Effect sizes from predictive models

Abhijit Dasgupta, PhD

ARAASTAT

@webbedfeet

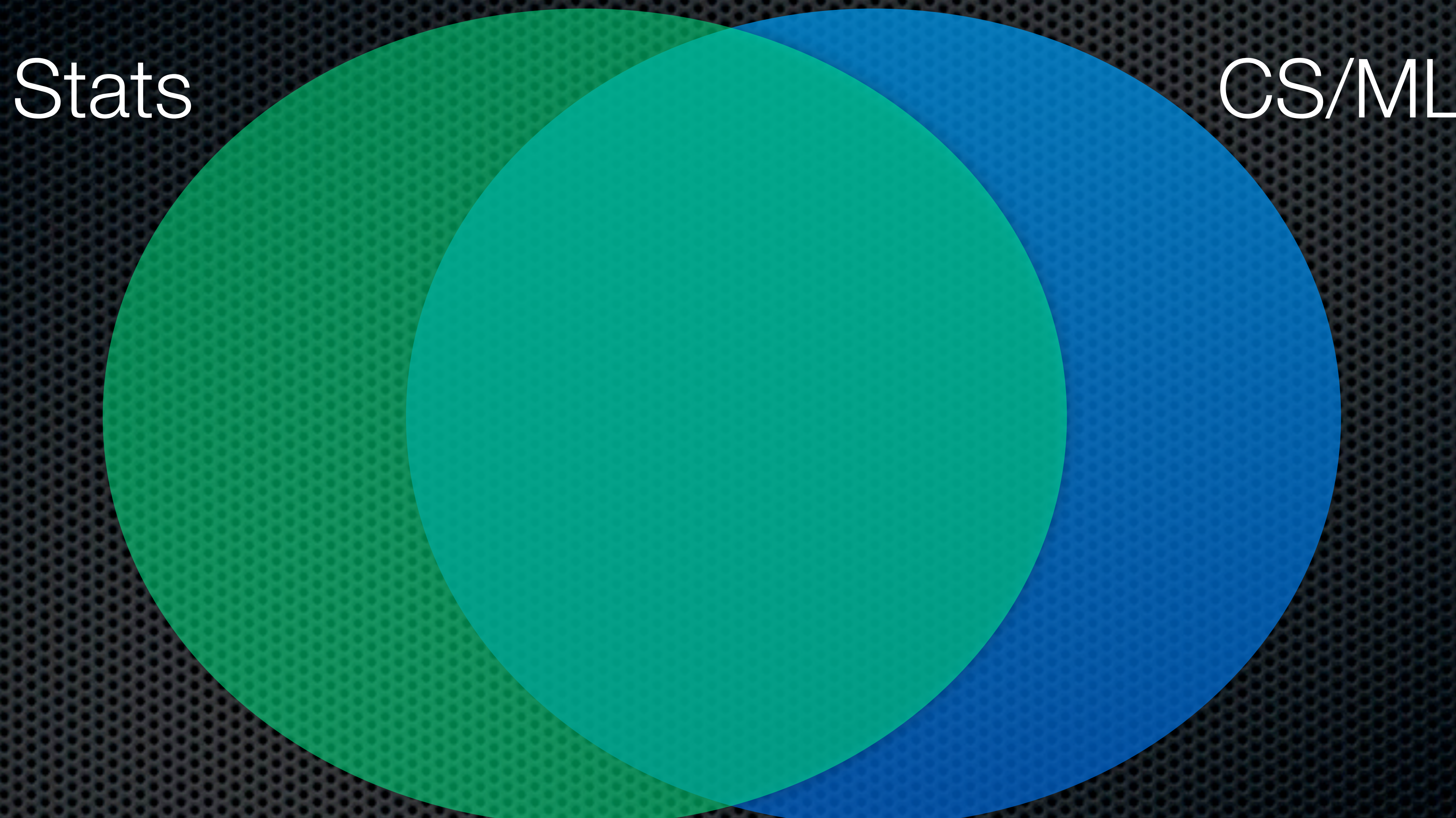
statbandit.wordpress.com

Statistics

- Parametric models
- Emphasis on explanation or “causality”
- Often hypothesis-driven
- End product is often a series of p-values

CS/ML

- Nonparametric models
- Emphasis on prediction
- Often hypothesis-generating
- End product is a set of predictions


Stats
CS/ML


Stats
CS/ML
I play here

We’ve trained our clients too well

Effect sizes and p-values

- In a supervised learning setting, demands are often
  - What is the effect of a feature?
  - Is it significant?
- The question is not
  - How well can I predict the outcome in new data?

Effect sizes and p-values

- In a supervised learning setting, demands are often
  - What is the effect of a feature?
  - Is it significant?

Associative

- The question is not
  - How well can I predict the outcome in new data?

Predictive

My experience/belief/faith

- Many good machine learning algorithms reflect patterns in the data very well, capturing structure in a “black box”
- Classical parametric models are limited by requirements of explicit structural specification and resulting mis-specification
- This affects both the associative and predictive views

Prediction

- Associative models are used for prediction
  - Linear regression
  - Logistic regression
  - Structural equations

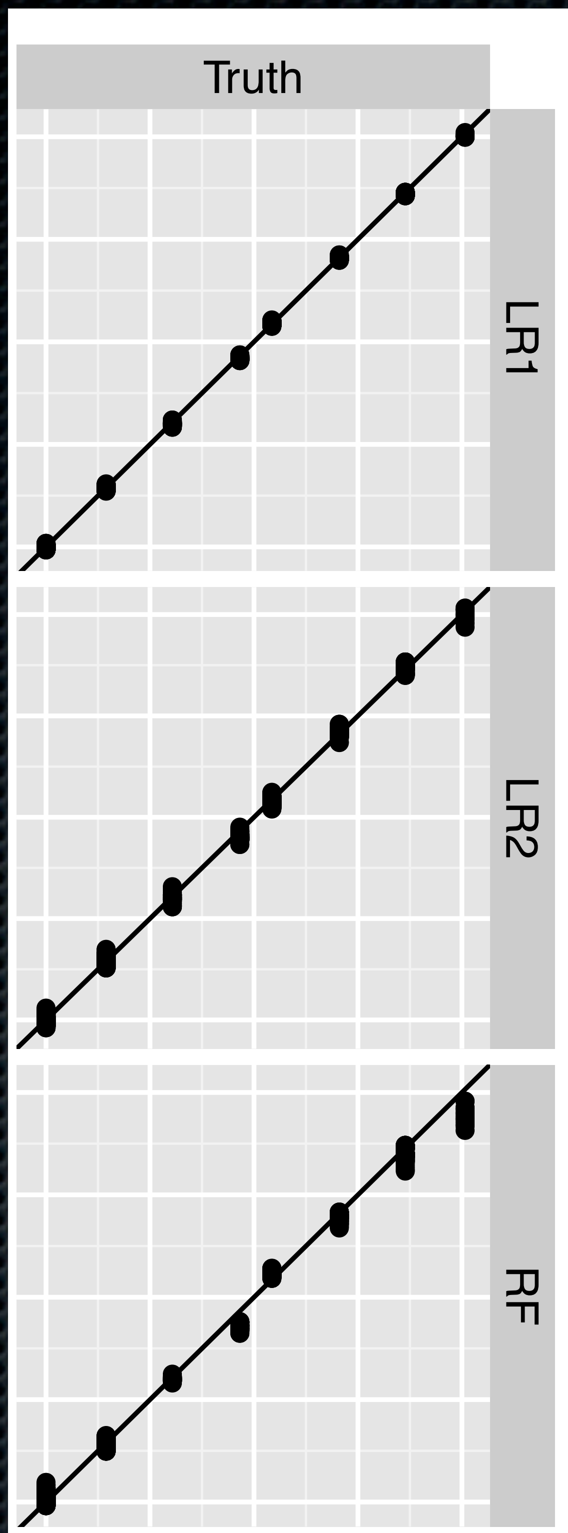
Prediction

- Associative models are used for prediction
  - Linear regression
  - Logistic regression
  - Structural equations

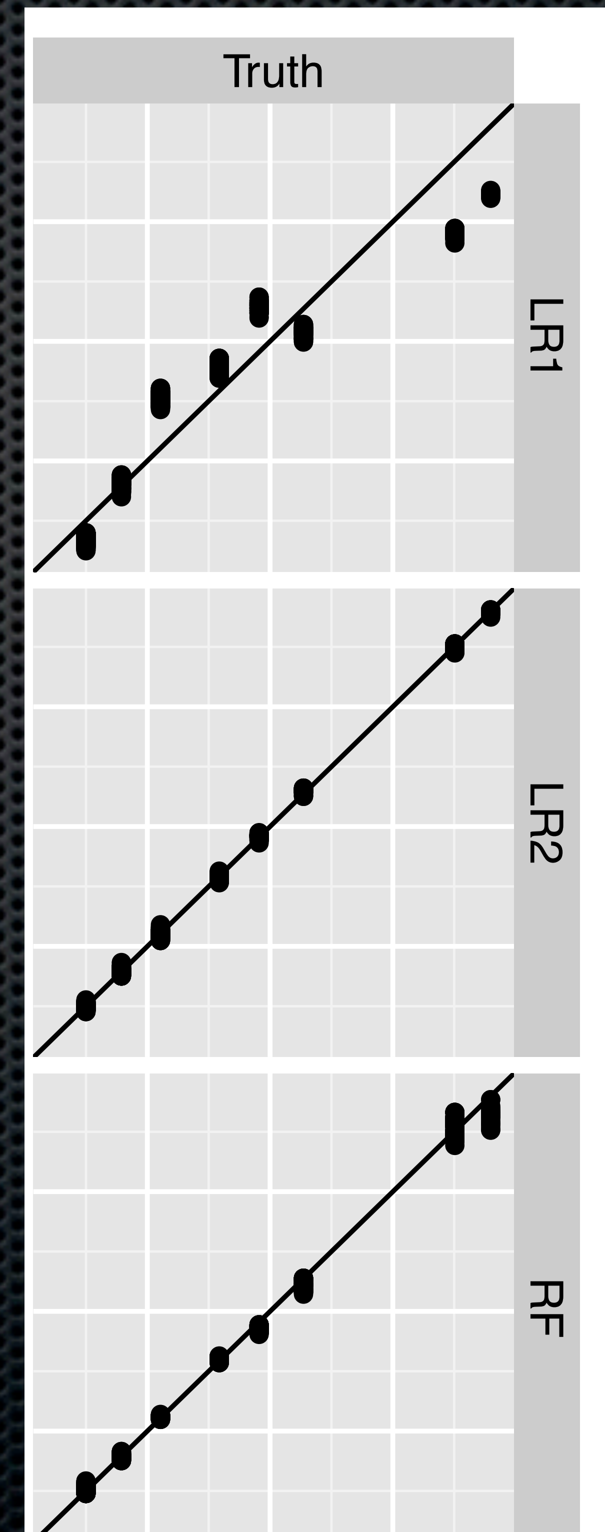
They require more careful model building to do well

Truth

No interactions



Interactions



Main effects only

$$y=\beta_0+\beta_1 x_1+\beta_2 x_2+\ldots$$

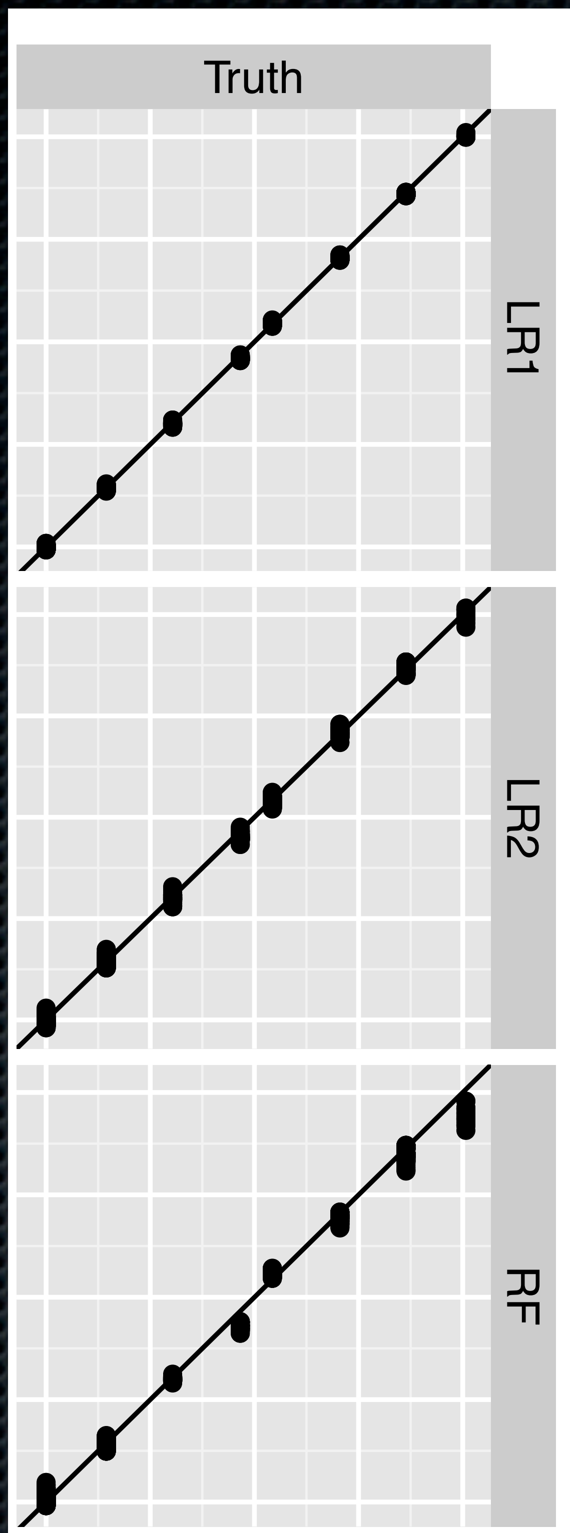
Main effects + interactions

$$y=\beta_0+\beta_1 x_1+\beta_2 x_2+\ldots+\beta_{12} x_1 x_2+\ldots$$

Same specification

Truth

No interactions



Main effects only

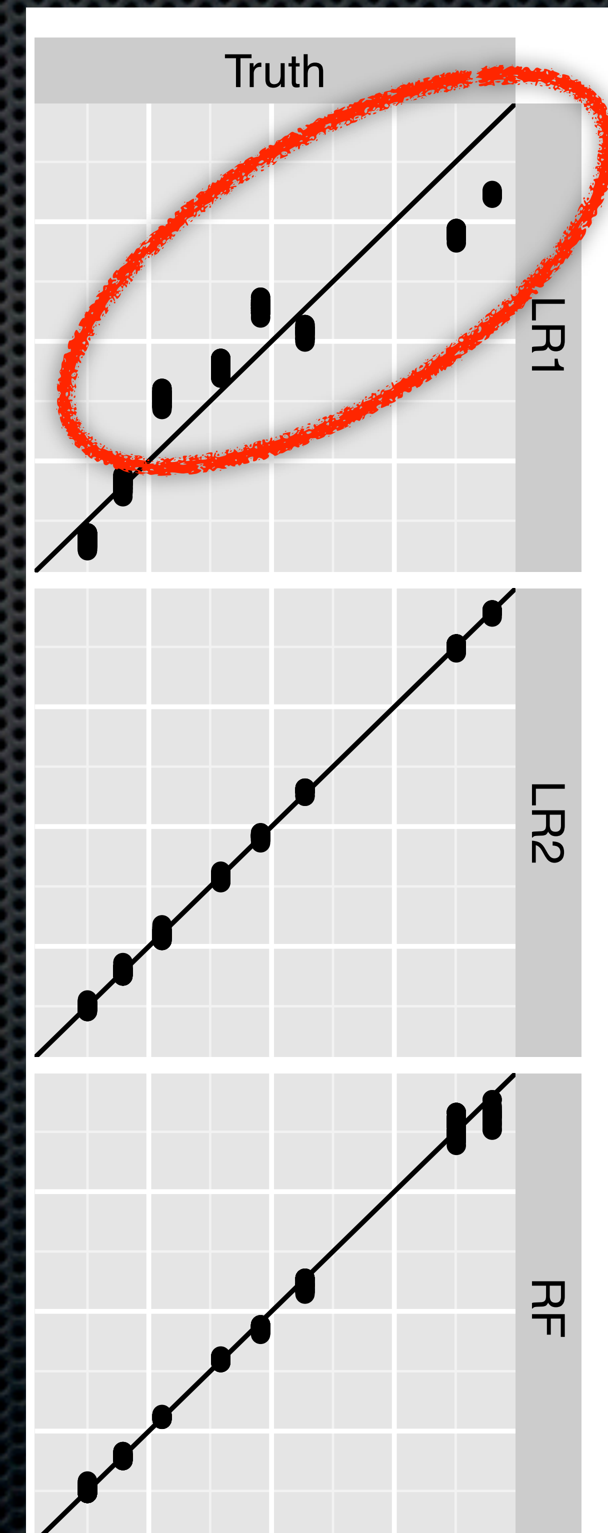
$$y=\beta_0+\beta_1 x_1+\beta_2 x_2+\ldots$$

Main effects + interactions

$$y=\beta_0+\beta_1 x_1+\beta_2 x_2+\ldots+\beta_{12} x_1 x_2+\ldots$$

Same specification

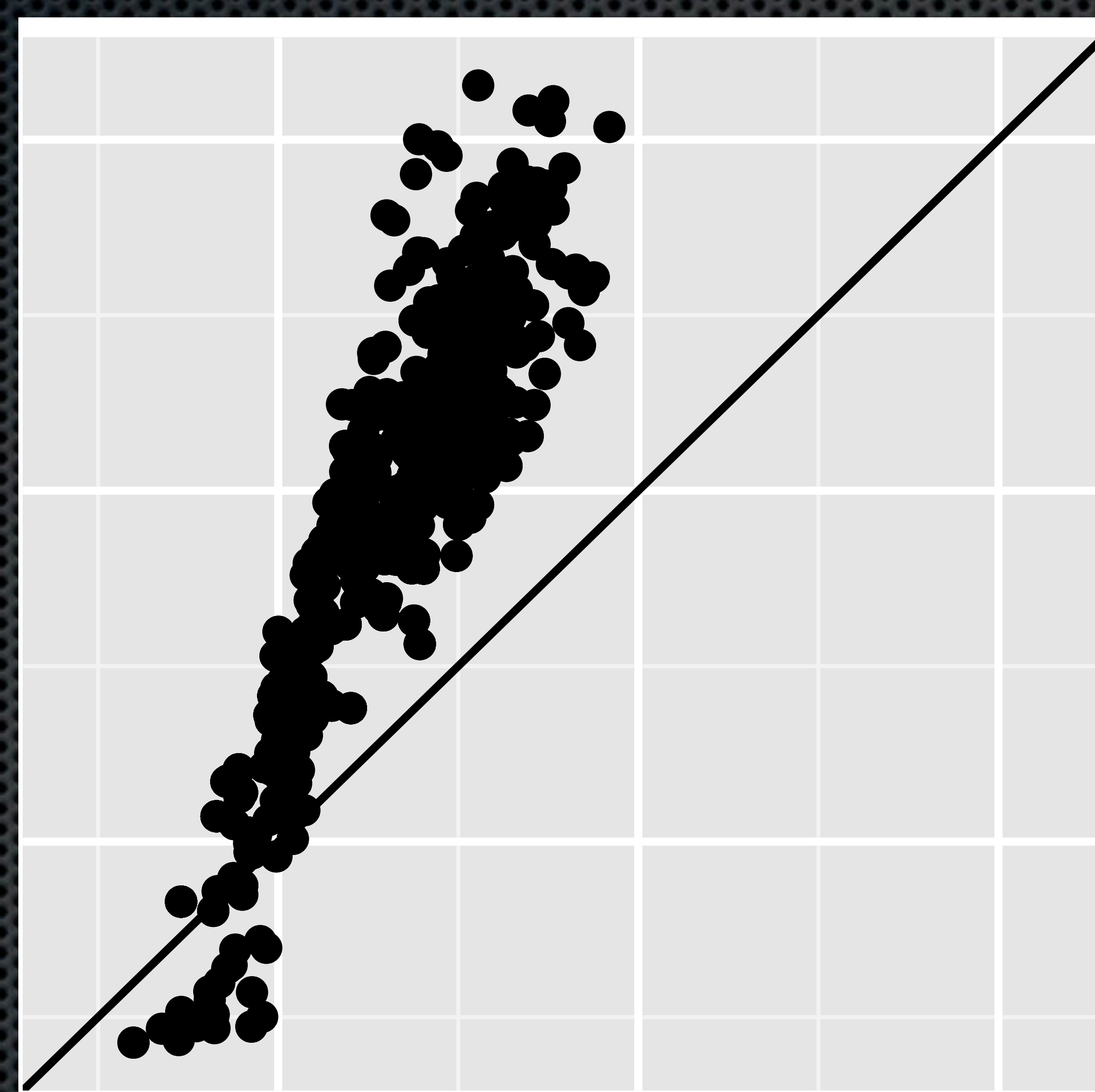
Interactions



Width of prediction intervals

Main effects + interactions

$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_{12} x_1 x_2 + \ldots$



Random Forest

The point is….

- With “black box” models you don’t have to worry about data structure explicitly, as long as you trust your “black box”
- With classical models, you spend a lot of time worrying about whether you got close enough to the data structure
- Often we don’t worry enough

The point is....

- With “black box” models you don’t have to worry about data structure explicitly, as long as you trust your “black box”
- With classical models, you spend a lot of time worrying about whether you got close enough to the data structure
- Often we don’t worry enough → CONSEQUENCES

But you can’t tell me how my favorite feature affects the outcome

But you can't tell me how my favorite feature affects the outcome

Whine, whine, whine!!

But you can't tell me how my favorite feature affects the outcome

Whine, whine, whine!!

Like some cheese with that?

Like some cheese with that?

A trip down memory lane ….

Regression

$$y = b_0 + b_1x_1 + b_2x_2 + \ldots + \text{error}$$

Regression

$$y = b_0 + b_1 x_1 + b_2 x_2 + \ldots + \text{error}$$

Regression

$$y = b_0 + b_1 x_1 + b_2 x_2 + \ldots + \text{error}$$

b_1 is the **average** change in y

Regression

$$y = b_0 + b_1 x_1 + b_2 x_2 + \ldots + \text{error}$$

b_1 is the **average** change in y

when x_1 changes by 1 unit

Regression

$$y = b_0 + b_1 x_1 + b_2 x_2 + \ldots + \text{error}$$

b_1 is the **average** change in y

when x_1 changes by 1 unit

everything else remaining the same

Regression

$$y = b_0 + b_1 x_1 + b_2 x_2 + \ldots + \text{error}$$

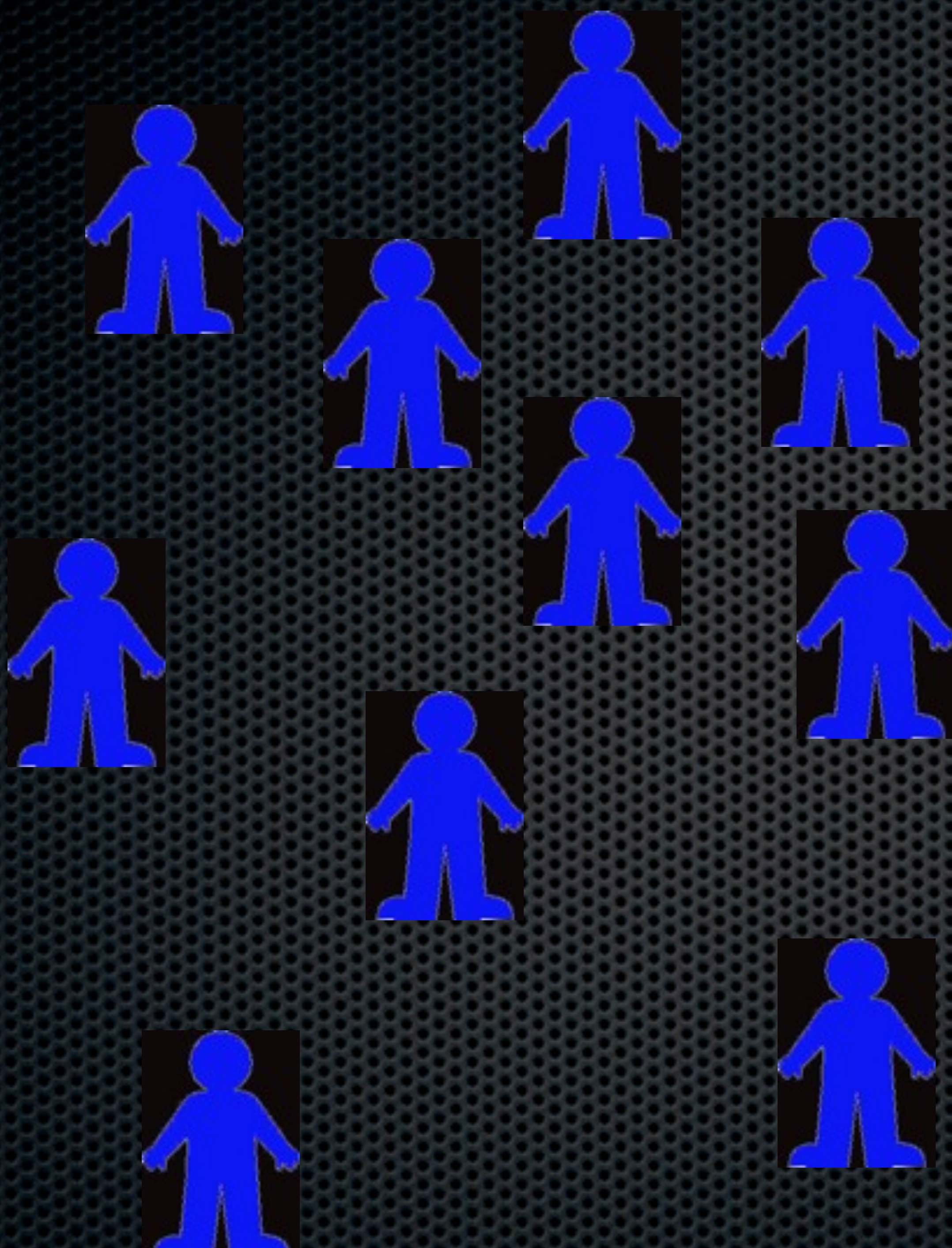
b_1 is the **average** change in y

when x_1 changes by 1 unit

everything else remaining the same

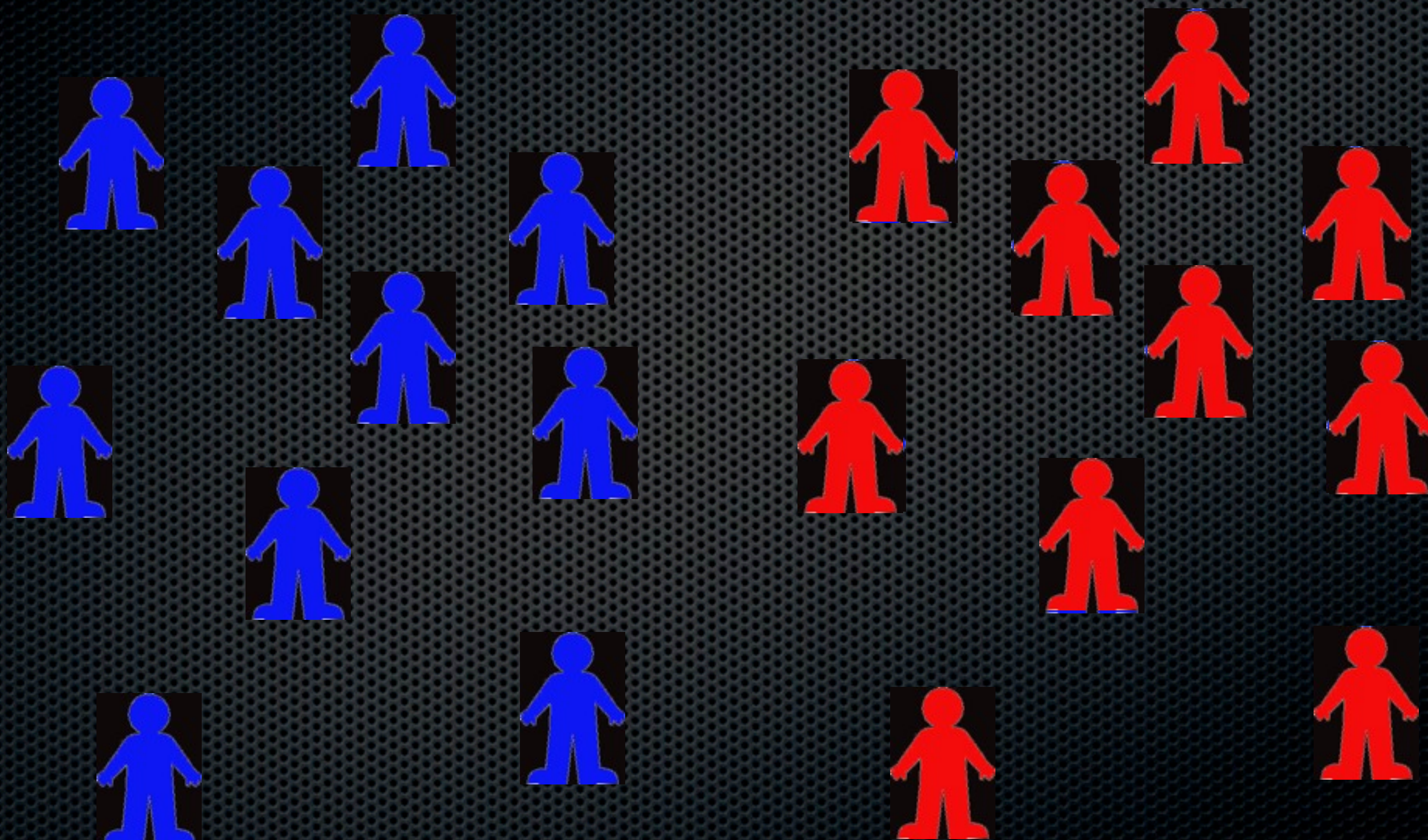
Counterfactual argument

The counterfactual argument



The counterfactual argument

The counterfactual argument



The counterfactual argument

The counterfactual argument

The counterfactual argument

The counterfactual argument

Can we predict outcome if we change environments?

Using predictive models

Counterfactuals as predictions

Counterfactuals as predictions

- Consider a binary feature (red/blue)

Counterfactuals as predictions

- Consider a binary feature (red/blue)
- Train a model only on the reds (Red model)
  - This captures the red environment

Counterfactuals as predictions

- Consider a binary feature (red/blue)
- Train a model only on the reds (Red model)
  - This captures the red environment
- Suppose you are blue, and have a true outcome y_t

Counterfactuals as predictions

- Consider a binary feature (red/blue)
- Train a model only on the reds (Red model)
  - This captures the red environment
- Suppose you are blue, and have a true outcome y_t
- If we drop you down the Red model, it would predict your outcome as if you were red (y_p)

Counterfactuals as predictions

- Consider a binary feature (red/blue)
- Train a model only on the reds (Red model)
  - This captures the red environment
- Suppose you are blue, and have a true outcome y_t
- If we drop you down the Red model, it would predict your outcome as if you were red (y_p)
- Your counterfactual effect would be $y_t - y_p$

Counterfactuals as prediction

Counterfactuals as prediction

- Train a Red model and a Blue model (same algorithm)
  - Red model trains only on Reds
  - Blue model trains only on Blues

Counterfactuals as prediction

- Train a Red model and a Blue model (same algorithm)
  - Red model trains only on Reds
  - Blue model trains only on Blues
- Send everyone down each model

Counterfactuals as prediction

- Train a Red model and a Blue model (same algorithm)
  - Red model trains only on Reds
  - Blue model trains only on Blues
- Send everyone down each model
- Everyone now has a Red prediction (y_R) and a Blue prediction (y_B)

Counterfactuals as prediction

- Train a Red model and a Blue model (same algorithm)
  - Red model trains only on Reds
  - Blue model trains only on Blues
- Send everyone down each model
- Everyone now has a Red prediction (y_R) and a Blue prediction (y_B)

We're calling this "counterfactual machines"

Counterfactuals as prediction

Counterfactuals as prediction

- Since the Red and Blue models are trained on just Reds and Blues

Counterfactuals as prediction

- Since the Red and Blue models are trained on just Reds and Blues
  - the models don't actually use the Red/Blue variable for prediction

Counterfactuals as prediction

- Since the Red and Blue models are trained on just Reds and Blues
  - the models don't actually use the Red/Blue variable for prediction
  - What these models capture is **everything else** about being Red and Blue

Counterfactuals as prediction

- Since the Red and Blue models are trained on just Reds and Blues
  - the models don't actually use the Red/Blue variable for prediction
  - What these models capture is **everything else** about being Red and Blue
    - Obvious patterns and intangibles

Counterfactuals as prediction

- Since the Red and Blue models are trained on just Reds and Blues
  - the models don't actually use the Red/Blue variable for prediction
  - What these models capture is **everything else** about being Red and Blue
    - Obvious patterns and intangibles
    - The Red and Blue "environment"

Counterfactuals to effect size

- If you want an overall effect measure of Red vs Blue **conditional on everything else**

Average (mean/median) over y_R-y_B

- If you want a subgroup-specific effect measure of Red vs Blue **conditional on everything else**

Average y_R-y_B only over the subgroup

Counterfactuals to effect size

- If you want an overall effect measure of Red vs Blue **conditional on everything else**

Average (mean/median) over y_R-y_B

- If you want a subgroup-specific effect measure of Red vs Blue **conditional on everything else**

Average y_R-y_B only over the subgroup

Try doing this in classical regression

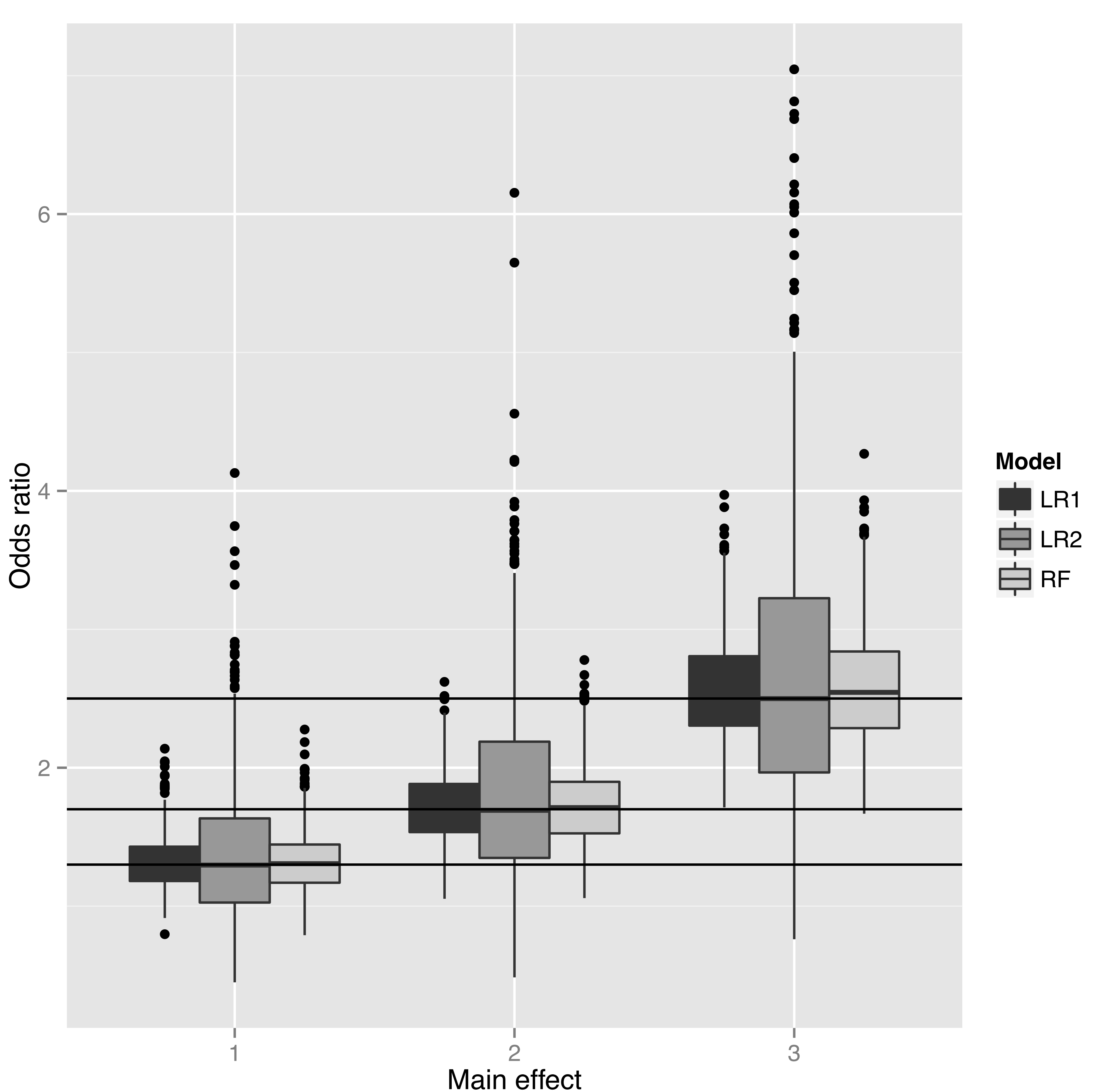
Counterfactuals to effect size

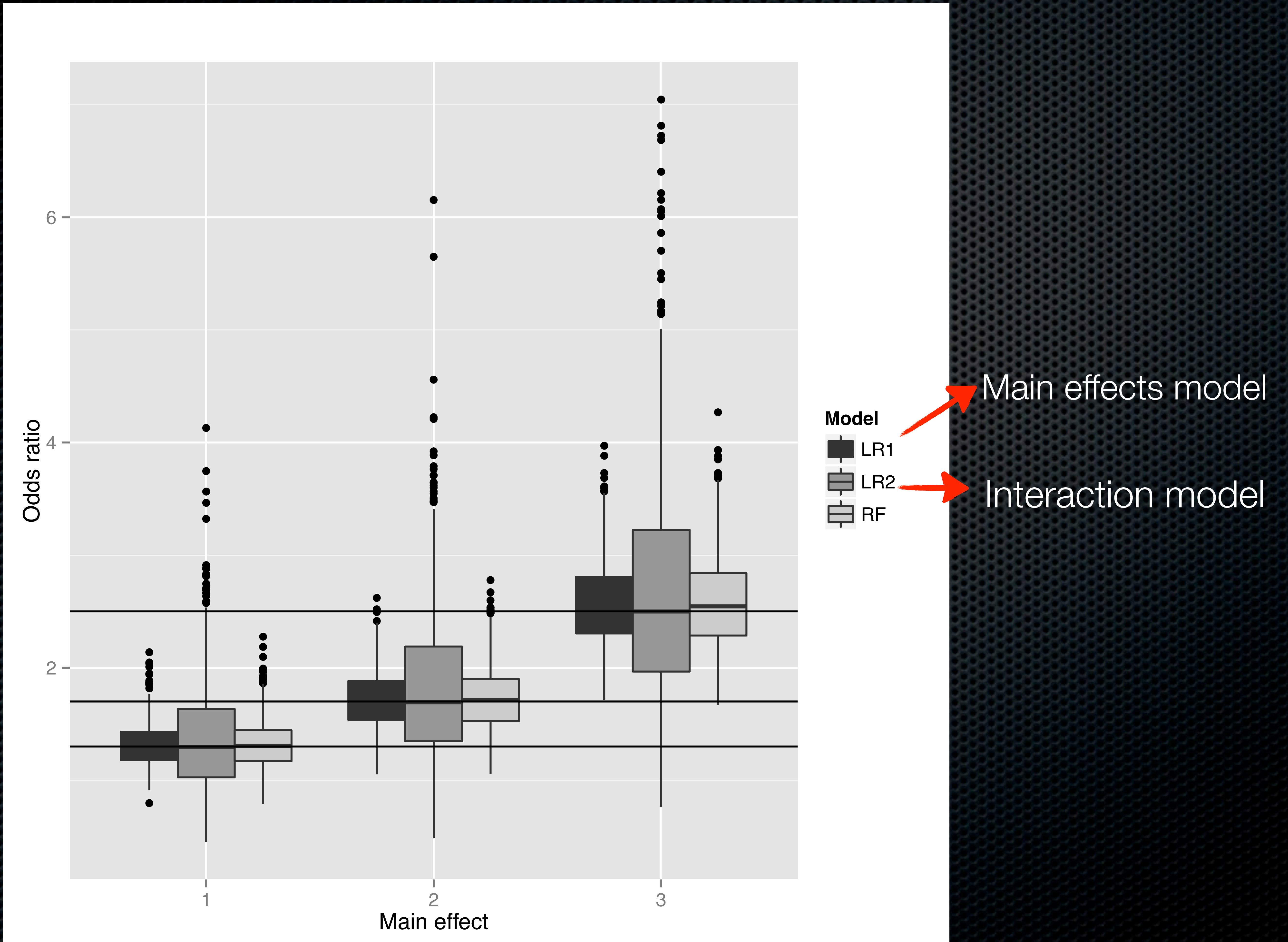
- In fact, since you now have individual “effects”
  - You can see how these change against a second variable
  - You can look at arbitrary subgroups
- You can extend this idea to look at interactions **at an individual level**

Can this possibly work?!!!

A simulation of binary regression

- Simulated binary outcomes under a logistic regression model with binary features
- Fit both a logistic regression model and a random forest model to the simulated data
- Used regression estimates from the logistic model and random forest based counterfactual machines


Odds ratio
6
4
2
1
2
3
Main effect
Model
LR1
LR2
RF



Odds ratio
6
4
2
1
2
3
Main effect
Model
LR1
LR2
RF
Main effects model
Interaction model

But wait, there’s more!!

Interactions

- In genetics there is a great deal of interest in gene-gene and gene-environment interactions
- Similarly, I'm sure, in other domains, there are situations where joint effects are interesting

Interactions

Interactions

- Is there an interaction between Red/Blue and Male/Female?

Interactions

- Is there an interaction between Red/Blue and Male/Female?
- Train predictive models within each of the 4 subgroups
  - Red Male, Red Female, Blue Male, Blue Female

Interactions

- Is there an interaction between Red/Blue and Male/Female?
- Train predictive models within each of the 4 subgroups
  - Red Male, Red Female, Blue Male, Blue Female
- Drop each person down each of the 4 models
  - (y_{RM}, y_{RF}, y_{BM}, y_{BF})

Interactions

- Is there an interaction between Red/Blue and Male/Female?
- Train predictive models within each of the 4 subgroups
  - Red Male, Red Female, Blue Male, Blue Female
- Drop each person down each of the 4 models
  - $(y_{RM}, y_{RF}, y_{BM}, y_{BF})$
- Interaction effect : $y_{RM}-y_{RF} - (y_{BM}-y_{BF})$

Interactions

- Is there an interaction between Red/Blue and Male/Female?
- Train predictive models within each of the 4 subgroups
  - Red Male, Red Female, Blue Male, Blue Female
- Drop each person down each of the 4 models
  - $(y_{RM}, y_{RF}, y_{BM}, y_{BF})$
- Interaction effect : $y_{RM}-y_{RF} - (y_{BM}-y_{BF})$
- Average over everyone to get overall interaction effect

But this gets tiring fast!!!

The Interactor

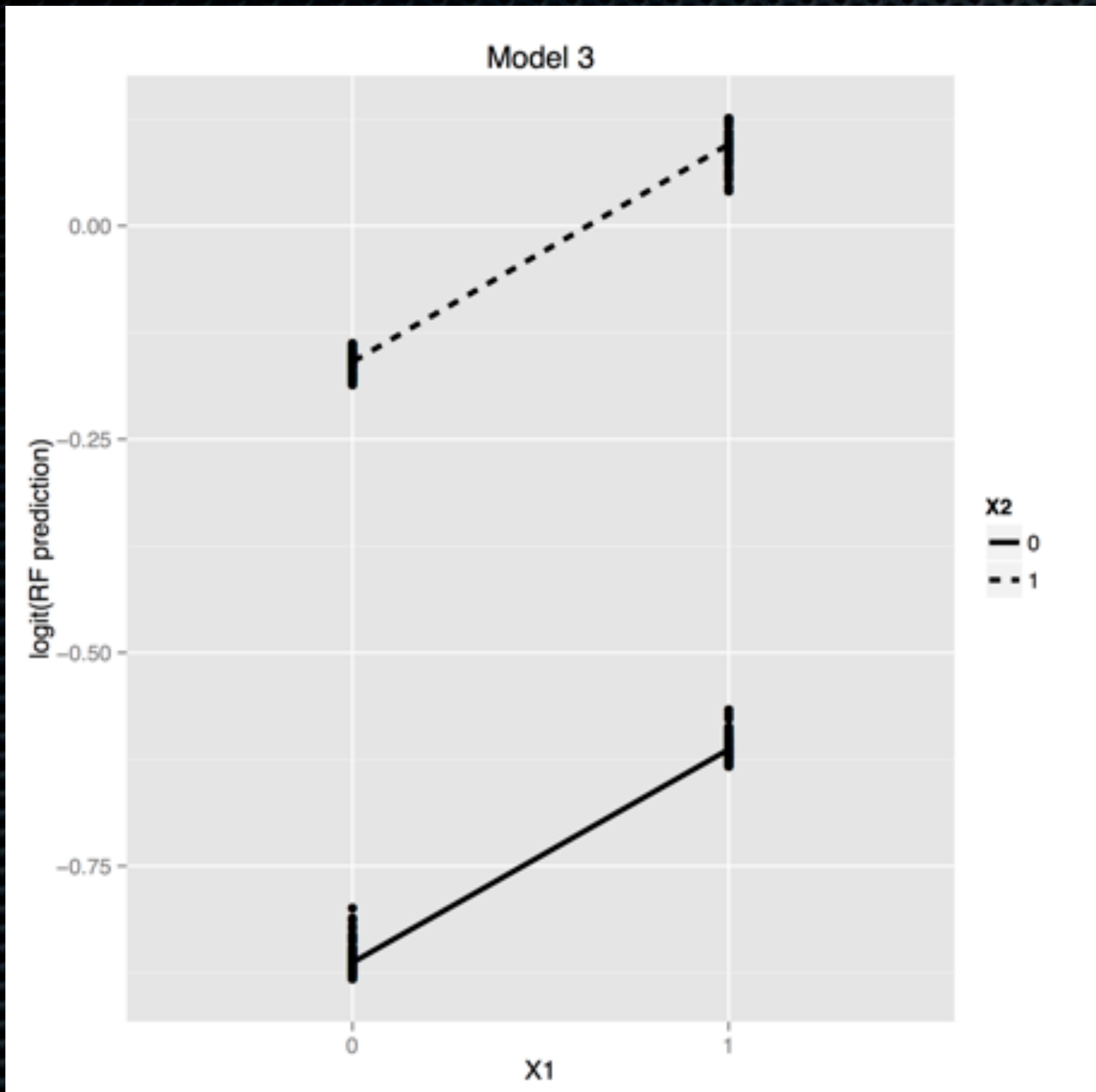
- Fit a <u>single</u> predictive model to your data
- Compute

$$\overline{y_{RM}} - \overline{y_{RF}} - (\overline{y_{BM}} - \overline{y_{BF}})$$

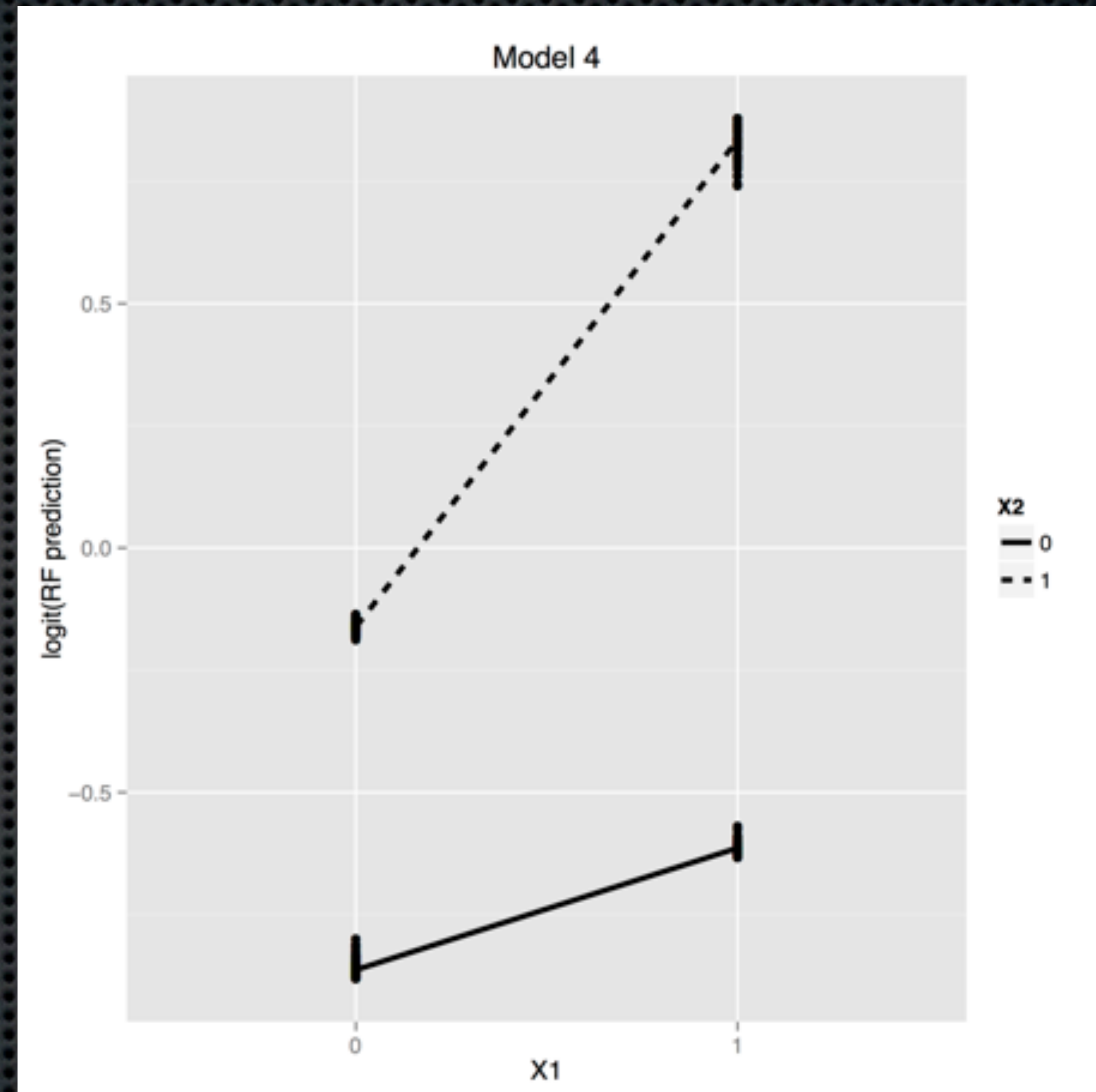
- This isn't a true interaction effect since it doesn't compute the counterfactuals
- It can be used to scan a large set of variables to see if interactions might exist

The Interactor

- This is of course trivial for continuous outcomes
- For binary outcomes, our models give us probability estimates, and so we can look at interactions in the predicted probabilities
  - Note, these probabilities aren't observed
  - In logistic regression, the predicted probabilities reflect the model, not necessarily what the data is saying.



Model 3
logit(RF prediction)
0.00
-0.25
-0.50
-0.75
X1
0
1
X2
0
1



Model 4
logit(RF prediction)
0.5
0.0
-0.5
X1
0
1
X2
0
1

Some notes

- Doing counterfactuals using classical parametric models only reflect the model, not the data
- Your effect size estimates will be only as good as the machine you use as your predictive model
- You can get standard errors by either bootstrapping or using bagged machines
- What I've described is straightforward for binary or categorical features
- You can get similar counterfactual machines for continuous features, but it is not so easy (ongoing research)

Using counterfactual machines

you **can** get conditional effects for particular features

Using counterfactual machines

you **can** get conditional effects for particular features

No more whining