

Statistical Computing with R

Masters in Data Science 503 (S9)

Second Batch, SMS, TU, 2023

Shital Bhandary

Associate Professor

Statistics/Bio-statistics, Demography and Public Health Informatics

Patan Academy of Health Sciences, Lalitpur, Nepal

Faculty, Data Analysis and Decision Modeling, MBA, Pokhara University, Nepal

Faculty, FAIMER Fellowship in Health Professions Education, India/USA.

Review Preview (Unit 2, Session 3)

- **Data Mining**

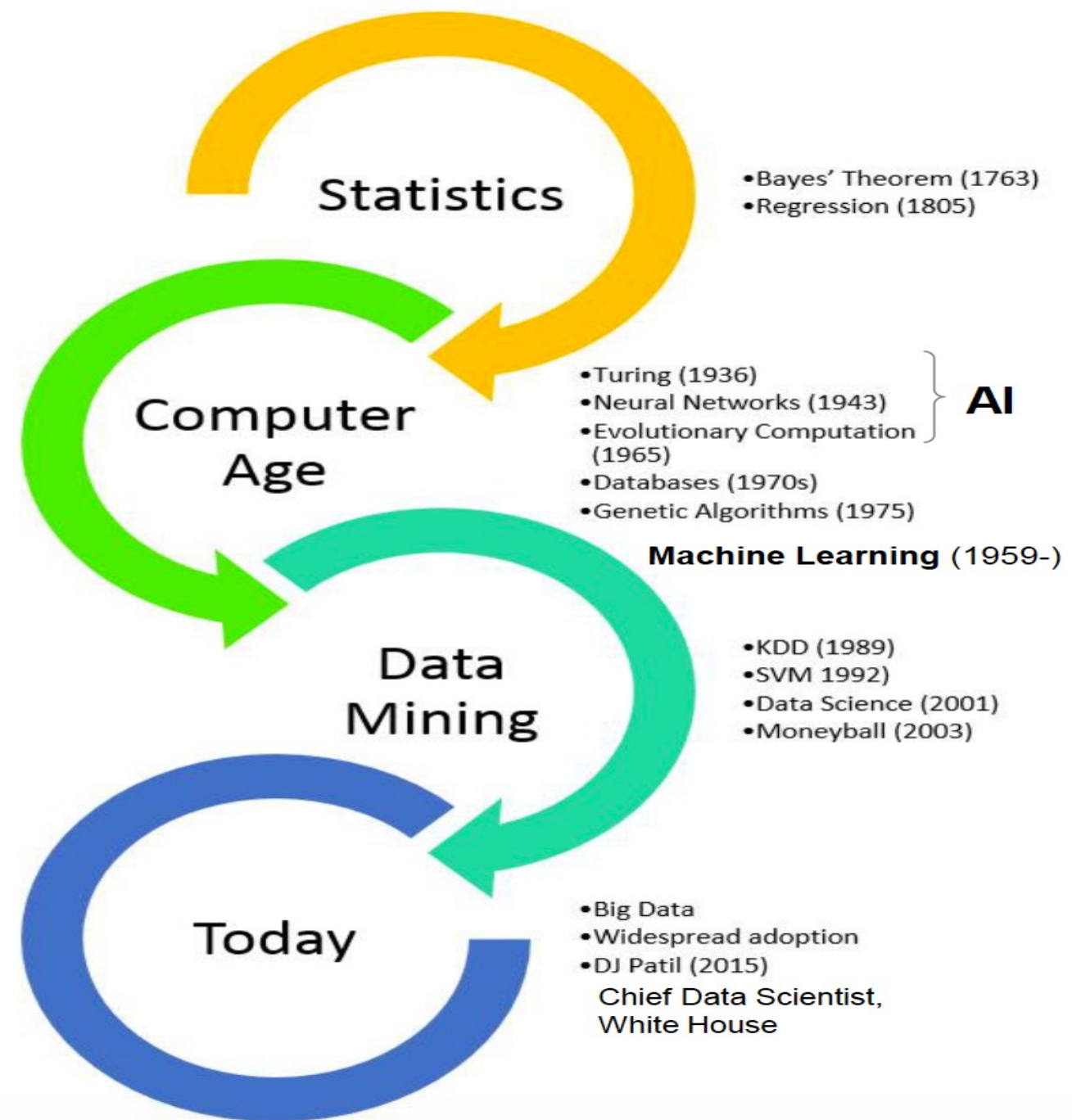
- What did you find in this reading provided to you?
- <https://bootcamp.pe.gatech.edu/blog/what-is-data-mining/>

- **Text Mining**

- What did you find in this reading provided to you?
- <https://rpubs.com/vipero7/introduction-to-text-mining-with-r>

Origins of Data Mining

- Draws ideas from AI, machine learning, pattern recognition, statistics, and database systems.
- There are differences in terms of
 - used data and
 - the goals.

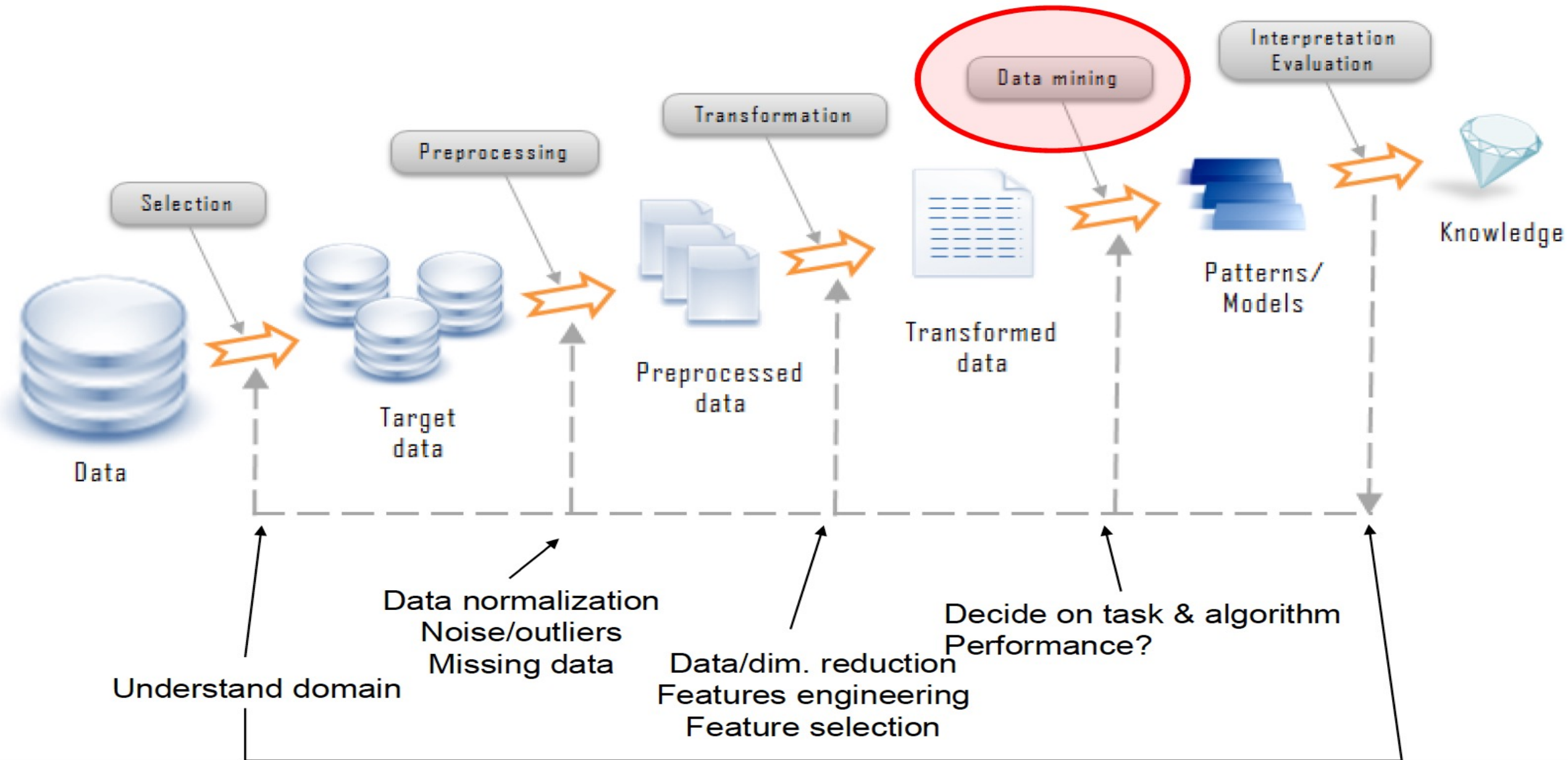


Data Mining (What is):

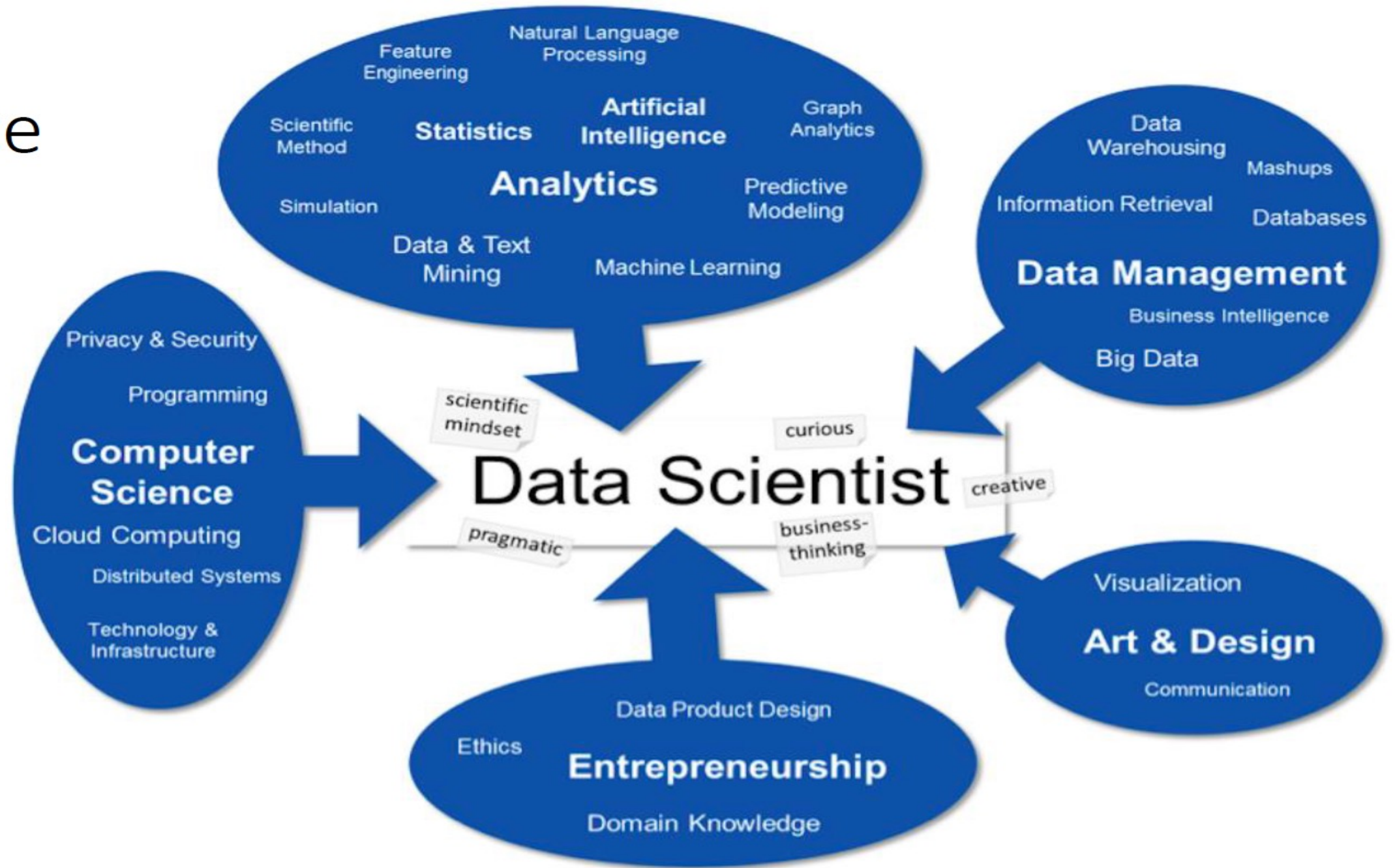
- Data Mining refers to a set of methods applicable to large and complex databases to eliminate the randomness and discover the hidden pattern.
(<https://online.stat.psu.edu/stat857/node/142/>)
- Data Mining is the science of **extracting useful information** from huge **data repositories/warehouse** (<http://www.kdd.org/curriculum>)
- Data Mining helps to:
 - identify patterns and relationships
 - classify and segment data
 - formulate hypothesis

KDD = Knowledge
Discovery in/from
Database

Knowledge Discovery in Databases (KDD) Process



Data Science



Source: T. Stadelmann, et al., Applied Data Science in Europe

For Data Science, Data Mining is:

- interdisciplinary and overlaps significantly with many fields such as
 - Statistics
 - Computer Science (Machine Learning, AI, Databases)
 - Optimization
- requires a team effort with members who have expertise in several areas such as
 - Data management
 - Statistics
 - Programming
 - Communication
 - + application domain (health, business, physics, biology etc.)

(IBM) CRISP-DM Reference Model:

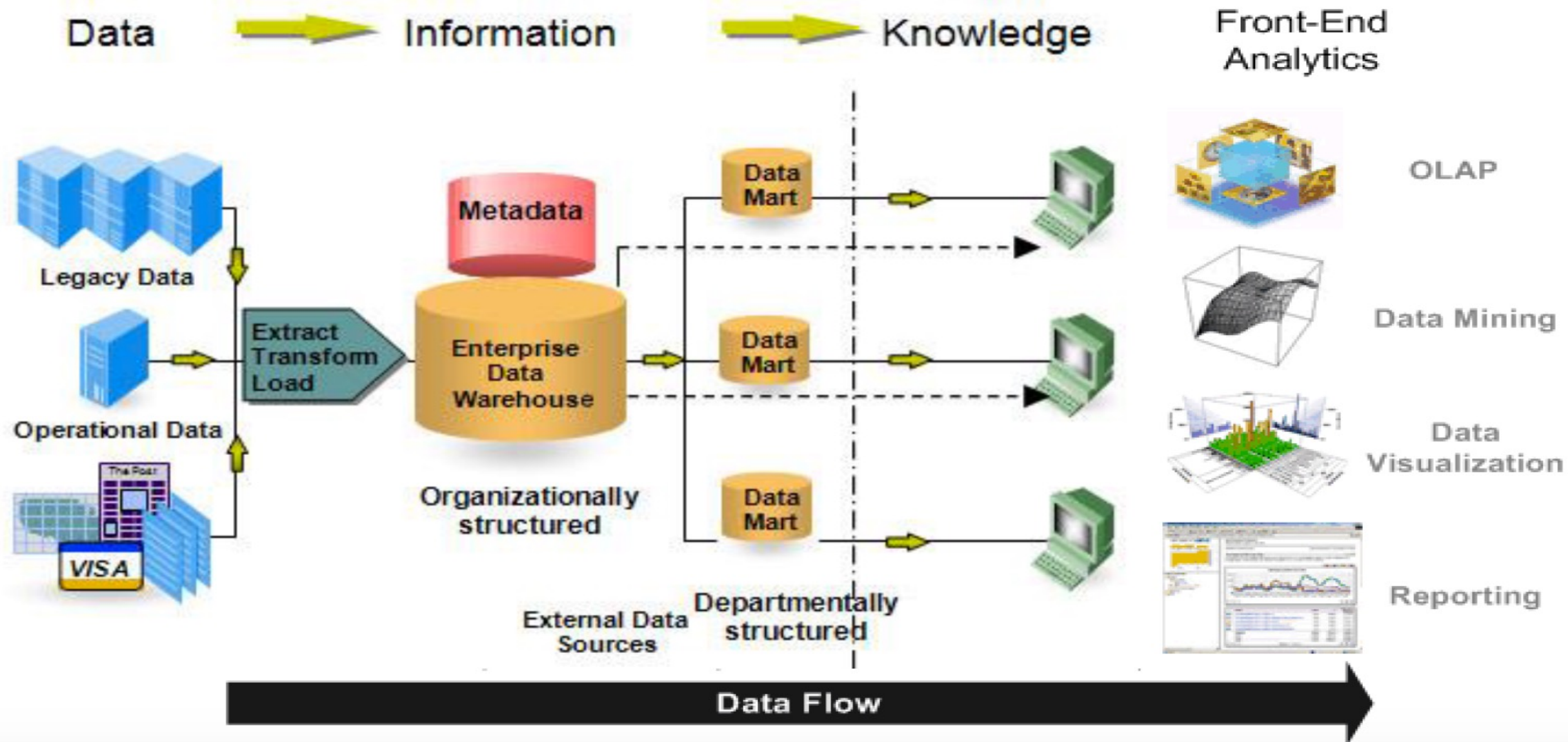
- Cross Industry Standard Process for Data Mining (CRISP-DM):
 - Business Understanding
 - Data understanding
 - Data Preparation
 - Modelling
 - Evaluation
 - Deployment

Tasks in the CRISP-DM Model

Business Understanding	Data Understanding	Data Preparation	Modeling	Evaluation	Deployment
Determine Business Objectives <i>Background</i> <i>Business Objectives</i> <i>Business Success Criteria</i>	Collect Initial Data <i>Initial Data Collection Report</i>	Select Data <i>Rationale for Inclusion/Exclusion</i>	Select Modeling Techniques <i>Modeling Technique</i> <i>Modeling Assumptions</i>	Evaluate Results <i>Assessment of Data Mining Results w.r.t. Business Success Criteria</i> <i>Approved Models</i>	Plan Deployment <i>Deployment Plan</i>
Assess Situation <i>Inventory of Resources</i> <i>Requirements, Assumptions, and Constraints</i> <i>Risks and Contingencies</i> <i>Terminology</i> <i>Costs and Benefits</i>	Describe Data <i>Data Description Report</i>	Clean Data <i>Data Cleaning Report</i>	Generate Test Design <i>Test Design</i>	Review Process <i>Review of Process</i>	Plan Monitoring and Maintenance <i>Monitoring and Maintenance Plan</i>
Determine Data Mining Goals <i>Data Mining Goals</i> <i>Data Mining Success Criteria</i>	Explore Data <i>Data Exploration Report</i>	Construct Data <i>Derived Attributes</i> <i>Generated Records</i>	Build Model <i>Parameter Settings</i> <i>Models</i> <i>Model Descriptions</i>	Determine Next Steps <i>List of Possible Actions</i> <i>Decision</i>	Produce Final Report <i>Final Report</i> <i>Final Presentation</i>
Produce Project Plan <i>Project Plan</i> <i>Initial Assessment of Tools and Techniques</i>	Verify Data Quality <i>Data Quality Report</i>	Integrate Data <i>Merged Data</i>	Assess Model <i>Model Assessment</i> <i>Revised Parameter Settings</i>		Review Project <i>Experience</i> <i>Documentation</i>
		Format Data <i>Reformatted Data</i>			
		<i>Dataset</i> <i>Dataset Description</i>			

Figure 3: Generic tasks (bold) and outputs (italic) of the CRISP-DM reference model

Data Warehouse



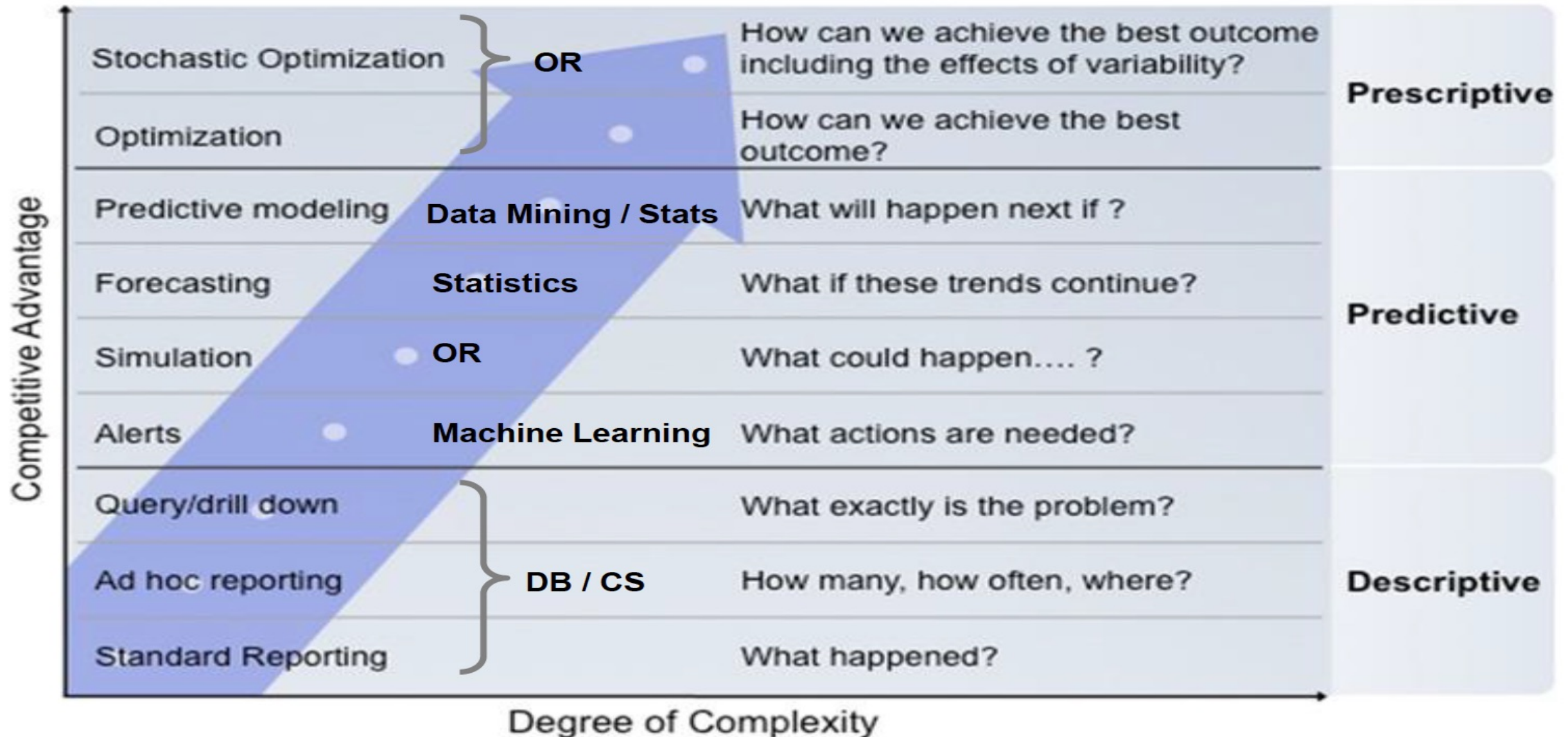
What is?

- Data:
 - Legacy data?
 - Operational data?
 - ETL process?
 - Information:
 - Metadata?
 - Enterprise Data Warehouse?
 - Data Mart?
 - Knowledge:
 - OLAP?
 - Data Mining?
- Data:
 - Old, obsolete but retained data
 - Highly volatile, real time analysis
 - Extract, transform, load
 - Information:
 - Directory
 - Integrated, static data, analytics
 - Simple form of data warehouse
 - Knowledge
 - Online analytical processing
 - Descriptive, predictive, prescriptive

Data Mining Tasks:

- Descriptive Methods:
 - Find human interpretable patterns that describes the data (Unit 1, 2 and 3 of this course)
- Predicting Methods:
 - Use some features (variables) to predict unknown or future value of other variable (Unit 4 and 5 of this course)
- Prescriptive Methods:
 - Optimization
 - Stochastic optimization

Data Mining & Analytics



Data Mining Tasks:

- **Predictive Modelling (Regression and classification) – Unit 4**
- **Dimensionality Reduction, Cluster Analysis – Unit 5**
- **Association Analysis – Unit 5**
- **Anomaly detection – Not covered in this course**

Predictive modelling:

- **Supervised Learning**
- Regression
 - Linear regression (simple and multiple)
 - Logistic regression (bi-variate and multivariate)
- Classification
 - Decision trees, Random forests, Neural networks
 - Support Vector Machines, Naïve Bayes
- **We will discuss more on these topics in Unit 4**

Dimensionality Reduction: Column/Variable Based Data Reduction Methods

- **Unsupervised Learning**
- Principal Component Analysis (PCA)
- Principal Axis Factoring (PAF)
- Multidimensional scaling (MDS)
 - Classical (Principal coordinate analysis)
 - Metric MDS, Non-metric MDS
 - Generalized MDS

Cluster Analysis: Row/Case Based Data Reduction Methods (HC, k-means etc.)

- **Unsupervised Learning**
- Data points in one cluster are more similar to one another
- Data points in separate clusters are less similar to one another
- **We will discuss more on dimension reductions/cluster analysis in Unit 5**

Data Mining Tools:

- Simple Graphical User Interface (GUI) based on R
 - Weka
 - Rattle
- Process oriented
 - Rapid Miner
 - IBM SPSS Modeler
 - SAS Enterprise Miner etc.
- Programming oriented
 - R, Rattle, R Studio (shiny), Microsoft R (reticulate package to run python in R)
 - Python, Numpy, Scipy scikit-learn, pandas, Jupyter notebook (rpy2 to run R in python)

Other Data Mining Tasks:

- **Text Mining (we will discuss it today with an example from web)**
- **Graph Mining (Unit 3)**
- Data stream mining – not covered in this course
- Mining spatiotemporal data (e.g. moving objects) – not covered
- Distributed data mining etc. – not covered in this course

Question/queries so far?

Text Mining:

- Import texts (Interviews, Twits, Facebook posts, Comments, Reviews etc.) in R
- Transform the texts to data frame and define the “Corpus”
- Perform pre-processing of the “Corpus” using standard methods
- Build document-term matrix (DTM)
- Find frequent terms and associations of key term with other terms
- Use network graph/word cloud to visualize the DTM
- Perform cluster analysis to find clusters of similar words
- Perform “topic modelling” and compare it with network graph result!

Packages required for Text Mining:

- Text mining: *tm*
(Details: <https://cran.r-project.org/web/packages/tm/tm.pdf>)
- Topic modelling: *topicmodels*, *lda*
- Word cloud: *wordcloud*
- Twitter data access: *twitteR* (Optional)

Example of tweet mining: rdatamining.com

(Alternative solution: https://rstudio-pubs-static.s3.amazonaws.com/66739_c4422a1761bd4ee0b0bb8821d7780e12.html)

Option 1: retrieve tweets from Twitter (must have API keys!)

- library(twitteR)
- tweets <- userTimeline("RDataMining", n = 3200)

Option 2: download @RDataMining tweets from RDataMining.com

- url <- "http://www.rdatamining.com/datasets/rdmTweets.RData"
download.file(url, destfile = "./rdmTweets.RData")

Option 3: Download @RDataMining tweets from RDataMining.com manually: <http://www.rdatamining.com/datasets/rdmTweets.RData>
and save it to the folder you want to use e.g. Downloads!

Load tweets in R, check length and its structure
(The “twitterR” package must be installed *a priori*):

- `load(file = "./rdmTweets.RData")` #Option 2 used!
- `tweets <- rdmTweets` #Assign tweets as rdmTweets
- `str(tweets)`

- (n.tweet <- length(tweets)) #If rmdTweets is assigned as “tweets”
- `[1] 154` #Option 2 used, 154 tweets only!

- strwrap(tweets[[154]]\$text, width = 55) #Text variable of tweet 154
- `[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."`

Checking the content of first three tweets

- **tweets[1:3]**
- **[[1]]**
- [1] "RDataMining: Postdoc/Research Scientist Position on Big Data at MIT
<http://t.co/hZ1ojAW2>"
- **[[2]]**
- [1] "RDataMining: Research scientist position for privacy-preserving data
publishing, Singapore <http://t.co/GPA0TyG5>"
- **[[3]]**
- [1] "RDataMining: Easier Parallel Computing in R with snowfall and sfCluster
<http://t.co/BPcinvzK>"

Checking the content of the last tweet

#With string wrap and line break at 55th and 62nd positions

- `strwrap(tweets[[154]]$text, width = 55)`
- `strwrap(tweets[[154]]$text, width = 62)`

#What happens if a single square bracket is used?

`strwrap(tweets[154]$text, width = 55)`

Output: ?

Why?

Text cleaning in R: Pre-processing I (tweets to data frame and text corpus formation)

- **library(twitteR)**

convert tweets to a data frame

- `df <- twListToDF(tweets)`

- `str(df)`

library(tm)

build a corpus

- `myCorpus <- Corpus(VectorSource(df$text))`

#Inspect first 3 elements

`inspect(myCorpus[1:3])`

Text cleaning in R: Pre-processing I (Corpus to lower case, remove punctuation/numbers)

convert to lower case

- `myCorpus <- tm_map(myCorpus, tolower)`
- `inspect(myCorpus[1:3])`

remove punctuations and numbers

- `myCorpus <- tm_map(myCorpus, removePunctuation)`
- `inspect(myCorpus[1:3])`
- `myCorpus <- tm_map(myCorpus, removeNumbers)`
- `inspect(myCorpus[1:3])`

Text cleaning in R: Pre-processing II (Remove URL)

remove URLs, http followed by non-space characters

- `removeURL <- function(x) gsub("http[^:space:]*", "", x)`
- `myCorpus <- tm_map(myCorpus, removeURL)`
- `inspect(myCorpus[1:3])`

Text cleaning in R: Pre-processing II (Remove Stop Words)

remove r and big from the list of stopwords

- `myStopwords <- setdiff(stopwords("english"), c("r", "big"))`

remove stopwords

- `myCorpus <- tm_map(myCorpus, removeWords, myStopwords)`
- `inspect(myCorpus[1:3])`

Text cleaning in R: Pre-processing III

(Stemming, **be careful with this process!**)

keep a copy of corpus

- **myCorpusCopy <- myCorpus**

stem words

- **myCorpus <- tm_map(myCorpus, stemDocument)**
- **inspect(myCorpus[1:3])**

stem completion (OPTIONAL:THIS MIGHT NOT WORK!)

- **myCorpus <- tm_map(myCorpus, stemCompletion, dictionary = myCorpusCopy)**
- **inspect(myCorpus[1:3])**

We must use
SnowballC package
for proper
stemming if this
does not work!

Text cleaning in R: Pre-processing III

(Stemming, **be careful with this process!**)

replace "posit" with "position", because "position" was first stemmed to "posit" and then completed to "posit"

- **myCorpus <- tm_map(myCorpus, gsub, pattern="posit", replacement="position")**
- **strwrap(myCorpus[154], width=55)** #check the corpus again (iteratively)!

[1] "r reference card data mining now available cran list mani use r"

[2] "functions packag data mining applic"

Create Term Document Matrix and Check “Frequent terms”:

#Create Term Document Matrix and check its structure

- `myTdm <- TermDocumentMatrix(myCorpus, control=list(wordLengths=c(1,Inf)))`
- `str(myTdm)`

inspect frequent words

`(freq.terms <- findFreqTerms(myTdm, lowfreq=10))`

- [1] "data" "posit" "research" "comput" "r" "packag" "tutori"
- [8] "use" "slide" "mine" "analysi" "network" "social" "introduc"
- [15] "exempl" "group"

Check “Associations” with word “r”:
Association ≥ 0.2 of “r” with other words!

- # which words are associated with r?

```
findAssocs(myTdm, "r", 0.2)
```

```
##           r
# mani      0.26   #function  0.24
# canberra  0.26   #use       0.23
# list      0.26
```

What is done here?

(This will not work if stemming is not corrected!)

which words are associated with

- `findAssocs(myTdm, "position", 0.25)`

`$position`

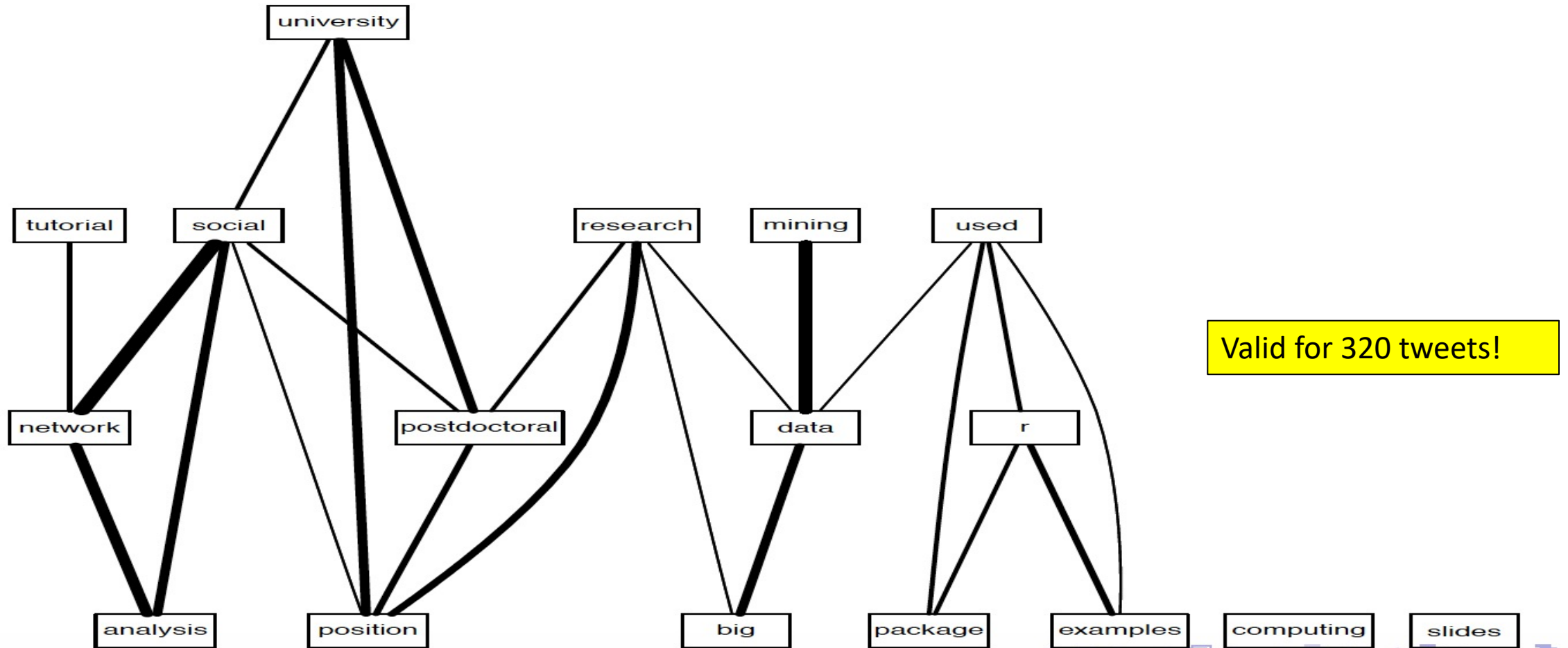
`numeric(0)`

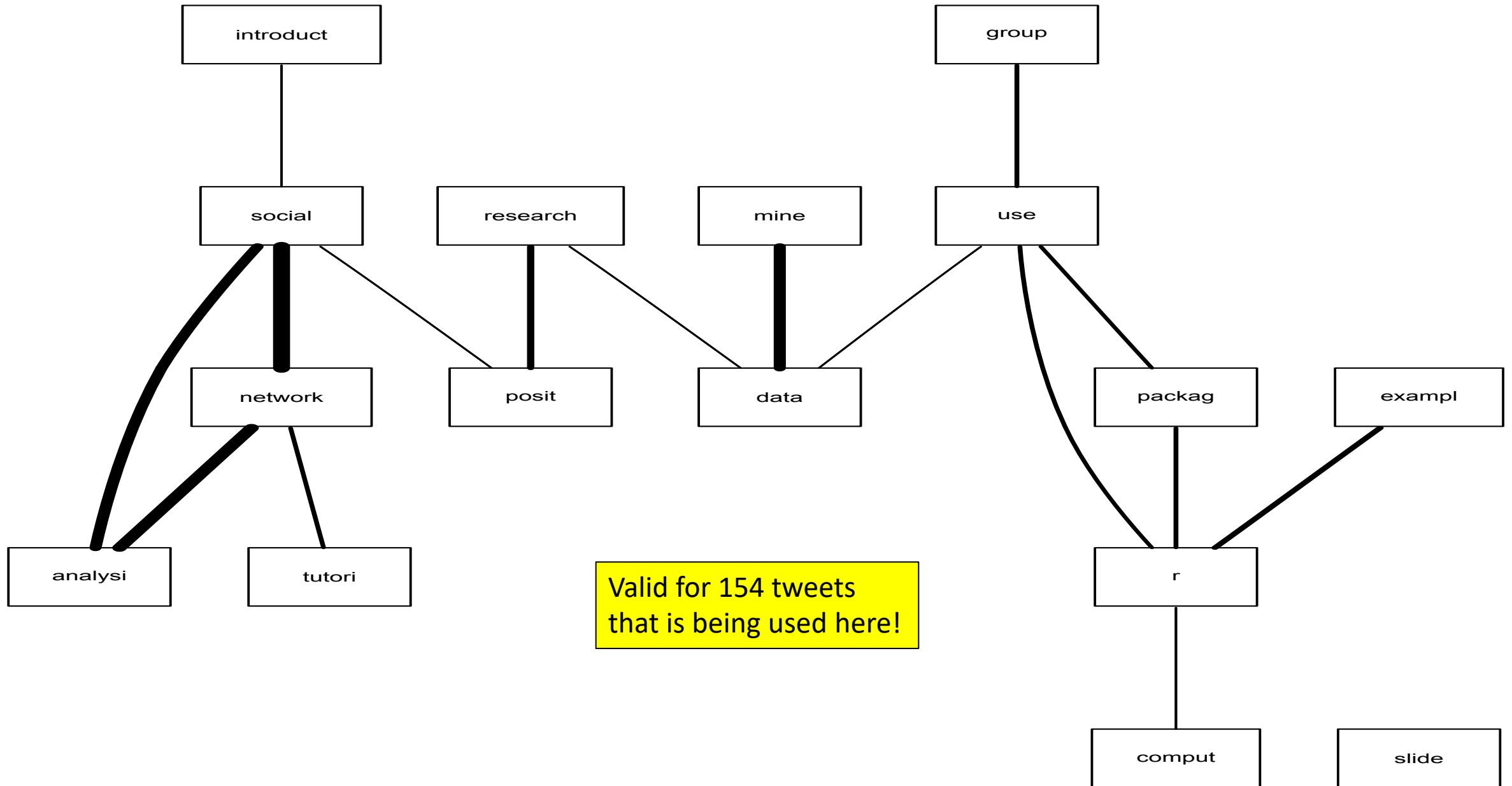
which words are associated with

- `findAssocs(myTdm, "posit", 0.25)`

Network of Terms

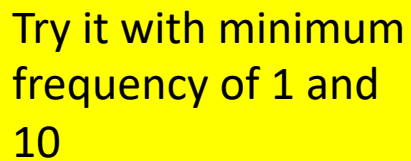
```
library(graph)
library(Rgraphviz)
plot(myTdm, term=freq.terms, corThreshold=0.1, weighting=T)
```





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)`



If you want colorful word cloud then use the RColorBrewer package!

More here:

<https://www.r-bloggers.com/2011/07/word-cloud-in-r/>

Topic Modelling: “topicmodels” package

- `library(topicmodels)`
- `set.seed(123)`
- `myLda <- LDA(as.DocumentTermMatrix(myTdm), k=5)` #5 topics
- `terms(myLda, 3)` #Three terms in each topic (can be changed)

Note: LDA = Latent Dirichlet Allocation: NLP->ML->AI (Self-learning)

- | | Topic 1 | Topic 2 | Topic 3 | Topic 4 | Topic 5 |
|--------|---------|----------|-----------|----------|---------|
| • [1,] | "data" | "r" | "analysi" | "r" | "r" |
| • [2,] | "mine" | "exampl" | "network" | "packag" | "data" |
| • [3,] | "r" | "code" | "social" | "comput" | "mine" |

- Compare this result with the Rgraphviz results obtained above
- Are you happy?
- Do you want to change LDA parameters?
- Change LDA parameters!

Refined model: Four topics with 3 terms each

- `set.seed(123)`
- `> myLda <- LDA(as.DocumentTermMatrix(myTdm), k=4)#5 topics`
- `> terms(myLda, 3)` `#Three terms in each topic (can be changed)`

	Topic 1	Topic 2	Topic 3	Topic 4
• [1,]	"data"	"data"	"r"	"r"
• [2,]	"mine"	"research"	"tutori"	"group"
• [3,]	"r"	"r"	"exampl"	"data"

Are you happy with these four topics?

Did it miss something important?

Project 1: Text mining with five pdf files of same content area (journal articles)!

- You must search and download first 5 free pdf files on this topic using **Google Scholar** (<https://scholar.google.com/>) (DO NOT USE GOOGLE.COM)
- You must put all the 5 pdf files in a folder called “MDS503P1”
- Use the “pdftools” package to read these five pdf files in R
- Once you read them in R, create a “corpus” and perform text mining
- Submit the R Script file, R markdown file and knitted pdf files for Project 1 work

Question/Queries?

Thank you!

@shitalbhandary