# データ解析・AIを使ってみよう

トルヴェ　アントワン

trouve@aibod.com
(株)AIBOD CTO

2016/10/26

## (株)AIBODの紹介

・住所：福岡市中央区大名

・活用：

　・解析案件を受ける

　・カスタム解析ソリューションを実装

　・会社向けセミナー（３日間AI合宿）

興味のある企業は
ご連絡ください


## 自己紹介

**名字：**トルヴェ（Trouvé）**名前：**アントワン（Antoine）

**出身：**ポワチエ（フランス）

**大学：**ボルドー大学（修士）・九州大学（博士）

**職歴：**九州大学助教（旧）・株式会社AIBOD CTO（現）

**講義**

（１時間）

＋

**ハンズオン**

（２時間半）

# 概要

① データ解析・AIをビジネスに活用する

- いい活用のためのヒント

② データ解析の基本

- データ解析のフロー
- 用語：モデル、説明変数・・・
- 基本的なアルゴリズムのイメージ

③ Rについて

- 機械学習用ツールについて
- Rとは

# ① データ解析・AI

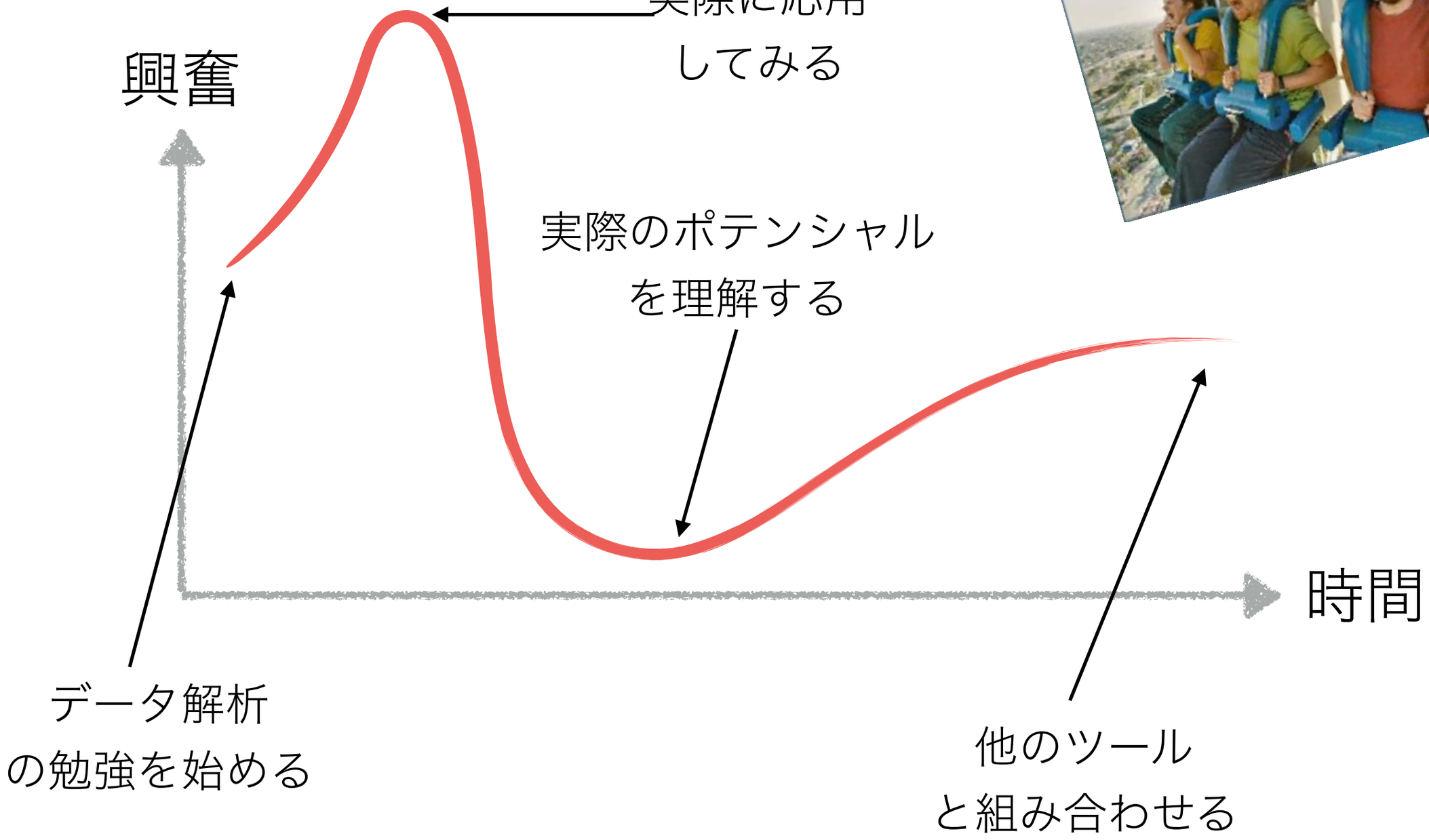
# をビジネスに活用する

# 応用事例

- マーケティング
  - LTV(リピーターなど)予測
  - 客の解約確率予測
  - マーケットセグメント分析
  - キャンペーンターゲッティング
- セールス
  - 売買成功確率予測
  - 売り上げ予測
- サプライ管理
  - 焦点ニーズ予測
- リスク管理
  - 不正行動検出
- 人事
  - ターンオーバー
  - 履歴書分類
  - 研修推薦（人材育成）

Daniel D. Gutierrez、”Machine Learning and Data Science: An Introduction to Statistical Learning Methods with R“, Technics Publication, 2015より

# データ解析のジェットコスター



# ジェットコースター

# 原因①

**魔法ではない**

アルゴリズムが魔法のように
答えを見つけてくれる

# データ解析は簡単にみえる

**未経験者の想像**

# データ解析は簡単にみえる

**未経験者の想像**


**実際は**

# ジェットコースター

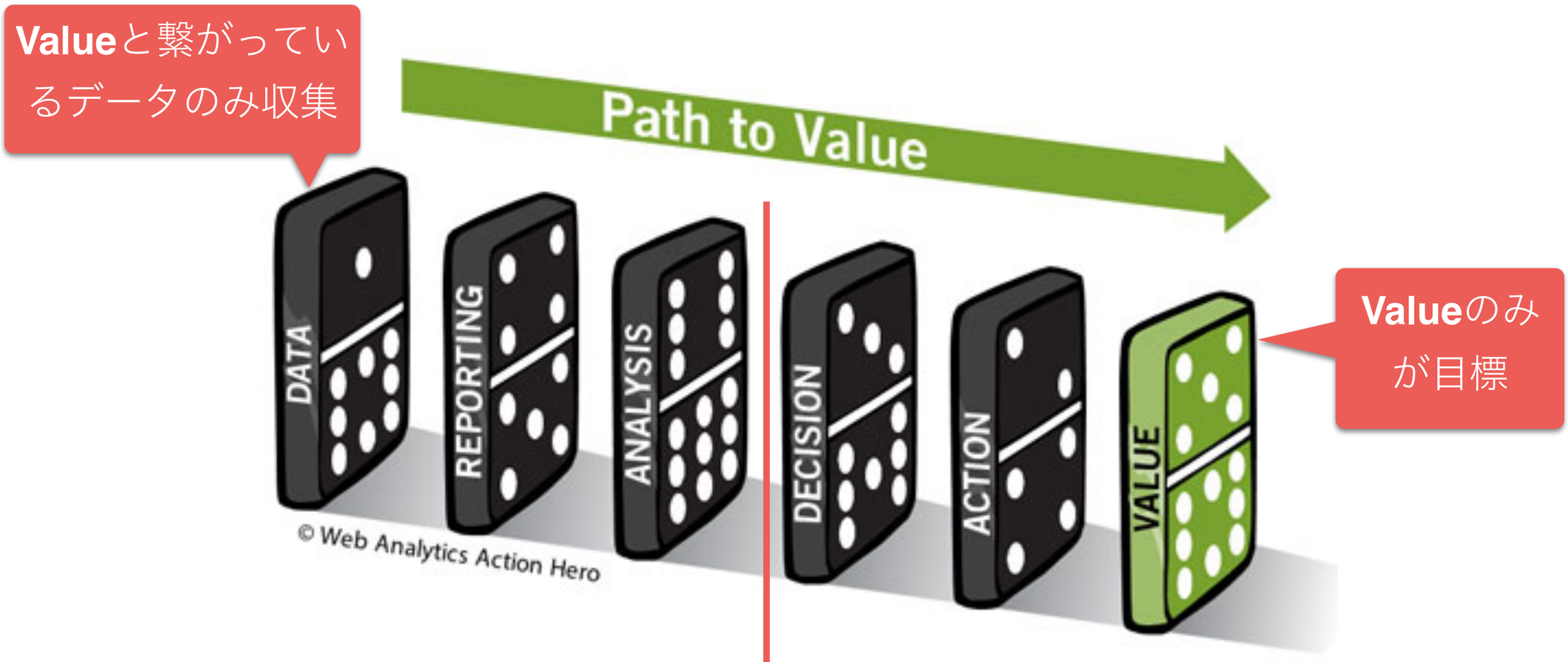
# 原因②

**VALUEに繋がる必要がある**

# 多くの企業が持っている
# るデータ応用のイメージはこれ

# 解析は**Value**と繋ぐ必要がある

最初からValueと解析の関係を明らかにする必要がある

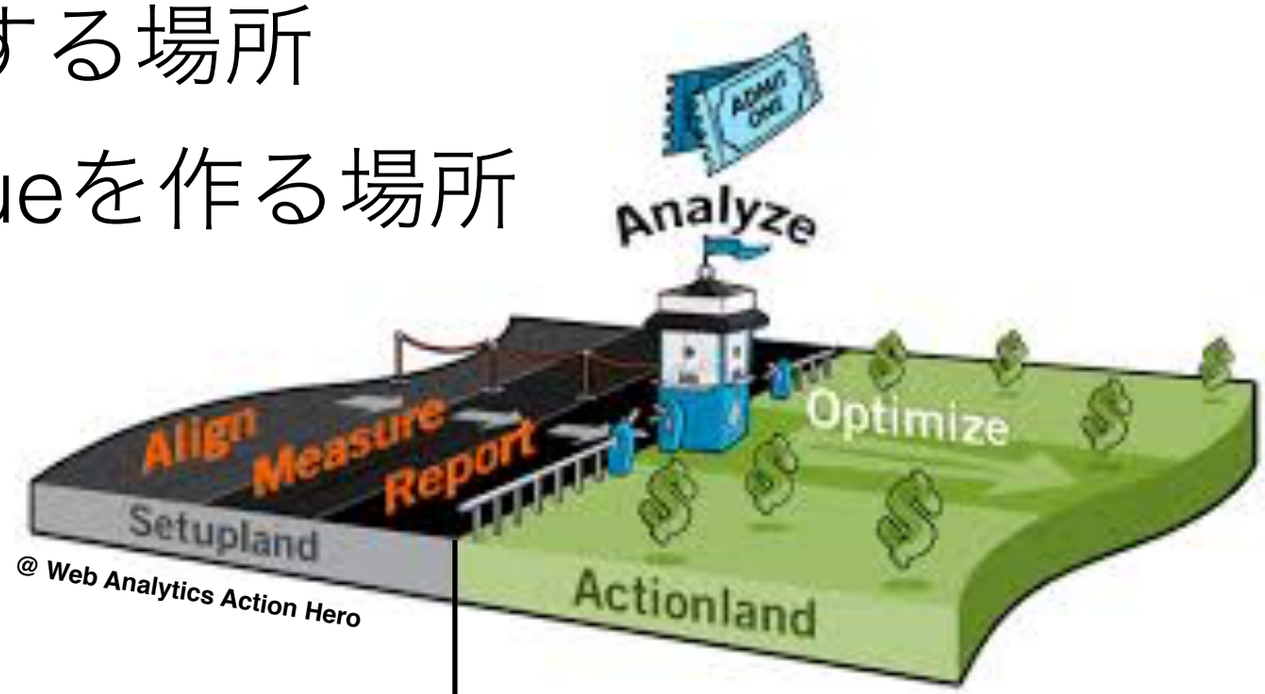



技術者の責任 | マネージャーの責任

# データソリューション構築のフロー

# 問題からシステムへ

- 「Setupland」：Valueと繋がっている解析フローを設計し、プロトタイプを納品する場所
- 「Actionland」：実際Valueを作る場所




**新しい問題**
**（＋データ）**

**解析フローを設計**

データ探索とも言う

プロトタイプ

**POC**

**本番システム**

「actionland」と「setupland」は「Web Analytics Action Hero」という本より

# 問題からシステムへ必要なソフト

**新しい問題**
**（＋データ）**

**解析フローを設計** → **POC** → **本番システム**

・使うソフト：R、Matlab、Python、BIツール
・貴重な知見：数学、統計学、情報学、応用分野(e.g.保険など)
**・課題：精度向上**

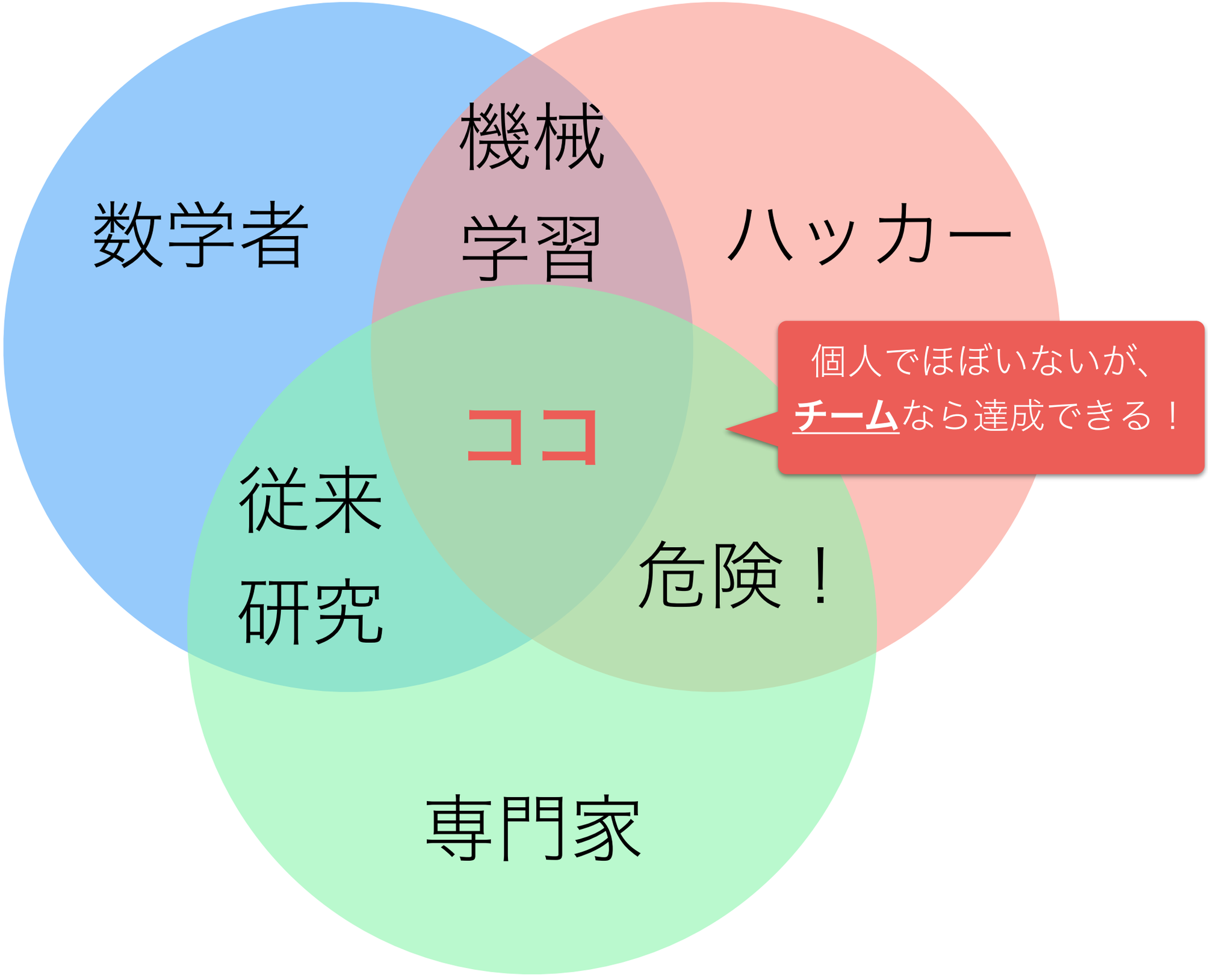
・使うソフト：Hadoop、Spark、クラウド(AWSなど)
・貴重な知見：情報学、応用分野(e.g.保険など)、システムインテグレーション
**・課題：精度と性能の妥協**

# データ解析・AIの活用に必要な人材

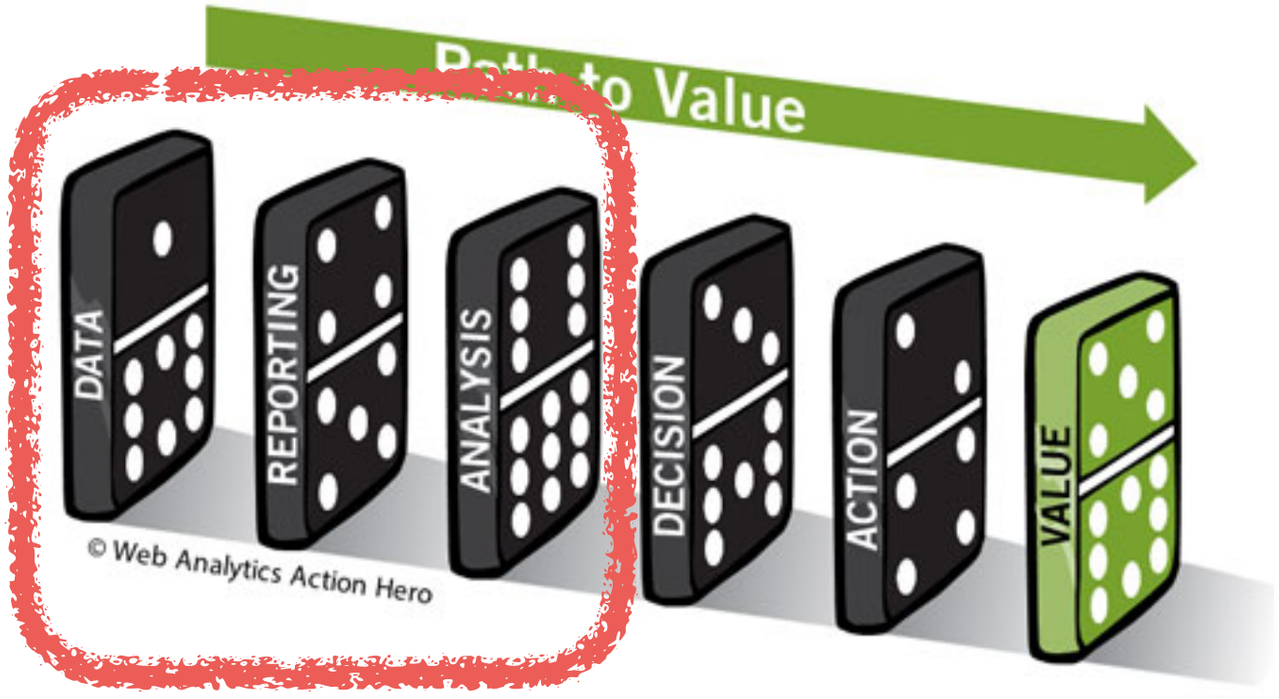
# 機械学習の専門家

+

# VALUEを理解する人

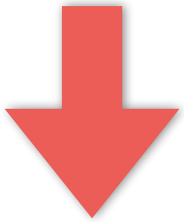

数学者
機械
学習
ハッカー
ココ
従来
研究
危険！
専門家
個人でほぼいないが、
チームなら達成できる！

# ② データ解析の基本

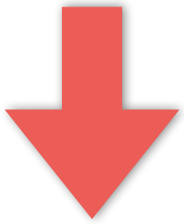

Path to Value
DATA
REPORTING
ANALYSIS
DECISION
ACTION
VALUE
© Web Analytics Action Hero

# データ利活用フロー

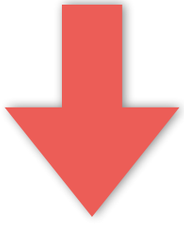
**問題設定**



**データ取得**



**データ解析**

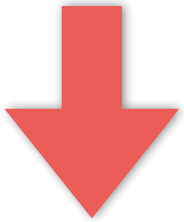


**評価・結果説明**

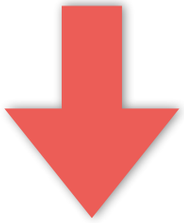
- データを収集する前に問題を設定する
  （VALUEから定義する）
- データの説明には２種類ある
  - 他の専門家に対して：誰でもできる
  - マネージャに対して：チャレンジング

# データ解析フロー

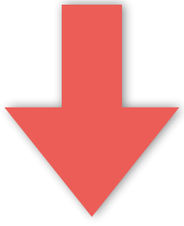
**クレンジング**



**探索**



**モデル化**



**モデル学習**

- **クレンジング**には役割が２つある：
  - 異常値の削除
  - 探索やモデル化の為のデータ変換
- **モデル化**は問題を表現すること
  - どの数値を使う？
  - どの問題の種類を使う？（回帰、分類）
  - モデル化は解析の半分以上を占める
- **モデル学習**とは
  - 実際に機械学習アルゴリズムを走らせて、精度を計算する
  - 機械学習アルゴリズムの選択はポイントではない：
    データの質（クレンジングとモデル化後）が一番重要

# 可視化について

- クレンジング・解析・レポートに貴重な技術
- 可視化することによってパターンや異常値を質的に検出できる
  - 解析の方針を決めるため
  - 最終的に定量的な方法を使う
- 可視化ツールのチャレンジ：
  - 探索しながら思いついた可視化を簡単に作れるか
  - 混雑などを解決できる（例：色などを使う、対話型にする）
- 最近はウエブテクノロジーによる可視化が人気
  - 特に「D3js」(http://d3js.orgをご参考）ライブラリを使って
  - 対話型可視化は簡単に作れる
- 可視化はレポートにも貴重な技術
  - おすすめサイト：http://www.informationisbeautiful.net

# 機械学習モデルとは


- **目的変数**
  - 数字か列挙で表現する
  - 予測したい情報
- **説明変数**
  - 数字か列挙で表現する
  - 持っている情報。この情報を使って、目的変数を求めたい
- **モデル**
  - 説明変数と目的変数を結ぶ「箱」
  - 「箱」の中に数式であったり、コンプイータプログラムであったりする

# データ型・回帰問題・分類問題

- ２種類の変数がある：数値とID型（列挙型）
  - ID型の例：Yes/No、Blue/Red/Yellow、Slow/Fast
- 目的変数の型に応じて問題の種類が違ってくる：
  - 目的変数は**数値型**：回帰問題（数値を求める）
  - 目的変数は**ID型**：分類問題（クラスを求める）
- 問題の種類に応じて使える機械学習アルゴリズムも限られてくる

# 回帰問題の例：売り上げ予測

回帰モデル。アルゴリズム：
・数式型：従来回帰分析、MARS分析
・アルゴリズム式：回帰木、SVR、k近傍法

**説明変数**
ビジネス
データ

機械学習
**モデル**

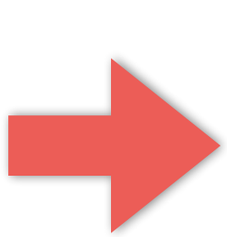
**目的変数**
売り上
の絶対値

例：昨年度売り上げ（数値）、新商品個数、各商品の値段（数値）、中国マーケットを狙うかどうか（Yes/NoのID型）

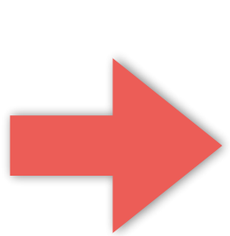
# **分類問題**の例：常連になるか予測

分類モデル。アルゴリズム：
・決定木、SVR、ニューラルネットワーク、k近傍法

**説明変数**
客データ



機械学習
**モデル**



**目的変数**
常連になる？
Yes/No

例：来店時間帯（ID型）、お店で過ごした時間（数値）、買い物の額（数値）、Pontaカードの有無（ID型）

# ID型と数値型の関係

- IDは数値へ変換できるが、おすすめしない
  - 例：Yes→0、No→1
  - しかし、選択した数値によって解析結果が違ってくる：気持ちがよくない
- 数値型はID型に変換できる
  - 例：0≤x≤10→Low、10<x→High
  - しかし、情報量が減るので、注意が必要
  - 目的変数の場合は問題を単純化する（回帰が失敗した時に）
  - 説明変数の場合は雑音をなくすためにありえる

僕はよくする。問題は回帰問題から分類問題になる

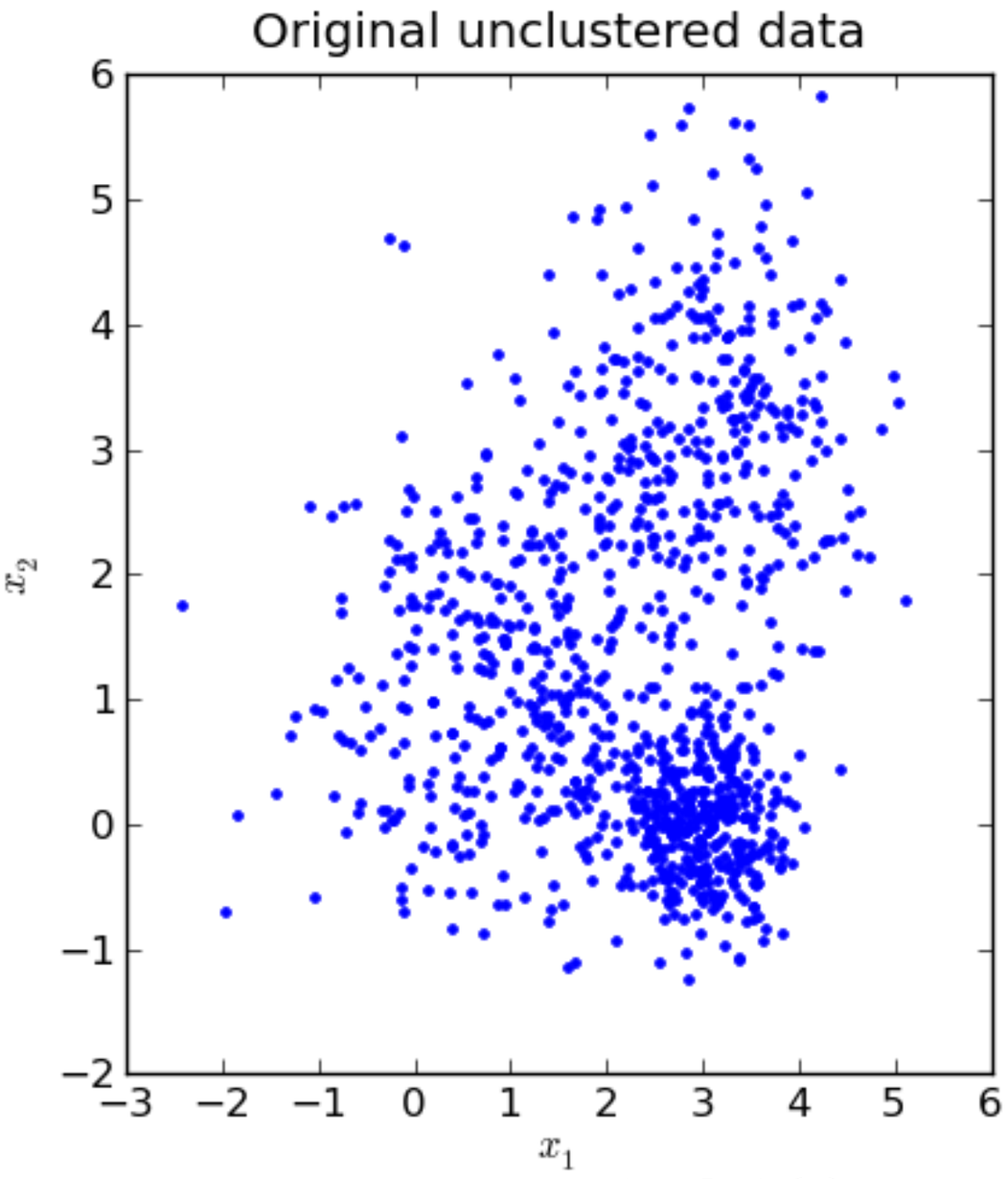
おすすめしない：
僕はしたことない。

# 教師なし学習について

- 目的変数の有無に応じて機会学習は２種類で分ける
  - 目的変数がある：**教師あり**学習。正解があって、機械学習モデルで求める
  - 目的変数がない：**教師なし**学習。データにパターンを見つけたいが、どのパターンがあるかどうか以前にわからない
- 教師なし学習の主な種類：
  - クラスター分析。例：カスタマーのセグメント化など
  - 異常検出。例：クレジットカード不正利用検出

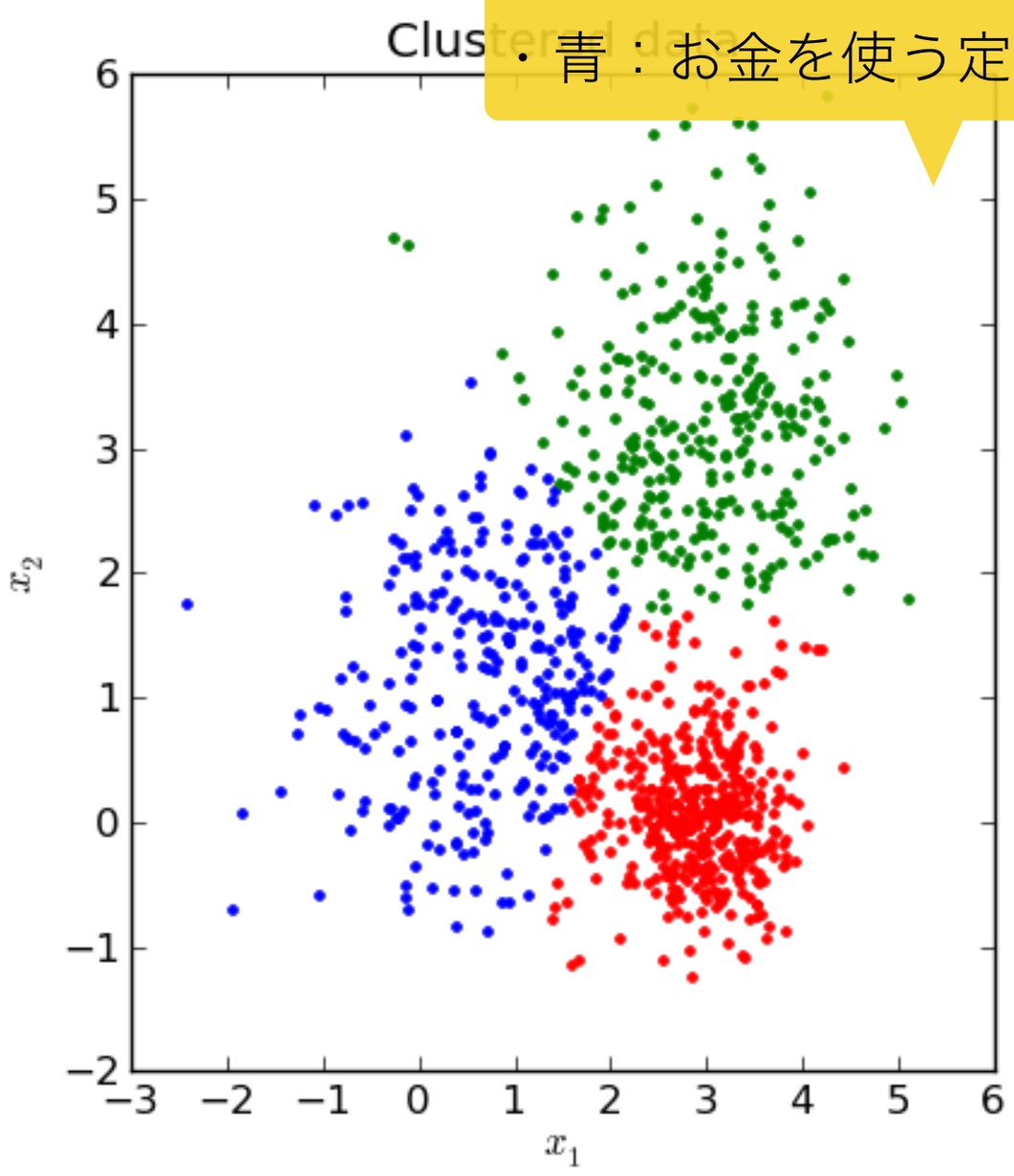
# クラスタ分析の例

例えば：
・赤：定連にならない客
・緑：定連になるが、お金が使わない客
・青：お金を使う定連になる客




**クラスター分析前**

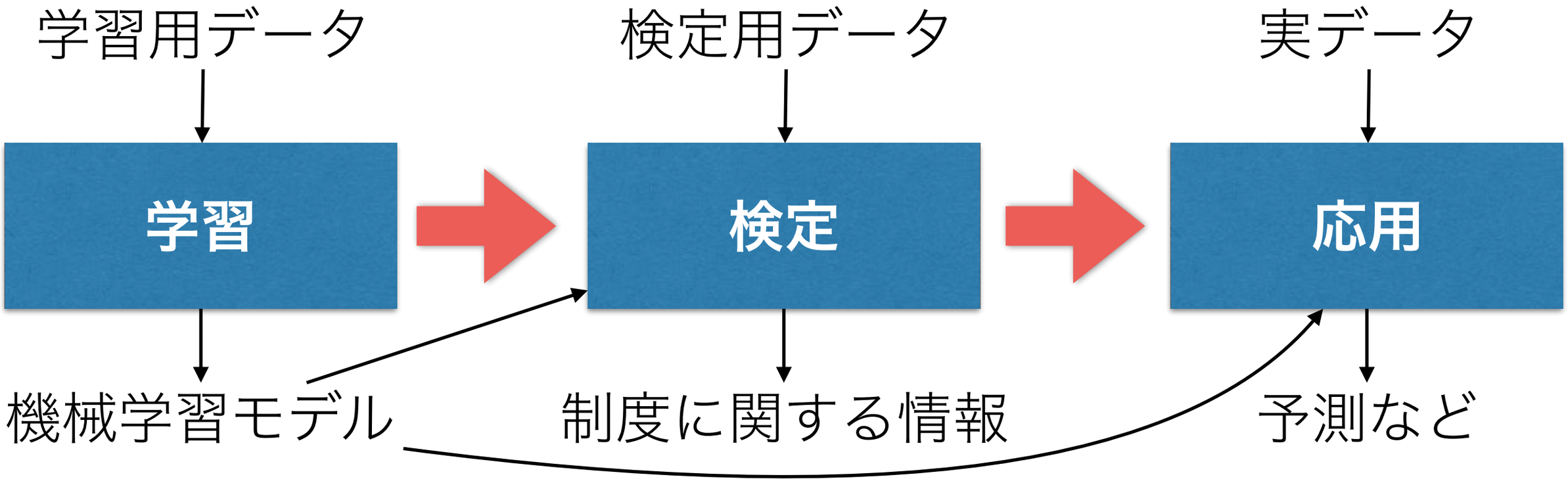
色付けなし




**クラスター分析後**

色付けあり

# モデル学習と検定とは




- **学習**：モデルを作る。教師あり学習の場合は説明変数と目的変数の両方は学習アルゴリズムに与える
- **検定**：モデルを確認する。モデルを使って予測し、正解と比較する（説明変数と目的変数を両方持っている）
- **応用**：実際にビジネスの場でモデルを使う（目的変数の実際値は未知）

# データ：学習用、検定用

- データは三種類がある：
  - 学習用データ、検定用データ、実データ
- 教師あり学習の場合：
  - **学習データ**と**検定データ**は説明変数と目的変数を含む
  - **実データ**は説明変数のみを含む
- **学習データと検定データの分け方**について
  - 異なるデータ：**予測モデル**を作る
  - 同じ：**再現モデル**を作る
- 再現モデルは価値が低い：オーバーフィットの可能性は高い

オーバーフィット問題は演習中に学ぶ

# データ解析の２つの目的

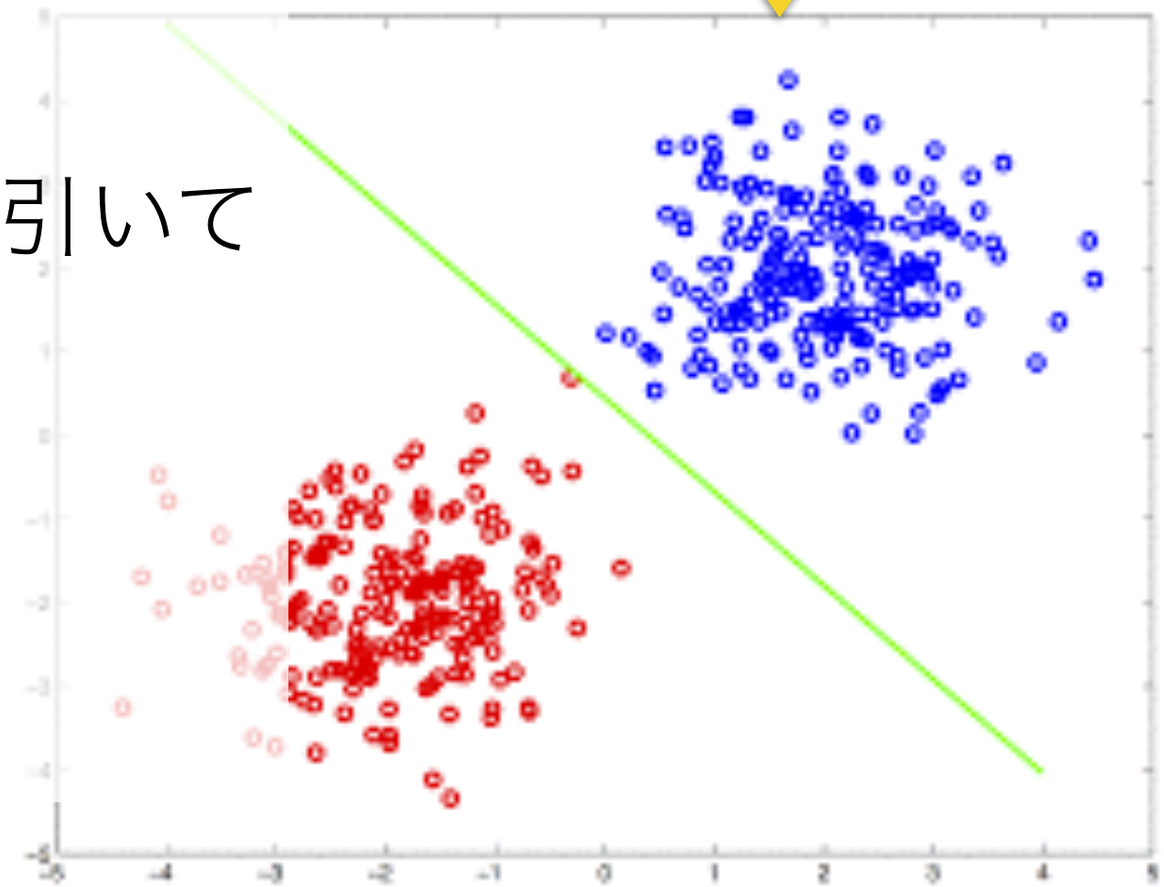
- 解析目的は説明か予測である
- 目的１：**データ説明**
  - 教師あり学習：目的変数と説明変数の関係を理解したい
  - 教師なし学習：説明変数の中にあるパタンを理解したい
- 目的２：**データ予測**（教師あり学習のみ）
  - 説明変数をもって、精度高く目的変数を求めたい

# 線形・非線形

- 線形という言葉はよく出るが、「線で解ける」と意味する
  - **回帰の場合**：目的変数対説明変数のグラフを書くと線になる
  - **分類の場合**：クラスは線を引いて分けられる
- 線形問題は一番簡単な問題

線は２次元の場合、n次元の場合は超平面をいう

この分類問題は線で解ける：赤と青は線で分離できる。このデータは**線形分離可能**という

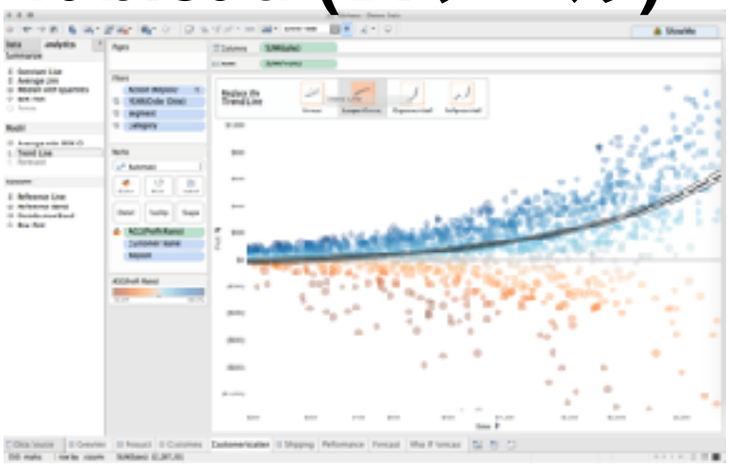


# ③ Rについて

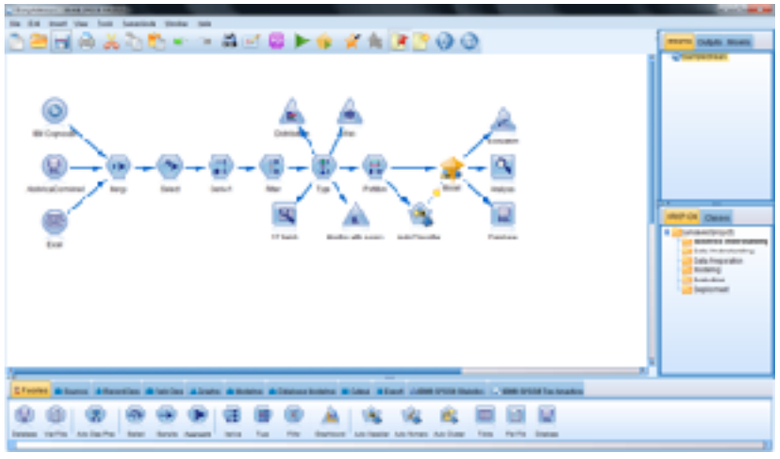
# GUI型対プログラミング型

| | 利点 | 欠点 | 例 |
|---|---|---|---|
| **GUI** | 使いやすい | 決められた解析のみ可能 | SPSS、BIツール |
| **プログラミング言語** | どの解析でも可能 | 学びにくい | **次のスライドをご参考** |

## Tableau (BIツール)



## IBM SPSS



## R Studio

# 必要に応じてプログラミング言語が決まる

| プログラミング言語について | 利点 | 欠点 | 例 |
|---|---|---|---|
| 解析用言語 | ・可視化まで可能<br>・探索に最適 | ・解析以外は難しい | R、Julia |
| 一般言語（スクリプト） | ・解析以外も楽（クレンジングなど）<br>・POCに最適 | ・探索に固い場合もある<br>・ライブラリは解析用言語　Pythonほど充実していない | |
| 一般言語（コンパイル式） | ・高性能ライブラリが充実<br>・本番環境に最適 | ・POCや探索には固い | Java 、C# |

機械学習専用
ライブラリーを使う

**言語周りも大きなポイント。例：**

- R：パッケージが多い。「R Studio」というIDEはすごく便利。
- Python：Rと連携する。深層学習ライブラリはだいたいPythonが一番利用しやすい
- Java：HadoopやSparkはJavaで実装されている

今日はRを使う

# Rは言語だけではない

- **言語**である
  - 1993にベル研究所から公開（C言語やUNIXの発祥地）
  - 当時人気のS言語のオープン版として開発された（S言語はほぼなくなりました）
  - コンパイラと標準ライブラリも充実
- データ解析**ツール**である
  - クレンシング、統計解析、機械学習、データ可視化など
- **エコシステム**である
  - パッケージが充実（“There’s  a package for that”）
- **コミュニティ**である
  - 統計学者と機械学習学者は皆使っている

# Rは人気？

- アメリカで一番給料の高いITスキル(dice.comのサーベィ,2014)
- データ科学分野にいおてRはSQLの次に一番使われている言語(O'Reilly Survey,2014)
- Rユーザは２億人を超えてしまう(Oracle社によるサーベィ)

# Google Trends より


**ネット検索数は過去6年間で2倍以上！**

# Rのユーザインターフェイス

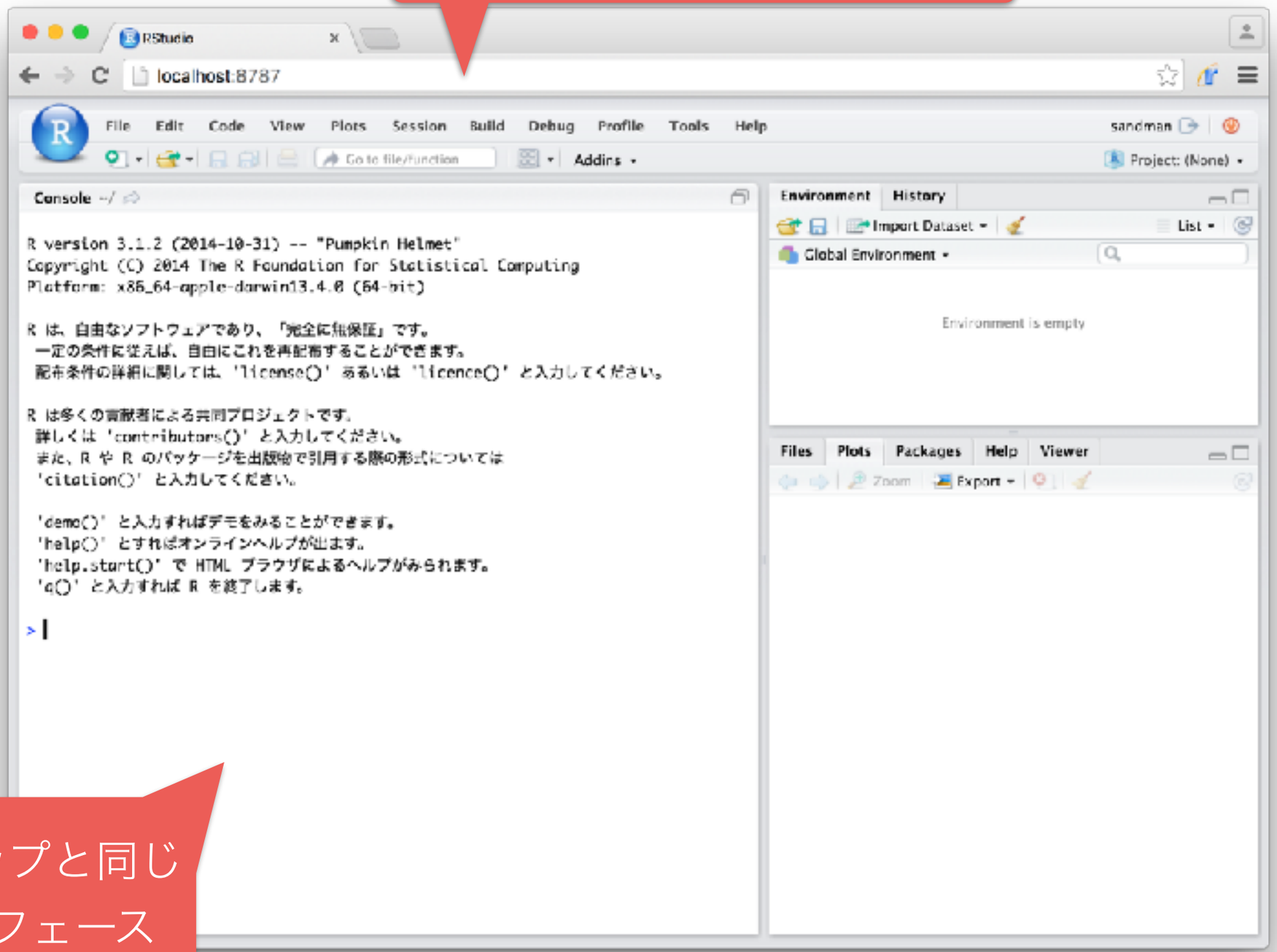
- R言語で書かれたコマンドを実行する
  - 裏はRコンパイラがマシンコードへ変換してプロセッサー上で実行する
- Rコマンドの実行方法
  - 方法①：Rプログラム（スクリプト）をファイルに保存してRコンパイラ（RScript）に渡す
  - 方法②：REPLインターフェイスを使う。ちなみに、REPLはRead Eval Print Loopの略で、ターミナルにコマンドを書いて、すぐに結果が表示される。
- 通常はロカルでRプログラムを実行するが、下記のやり方もある
  - 遠隔実行：「R Server」というソフトを使うと、重い計算を他のマシンで実行できる
  - 分散実行：R Serverや通常のロカルR実行環境をネットワークを通じて協調するように設定できる

# Rスタジオについて

- RStudioはRのグラフィックインターフェース
- コマンドラインを使うところは変わらないが、便利機能が沢山！
  - ベストIDEランキングによく出る
- デスクトップアプリケーションに加えてサーバーもある
  - サーバーはRStudioのWebサービス版。計算はサーバー側に実行する
- 今回は僕のパソコンにインストールしているRStudioサーバーに接続してもらう

# RStudioの画面（ウエブバージョン）

URLはボードをご参考




デスクトップと同じ
インターフェース

RStudio

localhost:8787

File Edit Code View Plots Session Build Debug Profile Tools Help

sandman

Go to file/function

Addins

Project: (None)

Console ~/

```
R version 3.1.2 (2014-10-31) -- "Pumpkin Helmet"
Copyright (C) 2014 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin13.4.0 (64-bit)

R は、自由なソフトウェアであり、「完全に無保証」です。
 一定の条件に従えば、自由にこれを再配布することができます。
 配布条件の詳細に関しては、'license()' あるいは 'licence()' と

R は多くの貢献者による共同プロジェクトです。
 詳しくは 'contributors()' と入力してください。
 また、R や R のパッケージを出版物で引用する際の形式については
 'citation()' と入力してください。

 'demo()' と入力すればデモをみることができます。
 'help()' とすればオンラインヘルプが出ます。
 'help.start()' で HTML ブラウザによるヘルプがみられます。
 'q()' と入力すれば R を終了します。

>
```

Environment History

Import Dataset

List

Global Environment

Environment is empty

・データインスペクター
・コマンド履歴

Files Plots Packages Help Viewer

Zoom Export

・グラフ
・ヘルプ（英語のみ）
など

コンソール。コマンドを入力するところ

# 今からRハンズオン