name= names(sumup) , amount = sumup)  
   
 # Merge v with lat\_long in order to have a data with Country/NbofAds/lattitude/longitude  
 data\_plot <- merge(v, lat\_long, by.x = "name", by.y = "Country" )  
   
 # Create a map of EUROPE with circles showing the amount of ads  
 map <- get\_map(location = 'Europe', zoom =4 )  
 mapPoints <- ggmap(map) + xlab("") + ylab("") + ggtitle("Number of ads of Drugs in Europe")+  
 geom\_point(data = data\_plot,aes(x =long, y = lat, size =amount)) +scale\_size\_continuous(limits=c(0,3000),breaks=c(0,500,1000,1500,2000), range = c(0,13))   
   
 # Display  
 mapPoints  
  
}  
  
MapEurope()

## Warning in plyr::split\_indices(scale\_id, n): '.Random.seed' is not an  
## integer vector but of type 'NULL', so ignored

## Warning: Removed 40 rows containing missing values (geom\_point).



## Figure 9 - European Drug Dealers

Circles show the amount of ads concerning drugs. The map confirms previous assumptions that there are a lot of Drug dealers in Europe with, as major dealer countries, *United Kingdom*, *Netherlands* and *Germany*.

It appears that the principal dealer-countries are located on the Atlantic Coast and own huge harbours where there is important merchant shipping. Whereas on the East part there are not a lot of activities. This is probably due to the fact that dealers are using *international commercial maritime traffics* in order to dispatch their drugs all around the world. Maritime transport is an option increasingly used since it allows them to carry large quantities at one time. Drugs can be transported in small and fast boats (Go-Fast-Boat between countries border) or in containers on commercial vessels. Thus, significant seaports in Europe such as Rotterdam in Netherlands or Antwerp in Belgium are key points for this type of trafficking. In 2014 "Dutch police estimated that 25-50 % of the cocaine reaching Europe now enters via the port, which handles around 11 million containers a year." [6]

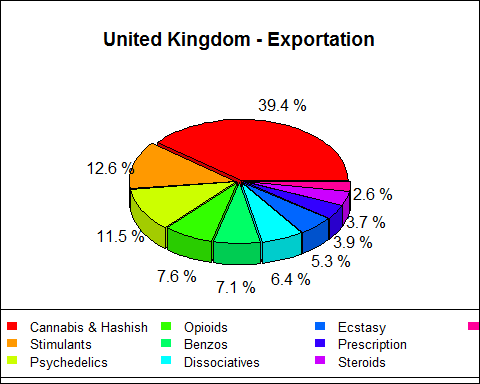
## - Product Flow By Country

Let's now focus more specifically on different countries and study their trend. To do so, export and import flows of the country have been investigated.

**1. United Kingdom exportation**

The pie below represents the repartition of each category that United Kingdom exports. Only the second subcategory has been kept because it appears to be the most relevant since the first one only gives information on the nature of the ad (for instance *"Drugs & Chemicals"*).

#-----------------------------------------------  
# Importation / Exportation of a country  
#-----------------------------------------------  
  
country\_Export <- function() {  
  
 #-------------------  
 # Initialization  
 #-------------------  
  
 country <- "United Kingdom"  
 num <- 0  
   
 # 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- Plot :  
 pie3D(piepercent,labels = lab,labelcex = 1, main = title ,col=c, theta = 0.9, explode = 0.04)  
   
 # 5- Legend :  
 legend(x=-2.3,y=-1.1,names(piepercent), cex = 0.8, fill = c,ncol=4,border=NA, xpd=NA)  
   
 # Frame  
 box(which = "inner", lty = "solid")  
}  
  
country\_Export()



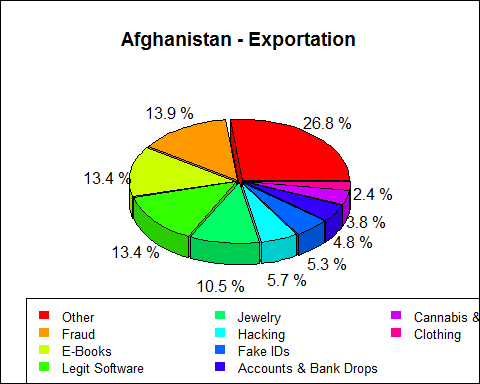
## Figure 10 - United Kingdom exportation

Given that most of exported items are drugs, that is not surprising that they are the most sold product, as it has been seen before. Once again this pie chart shows the market diversity. Although a huge part concerns *"Cannabis & Hashish"* category, *"Stimulants"* and other highly dangerous drugs are significantly present as we.

Most of European countries follows the same rules as *United Kingdom* and this confirms previous assumptions.

**2. Let's have a look at the exportations of Afghanistan which seem different to United Kingdom**

#-----------------------------------------------  
# Importation / Exportation of a country  
#-----------------------------------------------  
  
country\_Export <- function() {  
  
 #-------------------  
 # Initialization  
 #-------------------  
  
 country <- "Afghanistan"  
 num <- 0  
   
 # 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- Plot :  
 pie3D(piepercent,labels = lab,labelcex = 1, main = title ,col=c, theta = 0.9, explode = 0.04)  
   
 # 5- Legend :  
 legend(x=-2,y=-1,names(piepercent), cex = 0.8, fill = c,ncol=3,border=NA, xpd=NA)  
  
 # Frame  
 box(which = "inner", lty = "solid")  
   
}  
  
country\_Export()

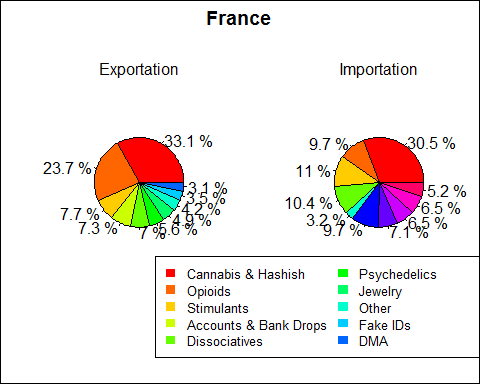


## Figure 11 - Afghanistan exportation

What is suprising is that, unlike most of countries, *Afghanistan* doesn't really retail drugs on *AlphaBay Market*. Actually, a vast majority of exported products are false identity, deception account... *Afghanistan* is also dealing electronic devices or softwares.

**3. Let's compare France export & import flows and see if there is a difference between them**

#-----------------------------------------------  
 # Importation / Exportation of a country  
 #-----------------------------------------------  
   
Country\_Export\_Import <- function() {  
  
 #-------------------  
 # Initialization  
 #-------------------  
   
 country <- "France"  
   
 #---------------------------  
 # Analysis - Exportation  
 #---------------------------  
   
 # Country as origin  
 matching\_vector <- str\_detect(data[,"origin"], 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\_exp <- table(cat[,3]) #cat[,3] : 2nd category   
 tab\_exp <- sort(tab\_exp, decreasing = TRUE) # Sorting (biggest in first)   
 tab\_exp <- tab\_exp[1:10] # Taking only the most important  
   
 #---------------------------  
 # Analysis - Importation  
 #---------------------------  
   
 # Country as destination  
 matching\_vector <- str\_detect(data[,"destination"], country)  
   
 # list of the categories (among the line that have "Country" as destination)  
 # -> Products (categories) importing 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\_imp <- table(cat[,3]) #cat[,3] : 2nd category   
 tab\_imp <- sort(tab\_imp, decreasing = TRUE) # Sorting (biggest in first)   
 tab\_imp <- tab\_imp[1:10] # Taking only the most important  
   
 #-------------------------  
 # Analysis - Fusion  
 #-------------------------  
   
 # Transformation in data frame  
 tab\_exp <- as.data.frame(tab\_exp)  
 tab\_imp <- as.data.frame(tab\_imp)  
   
 # Merger of the 2 data frame in order to have the same labels   
 tab <- merge(tab\_exp,tab\_imp,by.x="Var1",by.y="Var1",all = TRUE)  
   
 # Handling of the "NA" value (substitution by 0)  
 for (j in 2:3) {  
 for(i in 1:length(tab[,j])){  
 if(is.na(tab[i,j])) {tab[i,j] <-0}  
 }  
 }   
   
 #---------------------------  
 # Pie Chart - Exporation   
 #---------------------------  
   
 # ploting 2 graphics om the same picture  
 par(mfrow = c(1,2))  
   
 # 1- Labels :  
 # Calculation in percentage  
 piepercent <- round(100\*tab[,2]/sum(tab[,2]), 1)  
 # round(a,1) : one digit after the comma  
   
 lab <- c()  
   
 for(i in 1:length(piepercent)) {  
 if(piepercent[[i]] == 0) {lab[i] <- ""}   
 else {lab[i] <- paste(piepercent[[i]], "%", sep=" ")}  
 }  
  
 # 2- Colors :  
 c <- rainbow(length(tab[,1]))  
   
 # 3- Plot :  
 pie(piepercent,labels=lab,col=c)  
 mtext("Exportation",cex=1)  
   
 #----------------------------  
 # Pie Chart - Importation   
 #----------------------------  
   
 # 1- Labels :  
 # Calculation in percentage  
 piepercent <- round(100\*tab[,3]/sum(tab[,3]), 1)  
 # round(a,1) : one digit after the comma  
   
 lab <- c()  
   
 for(i in 1:length(piepercent)) {  
 if(piepercent[[i]] == 0) {lab[i] <- ""}  
 else {lab[i] <- paste(piepercent[[i]], "%", sep=" ")}  
 }  
  
 # 2- Plot :  
 pie(piepercent, labels=lab, col=c)  
 mtext("Importation",cex=1)  
   
 #------------------  
 # General - Plot   
 #-----------------  
   
 par(oma=c(0,0,1.8,0))  
 title("France",outer=TRUE)  
 legend(x=-4,y=-1.1,tab[,1], cex = 0.8, fill=c,ncol=3,border=NA, xpd=NA)  
   
 # Frame  
 box(which = "outer", lty = "solid")  
   
}  
  
Country\_Export\_Import()



## Figure 12 - France exportation and importation

It is noticeable that both pies are different. The percentages of each category are not equal and some of them don't appear systematicaly in the other pie chart. However it is obvious that significant exported drugs are also imported. Morevover, *France* is importing some drugs that are not local.

Nevertheless these conclusions should be moderated since targeting one particular country reduces significantly the number of information used for statistics.

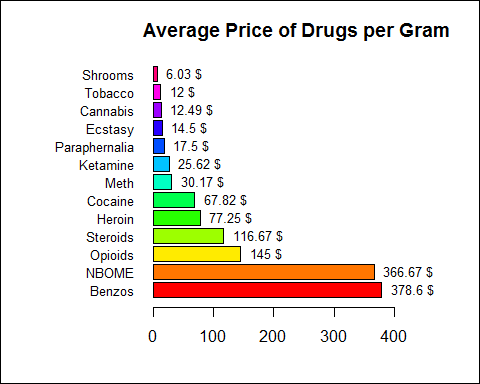
## - Market Prices

After analysing general trend and flows, one interesting topic to analyse is market prices. One may ask if sold products in *AlphaBay* are cheaper than in the streets.

**1. Average prices on the *AlphaBay Web Market***

Firstly, the average price of one gram of the most common drugs has been calculated and results below has been obtained.

DrugsPrices <- function() {  
   
 drugs <- c("Cocaine", "Meth", "Opioids", "Cannabis", "Steroids", "Ecstasy", "Ketamine", "Heroin", "NBOME","Shrooms", "Tobacco", "Benzos", "Paraphernalia")  
   
 med <-c()  
 for(i in 1:length(drugs)){  
 matching\_vector <- selectDrug(drugName = drugs[i]);  
 med[i] <- median((data[matching\_vector, "priceUnitDose"]))  
 }  
 priceDrugs <- data.frame(drugs, med);   
   
 priceDrugs$med <- round(priceDrugs$med,2)  
   
 priceDrugs <- priceDrugs[order(priceDrugs$med, decreasing=TRUE), ]  
   
 par(las=1) # Display yaxis horizontally  
 par(mar=c(4,8,3,2)) # Give space for yaxis  
   
 barp <- barplot(priceDrugs$med, main="Average Price of Drugs per Gram", names.arg = priceDrugs$drugs, xlim = c(0,max(priceDrugs$med+100)), cex.names = 0.8, col =rainbow(length(priceDrugs$drugs)), horiz =TRUE)  
 barp <- text(y = barp, x = priceDrugs$med, label = paste(priceDrugs$med, " $", sep=""), pos=4 , cex = 0.8, col= "Black")  
   
 # Frame  
 box(which = "outer", lty = "solid")  
   
 return (priceDrugs)  
  
}  
  
priceDrugs <- DrugsPrices()

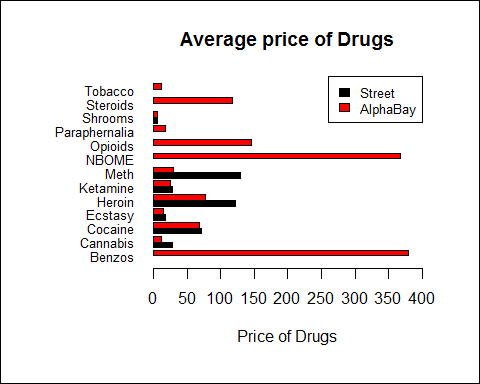


## Figure 13 - AlphaBay Drug Prices

**2. Comparison with the "street"**

Finally, some information about prices of street sellers have been collected, for same drugs as above. Thus, prices in articles and websites have been gathered. Unfortunately, values on some drugs are missing. Below are results of these investigations.

#-----------------------------------------  
# Prices find on articles  
#-----------------------------------------  
  
DrugsPricesDoc <- function(){  
   
 cols <- c("Cocaine", "Meth", "Opioids", "Cannabis" , "Steroids", "Ecstasy", "Ketamine", "Heroin", "NBOME","Shrooms", "Tobacco", "Benzos", "Paraphernalia" , "MDMA", "Amphetamine", "Crack", "LSD" , "URL")  
  
 ref1 <- c( 35 , 200 , NA , (5.3 + 7.85)/2 , NA , 15 , 25 , 100 , NA , NA , NA , NA , NA , 40 , 5 , NA , NA , "http://www.drugwise.org.uk/how-much-do-drugs-cost/")  
 ref2 <- c( 67 , NA , NA , 51 , NA , 15 , 32 , 129 , NA , NA , NA , NA , NA , 51 , 15 , 97 , NA , "http://www.telegraph.co.uk/news/uknews/crime/11346133/The-cost-of-street-drugs-in-Britain.html")  
 ref3 <- c( 110 , 80 , NA , NA , NA , NA , NA , 170 , NA , 5.7 , NA , NA , NA , 150 , NA , NA , 12000 , "http://www.rehabcenter.net/the-average-cost-of-illegal-drugs-on-the-street/ " )  
 ref4 <- c( 80 , 109 , NA , NA , NA , 19.12 , NA , 91.16 , NA , NA , NA , NA , NA , NA , NA , NA , NA , " http://o.canada.com/business/interactive-what-illegal-drugs-cost-on-the-street-around-the-world")  
 ref5 <- c( 64 , NA , NA , NA , NA , 20 , NA , NA , NA , NA , NA , NA , NA , NA , NA , NA , NA , " http://www.thestudentpocketguide.com/2012/01/student-life/health-and-relationships/facts-about-drugs/")  
   
 doc\_drugs <- t(data.frame(ref1, ref2, ref3, ref4, ref5))  
 colnames(doc\_drugs) <- cols  
   
 # Calculate the mean price of each drugs find on articles  
 price\_doc <- c()  
 for(i in 1 : length(cols)){  
 price\_doc[i] <- summary(as.numeric(doc\_drugs[,i]))[[4]]  
 }  
 price\_doc.data <- data.frame(cols, price\_doc)  
   
   
 # Merge the previous dataframe which correspond to the mean price of each drugs in the data  
 # with the dataframe created above  
 beside\_plot <- merge(price\_doc.data, priceDrugs, by.x ="cols", by.y ="drugs")   
   
 rownames(beside\_plot) <- beside\_plot[,1]  
 beside\_plot <- beside\_plot[,-1]  
   
 # Creating the barplot  
 par(las=1)#display yaxis horizontally  
 par(mar=c(6,8,4,3)) # Give space for yaxis  
   
 b <- barplot(rbind(beside\_plot[,1], beside\_plot[,2]), main="Average price of Drugs", xlim = c(0, 400),  
 xlab="Price of Drugs", beside=TRUE, names.arg = rownames(beside\_plot), col=1:2, horiz = TRUE, space = c(0,0.4), cex.names = 0.8)  
   
 axis(side=1,at=c(50,150,250,350),labels=c(50,150,250,350))  
   
   
 lab <- c("Street", "AlphaBay")  
   
 legend("topright",lab,fill=1:2, cex=0.8)  
   
 # Frame  
 box(which = "outer", lty = "solid")  
}  
  
DrugsPricesDoc()



## Figure 14 - Comparison of Prices between AlphaBay and Street

Globally, it appears that prices of *street sellers* are often largely higher than *AlphaBay* ads. In few cases both prices tend to be similar.(Please find in the last section all references used : [7], [8], [9], [10], [11])

# - Data mining

Secondly, data mining techniques have been performed in order to discover *hidden rules* and *correlations* in the database. Another goal is to predict the value of one variable given other values.  
Please note that for all this part the analysis is only on *Drug Market*.

## - Sellers Predictions

The first thing to wonder is how to guess the seller of an ad. To answer this question, different data mining methods have been used, especially *Decision Tree* and *Bayesian classification*.

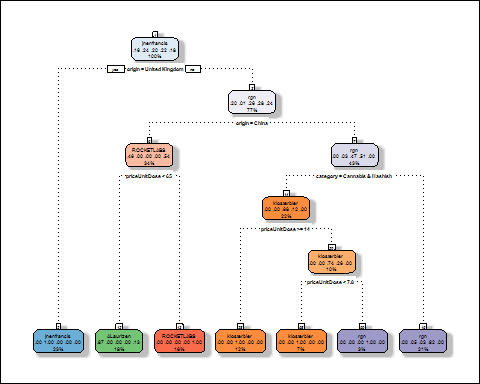
Algorithm has been run on a subset of the database with by rows ads and by columns the origin, category, seller and price. The aim is to predict who is selling each ads. By training the algorithm on one half of the data, predictions could be made on other half. Given that most of sellers own just few ads (occasional advertisements) only the main ones were selected, which represent at best the market. Otherwise, data mining techniques will fail in finding rules for them.

To check efficiency of the algorithm a measure of accuracy must be calculated. It is obtain by comparing the prediction of decision tree method with the true value.

**1. Decision tree**

Using *rpart package*, which is based on the *CART Alorithm*, a decision tree has been created. Thanks to it, predictions of the seller could be made. Prognoses on the five most significant sellers and the related tree can be found below.

#----------------------------------------------------------------------  
# Decision tree - CART algorithm  
# Prediction of the seller knowing the price / category / origin  
# Plot  
#-----------------------------------------------------------------------  
  
Dtsellers <- function(){   
  
 #-----------------  
 # New Data   
 #-----------------  
   
 # Select all "Drugs & Chemicals" ads  
 matching\_vector <- c( str\_detect(data$category, "Drugs & Chemicals"))  
 dectree.data <- data[matching\_vector,]  
   
 # Select the column of the data that are interesting for the tree  
 # ie removing colunm like "id" or "url" that don't give any informations  
 dectree.data <- subset(dectree.data, select=c(origin,category,seller,priceUnitDose))  
 # Subset : choose the colunm that you want  
   
 # Handling : column categorie  
 # Regular expression for spliting the categories  
 regex <- "/(.\*)/(.\*)/(.\*)"  
 cat <- str\_match(dectree.data$category, regex)  
 dectree.data$category <- cat[,3] # keep only the second part  
   
 # Handling : seller  
 tab\_sel <- table(dectree.data$seller)  
 tab\_sel <- sort(tab\_sel, decreasing=TRUE) # Sorting (biggest in first)  
 tab\_sel <- tab\_sel[1:5] # Taking only the most important : main sellers  
 name\_sel <- names(tab\_sel)  
 # New data keeping only the main sellers  
 dectree.data <-subset(dectree.data, seller %in% name\_sel)   
   
 # Random rows :  
 dectree.data <- dectree.data[sample(nrow(dectree.data),nrow(dectree.data),replace=FALSE), ]  
   
 #---------------------  
 # Decision tree  
 #---------------------  
   
 # Factor  
 dectree.data$seller <- factor(dectree.data$seller)  
   
 # Half of the data for making the decision tree  
 train.data <- dectree.data[1:(floor(nrow(dectree.data))/2),]  
   
 # Creation of the tree  
 tree <- rpart(seller ~.,data=train.data, method="class")   
   
 # Plot  
 fancyRpartPlot(tree, sub="")  
  
 # Frame  
 box(which = "outer", lty = "solid")  
   
 #--------------------  
 # Prediction  
 #--------------------  
   
 # The other half for the prediction  
 pred.data <- dectree.data[(floor(nrow(dectree.data)/2)+1):nrow(dectree.data),]  
   
 # Making prediction  
 pred <- predict(tree,pred.data,type="class")  
   
 # Analysis:  
   
 # Comparison between the result and the prediction (prediction in colunm)  
 conf <- table(pred.data[,match("seller",names(pred.data))],pred)  
   
 # Accurency  
 acc <- round((sum(diag(conf)) / sum(conf)\*100),2)  
   
 print(conf)   
 cat("\n")  
 cat(" ")  
 compTab <- TableCaption(compTab, "Sellers Prediction / Decision Tree method")  
   
 Result <- c(acc,compTab)  
   
 return(Result)  
}  
  
#accDTseller   
ResultDT1 <- Dtsellers()



## pred  
## ALaurizen jnenfrancis klosterbier rgn ROCKETLABS  
## ALaurizen 71 0 0 0 3  
## jnenfrancis 0 71 0 2 0  
## klosterbier 0 0 77 5 0  
## rgn 0 0 1 59 0  
## ROCKETLABS 11 0 0 0 49  
##   
## Table 2 - Sellers Prediction / Decision Tree method

## Figure 15 - Seller prediction / Decision Tree

## [1] "The accuracy is : 93.70 %"

It is striking to realise that predictions are very reliable since the accuracy is very high.

In the *Table 2* you can find by columns our Prognoses and by rows the real sellers. In other words the diagonal shows the number of correct predictions and all other values are mistakes made by the algortihm.

The tree enables to have a good visual aspect on it and gives a lot of essential information on these five main sellers.

Besides, with quick basic statistics, it is possible to validate this tree and its branchs, that is to say how it is splited. Indeed let's have a look at these five dealers. This array contains the main country, main category and median price of their ads.

#-------------------------------------------------------------  
# Statistics for decision tree  
#-------------------------------------------------------------  
  
DTanalysis <- function() {  
  
 # Select all "Drugs & Chemicals" ads  
 matching\_vector <- c( str\_detect(data$category, "Drugs & Chemicals"))  
 Drug.data <- data[matching\_vector,]  
   
 # Select the column of the data that are interesting for the tree  
 # ie removing colunm like "id" or "url" that don't give any informations  
 Drug.data <- subset(Drug.data, select=c(origin,category,seller,priceUnitDose))  
 # Subset : choose the colunm that you want  
   
 # Handling : column categorie  
 # Regular expression for spliting the categories  
 regex <- "/(.\*)/(.\*)/(.\*)"  
 cat <- str\_match(Drug.data$category, regex)  
 Drug.data$category <- cat[,3] # keep only the second part  
   
 # Handling : seller  
 tab\_sel <- table(Drug.data$seller)  
 tab\_sel <- sort(tab\_sel, decreasing=TRUE) # Sorting (biggest in first)  
 tab\_sel <- tab\_sel[1:5] # Taking only the 5 main sellers  
 seller <- names(tab\_sel) # Names of the 5 main sellers  
   
 # Initialization   
 ori <- c()  
 price <- c()  
 cat <- c()  
   
 # calculation of informations for each seller  
 for(i in 1:5) {  
 # Select ads from the ieme seller  
 matching\_vector <- c( str\_detect(Drug.data$seller, seller[i]))  
 sel.data <- Drug.data[matching\_vector,]  
   
 ori[i] <- names(sort(table(sel.data$origin),decreasing=TRUE)[1]) # First country   
 cat[i] <-names(sort(table(sel.data$category),decreasing=TRUE)[1]) # First category  
 price[i] <- round(summary(sel.data$priceUnitDose)[[3]],2) # Median price  
 }  
   
 sel.data <- as.data.frame(seller)  
 sel.data$origin <- ori  
 sel.data$category <- cat  
 sel.data$price <- price  
   
 print(sel.data)  
}  
  
DTanalysis()

## seller origin category price  
## 1 jnenfrancis United Kingdom Cannabis & Hashish 8.68  
## 2 klosterbier Worldwide Cannabis & Hashish 25.54  
## 3 rgn Worldwide Stimulants 17.75  
## 4 ALaurizen China Stimulants 6.96  
## 5 ROCKETLABS China Stimulants 247.50

## Table 3 - Seller Analyses

Obviously, these information are exactly the ones that are in the tree. *jnenfrancis* is guessed thanks to origin *United Kingdom*. For the sellers dealing mainly in *China*, that is to say *ALaurizen* and *ROCKETLABS*, the spliting is made regarding the price (*ALaurizen* is obviously cheaper). For other sellers the algorithm use the category *Cannabis & Hashish*.

Predictions have been done here on only five sellers in order to have a readable tree (otherwise the size of the tree is too big for being plotted). However, prognoses with more sellers can be made and with still a good accuracy. For instance, with 10 sellers, the results below are obtained.

#----------------------------------------------------------------------  
# Decision tree - CART algorithm  
# Prediction of the seller knowing the price / category / origin  
# Results  
#-----------------------------------------------------------------------  
  
Dtsellers2 <- function(){   
  
 #-----------------  
 # New Data   
 #-----------------  
   
 # Select all "Drugs & Chemicals" ads  
 matching\_vector <- c( str\_detect(data$category, "Drugs & Chemicals"))  
 dectree.data <- data[matching\_vector,]  
   
 # Select the column of the data that are interesting for the tree  
 # ie removing colunm like "id" or "url" that don't give any informations  
 dectree.data <- subset(dectree.data, select=c(origin,category,seller,priceUnitDose))  
 # Subset : choose the colunm that you want  
   
 # Handling : column categorie  
 # Regular expression for spliting the categories  
 regex <- "/(.\*)/(.\*)/(.\*)"  
 cat <- str\_match(dectree.data$category, regex)  
 dectree.data$category <- cat[,3] # keep only the second part  
   
 # Handling : seller  
 tab\_sel <- table(dectree.data$seller)  
 tab\_sel <- sort(tab\_sel, decreasing=TRUE) # Sorting (biggest in first)  
 tab\_sel <- tab\_sel[1:10] # Taking only the most important : main sellers  
 name\_sel <- names(tab\_sel)  
 # New data keeping only the main sellers  
 dectree.data <-subset(dectree.data, seller %in% name\_sel)   
   
 # Random rows :  
 dectree.data <- dectree.data[sample(nrow(dectree.data),nrow(dectree.data),replace=FALSE), ]  
   
 #---------------------  
 # Decision tree  
 #---------------------  
   
 # Factor  
 dectree.data$seller <- factor(dectree.data$seller)  
   
 # Half of the data for making the decision tree  
 train.data <- dectree.data[1:(floor(nrow(dectree.data))/2),]  
   
 # Creation of the tree  
 tree <- rpart(seller ~.,data=train.data, method="class")   
  
 #--------------------  
 # Prediction  
 #--------------------  
   
 # The other half for the prediction  
 pred.data <- dectree.data[(floor(nrow(dectree.data)/2)+1):nrow(dectree.data),]  
   
 # Making prediction  
 pred <- predict(tree,pred.data,type="class")  
   
 # Analysis:  
   
 # Comparison between the result and the prediction (prediction in colunm)  
 conf <- table(pred.data[,match("seller",names(pred.data))],pred)  
   
 # Accurency  
 acc <- round((sum(diag(conf)) / sum(conf)\*100),2)  
   
 # Display   
 conf <- as.data.frame(conf)  
 names(conf) <- c("Sellers","Prediction","Freq")  
 print(conf)  
   
 return(acc)  
   
}  
  
acc1 <- Dtsellers2()

## Sellers Prediction Freq  
## 1 ALaurizen ALaurizen 61  
## 2 empireteam ALaurizen 0  
## 3 Fapppylicious ALaurizen 0  
## 4 FelixUK ALaurizen 0  
## 5 GreenLeafLabs ALaurizen 0  
## 6 jnenfrancis ALaurizen 0  
## 7 klosterbier ALaurizen 0  
## 8 optiman ALaurizen 0  
## 9 rgn ALaurizen 0  
## 10 ROCKETLABS ALaurizen 17  
## 11 ALaurizen empireteam 0  
## 12 empireteam empireteam 44  
## 13 Fapppylicious empireteam 0  
## 14 FelixUK empireteam 0  
## 15 GreenLeafLabs empireteam 21  
## 16 jnenfrancis empireteam 0  
## 17 klosterbier empireteam 0  
## 18 optiman empireteam 0  
## 19 rgn empireteam 0  
## 20 ROCKETLABS empireteam 0  
## 21 ALaurizen Fapppylicious 0  
## 22 empireteam Fapppylicious 0  
## 23 Fapppylicious Fapppylicious 45  
## 24 FelixUK Fapppylicious 0  
## 25 GreenLeafLabs Fapppylicious 0  
## 26 jnenfrancis Fapppylicious 0  
## 27 klosterbier Fapppylicious 0  
## 28 optiman Fapppylicious 0  
## 29 rgn Fapppylicious 0  
## 30 ROCKETLABS Fapppylicious 0  
## 31 ALaurizen FelixUK 0  
## 32 empireteam FelixUK 0  
## 33 Fapppylicious FelixUK 0  
## 34 FelixUK FelixUK 41  
## 35 GreenLeafLabs FelixUK 0  
## 36 jnenfrancis FelixUK 0  
## 37 klosterbier FelixUK 0  
## 38 optiman FelixUK 0  
## 39 rgn FelixUK 0  
## 40 ROCKETLABS FelixUK 0  
## 41 ALaurizen GreenLeafLabs 0  
## 42 empireteam GreenLeafLabs 16  
## 43 Fapppylicious GreenLeafLabs 0  
## 44 FelixUK GreenLeafLabs 0  
## 45 GreenLeafLabs GreenLeafLabs 31  
## 46 jnenfrancis GreenLeafLabs 0  
## 47 klosterbier GreenLeafLabs 0  
## 48 optiman GreenLeafLabs 0  
## 49 rgn GreenLeafLabs 0  
## 50 ROCKETLABS GreenLeafLabs 0  
## 51 ALaurizen jnenfrancis 0  
## 52 empireteam jnenfrancis 0  
## 53 Fapppylicious jnenfrancis 0  
## 54 FelixUK jnenfrancis 2  
## 55 GreenLeafLabs jnenfrancis 0  
## 56 jnenfrancis jnenfrancis 71  
## 57 klosterbier jnenfrancis 0  
## 58 optiman jnenfrancis 0  
## 59 rgn jnenfrancis 0  
## 60 ROCKETLABS jnenfrancis 0  
## 61 ALaurizen klosterbier 0  
## 62 empireteam klosterbier 0  
## 63 Fapppylicious klosterbier 0  
## 64 FelixUK klosterbier 0  
## 65 GreenLeafLabs klosterbier 0  
## 66 jnenfrancis klosterbier 0  
## 67 klosterbier klosterbier 73  
## 68 optiman klosterbier 0  
## 69 rgn klosterbier 10  
## 70 ROCKETLABS klosterbier 0  
## 71 ALaurizen optiman 0  
## 72 empireteam optiman 0  
## 73 Fapppylicious optiman 0  
## 74 FelixUK optiman 0  
## 75 GreenLeafLabs optiman 1  
## 76 jnenfrancis optiman 0  
## 77 klosterbier optiman 0  
## 78 optiman optiman 54  
## 79 rgn optiman 0  
## 80 ROCKETLABS optiman 0  
## 81 ALaurizen rgn 0  
## 82 empireteam rgn 0  
## 83 Fapppylicious rgn 0  
## 84 FelixUK rgn 0  
## 85 GreenLeafLabs rgn 0  
## 86 jnenfrancis rgn 3  
## 87 klosterbier rgn 4  
## 88 optiman rgn 0  
## 89 rgn rgn 52  
## 90 ROCKETLABS rgn 0  
## 91 ALaurizen ROCKETLABS 0  
## 92 empireteam ROCKETLABS 0  
## 93 Fapppylicious ROCKETLABS 0  
## 94 FelixUK ROCKETLABS 0  
## 95 GreenLeafLabs ROCKETLABS 0  
## 96 jnenfrancis ROCKETLABS 0  
## 97 klosterbier ROCKETLABS 0  
## 98 optiman ROCKETLABS 0  
## 99 rgn ROCKETLABS 0  
## 100 ROCKETLABS ROCKETLABS 43

## Table 4 - Sellers Prediction / Decision Tree method

## [1] "The accuracy is : 87.44 %"

Once more, this array presents the number of mistakes and correct predictions.

**2. Bayesian classification - Naive alogrithm**

Bayesian Classification has been used in the same objective as decision tree, making predictions. Here are results when running *Bayesian Naive Algorithm* with the same data.

#----------------------------------------------------------------------------------  
# Bayesian Classification - Naive  
# Prediction of the Seller knowing the origin / price / category  
#-----------------------------------------------------------------------------------  
  
BayesSellersV1 <- function(){  
 #-----------------  
 # New Data   
 #-----------------  
   
 # Select all "Drugs & Chemicals" ads  
 matching\_vector <- c( str\_detect(data$category, "Drugs & Chemicals"))  
 bayesian.data <- data[matching\_vector,]  
   
 # Select the column of the data that are interesting  
 # ie removing colunm like "id" or "url" that don't give any informations  
 bayesian.data <- subset(bayesian.data, select=c(origin,category,seller,priceUnitDose))  
 # Subset : choose the colunm that you want  
   
 # Handling : column category  
 # Regular expression for spliting the categories  
 regex <- "/(.\*)/(.\*)/(.\*)"  
 cat <- str\_match(bayesian.data$category, regex)  
 bayesian.data$category <- cat[,3] # keep only the second part  
   
 sellers <- names(sort(table(bayesian.data$seller), decreasing = TRUE))[1:10]  
   
 bayesian.data <-subset(bayesian.data, seller %in% sellers)  
   
 bayesian.data$seller <- factor(bayesian.data$seller, labels = sellers)  
  
 #---------------------  
 # Bayesian stat  
 #---------------------  
   
 # Random rows :  
 bayesian.data <- bayesian.data[sample(nrow(bayesian.data),nrow(bayesian.data),replace=FALSE), ]  
   
 train.data <- bayesian.data[1:floor(nrow(bayesian.data)/2),]  
 pred.data <- bayesian.data[(floor(nrow(bayesian.data)/2)+1):nrow(bayesian.data),]  
   
 model <- naiveBayes(seller ~ ., data = train.data)  
   
   
 preds <- predict(model, newdata = pred.data)  
 conf\_matrix <- table(preds, pred.data$seller)  
 acc <- round(sum(diag(conf\_matrix)) / sum(conf\_matrix)\*100, 2)  
   
 # Display   
 conf\_matrix <- data.frame(conf\_matrix)  
 names(conf\_matrix) <- c("Sellers","Prediction","Freq")  
 print(conf\_matrix)  
   
   
 return(acc)  
}  
  
accBayesV1 <- BayesSellersV1()

## Sellers Prediction Freq  
## 1 jnenfrancis jnenfrancis 51  
## 2 klosterbier jnenfrancis 0  
## 3 rgn jnenfrancis 0  
## 4 ALaurizen jnenfrancis 0  
## 5 ROCKETLABS jnenfrancis 0  
## 6 empireteam jnenfrancis 0  
## 7 GreenLeafLabs jnenfrancis 0  
## 8 optiman jnenfrancis 0  
## 9 Fapppylicious jnenfrancis 0  
## 10 FelixUK jnenfrancis 3  
## 11 jnenfrancis klosterbier 0  
## 12 klosterbier klosterbier 9  
## 13 rgn klosterbier 0  
## 14 ALaurizen klosterbier 0  
## 15 ROCKETLABS klosterbier 55  
## 16 empireteam klosterbier 0  
## 17 GreenLeafLabs klosterbier 0  
## 18 optiman klosterbier 0  
## 19 Fapppylicious klosterbier 0  
## 20 FelixUK klosterbier 0  
## 21 jnenfrancis rgn 12  
## 22 klosterbier rgn 0  
## 23 rgn rgn 25  
## 24 ALaurizen rgn 0  
## 25 ROCKETLABS rgn 0  
## 26 empireteam rgn 0  
## 27 GreenLeafLabs rgn 10  
## 28 optiman rgn 0  
## 29 Fapppylicious rgn 1  
## 30 FelixUK rgn 0  
## 31 jnenfrancis ALaurizen 0  
## 32 klosterbier ALaurizen 0  
## 33 rgn ALaurizen 0  
## 34 ALaurizen ALaurizen 0  
## 35 ROCKETLABS ALaurizen 0  
## 36 empireteam ALaurizen 46  
## 37 GreenLeafLabs ALaurizen 0  
## 38 optiman ALaurizen 3  
## 39 Fapppylicious ALaurizen 2  
## 40 FelixUK ALaurizen 0  
## 41 jnenfrancis ROCKETLABS 0  
## 42 klosterbier ROCKETLABS 1  
## 43 rgn ROCKETLABS 0  
## 44 ALaurizen ROCKETLABS 0  
## 45 ROCKETLABS ROCKETLABS 48  
## 46 empireteam ROCKETLABS 0  
## 47 GreenLeafLabs ROCKETLABS 0  
## 48 optiman ROCKETLABS 0  
## 49 Fapppylicious ROCKETLABS 0  
## 50 FelixUK ROCKETLABS 0  
## 51 jnenfrancis empireteam 0  
## 52 klosterbier empireteam 0  
## 53 rgn empireteam 0  
## 54 ALaurizen empireteam 0  
## 55 ROCKETLABS empireteam 0  
## 56 empireteam empireteam 72  
## 57 GreenLeafLabs empireteam 0  
## 58 optiman empireteam 0  
## 59 Fapppylicious empireteam 4  
## 60 FelixUK empireteam 0  
## 61 jnenfrancis GreenLeafLabs 1  
## 62 klosterbier GreenLeafLabs 0  
## 63 rgn GreenLeafLabs 0  
## 64 ALaurizen GreenLeafLabs 0  
## 65 ROCKETLABS GreenLeafLabs 0  
## 66 empireteam GreenLeafLabs 0  
## 67 GreenLeafLabs GreenLeafLabs 76  
## 68 optiman GreenLeafLabs 0  
## 69 Fapppylicious GreenLeafLabs 3  
## 70 FelixUK GreenLeafLabs 0  
## 71 jnenfrancis optiman 0  
## 72 klosterbier optiman 0  
## 73 rgn optiman 0  
## 74 ALaurizen optiman 0  
## 75 ROCKETLABS optiman 27  
## 76 empireteam optiman 9  
## 77 GreenLeafLabs optiman 1  
## 78 optiman optiman 1  
## 79 Fapppylicious optiman 0  
## 80 FelixUK optiman 0  
## 81 jnenfrancis Fapppylicious 0  
## 82 klosterbier Fapppylicious 0  
## 83 rgn Fapppylicious 0  
## 84 ALaurizen Fapppylicious 0  
## 85 ROCKETLABS Fapppylicious 1  
## 86 empireteam Fapppylicious 0  
## 87 GreenLeafLabs Fapppylicious 54  
## 88 optiman Fapppylicious 0  
## 89 Fapppylicious Fapppylicious 15  
## 90 FelixUK Fapppylicious 0  
## 91 jnenfrancis FelixUK 21  
## 92 klosterbier FelixUK 0  
## 93 rgn FelixUK 0  
## 94 ALaurizen FelixUK 0  
## 95 ROCKETLABS FelixUK 0  
## 96 empireteam FelixUK 0  
## 97 GreenLeafLabs FelixUK 0  
## 98 optiman FelixUK 0  
## 99 Fapppylicious FelixUK 0  
## 100 FelixUK FelixUK 38

## Table 5 - Sellers Prediction / Naive Bayesian Classification

## [1] "The accuracy is : 56.88 %"

The accuracy is 56.88 % which is not very good comparing to decision tree. However one way to improve the prognosis is to add new variables to the data which could be relevant like the number of ads already sold and the creation date of the ad.

#----------------------------------------------------------------------------------  
# Bayesian Classification - Naive  
# Prediction of the Seller knowing the origin / price / category / products\_sold / date creation  
#-----------------------------------------------------------------------------------  
  
BayesSellers <- function(){  
 #-----------------  
 # New Data   
 #-----------------  
   
 # Select all "Drugs & Chemicals" ads  
 matching\_vector <- c( str\_detect(data$category, "Drugs & Chemicals"))  
 bayesian.data <- data[matching\_vector,]  
   
 # Select the column of the data that are interesting  
 # ie removing colunm like "id" or "url" that don't give any informations  
 bayesian.data <- subset(bayesian.data, select=c(origin,category,seller,priceUnitDose, sold\_since, products\_sold))  
 # Subset : choose the colunm that you want  
   
 # Handling : column category  
 # Regular expression for spliting the categories  
 regex <- "/(.\*)/(.\*)/(.\*)"  
 cat <- str\_match(bayesian.data$category, regex)  
 bayesian.data$category <- cat[,3] # keep only the second part  
   
 sellers <- names(sort(table(bayesian.data$seller), decreasing = TRUE))[1:10]  
   
 bayesian.data <-subset(bayesian.data, seller %in% sellers)  
   
 bayesian.data$seller <- factor(bayesian.data$seller, labels = sellers)  
  
 #---------------------  
 # Bayesian stat  
 #---------------------  
   
 # Random rows :  
 bayesian.data <- bayesian.data[sample(nrow(bayesian.data),nrow(bayesian.data),replace=FALSE), ]  
   
 train.data <- bayesian.data[1:floor(nrow(bayesian.data)/2),]  
 pred.data <- bayesian.data[(floor(nrow(bayesian.data)/2)+1):nrow(bayesian.data),]  
   
 model <- naiveBayes(seller ~ ., data = train.data)  
   
   
 preds <- predict(model, newdata = pred.data)  
 conf\_matrix <- table(preds, pred.data$seller)  
 acc <- round(sum(diag(conf\_matrix)) / sum(conf\_matrix)\*100, 2)  
   
 # Display   
 conf\_matrix <- data.frame(conf\_matrix)  
 names(conf\_matrix) <- c("Sellers","Prediction","Freq")  
 print(conf\_matrix)  
   
   
 return(acc)  
}  
  
accBayesV2 <- BayesSellers()

## Sellers Prediction Freq  
## 1 jnenfrancis jnenfrancis 66  
## 2 klosterbier jnenfrancis 0  
## 3 rgn jnenfrancis 0  
## 4 ALaurizen jnenfrancis 0  
## 5 ROCKETLABS jnenfrancis 0  
## 6 empireteam jnenfrancis 0  
## 7 GreenLeafLabs jnenfrancis 0  
## 8 optiman jnenfrancis 0  
## 9 Fapppylicious jnenfrancis 0  
## 10 FelixUK jnenfrancis 2  
## 11 jnenfrancis klosterbier 0  
## 12 klosterbier klosterbier 47  
## 13 rgn klosterbier 0  
## 14 ALaurizen klosterbier 0  
## 15 ROCKETLABS klosterbier 12  
## 16 empireteam klosterbier 0  
## 17 GreenLeafLabs klosterbier 1  
## 18 optiman klosterbier 0  
## 19 Fapppylicious klosterbier 0  
## 20 FelixUK klosterbier 0  
## 21 jnenfrancis rgn 0  
## 22 klosterbier rgn 0  
## 23 rgn rgn 45  
## 24 ALaurizen rgn 0  
## 25 ROCKETLABS rgn 0  
## 26 empireteam rgn 0  
## 27 GreenLeafLabs rgn 0  
## 28 optiman rgn 0  
## 29 Fapppylicious rgn 0  
## 30 FelixUK rgn 0  
## 31 jnenfrancis ALaurizen 0  
## 32 klosterbier ALaurizen 0  
## 33 rgn ALaurizen 0  
## 34 ALaurizen ALaurizen 3  
## 35 ROCKETLABS ALaurizen 0  
## 36 empireteam ALaurizen 44  
## 37 GreenLeafLabs ALaurizen 0  
## 38 optiman ALaurizen 1  
## 39 Fapppylicious ALaurizen 1  
## 40 FelixUK ALaurizen 0  
## 41 jnenfrancis ROCKETLABS 0  
## 42 klosterbier ROCKETLABS 3  
## 43 rgn ROCKETLABS 0  
## 44 ALaurizen ROCKETLABS 0  
## 45 ROCKETLABS ROCKETLABS 45  
## 46 empireteam ROCKETLABS 0  
## 47 GreenLeafLabs ROCKETLABS 0  
## 48 optiman ROCKETLABS 0  
## 49 Fapppylicious ROCKETLABS 0  
## 50 FelixUK ROCKETLABS 0  
## 51 jnenfrancis empireteam 0  
## 52 klosterbier empireteam 0  
## 53 rgn empireteam 0  
## 54 ALaurizen empireteam 0  
## 55 ROCKETLABS empireteam 0  
## 56 empireteam empireteam 76  
## 57 GreenLeafLabs empireteam 0  
## 58 optiman empireteam 0  
## 59 Fapppylicious empireteam 2  
## 60 FelixUK empireteam 0  
## 61 jnenfrancis GreenLeafLabs 0  
## 62 klosterbier GreenLeafLabs 0  
## 63 rgn GreenLeafLabs 0  
## 64 ALaurizen GreenLeafLabs 0  
## 65 ROCKETLABS GreenLeafLabs 0  
## 66 empireteam GreenLeafLabs 0  
## 67 GreenLeafLabs GreenLeafLabs 65  
## 68 optiman GreenLeafLabs 0  
## 69 Fapppylicious GreenLeafLabs 2  
## 70 FelixUK GreenLeafLabs 0  
## 71 jnenfrancis optiman 0  
## 72 klosterbier optiman 10  
## 73 rgn optiman 0  
## 74 ALaurizen optiman 0  
## 75 ROCKETLABS optiman 4  
## 76 empireteam optiman 3  
## 77 GreenLeafLabs optiman 0  
## 78 optiman optiman 31  
## 79 Fapppylicious optiman 0  
## 80 FelixUK optiman 0  
## 81 jnenfrancis Fapppylicious 0  
## 82 klosterbier Fapppylicious 0  
## 83 rgn Fapppylicious 0  
## 84 ALaurizen Fapppylicious 0  
## 85 ROCKETLABS Fapppylicious 0  
## 86 empireteam Fapppylicious 0  
## 87 GreenLeafLabs Fapppylicious 28  
## 88 optiman Fapppylicious 0  
## 89 Fapppylicious Fapppylicious 42  
## 90 FelixUK Fapppylicious 0  
## 91 jnenfrancis FelixUK 5  
## 92 klosterbier FelixUK 0  
## 93 rgn FelixUK 0  
## 94 ALaurizen FelixUK 0  
## 95 ROCKETLABS FelixUK 0  
## 96 empireteam FelixUK 0  
## 97 GreenLeafLabs FelixUK 0  
## 98 optiman FelixUK 0  
## 99 Fapppylicious FelixUK 0  
## 100 FelixUK FelixUK 51

## Table 6 - Sellers Prediction / Naive Bayesian Classification

## [1] "The accuracy is : 79.97 %"

Results show that the algorithm succeeds in predicting most of the sellers. Therefore, the accuracy is 79.97 %. Which is still a little bit less than with decision tree. However with more sellers (i.e more than 40 sellers for instance), this algorithm tends to be more accurate than the one based on decision tree method.

Later in the section *Prediction of profitability* , it will be discussed how to exploit at best this two values : the *creation date of ads* and the *number of sold products*.

**3. Text Mining**

One last method, but not least : Text mining. This is a fashion method since with the power of our computer, we are now able to perform wide analyses on *text data*. It can be used for text classification (determinating the topic, the tone, etc...), but here it will be used for predicting who has written the text (i.e sellers).

The main interest comparing to previous methods is that it is only based on the words used by the sellers. Thus, it is possible to identify the seller according to his *writting style* in other type websites, such as Social Network. What previous methods can not do because they are trained with variables specific to AlphaBay. Furthermore, there is no need to explain the interest of identifing in Social Network illegal sellers ...

The method consists in training the algortihm with words from the ad description of each seller. Then, it is tested with a part of the data which has not been used for training. The algorithm choosen for this method is *Support Vector Machine* but other alogrithms have been tested such as decision tree and random forest and results are similar. The result below has been obtained with a data containing the ads description from the 50 main sellers.

#----------------------------------------------------------------------  
# Algo to predict sellers given words from their text  
#-----------------------------------------------------------------------  
  
#https://journal.r-project.org/archive/2013/RJ-2013-001/RJ-2013-001.pdf  
  
TMPredictSeller <- function(TM\_nb\_sellers){  
   
   
 # Select all "Drugs & Chemicals" ads  
 matching\_vector <- c( str\_detect(data$category, "Drugs & Chemicals"))  
 new.data <- data[matching\_vector,]  
   
 # Random rows :  
 new.data <- new.data[sample(nrow(new.data),nrow(new.data),replace=FALSE), ]  
   
   
 # Handling : seller  
 tab\_sel <- table(new.data$seller)  
 tab\_sel <- sort(tab\_sel, decreasing=TRUE) # Sorting (biggest in first)  
 tab\_sel <- tab\_sel[1:TM\_nb\_sellers] # Taking only the most important : main sellers  
 name\_sel <- names(tab\_sel)  
   
 # New data keeping only the main sellers  
 new.data <-subset(new.data, seller %in% name\_sel)   
   
 new.data$seller <- factor(new.data$seller)  
   
 # CREATE THE DOCUMENT-TERM MATRIX  
 doc\_matrix <- create\_matrix(new.data$ad, language="english", removeNumbers=TRUE,  
 stemWords=TRUE, removeSparseTerms=.998)  
   
 container <- create\_container(doc\_matrix, new.data$seller, trainSize=1:round(0.75\*nrow(new.data)),  
 testSize=round(0.75\*nrow(new.data)+1,0):nrow(new.data), virgin=FALSE)  
   
   
 SVM <- train\_model(container,"SVM")  
   
 SVM\_CLASSIFY <- classify\_model(container, SVM)  
  
   
   
 test <- new.data[round(0.75\*nrow(new.data)+1,0):nrow(new.data),]  
 # Comparison between the result and the prediction (prediction in colunm)  
 conf <- table(test[,match("seller",names(test))],SVM\_CLASSIFY$SVM\_LABEL)  
   
 # Accuracy :  
 acc <- round((sum(diag(conf)) / sum(conf)\*100),2)  
   
 # Display   
 conf <- data.frame(conf)  
 names(conf) <- c("Sellers","Prediction","Freq")  
 #print(conf)  
   
 return (acc)  
}  
accTM <- TMPredictSeller(TM\_nb\_sellers)  
sprintf("The accuracy is : %.2f %%", accTM)

## [1] "The accuracy is : 93.95 %"

The accuracy is 93.95 % which is very high. For more sellers, (which own enough ads, in order to have some of them in the training dataset) the accuracy decreased but it is still very high. For instance for 150 main sellers it is around 90%.

However, this accuracy has to be taken with precaution. Indeed, most of the ads descriptions from one seller are very similar. The reason is that they create various ads for the same product with different quantities. Thus, ads description are almost the same. That is why, it is not surprising that the accuracy is so high.

Anyway, that show us that the algorithm is working properly and it can be used in order to find hidden identities of one seller in AlphaBay or in other Website and eventually discover his real identity.

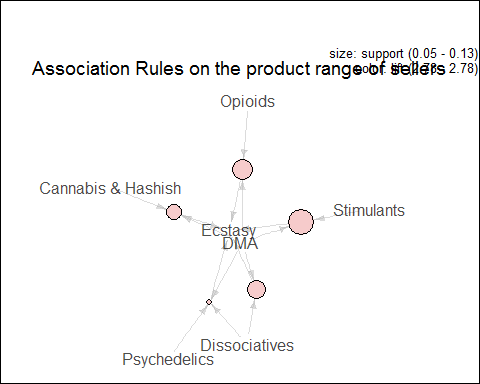
## - Clustering of drugs

Secondly, one can wonder if there were links between some drugs. That is to say, if this is possible to cluster some drugs.

To do so, firstly, a new data frame has been created with by rows sellers and by columns different sub-categories of *"Drugs & Chemicals"*. In each cell, value is True or False if the dealer has already sold something in this sub-category or not. Then, *Apriori algorithm* of *Association Rules* has been used. One drug has been selected that must be in the itemset. Here it is Ecstasy and results can be seen below.

#--------------------------------------------------------  
# Association Rules - Apriori algorithm   
# Guess if this dealer is selling this drugs  
#--------------------------------------------------------  
  
AssRSellersCat <- function(){  
   
 #------------------------------  
 # New Data frame for analysis  
 #------------------------------  
   
 # Select all ads of "Drugs & Chemicals"  
 matching\_vector <- c( str\_detect(data$category, "Drugs & Chemicals"))  
 data\_drugs <- data[matching\_vector, ]  
   
 # Handling of this categories  
 # Regular expression for spliting the categories  
 regex <- "/(.\*)/(.\*)/(.\*)"  
 cat\_exp <- str\_match(data\_drugs$category, regex)  
 data\_drugs$category <- cat\_exp[,3]  
   
 # Get rid of category "Other"  
 matching\_vector <- !c( str\_detect(data\_drugs$category, "Other"))  
 data\_drugs <- data\_drugs[matching\_vector, ]  
   
 # List all the sellers  
 sellers <-sort(table(data\_drugs$seller), decreasing = TRUE)  
 sellers <- sellers[ sellers != "Null"]  
 sellers <- sellers [1:100]  
   
 #List all categories concerning drugs  
 list\_category <- table(data\_drugs[,"category"])  
 list\_cat\_drugs <- list\_category [ list\_category != 0]  
   
# Step 1 : initialise a data.frame with the information of the first seller  
   
 # Select all categories of the seller   
 matching\_vector <- c( str\_detect(data$seller, names(sellers)[1]))  
 cat\_seller <-summary(data.frame(data[matching\_vector, "category"]))  
   
 # Loop which creates a boolean vector which tells if the seller sells stuffs in each category   
 bool\_cat <-c()  
 bool\_vec <-c()  
 for( i in 1: length(list\_cat\_drugs)){  
 bool\_vec <- str\_detect(cat\_seller, names(list\_cat\_drugs)[i])  
   
 bool <- FALSE   
 for(j in 1:length(bool\_vec)){  
 bool <- bool || bool\_vec[j]  
 }  
 bool\_cat[i] <- bool  
   
 }  
   
 cat\_seller.data <- t(data.frame(bool\_cat))  
 colnames(cat\_seller.data) <- names(list\_cat\_drugs)  
   
#Step 2 : Do the same for the other sellers  
   
 for(k in 2 : length(sellers)){  
 # Select all categories of the seller   
 matching\_vector <- c( str\_detect(data$seller, names(sellers)[k]))  
 cat\_seller <-summary(data.frame(data[matching\_vector, "category"]))  
   
 # Loop which creates a boolean vector which tells if the seller sells stuffs in each category   
 bool\_cat <-c()  
 bool\_vec <-c()  
 for( i in 1: length(list\_cat\_drugs)){  
 bool\_vec <- str\_detect(cat\_seller, names(list\_cat\_drugs)[i])  
   
 bool <- FALSE   
 for(j in 1:length(bool\_vec)){  
 bool <- bool || bool\_vec[j]  
 }  
 bool\_cat[i] <- bool  
   
 }  
   
 cat\_seller.data <- rbind(cat\_seller.data,bool\_cat)  
   
 }  
   
 rownames(cat\_seller.data)<- names(sellers)  
  
 #-------------------------  
 # Ass Rules  
 #-------------------------  
   
 # Association Rules with rhs containing "Ecstasy" only  
 rules <- apriori(cat\_seller.data,  
 parameter = list(minlen=2, supp=0.05, conf=0.8),  
 appearance = list(rhs=c("Ecstasy"),default="lhs"),  
 control = list(verbose=F))  
   
 rules.sorted <- sort(rules, by="lift")  
   
 rules.sorted@quality$support <- round(rules.sorted@quality$support, 3)  
 rules.sorted@quality$confidence <- round(rules.sorted@quality$confidence, 2)  
 rules.sorted@quality$lift <- round(rules.sorted@quality$lift, 2)  
   
 arules::inspect(rules.sorted, linebreak = TRUE)  
   
 cat("\n")  
 cat(" ")  
 compTab <- TableCaption(compTab, "Cluster of drugs / Association Rules")  
   
 # Plot graph of rules  
 plot(rules.sorted[1:5], method="graph", control=list(type="items"),main ="")  
 mtext("Association Rules on the product range of sellers" , cex = 1.2)  
   
 # Frame  
 box(which = "outer", lty = "solid")  
  
 return(compTab)  
   
}  
  
compTab <- AssRSellersCat()

## lhs rhs support confidence lift  
## [1] {Cannabis & Hashish,   
## DMA } => {Ecstasy} 0.09 1.00 2.78  
## [2] {Dissociatives,   
## DMA } => {Ecstasy} 0.10 1.00 2.78  
## [3] {DMA ,   
## Opioids} => {Ecstasy} 0.11 1.00 2.78  
## [4] {DMA ,   
## Stimulants} => {Ecstasy} 0.13 1.00 2.78  
## [5] {Dissociatives,   
## DMA ,   
## Psychedelics} => {Ecstasy} 0.05 1.00 2.78  
## [6] {DMA ,   
## Psychedelics,   
## Stimulants} => {Ecstasy} 0.06 1.00 2.78  
## [7] {Cannabis & Hashish,   
## Dissociatives,   
## DMA } => {Ecstasy} 0.05 1.00 2.78  
## [8] {Cannabis & Hashish,   
## DMA ,   
## Opioids} => {Ecstasy} 0.06 1.00 2.78  
## [9] {Cannabis & Hashish,   
## DMA ,   
## Stimulants} => {Ecstasy} 0.07 1.00 2.78  
## [10] {Dissociatives,   
## DMA ,   
## Opioids} => {Ecstasy} 0.05 1.00 2.78  
## [11] {Dissociatives,   
## DMA ,   
## Stimulants} => {Ecstasy} 0.09 1.00 2.78  
## [12] {DMA ,   
## Opioids,   
## Stimulants} => {Ecstasy} 0.08 1.00 2.78  
## [13] {Cannabis & Hashish,   
## Dissociatives,   
## DMA ,   
## Stimulants} => {Ecstasy} 0.05 1.00 2.78  
## [14] {Cannabis & Hashish,   
## DMA ,   
## Opioids,   
## Stimulants} => {Ecstasy} 0.05 1.00 2.78  
## [15] {Dissociatives,   
## DMA ,   
## Opioids,   
## Stimulants} => {Ecstasy} 0.05 1.00 2.78  
## [16] {DMA } => {Ecstasy} 0.19 0.95 2.64  
## [17] {DMA ,   
## Psychedelics} => {Ecstasy} 0.11 0.92 2.55  
## [18] {Dissociatives,   
## Psychedelics} => {Ecstasy} 0.09 0.82 2.27  
## [19] {Cannabis & Hashish,   
## Opioids} => {Ecstasy} 0.09 0.82 2.27  
## [20] {Psychedelics,   
## Stimulants} => {Ecstasy} 0.12 0.80 2.22  
## [21] {Cannabis & Hashish,   
## Dissociatives} => {Ecstasy} 0.08 0.80 2.22  
## [22] {Cannabis & Hashish,   
## Opioids,   
## Stimulants} => {Ecstasy} 0.08 0.80 2.22  
##   
## Table 7 - Cluster of drugs / Association Rules



## Figure 16 - Cluster of drugs / Association Rules

The algorithm succeeds in finding some rules in the data frame. That means that some drugs can effectively be clustered. The support is between 5% and 15% so it is frequent to have these itemsets. Moreover the confidence is more than 80%. In other words, if there is the itemset on the left we are most likely to have the drug on the right.

These rules can be interpreted as follows : sellers often deal more than one product. And these products can be clustered by type.

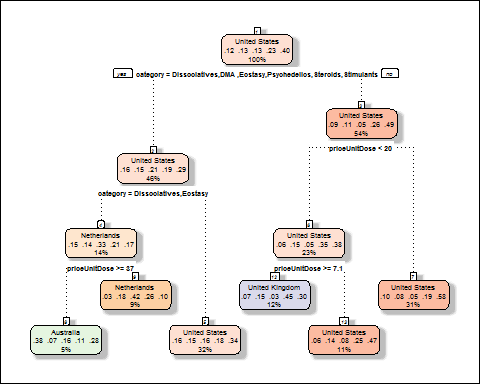
## - Prediction of ads origin

After trying to make predictions on sellers (cf. [Sellers Predictions](#sellers-predictions)), Prognoses on the origin of the ads have been made using Decision Tree method and Association Rules.

**1. Decision tree**

Looking at prices and categories, a *Decision Tree* has been created in order to predict the origin. In the same way the algorithm has been trained on one half of the data, and predictions made on the other half.

#----------------------------------------------------------------------  
# Decision tree - CART algorithm  
# Prediction of the country knowing the price / category  
#-----------------------------------------------------------------------  
  
DTorigin <- function(){  
   
 #-----------------  
 # New Data   
 #-----------------  
   
 # Select all "Drugs & Chemicals" ads  
 matching\_vector <- c( str\_detect(data$category, "Drugs & Chemicals"))  
 dectree.data <- data[matching\_vector,]  
   
 # Select the column of the data that are interesting for the tree  
 # ie removing colunm like "id" or "url" that don't give any informations  
 dectree.data <- subset(dectree.data, select=c(origin,category,priceUnitDose))   
 # Subset : choose the colunm that you want  
   
 # Handling : column categorie  
 # Regular expression for spliting the categories  
 regex <- "/(.\*)/(.\*)/(.\*)"  
 cat <- str\_match(dectree.data$category, regex)  
 dectree.data$category <- cat[,3] # keep only the second part  
   
 # Handling : country  
 dectree.data <- dectree.data[which(dectree.data$origin != "Worldwide"),]  
 tab\_coun <- table(dectree.data$origin)  
 tab\_coun <- sort(tab\_coun, decreasing=TRUE) # Sorting (biggest in first)  
 tab\_coun <- tab\_coun[1:5] # Taking only the most important : main sellers  
 name\_coun <- names(tab\_coun)  
 # New data keeping only the main dealers  
 dectree.data <-subset(dectree.data, origin %in% name\_coun)   
   
   
 # Random rows :  
 dectree.data <- dectree.data[sample(nrow(dectree.data),nrow(dectree.data),replace=FALSE), ]  
   
 #---------------------  
 # Decision tree  
 #---------------------  
   
 # Factor  
 dectree.data$origin <- factor(dectree.data$origin)  
   
 # Half of the data for making the decision tree  
 train <- dectree.data[1:(floor(nrow(dectree.data))/2),]  
   
 # Creation of the tree  
 tree <- rpart(origin ~.,data=train, method="class")   
   
 # Plot  
 fancyRpartPlot(tree, sub="")  
   
 # Frame  
 box(which = "outer", lty = "solid")  
   
 #--------------------  
 # Prediction  
 #--------------------  
   
 # The other half for the prediction  
 test <- dectree.data[(floor(nrow(dectree.data)/2)+1):nrow(dectree.data),]  
   
 # Making prediction  
 pred <- predict(tree,test,type="class")  
   
 # Analysis:  
   
 # Comparison between the result and the prediction (prediction in colunm)  
 conf <- table(test[,match("origin",names(test))],pred)  
   
 # Accuracy :  
 acc <- round((sum(diag(conf)) / sum(conf)\*100),2)  
   
 print(conf)   
 cat("\n")  
 cat(" ")  
 compTab <- TableCaption(compTab, "Origins Prediction / Decision Tree method")  
   
 Result <- c(acc,compTab)  
   
 return(Result)  
}  
  
ResultDT2 <- DTorigin()



## pred  
## Australia Germany Netherlands United Kingdom  
## Australia 93 0 13 46  
## Germany 16 0 108 110  
## Netherlands 51 0 177 32  
## United Kingdom 21 0 106 289  
## United States 66 0 52 215  
## pred  
## United States  
## Australia 478  
## Germany 478  
## Netherlands 383  
## United Kingdom 838  
## United States 1833  
##   
## Table 8 - Origins Prediction / Decision Tree method

## Figure 17 - Origins prediction / Decision Tree

Results don't seem to be very good, the accuracy is 44.26 % which is lower than previously. It turns out that without sellers, which give a lot of information on the origin, prognoses are not very reliable.

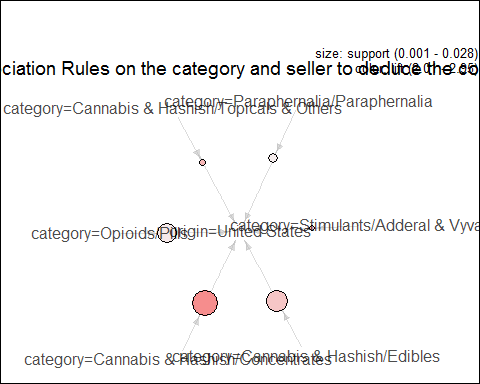
Now let's see if correlations can be found between categories and origins.

**2. Rules to determine cateogries of ads coming from United States**

To do so, *Association Rules Method* has been run with 2 variables : category and origin. Thus, one may be able to make a link with predictions of above decision tree. One country has been fixed, here United States.

#--------------------------------------------------------  
# Association Rules - Apriori algorithm   
# Guess if United States is the origin of the ad  
#--------------------------------------------------------  
  
AssROriginSellerCat <- function(){   
 #------------------------------  
 # New Data frame for analysis  
 #------------------------------  
  
 # Select all ads of "Drugs & Chemicals"  
 matching\_vector <- c( str\_detect(data$category, "Drugs & Chemicals"))  
 data\_drugs <- data[matching\_vector, ]  
  
   
 # Select some columns  
 asso.data <- subset(data\_drugs, select = c(origin,category))  
   
 # Get rid of the first part of the category name "/Drugs & Chemicals/"  
 asso.data$category <- gsub(pattern = "/Drugs & Chemicals/", replacement = "", asso.data$category)  
   
 asso.data$origin <- factor(asso.data$origin)  
 asso.data$category <-factor(asso.data$category)  
 # asso.data$seller <- factor(asso.data$seller)  
   
 # Association Rules with rhs containing one given country only  
 rules <- apriori(asso.data,  
 parameter = list(minlen=2, supp=0.0005, conf=0.5),  
 appearance = list(rhs=c("origin=United States"),default="lhs"),  
 control = list(verbose=F))  
   
 rules.sorted <- sort(rules, by="lift")  
   
   
 rules.sorted@quality$support <- round(rules.sorted@quality$support, 3)  
 rules.sorted@quality$confidence <- round(rules.sorted@quality$confidence, 2)  
 rules.sorted@quality$lift <- round(rules.sorted@quality$lift, 2)  
   
 arules::inspect(rules.sorted)  
   
 cat("\n")  
 cat(" ")  
 compTab <- TableCaption(compTab, "Origin Prediction / Association Rules")  
   
 # Plot graph of rules  
 plot(rules.sorted, method="graph", control=list(type="items"),main ="")  
 mtext("Association Rules on the category and seller to deduce the country" , cex = 1.2)  
   
 # Frame  
 box(which = "outer", lty = "solid")  
  
 return(compTab)  
}  
  
compTab <- AssROriginSellerCat()

## lhs rhs support confidence lift  
## [1] {category=Cannabis & Hashish/Concentrates} => {origin=United States} 0.028 0.76 2.95  
## [2] {category=Cannabis & Hashish/Topicals & Others} => {origin=United States} 0.002 0.68 2.63  
## [3] {category=Stimulants/Adderal & Vyvanse} => {origin=United States} 0.001 0.67 2.58  
## [4] {category=Cannabis & Hashish/Edibles} => {origin=United States} 0.023 0.65 2.51  
## [5] {category=Opioids/Pills} => {origin=United States} 0.020 0.53 2.07  
## [6] {category=Paraphernalia/Paraphernalia} => {origin=United States} 0.006 0.52 2.01  
##   
## Table 9 - Origin Prediction / Association Rules



## Figure 18 - Origin Prediction / Association Rules

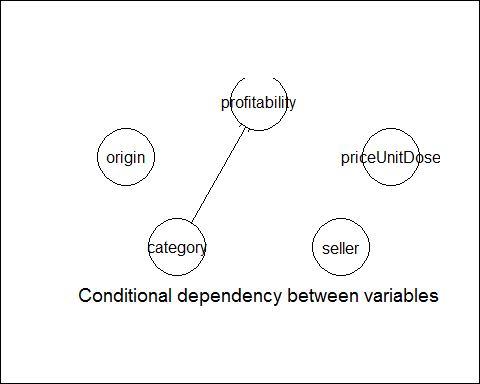
The results show that when there is an ad of the category on the left, it is likely to come from United States with a confidence higher than 50%. Thus, ads from United States are often on Cannabis & Hashish, this can be easily confirmed by plotting a pie chart of United States exportations, as in the section before.

It is striking to see that Cannabis & Hashish seems to be the main rule. This can be explained by the legalisation of Cannibis in some states. Thus, the sales of these products is easy in United States and may interest people from other countries where they are not legalized.

## - Prediction of ads profitability

Secondly, it worths to predict the profitability of an ad. That is to say, given an ad to predict if it will be sold a lot or not. Each ad have information on the category, origin, seller, price and a rate of profitability. This rate is caculated by dividing the number of product sold by the current lifetime of the ad and by times 30 to have a number of ads sold monthly. Bayesian Neural Network algorithm has beed accomplished on this new data and the results obtained are below.

#----------------------------------------------------------------------  
# Bayesian Network  
# with seller / origin / price / category / timestamp / sold\_since / product\_sold  
#-----------------------------------------------------------------------  
  
BayesNet <- function(){  
 #-----------------  
 # New Data   
 #-----------------  
   
 # Select all "Drugs & Chemicals" ads  
 matching\_vector <- c( str\_detect(data$category, "Drugs & Chemicals"))  
 bayesian.data <- data[matching\_vector,]  
   
   
 # Select the column of the data that are interesting  
 # ie removing colunm like "id" or "url" that don't give any information  
 bayesian.data <- subset(bayesian.data, select=c(origin,category,seller,priceUnitDose, products\_sold, sold\_since, timestamp ))  
 # Subset : choose the colunm that you want  
   
 # Handling : column category  
 # Regular expression for spliting the categories  
 regex <- "/(.\*)/(.\*)/(.\*)"  
 cat <- str\_match(bayesian.data$category, regex)  
 bayesian.data$category <- cat[,3] # keep only the second part  
   
 #Get rid of lines with Null as products\_sold value  
 bayesian.data <- bayesian.data[!is.element(bayesian.data$products\_sold, "NULL"),]  
   
 #Convert products\_sold to numeric and discretize it  
 bayesian.data$products\_sold <- as.numeric(as.character(bayesian.data$products\_sold))  
   
 #Given timestamp and sold\_since calculate the lifetime of the ad  
 bayesian.data$sold\_since <- as.Date(bayesian.data$sold\_since)  
 bayesian.data$timestamp <- as.Date(bayesian.data$timestamp)  
 bayesian.data$timestamp <- bayesian.data$timestamp - bayesian.data$sold\_since  
 bayesian.data$timestamp <- as.numeric(bayesian.data$timestamp)  
   
 # 1 day on the market at least  
 bayesian.data <- bayesian.data[which(bayesian.data$timestamp > 0),]  
   
 #Calculate profitability  
 bayesian.data$products\_sold <- bayesian.data$products\_sold / bayesian.data$timestamp \* 30  
 names(bayesian.data)[match("products\_sold",names(bayesian.data))] <- "profitability"  
   
 #Discretize profitability  
 nbCategory <- 5  
 bayesian.data$profitability <- arules::discretize(bayesian.data$profitability, method="frequency", categories = nbCategory)  
   
 bayesian.data <- subset(bayesian.data, select= -c(sold\_since, timestamp))  
   
 #Convert variables to factor  
 bayesian.data$category <- as.factor(bayesian.data$category)  
 bayesian.data$seller <- as.factor(bayesian.data$seller)  
 bayesian.data$origin <- as.factor(bayesian.data$origin)  
   
 #Get rid of lines with NA as products\_sold value  
 bayesian.data <- bayesian.data[!is.element(bayesian.data$profitability, NA),]  
   
 #---------------------  
 # Bayesian Network  
 #---------------------  
   
 res <- hc(bayesian.data)  
 plot(res)  
   
 fittedbn <- bn.fit(res, data = bayesian.data)  
   
 prob.data <- data.frame(fittedbn$profitability$prob)  
 colnames(prob.data) <- c("Profitability", "Category", "Probability")  
 print(prob.data)  
   
 compTab <- TableCaption(compTab, "Categories Profitability")  
   
 #Handling interval  
 interv <- levels(bayesian.data$profitability)  
 interv <- unlist(strsplit(interv, ","))  
 interv <- gsub(pattern = "[^0-9.]\*", replacement = "", interv)  
 interval <- data.frame(interv[seq(1, length(interv), 2)],interv[seq(2, length(interv), 2)])  
 colnames(interval) <- c("left", "right")  
 interval$left <- as.numeric(levels(interval$left))  
 interval$right <- as.numeric(levels(interval$right))  
   
 expectancy <- c()  
  
 #Calculate expectancy for each category   
 for(i in 0:(length(table(bayesian.data$category))-1)){  
 left <- 0  
 right <- 0  
 for(j in 1:nbCategory){  
 left<-left + interval$left[j] \* fittedbn$profitability$prob[i\*nbCategory + j]  
 right<-right + interval$right[j] \* fittedbn$profitability$prob[i\*nbCategory + j]  
 }  
 expectancy[i+1] <- paste("[", round(left,2) , "," , round(right,2) , "]")  
 }  
 affichage <-data.frame(names(table(bayesian.data$category)),expectancy)  
 colnames(affichage) <- c("Category", "Expectancy")  
 print(affichage)  
   
 compTab <- TableCaption(compTab, "Expectancies of category Profitability")  
  
 # Frame  
 box(which = "outer", lty = "solid")  
 mtext("Conditional dependency between variables" , cex = 1.2,side = 1)  
   
 return(compTab)  
   
}  
  
compTab <- BayesNet()



## Profitability Category Probability  
## 1 [0.0000, 0.0896) Benzos 0.466666667  
## 2 [0.0896, 0.8795) Benzos 0.296969697  
## 3 [0.8795, 3.1915) Benzos 0.042424242  
## 4 [3.1915, 9.1691) Benzos 0.060606061  
## 5 [9.1691,480.0000] Benzos 0.133333333  
## 6 [0.0000, 0.0896) Cannabis & Hashish 0.151891892  
## 7 [0.0896, 0.8795) Cannabis & Hashish 0.183783784  
## 8 [0.8795, 3.1915) Cannabis & Hashish 0.220810811  
## 9 [3.1915, 9.1691) Cannabis & Hashish 0.233513514  
## 10 [9.1691,480.0000] Cannabis & Hashish 0.210000000  
## 11 [0.0000, 0.0896) Dissociatives 0.370253165  
## 12 [0.0896, 0.8795) Dissociatives 0.199367089  
## 13 [0.8795, 3.1915) Dissociatives 0.155063291  
## 14 [3.1915, 9.1691) Dissociatives 0.151898734  
## 15 [9.1691,480.0000] Dissociatives 0.123417722  
## 16 [0.0000, 0.0896) DMA 0.342857143  
## 17 [0.0896, 0.8795) DMA 0.157142857  
## 18 [0.8795, 3.1915) DMA 0.242857143  
## 19 [3.1915, 9.1691) DMA 0.171428571  
## 20 [9.1691,480.0000] DMA 0.085714286  
## 21 [0.0000, 0.0896) Ecstasy 0.323928945  
## 22 [0.0896, 0.8795) Ecstasy 0.117032393  
## 23 [0.8795, 3.1915) Ecstasy 0.077324974  
## 24 [3.1915, 9.1691) Ecstasy 0.210031348  
## 25 [9.1691,480.0000] Ecstasy 0.271682341  
## 26 [0.0000, 0.0896) Opioids 0.001203369  
## 27 [0.0896, 0.8795) Opioids 0.204572804  
## 28 [0.8795, 3.1915) Opioids 0.297232250  
## 29 [3.1915, 9.1691) Opioids 0.259927798  
## 30 [9.1691,480.0000] Opioids 0.237063779  
## 31 [0.0000, 0.0896) Other 0.537815126  
## 32 [0.0896, 0.8795) Other 0.226890756  
## 33 [0.8795, 3.1915) Other 0.117647059  
## 34 [3.1915, 9.1691) Other 0.056722689  
## 35 [9.1691,480.0000] Other 0.060924370  
## 36 [0.0000, 0.0896) Paraphernalia 0.074468085  
## 37 [0.0896, 0.8795) Paraphernalia 0.638297872  
## 38 [0.8795, 3.1915) Paraphernalia 0.196808511  
## 39 [3.1915, 9.1691) Paraphernalia 0.058510638  
## 40 [9.1691,480.0000] Paraphernalia 0.031914894  
## 41 [0.0000, 0.0896) Prescription 0.000000000  
## 42 [0.0896, 0.8795) Prescription 0.000000000  
## 43 [0.8795, 3.1915) Prescription 0.327272727  
## 44 [3.1915, 9.1691) Prescription 0.466666667  
## 45 [9.1691,480.0000] Prescription 0.206060606  
## 46 [0.0000, 0.0896) Psychedelics 0.384871155  
## 47 [0.0896, 0.8795) Psychedelics 0.246882793  
## 48 [0.8795, 3.1915) Psychedelics 0.174563591  
## 49 [3.1915, 9.1691) Psychedelics 0.115544472  
## 50 [9.1691,480.0000] Psychedelics 0.078137988  
## 51 [0.0000, 0.0896) Steroids 0.000000000  
## 52 [0.0896, 0.8795) Steroids 0.011904762  
## 53 [0.8795, 3.1915) Steroids 0.404761905  
## 54 [3.1915, 9.1691) Steroids 0.309523810  
## 55 [9.1691,480.0000] Steroids 0.273809524  
## 56 [0.0000, 0.0896) Stimulants 0.113981763  
## 57 [0.0896, 0.8795) Stimulants 0.183130699  
## 58 [0.8795, 3.1915) Stimulants 0.174012158  
## 59 [3.1915, 9.1691) Stimulants 0.202127660  
## 60 [9.1691,480.0000] Stimulants 0.326747720  
## 61 [0.0000, 0.0896) Tobacco 0.416267943  
## 62 [0.0896, 0.8795) Tobacco 0.234449761  
## 63 [0.8795, 3.1915) Tobacco 0.177033493  
## 64 [3.1915, 9.1691) Tobacco 0.100478469  
## 65 [9.1691,480.0000] Tobacco 0.071770335  
## 66 [0.0000, 0.0896) Weight Loss 0.390697674  
## 67 [0.0896, 0.8795) Weight Loss 0.344186047  
## 68 [0.8795, 3.1915) Weight Loss 0.144186047  
## 69 [3.1915, 9.1691) Weight Loss 0.051162791  
## 70 [9.1691,480.0000] Weight Loss 0.069767442  
## Table 10 - Categories Profitability Category Expectancy  
## 1 Benzos [ 1.48 , 30.75 ]  
## 2 Cannabis & Hashish [ 2.88 , 114.89 ]  
## 3 Dissociatives [ 1.77 , 74.75 ]  
## 4 DMA [ 1.56 , 84.02 ]  
## 5 Ecstasy [ 3.24 , 103.68 ]  
## 6 Opioids [ 3.28 , 128.07 ]  
## 7 Other [ 0.86 , 28.41 ]  
## 8 Paraphernalia [ 0.71 , 29.57 ]  
## 9 Prescription [ 3.67 , 226.93 ]  
## 10 Psychedelics [ 1.26 , 56.99 ]  
## 11 Steroids [ 3.86 , 152.38 ]  
## 12 Stimulants [ 3.81 , 100.74 ]  
## 13 Tobacco [ 1.16 , 49.7 ]  
## 14 Weight Loss [ 0.96 , 26 ]  
## Table 11 - Expectancies of category Profitability

## Figure 19 - Variable Dependencies / Bayesian Neural Network

Neural Network shows that the profitability is conditionnaly dependant to category. That is to say category has a significant impact on profitability. Conditional probabilities are shown in the array. It is surprising that price and seller have no impacts on profitability.

Furthermore, expectancy of each event profitability X given category Y has been calculated. Thus, we can find the most profitable products to sold. Apparently it seems to be presciption, steroid and opioids. May be they are more profitable because they are not "common products" (and still wanted) contrary to Cannabis or Cocaine which can be found more easily on the street.

# - References

[1] Baravalle A, Lopez MS, Lee SW. " Mining the dark web: Drugs and fake ids ".

[2] Baraniuk C. " BBC - alphabay and hansa dark web markets shut down " 20/07/2017. [Online]. [Accessed : 24/07/2017]. Available: <http://www.bbc.co.uk/news/technology-40670010>.

[3] Kopan T. " CNN - doj announces takedown of dark web market alphabay " 20/07/2017. [Online]. [Accessed : 24/07/2017]. Available: <http://edition.cnn.com/2017/07/20/politics/doj-takes-down-dark-web-marketplace-alphabay/index.html>.

[4] " Google trends - explore: AlphaBay, dream market ". [Online] [Accessed : 22/07/2017]. Available: <https://trends.google.co.uk/trends/explore?date=2015-01-01%202017-07-01&q=alphabay,Dream%20Market>.

[5] " Google trends - explore: AlphaBay ". [Online] [Accessed : 22/07/2017]. Available: <https://trends.google.co.uk/trends/explore?date=2014-12-01%202017-07-01&q=alphabay>.

[6] " Cocaine - trafficking and supply (eu drug markets report) ". Paragraph: Modes of Transport [Online]. [Accessed : 24/07/2017]. Available: <http://www.emcdda.europa.eu/publications/eu-drug-markets/2016/online/cocaine/trafficking-and-supply>.

[7] " DrugWise - how much do drugs cost? ". [Online] [Accessed : 12/07/2017]. Available: <http://www.drugwise.org.uk/how-much-do-drugs-cost/>.

[8] " The telegraph - the cost of street drugs in britain. ". [Online] [Accessed : 12/07/2017]. Available: <http://www.telegraph.co.uk/news/uknews/crime/11346133/The-cost-of-street-drugs-in-Britain.html>.

[9] " RehabCenters.net - the average cost of illegal street drugs. ". [Online] [Accessed : 12/07/2017]. Available: <http://www.rehabcenter.net/the-average-cost-of-illegal-drugs-on-the-street/>.

[10] " Canada.com - what illegal drugs cost on the street around the world. ". [Online] [Accessed : 12/07/2017]. Available: <http://o.canada.com/business/interactive-what-illegal-drugs-cost-on-the-street-around-the-world>.

[11] " The student pocket guide - facts about drugs. ". [Online] [Accessed : 12/07/2017]. Available: <http://www.thestudentpocketguide.com/2012/01/student-life/health-and-relationships/facts-about-drugs/>.

[12] " R markdown from r studio - r markdown tutorial " . [Online] [Accessed : 29/06/2017]. Available: <http://rmarkdown.rstudio.com/lesson-1.html>.

[13] Tan P-N, Steinback M, Kumar V. " Introduction to data mining ". Addison Wesley; 2006.