# R+Hadoop 資料分析

華梵大學 圖書資訊處

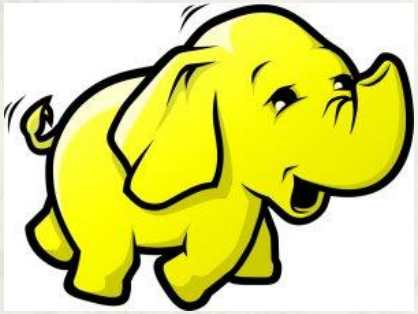
報告人:
李仁鐘、常紘瑜

2015/1/30

# Outline

- Hadoop 介紹
- R 介紹
- R+Hadoop 介紹
- Demo

# Hadoop



# What is Hadoop?

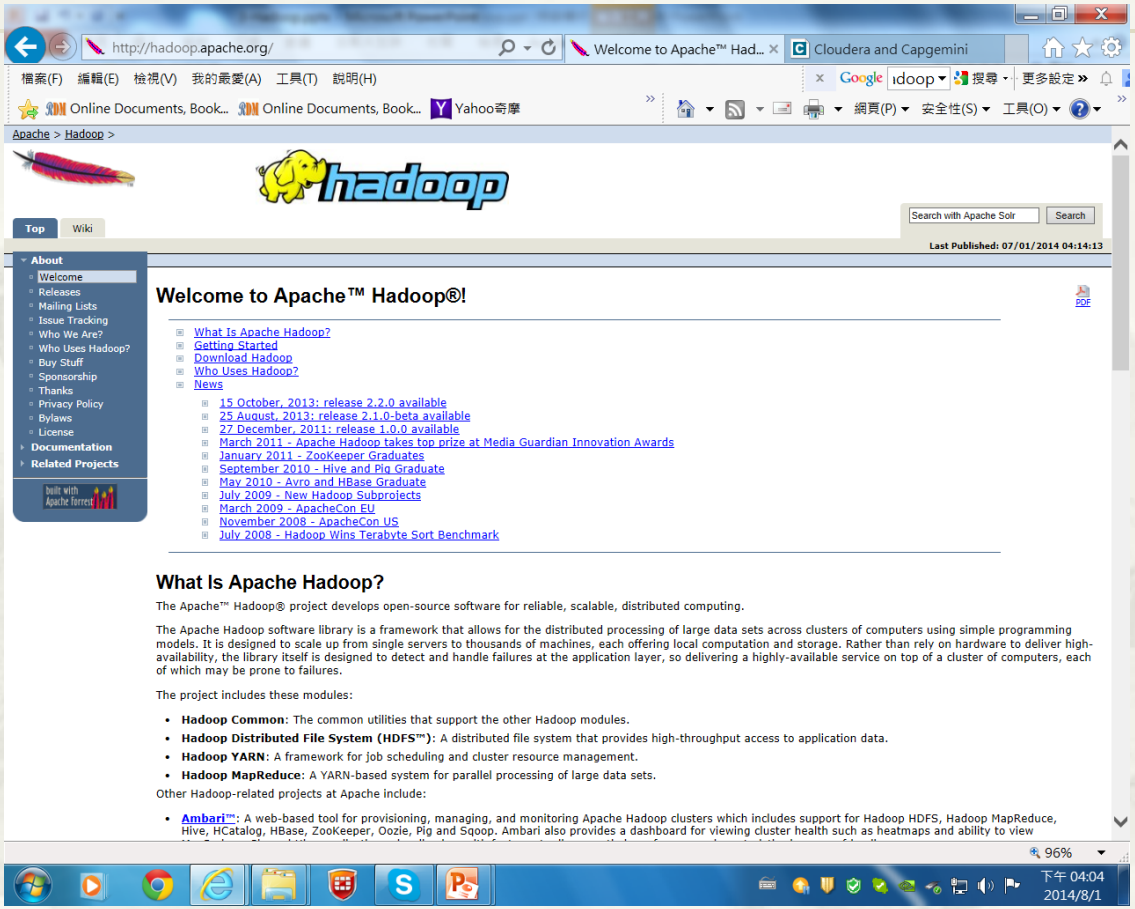
- Hadoop is a software platform that lets one easily write and run applications that process vast amount of data.
  - ✓ 它是軟體平台用來處理程式具巨量資料
- Hadoop can reliably store and process petabytes.
  - ✓ 它可用來可靠地儲存和處理PB級巨量資料
- It distributes the data and processing across clusters of commonly available computers. These clusters can number into the thousands of nodes.
  - ✓ 它將資料和處理程序分散到可以使用的電腦上，而且這些電腦的數量可以達到上千台之多

# What is Hadoop?

- By distributing the data, Hadoop can process it in parallel on the nodes where the data is located. This make it extremely rapid.
  - ✓ 藉由分散資料的處理，Hadoop可以平行的運算這些資料，使得處理速度變得非常快速

- Hadoop automatically maintains multiple copies of data and automatically redeploys computing tasks based on failures.
  - ✓ Hadoop可以將運算的程式和放置的資料在每一個可以運行的節點間進行複製和自動化的備份，可以避免執行中的程式或存放的資料，因為電腦的硬體或系統的上的損壞而使程式無法執行或檔案損毀

# Apache Hadoop




http://hadoop.apache.org/

# Hadoop

- 免費軟體
- 利用MapReduce作為分散式處理技術
- 利用HDFS作為分散式檔案系統

# MapReduce

# MapReduce

- 是一種軟體框架(Software Framework)
- 可在不同電腦組成的叢集(Clusters)上執行
- 能為巨量資料(Big Data)做分散運算處理
- 此框架的功能概念主要是映射(Map)和化簡(Reduce)兩種
- 實作上可用JAVA、R或其他程式語言來達成

# MapReduce

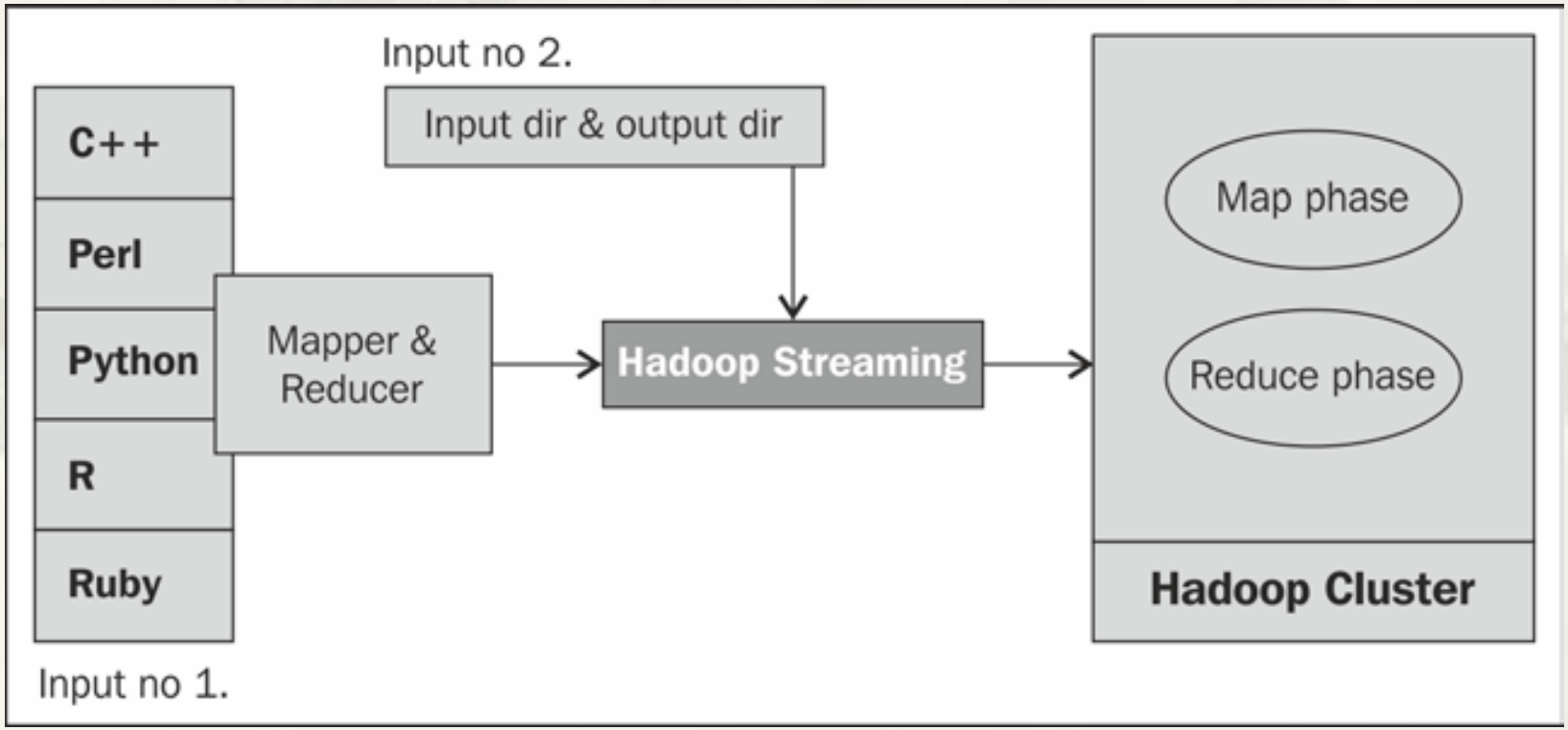
- Map
  - 從主節點(Master Node)輸入一組Input，此Input是一組 key/value序對，將這組輸入切分成好幾個小的子部分，分散到各個工作節點(Slave Nodes)去做運算
  - 輸入是一組 Key/Value 序對 ，輸出則為另一組中間過程(Intermediate)的 key/value 序對
    - $(K_{in}, V_{in})$ ➔ $list(K_{inter}, V_{inter})$

# MapReduce

- Reduce
  - ✓ 負責針對相同的中間過程 key 合併其所有相關聯的 Value，並產生輸出結果的 key/value 序對
  - ✓ 將多對具相同 Key但不同 Value 的資料，結合為多對的 Key/Value
    - ◆ $(K_{inter}, list(V_{inter})) \rightarrow list(K_{out}, V_{out})$

# Streaming API

- 可以讓開發者以其他語言撰寫Mapper/Reducer
  - R, Python, Perl

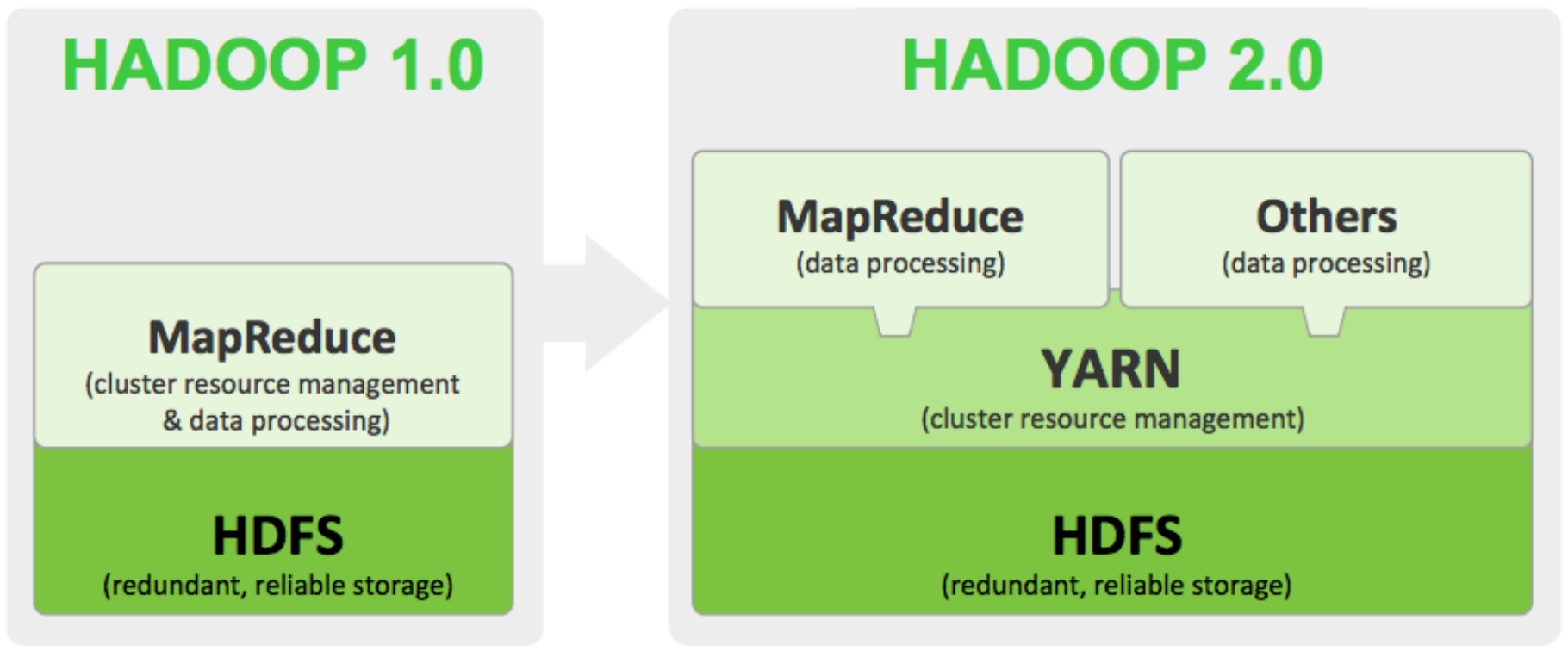


# HDFS

# Hadoop Distributed File System (HDFS)

- 在分散式儲存環境中，提供單一的目錄系統
- 資料以 Write-once-read-many 方式存取
- 每個檔案被分割成許多Block，每個Block複製許多複本(Replica)，並分散儲存於不同的DataNode上
  - ✓ NameNode：負責維護HDFS的檔案名稱空間 (File System Namespace)
  - ✓ DataNode：實際儲存檔案區塊(Blocks)的伺服器

# Hadoop 2.X



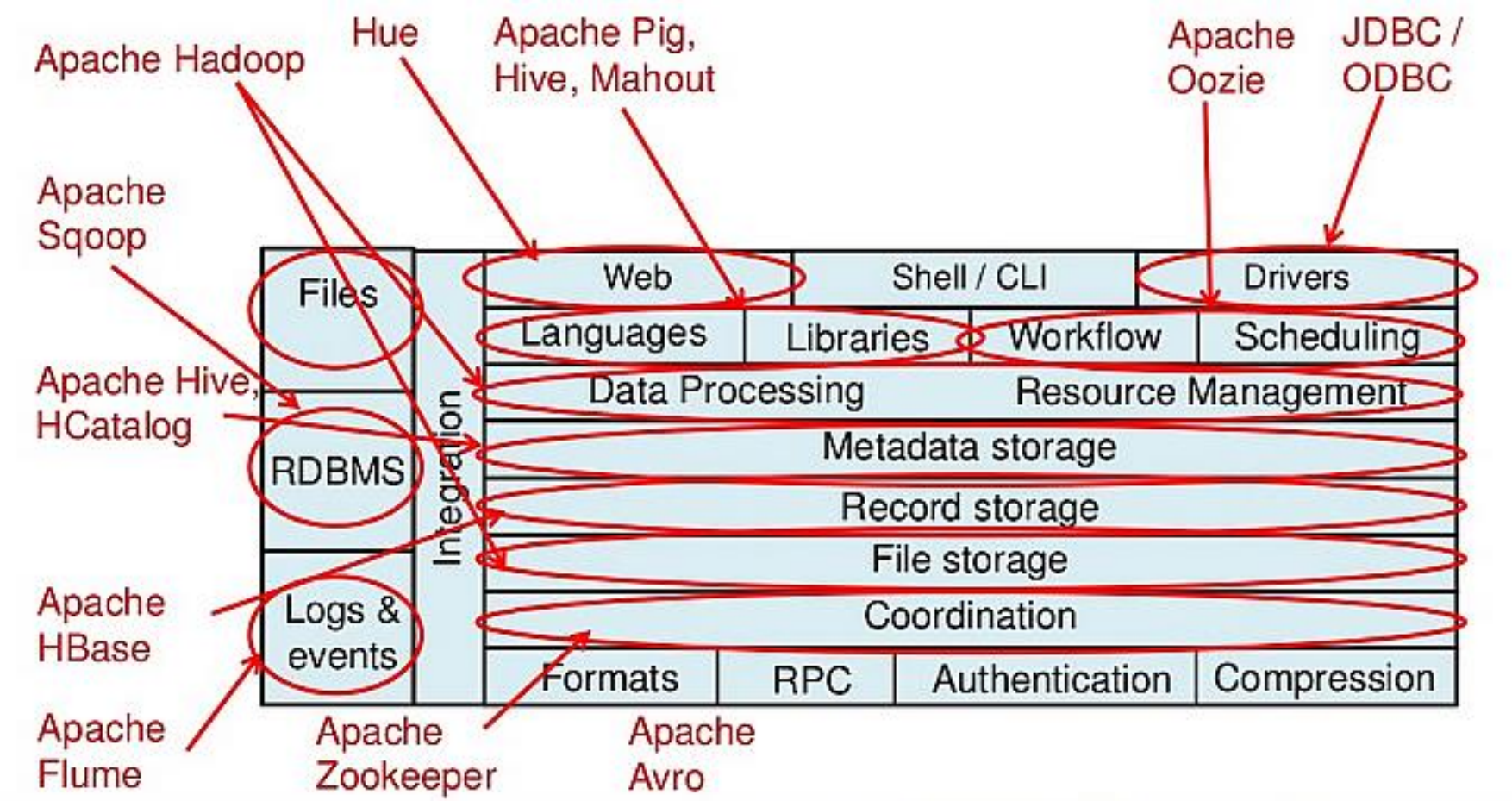

http://hortonworks.com/blog/office-hours-qa-on-yarn-in-hadoop-2/

# YARN

- Yet Another Resource Negotiator

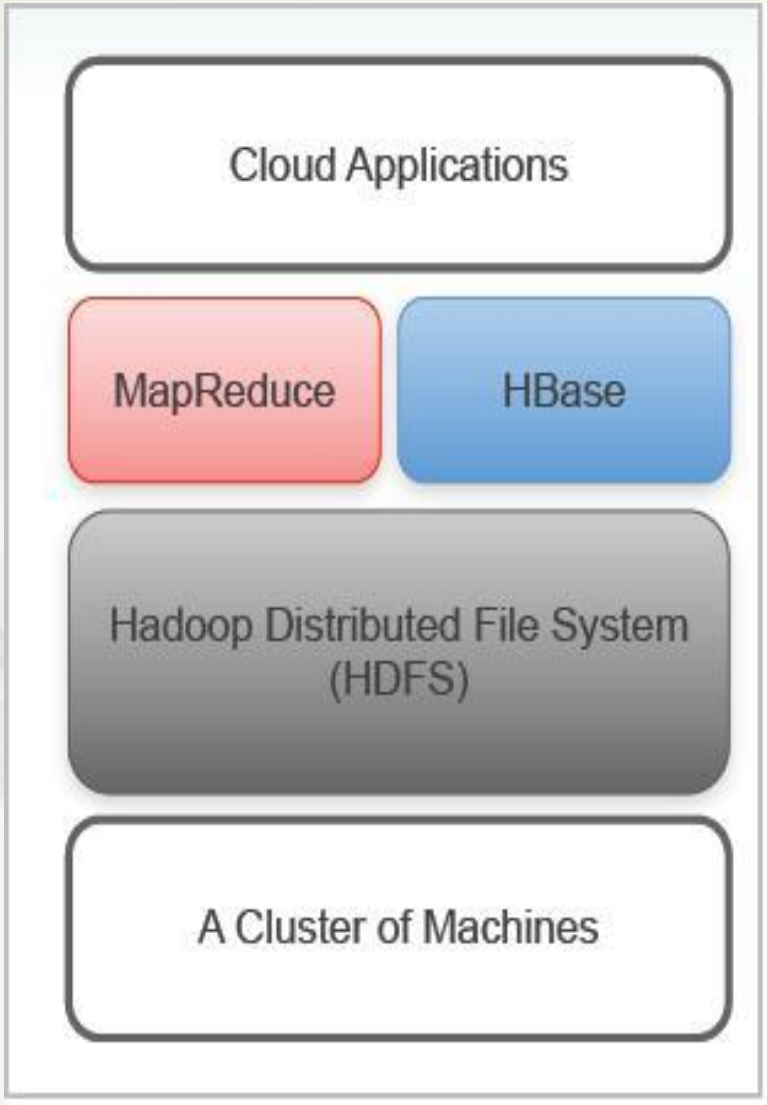
- YARN is a more general purpose framework of which classic MapReduce is one application.
  - ✓ YARN 是更通用的軟體框架，而 MapReduce 只是其中的一個應用

# Hadoop Ecosystem




http://www.slideshare.net/cloudera/the-hadoop-stack-then-now-and-in-the-future-eli-collins-charles-zedlewski-cloudera

# HBase




HBase:

- 是HDFS上的資料庫。
- 沒有正規化與Join的觀念
- 利用Family Columns將相似的欄位群聚在一起，用於強化效率。

# Hive

- Developed at Facebook
- "Relational database" built on Hadoop
  - ✓ Maintains list of table schemas
  - ✓ SQL-like query language (HiveQL)
  - ✓ Can call Hadoop Streaming scripts from HiveQL

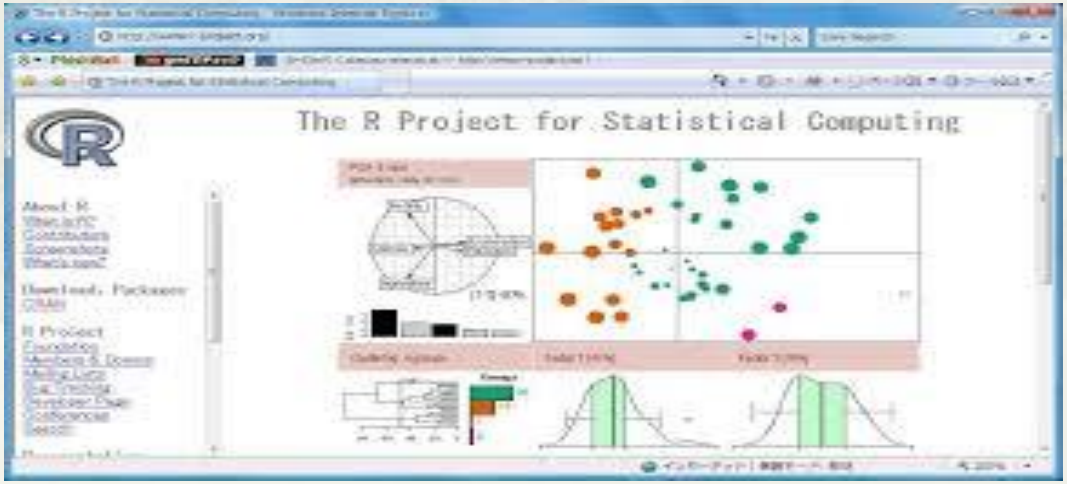
# Sqoop

- 是一個用來將 Hadoop 和關聯式資料庫中的資料相互轉移的工具，可以將一個關聯式資料庫（MySQL, SQL Server等）中的資料導入到 Hadoop 的 HDFS 中，也可以將 HDFS 的資料導入到關聯式資料庫中

# R 語言

# R 簡介

- Ross Ihaka 與 Robert Gentleman（1966）所開發出來之相似於 AT & T 貝爾實驗室所開發之 S 語言
- R 有 Windows、Unix、Linux 及 Apple MacOS 等不同作業系統的版本
- 免費軟體，其網站位於 http://www.r-project.org

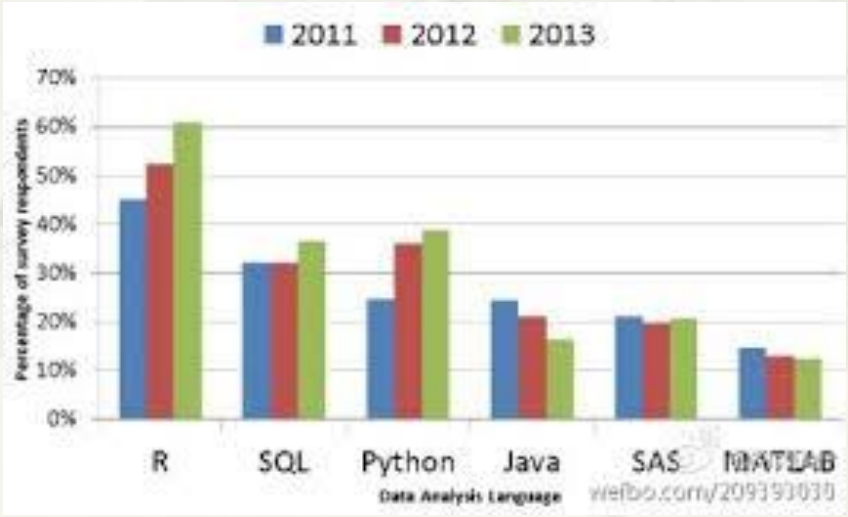


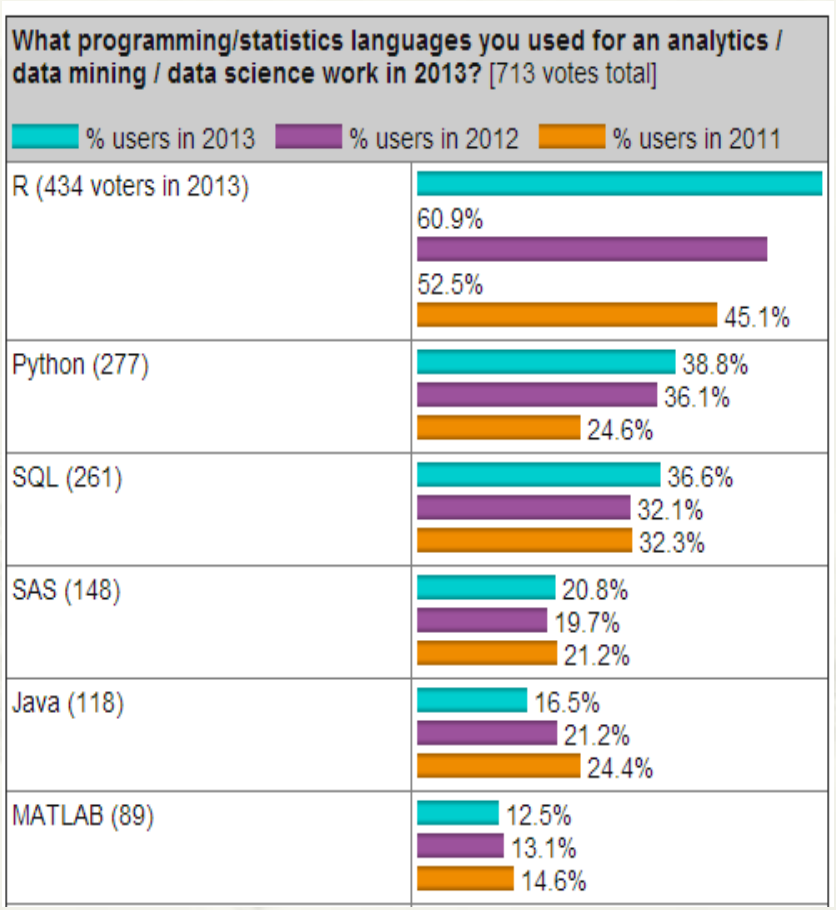
# R 語言

- 內建許多函式(Function)及約5000個免費套件
- R 是直譯式語言 (Interpreted Language)
  - ✓ 一行行執行，可直接看到執行結果
- R 是物件導向語言 (Object Oriented Language)

# 常用的資料分析語言

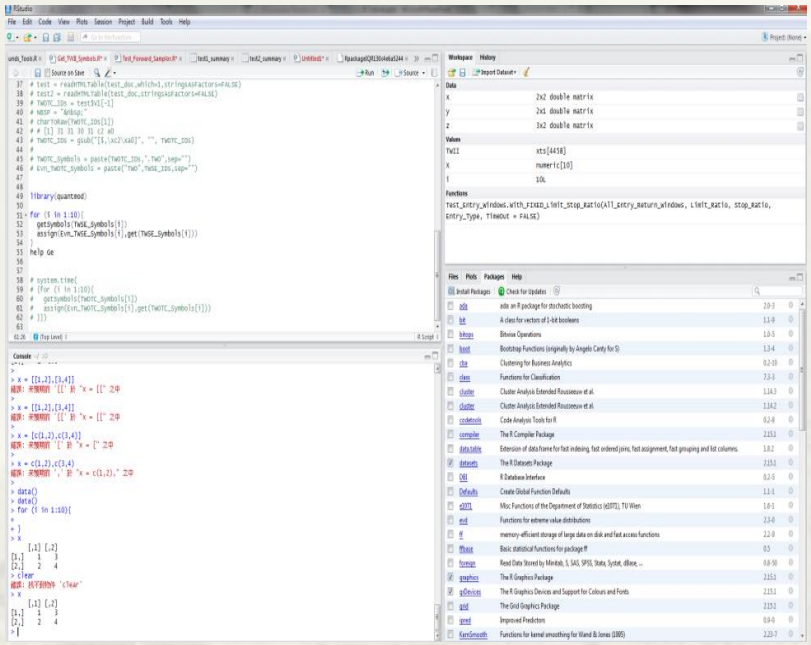
最近最受歡迎的資料分析語言 R



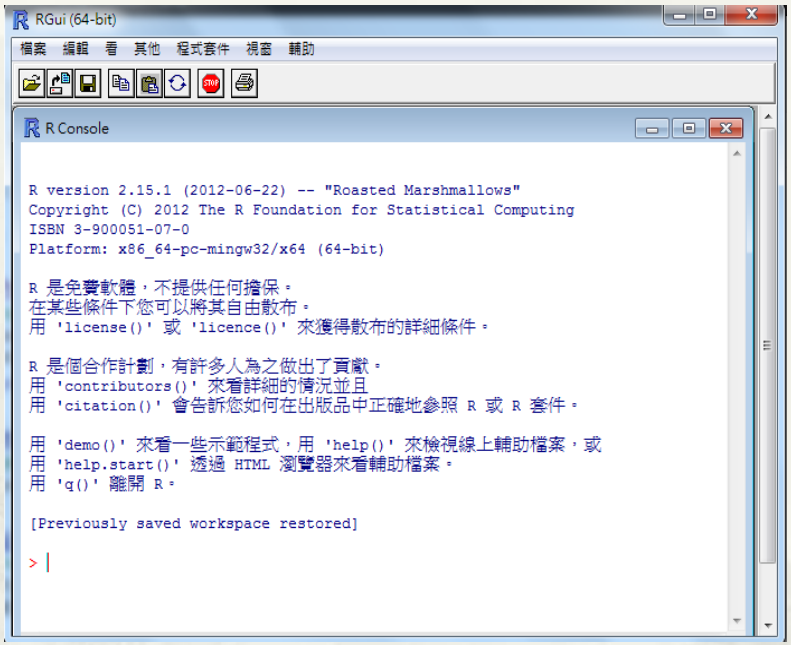

http://www.kdnuggets.com/polls/2013/languages-analytics-data-mining-data-science.html

# 整合式開發環境 IDE



R Studio

http://www.rstudio.com/




R

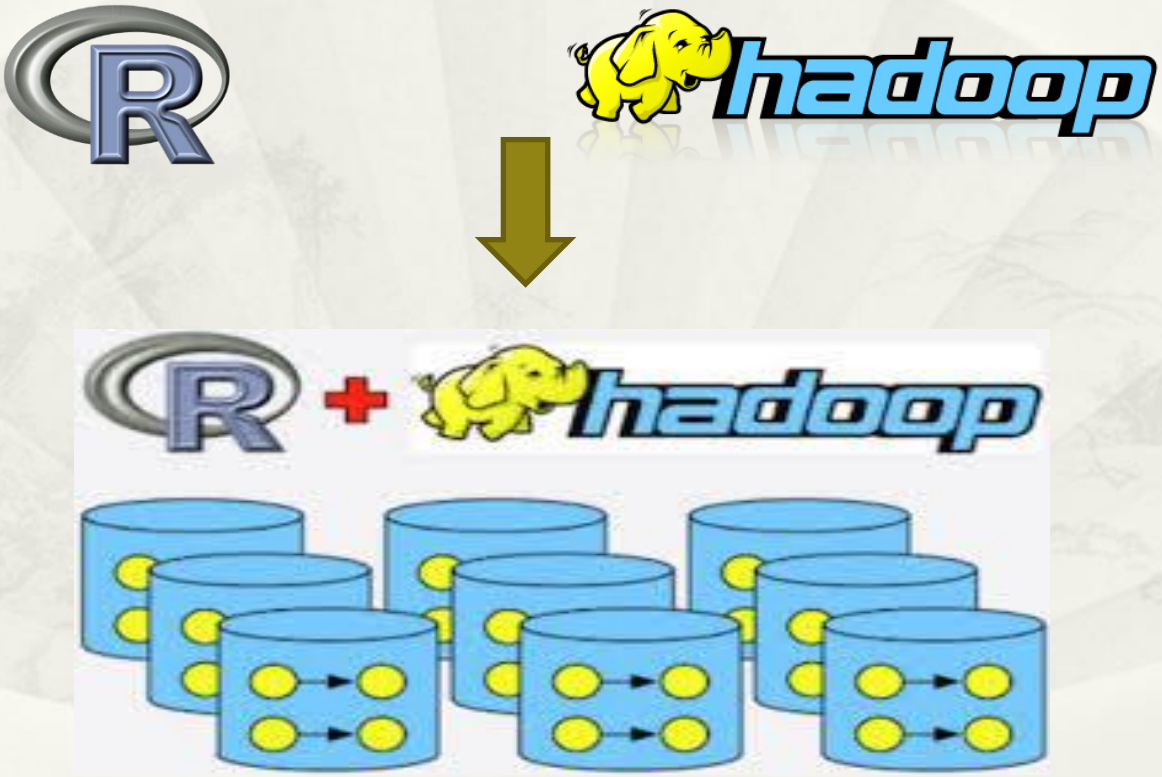
http://www.r-project.org/

# R 應用領域

- Big Data
- 統計分析
- 資料探勘
- 機器學習
- 推薦系統
- 文字探勘

…

# R+Hadoop



# R+Hadoop



- 擴大 R 處理資料能力
  - R 將資料全部讀進 Memory（無法讀入巨量資料）
  - Hadoop 讓 R 可以進行分散式運算
- 使用 R 語言就可輕易使用Hadoop功能

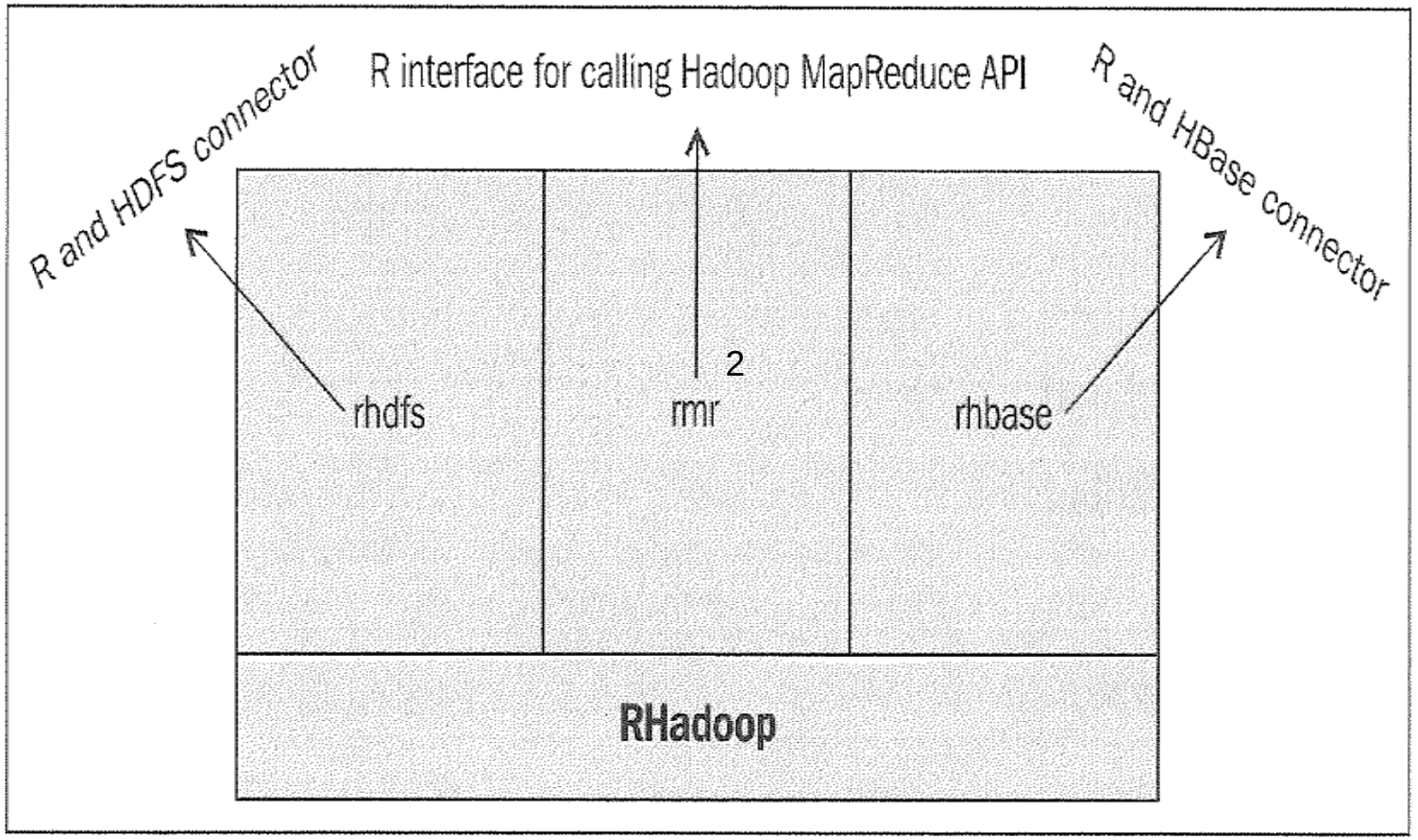
# RHadoop 套件

- 由 Revolution Analytics 開發
- 針對 MapReduce、HDFS 及 HBase發展三個免費套件
  - rmr2
  - rhdfs
  - rhbase

# RHadoop 套件功能

- rhdfs
  - 讓使用者可以透過 R 存取 HDFS
- rmr2
  - 可以讓使用者發展並呼叫 MapReduce 工作
- rhbase
  - 可以操作 HBase 資料

# RHadoop 套件架構




RHadoop Ecosystem

# rmr2 基本指令

```
mapreduce(
    input, output = NULL,
    map = to.map(identity),
    reduce = NULL,
    vectorized.reduce = FALSE,
    combine = NULL,
    in.memory.combine = FALSE,
    input.format = "native",
    output.format = "native",
    backend.parameters = list(),
    verbose = TRUE)
```

# rhbase 基本指令

- **Table Maninpulation**
  hb.new.table, hb.delete.table, hb.describe.table, hb.set.table.mode, hb.regions.table
- **Read/Write**
  hb.insert, hb.get, hb.delete, hb.insert.data.frame, hb.get.data.frame, hb.scan, hb.scan.ex
- **Utility**
  hb.list.tables
- **Initialization**
  hb.defaults, hb.init

https://github.com/RevolutionAnalytics/RHadoop/wiki/user%3Erhbase%3EHome

# rhdfs 基本指令

- 啟用rhdfs
  - hdfs.init ()
- 將檔案從本地端放置於HDFS.
  - hdfs.put('test.txt', './')
- 拷貝檔案
  - hdfs.copy('test.txt', 'test2.txt')
- 將檔案下載到本地端
  - hdfs.get('test.txt', '/home/cloudera/test3.txt')
- 將檔案搬移到不同位置
  - hdfs.move('test.txt', './test/q1.txt')
- 重新命名
  - hdfs.rename('./test/q1.txt', './test/test.txt')
- 刪除資料
  - hdfs.delete('./test/')
  - hdfs.rm('./test/')
- 觀看檔案資訊
  - hdfs.file.info('./')

# RHive 套件

- RHive is an R extension facilitating distributed computing via HiveQL.
  - ✓ RHive 是 R 可延伸使用 HiveQL 來加速分散式運算的套件

- It allows easy usage of HiveQL in R, and allows easy usage of R objects and R functions in Hive.
  - ✓ RHive 可在 R 中容易使用 HiveQL ，亦允許在 Hive 中使用 R 的物件及函式

http://cran.r-project.org/web/packages/RHive/RHive.pdf

# RHive 基本指令

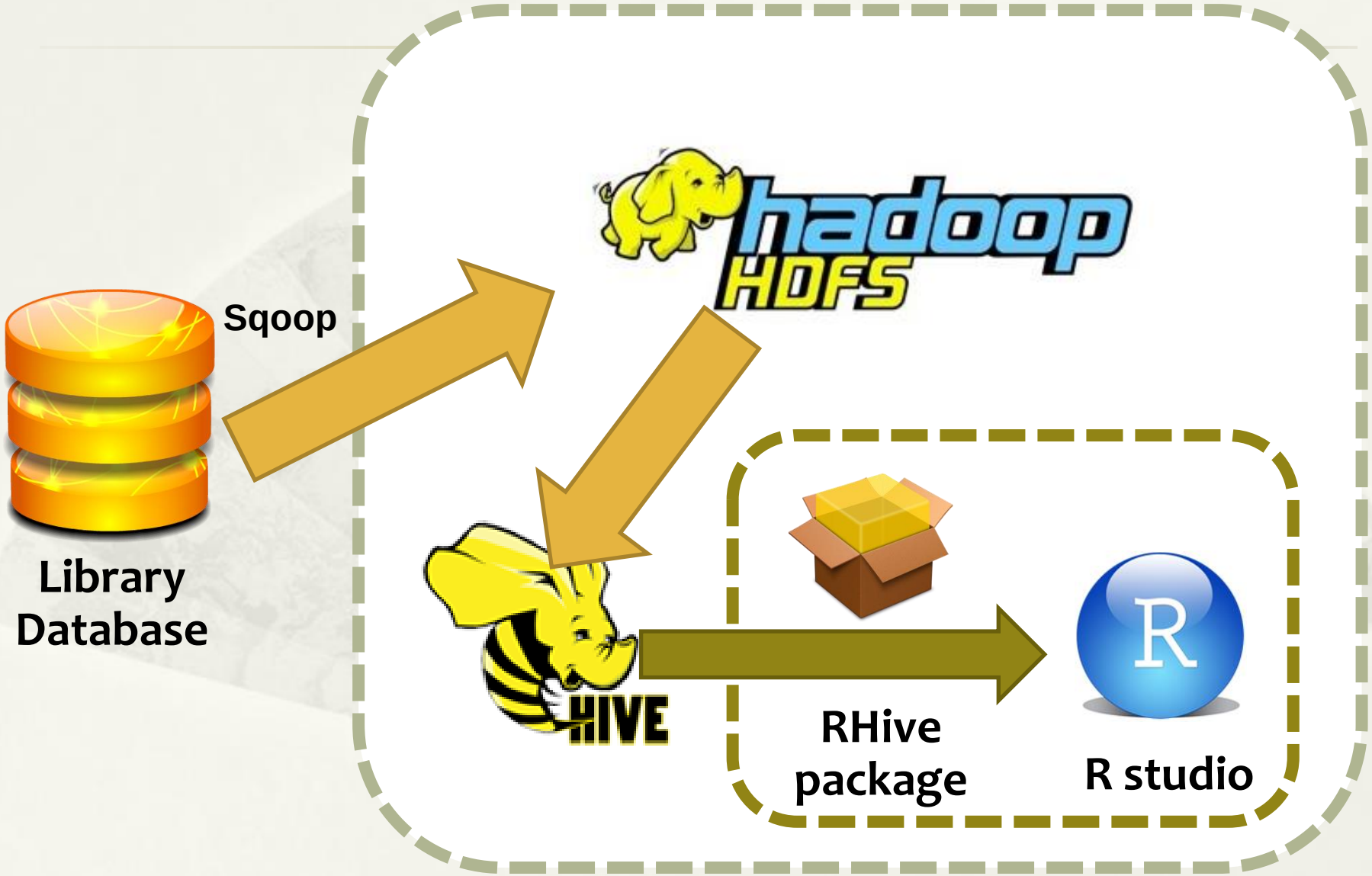
rhive.init
rhive.unset
rhive.big.query
rhive.export
rhive.rm.udf
rhive.close
rhive.use.database
rhive.desc.table
rhive.size.table
rhive.sapply
rhive.save
rhive.mapapply
rhive.hdfs.ls
rhive.hdfs.rm
rhive.hdfs.mkdirs
rhive.hdfs.du
rhive.hdfs.chmod
rhive.basic.mode
rhive.basic.xtabs
rhive.basic.by
rhive.block.sample

rhive.connect
rhive.query
rhive.assign
rhive.exportAll
rhive.script.export
rhive.list.databases
rhive.list.tables
rhive.load.table
rhive.drop.table
rhive.aggregate
rhive.sample
rhive.reduceapply
rhive.hdfs.get
rhive.hdfs.rename
rhive.hdfs.cat
rhive.hdfs.close
rhive.hdfs.chown
rhive.basic.range
rhive.basic.cut
rhive.basic.scale

rhive.set
rhive.execute
rhive.rm
rhive.list.udfs
rhive.script.unexport
rhive.show.databases
rhive.show.tables
rhive.exist.table
rhive.napply
rhive.load
rhive.mrapply
rhive.hdfs.connect
rhive.hdfs.put
rhive.hdfs.exists
rhive.hdfs.tail
rhive.hdfs.info
rhive.hdfs.chgrp
rhive.basic.merge
rhive.basic.cut2
rhive.basic.t.test

# Demo

# 執行流程


Sqoop
Library
Database
hadoop
HDFS
HIVE
RHive
package
R studio

# Sqoop 指令

```
sqoop import --connect
"jdbc:sqlserver://$SQL_SRV:$PORT;database=mitopac;username=$U
;password=$P" --hive-import -m 1 --table reader --warehouse-dir
$DEST_DIR --map-column-hive reader11=String --hive-overwrite
```

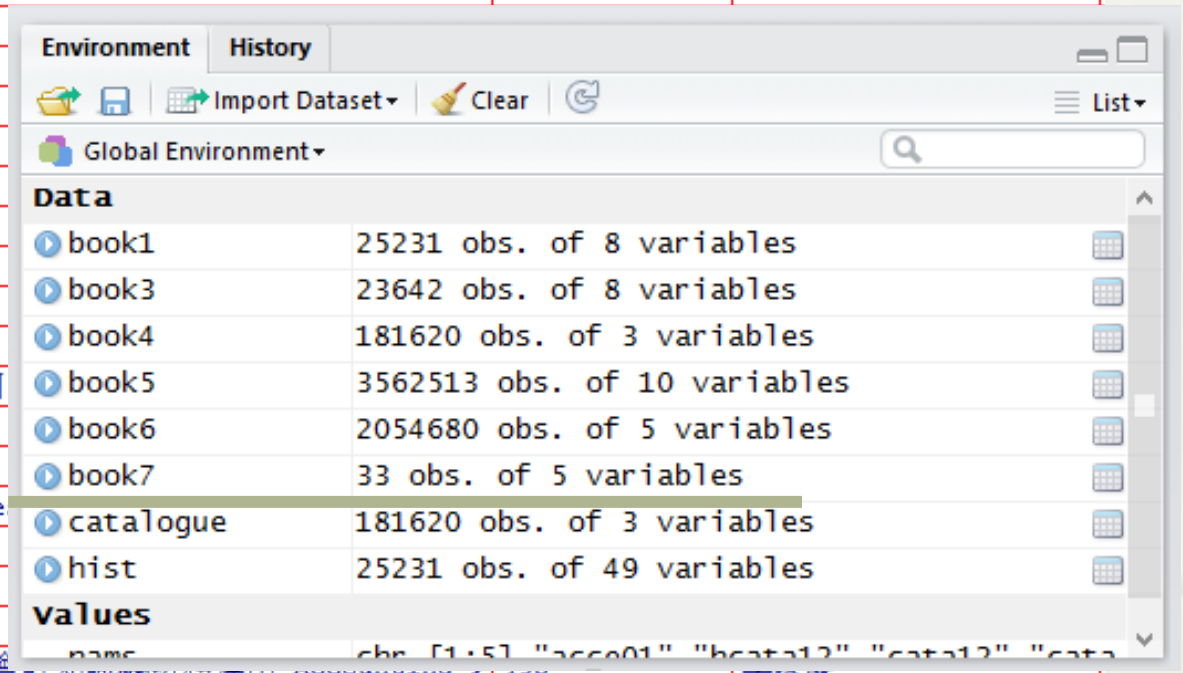
# RHive 指令

```
library('RHive')
library("plyr")
library("dplyr")
library("reshape2")
library("pipeR")

Sys.setenv(HIVE_HOME="/usr/local/hive")
Sys.setenv(HADOOP_HOME="/usr/local/hadoop")

rhive.init()
rhive.connect()

rhive.list.tables()

catalogue_data = rhive.query("select * FROM catalogue LIMIT 10")
str(catalogue_data)
```

# 相關書籍推薦結果

資料編輯器

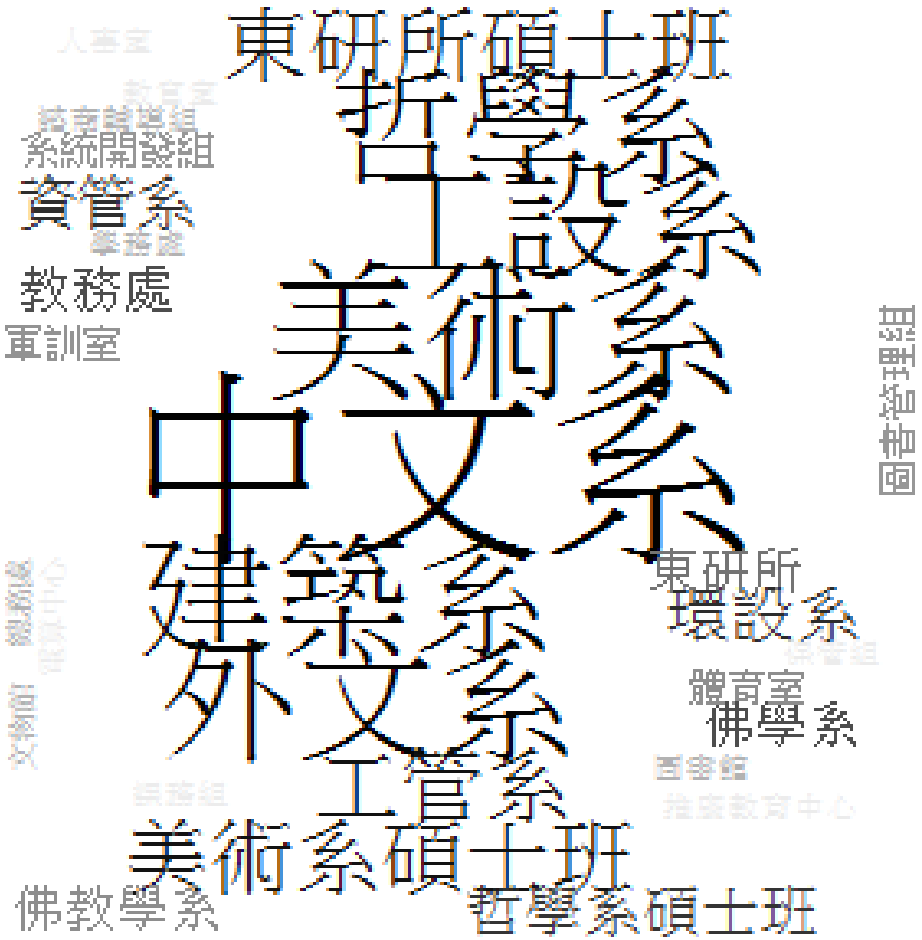
| | | | | | |
|---|---|---|---|---|---|
| 8 | C168163 | 圖解會計學 | 會計學:原理與應用=Accounting eng | 495 | 林朝彬 |
| 9 | C168163 | 圖解會計學 | 超越巔峰會計講義 | | |
| 10 | C168163 | 圖解會計學 | 初級會計學作業解答 | | |
| 11 | C168163 | 圖解會計學 | 會計學 | | |
| 12 | C168163 | 圖解會計學 | 會計學 | | |
| 13 | C168163 | 圖解會計學 | 會計學.(一):總復習 | | |
| 14 | C168163 | 圖解會計學 | 限制理論與產出會計 | | |
| 15 | C168163 | 圖解會計學 | 會計學原理 | | |
| 16 | C168163 | 圖解會計學 | 會計學實習 | | |
| 17 | C168163 | 圖解會計學 | 初級會計 | | |
| 18 | C168163 | 圖解會計學 | 會計系統設計方法指引 | | |
| 19 | C168163 | 圖解會計學 | 中級會計學.上冊 | | |
| 20 | C168163 | 圖解會計學 | 會計大戰 | | |
| 21 | C168163 | 圖解會計學 | 會計學原理=Principle | | |
| 22 | C168163 | 圖解會計學 | 中級會計 | | |
| 23 | C168163 | 圖解會計學 | 中級會計學 | | |
| 24 | C168163 | 圖解會計學 | 會計學 | | |
| 25 | C168163 | 圖解會計學 | 會計:商業人士必備的會 | | |
| 26 | C168163 | 圖解會計學 | 圖解會計學 | 495 | 黃士剛 |
| 27 | C168163 | 圖解會計學 | 會計學精析 | | |
| 28 | C168163 | 圖解會計學 | 會計制度設計之方法 | | |
| 29 | C168163 | 圖解會計學 | 會計學 | | |
| 30 | C168163 | 圖解會計學 | 會計學概要 | | |
| 31 | C168163 | 圖解會計學 | 高等會計學 | | |
| 32 | C168163 | 圖解會計學 | 中小型商業會計制度 | | |




5個欄位
33筆資料

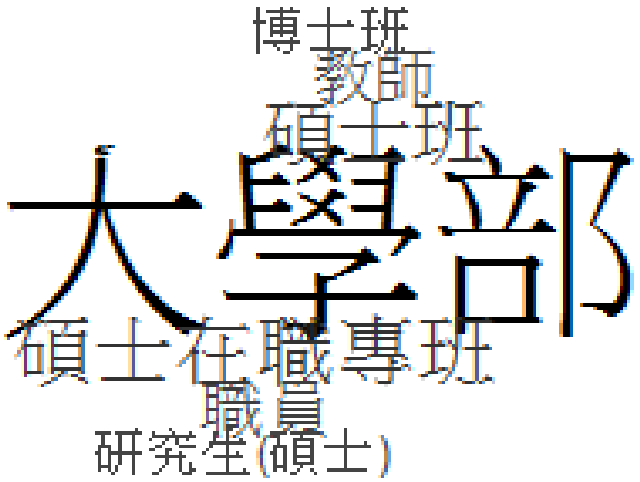
# Wordcloud

```
# 系所及處室借書狀況
require(RODBC)
require(wordcloud)
query <- "select hreader26 as s,COUNT(hreader26) as ss
FROM [mitopac].[dbo].[HIST]
where hacce43!='空間' and  hacce43!='借書證' and  hreader25!='' and
hreader25!='NDDS' and  hreader25!='測試'
group by hreader26
order by ss desc"
myData <- sqlQuery(connection, query)
set.seed(370)
m=as.matrix(myData[,2])
wordFreq=sort(rowSums(m),decreasing=TRUE)
grayLevels=gray((wordFreq+10)/ (max(wordFreq+10)))
words=as.matrix(myData[,1])
wordcloud(words, wordFreq,min.freq=1, random.order=F, colors=grayLevels)
```


東研所碩士班
哲學系
工設系
美術系
中文系
建築系
外文系
工管系
美術系碩士班
佛教學系
哲學系碩士班
資管系
教務處
軍訓室
東研所
環設系
佛學系
體育室
圖書館

# Wordcloud

```
# 各學制借書狀況
query <- "select hreader25 as s,COUNT(hreader25) as ss
FROM [mitopac].[dbo].[HIST]
where hacce43!='空間' and  hacce43!='借書證' and  hreader25!='' and
hreader25!='NDDS' and  hreader25!='測試'
group by hreader25
order by ss desc"
myData1 <- sqlQuery(connection, query)
myData1
set.seed(375)
m1=as.matrix(myData1[,2])
wordFreq1=sort(rowSums(m1),decreasing=TRUE)
grayLevels=gray((wordFreq1+10)/ (max(wordFreq1+10)))
words1=as.matrix(myData1[,1])
wordcloud(words1, wordFreq1,min.freq=1, random.order=F, colors=grayLevels)
```

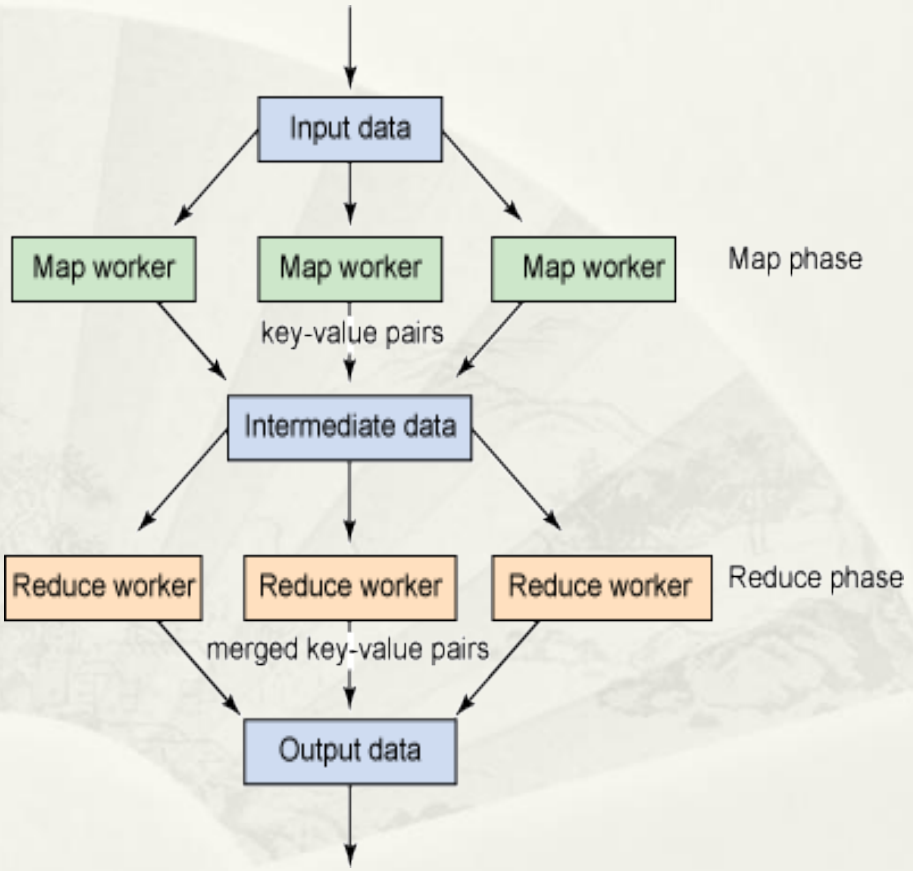

博士班
教師
碩士班
大學部
碩士在職專班
職員
研究生(碩士)

# Demo

# Process log with Hadoop Streaming

# Hadoop Streaming

- 允許使用者利用任何的程式或script進行mapper與reducer
- mapper與reducer從STDIN一行一行讀取資料處理後，送至STDOUT



# Hadoop Streaming

- 例：

```
$ export
    HADOOP_STREAMING=$HADOOP_HOME/share/hadoop/tools/lib/
$ hadoop jar $HADOOP_STREAMING/hadoop-streaming.jar \
-input myInputDirs \
-output myOutputDir \
-mapper cat \
-reducer wc
```

# Hadoop Streaming

- mapper: m.php

```
#!/bin/php
<?php
while (($line = fgets(STDIN)) !== false){
        $str = explode(']', $line);
        $str=preg_replace('/[^A-Za-z0-9]/','',$str["1"]);
        echo $str . " " .  count($str) . "\n";
}
?>
```

# Hadoop Streaming

- reducer: r.php

```
#!/bin/php
<?php
$log=array();
while (($line = fgets(STDIN)) !== false){
        $str = explode(' ', $line);
        if (array_key_exists("$str[0]", $log)){
                $num=$log[$str[0]];
                $num++;
                $log[$str[0]]=$num;
        }else{
                $log[$str[0]]="1";
        }
}
foreach ($log as $key => $value) {
    echo "$key, $value" , "\n";
}
?>
```

# Hadoop Streaming

- 測試m.php

```
$ head -10 www-error_log | php
    m.php
error 1
error 1
error 1
error 1
error 1
includewarn 1
error 1
error 1
error 1
error 1
```

➔(Intermediate Data)

- 測試r.php

```
$ head -10 www-error_log | php
    m.php | php r.php

error, 9
includewarn, 1
```

# Hadoop Streaming

- 上傳log至HDFS

```
$ hadoop fs -put www-error_log.12 /user/hadoop/error_log
```

- 檢視檔案

```
$ hadoop fs -ls /user/hadoop/error_log
-rw-r--r--   2 hadoop supergroup 1073071286 2015-01-07 11:30
    /user/hadoop/error_log/www-error_log.12
```

# Hadoop Streaming

- 執行

```
$ hadoop jar \ /usr/local/hadoop/share/hadoop/tools/lib/hadoop-
    streaming-2.5.1.jar \
  -input /user/hadoop/error_log/www-error_log.12 \
    -output www-error_log.out \
    -mapper "/home/hadoop/m.php" \
    -reducer "/home/hadoop/r.php" \
    -file /home/hadoop/m.php \
    -file /home/hadoop/r.php
```

# Hadoop Streaming

- 結果

$ hadoop fs -cat /user/hadoop/www-error_log.out/part-00000

accesscompaterror, 190
authzcoreerror, 1821
autoindexerror, 2226
cgiderror, 2529
corecrit, 139
coreerror, 12726
corenotice, 455
error, 4638941
includewarn, 288858

# References

1. Apache Hadoop Tutorial: http://hadoop.apache.org
http://hadoop.apache.org/core/docs/current/mapred_tutorial.html
2. Dean, J. and Ghemawat, S. 2008. **MapReduce: simplified data processing on large clusters.** *Communication of ACM* 51, 1 (Jan. 2008), 107-113.
3. Cloudera Videos by Aaron Kimball:
http://www.cloudera.com/hadoop-training-basic
4. http://www.cse.buffalo.edu/faculty/bina/mapreduce.html
5. D. Cutting and E. Baldeschwieler, "Meet Hadoop," OSCON, Portland, OR, USA, 25 July 2007 (Yahoo!)
6. R. E. Brayant, "Data Intensive Scalable computing: The case for DISC," Tech Report: CMU-CS-07-128, http://www.cs.cmu.edu/~bryant
7. **Vignesh Prajapati, Big Data Analytics with R and Hadoop, November 2013**

# Thank for your attention

# Q & A