Example Library Word Clouds

Gordon Erlebacher

What are some enhancements in R?

- Graphics
- Statistics
- Calculations
- Graphics Processing
- Interfaces to other computer languages
- Routines to simplify tasks

A package

- Package is composed of:
  - collection of functions
  - one or several datasets
  - documentation

Useful packages

- Set of R packages
- http://cran.r-project.org/web/packages/available_packages_by_name.html

- Useful packages:
- excel.link
- ez: Easy analysis and visualization of factorial experiments.
- USE IN CLASS?

- gdata: R tools for data manipulation
- read.xls (very useful, but needs perl)

- granova: graphical analysis of variance.

- All packages by name: http://cran.r-project.org/web/packages/available_packages_by_name.html

Useful packages

- Psych: for psychology research
- http://cran.r-project.org/web/packages/psych/index.html
- http://cran.r-project.org/web/packages/psych/psych.pdf

- rattle
- User interface for data mining
- http://cran.r-project.org/web/packages/rattle/index.html
- http://cran.r-project.org/web/packages/PairedData/PairedData.pdf

- scatterplot3d
- http://cran.r-project.org/web/packages/scatterplot3d/index.html
- http://cran.r-project.org/web/packages/scatterplot3d/scatterplot3d.pdf

- violinmplot
- violin plots with mean and standard deviation

Useful packages

- wordcloud *
- http://cran.r-project.org/web/packages/wordcloud/index.html
- http://cran.r-project.org/web/packages/wordcloud/wordcloud.pdf

- zipfR
- Statistical models for word frequency distributions
- http://cran.r-project.org/web/packages/zipfR/index.html
- http://cran.r-project.org/web/packages/zipfR/zipfR.pdf

- dae: Functions useful in the design and ANOVA of experiments
- http://cran.r-project.org/web/packages/dae/index.html
- http://cran.r-project.org/web/packages/dae/dae.pdf

List of Libraries
(with one-line description)

- http://cran.r-project.org/web/packages/available_packages_by_name.html

Objective of next few lessons

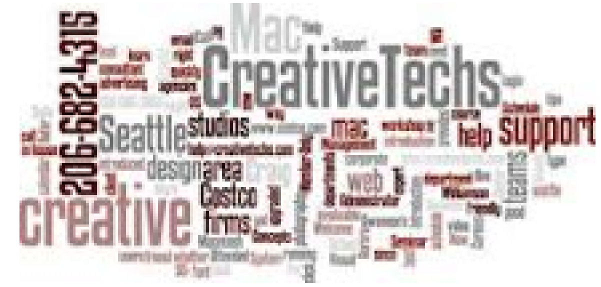
- Learn a library we know nothing about,
- with concepts we know almost nothing about
- using terminology we know nothing about
- How does one handle all the above on our own?

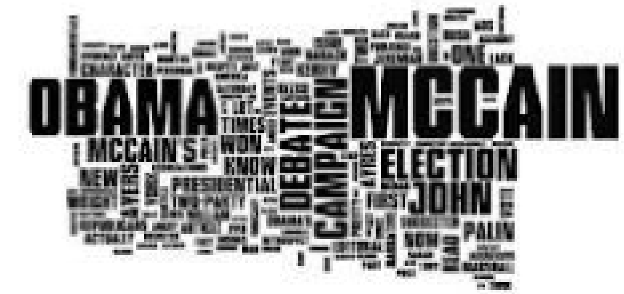
Word Clouds

- Read two newspaper articles on the economy:
  - Slant to the left
  - Slant to the right
- How to analyze?
- Construct a word cloud


Inept
Incompetent
Bad
Sucks
Dysfunctional
Confusion
Selfish
Poor
Corrupt
Lazy
Crooks
Ineffective
Disappointing
Gridlock


law
Kindle
Conference
Library
good
Session
School
digital
Great
presentation
like


Mac
CreativeTechs
206-682-4315
Seattle
creative
support
design
Costco

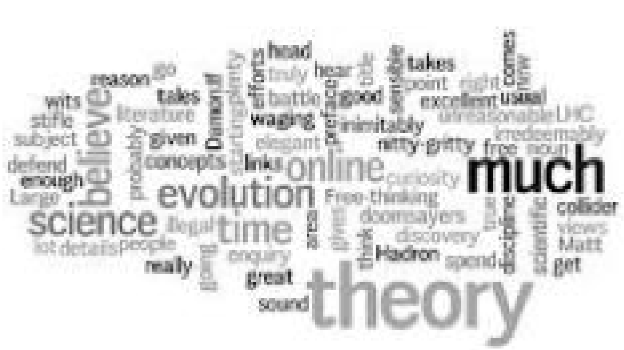

OBAMA
MCCAIN
CAMPAIGN
DEBATE
ELECTION
JOHN
MCCAIN'S


WHALE
AHAB
DOWN
INTO
YE
SHIP
SEA
CAPTAIN
OVER

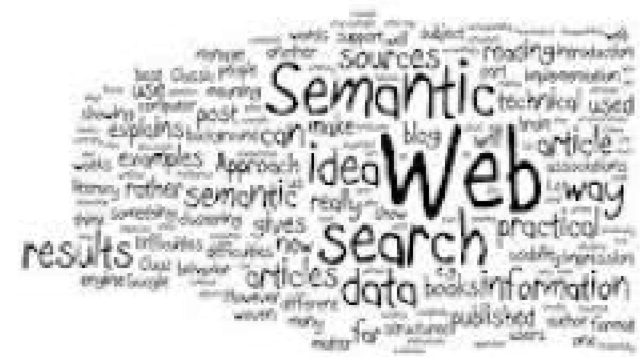

agree
aphra
behn
blog
defend
freedom
question
speech
persuade
wordpress
words
Aphra Behn


strategy
processes
organization
technology
information
change
future
strategic
systems
vision
process


kiss
we
love
meet
together
trust
marriage
engagement
dream
share
us
I do
date
join
talk

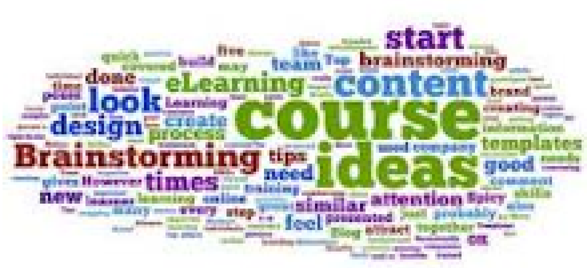

much
evolution
science
theory
online
time
believe

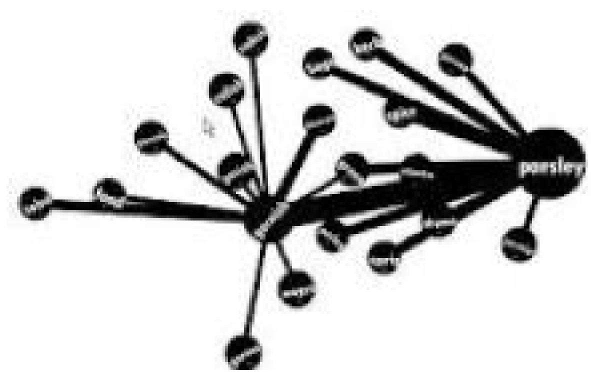

million
Speaker
billion
nation
Mr
Government
Budget
building

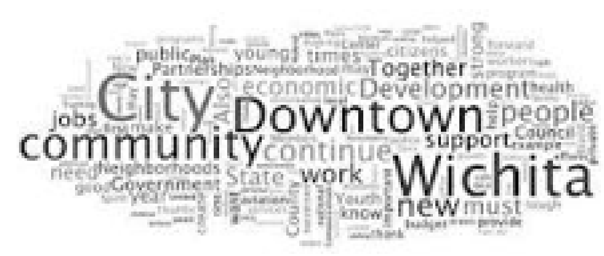

Semantic
Web
Search
idea
data
results
information


edits
file
Word
track
changes
comments
review
document
PDF

Page 3


start
content
course
ideas
Brainstorming
eLearning
look
design




City
Downtown
community
Wichita
Development
people


BOOKS
BEST
BOOK
TWO
TIME
STAR

word cloud psychology (google images)

Psychology suicide stress

Generate our own

- Library “wordcloud”
- How to use it
- Step 1:
  - What packages are related to “world clouds” if any?
- Step 2:
  - install.packages(“???? step 1 ???”)
  - Choose a repository: typically one in the USA, or use RStudio package installer
- Step 2:
  - look over some of the routines in the package

Use of WordCloud

- What is my objective?
  - given a file (ideally, pdf, but text should do)
  - read the file
  - produce a visual word cloud
  - produce a list of words + frequency and placement information (in the cloud)

Next few slides

- Installation of R without RStudio

Using R, install "wordcloud"

- install "wordcloud" via "install" button in RStudio did not work
- Use "install.packages("wordcloud") on the command line within RStudio, which worked, but ..
- Try loading "wordcloud" (press check box next to library name in RStudio), ...

library("wordcloud", lib.loc="/Library/Frameworks/R.framework/Versions/2.15/Resources/library")

Error: **package 'Rcpp' required by 'wordcloud' could not be found**

In addition: Warning message:

package 'wordcloud' was built under R version 2.15.1

Rcpp

- Installs easily and automatically using RStudio 0.95… and R version 3.03 on my mac.
- You might different experiences on different systems and different versions of R.
- The following pages give some information should have problems with Rcpp

Next ...

- Install earlier (or later) version of R for the Mac
  - R must be installed independently of RStudio
- Check documentation of Rcpp online
  - http://cran.r-project.org/web/packages/Rcpp/index.html
  - Rcpp is version 0.10.3 and requires R2.15.1
  - Another alternative: use an earlier version of Rcpp

Search Google
rcpp r2.15.0 mac

I have a problem installing "Rcpp" package in R. The install command have worked for all packages but not for this one. The R error is :

package ‘Rcpp’ is not available (for R version 2.15.0)

I am using windows Vista.Please help.

As can be seen on the CRAN status page for Rcpp, it now depends on R (>= 2.15.1).

So you either update R from 2.15.0 to 2.15.1 (which is not a bad idea), or you can **try to install an older Rcpp version such as 0.9.10 which will work with R 2.15.0.**

More searching

http://www.icesi.edu.co/CRAN/web/checks/check_results_Rcpp.html

CRAN Package Check Results for Package Rcpp

Last updated on 2012-06-02 05:49:56.

Flavor	Version	$T_{install}$	T_{check}	T_{total}	Status	Flags
r-devel-linux-x86_64-gcc-debian	0.9.10	34.29	271.37	305.66	WARN	
r-devel-linux-x86_64-gcc-fedora	0.9.10			358.16	WARN	
r-patched-linux-x86_64	0.9.10	34.44	263.41	297.85	NOTE	
r-patched-solaris-sparc	0.9.10				ERROR	
r-patched-solaris-x86	0.9.10				ERROR	
r-release-linux-ix86	0.9.10	34.11	218.49	252.60	NOTE	
r-release-macosx-ix86	0.9.10	150.00	594.00	744.00	NOTE	
r-release-windows-ix86+x86_64	0.9.10	110.00	341.00	451.00	NOTE	
r-oldrel-macosx-ix86	0.9.10	151.00	583.00	734.00	NOTE	
r-oldrel-windows-ix86+x86_64	0.9.10	93.00	287.00	380.00	NOTE	

http://www.icesi.edu.co/CRAN/web/packages/Rcpp/index.html

http://www.icesi.edu.co/CRAN/web/packages/Rcpp/index.html

(can find binaries for the mac, windows, linux)

- You'll notice that Rcpp is at a particular repository, different than the default used in RStudio by default (or else Rcpp would have been installed)

```
setRepositories(addURLs="http://www.icesi.edu.co/CRAN/web/packages")
```

Problem: tar.gz file with Rcpp not found on the link provided.

Search Google for file name Rcpp_0.9.10.tgz with mac in URL

Knowing the file name

http://www.filewatcher.com/m/Rcpp_0.9.10.tgz.7636884-0.html

Download mirrors for Rcpp_0.9.10.tgz (7.28 MB):

2012-02-17 ftp://ftp.psu.ac.th/pub/cran/bin/macosx/leopard/contrib/2.14/Rcpp_0.9.10.tgz
2012-02-17 ftp://ftp.uwsg.indiana.edu/pub/CRAN/bin/macosx/leopard/contrib/2.14/Rcpp_0.9.10.tgz
2012-02-17 ftp://ftp.ro.debian.org/pub/CRAN/bin/macosx/leopard/contrib/2.14/Rcpp_0.9.10.tgz
2012-02-17 ftp://ftp.fsn.hu/pub/CRAN/bin/macosx/leopard/contrib/2.14/Rcpp_0.9.10.tgz
2012-02-17 ftp://ftp.dk.debian.org/.disk1/cran/bin/macosx/leopard/contrib/2.14/Rcpp_0.9.10.tgz
2012-02-17 ftp://ftp.kaist.ac.kr/CRAN/bin/macosx/leopard/contrib/2.14/Rcpp_0.9.10.tgz
2012-02-17 ftp://ftp.drupal.org/.1/cran/bin/macosx/leopard/contrib/2.14/Rcpp_0.9.10.tgz
2012-02-17 ftp://ftp.musicbrainz.org/.1/cran/bin/macosx/leopard/contrib/2.14/Rcpp_0.9.10.tgz
2012-02-17 ftp://ftp.oregonstate.edu/.1/cran/bin/macosx/leopard/contrib/2.14/Rcpp_0.9.10.tgz
2012-02-17 ftp://ftp.sun.ac.za/pub/mirrors/cran.za.r/bin/macosx/leopard/contrib/2.14/Rcpp_0.9.10.tgz
2012-02-17 ftp://ftp.slackware.com/.1/cran/bin/macosx/leopard/contrib/2.14/Rcpp_0.9.10.tgz
2012-02-17 ftp://ftp.hu.debian.org/pub/CRAN/bin/macosx/leopard/contrib/2.14/Rcpp_0.9.10.tgz
2012-02-17 ftp://ftp2.ca.freebsd.org/CRAN/bin/macosx/leopard/contrib/2.14/Rcpp_0.9.10.tgz

Installing from RStudio created a problem:
for some reason, the package could not be installed.
Solution: install as administrator, from outside R

cloud library (conclusions)

- Conclusion:
  - Installation was quite difficult
  - Libraries are sometimes require a version of R more recent than what is installed
  - Sometimes, installation must proceed by hand
  - In many cases, an administrator, TA, friend can help you

```
> example(wordcloud)

Warning message:

In example(wordcloud) : no help found for ‘wordcloud’

> ?wordcloud

No documentation for ‘wordcloud’ in specified packages and libraries:
```

you could try ‘**??wordcloud**’

?wordcloud::wordcloud

Success!!

Plot a word cloud

Description

Plot a word cloud

Usage

```
wordcloud(words,freq,scale=c(4,.5),min.freq=3,max.words=Inf,random
.order=TRUE, random.color=FALSE,
      rot.per=.
1,colors="black",ordered.colors=FALSE,use.r.layout=FALSE,...)
```

Arguments

Regarding version of R

- See following slides
- Version of R was 2.13, and there was a compatible version of Rcpp (in 2012)
- Our version of R is 2.15, and Rcpp requires version 2.15.1 of R (in 2013)
- Problem: development of R and the libraries occur at different rates, and are done by different people

install.package

```
> install.packages("wordcloud")
Installing package(s) into '/Users/erlebach/Library/R/2.13/library'
(as 'lib' is unspecified)
--- Please select a CRAN mirror for use in this session ---
Loading Tcl/Tk interface ... done
CRAN mirror

 1: Argentina (La Plata)      2: Argentina (Mendoza)
 3: Australia (Canberra)      4: Australia (Melbourne)
 5: Austria                   6: Belgium
 7: Brazil (PR)               8: Brazil (RJ)
 9: Brazil (SP 1)            10: Brazil (SP 2)
11: Canada (BC)              12: Canada (NS)
13: Canada (ON)              14: Canada (QC 1)
15: Canada (QC 2)            16: Chile
17: China (Beijing 1)        18: China (Beijing 2)
19: China (Beijing 3)        20: China (Guangzhou)
21: China (Hefei)            22: China (Xiamen)
23: Colombia (Bogota)        24: Colombia (Cali)
25: Denmark                  26: France (Toulouse)
27: France (Lyon 1)          28: France (Lyon 2)
29: Germany (Berlin)         30: Germany (Goettingen)
31: Germany (Wiesbaden)      32: Greece
33: India                    34: Indonesia (Jakarta 1)
35: Indonesia (Jakarta 2)    36: Iran
37: Ireland                  38: Italy (Milano)
39: Italy (Padua)            40: Italy (Palermo)
41: Japan (Hyogo)            42: Japan (Tsukuba)
```

load library

```
> library("wordcloud")
Loading required package: Rcpp
Loading required package: RColorBrewer
Warning messages:
1: package 'wordcloud' was built under R version 2.13.2
2: package 'Rcpp' was built under R version 2.13.2
```

Functions in wordcloud

> library(help="wordcloud")
> package(help="wordcloud") # alternative to the previous line

Information on package 'wordcloud'

…

Index:

SOTU United States State of the Union Addresses (2010 and 2011)
commonality.cloud Plot a commonality cloud
comparison.cloud Plot a comparison cloud
wordcloud Plot a word cloud

wordcloud::wordcloud

- package : wordcloud
- function : wordcloud

help(wordcloud)

wordcloud package:wordcloud R Documentation

Plot a word cloud

Description:

Plot a word cloud

Usage:

```
wordcloud(words,freq,scale=c(4,.5),min.freq=3,max.words=Inf,random.order=TRUE,
random.color=FALSE,
                rot.per=.1,colors="black",ordered.colors=FALSE,use.r.layout=FALSE,...)
```

Next step

Or use RStudio installer

- Need package tm !!!
- install.packages("tm")
  # choose repository 69 (USA)
- 69 did not work
- 70 worked
- Now we try again

End 2012 notes

example(wordcloud)

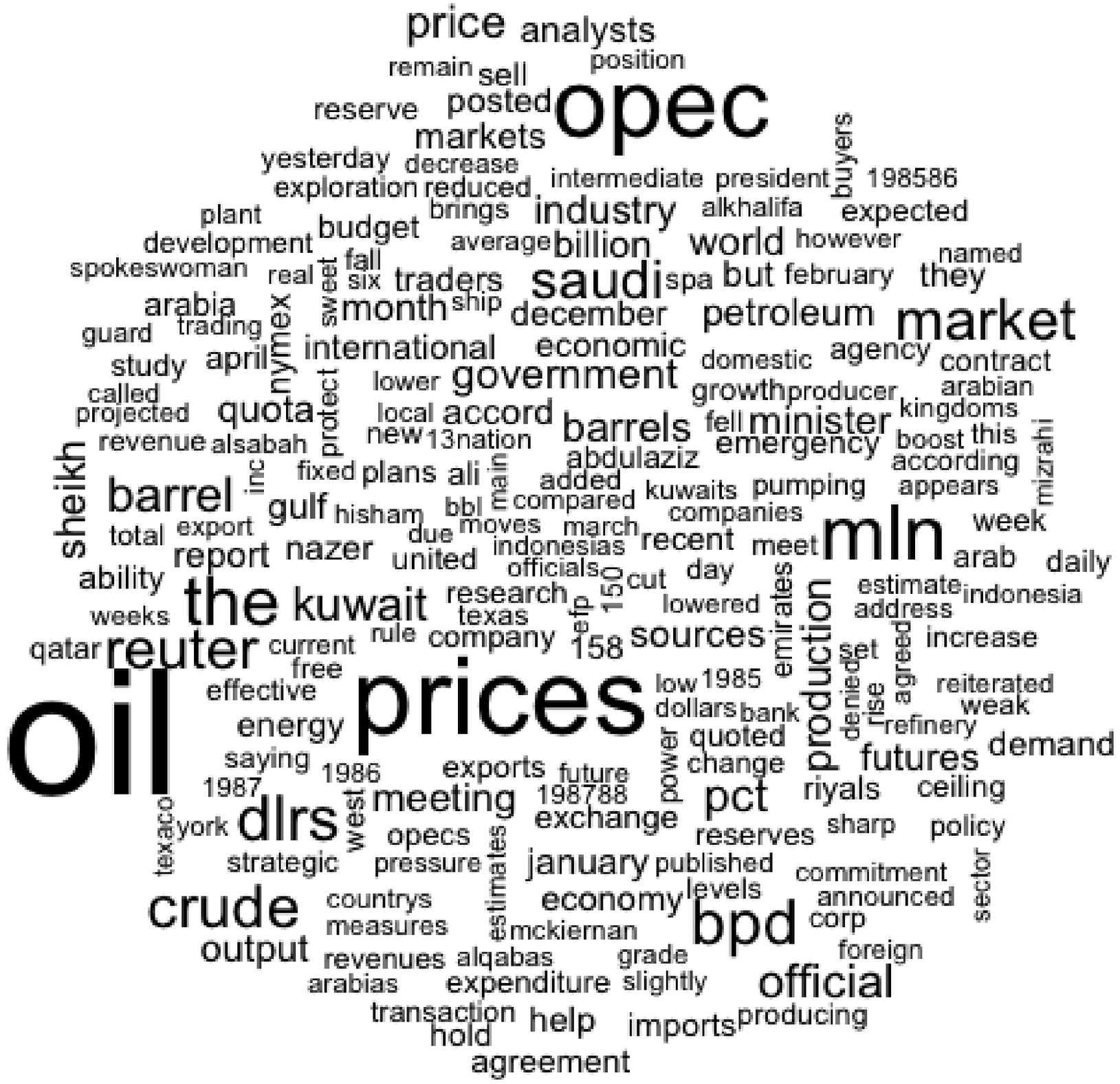
```
wrdcld> if(require(tm)){
wrdcld+  data(crude)
wrdcld+  crude <- tm_map(crude, removePunctuation)
wrdcld+  crude <- tm_map(crude, function(x)removeWords(x,stopwords()))
wrdcld+  tdm <- TermDocumentMatrix(crude)
…..

wrdcld+
wrdcld+  wordcloud(d$word,d$freq,c(8,.3),2,100,TRUE,,.15,pal,vfont=c("script","plain"))
wrdcld+
wrdcld+  wordcloud(d$word,d$freq,c(8,.3),2,100,TRUE,,.15,pal,vfont=c("serif","plain"))
wrdcld+ }
Loading required package: tm
Warning message:
In library(package, lib.loc = lib.loc, character.only = TRUE, logical.return = TRUE,  :
  there is no package called 'tm'
```

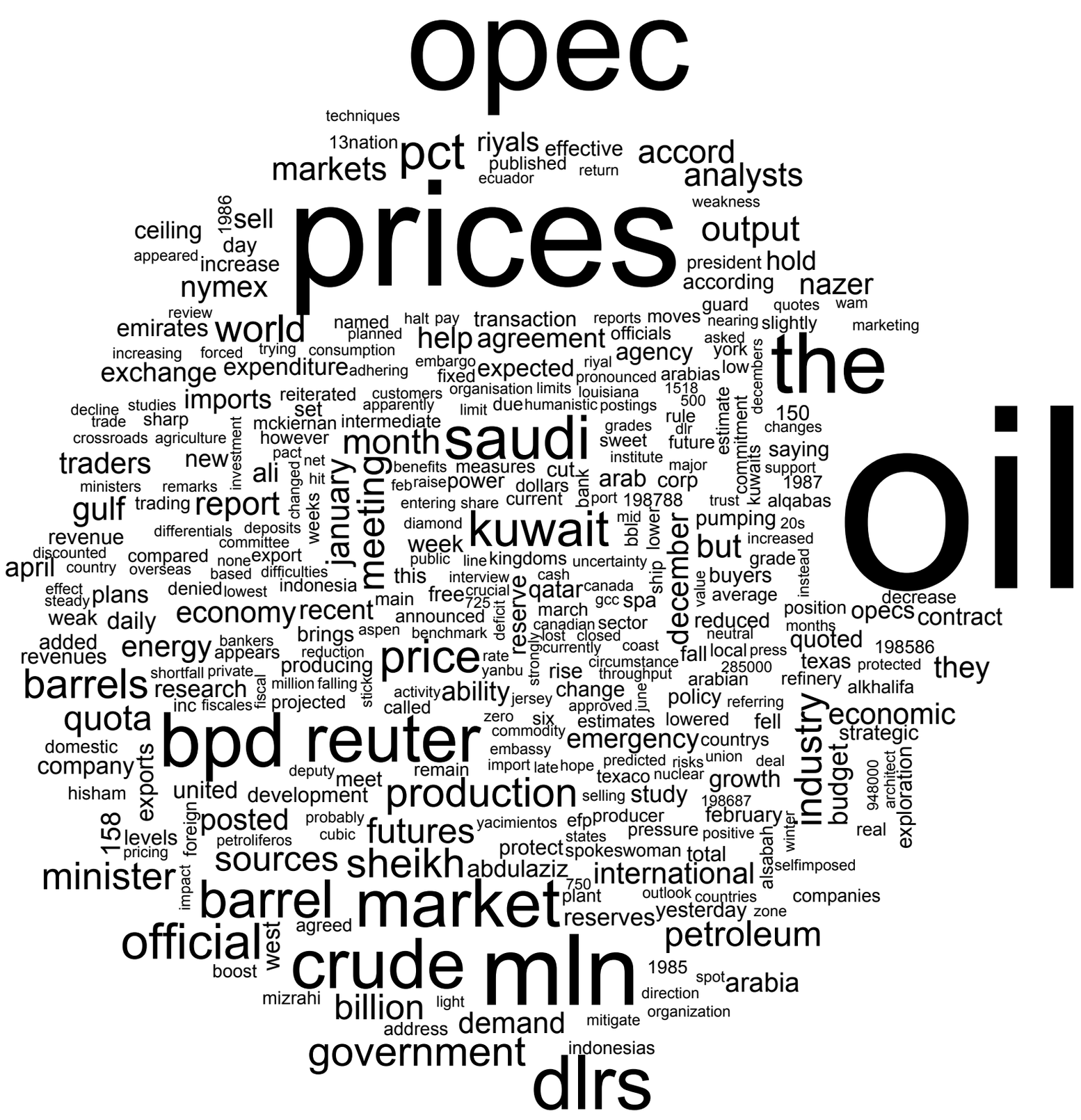
```
if(require(tm)){
    data(crude)
    crude <- tm_map(crude, removePunctuation)
    crude <- tm_map(crude, function(x)removeWords(x,stopwords()))
    tdm <- TermDocumentMatrix(crude)
    m <- as.matrix(tdm)
    v <- sort(rowSums(m),decreasing=TRUE)
    d <- data.frame(word = names(v),freq=v)

    wordcloud(d$word,d$freq)

    #A bigger cloud with a minimum frequency of 2
    wordcloud(d$word,d$freq,c(8,.3),2)

    #Now lets try it with frequent words plotted first
    wordcloud(d$word,d$freq,c(8,.5),2,,FALSE,.1)

    #####           with colors         #####
    if(require(RColorBrewer)){

        pal <- brewer.pal(9,"BuGn")
        pal <- pal[-(1:4)]
        wordcloud(d$word,d$freq,c(8,.3),2,,FALSE,,.15,pal)


        pal <- brewer.pal(6,"Dark2")
        pal <- pal[-(1)]
        wordcloud(d$word,d$freq,c(8,.3),2,,TRUE,,.15,pal)

        #random colors
        wordcloud(d$word,d$freq,c(8,.3),2,,TRUE,TRUE,.15,pal)
    }
    #####           with font           #####

    wordcloud(d$word,d$freq,c(8,.3),2,,TRUE,,.15,pal,
        vfont=c("gothic english","plain"))

    wordcloud(d$word,d$freq,c(8,.3),2,100,TRUE,,.15,pal,vfont=c("script","plain"))

    wordcloud(d$word,d$freq,c(8,.3),2,100,TRUE,,.15,pal,vfont=c("serif","plain"))
}
```

- Two results from running

  example(wordcloud)

- There are more


price
analysts
position
remain
sell
opec
reserve
posted
markets
buyers
yesterday
decrease
intermediate
president
198586
exploration
reduced
industry
alkhalifa
expected
plant
brings
budget
development
average
billion
world
however
named
spokeswoman
real
fall
six
traders
saudi
spa
but
february
they
arabia
sweet
month
ship
december
petroleum
market
guard
trading
nymex
international
economic
agency
contract
study
april
government
domestic
lower
growth
producer
arabian
called
projected
quota
protect
local
accord
kingdoms
barrels
fell
minister
revenue
alsabah
new
13nation
emergency
boost
this
abdulaziz
according
sheikh
barrel
inc
fixed
plans
ali
main
added
kuwaits
pumping
appears
mizrahi
gulf
hisham
bbl
compared
companies
mln
week
total
export
due
moves
march
recent
meet
report
nazer
united
indonesias
arab
daily
ability
officials
cut
day
estimate
the
kuwait
research
150
efp
lowered
emirates
production
indonesia
address
weeks
texas
sources
increase
qatar
reuter
current
rule
company
158
agreed
set
free
low
1985
reiterated
effective
prices
denied
rise
weak
dollars
bank
refinery
energy
quoted
demand
oil
power
saying
1986
exports
future
change
futures
1987
meeting
198788
pct
riyals
ceiling
texaco
york
dlrs
west
opecs
exchange
reserves
sharp
policy
strategic
pressure
january
published
commitment
estimates
sector
crude
countrys
economy
levels
announced
bpd
measures
mckiernan
corp
output
revenues
alqabas
grade
foreign
arabias
expenditure
slightly
official
transaction
help
imports
producing
hold
agreement


opec
prices
Oil
techniques
13nation
pct
riyals
effective
accord
markets
published
ecuador
return
analysts
weakness
1986
sell
ceiling
day
appeared
increase
output
president
hold
nymex
according
nazer
guard
quotes
wam
review
emirates
world
named
halt
pay
transaction
reports
moves
nearing
slightly
marketing
planned
help
agreement
officials
asked
the
increasing
forced
trying
consumption
agency
york
exchange
expenditure
adhering
embargo
expected
riyal
fixed
pronounced
arabias
low
imports
reiterated
customers
organisation
limits
louisiana
1518
decline
studies
apparently
due
humanistic
postings
500
estimate
decembers
150
trade
sharp
mckiernan
intermediate
limit
rule
dlr
changes
crossroads
agriculture
however
month
saudi
grades
sweet
future
saying
traders
new
pact
net
benefits
measures
cut
institute
major
commitment
kuwaits
support
investment
ali
changed
hit
feb
raise
power
dollars
bank
arab
corp
1987
ministers
remarks
gulf
trading
report
weeks
january
meeting
entering
share
current
port
198788
trust
alqabas
differentials
deposits
diamond
kuwait
week
bbl
lower
pumping
20s
revenue
committee
but
increased
discounted
compared
export
public
line
kingdoms
uncertainty
grade
april
country
overseas
none
based
difficulties
this
interview
cash
buyers
instead
effect
steady
plans
denied
lowest
indonesia
qatar
canada
ship
value
average
decrease
main
free
crucial
725
gcc
spa
position
opecs
contract
weak
daily
economy
recent
announced
march
reserve
december
deficit
canadian
sector
months
reduced
added
energy
bankers
brings
aspen
benchmark
lost
closed
neutral
quoted
198586
revenues
appears
reduction
price
rate
strongly
currently
coast
fall
local
press
texas
protected
they
barrels
shortfall
private
producing
yanbu
rise
circumstance
throughput
285000
arabian
refinery
alkhalifa
research
million
falling
stick
activity
ability
jersey
change
june
policy
referring
inc
fiscales
fiscal
projected
called
approved
industry
economic
quota
bpd
reuter
zero
six
commodity
estimates
lowered
fell
strategic
domestic
embassy
emergency
countrys
company
exports
deputy
meet
remain
import
late
hope
predicted
risks
union
deal
budget
948000
architect
exploration
hisham
united
development
production
selling
study
growth
158
foreign
posted
probably
futures
yacimientos
efp
producer
february
levels
pricing
petroliferos
cubic
states
pressure
positive
winter
real
protect
spokeswoman
total
minister
sources
sheikh
abdulaziz
international
alsabah
selfimposed
impact
barrel
market
750
plant
outlook
countries
companies
official
agreed
reserves
yesterday
zone
petroleum
west
crude
mln
1985
spot
arabia
boost
mizrahi
billion
light
direction
organization
address
demand
mitigate
government
indonesias
dlrs

- data(crude)
  wordcloud(crude)
  wordcloud(crude[2])
  wordcloud(crude[3:10])

- crude: corpus (collection) of 20 texts
  ?crude

How does it work?

- We analyze source code
- In the process, we learn more R
- We learn to accomplish more tasks
- Maybe we learn to apply what we learn to different problems
- We start off knowing nothing of how **wordcloud** works

```
if(require(tm)){
        data(crude)
        crude <- tm_map(crude, removePunctuation)
        crude <- tm_map(crude, function(x)removeWords(x,stopwords()))
        tdm <- TermDocumentMatrix(crude)
        m <- as.matrix(tdm)
        v <- sort(rowSums(m),decreasing=TRUE)
        d <- data.frame(word = names(v), freq=v)
```

In blue are functions we have never seen before

require: similar to library, but returns TRUE if already loaded
If require cannot be loaded, returns FALSE, and the code below is not executed. This helps avoid errors if there if not executing this section of code does not hurt the rest of the program

tm_map

Interface to apply transformation functions (also denoted as mappings) to corpora.

Usage:

```
## S3 method for class 'PCorpus'
tm_map(x, FUN, ..., useMeta = FALSE, lazy = FALSE)
## S3 method for class 'VCorpus'
tm_map(x, FUN, ..., useMeta = FALSE, lazy = FALSE)
```

tm_map

Arguments:

x: A corpus. #???? **What is a corpus?**

FUN: A transformation function returning a text document.

...: Arguments to ‘FUN’.

corpus

Help files with alias or concept or title matching ‘corpus’ using fuzzy matching:

tm::PCorpus	Permanent Corpus Constructor
tm::VCorpus	Volatile Corpus
tm::Zipf_plot	Explore Corpus Term Frequency Characteristics
tm::c.Corpus	Combine Corpora, Documents, and Term-Document Matrices
tm::inspect	Inspect Objects
tm::makeChunks	Split a Corpus into Chunks
tm::content_meta<-	Meta Data Management
tm::readRCV1	Read In a Reuters Corpus Volume 1 Document
tm::tm_filter	Filter and Index Functions on Corpora
tm::tm_map	Transformations on Corpora
tm::writeCorpus	Write a Corpus to Disk
timeDate::holidayDate	Public and Ecclesiastical Holidays

Type '?PKG::FOO' to inspect entries 'PKG::FOO', or 'TYPE?PKG::FOO' for entries like 'PKG::FOO-TYPE'.

I still have no idea what a corpus is

Next stop: the web

tm r corpus

About 402,000 results (0.32 seconds)

[PDF] Introduction to the **tm** Package Text Mining in **R**
cran.**r**-project.org/web/packages/**tm**/vignettes/**tm**.pdf
File Format: PDF/Adobe Acrobat - Quick View
by I Feinerer - 2011 - Cited by 3 - Related articles
object is not destroyed if the corresponding **R** object is released. Within the **corpus** constructor, x must be a Source object which abstracts the input location. **tm** ...

[PDF] Package '**tm**'
cran.**r**-project.org/web/packages/**tm**/**tm**.pdf
File Format: PDF/Adobe Acrobat - Quick View
by I Feinerer - 2012 - Cited by 1 - Related articles
Feb 3, 2012 – reut21578 <- system.file("texts", "crude", package = "**tm**") **r** <- **Corpus**(DirSource(reut21578),. readerControl = list(reader = readReu?1578XML)) ...

Text Data Mining with Twitter and **R** | Heuristic An
heuristically.wordpress.com/2011/04/08/text-data-mining-twitter-r/
Apr 8, 2011 – There is a specialized package for **R** called twitteR, but it isn't available for .... install.packages("**tm**.plugin.**webcorpus**", repos="http://**R**-Forge.

TM

http://cran.r-project.org/web/

Data Import
The main **structure** for managing documents in tm is a so-called Corpus, representing a collection of text documents. A corpus is an abstract concept, and there can exist several implementations in parallel. The default implementation is the so-called VCorpus (short for Volatile Corpus) which realizes a semantics as known from most R objects: corpora are R objects held fully in memory. We denote this as volatile since once the R object is destroyed, the whole corpus is gone. Such a volatile corpus can be created via the constructor Corpus(x, readerControl). Another implementation is the PCorpus which implements a Permanent Corpus semantics, i.e., the documents are physically stored outside of R (e.g., in a database), corresponding R objects are basically only pointers to external structures, and changes to the underlying corpus are reflected to all R objects associated with it. Compared to the volatile corpus the corpus encapsulated by a permanent corpus object is not destroyed if the corresponding R object is released.

What are structures?

- Vectors = collection of objects of same type
- Data.frames = list of columns constructed from vectors
- Class = structure composed of other objects and also functions
- Wave = sound structure which is a type of class
- Corpus = another type of structure, as yet unknown to us

Now in English

- Start with a text document
- Somehow this text document is transformed in a type of structure called a **corpus**
- We have already come across other structures (vector, list, data.frame, Wave)
- This gives me an idea …
  - look at the line **data(crude)**

Exploration commands

- class()
- str()
- summary()
- ?, ??
- names()

```
> data(crude)
> crude
A corpus with 20 text documents


> class(crude)
[1] "VCorpus" "Corpus"  "list"
```

```
> class(crude[1:3])
[1] "VCorpus" "Corpus"  "list"


> class(crude[[1]])
[1] "PlainTextDocument" "TextDocument"      "character"
```

[...] versus[[...]]

```
df1 = data.frame(name=c("john","mary"), age=c(50,60))
df2 = data.frame(cost=c(45,50,55), year=c(1957,1960,1963))
df3 = data.frame(c(1:10), rnorm(10), temp=90+rnorm(10))
```

```
frames = c(df1, df2, df3)

frames[1:2]
frames[1]

frames[[1]]
```

A **list** (can hold objects of *different* types)

[...] of a list returns another list
[...] of a data.frame returns another data.frame

[[...]] returns a single element from the list or a single column from a data.frame

[...] versus[[...]]

Imagine you have a "bag" filled with data.frames
Use [...] to select a subset of these data.frames.

As a special case, you can select a single data.frame,
for example: frames[3] returns a single data.frame from the collection.

Use [[...]] to select one column from the data.frame

[[...]] is used to select the "contents" of a single object

[...] versus[[...]]

```
df = data.frame(1:2, c("mary","john"))
grades = c(10,20,30,50)
cond = c(T,F,F,F,F)
```

2 columns
4 grades
5 conditions

```
collection = list(df, grades, cond)
collection[1]      # returns a list of one element
collection[2:3]   # returns a list of two elements


collection[[2:3]]  # illegal
collection[[2]]     # vector of numbers (grades)
collection[[1]]     # a data.frame
```

Store the above in collection.r
script("collection.r", echo=T)

```
> txt = crude[[1]]
> txt
```
Diamond Shamrock Corp said that
effective today it had cut its contract prices for crude oil by
1.50 dlrs a barrel.

The reduction brings its posted price for West Texas Intermediate to 16.00 dlrs a barrel, the company said.

"The price reduction today was made in the light of falling oil product prices and a weak crude oil market," a company spokeswoman said.

Diamond is the latest in a line of U.S. oil companies that have cut its contract, or posted, prices over the last two days citing weak oil markets.

Reuter

>

```
> str(txt)

Classes 'PlainTextDocument', 'TextDocument', 'character'  atomic [1:1] Diamond Shamrock Corp said that

effective today it had cut its contract prices for crude oil by 1.50 dlrs a barrel. The reduction brings its posted
price for West Texas Intermediate to 16.00 dlrs a barrel, the copany said.  "The price reduction today was
made in the light of falling oil product prices and a weak crude oil market," a company spokeswoman said.

    Diamond is the latest in a line of U.S. oil companies that have cut its contract, or posted, prices over the last
two days citing weak oil markets. Reuter

  ..- attr(*, "Author")= chr(0)
  ..- attr(*, "DateTimeStamp")= POSIXlt[1:1], format: "1987-02-26 17:00:56"
  ..- attr(*, "Description")= chr ""
  ..- attr(*, "Heading")= chr "DIAMOND SHAMROCK (DIA) CUTS CRUDE PRICES"
  ..- attr(*, "ID")= chr "127"
  ..- attr(*, "Language")= chr "en"
  ..- attr(*, "LocalMetaData")=List of 9

.. ..$ TOPICS    : chr "YES"
.. ..$ LEWISSPLIT: chr "TRAIN"
.. ..$ CGISPLIT  : chr "TRAINING-SET"
.. ..$ OLDID     : chr "5670"
.. ..$ Topics    : chr "crude"
 .. ..$ Places    : chr "usa"
 .. ..$ People    : chr(0)
  .. ..$ Orgs      : chr(0)
  .. ..$ Exchanges : chr(0)
  ..- attr(*, "Origin")= chr "Reuters-21578 XML"
```

str(txt)

Attributes are metadata that add additional information on an object

We are exploring ...

Access to txt metadata

```
> str(txt
> attr(txt,"LocalMetaData")$Topics
[1] "crude"
> attr(txt,"LocalMetaData")["Topics"]
$Topics
[1] “crude"
```

names(txt) does not work

Metadata

- Photos have metadata:
  - author
  - time the picture was taken
  - type of camera, lens, flash, etc.
  - etc.

From iPhoto

names(crude)

```
> names(crude)
 [1] "reut-00001.xml" "reut-00002.xml" "reut-00004.xml" "reut-00005.xml"
 [5] "reut-00006.xml" "reut-00007.xml" "reut-00008.xml" "reut-00009.xml"
 [9] "reut-00010.xml" "reut-00011.xml" "reut-00012.xml" "reut-00013.xml"
[13] "reut-00014.xml" "reut-00015.xml" "reut-00016.xml" "reut-00018.xml"
[17] "reut-00019.xml" "reut-00021.xml" "reut-00022.xml" "reut-00023.xml"
```

```
> crude$reut-00006.xml
Error: unexpected symbol in "crude$reut-00006.xml"
```

Probably the “-” that is creating problems since
it is probably interpreted as a negation operator

```
> crude[[3]]
Texaco Canada said it lowered the
contract price it will pay for crude oil 64 Canadian cts a
barrel effective today
    The decrease brings the companys posted price for the
benchmark grade EdmontonSwann Hills Light Sweet to 2226
Canadian dlrs a bbl
    Texaco Canada last changed its crude oil postings on Feb
19
 Reuter
```

```
crude2 <- tm_map(crude, function(x)removeWords(x,stopwords()))
```

```
> crude2[[3]]
Texaco Canada   lowered
contract price   pay  crude oil 64 Canadian cts
barrel effective
    The decrease brings  companys posted price
benchmark grade EdmontonSwann Hills Light Sweet  2226
Canadian dlrs  bbl
    Texaco Canada  changed  crude oil postings  Feb
19
 Reuter
```

```
> crude1[[3]]
Texaco Canada said it lowered the
contract price it will pay for crude oil 64 Canadian cts a
barrel, effective today.
    The decrease brings the company's posted price for the
benchmark grade, Edmonton/Swann Hills Light Sweet, to 22.26
Canadian dlrs a bbl.
    Texaco Canada last changed its crude oil postings on Feb
19.
```

crude1 <- **tm_map**(crude, **removePunctuation**)

```
> crude1[[3]]
Texaco Canada said it lowered the
contract price it will pay for crude oil 64 Canadian cts a
barrel effective today
    The decrease brings the companys posted price for the
benchmark grade EdmontonSwann Hills Light Sweet to 2226
Canadian dlrs a bbl
    Texaco Canada last changed its crude oil postings on Feb
19
 Reuter
```

punctuation removed

stopWords()

No longer exists on current version of R.
Find it.
Use findFn()

Use stopwords() instead

```
[356] "smaller"    "smallest"   "so"         "some"       "somebody"
[361] "someone"    "something"  "somewhere"  "state"      "states"
[366] "still"      "such"       "sure"       "t"          "take"
[371] "taken"      "than"       "that"       "that's"     "the"
[376] "their"      "theirs"     "them"       "themselves" "then"
[381] "there"      "therefore"  "there's"    "these"      "they"
[386] "they'd"     "they'll"    "they're"    "they've"    "thing"
[391] "things"     "think"      "thinks"     "this"       "those"
[396] "though"     "thought"    "thoughts"   "three"      "through"
[401] "thus"       "to"         "today"      "together"   "too"
[406] "took"       "toward"     "turn"       "turned"     "turning"
```

Some of the stopwords

```
71] "clear"       "clearly"     "come"        "could"       "couldn't"
 [76] "d"           "did"         "didn't"      "differ"      "different"
 [81] "differently" "do"          "does"        "doesn't"     "doing"
 [86] "done"        "don't"       "down"        "downed"      "downing"
 [91] "downs"       "during"      "e"           "each"        "early"
 [96] "either"      "end"         "ended"       "ending"      "ends"
[101] "enough"      "even"        "evenly"      "ever"        "every"
[106] "everybody"   "everyone"    "everything"  "everywhere"  "f"
[111] "face"        "faces"       "fact"        "facts"       "far"
```

```
if(require(tm)){
        data(crude)
        crude <- tm_map(crude, removePunctuation)
        crude <- tm_map(crude, function(x)removeWords(x,stopwords()))
        tdm <- TermDocumentMatrix(crude)
```

identical to

```
my.remove = function(x, words) {
     return(removeWords(x, stopwords()))
}

if(require(tm)){
            data(crude)
            crude <- tm_map(crude, removePunctuation)
            crude <- tm_map(crude, my.remove)
            tdm <- TermDocumentMatrix(crude)
```

```
if(require(tm)){
        data(crude)
        crude <- tm_map(crude, removePunctuation)
        crude <- tm_map(crude, function(x)removeWords(x,stopwords()))
        tdm <- TermDocumentMatrix(crude)
        m <- as.matrix(tdm)
        v <- sort(rowSums(m),decreasing=TRUE)
        d <- data.frame(word = names(v), freq=v)
```

What is **TermDocumentMatrix** ?

?TermDocumentMatrix

Term-Document Matrix

Description:

Constructs or coerces to a term-document matrix or a document-term matrix.

Usage:

```
TermDocumentMatrix(x, control = list())
DocumentTermMatrix(x, control = list())
as.TermDocumentMatrix(x, ...)
as.DocumentTermMatrix(x, ...)
```

?TermDocumentMatrix

Arguments:

x: a corpus for the constructors and either a term-document matrix or a document-term matrix or a simple triplet matrix (package 'slam') for the coercing functions.

control: a named list of control options. The component 'weighting' must be a weighting function capable of handling a 'TermDocumentMatrix'. **It defaults to 'weightTf' for term frequency weighting**. All other options are delegated internally to a 'termFreq' call.

...: the additional argument 'weighting' (typically a 'WeightFunction') is allowed when coercing a simple triplet matrix to a term-document or document-term matrix.

In English

- TermDocumentMatrix(…)
  - converts a corpus (essentially a document) to a special format according to specification of its arguments
  - this special format is used to help plot the word cloud

```
tdm <- TermDocumentMatrix(crude)
m <- as.matrix(tdm)
v <- sort(rowSums(m),decreasing=TRUE)
d <- data.frame(word = names(v), freq=v)
```

The above lines give us clues as to what is going on. We start with

```
m <- as.matrix(tdm)
```

crude

- crude is a corpus. It contains how many documents?
- > length(crude)
  20
- > dim(as.matrix(tdm))
  1132 20 # 1132 words in 20 docs

Test hypothesis

- Create a small document

  This class required much hard work and this work took many hard hours of work

- I constructed a sentence with word repetition to make testing more meaningful

Step 1

- Transform text to a corpus
- I do not know how
- ??corpus

tm::PCorpus Permanent Corpus Constructor # possibly what I want
tm::VCorpus Volatile Corpus
tm::Zipf_plot Explore Corpus Term Frequency Characteristics
tm::c.Corpus Combine Corpora, Documents, and Term-Document Matrices
tm::inspect Inspect Objects
tm::makeChunks Split a Corpus into Chunks
tm::content_meta<- Meta Data Management
tm::readRCV1 Read In a Reuters Corpus Volume 1 Document
tm::tm_filter Filter and Index Functions on Corpora
tm::tm_map Transformations on Corpora
tm::writeCorpus Write a Corpus to Disk
timeDate::holidayDate Public and Ecclesiastical Holidays

Reminder

- I'd like to read an article
- Print out the word cloud
- Count word frequencies
- Use in analysis so I can compare two articles with one another
- HOW???

Text file to Corpus

Google corpus reading data r

Search About 19,300,000 results (0.30 seconds)

Everything
Images
Maps
Videos
News
Shopping
More

Tallahassee, FL
Change location

All results
Related searches
More search tools

[PDF] Statistical Analysis of **Corpus Data** with **R** Outline Why do we ne…
cogsci.uni-osnabrueck.de/~severt/SIGIL/sigil_**R**/.../introduction.4up....
File Format: PDF/Adobe Acrobat - Quick View
Introduction to **R**: set-up, **data** manipulation and **...** The limitations of random sampling models for **corpus data**. **R** **....** **R** can also **read** and write files in CSV format **...**

Corpus Readers
nltk.googlecode.com/svn/trunk/doc/howto/**corpus**.html
The nltk.**corpus** package defines a collection of **corpus reader** classes, which can be **...** Each **corpus reader** provides a variety of methods to **read data** from the **corpus** **.....** for record in rotokas[1:]: ... lexeme = record.find('lx').text ... if re.match(**r**'(.

[PDF] Introduction to the tm Package Text Mining in **R**
cran.**r**-project.org/web/packages/tm/vignettes/tm.pdf
File Format: PDF/Adobe Acrobat - Quick View
by I Feinerer - 2011 - Cited by 3 - Related articles
readerControl = list(**reader** = readReut21578XML)). **Data** Export. For the case you have created a **corpus** via manipulating other objects in **R**, thus do not have **...**
You've visited this page 3 times. Last visit: 4/9/12

One **R** Tip A Day: Word Cloud in **R**
onertipaday.blogspot.com/2011/07/word-cloud-in-**r**.html

onertipaday.blogspot.com/2011/07/word-cloud-in-r.html

A big computer, a complex algorithm and a long time does not equal science

MERCOLEDÌ 27 LUGLIO 2011

Word Cloud in R

A word cloud (or tag cloud) can be an handy tool when you need to highlight the most commonly cited words in a text using a quick visualization. Of course, you can use one of the several on-line services, such as wordle or tagxedo , very feature rich and with a nice GUI. Being an R enthusiast, I always wanted to produce this kind of images within R and now, thanks to the recently released Ian Fellows' wordcloud package, finally I can!
In order to test the package I retrieved the titles of the XKCD web comics included in my RXKCD package and produced a word cloud based on the titles' word frequencies calculated using the powerful tm package for text mining (I know, it is like killing a fly with a bazooka!).

```
1  library(RXKCD)
2  library(tm)
3  library(wordcloud)
4  library(RColorBrewer)
5  path <- system.file("xkcd", package = "RXKCD")
6  datafiles <- list.files(path)
7  xkcd.df <- read.csv(file.path(path, datafiles))
8  xkcd.corpus <- Corpus(DataframeSource(data.frame(xkcd.df[, 3])))
9  xkcd.corpus <- tm_map(xkcd.corpus, removePunctuation)
10 xkcd.corpus <- tm_map(xkcd.corpus, tolower)
11 xkcd.corpus <- tm_map(xkcd.corpus, function(x) removeWords(x, stopwords("engl
12 tdm <- TermDocumentMatrix(xkcd.corpus)
13 m <- as.matrix(tdm)
14 v <- sort(rowSums(m),decreasing=TRUE)
15 d <- data.frame(word = names(v),freq=v)
16 pal <- brewer.pal(9, "BuGn")
17 pal <- pal[-(1:2)]
18 png("wordcloud.png", width=1280,height=800)
19 wordcloud(d$word,d$freq, scale=c(8,.3),min.freq=2,max.words=100, random.order
20 dev.off()
```

but very often we have whole texts. I will be very grateful if you can make a world cloud using a txt file.

Noam

Rispondi

Paolo Jul 29, 2011 11:35 PM

Dear Noam,
You can find both the answer to your question and a nice introduction to text mining in R in the vignette of the tm package:
install.packages("tm")
library("tm")
vignette("tm")
HIH!

Rispondi

Noam López Jul 31, 2011 09:09 AM

Thank you Paolo, I'm going to read about the tm package. This is my first meet with text mining bacause I just use R for my classes of statistics.

Noam

Rispondi

Paolo Jul 31, 2011 11:23 AM

You are welcome Noam!

Use Vignette("tm")

Introduction to the **tm** Package
Text Mining in R

Ingo Feinerer

April 15, 2011

Introduction

This vignette gives a short introduction to text mining in R utilizing the text mining framework provided by the **tm** package. We present methods for data import, corpus handling, preprocessing, meta data management, and creation of term-document matrices. Our focus is on the main aspects of getting started with text mining in R—an in-depth description of the text mining infrastructure offered by **tm** was published in the *Journal of Statistical Software* (Feinerer et al., 2008). An introductory article on text mining in R was published in *R News* (Feinerer, 2008).

Data Import

The main structure for managing documents in **tm** is a so-called Corpus, representing a collection of text documents. A corpus is an abstract concept, and there can exist several implementations in parallel. The default implementation is the so-called VCorpus (short for *Volatile Corpus*) which realizes a semantics as known from most R objects: corpora are R objects held fully in memory. We denote this as volatile since once the R object is destroyed, the whole corpus is gone. Such a volatile corpus can be created via the constructor `Corpus(x, readerControl)`. Another implementation is the PCorpus which implements a *Permanent Corpus* semantics, i.e., the documents are physically stored outside of R (e.g., in a database), corresponding R objects are basically only pointers to external structures, and changes to the underlying corpus are reflected to all R objects associated with it. Compared to the volatile corpus the corpus encapsulated by a permanent corpus object is not destroyed if the corresponding R object is released.

Within the corpus constructor, `x` must be a Source object which abstracts the input location. **tm** provides a set of predefined sources, e.g., DirSource, VectorSource, or DataframeSource, which handle a directory, a vector interpreting each component as document, data frame like structures (like CSV files), respectively. Except DirSource, which is designed solely for directories on a file system, and VectorSource, which only accepts (character) vectors, most other implemented sources can take connections as input (a character string is interpreted as file path). `getSources()` lists available sources, and users can create their own sources.

The second argument `readerControl` of the corpus constructor has to be a list with the named components `reader` and `language`. The first component `reader` constructs a text document from elements delivered by a source. The **tm** package ships with several readers (e.g., `readPlain()`, `readGmane()`, `readRCV1()`, `readReut21578XMLasPlain()`, `readPDF()`, `readDOC()`, ...). See `getReaders()` for an up-to-date list of available readers. Each source has a default reader which can be overridden. E.g., for `DirSource` the default just reads in the input files and interprets their content as text. Finally, the second component `language` sets the texts' language (preferably using ISO 639-2 codes).

Within the corpus constructor, x must be a Source object which abstracts the input location. tm provides a set of predefined sources, e.g., DirSource, VectorSource, or DataframeSource, which handle a directory, a vector interpreting each component as document, data frame like structures (like CSV files), respectively. Except **DirSource, which is designed solely for directories on a file system**, and **VectorSource, which only accepts (character) vectors**, most other implemented sources can take connections as input (a character string is interpreted as file path). **getSources() lists available sources**, and users can create their own sources.

The second argument readerControl of the corpus constructor has to be a list with the named components reader and language. The first component reader constructs a text document from elements delivered by a source. **The tm package ships with several readers (e.g., readPlain(), readGmane(), readRCV1(), readReut21578XMLasPlain(), readPDF(), readDOC(), . . .).** See getReaders() for an up-to-date list of avail- able readers. Each source has a default reader which can be overridden. E.g., for DirSource the default just reads in the input files and interprets their content as text. Finally, the second component language sets the texts' language (preferably using ISO 639-2 codes).

Let us experiment with a few readers …

plain text ==> corpus
pdf file ==> corpus
doc file ==> corpus (Microsoft Word)

getSources()

```
> getSources()

[1] "DataframeSource" "DirSource"       "GmaneSource"
"ReutersSource"

[5] "URISource"       "VectorSource"
```

getReaders()

```
> getReaders()
 [1] "readDOC"                  "readGmane"
 [3] "readPDF"                  "readReut21578XML"
 [5] "readReut21578XMLasPlain" "readPlain"
 [7] "readRCV1"                 "readRCV1asPlain"
 [9] "readTabular"              "readXML"
```

Experiment with readers

- I create a word file, a pdf file, and a plain text file

So e.g., plain text files in the directory txt containing Latin (lat) texts by the Roman poet Ovid can be read in with following code:

```
> txt <- system.file("texts", "txt", package = "tm")
> (ovid <- Corpus(DirSource(txt),
+ readerControl = list(language = "lat")))

A corpus with 5 text documents
```

For simple examples VectorSource is quite useful, as it can create a corpus from character vectors, e.g.:

```
> docs <- c("This is a text.", "This another one.")
> Corpus(VectorSource(docs))
```

Data Export

To create a corpus from text *not* stored on a hard disk, and want to save the text documents to disk, use writeCorpus()

> **writeCorpus(ovid)**

which writes a plain text representation of a corpus to multiple files on disk corresponding to the individual documents in the corpus.

Try it out!

- Starting with my own text:

"what is the meaning of this grade?"

- Transform the text into a **corpus**

```
library(tm)

txt = "What is the matter with this grade?"
cp = Corpus(VectorSource(txt))
print(class(cp))
```

```
> source("text_to_corpus.r")
[1] "VCorpus" "Corpus"  "list"
```

txt is a character vector
cp is a corpus

Start with
a single text file

- Idea
  - Since I know how to transform simple text into a corpus, …
  - Try and read a text file by assuming it is a csv file

Course on Corpus. Looks pretty extensive.
http://cogsci.uni-osnabrueck.de/~severt/SIGIL/sigil_R/

```
library("wordcloud")
library(tm)

v = VectorSource("I, of sound mind, have to go.")
c = Corpus(v)

removePunctuation(PlainTextDocument(v))
```

==========

```
removePunct = function(text) {
  v = VectorSource(text)
  c = Corpus(v)
  t = removePunctuation(PlainTextDocument(v))
  return(t)
}
```

```
removePunct("what, is up, Doc?")
```

will return

"what is up Doc"

File:

README

R Code

```
library(tm)

txt.file = "README"
txt = read.csv(txt.file, header=F)
print(txt)
```

```
> class(txt)
[1] "data.frame"
```

txt is not yet a Corpus

Convert to Corpus

```
cp = Corpus(VectorSource(txt))
print(class(cp))
print("----------------")
print(cp[[1]])
```

Output

```
sourc> source("file_to_corpus.r")
                                                    V1
1                          Course on Corpus. Looks pretty extensive.
2                  http://cogsci.uni-osnabrueck.de/~severt/SIGIL/sigil_R/
3  ----------------------------------------------------------------------
4                                          library(wordcloud)
5                                             library(tm)
6                       v = VectorSource(I, of sound mind, have to go.)
7                                             c = Corpus(v)
8                    removePunctuation(PlainTextDocument(v))
9                                        ================
10                          removePunct = function(text) {
11                             v = VectorSource(text)
12                               c = Corpus(v)
13                t = removePunctuation(PlainTextDocument(v))
14                               return(t)
15                                  }
16                          removePunct(what, is up, Doc?)
17                               will return
18                             what is up Doc
```

Return to the wordcloud

- Return to the example

```
data(crude)
crude <- tm_map(crude, removePunctuation)
crude <- tm_map(crude, function(x)removeWords(x,stopwords()))
tdm <- TermDocumentMatrix(crude)
m <- as.matrix(tdm)
v <- sort(rowSums(m),decreasing=TRUE)
d <- data.frame(word = names(v),freq=v)
```

wordcloud(d$word,d$freq)

Functions still unknown:
TermDocumentMatrix
sort

```
> class(tdm)
[1] "TermDocumentMatrix"     "simple_triplet_matrix"
```

Term-Document Matrix

Description:

Constructs or coerces to a term-document matrix or a document-term matrix.

In English: Converts a Corpus to an appropriate structure to transform into regular matrix

Usage:

TermDocumentMatrix(x, control = list())

x: a **corpus** for the constructors and either a term-document matrix or a document-term matrix or a simple triplet matrix (package 'slam') for the coercing functions.

control: a named list of control options.
The component 'weighting' must be a weighting function capable of handling a 'TermDocumentMatrix'. It defaults to 'weightTf' for term frequency weighting. All other options are delegated internally to a 'termFreq' call.

...

```
txt.file = "README"
txt = read.csv(txt.file, header=F)
cp = Corpus(VectorSource(txt))
tdm = TermDocumentMatrix(cp)
m = as.matrix(tdm)
print(m)
```

> m = as.matrix(tdm)

Terms	Docs V1
----------	2
==========	1
corpus.	1
corpus(v)	2
course	1
doc	1
doc?)	1
extensive.	1
function(text)	1
go.)	1
have	1
http://cogsci.uni-osnabrueck.de/~severt/sigil/sigil_r/	1
library(tm)	1
library(wordcloud)	1
looks	1
mind,	1
pretty	1
removepunct	1
removepunct(what,	1
removepunctuation(plaintextdocument(v))	2
return	1
return(t)	1
sound	1
up,	1
vectorsource(i,	1
vectorsource(text)	1
what	1
will	1

Word list + frequency

punctuation is considered a part of the word, which is why we'd like to remove punctuation

as.matrix(tdm)

```
> dim(tdm)
[1] 28  1
> dim(as.matrix(tdm))
[1] 28  1
```

The word cloud

```
m <- as.matrix(tdm)
 v <- sort(rowSums(m),decreasing=TRUE)
 d <- data.frame(word = names(v),freq=v)

 wordcloud(d$word,d$freq)
```

Error!!!

Error in strwidth(words[i], cex = size[i], ...) : invalid 'cex' value
In addition: Warning message:
In max(freq) : no non-missing arguments to max; returning -Inf

```
txt.file = "README"
txt = read.csv(txt.file, header=F)
cp = Corpus(VectorSource(txt))
cp = tm_map(cp, removePunctuation)
tdm = TermDocumentMatrix(cp)
m = as.matrix(tdm)
print(m)
```

```
> m
                                Docs
Terms                            V1
  corpus                          1
  corpusv                         2
  course                          1
  doc                             2
  extensive                       1
  functiontext                    1
  have                            1
  httpcogsciuniosnabrueckdesevertsigilsigilr  1
  librarytm                       1
  librarywordcloud                1
  looks                           1
  mind                            1
  pretty                          1
  removepunct                     1
  removepunctuationplaintextdocumentv  2
  removepunctwhat                 1
  return                          1
  returnt                         1
  sound                           1
  vectorsourcei                   1
  vectorsourcetext                1
  what                            1
  will                            1
```

No more punctuation
Error is still there!

Analysis

```
m <- as.matrix(tdm)
 v <- sort(rowSums(m),decreasing=TRUE)
 d <- data.frame(word = names(v),freq=v)
wordcloud(d$word,d$freq)
```

```
> class(d)
[1] "data.frame"
> names(d)
[1] "word" "freq"
```

We now examine the data frame

```
> d$word
 [1] ----------------------------------------------------------------
 [2] corpus(v)
 [3] removepunctuation(plaintextdocument(v))
 [4] ================
 [5] corpus.
 [6] course
 [7] doc
 [8] doc?)
 [9] extensive.
[10] function(text)
[11] go.)
[12] have
[13] http://cogsci.uni-osnabrueck.de/~severt/sigil/sigil_r/
[14] library(tm)
[15] library(wordcloud)
[16] looks
[17] mind,
[18] pretty
[19] removepunct
[20] removepunct(what,
[21] return
[22] return(t)
[23] sound
[24] up,
[25] vectorsource(i,
[26] vectorsource(text)
[27] what
[28] will
28 Levels: ----------------------------------------------------------------
```

```
> d$freq
 [1] 2 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

The dashed lines “might” be the reason for the error! Let us remove punctuation

What to do now?

- The current situation
  - I cannot get a word cloud to work with my file
- Take original example
  - create a word cloud
  - simplify the example as much as possible without generating an error

Original example

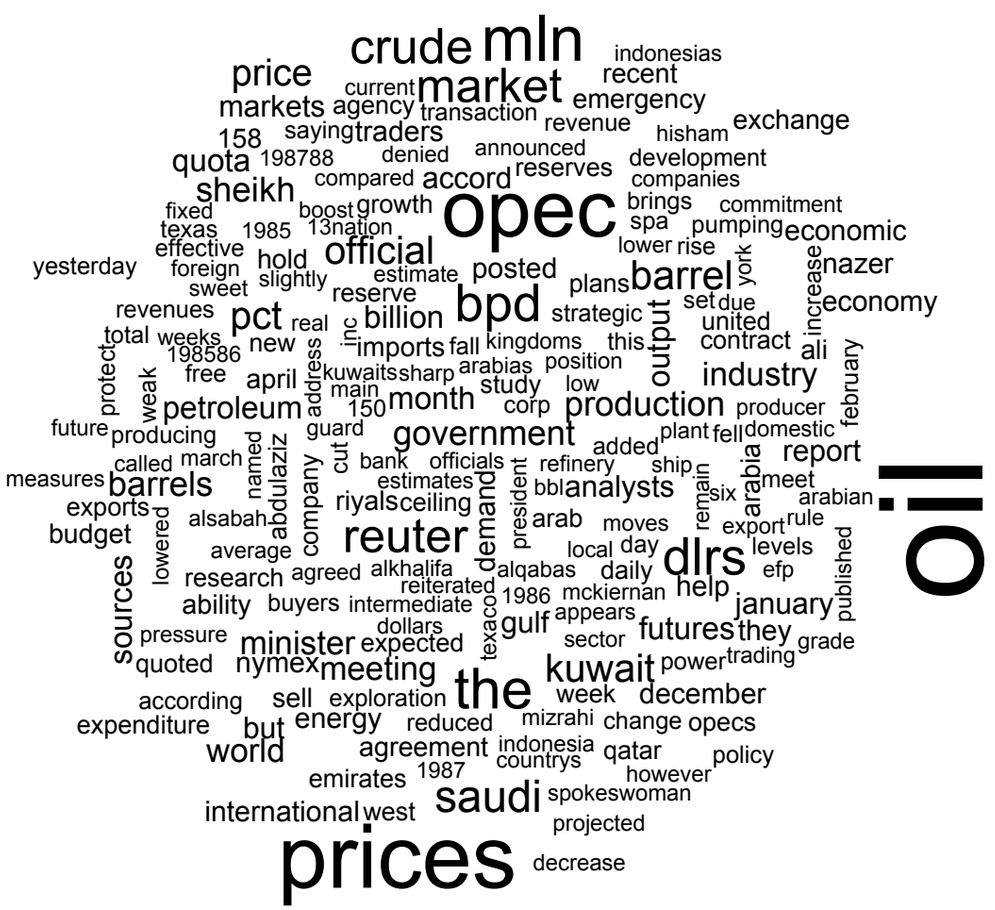
```
library(wordcloud)
library(tm)
data(crude)
crude <- tm_map(crude, removePunctuation)
crude <- tm_map(crude, function(x)removeWords(x,stopwords()))
tdm <- TermDocumentMatrix(crude)
m <- as.matrix(tdm)
v <- sort(rowSums(m),decreasing=TRUE)
 d <- data.frame(word = names(v),freq=v)
wordcloud(d$word,d$freq)
```

Remove (potentially) unnecessary lines

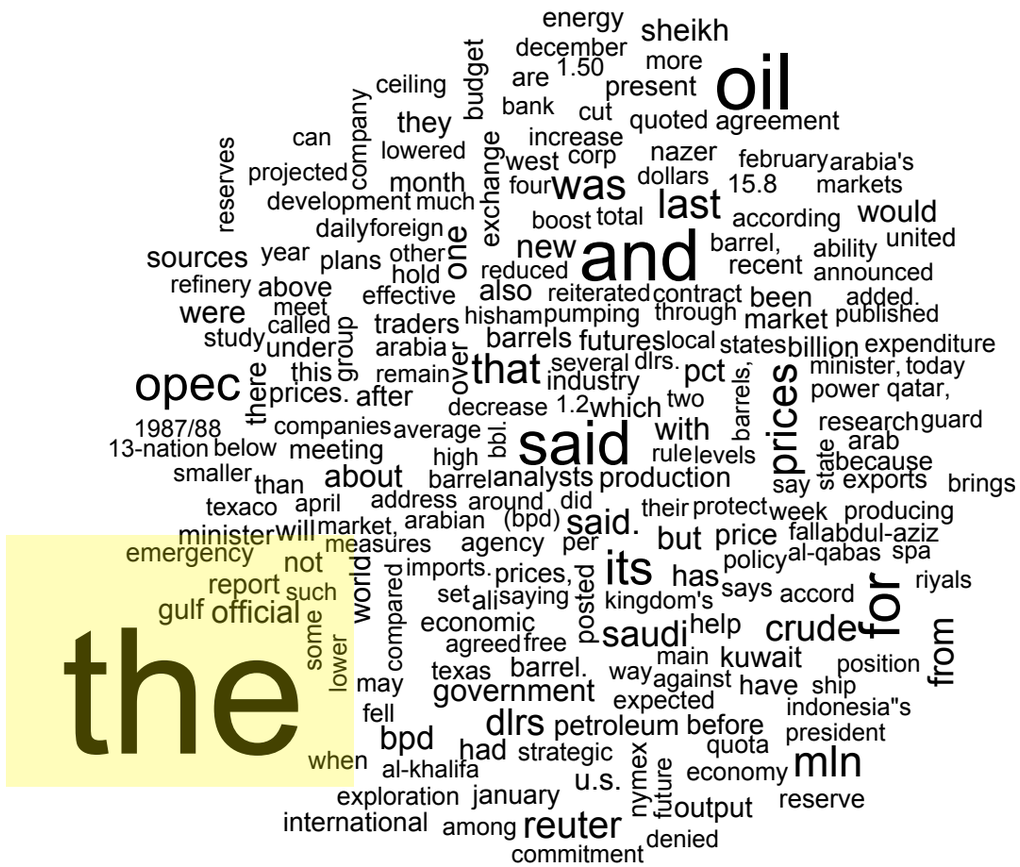
Simplify

```
library(wordcloud)
library(tm)
data(crude)
tdm <- TermDocumentMatrix(crude)
m <- as.matrix(tdm)
v <- sort(rowSums(m),decreasing=TRUE)
 d <- data.frame(word = names(v),freq=v)
wordcloud(d$word,d$freq)
```

without punctuation and stop words



with punctuation and stop words



as.matrix, sort

```
m <- as.matrix(tdm)
v <- sort(rowSums(m),decreasing=TRUE)
d <- data.frame(word = names(v),freq=v)
```

Let us take a simpler file to better understand

```
library(wordcloud)
library(tm)

txt.file = "TEXT.txt"
txt = read.csv(txt.file, header=F)
txt = Corpus(VectorSource(txt))
tdm <- TermDocumentMatrix(txt)
m <- as.matrix(tdm)
v <- sort(rowSums(m),decreasing=TRUE)
d <- data.frame(word = names(v),freq=v)

wordcloud(d$word,d$freq)
```

```
> dim(m)
8 1
```

```
> m  # matrix
       Docs
Terms     V1
  class    1
  full.    1
  going    1
  room?    1
  seminar  1
  the      2
  today    1
  what     1
```

```
> dimnames(m)
$Terms
[1] "class"   "full."   "going"   "room?"   "seminar" "the"     "today"
[8] "what"

$Docs
[1] "V1"
```

```
> colnames(m)
[1] "V1"
> rownames(m)
[1] "class"   "full."   "going"   "room?"
"seminar" "the"     "today"
[8] "what"
```

```
> v
   the  class  full.  going  room? seminar  today   what
     2      1      1      1      1       1      1      1
```

```
> d       # dataframe
          word freq
the        the    2
class    class    1
full.    full.    1
going    going    1
room?    room?    1
seminar seminar   1
today    today    1
what      what    1
```

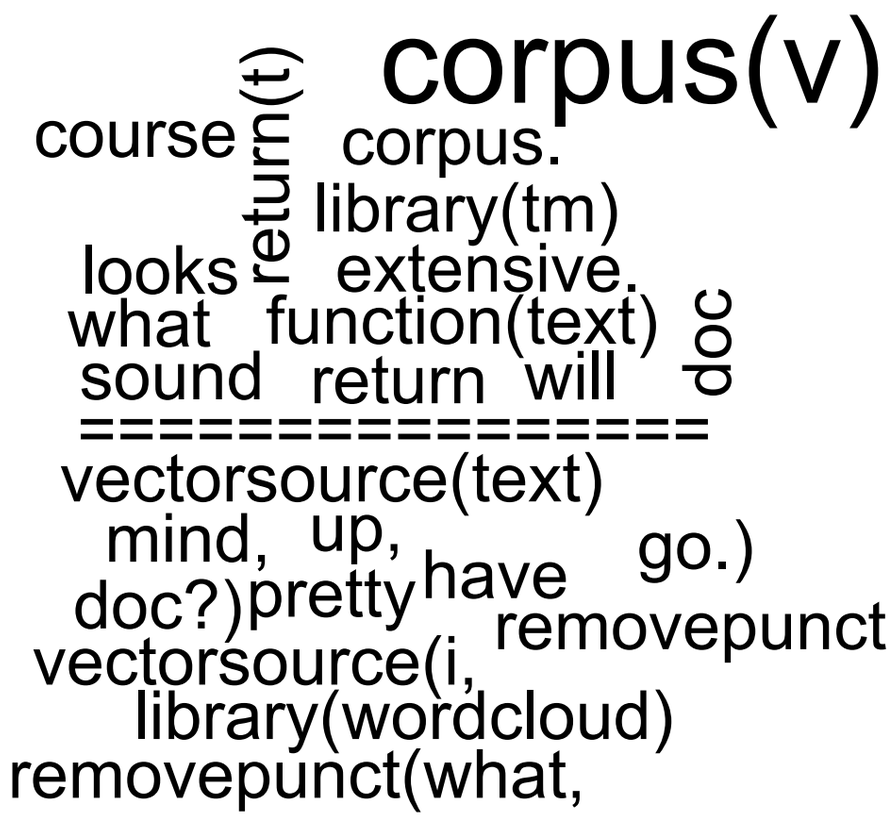
Create a function

- mycloud(filename)
- output a cloud
- hide all the details

```
# Example
library(wordcloud)
library(tm)

mycloud = function(filename)
{
    txt = read.csv(filename, header=F)
    cp = Corpus(VectorSource(txt))
    tdm <- TermDocumentMatrix(cp)
    m <- as.matrix(tdm)
    v <- sort(rowSums(m),decreasing=TRUE)
    d <- data.frame(word = names(v),freq=v)
    wordcloud(d$word,d$freq,min.freq=1)
}
```

```
> source("mycloud.r")
> mycloud("README")
```

warning messages
(next slide)



Warning messages

Warning messages:
1: In wordcloud(d$word, d$freq, min.freq = 1) :
removepunctuation(plaintextdocument(v)) could not be fit on page. It will not be plotted.
2: In wordcloud(d$word, d$freq, min.freq = 1) :
---------- could not be fit on page. It will not be plotted.
3: In wordcloud(d$word, d$freq, min.freq = 1) :
http://cogsci.uni-osnabrueck.de/~severt/sigil/sigil_r/ could not be fit on page. It will not be plotted.