Exemplo

Dados de anúncios de apartamentos em um bairro de Belo Horizonte (bairro Sion):

// Area (em metros quadrados)

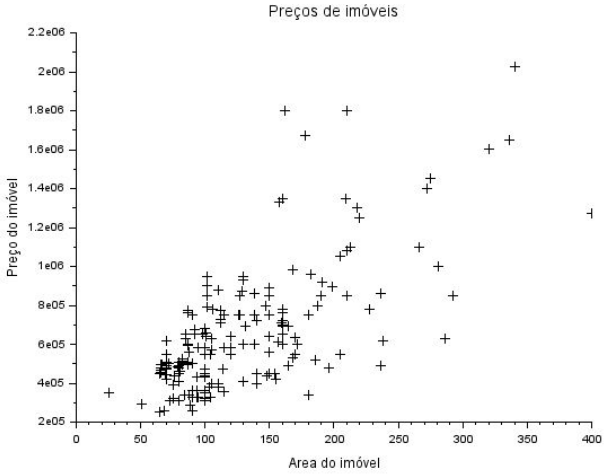
// Numero de Quartos

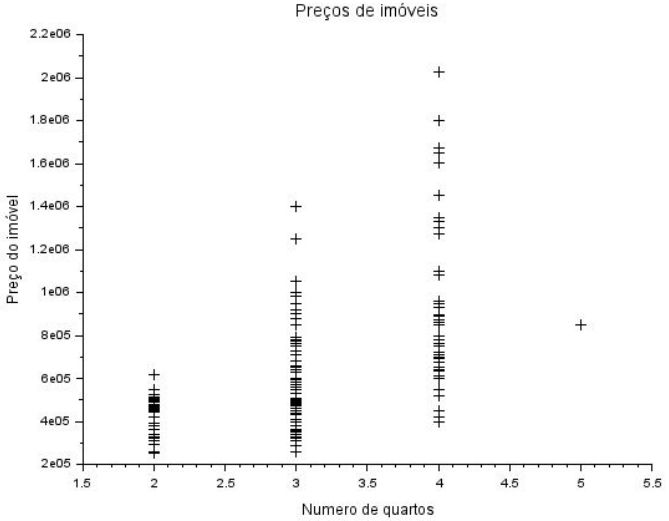
// Numero de Suites

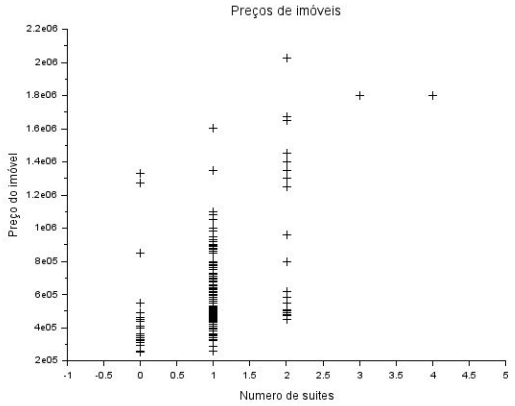
// Numero de Vagas de garagem

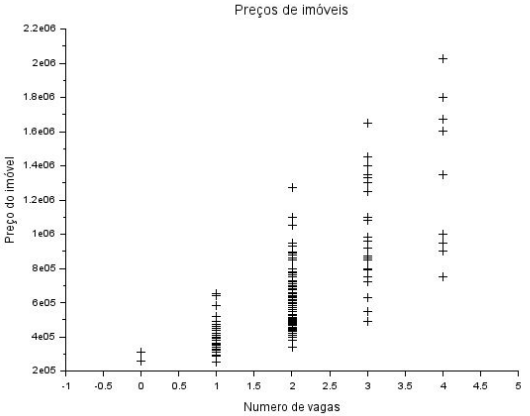
// Preço (em reais)

Regressão linear de preço com as demais features


Preços de imóveis
Preço do imóvel
Area do imóvel


Preços de imóveis
Preço do imóvel
Numero de quartos


Preços de imóveis
Preço do imóvel
Numero de suites


Preços de imóveis
Preço do imóvel
Numero de vagas

Interpretação dos coeficientes no modelo de regressão linear

$$E\left(\text{Preço do apto em reais} \mid \underset{\sim}{x}\right) = \beta_0 + \beta_1(\text{área em } m^2) + \beta_2(\text{nº quartos}) + \beta_3(\text{nº suítes}) + \beta_4(\text{nº de vagas de garagem})$$

$$\approx \hat{\beta}_0 + \hat{\beta}_1(\text{área}) + \hat{\beta}_2(\text{nº quartos}) + \hat{\beta}_3(\text{suítes}) + \beta_4(\text{vag}$$

$$= (-269382.13) + 1915.90 * \text{área} + 59637.00 * \text{quartos} + 11174383 * \text{suítes} + 19140 4.03 * \text{vagas}$$

-269382.13 + 1915.90*area + 59637.00*(quartos) + 111743.83*(suites) + 191404.13*(vagas)

Pegue um perfil $\underset{\sim}{x} = (1, \text{área}, \text{quartos}, \text{suítes}, \text{vagas})$ arbitrário

$= (1, x_1, x_2, x_3, x_4)$

Se o nº de quartos (e só o nº de quartos) passar de seu valor x_2 para $x_2 + 1$ teremos o perfil $\underset{\sim}{x}^* = (1, x_1, x_2 + 1, x_3, x_4)$

Compare $\mathbb{E}(\underset{Y}{\text{preço}} | \underset{\sim}{x})$ e $\mathbb{E}(\underset{Y}{\text{preço}} | \underset{\sim}{x}^*)$

$$\mathbb{E}\left(\underset{Y}{\text{preço}} \,\middle|\, \underset{\sim}{X}\right) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \beta_4 X_4$$

$$\mathbb{E}\left(\underset{Y^*}{\text{preço}} \,\middle|\, \underset{\sim}{X^*}\right) = \beta_0 + \beta_1 X_1 + \beta_2 (X_2 + 1) + \beta_3 X_3 + \beta_4 X_4$$

$$= \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \beta_4 X_4 + \boxed{\beta_2}$$

$$= \mathbb{E}\left(\underset{Y}{\text{preço}} \,\middle|\, \underset{\sim}{X}\right) + \boxed{\beta_2}$$

Assim, β_2 é o acréscimo no preço esperado quando X_2 passa para $X_2 + 1$ (uma unidade). Este é o efeito médio em Y de aumentar X_2 em 1.

Ajuste OLS

Predição de preços baseados no modelo de regressão linear é:

*-269382.13 + 1915.90*area + 59637.00*(quartos) + 111743.83*(suites) + 191404.13*(vagas)*

Uma vaga adicional de garagem aumenta o preço em 19 mil reais em média

Uma suíte adicional aumenta o preço em aproximadamente 11 mil reais.

Um metro quadrado adicional aumenta o preço em 1900 reais, em média

Pontos relevantes:

a) O efeito em $\mathbb{E}(Y|\underset{\sim}{x})$ da variável x_2 é β_2
(isto é, β_2 é o impacto médio de alterar x_2 em uma unidade adicional)

b) Este efeito é o mesmo para todo valor inicial de x_2 e também dos demais valores das outras features.

O efeito β_2 em $\mathbb{E}(Y|\underset{\sim}{x})$ não depende dos valores das features em $\underset{\sim}{x}$

(dos valores)

O efeito de aumentar 1 quarto é o mesmo de passarmos de 1 quarto para 2 ou se passarmos de 2 para 3

O efeito β_2 é o mesmo se o apto tem 100 m^2 ou se ele tem 200 m^2

- Isto torna o modelo de regressão linear muito atrativo para tomada de decisões:
  - ele fornece uma explicação de como cada feature afeta a resposta.
  - ele "explica o mecanismo"
  - fornece indicações de como alterar o sistema para obter resultados em Y.
    { É melhor aumentar uma vaga de garagem ou aumentar nº de suítes?
- Ao invés de apenas predizer Y "cegamente", passamos a entender como Y é formado e como pode ser alterado

A stochastic view for ML

Renato Assuncao

The need for a stochastic view

- We studied the linear regression model based on the least squares minimization of the sum of the (squared) residuals

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

- We can not go much further to understand the properties of this method unless we introduce a stochastic (probabilistic) view of the data.
- We assume that there is a probabilistic mechanism generating the data, possibly an infinite amount of them.
- We are allowed to observe a small portion portion of these data, the empirical sample with n training examples (possibly another portion for later testing).

Stochastic model

- The observed data are $(y_1, \mathbf{x}_1), (y_2, \mathbf{x}_2), \ldots, (y_n, \mathbf{x}_n)$

- The feature vector is composed of p variables: $\mathbf{x}_i = (x_{i1}, \ldots, x_{ip})$
- Organize the data in a matrix

$$\begin{bmatrix} y_1 & x_{11} & x_{12} & \cdots & x_{1p} \\ y_2 & x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ y_i & x_{i1} & x_{i2} & \cdots & x_{ip} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ y_n & x_{n1} & x_{n2} & \cdots & x_{np} \end{bmatrix}$$

i.i.d. sample

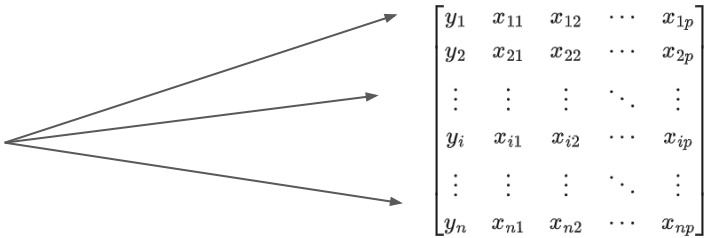
- Most ML models assume that the data is an i.i.d. sample from a random vector $(Y, \mathbf{X}) = (Y, X_1, X_2, \ldots, X_p)$

- This random vector has a generic joint distribution with density

$$f(y, \mathbf{x}) = f(y, x_1, x_2, \ldots, x_p)$$

- Rows in data matrix are independent instances or realizations of $(Y, \mathbf{X})$

i.i.d.



Generative models

- Generative models aim to model the joint distribution $f(y, \mathbf{x})$
- Examples of classical generative models:
  - Bayesian networks
  - Markov random fields
  - Naive Bayes classifier
  - Gaussian mixture models
  - Hidden Markov models
  - Linear discriminant analysis
- This is in contract with discriminative models

Discriminative models

- The joint density can always be decomposed as:

$$\underbrace{f(y, \mathbf{x})}_{\text{joint}} = \underbrace{f(y|\mathbf{x})}_{\text{conditional}} \underbrace{f(\mathbf{x})}_{\text{marginal}}$$

- Discriminative models focus on the conditional, ignoring the marginal
- Rather than modelling many variables at once, we focus on a model for a SINGLE random variable Y.
-

Discriminative models

$$\underbrace{f(y, \mathbf{x})}_{\text{joint}} = \underbrace{f(y|\mathbf{x})}_{\text{conditional}} \underbrace{f(\mathbf{x})}_{\text{marginal}}$$

- Na verdade, the notation $f(y|\mathbf{x})$ hids a very large set of distinct distributions: we have one distribution for each specific $\mathbf{x}$

different distributions

$$f(y^{\text{price}}|x_1^{\text{area}} = 100m^2, x_2^{\text{rooms}} = 2, x_3^{\text{restroom}} = 1, \ldots)$$

$$f(y^{\text{price}}|x_1^{\text{area}} = 150m^2, x_2^{\text{rooms}} = 2, x_3^{\text{restroom}} = 2, \ldots)$$

- This means that if $\mathbf{x}_i \neq \mathbf{x}_j$ then $(Y_i|\mathbf{x}_i)$ and $(Y_j|\mathbf{x}_j)$ are NOT identically distributed
- They are still independent, but not i.d.

Discriminative models

- Porque alguns aptos tem preços altos e outros possuem preços baixos?

- Vamos explicar como esta variação ocorre quebrando suas causas em
- dois componentes:
  - Causas determinadas pelos atributos ou features
  - Outras causas não medidas ou desconhecidas.
-
- Além disso, vamos também decompor a variável aleatória (Y|x) como a soma de sua esperança e do desvio em relação em a esta esperança

Decomposição de uma v.a.

- Seja Y uma v.a. qualquer
- Temos o valor numérico E(Y)
- O que e' E(Y)?
  - Uma v.a.?
  - Um valor numérico fixo determinado por f(y)?

Decomposição de uma v.a.

- Seja Y uma v.a. qualquer
- Temos o valor numérico E(Y)
- O que e' E(Y)?
  - Uma v.a.?
  - Um valor numérico fixo determinado por f(y)?
-
- Por exemplo,
  - Y tem distribuição Gaussiana e E(Y) = 15.3
  - Y tem distribuição Bernoulli (binária) com valores Y=1 ou 0 e temos E(Y) = P(Y=1) = 0.81

Decomposição de uma v.a.

- Seja Y uma v.a. qualquer
- Temos o valor numérico E(Y)
- Seja $\varepsilon = Y - \mathbb{E}(Y)$

- O que e' $\mathcal{E}$?
  - Uma v.a.?
  - Uma constante?
  - Uma função matemática?

Decomposição de uma v.a.

- Se $\varepsilon = Y - \mathbb{E}(Y)$ então podemos SEMPRE escrever

$$\underbrace{Y}_{\text{random}} = \underbrace{\mathbb{E}(Y)}_{\text{not random}} + \underbrace{\varepsilon}_{\text{random}} = \mu + \varepsilon$$

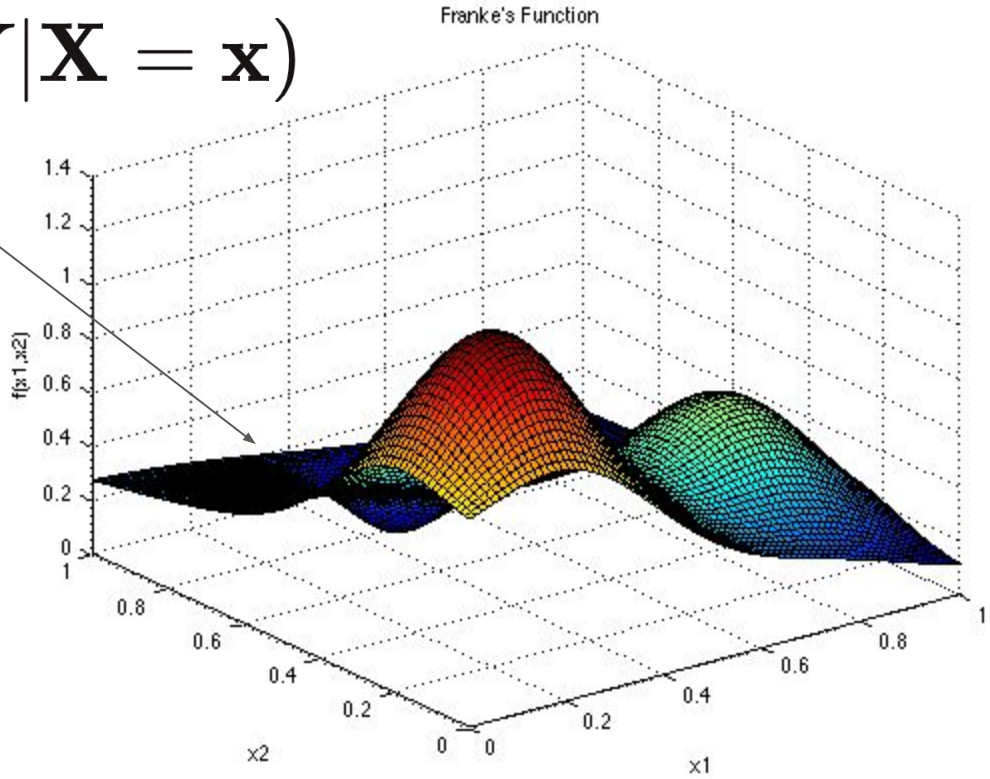
- No caso da distribuição condicional $(Y|\mathbf{X} = \mathbf{x})$, temos

$$\underbrace{(Y|\mathbf{X} = \mathbf{x})}_{\text{random}} = \underbrace{\mathbb{E}(Y|\mathbf{X} = \mathbf{x})}_{\text{not random}} + \underbrace{\varepsilon}_{\text{random}} = \mu(\mathbf{x}) + \underbrace{\varepsilon}_{\text{random}}$$

- Sabemos que $\mu(\mathbf{x}) = \mathbf{E}(Y|\mathbf{X} = \mathbf{x})$ ‘e o melhor preditor da v.a. (Y|x) no sentido de minimizar o erro de predicao esperado (erro-ao-quadrado)

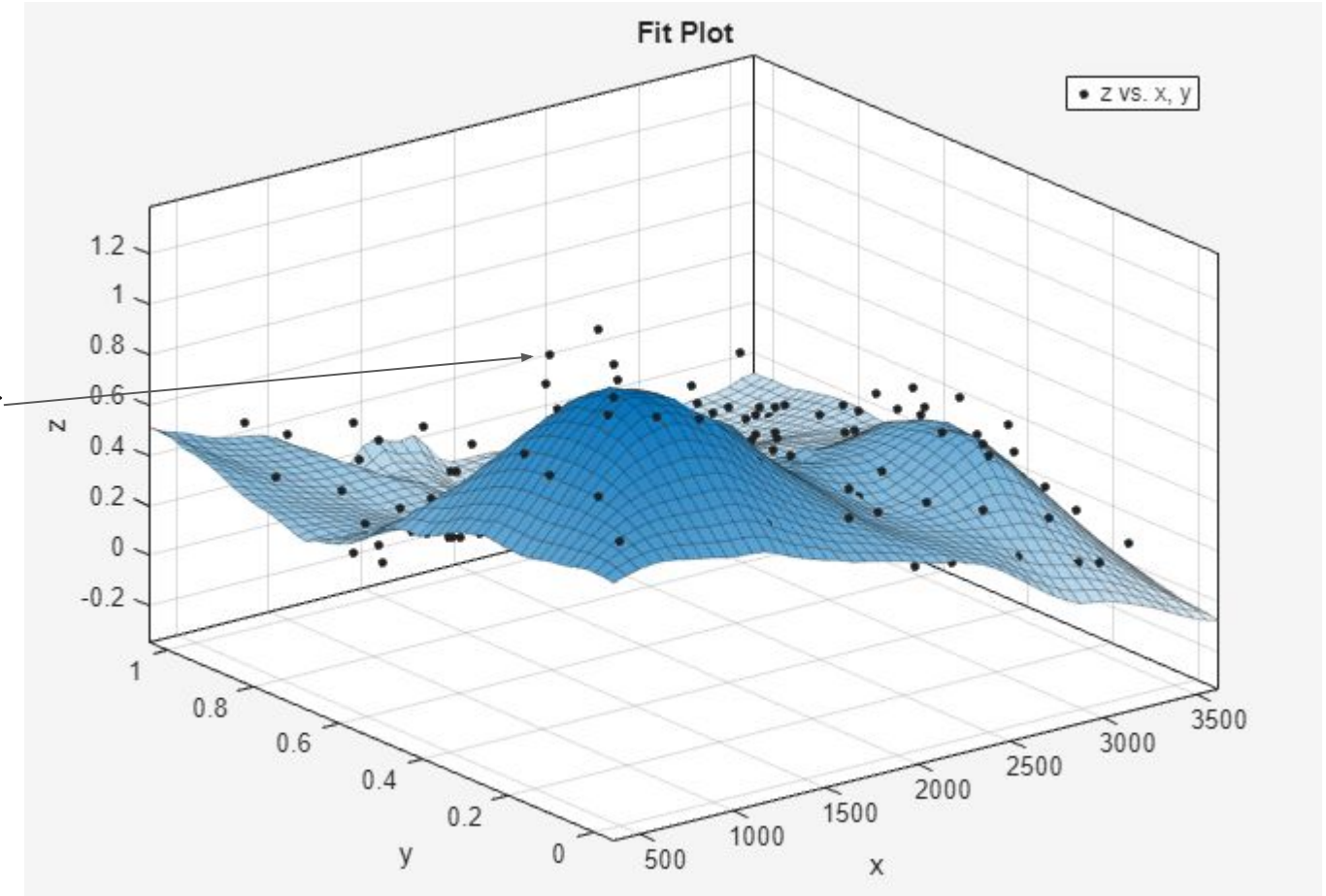
Superfície

$$\mu(\mathbf{x}) = \mathbf{E}(Y|\mathbf{X} = \mathbf{x})$$



Dados =
superfície mu(x)
+ erros

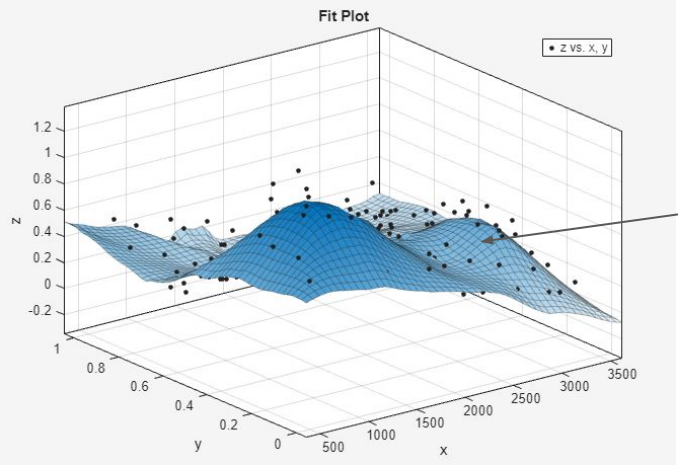
$$(Y|\mathbf{X} = \mathbf{x}) = \mu(\mathbf{x}) + \varepsilon$$



O modelo de regressão linear

- Precisamos dizer algo acerca dos dois componentes: $\mu(x)$ and ε
- Na regressão linear, aproximamos

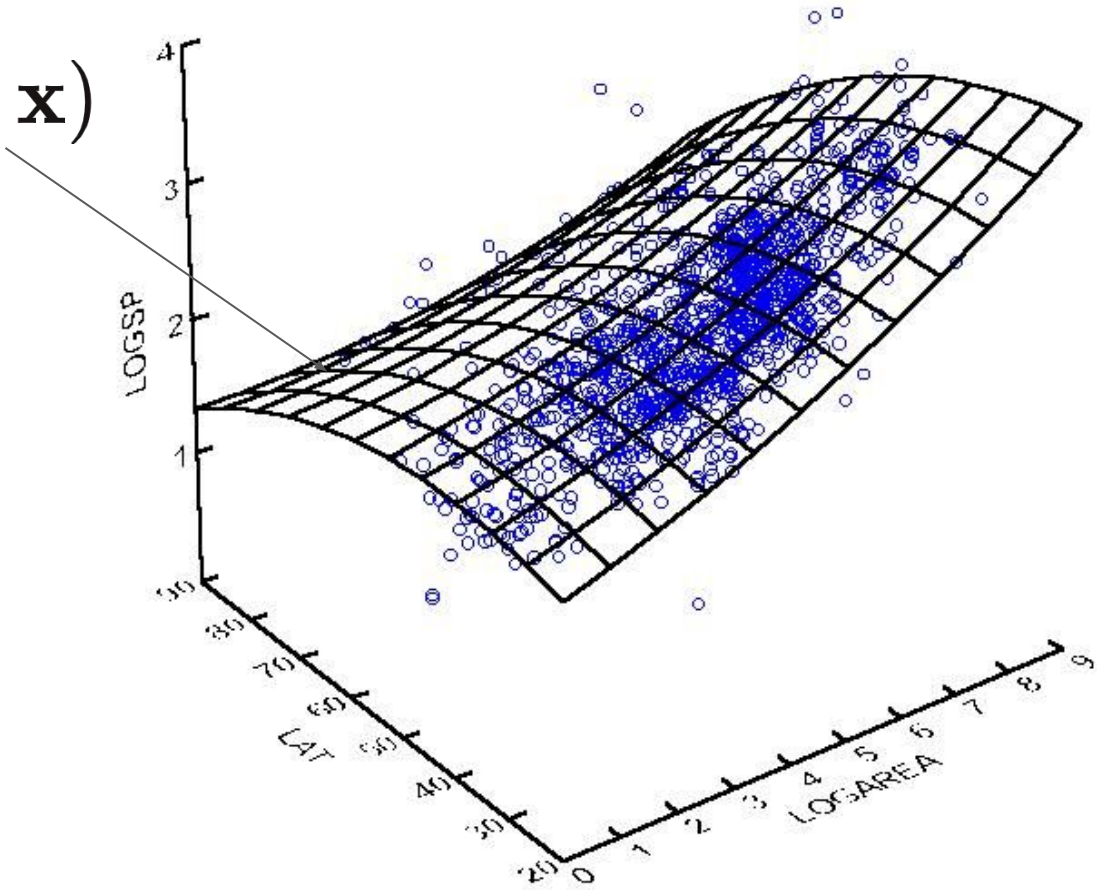
$$\mu(\mathbf{x}) \approx \beta_0 + \beta_1 x_1 + \ldots + \beta_p x_p$$

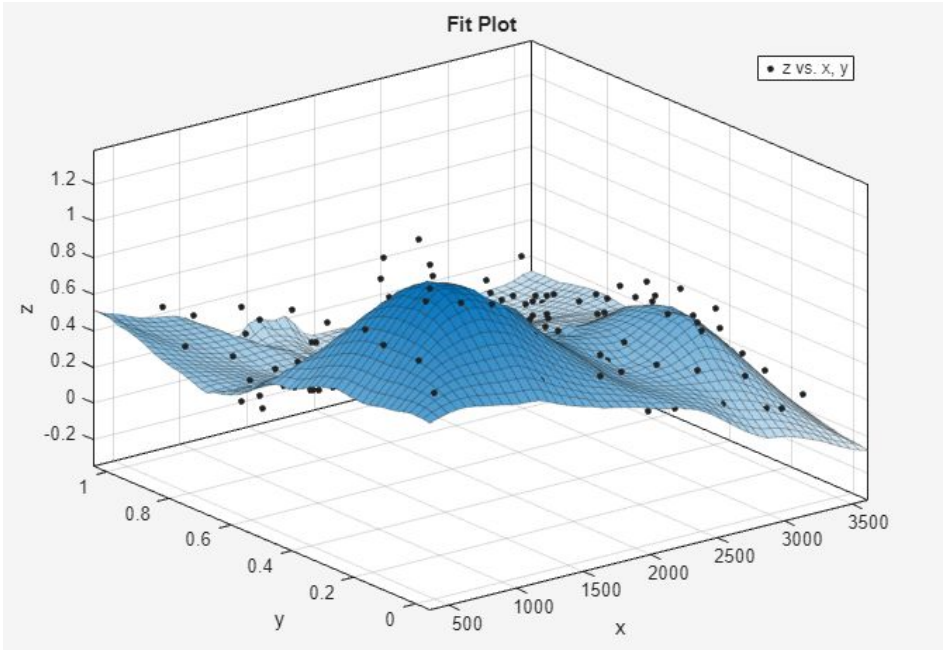


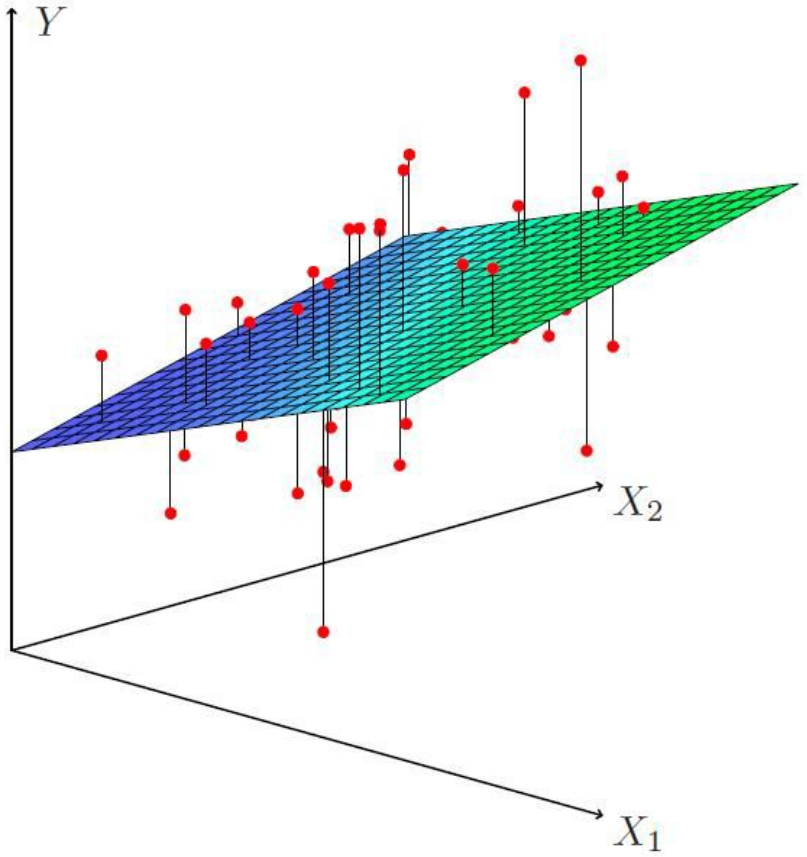
Com duas features, aproximar a superfície mu(x) por um plano

Superfície

$$\mu(\mathbf{x}) = \mathbf{E}(Y|\mathbf{X} = \mathbf{x})$$




Fit Plot
z vs. x, y
z
1.2
1
0.8
0.6
0.4
0.2
0
-0.2
y
1
0.8
0.6
0.4
0.2
0
x
500
1000
1500
2000
2500
3000
3500


Y
X_2
X_1

Aproximação linear

- Temos

$$\mu(\mathbf{x}) \approx \beta_0 + \beta_1 x_1 + \ldots + \beta_p x_p$$

$$= [1, x_1, x_2, \ldots, x_p] \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \ldots \\ \beta_p \end{bmatrix}$$

$$= \mathbf{x}'\beta$$

E a parte estocástica?

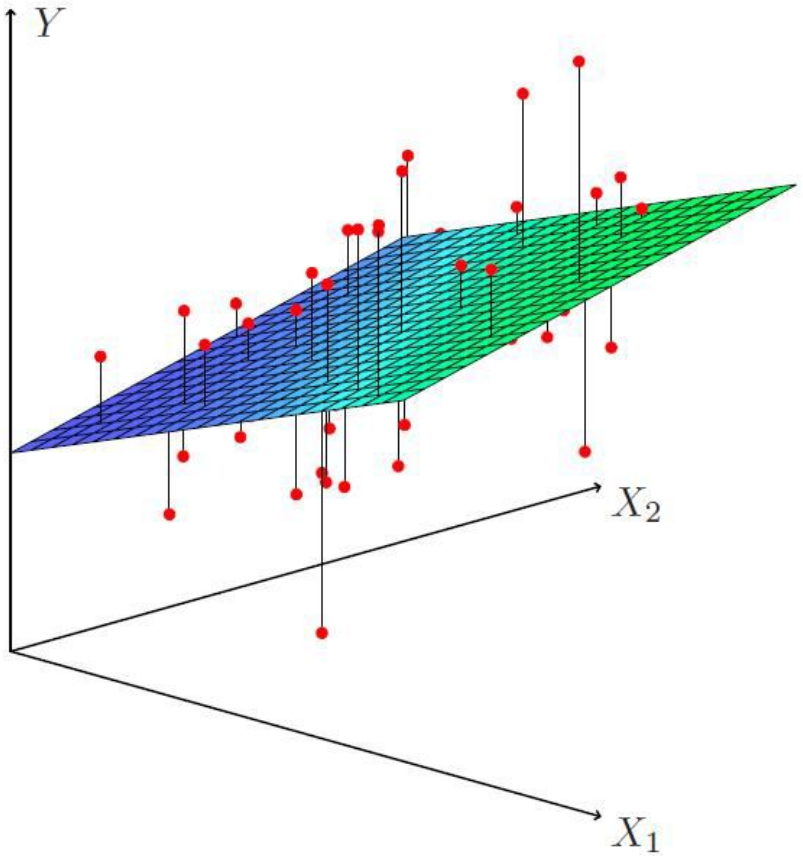
- Como fica a decomposição de (Y|X=x) quando fazemos esta aproximação linear?

$$
\begin{aligned}
(Y, \mathbf{X}) &= \mu(\mathbf{x}) + \varepsilon \\
&= \mathbf{x}'\beta + (\mu(\mathbf{x}) - \mathbf{x}'\beta) + \varepsilon \\
&= \mathbf{x}'\beta + \varepsilon^*
\end{aligned}
$$

Novo erro incorpora erros de má-especificação (linear) da média mu(x)

Vamos voltar a usar simplesmente ε

A parte estocástica



- Suponha que o modelo linear e' razoável: que um plano e' uma boa aproximação para mu(x)
- Os erros ε devem estar espalhados acima (positivos) e abaixo (negativos) do plano.
- Isto justifica supor que E(ε) = 0

- Mas e se E(ε) != 0 ???

O erro ε_i

- Considere o "erro" aletório

$$\varepsilon_i = Y_i - (\beta_0 + \beta_1 x_{i1} + \ldots + x_{i,p-1}\beta_p)$$

- Podemos SEMPRE assumir que $\mathbb{E}(\varepsilon_i) = 0$.
- Para ver isto, suponha que $\mathbb{E}(\varepsilon_i) = \alpha \neq 0$.
- Defina um novo erro aleatório ε_i^* da seguinte forma:

$$\begin{aligned} Y_i &= \beta_0 + \beta_1 x_{i1} + \ldots + x_{i,p-1}\beta_p + \varepsilon_i \\ &= \beta_0 + \beta_1 x_{i1} + \ldots + x_{i,p-1}\beta_p + \varepsilon_i - \alpha + \alpha \\ &= (\beta_0 - \alpha) + \beta_1 x_{i1} + \ldots + x_{i,p-1}\beta_p + (\varepsilon_i - \alpha) \\ &= \beta^* + \beta_1 x_{i1} + \ldots + x_{i,p-1}\beta_p + \varepsilon_i^* \end{aligned}$$

- O 1o. termo do lado direito é uma combinação linear dos atributos
- O novo erro tem

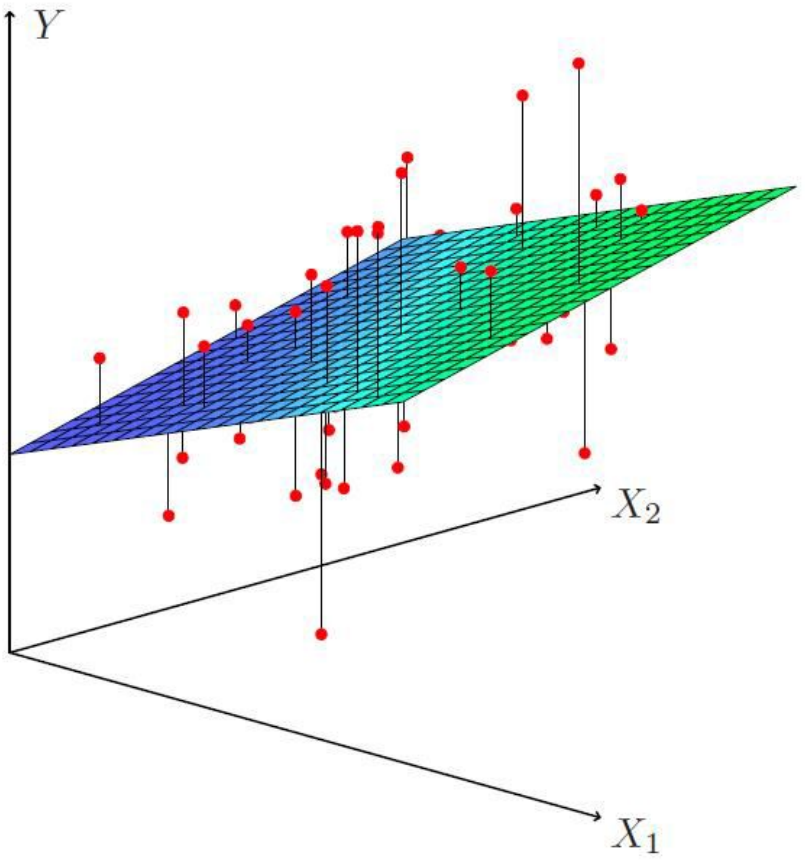
$$\mathbb{E}(\varepsilon_i^*) = \mathbb{E}(\varepsilon_i - \alpha) = \mathbb{E}(\varepsilon_i) - \alpha = \alpha - \alpha = 0$$

$$\mathbb{E}(\varepsilon) = 0 \qquad \mathbb{V}(\varepsilon) = ???$$

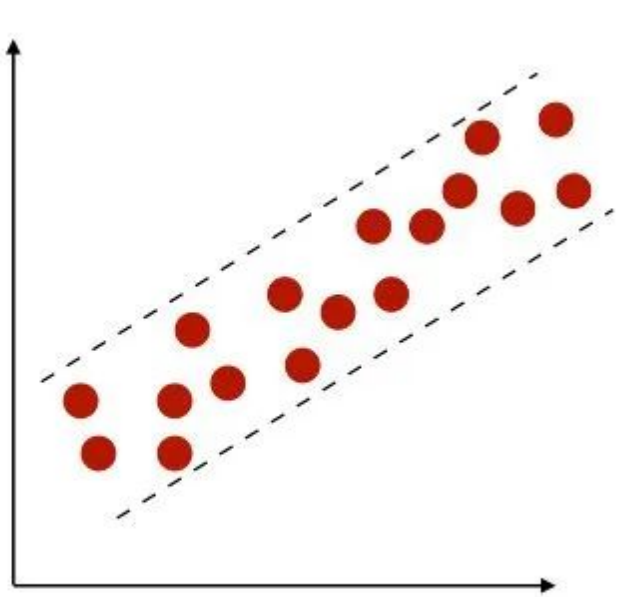
- Para a variância, $\mathbb{V}(\varepsilon) = \sigma^2$

- Variância não varia com $\mathbf{x}$

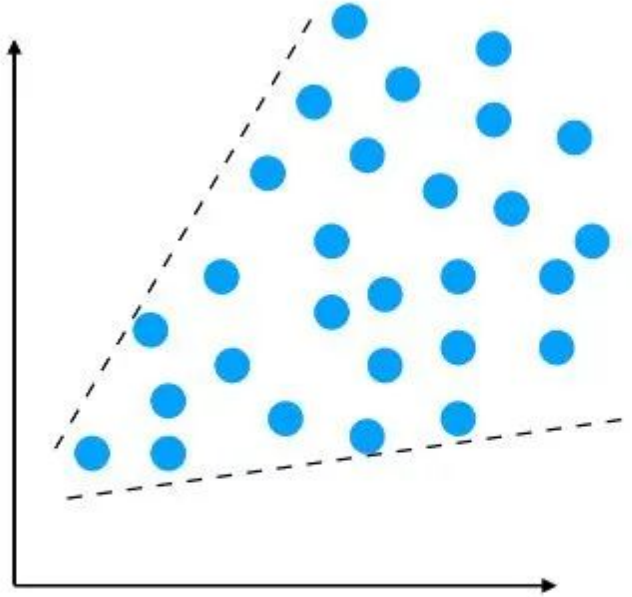
- Os tamanhos típicos dos desvios são os mesmos para todo $\mathbf{x}$
- Esta hipótese é chamada de homocedasticidade



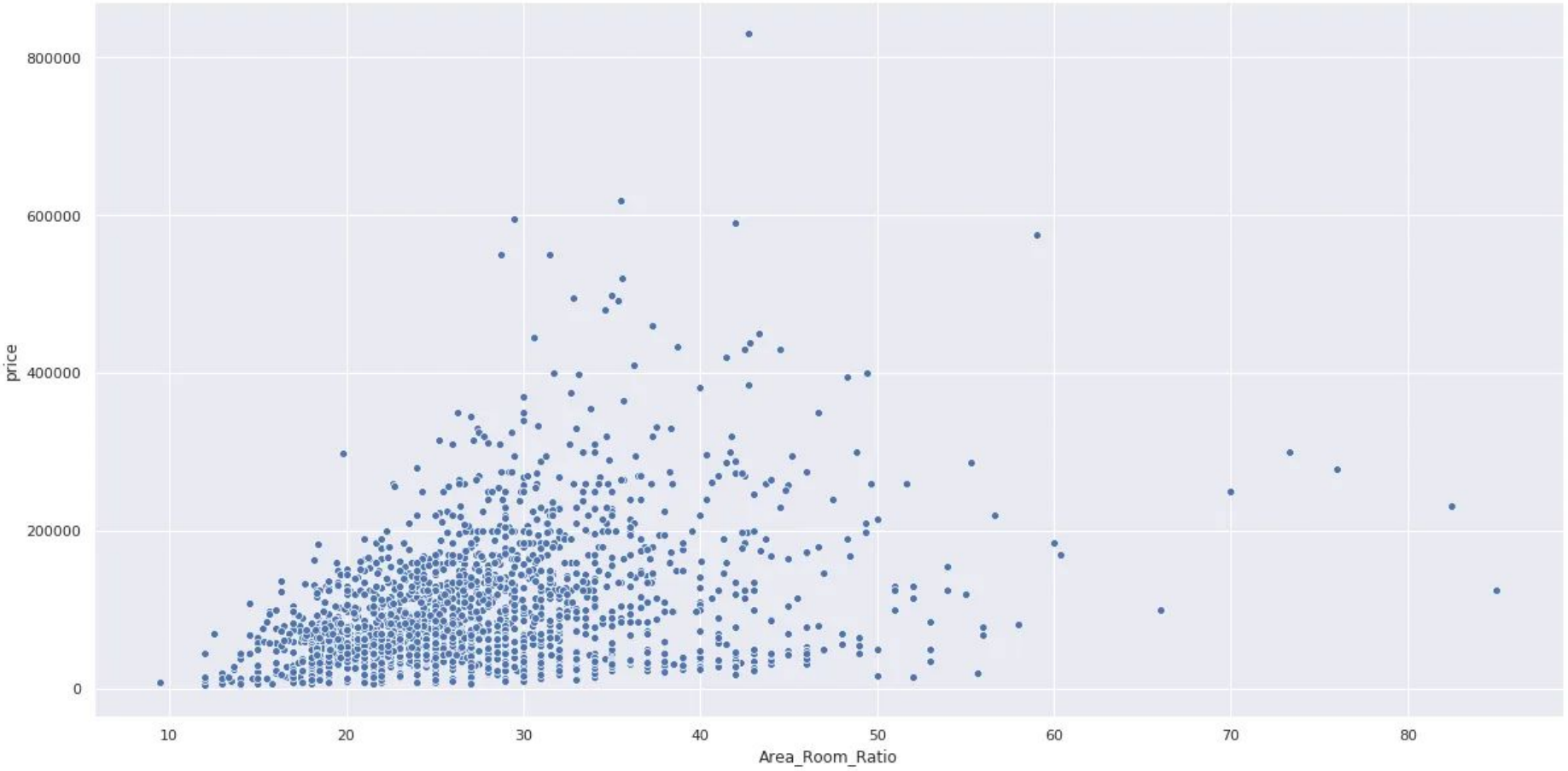
No caso de uma única feature x

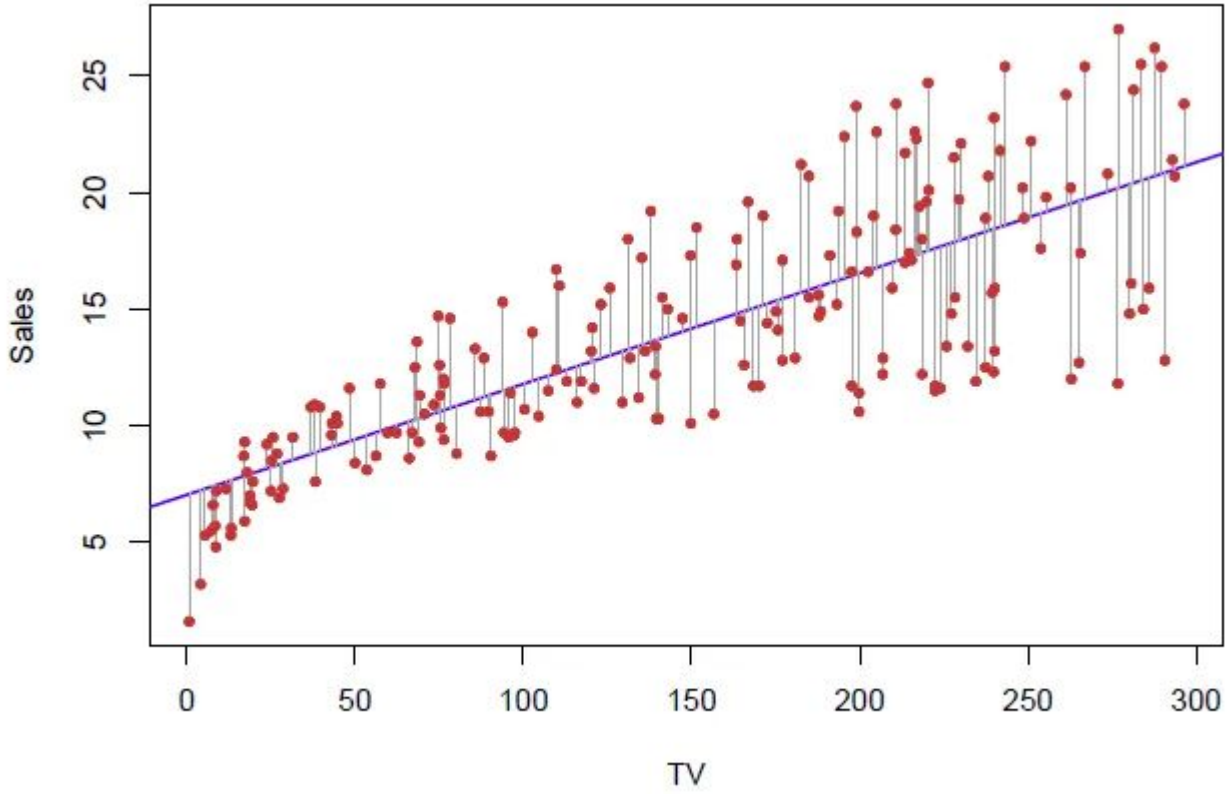


Homoscedasticity



Heteroscedasticity


800000
600000
400000
200000
0
price
10
20
30
40
50
60
70
80
Area_Room_Ratio


Sales
5
10
15
20
25
0
50
100
150
200
250
300
TV

Distribuição Gaussiana

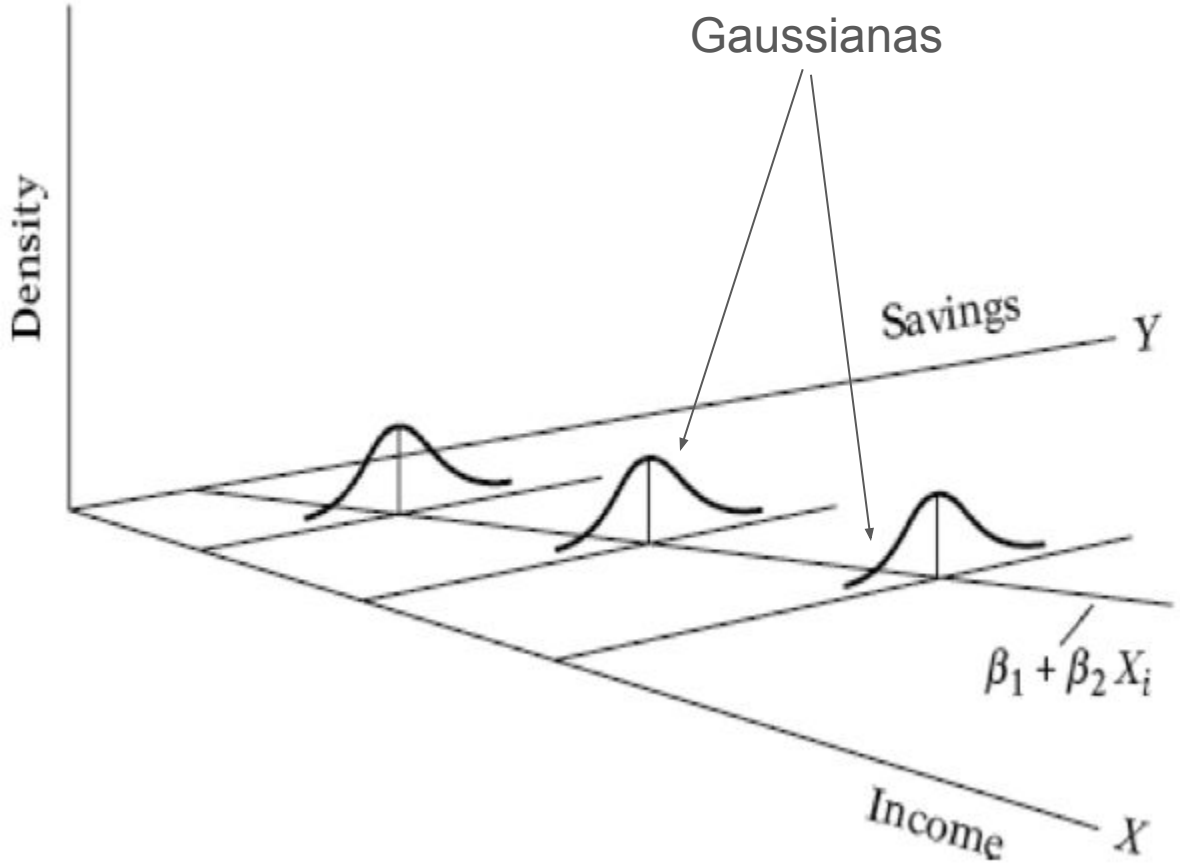
- Finalmente, além de $\mathbb{E}(\varepsilon) = 0$ e $\mathbb{V}(\varepsilon) = \sigma^2$, vamos agora falar da distribuição de probabilidade.
- Vamos assumir um erro gaussiano
- Assim, para uma observação com features $\mathbf{X}$, temos

$$(Y|\mathbf{X} = \mathbf{x}) = \mathbf{x}'\beta + \underbrace{\varepsilon}_{N(0,\sigma^2)}$$

- Isto implica que

$$(Y|\mathbf{X} = \mathbf{x}) \sim N(\mathbf{x}'\beta, \sigma^2)$$

- Além disso, as diferentes observações são v.a.'s independentes


Gaussianas
Density
Savings
Y
$\beta_1 + \beta_2 X_i$
Income
X

Versão estocástica matricial

$$\underset{\mu\times 1}{\underset{\sim}{Y}} = \begin{pmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_m \end{pmatrix}, \text{ vetor aleatório } m\text{-dimensional}$$

$$\underset{\mu\times 1}{\underset{\sim}{\mu}} = \begin{pmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_m \end{pmatrix} = \begin{pmatrix} \underset{\sim}{x}_1' \cdot \underset{\sim}{\beta} \\ \underset{\sim}{x}_2' \cdot \underset{\sim}{\beta} \\ \vdots \\ \underset{\sim}{x}_m' \cdot \underset{\sim}{\beta} \end{pmatrix} = \begin{bmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1p} \\ 1 & x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_{m1} & x_{m2} & \cdots & x_{mp} \end{bmatrix} \cdot \underset{(p+1)\times 1}{\underset{\sim}{\beta}}, \quad \underset{\sim}{\beta} = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_p \end{pmatrix}$$

$\mu \times (p+1)$

$$\underset{\mu\times 1}{\underset{\sim}{\mu}}(X) = \underset{\mu\times(p+1)}{X} \cdot \underset{(p+1)\times 1}{\underset{\sim}{\beta}}$$

$$\left(\underset{\sim}{Y} \mid X\right) \sim N_n\left(\underset{\sim}{\mu}(X),\ \sigma^2 I_n\right) \qquad (2)$$

$$\left(\underset{\sim}{Y} \mid X\right) \sim N_n\left(\begin{bmatrix} 1 & x_{11} & \cdots & x_{1p} \\ \vdots & \vdots & & \vdots \\ 1 & x_{n1} & \cdots & x_{np} \end{bmatrix} \cdot \underset{\sim}{\beta},\ \sigma^2 \begin{bmatrix} 1 & & 0 \\ & \ddots & \\ 0 & & 1 \end{bmatrix}\right)$$

$$\mathbb{E}\left(\underset{\sim}{Y} \mid X\right) = \underbrace{\underset{n\times(p+1)}{X} \cdot \underset{(p+1)\times 1}{\beta}}_{n\times 1} \qquad \mathbb{V}\mathrm{ar}\left(\underset{\sim}{Y} \mid X\right) = \sigma^2 I_n$$

Propriedades de uma gaussiana (ver FECD-A) ③

(A) Seja $\underset{\sim}{Y} \sim N_m\left(\underset{m\times 1}{\underset{\sim}{\mu}}, \underset{m\times m}{\Sigma}\right)$ e $\underset{k\times m}{A}$ uma matriz de constantes

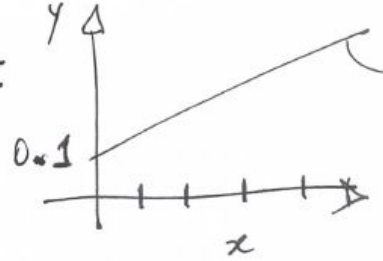
Então:

O vetor aleatório $\underbrace{\underset{k\times m}{A}\ \underset{m\times 1}{\underset{\sim}{Y}}}_{k\times 1} \sim N_k\left(A.\underset{\sim}{\mu}, A\Sigma A^t\right)$

Propriedades do estimador de mínimos quadrados (4)

- Vamos assumir que o modelo de regressão linear é perfeito, que os dados observados realmente foram gerados de acordo com o modelo.
- Isto é, assuma que existe um vetor $\underset{\sim}{\beta}$ fixo mas desconhecido e que os dados $\underset{\sim}{Y}$ foram gerados de acordo com: $\underset{\sim}{Y} \sim N_n(X\underset{\sim}{\beta}, \sigma^2 I_n)$
- Note que a matriz X é mantida fixa (gera X primeiro e depois gera $\underset{\sim}{Y}|X$).

Exemplo



$\mu(x) = 0.1 + 0.7x$

4 valores de x escolhidos
Geram 8 valores repetindo cada x duas vezes

$$y_1 = 0.1 + 0.7*1 + N(0, 0.3^2)$$
$$y_2 = 0.1 + 0.7*1 + N(0, 0.3^2)$$
$$y_3 = 0.1 + 0.7*2 + N(0, 0.3^2)$$
$$y_4 = 0.1 + 0.7*2 + N(0, 0.3^2)$$
$$\vdots$$
$$y_7 = 0.1 + 0.7*4 + N(0, 0.3^2)$$
$$y_8 = 0.1 + 0.7*4 + N(0, 0.3^2)$$

$$\underset{8\times 1}{Y} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 2 \\ 1 & 2 \\ 1 & 3 \\ 1 & 3 \\ 1 & 4 \\ 1 & 4 \end{bmatrix} \cdot \begin{bmatrix} 0.1 \\ 0.7 \end{bmatrix} + \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_8 \end{bmatrix}$$

onde $\underset{\sim}{\varepsilon} \sim N_8\left(\underset{\sim}{0}, 0.3^2 I\right)$

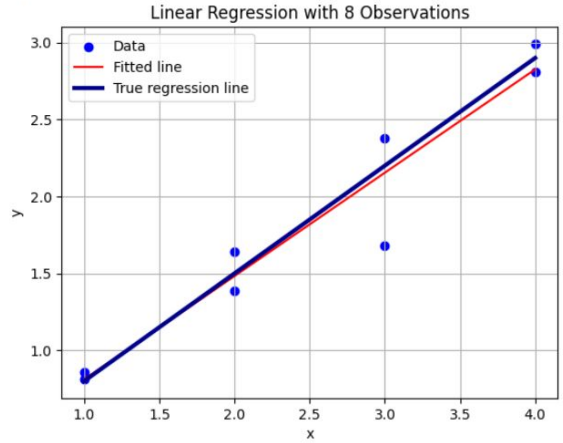
```
# Install statsmodels (if not already installed)
!pip install -q statsmodels

# Import required libraries
import numpy as np
import pandas as pd
import statsmodels.api as sm
import matplotlib.pyplot as plt

# Set seed for reproducibility
np.random.seed(432)
# Create predictor variable x
x = np.repeat([1, 2, 3, 4], 2)
# Generate noise ~ N(0, 0.3^2)
noise = np.random.normal(0, 0.3, size=8)
# Compute response y = 0.1 + 0.7 * x + noise
y = 0.1 + 0.7 * x + noise
# Create DataFrame
df = pd.DataFrame({'x': x, 'y': y})
# Fit linear regression model
X = sm.add_constant(df['x'])  # Add intercept
model = sm.OLS(df['y'], X).fit()
# Print model summary
print(model.summary())

# Optional: Plot the data and fitted line
plt.scatter(df['x'], df['y'], color='blue', label='Data')
plt.plot(df['x'], model.predict(X), color='red', label='Fitted line')
mu_true = 0.1 + 0.7 * x
plt.plot(x, mu_true, color='darkblue', linewidth=3, label='True regression line')
plt.xlabel('x')
plt.ylabel('y')
plt.title('Linear Regression with 8 Observations')
plt.legend()
plt.grid(True)
plt.show()
```

OLS Regression Results

Dep. Variable:	y	R-squared:	0.930
Model:	OLS	Adj. R-squared:	0.918
Method:	Least Squares	F-statistic:	79.36
Date:	Mon, 31 Mar 2025	Prob (F-statistic):	0.000111
Time:	13:10:22	Log-Likelihood:	1.2695
No. Observations:	8	AIC:	1.461
Df Residuals:	6	BIC:	1.620
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	0.1395	0.206	0.676	0.524	-0.366	0.645
x	0.6716	0.075	8.909	0.000	0.487	0.856

Omnibus:	7.386	Durbin-Watson:	2.389
Prob(Omnibus):	0.025	Jarque-Bera (JB):	2.297
Skew:	-1.252	Prob(JB):	0.317
Kurtosis:	3.786	Cond. No.	7.47



With these 8 data points, obtain the LS (least squares) estimator:

$$\text{true } \underset{\sim}{\beta} = \begin{pmatrix} \beta_0 \\ \beta_1 \end{pmatrix} = \begin{pmatrix} 0.1 \\ 0.7 \end{pmatrix} \qquad \text{Estimated } \hat{\underset{\sim}{\beta}} = \begin{pmatrix} \hat{\beta}_0 \\ \hat{\beta}_1 \end{pmatrix} = \begin{pmatrix} 0.14 \\ 0.67 \end{pmatrix}$$

$$\text{Estimated } \hat{\underset{\sim}{\beta}} \neq \underset{\sim}{\beta} \text{ true} \qquad \underbrace{\hat{\underset{\sim}{\beta}} - \underset{\sim}{\beta}}_{\text{estimation error}} = \begin{pmatrix} 0.14 \\ 0.67 \end{pmatrix} - \begin{pmatrix} 0.1 \\ 0.7 \end{pmatrix} = \begin{pmatrix} 0.04 \\ -0.03 \end{pmatrix}$$

- ~~What is this~~
- Simulate a second set of 8 observations following this linear regression model ~~(same seed)~~ (use a ~~seed~~ different)

With the second set of observations:

$$\text{True } \underset{\sim}{\beta} = \begin{pmatrix} \beta_0 \\ \beta_1 \end{pmatrix} = \begin{pmatrix} 0.1 \\ 0.7 \end{pmatrix} \qquad \underset{\sim}{\hat{\beta}} = \begin{pmatrix} \hat{\beta}_0 \\ \hat{\beta}_1 \end{pmatrix} = \begin{pmatrix} 0.06 \\ 0.73 \end{pmatrix}$$

$$\text{Estimation error } \underset{\sim}{\hat{\beta}} - \underset{\sim}{\beta} = \begin{pmatrix} 0.06 \\ 0.73 \end{pmatrix} - \begin{pmatrix} 0.1 \\ 0.7 \end{pmatrix} = \begin{pmatrix} -0.04 \\ 0.03 \end{pmatrix}$$

The new sample implied a <u>different</u> value for $\underset{\sim}{\hat{\beta}}$ and a different estimation error.

<u>The nature of $\underset{\sim}{\hat{\beta}}$ and $\underset{\sim}{\hat{\beta}} - \underset{\sim}{\beta}$</u>

What is $\underset{\sim}{\hat{\beta}}$?
- a fixed vector?
- a random vector?
- a mathematical function?

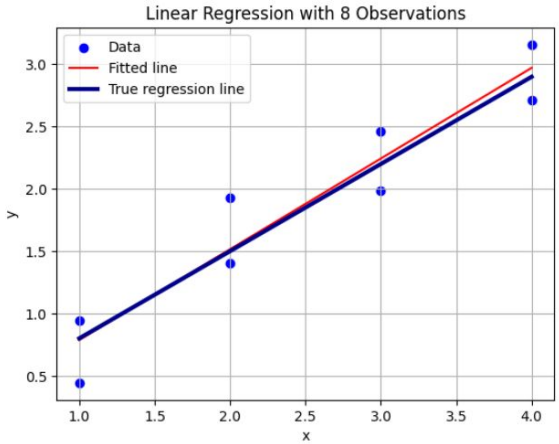
```
# A second set of observations follwoing the same linear regression model.

# New sedd
np.random.seed(1234)

# New noise ~ N(0, 0.3^2)
noise2 = np.random.normal(0, 0.3, size=8)

# New response
y2 = 0.1 + 0.7 * x + noise2

# Create DataFrame
df2 = pd.DataFrame({'x': x, 'y': y2})

# Fit linear regression model
X = sm.add_constant(df2['x'])  # Add intercept
model2 = sm.OLS(df2['y'], X).fit()

# Print model summary
print(model2.summary())

# Optional: Plot the data and fitted line
plt.scatter(df2['x'], df2['y'], color='blue', label='Data')
plt.plot(df2['x'], model2.predict(X), color='red', label='Fitted line')
plt.plot(x, mu_true, color='darkblue', linewidth=3, label='True regression line')
plt.xlabel('x')
plt.ylabel('y')
plt.title('Linear Regression with 8 Observations')
plt.legend()
plt.grid(True)
plt.show()
```

OLS Regression Results

Dep. Variable:	y	R-squared:	0.907
Model:	OLS	Adj. R-squared:	0.891
Method:	Least Squares	F-statistic:	58.17
Date:	Mon, 31 Mar 2025	Prob (F-statistic):	0.000265
Time:	13:53:45	Log-Likelihood:	-0.61906
No. Observations:	8	AIC:	5.238
Df Residuals:	6	BIC:	5.397
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	0.0593	0.261	0.227	0.828	-0.580	0.699
x	0.7281	0.095	7.627	0.000	0.495	0.962

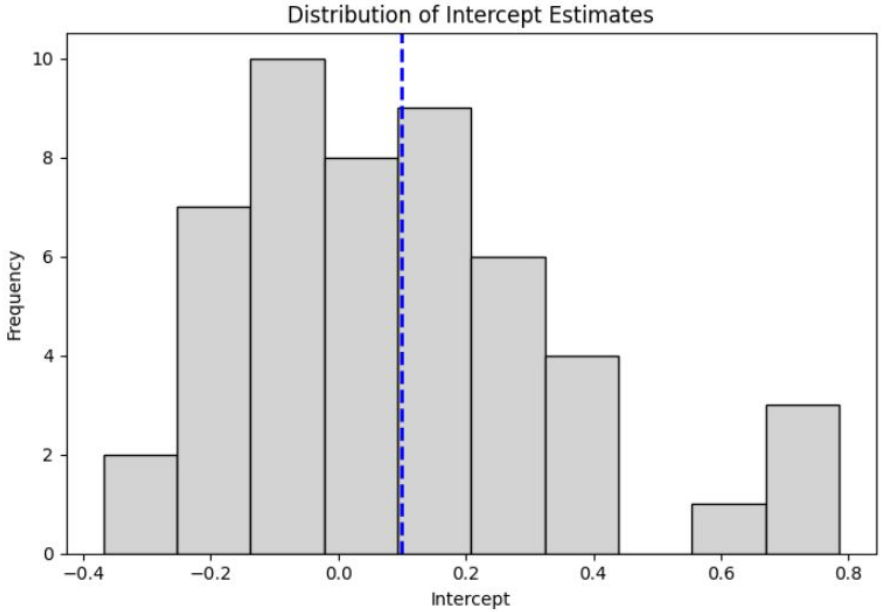
Omnibus:	1.813	Durbin-Watson:	2.848
Prob(Omnibus):	0.404	Jarque-Bera (JB):	0.732
Skew:	0.099	Prob(JB):	0.694
Kurtosis:	1.532	Cond. No.	7.47

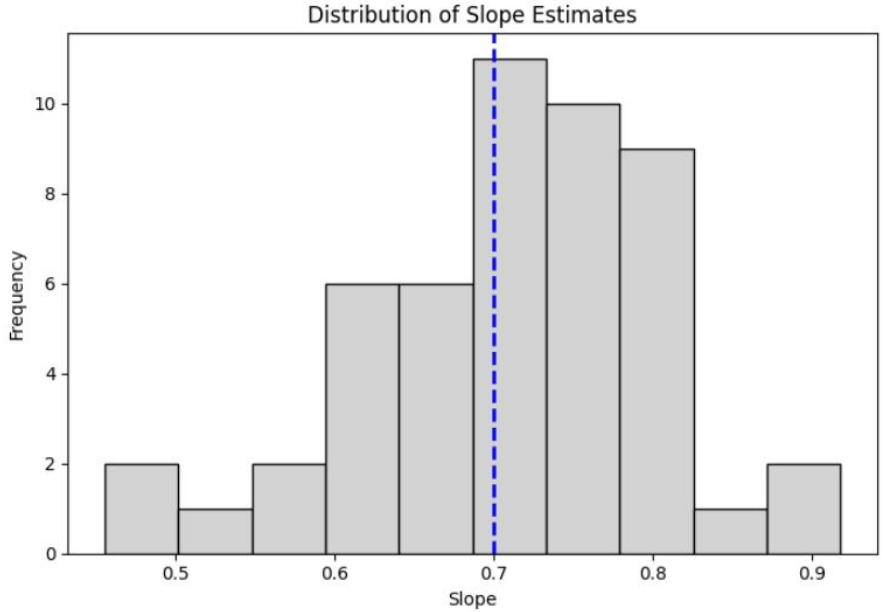


Learning the random nature of $\hat{\beta}$ by simulation (8)

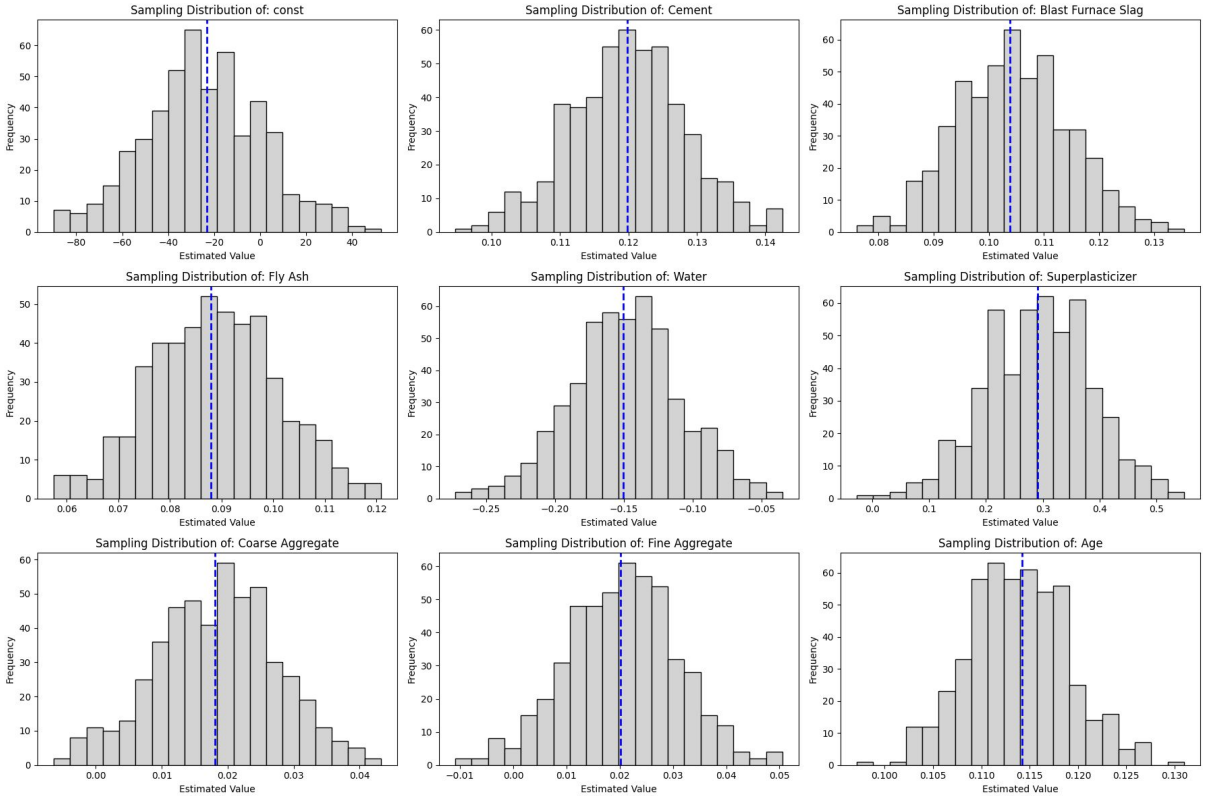
- Repeat the data generation (holding X fixed)
- for i in (n_sim):
  - generate noise and Y
  - fit OLS model
  - save estimates
- Plot results.


Monte Carlo Simulation: 50 Fitted Regression Lines
True regression line
3.0
2.5
2.0
1.5
1.0
0.5
y
1.0
1.5
2.0
2.5
3.0
3.5
4.0
x


Distribution of Intercept Estimates
Frequency
Intercept
0
2
4
6
8
10
−0.4
−0.2
0.0
0.2
0.4
0.6
0.8


Distribution of Slope Estimates
Frequency
Slope
0
2
4
6
8
10
0.5
0.6
0.7
0.8
0.9

Caso geral: vários preditores (Cement strength)



So, $\hat{\beta}$ is a random vector.

This conclusion is valid for the general linear regression model.

- Simulation with Concrete Compressive Strength Dataset
- Fit the linear regression and obtain $\hat{\beta}$.
- Take the residuals $\frac{1}{M}\sum_{i=1}^{M}(y_i - \hat{y}_i)^2 = MSE$
- Simulate new data $\underset{\sim}{Y}^*$ with $\underset{\sim}{Y}^* = X \cdot \hat{\beta} + \underset{\sim}{\varepsilon}$ with $\underset{\sim}{\varepsilon} \sim N_M\left(\underset{\sim}{0}, (MSE) \cdot I_M\right)$
- Fit the model to each new dataset.

OK, $\hat{\beta}$ is a random vector.

What else can be said about $\hat{\beta}$ assuming that the linear regression model is the true data generating mechanism?

Much can be said:

$$\underset{(p+1)\times 1}{\hat{\beta}} = \underbrace{(X'X)^{-1}X'}_{A}\underset{\sim}{Y} = A \cdot \underset{\sim}{Y}$$

$$A = \underbrace{(X'X)^{-1}}_{[(p+1)\times(p+1)]} \cdot \underbrace{X'}_{(p+1)\times n}$$

$$(p+1)\times n$$

Using the property of Multivariate Gaussian (11)

$$\hat{\underset{\sim}{\beta}} = A \cdot \underset{\sim}{Y} \quad \text{and} \quad \underset{\sim}{Y} \sim N_m\left(X\underset{\sim}{\beta}, \sigma^2 I_m\right)$$

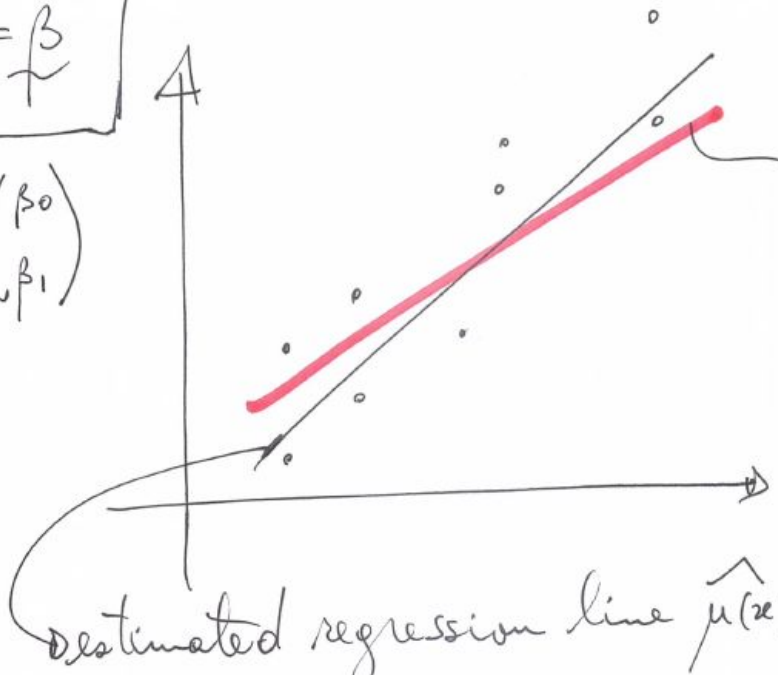
$$\Rightarrow \hat{\underset{\sim}{\beta}} \sim N_{p+1}\left(A.(X\underset{\sim}{\beta}), \sigma^2 A I_m A^t\right)$$

$$A(X\underset{\sim}{\beta}) = (X'X)^{-1}X'(X\underset{\sim}{\beta}) = (X'X)^{-1}(X'X)\cdot\underset{\sim}{\beta} = \underset{\sim}{\beta}$$

that is, $\hat{\underset{\sim}{\beta}}$ is random, Gaussian and $\mathbb{E}(\hat{\underset{\sim}{\beta}}) = \underset{\sim}{\beta}$

$$\mathbb{E}(\hat{\beta}) = \beta$$

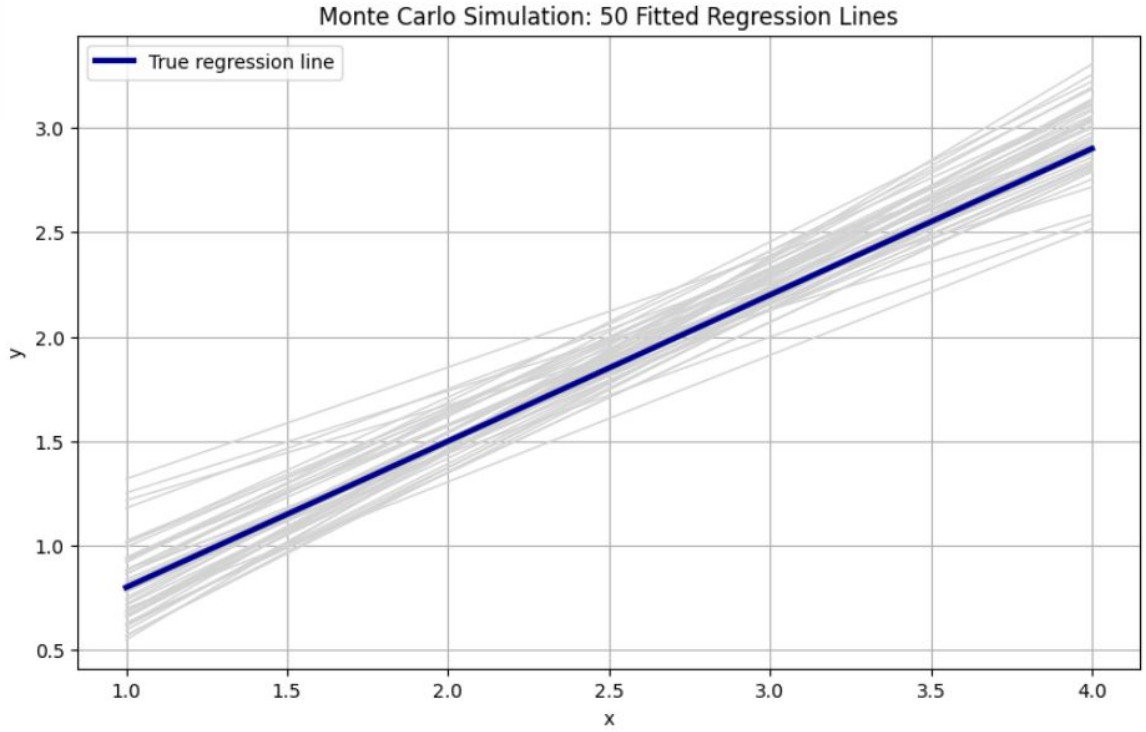
$$\mathbb{E}\begin{pmatrix}\hat{\beta}_0\\ \hat{\beta}_1\end{pmatrix} = \begin{pmatrix}\beta_0\\ \beta_1\end{pmatrix}$$



Sometimes $\hat{\beta}_1 > \beta_1$, sometimes $\hat{\beta}_1 < \beta_1$

However, on average, $\mathbb{E}(\hat{\beta}_1) = \beta_1$

Dizemos que $\hat{\beta}$ é não-viciado para estimar β quando $\mathbb{E}(\hat{\beta}) = \beta$



Covariance Matrix

$\hat{\beta}$ is vector aleatório. $\hat{\beta} = \begin{pmatrix} \hat{\beta}_0 \\ \hat{\beta}_1 \end{pmatrix}$

$$\mathbb{E}(\hat{\beta}_0) = \beta_0^{(\text{true})} \qquad \mathbb{E}(\hat{\beta}_1) = \beta_1$$

But, in any sample, $\hat{\beta}_0 \neq \beta_0$ and $\hat{\beta}_1 \neq \beta_1$

there will be estimation errors. What is the typical size of these errors? estimation

$$\mathbb{E}(\hat{\beta}_1 - \beta_1)^2 = \text{Var}(\hat{\beta}_1)$$

. Likewise, $\mathbb{E}(\hat{\beta}_0 - \beta_0)^2 = \text{Var}(\hat{\beta}_0)$

$$Var(\hat{\beta}) = Var\left(\begin{pmatrix}\hat{\beta}_0\\ \hat{\beta}_1\end{pmatrix}\right) = \begin{bmatrix} Var(\hat{\beta}_0) & Cov(\hat{\beta}_0, \hat{\beta}_1) \\ Cov(\hat{\beta}_0, \hat{\beta}_1) & Var(\hat{\beta}_1) \end{bmatrix}$$

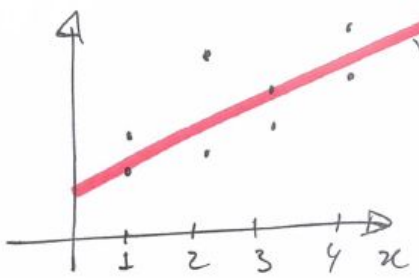
$= \sigma^2 (X'X)^{-1} X'$ (ver slide anterior com distribuição Gaussiana)

$= \sigma^2 A I_n A^t = \sigma^2 (X'X)^{-1} X (X'X)^{-1}$

$= \sigma^2 (X'X)^{-1}$

Isto é, $Var(\hat{\beta}) = \sigma^2 (X'X)^{-1}$

Exemplo



$$\mu(x) = 0.1 + 0.7x$$

$$\varepsilon \sim N(0, 0.3^2)$$

$$X = \begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 2 \\ 1 & 2 \\ 1 & 3 \\ 1 & 3 \\ 1 & 4 \\ 1 & 4 \end{bmatrix} \qquad X'X = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 2 & 2 & 3 & 3 & 4 & 4 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 2 \\ 1 & 2 \\ 1 & 3 \\ 1 & 3 \\ 1 & 4 \\ 1 & 4 \end{bmatrix} = \begin{bmatrix} 8 & 20 \\ 20 & 30 \end{bmatrix}$$

$$(X'X)^{-1} = \begin{pmatrix} 0.75 & -0.25 \\ -0.25 & 0.10 \end{pmatrix}$$

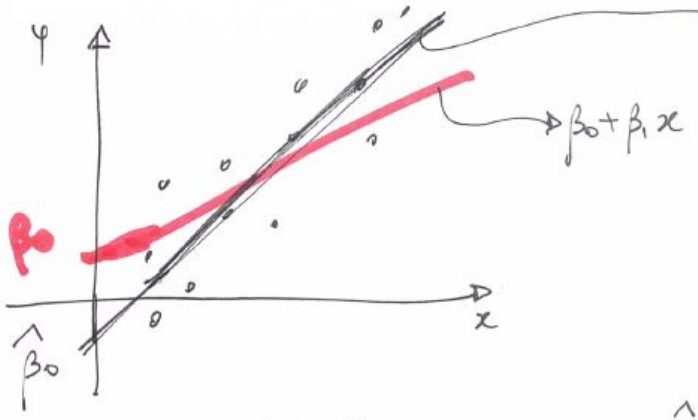
$$Cov\left(\hat{\beta}\right) = \sigma^2 (X'X)^{-1} = (0.3)^2 \begin{pmatrix} 0.75 & -0.25 \\ -0.25 & 0.10 \end{pmatrix} = \begin{pmatrix} Var(\hat{\beta}_0) & Cov(\hat{\beta}_0, \hat{\beta}_1) \\ Cov(\hat{\beta}_0, \hat{\beta}_1) & Var(\hat{\beta}_1) \end{pmatrix}$$

$$\oplus \quad \mathrm{Var}\begin{pmatrix}\hat{\beta}_0\\ \hat{\beta}_1\end{pmatrix} = (0.3)^2\begin{pmatrix}0.75 & -0.25\\ -0.25 & 0.10\end{pmatrix} = \begin{pmatrix}\mathrm{Var}(\hat{\beta}_0) & \mathrm{Cov}(\hat{\beta}_0,\hat{\beta}_1)\\ \mathrm{Cov}(\hat{\beta}_0,\hat{\beta}_1) & \mathrm{Var}(\hat{\beta}_1)\end{pmatrix} \quad (17)$$

$$\mathrm{Cov}(\hat{\beta}_0,\hat{\beta}_1) = \sqrt{\mathrm{Var}(\hat{\beta}_0)\,\mathrm{Var}(\hat{\beta}_1)} \cdot \left(\text{Correlation}\ (\hat{\beta}_0,\hat{\beta}_1)\right)$$

$$\underline{\text{Corr}}(\hat{\beta}_0,\hat{\beta}_1) = \frac{-0.25\,(0.3)^2}{\sqrt{0.75}\,\sqrt{0.10}\,(0.3)^2} = \underline{-0.91}$$

⊕ Why such large negative correlation between $\hat{\beta}_0$ and $\hat{\beta}_1$??



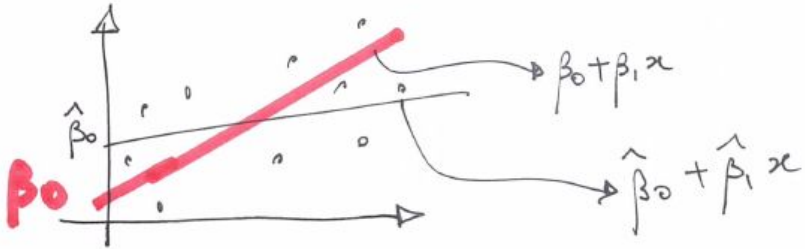
Neste dataset, tivemos

$\hat{\beta}_0 < \beta_0 = \mathbb{E}(\hat{\beta}_0)$

$\hat{\beta}_1 > \beta_1 = \mathbb{E}(\hat{\beta}_1)$

Existe a tendência: se $\hat{\beta}_1 > \beta_1$ ~~usualmente~~ é comum termos $\hat{\beta}_0 < \beta_0$

Por outro lado, se $\hat{\beta}_1 < \beta_1 \Rightarrow \hat{\beta}_0 > \beta_0$, usualmente



Kaggle Dataset

- Aim: To predict the compressive strength of concrete based on material composition.

🎯 Target Variable (Response Variable)

Feature Name	Description	Units	Typical Range
Compressive Strength	The maximum compressive stress the concrete can withstand.	MPa (MegaPascals)	2.33 - 82.6

- Number of Samples: 1,030 observations
- Number of Features: 8 predictors

```
X'X matrix (9x9) scaled by 10^7:
 [[ 0.    0.03  0.01  0.01  0.02  0.    0.1   0.08  0.  ]
 [ 0.03  9.27  1.88  1.3   5.24  0.19 28.08 22.21  1.38]
 [ 0.01  1.88  1.33  0.23  1.4   0.05  7.21  5.69  0.32]
 [ 0.01  1.3   0.23  0.72  0.98  0.05  5.43  4.36  0.19]
 [ 0.02  5.24  1.4   0.98  3.44  0.11 18.16 14.39  0.89]
 [ 0.    0.19  0.05  0.05  0.11  0.01  0.61  0.51  0.02]
 [ 0.1  28.08  7.21  5.43 18.16  0.61 98.12 77.41  4.57]
 [ 0.08 22.21  5.69  4.36 14.39  0.51 77.41 62.3   3.56]
 [ 0.    1.38  0.32  0.19  0.89  0.02  4.57  3.56  0.63]]
X'Y vector (9x1) scaled by 10^7:  [0.   1.13 0.29 0.19 0.66 0.03 3.57 2.83 0.2 ]

The Normal Equations are: X'X * B = X'Y
Where B is the vector of regression coefficients (intercept + slopes).
```

```python
#generate OLS regression results for all features
import statsmodels.api as sm

X_sm = sm.add_constant(X)
model = sm.OLS(y,X_sm)
print(model.fit().summary())
```

OLS Regression Results

Dep. Variable:	csMPa	R-squared:	0.616
Model:	OLS	Adj. R-squared:	0.613
Method:	Least Squares	F-statistic:	204.3
Date:	Fri, 15 Oct 2021	Prob (F-statistic):	6.29e-206
Time:	16:43:15	Log-Likelihood:	-3869.0
No. Observations:	1030	AIC:	7756.
Df Residuals:	1021	BIC:	7800.
Df Model:	8		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	-23.3312	26.586	-0.878	0.380	-75.500	28.837
cement	0.1198	0.008	14.113	0.000	0.103	0.136
slag	0.1039	0.010	10.247	0.000	0.084	0.124
flyash	0.0879	0.013	6.988	0.000	0.063	0.113
water	-0.1499	0.040	-3.731	0.000	-0.229	-0.071
superplasticizer	0.2922	0.093	3.128	0.002	0.109	0.476
coarseaggregate	0.0181	0.009	1.926	0.054	-0.000	0.037
fineaggregate	0.0202	0.011	1.887	0.059	-0.001	0.041
age	0.1142	0.005	21.046	0.000	0.104	0.125

O coeficiente estimado $\hat{\beta}_j$ é uma variável aleatória

$$\hat{\beta}_j \sim N(\beta_{\text{true}}, v^2)$$

não-viciado

$$v^2 = \mathbb{V}(\hat{\beta}_j) = \sigma^2 \text{diag}(\mathbf{X}'\mathbf{X})^{-1}[jj]$$

```python
#generate OLS regression results for all features
import statsmodels.api as sm

X_sm = sm.add_constant(X)
model = sm.OLS(y,X_sm)
print(model.fit().summary())
```

$$\hat{\beta}_j \sim N(\beta_j, \sigma^2(X'X)^{-1}[jj])$$

OLS Regression Results

Dep. Variable:	csMPa	R-squared:	0.616
Model:	OLS	Adj. R-squared:	0.613
Method:	Least Squares	F-statistic:	204.3
Date:	Fri, 15 Oct 2021	Prob (F-statistic):	6.29e-206
Time:	16:43:15	Log-Likelihood:	-3869.0
No. Observations:	1030	AIC:	7756.
Df Residuals:	1021	BIC:	7800.
Df Model:	8		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	-23.3312	26.586	-0.878	0.380	-75.500	28.837
cement	0.1198	0.008	14.113	0.000	0.103	0.136
slag	0.1039	0.010	10.247	0.000	0.084	0.124
flyash	0.0879	0.013	6.988	0.000	0.063	0.113
water	-0.1499	0.040	-3.731	0.000	-0.229	-0.071
superplasticizer	0.2922	0.093	3.128	0.002	0.109	0.476
coarseaggregate	0.0181	0.009	1.926	0.054	-0.000	0.037
fineaggregate	0.0202	0.011	1.887	0.059	-0.001	0.041
age	0.1142	0.005	21.046	0.000	0.104	0.125

A coluna std error e´ a raiz da diagonal dessa matriz de covariância

É o desvio-padrão do coeficiente estimado.

```python
#generate OLS regression results for all features
import statsmodels.api as sm

X_sm = sm.add_constant(X)
model = sm.OLS(y,X_sm)
print(model.fit().summary())
```

OLS Regression Results

Dep. Variable:	csMPa	R-squared:	0.616
Model:	OLS	Adj. R-squared:	0.613
Method:	Least Squares	F-statistic:	204.3
Date:	Fri, 15 Oct 2021	Prob (F-statistic):	6.29e-206
Time:	16:43:15	Log-Likelihood:	-3869.0
No. Observations:	1030	AIC:	7756.
Df Residuals:	1021	BIC:	7800.
Df Model:	8		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	-23.3312	26.586	-0.878	0.380	-75.500	28.837
cement	0.1198	0.008	14.113	0.000	0.103	0.136
slag	0.1039	0.010	10.247	0.000	0.084	0.124
flyash	0.0879	0.013	6.988	0.000	0.063	0.113
water	-0.1499	0.040	-3.731	0.000	-0.229	-0.071
superplasticizer	0.2922	0.093	3.128	0.002	0.109	0.476
coarseaggregate	0.0181	0.009	1.926	0.054	-0.000	0.037
fineaggregate	0.0202	0.011	1.887	0.059	-0.001	0.041
age	0.1142	0.005	21.046	0.000	0.104	0.125

$$\hat{\beta}_j \sim N(\beta_j, \sigma^2 (X'X)^{-1}[jj])$$

A matriz X de features é conhecida.

Mas, e σ^2 ??

Ele é um parâmetro tão desconhecido quanto o verdadeiro β

Ele precisa ser estimado (aprendido). Como?