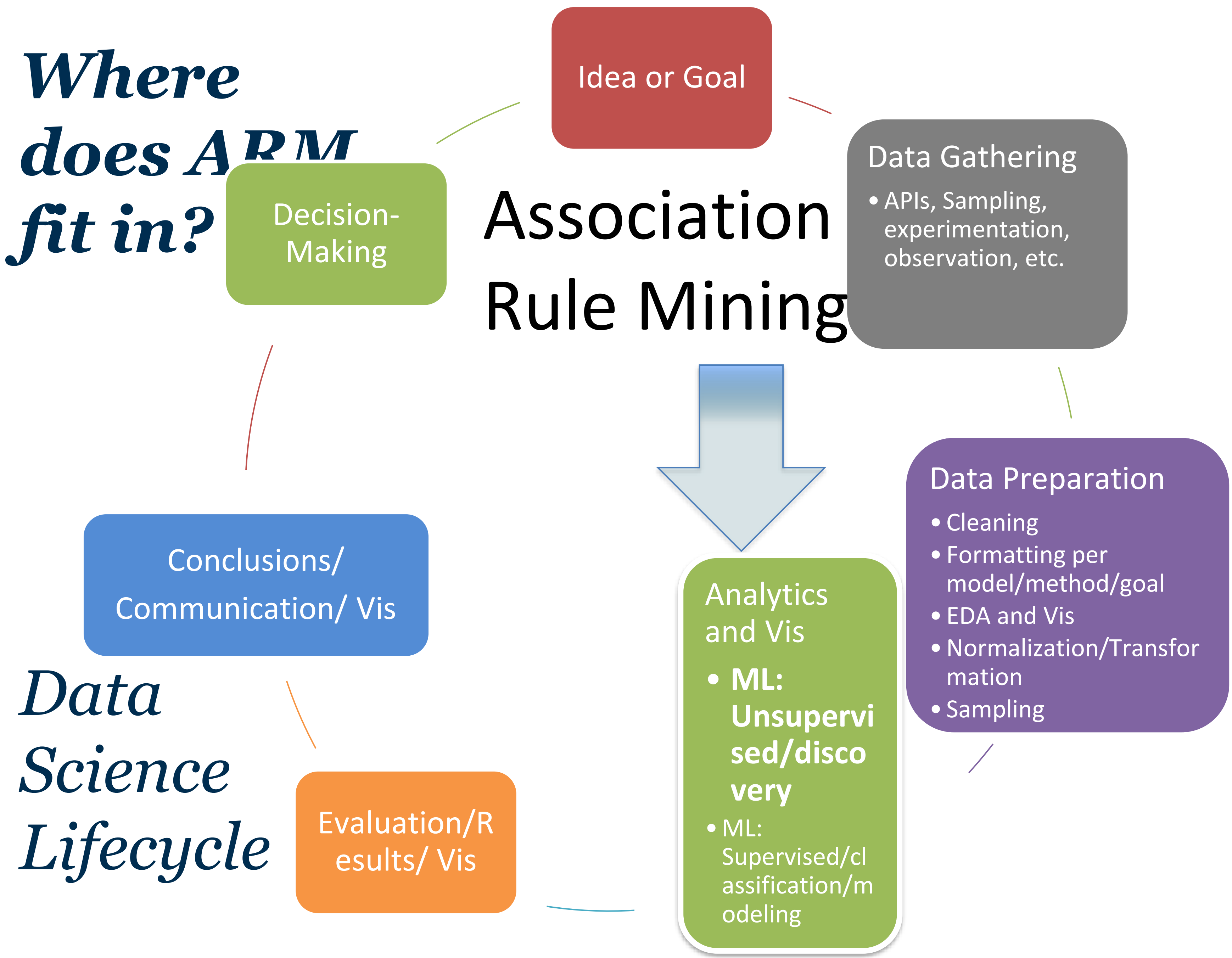
```
> inspect(SortedRules_lift[1:20])
     lhs                    rhs     support   confidence coverage  lift count
[1]  {mainland}          => {fresh} 0.1764706 1          0.1764706 4.25 3
[2]  {mainland}          => {roast} 0.1764706 1          0.1764706 4.25 3
[3]  {fresh}             => {roast} 0.2352941 1          0.2352941 4.25 4
[4]  {roast}             => {fresh} 0.2352941 1          0.2352941 4.25 4
[5]  {fresh,mainland}    => {roast} 0.1764706 1          0.1764706 4.25 3
[6]  {mainland,roast}    => {fresh} 0.1764706 1          0.1764706 4.25 3
[7]  {dear,mainland}     => {fresh} 0.1764706 1          0.1764706 4.25 3
[8]  {lover,mainland}    => {fresh} 0.1764706 1          0.1764706 4.25 3
[9]  {coffee,mainland}   => {fresh} 0.1764706 1          0.1764706 4.25 3
[10] {dear,mainland}     => {roast} 0.1764706 1          0.1764706 4.25 3
```

Introduction to ***Association Rule Mining (ARM)***

-and –

Thinking Outside the Basket with Twitter

Dr. Ami Gates



Where does ARM fit in?
Idea or Goal
Data Gathering
APIs, Sampling, experimentation, observation, etc.
Data Preparation
Cleaning
Formatting per model/method/goal
EDA and Vis
Normalization/Transformation
Sampling
Association Rule Mining
Analytics and Vis
ML: Unsupervised/discovery
ML: Supervised/classification/modeling
Evaluation/Results/ Vis
Conclusions/ Communication/ Vis
Decision-Making
Data Science Lifecycle

Dr. Gates Talks on ARM

Talk 1: Association Rule Mining on Tweets in R

https://www.youtube.com/watch?v=eOOhn9CX2qU

Talk 2: Association Rule Mining on Twitter and Building Network Visualizations in R

https://www.youtube.com/watch?v=AIK_FRzyUEE

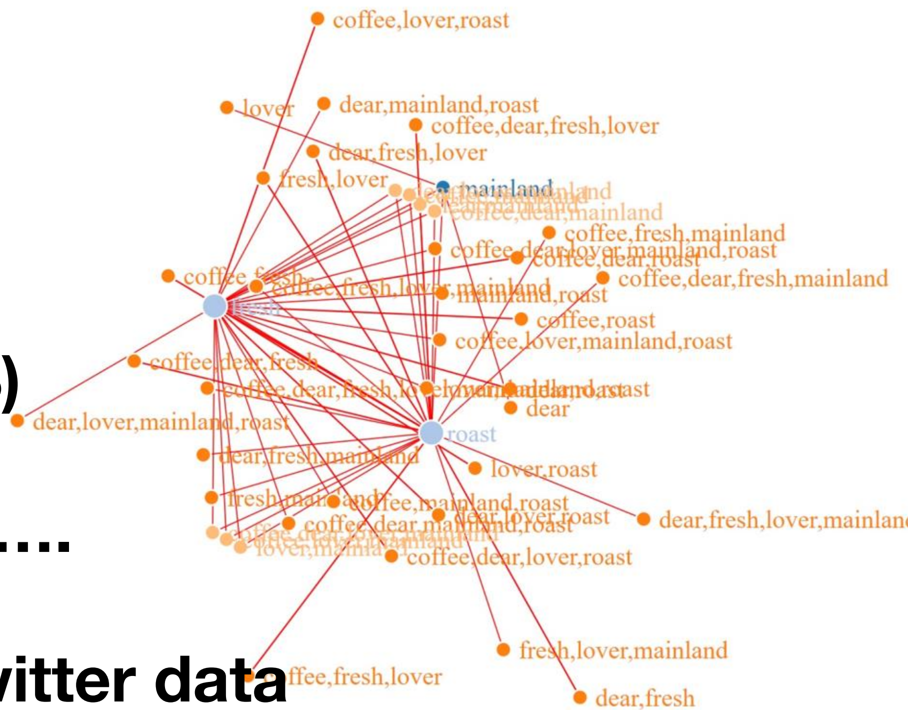
What is Association Rule Mining (ARM)

Unsupervised Learning – no labels – discovery

Evaluates "**transactions**" for correlations/associations.

Most common example:

Market Basket (Kumar, 2008)



Many applications, including….

- Image identification
- Text Analytics: like **Twitter data**
- Click streams
- Bio data – binding sites, AA's in proteins

Thinking About ARM (with Tweets and Coffee)

#coffee

Tweet 1:

The **coffee festival** was **delicious**. **Loved** it. Coffee **good**.

Tweet 2:

Coffee is **good** with **soymilk**. **Go** to the **Festival**.

Tweet 3:

The **coffee festival** had **soymilk, almond**, and **coconut creamers. Delicious! Go!**

Convert to **Transaction** Data

Each Tweet is now a "transaction" made up of words

coffee	**festival**	**delicious**	**love**	**good**			
coffee	**good**	**soymilk**	**go**	**festival**			
coffee	**festival**	**soymilk**	**almond**	**coconut**	**creamer**	**delicious**	**go**

Where else can we find associations?

1) Reviews – patient, purchase, people, movie, book, etc.
2) Documents – speeches, novels
3) Articles – journal papers, news
4) Social Posts – Twitter, FB, etc.
5) Click Streams

Applications and Visual Options

1) Networks – how are things related?
2) Sentiment
3) Topic Modeling
4) Purchase preferences
5) Product placement/suggestion

Example 1
The Rules

TID	*Items*
1	Bread, Coke, Milk
2	Beer, Bread
3	Beer, Coke, Diaper, Milk
4	Beer, Bread, Diaper, Milk
5	Coke, Diaper, Milk

Introduction to Data Mining, 2nd Edition

{Diapers} → {Beer}
{Milk, Bread} → {Coke}
{Milk, Bread} → {Coke, Diaper}
{Diapers} → {Beer, Bread}

** Association (like correlation) is a measure of **co-occurrence** NOT causality.

The Market Basket Example

The following few slides will cover the most classic example of ARM.

You can find this is Kumar’s Data Mining book.

The Measures:
Support, Confidence, Lift

Let A and B be sets and assume rule A $\rightarrow$ B

(Remember, A and B are sets of zero or more items/words)

1) Support:

Sup(A, B) = P(A, B)

[How often items in A and items in B occur together relative to all transactions.]

(Count of A and B together) / (Total # Trans)

2) Confidence:

Conf(A, B) = P(B|A) = P(A,B)/ P(A)

[How often items in A and items in B occur together – relative to transactions that contain A]

(Count of A and B together) / (Count of A)

Lift

For Rules $A \rightarrow B$

Lift (A, B) = **P(A, B) / P(A)P(B)** = P(A | B) / P(A)

1) What is true if Lift(A,B) = 1?
2) What is true if Lift (A,B) < 1?
3) What is true if Lift (A,B) > 1?

Lift

For Rules A ⍰ B

Lift (A, B) = P(A, B) / P(A)P(B)

1) What is true if Lift(A,B) = 1? **A and B are independent!**
2) What is true if Lift (A,B) < 1?
3) What is true if Lift (A,B) > 1?

Lift

For Rules A B

Lift (A, B) = P(A, B) / P(A)P(B)

= P(A | B) P(B) / P(A) P(B)

= P(A | B) / P(A)

1) What is true if Lift(A,B) = 1? A and B are independent
2) What is true if Lift (A,B) < 1? **A and B are negatively correlated**
3) What is true if Lift (A,B) > 1?

Lift

For Rules A □ B

Lift (A, B) = P(A, B) / P(A)P(B)

= P(A | B) P(B) / P(A) P(B)

= P(A | B) / P(A)

1) What is true if Lift(A,B) = 1? A and B are independent
2) What is true if Lift (A,B) < 1? A and B are negatively correlated
3) What is true if Lift (A,B) > 1? **A and B are positively correlated**

We will consider only the rules with **Lift > 1** because we are looking for associations.

Quick Measure Examples

TID	Items
1	Bread, Coke, Milk
2	Beer, Bread
3	Beer, Coke, Diaper, Milk
4	Beer, Bread, Diaper, Milk
5	Coke, Diaper, Milk

Introduction to Data Mining, 2nd Edition

Given: **{Diaper}→ {Beer}**

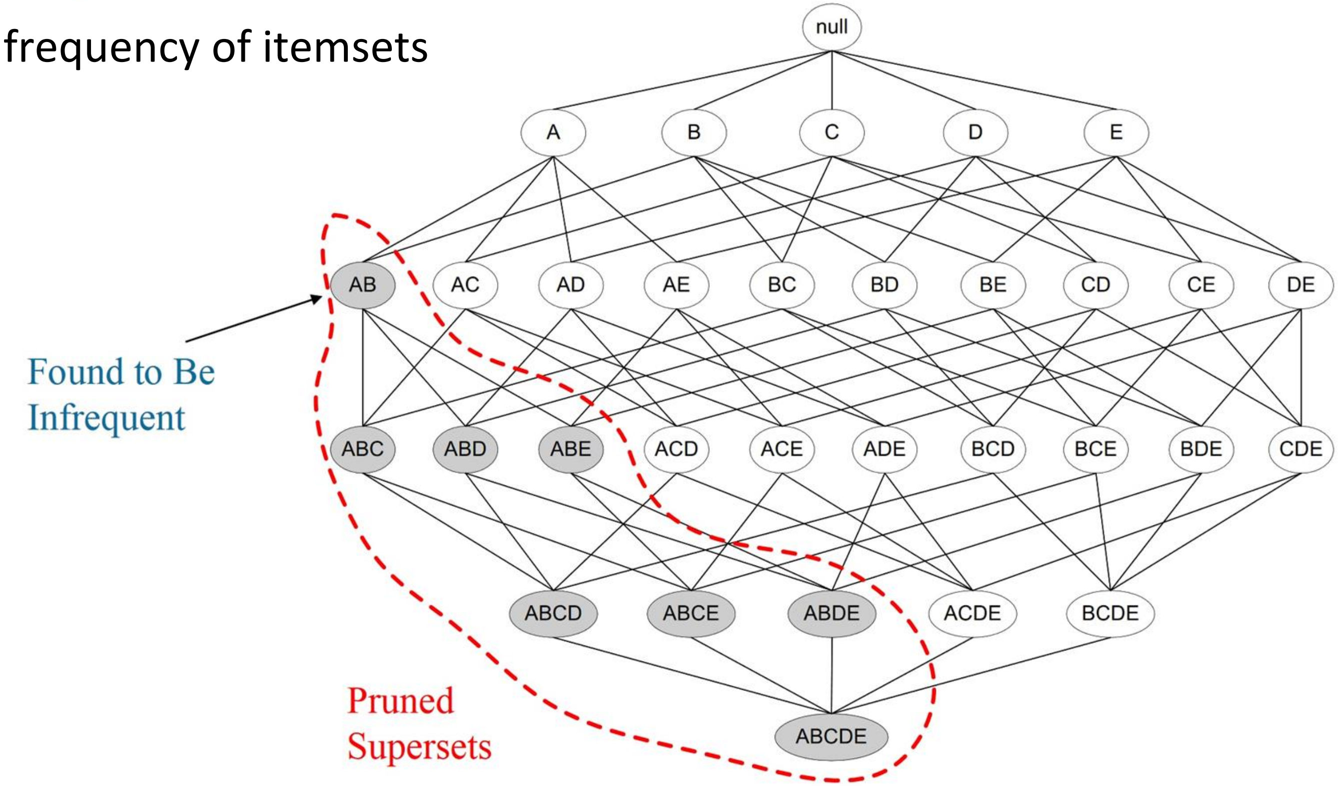
Sup({Diaper}, {Beer}) = 2/5 = .40 = 40%

Conf({Diaper}, {Beer})
= P ({Diaper}, {Beer}) / P({Diaper})
= (2/5) / (3/5) = 66.7%

Lift ({Diaper}, {Beer}) = Sup({Diaper}, {Beer}) / P({Diaper}) * P({Beer}) = (2/5) / (3/5)* (3/5) = 1.11

Quick Reminder: The apriori algorithm

Based on frequency of itemsets



Other Ways to Represent Transaction Data

TID	*Items*
1	Bread, Coke, Milk
2	Beer, Bread
3	Beer, Coke, Diaper, Milk
4	Beer, Bread, Diaper, Milk
5	Coke, Diaper, Milk

Introduction to Data Mining, 2nd Edition

```
> inspect(Foods)
    items                      transactionID
[1] {Bread,Coke,Milk}          1
[2] {Beer,Bread}               2
[3] {Beer,Coke,Diaper,Milk}    3
[4] {Beer,Bread,Diaper,Milk}   4
[5] {Coke,Diaper,Milk}         5
>
```

1	Bread
1	Coke
1	Milk
2	Beer
2	Bread
3	Beer
3	Coke
3	Diaper
3	Milk
4	Beer
4	Bread
4	Diaper
4	Milk
5	Coke
5	Diaper
5	Milk

TID	Bread	Coke	
	Milk	Beer	Diaper
1	1		1
	1	0	0
2	1		0
	0	1	0
3	0		1
	1	1	1
4	1		0
	1	1	1
5	0		1
	1	0	1

quinoa	soymilk	coffee	chocloate	
quinoa	soymilk	kale	tea	
quinoa	kale			
quinoa	soymilk	coffee	chocloate	
quinoa	soymilk	carrot	tea	
quinoa	kale			
quinoa	soymilk	coffee	chocloate	carrot
quinoa	soymilk	kale	tea	
quinoa	carrot			
quinoa	soymilk	coffee	chocloate	
quinoa	soymilk	kale	tea	
quinoa	carrot			
quinoa	soymilk	coffee	chocloate	carrot
quinoa	soymilk		tea	
quinoa	kale			
quinoa	soymilk	coffee	chocloate	
quinoa	soymilk	carrot		
quinoa	carrot			
quinoa	soymilk	coffee	chocloate	
quinoa	soymilk			

Transaction Data

Notice: It is not necessary to have a numbered transaction ID

Basic ARM R Code

```
library(arules)

Foods <- read.transactions("HealthyBasketData.csv",
                           rm.duplicates = FALSE,
                           format = "basket",
                           sep=",",
                           cols=NULL)
inspect(Foods)

rules <- arules::apriori(Foods, parameter = list(support=.2,
                  confidence=.2, minlen=2))
inspect(rules)

SortedRules <- sort(rules, by="confidence", decreasing=TRUE)
inspect(SortedRules[1:10])

SortedRulesL <- sort(rules, by="lift", decreasing=TRUE)
inspect(SortedRulesL[1:10])
```

```
> SortedRules <- sort(rules, by="confidence", decreasing=TRUE)
> inspect(SortedRules[1:10])
     lhs             rhs          support confidence lift     count
[1]  {kale}       => {quinoa}     0.30    1          1.000000 6
[2]  {tea}        => {soymilk}    0.25    1          1.428571 5
[3]  {tea}        => {quinoa}     0.25    1          1.000000 5
[4]  {carrot}     => {quinoa}     0.35    1          1.000000 7
[5]  {coffee}     => {chocloate}  0.35    1          2.857143 7
[6]  {chocloate}  => {coffee}     0.35    1          2.857143 7
[7]  {coffee}     => {soymilk}    0.35    1          1.428571 7
[8]  {coffee}     => {quinoa}     0.35    1          1.000000 7
[9]  {chocloate}  => {soymilk}    0.35    1          1.428571 7
[10] {chocloate}  => {quinoa}     0.35    1          1.000000 7
>
> SortedRulesL <- sort(rules, by="lift", decreasing=TRUE)
> inspect(SortedRulesL[1:10])
     lhs                            rhs         support confidence lift     count
[1]  {coffee}                    => {chocloate} 0.35    1.0000000  2.857143 7
[2]  {chocloate}                 => {coffee}    0.35    1.0000000  2.857143 7
[3]  {coffee,soymilk}            => {chocloate} 0.35    1.0000000  2.857143 7
[4]  {chocloate,soymilk}         => {coffee}    0.35    1.0000000  2.857143 7
[5]  {coffee,quinoa}             => {chocloate} 0.35    1.0000000  2.857143 7
[6]  {chocloate,quinoa}          => {coffee}    0.35    1.0000000  2.857143 7
[7]  {coffee,quinoa,soymilk}     => {chocloate} 0.35    1.0000000  2.857143 7
[8]  {chocloate,quinoa,soymilk}  => {coffee}    0.35    1.0000000  2.857143 7
[9]  {tea}                       => {soymilk}   0.25    1.0000000  1.428571 5
[10] {soymilk}                   => {tea}       0.25    0.3571429  1.428571 5
```

Read Two Common Formats

```
Foods <- read.transactions("KumarGroceriesTransData.csv",
                    rm.duplicates = FALSE,
                    format = "single", ##or basket
                    sep=",",
                    skip=0,
                    cols=c(1,2) ## for single, 1 ID col , 2 is item
                                ## default is NULL for basket. Null means no IDs
 )
arules::inspect(Foods)
```

```
Foods2 <- read.transactions("KumarGroceriesTransData_ASTRANS.csv",
                    rm.duplicates = FALSE,
                    format = "basket",
                    sep=",",
                              cols=1 ##ID in col 1 if no ID then cols=NULL
)
arules::inspect(Foods2)
```

Thinking Outside the Basket
Twitter Data

1) Will need to create a "**document of transactions**" – one for each Tweet.
2) ** **Each row is a Tweet.**
3) Each column is a word (token) in that Tweet.
4) Order does not matter.
5) No duplicates

R Association Rules and Twitter: libraries

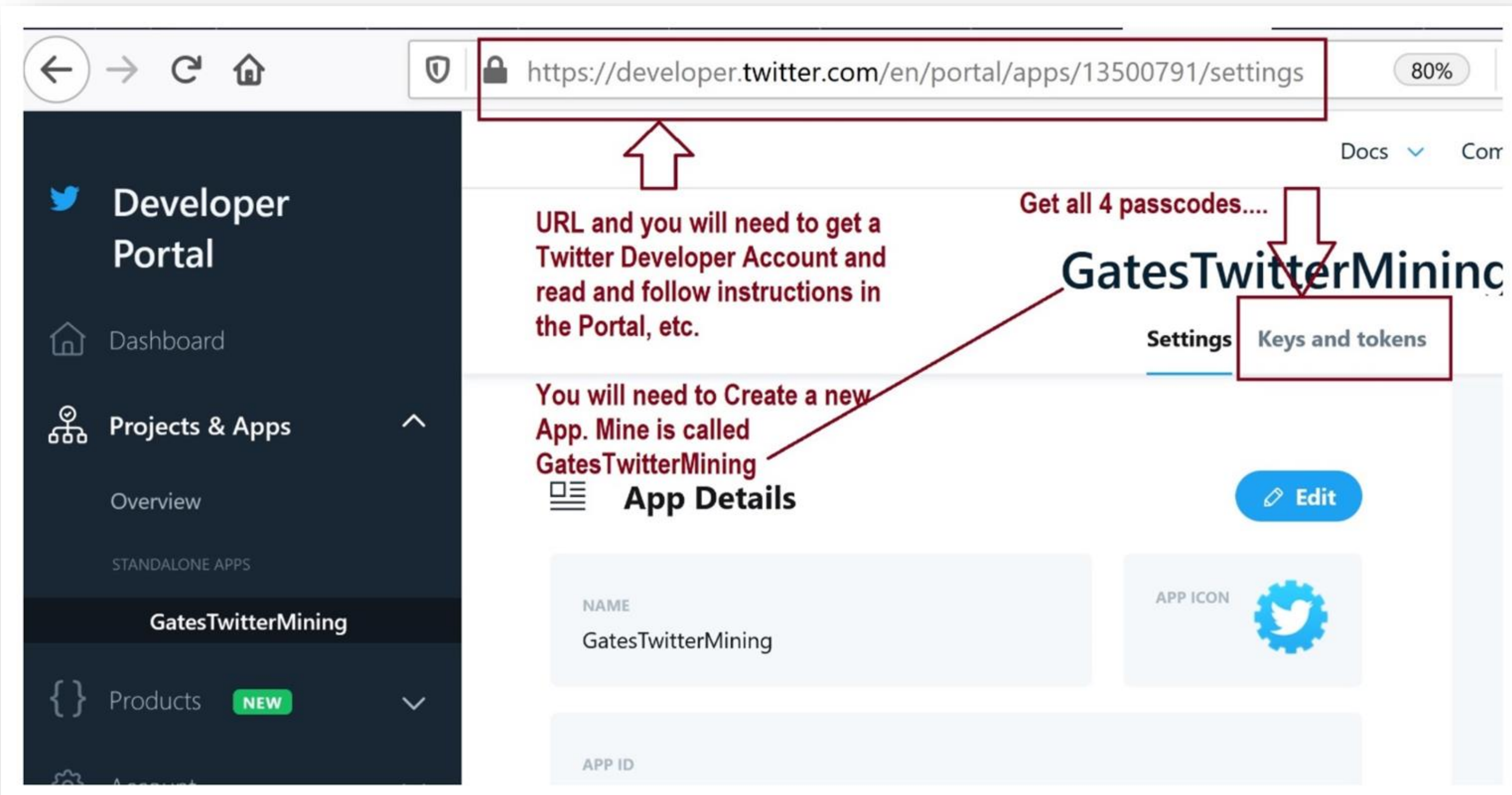
library(arules)
library(rtweet)
library(twitteR)
library(ROAuth)
library(jsonlite)
#library(streamR)
library(rjson)
library(tokenizers)
library(tidyverse)
library(plyr)
library(dplyr)
library(ggplot2)
#install.packages("syuzhet")
sentiment analysis
library(syuzhet)
library(stringr)
library(arulesViz) ## load last

Trouble with arulesViz?
FIRST - you MUST register and log into github
install_github("mhahsler/arulesViz")
RE: https://github.com/mhahsler/arulesViz

Trouble with arules not working suddenly
detach("package:arules", unload=TRUE)
library("arules")

Set Up Twitter Dev Account First

https://developer.twitter.com/en/portal/apps



R Twitter Options

```
##############  Using twittR ##########################################
setup_twitter_oauth(consumerKey,consumerSecret,access_Token,access_Secret)

Search<-twitteR::searchTwitter("#ILoveChocolate",n=100,since="2018-09-09")
(Search_DF <- twListToDF(Search))
TransactionTweetsFile = "Choc.csv"
```

(Search_DF <- twListToDF(Search))

```
                                  text
1 The other day I woke up craving chocolate cupcakes. Today I'm craving @HersheyCompany chocolate bars. 
 think the u… https://t.co/NtGH4eaSRc
2            WHO SAID "CHOCOLATE"?\n__________________________________\n#feed #feedsmartfood #honey #we
ovechocolate… https://t.co/DzzmvJlKEh
3                                                           @ClaireValy @LowngSnake @firebox #ILOVECHOCO
ATE\nI love Chocolate very very much.
4                 #HealthTips #momlife #sahmlife #toddlers #ilovechocolate #homeschoolmom #bethechange #
oingitformygirls #fitmom #feeltheburn
5   RT @Kelly_Hawrylysh: #Fairtrade sourcing needed more than ever to avoid chocapocalypse!!! https://t.
o/dbxw3eQfTc #SDG12 @FairtradeAfrica…
6   RT @Kelly_Hawrylysh: #Fairtrade sourcing needed more than ever to avoid chocapocalypse!!! https://t.
o/dbxw3eQfTc #SDG12 @FairtradeAfrica…
  favorited favoriteCount     replyToSN             created truncated          replyToSID
1     FALSE             0          <NA> 2018-09-27 12:12:52      TRUE                <NA>
2     FALSE             0          <NA> 2018-09-27 10:51:42      TRUE                <NA>
3     FALSE             0    ClaireValy 2018-09-27 00:45:43     FALSE 1044897146326208513
4     FALSE             0 templin_katie 2018-09-26 19:49:55     FALSE 1045037612388536321
5     FALSE             0          <NA> 2018-09-26 16:24:22     FALSE                <NA>
6     FALSE             0          <NA> 2018-09-26 16:23:42     FALSE                <NA>
                   id           replyToUID
1 1045285140505735169                 <NA>
2 1045264712118734848                 <NA>
3 1045112213915226113           2878148959
4 1045037771050618881 1035584652722036736
5 1044986045975220224                 <NA>
6 1044985877456392194                 <NA>
                                                                             statusSource
1 <a href="http://twitter.com/download/android" rel="nofollow">Twitter for Android</a>
2                         <a href="http://instagram.com" rel="nofollow">Instagram</a>
3                  <a href="http://twitter.com" rel="nofollow">Twitter Web Client</a>
4   <a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a>
5 <a href="http://twitter.com/download/android" rel="nofollow">Twitter for Android</a>
6 <a href="http://twitter.com/download/android" rel="nofollow">Twitter for Android</a>
      screenName retweetCount isRetweet retweeted longitude latitude
1     RachelTBue            0     FALSE     FALSE      <NA>     <NA>
2      Niklaus_R            0     FALSE     FALSE   4.35008   50.845
3    saminaseem16           0     FALSE     FALSE      <NA>     <NA>
```

Build the Transaction File: Step 1

1) Each tweet should be one transaction.
2) Each word (token) in the tweet should be in its own column.

```
> (Search_DF$text[1])
[1] "The other day I woke up craving chocolate cupcakes. Today I'm craving @HersheyCompany chocolate bars. I
think the u… https://t.co/NtGH4eaSRc"
```

Build The Transaction File: Step 2

```
## Start the file
Trans <- file(TransactionTweetsFile)
## Tokenize to words
Tokens<-tokenizers::tokenize_words(Search_DF$text[1],stopwords = stopwords::stopwords("en"),
          lowercase = TRUE,  strip_punct = TRUE, strip_numeric = TRUE,simplify = TRUE)
## Write squished tokens
cat(unlist(str_squish(Tokens)), "\n", file=Trans, sep=",")
close(Trans)

## Append remaining lists of tokens into file
## Recall - a list of tokens is the set of words from a Tweet
Trans <- file(TransactionTweetsFile, open = "a")
for(i in 2:nrow(Search_DF)){
  Tokens<-tokenize_words(Search_DF$text[i],stopwords = stopwords::stopwords("en"),
          lowercase = TRUE,  strip_punct = TRUE, simplify = TRUE)
  cat(unlist(str_squish(Tokens)), "\n", file=Trans, sep=",")
}
close(Trans)
```

Transaction File: Each Row is a Tweet (Opened with Excel)

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	day	woke	craving	chocolate	cupcakes	today	craving	hersheyco	chocolate	bars	think	u	https	t.co	ntgh4easr
2	said	chocolate	________	feed	feedsmart	honey	welovechc	https	t.co	dzzmvjlkeh					
3	clairevaly	lowngsnak	firebox	ilovechocc	love	chocolate	much								
4	healthtips	momlife	sahmlife	toddlers	ilovechocc	homescho	bethechar	doingitfori	fitmom	feeltheburn					
5	rt	kelly_haw	fairtrade	sourcing	needed	ever	avoid	chocapoca	https	t.co	dbxw3eqft	sdg12	fairtradeafrica		
6	rt	kelly_haw	fairtrade	sourcing	needed	ever	avoid	chocapoca	https	t.co	dbxw3eqft	sdg12	fairtradeafrica		
7	cada	día	estamos	mas	listos	para	navidad	taza	3	pack	de	venta	en	cityclub	navidad
8	fairtrade	sourcing	needed	ever	avoid	chocapoca	https	t.co	dbxw3eqft	sdg12	https	t.co	rgmtaombom		
9	ilovechocc	chocolate	adictaalch	https	t.co	kpzofu8ix2									
10	see	big	chocolate	show	saturday	night	ilovechocolate								
11	else	can	say	thehousec	braziliantr	truffles	brigadeiro	desserts	https	t.co	pzayia63ir				
12	touch	cocoa	please	ilovechocc	bless	https	t.co	vx7v7csfr5							
13	bako_nw	weekendb	choc	dome	hiding	double	chocolate	cheesecak	ilovechocc	https	t.co	f2ginuvtfq			
14	ilovechocolate														
15	los	lunes	lucen	tan	malos	si	los	ves	con	la	actitud	correcta	chocolate	iniciodese	felizlunes
16	enough	words	express	thankful	amazing	coworkers	thank	ccriheathe	onl	https	t.co	2gljgtudhh			
17	casa	ino	nostra	przedstaw	hotel	hotelwgór	tatry	podhale	zakopane	nowytarg	deser	slodycz	suflet	https	t.co
18	rt	ccfchocola	crunchy	biscuit	dipped	chocolate	foodporn	yummy	sweets	love	instafood	food	delicious	choco	dessert
19	crunchy	biscuit	dipped	chocolate	foodporn	yummy	sweets	love	instafood	food	delicious	choco	dessert	https	t.co
20	bbcmiami:	light	ilovechocolate												

Read and Inspect the Transactions

```
######### Read in the tweet transactions
TweetTrans <- read.transactions(TransactionTweetsFile,
                                rm.duplicates = FALSE,
                                format = "basket",
                                sep=","
                                ## cols =
                                )
inspect(TweetTrans)
## See the words that occur the most
Sample_Trans <- sample(TweetTrans, 50)
summary(Sample_Trans)
```

```
most frequent items:
          https            t.co       chocolate ilovechocolate              rt
             35              35              25              23               9
```

Transaction Sets and Summary

```
[59] {1,
      along,
      box,
      chocolates,
      days,
      domme,
      findom,
      finsub,
      godiva,
      ilovechocolate,
      pay,
      send}
[60] {chocolate,
      delicious,
      food,
      foodporn,
      https,
      instafood,
      introducing,
      love,
      mango,
      marzipan,
      sweets,
      t.co,
      truffles,
      u17wpqhhxh,
      yummy}
```

Clean Up

```
## Read the transactions data into a dataframe
TweetDF <- read.csv(TransactionTweetsFile, header = FALSE, sep = ",")
head(TweetDF)
```

```
> TweetDF <- read.csv(TransactionTweetsFile, header = FALSE, sep = ",")
> head(TweetDF)
          V1              V2                                        V3            V4            V5
1        day            woke                                   craving     chocolate      cupcakes
2       said       chocolate ___________________________________             feed feedsmartfood
3 clairevaly      lowngsnake                                   firebox ilovechocolate         love
4 healthtips         momlife                                  sahmlife      toddlers ilovechocolate
5         rt kelly_hawrylysh                                 fairtrade      sourcing        needed
6         rt kelly_hawrylysh                                 fairtrade      sourcing        needed
            V6              V7                V8        V9        V10        V11   V12             V13
1        today         craving     hersheycompany chocolate       bars      think     u           https
2        honey welovechocolate              https      t.co dzzmvjlkeh
3    chocolate            much
4 homeschoolmom     bethechange doingitformygirls    fitmom feeltheburn
5         ever           avoid     chocapocalypse     https       t.co dbxw3eqftc sdg12 fairtradeafrica
6         ever           avoid     chocapocalypse     https       t.co dbxw3eqftc sdg12 fairtradeafrica
```

```
most frequent items:
          https            t.co       chocolate ilovechocolate              rt
             35              35              25             23               9
```

Specifically Remove Words

```
## Convert all columns to char
TweetDF<-TweetDF %>%
  mutate_all(as.character)
(str(TweetDF))
# We can now remove certain words
TweetDF[TweetDF == "t.co"] <- ""
TweetDF[TweetDF == "rt"] <- ""
TweetDF[TweetDF == "http"] <- ""
TweetDF[TweetDF == "https"] <- ""

## Clean with grepl - every row in each column
MyDF<-NULL
for (i in 1:ncol(TweetDF)){
  MyList=c() # each list is a column of logicals ...
  MyList=c(MyList,grepl("[[:digit:]]", TweetDF[[i]]))
  MyDF<-cbind(MyDF,MyList)  ## create a logical DF
  ## TRUE is when a cell has a word that contains digits
}
## For all TRUE, replace with blank
TweetDF[MyDF] <- ""
(TweetDF)
```

```r
## Clean with grepl - every row in each column
MyDF<-NULL
MyDF2<-NULL
MyDF3<-NULL
for (i in 1:ncol(TweetDF)){
  MyList=c() # each list is a column of logicals ...
  MyList=c(MyList,grepl("[[:digit:]]", TweetDF[[i]]))

  MyList2=c()## for small words
  MyList2=c(MyList2,grepl("[A-z]{4,}", TweetDF[[i]]))

  MyList3=c()## for large words
  MyList3=c(MyList3,grepl("[A-z]{12,}", TweetDF[[i]]))

  MyDF<-cbind(MyDF,MyList)  ## create a logical DF
  MyDF2<-cbind(MyDF2,MyList2)
  MyDF3<-cbind(MyDF3,MyList3)
}
## For all TRUE, replace with blank
TweetDF[MyDF] <- ""
TweetDF[!MyDF2] <- ""
TweetDF[MyDF3] <- ""
(head(TweetDF,10))
```

Our Transactions

	V1	V2	V3	V4	V5	V6	V7	V8	V9
1	looking	healthy	food	swaps				eathealthy	nutrition
2			dates	stuffed	roasted	almonds	dipped	chocolate	weekend
3	dates	stuffed	roasted	almonds	dipped	chocolate	weekend	indulgences	giftbox
4		reese's	cups	mini's	they're	half	calories	regular	ones
5		cuando		aburre		helado		vainilla	tenemos
6	bigass	chocolate	happydiwali	corporate	tings	bigting			africa
7		birthday	cake	chocolate	orange	shared	friends		
8	doubt		chocolate	person	tries	make	move	chocolate	
9	another	great	gift	idea	just	love	bottles	come	
10									

	V10	V11	V12	V13	V14	V15	V16	V17	V18	V19	V20	V21	V22
1													
2	indulgences	giftbox	foodporn	yummy	sweets								
3	foodporn	yummy											
4	unless			they're									
5	siempre		chocolate	pero	blanco	pata	hacer		nuestro				
6	umhlanga												
7	birthday			gmakffbpaj									
8													
9		exciting	christmas										
10													

Association Rule Mining

```
[70] {chocolate,
      delicious,
      food,
      foodporn,
      instafood,
      introducing,
      love,
      mango,
      marzipan,
      sweets,
      truffles,
      yummy}
[71] {bali's,
      big,
      check,
      chocolatiers,
      ilovechocolate,
      six,
      theyakmag,
      theyakmagazine,
      yak}
```

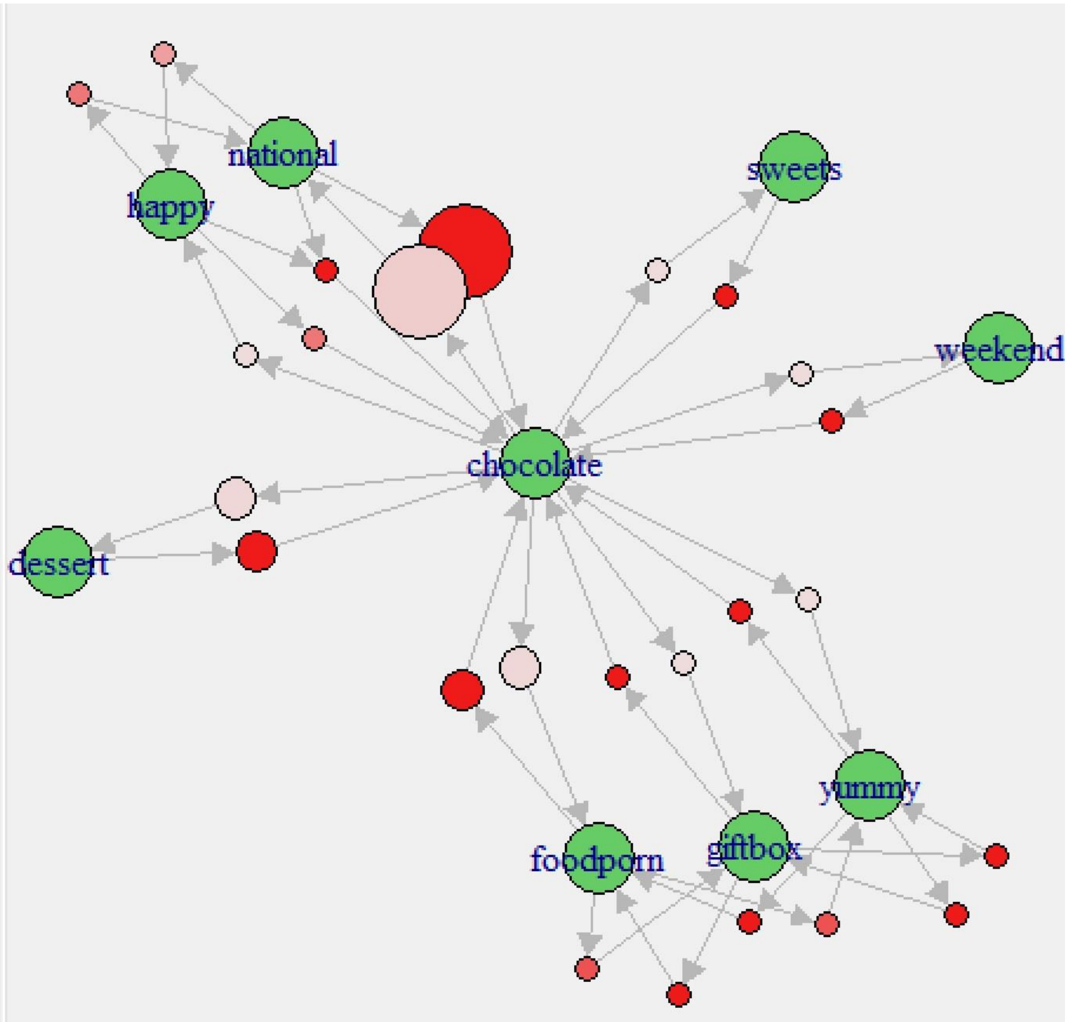
Example cleaner tweets as individual transactions.

```
TweetTrans_rules = arules::apriori(TweetTrans,
            parameter = list(support=.0001, confidence=.0001, minlen=2, maxlen=6))
                              #minlen=2, maxtime=10))
inspect(TweetTrans_rules[1:30])
## sorted
SortedRules_sup <- sort(TweetTrans_rules, by="support", decreasing=TRUE)
inspect(SortedRules_sup[1:20])
```

```
> inspect(SortedRules_sup[1:20])
     lhs            rhs         support    confidence lift      count
[1]  {national}  => {chocolate} 0.11267606 1.00000000  1.731707 8
[2]  {chocolate} => {national}  0.11267606 0.19512195  1.731707 8
[3]  {dessert}   => {chocolate} 0.07042254 1.00000000  1.731707 5
[4]  {chocolate} => {dessert}   0.07042254 0.12195122  1.731707 5
[5]  {foodporn}  => {chocolate} 0.07042254 1.00000000  1.731707 5
[6]  {chocolate} => {foodporn}  0.07042254 0.12195122  1.731707 5
[7]  {happy}     => {national}  0.05633803 0.66666667  5.916667 4
[8]  {national}  => {happy}     0.05633803 0.50000000  5.916667 4
[9]  {happy}     => {chocolate} 0.05633803 0.66666667  1.154472 4
[10] {chocolate} => {happy}     0.05633803 0.09756098  1.154472 4
[11] {weekend}   => {chocolate} 0.05633803 1.00000000  1.731707 4
[12] {chocolate} => {weekend}   0.05633803 0.09756098  1.731707 4
[13] {sweets}    => {chocolate} 0.05633803 1.00000000  1.731707 4
[14] {chocolate} => {sweets}    0.05633803 0.09756098  1.731707 4
[15] {giftbox}   => {yummy}     0.05633803 1.00000000 17.750000 4
[16] {yummy}     => {giftbox}   0.05633803 1.00000000 17.750000 4
[17] {giftbox}   => {foodporn}  0.05633803 1.00000000 14.200000 4
[18] {foodporn}  => {giftbox}   0.05633803 0.80000000 14.200000 4
[19] {giftbox}   => {chocolate} 0.05633803 1.00000000  1.731707 4
[20] {chocolate} => {giftbox}   0.05633803 0.09756098  1.731707 4
```

A Quick Plot

```
library(arulesViz)
SortedRules_sup <- sort(TweetTrans_rules, by="support", decreasing=TRUE)
inspect(SortedRules_sup[1:20])
plot (SortedRules_sup[1:25],method="graph",engine='interactive',shading="confidence")
```



Size: support
Color (dark=higher): conf