教師あり・なし学習

機械学習の分類

	入力に関するデータ	出力に関するデータ(正答)	活用例
教師あり学習	有り	有り	不動産価格の予測(回帰)、検査結果に基づく陽性判定(分類)
教師なし学習	有り	無し	顧客の属性に基づく分類、情報の集約
強化学習	有り	直接的な正答は無いが、報酬が与えられる	ロボットの歩行学習、将棋、囲碁

教師あり学習

- 正解の用意された問題集（教師データ）がある
- 教師は正解の導き方を詳しく説明してくれない
- 学習者（コンピュータ）は問題と正解の関係を推測する
- これまで見たことの無い新しい類似の問題に対して正解を導けるようになることを目指す

教師あり学習の概要

- 入力とそれに対応する正解出力の組からなる教師データを用いる
- 教師データに基づいて入力と出力の関係を表現するモデルを構築する
- 新しい入力に対して精度の高い予測出力ができることを目指す
- 出力が連続値の場合「回帰」モデル、出力がカテゴリの場合「分類」モデルを構築する

ex. 不動産価格の予測（回帰）、検査結果に基づく陽性判定（分類）

教師なし学習

出力に関するデータ（正解）が与えられていない

クラスタリングか次元圧縮を目的とする

- クラスタリング
  - 入力データに基づいてサンプルをいくつかのクラスタに分類する
- 次元圧縮
  - 多種類の入力を少数種類の入力にまとめる

ex. 顧客の属性に基づいて顧客をいくつかのクラスタに分類する（クラスタリング）、学生の複数科目の試験成績データに基づいて学生の能力を説明できる少数の変数を作る（次元圧縮）

強化学習（本授業の範囲外）

試行錯誤を通じて報酬が最大となるような行動や選択を学習する

ex.

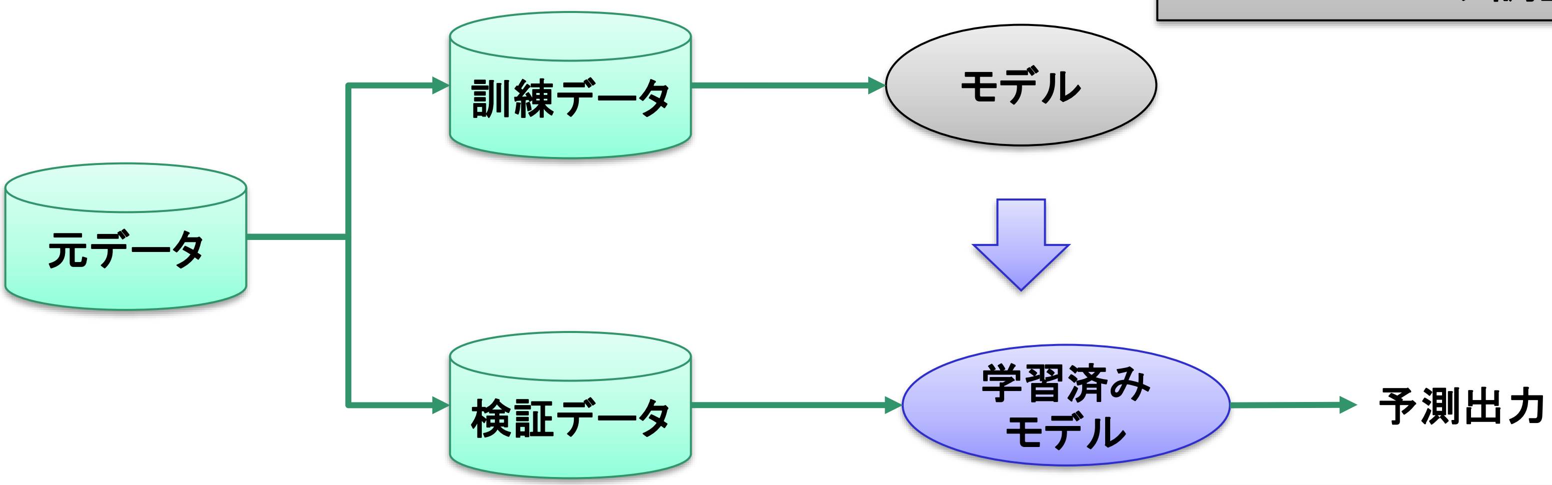
- ロボットの歩行学習
  - 歩けた距離を報酬とする。機械は手足の動かし方を試行錯誤して報酬が最大になるような動かし方を学習する
- 将棋
  - 敵の王将を取ることを報酬とする。駒の動かし方を試行錯誤して報酬が最大になる戦略を学習する

機械学習のプロセス

1. データ分割

2. モデル（アルゴリズム）選択

3. 学習：訓練データ（入出力の組）をモデルに与えてパラメータ調整



4. 予測：検証データ（入力）を学習済みモデルに与えて予測出力

5. 評価：予測出力と検証データ（出力）を比較して予測精度を評価

機械学習タスクの例：

- 腫瘍の検査値X（直径や面積、滑らかさ等）と良性・悪性の診断結果yが組になったデータ (X,y) が100サンプルある場合を考える
- 入力Xからyを予測できるモデル y=f(X,C) を構築したい（C:モデルのパラメータ）
- 80サンプルでモデルを訓練して、20サンプルでモデルの予測精度を評価する

機械学習のアルゴリズム

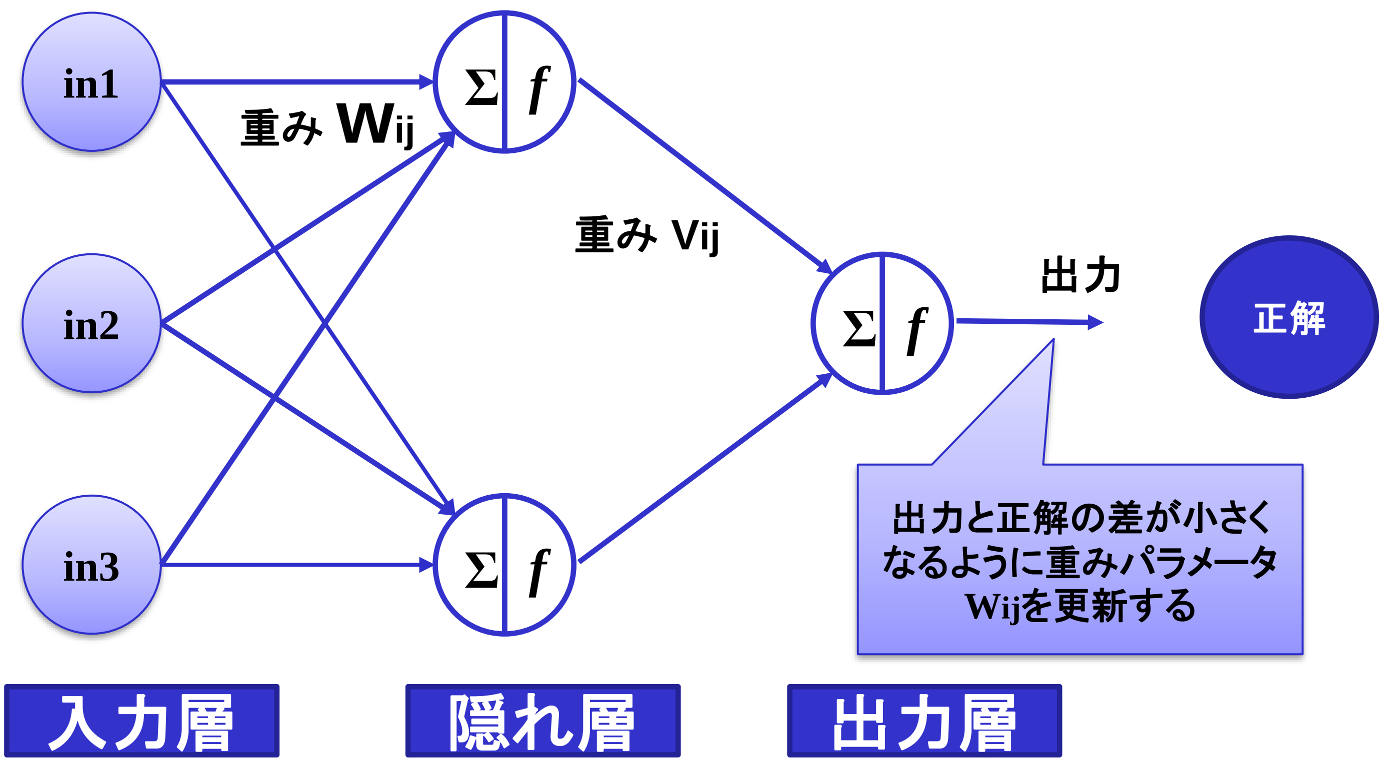
教師あり学習
- 分類
  - k近傍法、SVM、決定木、ランダムフォレスト、パーセプトロン，ロジスティック回帰，ニューラルネットワーク
- 回帰
  - 線形回帰、Ridge回帰、Lasso回帰、Elastic Net

教師なし学習
- クラスタリング
  - k平均法
- 次元削減
  - PCA、自己組織化マップ

強化学習（本授業の範囲外）
- Q-learning、SARSA、モンテカルロ法

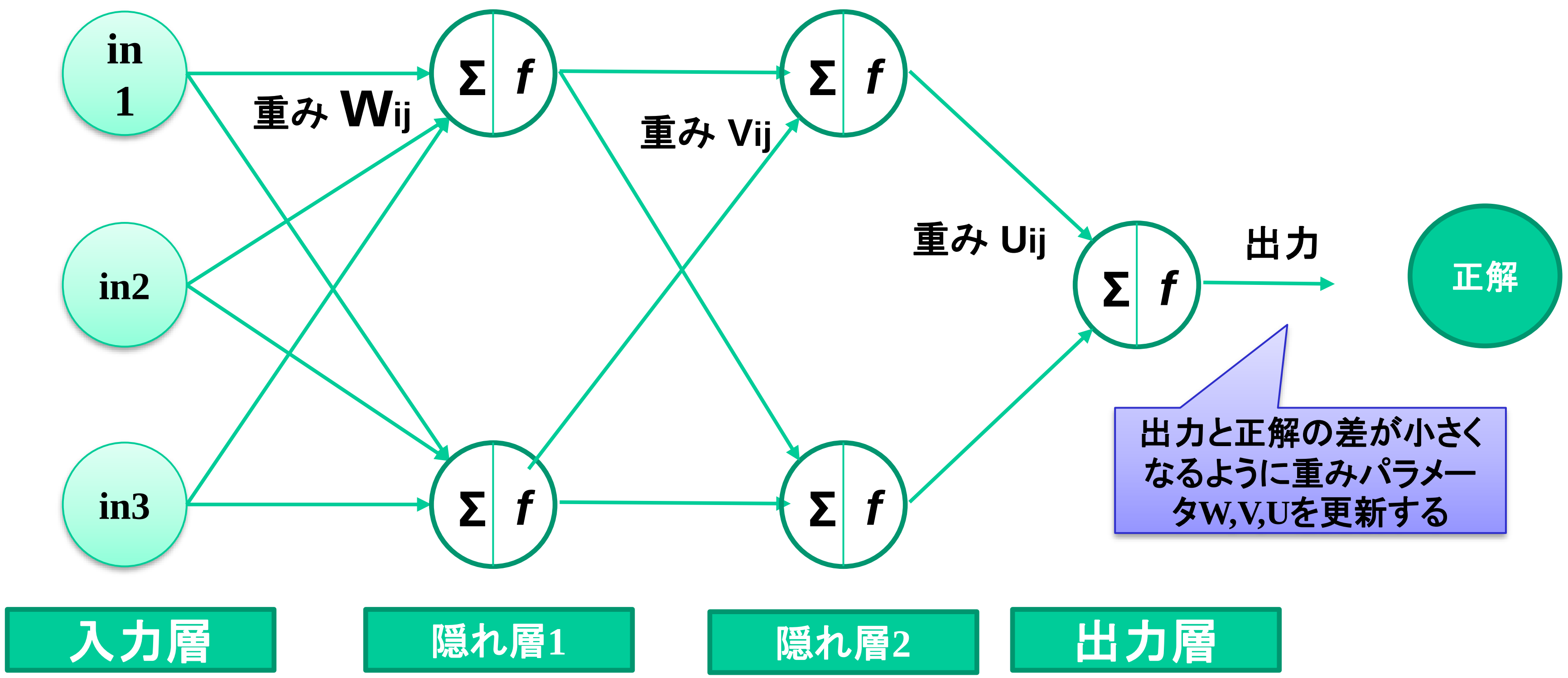
ニューラルネットワーク

脳の神経回路の仕組みを真似たモデル．機械学習アルゴリズムの一種
入力と対応する正解出力の組からなる教師データを与える
出力データと正解データの差が小さくなるように重みパラメータを更新する



深層学習

ニューラルネットワークの隠れ層を増やしたモデル
画像や音声等入力と対応する正解出力の組からなる教師データを与える
出力データと正解データの差が小さくなるように重みパラメータを更新する
機械は与えられたデータに基づいて特徴量を抽出・学習する



機械学習・深層学習のための言語

R
- ベクトル・行列やデータフレーム等データ処理機能、描画機能が標準で用意されており操作が統一されている
- 統計解析の機能が標準で用意されている
- 機械学習の各種アルゴリズムがパッケージとして豊富に提供されている
- 深層学習のライブラリはあるものの主流ではない

Python
- ベクトル・行列計算にはNumPyやSciPy、データフレーム処理にはpandas、描画にはMatplotlibのように外部ライブラリが必要で操作が統一されていない
- 統計解析の機能が標準で用意されていない（別途StatsModelsが必要）
- 機械学習のライブラリ（scikit-learn）がある
- 深層学習のライブラリ（TensorFlow, Keras, PyTorch, Caffe等）が充実しており主流となっている

CSAJ

演習で採用する環境

機械学習

- 言語と開発環境：RとR Studio
- ライブラリ：Rの各種パッケージ
- 実行環境：ローカル

深層学習

- 言語と開発環境：PythonとGoogle Colaboratory
- ライブラリ：Keras
- 実行環境：クラウド

教師あり学習（分類）

教師あり学習（再掲）

- 正解の用意された問題集（教師データ）がある
- 教師は正解の導き方を詳しく説明してくれない
- 学習者（コンピュータ）は問題と正解の関係を推測する
- これまで見たことの無い新しい類似の問題に対して正解を導けるようになることを目指す

教師あり学習の概要

- 入力とそれに対応する正解出力の組からなる教師データを用いる
- 教師データに基づいて入力と出力の関係を表現するモデルを構築する
- 新しい入力に対して精度の高い予測出力ができることを目指す
- 出力が連続値の場合「回帰」モデル、出力がカテゴリの場合「分類」モデルを構築する

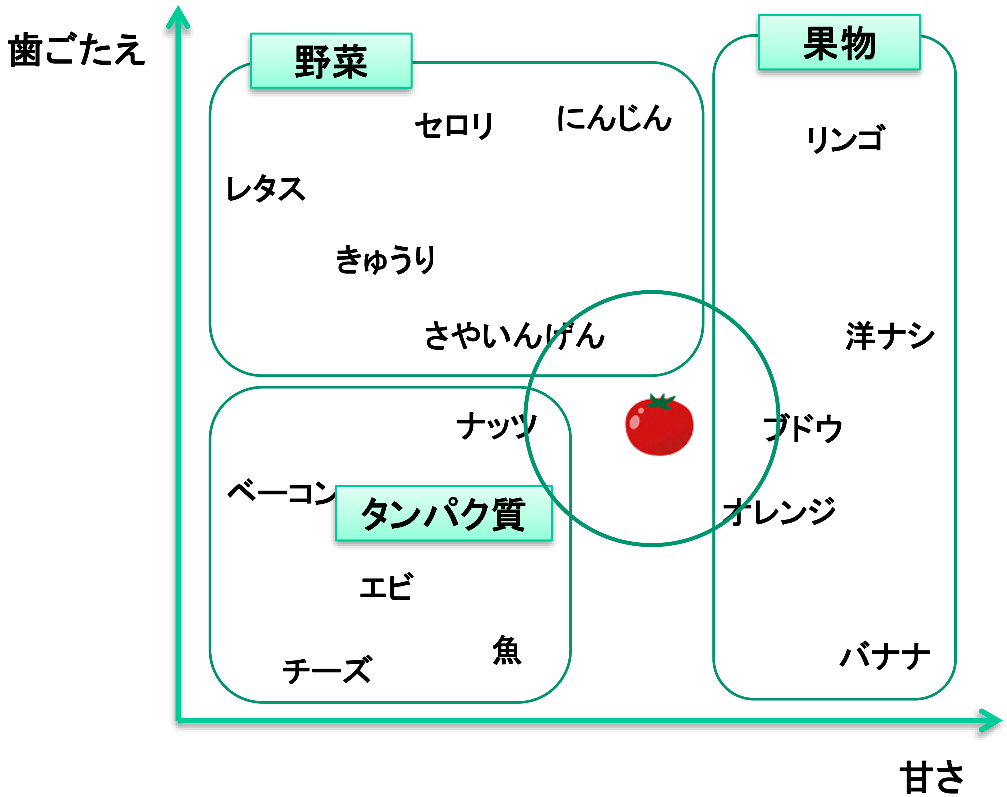
ex. 不動産価格の予測（回帰）、検査結果に基づく陽性判定（分類）

k-近傍法

教師あり学習（分類）

k-近傍法の概要と例

k-近傍法：特徴空間上で最も近い距離にある訓練データに基づいて分類を行う方法



例：トマトは野菜か果物か

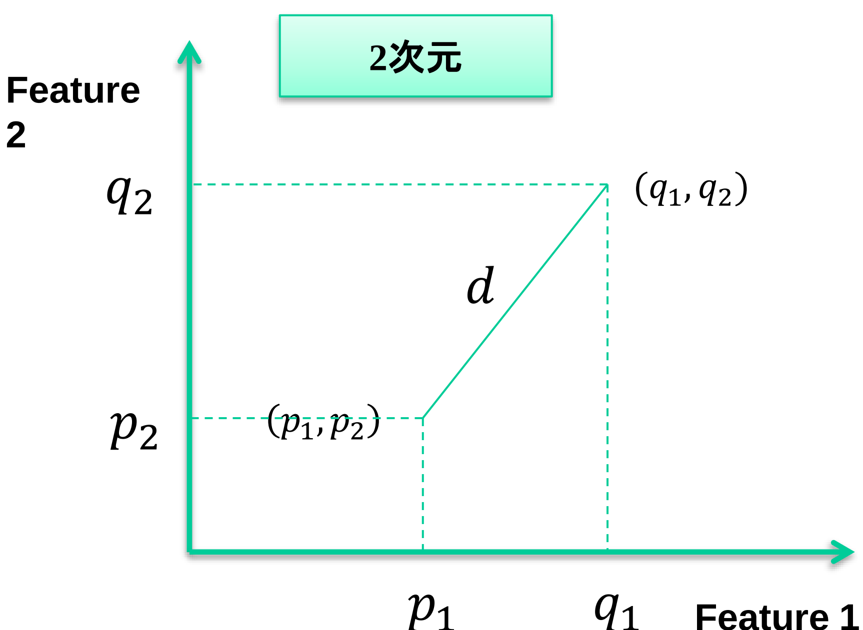
- 甘さと歯ごたえの2つの特徴量に基づいて判定すると仮定する
- 2次元特徴空間（平面）における近傍点が野菜なら野菜、果物なら果物と判定
- この例ではトマトの4つの近傍はブドウ、オレンジ、さやいんげん、ナッツだが、4つのうち2つが果物なので果物！

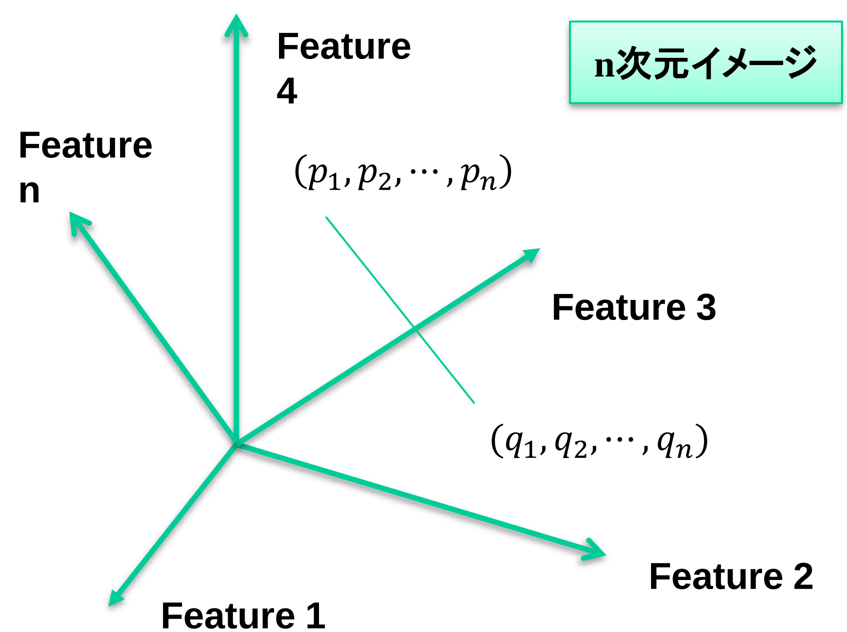
出典：Brett Lantz：「Rによる機械学習」翔泳社（2017）を基に鈴木作成

k-近傍法における次元と距離

例：トマトは野菜か果物か
- 甘さと歯ごたえの2つの特徴量に基づいて判定
- 2次元特徴空間（平面）における近傍点が野菜なら野菜、果物なら果物
- 2点(p_1, p_2), (q_1, q_2)間の距離
  - $d = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2}$

例：注目する腫瘍が良性か悪性か
- ガン細胞のサイズや滑らかさなど30の特徴量に基づいて判定
- 30次元特徴空間における近傍点が良性なら良性、悪性なら悪性
- 2点$(p_1, p_2, \cdots, p_n)$, $(q_1, q_2, \cdots, q_n)$間の距離
  - $\sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \cdots + (p_n - q_n)^2}$

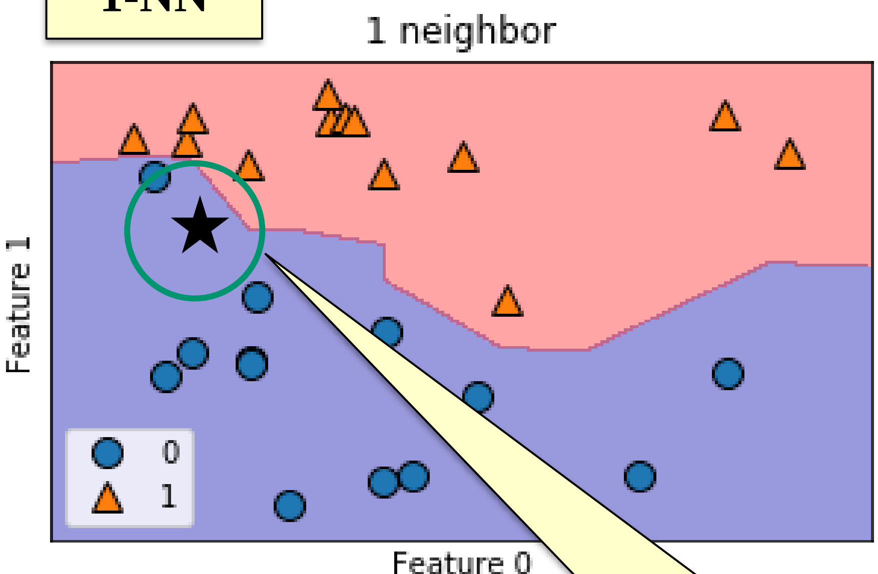




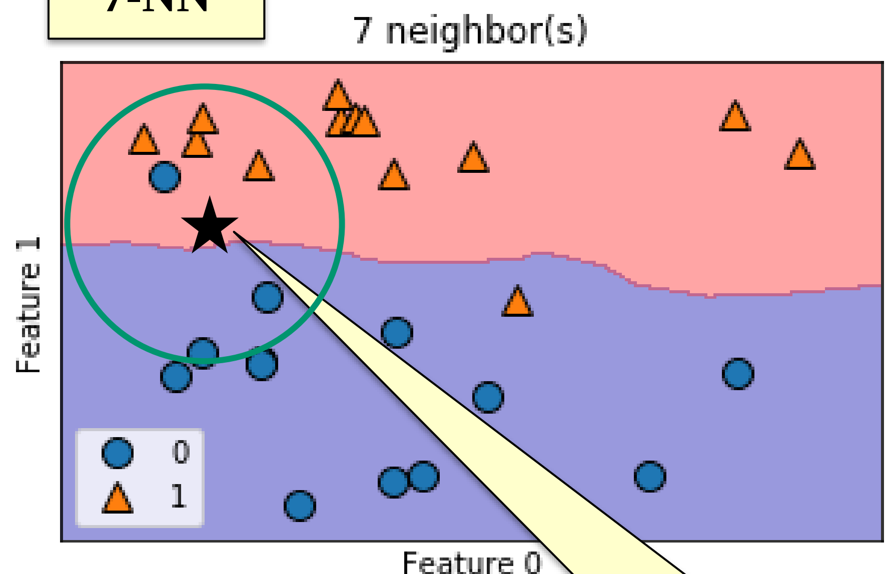
k-近傍法による予測方法とkの値による違い

新しいデータ点は最近傍点（もしくはk個の近傍点うち多数派）と同じクラスと予測

1-NN



7-NN



★の最近傍点は●
⇒　★は●と同じクラスと予測

★の7つの近傍点は
▲が4、●が3
⇒★は▲と同じクラスと予測
（多数決）

- 線形分離不可能なデータセットでも分類可能
- kが小さい場合きめ細かく分類、大きい場合滑らかに分類される

※データ出典：A. C. Muller et al. mglearn.datasets

k-近傍法の処理手順と実装

処理手順

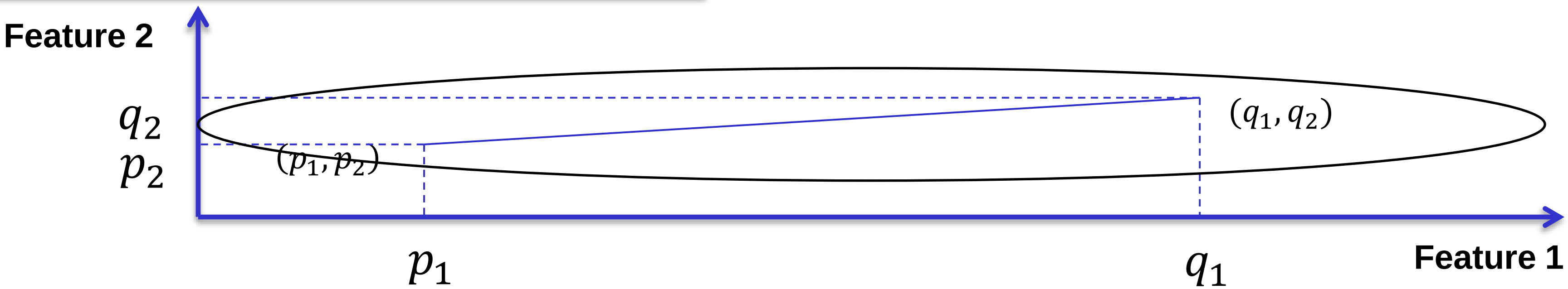
- 学習
  - 訓練データセット（特徴量・クラス）を格納するだけ（学習不要）
- 予測
  1. 未分類のデータ点1個に対して、分類済みの近傍データ点をk個見つける
  2. 未分類データ点は、分類済み近傍データ点の多数派が所属するクラスに分類する

実装

- Rのclassパッケージに含まれるknn実装など

k近傍法で用いるデータの前処理の必要性

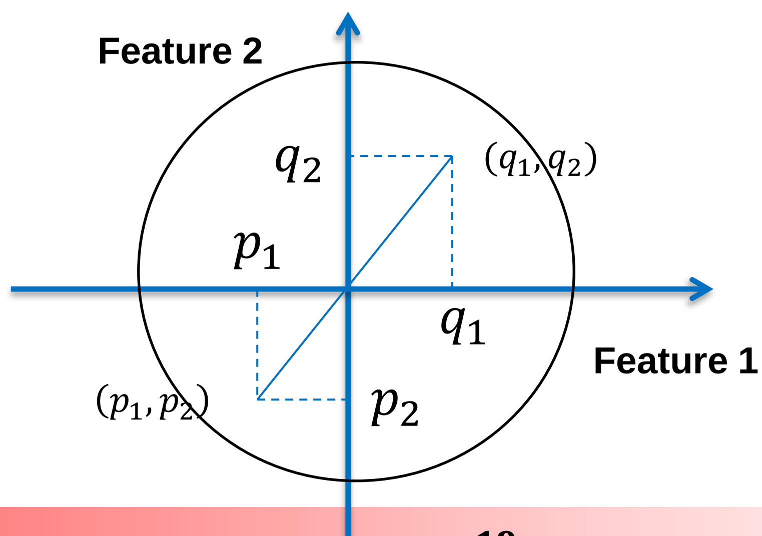
特徴量間で値の範囲が大きく異なる場合



$$\sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2} \sim \sqrt{(p_1 - q_1)^2} = |p_1 - q_1|$$

値の範囲が狭い特徴量がモデル構築において評価されない

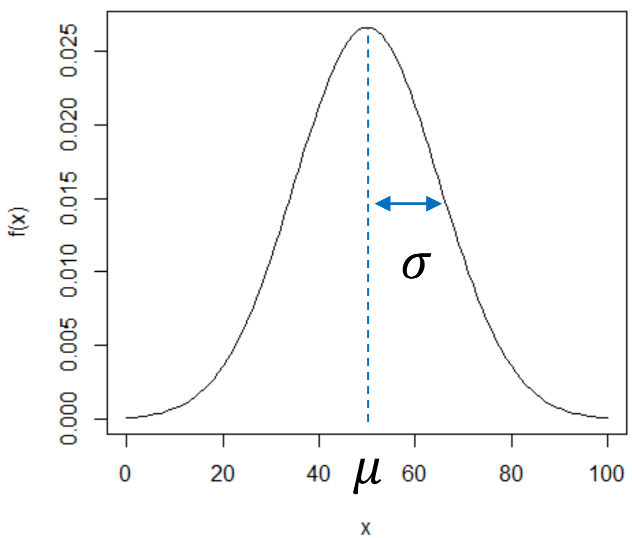
正規化・標準化



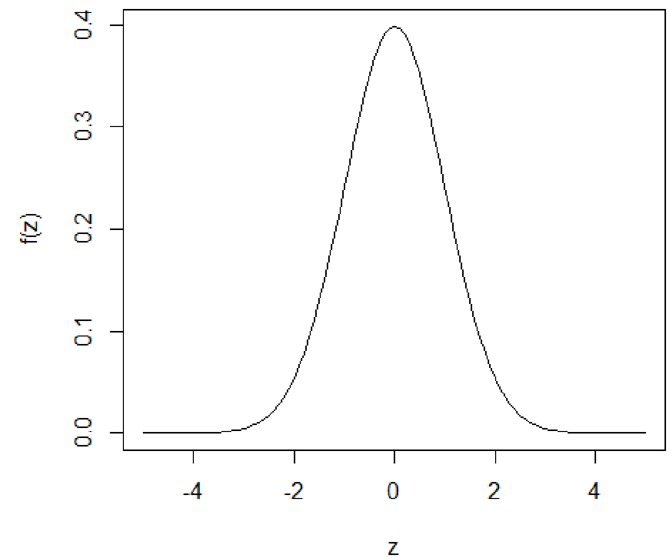
すべての特徴量について正規化・標準化を行い、どの特徴量も同等に評価されるようにする必要がある

Zスコア標準化・最大最小正規化

Zスコア標準化



$$Z = \frac{X - \mu}{\sigma}$$



値の範囲が広すぎる

値の範囲が標準化された

最大最小正規化

$$X' = \frac{X - \min(X)}{\max(X) - \min(X)}$$

$$X' \in [0,1]$$

値の範囲が0以上1以下に標準化された

※参考：k-近傍法を利用した回帰

処理手順

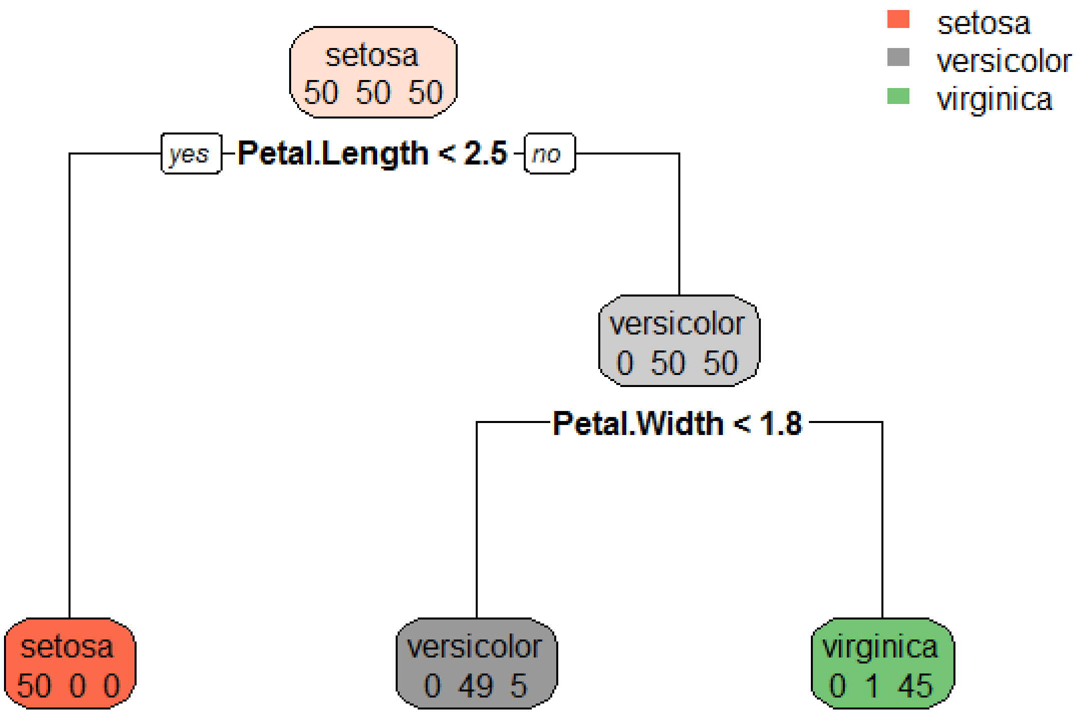
- 学習
  - 訓練データセット（特徴量・目的変数）を格納するだけ（学習不要）
- 予測
  1. 目的変数の値が未知のデータ点1個に対して、特徴空間における近傍データ点をk個見つける
  2. k個の近傍データ点の目的変数の平均を、注目データ点の目的変数の値とする

決定木

教師あり学習（分類）

決定木の概要と例

決定木：特徴量に関する条件分岐をたどると最終的に分類が決定する木構造

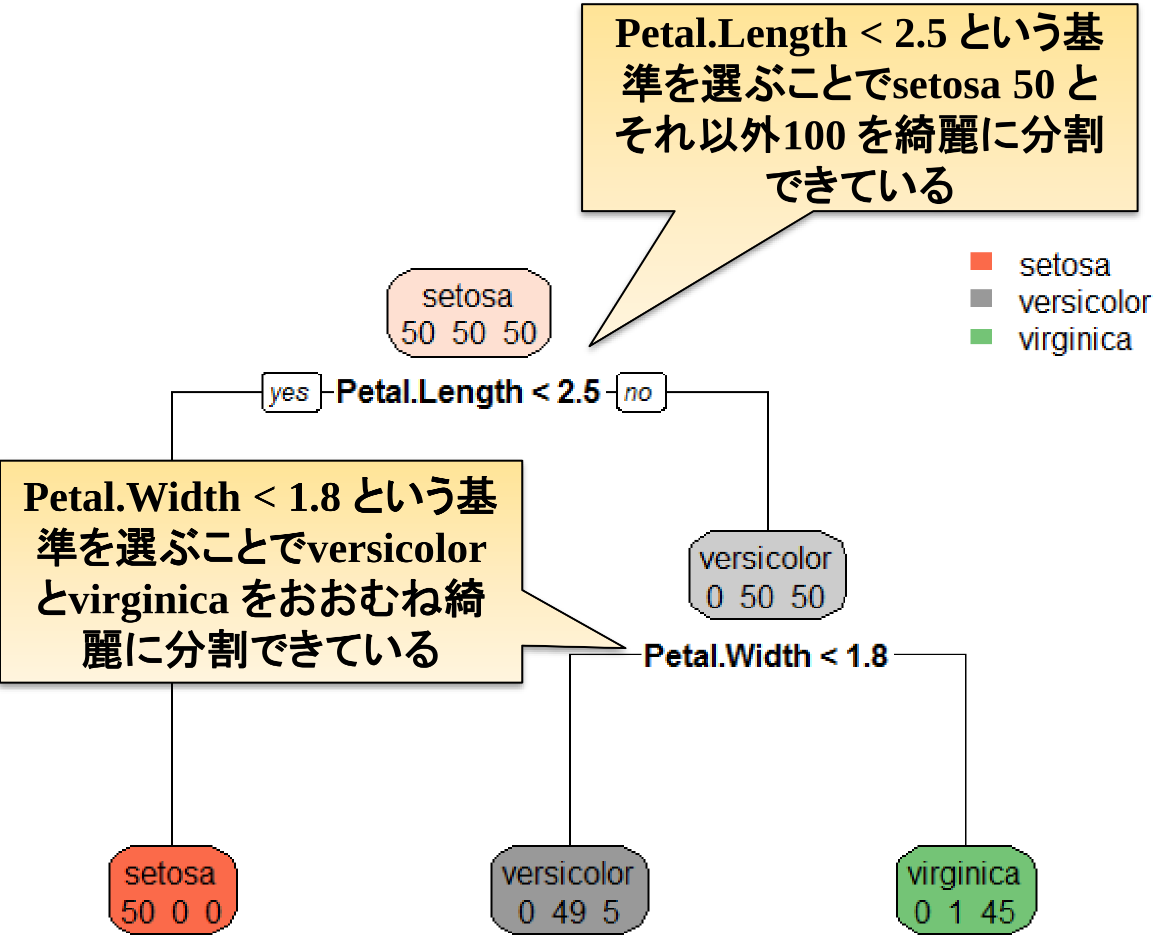


例：アヤメの種類を咢と花弁の長さと幅から決定する決定木

- **Petal（花弁）の長さが2.5未満ならsetosa**
  - **2.5以上で幅が1.8未満ならversicolor**
  - **2.5以上で幅が1.8以上ならvirginica**

- **setosa 50個体、versicolor 50個体、virginica 50個体からなる全150個体を、本モデルによっておおむね綺麗に分けられている**

決定木の学習手順



学習手順

1. 最も綺麗にデータを分割できる基準を選ぶ
   ※「情報利得」が最大となる基準
2. データを分割する
3. 終了条件を満たすまで1と2を繰り返す
   ※終了条件は、木の深さ、終端ノード数、ノードに含まれるデータ数、誤り率などを考慮する

実装

- RのrpartやC50など

決定木：
情報利得による分割基準の選択

エントロピー

エントロピー： 乱雑さ

- ノード N_tのエントロピー

$$H(N_t) = -\sum_{i=1}^{n} p_i \log_2 p_i$$

- p_i ：ノードN_tにおけるクラスi に属するインスタンスの割合
- n ：クラス数

ノードN_t

クラス1: p_1
クラス2: p_2
クラス3: p_3
…
クラスn: p_n

ただし $\sum_{i=1}^{n} p_i = 1$

ノードN_tのエントロピー

$$H(N_t) = -\sum_{i=1}^{n} p_i \log_2 p_i$$

エントロピーの性質（2クラス）

$$H(N_t) = -\sum_{i=1}^{n} p_i \log_2 p_i$$

$n = 2$

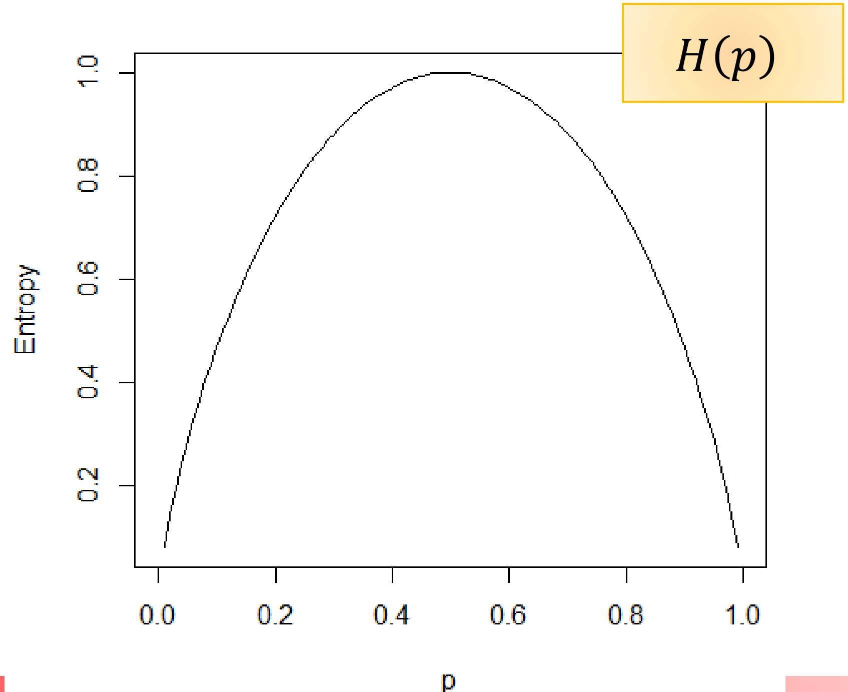
ノードN_t

クラス1: $p_1 = p$
クラス2: $p_2 = 1 - p$

※一方のクラスの割合をpとおいた

ノードN_tのエントロピー

$$H(p) = -p\log_2 p \; - (1-p)\log_2(1-p)$$



エントロピーの性質（2クラス）

- $p = \frac{1}{2}$ で$H\left(\frac{1}{2}\right) = 1.0$（最大値）
  - データの割合が50:50 のときエントロピー最大
- $p \to 0 \; or \; p \to 1$ で$H(p) \to 0$
  - いずれか一方のクラスしかない場合エントロピーは0

クラスの混在の程度が高いほどエントロピーが大きい

エントロピーの性質(nクラス)

ノードN_t

クラス1: p_1
クラス2: p_2
クラス3: p_3
…
クラスn: p_n

ただし $\sum_{i=1}^{n} p_i = 1$

ノードN_tのエントロピー

$$H(N_t) = -\sum_{i=1}^{n} p_i \log_2 p_i$$

エントロピーの性質(nクラス)

- データの割合が均等なとき($p_i = \frac{1}{n}$)、エントロピー最大

$$H(N_t) = -\sum_{i=1}^{n} \frac{1}{n} \log_2 \frac{1}{n} = \log_2 n$$

- いずれか一つのクラスしか含まない場合エントロピーは0

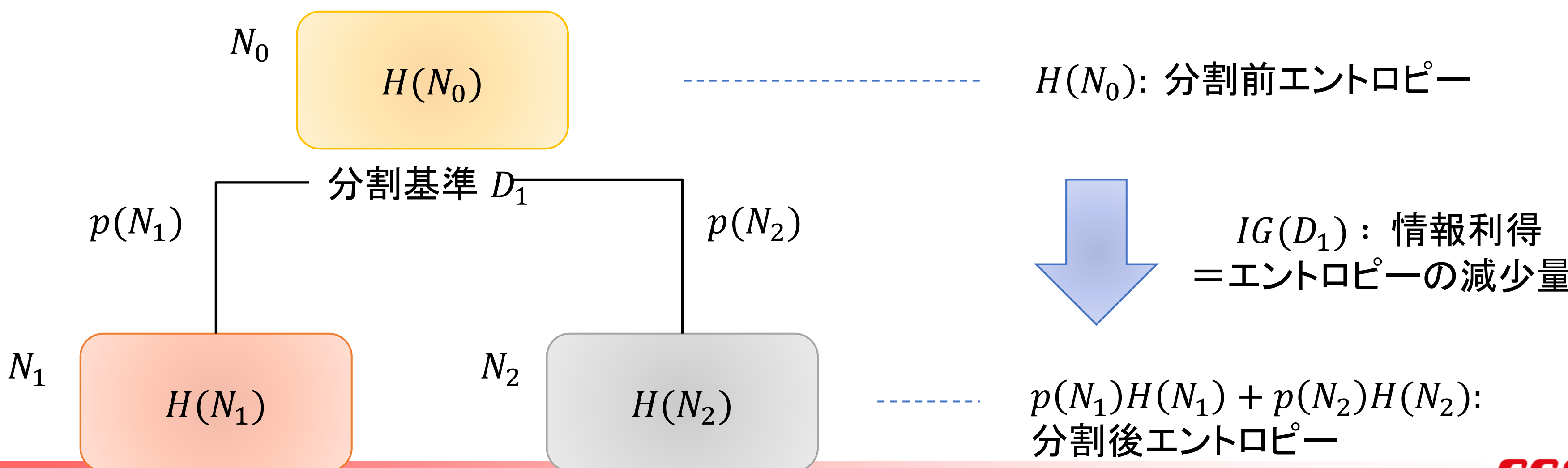
クラスの混在の程度が高いほど
エントロピーが大きい

情報利得

- **情報利得**:分割によるエントロピーの減少量
  - ある分割D_1 によりノードN_0を2つのノード N_1, N_2に分割した場合の情報利得

$$IG(D_1) = \underbrace{H(N_0)}_{\text{分割前エントロピー}} - \underbrace{\{ p(N_1)H(N_1) + p(N_2)H(N_2) \}}_{\text{分割後エントロピー}}$$

  - H:エントロピー
  - p:分割比率($p(N_1) + p(N_2) = 1$)



ある分割D1による情報利得の計算例

$$H(N_0) = -\frac{1}{3}\log_2\frac{1}{3} - \frac{1}{3}\log_2\frac{1}{3} - \frac{1}{3}\log_2\frac{1}{3} = \log_2 3 = 1.584$$

分割D1:
Petal.Length < 2.5

N_0
setosa: 50
versicolor: 50
virginica: 50

Petal.Length < 2.5

$\frac{50}{150}$

$\frac{100}{150}$

N_1
setosa: 50
versicolor: 0
virginica: 0

N_2
setosa: 0
versicolor: 50
virginica: 50

$$H(N_1) = 0$$

$$H(N_2) = -\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{2}\log_2\frac{1}{2} = \log_2 2 = 1.0$$

情報利得

$$IG(D_1) = H(N_0) - \left\{\frac{50}{150} * H(N_1) + \frac{100}{150} * H(N_2)\right\} = 1.584 - 0.666 = 0.918 \text{ bit}$$

分割前エントロピー

分割後エントロピー

ある分割D2による情報利得の計算例

分割D2:
Sepal.Length < 5.8

$$H(N_0) = -\frac{1}{3}\log_2\frac{1}{3} - \frac{1}{3}\log_2\frac{1}{3} - \frac{1}{3}\log_2\frac{1}{3} = \log_2 3 = 1.584$$

N_0
setosa: 50
versicolor: 50
virginica: 50

Sepal.Length < 5.8

$\frac{73}{150}$ $\frac{77}{150}$

N_1
setosa: 49
versicolor: 21
virginica: 3

N_2
setosa: 1
versicolor: 29
virginica: 47

$$H(N_1) = -\frac{49}{73}\log_2\frac{49}{73} - \frac{21}{73}\log_2\frac{21}{73} - \frac{3}{73}\log_2\frac{3}{73} = 1.092$$

$$H(N_2) = -\frac{1}{77}\log_2\frac{1}{77} - \frac{29}{77}\log_2\frac{29}{77} - \frac{47}{77}\log_2\frac{47}{77} = 1.046$$

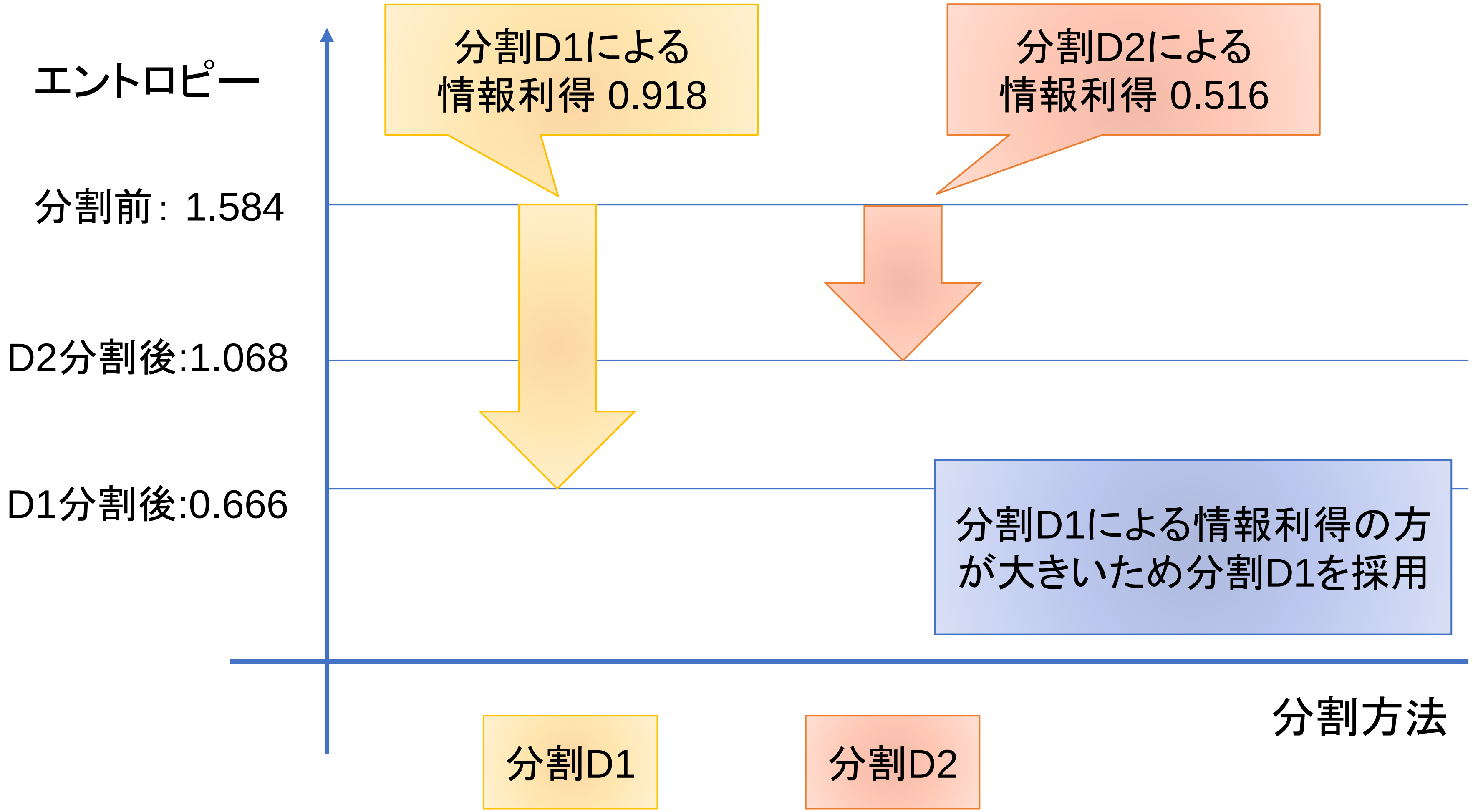
$$IG(D_2) = H(N_0) - \left\{\frac{73}{150} * H(N_1) + \frac{77}{150} * H(N_2)\right\} = 1.584 - 1.068 = 0.516\text{bit}$$

情報利得

分割前エントロピー

分割後エントロピー

分割D1と分割D2の情報利得の比較



決定木のメリット

分析結果の解釈や実装が容易

特徴量の正規化・標準化が不要

複数の決定木を組み合わせることで、ランダムフォレストなどの強力なアルゴリズムに発展させることが可能

ランダムフォレスト

ランダムサンプリングされた訓練データによって学習した多数の決定木を使用する方法

学習

1. 学習に用いるデータや特徴量をランダムに抽出する
2. 抽出したデータと特徴量で決定木を生成する
3. 1．と2．を繰り返す

予測

– 予測したい事例について、個々の決定木が予測した各クラスへの所属確率を平均し、確率が高いと判断されるクラスを予測結果とする

実装

– RのrandomForestパッケージなど

演習：教師あり学習（分類）演習ガイド参照

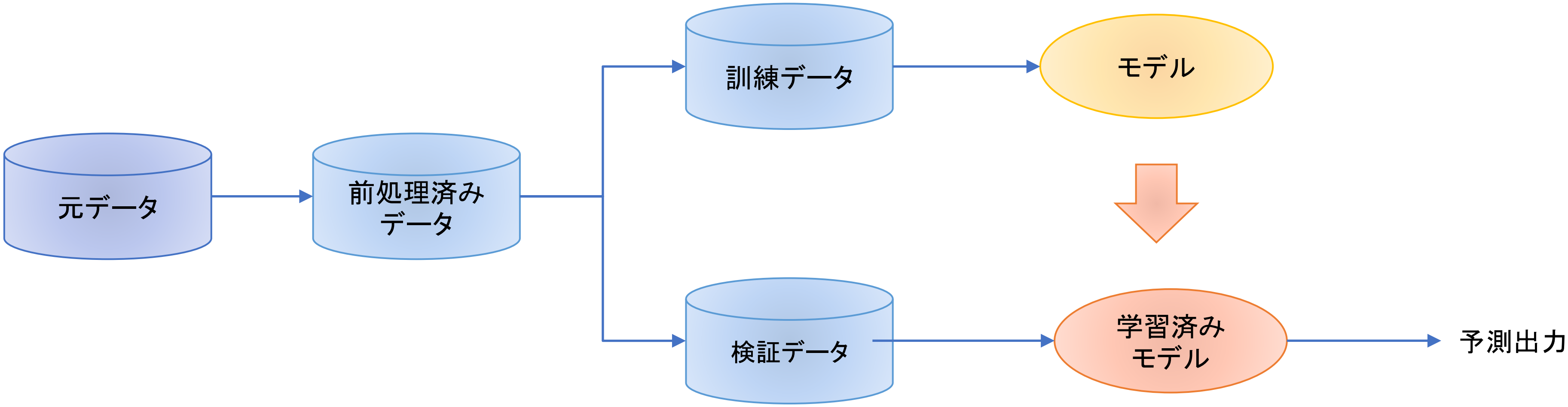
機械学習モデル構築（演習）手順

0. 前処理

1. データ分割

2. モデル（アルゴリズム）選択

3. 学習：訓練データ（入出力の組）をモデルに与えてパラメータ調整



4. 予測：検証データ（入力）を学習済みモデルに与えて予測出力

5. 評価：予測出力と検証データ（出力）を比較して予測精度を評価

演習：Wisconsin乳がんデータセットを利用

- 腫瘍細胞の30の特徴量から良性・悪性の診断を行う機械学習モデルを構築
- k近傍法か決定木を採用
- 必要なら正規化・標準化等の前処理を実施
- 全569事例のデータを訓練データ469事例、検証データ100事例に分割
- 訓練データでモデルを訓練
- 検証データを学習済みモデルに与え、その予測精度を評価

RとR Studio

RとR Studio

R

- 統計解析・可視化のための言語・開発実行環境
- オープンソース・フリーソフトウェアであり、Windows、MAC、Linux等様々なOSで利用可能
- 機械学習のためのライブラリも数多く提供されており、少ないコマンドで簡単にインストール・利用可能

R studio

- Rのための統合開発環境
- コンソールや高機能なエディタ、描画やコマンド履歴、デバッグ、ワークスペース管理のためのツールなどを備える
- クロスプラットフォームであり、各種OSで利用可能

R・Rtoolsのインストール

https://cran.r-project.org/ にアクセスし、Download and Install Rから自分のOSを選択

- Windowsの場合はbaseとRtools（recommended）をそれぞれダウンロードし、exeファイルを実行してインストール
- Rtoolsインストール中に”Add rtools to system PATH”に✓
  - Rtoolsはソースで配布されているライブラリ等をコンパイルして利用する場面で必要

R Studioのインストールと起動

https://rstudio.com/にアクセスし、DOWNLOADからRStudio Desktop (Free)を選択

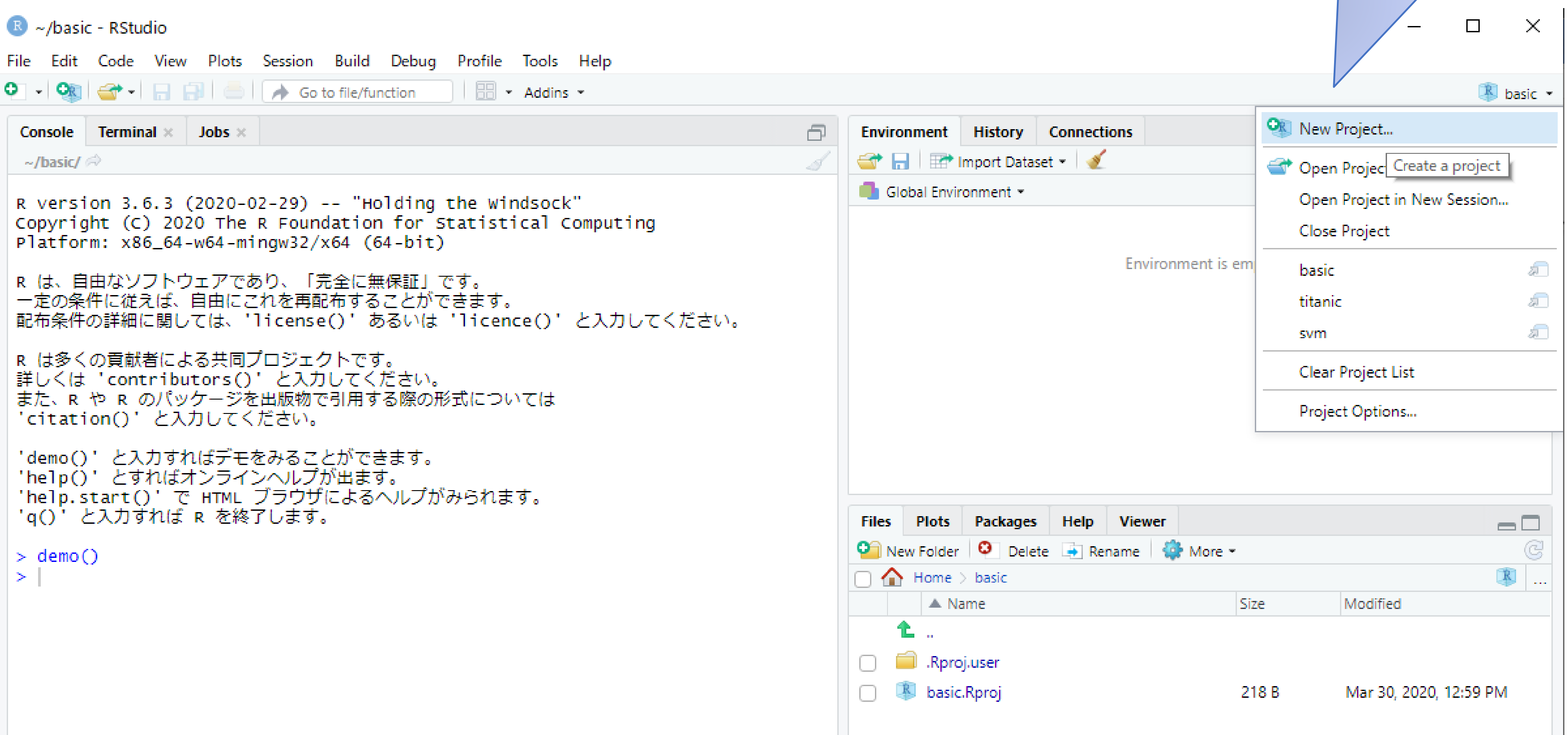
- Windowsの場合はDOWNLOAD RSTUDIO FOR WINDOWSからexeファイルをダウンロード・実行してインストール

R をクリックしてR Studioを起動

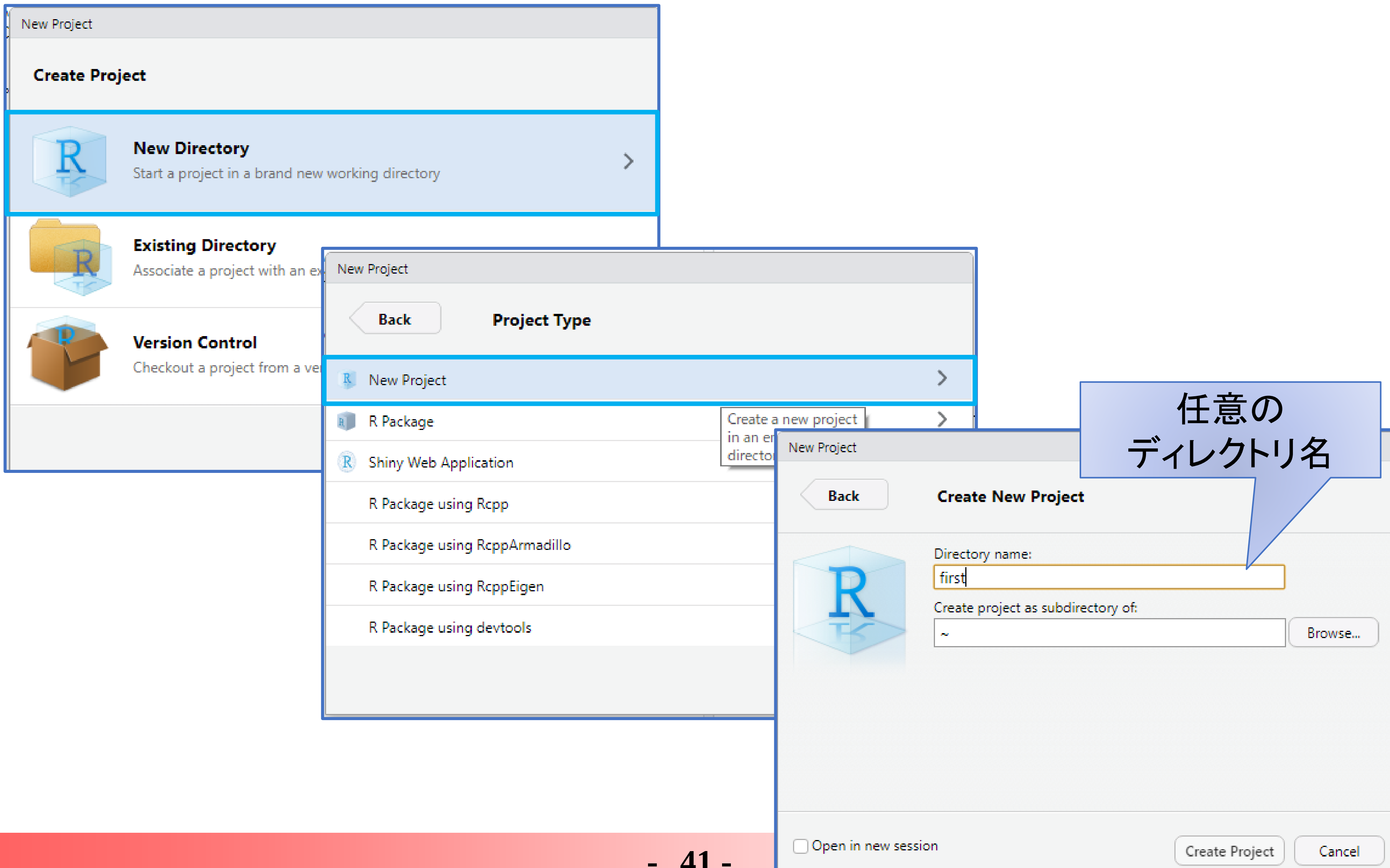
プロジェクトの概念：データやRのコードをまとめて管理する

R Studio起動後、右上からNew Project

New Project



New Directory ⇒ New Project ⇒ 任意のディレクトリ名

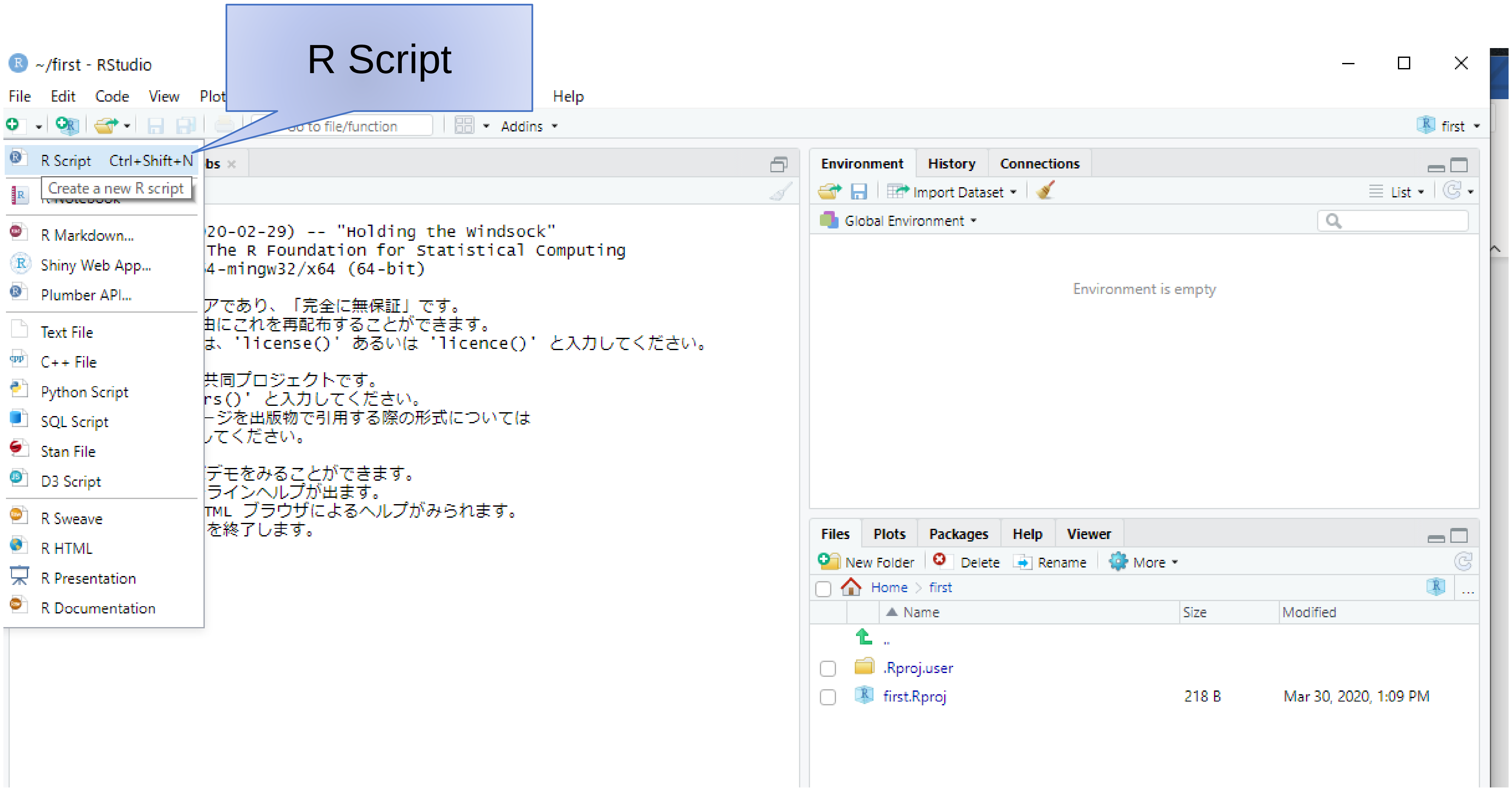


R Studioのペインと概要

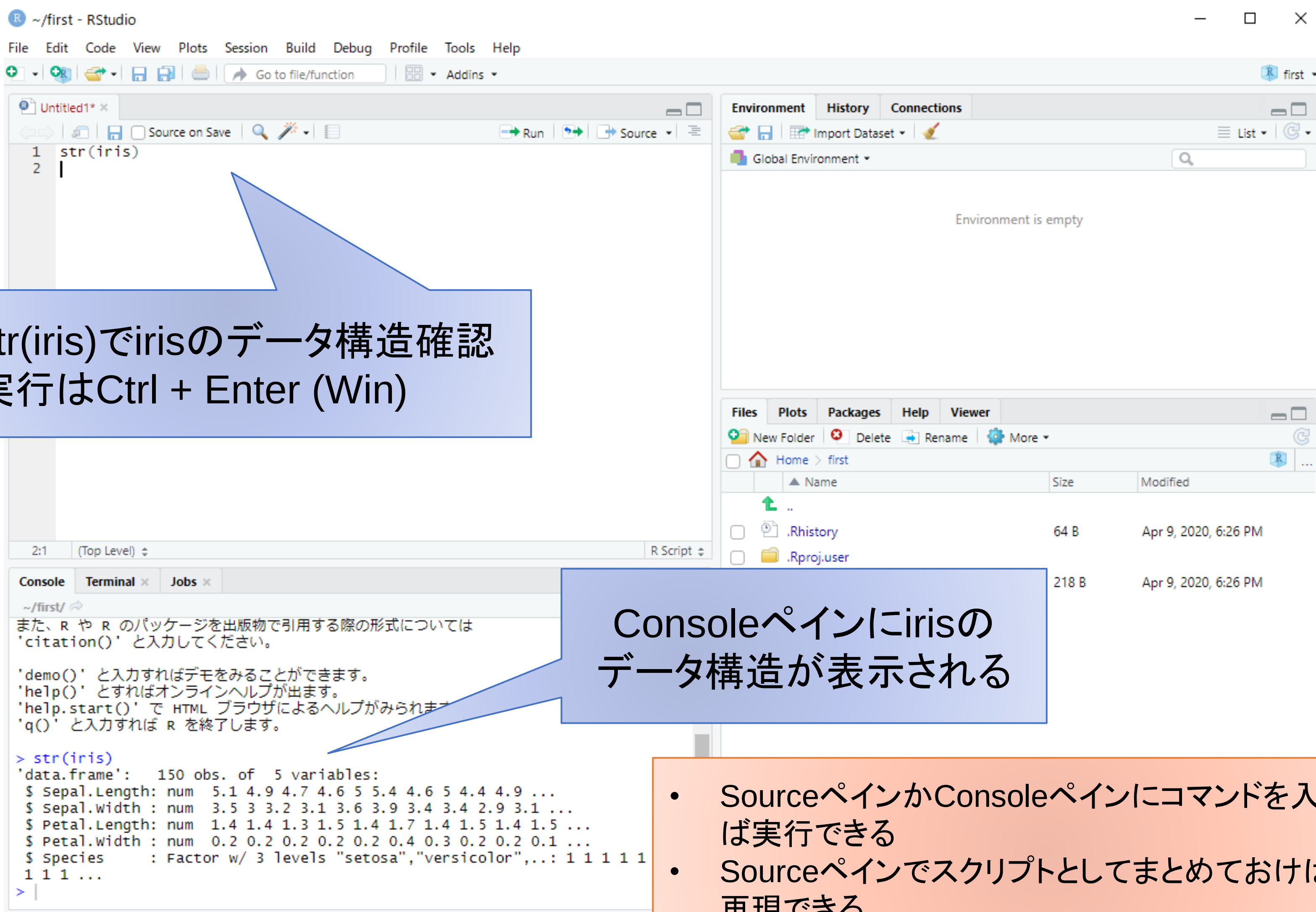
Sourceペインか Consoleペインに
コマンド入力すれば実行できる

ペイン名	ショートカット	概要
Source	Ctrl + 1	スクリプトやマークダウンの編集・表示
Console	Ctrl + 2	Sourceペインに書かれたコマンドの実行。直接コマンドを入力して実行
Terminal	Shift + Alt + T	R studio内でターミナル操作するためのペイン
Help	Ctrl + 3	関数やパッケージのヘルプ
History	Ctrl + 4	コマンド履歴
Files	Ctrl + 5	ワーキングディレクトリにあるファイルを表示
Plots	Ctrl + 6	図の描画
Packages	Ctrl + 7	インストールされているパッケージの表示
Environment	Ctrl + 8	読み込んだデータセットや作成した変数など、現在のワークスペースにある環境を表示

左上からR Scriptを選択



ソースペインにコマンド入力



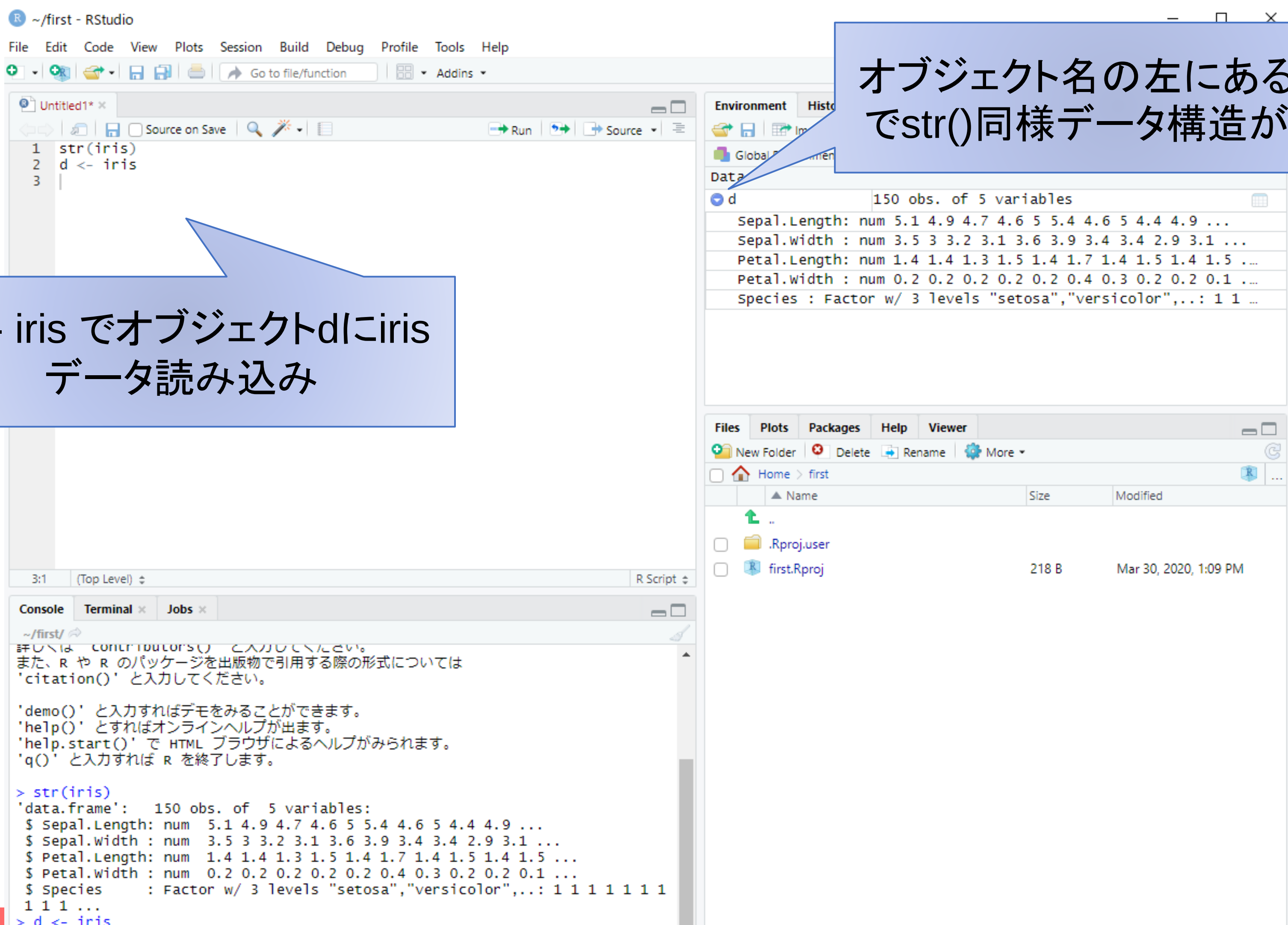
- str(iris)でirisのデータ構造確認
- 実行はCtrl + Enter (Win)

Consoleペインにirisの
データ構造が表示される

- SourceペインかConsoleペインにコマンドを入力すれば実行できる
- Sourceペインでスクリプトとしてまとめておけば保存・再現できる

CSAJ

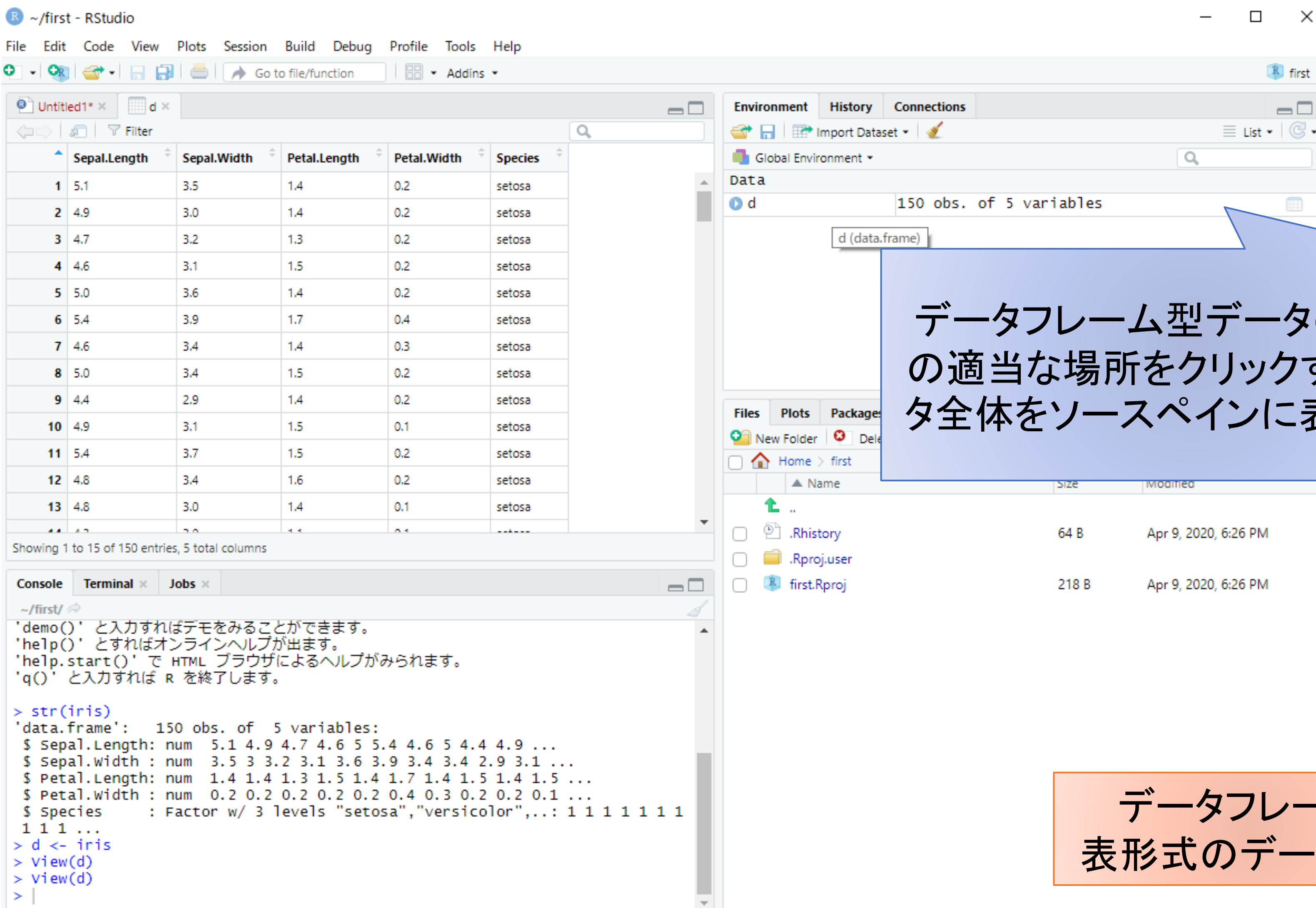
データ読込み･データ構造確認



オブジェクト名の左にある矢印ボタンでstr()同様データ構造が確認できる

d <- iris でオブジェクトdにirisデータ読み込み

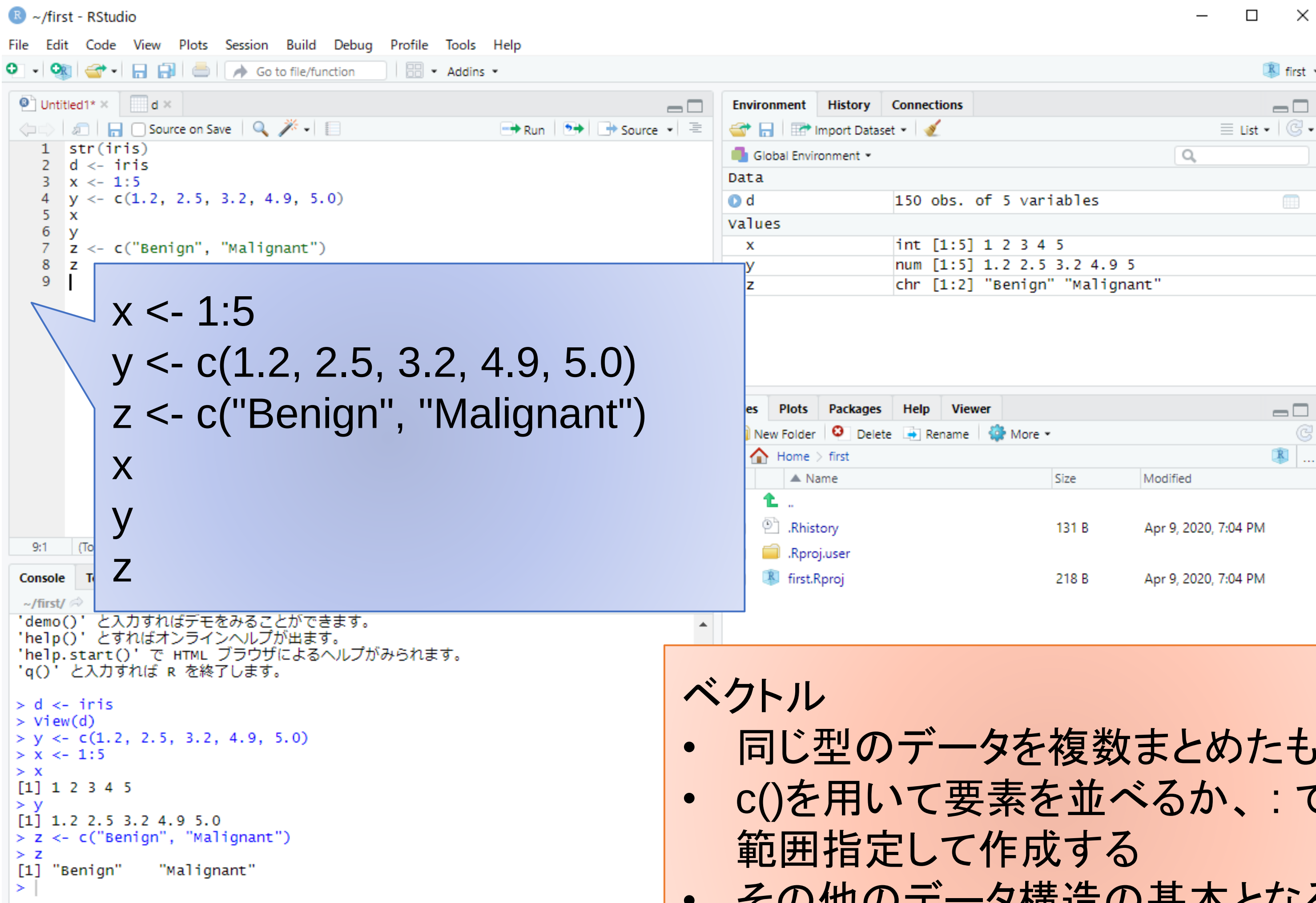
データフレーム型データの表示



データフレーム型データの場合行の適当な場所をクリックするとデータ全体をソースペインに表示できる

データフレーム：
表形式のデータ構造

データ構造：ベクトル



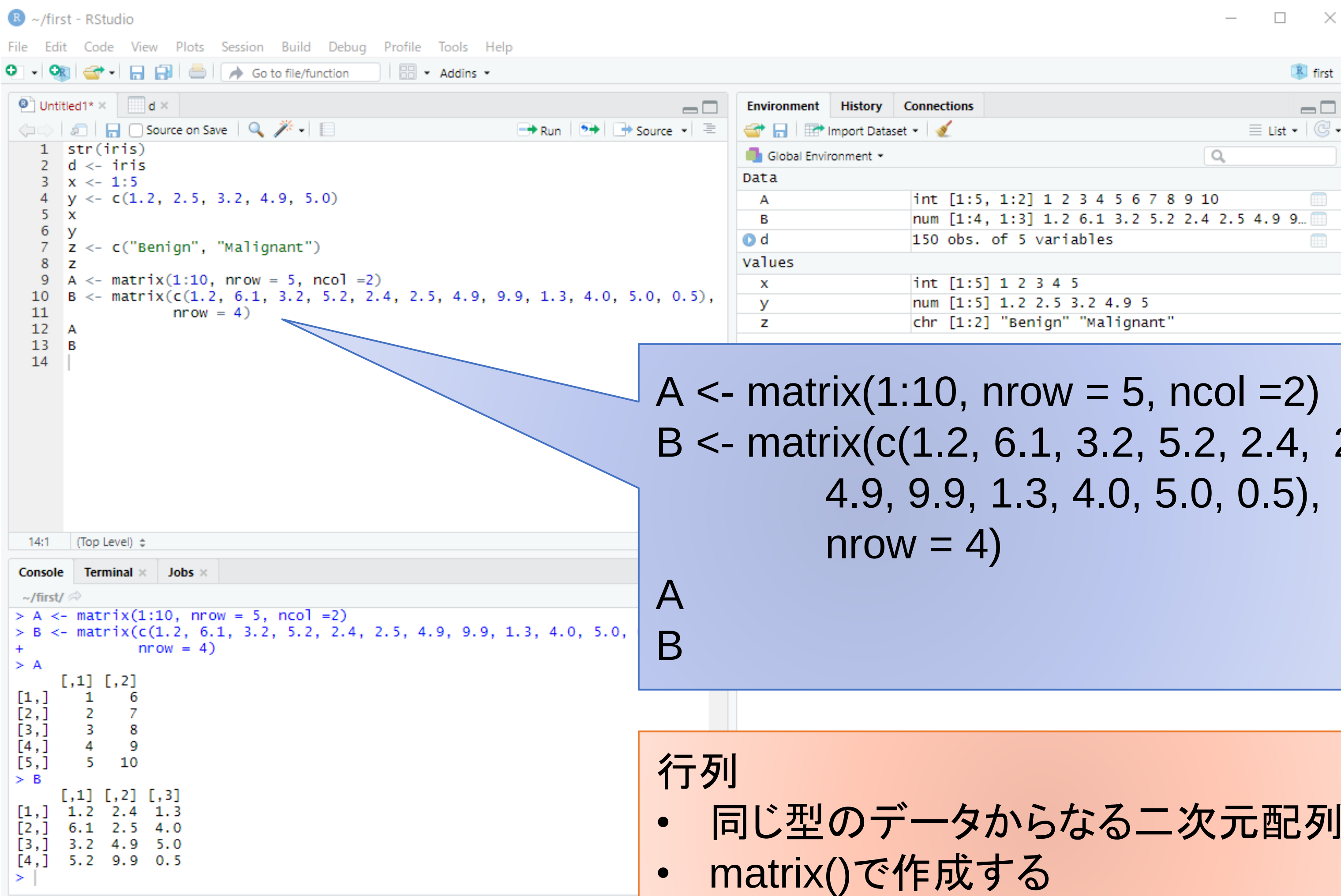

```
x <- 1:5
y <- c(1.2, 2.5, 3.2, 4.9, 5.0)
z <- c("Benign", "Malignant")
x
y
z
```

ベクトル

- 同じ型のデータを複数まとめたもの
- c()を用いて要素を並べるか、：で数値範囲指定して作成する
- その他のデータ構造の基本となる

CSAJ

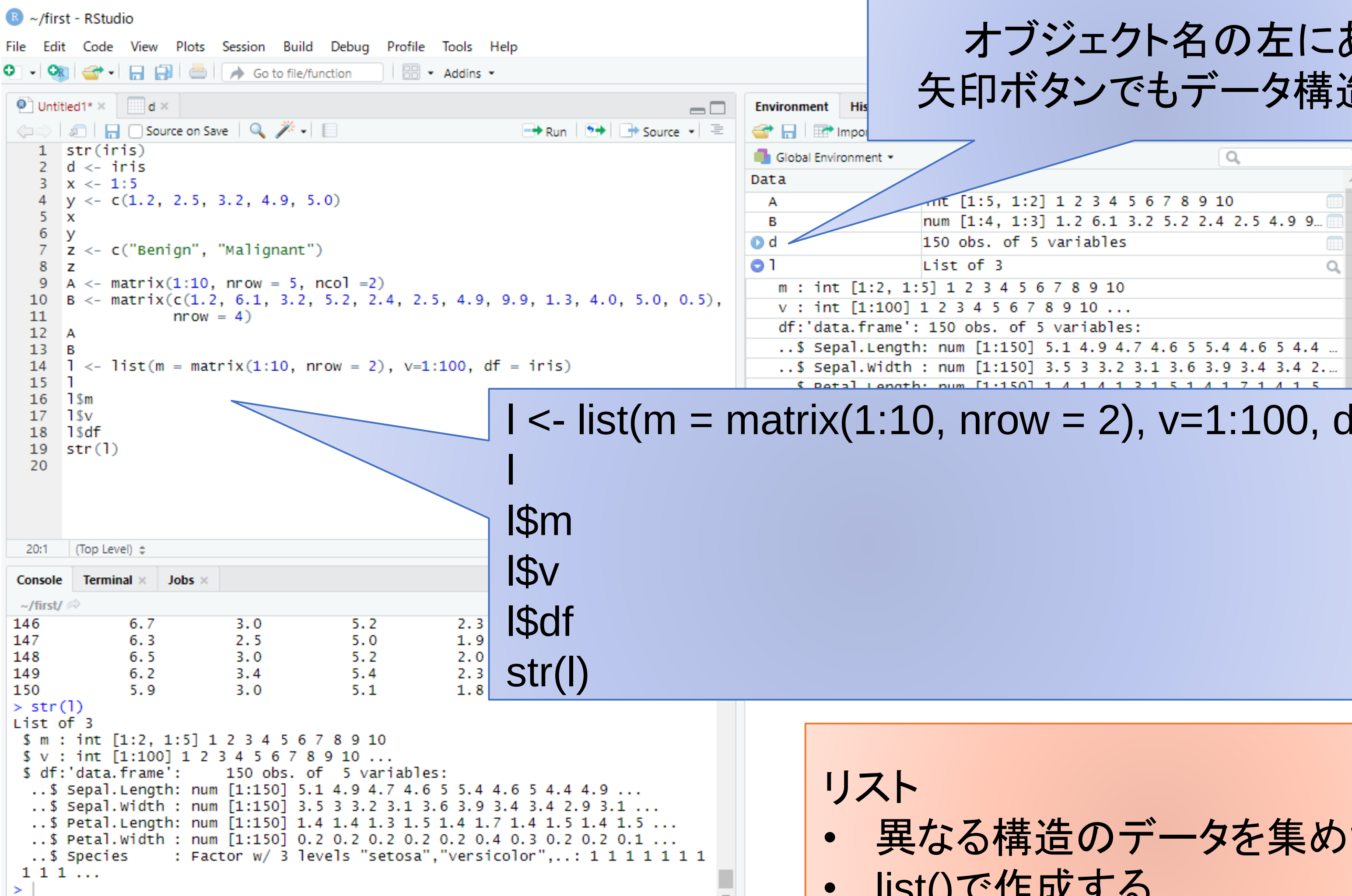
データ構造：行列




```
A <- matrix(1:10, nrow = 5, ncol =2)
B <- matrix(c(1.2, 6.1, 3.2, 5.2, 2.4,  2.5,
          4.9, 9.9, 1.3, 4.0, 5.0, 0.5),
          nrow = 4)
A
B
```

行列
- 同じ型のデータからなる二次元配列
- matrix()で作成する

データ構造:リスト



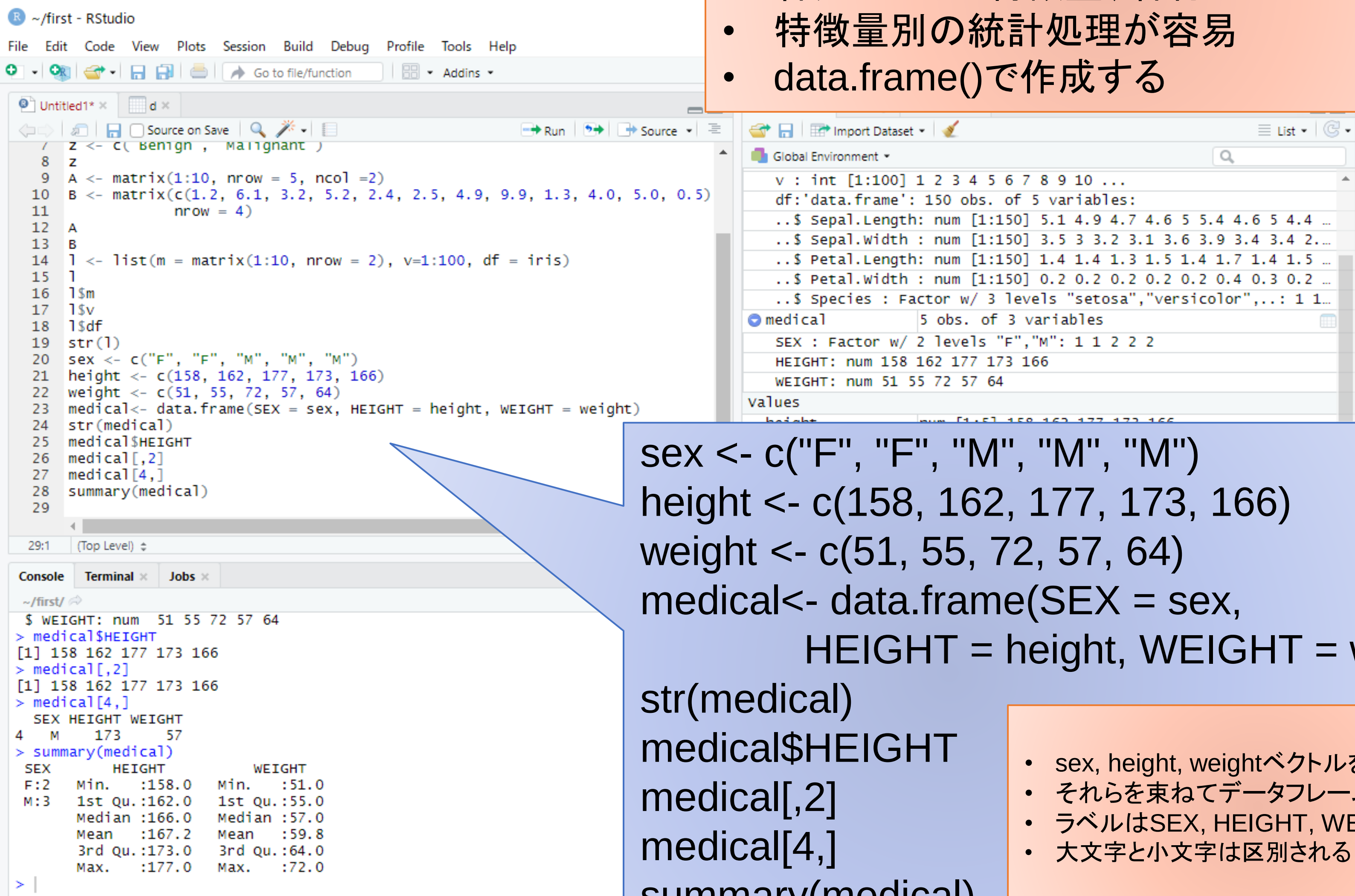
オブジェクト名の左にある
矢印ボタンでもデータ構造確認

```
l <- list(m = matrix(1:10, nrow = 2), v=1:100, df = iris)
l
l$m
l$v
l$df
str(l)
```

リスト
- 異なる構造のデータを集めたもの
- list()で作成する

データフレーム

データフレーム
- リストの一種
- 各列は1つの特徴量、各行は1つの観測値
- 特徴量別の統計処理が容易
- data.frame()で作成する




```
sex <- c("F", "F", "M", "M", "M")
height <- c(158, 162, 177, 173, 166)
weight <- c(51, 55, 72, 57, 64)
medical<- data.frame(SEX = sex,
              HEIGHT = height, WEIGHT = weight)
str(medical)
medical$HEIGHT
medical[,2]
medical[4,]
summary(medical)
```

- sex, height, weightベクトルを作成
- それらを束ねてデータフレーム作成
- ラベルはSEX, HEIGHT, WEIGHTとした
- 大文字と小文字は区別される

教師あり学習（分類）のためのデータ準備

R studioで新規プロジェクトとスクリプト作成

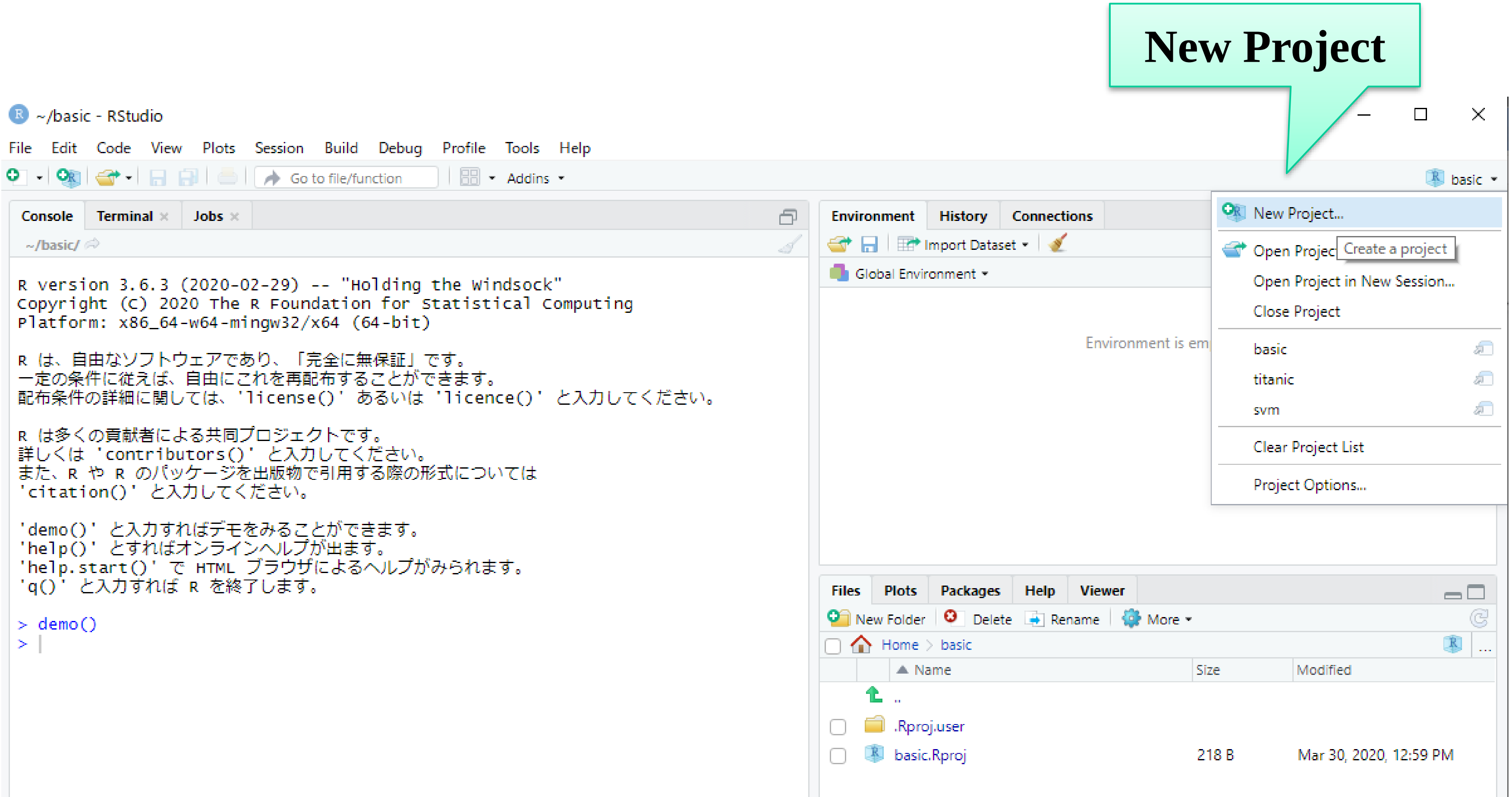
プロジェクト名は任意だが一応 wbcd とする

- Wisconsin Breast Cancer Diagnosticの意味

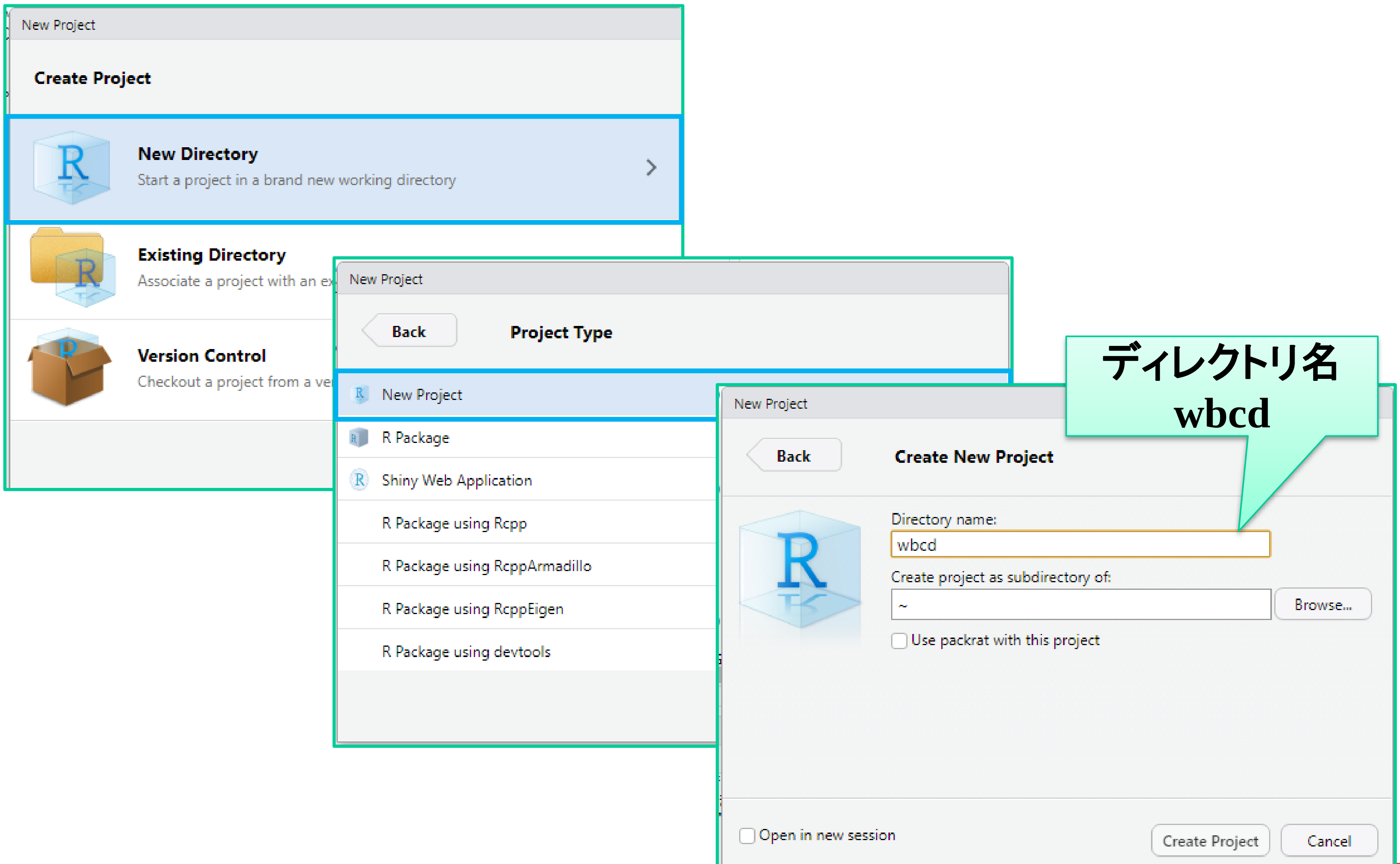
プロジェクトを立ち上げたら、新しいスクリプトを作成する

「新規プロジェクト作成」「スクリプト作成」の手順は覚えていると思うが、忘れた人は次のページを参照のこと

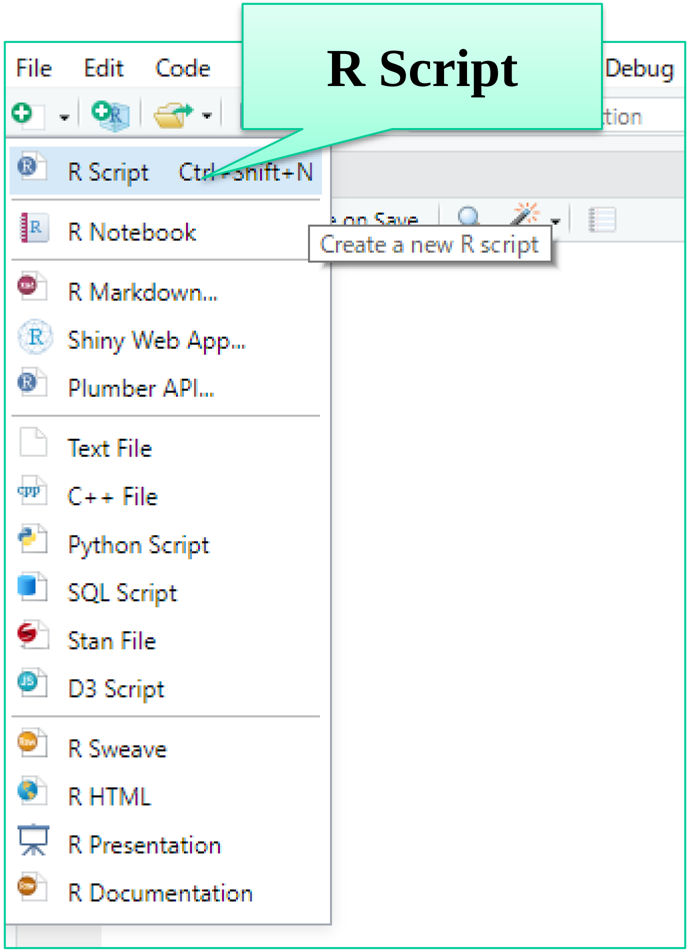
R Studio起動後、右上からNew Project



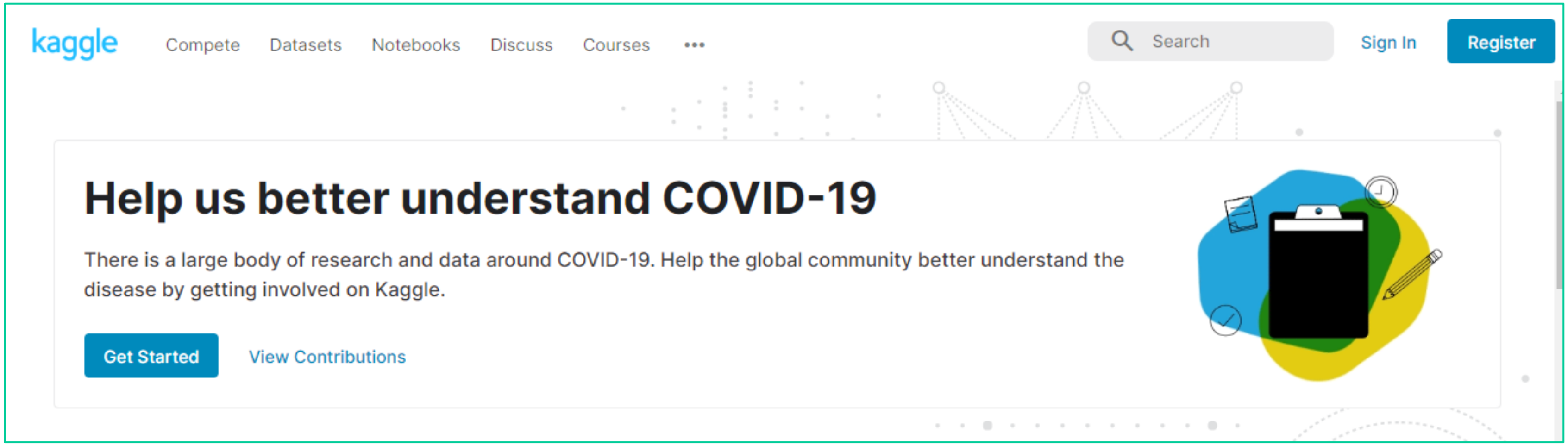
New Directory ⇒ New Project ⇒ ディレクトリ名 wbcd



左上からR Scriptを選択



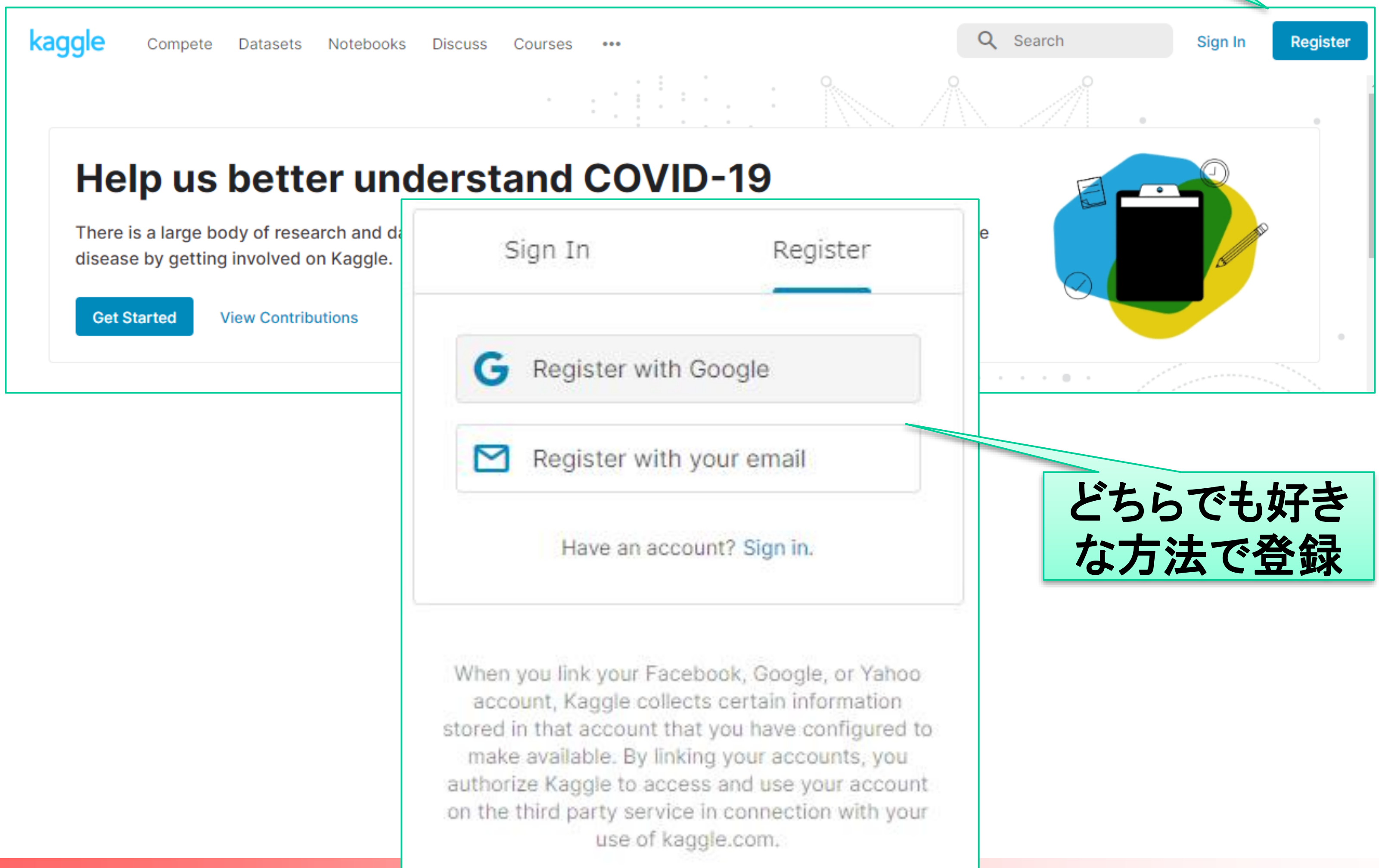
kaggleの登録



- 機械学習・データサイエンスに関わる人が集まるコミュニティ
- 企業等から実データが提供されたうえで賞金付きのコンペ（競技会）も実施される
- 提供されている実データを機械学習の学習のために利用可能
- https://www.kaggle.com/

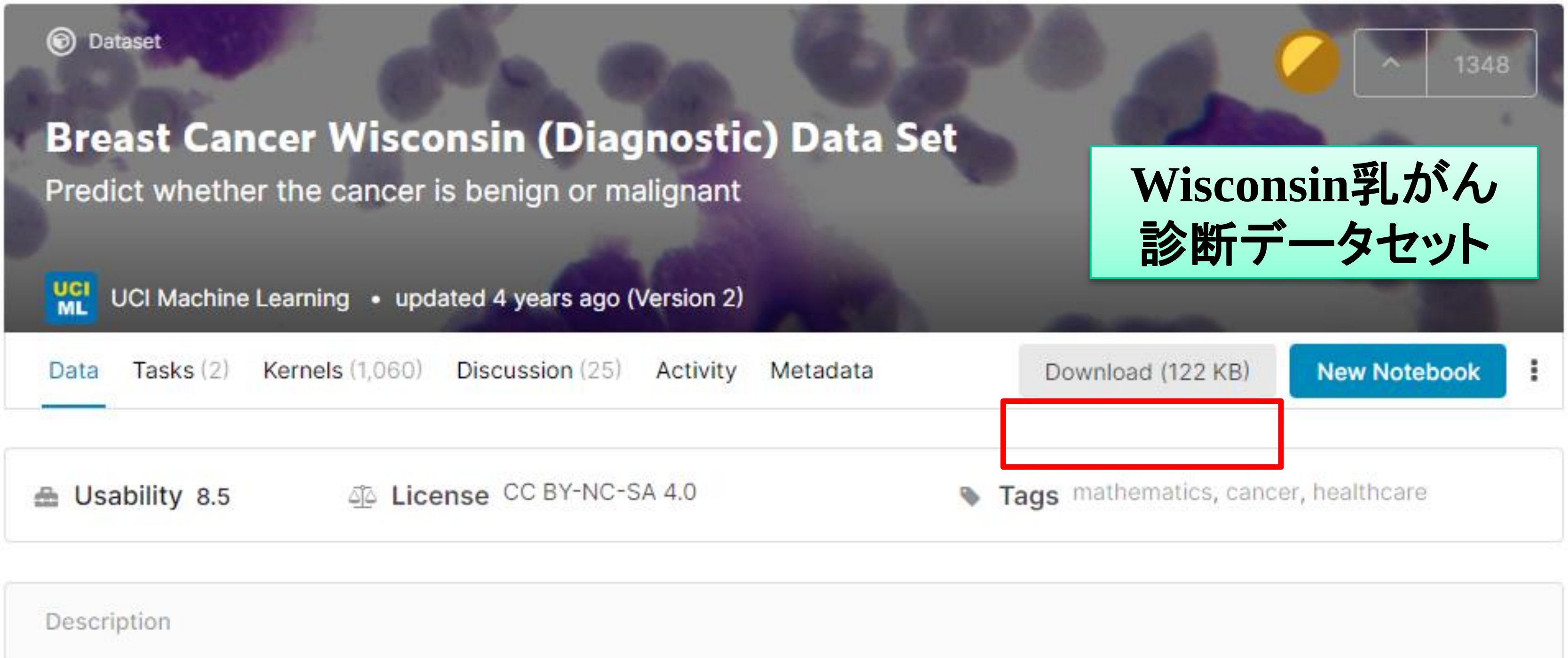
kaggleの登録

Register



どちらでも好きな方法で登録

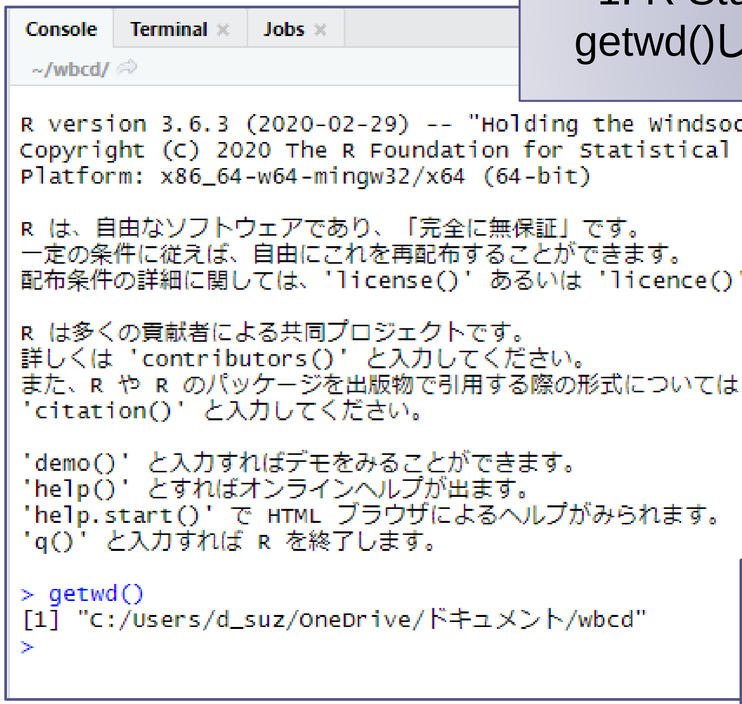
データのダウンロード



- https://www.kaggle.com/uciml/breast-cancer-wisconsin-data のDownloadから圧縮ファイル(.zip)をダウンロード
- ダウンロードしたzipファイルを展開する(zipファイルを右クリックして「すべて展開」を選択)
- 展開後のフォルダの中に"data.csv"があることを確認

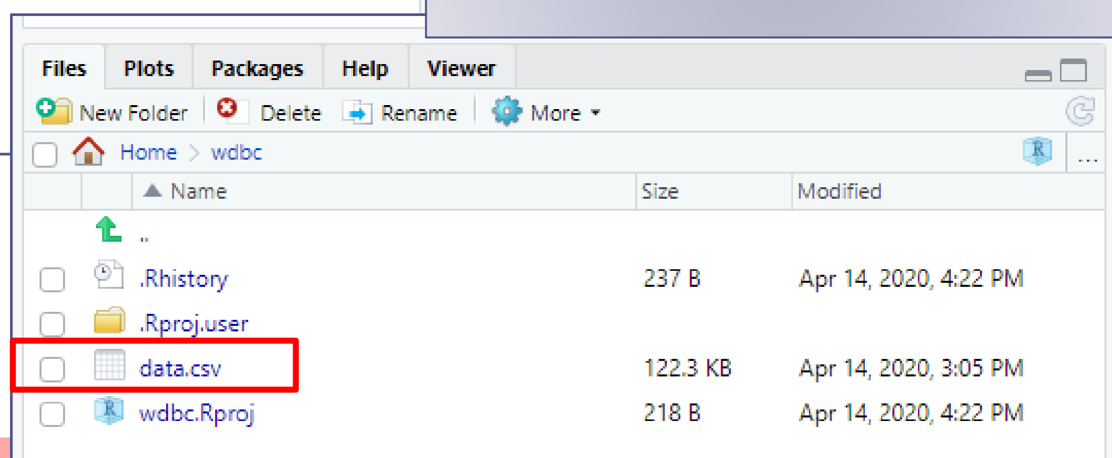
データの準備

1. R Studio のConsoleペインでgetwd()して作業ディレクトリを確認

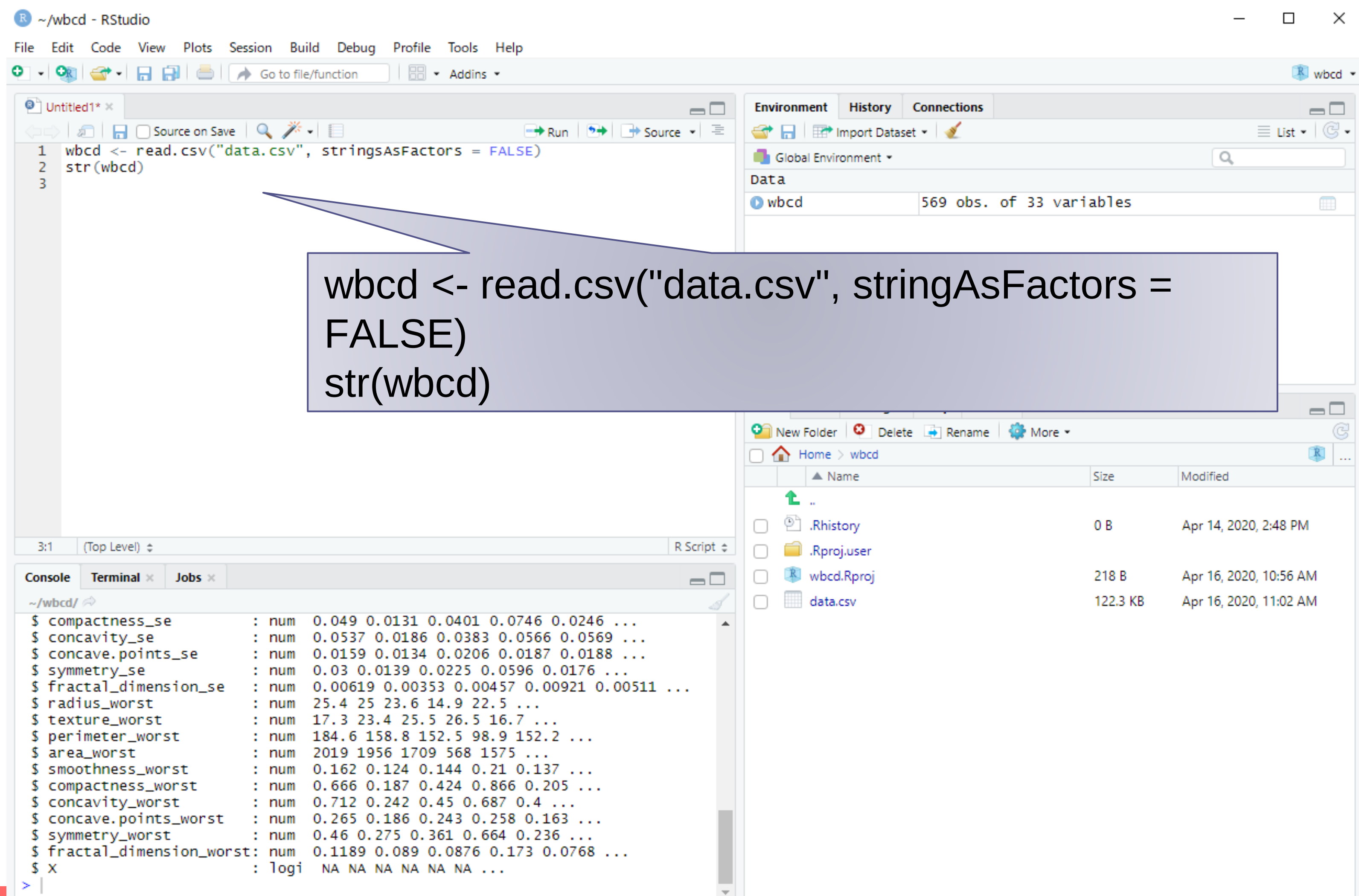


2. 先ほど用意した"data.csv"を作業ディレクトリに移動

3. R Studio のFilesペインでdata.csvの存在が確認できればOK



データの読込みとデータ構造確認



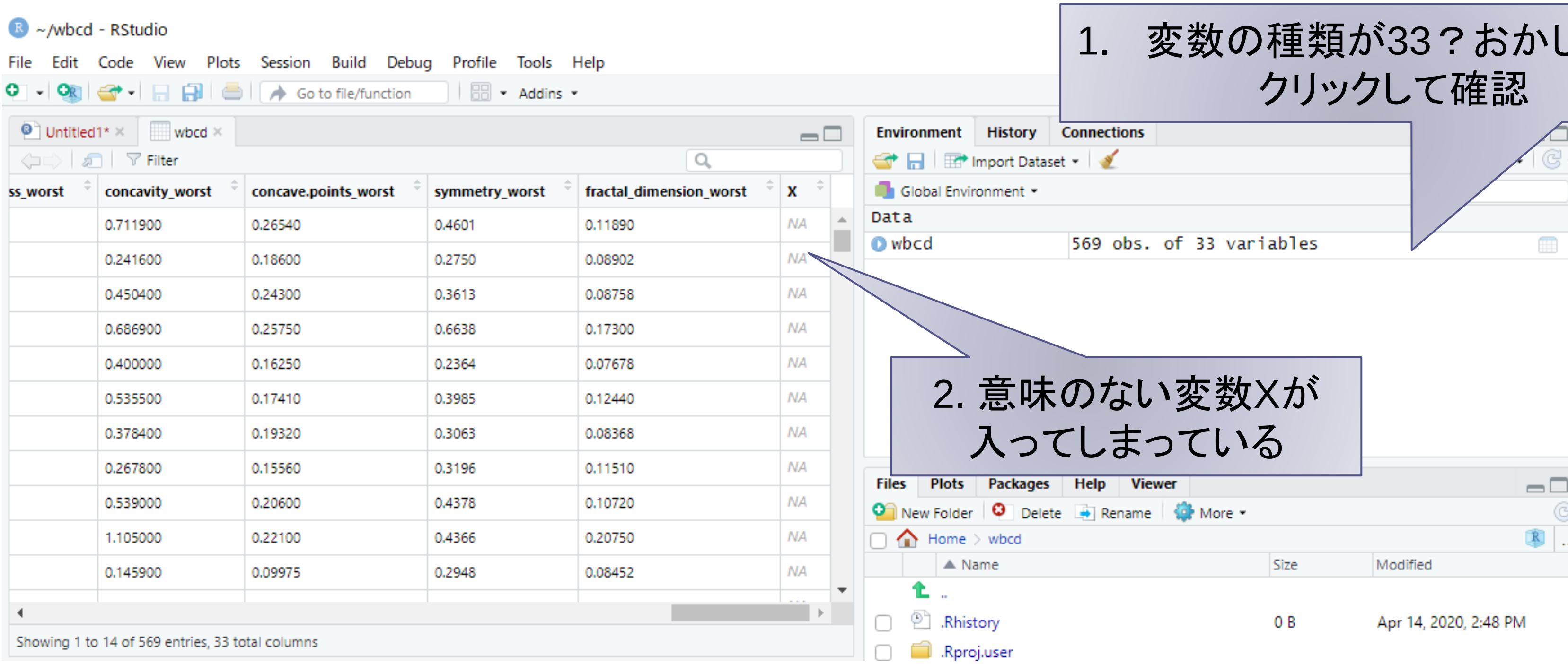

```
wbcd <- read.csv("data.csv", stringsAsFactors = FALSE)
str(wbcd)
```

569事例、32変数
1) ID, 2) 診断結果（M=悪性、B=良性）, 3-32) 細胞核の10の特徴量各々について平均、標準偏差、最大値
欠損値なし
357 良性、212 悪性

データの意味

- Number of instances: 569
- Number of attributes: 32 (ID, diagnosis, 30 real-valued input features)
- Attribute information
  - 1) ID number
  - 2) Diagnosis (M = malignant, B = benign)
  - 3-32)Ten real-valued features are computed for each cell nucleus:
    - a) radius (mean of distances from center to points on the perimeter)
    - b) texture (standard deviation of gray-scale values)
    - c) perimeter
    - d) area
    - e) smoothness (local variation in radius lengths)
    - f) compactness (perimeter^2 / area - 1.0)
    - g) concavity (severity of concave portions of the contour)
    - h) concave points (number of concave portions of the contour)
    - i) symmetry
    - j) fractal dimension ("coastline approximation" - 1)
    - The mean, standard error, and "worst" or largest (mean of the three largest values) of these features were computed for each image, resulting in 30 features. For instance, field 3 is Mean Radius, field13 is Radius SE, field 23 is Worst Radius.
    - All feature values are recoded with four significant digits.
- Missing attribute values: none
- Class distribution: 357 benign, 212 malignant

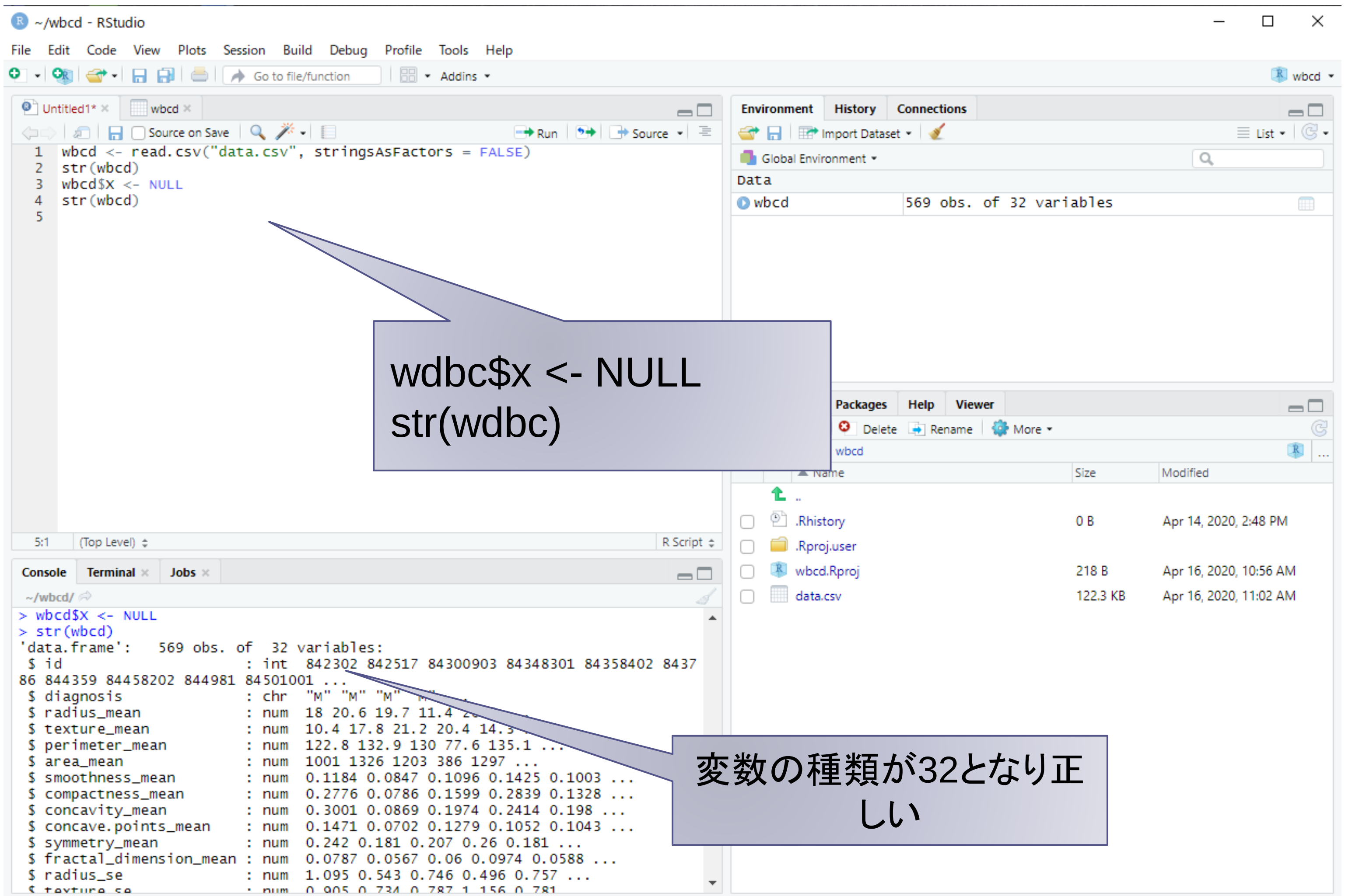
データ確認と不要な変数の発見



1. 変数の種類が33？おかしい.
クリックして確認

2. 意味のない変数Xが
入ってしまっている

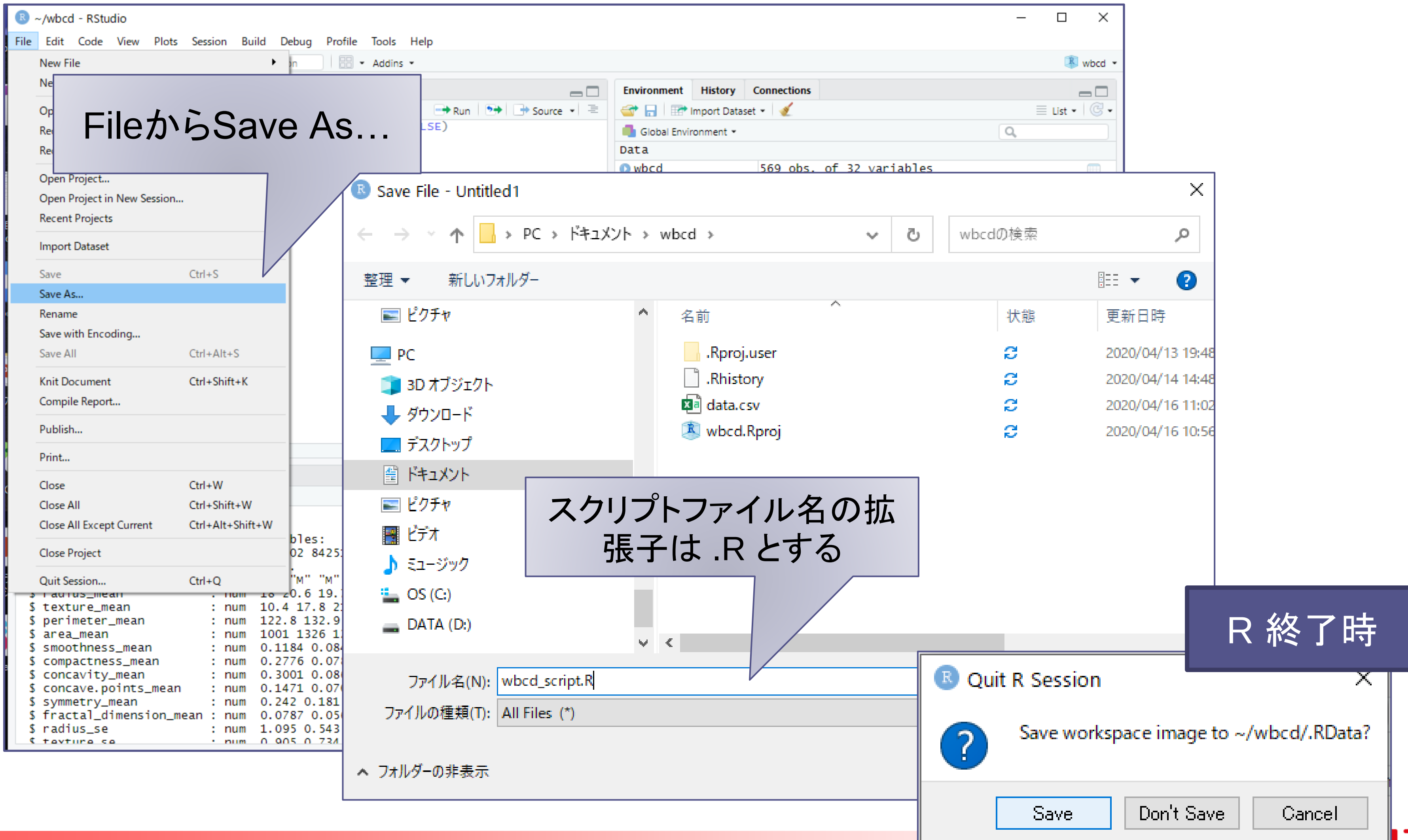
不要な変数xの削除



wdbc$x <- NULL
str(wdbc)

変数の種類が32となり正しい

スクリプトの保存・ワークスペースの保存



教師あり学習（回帰）

教師あり学習

- 正解の用意された問題集（教師データ）がある
- 教師は正解の導き方を詳しく説明してくれない
- 学習者（コンピュータ）は問題と正解の関係を推測する
- これまで見たことの無い新しい類似の問題に対して正解を導けるようになることを目指す

- 教師あり学習の概要
  - 入力とそれに対応する正解出力の組からなる教師データを用いる
  - 教師データに基づいて入力と出力の関係を表現するモデルを構築する
  - 新しい入力に対して精度の高い予測出力ができることを目指す
  - 出力が連続値の場合「回帰」モデル、出力がカテゴリの場合「分類」モデルを構築する

- ex. 不動産価格の予測（回帰）、検査結果に基づく陽性判定（分類）

回帰（Regression）

教師あり学習
入力データから連続値を予測する
– ex. 都市の電力消費量や住宅価格など

回帰のアルゴリズム
– 線形回帰（通常最小二乗法）
– Ridge回帰、Lasso回帰 、Elastic Net
– SVR(Support Vector Regression)

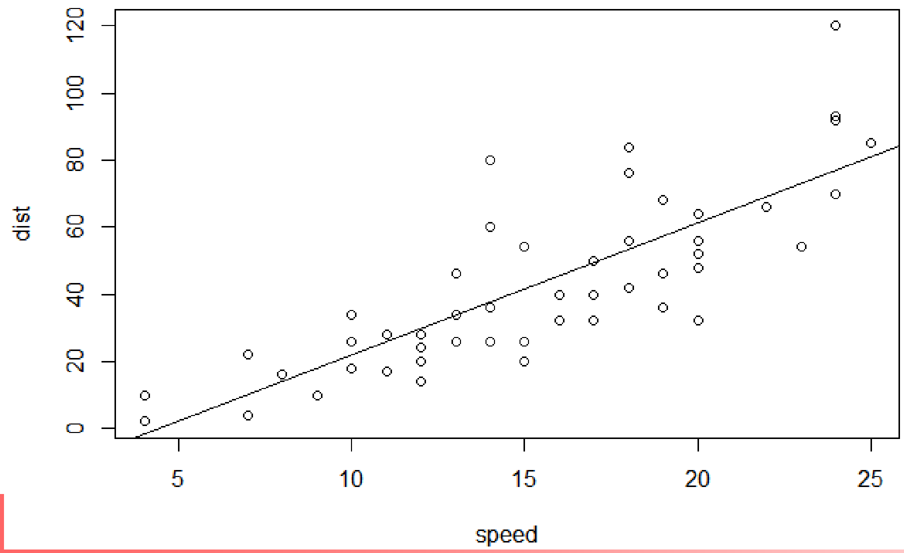
線形回帰

教師あり学習（回帰）

線形回帰の概要と例

説明変数の線型結合で目的変数を表現するモデル

- 単回帰
  - 説明変数が1つ
  - 目的変数 y，説明変数 x として$y = ax + b$ と表現される. aは回帰係数, bは切片
- 重回帰
  - 説明変数が2つ以上
  - 目的変数 y，説明変数 $x_1, x_2, \cdots, x_d$ として$y = w_0 + w_1 x_1 + w_2 x_2 + \cdots + w_d x_d$ と表現される. $w_1, w_2, \cdots, w_d$は偏回帰係数, w_0は切片



単回帰の例

- R標準データセットcars
- 車速(mph)と停止距離(ft)の組50組
- 車速を説明変数、停止距離を目的変数として単回帰
- 回帰式$y = 3.9\,x\ - 17.6$
- 回帰式は最小二乗法により求まる

※実際は、空走距離は車速に比例するが、制動距離は速度の二乗に比例する

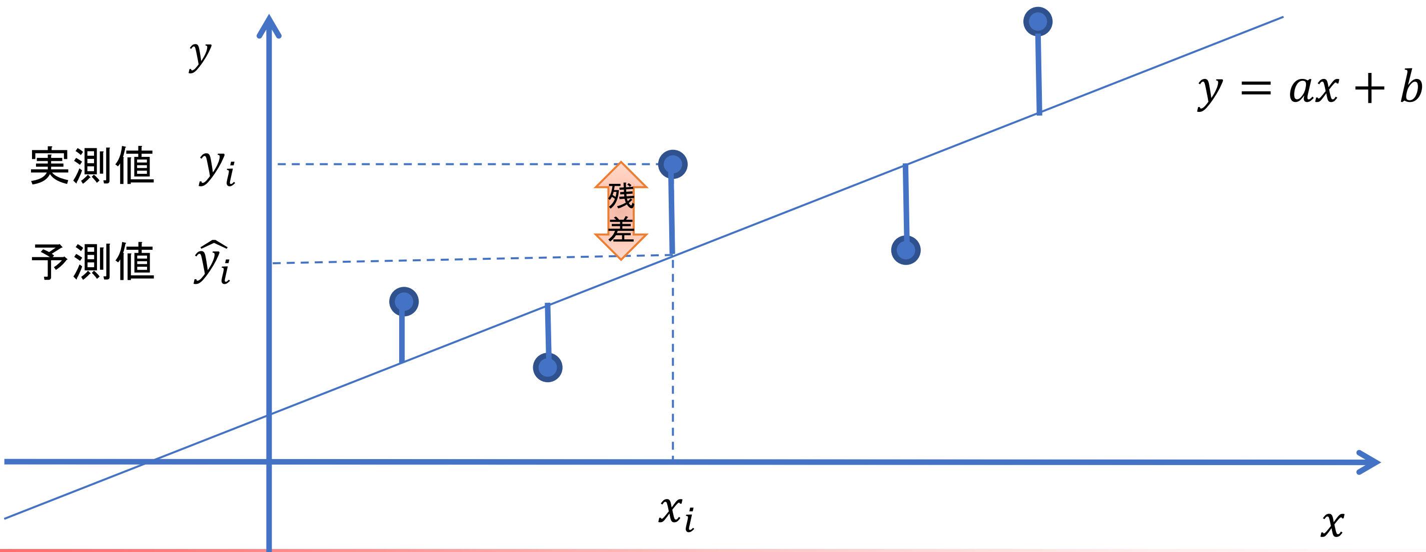
最小二乗法による回帰係数の導出(1/2)

予測値と実測値の差（残差）の平方和を最小にするよう回帰係数を決定する方法

ex. 単回帰式

- データセット $(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n)$に基づき回帰式$y = ax + b$を導出する
- 予測値を$\hat{y}_i = ax_i + b$と書くと、残差平方和は以下の式であらわされる

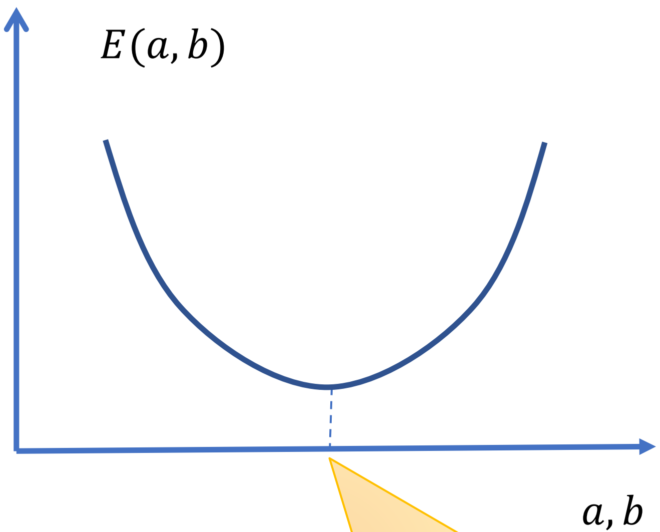
$$E = \sum_{i=1}^{n} (\hat{y}_i - y_i)^2 = \sum_{i=1}^{n} (ax_i + b - y_i)^2$$



最小二乗法による回帰係数の導出(2/2)

$$E=\sum_{i=1}^{n}(\hat{y}_i-y_i)^2=\sum_{i=1}^{n}(ax_i+b-y_i)^2$$

Eはa,bの二変数関数$E(a,b)$と見ることができ、下に凸の二次関数となる



$\frac{\partial E}{\partial a}=0,\frac{\partial E}{\partial b}=0$ なる (a,b)

$\frac{\partial E}{\partial a}=0,\frac{\partial E}{\partial b}=0$ でEは最小となるため、

$$\frac{\partial E}{\partial a}=\sum 2(ax_i+b-y_i)x_i=0$$

$$\frac{\partial E}{\partial b}=\sum 2(ax_i+b-y_i)=0$$

連立方程式を解くと、$\bar{x}=\frac{1}{n}\sum x_i,\ \ \bar{y}=\frac{1}{n}\sum y_i$として

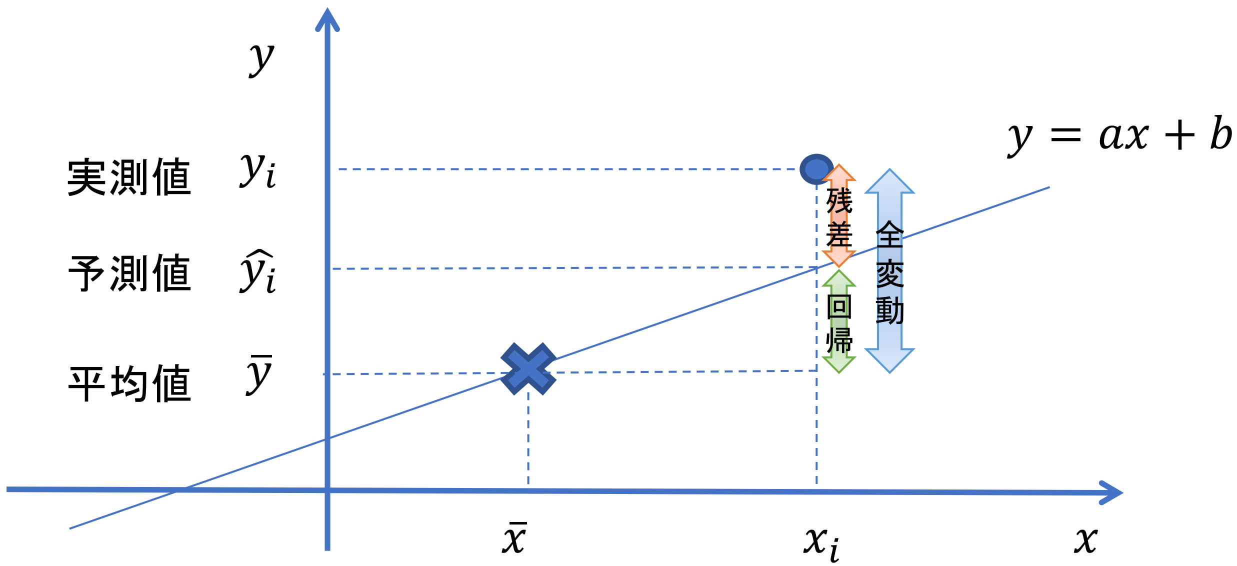
$$a=\frac{\frac{1}{n}\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y})}{\frac{1}{n}\sum_{i=1}^{n}x_i^2-\bar{x}^2}=\frac{\sigma_{xy}}{\sigma_x^2},$$

$$\mathrm{b}=\bar{\mathrm{y}}-a\bar{x}$$

σ_x^2: xの分散
σ_{xy}: xとyの共分散

決定係数

回帰式が全体の変動のうちどの程度を説明できるか示す係数



- 全変動平方和＝回帰変動平方和＋残差変動平方和

$$\sum(y_i - \bar{y})^2 = \sum(\hat{y}_i - \bar{y})^2 + \sum(y_i - \hat{y}_i)^2$$

- 決定係数＝回帰変動平方和／全変動平方和

$$R^2 = \frac{\sum(\hat{y}_i - \bar{y})^2}{\sum(y_i - \bar{y})^2} = 1 - \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i - \bar{y})^2}$$

どの残差も0ならば
決定係数は1になる

重回帰の場合の最小二乗法による回帰係数の導出 (1/3)

重回帰式は

$$y = w_0 + w_1 x_1 + w_2 x_2 + \cdots + w_d x_d$$

$\boldsymbol{x} = (x_1, x_2, \cdots, x_d)^T$ は入力変数、$w_0, w_1, \cdots, w_d$はパラメータ

ここで $\tilde{\boldsymbol{x}} = (1, x_1, x_2, \cdots, x_d)^T, \boldsymbol{w} = (w_0, w_1, \cdots, w_d)^T$ と定義すると

重回帰式は

$$y = \boldsymbol{w}^T \tilde{\boldsymbol{x}}$$

とあらわせる

重回帰の場合の最小二乗法による回帰係数の導出 (2/3)

重回帰式は

$$y = \boldsymbol{w}^T \tilde{\boldsymbol{x}}$$

データ行列

$$\boldsymbol{X} = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1d} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{nd} \end{pmatrix}$$

をもとに

$$\tilde{\boldsymbol{X}} = \begin{pmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1d} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & x_{n2} & \cdots & x_{nd} \end{pmatrix}$$

を定義すると、各サンプル行（1~n）に対して回帰式をあてはめた結果は

$$\begin{pmatrix} w_0 + w_1 x_{11} + & \cdots & + w_d x_{1d} \\ w_0 + w_1 x_{21} + & \ldots & + w_d x_{2d} \\ \vdots & \vdots & \vdots \\ w_0 + w_1 x_{n1} + & \cdots & + w_d x_{nd} \end{pmatrix} = \begin{pmatrix} \boldsymbol{w}^T \tilde{\boldsymbol{x}}_1 \\ \boldsymbol{w}^T \tilde{\boldsymbol{x}}_2 \\ \vdots \\ \boldsymbol{w}^T \tilde{\boldsymbol{x}}_n \end{pmatrix} = \tilde{\boldsymbol{X}} \boldsymbol{w}$$

とあらわすことができる

重回帰の場合の最小二乗法による回帰係数の導出(3/3)

重回帰式を$\boldsymbol{w}$の関数とみて

$$\hat{\boldsymbol{y}}(\boldsymbol{w}) = \widetilde{\boldsymbol{X}}\boldsymbol{w}$$

目的変数yとの差の二乗和を目的関数として、これを最小にするような$\boldsymbol{w}$を求める

$$E(\boldsymbol{w}) = \left|\boldsymbol{y} - \widetilde{\boldsymbol{X}}\boldsymbol{w}\right|^{2} = \left(\boldsymbol{y} - \widetilde{\boldsymbol{X}}\boldsymbol{w}\right)^{\mathrm{T}}\left(\boldsymbol{y} - \widetilde{\boldsymbol{X}}\boldsymbol{w}\right)$$
$$= \boldsymbol{y}^{T}\boldsymbol{y} - \boldsymbol{w}^{T}\widetilde{\boldsymbol{X}}^{T}\boldsymbol{y} - \boldsymbol{y}^{T}\widetilde{\boldsymbol{X}}\boldsymbol{w} + \boldsymbol{w}^{T}\widetilde{\boldsymbol{X}}^{T}\widetilde{\boldsymbol{X}}\boldsymbol{w}$$

目的関数$E(\boldsymbol{w})$の勾配を求めると

$$\nabla E(\boldsymbol{w}) = -2\widetilde{\boldsymbol{X}}^{T}\boldsymbol{y} + 2\widetilde{\boldsymbol{X}}^{T}\widetilde{\boldsymbol{X}}\boldsymbol{w}$$

これを $= 0$とおくことで

$$\widetilde{\boldsymbol{X}}^{T}\boldsymbol{y} = \widetilde{\boldsymbol{X}}^{T}\widetilde{\boldsymbol{X}}\boldsymbol{w}$$

よって

$$\boldsymbol{w} = \left(\widetilde{\boldsymbol{X}}^{T}\widetilde{\boldsymbol{X}}\right)^{-1}\widetilde{\boldsymbol{X}}^{T}\boldsymbol{y}$$

データ行列と目的変数ベクトルからパラメータベクトルが決まる

線型回帰の処理手順と実装

処理手順

- 学習
  - 訓練データセット（特徴量・目的変数）を与えると、残差平方和を最小化する回帰式が得られる
- 予測
  - 回帰式にテストデータセットの特徴量を与えると目的変数の予測値が得られる

実装

- Rのstatsパッケージに含まれるlm実装など

Ridge回帰・Lasso回帰

Ridge回帰

- 最小にすべき目的関数にペナルティ$\lambda|\boldsymbol{w}|^2$（L2ノルム)を付加

$$E(\boldsymbol{w}) = \left|\boldsymbol{y} - \widetilde{\boldsymbol{X}}\boldsymbol{w}\right|^{\boldsymbol{2}} + \lambda|\boldsymbol{w}|^2$$

- 係数があまりに大きくならないようにしている

Lasso回帰

- 最小にすべき目的関数にペナルティ $\lambda|w|_1$（L1ノルム）を付加

$$E(\boldsymbol{w}) = \left|\boldsymbol{y} - \widetilde{\boldsymbol{X}}\boldsymbol{w}\right|^{\boldsymbol{2}} + \lambda|w|_1$$

- ここで$|w|_1 = \sum_{i=1}^{d}|w_i|$
- 係数があまりに大きくならないようにしている

Ridge回帰・Lasso回帰

Ridge回帰

- 最小にすべき目的関数にペナルティ$\lambda|\boldsymbol{w}|^2$（L2ノルム)を付加

$$E(\boldsymbol{w}) = \left|\boldsymbol{y} - \widetilde{\boldsymbol{X}}\boldsymbol{w}\right|^{\mathbf{2}} + \lambda|\boldsymbol{w}|^2$$

- 係数があまりに大きくならないようにしている

Lasso回帰

- 最小にすべき目的関数にペナルティ $\lambda|w|_1$（L1ノルム）を付加

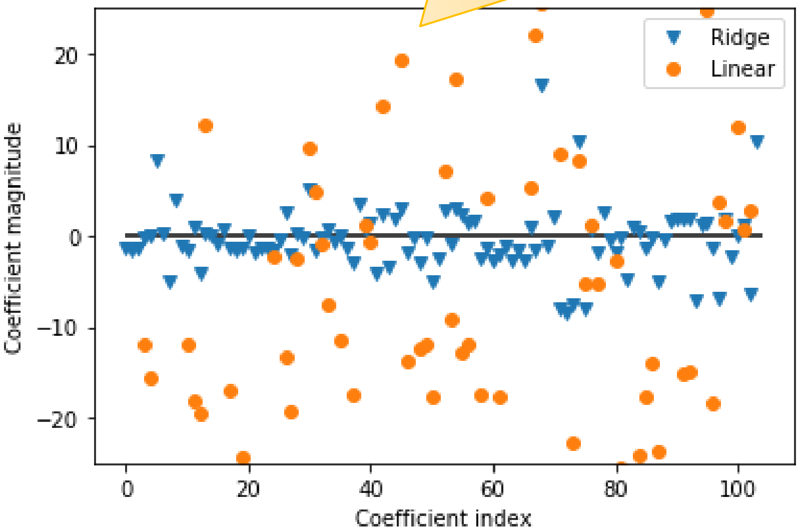
$$E(\boldsymbol{w}) = \left|\boldsymbol{y} - \widetilde{\boldsymbol{X}}\boldsymbol{w}\right|^{\mathbf{2}} + \lambda|w|_1$$

- ここで$|w|_1 = \sum_{i=1}^{d}|w_i|$
- 係数があまりに大きくならないようにしている

線形回帰とRidge回帰の比較

- Boston house prices dataset
- 目的変数：ボストン近郊の住宅地の住宅価格の中央値
- 説明変数：犯罪率、固定資産税率、生徒教師比率等13の特徴量と、13から2つの特徴量を重複を許して選んだ組み合わせ91の合計104の特徴量
- サンプル数506
  - 379を訓練データにする
  - 127をテストデータにする
- 訓練データを用いて104の説明変数の重みを求めて、テストデータで推定する

▼Ridgeは係数小さい
●Linearはどれも係数大きい



(Linear) Training set score: 0.95
(Linear) Test set score: 0.61
(Ridge) Training set score: 0.89
(Ridge) Test set score: 0.75

Ridgeの方がテストのscoreが高くなる

Ridge回帰は係数の大きさにペナルティを与えることで過剰適合を防ぐ

※データ出典：A. C. Muller et al. mglearn.datasets

SVM

教師あり学習（分類／回帰）

サポートベクターマシン（SVM）

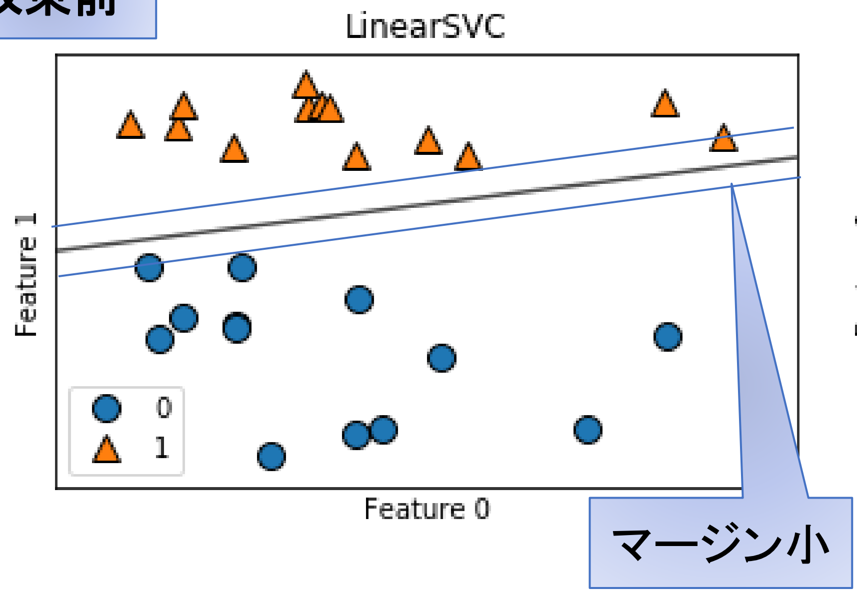
教師あり学習のひとつ
- 分類もしくは回帰に利用可能

SVMの種類
- ハードマージンSVM
  - 各クラスのデータが分類境界からなるべく離れるように分類境界を定める
- ソフトマージンSVM
  - 線形分離できない場合に、スラック変数ξを導入して、分類境界で分離できないデータを許容する
- カーネルSVM
  - カーネル法によって入力データをより高次元の特徴空間に写像し、特徴空間上で線形分離することで、実空間での非線形分離を可能にする
- SV回帰
  - SVMを回帰問題へ応用したもの

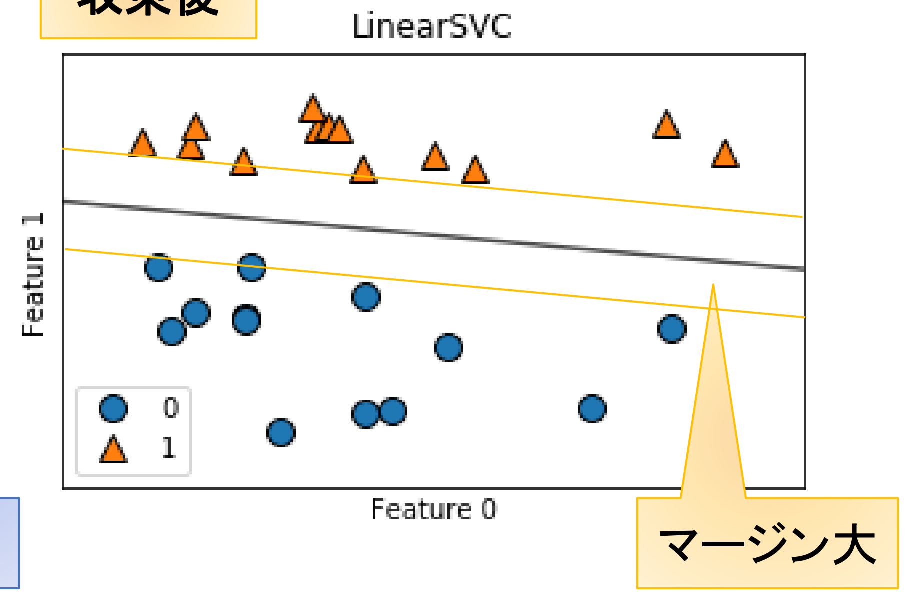
SVMの実装
- Rのkernlabパッケージなどで実装されている

ハードマージンSVM

収束前



収束後

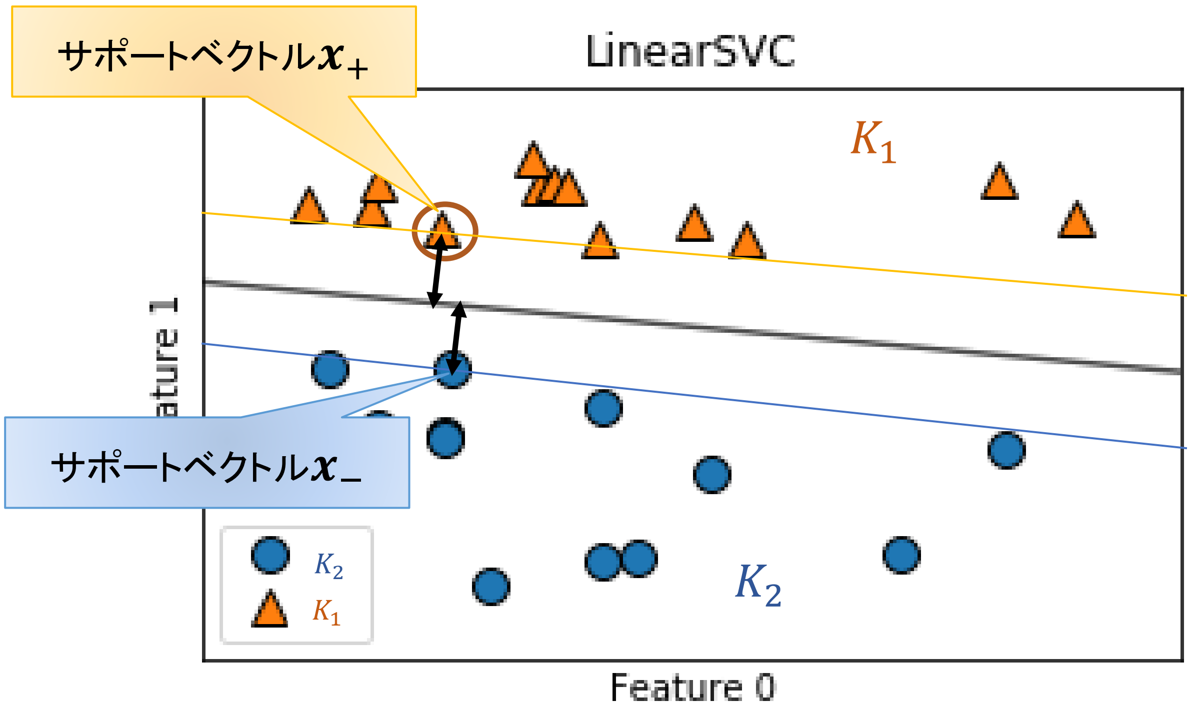


それぞれのクラスのデータが分類境界からなるべく離れる(マージンを最大化する)ように分類境界を定める

サポートベクトル：分類境界に最も近いサンプル
マージン：サポートベクトルと分類境界の距離

※データ出典：A. C. Muller et al. mglearn.datasetsを改変

マージン最大化による分類境界の導出(1/3)



分類境界

$$\boldsymbol{w}^T\boldsymbol{x} + b = 0$$

$$\boldsymbol{w}^T\boldsymbol{x}_i + b > 0$$

$$t_i = \begin{cases} 1\ (\boldsymbol{x}_i \in K_1) \\ -1\ (\boldsymbol{x}_i \in K_2) \end{cases}$$

$$\boldsymbol{w}^T\boldsymbol{x}_i + b < 0$$

なるラベル変数t_iを導入すると,

任意の点$\boldsymbol{x}_i$と分類境界平面との距離は $\dfrac{t_i(\boldsymbol{w}^T\boldsymbol{x}_i + b)}{|\boldsymbol{w}|}$ と表される．これより，マージン最大化は

$$\underset{\boldsymbol{w},b}{\text{Maximize}}\ \min_i \frac{t_i(\boldsymbol{w}^T\boldsymbol{x}_i + b)}{|\boldsymbol{w}|}$$

※データ出典：A. C. Muller et al. mglearn.datasetsを改変

マージン最大化による分類境界の導出(2/3)

$$\underset{\boldsymbol{w},b}{\text{Maximize}}\ \min_i \frac{t_i(\boldsymbol{w}^T\boldsymbol{x}_i + b)}{|\boldsymbol{w}|} = \underset{\boldsymbol{w},b}{\text{Maximize}} \frac{1}{|\boldsymbol{w}|}\ \min_i t_i(\boldsymbol{w}^T\boldsymbol{x}_i + b)$$

今、$\min_i t_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) = 1$ の場合だけ考えればよい．これより最適化の式は，

$$\underset{\boldsymbol{w},b}{\text{Maximize}} \frac{1}{|\boldsymbol{w}|} \qquad \text{Subject to } t_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) \geq 1\ (i = 1,2,\cdots,n)$$

さらに簡単化して，

$$\underset{\boldsymbol{w},b}{\text{Minimize}} \frac{1}{2}|\boldsymbol{w}|^2 \quad \text{Subject to } t_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) \geq 1\ (i = 1,2,\cdots,n)$$

ラグランジュ未定乗数$\boldsymbol{a}$を導入して以下のラグランジュ関数を定義する

$$L(b,\boldsymbol{w},\boldsymbol{a}) = \frac{1}{2}|\boldsymbol{w}|^2 - \sum_{i=1}^{n} a_i\{t_i(\boldsymbol{b} + \boldsymbol{w}^T\boldsymbol{x}_i) - 1\}$$

Lをb, wで偏微分して0とおいたものを用いてLを変形すると以下のような$\boldsymbol{a}$だけの関数になる

$$L(\boldsymbol{a}) = \sum_{i=1}^{n} a_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} a_i a_j t_i t_j \boldsymbol{x}_i^T \boldsymbol{x}_j$$

マージン最大化による分類境界の導出（3/3）

主問題であったマージン最大化問題は双対問題である以下の二次計画問題に帰着される

$$\text{Maximize } L(\boldsymbol{a}) = \sum_{i=1}^{n} a_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} a_i a_j t_i t_j \boldsymbol{x}_i^T \boldsymbol{x}_j$$

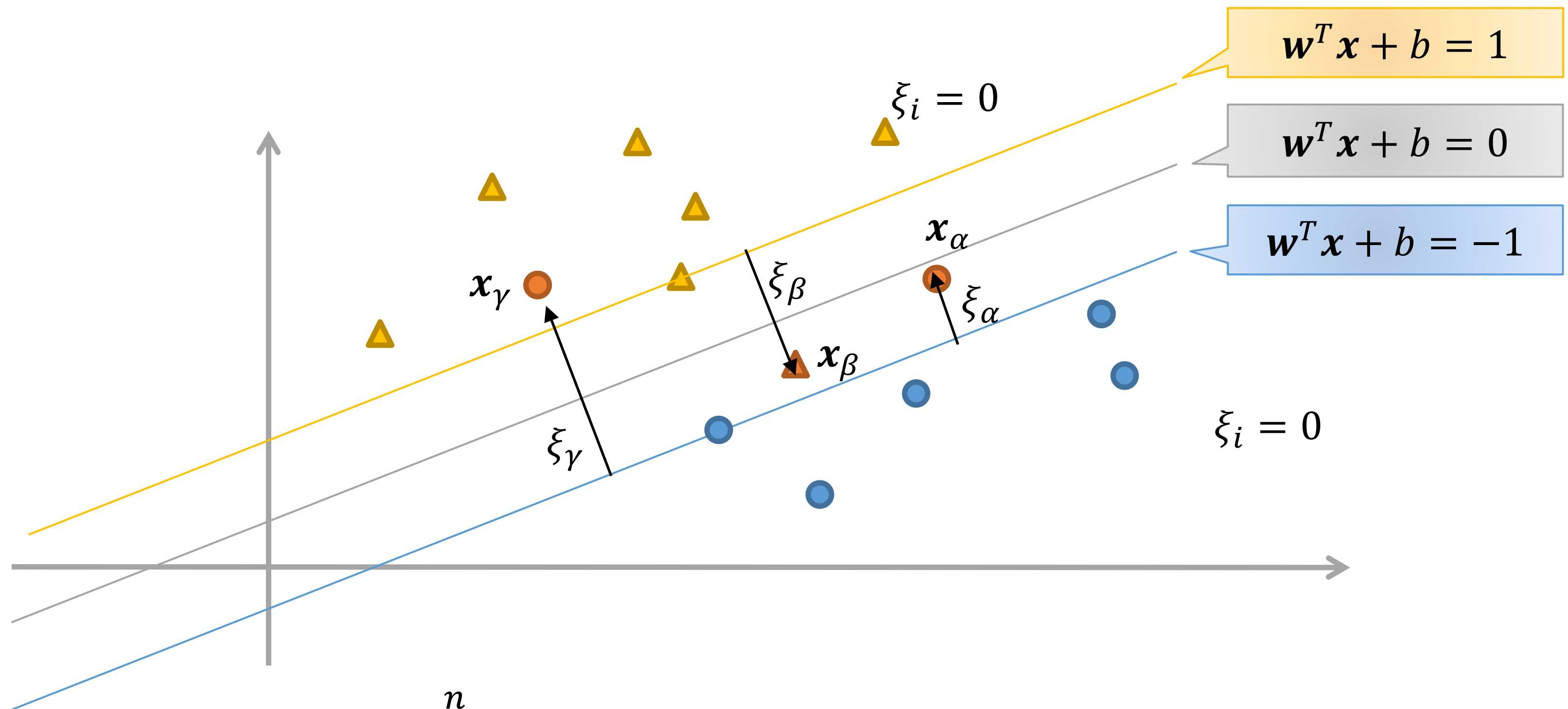
$$\text{Subject to } \sum_{i=1}^{n} a_i t_i = 0$$

$$a_i \geq 0 \qquad a_i\{t_i(b + \boldsymbol{w}^T \boldsymbol{x}_i) - 1\} = 0$$

SVMのライブラリ内部では、このような双対問題を解くような実装がなされている

ソフトマージンSVM

- 線形分離できない場合に、スラック変数ξを導入して、分離境界で分離できないデータを許容する方法



$$\underset{\boldsymbol{w},b,\xi}{\text{Minimize}} \quad \frac{1}{2}|\boldsymbol{w}|^2 + C\sum_{i=1}^{n}\xi_i$$

$$\text{Subject to } t_i(\boldsymbol{w}^T\boldsymbol{x_i} + b) \geq 1 - \xi_i \ (i = 1,2,\cdots,n)$$

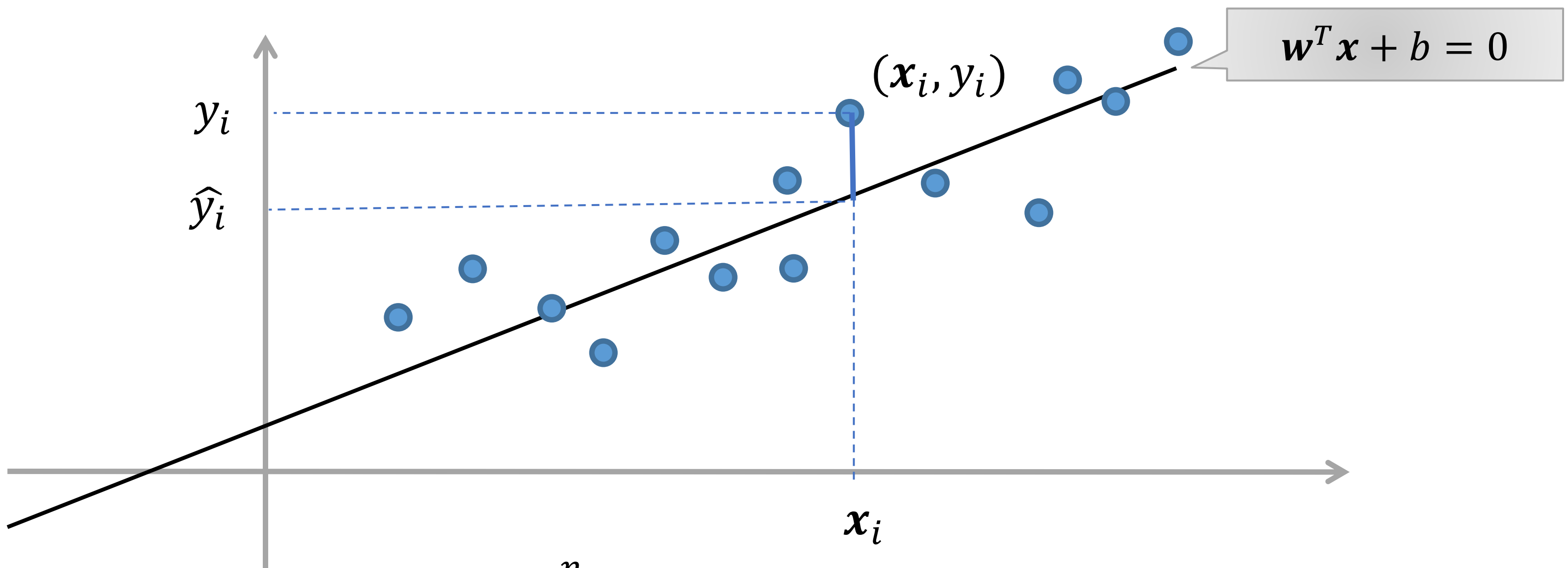
$$\xi_i \geq 0 \ (i = 1,2,\cdots,n)$$

C:小さい⇒大きいξを許容

C:→ ∞ ⇒ $\xi = 0$のみ許容（ハードマージン）

SV回帰

- SVMの回帰問題への応用



$$\underset{\boldsymbol{w},b}{\text{Minimize}} \quad \frac{1}{2}|\boldsymbol{w}|^2 + C\sum_{i=1}^{n}\max\{|y_i - (\boldsymbol{w}^T\boldsymbol{x}_i + b)| - \epsilon, 0\}$$

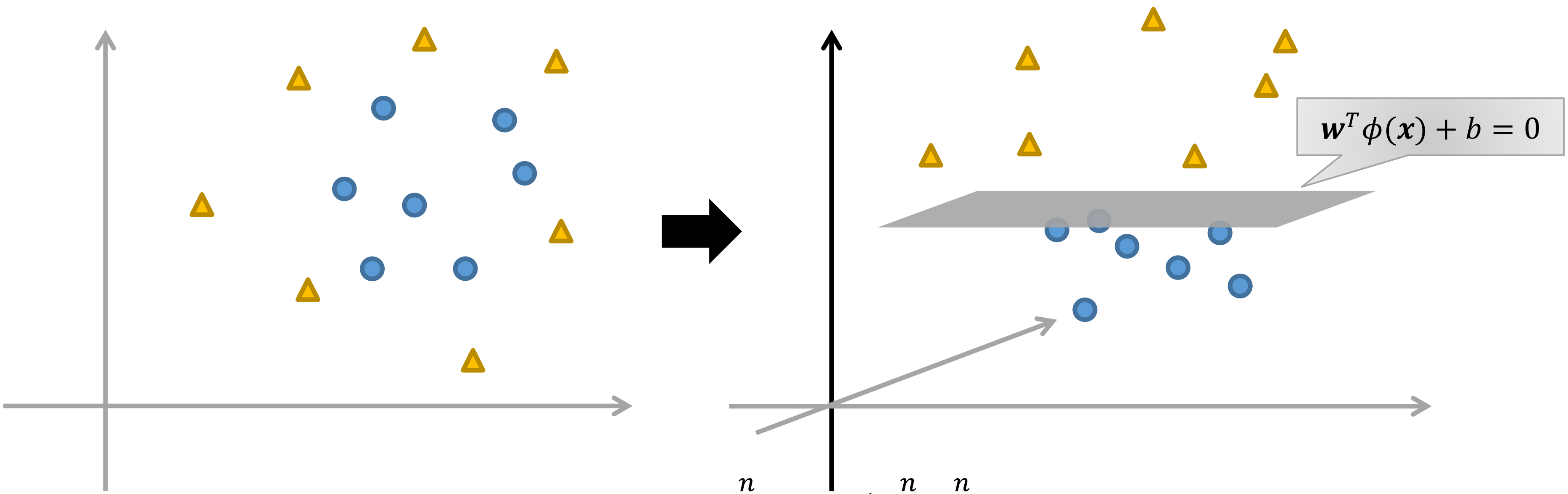
$\epsilon > 0$： 不感度係数
$C > 0$ ： 正則化係数

不感度：残差がϵ以下の時は損失0
正則化：小さい ⇒ 残差がもたらす損失に対して寛容

※カーネル法を用いれば非線形回帰も可能

非線形SVM

入力データをより高次元の特徴空間に写像し、特徴空間上で線形分離することで、実空間での非線形分離を可能にする方法



$$\text{Maximize } L(\boldsymbol{a}) = \sum_{i=1}^{n} a_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} a_i a_j t_i t_j \phi(\boldsymbol{x}_i)^T \phi(\boldsymbol{x}_j)$$

$$\text{Subject to } \sum_{i=1}^{n} a_i t_i = 0 \qquad a_i \geq 0$$

カーネル法

$\phi(\boldsymbol{x}_i)^T\phi(\boldsymbol{x}_j) = K(\boldsymbol{x}_i, \boldsymbol{x}_j)$ **とおくと、目的関数は次のようになる**

$$L(\boldsymbol{a}) = \sum_{i=1}^{n} a_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} a_i a_j t_i t_j K(\boldsymbol{x}_i, \boldsymbol{x}_j)$$

ここでKはカーネル関数と呼ばれる

カーネル関数を使うことで、特徴空間における内積$\phi(\boldsymbol{x}_i)^T\phi(\boldsymbol{x}_j)$を明示的に計算せずに、$K$だけの評価で問題を解くことができる

特徴空間中の座標を計算しないことで計算量を抑えることができる

カーネル関数の一例

- RBFカーネル（Radial Basis Function kernel）
- 実データ　$\boldsymbol{x}_i$ と　$\boldsymbol{x}_j$の特徴空間における内積

$$K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\left(-\frac{\left|\boldsymbol{x}_i - \boldsymbol{x}_j\right|^2}{2\sigma^2}\right)$$

演習：教師あり学習（回帰）演習ガイド参照

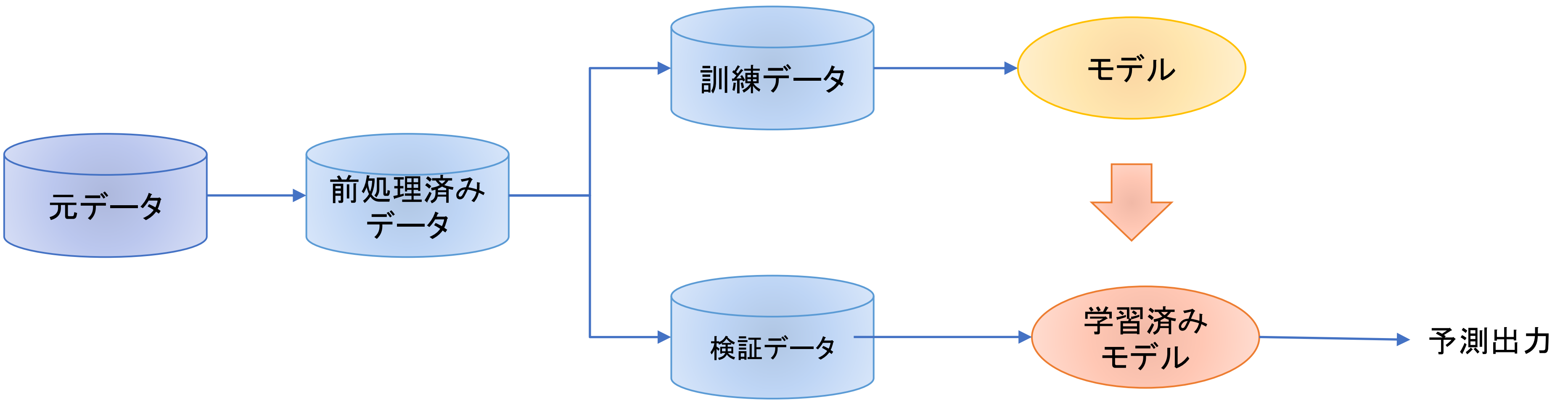
機械学習モデル構築（演習）手順

0. 前処理

1. データ分割

2. モデル（アルゴリズム）選択

3. 学習：訓練データ（入出力の組）をモデルに与えてパラメータ調整



4. 予測：検証データ（入力）を学習済みモデルに与えて予測出力

5. 評価：予測出力と検証データ（出力）を比較して予測精度を評価

演習：Wine Qualityデータセットを利用
- 11種類の科学的特性に基づいて品質スコアを推定する機械学習モデルを構築
- 線形重回帰かSVMを採用
- 必要なら正規化・標準化等の前処理を実施
- 白ワイン4,898事例のデータを訓練データ3,750事例、検証データ1,148事例に分割
- 訓練データでモデルを訓練
- 検証データを学習済みモデルに与え、その予測精度を評価