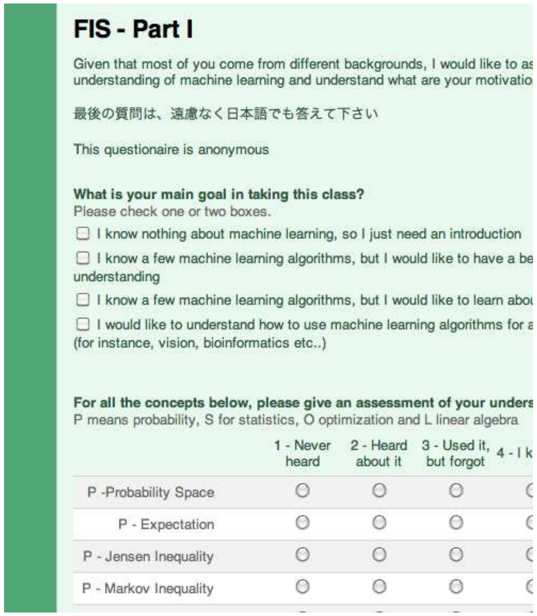
Foundation of Intelligent Systems, Part I

Regression

mcuturi@i.kyoto-u.ac.jp

Before starting

- Please take this **survey** before the end of this week.



FIS - Part I

Given that most of you come from different backgrounds, I would like to as
understanding of machine learning and understand what are your motivatio

最後の質問は、遠慮なく日本語でも答えて下さい

This questionaire is anonymous

What is your main goal in taking this class?
Please check one or two boxes.

- ☐ I know nothing about machine learning, so I just need an introduction
- ☐ I know a few machine learning algorithms, but I would like to have a be understanding
- ☐ I know a few machine learning algorithms, but I would like to learn abou
- ☐ I would like to understand how to use machine learning algorithms for a (for instance, vision, bioinformatics etc..)

For all the concepts below, please give an assessment of your unders
P means probability, S for statistics, O optimization and L linear algebra

	1 - Never heard	2 - Heard about it	3 - Used it, but forgot	4 - I k
P -Probability Space	○	○	○	○
P - Expectation	○	○	○	○
P - Jensen Inequality	○	○	○	○
P - Markov Inequality	○	○	○	○

- Here are a few books which you can check beyond the slides.
  - Elements of Statistical Learning, Hastie Tibshirani Friedman
  - Pattern Recognition, Theodoridis Koutroumbras
  - Pattern Recognition & Machine Learning, Bishop
- You can also check Andrew Ng's video lectures (Stanford)

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→)
  - SVM for regression (1996→)
  - Compressed Sensing (2002→)

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 $x_{i,j}$ of a given point $\mathbf{x}_j$ $1 \leq i \leq d$ is a number.

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

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

Within such variables...

- Some variables are very **cheap to measure**, others **very expensive**
- Some variables might have a strong **influence** on other variables

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

  ...to highlight such a difference or **guess** **expensive** variables from **cheap** ones.

The Regression Problem

- Given,
  - A database $\{\mathbf{x}_1, \cdots, \mathbf{x}_N\} \Longleftrightarrow X = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots x_{1,N} \\ x_{2,1} & x_{2,2} & \cdots x_{2,N} \\ x_{3,1} & x_{3,2} & \cdots x_{3,N} \\ \vdots & \vdots & \vdots \\ x_{d,1} & x_{d,2} & \cdots x_{d,N} \end{bmatrix}$
  - A set of k **regressors** variables **Reg** $\subset \{1, \cdots, d\}$
  - A set of $d - k$ **response** variable **Res** $\subset \{1, \cdots, d\}$
- Regression = **build a function** $f : \mathbb{R}^{\mathbf{k}} \rightarrow \mathbb{R}^{\mathbf{d\text{-}k}}$ such that,

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

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

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

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

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

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

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

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.

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** $\not\subset$ **binary**... we'll discuss that later.

Today's Example: Your apartment

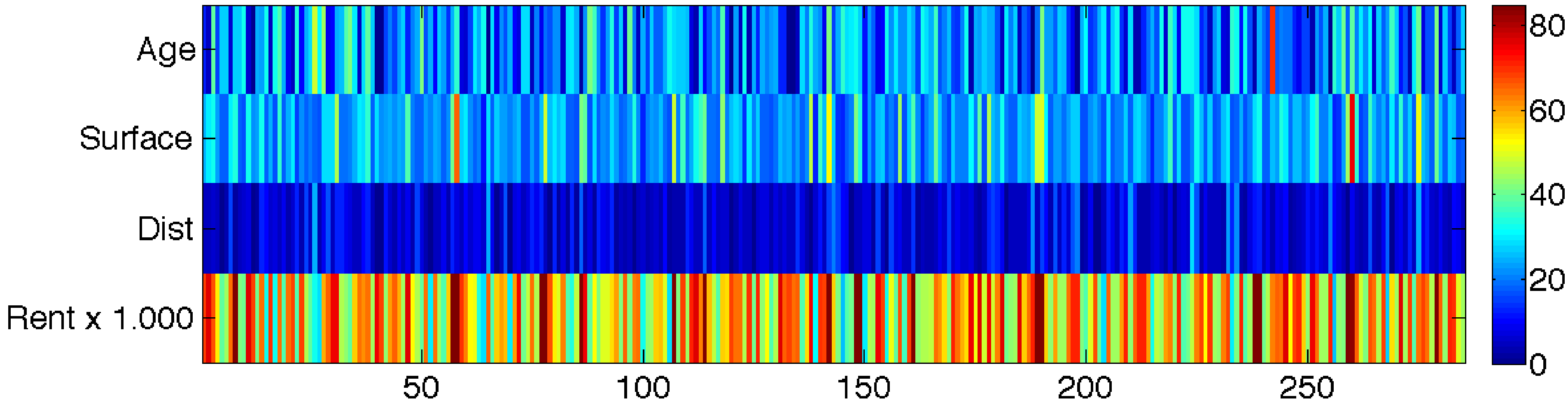
画像	路線名/駅名 住所	バス 徒歩	賃料 管理費等	敷金または保証金 礼金(敷引)	間取り 専有面積	築年月 (築年数)	選択
NEW!	京阪鴨東線/出町柳 京都市左京区田中大堰町	ー 5分	4.20万円 3,000円	5万円 5万円(ー)	1R 16.00m²	'89/09 (築22年)	
	叡山本線/修学院 京都市左京区山端滝ケ鼻町	ー 7分	4.30万円 2,000円	5万円 なし(なし)	1K 20.44m²	'95/03 (築17年)	
	叡山本線/修学院 京都市左京区山端滝ケ鼻町	ー 7分	4.30万円 2,000円	5万円 なし(なし)	1K 20.44m²	'95/03 (築17年)	
	叡山本線/一乗寺 京都市左京区一乗寺梅ノ木町	ー 6分	4.30万円 2,000円	5万円 なし(ー)	1K 20.00m²	'88/03 (築24年)	
	京阪鴨東線/出町柳 京都市左京区吉田上阿達町	ー 7分	4.30万円 2,000円	5万円 5万円(なし)	1K 19.02m²	'85/02 (築27年)	
	烏丸線/今出川 京都市上京区柳図子町	ー 3分	4.30万円 2,000円	なし 10万円(ー)	1R 17.70m²	'84/10 (築27年)	
NEW!	烏丸線/今出川 京都市上京区北小路室町	ー 2分	4.50万円 なし	5万円 5万円(ー)	1K 16.00m²	'94/03 (築18年)	
	京阪鴨東線/出町柳 京都市左京区田中下柳町	ー 3分	4.30万円 2,500円	5万円 8万円(ー)	1K 20.00m²	'85/03 (築27年)	
	叡山本線/修学院 京都市左京区高野泉町	ー 5分	4.05万円 5,500円	6万円 5万円(なし)	1K 14.58m²	'79/01 (築33年)	

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;`

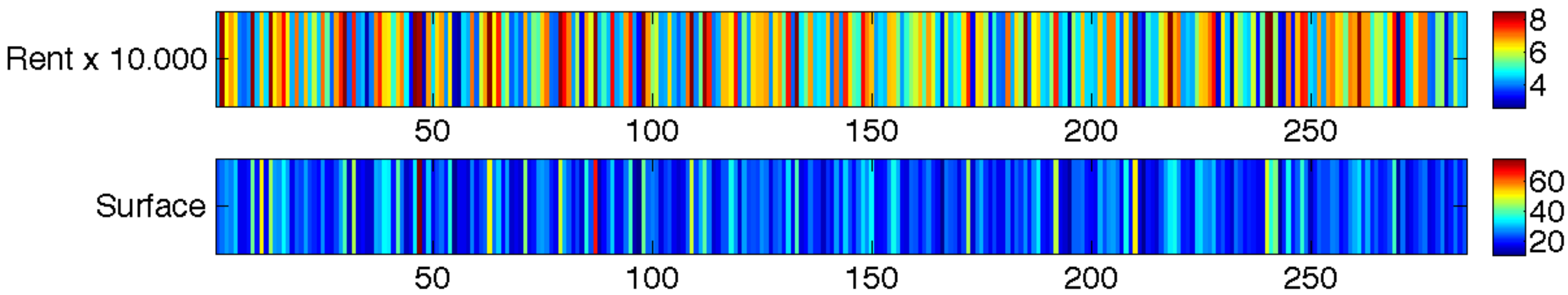


285 columns, 4 lines.

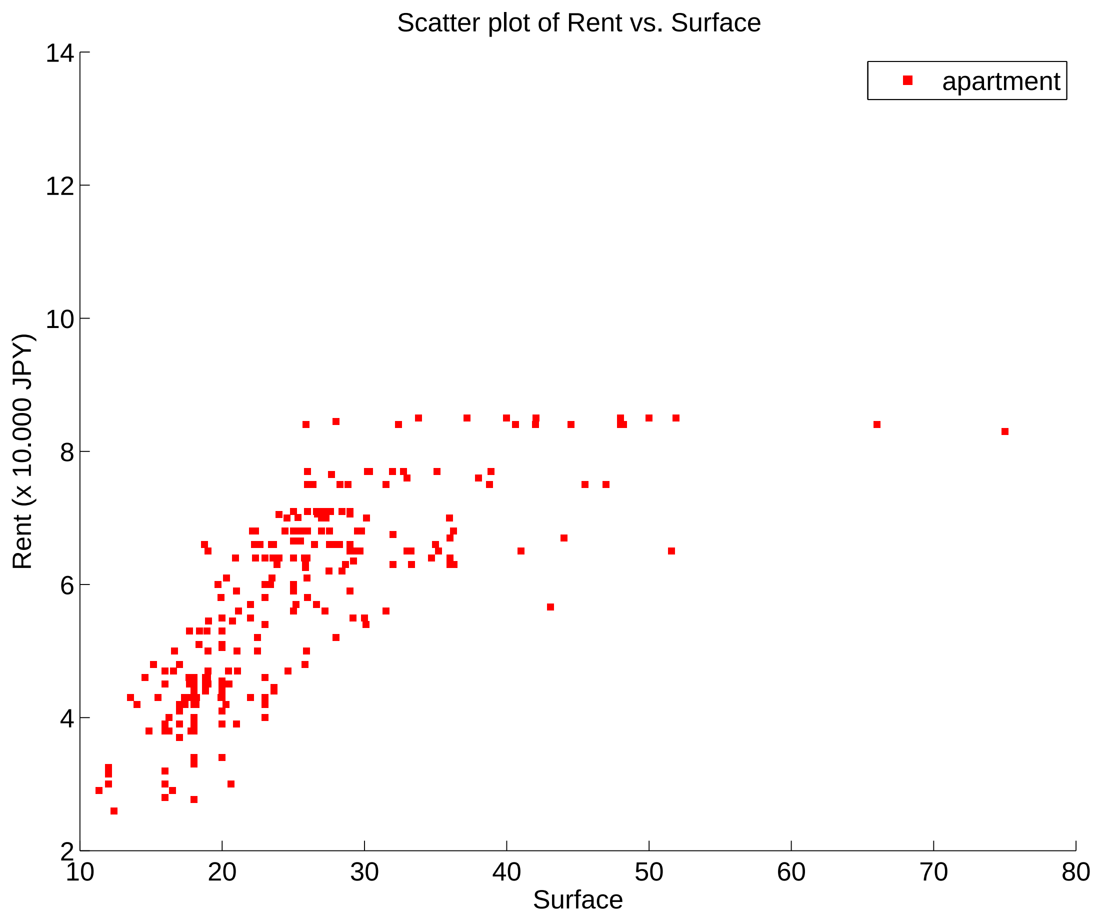
Each column represents one apartment.

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

Regression: one variable vs. another

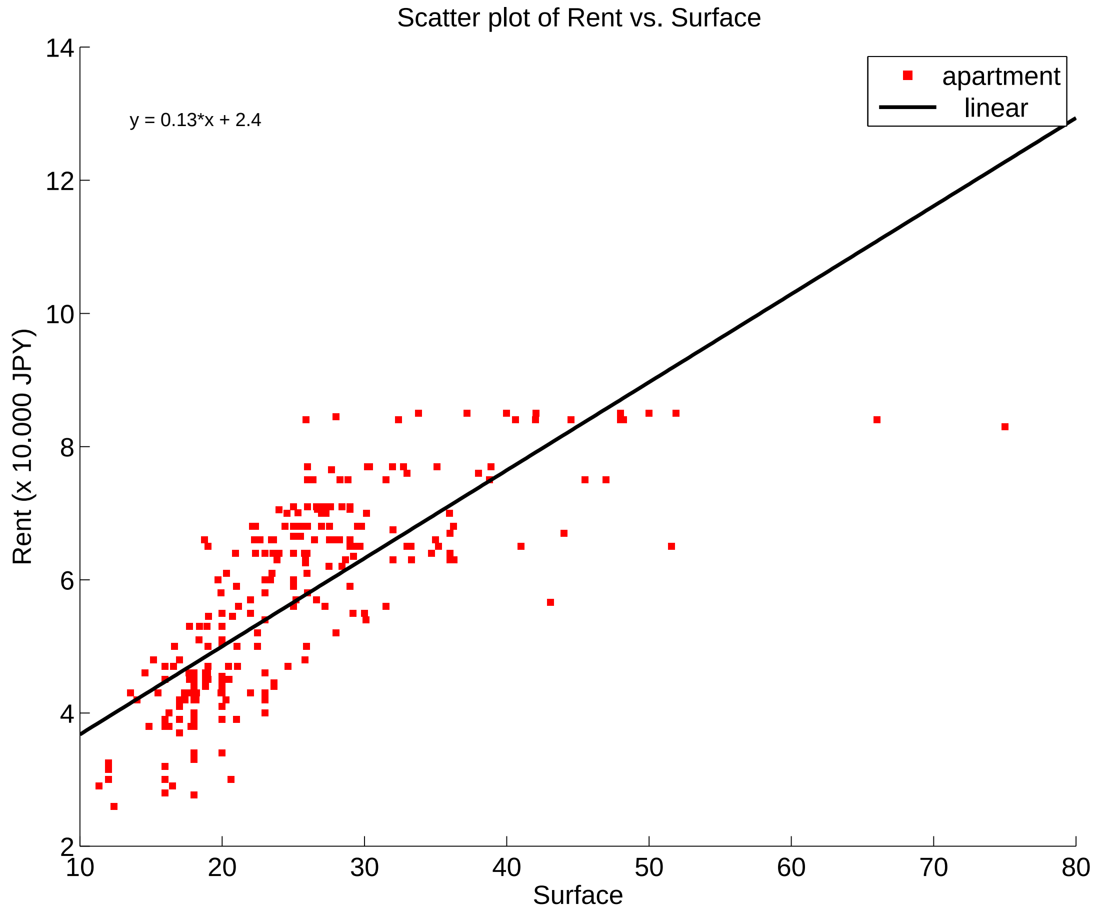


Rent vs. Surface



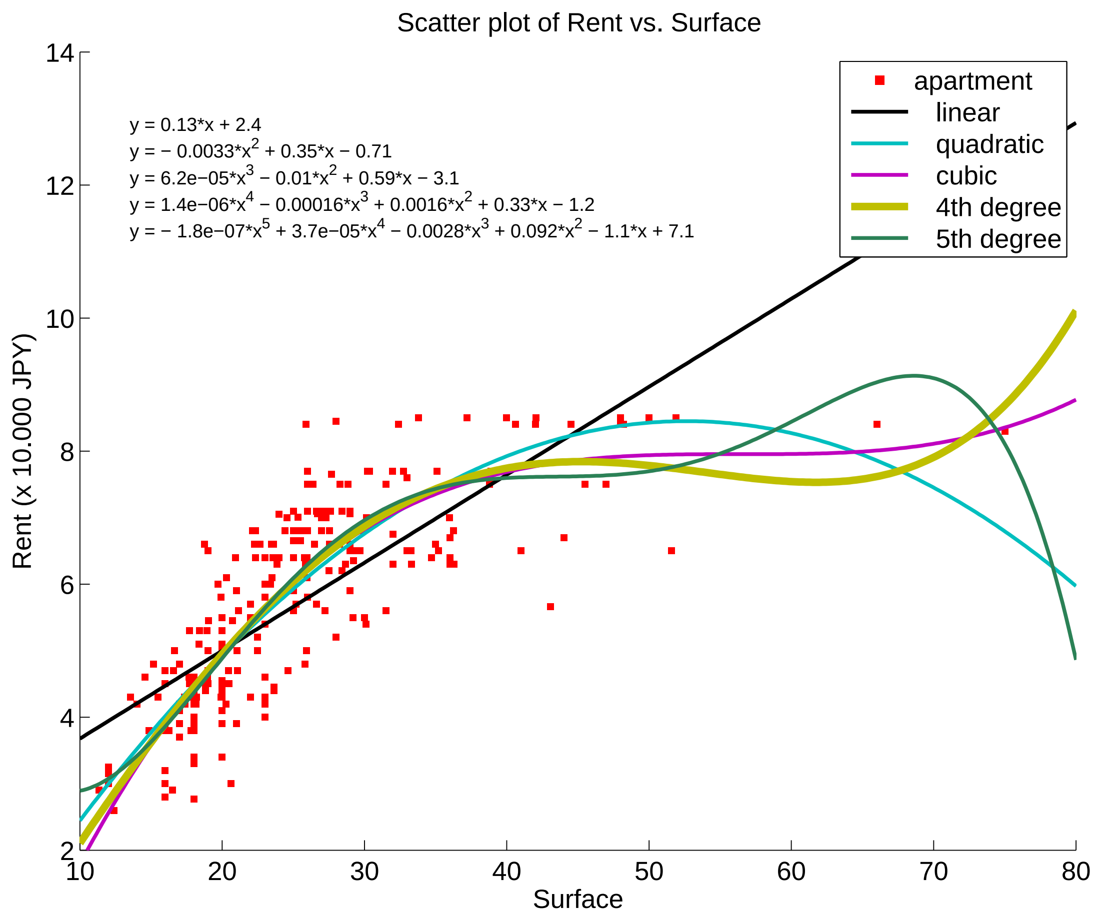
Note that the dataset has been censored above 85.000 JPY

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 tool

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

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

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

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

How can we solve this? The linear case

- Let's take a look at the function

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

- Using the notations

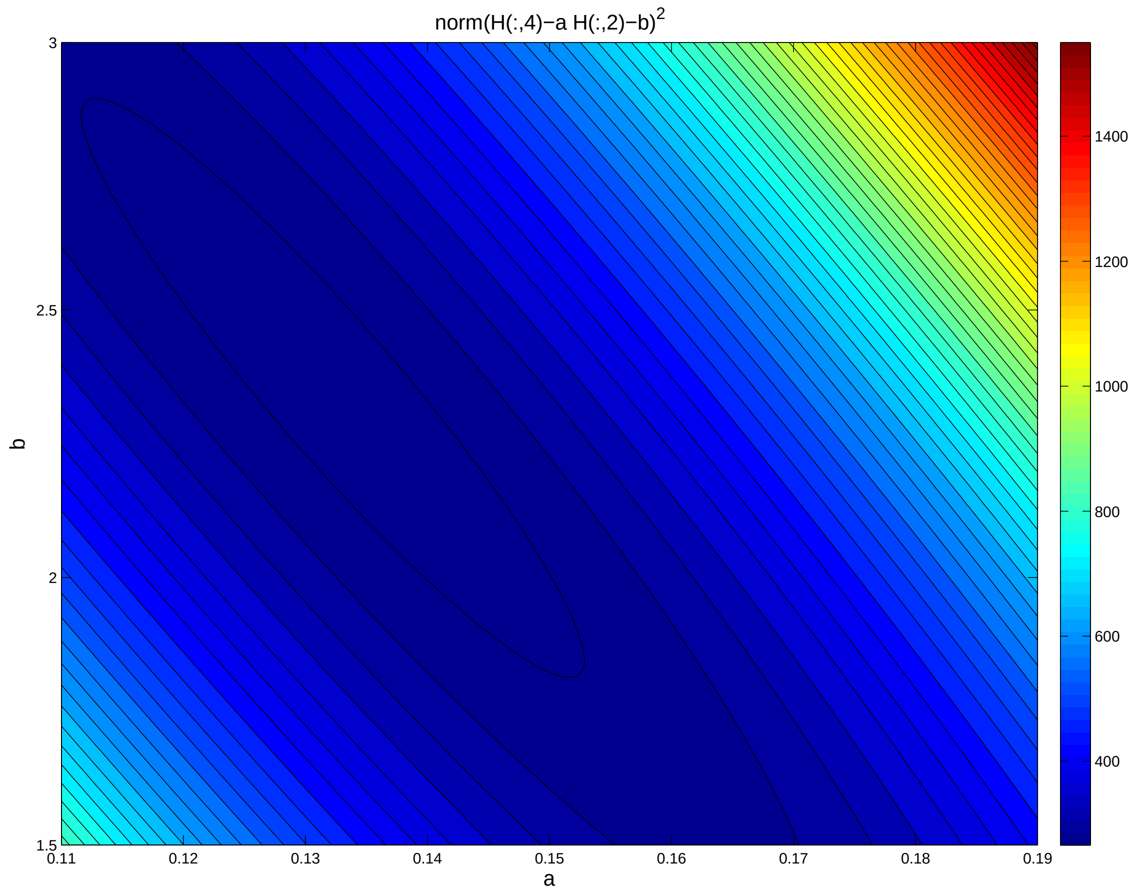
$$\begin{aligned} \text{Rent} \quad Y &= \begin{bmatrix} y_1 & y_2 & \cdots & y_N \end{bmatrix} \\ \text{Surface} \quad X &= \begin{bmatrix} x_1 & x_2 & \cdots & x_N \end{bmatrix} \\ \text{Constant} \quad \mathbf{1}_N &= \begin{bmatrix} 1 & 1 & \cdots & 1 \end{bmatrix} \end{aligned}$$

we have that

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

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

- 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

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

- The partial derivatives of L can be computed.

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

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

- 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

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

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

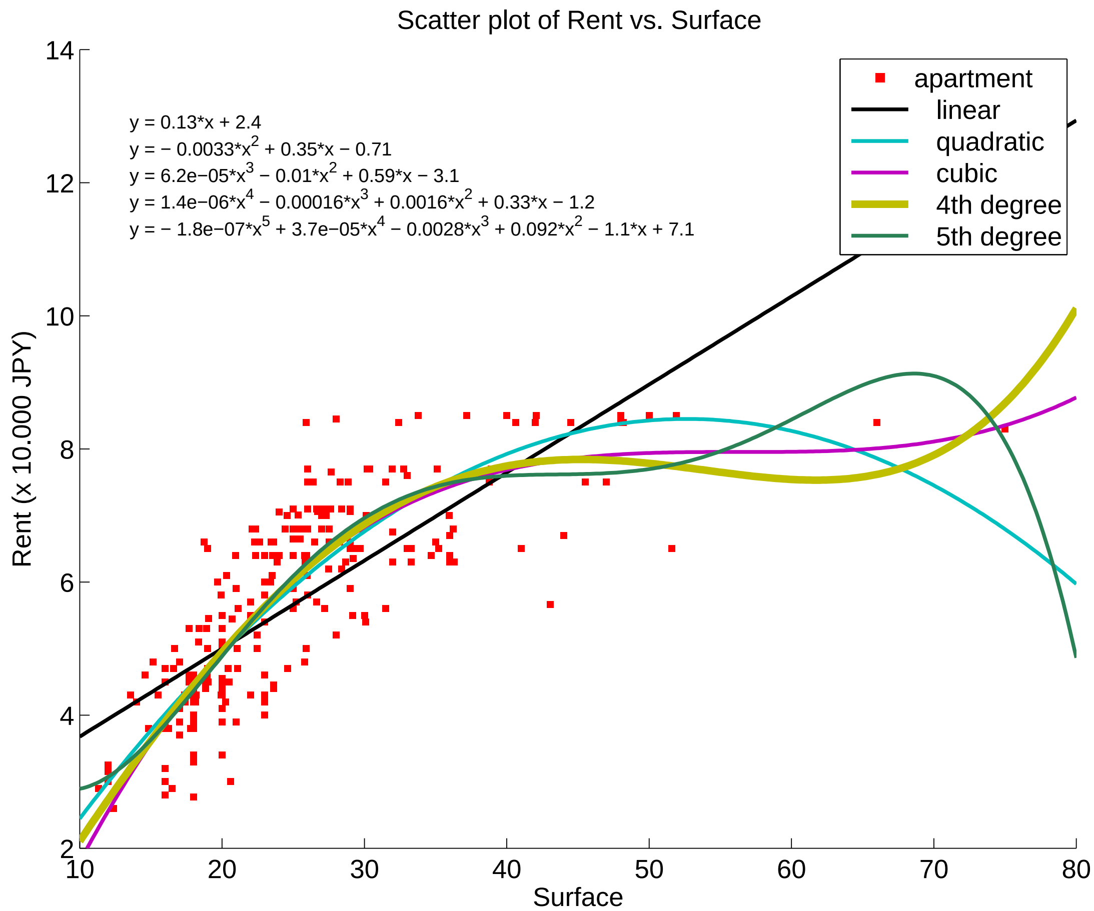
$$0 = a^\star \sum x_j^2 + b^\star \sum x_j - \sum y_j x_j$$
$$0 = N b^\star - \sum y_j + a^\star \sum x_j$$

- Namely,

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

- `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?

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

- same idea... define

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

- look at the objective's derivatives...

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

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

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

What about the quadratic case?

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

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

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

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

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

- `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

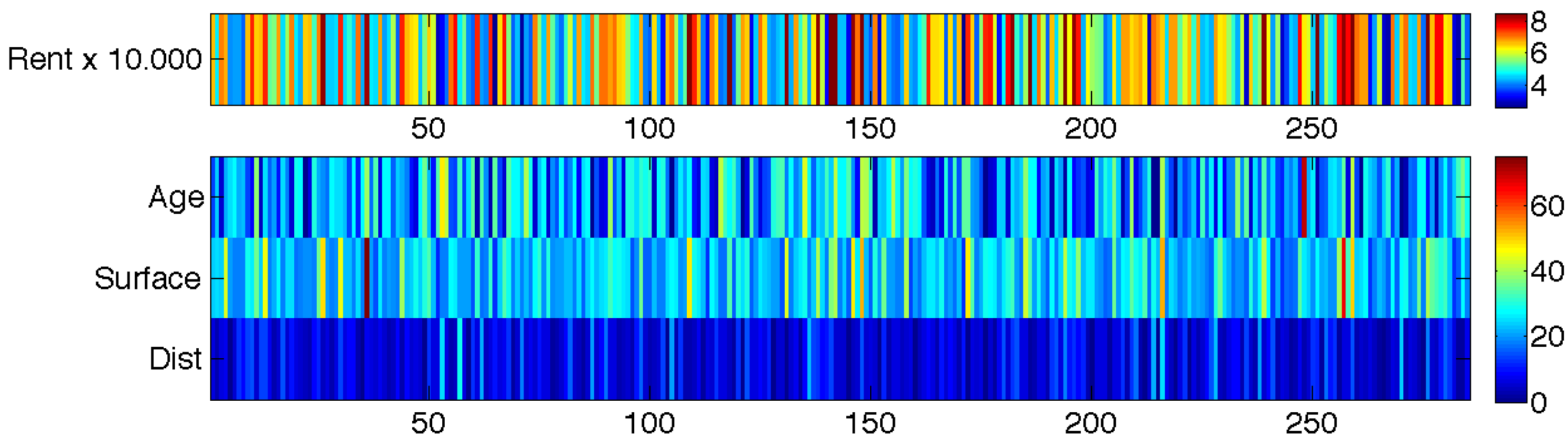
- Not so surprising after all:
  - Least-squares: objective of degree 2 in coefficients
  - Minimum $\Leftrightarrow$ saddle point $\Leftrightarrow$ system of degree 1..
  - Least-squares has been chosen **because** it yields a linear system...

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**:

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

The general case

- Notice that

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

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

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

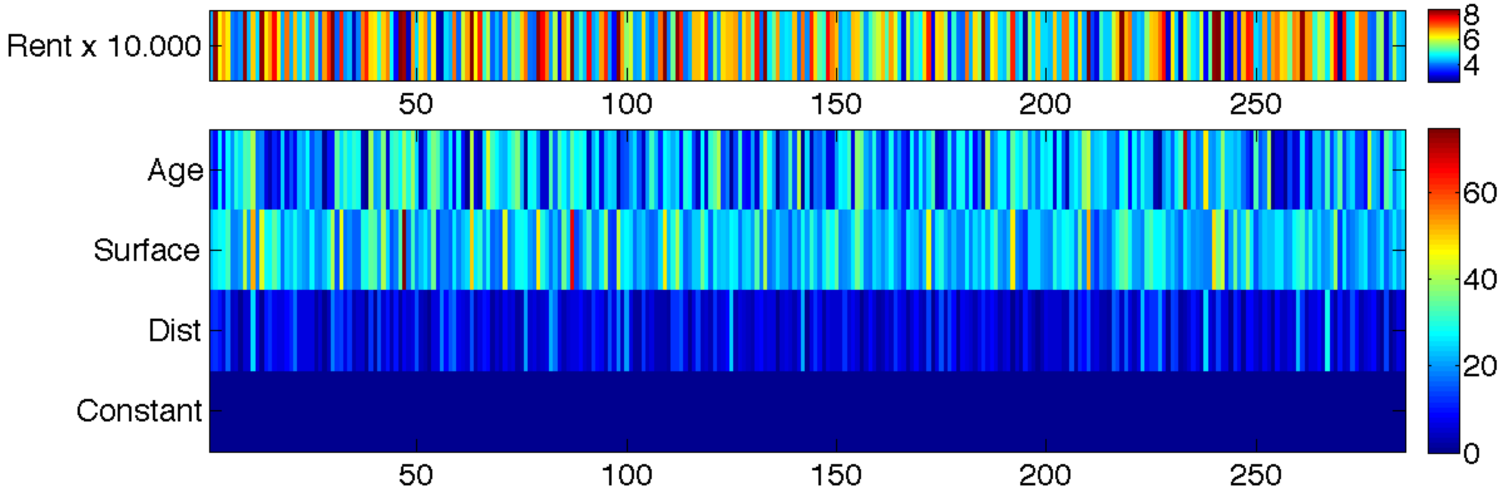
- Hence this gradient is zero for

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

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

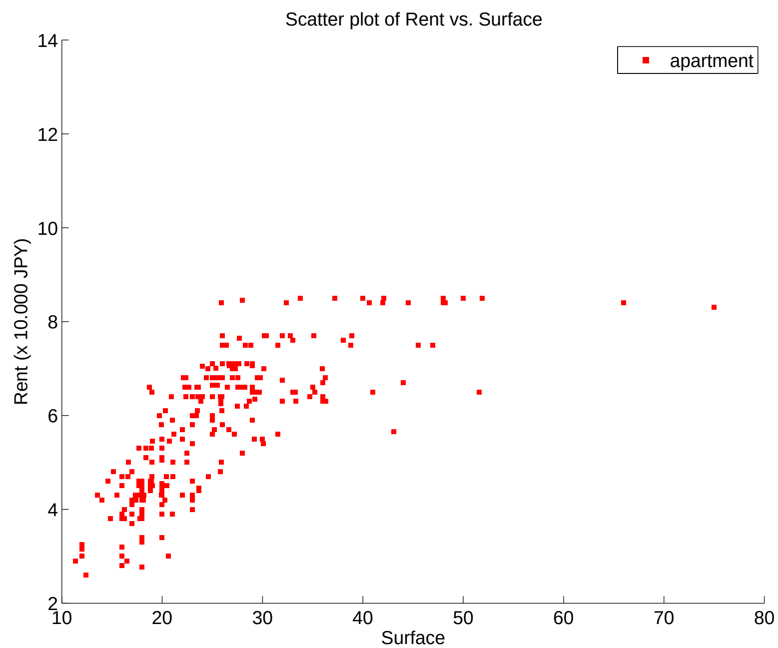
Considering again rents vs the rest

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



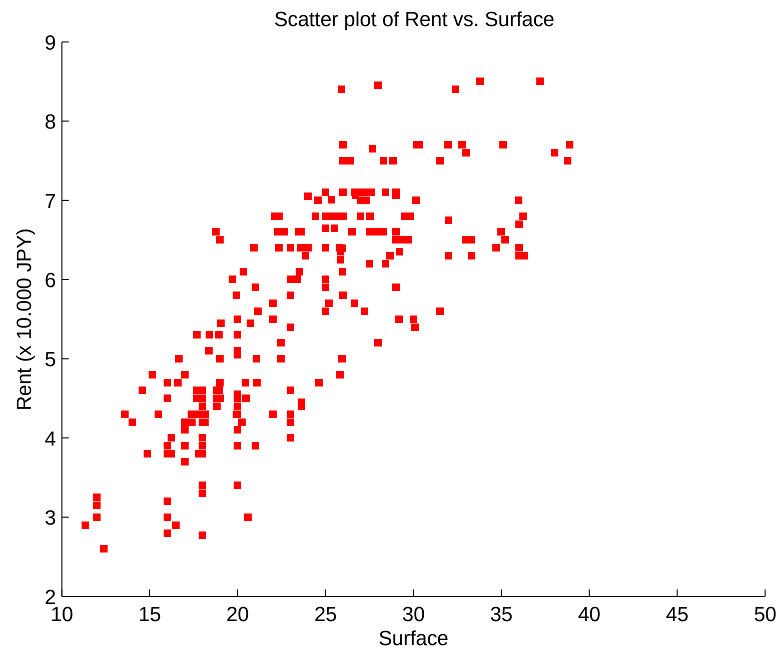

```
>> (X*X')\(X*Y')
ans =
    0.000141678721821
    0.004226687659299
   -0.012599982792209
    5.611128285287092
```

What went wrong?



rent= 0.00014 age + 0.00422 surf -0.0125 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 β such that $\|\beta\|_0 = \operatorname{card} \beta_i \neq 0$ is small.