Statistical Machine Learning, Part I

Regression

mcuturi@i.kyoto-u.ac.jp

Fundamentals in Regression

- Can be studied from different viewpoints: statistical, linear algebra, AI... *etc.*.
- Linear regression is currently revived by different ideas in **sparsity**
  - Lasso (1996$\rightarrow$)
  - SVM for regression (1996$\rightarrow$)
  - Compressed Sensing (2002$\rightarrow$)

One of the most standard data analysis tasks: Regression

Data: many observations of the same data type

- We have a database $\{\mathbf{x}_1, \cdots, \mathbf{x}_N\}$.
- Each datapoint $\mathbf{x}_j$ can be encoded as a vector of features $\mathbf{x}_j = \begin{bmatrix} x_{\mathbf{1},j} \\ x_{\mathbf{2},j} \\ \vdots \\ x_{\boldsymbol{d},j} \end{bmatrix}$
- Each feature $x_{i,j}$ $(1 \leq i \leq d)$ of a given point $\mathbf{x}_j$ is a number.

One of the most standard data analysis tasks: Regression

This database can be seen as a $\mathbb{R}^{d\times N}$ **matrix**

$$\{\mathbf{x}^1, \cdots, \mathbf{x}^N\} \Longleftrightarrow \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,N} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,N} \\ x_{3,1} & x_{3,2} & \cdots & x_{3,N} \\ \vdots & \vdots & \cdots & \vdots \\ x_{d,1} & x_{d,2} & \cdots & x_{d,N} \end{bmatrix}$$

Examples

$$\text{Credit card holder } \mathbf{x}_j = \begin{bmatrix} \text{Income} \\ \text{Age} \\ \vdots \\ \text{Work history (months)} \\ \text{Family} \\ \#\ \text{Credit Incidents} \end{bmatrix}$$

$$\text{Patient } \mathbf{x}_j = \begin{bmatrix} \text{height} \\ \text{weight} \\ \vdots \\ \#\ \text{minutes exercise/week} \\ \text{LDL cholesterol} \\ \text{HDL cholesterol} \end{bmatrix}$$

$$\text{Blog } \mathbf{x}_j = \begin{bmatrix} \text{avg. pages view/month} \\ \#\ \text{posts} \\ \vdots \\ \text{avg. } \#\ \text{comments/month} \\ \text{revenue from ads/month} \end{bmatrix}$$

Within such variables...

- Some variables are very **cheap to measure**, others **very expensive**.
- Some variables might have a **causal** effect on other variables.

- In the regression setting, the d variables are split between
  - k **regressor** (or **predictor**) variables
  - $d-k$ **response** (or **predicted**) variables.

Regression

- In the regression setting, the d variables are split between
  - k **regressor** (or **predictor**) variables
  - $d - k$ **response** (or **predicted**) variables.

guess **response**(**expensive**) variables using **regressor**(**cheap**) ones.

The Regression Problem

- Given,
  - A database $\{\mathbf{x}_1, \cdots, \mathbf{x}_N\} \Longleftrightarrow X = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots x_{1,N} \\ x_{2,1} & x_{2,2} & \cdots x_{2,N} \\ x_{3,1} & x_{3,2} & \cdots x_{3,N} \\ \vdots & \vdots & \vdots \\ x_{d,1} & x_{d,2} & \cdots x_{d,N} \end{bmatrix}$

The Regression Problem

- Given,
  - A database $\{\mathbf{x}_1, \cdots, \mathbf{x}_N\} \Longleftrightarrow X = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots x_{1,N} \\ x_{2,1} & x_{2,2} & \cdots x_{2,N} \\ x_{3,1} & x_{3,2} & \cdots x_{3,N} \\ \vdots & \vdots & \vdots \\ x_{d-1,1} & x_{d-1,2} & \cdots x_{d-1,N} \\ x_{d,1} & x_{d,2} & \cdots x_{d,N} \end{bmatrix}$
  - A set of k **regressors** variables **Reg** $\subset \{1, \cdots, d\}$
  - A set of $d-k$ **response** variable **Res** $\subset \{1, \cdots, d\}$

The Regression Problem

- Regression = **build a function** $f : \mathbb{R}^{\mathsf{k}} \to \mathbb{R}^{\mathsf{d\text{-}k}}$ such that,

$$\forall \mathbf{x}, f((\boldsymbol{x_i})_{\boldsymbol{i} \in \text{Reg}}) \approx (\boldsymbol{x_k})_{\boldsymbol{k} \in \text{Res}}.$$

- *e.g.* if $d = 6$, $k = 4$, **Reg**$= \{1, 2, 3, 4\}$, **Res**$= \{5, 6\}$ we look for a function $f : \mathbb{R}^4 \to \mathbb{R}^2$,

$$f(\boldsymbol{x_1, x_2, x_3, x_4}) \approx (\boldsymbol{x_5, x_6})$$

Examples continued

$$\text{Credit card holder } \mathbf{x}_j = \begin{bmatrix} \text{Income} \\ \text{Age} \\ \vdots \\ \text{Work history (months)} \\ \text{Family} \\ \#\text{ Credit Incidents} \end{bmatrix}$$

$$\text{Patient } \mathbf{x}_j = \begin{bmatrix} \text{height} \\ \text{weight} \\ \vdots \\ \#\text{ minutes exercise/week} \\ \text{LDL cholesterol} \\ \text{HDL cholesterol} \end{bmatrix}$$

$$\text{Blog } \mathbf{x}_j = \begin{bmatrix} \text{avg. pages view/month} \\ \#\text{ posts} \\ \vdots \\ \text{avg. }\#\text{ comments/month} \\ \text{revenue from ads/month} \end{bmatrix}$$

Examples continued

Credit card holder $\mathbf{x}_j = \begin{bmatrix} \text{Income} \\ \text{Age} \\ \vdots \\ \text{Work history (months)} \\ \text{Family} \\ \text{\# Credit Incidents} \end{bmatrix}$

Patient $\mathbf{x}_j = \begin{bmatrix} \text{height} \\ \text{weight} \\ \vdots \\ \text{\# minutes exercise/week} \\ \text{LDL cholesterol} \\ \text{HDL cholesterol} \end{bmatrix}$

Blog $\mathbf{x}_j = \begin{bmatrix} \text{avg. pages view/month} \\ \text{\# posts} \\ \vdots \\ \text{avg. \# comments/month} \\ \text{revenue from ads/month} \end{bmatrix}$

In the following slides...

We only consider tasks with **one response** variable

- All other variables are **regressors**.
- We rename the **response** variable **y** and reassign $x_1, \cdots, x_d$ for the **regressors**
- predicting **more than one** variable? heavier mathematically, but similar.

In the following slides...

We assume that **y** takes **continuous values**.

- When **y** takes discrete values, notably binary $\{0, 1\}$ things change a bit.
- Yet... **binary** $\subset$ **real** : regression techniques "work" on discrete data
- but **real** $\not\subseteq$ **binary**... we'll discuss that later.

Today's Example: Your apartment

Collected information about 285 (out of 1226) apartments close to Kyoto U.

Source: http://realestate.yahoo.co.jp/

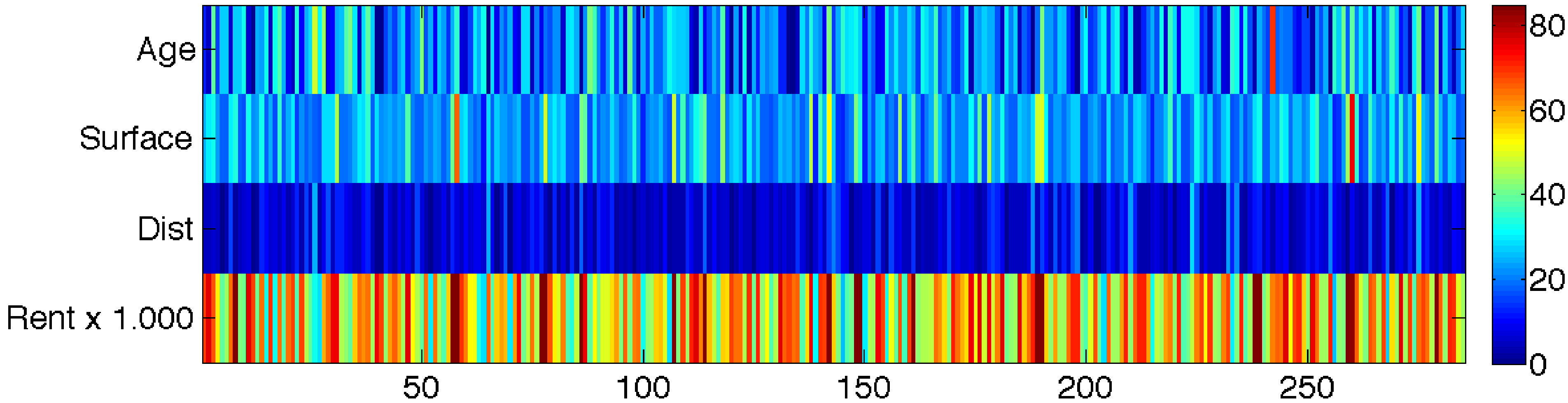
Today's Example: Your apartment

Collected information about 285 (out of 1226) apartments close to Kyoto U.

Kept 4 variables: **Surface**, **Rent**, **Age of Building**, **Walking distance to station**.

What does the matrix look like?

```
imagecs(H); colorbar;
```

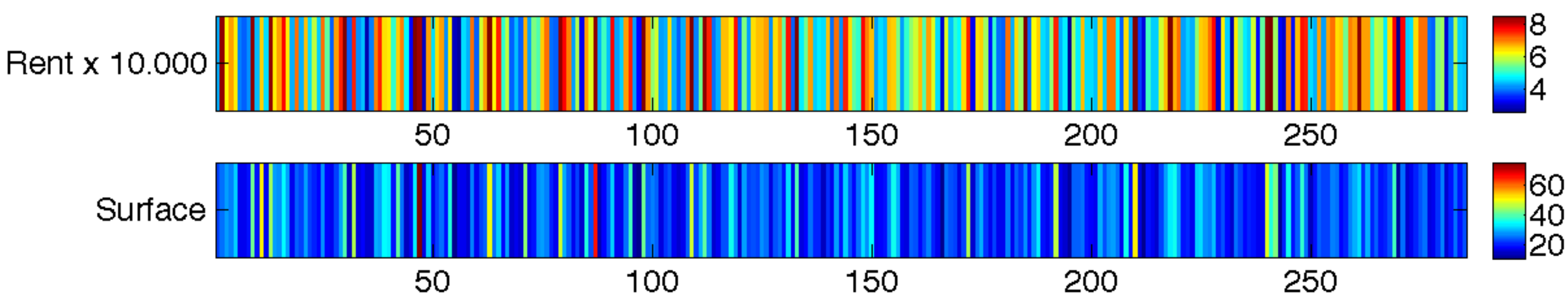


285 columns, 4 lines.

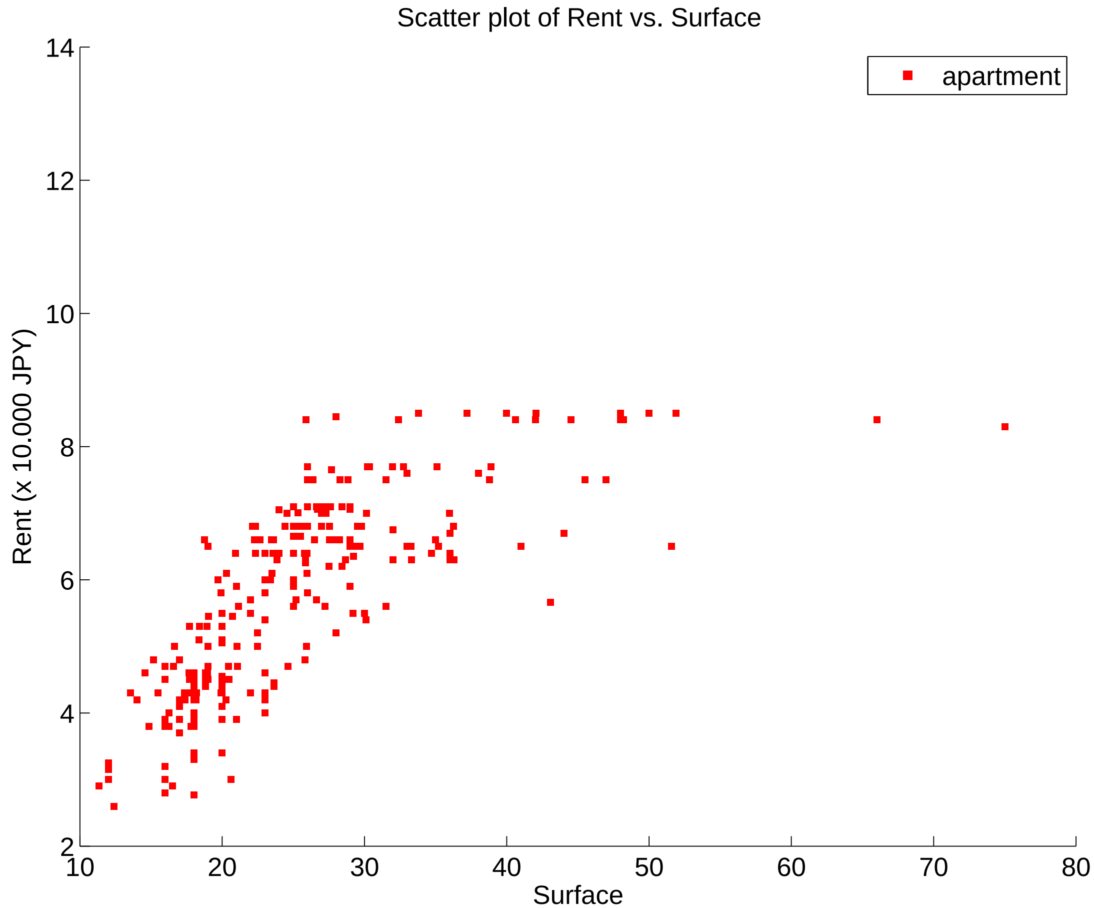
Each column represents one apartment.

In these slides, we will **regress** the rent using age, surface and distance

Regression: one variable vs. another

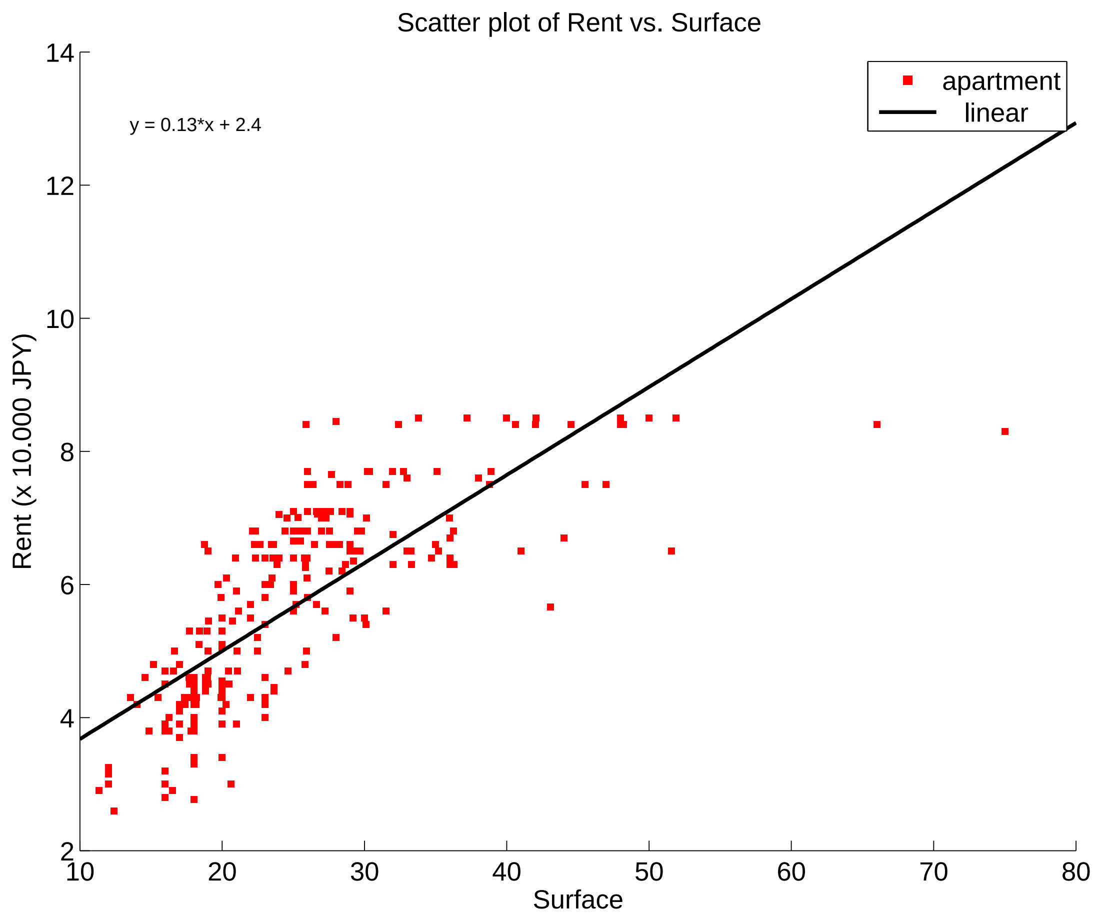


Rent vs. Surface



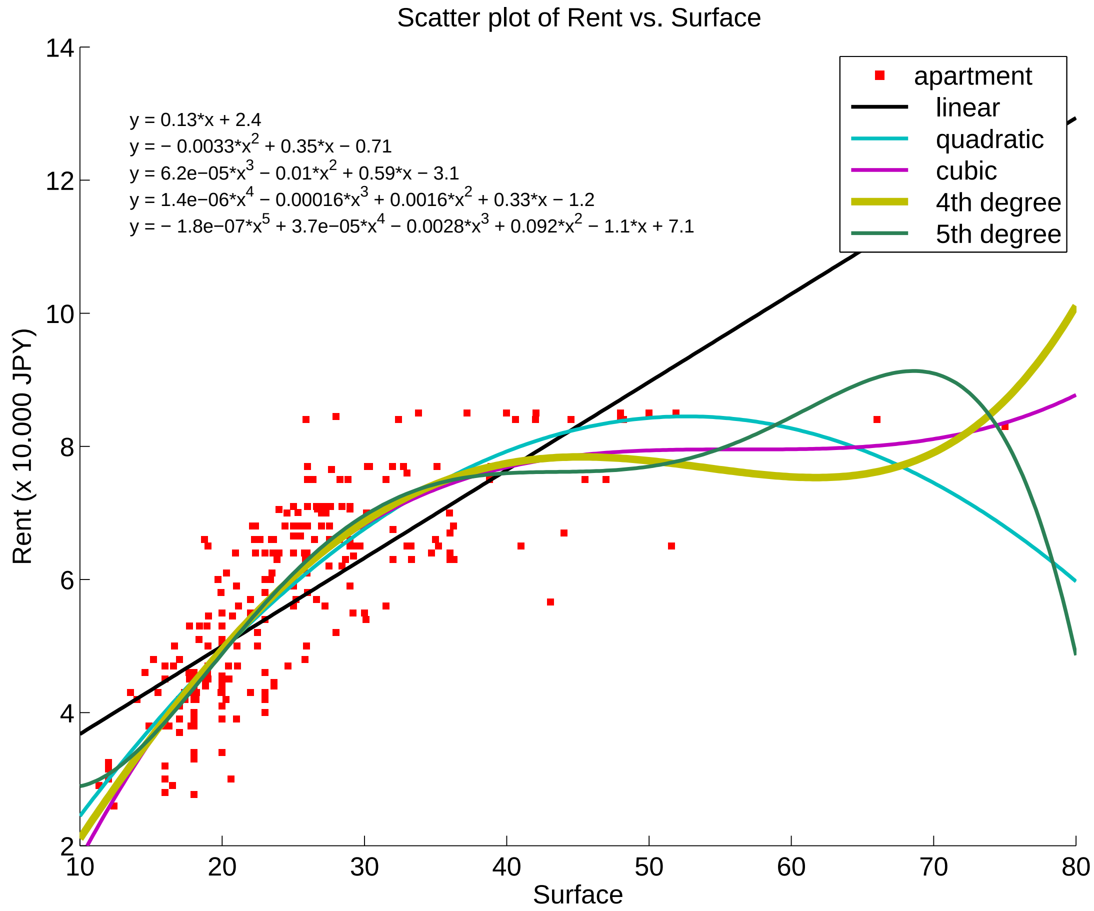
Note that the dataset has been censored above 85.000 JPY

Rent vs. Surface



Using the linear tool in curve fitting, we obtain the approximation $\mathbf{y} = 0.13x + 2.4$

Rent vs. Surface



We can use higher order polynomials... yet look at the results.

Behind the curve fitting tool

- Matlab selects these curves using the **least-squares** criterion e.g

$$\min_{\boldsymbol{f}\in\mathcal{F}} \sum_{j=1}^{N} (\boldsymbol{y_j} - \boldsymbol{f}(\boldsymbol{x_j}))^2$$

  where $\mathcal{F}$ is a **class of functions**

- Matlab considers a few function classes for $\mathcal{F}$.

Behind the curve fitting tool

- Matlab selects these curves using the **least-squares** criterion e.g

$$\min_{\boldsymbol{f}\in\mathcal{F}} \sum_{j=1}^{N} (\boldsymbol{y_j} - \boldsymbol{f}(\boldsymbol{x_j}))^2$$

  where $\mathcal{F}$ is a **class of functions**

- Matlab considers a few function classes for $\mathcal{F}$. Among them..
  - **Linear** $\min_{b,a_1\in\mathbb{R}} \sum_{j=1}^{N} \left(\boldsymbol{y_j} - (\boldsymbol{b} + \boldsymbol{a_1}\boldsymbol{x_j})\right)^2$

Behind the curve fitting tool

- Matlab selects these curves using the **least-squares** criterion e.g

$$\min_{f \in \mathcal{F}} \sum_{j=1}^{N} (y_j - f(x_j))^2$$

where $\mathcal{F}$ is a **class of functions**

- Matlab considers a few function classes for $\mathcal{F}$. Among them..
  - **Linear** $\min_{b,a_1 \in \mathbb{R}} \sum_{j=1}^{N} (y_j - (b + a_1 x_j))^2$
  - **Quadratic** $\min_{b,a_1,a_2 \in \mathbb{R}} \sum_{j=1}^{N} \left(y_j - (b + a_1 x_j + a_2 x_j^2)\right)^2$

Behind the curve fitting tool

- Matlab selects these curves using the **least-squares** criterion e.g

$$\min_{f \in \mathcal{F}} \sum_{j=1}^{N} (y_j - f(x_j))^2$$

where $\mathcal{F}$ is a **class of functions**

- Matlab considers a few function classes for $\mathcal{F}$. Among them..
  - **Linear** $\min_{b, a_1 \in \mathbb{R}} \sum_{j=1}^{N} (y_j - (b + a_1 x_j))^2$
  - **Quadratic** $\min_{b, a_1, a_2 \in \mathbb{R}} \sum_{j=1}^{N} \left(y_j - (b + a_1 x_j + a_2 x_j^2)\right)^2$
  - **Cubic** $\min_{b, a_1, a_2, a_3 \in \mathbb{R}} \sum_{j=1}^{N} \left(y_j - (b + a_1 x_j + a_2 x_j^2 + a_3 x_j^3)\right)^2$
  - *etc.*

How can we solve this? The linear case

- Let's take a look at the function

$$(a, b) \mapsto \sum_{j=1}^{N} \left(\boldsymbol{y_j} - (\boldsymbol{b} + \boldsymbol{a} x_j)\right)^2.$$

- Using the notations

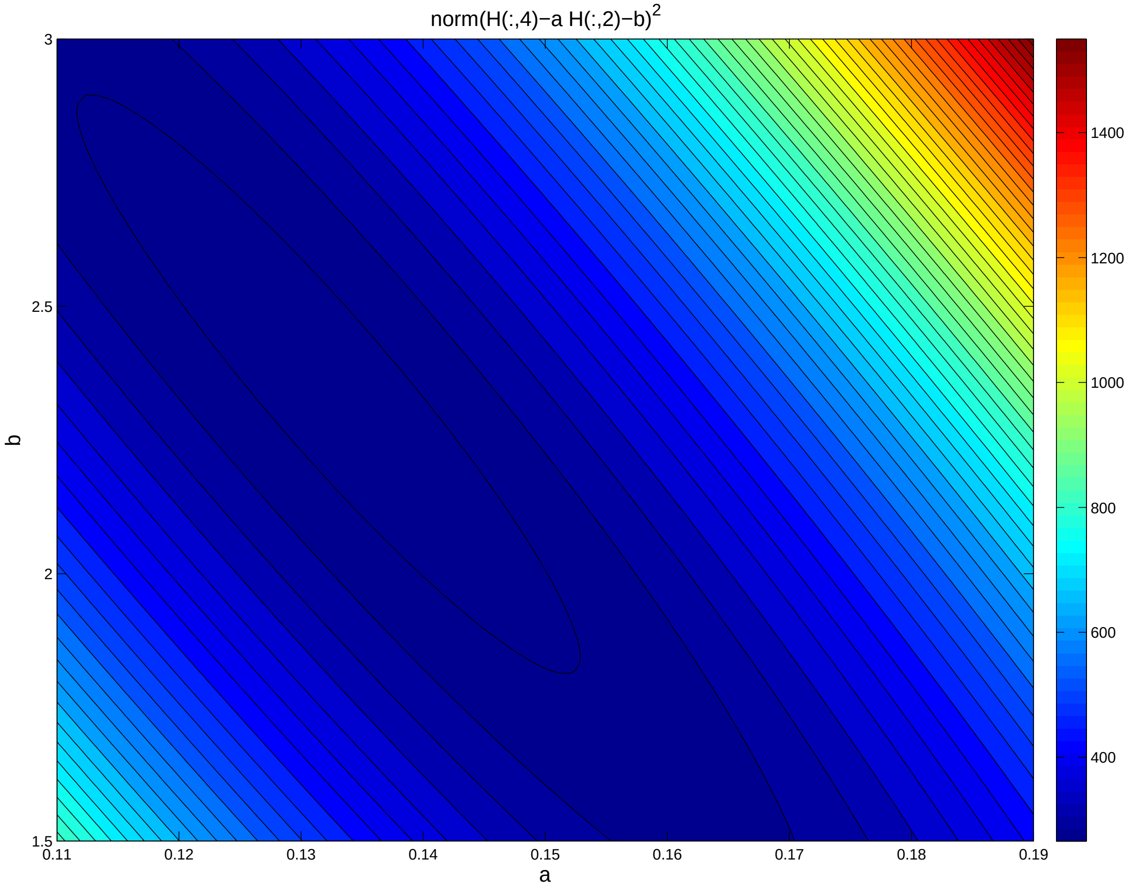
$$\begin{aligned} \text{Rent} \quad Y &= \begin{bmatrix} y_1 & y_2 & \cdots & y_N \end{bmatrix} \\ \text{Surface} \quad X &= \begin{bmatrix} x_1 & x_2 & \cdots & x_N \end{bmatrix} \\ \text{Constant} \quad \mathbf{1}_N &= \begin{bmatrix} 1 & 1 & \cdots & 1 \end{bmatrix} \end{aligned}$$

we have that

$$\sum_{j=1}^{N} \left(\boldsymbol{y_j} - (\boldsymbol{a}\boldsymbol{x_j} + \boldsymbol{b})\right)^2 = \|\boldsymbol{Y} - \boldsymbol{a}\boldsymbol{X} - \boldsymbol{b}\mathbf{1}_N\|^2$$

Contour plot of $(a, b) \to \|Y - aX - b\mathbf{1}_N\|^2$

- Since we only handle 2 parameters, we can make a contour plot



- This validates the equation $\mathbf{y} = 0.13x + 2.4$. How to get there?

Some linear algebra

- We define the function L as

$$L : (a, b) \mapsto \sum_{j=1}^{N} \left(\boldsymbol{y_j} - (\boldsymbol{b} + \boldsymbol{a} x_j)\right)^2$$

- The partial derivatives of L can be computed.

$$\frac{\partial L}{\partial a} = -2 \sum_{j=1}^{N} \left(\boldsymbol{y_j} - (\boldsymbol{b} + \boldsymbol{a} x_j)\right) \boldsymbol{x_j}$$

$$\frac{\partial L}{\partial b} = -2 \sum_{j=1}^{N} \boldsymbol{y_j} - (\boldsymbol{b} + \boldsymbol{a} x_j)$$

- Any minimum $(a^\star, b^\star)$ of L must be a saddle point.

Some linear algebra

- Namely, the partial derivatives of L at $(a^\star, b^\star)$ must be zero

$$\frac{\partial L}{\partial a} = 2\left(\boldsymbol{a}\sum \boldsymbol{x_j^2} + \boldsymbol{b}\sum \boldsymbol{x_j} - \sum y_j x_j\right)$$
$$\frac{\partial L}{\partial b} = 2\left(N\boldsymbol{b} - \sum y_j + \boldsymbol{a}\sum x_j\right)$$

- Hence $(a^\star, b^\star)$ **must satisfy** the linear system

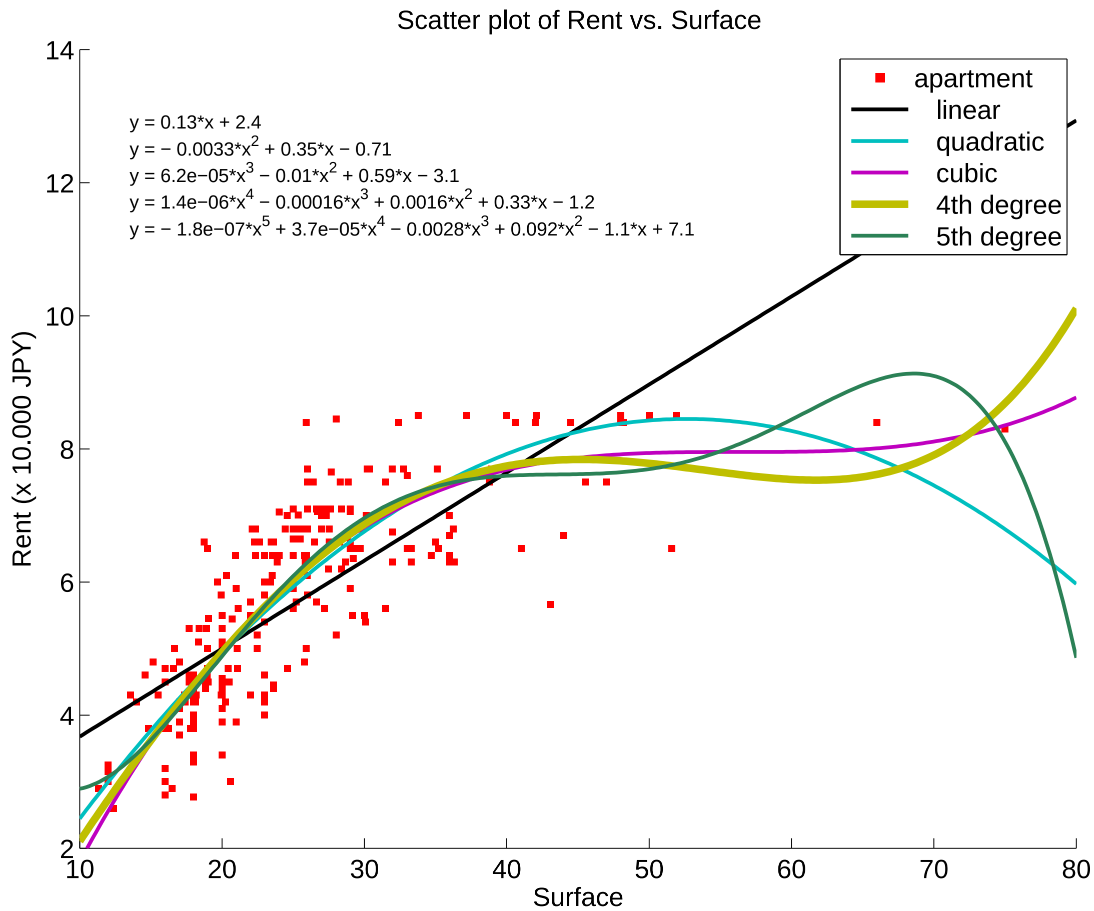
$$0 = a^\star \sum \boldsymbol{x_j^2} + b^\star \sum \boldsymbol{x_j} - \sum y_j \boldsymbol{x_j}$$
$$0 = N b^\star - \sum y_j + a^\star \sum \boldsymbol{x_j}$$

- Namely,

$$\begin{bmatrix} a^\star \\ b^\star \end{bmatrix} = \begin{bmatrix} \sum \boldsymbol{x_j^2} & \sum \boldsymbol{x_j} \\ \sum \boldsymbol{x_j} & N \end{bmatrix}^{-1} \begin{bmatrix} \sum y_j x_j \\ \sum y_j \end{bmatrix}$$

- `ans = 0.132248772789152 2.354203561671262`

Rent vs. Surface



We understood how to get the linear curve. What about the quadratic?

What about the quadratic case?

Quadratic $\min_{b,a_1,a_2\in\mathbb{R}} \sum_{j=1}^{N} \left(\boldsymbol{y_j} - (\boldsymbol{b} + \boldsymbol{a_1}\boldsymbol{x_j} + \boldsymbol{a_2}\boldsymbol{x_j^2})\right)^2$

- same idea... define

$$L : (a_1, a_2, b) \mapsto \sum_{j=1}^{N} \left(\boldsymbol{y_j} - \left(\boldsymbol{b} + \boldsymbol{a_1}\boldsymbol{x_j} + \boldsymbol{a_2}\boldsymbol{x_j^2}\right)\right)^2$$

- look at the objective's derivatives...

$$\frac{\partial L}{\partial a_2} = -2\sum_{j=1}^{N} \left(\boldsymbol{y_j} - \left(\boldsymbol{b} + \boldsymbol{a_1}\boldsymbol{x_j} + \boldsymbol{a_2}\boldsymbol{x_j^2}\right)\right) \boldsymbol{x_j^2}$$

$$\frac{\partial L}{\partial a_1} = -2\sum_{j=1}^{N} \left(\boldsymbol{y_j} - \left(\boldsymbol{b} + \boldsymbol{a_1}\boldsymbol{x_j} + \boldsymbol{a_2}\boldsymbol{x_j^2}\right)\right) \boldsymbol{x_j}$$

$$\frac{\partial L}{\partial b} = -2\sum_{j=1}^{N} \left(\boldsymbol{y_j} - \left(\boldsymbol{b} + \boldsymbol{a_1}\boldsymbol{x_j} + \boldsymbol{a_2}\boldsymbol{x_j^2}\right)\right)$$

What about the quadratic case?

- We consider the equations that a saddle point must verify:

$$0 = \sum_{j=1}^{N} \left(\boldsymbol{y_j} - \left(b^\star + a_1^\star \boldsymbol{x_j} + a_2^\star \boldsymbol{x_j^2} \right) \right) \boldsymbol{x_j^2}$$

$$0 = \sum_{j=1}^{N} \left(\boldsymbol{y_j} - \left(b^\star + a_1^\star \boldsymbol{x_j} + a_2^\star \boldsymbol{x_j^2} \right) \right) \boldsymbol{x_j}$$

$$0 = \sum_{j=1}^{N} \left(\boldsymbol{y_j} - \left(b^\star + a_1^\star \boldsymbol{x_j} + a_2^\star \boldsymbol{x_j^2} \right) \right)$$

$$\begin{bmatrix} a_2^\star \\ a_1^\star \\ b^\star \end{bmatrix} = \begin{bmatrix} \sum \boldsymbol{x_j^4} & \sum \boldsymbol{x_j^3} & \sum \boldsymbol{x_j^2} \\ \sum \boldsymbol{x_j^3} & \sum \boldsymbol{x_j^2} & \sum \boldsymbol{x_j} \\ \sum \boldsymbol{x_j^2} & \sum \boldsymbol{x_j} & N \end{bmatrix}^{-1} \begin{bmatrix} \sum \boldsymbol{y_j x_j^2} \\ \sum \boldsymbol{y_j x_j} \\ \sum \boldsymbol{y_j} \end{bmatrix}$$

- `ans = -0.003306463076068 0.347969105896777 -0.705157514974559`

Higher order polynomials

- Intuitively, for polynomial up to degree p we would have to
  - Build the corresponding Toeplitz Matrix
  - Build the corresponding vector with y and x combined at different exponents
  - Solve the linear system
- Surprisingly

Finding the **best p^{th} order polynomial** with **least-squares**

$$\Updownarrow$$

Solving a p **dimensional** linear system

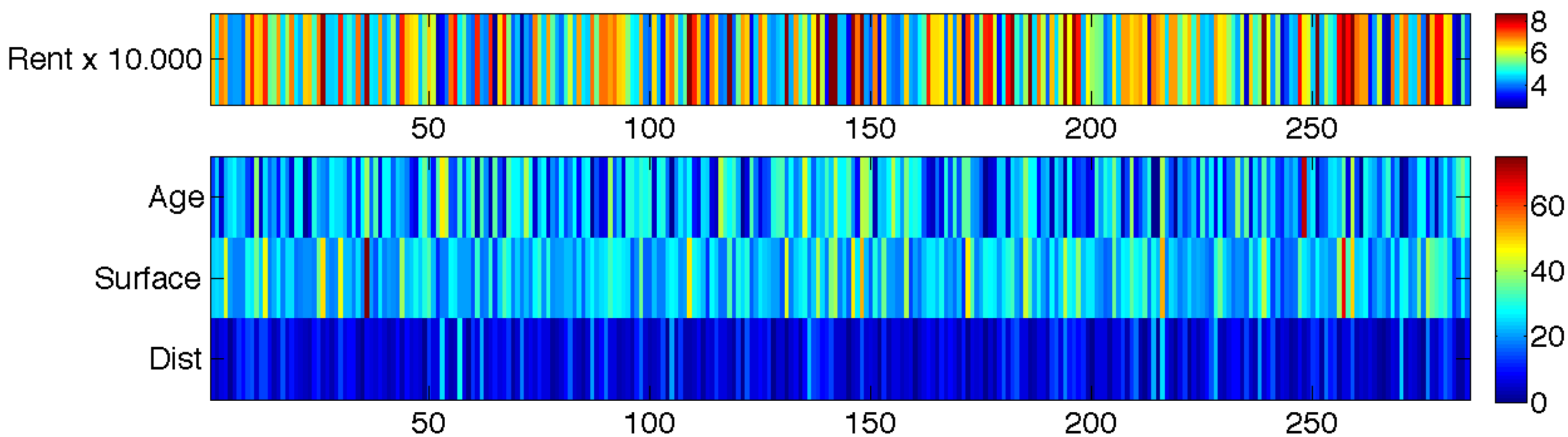
- Not so surprising after all:
  - Least-squares: objective of degree 2 in coefficients
  - Minimum $\Leftrightarrow$ saddle point $\Leftrightarrow$ system of degree 1..
  - Least-squares has been chosen **because** it yields a linear system...

The general case: one vs. all rest

- What about using all other variables?

$$\text{Rent} \quad Y = \begin{bmatrix} y_1 & y_2 & \cdots & y_N \end{bmatrix}$$

$$\text{All other variables} \quad X = \begin{bmatrix} \vdots & \vdots & \cdots & \vdots \\ \mathbf{x}_1 & \mathbf{x}_2 & \cdots & \mathbf{x}_N \\ \vdots & \vdots & \cdots & \vdots \end{bmatrix}$$



The general case

- We assume that we have d **regressor** variables, 1 **response** variable.
- Consider again the **linear** approach. We look for a function f of the form

$$f(\mathbf{x}) = \boldsymbol{\alpha_0} + \boldsymbol{\alpha_1} x_1 + \boldsymbol{\alpha_2} x_2 + \cdots + \boldsymbol{\alpha_d} x_d.$$

- We want to determine $d+1$ weights,
  - a constant $\boldsymbol{\alpha_0}$
  - $\boldsymbol{1 \leq i \leq d, \alpha_i}$ weights for each variable.

- **Least squares**:

$$L(\alpha_0, \alpha_1, \alpha_2, \cdots, \alpha_d) = \sum_{j=1}^{N} \Big(\boldsymbol{y_j} - (\boldsymbol{\alpha_0} + \boldsymbol{\alpha_1} x_{1,j} + \boldsymbol{\alpha_2} x_{2,j} + \cdots + \boldsymbol{\alpha_d} x_{d,j}) \Big)^2$$

The general case

- Notice that

$$L(\alpha_0, \alpha_1, \alpha_2, \cdots, \alpha_d) \to \sum_{i=1}^{N} \left(y_i - \left(\alpha_0 + \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_d \end{bmatrix}^T \mathbf{x}_i \right) \right)^2 = \left\| \begin{bmatrix} \alpha_0 \\ \vdots \\ \alpha_d \end{bmatrix}^T X - Y \right\|^2,$$

where

$$X = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ \vdots & \vdots & \cdots & \vdots \\ \mathbf{x}_1 & \mathbf{x}_2 & \cdots & \mathbf{x}_N \\ \vdots & \vdots & \cdots & \vdots \end{bmatrix} \in \mathbb{R}^{d+1 \times N}$$

and

$$Y = \begin{bmatrix} y_1 & \cdots & y_N \end{bmatrix} \in \mathbb{R}^N.$$

- We write α for the $d+1$ dimensional vector $\begin{bmatrix} \alpha_0 \\ \vdots \\ \alpha_d \end{bmatrix}$.

Linear least squares

- Expanding this expression,

$$L(\alpha) = \left(\alpha^T XX^T\alpha - 2YX^T\alpha + \|Y\|^2\right)$$

- Consider the **gradient** of that function

$$\nabla L = 2XX^T\alpha - 2XY^T$$

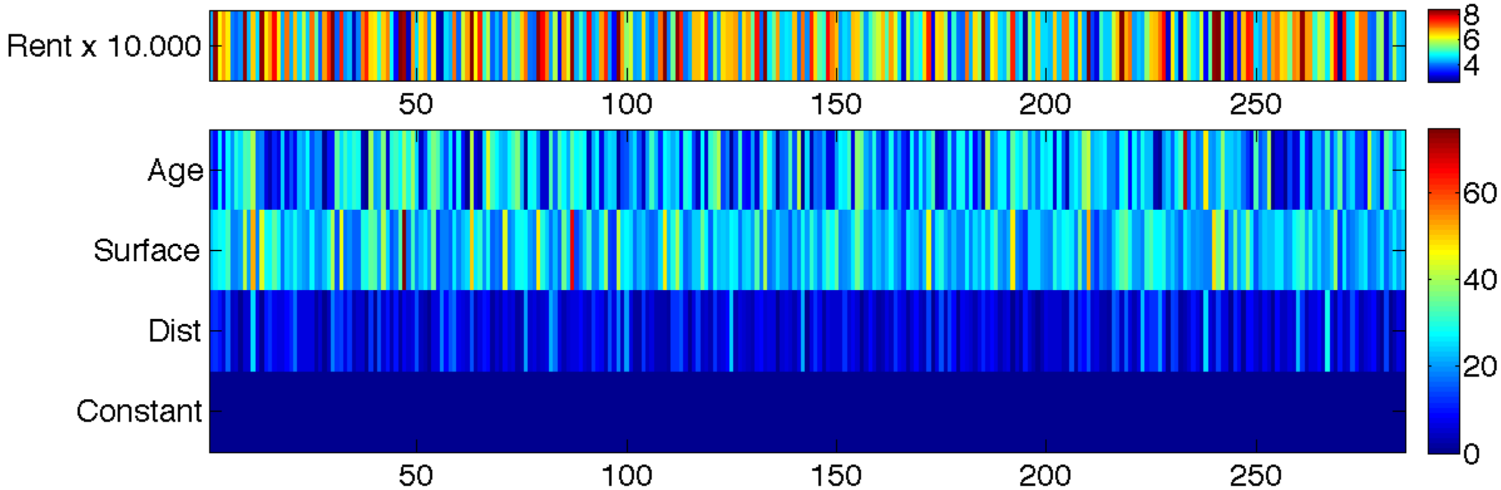
- Hence this gradient is zero for

$$\alpha^\star = (XX^T)^{-1}XY^T$$

- $XX^T \in \mathbf{S}_+^n$, that is XX^T is a positive (semi)definite matrix.
- This works if $XX^T \in \mathbb{R}^{d+1}$ is **invertible**, that is $XX^T \in \mathbf{S}_{++}^n$.

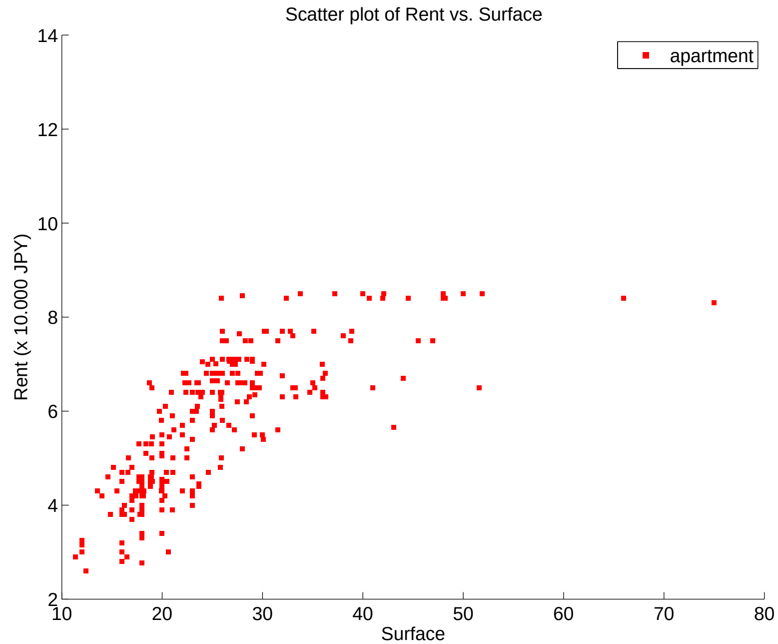
Considering again rents vs the rest

- Getting the data again, adding a line of $\mathbf{1}'s$



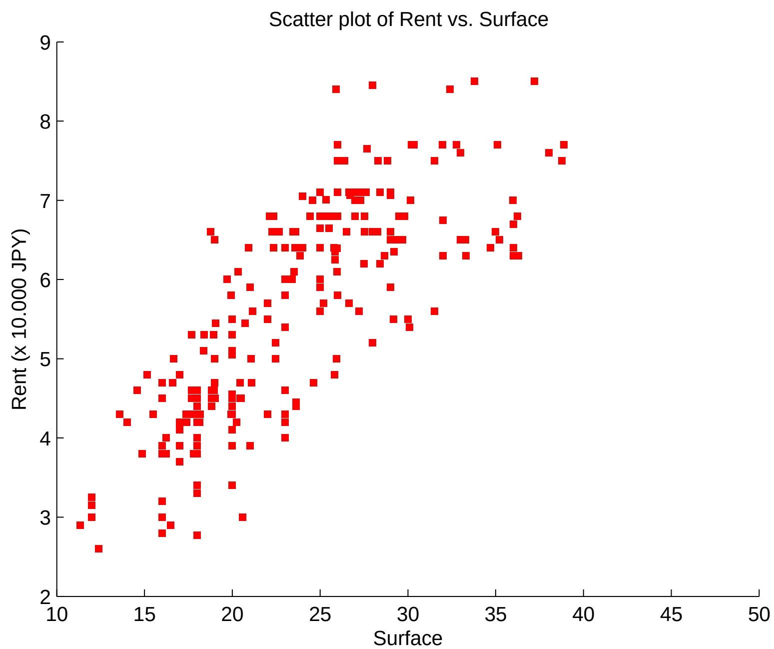

```
>> (X*X')\(X*Y')
ans =
    0.000141678721821
    0.004226687659299
   -0.012599982792209
    5.611128285287092
```

What went wrong?



rent= 1.4 age + 42.2 surf -125 dist + 56.110 JPY

What happens if we remove outliers? (surf> 40)




```
>> (X*X')\(X*Y')
ans =
   -0.049332605603095     x age
    0.163122792160298     x surface
   -0.004411580036614     x distance
    2.731204399433800     + 27.300 JPY
```

Moral of the story: easy to draw wrong conclusions **even with simple tools**

What else can go wrong? Next time...

- What happens when $d \gg n$? (XX^T) is **no longer invertible**...
  - high-dimensional data in genomics,
  - images analysis (lots of features)

- What happens when (XX^T) is **badly conditioned** ($\frac{\lambda_{\min}(XX^T)}{\lambda_{\max}(XX^T)} \approx 0$)?
  - if $\lambda_{\min}(XX^T) = 1e - 10$, $\lambda_{\max}\left((XX^T)^{-1}\right) = 1e10$!!
  - Very bad numerical stability of the solution...

- When $d \gg n$, we might want to do **variable selection**,
  - $i.e.$pick a subset d' of the d variables which is relevant to predict **y**.
  - $i.e.$favor vectors α such that $\|\alpha\|_0 = \text{card}\{\alpha_i \neq 0\}$ is small.