CS145 Discussion
Week 1

Junheng, Shengming, Yunsheng
10/05/2018

Roadmap

- Class logistics & Reminders
- Math review: Basics of
  - Probability
  - Linear Algebra
  - Optimization
  - Matrix Calculus
- Linear Regression

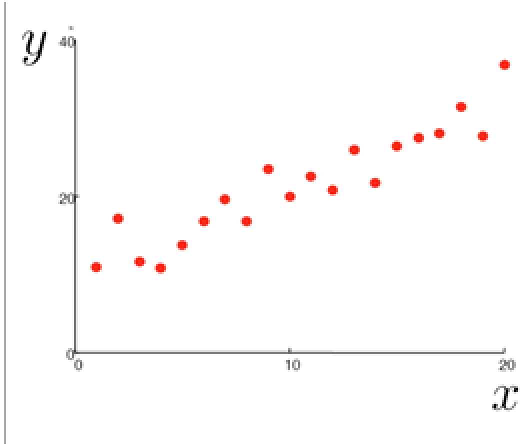
Announcement

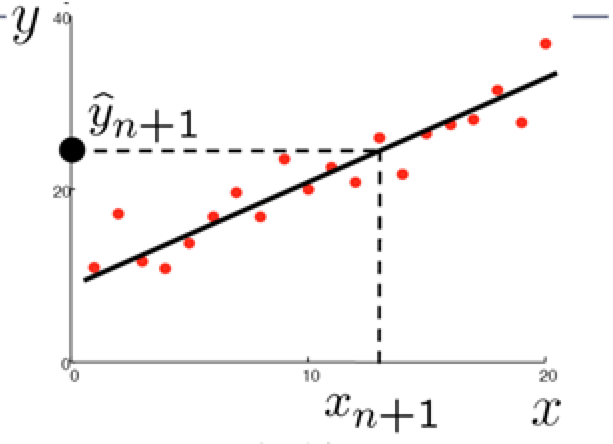
- Course Website:
  - http://web.cs.ucla.edu/~yzsun/classes/2018Fall_CS145/index.html
- Dates:
  - Project introduction on next Monday (10/08)
  - Homework 1 out on next Wednesday (10/10)
  - In-class midterm on Nov. 14 (12-2pm)
  - Final exam schedules on Dec 13 (11:30am-2:30pm)
- Update: TA office hours
  - Yizhou Sun (yzsun@cs.ucla.edu), **office hours: 3-5pm Wednesdays** @BH 3531E
  - Yunsheng Bai (yba@cs.ucla.edu), **office hours: 1-3pm Thursdays** @BH 3256S
  - Junheng Hao (haojh.ucla@gmail.com), **office hours: 1-3pm Tuesdays** @BH 3256S
  - Shengming Zhang (michaelzhang@cs.ucla.edu), **office hours: 3-5pm Mondays** @BH 3256S

Math Review

1. Probability
2. Linear Algebra
3. Optimization
4. Matrix Calculus

Linear Regression





Can this be fitted using a straight line?
No!
If not, can we still use the idea of linear regression?
Yes! (sometimes called Polynomial Regression)

Linear Regression → Polynomial Regression

- Data: n independent data objects
  - $y_i, i = 1, \dots, n$
  - $\boldsymbol{x}_i = \boxed{(x_{i1}, x_{i2}, \dots, x_{ip})^{\mathrm{T}}}, i = 1, \dots, n$ $\quad \boxed{(x, x^2, x^3, \dots)^{\mathrm{T}}}$
    - A constant factor is added to model the bias term, i.e., $x_{i0} = 1$
    - New x: $\boldsymbol{x}_i = (x_{i0}, x_{i1}, x_{i2}, \dots, x_{ip})^{\mathrm{T}}$
- Model:
  - y: *dependent variable*
  - $\boldsymbol{x}$: *explanatory variables*
  - $\boldsymbol{\beta} = (\beta_0, \beta_1, \dots, \beta_p)^T$: *weight vector*
  - $y = \boldsymbol{x}^T \boldsymbol{\beta} = \boxed{\beta_0 + x_1\beta_1 + x_2\beta_2 + \cdots + x_p\beta_p} \rightarrow \boxed{\beta_0 + x\beta_1 + x^2\beta_2 + \dots}$

Matrix form of Least Square Estimation

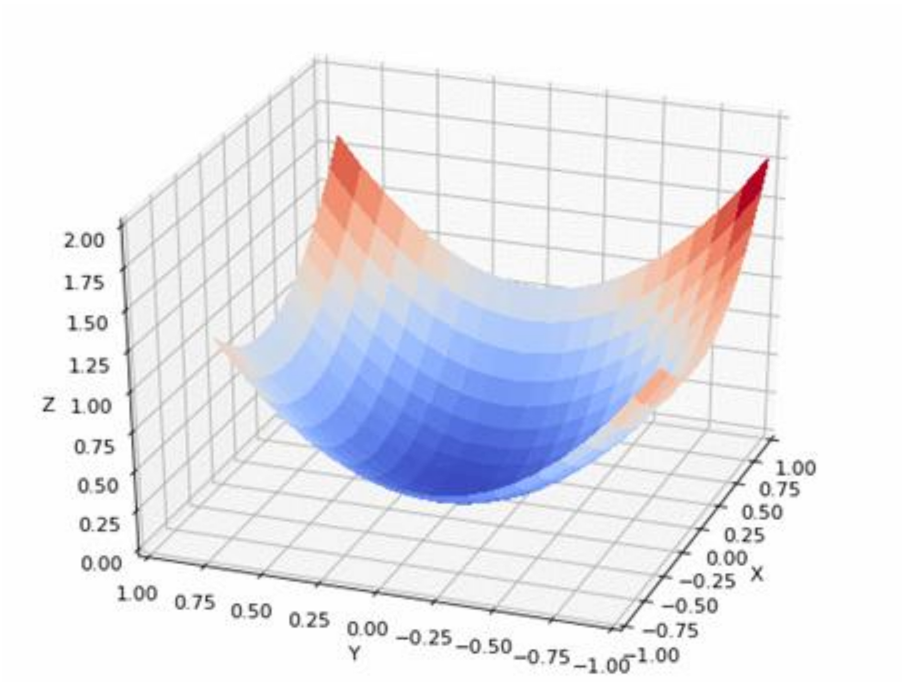
$$J(\boldsymbol{\beta}) = (X\boldsymbol{\beta} - \mathbf{y})^T (X\boldsymbol{\beta} - \mathbf{y})/2$$

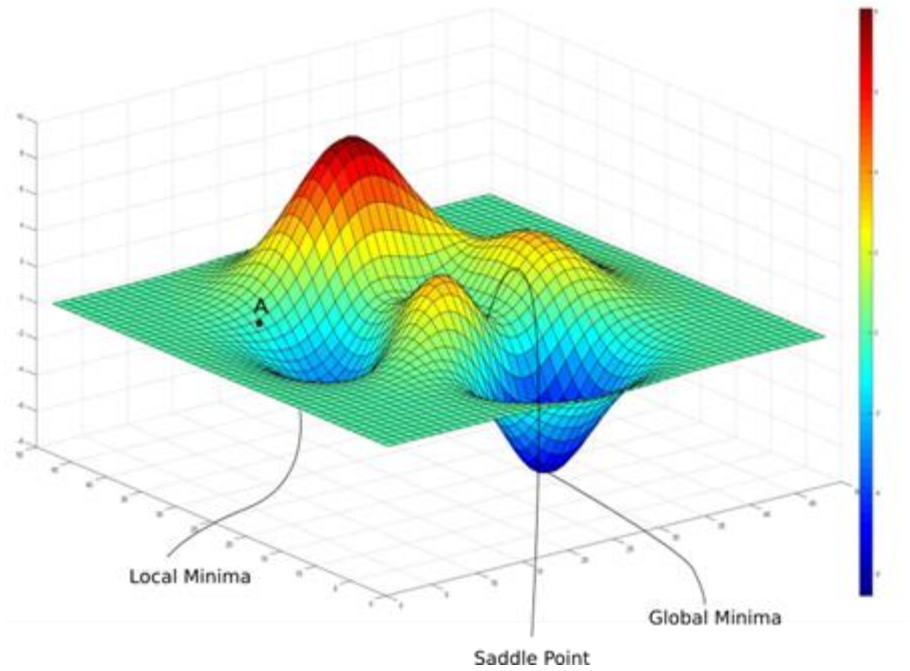
$$\begin{bmatrix} 1, x_{11} & \cdots & x_{1f} & \cdots & x_{1p} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 1, x_{i1} & \cdots & x_{if} & \cdots & x_{ip} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 1, x_{n1} & \cdots & x_{nf} & \cdots & x_{np} \end{bmatrix} \times \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_p \end{pmatrix} - \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} = \begin{pmatrix} X_1\beta - y_1 \\ X_2\beta - y_2 \\ \vdots \\ X_n\beta - y_n \end{pmatrix}$$

X: $n \times (p+1)$ matrix $\quad$ β: (p + 1) x 1 matrix $\quad$ y: n x 1 matrix $\quad$ $X\beta$ - y: n x 1 matrix

$$J(\boldsymbol{\beta}) = \|X\boldsymbol{\beta} - \mathbf{y}\|^2/2$$

Gradient Descent



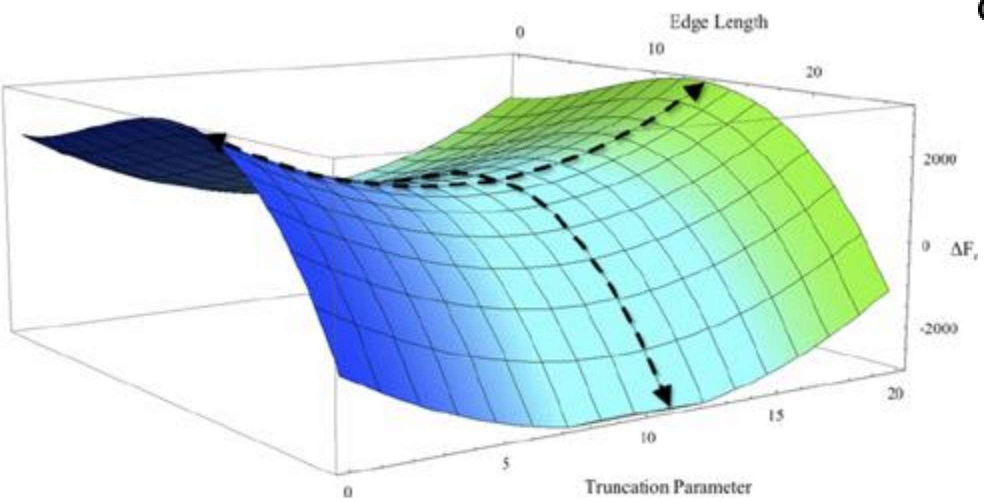


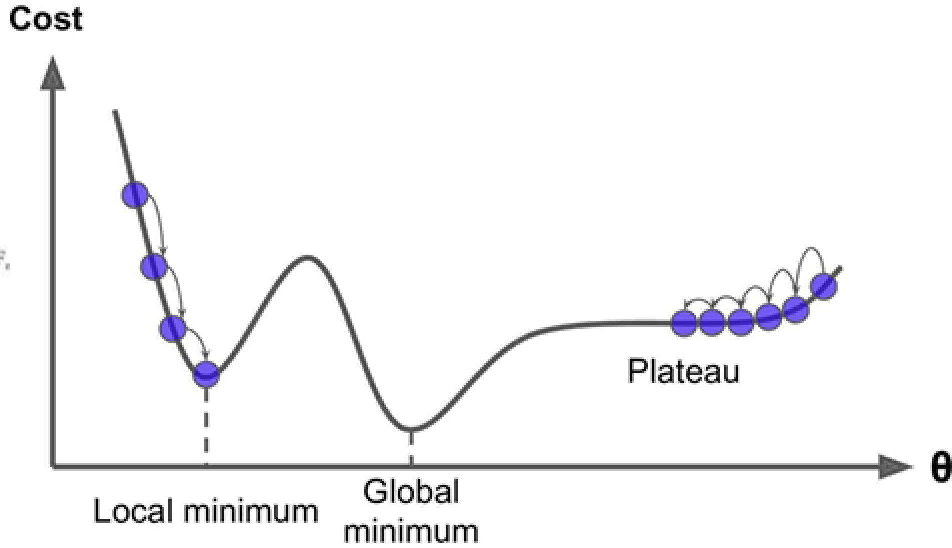
https://blog.paperspace.com/intro-to-optimization-in-deep-learning-gradient-descent/

Batch vs Stochastic Gradient Descent

Why do we need Stochastic GD besides the efficiency/scalability reason?

Think about saddle points and local minima.

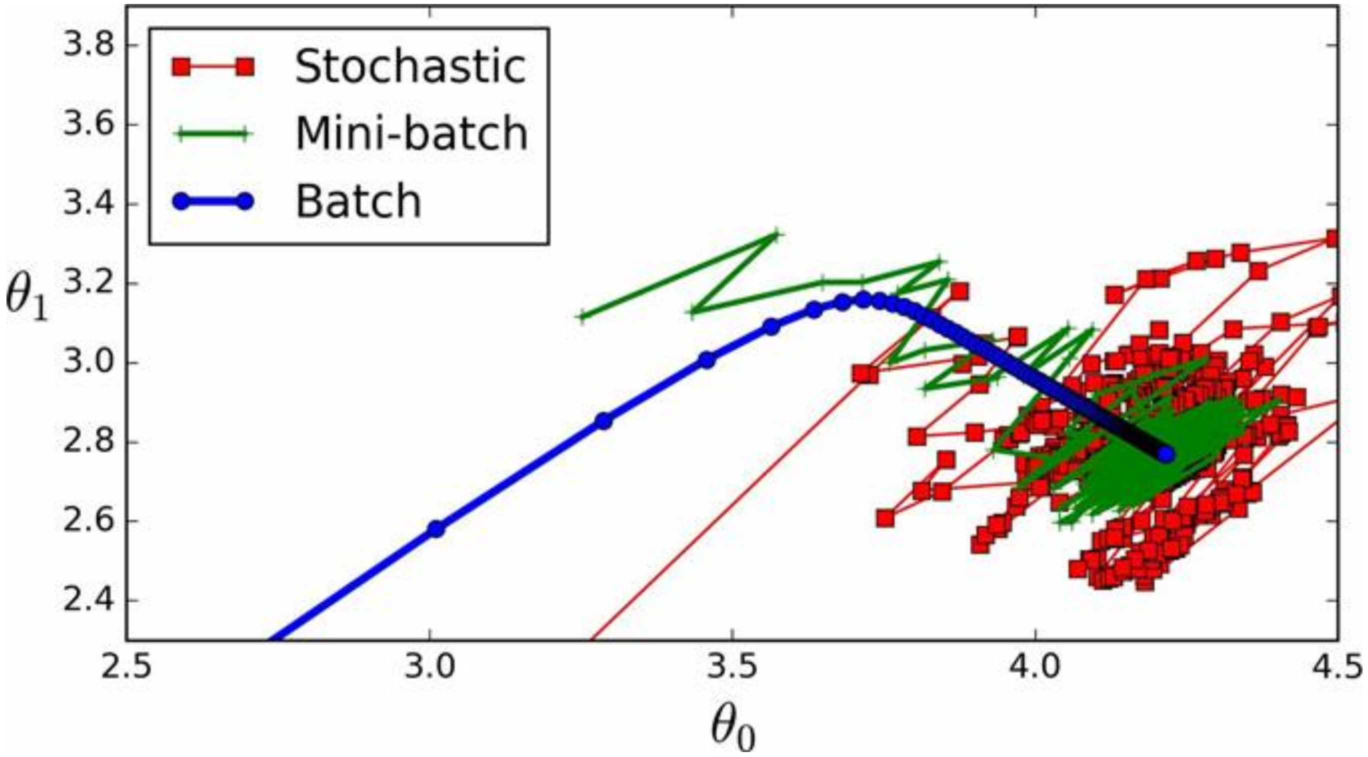




https://blog.paperspace.com/intro-to-optimization-in-deep-learning-gradient-descent/

Géron, Aurélien. *Hands-on machine learning with Scikit-Learn and TensorFlow: concepts, tools, and techniques to build intelligent systems*. " O'Reilly Media, Inc.", 2017.

Comparison



Géron, Aurélien. *Hands-on machine learning with Scikit-Learn and TensorFlow: concepts, tools, and techniques to build intelligent systems*. " O'Reilly Media, Inc.", 2017.

Linear Regression: When to use...

1. Closed form solution? $\Rightarrow \hat{\boldsymbol{\beta}} = \left(X^T X\right)^{-1} X^T y$
2. Gradient descent
    a. Batch GD?
    b. Stochastic GD?
    c. Mini-batch GD?

https://stats.stackexchange.com/questions/49528/batch-gradient-descent-versus-stochastic-gradient-descent

Feature Extraction from Real Data

- Types of Features
  - Numerical
  - Categorical
    - Nominal, Binary, Ordinal

- Real data may be messy for extracting features
  - Unorganized structure
  - Hidden and deep information

Example of a tweet data

Numerical Features

- Numerical attributes
  - Raw data with numerical formats
  - E.g., numbers of friends and followers, timestamps
- Numerical statistics
  - Numerical statistics towards a characteristic
  - E.g., the length of text, the average daily number of tweets for the user
- Numerical hidden representations
  - Represent data in optimized hidden spaces
  - E.g, pLSA and LDA for text (Week 10)

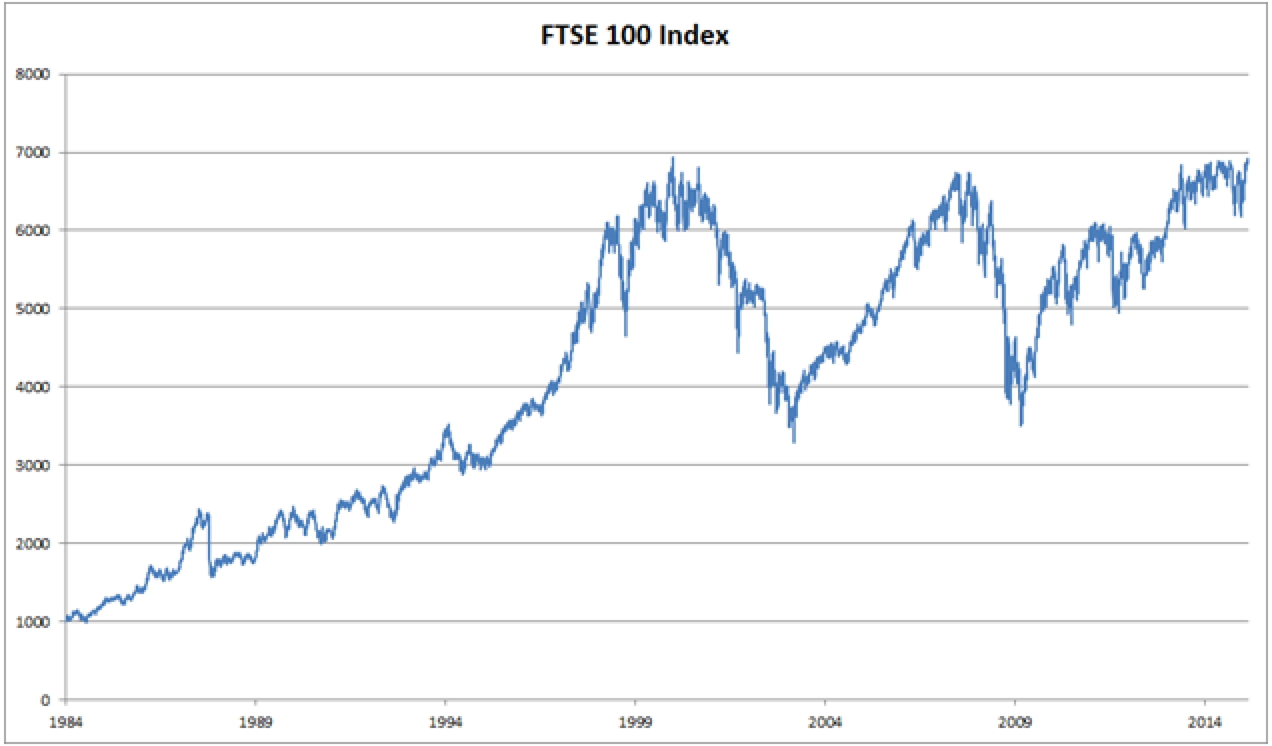
Categorical Features

- Categorical attributes
  - Raw data which originally have a set of discrete categories
  - E.g., cities of users, languages of text,

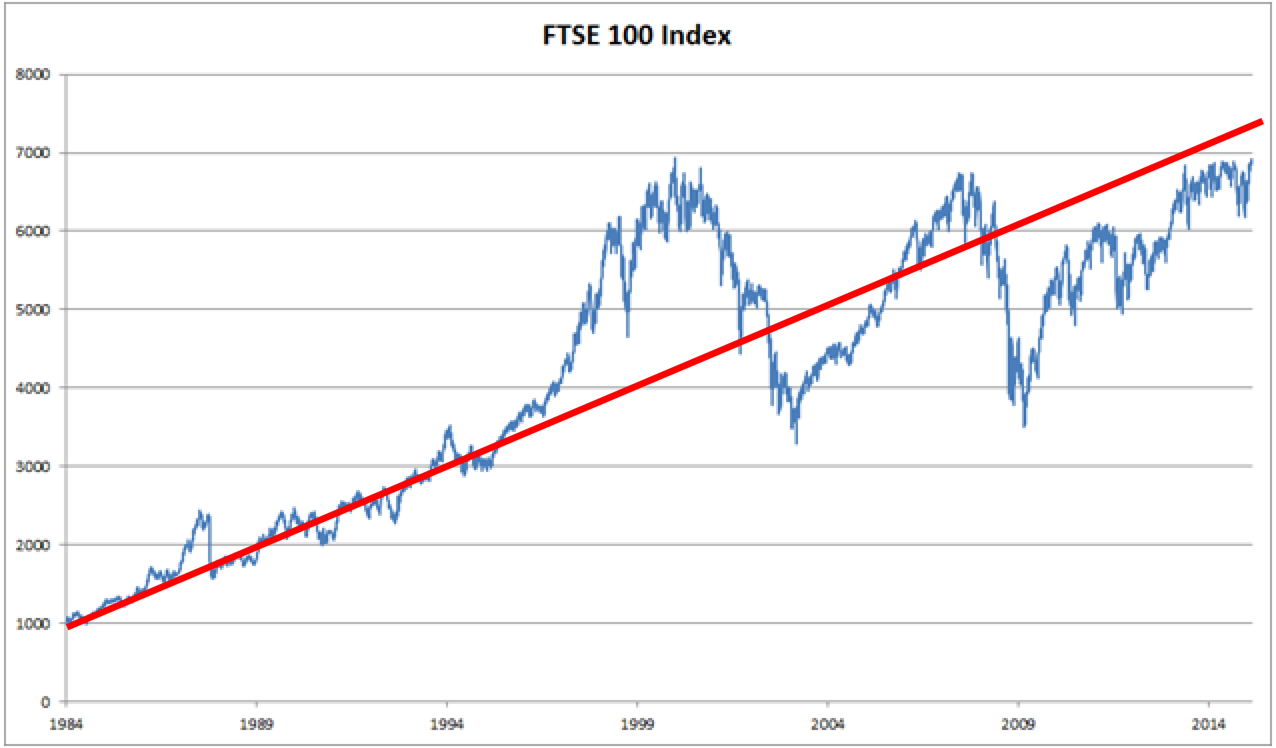
- Discretization for numerical attributes
  - Transform numerical features into categorical features
  - E.g., Morning/Afternoon/Night, Long/Short Text (more than k words?)

- Categorical statistics
  - Categorical statistics towards a characteristic
  - E.g., If the user posts more than k tweets in a week, Few/Usual/Many tweets posted in near regions

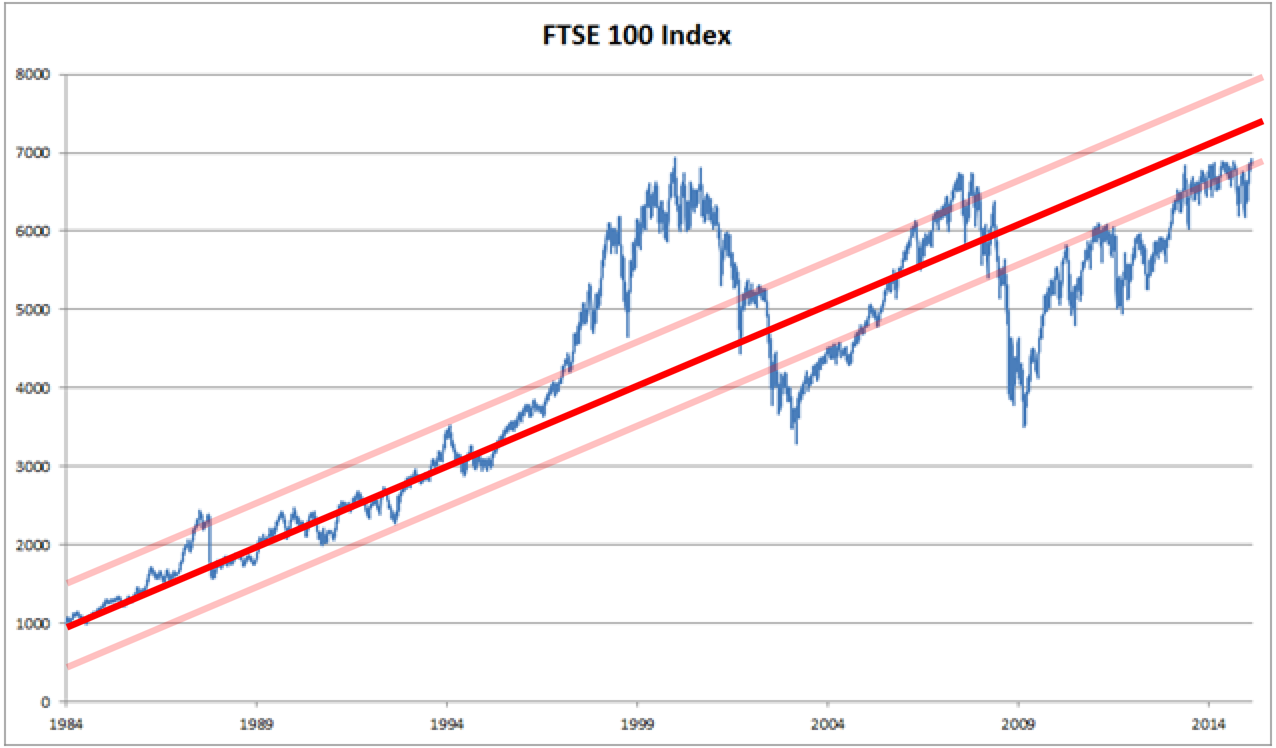
An application of linear regression: Stock Prices



An application of linear regression: Stock Prices



An application of linear regression: Stock Prices



Appendix: Unbiased Estimator

Sample variance vs Variance

Proof & Explanation:https://en.wikipedia.org/wiki/Bessel%27s_correction

Thank you!

Q & A