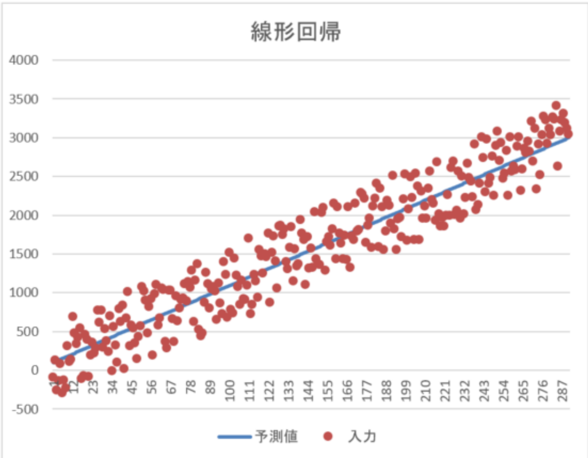
# 1 線形回帰

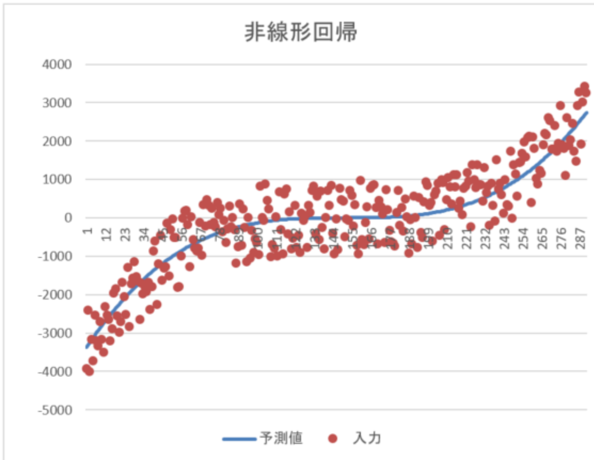
・回帰とは?:ある入力に対してある出力値を予測する問題

・直線 (多次元の場合は超平面) で予測することを線形回帰という

・一次式回帰は単回帰、多次元回帰を重回帰、ともいう

・(対して曲線 (曲面) で予測するのを非線形回帰という)




・入力値を説明変数または特徴量とよぶ (入力は m 次元のベクトル)

$$\text{説明変数 } \mathbf{x} = (x_1, x_1, \ldots, x_m)^T \in \mathbb{R}^m$$

・出力値 (予測値) を目的変数とよぶ (出力はスカラー値)

$$\text{目的変数 } y \in \mathbb{R}^1$$

## 2　線形結合 (入力とパラメータの内積)

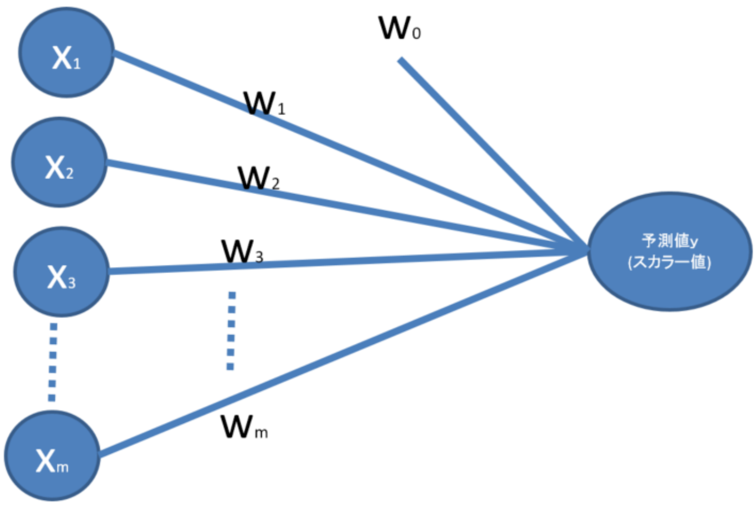
・m 次元の入力と m 次元のパラメータ (重み、傾き、とも言う) を線形結合する

$$\text{パラメータ } \mathbf{w} = (w_1, w_1, \ldots, w_m)^T \in \mathbb{R}^m$$

$$\text{線形結合} \hat{y} = \mathbf{w}^T\mathbf{x} + w0 = \sum_{j=1}^{m} w_j x_j + w_0$$

(予測値 (ここでは y) には慣例としてハットをつける)
($w_0$ は切片 (バイアスとも言う))

線形結合を図示すると以下のようになると説明



## 3　線形回帰モデル (単回帰を例とする)

・回帰問題を解くための機械学習モデルの１つ
・教師あり学習である (入力値とその出力を教師データとして、パラメータ w を学習する)
・データは回帰直線に誤差 ($\epsilon$) が加わり観察されていると仮定
・モデルを連立方程式で表すと以下になると説明されている

$$\begin{aligned} y_1 &= w_0 + w_1 x_1 + \varepsilon_1 \\ y_2 &= w_0 + w_2 x_2 + \varepsilon_2 \\ &\vdots \\ y_n &= w_0 + w_n x_n + \varepsilon_n \end{aligned}$$

・上記連立方程式を行列式で表すと以下

$$\mathbf{y} = \mathbf{X}\mathbf{w} + \epsilon$$

ただし、

$\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n)^T$　　→ ($\mathbf{x}_i$ はそれぞれベクトルを示す)
$\mathbf{y} = (y_1, y_2, \ldots, y_n)^T$
$\epsilon = (\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_n)^T$
$\mathbf{x}_i = (1, x_1)^T$　　→ (切片 $w_0$ に対して 1 倍、$w_1$ に $x_1$ をそれぞれ乗算)
$\mathbf{w} = (w_0, w_1)^T$　　→ (切片 $w_0$ に対して 1 倍、$w_1$ に $x_1$ をそれぞれ乗算)

## 4　線形回帰モデル (重回帰を例とする)

・データは回帰曲面 (超平面) に誤差 ($\epsilon$) が加わり観察されていると仮定

$$y_1 = w_0 + w_1 x_1 + w_2 x_2 + \ldots + \epsilon$$

・上記連立方程式を行列式で表すと以下 (説明の為２次元に限定する)

$$\mathbf{y} = \mathbf{X}\mathbf{w} + \epsilon$$

ただし、

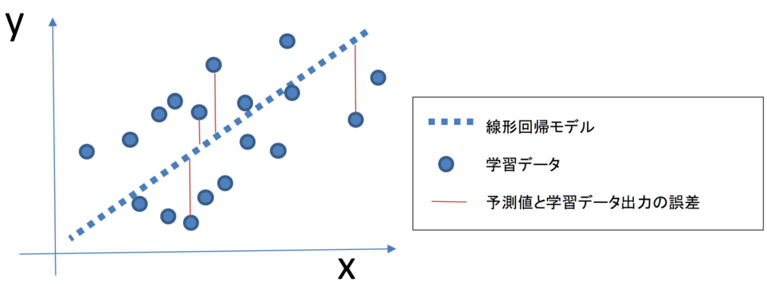
$\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n)^T$　　→ ($\mathbf{x}_i$ はそれぞれベクトルを示す)
$\mathbf{y} = (y_1, y_2, \ldots, y_n)^T$
$\epsilon = (\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_n)^T$
$\mathbf{x}_i = (1, x_1, x_2)^T$　　→ (切片 $w_0$ に対して 1 倍、$w_1$ に $x_1$、$w_2$ に $x_2$ を乗算)
$\mathbf{w} = (w_0, w_1, w_2)^T$　　→ (切片 $w_0$ に対して 1 倍、$w_1$ に $x_1$、$w_2$ に $x_2$ を乗算)

# 5　線形回帰の学習

・学習の過程で最小二乗法を用いてパラメータ推定する
・誤差はデータ (出力) とモデル出力の二乗誤差の和で算出
・二乗する事で符号をキャンセルする (誤差同士の打ち消しあいを排除)
・絶対値でなく二乗する意味→計算が容易なのでと説明されている
・平均二乗誤差 (MSE) を最小にするように直線 (モデル) を学習する

$$MSE_{train} = \frac{1}{n_{train}} \sum_{i=1}^{n_{train}} (\hat{y}_i^{train} - y_i^{train})^2$$

($\hat{y}$ が予測値で $y$ が学習データの出力)




MSE を最小とする様なパラメータ $\mathbf{w}$ を求める

→ MSE をパラメータ $\mathbf{w}$ に関して偏微分したものが 0、となる様な $\mathbf{w}$ を求める

・・・MSE が $\mathbf{w}$ の二次式→傾き 0 の点が底→最小、という理解でよい？

MSE をパラメータ $\mathbf{w}$ で偏微分する

$$\begin{aligned}
&\frac{\partial}{\partial w} MSE_{train} \\
&= \frac{\partial}{\partial w}\{\sum_{i=1}^{n_{train}} (\hat{y}_i^{(train)} - y_i^{(train)})^2\} \\
&= \frac{1}{n_{train}} \frac{\partial}{\partial w}\{(\mathbf{X}^{(train)}\mathbf{w} - \mathbf{y}^{(train)})^T(\mathbf{X}^{(train)}\mathbf{w} - \mathbf{y}^{(train)})\} \quad \rightarrow \text{(行列表現)} \\
&= \frac{1}{n_{train}} \frac{\partial}{\partial w}\{\mathbf{w}^T\mathbf{X}^{(train)T}\mathbf{X}^{(train)}\mathbf{w} - 2\mathbf{w}^T\mathbf{X}^{(train)T}\mathbf{y}^{(train)} + \mathbf{y}^{(train)T}\mathbf{y}^{(train)}\} \quad \rightarrow \text{(展開)}
\end{aligned}$$

パラメータ $\mathbf{w}$ で偏微分→ 0 となるので以下等式が成り立つ ($\frac{2}{n_{train}}$ の逆数を両辺にかけて消している)

$$\rightarrow \mathbf{X}^{(train)T}\mathbf{X}^{(train)}\mathbf{w} - \mathbf{X}^{(train)T}\mathbf{y}^{(train)} = 0$$

上より MSE が最小になる様なパラメータ $\mathbf{w}$ は

$$\mathbf{w} = (\mathbf{X}^{(train)T}\mathbf{X}^{(train)})^{-1}\mathbf{X}^{(train)T}\mathbf{y}^{(train)}$$

となり、結局学習データの入力値と出力値のみで計算できることになる

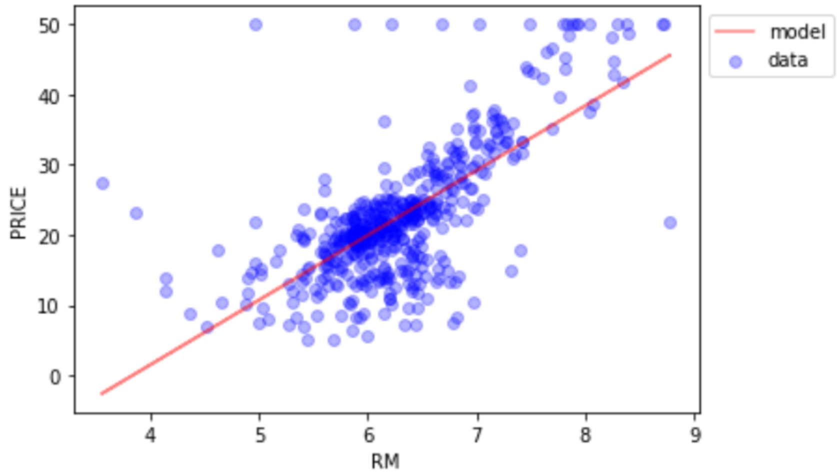
ちなみに・・・学習実施に際しては全データを使うのでなく、train データと test データに分けて、train データで学習、test データで評価、するように念押しされている (test は汎化性能評価の為)

## 6 実習

単回帰について考察

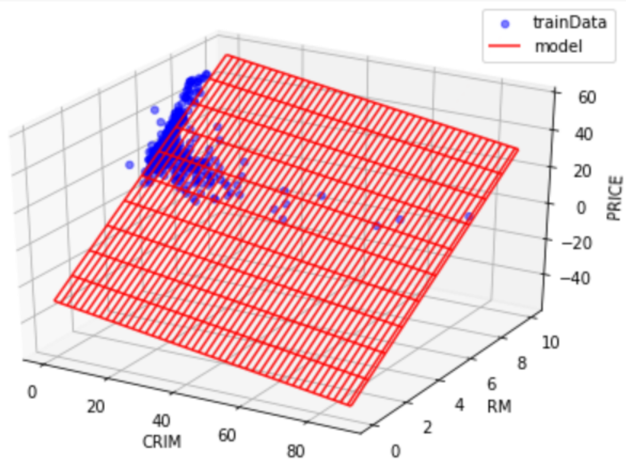
```
#モデルの回帰直線を現物と目視比較
#密度分布の偏りがモデルに見えているっぽい
#(二乗誤差最小でモデルが作られているのを何となく目視体感できる)
#上で「一部屋しかない物件の予想価格がマイナス」なのは
#目の前に見えているデータにfitさせた結果に過ぎず仕方なし
```

```
<matplotlib.legend.Legend at 0x7fa94f520810>
```




重回帰について考察

```
#モデルの回帰平面を現物と目視比較(3Dプロットで)
#まぁ・・・・可視性が良いとは言えないけど・・・・
#分布が密集しているゾーンでモデルの上下にそれなりに分布している様子は見えるけど程度
#犯罪率低めだと価格が高めで部屋数に正比例して上がる
#犯罪率高めだと価格が低め推移、そもそも犯罪率超高めだと物件もまばらになりがちに見える
```



課題に対する解答

```
#課題(部屋数4で犯罪率0.3の物件価格)を予測して終わりにしておく
print("部屋数4で犯罪率0.3の物件価格は%.2fになる"%(model2.predict([[0.3,4]])))
```

```
部屋数4で犯罪率0.3の物件価格は4.24になる
```

コードは本ドキュメントと同じレポジトリに提出する