クラウドからビックデータへ
-データサイエンティストのすすめ-

大阪大学准教授　鈴木譲

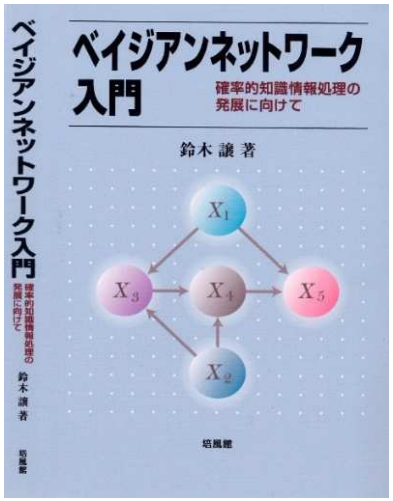
2015年6月12日

SAMURAI CLOUD
POWERED BY NIPPON CLOUD WORKING GROUP

R

鈴木譲(すずきじょう, Joe Suzuki)

- 大阪大学准教授 (情報数理、統計数理、機械学習)
- 日本のBayesianネットワーク研究の第１人者



ロードマップ

1. データサインエンスの進展とそれをとりまく背景
2. R言語の概要
3. 2015年前期で、大阪大学で教えている講義の紹介
4. Rパッケージ twitterR によるキーワード解析 (実行例)
5. Rパッケージ Rmecabによる形態素解析 (実行例)
6. データサイエンスでどのようなビジネスがあるのか
7. 今後の課題

大学の先生の講演というよりは、
オープンソースカンファレンスのLTに近い(デモばかり)

ビックデータ: ただのデータ解析、されど…

- 統計理論、機械学習技術の進展
- インターネットなど、一般IT技術の進展
- R言語の普及

データサイエンス == データから、法則性を見出す営み

- 医者など、科学者のほとんどがデータサイエンティスト
- データから因果関係・法則性を見出し、ビジネス上の判断

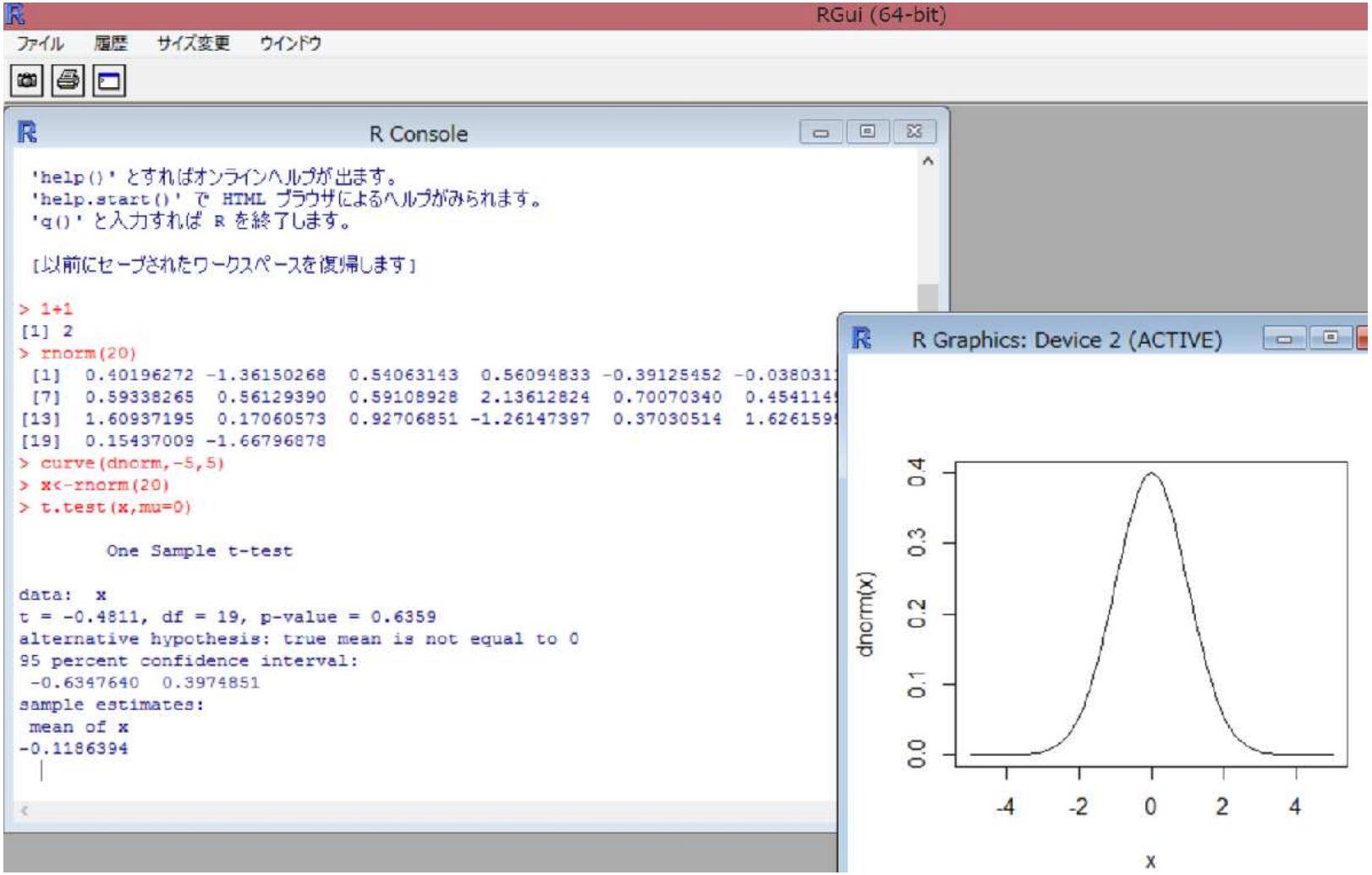
サイエンス = データサイエンス
サイエンティスト = データサイエンティスト

データサイエンティスト

項目	概要
魅力	身近なデータから、自分が初めて法則性を見出す喜び 事実にもとづいて判断する姿勢が強化される 「自分にしかできない」という、専門職としての満足感 転職や新規事業のチャンスになる
応用	ビジネス: 　経営分析、マーケッティング分析、金融(為替、株価予測) 科学: 　ゲノム解析、医学全般、心理学、経済学
必要なスキル	分析力・洞察力 開発力 ビジネス力 知識獲得力

できる人でないと、一人前にはなれないかもしれない

R言語: 統計を扱うための専用の言語

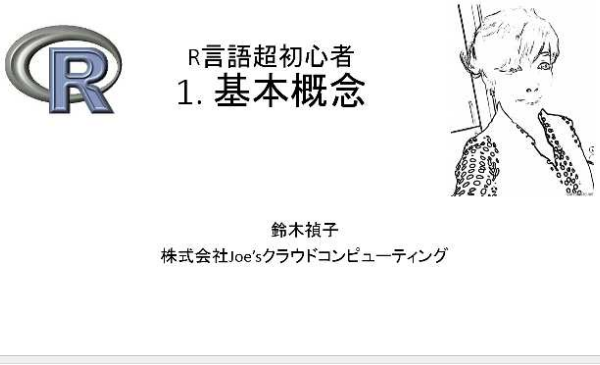


なぜ、Excelではなく、R言語なのか

	Excel	R言語
学習するためのコスト	ゼロ	完全に理解するのに1年かかるといわれる
購入コスト	有償 (アドインソフトの費用も)	無料 (オープンソース)
環境	Microsoft Mac	環境を選ばない
マニュアル	購入したものには用意されている	ネット検索でおびただしい数の情報が得られる
プログラミング	用意されているマクロを組み合わせる	細かいプログラミングができる
グラフィック	解像度に限界がある	美しいといわれる
開発部隊	マイクロソフトでしか開発されていない	オープンソースなので、ニーズの高いパッケージ(無料)が毎日アップロードされている。したがって、工数(プログラムステップ数)が少ない。
手作業量	手作業が多い	関数が多数あるので、データ処理が自動化される
主なユーザ層	一般ビジネスマン	大学、研究者、データサイエンティスト

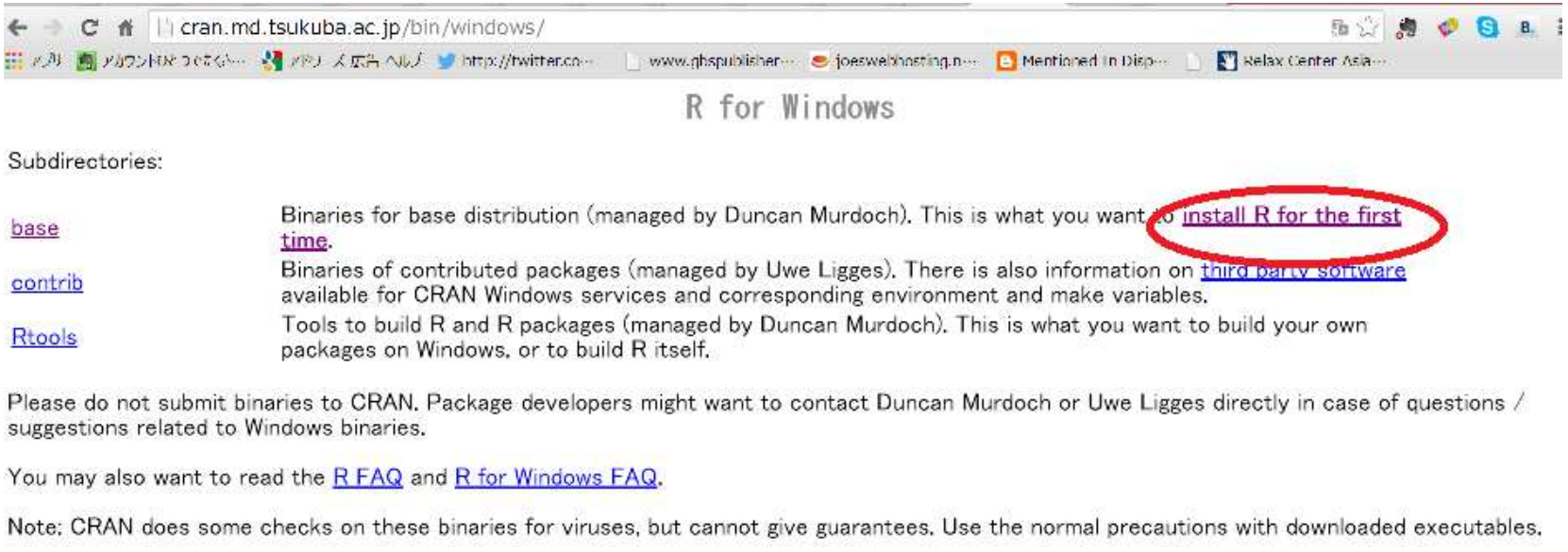
R言語の学習は、難しくない

- ベクトル同士の演算が簡単にできる。
- 統計関係の関数、パッケージが多数そろっているので、統計をブラックボックスとみてよい。
- プログラミング言語というより、データを入力したら解析結果を返す、「そろばん」のようなイメージが強い。
- ノートPCにいれておいて、いつでも触れられるようにしておく。
- ネット検索が基本: 疑問点が頭をさえぎったら、「R (関数名)」のように



Rのインストール: 素人でも1-2分

「Rインストール」で検索して、ダウンロードして、Yesばかり



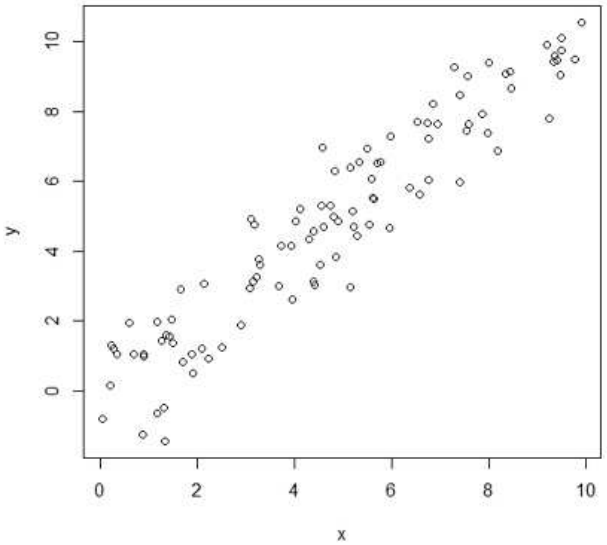
データ解析に必要な統計学

統計学の分野	例
可視化	散布図、ヒストグラム、箱ひげ図、円グラフ
推定	回帰分析 (直線へのあてはめ)
検定	正規性の検定 (データが正規分布にしたがって、発生したか) 平均値の検定 (平均値がある値であったか) 平均値の差の検定 (2変量の平均値の差が0であったか) カイ2乗検定 (カテゴリー間の相関性) 相関性の検定 (2変量の相関係数が0であったか)

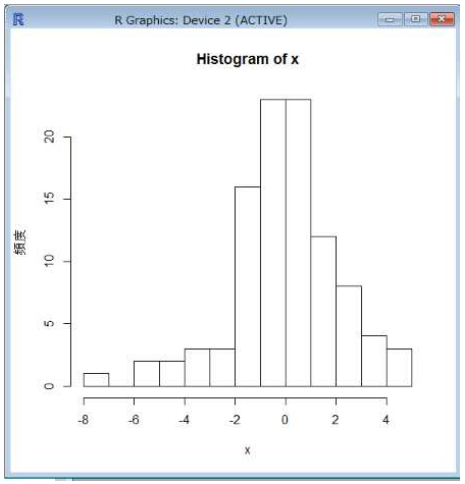
分布は未知で、データのみが存在。データと事前情報から結論を導く

可視化

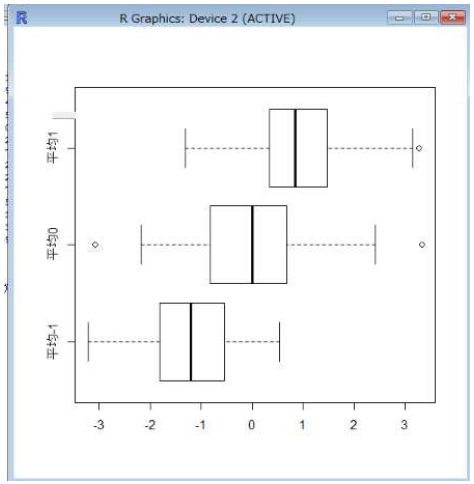
散布図



ヒストグラム



箱ひげ図



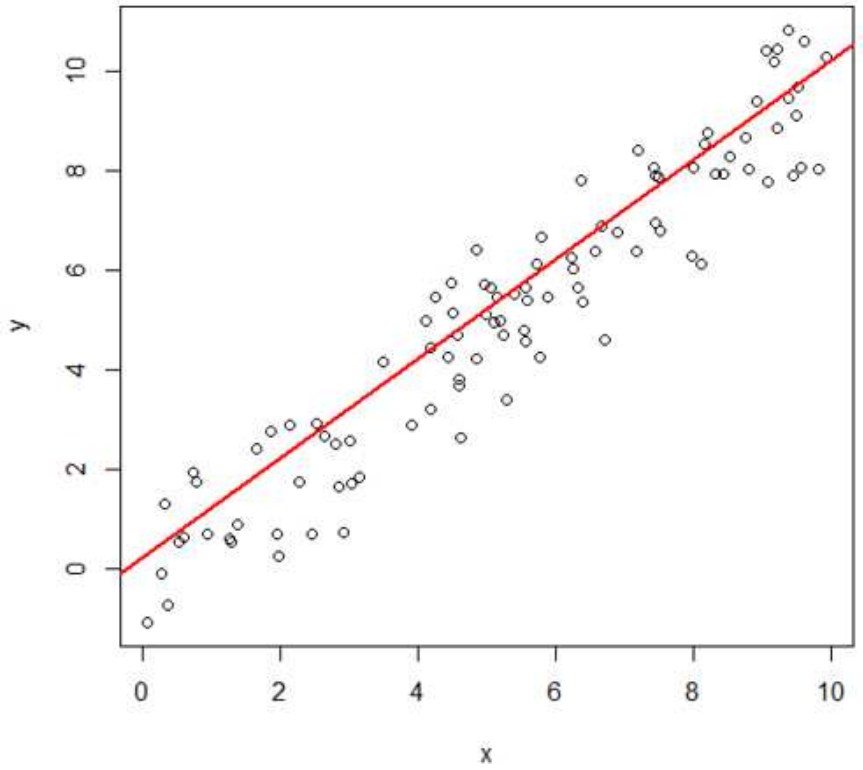
円グラフ

回帰分析

$(x_1, y_1), \cdots, (x_n, y_n)$ から

$Y = aX + b$ なる a, b を求める

$$\sum_{i=1}^{n} (ax_i + b - y_i)^2 \to min$$



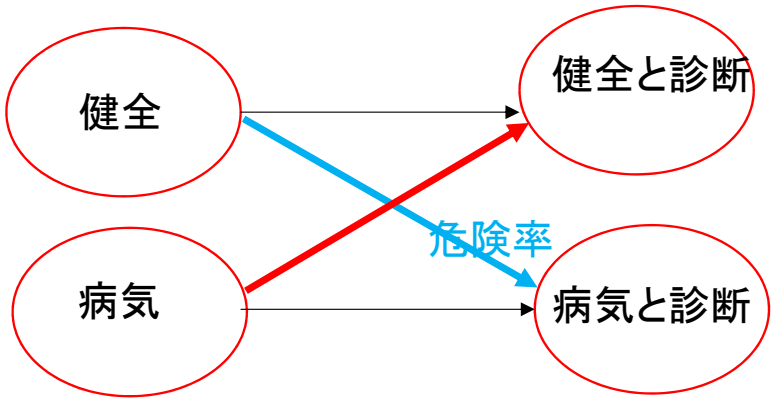
統計的検定

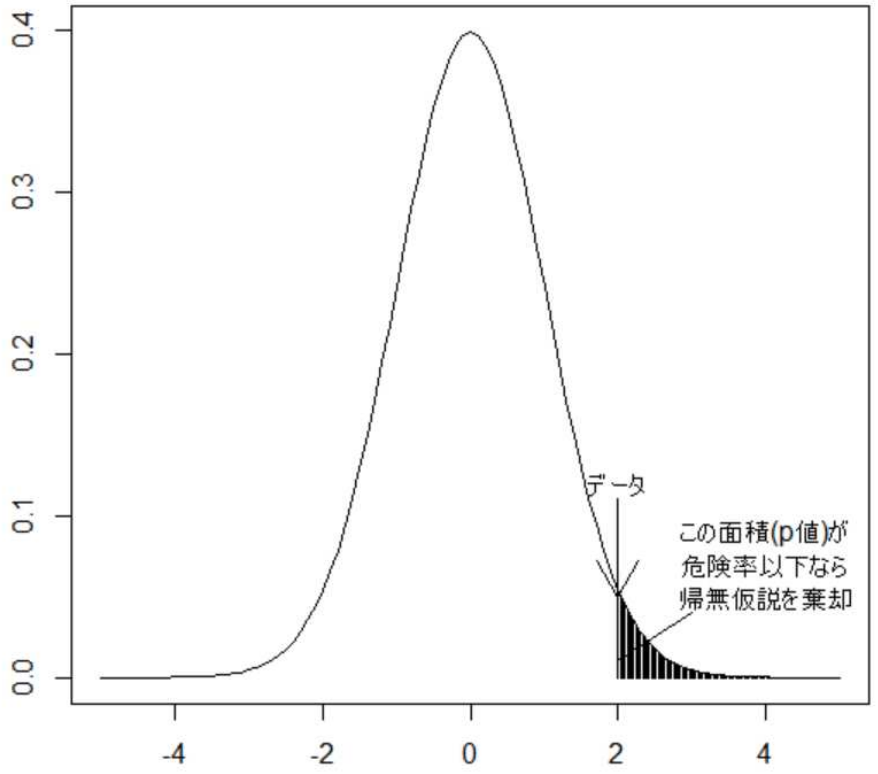
帰無仮説: データがある分布にしたがう。

対立仮説: 帰無仮説は正しくない

帰無仮説の分布を仮定したとき、

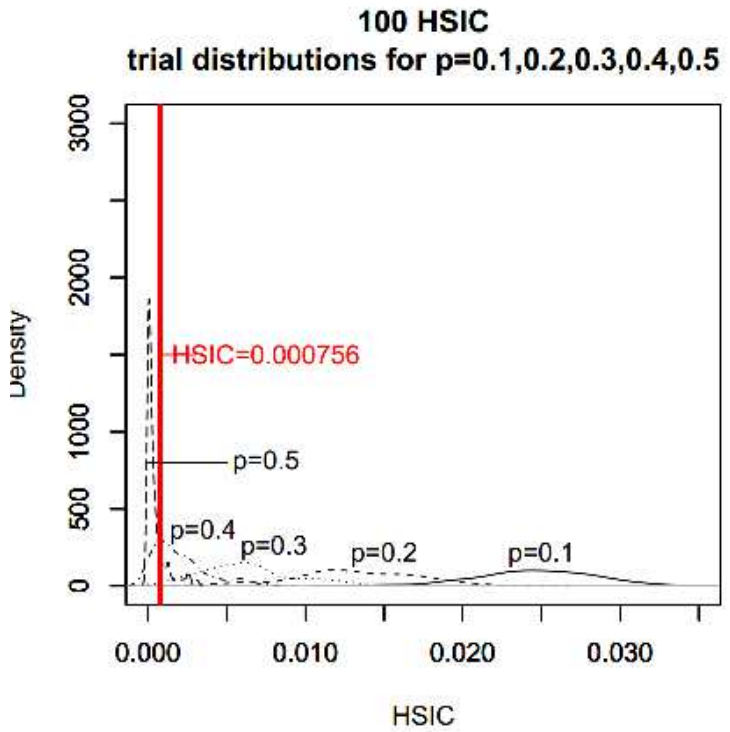
1. そのようなデータが生じる確率が危険率 (0.05 など) 未満のとき、危険率で、その帰無仮説を棄却
2. それ以外では、帰無仮説を棄却できない





危険率を一定以内に抑えながら、
帰無仮説を棄却すべきなのに、棄却しない確率を最小にする

2014年度後期 留学生向け(Experimental Mathematics)
http://math.lms.ac



thank you for the hints and also for the last party.
We enjoyed it allot.
Furthermore we are very sorry for not attending the last lecture but we thought classes ended the week before.
If you are in München again (for the Octoberfest or any other reason) just write me, and I will invite you to a party in München.

Dear Prof. Suzuki

Thank you for this course and for all the help with experiencing Japanese culture.
I really appreciate it a lot.
Thank you!

Best regards

2015年前期(阪大数学科3年): 金融工学のビックデータ的アプローチ

1. ガイダンス: 金融時系列の事例
2. R言語(1): 基本概念とベクトル
3. R言語(2): 行列・配列・リスト
4. R言語(3): 関数とプログラミング
5. R言語(4): データフレーム
6. R言語(5): グラフィックス
7. 回帰分析
8. ARMA過程(1)
9. ARMA過程(2)
10. 回帰分析、ARMA過程へのAICの適用
11. ARCH過程とGARCH過程
12. VARモデル
13. 単位根過程
14. 金融データ処理(1): 為替レートの予測
15. 金融データ処理(2): 株価の予測

無限の可能性: Rによるビックデータの応用

業務	テーマ
講義	金融工学 (FX, 株価予測)
研究	遺伝子ネットワーク (ゲノム解析)
研究指導(学生他)	Twitterによるのテキスト解析

- 転職のマッチング
- リスティング
- ABテスト

データ解析の求人が、格段に増えている

Rからtwitterのデータを読込む

```
Install.packages(“twitteR”)
library(twitteR)

api_key <- "KaOM8Edi7x0oECtL???????"
api_secret <-"lrcwL4irmkOKixQoF0aIv3wYKF??????????"
access_token <- "47486543-JZlQOBx8nrzVRebiWYwTe??????????"
access_token_secret <- "B17VHMqEWFLcgQdOvkoI6????????????????"

setup_twitter_oauth(api_key,api_secret,access_token,access_token_secret)
searchTwitter("iphone")
```

```
#データフレームの整形
wday.abb <- c("月", "火", "水", "木", "金", "土", "日")
statusDF <- within(statusDF, {
    attr(created, "tzone") <- "Asia/Tokyo";
    statusSource <- factor(gsub("<a .*?>(.*?)</a>","¥¥1",statusSource));
    date <- factor(format(created, "%Y 年 %m 月 %d 日"));
    hour <- NULL; month <- NULL; year <- NULL; wday <- NULL;
    with(as.POSIXlt(created), { hour <<- factor(hour); month <<- factor(mon + 1); year <<- factor(year + 1900);; wday <<- factor((wday +
6) %% 7, labels = wday.abb)}
)
    textLength <- nchar(text)
    text.1 <- gsub("¥¥B[#@]¥¥S+¥¥b", "", text);
    text.1 <- gsub("^¥¥s+|¥¥s+$","",text.1);text.1 <- gsub("[ ¥t]{2,}","",text.1);
    text.1 <- gsub("(RT|via)((?:¥¥b¥¥W*@¥¥w+)+)", "", text.1);
    text.1 = gsub("[[:punct:]]", "", text.1);  text.1 = gsub("[[:digit:]]", "", text.1)
    cleanText <- gsub("http¥¥w+", "", text.1)
    cleanTextLength <- nchar(cleanText)
})
topSources <- names(head(sort(table(statusDF$statusSource),decreasing = TRUE), 5));
statusDF <- within(statusDF, {statusSource <- as.character(statusSource); statusSource[!statusSource %in% topSources] <- "other"
    statusSource <- factor(statusSource, levels = names(sort(table(statusSource), decreasing = TRUE)))
})
```

```
#統計情報の算出
#install.packages("reshape2");
library(reshape2)
acast(melt(statusDF, id.vars = c("statusSource", "wday", "month", "hour"),measure.vars = c("textLength")),month + statusSource ~ wday, mean,
subset = .(statusSource %in% c("Foursquare", "Twitter for iPhone") & month %in% 9:11 & hour %in% 12:23 & wday %in% c("月", "土", "日")))
mstatus <- melt(statusDF,id.vars = c("statusSource", "wday", "year", "month", "hour", "date"),measure.vars = c("textLength", "cleanTextLength"))
mstatus[857:862, ]
args(acast); args(dcast); acast(mstatus, . ~ wday, length, subset = .(variable == "textLength"))
acast(mstatus, hour ~ wday, length, subset = .(variable == "textLength"))
acast(mstatus, hour ~ wday + month, length, subset = .(variable == "textLength"))
acast(mstatus, hour ~ wday ~ month, length, subset = .(variable == "textLength"))
dcast(mstatus, statusSource ~ ., function(x) list(c(mean = mean(x), sd =sd(x))), fill = list(c(mean = NaN, sd = NA)))
pc <- unlist(subset(statusDF, grepl("Web", statusSource), textLength)); sp <- unlist(subset(statusDF, grepl("(iPhone|iPad|Foursquare)",statusSource),textLength)); t.test(sp, pc, var.equal = FALSE)
extractScreenNames <- function(text) {
    screenNames <- gsub("(?:(@)(¥¥w+)|[¥¥s¥¥S])", "¥¥1¥¥2", text, perl = TRUE)
    unique(unlist(strsplit(substring(screenNames, 2), "@")))
}
screenNames <- unlist(lapply(statusDF$text, extractScreenNames)); head(sort(table(screenNames), decreasing = TRUE), 10)
```

```
# RMeCabを用いた日本語の単語解析
#http://taku910.github.io/mecab/
#install.packages ("RMeCab", repos = "http://rmecab.jp/R")
library(RMeCab)
(docDF(data.frame(“わたしはかもめわたしはかもめ"), column = 1, type = 1))
#http://www.lr.pi.titech.ac.jp/~takamura/pubs/pn_ja.dic
pndic <- read.table("http://www.lr.pi.titech.ac.jp/~takamura/pubs/pn_ja.dic", sep = ":",col.names = c("term", "kana", "pos", "value"),colClasses = c("character", "character", "factor","numeric"), fileEncoding = "Shift_JIS")
pndic2 <- aggregate(value ~ term + pos, pndic, mean); pos <- unique(pndic2$pos)
tweetDF <- docDF(statusDF, column = "cleanText", type = 1, pos = pos); tweetDF[900:904, 1:5]
tweetDF <- subset(tweetDF, TERM %in% pndic2$term); tweetDF <- merge(tweetDF, pndic2, by.x=c("TERM", "POS1"), by.y=c("term", "pos"))
score <- colSums(tweetDF[4:(ncol(tweetDF) - 1)] * tweetDF$value)
sum(score < 0); sum(score > 0); sum(score == 0)
table(ifelse(pndic$value > 0, "positive",ifelse(pndic$value == 0, "neutral", "negative")))
m <- mean(score);
tweetType <- factor(ifelse(score > m, "positive", ifelse(score == m, "neutral", "negative")),levels = c("positive", "neutral", "negative"));
table(tweetType); statusDF$tweetType <- droplevels(tweetType);
qplot(month, data = statusDF, geom = "bar", fill = tweetType, position = "fill")
```

テキストマイニングのまとめ

Rmecab (日本語で扱える)

- クラスタリング
- トピック分析

Python vs R

- Rでは、遅すぎる、バグが多すぎるので、お客様に納品は難しい。
- Deep Learningなど、細かい処理を書くのは、Rではむずかしい。
- Rは、プロトタイプを構築するためのものと思えば良い
- Rは、プログラムが余り組めなくても、データ解析できる (しきいが低い)
- 機械学習の研究では、まだRが主流

まとめ

- ビックデータビジネスの可能性をビジュアライズしてみては
- 地図をみているよりは、歩き始めてみては
  - 家計簿や体重の推移など、身近なところのデータ解析
  - Rをインストールして、1+1を実行してみては
  (話を聞くだけより、ずっと意味がある)