Introduction to Data Mining with R[1]

Yanchang Zhao
http://www.RDataMining.com

Statistical Modelling and Computing Workshop at Geoscience Australia

8 May 2015

[1] Presented at AusDM 2014 (QUT, Brisbane) in Nov 2014, at Twitter (US) in Oct 2014, at UJAT (Mexico) in Sept 2014, and at University of Canberra in Sept 2013

Questions

- Do you know data mining and its algorithms and techniques?

Questions

- Do you know data mining and its algorithms and techniques?
- Have you heard of R?

Questions

- Do you know data mining and its algorithms and techniques?
- Have you heard of R?
- Have you ever used R in your work?

Outline

What is R?

- R [2] is a free software environment for statistical computing and graphics.
- R can be easily extended with 6,600+ packages available on CRAN[3] (as of May 2015).
- Many other packages provided on Bioconductor[4], R-Forge[5], GitHub[6], etc.
- R manuals on CRAN[7]
  - *An Introduction to R*
  - *The R Language Definition*
  - *R Data Import/Export*
  - ...

[2] http://www.r-project.org/
[3] http://cran.r-project.org/
[4] http://www.bioconductor.org/
[5] http://r-forge.r-project.org/
[6] https://github.com/
[7] http://cran.r-project.org/manuals.html

Why R?

- R is widely used in both academia and **industry**.
- R was ranked no. 1 in the KDnuggets 2014 poll on *Top Languages for analytics, data mining, data science*[8] (actually, no. 1 in 2011, 2012 & 2013!).
- The CRAN Task Views [9] provide collections of packages for different tasks.
  - Machine learning & statistical learning
  - Cluster analysis & finite mixture models
  - Time series analysis
  - Multivariate statistics
  - Analysis of spatial data
  - ...

[8] http://www.kdnuggets.com/polls/2014/languages-analytics-data-mining-data-science.html
[9] http://cran.r-project.org/web/views/

Outline

Classification with R

- Decision trees: *rpart*, *party*
- Random forest: *randomForest*, *party*
- SVM: *e1071*, *kernlab*
- Neural networks: *nnet*, *neuralnet*, *RSNNS*
- Performance evaluation: *ROCR*

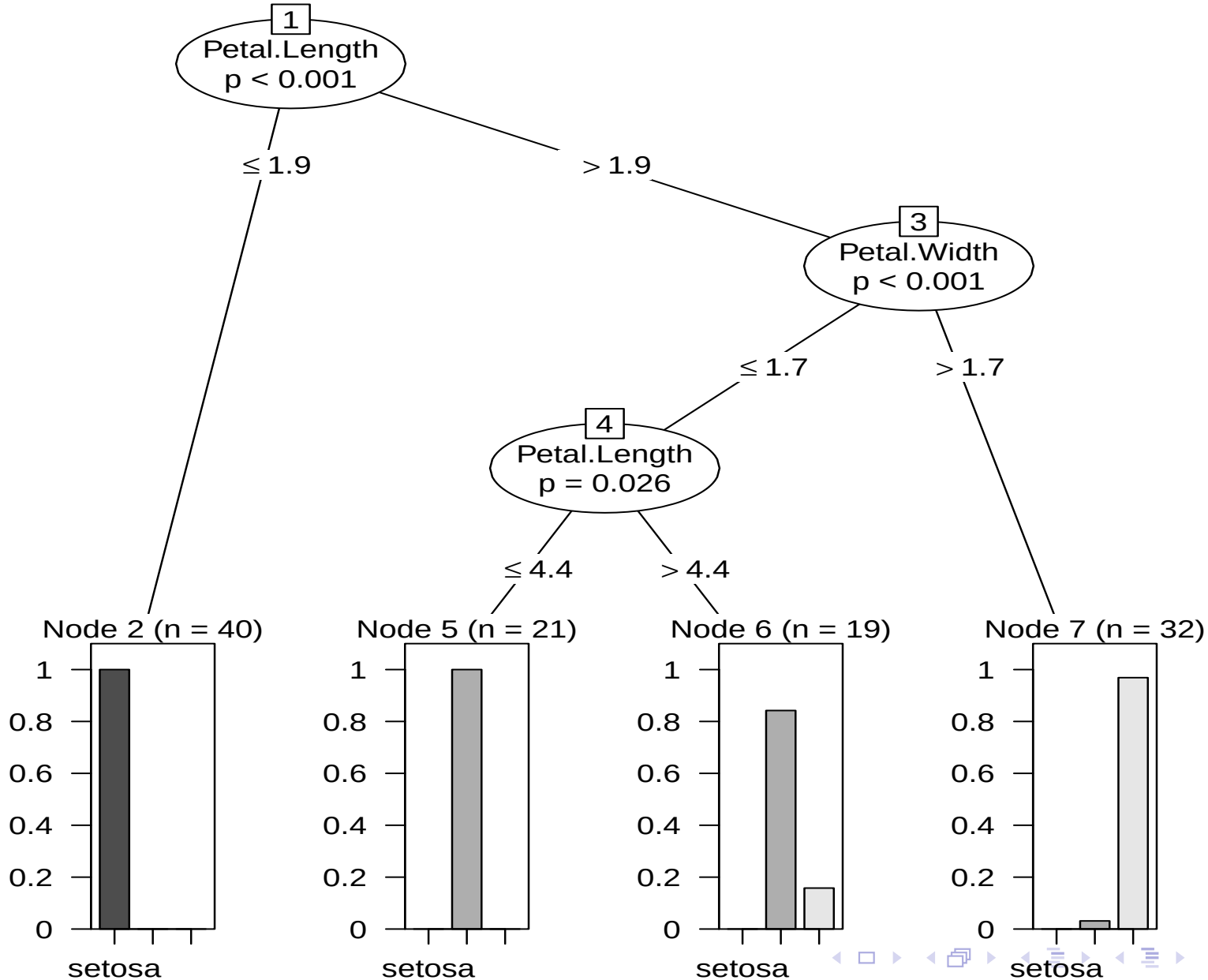
The Iris Dataset

```
# iris data
str(iris)

## 'data.frame': 150 obs. of  5 variables:
##  $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
##  $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1..
##  $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1..
##  $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0..
##  $ Species     : Factor w/ 3 levels "setosa","versicolor",...

# split into training and test datasets
set.seed(1234)
ind <- sample(2, nrow(iris), replace=T, prob=c(0.7, 0.3))
iris.train <- iris[ind==1, ]
iris.test <- iris[ind==2, ]
```

Build a Decision Tree

```
# build a decision tree
library(party)
iris.formula <- Species ~ Sepal.Length + Sepal.Width +
                          Petal.Length + Petal.Width
iris.ctree <- ctree(iris.formula, data=iris.train)
```

```
plot(iris.ctree)
```



Prediction

```
# predict on test data
pred <- predict(iris.ctree, newdata = iris.test)
# check prediction result
table(pred, iris.test$Species)

##
## pred          setosa versicolor virginica
##   setosa          10          0         0
##   versicolor       0         12         2
##   virginica        0          0        14
```

Outline

Clustering with R

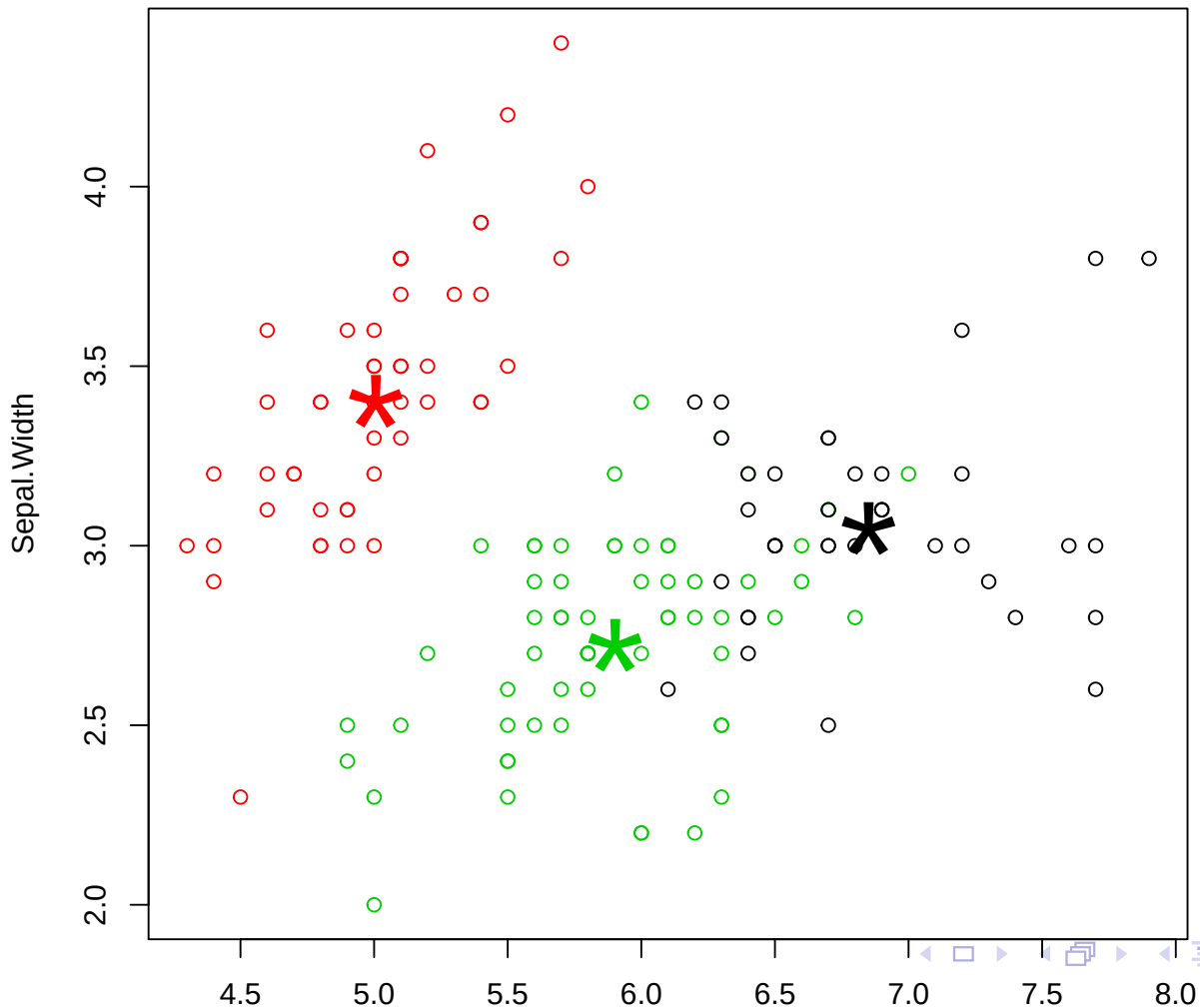
- *k*-means: *kmeans()*, *kmeansruns()*[10]
- *k*-medoids: *pam()*, *pamk()*
- Hierarchical clustering: *hclust()*, *agnes()*, *diana()*
- DBSCAN: *fpc*
- BIRCH: *birch*
- Cluster validation: packages *clv*, *clValid*, *NbClust*

[10] Functions are followed with "()", and others are packages.

k-means Clustering

```
set.seed(8953)
iris2 <- iris
# remove class IDs
iris2$Species <- NULL
# k-means clustering
iris.kmeans <- kmeans(iris2, 3)
# check result
table(iris$Species, iris.kmeans$cluster)

##
##                1  2  3
##   setosa       0 50  0
##   versicolor   2  0 48
##   virginica   36  0 14
```

```r
# plot clusters and their centers
plot(iris2[c("Sepal.Length", "Sepal.Width")], col=iris.kmeans$cluster)
points(iris.kmeans$centers[, c("Sepal.Length", "Sepal.Width")],
       col=1:3, pch="*", cex=5)
```

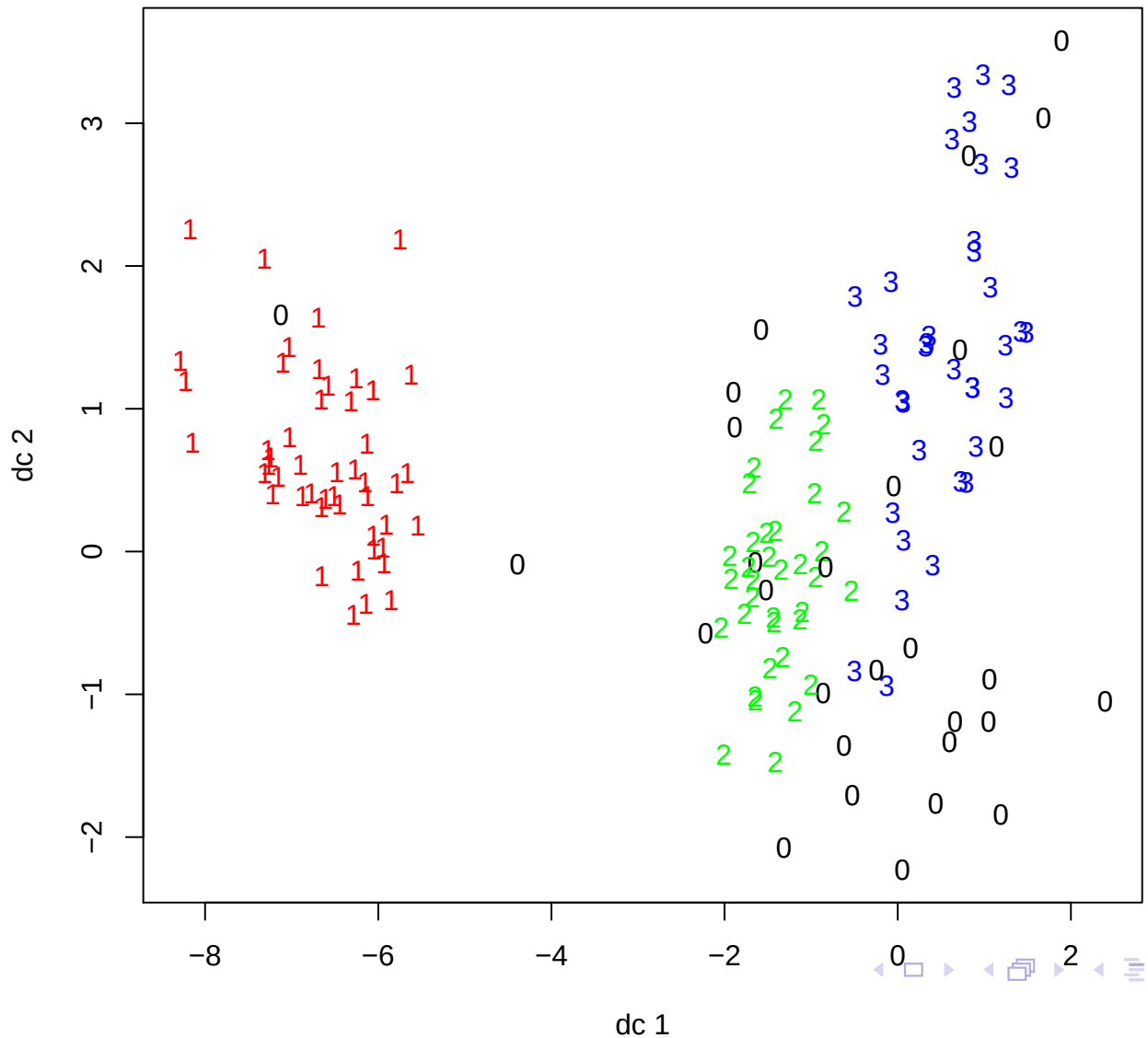


Density-based Clustering

```
library(fpc)
iris2 <- iris[-5]  # remove class IDs
# DBSCAN clustering
ds <- dbscan(iris2, eps = 0.42, MinPts = 5)
# compare clusters with original class IDs
table(ds$cluster, iris$Species)
```

```
##
##     setosa versicolor virginica
##   0      2         10        17
##   1     48          0         0
##   2      0         37         0
##   3      0          3        33
```

```
# 1-3: clusters; 0: outliers or noise
plotcluster(iris2, ds$cluster)
```



Outline

Association Rule Mining with R

- Association rules: *apriori()*, *eclat()* in package *arules*
- Sequential patterns: *arulesSequence*
- Visualisation of associations: *arulesViz*

The Titanic Dataset

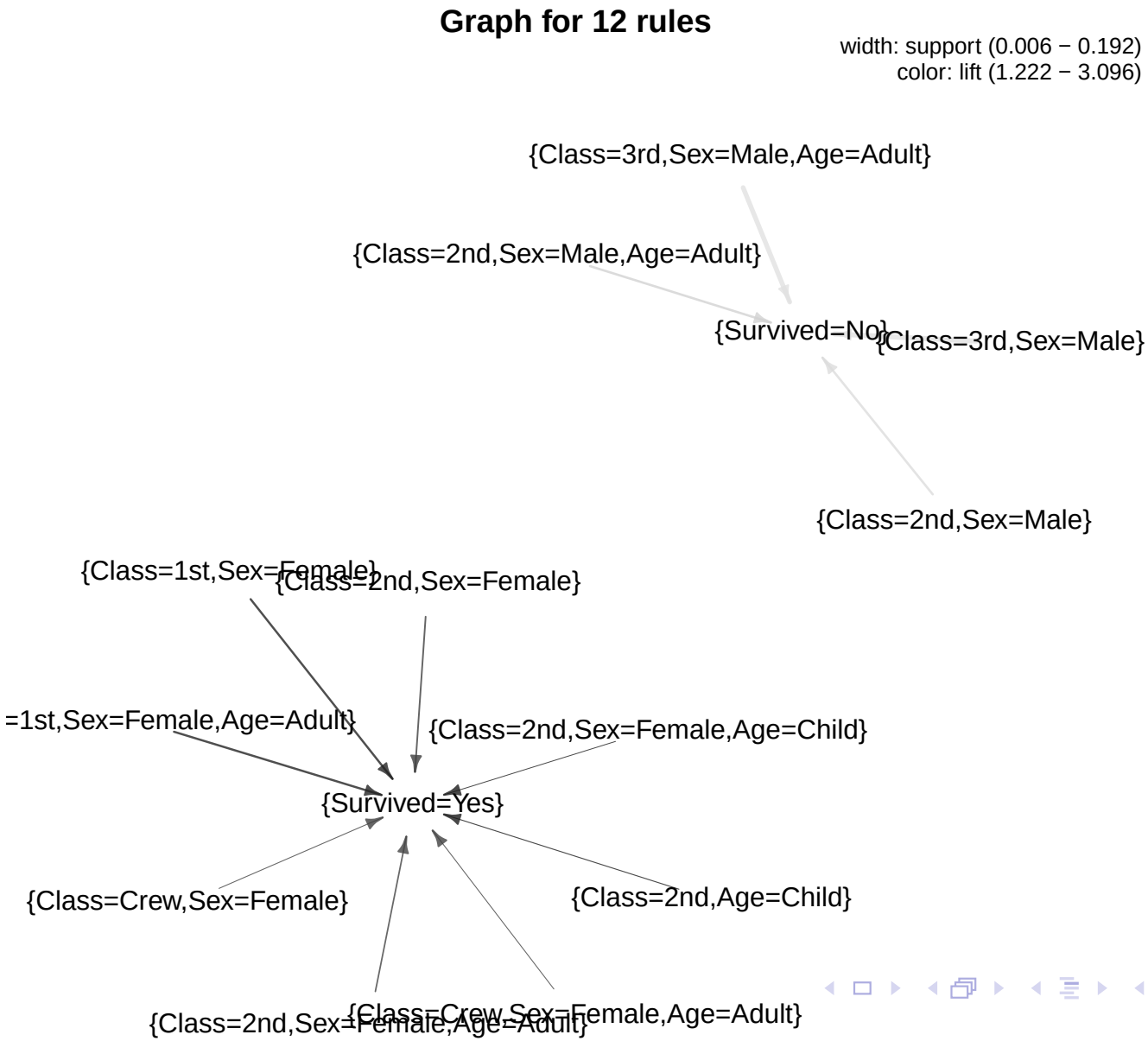
```
load("./data/titanic.raw.rdata")
dim(titanic.raw)

## [1] 2201    4

idx <- sample(1:nrow(titanic.raw), 8)
titanic.raw[idx, ]

##       Class    Sex   Age Survived
## 501     3rd   Male Adult       No
## 477     3rd   Male Adult       No
## 674     3rd   Male Adult       No
## 766    Crew   Male Adult       No
## 1485    3rd Female Adult       No
## 1388    2nd Female Adult       No
## 448     3rd   Male Adult       No
## 590     3rd   Male Adult       No
```

Association Rule Mining

```
# find association rules with the APRIORI algorithm
library(arules)
rules <- apriori(titanic.raw, control=list(verbose=F),
                 parameter=list(minlen=2, supp=0.005, conf=0.8),
                 appearance=list(rhs=c("Survived=No", "Survived=Yes"),
                                 default="lhs"))
# sort rules
quality(rules) <- round(quality(rules), digits=3)
rules.sorted <- sort(rules, by="lift")
# have a look at rules
# inspect(rules.sorted)
```

```
#     lhs                  rhs                support confidence  lift
# 1   {Class=2nd,
#      Age=Child}       => {Survived=Yes}    0.011      1.000 3.096
# 2   {Class=2nd,
#      Sex=Female,
#      Age=Child}       => {Survived=Yes}    0.006      1.000 3.096
# 3   {Class=1st,
#      Sex=Female}      => {Survived=Yes}    0.064      0.972 3.010
# 4   {Class=1st,
#      Sex=Female,
#      Age=Adult}       => {Survived=Yes}    0.064      0.972 3.010
# 5   {Class=2nd,
#      Sex=Male,
#      Age=Adult}       => {Survived=No}     0.070      0.917 1.354
# 6   {Class=2nd,
#      Sex=Female}      => {Survived=Yes}    0.042      0.877 2.716
# 7   {Class=Crew,
#      Sex=Female}      => {Survived=Yes}    0.009      0.870 2.692
# 8   {Class=Crew,
#      Sex=Female,
#      Age=Adult}       => {Survived=Yes}    0.009      0.870 2.692
# 9   {Class=2nd,
#      Sex=Male}        => {Survived=No}     0.070      0.860 1.271
# 10  {Class=2nd,
```

```
library(arulesViz)
plot(rules, method = "graph")
```



Outline

Text Mining with R

- Text mining: *tm*
- Topic modelling: *topicmodels*, *lda*
- Word cloud: *wordcloud*
- Twitter data access: *twitteR*

Retrieve Tweets

Retrieve recent tweets by @RDataMining

```
## Option 1: retrieve tweets from Twitter
library(twitteR)
tweets <- userTimeline("RDataMining", n = 3200)
## Option 2: download @RDataMining tweets from RDataMining.com
url <- "http://www.rdatamining.com/data/rdmTweets.RData"
download.file(url, destfile = "./data/rdmTweets.RData")
```

```
## load tweets into R
load(file = "./data/rdmTweets.RData")
(n.tweet <- length(tweets))

## [1] 320

strwrap(tweets[[320]]$text, width = 55)

## [1] "An R Reference Card for Data Mining is now available"
## [2] "on CRAN. It lists many useful R functions and packages"
## [3] "for data mining applications."
```

Text Cleaning

```
library(tm)
# convert tweets to a data frame
df <- twListToDF(tweets)
# build a corpus
myCorpus <- Corpus(VectorSource(df$text))
# convert to lower case
myCorpus <- tm_map(myCorpus, tolower)
# remove punctuations and numbers
myCorpus <- tm_map(myCorpus, removePunctuation)
myCorpus <- tm_map(myCorpus, removeNumbers)
# remove URLs, 'http' followed by non-space characters
removeURL <- function(x) gsub("http[^[:space:]]*", "", x)
myCorpus <- tm_map(myCorpus, removeURL)
# remove 'r' and 'big' from stopwords
myStopwords <- setdiff(stopwords("english"), c("r", "big"))
# remove stopwords
myCorpus <- tm_map(myCorpus, removeWords, myStopwords)
```

Stemming

```
# keep a copy of corpus
myCorpusCopy <- myCorpus
# stem words
myCorpus <- tm_map(myCorpus, stemDocument)
# stem completion
myCorpus <- tm_map(myCorpus, stemCompletion,
                   dictionary = myCorpusCopy)
# replace "miners" with "mining", because "mining" was
# first stemmed to "mine" and then completed to "miners"
myCorpus <- tm_map(myCorpus, gsub, pattern="miners",
                   replacement="mining")
strwrap(myCorpus[320], width=55)

## [1] "r reference card data mining now available cran list"
## [2] "used r functions package data mining applications"
```

Frequent Terms

```
myTdm <- TermDocumentMatrix(myCorpus,
                            control=list(wordLengths=c(1,Inf)))
# inspect frequent words
(freq.terms <- findFreqTerms(myTdm, lowfreq=20))
```

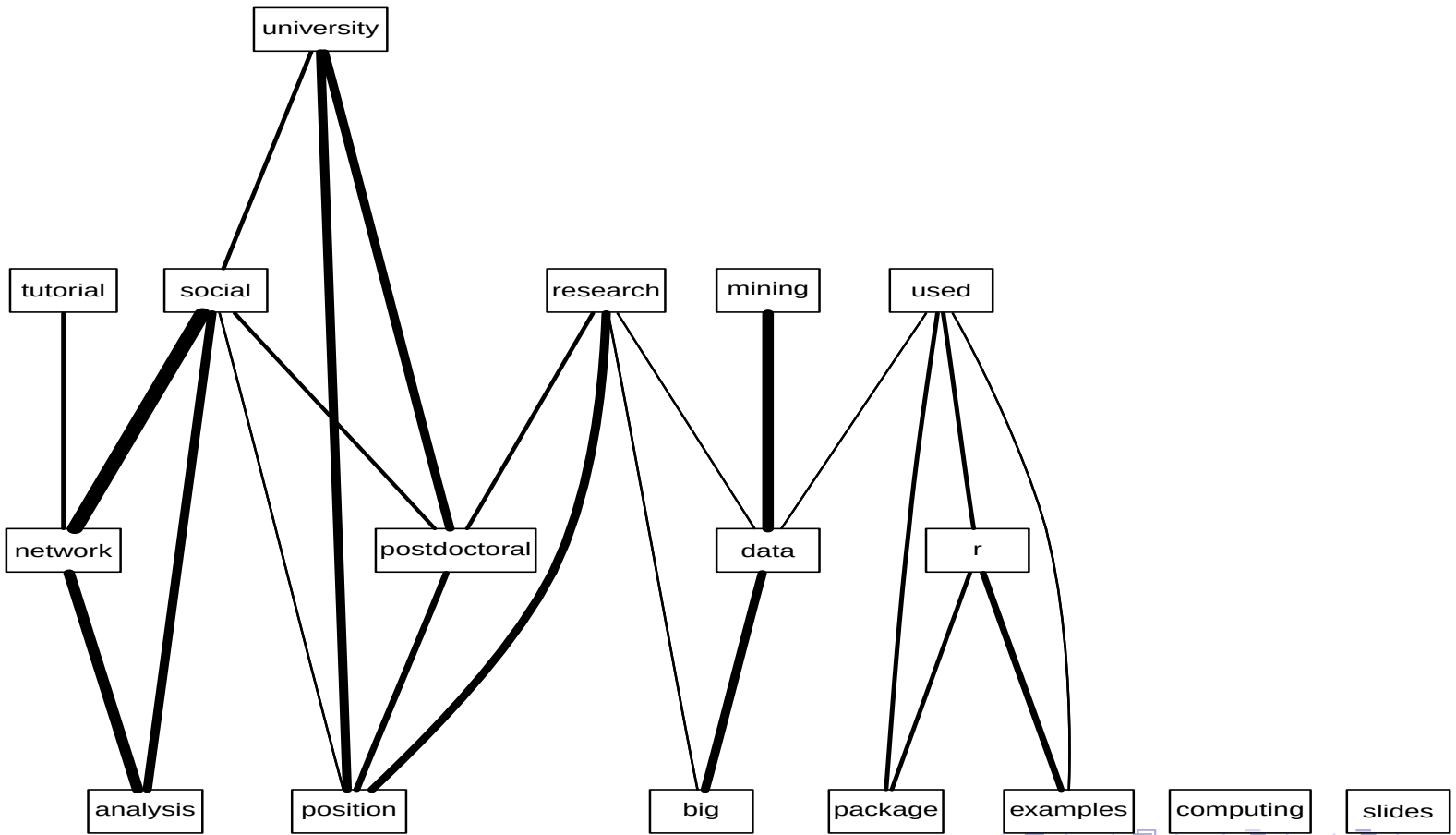
```
##  [1] "analysis"     "big"          "computing"    "data"    ..
##  [5] "examples"     "mining"       "network"      "package"..
##  [9] "position"     "postdoctoral" "r"            "research"..
## [13] "slides"       "social"       "tutorial"     "universi"..
## [17] "used"
```

Associations

```
# which words are associated with 'r'?
findAssocs(myTdm, "r", 0.2)

##              r
## examples 0.32
## code     0.29
## package  0.20

# which words are associated with 'mining'?
findAssocs(myTdm, "mining", 0.25)

##                  mining
## data               0.47
## mahout             0.30
## recommendation     0.30
## sets               0.30
## supports           0.30
## frequent           0.26
## itemset            0.26
```

Network of Terms

```
library(graph)
library(Rgraphviz)
plot(myTdm, term=freq.terms, corThreshold=0.1, weighting=T)
```



Word Cloud

```
library(wordcloud)
m <- as.matrix(myTdm)
freq <- sort(rowSums(m), decreasing=T)
wordcloud(words=names(freq), freq=freq, min.freq=4, random.order=F)
```

Topic Modelling

```
library(topicmodels)
set.seed(123)
myLda <- LDA(as.DocumentTermMatrix(myTdm), k=8)
terms(myLda, 5)
```

```
##      Topic 1    Topic 2  Topic 3    Topic 4
## [1,] "mining"   "data"   "r"        "position"
## [2,] "data"     "free"   "examples" "research"
## [3,] "analysis" "course" "code"     "university"
## [4,] "network"  "online" "book"     "data"
## [5,] "social"   "ausdm"  "mining"   "postdoctoral"
##      Topic 5        Topic 6     Topic 7     Topic 8
## [1,] "data"         "data"      "r"         "r"
## [2,] "r"            "scientist" "package"   "data"
## [3,] "mining"       "research"  "computing" "clustering"
## [4,] "applications" "r"         "slides"    "mining"
## [5,] "series"       "package"   "parallel"  "detection"
```

Outline

Time Series Analysis with R

- Time series decomposition: *decomp()*, *decompose()*, *arima()*, *stl()*
- Time series forecasting: *forecast*
- Time Series Clustering: *TSclust*
- Dynamic Time Warping (DTW): *dtw*

Outline

Social Network Analysis with R

- Packages: *igraph*, *sna*
- Centrality measures: *degree()*, *betweenness()*, *closeness()*, *transitivity()*
- Clusters: *clusters()*, *no.clusters()*
- Cliques: *cliques()*, *largest.cliques()*, *maximal.cliques()*, *clique.number()*
- Community detection: *fastgreedy.community()*, *spinglass.community()*
- Graph database Neo4j: package *RNeo4j*

  `http://nicolewhite.github.io/RNeo4j/`

Outline

R and Big Data Platforms

- Hadoop
  - Hadoop (or YARN) - a framework that allows for the distributed processing of large data sets across clusters of computers using simple programming models
  - R Packages: *RHadoop*, *RHIPE*
- Spark
  - Spark - a fast and general engine for large-scale data processing, which can be 100 times faster than Hadoop
  - *SparkR* - R frontend for Spark
- H2O
  - H2O - an open source in-memory prediction engine for big data science
  - R Package: *h2o*
- MongoDB
  - MongoDB - an open-source document database
  - R packages: *rmongodb*, *RMongo*

R and Hadoop

- Packages: *RHadoop*, *RHive*
- RHadoop[11] is a collection of R packages:
  - *rmr2* - perform data analysis with R via MapReduce on a Hadoop cluster
  - *rhdfs* - connect to Hadoop Distributed File System (HDFS)
  - *rhbase* - connect to the NoSQL HBase database
  - ...
- You can play with it on a single PC (in standalone or pseudo-distributed mode), and your code developed on that will be able to work on a cluster of PCs (in full-distributed mode)!
- Step-by-Step Guide to Setting Up an R-Hadoop System

  `http://www.rdatamining.com/big-data/r-hadoop-setup-guide`

[11] `https://github.com/RevolutionAnalytics/RHadoop/wiki`

An Example of MapReducing with R[12]

```
library(rmr2)
map <- function(k, lines) {
    words.list <- strsplit(lines, "\\s")
    words <- unlist(words.list)
    return(keyval(words, 1))
}
reduce <- function(word, counts) {
    keyval(word, sum(counts))
}
wordcount <- function(input, output = NULL) {
    mapreduce(input = input, output = output, input.format = "text",
        map = map, reduce = reduce)
}
## Submit job
out <- wordcount(in.file.path, out.file.path)
```

[12] From Jeffrey Breen's presentation on *Using R with Hadoop*
http://www.revolutionanalytics.com/news-events/free-webinars/2013/using-r-with-hadoop/

Outline

Online Resources

- RDataMining website: `http://www.rdatamining.com`
  - R Reference Card for Data Mining
  - RDataMining Slides Series
  - R and Data Mining: Examples and Case Studies
- RDataMining Group on LinkedIn (12,000+ members)

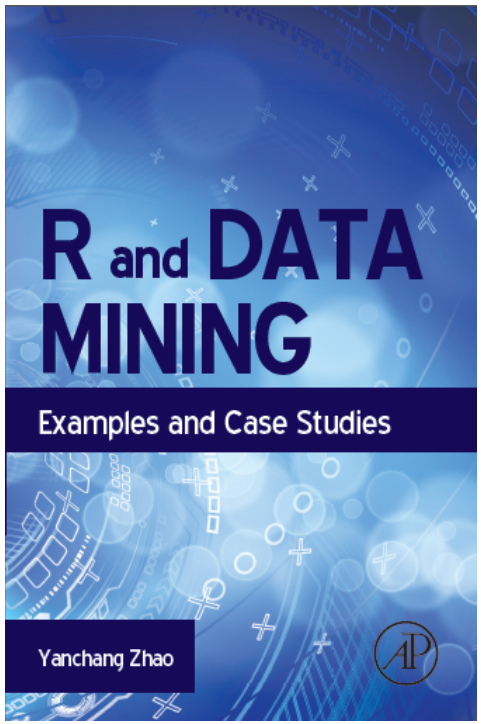
  `http://group.rdatamining.com`
- RDataMining on Twitter (2,000+ followers)

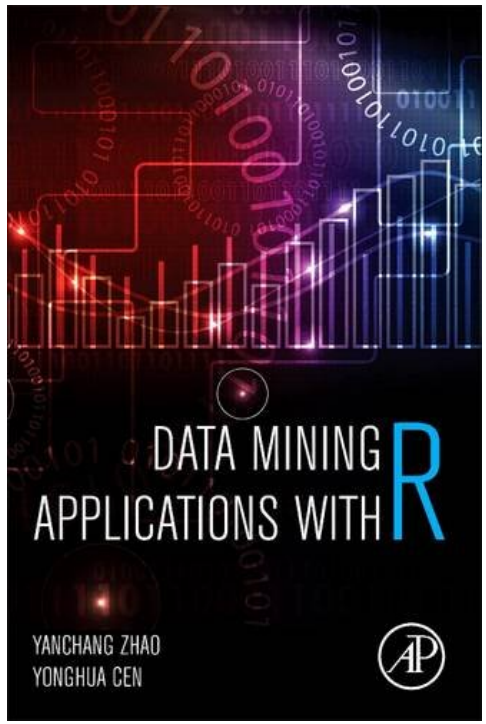
  `@RDataMining`
- Free online courses

  `http://www.rdatamining.com/resources/courses`
- Online documents

  `http://www.rdatamining.com/resources/onlinedocs`

The End





Thanks!

Email: yanchang(at)rdatamining.com