データマイニング

第10回　集団学習（アンサンブル学習）

2023年春学期

宮津和弘

本日の講義・演習

日付	講義・演習内容
04/14/23	(1) イントロダクション
04/21/23	(2) ビジネスシミュレーション
04/28/23	(3) ID-POSデータ分析
05/12/23	(4) 対応分析
05/19/23	(5) クラスター分析
05/26/23	(6) 自己組織化マップ
06/02/23	(7) 線形判別分析
06/09/23	(8) 非線形判別分析
06/16/23	(9) ツリーモデル
06/23/23	**(10) 集団学習**
~~06/30/23~~	(11) サポートベクターマシン
07/04/23	~~(12) ネットワーク分析~~
07/14/23	(13) 共分散構造分析
07/21/23	(14) テキスト分析
07/28/23	(15) まとめ

※来週（6/30）は **黒門祭** のためお休みです！

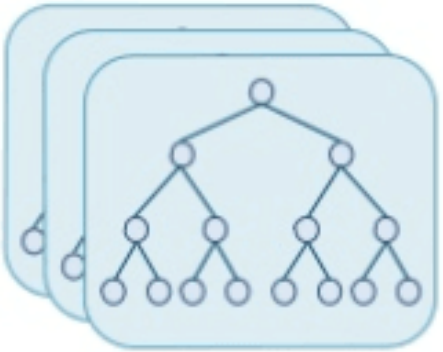
本日の演習概要とポイント

- **集団学習（アンサンブル学習）**

  → バッギング、ブースティング、スタッキング

- **ランダムフォレスト**

  → 乳癌データ、オゾン濃度データ を用いた演習



機械学習の手法

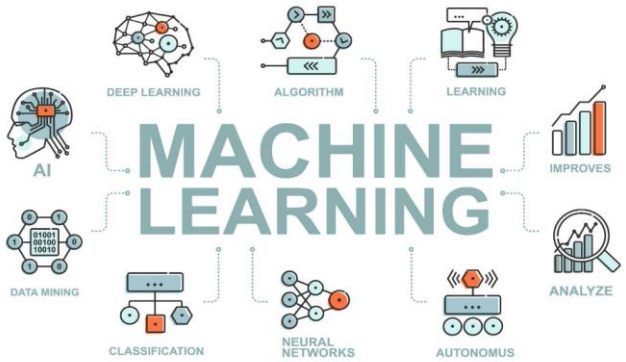
教師データあり

- ✓ 線形回帰
- ✓ ロジスティック回帰
- • サポートベクターマシーン
- ✓ 分類木
- ✓ 回帰木
- • **ランダムフォレスト**
- • 勾配ブースティング木
- ✓ ニューラルネットワーク
- • 畳み込みニューラルネットワーク
- • 再起型ニューラルネットワーク
- ✓ ナイーブベイズ
- • k近傍法ブースティング
- • **バギング**

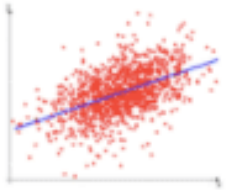
教師データなし

- ✓ 階層型クラスタリング(ウォード法など)
- ✓ 非階層型クラスタリング（k-meansなど)
- • トピックモデル（LDAなど)
- ✓ 自己組織化マップ
- ✓ アソシエーション分析（*）
- ✓ 協調フィルタリング（*）
- ✓ ベイジアンネットワーク（*）

＊ データサイエンス演習 1



機械学習の手法２　－ 分類と予測 －



回帰

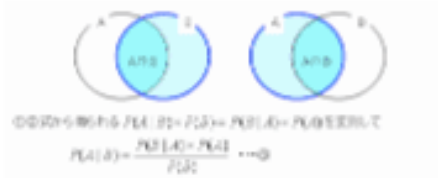
・線形回帰
・ロジスティック回帰
・サポートベクターマシン（SVM）

木

・決定木
・回帰木
・ランダムフォレスト
・XGBoost

ニューラルネット

・単純パーセプトロン
・DNN
・CNN
・RNN

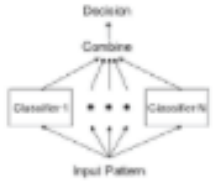


ベイズ（事後確率）

・ナイーブベイズ

クラスタリング

・k-means
・k-means++



アンサンブル学習

・Boosting
・Adaboost

https://spjai.com/regression-tree/

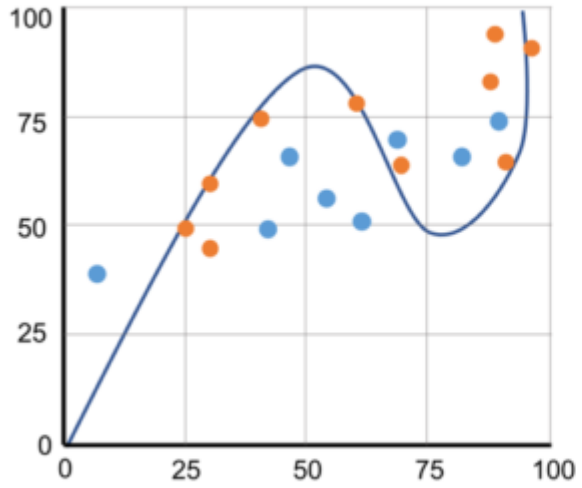
予測と適合（トレードオフの関係）

学習データに限りなく適合するようにモデル化すると、それ以外の評価データに対する予測能力が劣化する。これを過学習と呼び、適合性と予測性はトレードオフの関係にある。

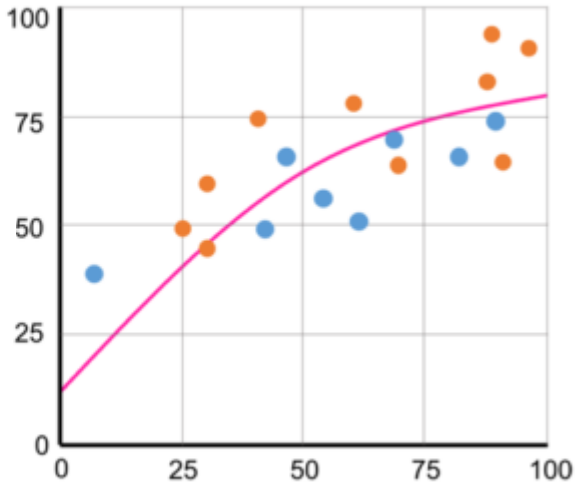
学習データに対して
過度の適合している

↓

データへの当てはまりは良いが、予測能力が劣化する



vs.



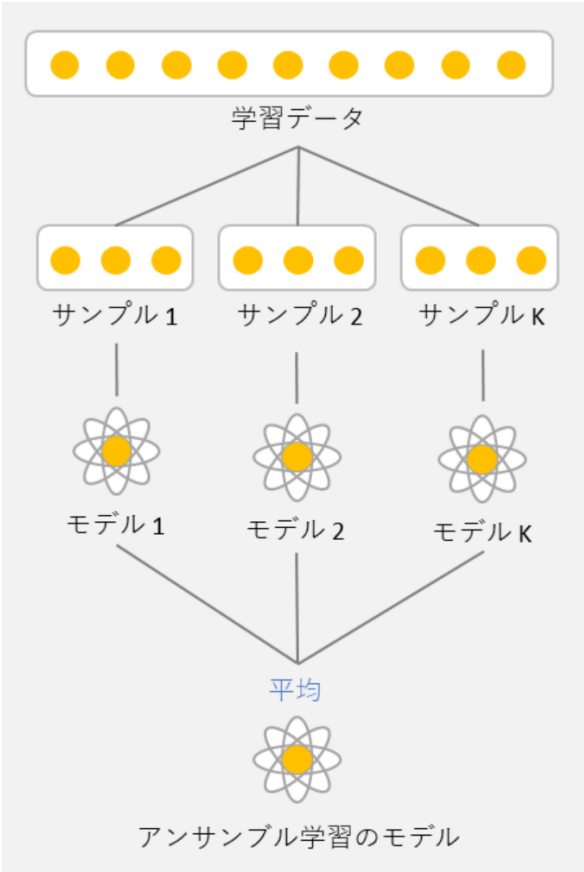
学習データに対して
緩く適合している

↓

データへの当てはまりは平均的だが、予測能力は良い

バギング

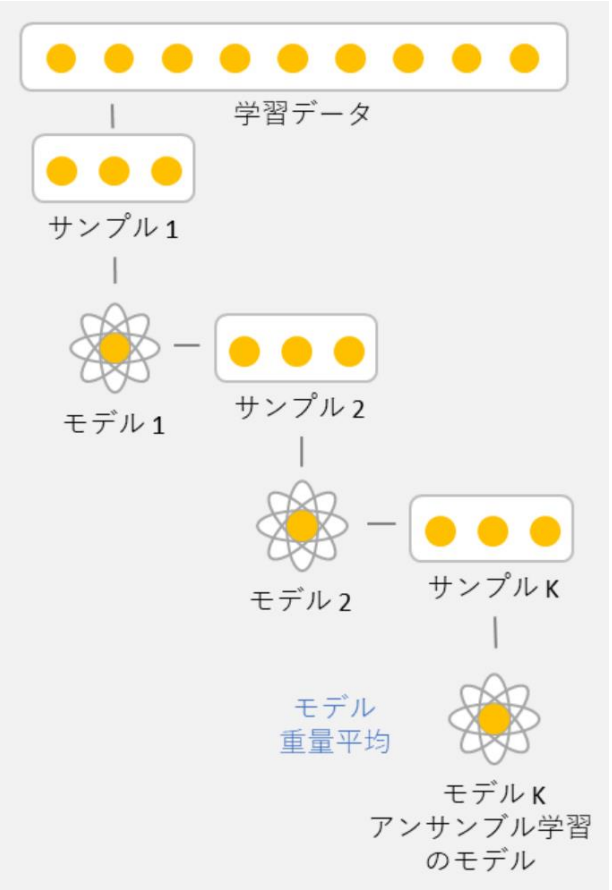
- 学習データからランダムに抽出したデータセットを用いたモデル構築を繰り返し実行して、複数モデルを生成する
- 生成された複数モデルを用いて１つのモデルを構築する
  → 例えば、複数モデルの平均を最終モデルとする
- 学習データの一部を用いて構築したモデルを“**弱学習器**”と呼ぶ（全学習データから構築したモデルと区別する）
- バギングを用いた手法として「**ランダムフォレスト**」がある
  → 後述、データ演習を行う
- 並列処理が可能で、ブースティングよりも効率的である



出典：S-Analysis社

ブースティング

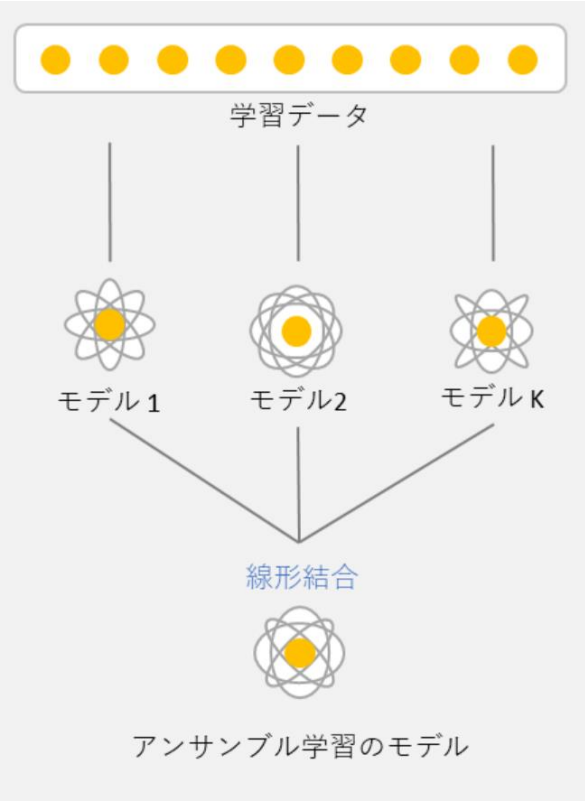
- 学習データからランダム抽出したデータセットを用いて、最初のモデル(“弱学習器”)構築する
- 構築された弱学習器が誤った部分に重みをかけて、次の弱学習器を構築する
- ブースティングを用いた手法として、**XGBoost**や**勾配ブースティング**などがある
- 直列計算のため、バギングよりも時間がかかる



出典：S-Analysis社

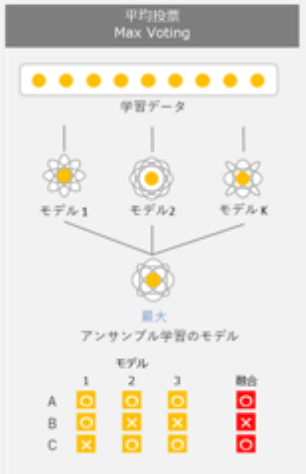
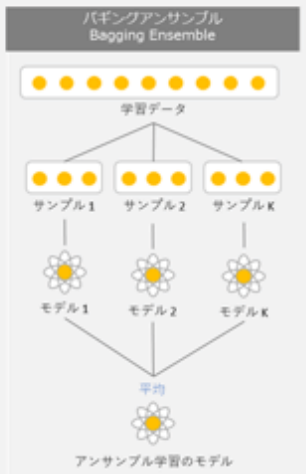
スタッキング

- ランダムフォレストや勾配ブースティングなどの異なる手法で複数モデルを構築する
  → 各モデル構築はサンプルデータセットからではなく全学習データを用いて構築する
- 各モデルから予測値を算出し、結果を統合する
  → **メタモデル**の作成（線形結合）
- 並列処理が可能で、データ処理は効率的である



出典：S-Analysis社

アンサンブル学習手法
Ensemble Learning

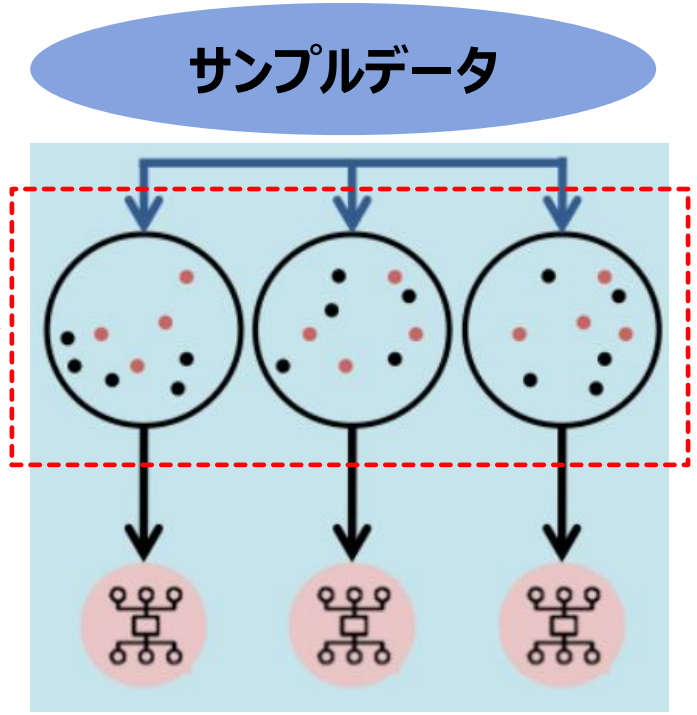
	平均投票 Max Voting	重量平均投票 Weighted Average Voting	バギング Bagging	ブースティング Boosting	スタッキング Stacking
モデルの構成					
複数サンプル	✕	✕	○	○	✕
複数モデル	○	○	○	○	○
モデル作成方法	平行	平行	平行	階段	平行
結果の融合方法	平行	重量平均	平均	重量平均	線形結合
バイアスとバリアンスのエラー処理			バリアンス		バイアス バリアンス

出典：S-Analysis社

ブートストラップ法

サンプルを母集団と見立てて、この疑似母集団から重複を許容せず、無作為に繰り返しサンプルを抽出すること。

バッギングではブートストラップ法で**リサンプリング**された複数のサンプルセットを学習データとして機械学習に用いる。各リサンプルは独立のため(同時に)並列で複数のモデル学習が可能となる



オゾン濃度データの読込み

① Rstudio起動する

② > library(Rcmdr)　　※コマンドラインから Rコマンダー を起動する

③ 演習ファイル “cancer.csv” を読み込む

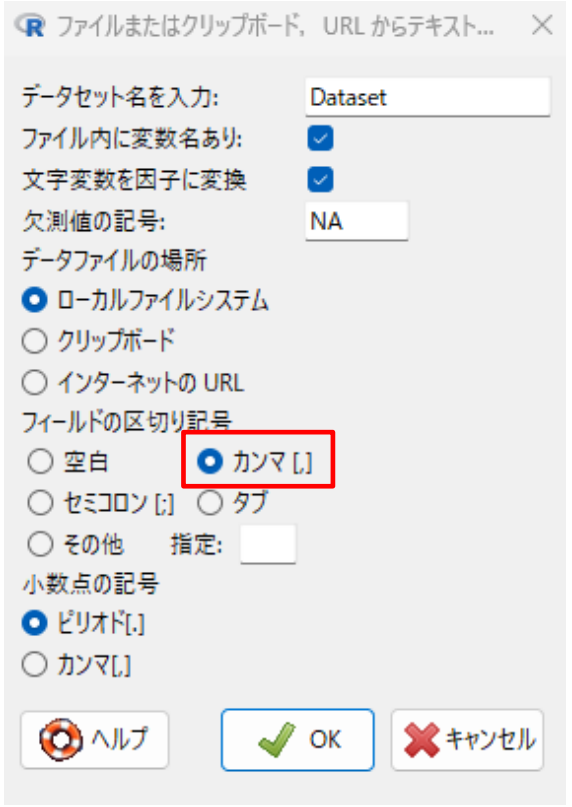
- Rstudio　> Dataset<-read.csv(“ozone.csv”)

  又は

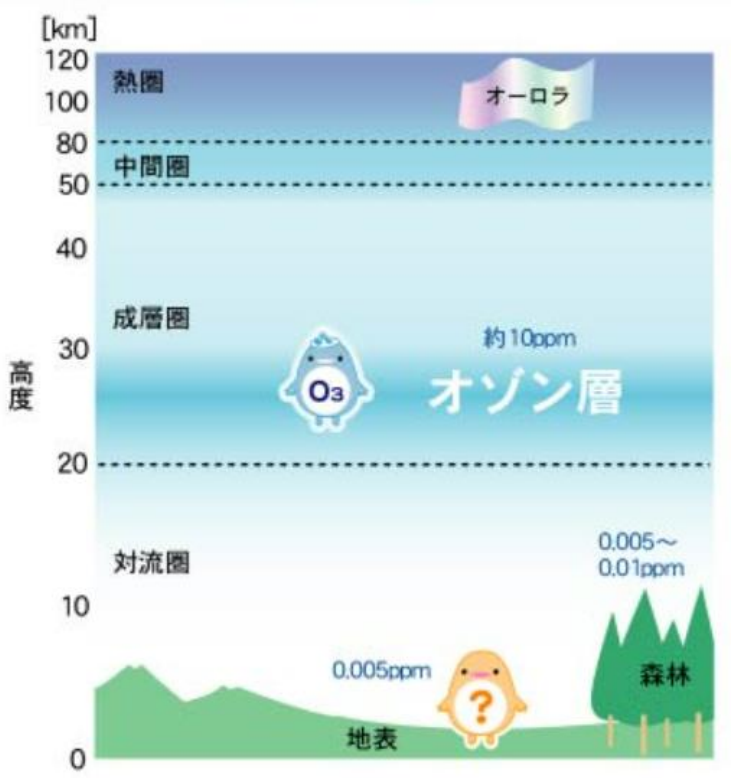
- Rコマンダー　(データ) → (データインポート) → (テキストファイルまたはクリップボード・・・) →

  ✓OKを選択して、ozone.csv を指定する

④ 演習データが Dataset に読込まれる



オゾン濃度データ



https://www.rgl.co.jp/html/page120.html

気圧500mmbに達する**高度**

逆転温度

逆転基準温度

風速 **湿度** **気温** **気圧差** **視界** **測定日**

オゾン濃度

o3	vh	wind	humidity	temp	ibh	dpg	ibt	vis	doy
3	5710	4	28	40	2693	-25	87	250	33
5	5700	3	37	45	590	-24	128	100	34
5	5760	3	51	54	1450	25	139	60	35
6	5720	4	69	35	1568	15	121	60	36
4	5790	6	19	45	2631	-33	123	100	37
4	5790	3	25	55	554	-28	182	250	38
6	5700	3	73	41	2083	23	114	120	39
7	5700	3	59	44	2654	-2	91	120	40
4	5770	8	27	54	5000	-19	92	120	41
6	5720	3	44	51	111	9	173	150	42
5	5760	6	33	51	492	-44	181	40	43
4	5780	6	19	54	5000	-44	135	200	44
4	5830	3	19	58	1249	-53	243	250	45
7	5870	2	19	61	5000	-67	186	200	46
5	5840	5	19	64	5000	-40	174	200	47
9	5780	4	59	67	639	1	189	150	48
4	5680	5	73	52	393	-68	210	10	49
3	5720	4	19	54	5000	-66	126	140	50
4	5760	3	19	54	5000	-58	111	250	51
4	5730	4	26	58	5000	-26	111	200	52

バギングの実行

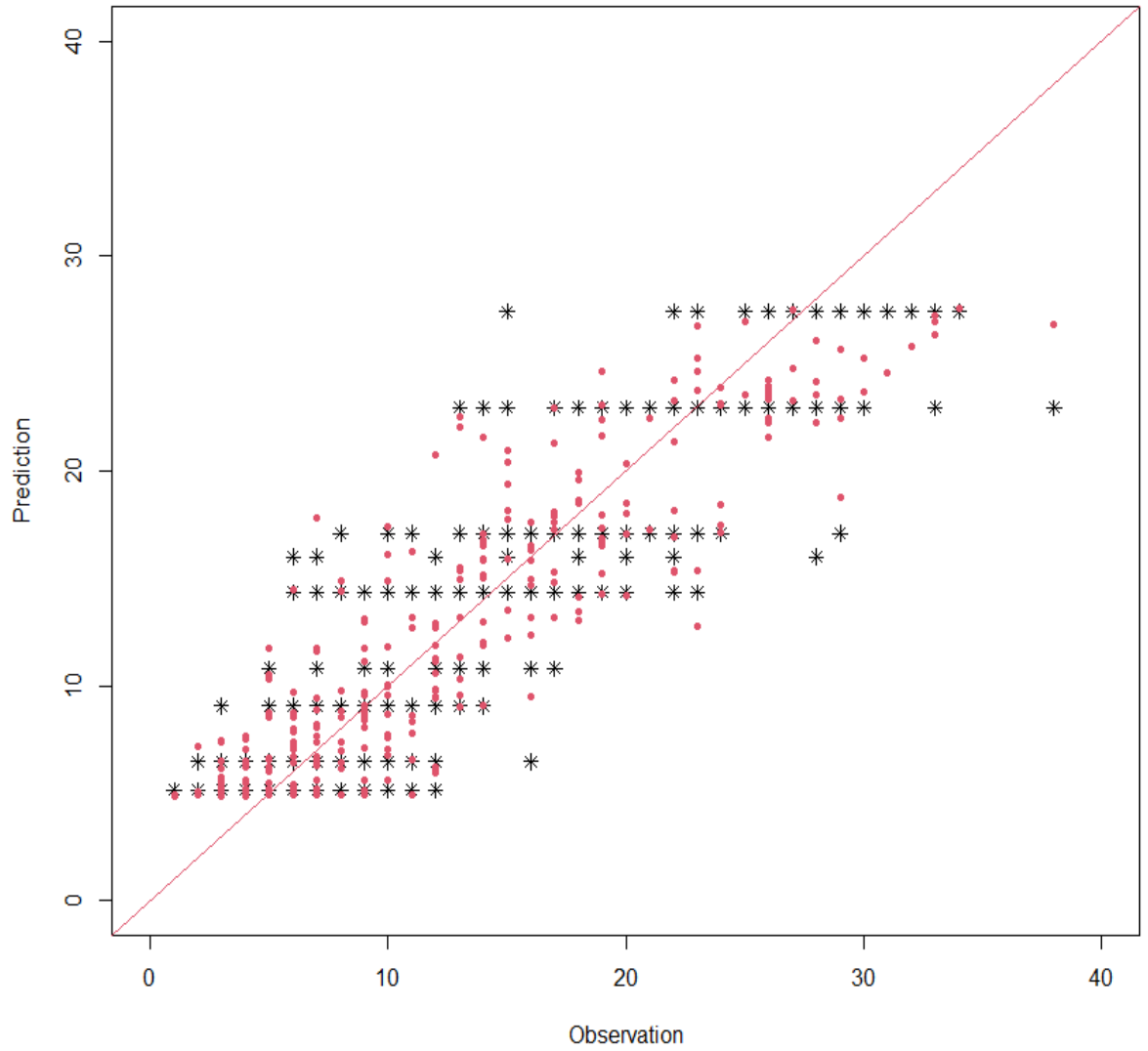
※　バギングには”**ipred**”パッケージのインストールが必須

```
> ozone<-Dataset
> library(ipred)
> resBB <- bagging(O3~.,data=ozone,nbagg=500,cob=TRUE)
> print(resBB)

Bagging regression trees with 500 bootstrap replications

Call: bagging.data.frame(formula = O3 ~ ., data = ozone, nbagg = 500,
    cob = TRUE)
```

※　損失に対するOOB
(Out-Of-Bag)推定あり

※　ここのバギングでは、**ブートストラップ法**で
500回のリサンプリングを実施

```
> library(rpart)
> resPR<-rpart(O3~.,data=ozone)
> prePR<-predict(resPR,ozone)
> preBB<-predict(resBB,ozone)
> matplot(ozone[,1],cbind(prePR,preBB),pch=c(8,20),col=3:4,
+ xlim=c(0,40),ylim=c(0,40),xlab="observation",ylab="prediction")
> abline(0,1,type=2)
```

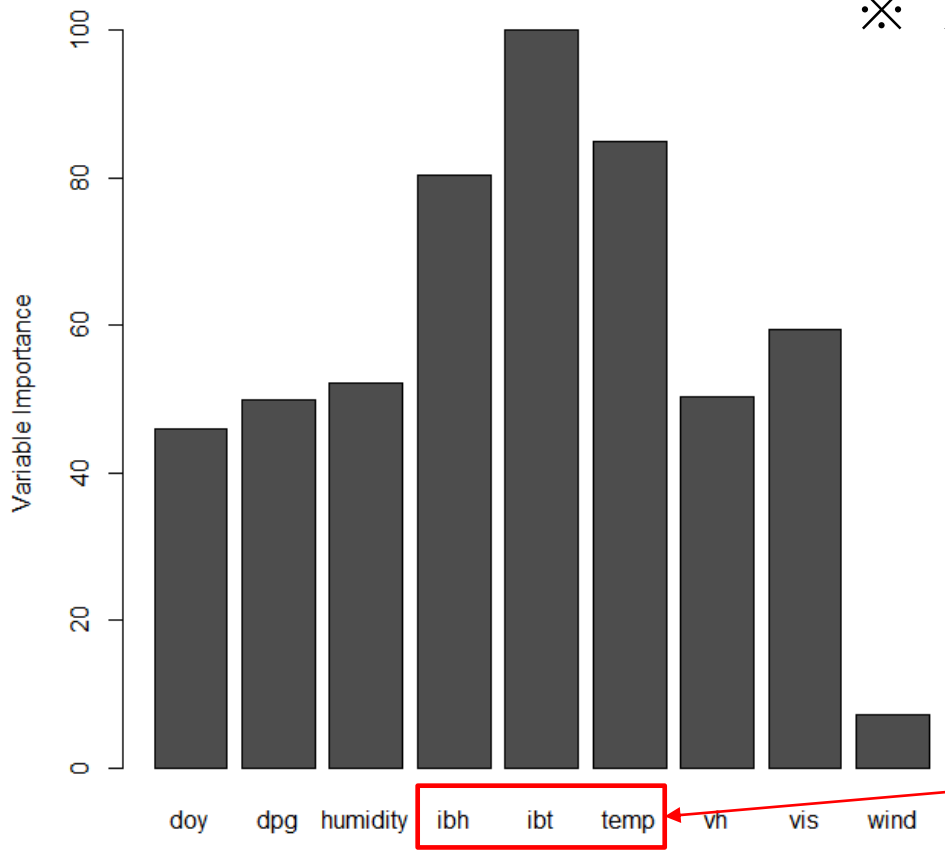


※ 決定木（rpart）を実行するには、
> library(rpart)　※　予め実行しておくこと

バギング法の方がｙ＝ｘの線に集結しており、
適合率は良いと考えられる

＊　CART法（単一サンプル）
● バギング法（複数サンプル）

バギングにおける変数の重要度



※　影響度には”**caret**”パッケージのインストールが必須

```
> library(caret)
> vimp<-varImp(resBB)
> (rimp<-vimp/max(vimp)*100)
             Overall
doy        45.908250
dpg        49.819627
humidity   52.221743
ibh        80.295447
ibt       100.000000
temp       84.814133
vh         50.301054
vis        59.383792
wind        7.284144
> barplot(t(rimp),ylab="Variable Importance")
```

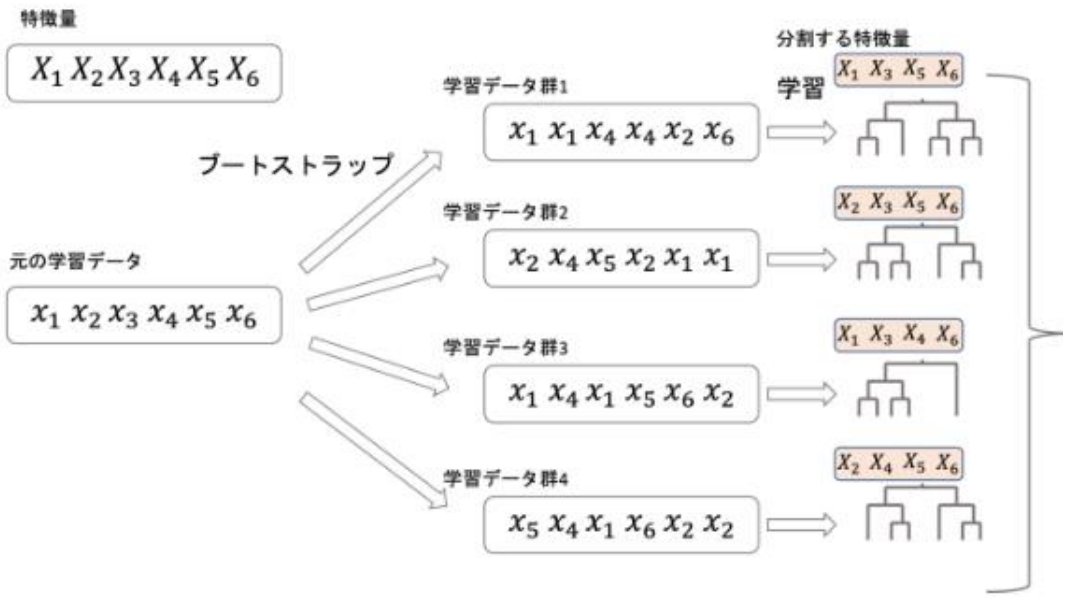
※　逆転温度と気温による影響度合いが大きい

ランダムフォレスト

ランダムフォレスト

決定木を弱学習器として”**バギング**”による**アンサンブル学習**の手法

→　単体の決定木よりも、**適合性**と**予測性**に優れる！



https://datawokagaku.com/random_forest/

学習データの一部をブートストラップ法で抽出して複数の決定木を生成して**アンサンブル学習**する

→　**回帰 & 分類**

乳癌データの読込み

① Rstudio起動する

② > library(Rcmdr)　　※コマンドラインから Rコマンダー を起動する

③ 演習ファイル “cancer.csv” を読み込む

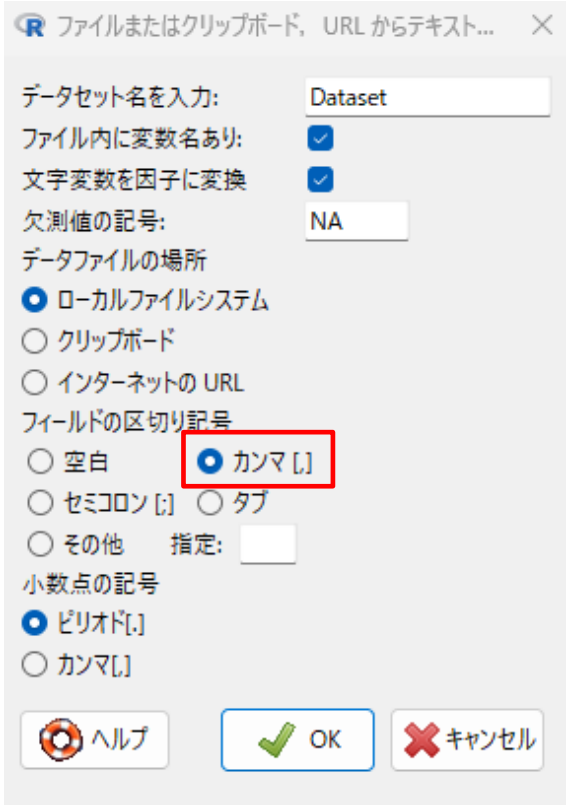
- Rstudio　> Dataset<-read.csv(“cancer.csv”)

  又は

- Rコマンダー　(データ) → (データインポート) → (テキストファイルまたはクリップボード・・・) →

  ✔OKを選択して、cancer.csv を指定する

④ 演習データが Dataset に読込まれる



乳癌データ

判定：benign（良性）
malignant（悪性）

患者ID	細胞厚さ	細胞大きさ	細胞形状	周縁癒着	上皮細胞大きさ	裸核	クロマチン	正常核小体	有糸分裂	
Id	Cl.thickness	Cell.size	Cell.shape	Marg.adhesion	Epith.c.size	Bare.nuclei	Bl.cromatin	Normal.nucleoli	Mitoses	Class
1000025	5	1	1	1	2	1	3	1	1	benign
1002945	5	4	4	5	7	10	3	2	1	benign
1015425	3	1	1	1	2	2	3	1	1	benign
1016277	6	8	8	1	3	4	3	7	1	benign
1017023	4	1	1	3	2	1	3	1	1	benign
1017122	8	10	10	8	7	10	9	7	1	malignant
1018099	1	1	1	1	2	10	3	1	1	benign
1018561	2	1	2	1	2	1	3	1	1	benign
1033078	2	1	1	1	2	1	1	1	5	benign
1033078	4	2	1	1	2	1	2	1	1	benign
1035283	1	1	1	1	1	1	3	1	1	benign
1036172	2	1	1	1	2	1	2	1	1	benign
1041801	5	3	3	3	2	3	4	4	1	malignant
1043999	1	1	1	1	2	3	3	1	1	benign
1044572	8	7	5	10	7	9	5	5	4	malignant
1047630	7	4	6	4	6	1	4	3	1	malignant
1048672	4	1	1	1	2	1	2	1	1	benign
1049815	4	1	1	1	2	1	3	1	1	benign
1050670	10	7	7	6	4	10	4	1	2	malignant
1050718	6	1	1	1	2	1	3	1	1	benign
1054590	7	3	2	10	5	10	5	4	4	malignant
1054593	10	5	5	3	6	7	7	10	1	malignant
1056784	3	1	1	1	2	1	2	1	1	benign
1057013	8	4	5	1	2	NA	7	3	1	malignant
1059552	1	1	1	1	2	1	3	1	1	benign

ランダムフォレストの実行（分類）

```
> library(randomForest)
> cancer <- na.omit(Dataset[,-1])
> y<- as.factor(as.character(cancer[,10]))
> x<- as.matrix(cancer[,-10])
> classRF<- randomForest(x,y,proximity=TRUE)
> print(classRF)
> plot(classRF)
```

※　ランダムフォレスト初回はパッケージ
“**randomForest**” をインストールする

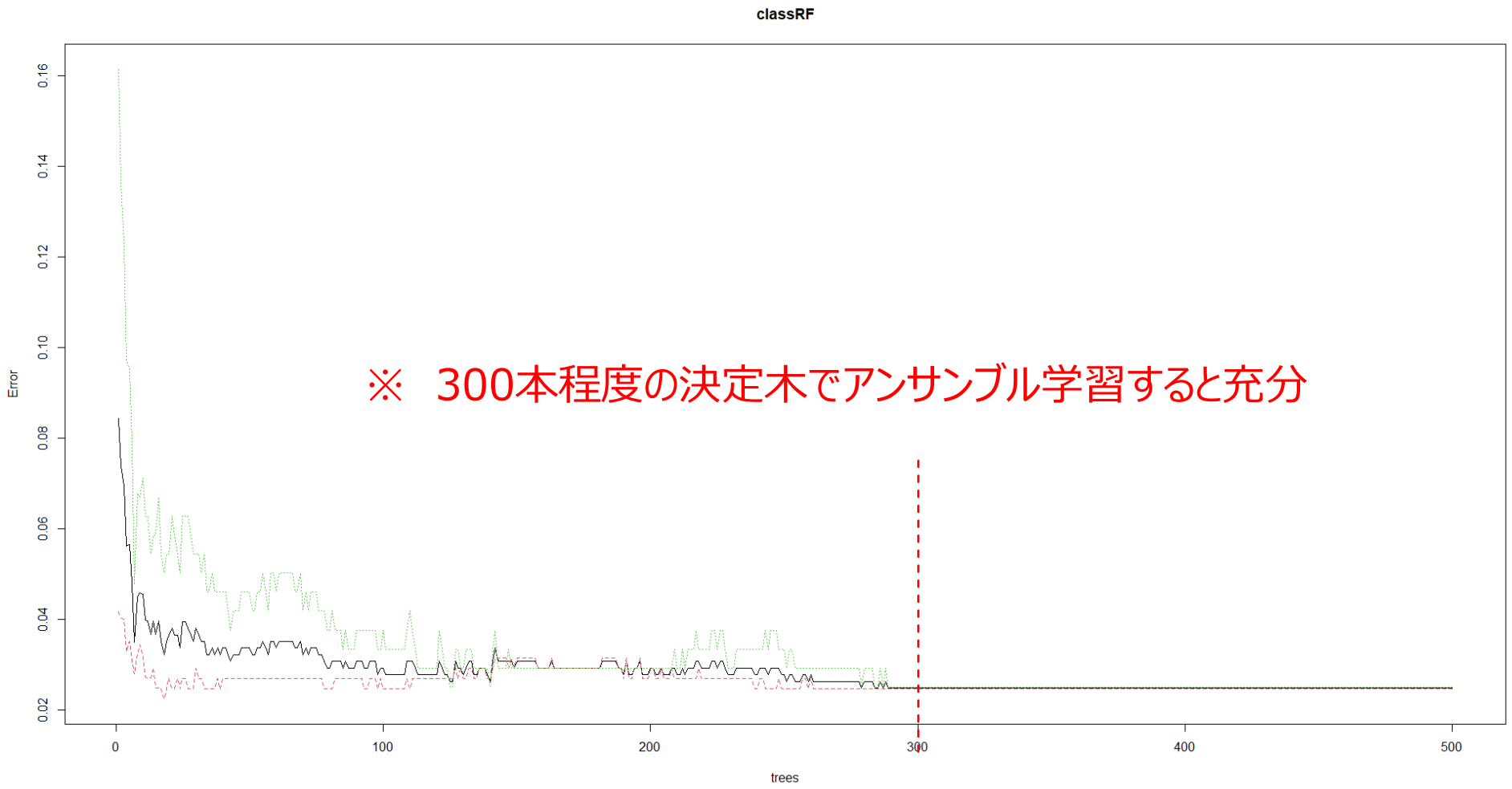
```
Call:
 randomForest(x = x, y = y, proximity = TRUE)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 3

        OOB estimate of  error rate: 2.49%
Confusion matrix:
          benign malignant class.error
benign       433        11  0.02477477
malignant      6       233  0.02510460
```

バギングに用いた決定木の数は500本！

各決定木で採用した特徴量は３つ！

最終的な誤り率は2.49%

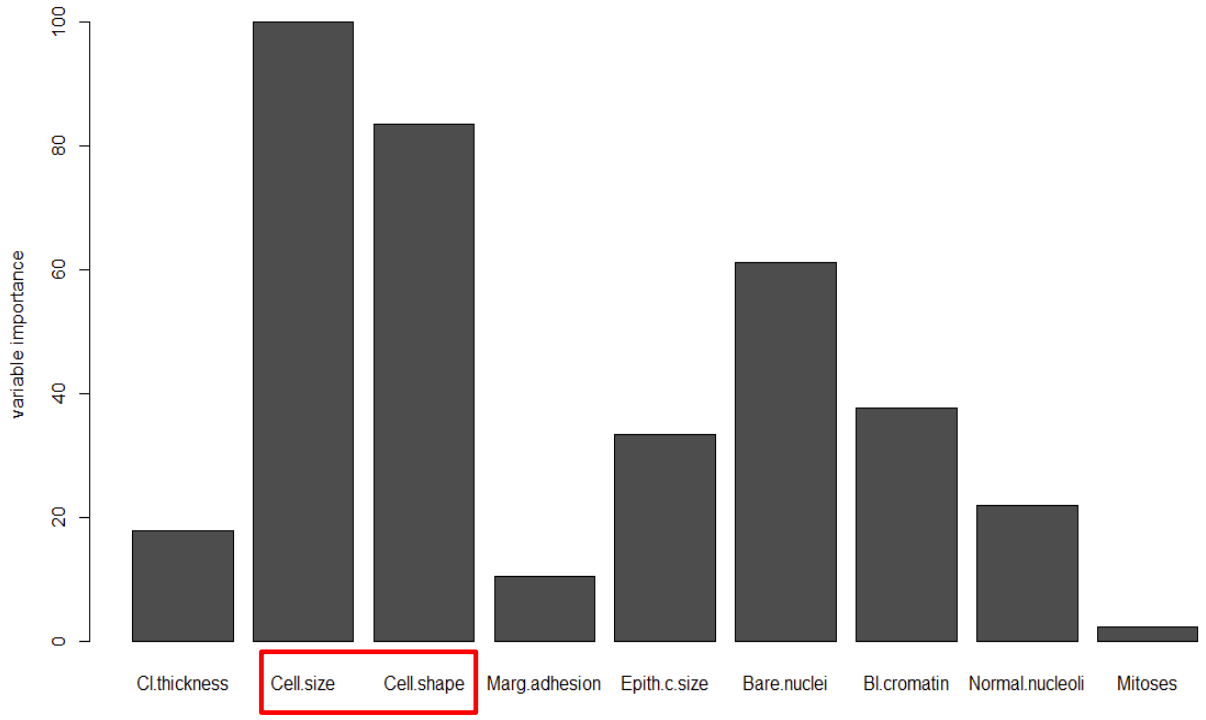
ランダムフォレストの樹木増加に対する平均二乗誤差（分類）



ランダムフォレストの変数重要度（分類）

Gini係数で評価した特徴量の重要度

```
> vimp<-importance(classRF)
> rimp<-vimp/max(vimp)*100
> print(vimp)
                MeanDecreaseGini
Cl.thickness            15.079117
Cell.size               84.036201
Cell.shape              70.175303
Marg.adhesion            8.815259
Epith.c.size            28.073652
Bare.nuclei             51.525253
Bl.cromatin             31.706537
Normal.nucleoli         18.521848
Mitoses                  2.063725
> barplot(t(rimp),ylab="variable importance")
```



※乳癌の分類に最も重要な２つの変数

細胞大きさ

細胞形状

オゾン濃度データの読込み（再掲）

① Rstudio起動する

② ＞ library(Rcmdr)　　※コマンドラインから Rコマンダー を起動する

③ 演習ファイル “cancer.csv” を読み込む

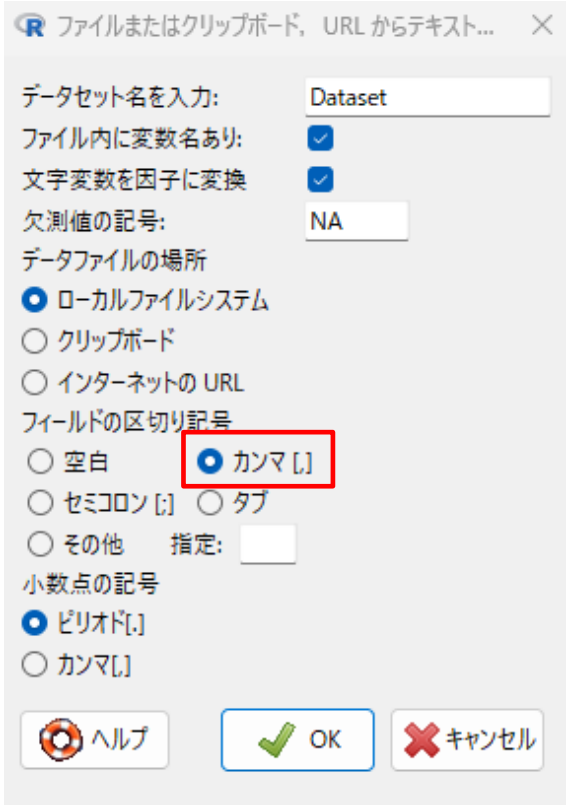
- Rstudio　＞ Dataset<-read.csv(“ozone.csv”)

　　又は

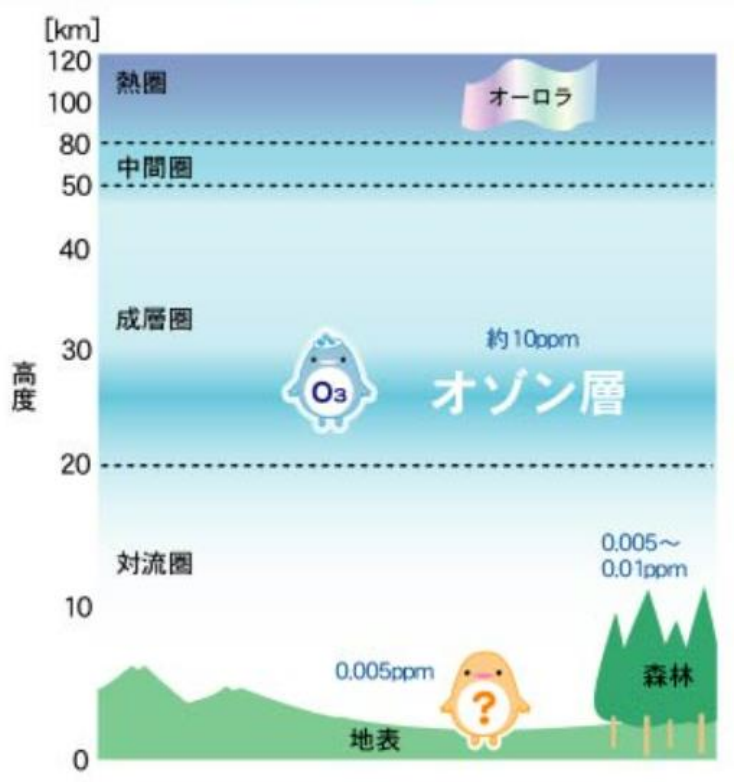
- Rコマンダー　(データ) → (データインポート) → (テキストファイルまたはクリップボード・・・) →

　✔OKを選択して、ozone.csv を指定する

④ 演習データが Dataset に読込まれる



オゾン濃度データ（再掲）



https://www.rgl.co.jp/html/page120.html

気圧500mmbに達する**高度**　　**逆転温度**　　**逆転基準温度**

風速　**湿度**　**気温**　**気圧差**　**視界**　**測定日**

オゾン濃度

O3	vh	wind	humidity	temp	ibh	dpg	ibt	vis	doy
3	5710	4	28	40	2693	-25	87	250	33
5	5700	3	37	45	590	-24	128	100	34
5	5760	3	51	54	1450	25	139	60	35
6	5720	4	69	35	1568	15	121	60	36
4	5790	6	19	45	2631	-33	123	100	37
4	5790	3	25	55	554	-28	182	250	38
6	5700	3	73	41	2083	23	114	120	39
7	5700	3	59	44	2654	-2	91	120	40
4	5770	8	27	54	5000	-19	92	120	41
6	5720	3	44	51	111	9	173	150	42
5	5760	6	33	51	492	-44	181	40	43
4	5780	6	19	54	5000	-44	135	200	44
4	5830	3	19	58	1249	-53	243	250	45
7	5870	2	19	61	5000	-67	186	200	46
5	5840	5	19	64	5000	-40	174	200	47
9	5780	4	59	67	639	1	189	150	48
4	5680	5	73	52	393	-68	210	10	49
3	5720	4	19	54	5000	-66	126	140	50
4	5760	3	19	54	5000	-58	111	250	51
4	5730	4	26	58	5000	-26	111	200	52

ランダムフォレストの実行（回帰）

※　ランダムフォレスト初回はパッケージ
“**randomForest**” をインストールする

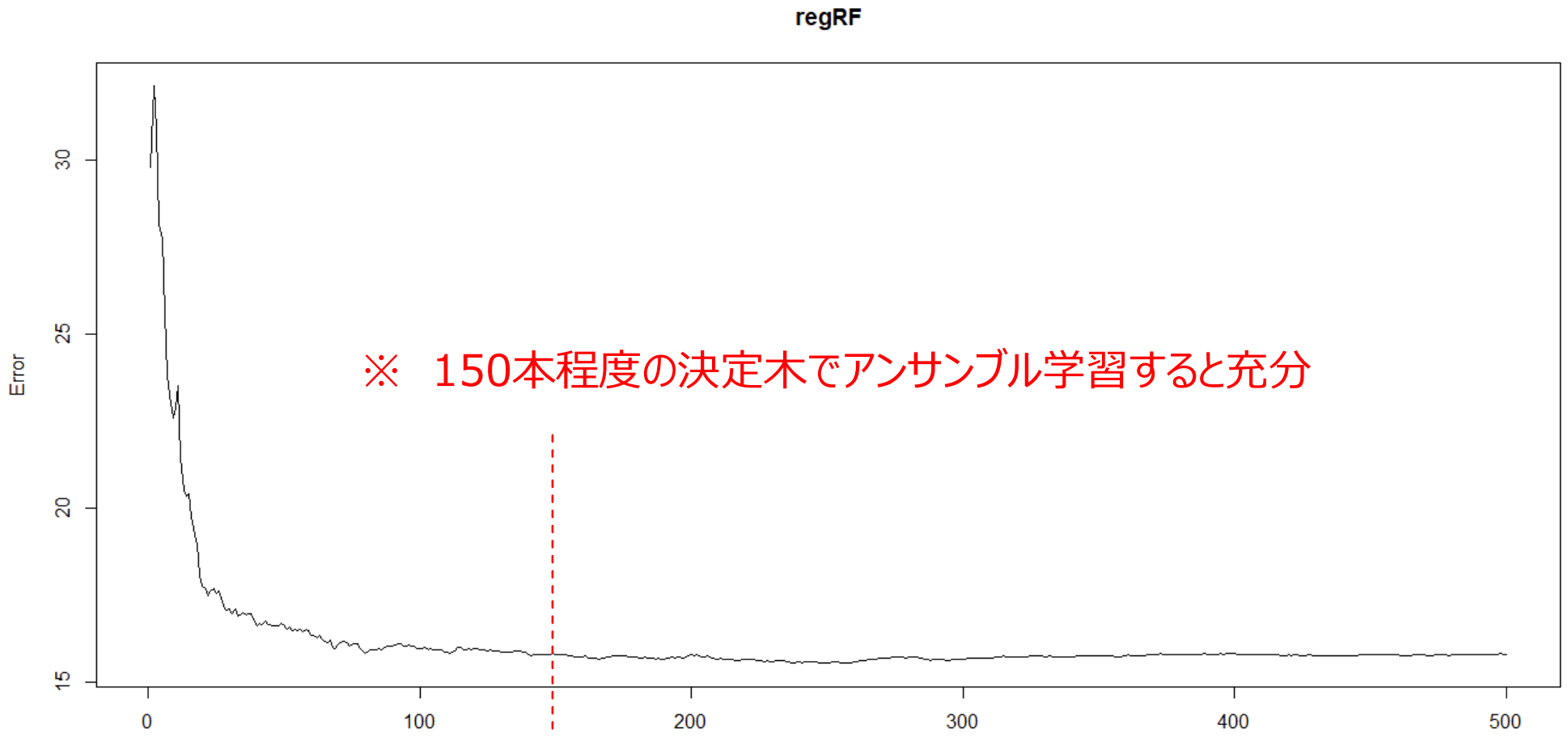
```
> library(randomForest)
> y<- as.vector(ozone[,1])
> x<- as.matrix(ozone[,-1])
> regRF<- randomForest(x,y)
> print(regRF)
> plot(regRF)
```

回帰(**regression**)型のランダムフォレストでは目的変数と説明変数(特徴量)を指定する

```
Call:
 randomForest(x = x, y = y)
               Type of random forest: regression
                     Number of trees: 500
No. of variables tried at each split: 3

          Mean of squared residuals: 15.81142
                    % Var explained: 75.29
```

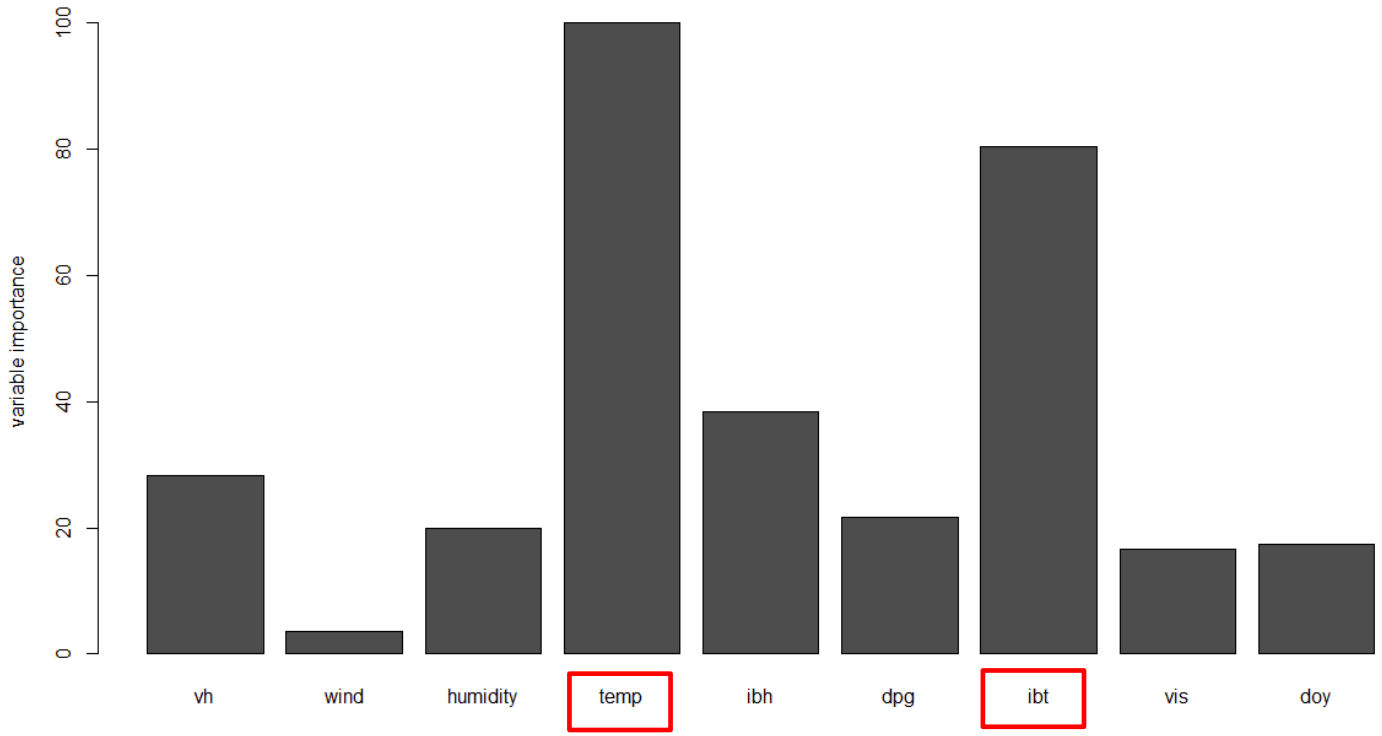
※　ランダムフォレストに採用した
決定木は**500本**、各決定木は
３つの特徴量でモデル化

ランダムフォレストの樹木増加に対する平均二乗誤差（回帰）



ランダムフォレストの変数重要度（回帰）

```
> vimp<- importance(regRF)
> rimp <- vimp/max(vimp)*100
> print(rimp)
          IncNodePurity
vh             28.376157
wind            3.655413
humidity       19.921509
temp          100.000000
ibh            38.438453
dpg            21.687413
ibt            80.343881
vis            16.520258
doy            17.450920
> barplot(t(rimp),ylab="variable importance")
```



※オゾン濃度の回帰に最も重要な２つの変数

気温

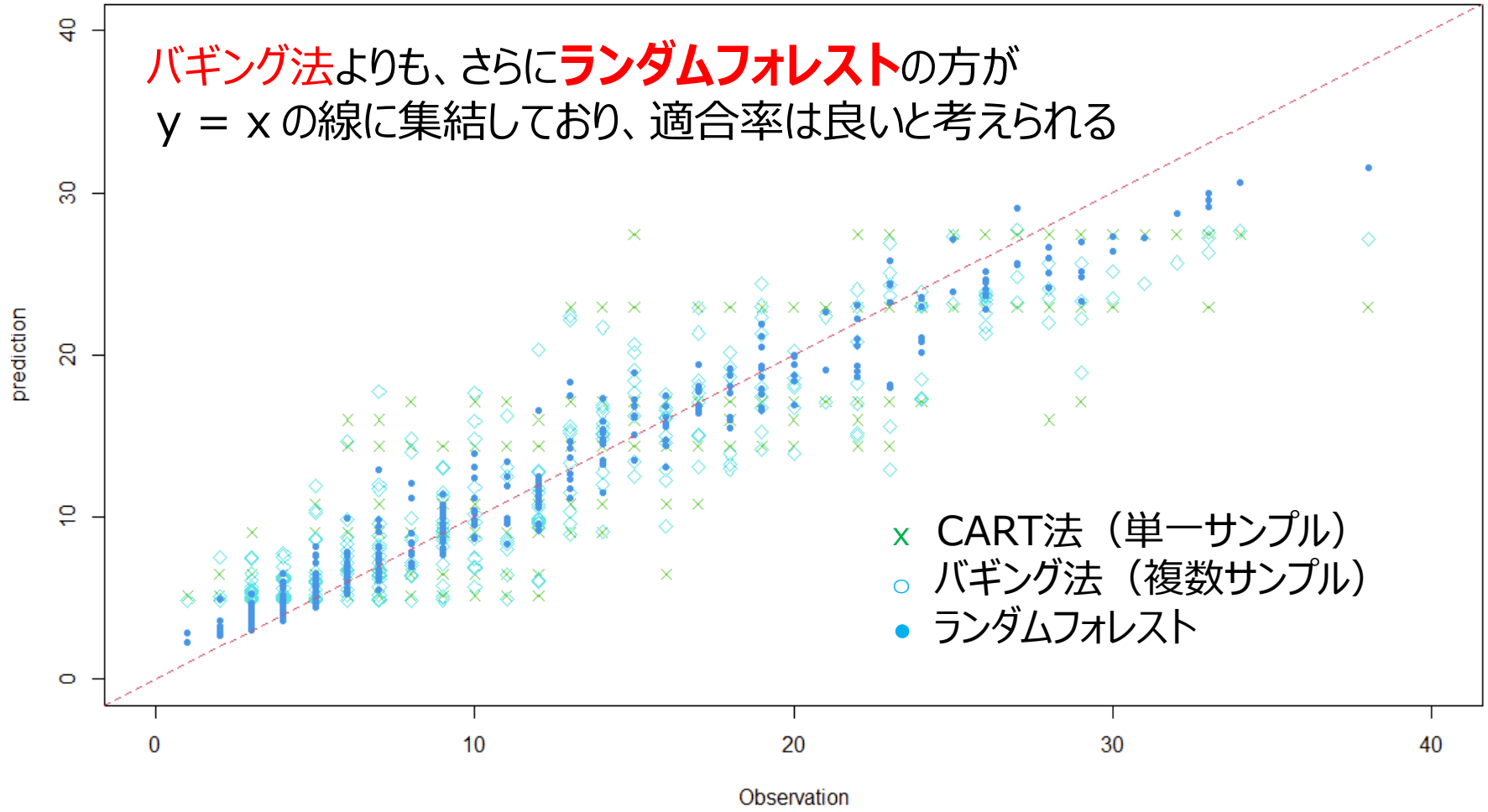
逆転基準温度

各種アルゴリズムによる予測評価の比較

※　単純の決定木、バギングによる決定木、ランダムフォレストによる予測評価

```
> library(ipred)
> library(rpart)
> yhat<-predict(regRF,x)
> predBagg<-predict(bagging(O3~.,data=ozone,nbagg=500),ozone)
> predCART<-predict(rpart(O3~.,data=ozone),ozone)
> matplot(ozone[,1],cbind(predCART,predBagg,yhat),pch=c(4,5,20),col=c(3,5,4))
> library(ipred)
> library(rpart)
> yhat<-predict(regRF,x)
> predBagg<-predict(bagging(O3~.,data=ozone,nbagg=500),ozone)
> predCART<-predict(rpart(O3~.,data=ozone),ozone)
> matplot(ozone[,1],cbind(predCART,predBagg,yhat),pch=c(4,5,20),col=c(3,5,4)
+ ,xlim=c(0,40),ylim=c(0,40),xlab="Observation",ylab="prediction")
> abline(0,1,col=2,lty=2)
```

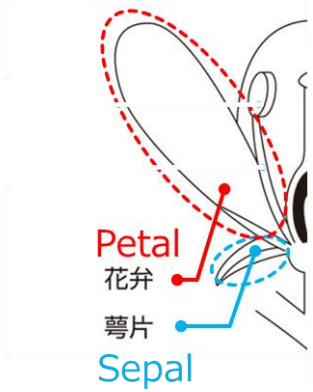
ランダムフォレストは適合と予測に優れる！



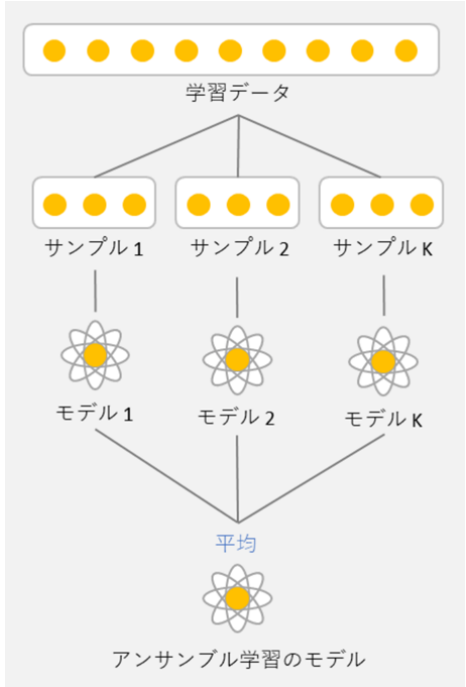
課題：IRISデータを用いたアンサンブル学習

IRISデータを用いて、ランダムフォレストでアンサンブル学習を実行し、学習結果をについて評価しなさい。

アヤメの種類によって、
<u>花弁</u>と**<u>がく片</u>**の幅と長さで分類するためのデータとして用いられる



Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
5.1	3.5	1.4	0.2	setosa
4.9	3.0	1.4	0.2	setosa
4.7	3.2	1.3	0.2	setosa
4.6	3.1	1.5	0.2	setosa
5.0	3.6	1.4	0.2	setosa
5.4	3.9	1.7	0.4	setosa
4.6	3.4	1.4	0.3	setosa
5.0	3.4	1.5	0.2	setosa
4.4	2.9	1.4	0.2	setosa
4.9	3.1	1.5	0.1	setosa
7.0	3.2	4.7	1.4	versicolor
6.4	3.2	4.5	1.5	versicolor
6.9	3.1	4.9	1.5	versicolor
5.5	2.3	4.0	1.3	versicolor
6.5	2.8	4.6	1.5	versicolor
5.7	2.8	4.5	1.3	versicolor
6.3	3.3	4.7	1.6	versicolor
4.9	2.4	3.3	1.0	versicolor
6.6	2.9	4.6	1.3	versicolor
5.2	2.7	3.9	1.4	versicolor
6.3	3.3	6.0	2.5	virginica
5.8	2.7	5.1	1.9	virginica
7.1	3.0	5.9	2.1	virginica
6.3	2.9	5.6	1.8	virginica
6.5	3.0	5.8	2.2	virginica
7.6	3.0	6.6	2.1	virginica
4.9	2.5	4.5	1.7	virginica
7.3	2.9	6.3	1.8	virginica
6.7	2.5	5.8	1.8	virginica
7.2	3.6	6.1	2.5	virginica




データマイニングを楽しもう！