Advanced Analytics with R

Course Introduction

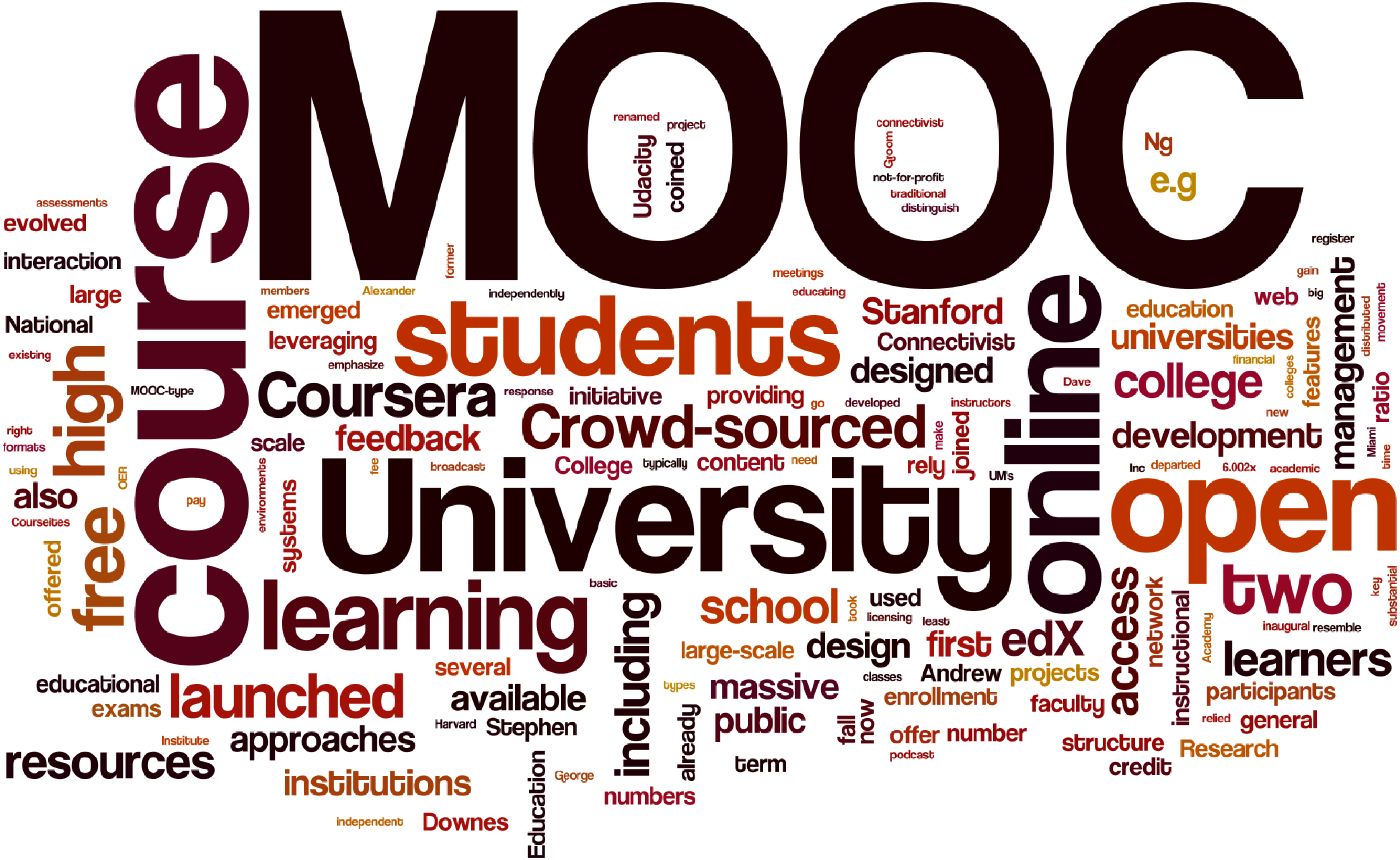
Forbes Earning Preview: H.J. Heinz

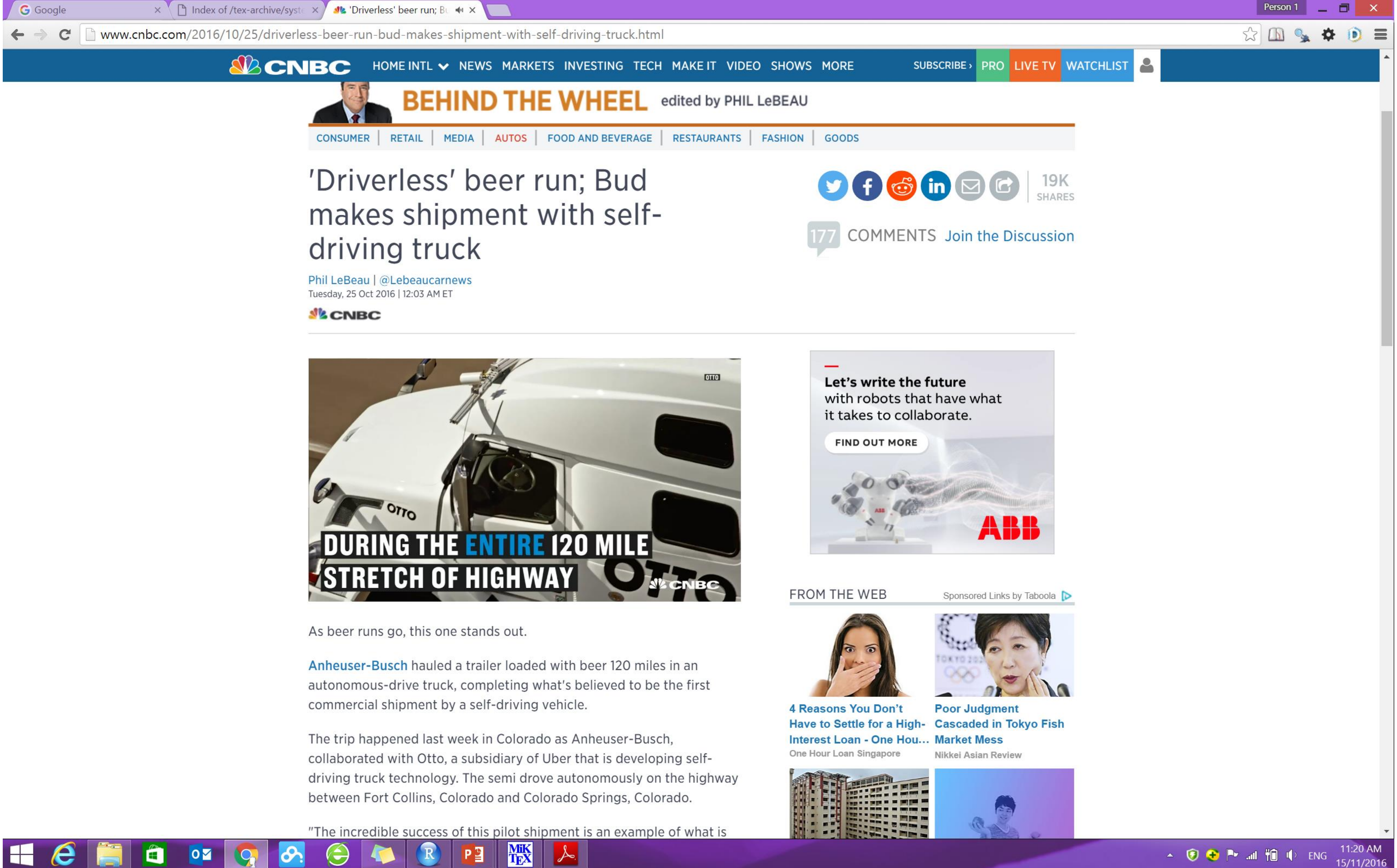
A quality first quarter earnings announcement could push shares of H.J. Heinz (HNZ) to a new 52-week high as the price is just 49 cents off the milestone heading into the company’s earnings release on Wednesday, August 29, 2012.

The Wall Street consensus is 80 cents per share, up 2.6 percent from a year ago when H.J reported earnings of 78 cents per share. The consensus estimate remains unchanged over the past month, but it has decreased from three months ago

when it was 82 cents. Analysts are expecting earnings of $3.52 per share for the fiscal year. Analysts project revenue to fall 0.3 percent year-over-year to $2.84 billion for the quarter, after being $2.85 billion a year ago . For the year, revenue is projected to roll in at $11.82 billion.

Paging DR. WATSON


MOOC
course
students
University
online
open
learning
Coursera
Crowd-sourced
Stanford
launched
two
high
free
resources
learners
edX
school
designed
development
college
universities
education
Connectivist
feedback
including
massive
public
approaches
available
institutions
design
first
access
management
features
ratio
web
evolved
interaction
large
National
assessments
existing
right
formats
using
also
Courseites
offered
educational
exams
Institute
independent
Downes
Education
George
numbers
already
term
Stephen
Harvard
several
large-scale
types
enrollment
offer number
podcast
fall
now
Andrew
projects
faculty
structure
credit
Research
general
relied
participants
inaugural
resemble
instructional
network
Academy
key
substantial
classes
least
licensing
took
used
basic
systems
environments
fee
pay
OER
MOOC-type
emphasize
leveraging
emerged
members
Alexander
former
independently
response
initiative
providing
go
developed
instructors
Dave
scale
broadcast
College
typically
content
need
rely
make
joined
UM's
Inc
departed
6.002x
academic
Miami
time
new
colleges
financial
big
gain
register
distributed
movement
meetings
educating
renamed
project
Udacity
coined
connectivist
Groom
not-for-profit
traditional
distinguish
Ng
e.g



'Driverless' beer run; Bud makes shipment with self-driving truck

Phil LeBeau | @Lebeaucarnews
Tuesday, 25 Oct 2016 | 12:03 AM ET

As beer runs go, this one stands out.

Anheuser-Busch hauled a trailer loaded with beer 120 miles in an autonomous-drive truck, completing what's believed to be the first commercial shipment by a self-driving vehicle.

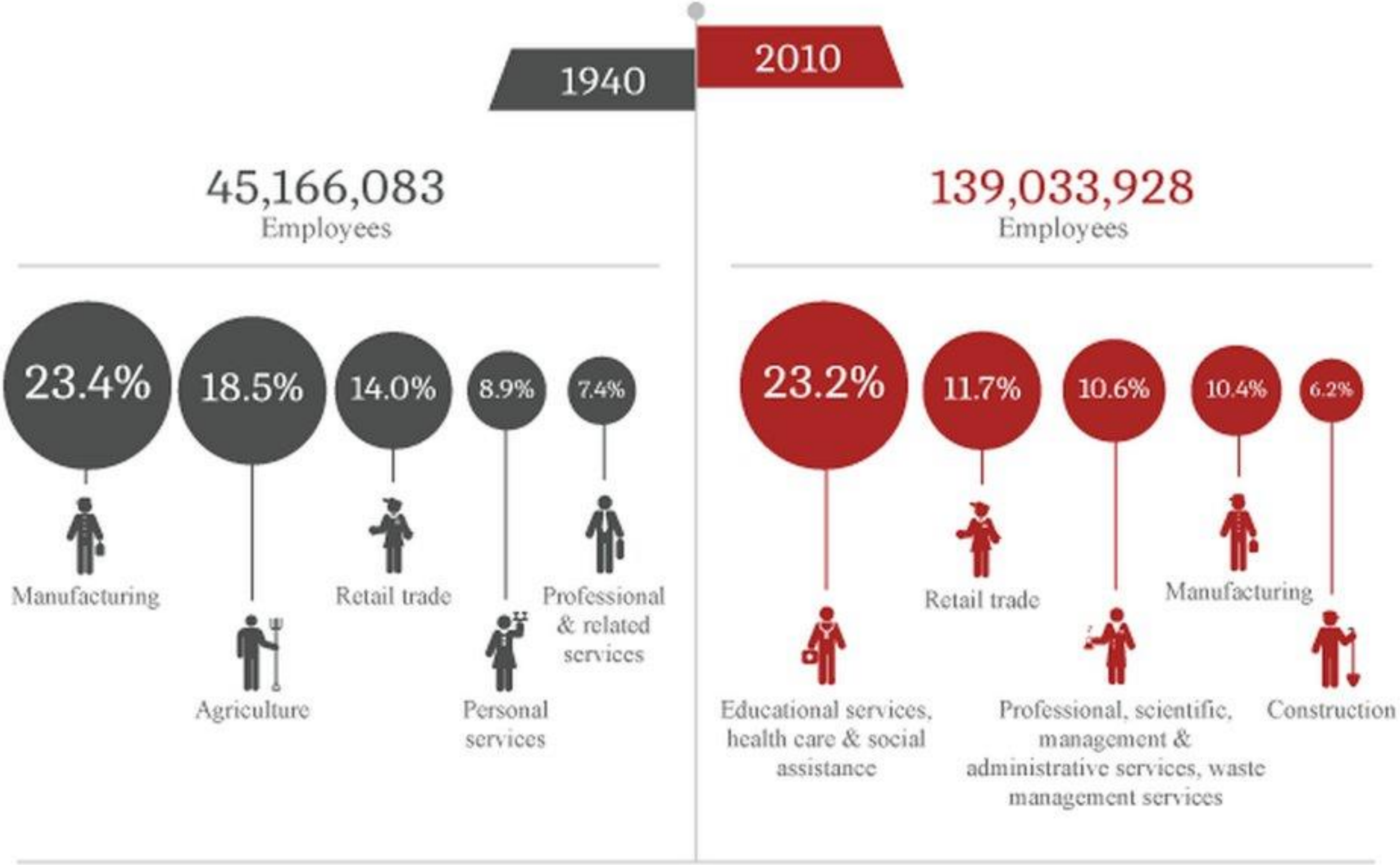
The trip happened last week in Colorado as Anheuser-Busch, collaborated with Otto, a subsidiary of Uber that is developing self-driving truck technology. The semi drove autonomously on the highway between Fort Collins, Colorado and Colorado Springs, Colorado.

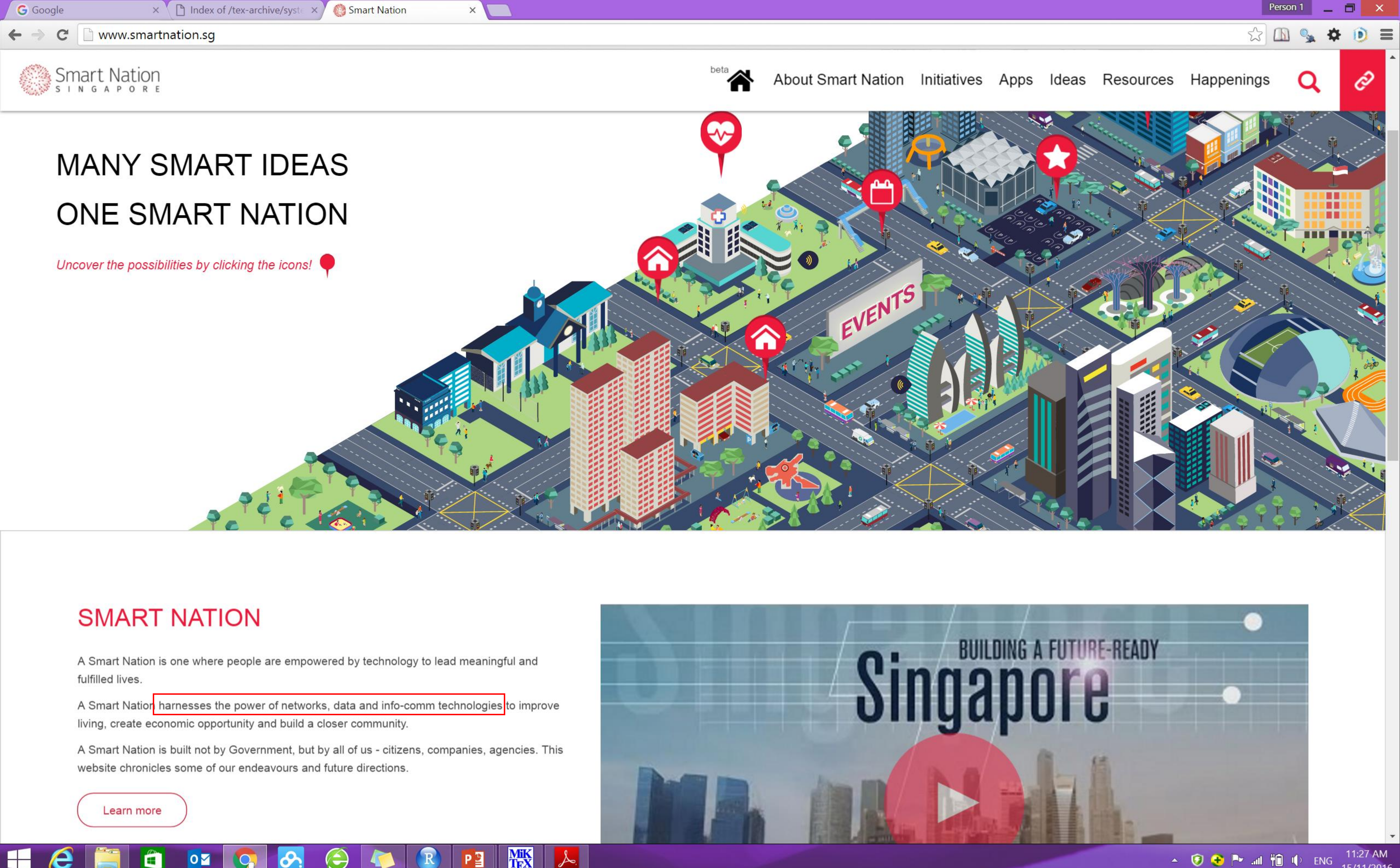
"The incredible success of this pilot shipment is an example of what is

Let's write the future with robots that have what it takes to collaborate.
FIND OUT MORE
ABB

FROM THE WEB
Sponsored Links by Taboola
4 Reasons You Don't Have to Settle for a High-Interest Loan - One Hou…
One Hour Loan Singapore
Poor Judgment Cascaded in Tokyo Fish Market Mess
Nikkei Asian Review

Our workforce and industries have changed dramatically over time.





MANY SMART IDEAS

ONE SMART NATION

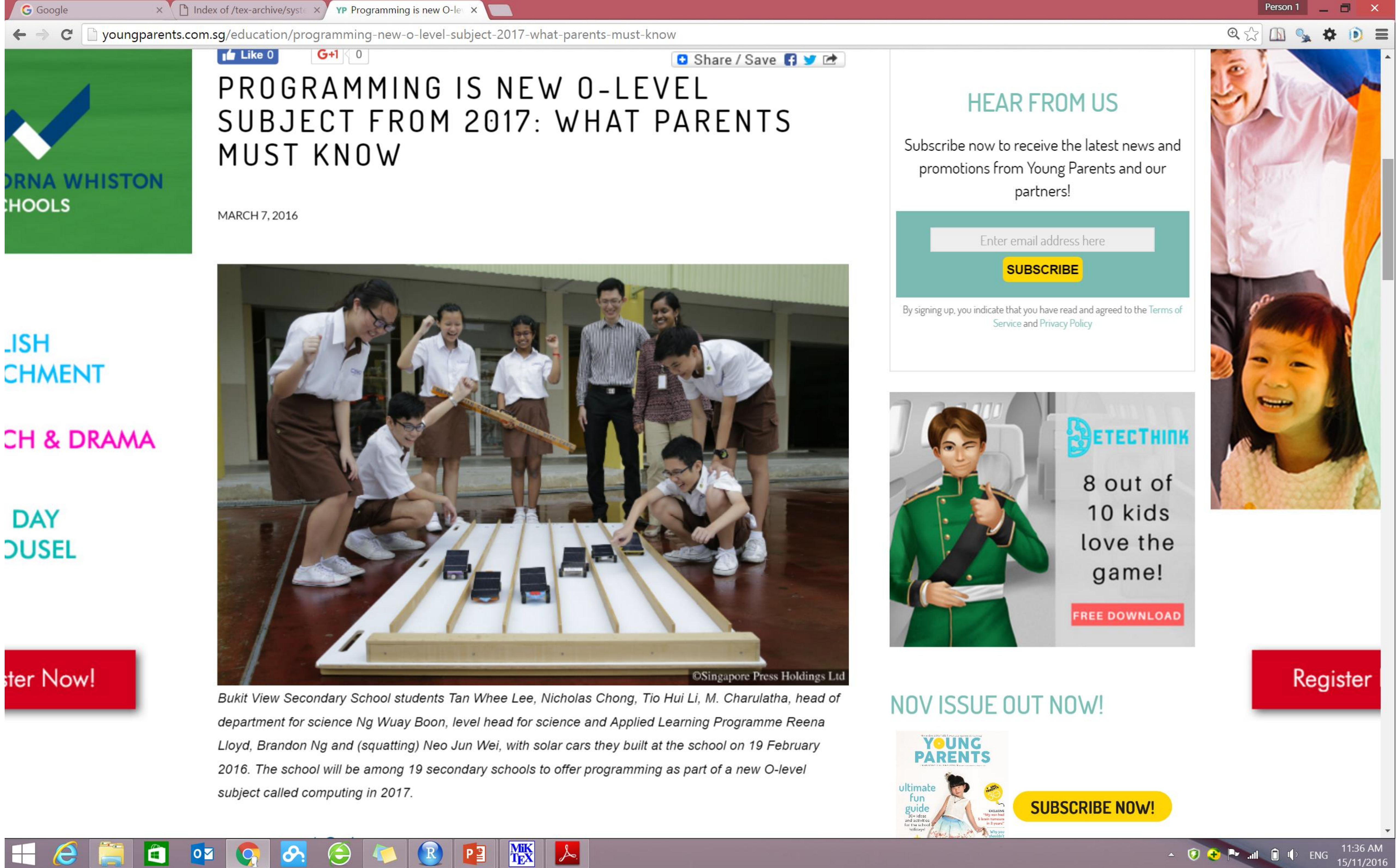
Uncover the possibilities by clicking the icons!

SMART NATION

A Smart Nation is one where people are empowered by technology to lead meaningful and fulfilled lives.

A Smart Nation harnesses the power of networks, data and info-comm technologies to improve living, create economic opportunity and build a closer community.

A Smart Nation is built not by Government, but by all of us - citizens, companies, agencies. This website chronicles some of our endeavours and future directions.


Google
Index of /tex-archive/syst
YP Programming is new O-le
Person 1
youngparents.com.sg/education/programming-new-o-level-subject-2017-what-parents-must-know
Like 0
G+1 0
Share / Save

PROGRAMMING IS NEW O-LEVEL SUBJECT FROM 2017: WHAT PARENTS MUST KNOW

MARCH 7, 2016

©Singapore Press Holdings Ltd

Bukit View Secondary School students Tan Whee Lee, Nicholas Chong, Tio Hui Li, M. Charulatha, head of department for science Ng Wuay Boon, level head for science and Applied Learning Programme Reena Lloyd, Brandon Ng and (squatting) Neo Jun Wei, with solar cars they built at the school on 19 February 2016. The school will be among 19 secondary schools to offer programming as part of a new O-level subject called computing in 2017.

HEAR FROM US

Subscribe now to receive the latest news and promotions from Young Parents and our partners!

Enter email address here

SUBSCRIBE

By signing up, you indicate that you have read and agreed to the Terms of Service and Privacy Policy

DETECTHINK
8 out of 10 kids love the game!
FREE DOWNLOAD

NOV ISSUE OUT NOW!

YOUNG PARENTS
ultimate fun guide

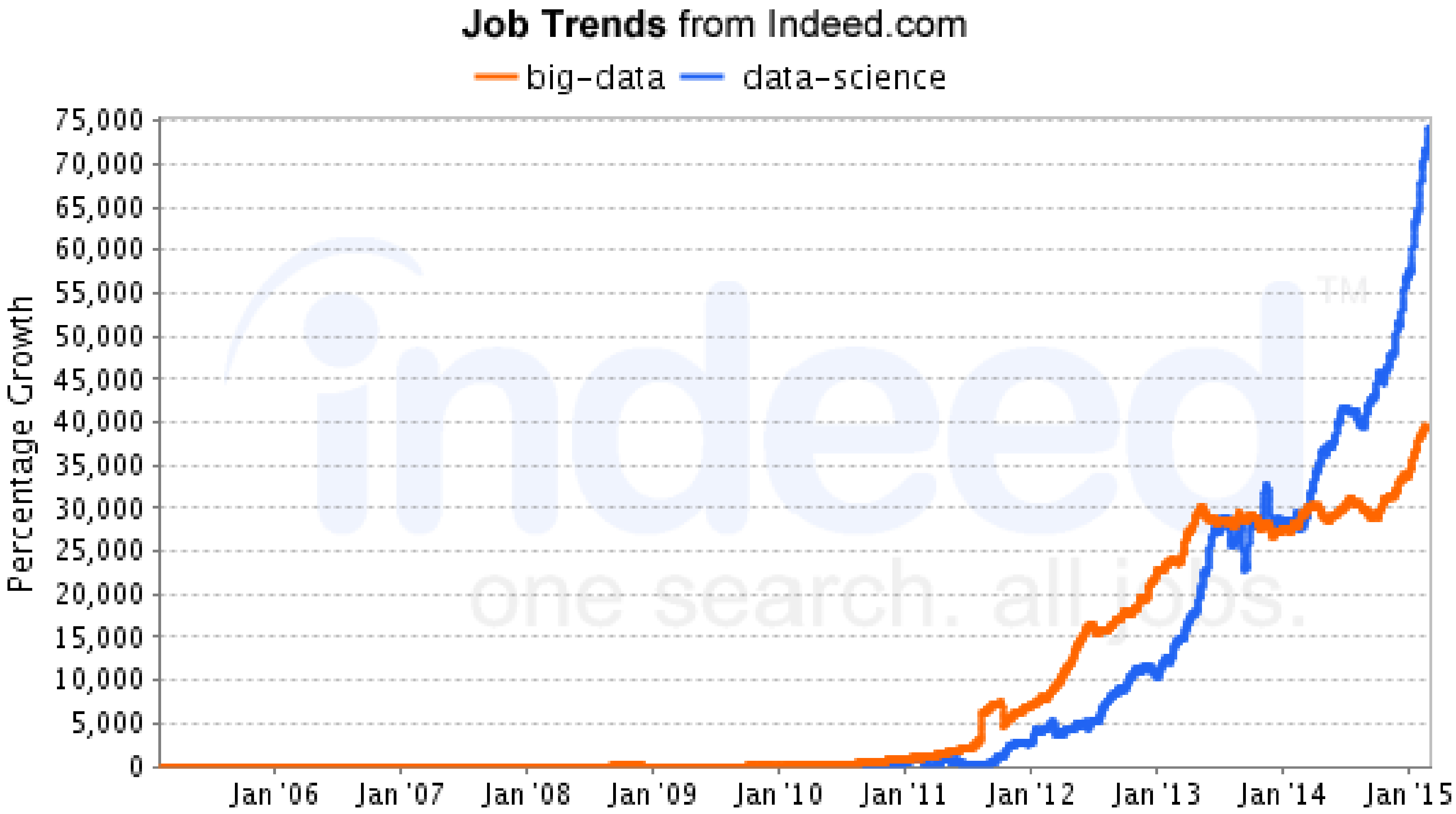
SUBSCRIBE NOW!

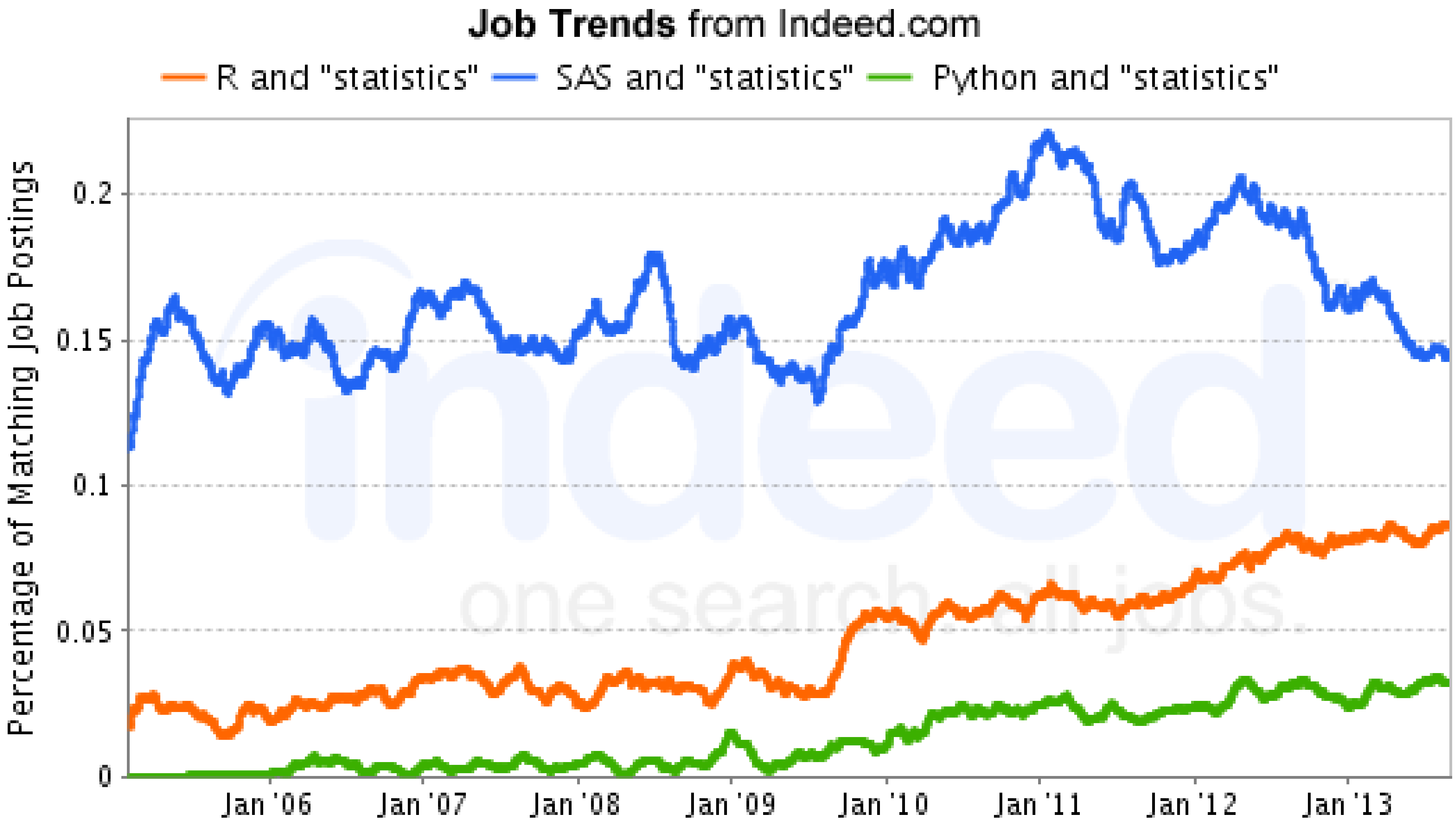
RNA WHISTON
HOOLS
LISH
CHMENT
CH & DRAMA
DAY
OUSEL
ter Now!

Register

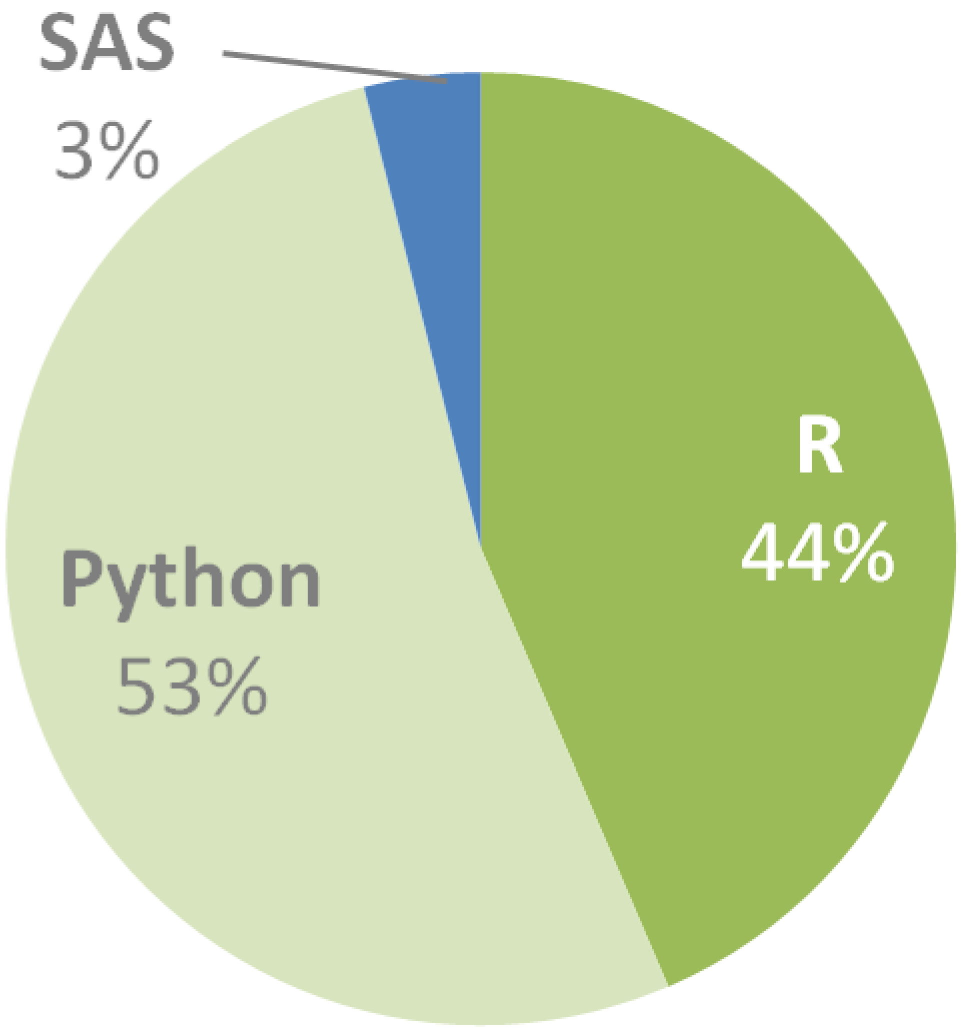
Mining

Data Mining

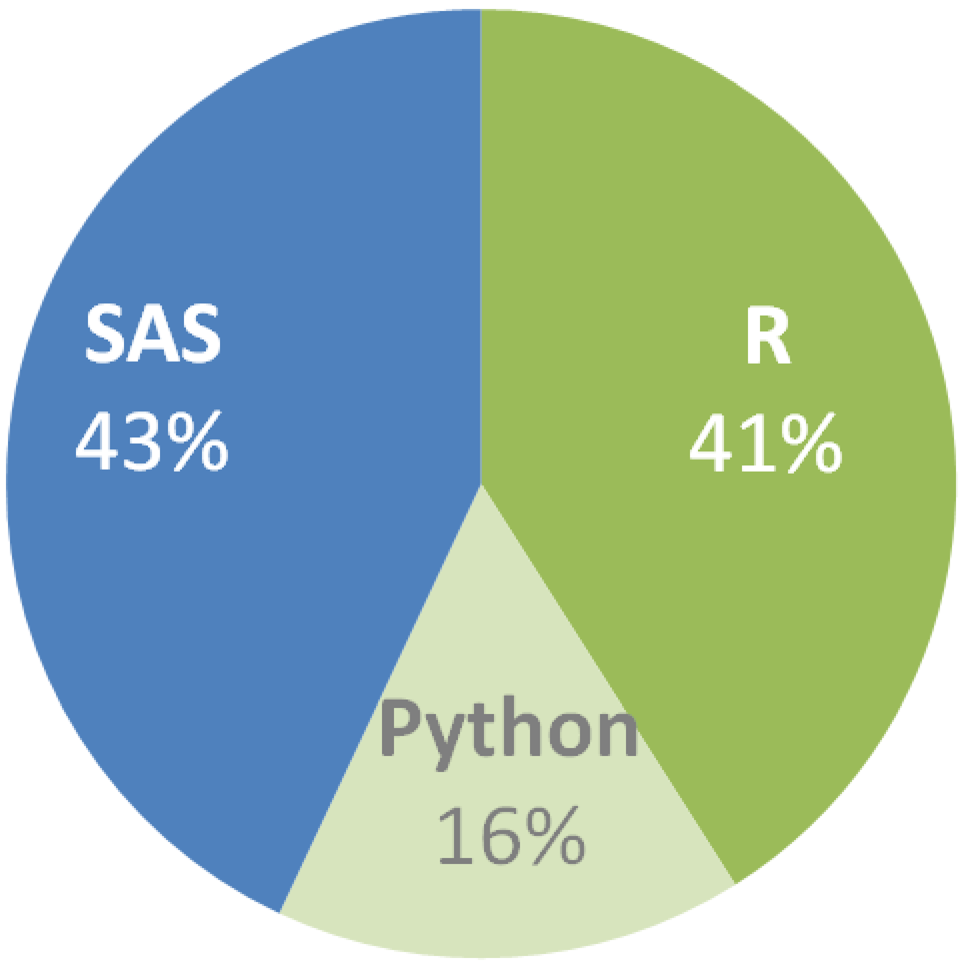

Job Trends from Indeed.com
big-data
data-science
Percentage Growth
75,000
70,000
65,000
60,000
55,000
50,000
45,000
40,000
35,000
30,000
25,000
20,000
15,000
10,000
5,000
0
Jan '06
Jan '07
Jan '08
Jan '09
Jan '10
Jan '11
Jan '12
Jan '13
Jan '14
Jan '15
indeed
one search. all jobs.


Job Trends from Indeed.com
R and "statistics"
SAS and "statistics"
Python and "statistics"
Percentage of Matching Job Postings
0.2
0.15
0.1
0.05
0
Jan '06
Jan '07
Jan '08
Jan '09
Jan '10
Jan '11
Jan '12
Jan '13
indeed
one search. all jobs.

Data Scientist



Predictive Analytics



Why R?

- R is free.
- R is a language. (You have the full flexibility and control)
- R is powerful.
- R has an amazing ecosystem of developers.

Learning Objectives

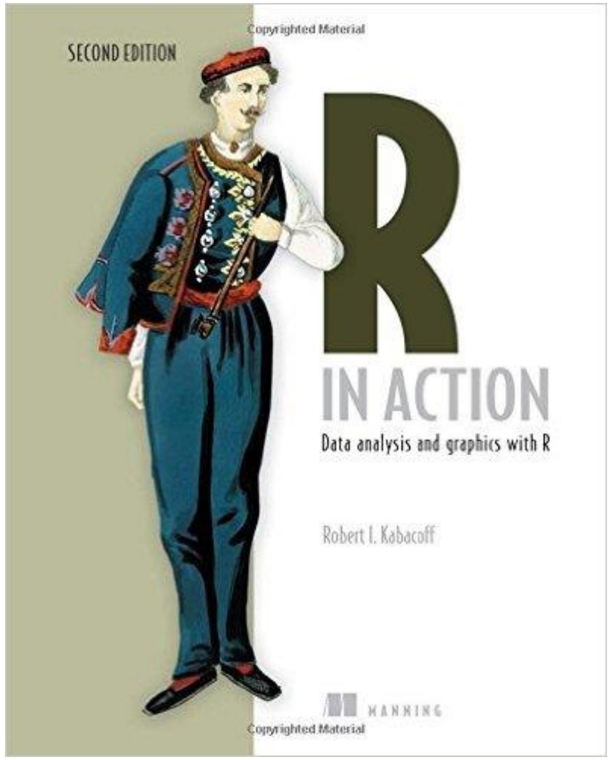
- Master basic R programming techniques to model and analyse data.
- Self-explore R packages for analytics needs.
- Understand some common analytics methodologies used in current business environment.
- Appreciate what data science is.

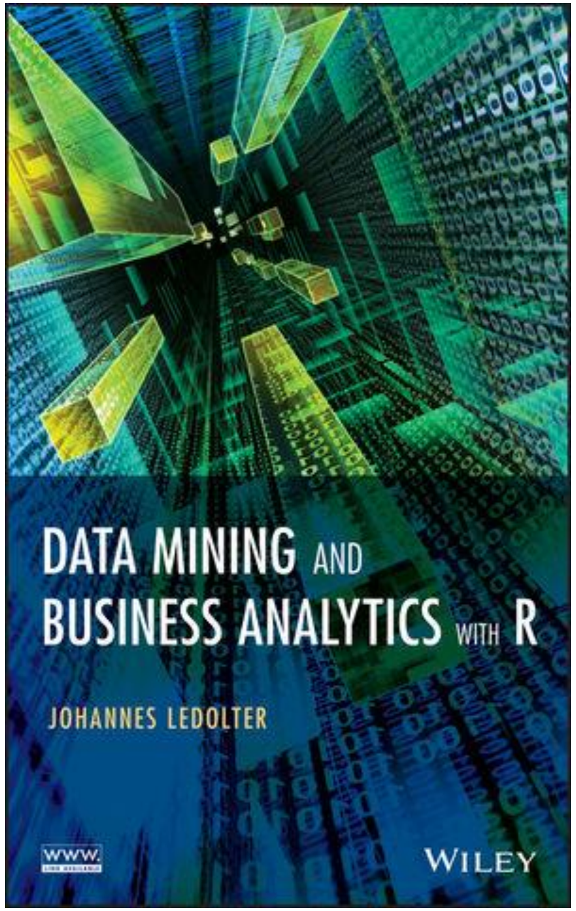
I am training what industry wants …

Right expectations

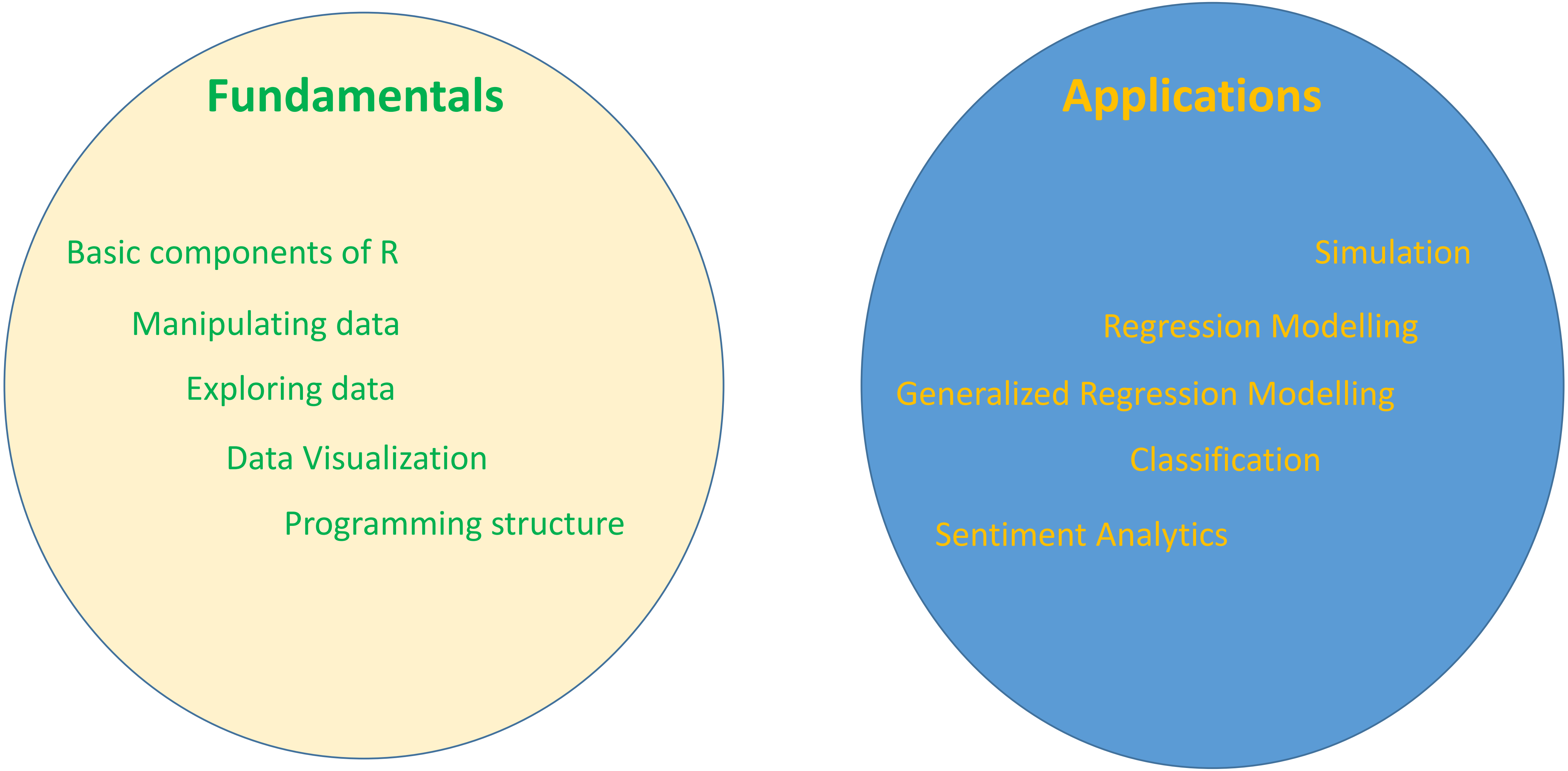
- Learning programming can be very painful
    - But, believe me, it will be very rewarding
- Be prepared to solve problems yourself
- Be prepared to be better than the professor!!!

Readings


Copyrighted Material
SECOND EDITION
R
IN ACTION
Data analysis and graphics with R
Robert I. Kabacoff
MANNING
Copyrighted Material


DATA MINING AND
BUSINESS ANALYTICS WITH R
JOHANNES LEDOLTER
WWW.
WILEY

Topics



Fundamentals

Basic components of R

Manipulating data

Exploring data

Data Visualization

Programming structure

Applications

Simulation

Regression Modelling

Generalized Regression Modelling

Classification

Sentiment Analytics

Assessment

• Class Participation	20%
• Group Project	40%
• Test 1	20%
• Test 2	20%

Public datasets

- http://hadoopilluminated.com/hadoop_illuminated/Public_Bigdata_Sets.html
- https://www.quora.com/Data/Where-can-I-find-large-datasets-open-to-the-public
- http://www.kdnuggets.com/datasets/index.html
- http://www.researchpipeline.com/mediawiki/index.php?title=Main_Page

Places to get help

- IVLE
- https://www.r-project.org/
- Stackoverflow.com
- Cross Validated

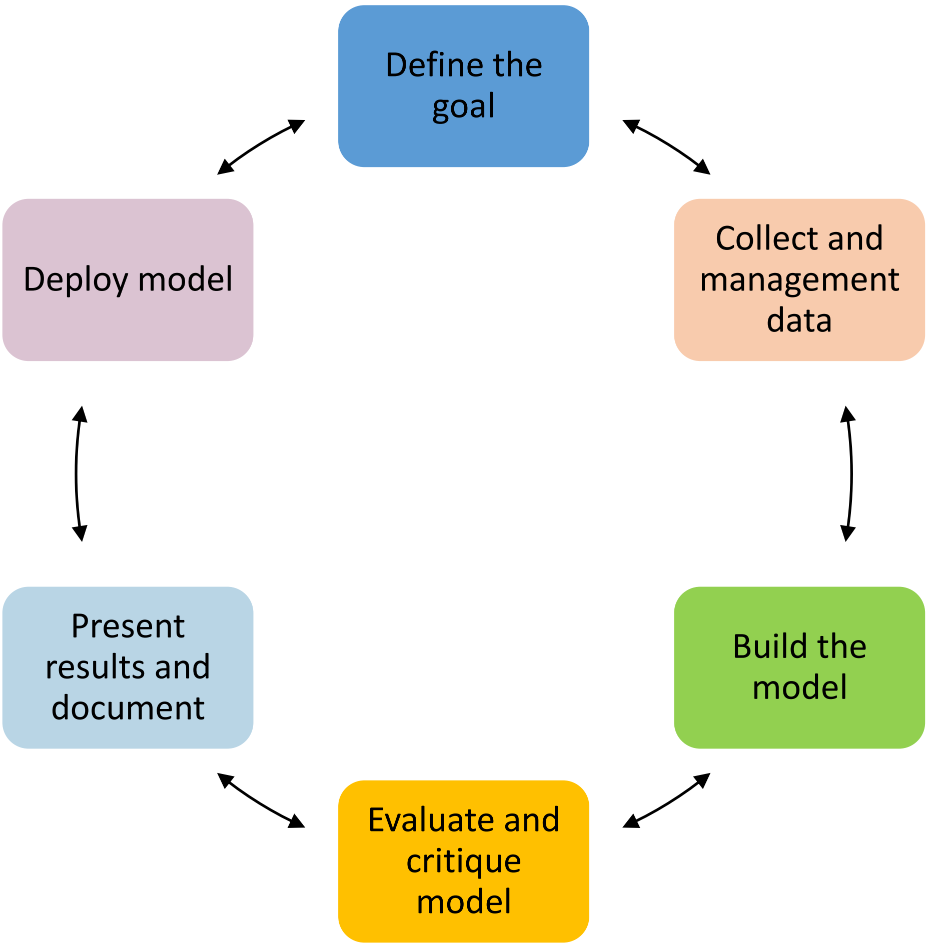
R Communities

- http://www.r-bloggers.com/
- https://twitter.com/search?q=%23rstats&src=typd
- http://www.revolutionanalytics.com/r-community

Good about R

- What you can think of, someone may have done it for you.

Data Science Project



What problem am I solving?

Define the goal

Collect and management data

What information do I need?

Build the model

Find patterns in the data that lead to solutions

Evaluate and critique model

Does the model solve my problem?

Present results and document

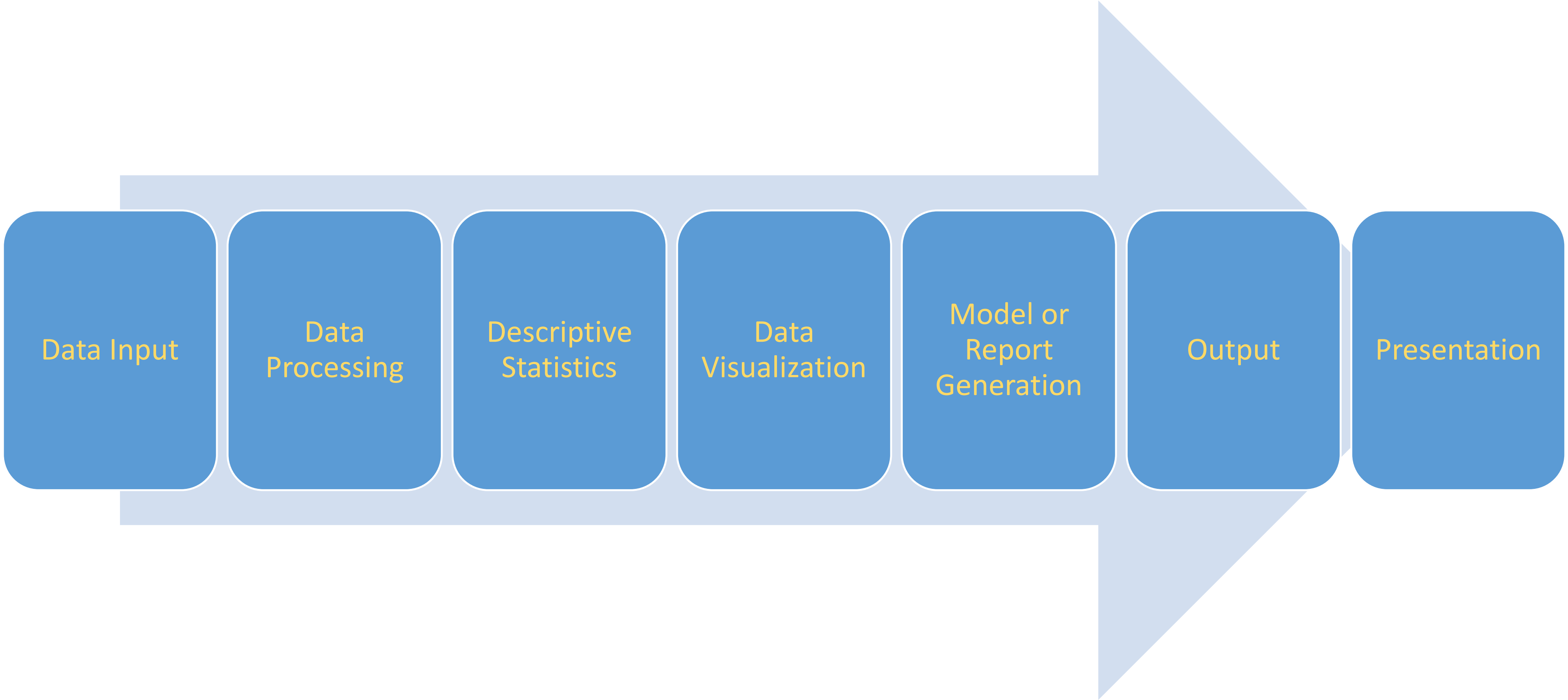
Establish that I can solve the problem and how

Deploy model

Deploy the model to solve the problem in the real world

Source: “Practical Data Science with R”

Basic Analytics Steps



Data Input

Data Processing

Descriptive Statistics

Data Visualization

Model or Report Generation

Output

Presentation

Fundamental of R Programming

Installation

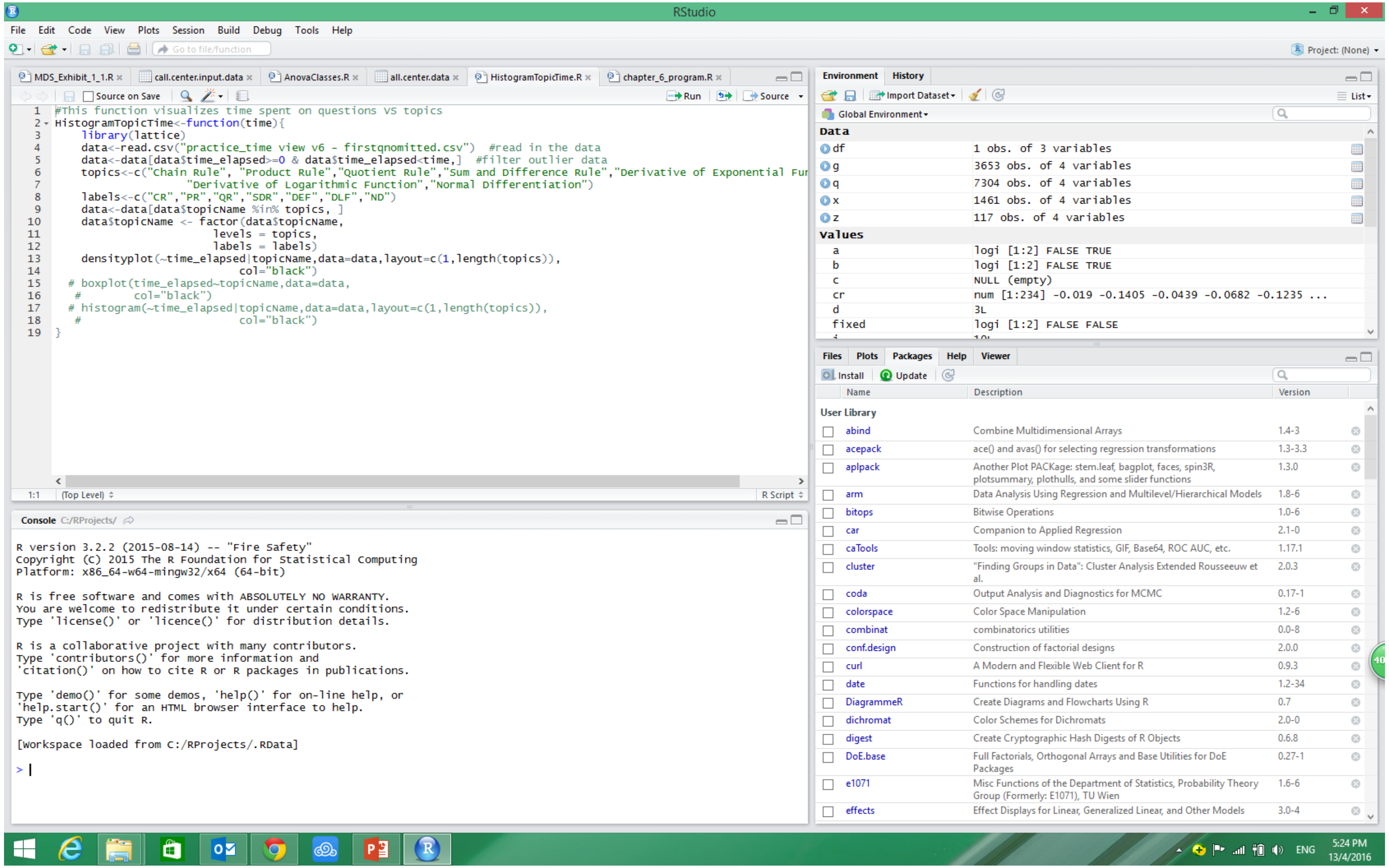
- R: the core engine

  Download at http://www.r-project.org/ and install

- RStudio: the IDE

  Download at http://www.rstudio.com/ide/ and install

Introduction to R Studio



```r
#This function visualizes time spent on questions VS topics
HistogramTopicTime<-function(time){
    library(lattice)
    data<-read.csv("practice_time view v6 - firstqnomitted.csv")  #read in the data
    data<-data[data$time_elapsed>=0 & data$time_elapsed<time,]  #filter outlier data
    topics<-c("Chain Rule", "Product Rule","Quotient Rule","Sum and Difference Rule","Derivative of Exponential Fur
                    "Derivative of Logarithmic Function","Normal Differentiation")
    labels<-c("CR","PR","QR","SDR","DEF","DLF","ND")
    data<-data[data$topicName %in% topics, ]
    data$topicName <- factor(data$topicName,
                             levels = topics,
                             labels = labels)
    densityplot(~time_elapsed|topicName,data=data,layout=c(1,length(topics)),
                                  col="black")
  # boxplot(time_elapsed~topicName,data=data,
   #         col="black")
  # histogram(~time_elapsed|topicName,data=data,layout=c(1,length(topics)),
   #                              col="black")
}
```

Console C:/RProjects/

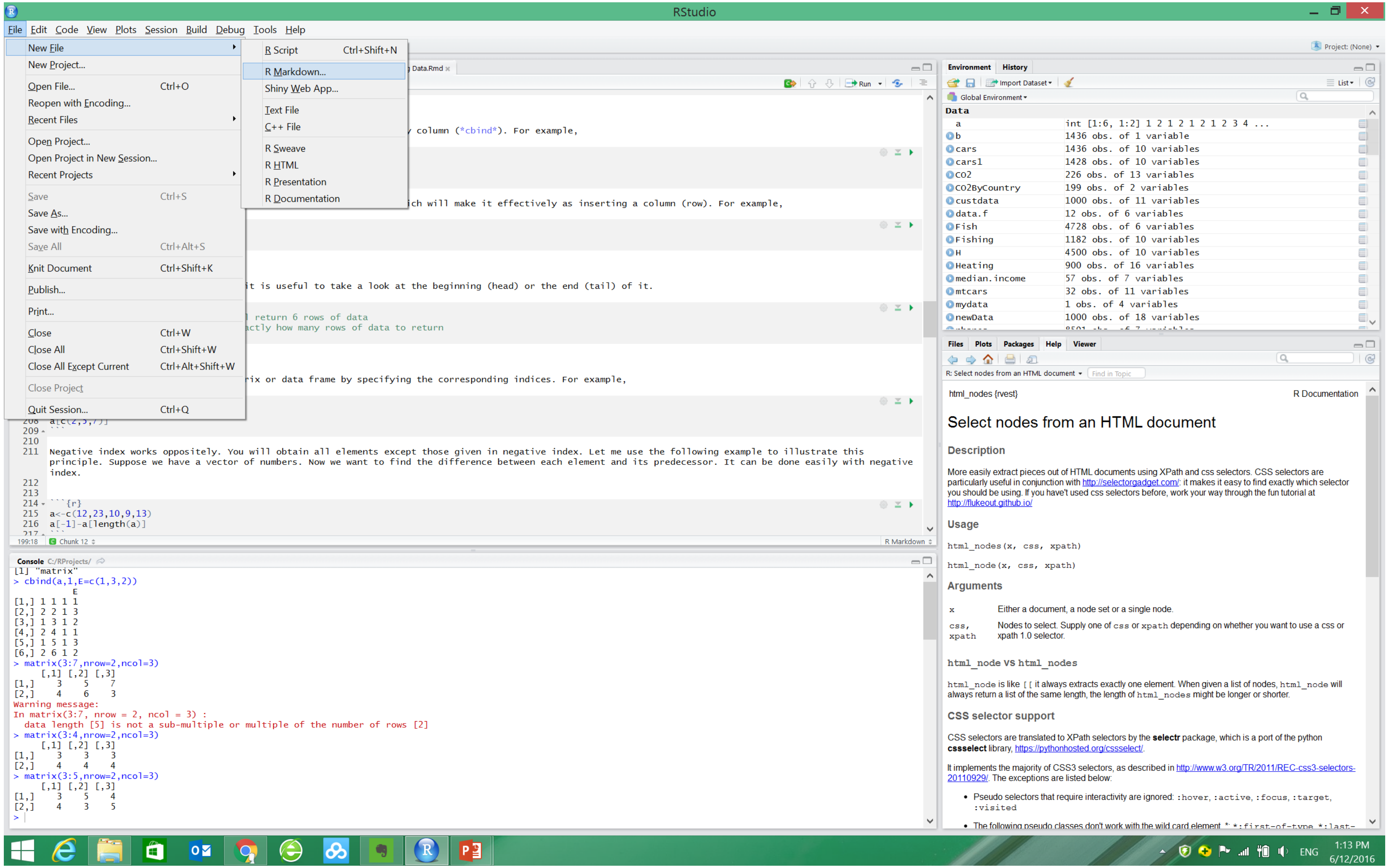
```
R version 3.2.2 (2015-08-14) -- "Fire Safety"
Copyright (C) 2015 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

[workspace loaded from C:/RProjects/.RData]

> |
```

```
Data
df       1 obs. of 3 variables
g        3653 obs. of 4 variables
q        7304 obs. of 4 variables
x        1461 obs. of 4 variables
z        117 obs. of 4 variables
Values
a        logi [1:2] FALSE TRUE
b        logi [1:2] FALSE TRUE
c        NULL (empty)
cr       num [1:234] -0.019 -0.1405 -0.0439 -0.0682 -0.1235 ...
d        3L
fixed    logi [1:2] FALSE FALSE
```

User Library

	Name	Description	Version
□	abind	Combine Multidimensional Arrays	1.4-3
□	acepack	ace() and avas() for selecting regression transformations	1.3-3.3
□	aplpack	Another Plot PACKage: stem.leaf, bagplot, faces, spin3R, plotsummary, plothulls, and some slider functions	1.3.0
□	arm	Data Analysis Using Regression and Multilevel/Hierarchical Models	1.8-6
□	bitops	Bitwise Operations	1.0-6
□	car	Companion to Applied Regression	2.1-0
□	caTools	Tools: moving window statistics, GIF, Base64, ROC AUC, etc.	1.17.1
□	cluster	"Finding Groups in Data": Cluster Analysis Extended Rousseeuw et al.	2.0.3
□	coda	Output Analysis and Diagnostics for MCMC	0.17-1
□	colorspace	Color Space Manipulation	1.2-6
□	combinat	combinatorics utilities	0.0-8
□	conf.design	Construction of factorial designs	2.0.0
□	curl	A Modern and Flexible Web Client for R	0.9.3
□	date	Functions for handling dates	1.2-34
□	DiagrammeR	Create Diagrams and Flowcharts Using R	0.7
□	dichromat	Color Schemes for Dichromats	2.0-0
□	digest	Create Cryptographic Hash Digests of R Objects	0.6.8
□	DoE.base	Full Factorials, Orthogonal Arrays and Base Utilities for DoE Packages	0.27-1
□	e1071	Misc Functions of the Department of Statistics, Probability Theory Group (Formerly: E1071), TU Wien	1.6-6
□	effects	Effect Displays for Linear, Generalized Linear, and Other Models	3.0-4

Making notes using R Markdown



```
y column (*cbind*). For example,

ich will make it effectively as inserting a column (row). For example,

it is useful to take a look at the beginning (head) or the end (tail) of it.

] return 6 rows of data
actly how many rows of data to return

rix or data frame by specifying the corresponding indices. For example,

a[c(2,3,7)]
```

Negative index works oppositely. You will obtain all elements except those given in negative index. Let me use the following example to illustrate this
principle. Suppose we have a vector of numbers. Now we want to find the difference between each element and its predecessor. It can be done easily with negative
index.

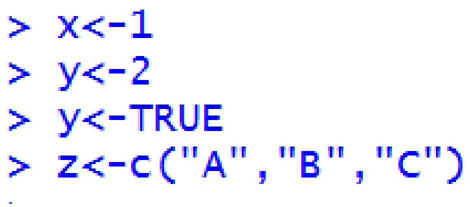
```{r}
a<-c(12,23,10,9,13)
a[-1]-a[length(a)]
```
```

```
[1] "matrix"
> cbind(a,1,E=c(1,3,2))
          E
[1,] 1 1 1 1
[2,] 2 2 1 3
[3,] 1 3 1 2
[4,] 2 4 1 1
[5,] 1 5 1 3
[6,] 2 6 1 2
> matrix(3:7,nrow=2,ncol=3)
     [,1] [,2] [,3]
[1,]    3    5    7
[2,]    4    6    3
Warning message:
In matrix(3:7, nrow = 2, ncol = 3) :
  data length [5] is not a sub-multiple or multiple of the number of rows [2]
> matrix(3:4,nrow=2,ncol=3)
     [,1] [,2] [,3]
[1,]    3    3    3
[2,]    4    4    4
> matrix(3:5,nrow=2,ncol=3)
     [,1] [,2] [,3]
[1,]    3    5    4
[2,]    4    3    5
> 
```

Data

a	int [1:6, 1:2] 1 2 1 2 1 2 1 2 3 4 ...
b	1436 obs. of 1 variable
cars	1436 obs. of 10 variables
cars1	1428 obs. of 10 variables
co2	226 obs. of 13 variables
CO2ByCountry	199 obs. of 2 variables
custdata	1000 obs. of 11 variables
data.f	12 obs. of 6 variables
Fish	4728 obs. of 6 variables
Fishing	1182 obs. of 10 variables
H	4500 obs. of 10 variables
Heating	900 obs. of 16 variables
median.income	57 obs. of 7 variables
mtcars	32 obs. of 11 variables
mydata	1 obs. of 4 variables
newData	1000 obs. of 18 variables

html_nodes {rvest} R Documentation

Select nodes from an HTML document

Description

More easily extract pieces out of HTML documents using XPath and css selectors. CSS selectors are particularly useful in conjunction with http://selectorgadget.com/: it makes it easy to find exactly which selector you should be using. If you have't used css selectors before, work your way through the fun tutorial at http://flukeout.github.io/

Usage

```
html_nodes(x, css, xpath)

html_node(x, css, xpath)
```

Arguments

x	Either a document, a node set or a single node.
css, xpath	Nodes to select. Supply one of css or xpath depending on whether you want to use a css or xpath 1.0 selector.

html_node vs html_nodes

html_node is like [[it always extracts exactly one element. When given a list of nodes, html_node will always return a list of the same length, the length of html_nodes might be longer or shorter.

CSS selector support

CSS selectors are translated to XPath selectors by the **selectr** package, which is a port of the python **cssselect** library, https://pythonhosted.org/cssselect/.

It implements the majority of CSS3 selectors, as described in http://www.w3.org/TR/2011/REC-css3-selectors-20110929/. The exceptions are listed below:

- Pseudo selectors that require interactivity are ignored: :hover, :active, :focus, :target, :visited
- The following pseudo classes don't work with the wild card element, *: *:first-of-type, *:last-
```

Working Environment

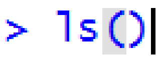
- getwd() - Get the current working directory
- setwd() - Set the working directory to a specific folder
- list.files() - list all the files/directories under a specific folder
- objects() – list all the objects stored in current environment

Create Objects



```
> x<-1
> y<-2
> y<-TRUE
> z<-c("A","B","C")
```

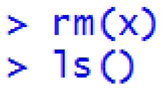
Create three new objects



```
> ls()
```

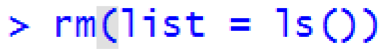
Check that the new objects are in the working environment

Remove Object



```
> rm(x)
> ls()
```

Remove x from the environment



```
> rm(list = ls())
```

Remove all objects from the environment

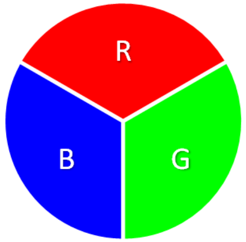
Warning: Don’t do this unless necessary!!!

Exercise

- Open “Lasagna Triers.csv” file.
- Discuss the various types of fields in the file.

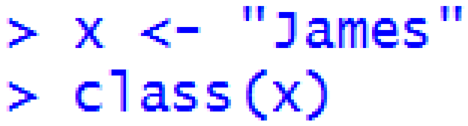
Data Type

- R has five basic or “atomic” classes of objects
  - Character
  - Numeric
  - Integer
  - Complex
  - logical
- On top of basic classes, you can build objects such as
  - Vector
  - List
  - Matrix
  - Factor
  - Data Frames



Character

- A ***character*** object is used to represent string values in R. For example, “James”, “China”, etc.



```
> x <- "James"
> class(x)
```

Commonly used functions on character

Function	Description	Example
nchar()	Get the number of characters in the string	*a<-"Singapore"* *nchar(a)*
regexpr()	Find the starting position of a small string in a large string	*regexpr("ex", "longtext")*
gregexpr()	find positions of every match of a small sting in a large string	*gregexpr("a","banana")*
grep()	find the positions of a regular expression in a vector of text strings	*txt<-c("arm","foot","lefroo", "bafoobar")* *grep("foo", txt)*
substr()	extract part of a text string based on position in the text string	*substr("Singapore",2,4)*
sub()	replace the first match of a string with a new string	*sub("or","es","Singapore")*
gsub()	replace every match of a given sub string in a string with a new string	*gsub("a","o","banana")*
paste()	combine two strings into a new string	*paste("Liu", "Qizhang")*

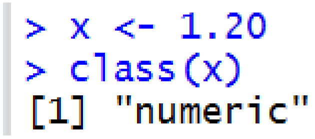
Logical

- A ***logical*** value is normally produced by some logical operations.

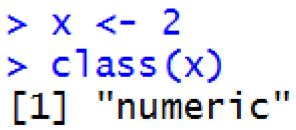
Operator	Description	Example
>	Check if a number is more than the other	5 > 3
<	Check if a number is less than the other	3 < 5
==	Check if two values are the same	3 == 5
<=	Check if a number is less than or equal to the other	3 <= 5
>=	Check if a number is greater than or equal to the other	3 >= 5
!=	Check if a number is unequal to the other	3 != 5
&	(AND) It combines two logical values. It is TRUE only if both logical values are TRUE	(5>3) & (4>2)
|	(OR) It combines two logical values. It is FALSE only if both logical values are FALSE	(5>3) | (4>2)
!	(NOT) Gives opposite value of the given logical value	!(5 > 3)

Numeric

- Decimal values are called numeric in R. It is the **default** computational data type.



```
> x <- 1.20
> class(x)
[1] "numeric"
```



```
> x <- 2
> class(x)
[1] "numeric"
```

Integer

- Integer is a special data type for integer values.

```
x<-as.integer(2)
class(x)

x<-2L
class(x)
```

```
x<-as.integer(5>6)
x
```

Coercion

- Use as(object) function to convert an object from one data type to another. This is called ***coercion***.
- Coercion order: logical < integer < numeric < complex < character

Vector

- Vector is the most basic data structure. It stores an ordered array of elements of the same data type.

```
a<-c(3,1,5)
a
```

```
b<-1:4
b
```

```
c<-c(a,b)
c
```

```
d<-c("Name",1)
d
```

Special vectors

- seq() creates a vector of numbers in a special sequence.

```
seq(1,9,2)
```

- rep() creates a vector which is a replication of a number or a vector

```
rep(c(2,3,4), 3)
```

Computing basic statistics

- mean(x)
- median(x)
- sum(x)
- sd(x)
- var(x)
- cor(x,y)
- cov(x,y)
- max(x)
- min(x)

Selecting Vector Elements

- Use [] to select vector elements by their positions.
- Use negative indexes to exclude elements
- Use a vector of indexes to select multiple values
- Use a logical vector to select elements based on a condition
- Use names to access named elements

List

- List is more flexible than vector. It allows multiple types of elements, including list itself.

- A list is constructed by *list()* command.

```
Mike<-list(Name="Mike",Salary=10000,Age=43,Children=c("Tom","Lily","Alice"))
Mike
```

Matrix

- ***Matrix*** is a vector of elements arranged in two dimensions.

```
m1<-matrix(3:8,ncol=3,nrow=2)
m1
```

- It can be formed by introducing *dim()*.

```
m2<-3:8
```

```
dim(m2)<-c(3,2)
m2
```

```
dim(m2)<-c(2,3)
m2
```

- Matrix is the data structure in R corresponding to matrix in linear algebra.

Naming a matrix

```
m3<-matrix(3:8,ncol = 3,nrow = 2,byrow = TRUE)
```

```
rownames(m3)<-c("Row1","Row2")
colnames(m3)<-c("Col.1","Col.2","Col.3")
m3
```

```
m3["Row1","Col.2"]
```

Data Frame

- The most commonly used data structure in R is *data frame,* which is an extension of matrix by allowing co-existence of data of various types.
- A column in a data frame normally represents a data field.
- A row in a data frame normally corresponds to a record of the data.

Construct a data frame

```r
df<-data.frame(ID=c(1,2,3),Names=c("James","Jack","Tom"),Values=rnorm(3,mean=100,sd=10))
df
```

Explore data set

- Explore data set *mtcars*

Factor

- ***Factors*** are special variables used to store categorical variables.

- Factor is not an atomic data type and it can be either a numeric data or a character data.

Advantage of using factor

- Factor variables are stored as a vector of integer values, thus it is a more efficient use of memory. Yet, the original set of character values will be used when the factors are displayed, which will make the data presentation more meaningful.

- Many statistical models will automatically handle factor variables properly.

- Factor variables are also very useful in many different types of graphics.

Example

```
a<-c("Snake", "Dog", "Elephant", "Dog", "Cat" )

b <- factor(a)
b
as.integer(b)
```

Levels of factor by default is by alphabet order

```
b <-factor(a, levels=c("Snake", "Cat", "Dog", "Elephant"))
b

as.integer(b)
```

You can manually define the order