# Linear Regression and Resampling Methods for Uncertainty

## Big Data y Machine Learning para Economía Aplicada

Ignacio Sarmiento-Barbieri

Universidad de los Andes

# Agenda

1. Review
2. OLS
   - Traditional Computation
   - Gradient Descent
   - Numerical Properties
3. Uncertainty: Motivation
   - What are resampling methods?
   - The Bootstrap
     - Example: Elasticity of Demand for Gasoline
4. Review

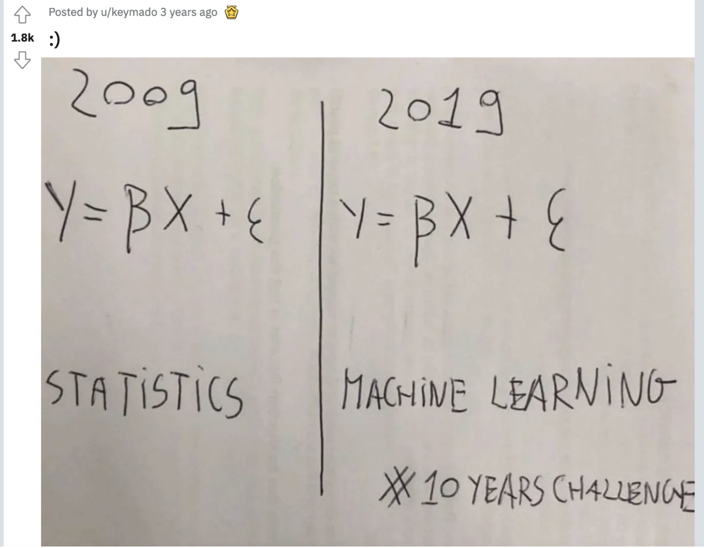
# Agenda

# Predicting Well

$$y = f(X) + u \tag{1}$$

- Interest on predicting $y$
- Under quadratic loss $\Rightarrow E[y|X = x]$

# Linear Regression



# Linear Regression

$$y = f(X) + u \tag{2}$$
$$= \beta_0 + \beta_1 X_1 + \cdots + \beta_k X_k + u \tag{3}$$
$$= X\beta + u \tag{4}$$

- If $f(X) = X\beta$, obtaining $f(.)$ boils down to obtaining $\beta$

# Linear Regression

- OLS says we should choose the estimators $\hat{\beta}$ such that we minimize the Sum of Square Residual (SSR)

$$
\begin{aligned}
\mathcal{L} &= \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \qquad (5) \\
&= \sum_{i=1}^{n} \left( y_i - \hat{\beta}_0 - \sum_{j=1}^{k} \hat{\beta}_j x_{ji} \right)^2 \qquad (6)
\end{aligned}
$$

# Agenda

# Agenda

# Linear Regression

- Using matrix algebra, the loss function:

$$SSR(\tilde{\beta}) \equiv \sum_{i=1}^{n} \tilde{e}_i^2 = \tilde{e}'\tilde{e} = (y - X\tilde{\beta})'(y - X\tilde{\beta}) \tag{7}$$

  - $SSR(\tilde{\beta})$ is the aggregation of squared errors if we choose $\tilde{\beta}$ as an estimator.

- The **least squares estimator** $\hat{\beta}$ will be

$$\hat{\beta} = \underset{\tilde{\beta}}{argmin}\, SSR(\tilde{\beta}) \tag{8}$$

# OLS

- FOC are

$$\frac{\partial \tilde{e}'\tilde{e}}{\partial \tilde{\beta}} = 0 \tag{9}$$

$$-2X'y + 2X'X\tilde{\beta} = 0 \tag{10}$$

- SOC (H.W.)

# OLS

- Let $\hat{\beta}$ be the solution.Then $\hat{\beta}$ satisfies the following normal equation

$$X'X\hat{\beta} = X'y \tag{11}$$

- If the inverse of $X'X$ exists, then

$$\hat{\beta} = (X'X)^{-1}X'y \tag{12}$$

- Pro
  - Closed solution (a bonus!!)
- Cons
  - Involves inverting a $k \times k$ matrix $X'X$
  - requires allocating $O(nk + k^2)$ if n is "big" we cannot store in memory

# QR decomposition

- To avoid inverting $X'X$ we can use matrix decomposition: QR decomposition
- Most software use it

**Theorem** If $A \in \mathbb{R}^{n \times k}$ then there exists an orthogonal $Q \in \mathbb{R}^{n \times k}$ and an upper triangular $R \in \mathbb{R}^{k \times k}$ so that $A = QR$

- Orthogonal Matrices:
  - Def: $Q'Q = QQ' = I$ and $Q' = Q^{-1}$
  - Prop: product of orthogonal is orthogonal, e.g $A'A = I$ and $B'B = I$ then $(AB)'(AB) = B'(A'A)B = B'B = I$
- **(Thin QR)** If $A \in \mathbb{R}^{n \times k}$ has full column rank then $A = Q_1 R_1$ the QR factorization is unique, where $Q_1 \in \mathbb{R}^{n \times k}$ and $R$ is upper triangular with positive diagonal entries

# `QR` decomposition

- $\hat{\beta}$?

$$(X'X)\hat{\beta} = X'y \tag{13}$$
$$(R'Q'QR)\hat{\beta} = R'Q'y \tag{14}$$
$$(R'R)\hat{\beta} = R'Q'y \tag{15}$$
$$R\hat{\beta} = Q'y \tag{16}$$

- Solve by back substitution

# QR decomposition

$$X = \begin{bmatrix} 1 & 2 \\ 1 & 2 \\ 1 & 3 \end{bmatrix} \quad y = \begin{pmatrix} 1 \\ 4 \\ 2 \end{pmatrix} \tag{17}$$

1. QR factorization X=QR

$$Q = \begin{bmatrix} -0.57 & -0.41 \\ -0.57 & -0.41 \\ -0.57 & 0.82 \end{bmatrix} \quad R = \begin{bmatrix} -1.73 & -4.04 \\ 0 & 0.81 \end{bmatrix} \tag{18}$$

2. Calculate $Q'y = [-4.04, -0.41]'$
3. Solve

$$\begin{bmatrix} -1.73 & -4.04 \\ 0 & 0.81 \end{bmatrix} \begin{bmatrix} \beta_1 \\ \beta_2 \end{bmatrix} = \begin{bmatrix} -4.04 \\ -0.41 \end{bmatrix} \tag{19}$$

Solution is $(3.5, -0.5)$

# QR decomposition

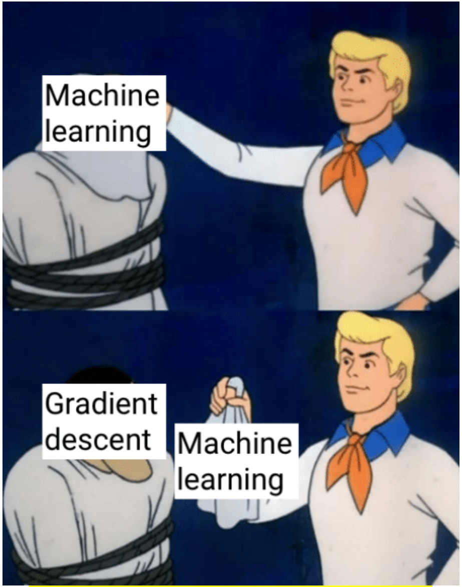
This is actually what R does under the hood

| | | |
|---|---|---|
| obj | list [12] (S3: lm) | List of length 12 |
| coefficients | double [2] | -1.71e+08 3.01e+08 |
| residuals | double [207607] | 1.17e+09 -2.38e+08 -5.21e+08 -1.96e+08 -5.12e+07 -1.91e+08 ... |
| effects | double [207607] | -3.15e+11 2.10e+11 -5.24e+08 -1.98e+08 -5.34e+07 -1.93e+08 ... |
| rank | integer [1] | 2 |
| fitted.values | double [207607] | 4.31e+08 4.31e+08 7.32e+08 4.31e+08 4.31e+08 4.31e+08 ... |
| assign | integer [2] | 0 1 |
| qr | list [5] (S3: qr) | List of length 5 |
| df.residual | integer [1] | 207605 |
| xlevels | list [0] | List of length 0 |
| call | language | lm(formula = price ~ bathrooms, data = dta0) |
| terms | formula | price ~ bathrooms |
| model | list [207607 x 2] (S3: data.fra | A data.frame with 207607 rows and 2 columns |

Note that `R's` `lm` also returns many objects that have the same size as X and Y

# Agenda

# Gradient Descent



# Gradient Descent

- Gradient Descent is a very generic optimization algorithm capable of finding optimal solutions to a wide range of problems.
- The general idea of Gradient Descent is to tweak parameters iteratively in order to minimize a loss function.

$$\min_f \sum_{i=1}^{n} L\left(y_i, f\left(\boldsymbol{X_i}\right)\right) \tag{20}$$

$$L\left(y_i, f\left(\boldsymbol{X_i}\right)\right) = \left(y_i - f\left(\boldsymbol{X_i}\right)\right)^2 \tag{21}$$

# Gradient Descent

## Linear regression

- The problem boils down to estimating the coefficients of vector $\beta$ which minimise an objective function:

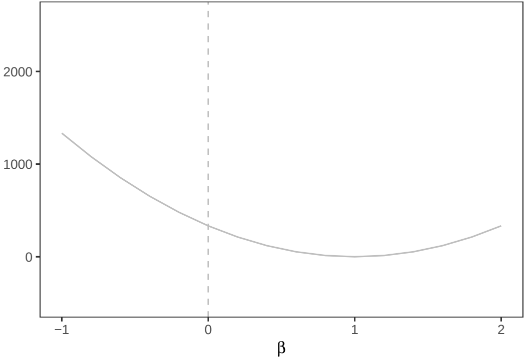
$$\arg\min_{\beta} \sum_{i=1}^{n} L\left(y_i, f\left(\boldsymbol{X_i}\right)\right), \tag{22}$$

$$\arg\min_{\beta} \sum_{i=1}^{n} \left(y_i - \beta \boldsymbol{X_i}\right)^2 \tag{23}$$

# Gradient Descent

## Linear regression

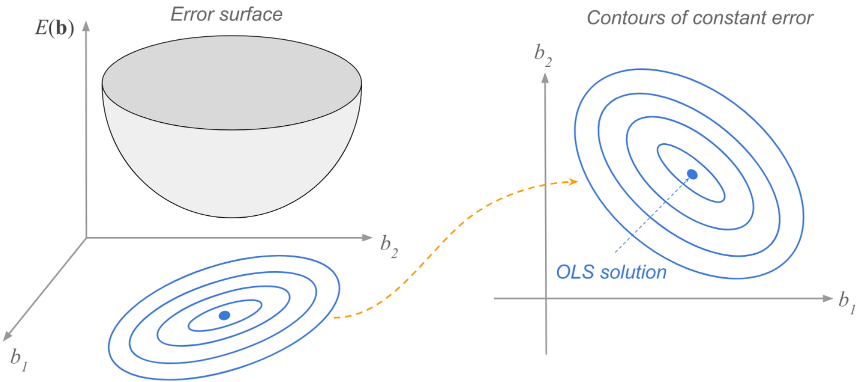
- Intuition: Loss Function 1 dimension (App)



# Gradient Descent

## Linear regression

- Intuition: Loss Function 2 dimension (App)



# Gradient Descent

- In a more general context, when at a point $\beta \in \mathbb{R}^k$, at any step $i$, the gradient descent algorithm tries to move in a direction $\delta\beta$ such that:

$$L(\beta^{(t)} + \delta\beta) < L(\beta^{(t)}) \tag{24}$$

- The choice of $\delta\beta$ is made such that $\delta\beta = -\epsilon\nabla_\beta L(\beta^{(t)})$:

$$\beta^{(t+1)} = \beta^{(t)} - \epsilon\nabla_\beta L(\beta^{(t)}) \tag{25}$$

- In other words, you need to calculate how much the cost function will change if you change $\beta$ just a little bit.

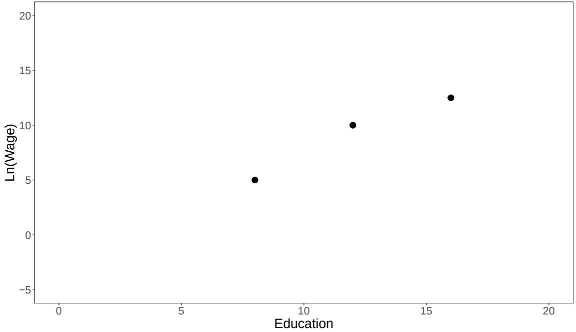
# Gradient Descent

- Algorithm

  1. Randomly pick starting values for the parameters
  2. Compute the gradient of the objective function at the current value of the parameters using all the observations from the training sample
  3. Update the parameters
  4. Repeat from step 2 until a fixed number of iteration or until convergence.

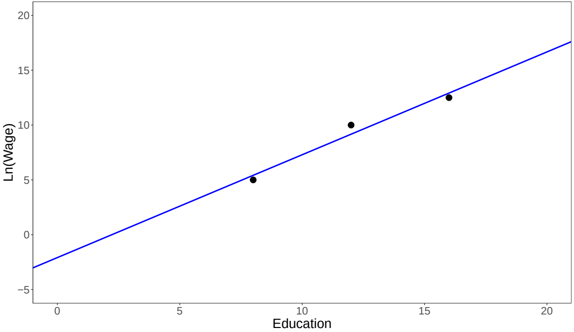
# Gradient Descent: Example

| log(wage) | Education (years) |
|---|---|
| 5 | 8 |
| 10 | 12 |
| 12.5 | 16 |



# Gradient Descent: Example

| log(wage) | Education (years) |
|---|---|
| 5 | 8 |
| 10 | 12 |
| 12.5 | 16 |




$$\hat{\beta} = (X'X)^{-1}X'y$$

```
beta<-solve(t(X)%*%X)%*%t(X)%*%y

lm(y~x,data)
```

$$y = -2.0833 + 0.9375 \times Educ$$

# Gradient Descent: Example

## The Risk Function

$$SSR = f(\theta) = \sum_{i=1}^{n}(y_i - (\alpha + \beta x_i))^2$$

## The Gradient

$$\nabla f_\theta(\theta) = \begin{pmatrix} \frac{\partial f}{\partial \alpha} \\ \frac{\partial f}{\partial \beta} \end{pmatrix} = \begin{pmatrix} -2\sum_{i=1}^{n}(y_i - \alpha - \beta x_i) \\ -2\sum_{i=1}^{n} x_i(y_i - \alpha - \beta x_i) \end{pmatrix}$$

## Updating

$$\alpha^{(t+1)} = \alpha^{(t)} - \epsilon \frac{\partial f}{\partial \alpha}$$

$$\beta^{(t+1)} = \beta^{(t)} - \epsilon \frac{\partial f}{\partial \beta}$$

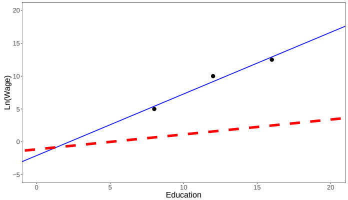
# Gradient Descent: Example

First Iteration

| log(wage) | Education (years) |
|---|---|
| 5 | 8 |
| 10 | 12 |
| 12.5 | 16 |

Start with an initial guess: $\alpha = -1; \beta = 2$ , and a learning rate ($\epsilon = 0.005$). Then we have

$$
\begin{aligned}
\alpha' &= (-1) - 0.005\,(-2\,((5-(-1)-2\times 8) + (10-(-1)-2\times 12) + (12.5-(-1)-2\times 16)) \\
\beta' &= 2 + 0.005\,(-2\,(8(5-(-1)-2\times 8) + 12(10-(-1)-2\times 12) + 16(12.5-(-1)-2\times 16)) \\
\alpha' &= -1.1384 \\
\beta' &= 0.2266
\end{aligned}
$$



# Gradient Descent: Example

Second Iteration

| log(wage) | Education (years) |
|---|---|
| 5 | 8 |
| 10 | 12 |
| 12.5 | 16 |

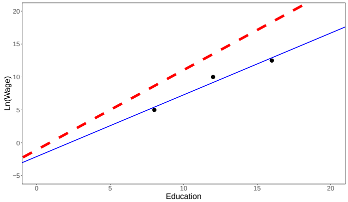
Start with an initial guess: $\alpha = -1; \beta = 2$ , and a learning rate ($\epsilon = 0.005$). Then we have

$$\alpha^2 = (-1.1384) - 0.005\,(-2\,((5 - (-1.1384) - (0.2266) \times 8) + (10 - (-1.1384) - (0.2266) \times 12) + (12.5 - (-1.1384) - (0.2266) \times 16))$$

$$\beta^2 = (0.2266) + 0.005\,(-2\,(8(5 - (-1.1384) - (0.2266) \times 8) + 12(10 - (-1.1384) - (0.2266) \times 12) + 16(12.5 - (-1.1384) - (0.2266) \times 16))$$

$$\alpha^2 = -1.0624$$

$$\beta^2 = 1.212689$$



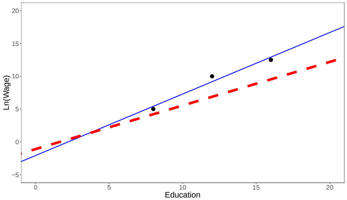
# Gradient Descent: Example

Third Iteration

| log(wage) | Education (years) |
|---|---|
| 5 | 8 |
| 10 | 12 |
| 12.5 | 16 |

$$\alpha^3 = -1.0624$$

$$\beta^3 = 1.212689$$



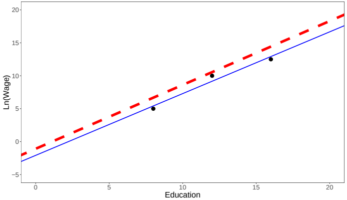
# Gradient Descent: Example

Fourth Iteration

| log(wage) | Education (years) |
|---|---|
| 5 | 8 |
| 10 | 12 |
| 12.5 | 16 |

$$\alpha^4 = -1.082738$$

$$\beta^4 = 0.9693922$$



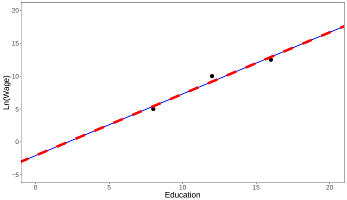
# Gradient Descent: Example

7211 Iteration

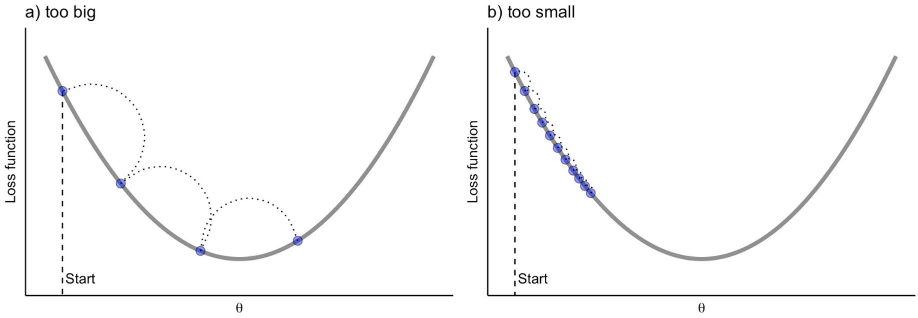
| log(wage) | Education (years) |
|---|---|
| 5 | 8 |
| 10 | 12 |
| 12.5 | 16 |

$$\alpha^{7211} = -2.076246$$
$$\beta^{7211} = 0.9369499$$

$$y^{ols} = -2.0833 + 0.9375 \times Educ$$



# The learning rate




Source: Boehmke, B., & Greenwell, B. (2019)

- We can choose $\epsilon$ in several different ways:
  - Set $\epsilon$ to a small constant.
  - Use varying learning rates.

# Agenda

# Numerical Properties

- Numerical properties have nothing to do with how the data was generated
- These properties hold for every data set, just because of the way that $\hat{\beta}$ was calculated
- Davidson & MacKinnon, Greene y Ruud have nice geometric interpretations
- Helps in computing with big data

# Projection

OLS Residuals:

$$e = y - \hat{y} \tag{26}$$
$$= y - X\hat{\beta} \tag{27}$$

replacing $\hat{\beta}$

$$e = y - X(X'X)^{-1}X'y \tag{28}$$
$$= (I - X(X'X)^{-1}X')y \tag{29}$$

Define two matrices

- Projection matrix $P_X = X(X'X)^{-1}X'$
- Annihilator (residual maker) matrix $M_X = (I - P_X)$

# Projection

- $P_X = X(X'X)^{-1}X'$
- $M_X = (I - P_X)$
- Both are symmetric
- Both are idempotent $(A'A) = A$
- $P_X X = X \Rightarrow$ projection matrix
- $M_X X = 0 \Rightarrow$ annihilator matrix

We can write

$$SSR = e'e = u'M_X u \tag{30}$$

So we can relate SSR to the true error term u

# Frisch-Waugh-Lovell (FWL) Theorem

- Lineal Model: $Y = X\beta + u$
- Split it: $Y = X_1\beta_1 + X_2\beta_2 + u$
  - $X = [X_1\, X_2]$, $X$ is $n \times k$, $X_1$ $n \times k_1$, $X_2$ $n \times k_2$, $k = k_1 + k_2$
  - $\beta = [\beta_1\, \beta_2]$

**Theorem**

1. The OLS estimates of $\beta_2$ from these equations

$$y = X_1\beta_1 + X_2\beta_2 + u \tag{31}$$

$$M_{X_1}y = M_{X_1}X_2\beta_2 + residuals \tag{32}$$

   are numerically identical

2. the OLS residuals from these regressions are also numerically identical

# Applications

- Why FWL is useful in the context of big volume of data?
- An computationally inexpensive way of
  - Removing nuisance parameters
    - E.g. the case of multiple fixed effects. The traditional way is either apply the within transformation with respect to the FE with more categories then add one dummy for each category for all the subsequent FE
    - Not feasible in certain instances.
  - Computing certain diagnostic statistics: Leverage, $R^2$, LOOCV.
  - Way to add more data without having to compute everything again

# Applications: Fixed Effects

- For example: Carneiro, Guimarães, & Portugal (2012) *AEJ: Macroeconomics*

$$\ln w_{ijft} = x_{it}\beta + \lambda_i + \theta_j + \gamma_f + u_{ijft} \tag{33}$$

$$Y = X\beta + D_1\lambda + D_2\theta + D_3\gamma + u \tag{34}$$

  - Data set 31.6 million observations, with 6.4 million individuals (i), 624 thousand firms (f), and 115 thousand occupations (j), 11 years (t).
  - Storing the required indicator matrices would require 23.4 terabytes of memory
  - From their paper
    *"In our application, we first make use of the Frisch-Waugh-Lovell theorem to remove the influence of the three high- dimensional fixed effects from each individual variable, and, in a second step, implement the final regression using the transformed variables. With a correction to the degrees of freedom, this approach yields the exact least squares solution for the coefficients and standard errors"*

# Applications: Outliers and High Leverage Data

- Note the following

$$\hat{\beta} = (X'X)^{-1}X'y \tag{35}$$

- each element of the vector of parameter estimates $\hat{\beta}$ is simply a weighted average of the elementes of the vector $y$
- Let's call $c_j$ the j-th row of the matrix $(X'X)^{-1}X'$ then

$$\hat{\beta}_j = c_j y \tag{36}$$

- App

# Applications: Outliers and High Leverage Data

Consider a dummy variable $e_j$ which is an $n - vector$ with element $j$ equal to 1 and the rest is 0. Include it as a regressor

$$y = X\beta + \alpha e_j + u \tag{37}$$

using FWL we can do

$$M_{e_j} y = M_{e_j} X\beta + r \tag{38}$$

- $\beta$ and *residuals* from both regressions are identical
- Same estimates as those that would be obtained if we deleted observation $j$ from the sample. We are going to denote this as $\beta^{(j)}$

Note:

- $M_{e_j} = I - e_j(e_j'e_j)^{-1}e_j'$
- $M_{e_j} y = y - e_j(e_j'e_j)^{-1}e_j'y = y - y_j e_j$
- $M_{e_j} X$ is X with the $j$ *row* replaced by zeros

# Applications: Outliers and High Leverage Data

Let's define a new matrix $Z = [X, e_j]$

$$y = X\beta + \alpha e_j + u \tag{39}$$

$$y = Z\theta + u \tag{40}$$

then the fitted values and residuals

$$y = P_Z y + M_z y \tag{41}$$

$$= X\hat{\beta}^{(j)} + \hat{\alpha} e_j + M_Z y \tag{42}$$

Pre-multiply by $P_X$ (remember $M_Z P_X = 0$)

$$P_X y = X\hat{\beta}^{(j)} + \hat{\alpha} P_X e_j \tag{43}$$

$$X\hat{\beta} = X\hat{\beta}^{(j)} + \hat{\alpha} P_X e_j \tag{44}$$

$$X(\hat{\beta} - \beta^{(j)}) = \hat{\alpha} P_X e_j \tag{45}$$

# Applications: Outliers and High Leverage Data

How to calculate $\alpha$? FWL once again

$$M_X y = \hat{\alpha} M_X e_j + res \tag{46}$$

$$\hat{\alpha} = (e_j' M_X e_j)^{-1} e_j' M_X y \tag{47}$$

- $e_j' M_X y$ is the $j$ element of $M_X y$, the vector of residuals from the regression including all observations
- $e_j' M_x e_j$ is just a scalar, the diagonal element of $M_X$
  Then

$$\hat{\alpha} = \frac{\hat{u}_j}{1 - h_j} \tag{48}$$

where $h_j$ is the $j$ diagonal element of $P_X$

# Applications: Outliers and High Leverage Data

Finally we get

$$(\hat{\beta}^{(j)} - \hat{\beta}) = -\frac{1}{1-h_j}(X'X)^{-1}X_j'\hat{u}_j \tag{49}$$

# Agenda

# Motivation

- The real world is messy.
- Recognizing this mess will differentiate a sophisticated and useful analysis from one that is hopelessly naive.
- This is especially true for highly complicated models, where it becomes tempting to confuse signal with noise and hence "overfit."
- The ability to deal with this mess and noise is the most important skill you need.

# Uncertainty in Linear Regression

- To get a measure of the uncertainty, precision or variability of our estimates we need a measure
- We can estimate the Variance of our estimators
- Linear regression

$$Var(\hat{\beta}) = Var((X'X)^{-1}X'y) \tag{50}$$

# Uncertainty and Resampling

- Sometimes the analytical expression of the variance can be quite complicated.
- In these cases we can use the bootstrap
- The bootstrap provides a way to perform statistical inference by resampling from the sample.

# Agenda

# What are resampling methods?

- Tools that involves repeatedly drawing samples from a training set and refitting a model of interest on each sample in order to obtain more information about the fitted model
  - Parameter Assessment: estimate standard errors
  - Model Assessment: estimate test error rates
  - They are computationally expensive! But these days we have powerful computers

# Agenda

# The Bootstrap
## Introduction

- Suppose we have $y_1, y_2, \ldots, y_n$ iid $Y \sim (\mu, \sigma^2)$ (both finite)
- We want to estimate

$$Var(\bar{Y}) \tag{51}$$

# The Bootstrap

## Introduction

- Alternative way (no formula!)
  1. From the $n$ original data points $y_1, y_2, \ldots, y_n$ take a sample *with replacement* of size $n$
  2. Calculate the sample average of this *"pseudo-sample"*
  3. Repeat this B times.
  4. Compute the variance of the B means

# The Bootstrap
## Introduction

- The bootstrap is a widely applicable and extremely powerful statistical tool that can be used to quantify the uncertainty associated with a given estimator or statistical learning method.
- In German the expression *an den eigenen Haaren aus dem Sumpf zu ziehen* nicely captures the idea of the bootstrap – *"to pull yourself out of the swamp by your own hair."*

# The Bootstrap
## Introduction

- **Key Innovation**: The sample itself is used to assess the precision of the estimate.
- Why would this work?
- Remember that uncertainty arises from the randomness inherent to our data-generating process
- So if we can approximately simulate this randomness, then we can approximately quantify our uncertainty.

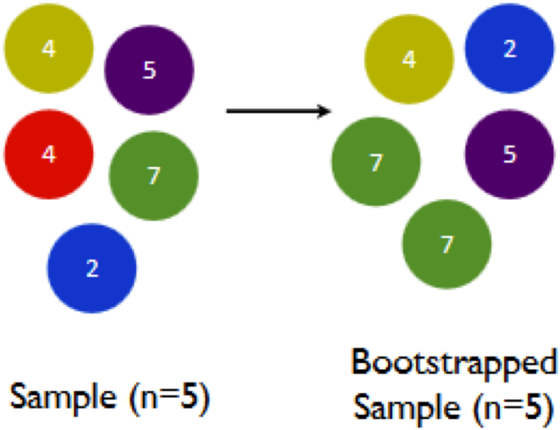
# The Bootstrap

## Introduction

- There are two key properties of bootstrapping that make this seemingly crazy idea actually work.

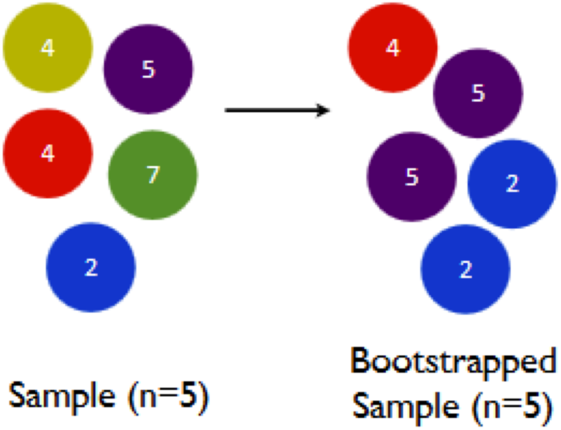
  1. Each bootstrap sample must be of the same size (N) as the original sample
  2. Each bootstrap sample must be taken with replacement from the original sample

# Sampling with replacement



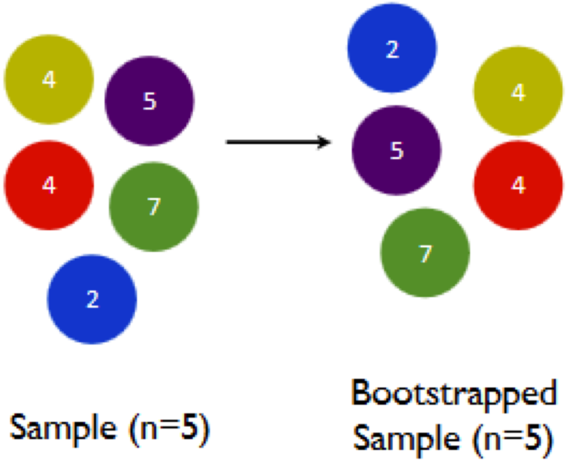
# Sampling with replacement

## Resampling creates synthetic variability



# Sampling with replacement

## Resampling creates synthetic variability

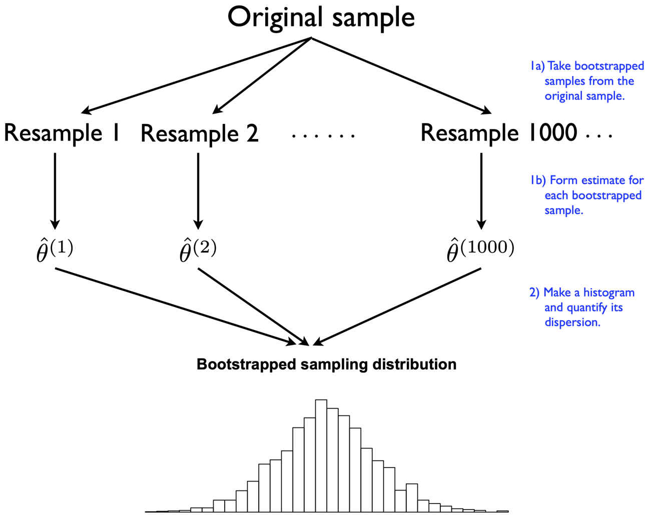


# The Bootstrap

- In general terms:
  - $Y_i \; i = 1, \ldots, n$
  - $\theta$ is the magnitude of interest
- To calculate it's variance
  1. Sample of size $n$ with replacement (*bootstrap sample*)
  2. Compute $\hat{\theta}_j \; j = 1, \ldots, B$
  3. Repeat $B$ times
  4. Calculate

$$\hat{V}(\hat{\theta})_B = \frac{1}{B}\sum_{j=1}^{B}(\hat{\theta}_j - \bar{\hat{\theta}})^2 \tag{52}$$

# The Bootstrap



# Example: Elasticity of Demand for Gasoline


photo from https://www.dailydot.com/parsec/batman-1966-labels-tumblr-twitter-vine/

# Review and Caveats

- The bootstrap is a widely applicable and extremely powerful statistical tool that can be used to quantify the uncertainty associated with a given estimator or statistical learning method.
- The power of the bootstrap, and resampling in general, lies in the fact that it can be easily applied to a wide range of statistical learning methods.
- In particular, it does not assume that the regression errors are iid so it can accommodate heteroscedasticity.
- Of course it does still assume that the observations are independent.
- Resampling dependent observations is an inherently more difficult task which has generated its own rather large literature.

# Agenda

1. Review
2. OLS
   - Traditional Computation
   - Gradient Descent
   - Numerical Properties
3. Uncertainty: Motivation
   - What are resampling methods?
   - The Bootstrap
     - Example: Elasticity of Demand for Gasoline
4. Review

# Review

- So far: The predictive paradigm and linear regression
- Next Week: Out of sample prediction. Overfit, Resampling Methods.