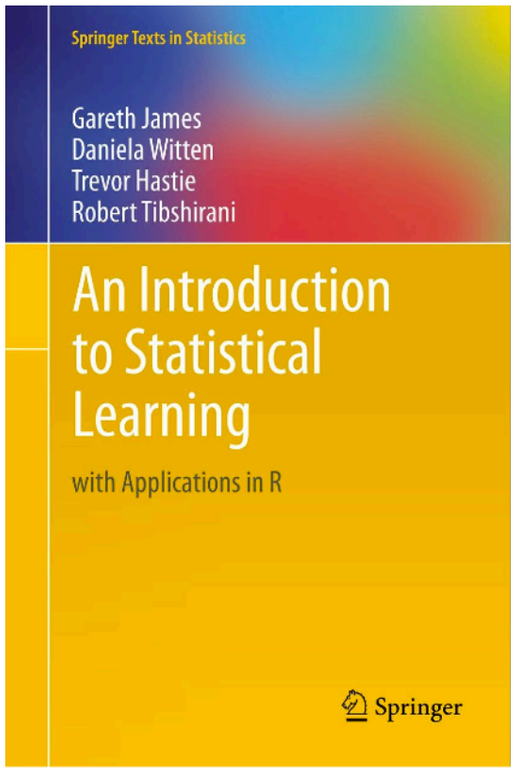
演習

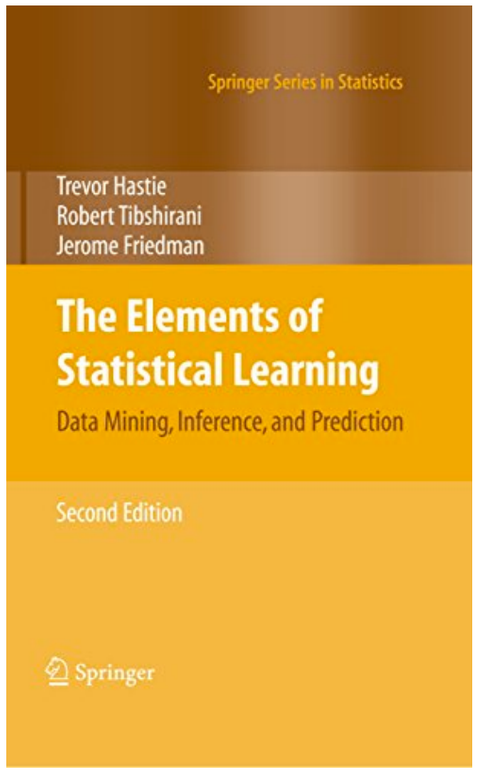
ML入門, 線形回帰分析

片山 翔太

参考書

- G.James他, An introduction to statistical learning with applications in R
  - 主にこの書籍を参考にしています
- T.Hastie他, The elements of statistical learning
  - より理論的なことが知りたければこっちを参照





Introduction

- **統計的学習とは**
  - データを理解するためのツールの総称
  - 教師**あり**(supervised)と教師**なし**(unsupervised)

	入力	出力	備考
教師あり	○	○	入力から出力を予測(or 推定)
教師なし	○	✖	入力の構造を学習

教師あり学習

入力 x

入力
x_1
x_2
x_3
x_4
…
…
x_n

$f(x)$

Ex. $f(x) = \alpha + \beta x$

出力 y

出力
y_1
y_2
y_3
y_4
…
…
y_n

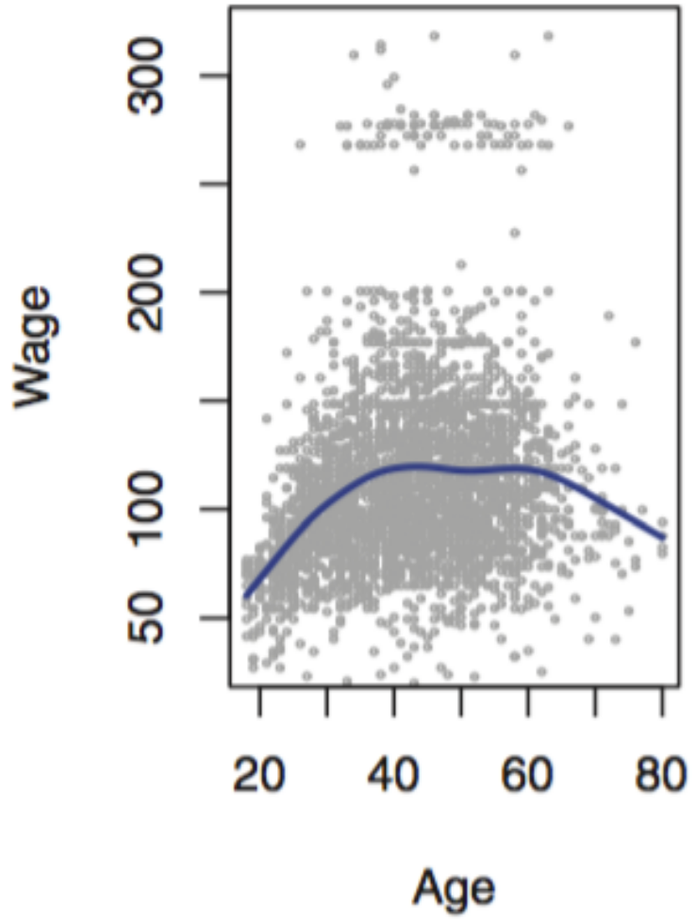
- 入出力が**多変数 (多次元)** になることもある
- 基本的にやってることは **f の推定**

学習とも呼ばれる

給料(wage)データ

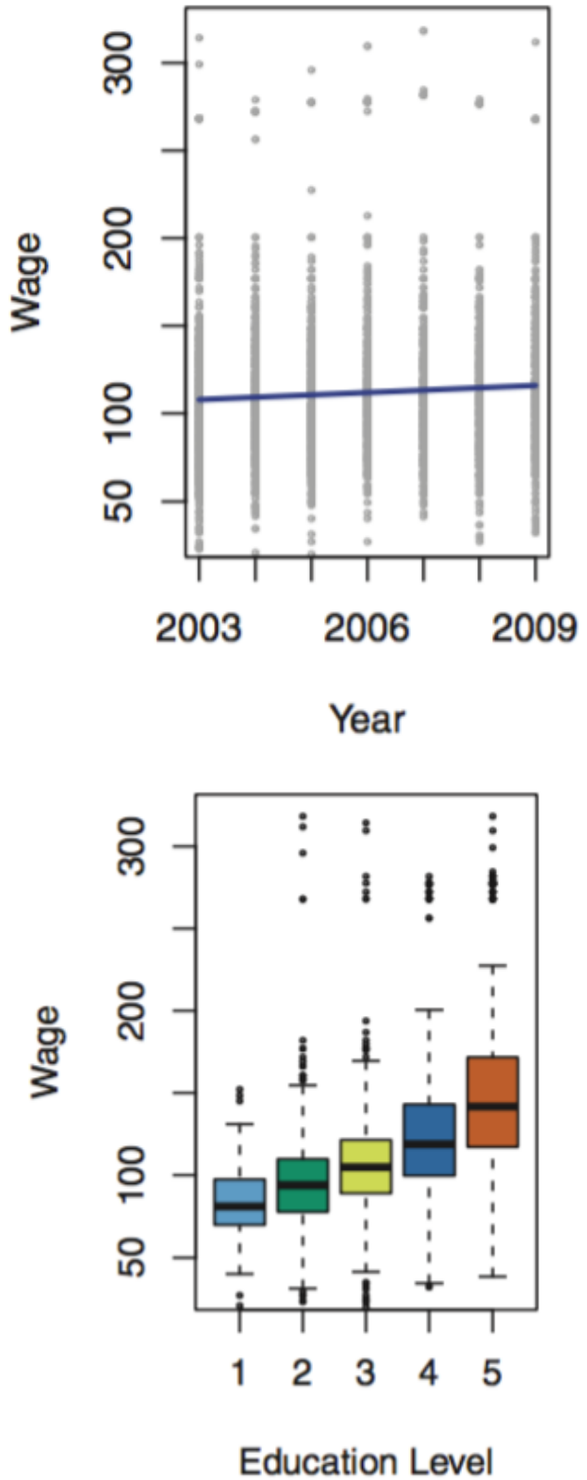
- **データの概要**
  - アメリカ合州国(東海岸)の男性
  - 給料に関連する要因を調べたい (教師あり学習)
    - 特に年齢，教育レベル，カレンダー年

- **給料と年齢の関係**
  - 給料×年齢の散布図
    - 年齢が増加すると給料も増加
    - ただし60歳付近から減少する



給料(wage)データ

- **給料とカレンダー年の関係**
  - 僅かに上昇傾向
  - 2003年から2009年にかけて約$10,000上昇
- **給料と教育レベルの関係**
  - 低レベル(1)から高レベル(5)
  - 教育レベルが高いほど高給料
- **給料の予測のためには...**
  - 年齢, カレンダー年, 教育レベルの組み合わせ
  - 線形 (第5回) と非線形 (秋学期)



株取引 (stock market) データ

- **データの概要**
  - 2001年から2005年までの日次S&P500株価指標
  - 「過去5日間の変動」から「当日の変動」を予測
    - ただしUp or Downの２択で十分
    - 教師あり学習

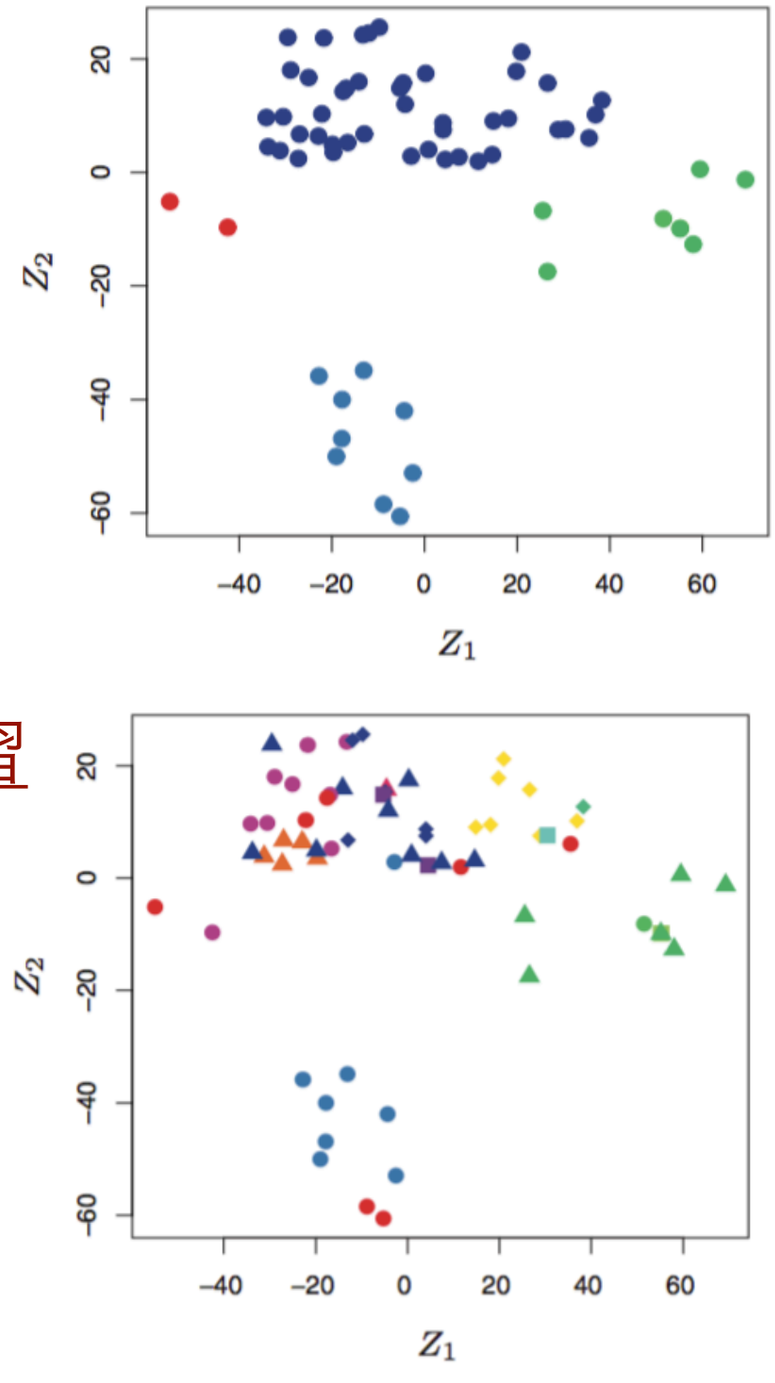
- **備考**
  - 賃金データの場合は出力が**連続値** → **回帰**問題
  - 株取引データの場合は出力が**２値** → **分類**問題
    - 3値以上(多値)の場合も分類問題

教師なし学習

- **データは入力のみ**
  - 入力の持つ構造自体に興味がある
  - 基本的には**クラスタリング**がメイン

- **遺伝子発現データ (gene expression)**
  - 64癌細胞株の6,830遺伝子発現量
  - 細胞株をグループ分けしたい
    - 出力がない(予測が目的でない)教師なし学習

- **主成分分析**
  - 6,830次元を2次元に圧縮
  - 上：視覚による色分け
  - 下：実際のグループ情報 (14グループ)



ここまでのまとめ

- **統計的学習の分類**
  - 教師あり学習と教師なし学習
    - 予測が目的であるかどうかで分かれる

- **教師あり学習**
  - 回帰問題(給料)と分類問題(株価)
    - 予測したい出力が連続であるか離散であるかで分かれる

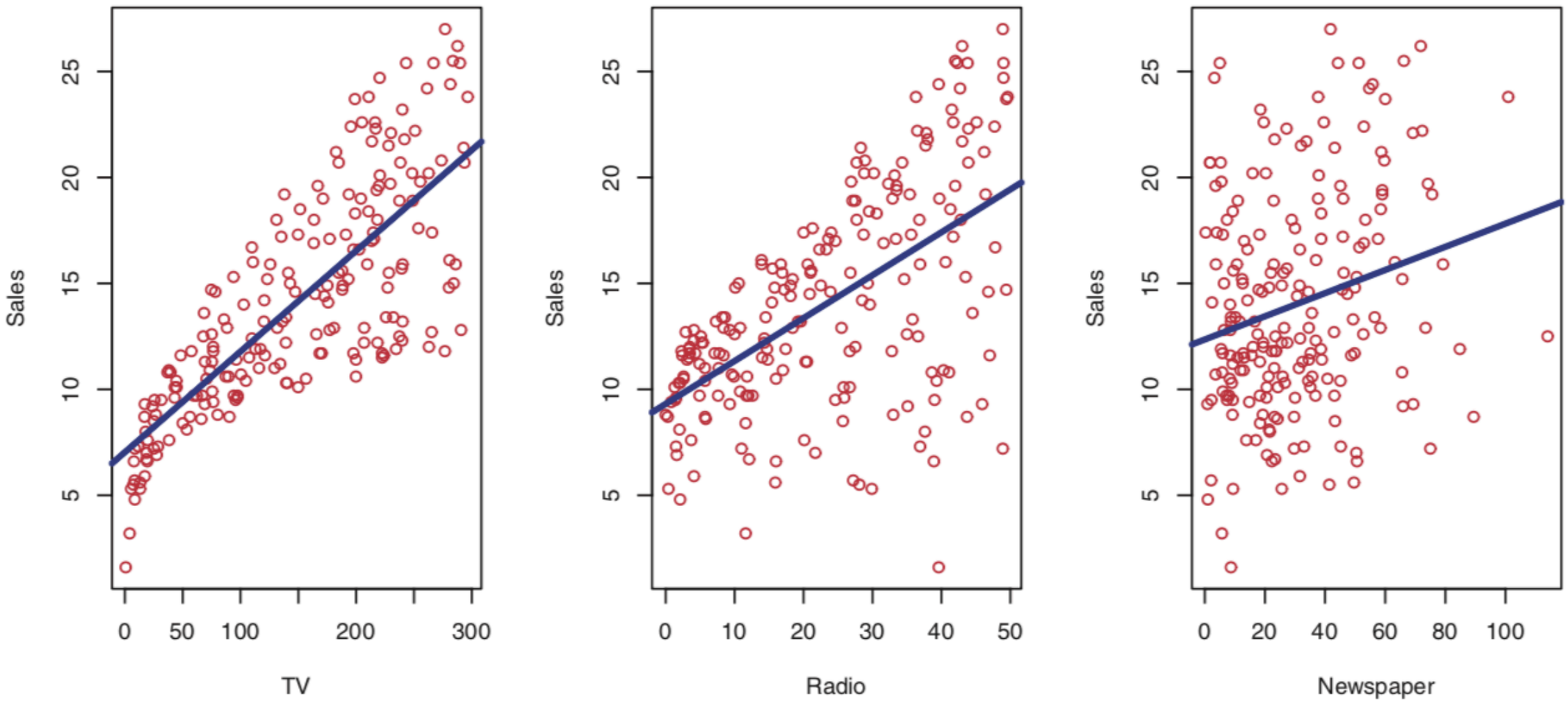
- **教師なし学習**
  - 出力がなく入力だけのデータ
  - 主にクラスタリングが目的

教師あり学習：数学的な記述

- **用語の定義**
  - 入力：**説明変数** or **予測変数**
  - 出力：**目的変数** or **結果変数**

入力			出力
x_{11}	x_{21}	x_{31}	y_1
x_{21}	x_{22}	x_{32}	y_2
x_{31}	x_{23}	x_{33}	y_3
x_{41}	x_{24}	x_{34}	y_4
…	…	…	…
…	…	…	…
x_{n1}	x_{2n}	x_{3n}	y_n

- **Example**
  - 3つの媒体の広告費と売り上げ
  - 説明変数：テレビ(x_1)，ラジオ(x_2)，新聞(x_3)の広告費
  - 目的変数：商品の売り上げ(y)


Sales
5
10
15
20
25
0
50
100
200
300
TV
Sales
5
10
15
20
25
0
10
20
30
40
50
Radio
Sales
5
10
15
20
25
0
20
40
60
80
100
Newspaper

一般的な定式化

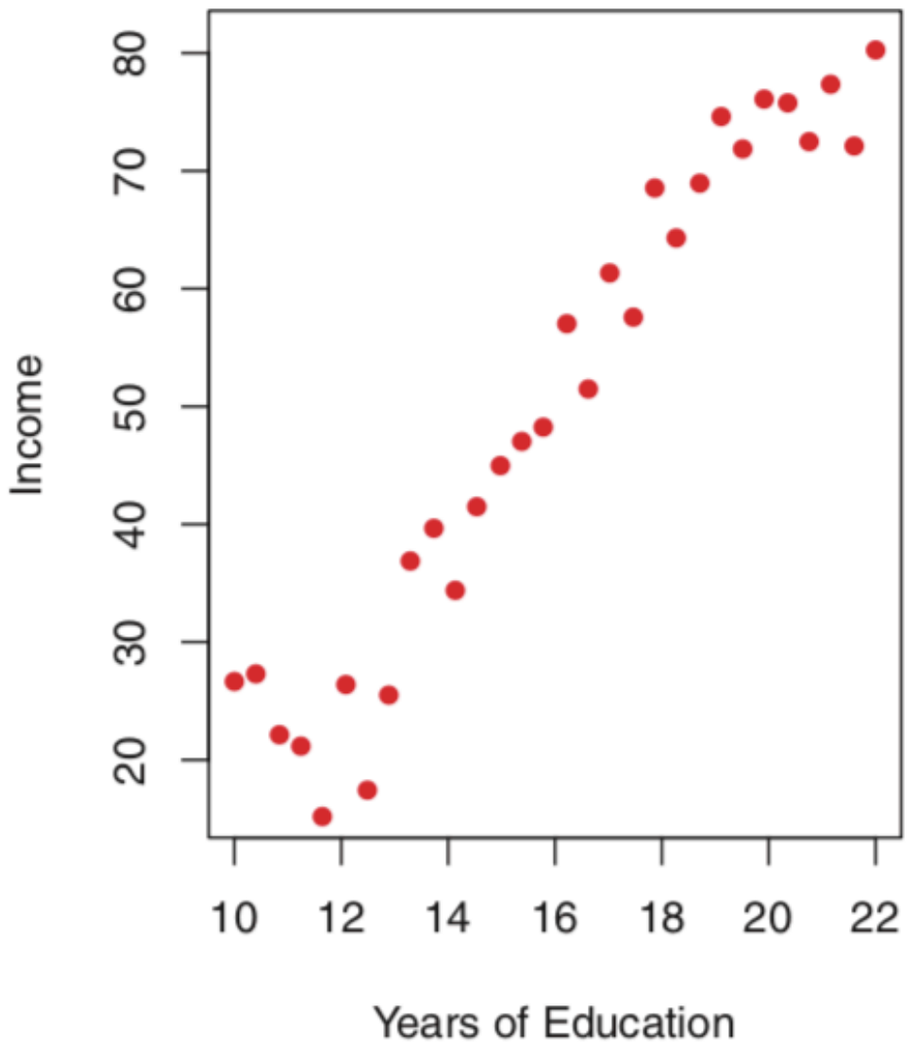
- **説明変数 $\boldsymbol{X} = (X_1, X_2, \ldots, X_p)$ と目的変数 Y**
  - これらの関係性を次のように想定

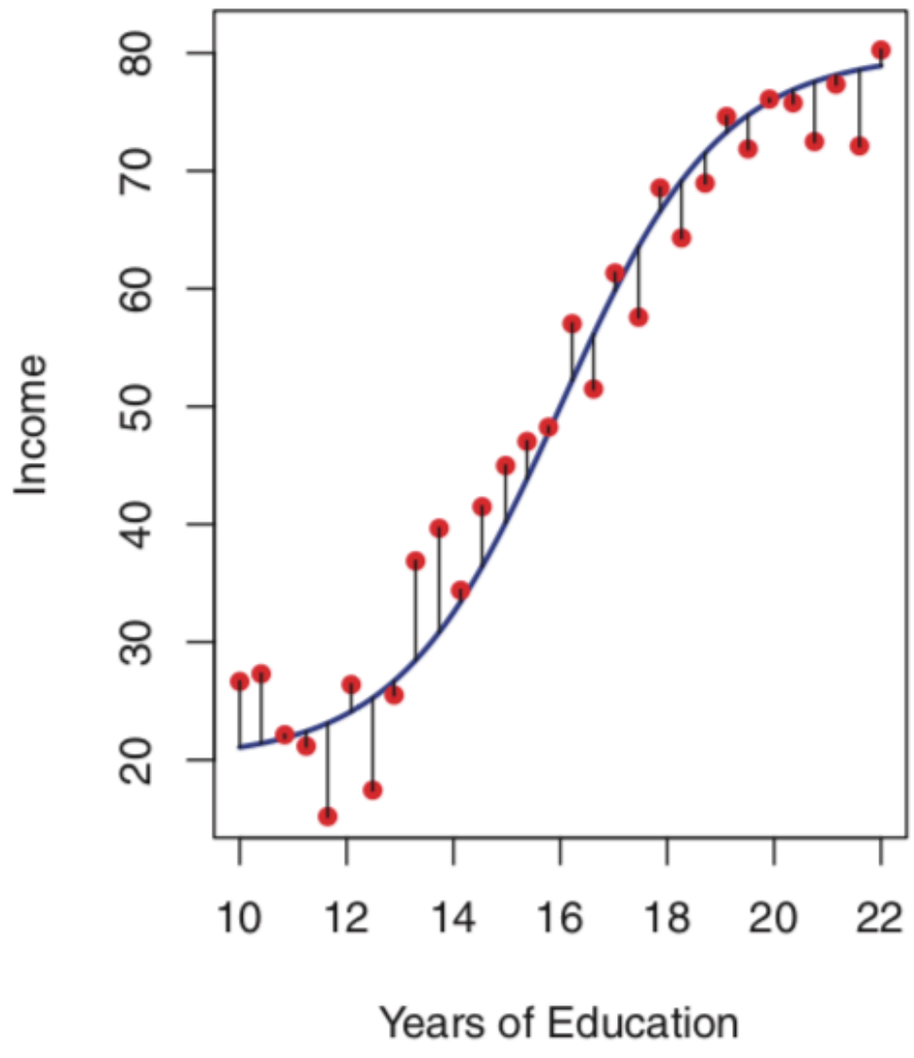
$$Y = f(\boldsymbol{X}) + \varepsilon$$

- 関数 $f(\cdot)$ はどんな形か分からない (未知)
- 誤差 ε : 関数f(X)では説明できない確率的な変動

Example

- **教育年数と収入の関係性**





教師あり学習

得られたデータ(X,Y)からfを推定

- **目的(1) 予測** $\hat{Y} = \hat{f}(X)$
  - Xが簡単に得られる一方でYが難しい場合に有用
  - Ex. 血圧(X)と副作用の大きい薬への患者の反応(Y)
  - 関数fの正確な形が分かる必要はない
    - 予測の精度さえ良ければOK
- **関連手法**
  - ランダムフォレスト
  - 非線形回帰 (スプライン回帰, Gaussian process回帰)
  - 深層学習

教師あり学習

得られたデータ(X,Y)からfを推定

- **目的(2) 推測**
  - XのYの関係性を知りたい場合に有用
  - Ex. 広告費(X)と売り上げ(Y)
  - 関数fの正確な形 (モデル)が必要

- **関連手法**
  - 線形回帰, ロジスティック回帰
  - サポートベクトルマシン
  - Explainable ML (説明可能な機械学習)

教師あり学習

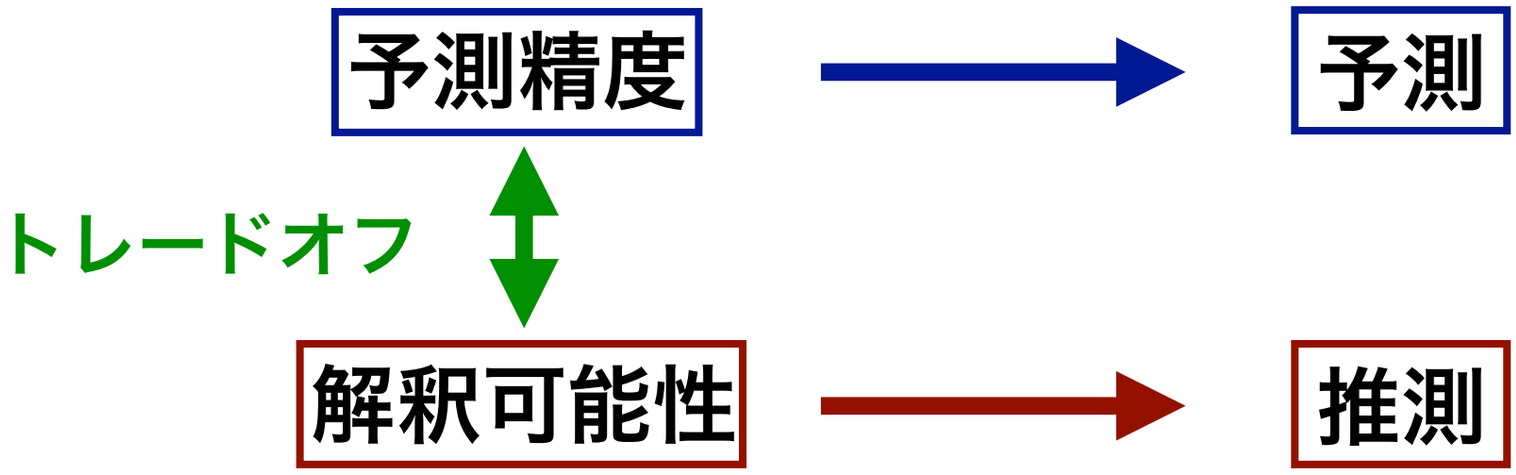
得られたデータ(X,Y)からfを推定

- **目的(3) 予測 + 推測**
  - Ex. 不動産の価値(Y)とその特性(X)
    - 予測：適切な不動産価格を決定したい
    - 推測：海や川が見えたらいくら価値が上がるのか

suumo

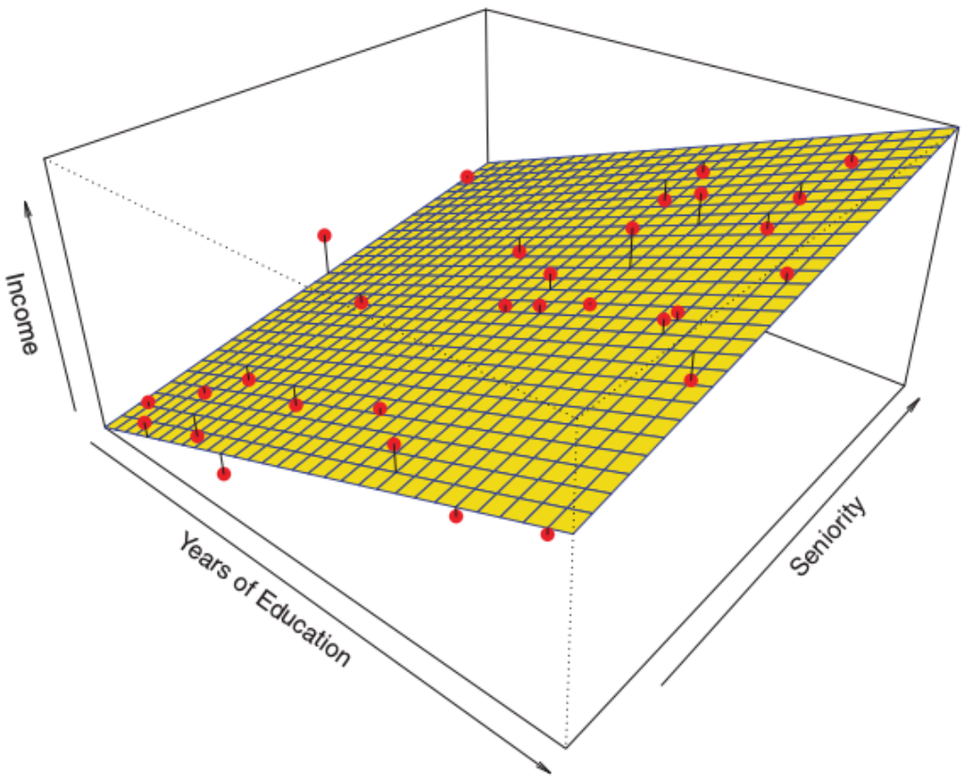
suumo

予測精度と解釈可能性のトレードオフ

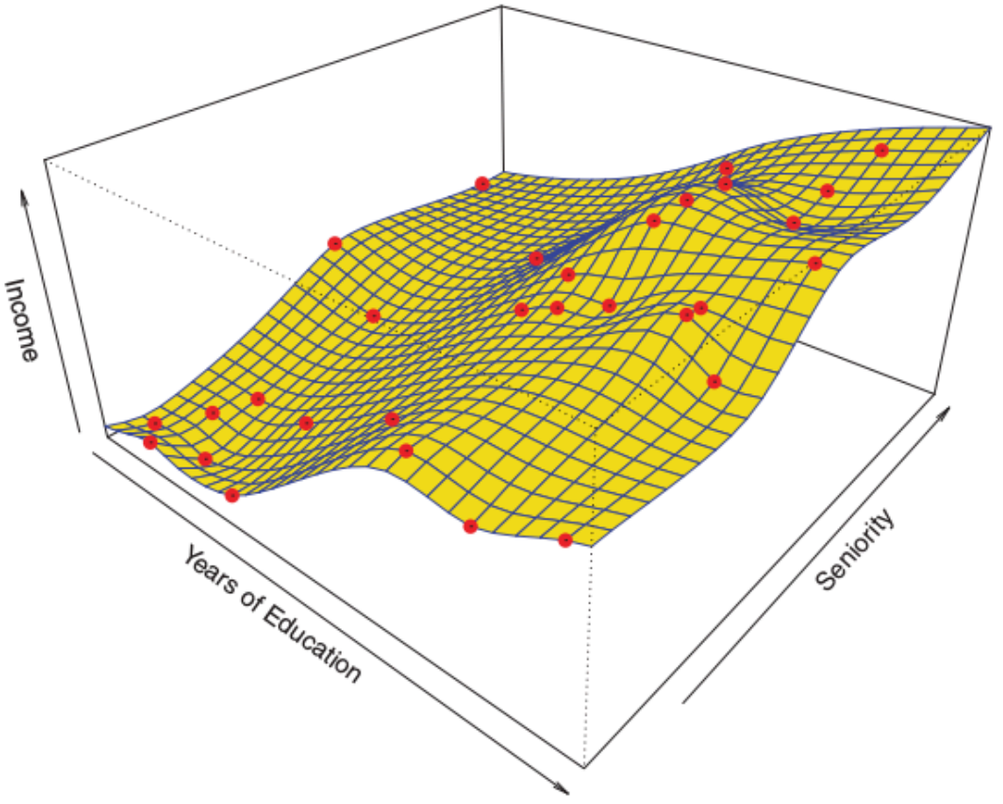


- **予測精度を向上**
  - 関数fを複雑に設定する (or ブラックボックス化)
  - Ex. $Y = \beta_1 X_1^2 + \beta_2 \log X_2 + \beta_3 X_1^{1/5} X_3^3 + \varepsilon$
- **解釈可能性**
  - 関数fを簡潔に設定する
  - Ex. $Y = \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \varepsilon$

・教育年数・年齢 (X1, X2)と収入(Y)の関係性



線形回帰



非線形回帰

まとめ

- **教師あり学習**
  - 定式化： $Y = f(\boldsymbol{X}) + \varepsilon$
- **教師あり学習の目的**
  - 予測 $\hat{Y} = \hat{f}(X)$
  - 推測 (XとYの間の関係性を解釈)
  - 予測 + 推測
- **予測精度と解釈可能性**
  - どちらか一方を向上させると他が下がる
  - トレードオフが存在

線形回帰分析入門

データ

- **TV, ラジオ, 新聞への広告料と売上**
  - 製品の広告に使った費用とその売上

- **Goal**
  - 広告料と売上の関係性を見たい
    - どの広告媒体が売上に効いているか
  - 広告料だけ与えられた場合に売上を予測できるか

先ずはデータを眺める

作業ディレクトリの指定 (各自の設定によって異なる)

```
> setwd("/Users/shota/R")
> adv <- read.csv("Advertising.csv",header=T,fileEncoding="Shift_JIS")
> adv <- adv[,-1] ##remove first column
> head(adv)
     TV radio newspaper sales
1 230.1  37.8      69.2  22.1
2  44.5  39.3      45.1  10.4
3  17.2  45.9      69.3   9.3
4 151.5  41.3      58.5  18.5
5 180.8  10.8      58.4  12.9
6   8.7  48.9      75.0   7.2
>
```

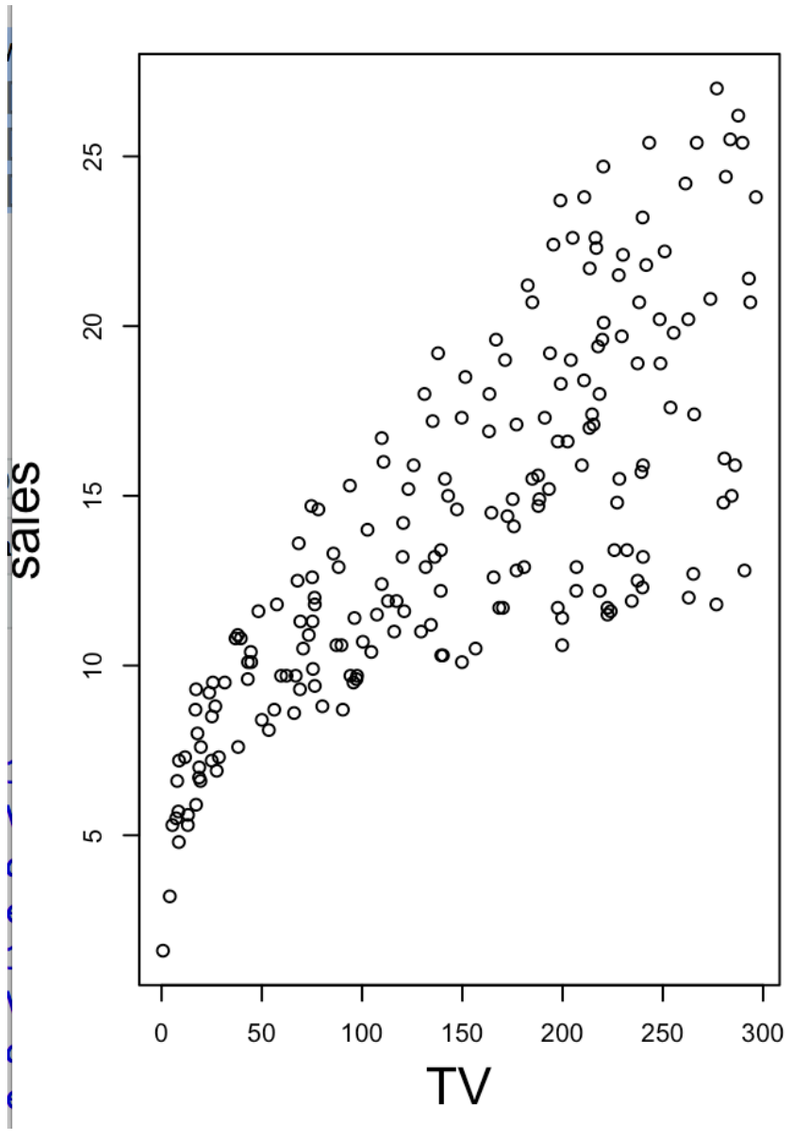
csvファイルを読み込む

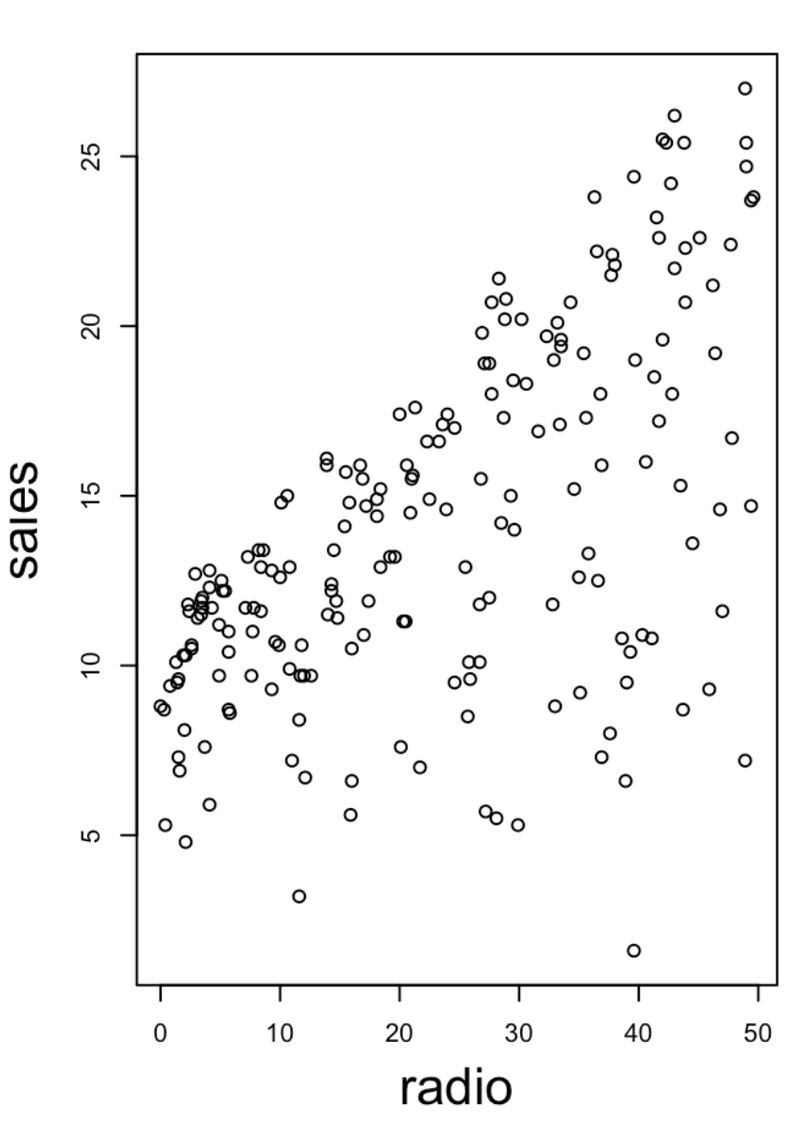
salesの単位：千台

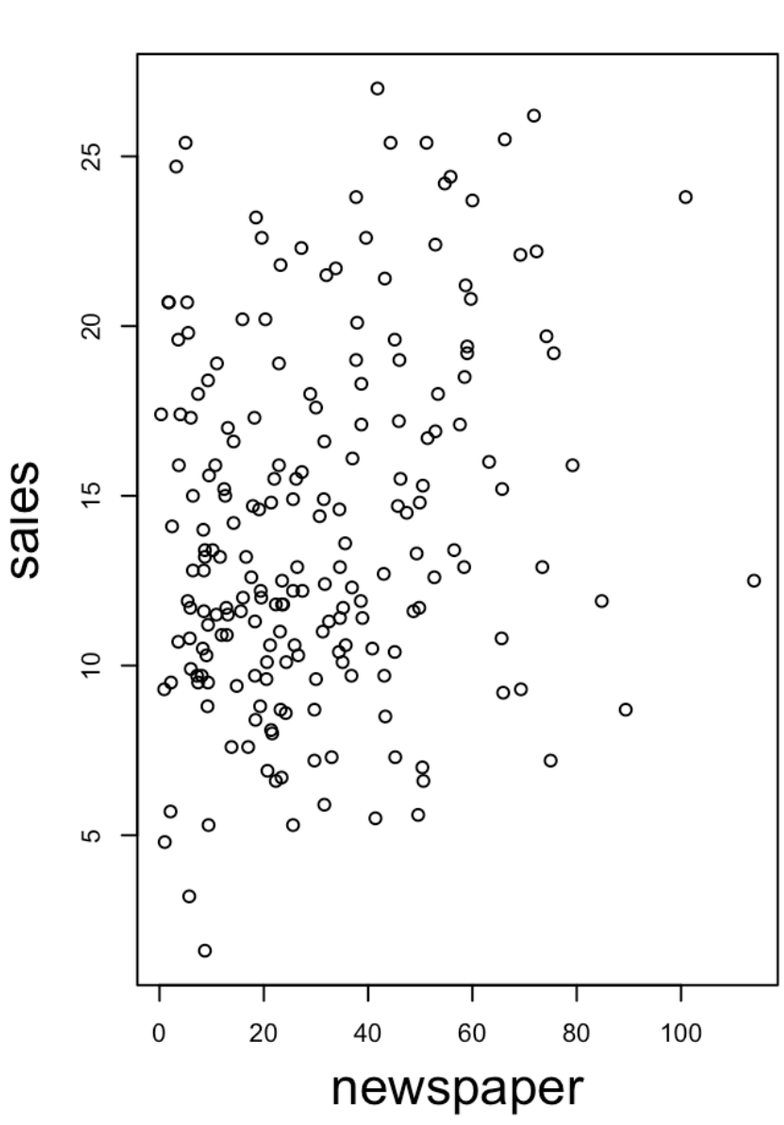
その他の単位：$1000

先ずはデータを眺める

- **Exercise**
  - 次の散布図を描こう







線形回帰分析 (単変数)

- **モデル式**

$$Y = \beta_0 + \beta_1 X + \varepsilon, \quad \varepsilon \sim N(0, \sigma^2)$$

切片　傾き　　誤差

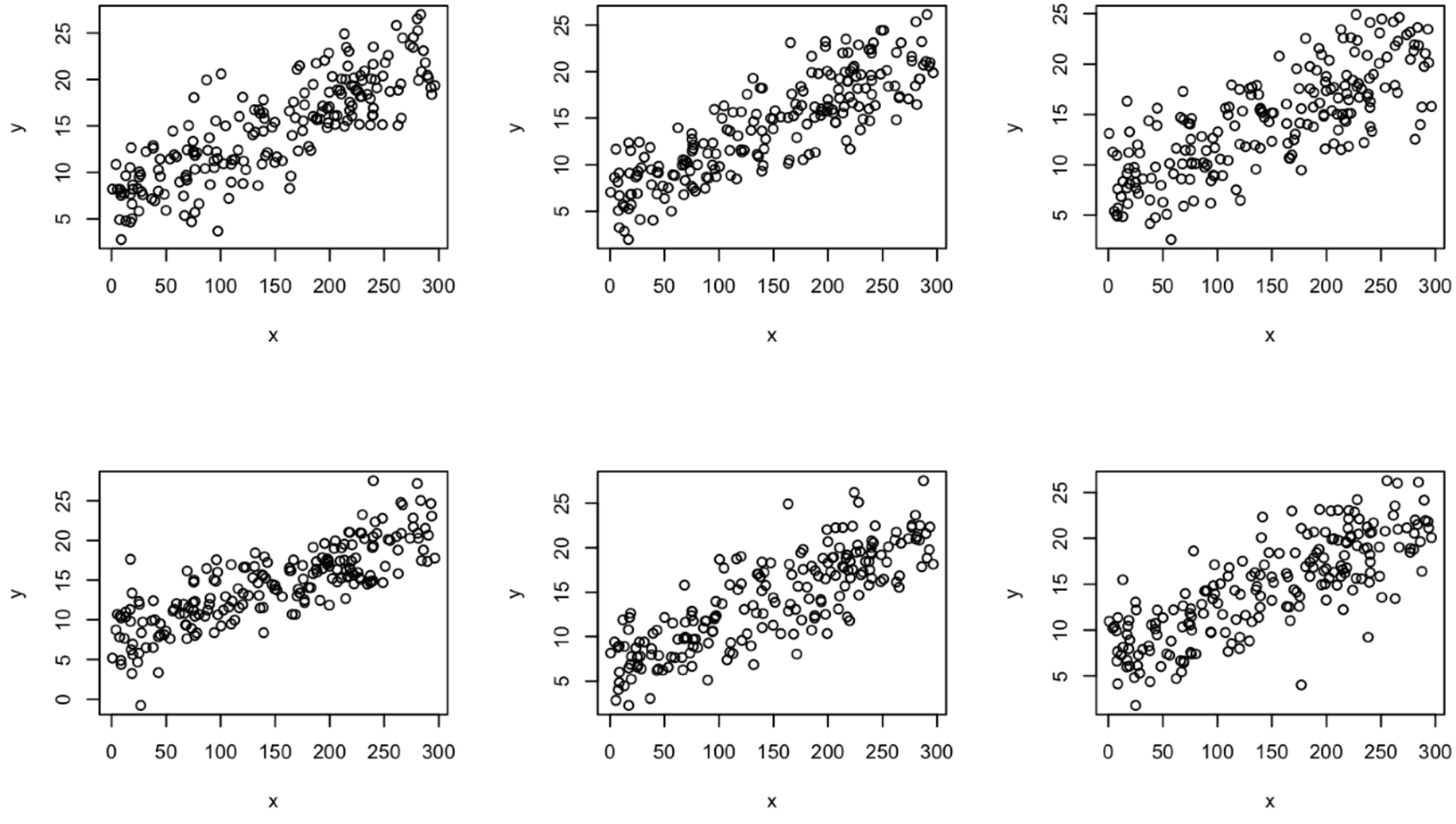
- Y : sales, X : TV, ラジオ or 新聞への広告料

このモデルに従って
データが発生していると**想定する**

線形回帰分析 (単変数)

- **例えば...** $Y = 5 + 0.8X + \varepsilon, \quad \varepsilon \sim N(0, 3)$

```
x <- as.vector(adv[,1])
par(mfrow=c(2,3))
plot(x,7+0.05*x+rnorm(200,0,3),ylab="y")
```



線形回帰分析 (単変数)

- **パラメータの推定 (最小二乗法)**

$$\sum_{i=1}^{n}(y_i - \beta_0 - \beta_1 x_i)^2 \longrightarrow \text{minimize} \longrightarrow \hat{\beta}_0,\ \hat{\beta}_1$$

- **Rでの実装**
  - Y : 売上, X : TVへの広告料

```
> result <- lm(sales ~ TV, data=adv)
> summary(result)
```

結果の解釈

$\hat{\beta}_0,\ \hat{\beta}_1$

$H_0 : \beta_0 = 0$ vs. $H_1 : \beta_0 \neq 0$

$H_0 : \beta_1 = 0$ vs. $H_1 : \beta_1 \neq 0$

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 7.032594   0.457843   15.36   <2e-16 ***
TV          0.047537   0.002691   17.67   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

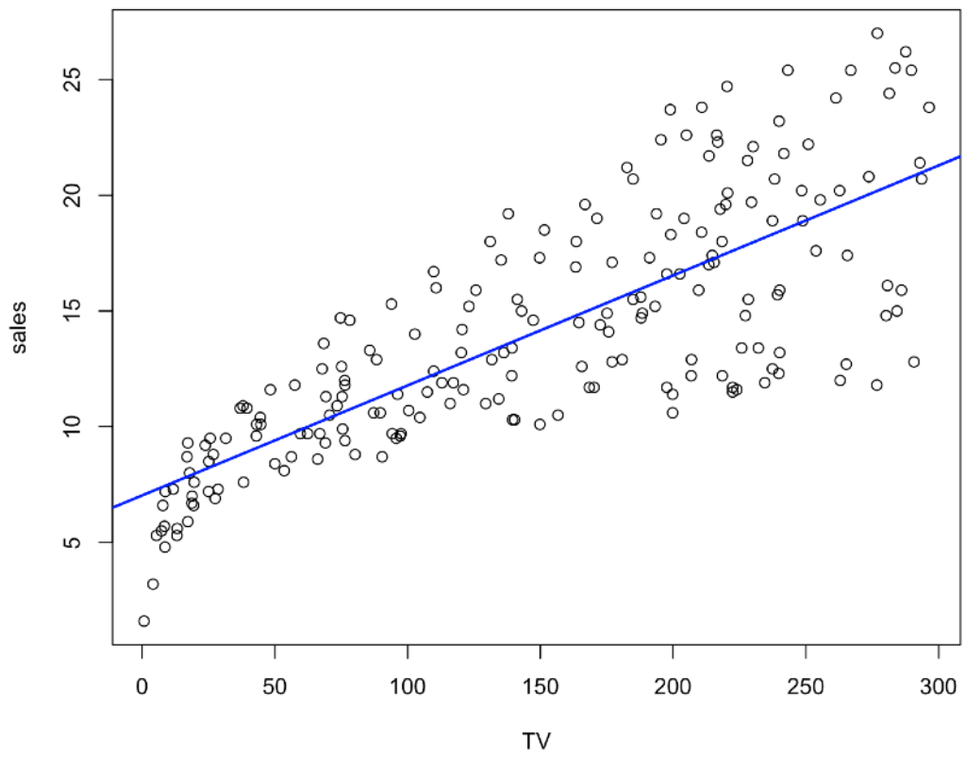
$$\mathbb{P}(|\hat{\beta}_j - \beta_j| \leq 2\ \texttt{Std.Error}) \simeq 0.95$$

95%信頼区間： $[\hat{\beta}_j - 2\texttt{Std.Error}, \hat{\beta}_j + 2\texttt{Std.Error}]$

結果の解釈

- **回帰直線の推定** $\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x$

- **Exercise**
  - TV × salesの散布図に回帰直線を重ね書きしよう
  - Rで直線はabline(切片, 傾き)




```
> coefficients(result)
(Intercept)          TV
 7.03259355  0.04753664
>
```

結果の解釈

- **予測値** $\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i$

```
Residuals:
    Min      1Q  Median      3Q     Max
-8.3860 -1.9545 -0.1913  2.0671  7.2124
```

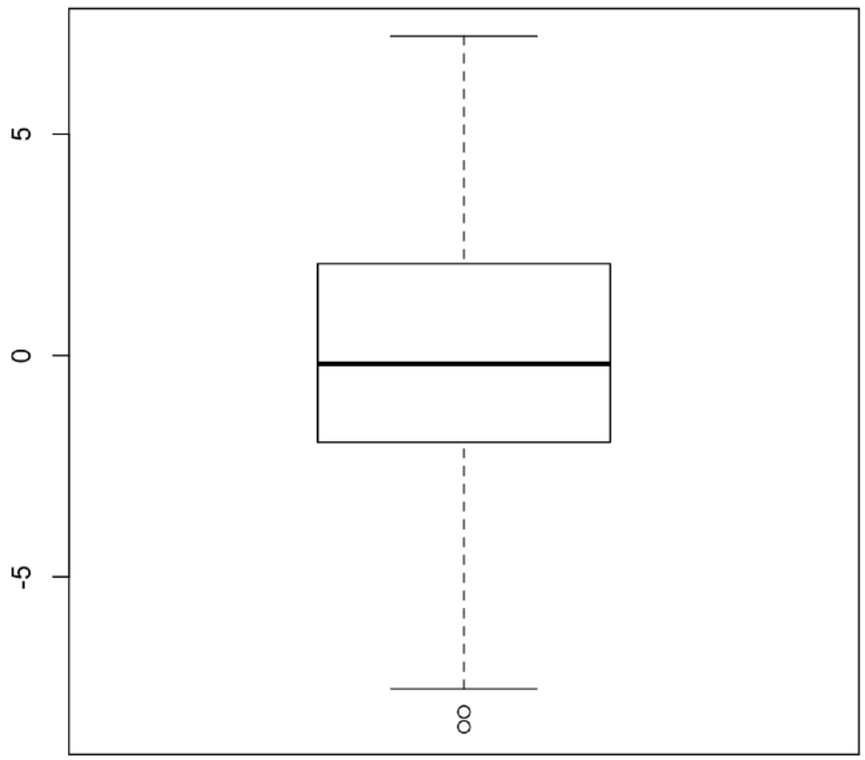
残差

$y_i - \hat{y}_i$

残差の箱ひげ図

```
> boxplot(adv[,"sales"] - predict(result))
```

予測値の計算



結果の解釈

実測値と予測値のずれ $\sqrt{\frac{1}{n-2}\sum_{i=1}^{n}(y_i - \hat{y}_i^2)}$

```
Residual standard error: 3.259 on 198 degrees of freedom
Multiple R-squared:  0.6119,    Adjusted R-squared:  0.6099
F-statistic: 312.1 on 1 and 198 DF,  p-value: < 2.2e-16
```

$$1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \in [0, 1]$$

当てはまりの良さ

- **Exercise**
  - ラジオ, 新聞に対しても同様に回帰し，次の問いに答えよ

- **Question**
  - 一番当てはまりの良い回帰モデルはどれ？
  - ラジオに広告費を$25000(データでは25)を使った場合の売上は？
  - 新聞への広告費を$1000増やすと売上はどの程度伸びる？

単変量から多変量へ

より複雑な現象を捉える

- **多変量線形回帰モデル**

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p + \varepsilon, \quad \varepsilon \sim N(0, \sigma^2)$$

- **売上データの場合**

ひとつの広告費用で売上を説明するよりも**合理的**

$$\texttt{sales} = \beta_0 + \beta_1 \texttt{TV} + \beta_2 \texttt{radio} + \beta_3 \texttt{newspaper} + \varepsilon$$

- **Rで実装**

```
> result <- lm(sales ~ TV + radio + newspaper, data=adv)
> summary(result)

Multiple R-squared:  0.8972,    Adjusted R-squared:  0.8956
```

結果の解釈を行うために：準備 (1)

結果変数

$$\boldsymbol{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}$$

説明変数

$$\boldsymbol{X} = \begin{pmatrix} 1 & x_{11} & \cdots & x_{1p} \\ 1 & x_{21} & \cdots & x_{2p} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_{n1} & \cdots & x_{np} \end{pmatrix}$$

パラメータ

$$\boldsymbol{\beta} = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_p \end{pmatrix}$$

誤差

$$\boldsymbol{\varepsilon} = \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{pmatrix}$$

各観測値に対するモデル式： $\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$

結果の解釈を行うために：準備 (2)

- **ベクトルとノルム**

$$\boldsymbol{a} = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \end{pmatrix}$$

長さmの適当な
ベクトル

$$\|\boldsymbol{a}\|_2 := \sqrt{a_1^2 + a_2^2 + \cdots + a_m^2} = \sqrt{\boldsymbol{a}^T \boldsymbol{a}}$$

二乗和の平方根
2ノルムとも呼ばれる

- **データから分かっているもの**
  - 結果変数と説明変数 (赤字)
- **推定したいもの**
  - パラメータ (青字)
- **どうにもできないもの**
  - 誤差 (ただし分散の推定は可能)

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$$

パラメータの推定 (最小二乗法)

$$\sum_{i=1}^{n} \left\{ y_i - (\beta_0 + \beta_1 x_{i1} + \cdots \beta_p x_{ip}) \right\}^2 \longrightarrow \text{minimize}$$

誤差が**二乗和の意味**で最小になるように推定

モデル式 (行列形式)

$$\sum_{i=1}^{n} \left\{ y_i - (\beta_0 + \beta_1 x_{i1} + \cdots \beta_p x_{ip}) \right\}^2 \longrightarrow \text{minimize}$$

$$\Vert \boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta} \Vert_2^2$$

βに関して最小に！

- **パラメータの求め方**
  - βに関して微分して0とおき，それを解く
  - 天下り的なやり方 (ほんとは幾何的な解釈)

Advanced

$$L := \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|_2^2 = \sum_{i=1}^{n} \{y_i - (\beta_0 + \beta_1 x_{i1} + \cdots \beta_p x_{ip})\}^2$$

$$\frac{\partial L}{\partial \boldsymbol{\beta}} := \begin{pmatrix} \frac{\partial L}{\partial \beta_0} \\ \frac{\partial L}{\partial \beta_1} \\ \vdots \\ \frac{\partial L}{\partial \beta_p} \end{pmatrix} = -2\boldsymbol{X}^T\boldsymbol{y} + 2\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{\beta}$$

これを0 (ベクトル) とおいてβに関して解くと...

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}$$

天下り的な確認

- **次のような分解が可能**　βは任意で成立

$$\|\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta}\|_2^2=\|\boldsymbol{y}-\boldsymbol{X}\hat{\boldsymbol{\beta}}\|_2^2+\|\boldsymbol{X}\hat{\boldsymbol{\beta}}-\boldsymbol{X}\boldsymbol{\beta}\|_2^2$$

- **Check!!**

$$\|\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta}\|_2^2 = \|\boldsymbol{y}-\boldsymbol{X}\hat{\boldsymbol{\beta}}+\boldsymbol{X}\hat{\boldsymbol{\beta}}-\boldsymbol{X}\boldsymbol{\beta}\|_2^2$$
$$= \|\boldsymbol{y}-\boldsymbol{X}\hat{\boldsymbol{\beta}}\|_2^2 + \|\boldsymbol{X}\hat{\boldsymbol{\beta}}-\boldsymbol{X}\boldsymbol{\beta}\|_2^2 + 2(\boldsymbol{y}-\boldsymbol{X}\hat{\boldsymbol{\beta}})^T(\boldsymbol{X}\hat{\boldsymbol{\beta}}-\boldsymbol{X}\boldsymbol{\beta})$$

が成立するから， $(\boldsymbol{y}-\boldsymbol{X}\hat{\boldsymbol{\beta}})^T(\boldsymbol{X}\hat{\boldsymbol{\beta}}-\boldsymbol{X}\boldsymbol{\beta})=0$ **を示せばOK**

$\hat{\boldsymbol{\beta}}=(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}$ **を代入する**

$$\boldsymbol{y}-\boldsymbol{X}\hat{\boldsymbol{\beta}} = \boldsymbol{y}-\boldsymbol{X}(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y} = (\boldsymbol{I}_n-\boldsymbol{P}_X)\boldsymbol{y}$$

$$\boldsymbol{X}\hat{\boldsymbol{\beta}}-\boldsymbol{X}\boldsymbol{\beta} = \boldsymbol{P}_X\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta} = \boldsymbol{P}_X\boldsymbol{y}-\boldsymbol{P}_X\boldsymbol{X}\boldsymbol{\beta} \quad \because\ \boldsymbol{P}_X\boldsymbol{X}=\boldsymbol{X}$$

したがって...

$$(\boldsymbol{y}-\boldsymbol{X}\hat{\boldsymbol{\beta}})^T(\boldsymbol{X}\hat{\boldsymbol{\beta}}-\boldsymbol{X}\boldsymbol{\beta}) = \boldsymbol{y}^T\underline{(\boldsymbol{I}_n-\boldsymbol{P}_X)^T\boldsymbol{P}_X}(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta})$$

であるが,

$$\underline{(\boldsymbol{I}_n-\boldsymbol{P}_X)^T\boldsymbol{P}_X} = (\boldsymbol{I}_n-\boldsymbol{P}_X)\boldsymbol{P}_X = \boldsymbol{P}_X-\boldsymbol{P}_X\boldsymbol{P}_X = \boldsymbol{P}_X-\boldsymbol{P}_X = \underline{\boldsymbol{O}}$$

推定された回帰係数の分布

- **記法**

$$\hat{\boldsymbol{\beta}} = (\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_{p+1}),\ \boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_{p+1})$$

$$v_j : (j, j) \text{ element of } (\boldsymbol{X}^T \boldsymbol{X})^{-1}$$

- **回帰係数の分布**

添字のズレに注意

$$\hat{\beta}_j \sim N(\beta_j, \sigma^2 v_{j+1}),\quad j = 0, 1, 2, \ldots, p$$

- **分散の推定とその分布**

$$\hat{\sigma}^2 = \frac{1}{n-p-1} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2, \quad \hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_{i1} + \cdots + \hat{\beta}_p x_{ip}$$

$$(n-p-1)\hat{\sigma}^2/\sigma^2 \sim \chi^2_{n-p-1}$$

- **t分布**
  - $Z \sim N(0,1), W \sim \chi^2_m$, Z and W are independent

$$\frac{Z}{\sqrt{W/m}} \sim t_m$$

- **回帰係数の分布**

実は係数と分散の推定量は独立

$$\frac{\beta_j - \beta_j}{\sqrt{\sigma^2 v_j}} \sim N(0,1) \qquad (n-p-1)\hat{\sigma}^2/\sigma^2 \sim \chi^2_{n-p-1}$$

$$\Rightarrow \quad \frac{\hat{\beta}_j - \beta_j}{\sqrt{\sigma^2 v_j}} \Big/ \sqrt{\frac{\hat{\sigma}^2}{\sigma^2}} \sim t_{n-p-1}$$

左辺を整理すると $T_j := \dfrac{\hat{\beta}_j - \beta_j}{\sqrt{\hat{\sigma}^2 v_j}} \sim t_{n-p-1}$

- **帰無仮説と対立仮説**
  - $H_0 : \beta_j = 0$ vs. $H_1 : \beta_j \neq 0$
  - 帰無仮説H0の下では

$$T_j := \frac{\hat{\beta}_j}{\sqrt{\hat{\sigma}^2 v_j}} \sim t_{n-p-1}$$

  - 備考：自由度の大きなt分布はほぼ標準正規分布

$\sqrt{\hat{\sigma}^2 v_j}$

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.938889   0.311908   9.422   <2e-16 ***
TV           0.045765   0.001395  32.809   <2e-16 ***
radio        0.188530   0.008611  21.893   <2e-16 ***
newspaper   -0.001037   0.005871  -0.177     0.86
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

$$T_j := \frac{\hat{\beta}_j}{\sqrt{\hat{\sigma}^2 v_j}}$$

Rによる実装

- **Exercise**
  - 赤で囲まれた部分をlmを使わずに計算せよ

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.938889   0.311908   9.422   <2e-16 ***
TV           0.045765   0.001395  32.809   <2e-16 ***
radio        0.188530   0.008611  21.893   <2e-16 ***
newspaper   -0.001037   0.005871  -0.177     0.86
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

F検定

```
Residual standard error: 1.686 on 196 degrees of freedom
Multiple R-squared:  0.8972,    Adjusted R-squared:  0.8956
F-statistic: 570.3 on 3 and 196 DF,  p-value: < 2.2e-16
```

- **次の帰無仮説を検定**

β_0が入っていないことに注意

$$H_0 : \beta_1 = \beta_2 = \cdots = \beta_p = 0$$

$$H_1 : \text{at least one } \beta_j \text{ is non-zero}$$

TV, radio, newspaperの**どれか**

は売上に効果がある

どれが効いているかは不明

F統計量

- **F分布**
  - $X_1 \sim \chi^2_{d_1}$, $X_2 \sim \chi^2_{d_2}$, X_1 and X_2 are independent

$$\frac{X_1/d_1}{X_2/d_2} \sim F_{(d_1,d_2)}$$

- **F統計量の計算**

RSS$_0$とRSS$_F$は独立

- H$_0$の下での残差平方和　$\mathrm{RSS}_0 = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2$
- フルモデルでの残差平方和　$\mathrm{RSS}_\mathrm{F} = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2$

$$\frac{\mathrm{RSS}_0 - \mathrm{RSS}_\mathrm{F}}{\sigma^2} \sim \chi^2_p \ \text{ under } H_0 \qquad \frac{\mathrm{RSS}_\mathrm{F}}{\sigma^2} \sim \chi^2_{n-p-1}$$

$$\Rightarrow \quad F = \frac{(\mathrm{RSS}_0 - \mathrm{RSS}_\mathrm{F})/p}{\mathrm{RSS}_\mathrm{F}/(n-p-1)} \sim F_{(p,n-p-1)} \ \text{ under } H_0$$

- **Exercise**
  - 下記の赤部分をRで実際に計算せよ

```
Residual standard error: 1.686 on 196 degrees of freedom
Multiple R-squared:  0.8972,    Adjusted R-squared:  0.8956
F-statistic: 570.3 on 3 and 196 DF,  p-value: < 2.2e-16
```

```
> result0 <- lm(sales ~ NULL, data=adv)
> predict(result0)
```

H_0での回帰分析

H_0での予測値 $\hat{y}_i$ の計算

F検定 (続き)

- **部分的な検定も可能**

$$H_0 : \beta_{p-q+1} = \beta_{p-q+2} = \cdots = \beta_p = 0$$

  - 検定の手順はほぼ同じ
    - H_0でのRSSとフルモデルでのRSSから計算される

$$\frac{\mathrm{RSS}_0 - \mathrm{RSS}_\mathrm{F}}{\sigma^2} \sim \chi^2_q \ \text{ under } H_0 \qquad \frac{\mathrm{RSS}_\mathrm{F}}{\sigma^2} \sim \chi^2_{n-p-1}$$

$$F = \frac{(\mathrm{RSS}_0 - \mathrm{RSS}_\mathrm{F})/q}{\mathrm{RSS}_\mathrm{F}/(n-p-1)} \sim F_{(q,n-p-1)} \ \text{ under } H_0$$

Advanced

一箇所だけの場合はＦの平方根がＴの絶対値に一致する

F検定の実際

- **白ワインデータを読み込む**
  - ワインの品質に関するデータ

```
> wine <- read.csv("winequality-white.csv",sep=";")
> head(wine)
```

```
> names(wine)
 [1] "fixed.acidity"        "volatile.acidity"     "citric.acid"
 [4] "residual.sugar"       "chlorides"            "free.sulfur.dioxide"
 [7] "total.sulfur.dioxide" "density"              "pH"
[10] "sulphates"            "alcohol"              "quality"
```

結果変数

```
> lmwine <- lm(quality~., data=wine)
> summary(lmwine)

Coefficients:
                       Estimate Std. Error t value Pr(>|t|)
(Intercept)          1.502e+02  1.880e+01   7.987 1.71e-15 ***
fixed.acidity        6.552e-02  2.087e-02   3.139  0.00171 **
volatile.acidity    -1.863e+00  1.138e-01 -16.373  < 2e-16 ***
citric.acid          2.209e-02  9.577e-02   0.231  0.81759
residual.sugar       8.148e-02  7.527e-03  10.825  < 2e-16 ***
chlorides           -2.473e-01  5.465e-01  -0.452  0.65097
free.sulfur.dioxide  3.733e-03  8.441e-04   4.422 9.99e-06 ***
total.sulfur.dioxide -2.857e-04  3.781e-04  -0.756  0.44979
density             -1.503e+02  1.907e+01  -7.879 4.04e-15 ***
pH                   6.863e-01  1.054e-01   6.513 8.10e-11 ***
sulphates            6.315e-01  1.004e-01   6.291 3.44e-10 ***
alcohol              1.935e-01  2.422e-02   7.988 1.70e-15 ***
```

$$H_0 : \beta_{\texttt{citric.acid}} = \beta_{\texttt{chlorides}} = \beta_{\texttt{total.sulfur.dioxide}} = 0$$

モデルに組み込むべきかどうかの検定

なぜ **”同時”** 検定が必要か？

- **各々の検定を繰り返した場合**

$$H_{0,1} : \beta_{\texttt{citric.acid}} = 0 \qquad H_{0,2} : \beta_{\texttt{chlorides}} = 0$$

$$H_{0,3} : \beta_{\texttt{total.sulfur.dioxide}} = 0$$

  - それぞれ独立に有意水準5%の検定を実施したとすると

$$\begin{aligned}
&\mathbb{P}(\texttt{at least one } H_{0,j} \texttt{ is uncorrectly rejected}) \\
&\quad = 1 - \prod_{j=1}^{3} \mathbb{P}(H_{0,j} \texttt{ is correctly accepted}) \\
&\quad = 1 - (0.95)^3 \simeq 0.143
\end{aligned}$$

検定の個数が増加するとこの確率も増加する

- **Exercise**
  - 実際に白ワインデータで次の検定を行ってみよう

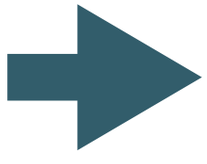
$$H_0 : \beta_{\texttt{citric.acid}} = \beta_{\texttt{chlorides}} = \beta_{\texttt{total.sulfur.dioxide}} = 0$$

性能評価

Motivation

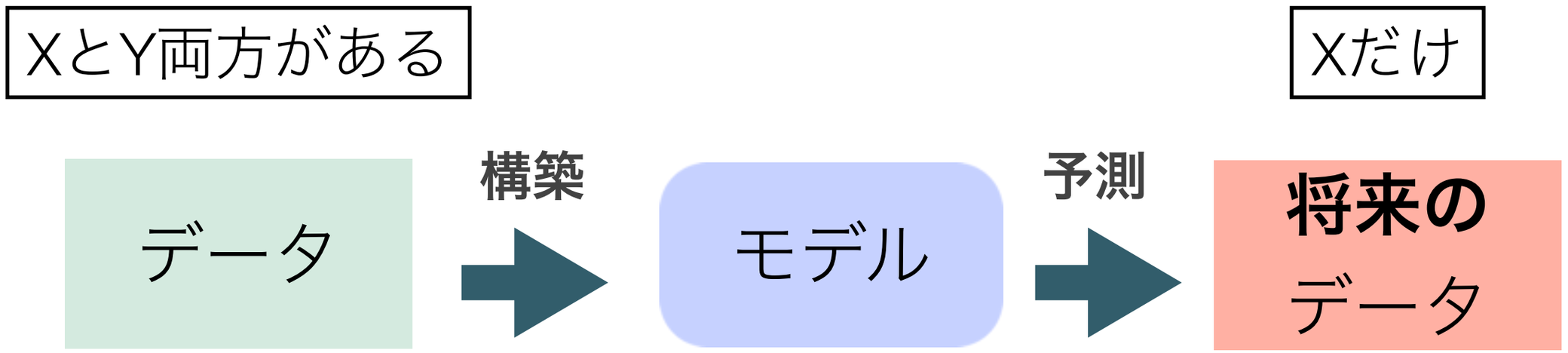
- **予測に重点が置かれる場合**
  - 社運をかけた商品の売上
    - どの程度広告費を計上すれば希望の売上を達成できるか
  - 不動産業者
    - 住宅価格の設定が利益に直結

手元のデータへの適合だけで本当に良いの？



過剰適合が起こり得る

モデル作成プロセス



- **得られたモデルがデータに極端に当てはまる場合**
  - そのデータを説明するだけのモデルになる
  - 将来のデータへは適合していない可能性
  - **過剰適合 (over-fitting)** の問題

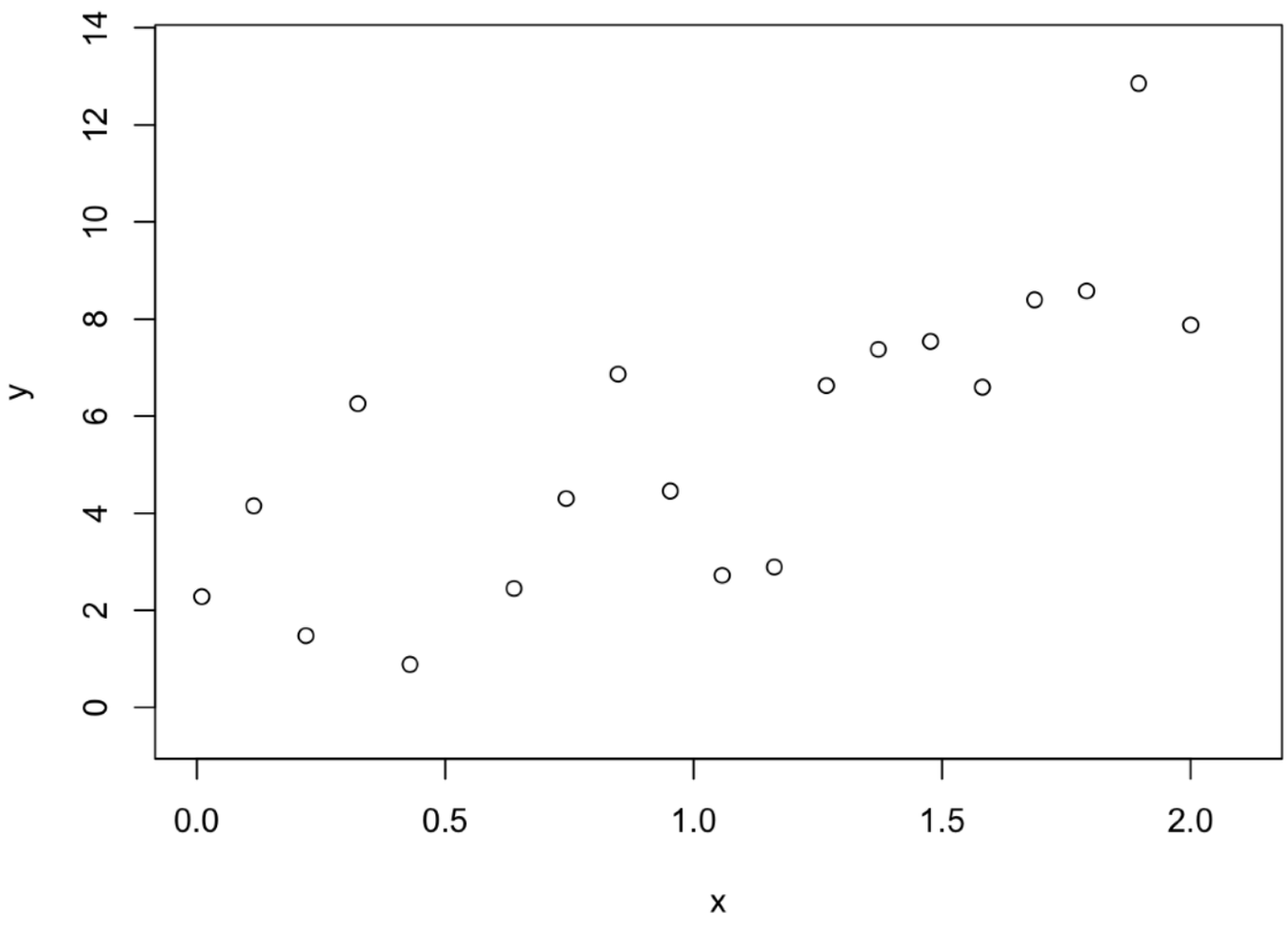
Multiple R-squared

$$1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \in [0, 1]$$

実際に確認してみる

- **仮想データを読み込んで散布図を描く**

```
> vert.data <- read.csv("vert_data.csv")[,-1]
> x <- vert.data[,1]; y <- vert.data[,2]
> plot(x,y,xlim=c(0,2.1),ylim=c(-0.5,13.5))
```



線形モデルでは
説明できなさそう…

- **多項式回帰モデル**

$$Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \cdots + \beta_k X^k + \varepsilon$$

  - 実装は簡単 → 多変量線形回帰モデルの応用！

- **次数kと複雑さの関係**
  - 次数を増やせば曲線が複雑になる
  - データへの適合は次数を増やすほど良い

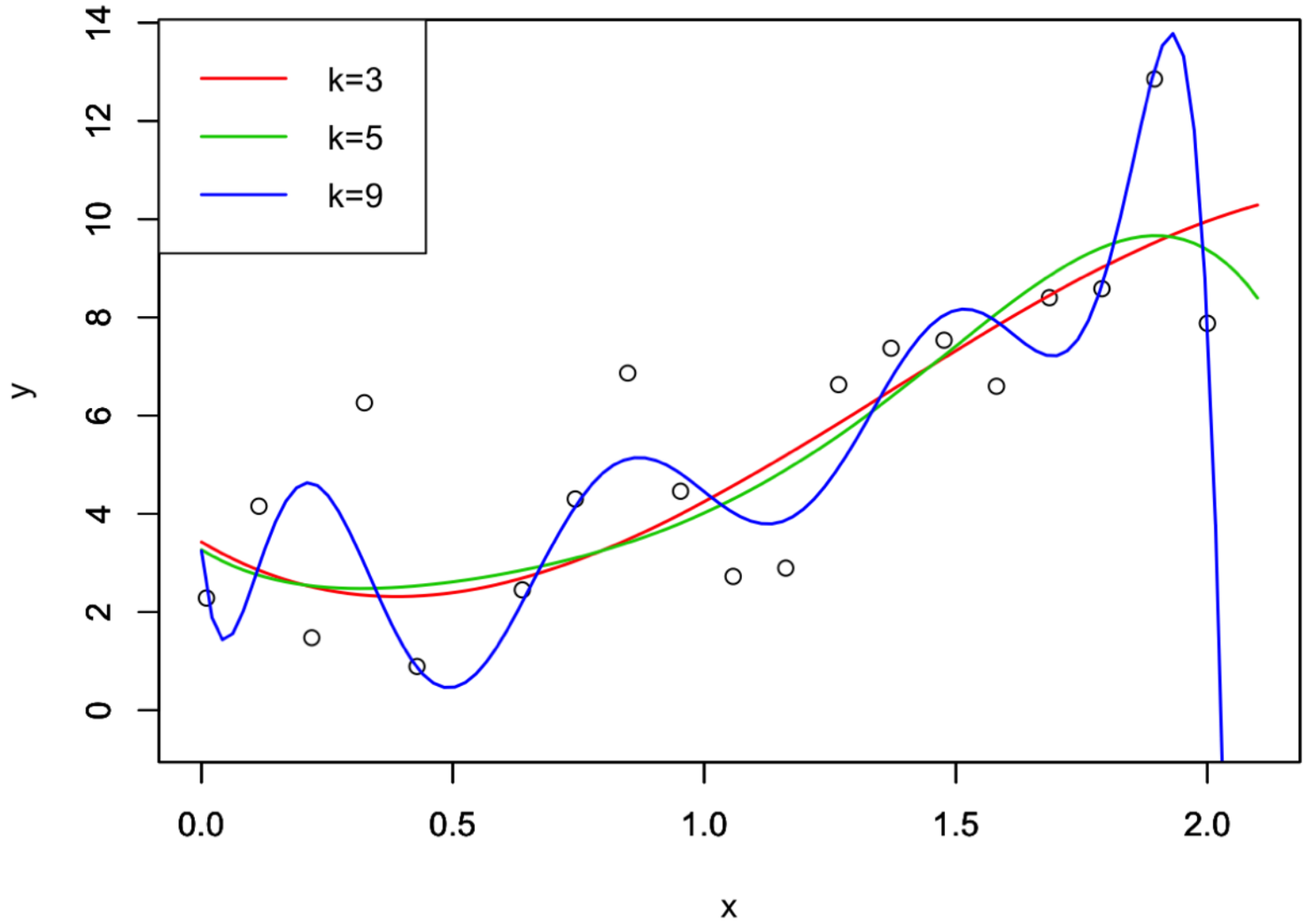
k=3 : Multiple R-squared: 0.6352

k=5 : Multiple R-squared: 0.6408

k=9 : Multiple R-squared: 0.8151

Q. 次数9がベスト？

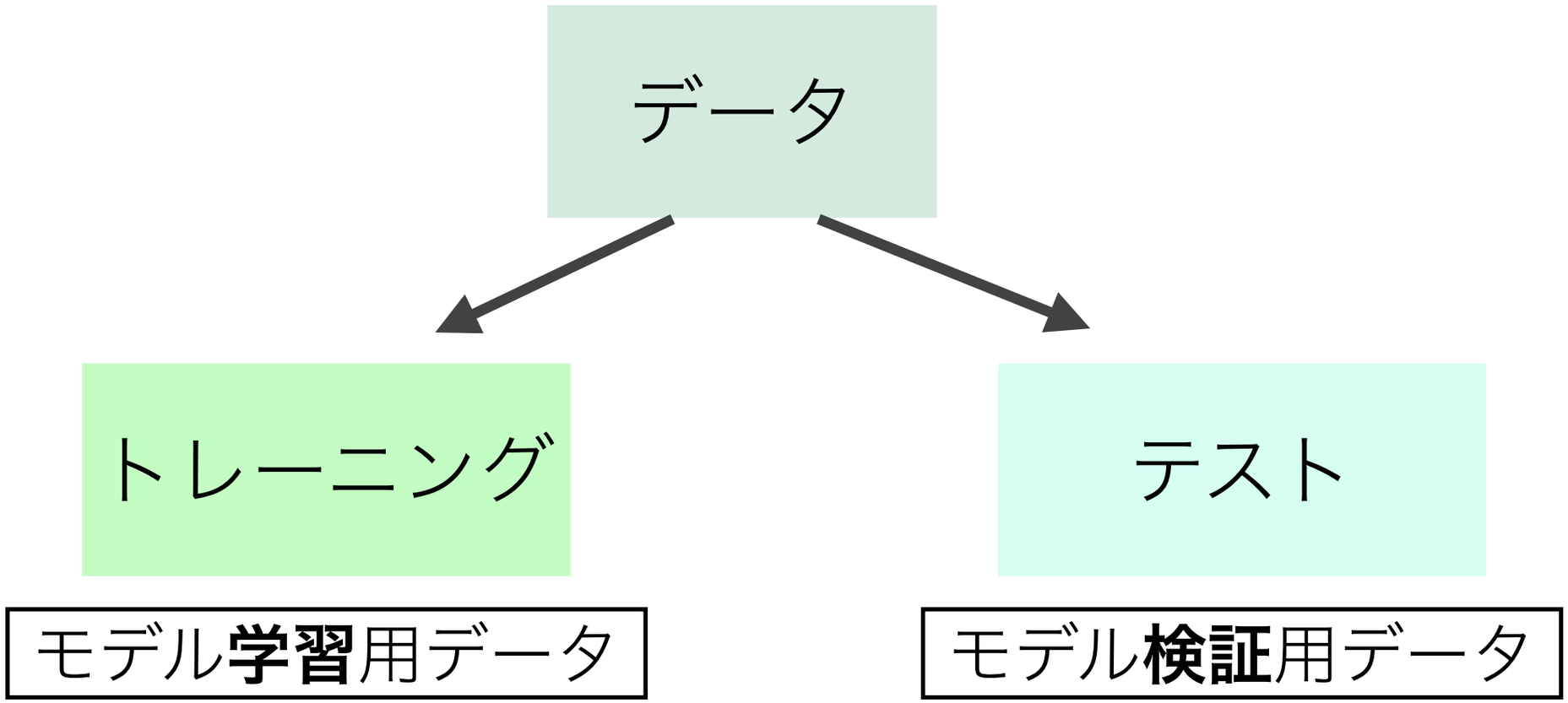
- **推定された曲線を描いてみる**



- **k=9の曲線を見てみる**
  - x > 2で急激にyが下がる; xが0.7付近で上昇傾向
  - 実は正解が**k=3**

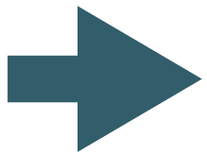
どう評価するか

- **データを2つに分割(互いに素)する**
  - トレーニング(訓練)データとテストデータ



トレーニング

学習

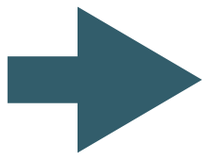


モデル

- 線形回帰モデル
- Nearest Neighbor
- ランダムフォレスト

テスト

性能評価



- 線形回帰モデル： 3.508
- Nearest Neighbor : 2.441
- ランダムフォレスト： **0.551**

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

$$\sum_{i=1}^{n} I(y_i = \hat{y}_i)$$

実際にやってみる

- **Bostonデータを読み込む**
  - Bostonの住宅価格に関するデータ

```
> library(MASS)
> head(Boston)
```

X Y

	crim	zn	indus	chas	nox	rm	age	dis	rad	tax	ptratio	black	lstat	medv
1	0.00632	18	2.31	0	0.538	6.575	65.2	4.0900	1	296	15.3	396.90	4.98	24.0
2	0.02731	0	7.07	0	0.469	6.421	78.9	4.9671	2	242	17.8	396.90	9.14	21.6
3	0.02729	0	7.07	0	0.469	7.185	61.1	4.9671	2	242	17.8	392.83	4.03	34.7
4	0.03237	0	2.18	0	0.458	6.998	45.8	6.0622	3	222	18.7	394.63	2.94	33.4
5	0.06905	0	2.18	0	0.458	7.147	54.2	6.0622	3	222	18.7	396.90	5.33	36.2
6	0.02985	0	2.18	0	0.458	6.430	58.7	6.0622	3	222	18.7	394.12	5.21	28.7

犯罪率, 非小売業の割合, 所得税率, etc **住宅価格**

- **トレーニングとテストデータに分ける**

```
> nt <- 100 ##number of test sample
> test.boston <- Boston[1:nt,] ##set first nt rows as test sample
> train.boston <- Boston[(nt+1):506,]
```

- 次の線形モデルを比較したい

$$\texttt{medv} = \beta_0 + \beta_1 \texttt{crim} + \varepsilon$$
$$\texttt{medv} = \beta_0 + \beta_1 \texttt{rm} + \beta_2 \texttt{age} + \varepsilon$$
$$\texttt{medv} = \beta_0 + \beta_1 \texttt{tax} + \beta_2 \texttt{lstat} + \varepsilon$$
$$\texttt{medv} = \beta_0 + \beta_1 \texttt{zn} + \beta_2 \texttt{indus} + \beta_3 \texttt{chas} + \varepsilon$$

- 最初のモデル

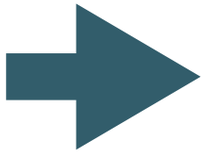
```
> lm1 <- lm(medv~crim,data=train.boston)
> cf1 <- coef(lm1)
> yhat1 <- cf1[1] + cf1[2] * test.boston[,"crim"]
> mse1 <- sum((test.boston[,"medv"] - yhat1)^2)/nt
> mse1
[1] 37.96788
```

- **Exercise**
  - 同様にしてMSEを導出し, ベストモデルを探し出そう

クロスバリデーション

モデルの調整

- **モデルチューニング**
  - 多くのモデルはチューニングが必要なパラメータを持つ
    - 多項式回帰モデルにおける次数
    - 線形回帰モデルにおける説明変数の個数
    - 回帰係数の取る範囲
- **どうチューニングするか？**
  - トレーニングに適合するように決めてはもちろんダメ
  - テストデータは使えない (最後の検証用に使うべき)

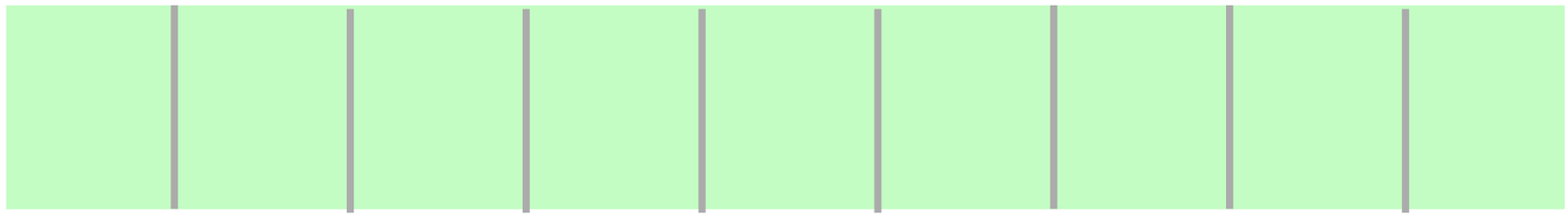


クロスバリデーション (交差検証法)

D-foldクロスバリデーション

トレーニングデータ

D個に分割



ひとつを検証用に取っておき，残りで学習

検証	**学習**

取っておいたデータを使って性能評価

性能指標　Ex. $\sum_{i \in test} (y_i - \hat{y}_i)^2 / n_{test}$

検証用データを変えながらD回これを繰り返し，性能指標の平均を取る

D-foldクロスバリデーションの実際

- **Carデータを読み込む**

```
> Auto <- read.csv("Auto.csv",header=T)[,-1]
> head(Auto)
  mpg cylinders displacement horsepower weight acceleration year origin
1  18         8          307        130   3504         12.0   70      1
2  15         8          350        165   3693         11.5   70      1
3  18         8          318        150   3436         11.0   70      1
4  16         8          304        150   3433         12.0   70      1
5  17         8          302        140   3449         10.5   70      1
6  15         8          429        198   4341         10.0   70      1
```

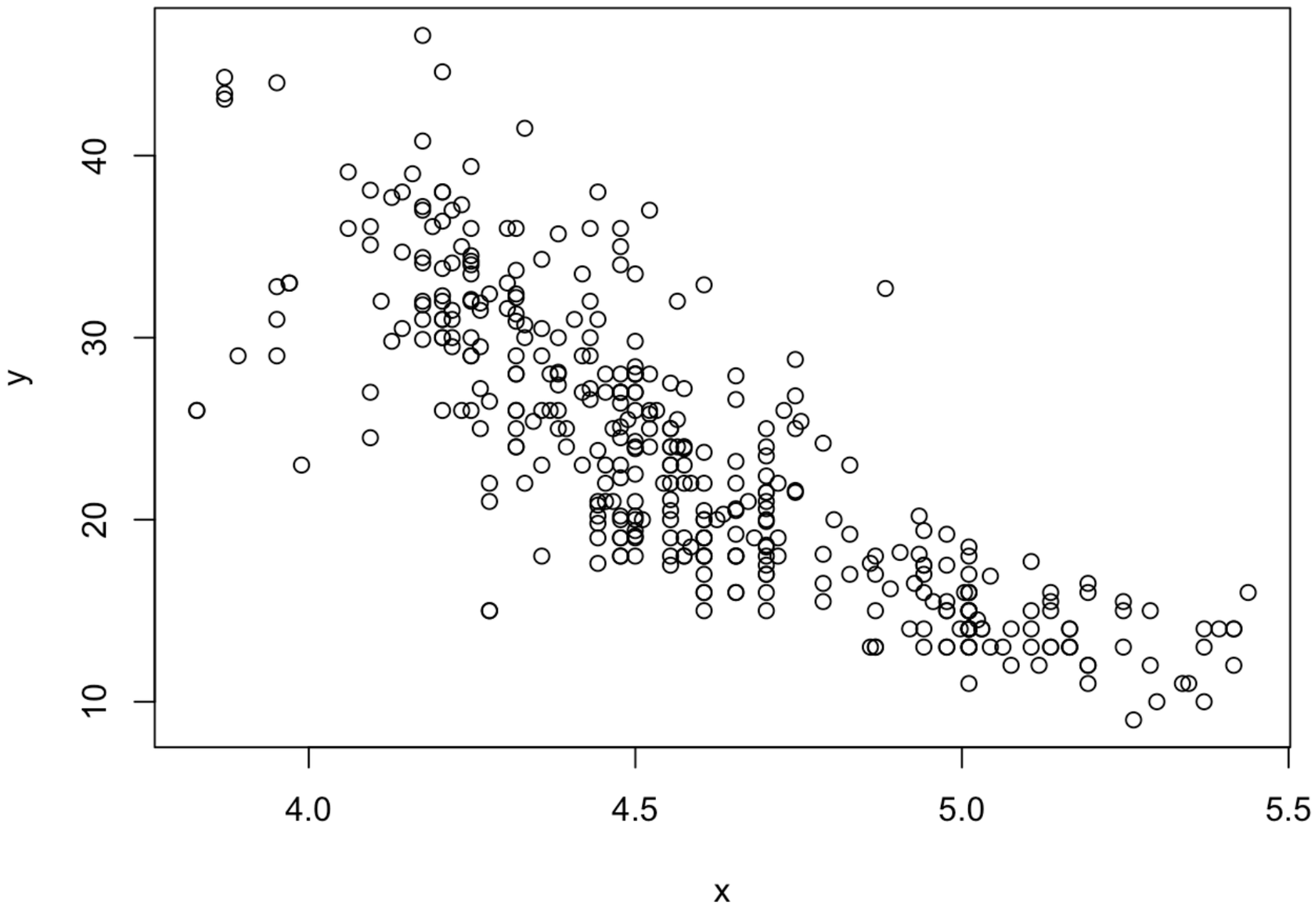
Y 走行距離　　X 馬力

K-foldクロスバリデーションの実際

- **散布図を描く**

```
> x <- log(Auto[,"horsepower"])
> y <- Auto[,"mpg"]
> plot(x,y)
```

スケールを小さく



多項式回帰が良さそう

次数kを選びたい

- **次数kの選択手順 (k=1,2,3,4,5の中から選ぶ)**
  - kを固定してD-foldクロスバリデーションを実行
    - データをD個に分割
    - 最初のデータを検証用に確保し，残りでβ_0 ,..., β_kを推定
    - 検証データ上で予測値を計算：

$$\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i + \hat{\beta}_2 x_i^2 + \cdots + \hat{\beta}_k x_i^k,\ i \in test$$

    - 残差平方和を計算：$\mathrm{Error}_{k,1} = \sum_{i \in test} (y_i - \hat{y}_i)^2 / n_{test}$
    - 検証データを動かしてD個の$\mathrm{Error}_{k,1}, \ldots, \mathrm{Error}_{k,D}$ を計算
    - その平均を計算する：$\mathrm{Error}_k = \sum_{i=1}^{D} \mathrm{Error}_{k,i} / D$
  - kを動かしてError1~Error5を計算する
  - 一番小さいErrorの値に対応するkを選択

・空白部分を埋めて試してみよう

トレーニングデータ

```
X <- as.data.frame(cbind(y,x,x^2,x^3,x^4,x^5))
n <- dim(X)[1]

##8-fold cross-validation since n=392
fold.n <- n/8
accuracy <- rep(0,5)
for(i in 1:5){
  error <- rep(0,8)
  data <- X[,1:(i+1)]
  for(D in 0:7){
    test.data <-
    train.data <-
    result <- lm(y~.,data=train.data)
    cf <- coef(result)
    yhat <-
    error[D+1] <- sum((test.data[,1] - yhat)^2)/fold.n
  }
  accuracy[i] <- mean(error)
}
which.min(accuracy)
```