Three Algorithms:
go-to tools in the tooloox

Marco Morales
marco.morales@columbia.edu

GR5069
Topics in Applied Data Science
for Social Scientists

Spring 2019
Columbia University

Three Algorithms

- On a day-to-day basis, you'll commonly use just a few algorithms that help with 80% of your needs:
  1. **OLS**
  2. **logistic regression**
  3. **random forest**
- useful for both **inferential** and **predictive** purposes

Algorithm I: OLS

What is a linear regression?

an inferential perspective

- a **linear regression function** characterizes the relationship between an independent variable ($\mathbf{X}$) and a dependent variable (Y)

$$Y = E[Y|\mathbf{X}] + \epsilon, \quad E[\epsilon|\mathbf{X}] = 0 \tag{1}$$

- a **conditional mean function** that decomposes Y into a component related to $\mathbf{X}$ and another that is not

$$\begin{aligned} Y &= E[Y|\mathbf{X}] + (Y - E[Y|\mathbf{X}]) \\ &= E[Y|\mathbf{X}] + \epsilon \\ &= \mathbf{X}\beta + \epsilon \end{aligned} \tag{2}$$

- β is found by minimizing $(Y - E[Y|X])^2$

- empirically: what do we get from a regression?

What is a linear regression?

an inferential perspective

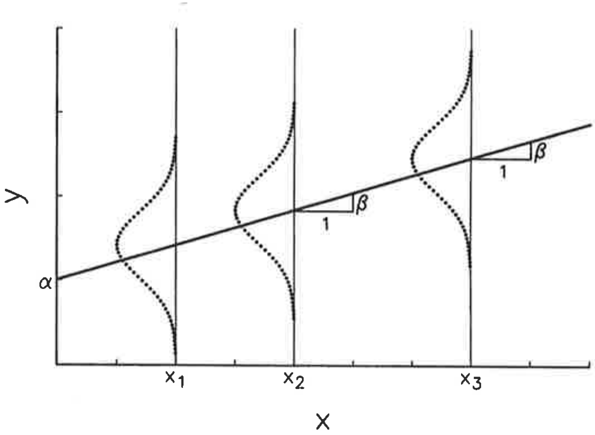


Figure 2.1. Simple Linear Regression Model With the Distribution of y Given x

Figure: Long (1997)

a Gauss-Markov assumptions refresher

1. **linear relationship in the parameters**

$$E[Y|\mathbf{X}] = \beta_1 f_1(\dots) + \beta_2 f_2(\dots) + \dots \beta_k f_k(\dots) + \epsilon$$

   - does not mean a linear relationship in the variables
   - $f_k(\dots)$ can be any transformation of variable k

2. **No <u>linear</u> dependencies in $\mathbf{X}$**

   - no x_k may be described as a **linear function** of other variables in $\mathbf{X}$
   - why would this be problem?

a Gauss-Markov assumptions refresher

3. **Zero conditional mean of ϵ**

$$E[\epsilon|\mathbf{X}] = 0, \quad Cov(\mathbf{X}, \epsilon) = 0$$

   - no x_j has information about $E[\epsilon_k]$
   - why would a violation of this be problem?
   - remember also eqs. (1) and (2)

4. **Spherical errors: conditional homoscedasticity & no autocorrelation**

$$Var(\epsilon|\mathbf{X}) = \sigma_\epsilon^2\mathbf{I}$$

   - disturbances are constant and provide no information about each other
   - why would a violation of this be problem?

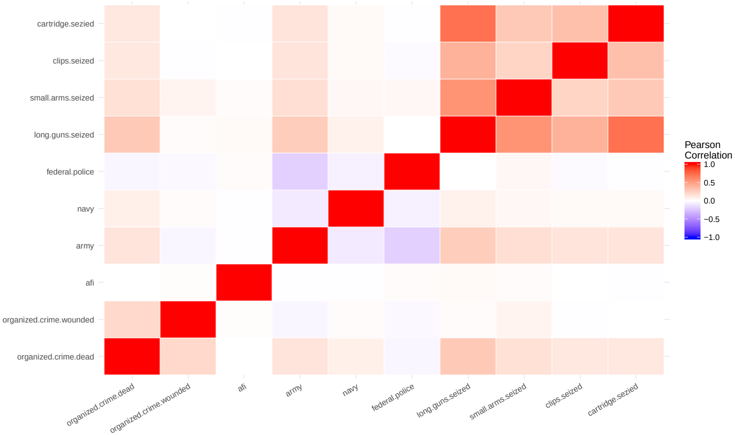
OLS for inference: an example

- suppose we need to better understand dynamics in `organized_crime_dead` and use available data
  - could we extract some causal insights from this data?
  - what could we learn from an OLS algorithm?
- remember: OLS works through a conditional mean function...
  - what does this mean in practice?
  - how generalizable is what we find?

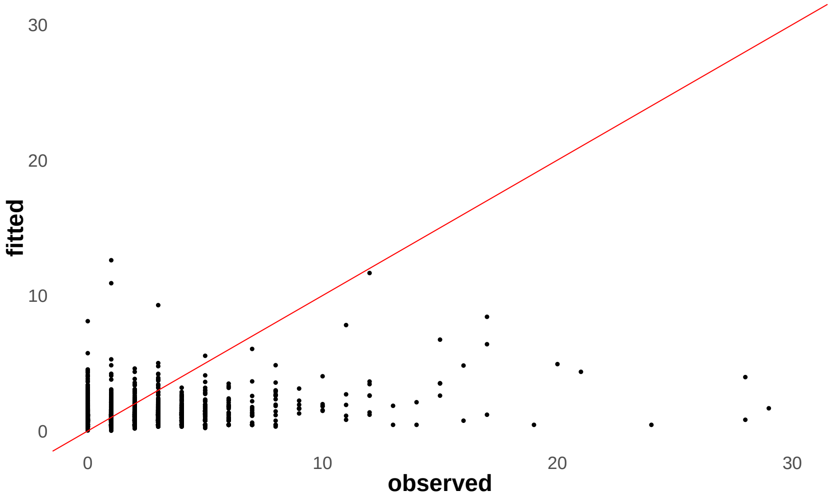
OLS for inference: an example

```
Call:
lm(formula = organized_crime_dead ~ organized_crime_wounded +
    afi + army + navy + federal_police + long_guns_seized + small_arms_seized +
    clips_seized + cartridge_seized, data = AllData)

Residuals:
     Min       1Q   Median       3Q      Max
-11.6058  -0.7274  -0.4506   0.2192  27.3262

Coefficients:
                          Estimate Std. Error t value Pr(>|t|)
(Intercept)              0.4505553  0.0332307  13.558  < 2e-16 ***
organized_crime_wounded  0.3736900  0.0239171  15.624  < 2e-16 ***
afi                     -0.2261752  0.4210396  -0.537   0.5912
army                     0.3066898  0.0532594   5.758 8.96e-09 ***
navy                     0.7150402  0.1389449   5.146 2.75e-07 ***
federal_police          -0.1271515  0.0773309  -1.644   0.1002
long_guns_seized         0.1478424  0.0085972  17.197  < 2e-16 ***
small_arms_seized       -0.0437447  0.0184592  -2.370   0.0178 *
clips_seized             0.0004374  0.0003152   1.388   0.1653
cartridge_seized        -0.0001690  0.0000193  -8.760  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1  ' ' 1

Residual standard error: 1.731 on 5386 degrees of freedom
Multiple R-squared:  0.1413,Adjusted R-squared:  0.1398
F-statistic: 98.44 on 9 and 5386 DF,  p-value: < 2.2e-16
```

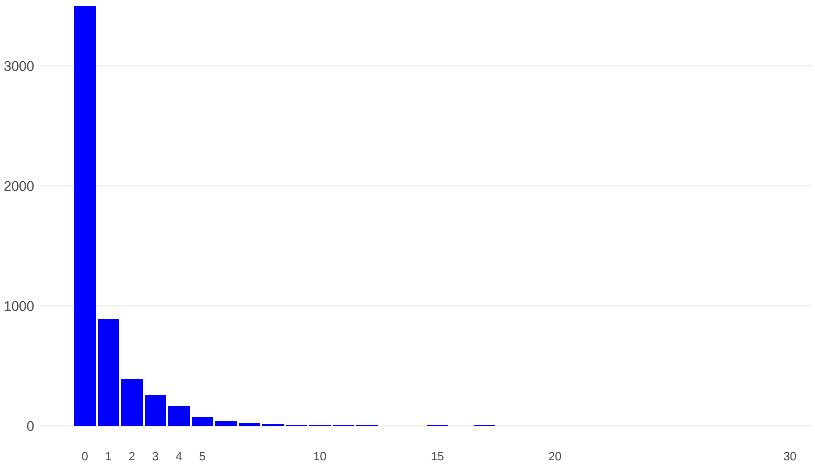
are these "real" results, or just a mirage from reiterated information in our variables?



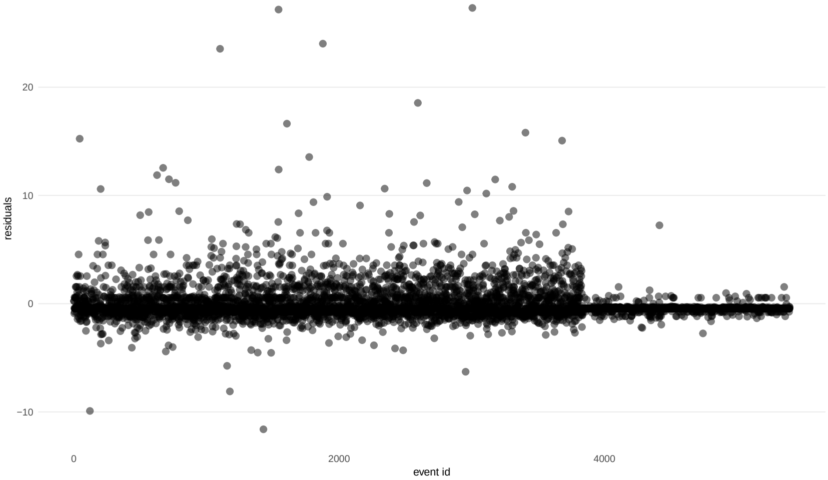
but wait... how does my model fit?



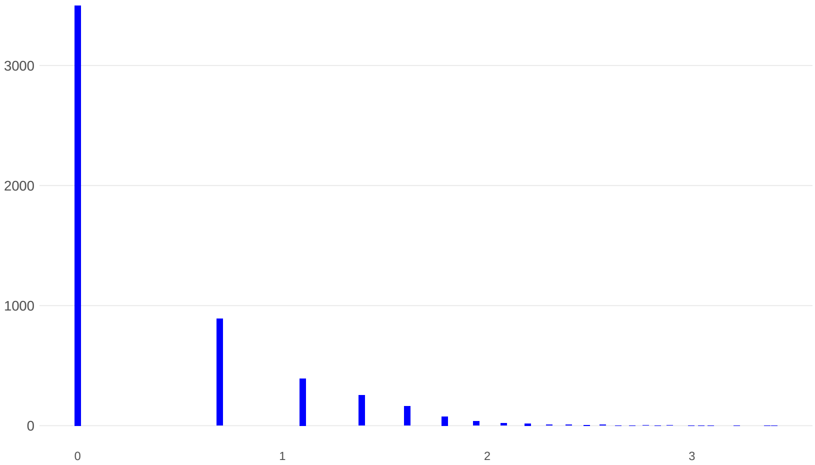
what does my DV look like?



what’s the problem with this?



we can always log it, right?... think again



BLUE is good for inference but bad for prediction

- if Gauss-Markov assumptions are fulfilled, OLS produces the **B**est **L**inear **U**nbiased **E**stimator...
  - which is great for inference... but...
- remember the Hastie et al. (2009) equation?

$$EPE = Var(Y) + Bias^2 + Var(\hat{f}(x))$$

- OLS has little bias ($Bias^2$) but high variance ($Var(\hat{f}(x))$)
  - typically high variance is bad for prediction
  - we may need a tradeoff that increases bias - and reduces variance - to improve prediction

back to the bias-variance tradeoff

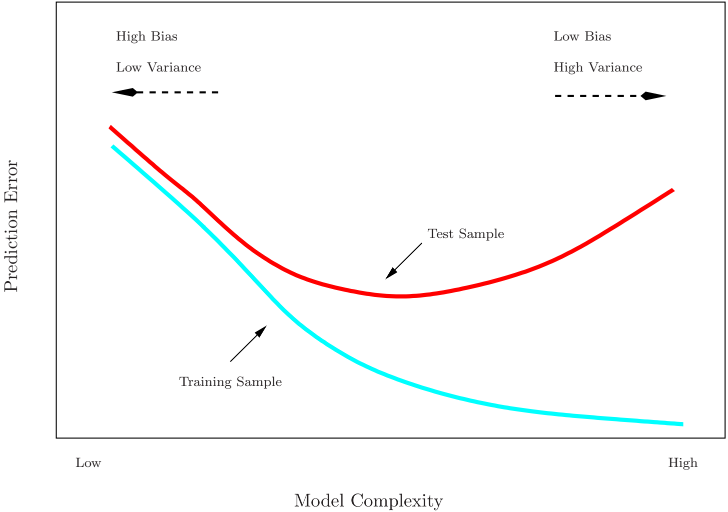


Figure: James et al (2013)

the bias-variance tradeoff in practice

- most important characteristic of a predictive model: **generalization**
- **objective**: optimize bias-variance **tradeoff** to **improve predictons**
- **model selection methods**: constrain [number/estimates] of parameters $k \in \{0, 1, 2, \ldots, p\}$ to minimize expected prediction error
  1. **best subset** (analytical solution criteria)
  2. **(forward-backward) stepwise selection** (analytical solution criteria))
  3. **cross-validation** (cross-validation prediction error)
  4. **shrinkage** (analytical solution criteria)
- we'll review examples of 1 and 3

best subset selection in practice

- **best subset selection** searches for the minimal optimal combination of variables that **minimize expected prediction error**
- **best subset selection algorithm**:
  1. fit a null model $\mathcal{M}_0$
  2. for each $k = 1, 2, \ldots, p$, fit all $\binom{p}{k}$ models that contain k predictors (on the training data), and pick the one with the **lowest train error** among them $\mathcal{M}_k$
  3. select the model with the **lowest prediction error** among $\mathcal{M}_0, \ldots, \mathcal{M}_p$

best subset selection in practice

- **best subset selection** relies on different criteria to select the “best” subset model
  - **step 2** selects models with the lowest **train error**:
    - determined by a low residual sum of squares (RSS)
  - **step 3** selects the model with the lowest **test error**:
    - **indirectly**, with a statistic that “adjusts” the train error with a penalty for the number of variables in the model: BIC, AIC, C_p, etc
    - **directly**, by cross-validation

AIC and BIC “chose” OLS regressions with different parameters subsets as best for prediction

Best Subset selection using AIC

```
##
## Call:
## lm(formula = y ~ ., data = data.frame(Xy[, c(bestset[-1], FALSE),
##     drop = FALSE], y = y))
##
## Coefficients:
##            (Intercept)  organized_crime_wounded         long_guns_seized
##              0.4498740                0.3730898                0.1500302
##      small_arms_seized         cartridge_sezied                     army
##             -0.0434190               -0.0001668                0.3097144
##         federal_police                     navy
##             -0.1296465                0.7166220
```

Best Subset selection using BIC

```
##
## Call:
## lm(formula = y ~ ., data = data.frame(Xy[, c(bestset[-1], FALSE),
##     drop = FALSE], y = y))
##
## Coefficients:
##            (Intercept)  organized_crime_wounded         long_guns_seized
##              0.4237166                0.3713140                0.1389487
##       cartridge_sezied                     army                     navy
##             -0.0001567                0.3263833                0.7347481
```

cross-validation in practice

- **cross-validation** combines a holdout and resampling to estimate a model's **expected prediction error**

- **cross-validation algorithm:**
  1. randomly split the data into K folds

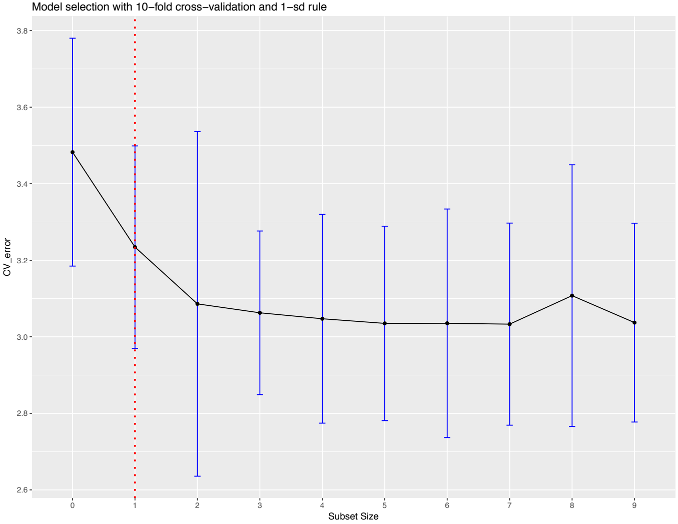
1	2	3	4	5
Train	Train	Validation	Train	Train

  2. fit the model on all but the $k-th$ fold
  3. compute the **prediction error** by predicting the $k-th$ fold
  4. iterate steps 2 and 3 over $k = 1, \ldots, K$
  5. average over K prediction errors to compute the **cross-validation prediction error**

cross-validation in practice

- in practice, $K = 5$ or $K = 10$ provides a good estimate of the **expected prediction error**
- cross-validation could serve two different-but-related purposes
  - estimate a single model's **expected prediction error**
  - produce an **estimated prediction error curve** where its lower level can be identified to compare different algorithms or different levels of flexibility in a single algorithm
- **one-standard deviation rule:** choose the simplest model with an error within one standard deviation of the minimal error model

10-fold cross-validation identified a 1-parameter OLS regression as best for prediction in our example



10-fold cross-validation identified a 1-parameter OLS regression as best for prediction in our example

```
##
## Call:
## lm(formula = y ~ ., data = data.frame(Xy[, c(bestset[-1], FALSE),
##     drop = FALSE], y = y))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -15.4137  -0.6698  -0.6698   0.3302  27.6742
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)      0.669800   0.025822   25.94   <2e-16 ***
## long_guns_seized 0.109332   0.005091   21.47   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.791 on 5394 degrees of freedom
## Multiple R-squared:  0.07876,    Adjusted R-squared:  0.07859
## F-statistic: 461.2 on 1 and 5394 DF,  p-value: < 2.2e-16
```

Algorithm II: Logistic Regression

What is a logistic regression?

an inferential perspective

- different question: **did something happen or not?**
  - essentially, binary outcome classification
  - why not just use OLS?
- one way to think about this: let y^* be a continuous (latent) variable

$$y^* = x\beta + \epsilon$$

- for which we only observe two outcomes

$$y_i = \begin{cases} 1 & \textit{if} \quad y_i^* > \tau \\ 0 & \textit{if} \quad y_i^* \leq \tau \end{cases}$$

What is a logistic regression?

an inferential perspective

- we're interested in the probability that $y = 1$

$$\pi_i = Pr(y = 1) = F(\beta x)$$

- in the case of a logit, we estimate

$$\pi_i = \Lambda(\beta x) = \frac{e^{\beta x}}{1 + e^{\beta x}}$$

- but there's also additional "flavors" (i.e. probit)

a logistic regression assumptions refresher

1. **linear relationship between parameters**

$$\pi_i = F(\beta_1 f_1(\dots) + \beta_2 f_2(\dots) + \dots \beta_k f_k(\dots) + \epsilon_i)$$

   - does not mean a linear relationship in the variables
   - $f_k(\dots)$ can be any transformation of variable k

2. **no <u>linear</u> dependencies in $\mathbf{X}$**
   - no x_k may be described as a **linear function** of other variables in $\mathbf{X}$
   - why would this be a problem?

a logistic regression assumptions refresher

3. **no autocorrelation**

$$Cov(\epsilon_i, \epsilon_j) = 0; \;\; \forall\, i \neq j$$

   - why would a violation of this be a problem?

4. **a balanced sample in Y**

   - what does this mean?
   - why would a violation of this be a problem?

logistic regression for inference: back to our example

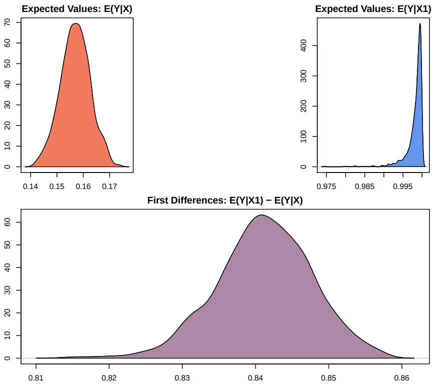
- we have a natural dual category: **events with deaths / no deaths**
- **could we learn something about correlates to events with organized crime deaths?**
  - we have information on federal forces involved
  - also on materiel seizures
- **can this relationship ever be causal?**

logistic regression for inference: back to our example

```
Call:
glm(formula = organized_crime_death ~ organized_crime_wounded +
    afi + army + navy + federal_police + long_guns_seized + small_arms_seized +
    clips_seized + cartridge_sezied, family = binomial(link = "logit"),
    data = AllData)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-4.5396  -0.6657  -0.4731  -0.4592   2.7612

Coefficients:
                          Estimate Std. Error z value Pr(>|z|)
(Intercept)             -2.1337831  0.0599578 -35.588  < 2e-16 ***
organized_crime_wounded  0.2839835  0.0376519   7.542 4.62e-14 ***
afi                     -0.6960636  0.7234004  -0.962    0.336
army                     0.7395036  0.0812191   9.105  < 2e-16 ***
navy                     0.9292565  0.1827726   5.084 3.69e-07 ***
federal_police          -0.0628413  0.1331772  -0.472    0.637
long_guns_seized         0.1544432  0.0141145  10.942  < 2e-16 ***
small_arms_seized       -0.0137429  0.0271923  -0.505    0.613
clips_seized            -0.0004430  0.0004284  -1.034    0.301
cartridge_sezied        -0.0002413  0.0000510  -4.730 2.25e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1  ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 5185.2  on 5395  degrees of freedom
Residual deviance: 4721.3  on 5386  degrees of freedom
AIC: 4741.3
```

we'd want to translate estimated coefficients into meaningful insights

- let's look at the change in probablity between `organized_crime_wounded == 0` (X) and `organized_crime_wounded == 30` (X1)



but before that, we want to assess model fit which is a bit more complicated for LDV models

i) **likelihood-based approaches**: assess the likelihood that a model produced the observed (sample) data

- **focus:** a model's log-likelihood, transformed to produce specific test statistics: likelihood-ratio test, AIC, etc
- well suited to **choose among different models**

ii) **classification-based approaches**: assess how good a model is at classifying cases

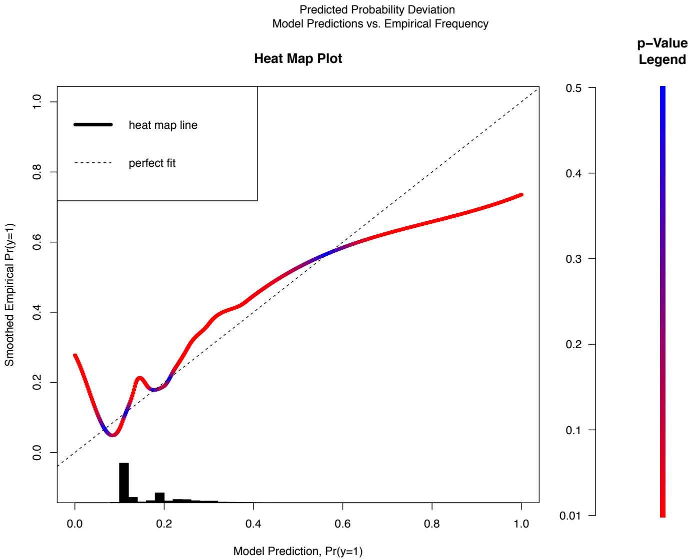
- **focus:** difference between estimated outcomes ($\hat{y}$) and observed outcomes (y)
- percent correctly predicted (PCP), receiver operating characteristic (ROC)

but before that, we want to assess model fit which is a bit more complicated for LDV models

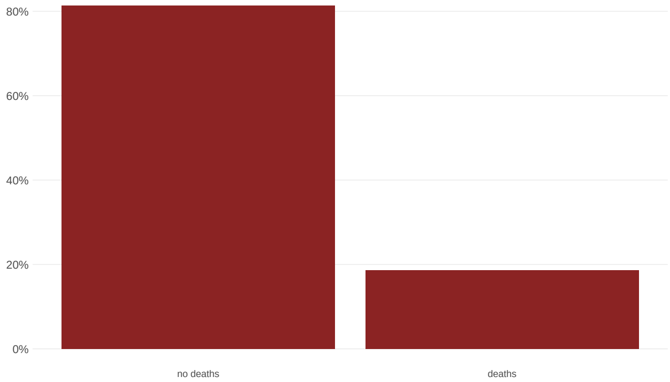
iii) **probablity-based approaches**: assess how good a model is at generating estimated probabilities

- LDV models estimate a probability of an event happening ($\hat{p} \in [0, 1]$) that is converted to a binary outcome ($\hat{y} = \{0, 1\}$) when a threshold ($\tau \in [0, 1]$) is overcome
- **focus:** difference between estimated probabilities ($\hat{p}$) and some empirical probability ($R(\hat{p})$)
- heat map plot and statistic

sadly, our model has a terrible fit!



wait again, what does my DV look like?



- what does your "plain vanilla" logistic regression assume?

onto prediction: AIC and BIC “chose” logistic regressions with same parameter subsets

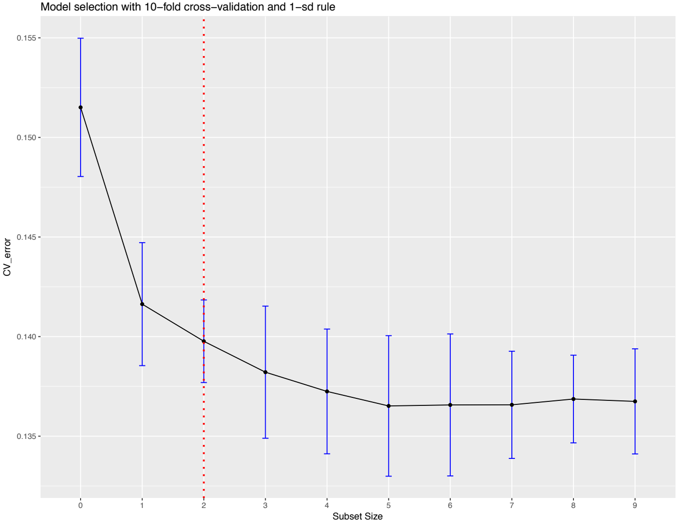
Best Subset Selection (AIC)

```
##
## Call:  glm(formula = y ~ ., family = family, data = Xi, weights = weights)
##
## Coefficients:
##            (Intercept)  organized_crime_wounded         long_guns_seized
##             -2.1465619                0.2831332                0.1479253
##       cartridge_sezied                     army                     navy
##             -0.0002407                0.7477216                0.9415283
##
## Degrees of Freedom: 5395 Total (i.e. Null);  5390 Residual
## Null Deviance:       5185
## Residual Deviance: 4724  AIC: 4736
```

Best Subset Selection (BIC)


```
##
## Call:  glm(formula = y ~ ., family = family, data = Xi, weights = weights)
##
## Coefficients:
##            (Intercept)  organized_crime_wounded         long_guns_seized
##             -2.1465619                0.2831332                0.1479253
##       cartridge_sezied                     army                     navy
##             -0.0002407                0.7477216                0.9415283
##
## Degrees of Freedom: 5395 Total (i.e. Null);  5390 Residual
## Null Deviance:       5185
## Residual Deviance: 4724  AIC: 4736
```

10-fold cross-validation identified a 2-parameter logistic regression as best for prediction



10-fold cross-validation identified a 2-parameter logistic regression as best for prediction

```
##
## Call:
## glm(formula = y ~ ., family = family, data = data.frame(Xy[,
##     c(bestset[-1], FALSE), drop = FALSE], y = y))
##
## Deviance Residuals:
##    Min       1Q   Median       3Q      Max
## -5.339  -0.690  -0.514  -0.514   2.044
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)     -1.957360   0.050712 -38.598   <2e-16 ***
## long_guns_seized 0.108097   0.009482  11.400   <2e-16 ***
## army             0.643469   0.076039   8.462   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 5185.2  on 5395  degrees of freedom
## Residual deviance: 4859.9  on 5393  degrees of freedom
## AIC: 4865.9
##
## Number of Fisher Scoring iterations: 4
```

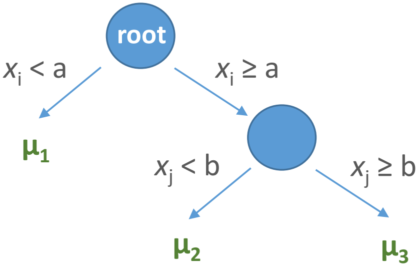
Algorithm III: Random Forests

Let's start with a single tree model

$$Y = g(x; T, M) + \epsilon$$

where

- T : a tree structure (decision rules, internal and terminal nodes)
- $M = \{\mu_1, \mu_2, ..., \mu_b\}$: set of terminal node μ's
- $g(x; T, M)$: the function that assigns a μ to x



What are random forests?

the predictive approach

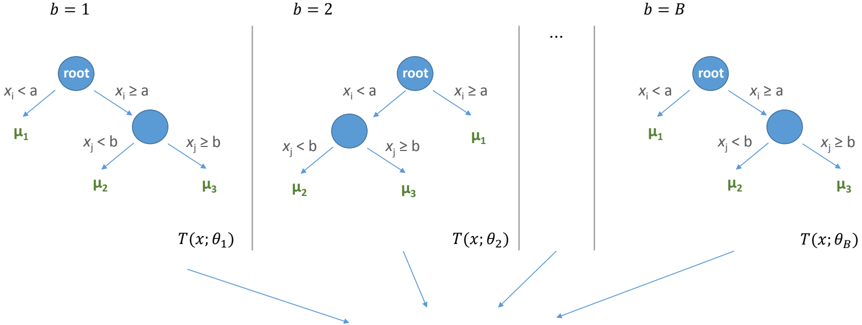
- **random forests**: tree-based algorithm (for prediction) with **regression** and **classification** flavors
  - **forests** because the algorithm uses many trees
  - **random** because the algorithm selects features randomly
- important **bias-variance consequences** from these characteristics:
  - **trees** tend to have **low bias** as depth increases
  - **random selection** of features **de-correlates** trees
  - **averaging** over de-correlated trees **reduces variance**

the random forests algorithm

Hastie et al. (2009)

1. for $b = 1$ to B:
   (a) draw a bootstrap sample $\mathbf{Z}^*$ of size N from the training data.
   (b) grow a random-forest tree T_b to the bootstrapped data, by recursively repeating the following steps for each terminal node of the tree, until the minimum node size n_{min} is reached.
      i. select m variables at random from the p variables.
      ii. pick the best variable/split-point among the m.
      iii. split the node into two daughter nodes.
2. output the ensemble of trees $\{T_b\}_1^B$.

the random forests algorithm



$$\{T_b\}_1^B = \frac{1}{B} \sum_{b=1}^{B} T(x; \theta_b)$$

what makes random forests such a special algorithm?

- **Assumptions:**
  - no distributional assumptions
  - no linear relationship in parameters assumption
- **Advantages:**
  - works for regression and classification problems
  - uses categorical features (variables)"naturally"
  - detects "important" variables and selects them
  - handles non-linear interactions and boundaries
  - performs cross-validation on the fly
  - (under certain conditions) not too prone to overfitting

random forests for prediction: back to our example

- **could we learn something about predictors of organized crime deaths?**
  - we have information on a number of predictors
  - perhaps thinking of this problem as trees may help

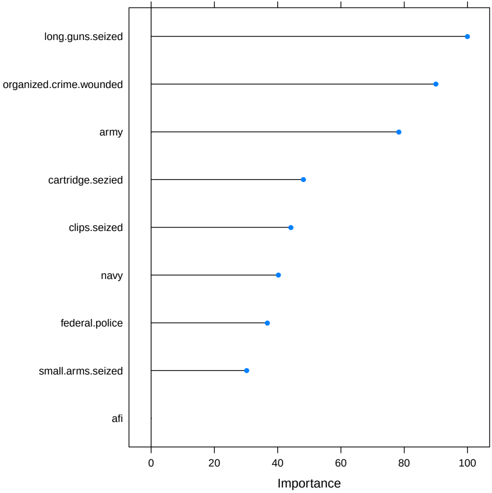
random forests for prediction: back to our example

```
Call:
 randomForest(formula = organized_crime_dead ~ organized_crime_wounded +
                                afi + army + navy + federal_police +
                                long_guns_seized + small_arms_seized +
                                clips_seized + cartridge_sezied,
                                data = training, method = "rf",
                                importance = TRUE,
                                prox = TRUE,
                                preProc = c("center", "scale"))
               Type of random forest: regression
                     Number of trees: 500
No. of variables tried at each split: 3

          Mean of squared residuals: 3.275263
                    % Var explained: 11.8
```

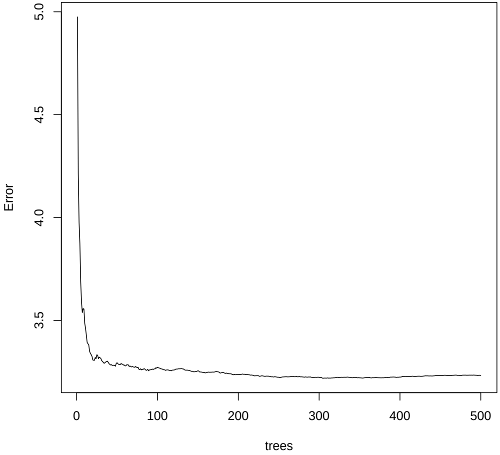
10-fold cross-validation confirms that using nearly all nine predictors produces the least error

```
       9        8        7        6        5        4        3        2        1
3.270985 3.219273 3.231779 3.244063 3.271915 3.446298 3.377251 3.483434 3.485249
```

what does variance importance tell us?



a quick look at MSE for this model by number of trees



What did we learn from these algorithms in an inferential context?

- if our **inferential models** were correctly specified
- the number of organized crime deaths (**OLS**) and the likelihood of observing a death among organized crime members (**logistic regression**) tend to be higher in events where:
  - the **navy** or **army** participate
  - **organized crime wounded** exist
  - **long guns** and **catridges** are seized

What did we learn from these algorithms in a predictive context?

- the best **predictors** of the number of organized crime deaths (**OLS**) are:
  - the number of **organized crime wounded**, the participation of armed forces (**army**, **navy**, **federal police**), and the seizure of **long guns**, **small arms** and **cartridges** (AIC)
  - the number of **organized crime wounded**, the participation of **army** and **navy**, and the seizure of **long guns** and **cartridges** (BIC)
  - the number of **long guns seized** (cross-validation)

What did we learn from these algorithms in a predictive context?

- the best **predictors** of the existence of at least one organized crime death (**logistic regression**) are:
  - the number of **organized crime wounded**, the participation of **army** or **navy**, and the seizure of **long guns** or **cartridges** (AIC, BIC)
  - the participation of the **army** and the seizure of **long guns** (cross-validation)
- the best **predictors** of the number of deaths among organized crime (**random forests**) are:
  - the presence of seized **long guns**, **organized crime wounded**, and the participation of the **army**

Three Algorithms: go-to tools in the tooolboox

Marco Morales
marco.morales@columbia.edu

GR5069
Topics in Applied Data Science
for Social Scientists

Spring 2019
Columbia University