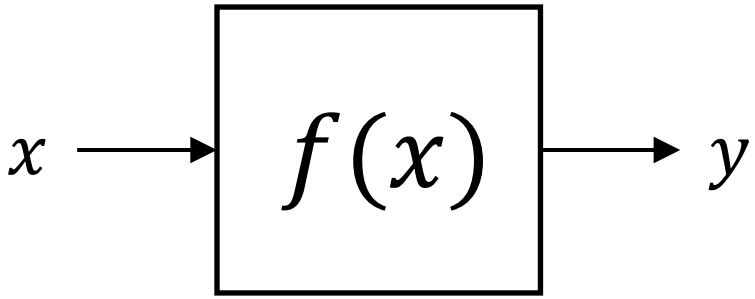
情報システム実験実習II（後期）
機械学習によるデータ解析
1回目

情報システムコース 内田雅人

最小二乗法による一次関数のパラメータ推定

関数のおさらい

- $y = f(x)$
  - ◆入力xから何かしらの処理をして出力yを求める
  - ◆この処理内容は$f(x)$で決まる



- 関数$y = f(x)$の例
  - ◆一次関数：$y = ax + b$
  - ◆二次関数：$y = x^2$
  - ◆三角関数：$y = \sin(x)$，$y = \cos(x)$，$y = \tan(x)$

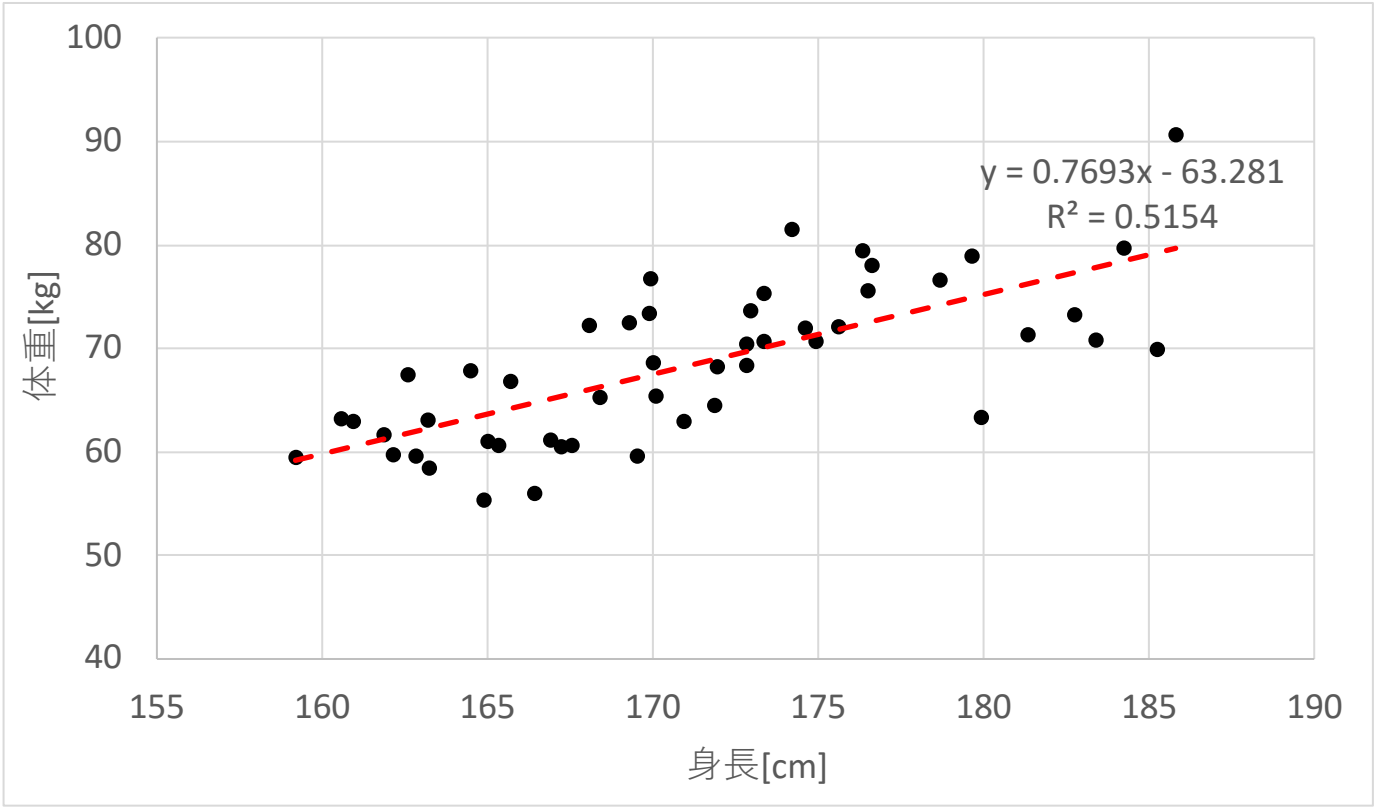
回帰分析

- 変数同士の関係を表す最適な線を数式化（モデル化）する分析手法
- 直線回帰（線形回帰）
  - ◆2つの数の関係を直線関係（一次関数）と仮定し近似
  - ◆近似された直線は回帰直線という
- 非線形回帰
  - ◆直線以外で近似したもの

- モデル関数
  - ◆変数間の関係を関数$y = f(x)$として表したもの

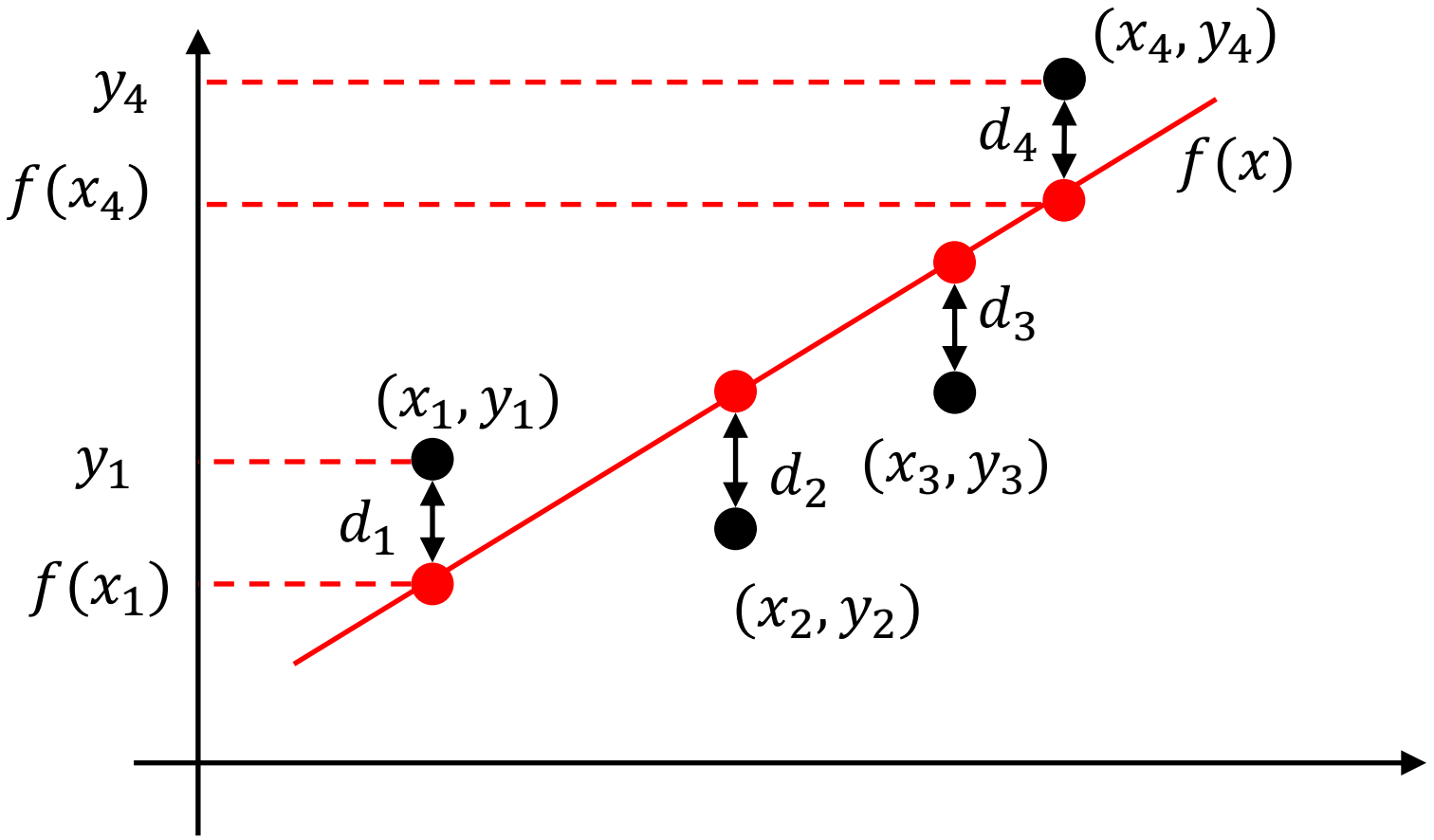
例：身長と体重

- 身長xが高いほど体重yが重い傾向
  - ◆一次関数で表現：$y = 0.7693x - 63.281$
    → この傾きと切片を今回は求める！
  - ◆相関係数r：$r = \sqrt{r^2} = \sqrt{0.5154} \fallingdotseq 0.7179$



最小二乗法

- 推定する関数$y = f(x)$と各データの残差$d_i = y_i - f(x_i)$（あるいは誤差$e_i = f(x_i) - y_i$）の二乗誤差が最小になる関数を求める方法（非線形でも適用可能）



最小二乗法による直線回帰(1)

- 目的関数：$y = f(x) = ax + b$
- ある点の真値(x_i, y_i)とその予測値$(x_i, ax_i + b)$の誤差e_iは

$$e_i = f(x_i) - y_i = (ax_i + b) - y_i$$

- nを要素数とし，二乗誤差の総和Eを考える

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

→誤差が最小となるa，bが最も当てはめの良い関数

※ この後，簡略化のため総和を次のように表現する： $\sum_{i=1}^{n} x_i = \sum x_i$

最小二乗法による直線回帰(2)

- 誤差Eの式を展開するとa^2，b^2は常に正を取るので下に凸の関数となる
  → a，bそれぞれ微分した値が0 = 誤差最小

$$\frac{\partial E}{\partial a} = 0, \quad \frac{\partial E}{\partial b} = 0$$

係数の2は両辺を2で割って消している

- それぞれ展開

$$\frac{\partial E}{\partial a} = \sum (ax_i^2 + bx_i - x_i y_i) = 0 \quad \cdots (1)$$

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

最小二乗法による直線回帰(3)

■ (2)式より

両辺をnで割った

$$\frac{1}{n}\sum(ax_i + b - y_i) = 0$$

$$a\frac{1}{n}\sum x_i + \frac{1}{n}nb - \frac{1}{n}\sum y_i = 0 \quad \cdots (3)$$

■ $\mu_x = \frac{1}{n}\sum x_i$，$\mu_y = \frac{1}{n}\sum y_i$とし，$b$について整理

※ 統計の分野では平均をμで表す

$$b = \mu_y - a\mu_x \quad \cdots (4)$$

aの求め方は次ページ以降

最小二乗法による直線回帰(4)

- (4)式を(1)式に代入

$$a\frac{1}{n}\sum x_i^2 + (\mu_y - a\mu_x)\frac{1}{n}\sum x_i - \frac{1}{n}\sum x_i y_i = 0$$

$$a\left(\frac{1}{n}\sum x_i^2 - \mu_x^2\right) + \mu_x\mu_y - \frac{1}{n}\sum x_i y_i = 0$$

$$a = \frac{\frac{1}{n}\sum x_i y_i - \mu_x\mu_y}{\frac{1}{n}\sum x_i^2 - \mu_x^2} \quad \cdots (5)$$

aについて，この後もう少し整理していく

最小二乗法による直線回帰(5)

- xとyの共分散σ_{xy}は「xの偏差×yの偏差」の平均
  ※ 偏差は平均との差のこと

$$\sigma_{xy} = \frac{1}{n}\left(\sum (x_i - \mu_x) \sum (y_i - \mu_y)\right)$$

- 期待値の記号で表すと

$$\sigma_{xy} = E[(x - \mu_x)(y - \mu_y)]$$

※ 期待値は$E[x] = \frac{1}{n}\sum x_i = \mu_x$のように表せる

最小二乗法による直線回帰(6)

■ 展開すると

$$\begin{aligned}\sigma_{xy} &= E\left[xy - x\mu_y - y\mu_x + \mu_x\mu_y\right] \\ &= E[xy] - E\left[x\mu_y\right] - E[y\mu_x] + E\left[\mu_x\mu_y\right] \\ &= E[xy] - \mu_y E[x] - \mu_x E[y] + \mu_x\mu_y \\ &= E[xy] - \mu_x\mu_y - \mu_x\mu_y + \mu_x\mu_y \\ &= E[xy] - \mu_x\mu_y \\ &= \frac{1}{n}\sum(x_i y_i) - \mu_x\mu_y \cdots (6)\end{aligned}$$

$\boldsymbol{x} = [x_1, x_2, \cdots, x_n]^T$, $\boldsymbol{y} = [y_1, y_2, \cdots, y_n]^T$

$\sum(x_i y_i) = \boldsymbol{x}^T\boldsymbol{y}$ → $\boldsymbol{x}$と$\boldsymbol{y}$の内積で表せる

$$\frac{1}{n}\sum(x_i y_i) = \frac{\boldsymbol{x}^T\boldsymbol{y}}{n}$$

最小二乗法による直線回帰(7)

- 次に，xの分散σ_x^2は次のようになる

$$\sigma_x^2 = \frac{1}{n}\left(\sum (x_i - \mu_x)\right)^2 = E[(x - \mu_x)^2]$$

- 展開すると

$$\begin{aligned}\sigma_x^2 &= E[x^2 - 2x\mu_x + \mu_x^2] \\ &= E[x^2] - \mu_x^2 \\ &= \frac{1}{n}\sum x_i^2 - \mu_x^2 \quad \cdots (7)\end{aligned}$$

$\boldsymbol{x} = [x_1, x_2, \cdots, x_n]^T$
$\sum x_i^2 = \boldsymbol{x}^T\boldsymbol{x}$

- (6)，(7)式を(5)式に代入すると

$$a = \frac{\sigma_{xy}}{\sigma_x^2} \quad \cdots (8)$$

※ (8)式を見ると統計的な処理だとわかる

最小二乗法による直線回帰（まとめ）

- $\boldsymbol{x} = [x_1, x_1, \cdots, x_n]^T$, $\boldsymbol{y} = [y_1, y_1, \cdots, y_n]^T$
- $\sigma_{xy} = \frac{1}{n}\sum(x_i y_i) - \mu_x \mu_y = \frac{\boldsymbol{x}^T \boldsymbol{y}}{n} - \mu_x \mu_y$
- $\sigma_x^2 = \frac{1}{n}\sum x_i^2 - \mu_x^2 = \frac{\boldsymbol{x}^T \boldsymbol{x}}{n} - \mu_x^2$

$$a = \frac{\sigma_{xy}}{\sigma_x^2}$$

$$b = \mu_y - a\mu_x$$

演習(1)

- 最小二乗法により一次関数$f(x) = ax + b$のパラメータを求めるプログラムを完成させよ

- 演習で使って良いnumpyの関数
  - ◆内積：np.dot()
  - ◆累乗：np.pow()
  - ◆平均：np.mean()
  - ◆合計：np.sum()

- プログラムの実行（/scripts内で実行）

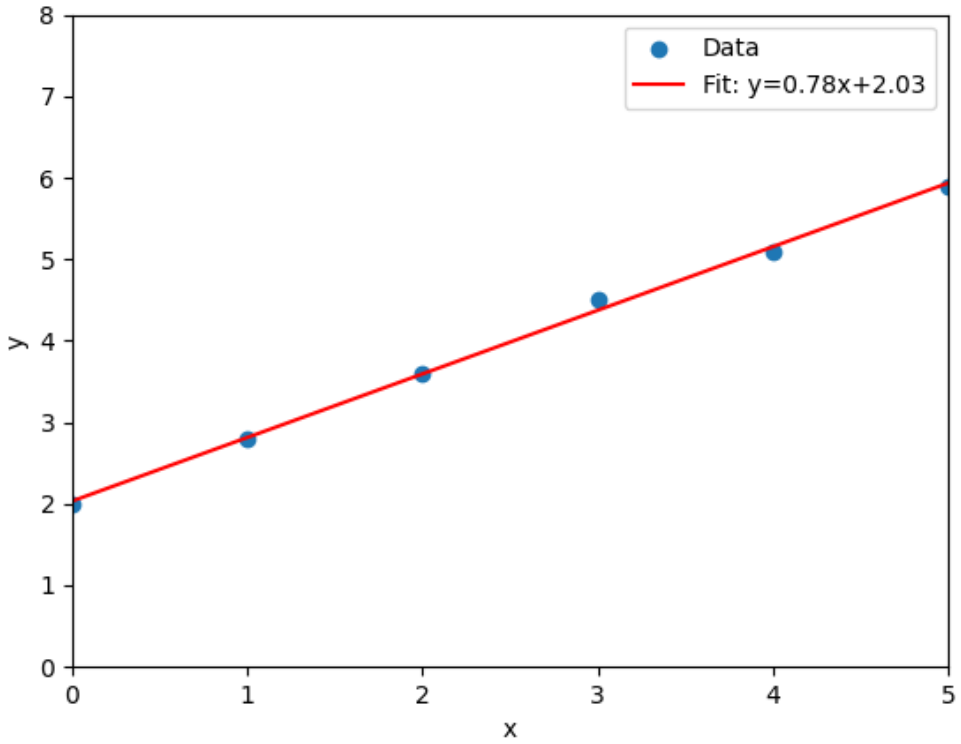
```
$ uv run least_squares_method.py
```

ディレクトリ構造

```
01_regression_analysis
├ outputs
|       └ least_squares_method.png（出力結果）
├ scripts
|       ├ least_squares_method.py（演習）
|       └ regression_analysis.py
├ pyproject.toml
└ requirements.txt
```

演習1の結果と比較

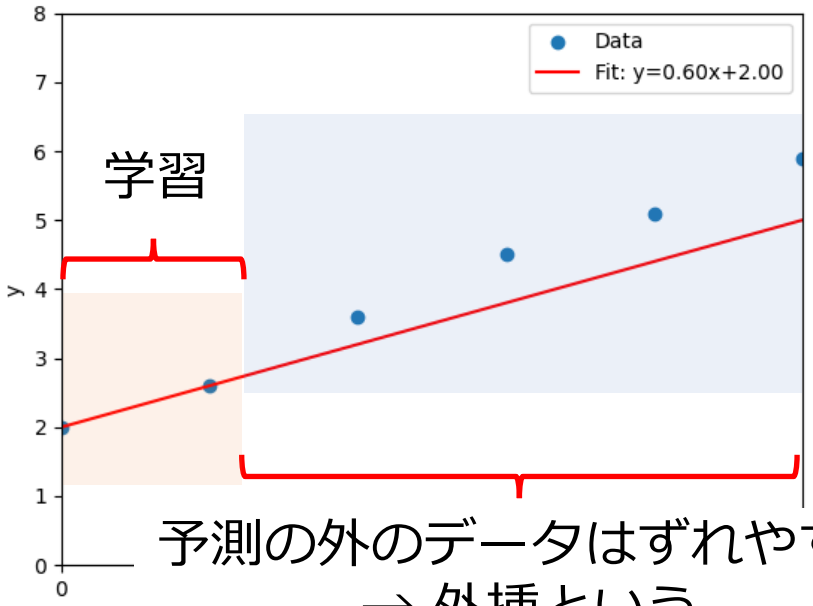
- 概ね，全てのデータにフィッテイング
- 大事なポイントは統計的な処理だけでパラメータを求められること



$a = 0.780000,\ b = 2.033333$

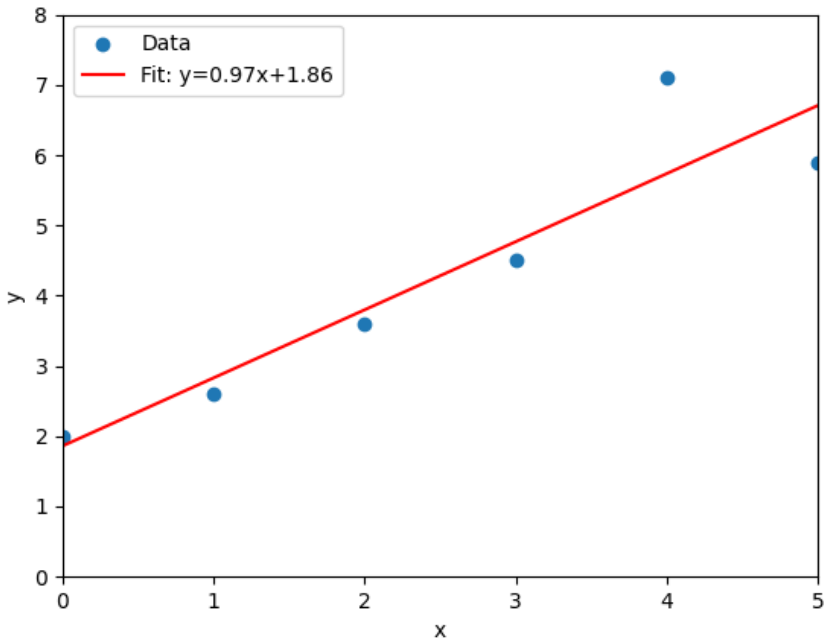
最小二乗法の注意点

1. データ点数が極端に少ないと，学習データより外のデータは予測を外しやすい
2. 外れ値があると引っ張られる



予測の外のデータはずれやすい
→ 外挿という

$a = 0.600000,\ b = 2.000000$
ただし，$y = [2, 2.6]$



$a = 0.968571,\ b = 1.861905$
外れ値を混入させた

正規方程式による回帰分析の書き換えと重回帰分析への拡張

直線回帰$y = wx + b$の書き直し(1)

- $b \to w_0$，$w \to w_1$，$x \to x_1$と置き換え，
  $w = [w_0, w_1]^T$，$x = [x_0, x_1]^T$，予測値$\hat{y}$とする
  $$\hat{y}(\boldsymbol{w}) = w_0 + w_1 x_1 = \boldsymbol{w}^T \boldsymbol{x}$$
- 実際はn個のデータセットがあることを
  踏まえて以下のように修正

$$\boldsymbol{x} = \begin{bmatrix} 1 & x_{1,1} \\ 1 & x_{2,1} \\ \vdots & \vdots \\ 1 & x_{n,1} \end{bmatrix}, \quad \boldsymbol{w} = \begin{bmatrix} w_0 \\ w_1 \end{bmatrix}, \quad \boldsymbol{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix},$$

$$\hat{\boldsymbol{y}} = \boldsymbol{x}\boldsymbol{w} = \begin{bmatrix} 1 & x_{1,1} \\ 1 & x_{2,1} \\ \vdots & \vdots \\ 1 & x_{n,1} \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \end{bmatrix} = \begin{bmatrix} w_0 + w_1 x_{1,1} \\ w_0 + w_1 x_{2,1} \\ \vdots \\ w_0 + w_1 x_{n,1} \end{bmatrix}$$

直線回帰のモデルの書き直し(2)

- 残差$\boldsymbol{\varepsilon} = \boldsymbol{y} - \hat{\boldsymbol{y}} = \boldsymbol{y} - \boldsymbol{xw}$
- 誤差関数

$$E(\boldsymbol{w}) = \sum \varepsilon_i^2 = \|\boldsymbol{\varepsilon}\|^2 = \|\boldsymbol{y} - \hat{\boldsymbol{y}}\|^2$$

- 目的は $\boldsymbol{w} = \arg\min(\|\boldsymbol{y} - \boldsymbol{xw}\|^2)$を解くこと

$$\begin{aligned} E(\boldsymbol{w}) &= \|\boldsymbol{y} - \hat{\boldsymbol{y}}\|^2 \\ &= (\boldsymbol{y} - \hat{\boldsymbol{y}})^T(\boldsymbol{y} - \hat{\boldsymbol{y}}) \\ &= (\boldsymbol{y} - \boldsymbol{xw})^T(\boldsymbol{y} - \boldsymbol{xw}) \\ &= \boldsymbol{y}^T\boldsymbol{y} - \boldsymbol{w}^T\boldsymbol{x}^T\boldsymbol{y} - \boldsymbol{y}^T\boldsymbol{xw} + \boldsymbol{w}^T\boldsymbol{x}^T\boldsymbol{xw} \end{aligned}$$

$$\frac{\partial E(\boldsymbol{w})}{\partial \boldsymbol{w}} = -2\boldsymbol{x}^T\boldsymbol{y} + 2\boldsymbol{x}^T\boldsymbol{xw} = 0$$

$$\boldsymbol{x}^T\boldsymbol{xw} = \boldsymbol{x}^T\boldsymbol{y}$$

$$\boldsymbol{w} = (\boldsymbol{x}^T\boldsymbol{x})^{-1}\boldsymbol{x}^T\boldsymbol{y}$$

数学記号の補足

- L2ノルム

$$\|\boldsymbol{x}\| = \sqrt{x_1^2 + x_1^2 + \cdots + x_n^2} = \sqrt{\sum x_i^2}$$

- 二乗和をL2ノルムと内積で表現

$$\sum x_i^2 = \|\boldsymbol{x}\|^2 = \boldsymbol{x}^T\boldsymbol{x} = [x_1, x_2, \cdots, x_n]\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

- $f(x)$ が最小値となるxの集合

$$x = \arg\min(f(x))$$

用語の補足

- 説明変数
  - ◆何かの原因となっている変数
  - ◆$y = ax + b$のうち，xが説明変数
- 目的変数
  - ◆その原因を受けた結果となっている変数
  - ◆$y = ax + b$のうち，yが目的変数
- 設計行列（計画行列，デザイン行列）
  - ◆回帰モデルに含まれる説明変数を行列の形で整理した行列
  - ◆1列目は切片項（1の列）
  - ◆様々な項を追加しやすいうえ，正規方程式の形は変わらない
    → この後の重回帰分析を考える上で重要な考え

重回帰分析

- 複数の説明変数があるとして拡張

$$y = w_0 + w_1 x_1 + w_2 x_2 + \cdots + w_p x_p = \boldsymbol{w}^T \boldsymbol{x}$$

$$\hat{\boldsymbol{y}}(\boldsymbol{w}) = \boldsymbol{x}\boldsymbol{w}$$

- n個のデータセットがあることを踏まえ，以下のように修正

$$\boldsymbol{x} = \begin{bmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1p} \\ 1 & x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & x_{n1} & x_{n2} & \cdots & x_{np} \end{bmatrix}, \quad \boldsymbol{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}, \quad \boldsymbol{w} = \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_p \end{bmatrix}$$

- 最終的に，単回帰分析と同じ結果になる

$$\boldsymbol{w} = (\boldsymbol{x}^T \boldsymbol{x})^{-1} \boldsymbol{x}^T \boldsymbol{y}$$

モデルの評価(1)

- R^2（決定係数，寄与率）
  - ◆説明変数だけで目的変数を
    何%説明できるかの指標，R^2の平方根が相関係数

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

  - ◆値域：$0 < R^2 < 1$
  - ◆1に近づくほどモデルの当てはまりが良い
  - ◆0に近づくほどモデルの当てはまりが悪い

モデルの評価(2)

- RMSE（Root Mean Squared Error）
  - ◆予測値が実測値から平均的に
    どれくらいズレているかの指標
  - ◆平均二乗誤差の平方根

$$RMSE = \sqrt{\frac{1}{n}\sum (y_i - \hat{y}_i)^2} = \sqrt{\frac{(\boldsymbol{y} - \hat{\boldsymbol{y}})^T(\boldsymbol{y} - \hat{\boldsymbol{y}})}{n}}$$

  - ◆値域：$0 < RMSE$

演習(2)

- 正規方程式による重回帰分析を実装
- データセットはdata/advertising.csvを使う
  - ◆Advertising Dataset（広告費と売上）
  - ◆説明変数：TV広告費、ラジオ広告費、新聞広告費
  - ◆目的変数：Sales（売上）

- numpyの関数など
  - ◆逆行列：np.linalg.inv()
  - ◆内積：np.dot()
  - ◆行列の水平方向への結合：
    np.hstack((行列1, 行列2, ...))
  - ◆要素が1の行列を作成：np.ones((行数, 列数))

演習（続き）

- その他の記法
  - ◆各次元の要素数を取得：変数名.shape → x.shape
  - ◆指定した次元の要素数を取得：
    変数名.shape[次元番号]
    x.shape[0] → xの0次元目の要素数
    x.shape[1] → xの1次元目の要素数
  - ◆行列の転置：変数名.T → x.Tなど
  - ◆内積：変数1 @ 変数2 → x.T @ xなど
  - ◆二乗：変数 ** 2

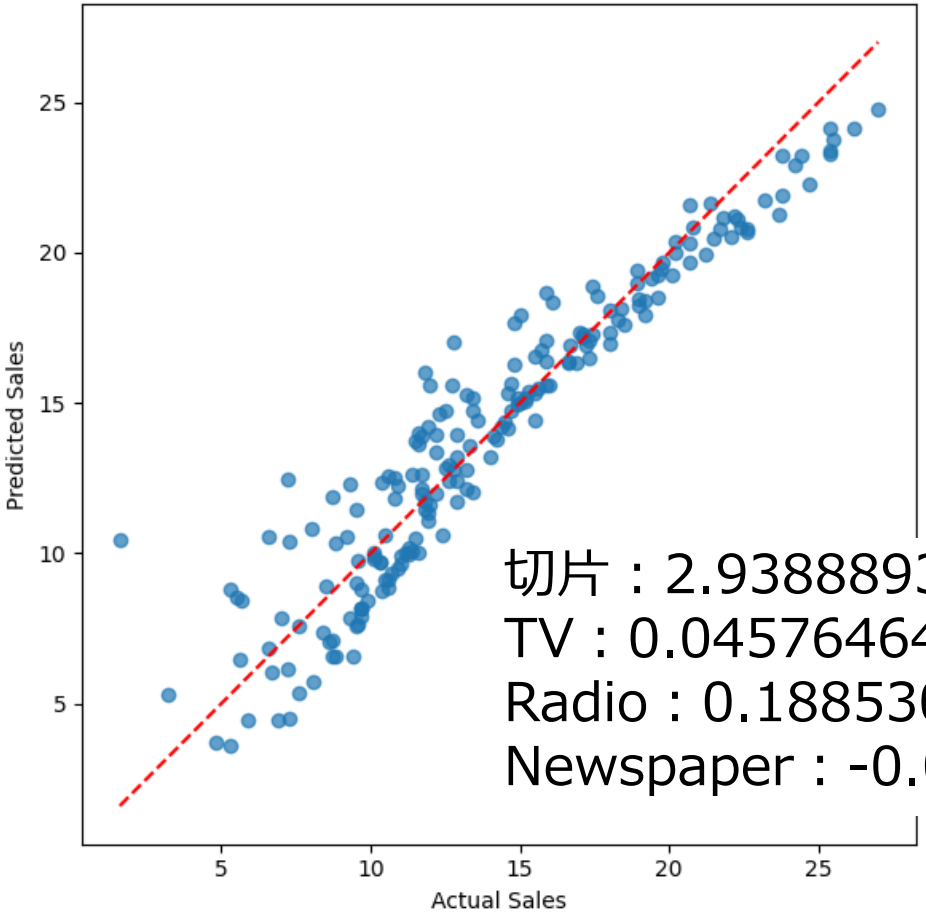
- プログラムの実行（/scripts内で実行）
  $ uv run regression_analysis.py

ディレクトリ構造

```
01_regression_analysis
 ├ data
 ┃      └ advertising.csv
 ├ outputs
 ┃      └ advertising_actual_vs_predicted.png（出力結果）
 ├ scripts
 ┃      ├ least_squares_method.py
 ┃      └ regression_analysis.py（演習）
 ├ pyproject.toml
 └ requi
rements.txt
```

出力結果

- 横軸に正解，縦軸に予測の散布図
- 直線上にデータがあるほど良いモデル



切片：2.9388893694594294
TV：0.045764645455397 55
Radio：0.18853001691820445
Newspaper：-0.0010374930424763527

課題・考察

課題・考察(1)

- regression_analysis.pyを以下の条件で改良する．改良後，各説明変数について，各項を追加・削除したときのR^2変化を比較し，理由を考察せよ．プログラムはex_01_学籍番号.pyとして作成せよ．
  - ◆R^2とRMSEの算出する関数を追加し，main関数で値を表示する
  - ◆TV * Radio，TV * Newspaper，Radio * Newspaperを設計行列に追加※
  - ◆設計行列から任意の列（説明変数）を選択できるようにする（切片は必ず残す）

※ 説明変数同士を乗じた項を交互作用項という

課題のヒント

- 交互作用項の計算
  配列名 = x[:, 列番号] * x[:, 列番号]

- 配列の追加（ reshape()をつけないとエラー）
  x = np.hstack([x, 配列名.reshape(-1, 1)])

- for文でのprintに対応するため，ラベル名を追加
  header.append(“適当な文字列")

- 列を選択する方法の例
  x = x[:, [0, 1, 3]]