

Mixed Integer Programming (MIP) for Daily Fantasy Sports, Statistics and Marketing

Juan Pablo Vielma

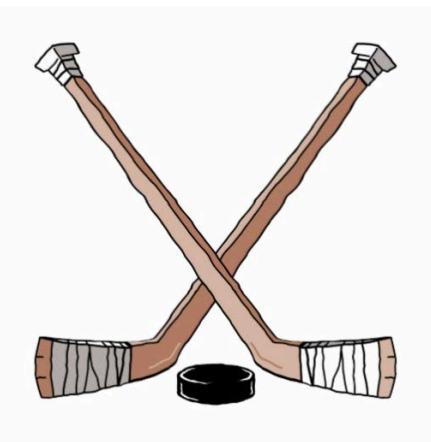
Massachusetts Institute of Technology

AM/ES 121, SEAS, Harvard.
Boston, MA, November, 2016.

MIP & Daily Fantasy Sports

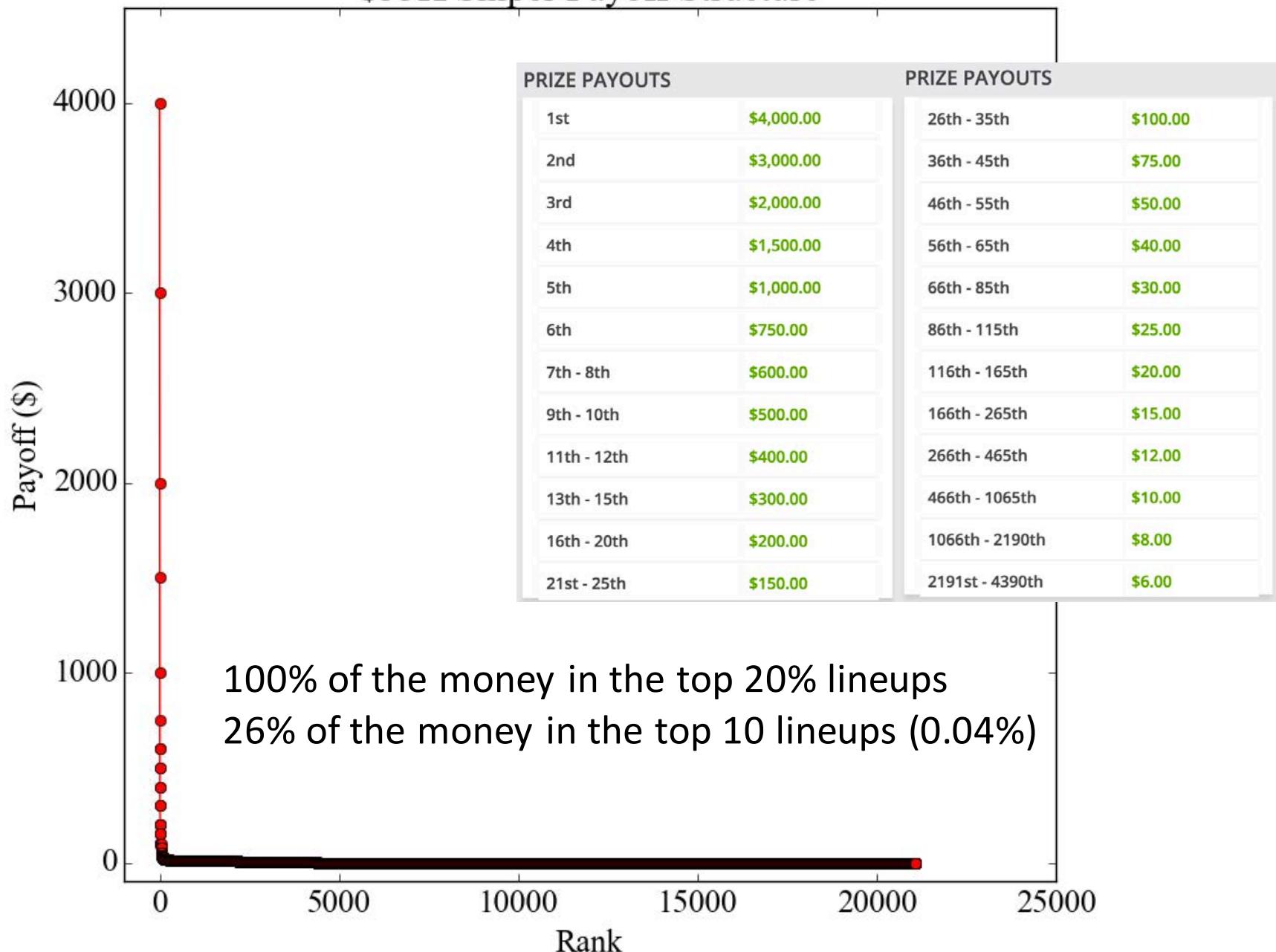


Example Entry



LINEUP					Avg. Rem. / Player: \$0	Rem. Salary: \$0
POS	PLAYER	OPP	FPPG	SALARY		
C	Jussi Jokinen	Fla@Anh	3.1	\$5,300	X	
C	Brandon Sutter	Pit@Van	3.0	\$4,400	X	
W	Nikolaj Ehlers	Wpg@Tor	3.9	\$4,800	X	
W	Daniel Sedin	Pit@Van	3.8	\$6,400	X	
W	Radim Vrbata	Pit@Van	3.4	\$5,800	X	
D	Brian Campbell	Fla@Anh	2.6	\$4,100	X	
D	Morgan Rielly	Wpg@Tor	3.5	\$4,200	X	
G	Corey Crawford P	StL@Chi	6.3	\$7,800	X	
UTIL	Blake Wheeler	Wpg@Tor	4.8	\$7,200	X	

\$55K Sniper Payoff Structure



Building a Lineup



Integer Programming Formulation

- We will make a bunch of lineups consisting of 9 players each
- Use an integer programming approach to find these lineups

Decision variables

$$x_{pl} = \begin{cases} 1, & \text{if player } p \text{ in lineup } l \\ 0, & \text{otherwise} \end{cases}$$

Basic Feasibility

- 9 different players
- Salary less than \$50,000

Basic constraints

$$\sum_{p=1}^N c_p x_{pl} \leq \$50,000, \quad (\text{budget constraint})$$

$$\sum_{p=1}^N x_{pl} = 9, \quad (\text{lineup size constraint})$$

$$x_{pl} \in \{0, 1\}, \quad 1 \leq p \leq N.$$

Position Feasibility

- Between 2 and 3 centers
- Between 3 and 4 wingers
- Between 2 and 3 defensemen
- 1 goalie

Position constraints

$$2 \leq \sum_{p \in C} x_{pl} \leq 3, , \quad (\text{center constraint})$$

$$3 \leq \sum_{u \in W} x_{pl} \leq 4, \quad (\text{winger constraint})$$

$$2 \leq \sum_{u \in D} x_{pl} \leq 3, \quad (\text{defensemen constraint})$$

$$\sum_{u \in G} x_{pl} = 1 \quad (\text{goalie constraint})$$

Team Feasibility

- At least 3 different NHL teams

Team constraints

$$t_i \leq \sum_{p \in T_i} x_{pl}, \quad \forall i \in \{1, \dots, N_T\}$$

$$\sum_{i=1}^{N_T} t_i \geq 3,$$

$$t_i \in \{0, 1\}, \quad \forall i \in \{1, \dots, N_T\}.$$

Maximize Points

- Forecasted points for player p: f_p



Score type	Points
Goal	3
Assist	2
Shot on Goal	0.5
Blocked Shot	0.5
Short Handed Point Bonus (Goal/Assist)	1
Shootout Goal	0.2
Hat Trick Bonus	1.5
Win (goalie only)	3
Save (goalie only)	0.2
Goal allowed (goalie only)	-1
Shutout Bonus (goalie only)	2

Table 1 Points system for NHL contests in DraftKings.



Points Objective Function

$$\sum_{p=1}^N f_p x_{pl}$$

Lineup

Projections: 5.4 2.5 3.4 3.0 3.2 4.2 3.5 3.4 5.7

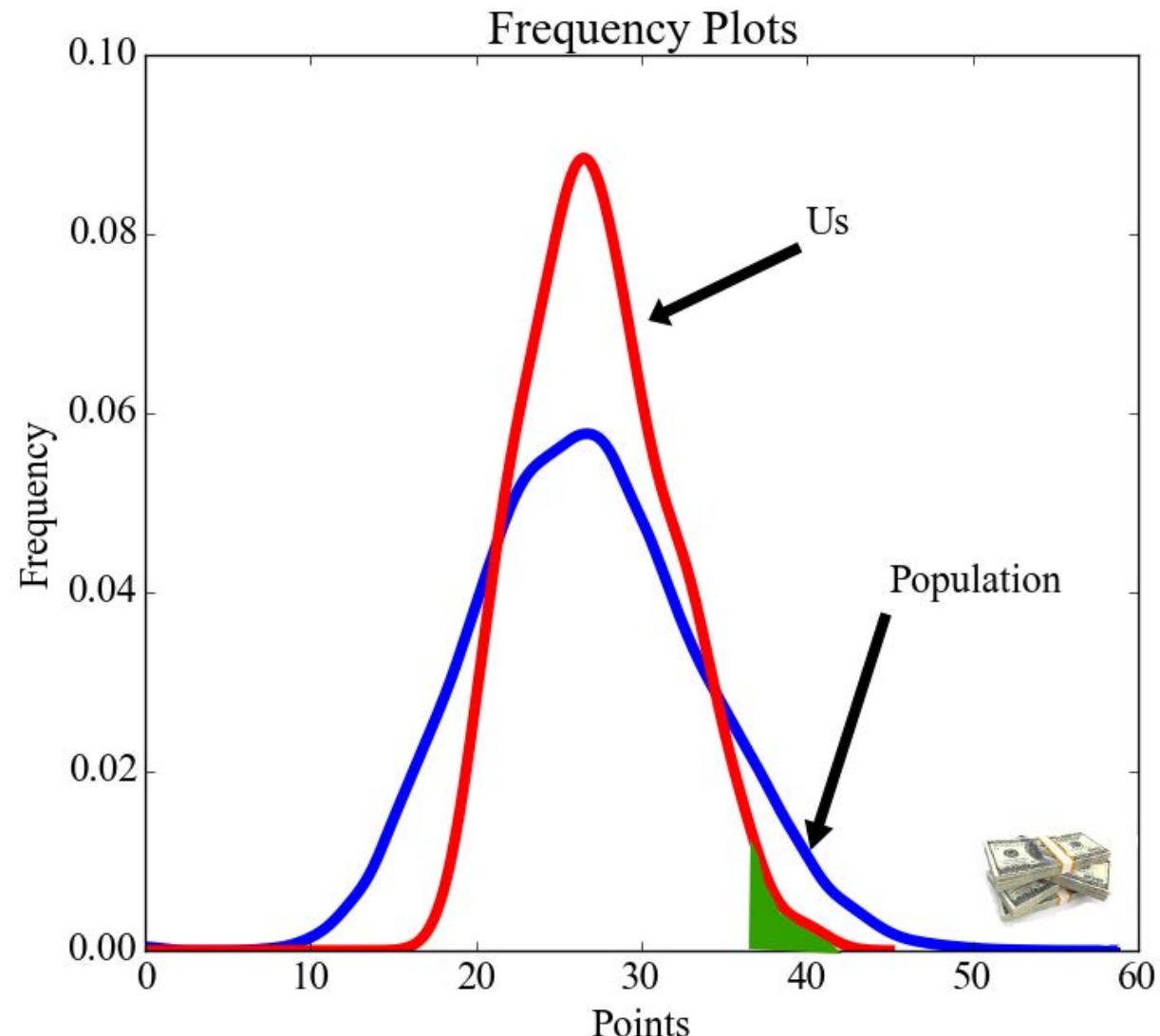
\$9500 \$2700 \$4600 \$3800 \$4600 \$6400 \$5200 \$5100 \$8000

W UTIL D D C C W W G

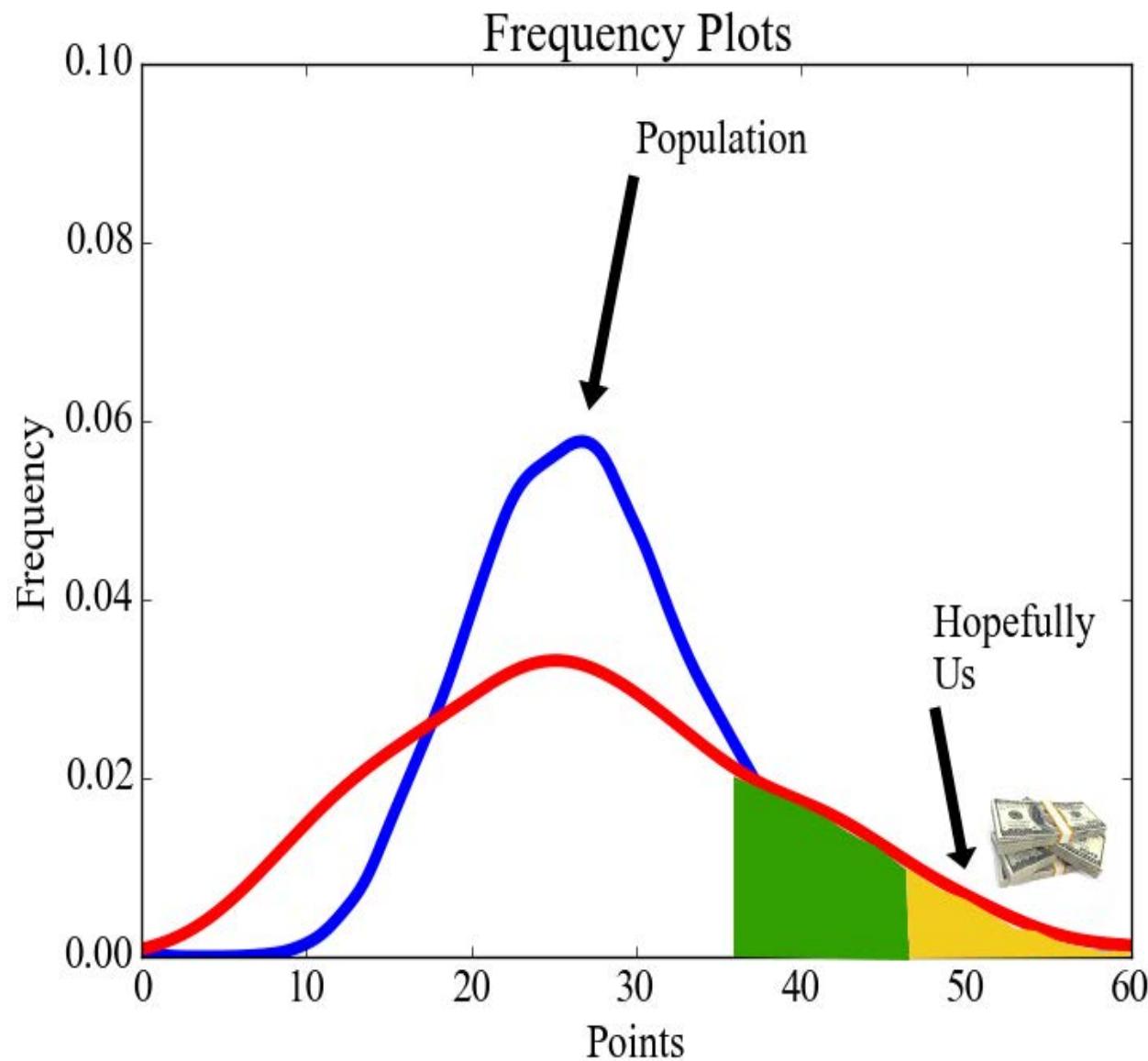


23 points on average

Need > 38 points for a chance to win



Increase variance to have a chance

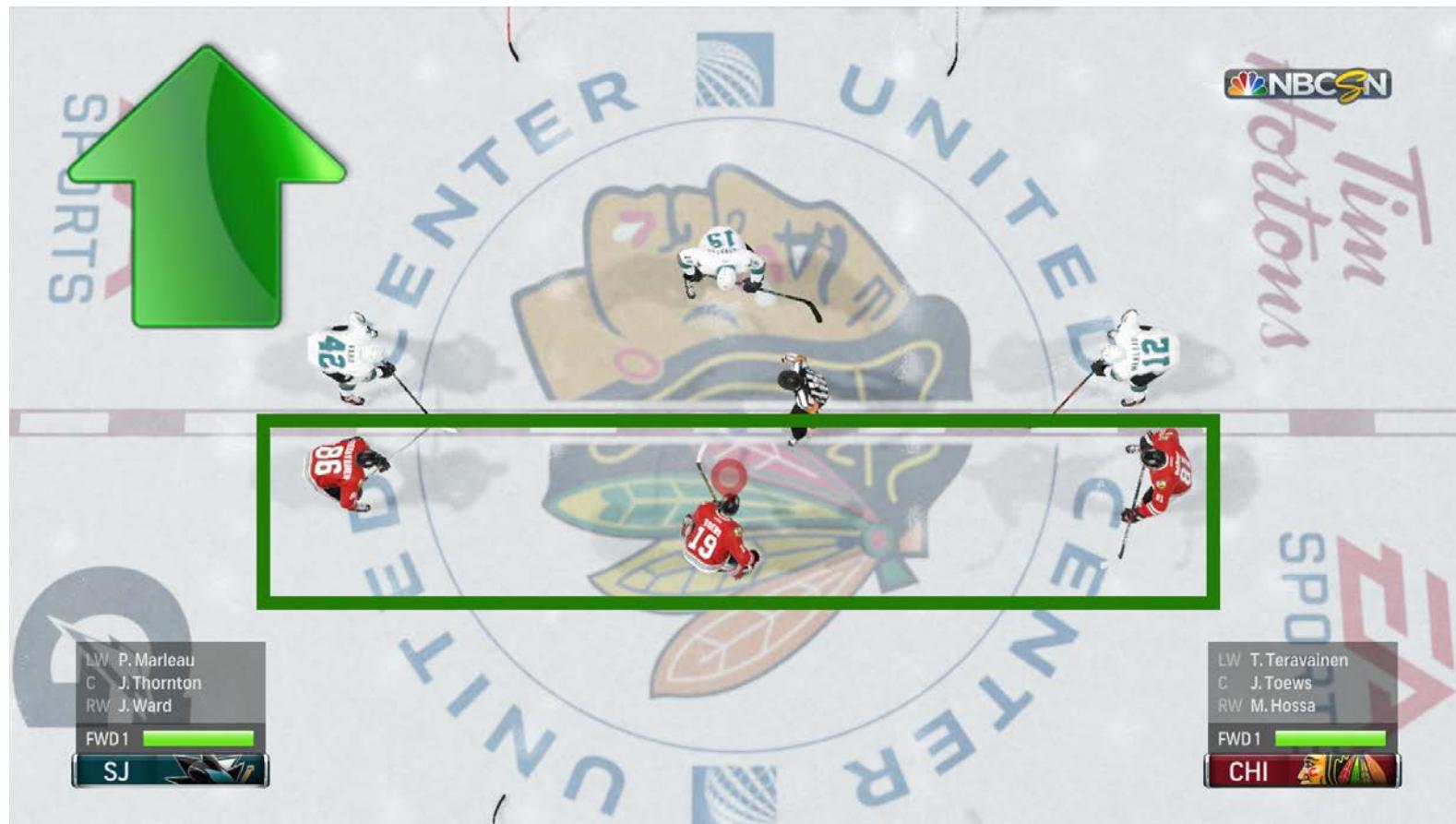


Structural Correlations - Teams



Structural Correlations - Lines

- Goal = 3 pt, assist = 2 pt



Structural Correlations – Lines = Stacking

- At least 1 complete line (3 players per line)
- At least 2 partial lines (at least 2 players per line)

1 complete line constraint

$$3v_i \leq \sum_{p \in L_i} x_{pl}, \quad \forall i \in \{1, \dots, N_L\}$$

$$\sum_{i=1}^{N_L} v_i \geq 1$$

$$v_i \in \{0, 1\}, \quad \forall i \in \{1, \dots, N_L\}.$$

2 partial lines constraint

$$2w_i \leq \sum_{p \in L_i} x_{pl}, \quad \forall i \in \{1, \dots, N_L\}$$

$$\sum_{i=1}^{N_L} w_i \geq 2$$

$$w_i \in \{0, 1\}, \quad \forall i \in \{1, \dots, N_L\}.$$

Structural Correlations – Goalie Against Opposing Players



Structural Correlations – Goalie Against Skaters

- No skater against goalie

No skater against goalie constraint

$$6x_{pl} + \sum_{q \in Opponents_p} x_{ql} \leq 6, \quad \forall p \in G$$

Good, but not great chance

Feasible

Line

Team

Line

Goalie

Not
Against



Play many diverse Lineups

- Make sure lineup l has no more than γ players in common with lineups 1 to $l-1$

Diversity constraint

$$\sum_{p=1}^N x_{pk}^* x_{pl} \leq \gamma, k = 1, \dots, l-1$$

Were we able to do it?

GameCenter
STANDINGS | ENTRIES | DETAILS | GAMES
NHL \$2K Sniper [\$2,000 Guaranteed]

Rank	User	Score	PMR
1st	zlisto	54.50	PMR 0
3rd	zlisto	51.50	PMR 0
9th	zlisto	49.50	PMR 0
23rd	zlisto	46.00	PMR 0
28th	zlisto	45.50	PMR 0
28th	zlisto	45.50	

GameCenter
STANDINGS | ENTRIES | DETAILS | GAMES
NHL \$40K Sniper [\$40,000 Guaranteed]

Rank	User	Score	PMR
2nd	zlisto	61.30	PMR 0
21st	zlisto	57.30	PMR 0
21st	zlisto	57.30	PMR 0
40th	zlisto	56.10	PMR 0
42nd	zlisto	55.70	PMR 0
81st	zlisto	54.10	

GameCenter
STANDINGS | ENTRIES | DETAILS | GAMES
NHL \$80K Tuesday Special [\$80,000 Guaranteed]

Rank	User	Score	PMR
3rd	zlisto	54.60	PMR 0
6th	zlisto	52.80	PMR 0
7th	zlisto	52.30	PMR 0
10th	zlisto	50.60	PMR 0
11th	zlisto	50.30	PMR 0
15th	zlisto	50.10	

GameCenter
STANDINGS | ENTRIES | DETAILS | GAMES
NHL \$45K Sniper [\$45,000 Guaranteed]

Rank	User	Score	PMR
1st	zlisto	52.60	PMR 0
8th	zlisto	49.60	PMR 0
57th	zlisto	45.60	PMR 0
57th	zlisto	45.60	PMR 0
83rd	zlisto	44.60	PMR 0
83rd		44.60	

November 15, 2015

November 16, 2015

November 17, 2015

November 23, 2015

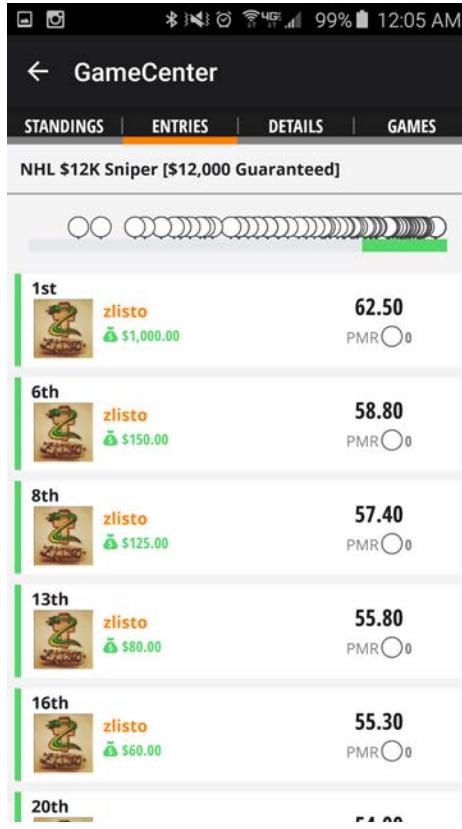
200 lineups

Policy Change



200 lineups -> 100 lineups

Were we able to continue it?



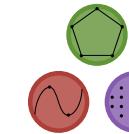
> \$15K

December 12, 2015

100 lineups



How can you do it?



JuMP

Download Code from Github:

<https://github.com/dscotthunter/Fantasy-Hockey-IP-Code>

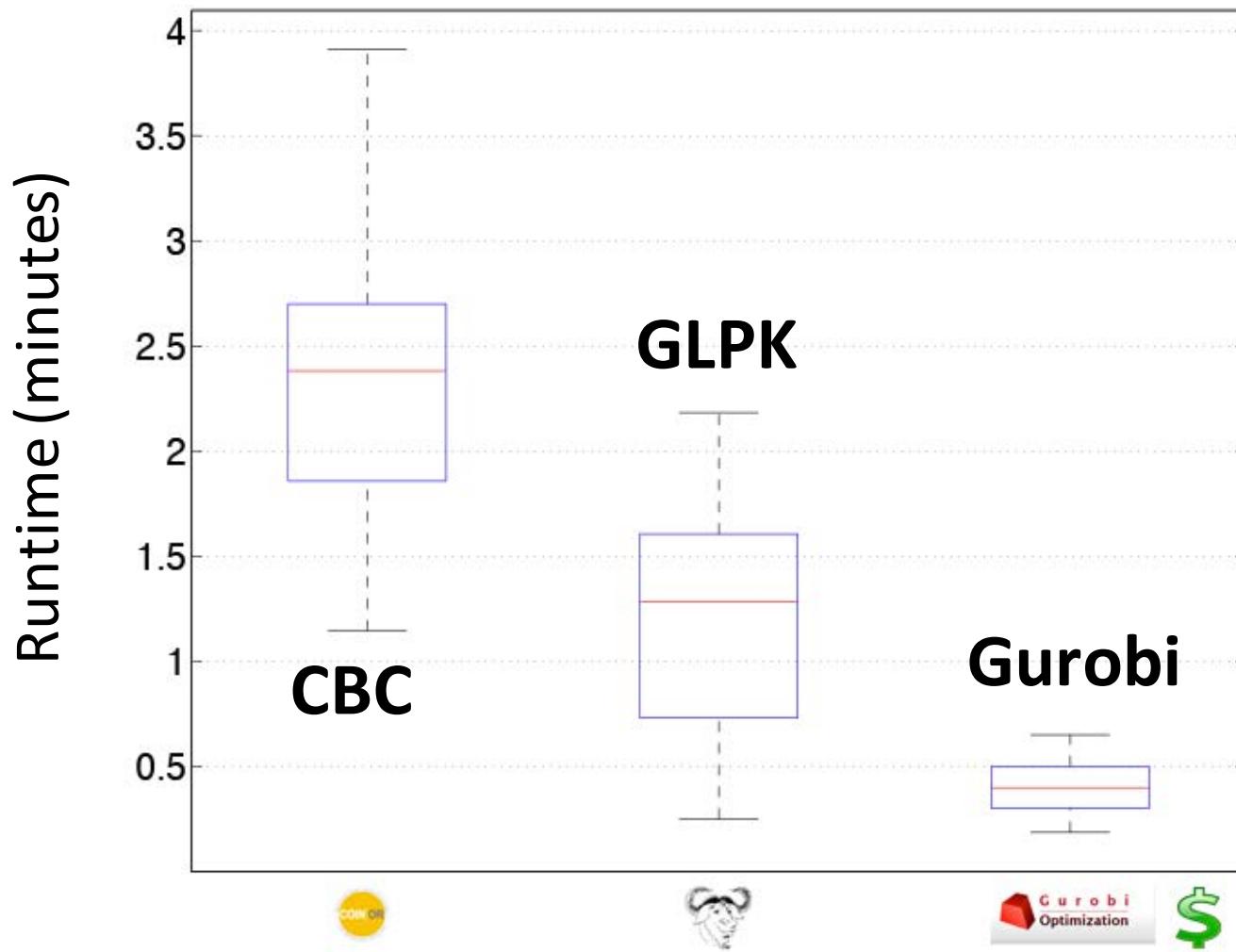
The screenshot shows a Julia code editor with a large amount of JuMP code. The code defines variables for players (skaters, goalies, lines), constraints for team composition (e.g., 6 skaters, 1 goalie per line, 2 centers, 3 wings, 1 center/goalie overlap), and objective functions for salary and points. It also includes logic for stack constraints (stacks of 2 or 3) and specific rules for defensemen (must be on PTC). The code ends with a solve command.

```
175 function one_lineup_Type4(skaters, goalies, lines, num_overlap, num_skaters, num_goalies, centers, wingers, defenders, num_teams, skaters_teams, goalie_oppents, team_lines, num_lines, P1_info)
176     m = Model(optimizer=GurobiSolver())
177
178     # Variables for skaters
179     num_skaters = IntVar(m, 0, num_skaters, Bin)
180     num_goalies = IntVar(m, 0, num_goalies, Bin)
181     num_centers = IntVar(m, 0, num_centers, Bin)
182     num_wingers = IntVar(m, 0, num_wingers, Bin)
183
184     # Skaters
185     addConstraint(m, sum(goalies_lineup[i], 1:1:num_goalies) == 1)
186     # Points Skaters
187     addConstraint(m, sum(skaters_lineup[i], 1:1:num_skaters) == 6)
188
189     # Centers
190     addConstraint(m, sum(centers[i], skaters_lineup[i], 1:1:num_skaters) == 2)
191     # Wings
192     addConstraint(m, sum(wingers[i], skaters_lineup[i], 1:1:num_skaters) == 3)
193
194     # Defense
195     addConstraint(m, sum(defenders[i], skaters_lineup[i], 1:1:num_skaters) == 1)
196     addConstraint(m, sum(defenders[i], skaters_lineup[i], 1:1:num_skaters) == 1)
197
198     # Between 2 and 3 defenders
199     addConstraint(m, 2 == sum(defenders[i], skaters_lineup[i], 1:1:num_skaters))
200     addConstraint(m, 3 == sum(defenders[i], skaters_lineup[i], 1:1:num_skaters))
201
202     # Salary
203     addConstraint(m, sum(skaters[i].Salary, skaters_lineup[i], 1:1:num_skaters) + sum(goalies[i].Salary, goalie_oppents[i], 1:1:num_goalies) == 50000)
204
205     # Must have different teams for the 2 centers constraint
206     num_on_team = IntVar(m, 0, num_teams)
207     userVar1, num_on_team, neg_num_on_team = Var(m, 1:1:num_teams)
208     userVar2, num_on_team, neg_num_on_team = Var(m, 1:1:num_teams)
209     userVar3, num_on_team, neg_num_on_team = Var(m, 1:1:num_teams)
210     userVar4, num_on_team, neg_num_on_team = Var(m, 1:1:num_teams)
211
212     addConstraint(m, num_on_team == num_on_team - neg_num_on_team, 1:1:num_teams)
213     addConstraint(m, num_on_team == num_on_team - neg_num_on_team, 2:2:num_teams)
214     addConstraint(m, num_on_team == num_on_team - neg_num_on_team, 3:3:num_teams)
215     addConstraint(m, num_on_team == num_on_team - neg_num_on_team, 4:4:num_teams)
216
217     # No goalie going against skaters
218     addConstraint(m, constri[1:num_skaters, 1:num_goalies] == 0)
219
220     # Must have at least 1 line up with at least 2 people
221     pos_num_in_line, neg_num_in_line = Var(m, 1:1:num_lines)
222     userVar5, pos_num_in_line, neg_num_in_line = Var(m, 1:1:num_lines)
223     userVar6, pos_num_in_line, neg_num_in_line = Var(m, 1:1:num_lines)
224     userVar7, pos_num_in_line, neg_num_in_line = Var(m, 1:1:num_lines)
225     userVar8, pos_num_in_line, neg_num_in_line = Var(m, 1:1:num_lines)
226
227     addConstraint(m, pos_num_in_line == 1)
228     addConstraint(m, pos_num_in_line == 2)
229     addConstraint(m, pos_num_in_line == 3)
230
231     # Line Stack
232     line_stack1, line_stack2 = Var(m, 1:1:num_lines)
233     line_stack1, line_stack2 = Var(m, 1:1:num_lines)
234     line_stack1, line_stack2 = Var(m, 1:1:num_lines)
235
236     addConstraint(m, line_stack1 == 1)
237     addConstraint(m, line_stack2 == 2)
238     addConstraint(m, line_stack1 == 3)
239
240     # Must have at least 1 line up with at least 2 people
241     pos_num_in_line2, neg_num_in_line2 = Var(m, 1:1:num_lines)
242     userVar9, pos_num_in_line2, neg_num_in_line2 = Var(m, 1:1:num_lines)
243     userVar10, pos_num_in_line2, neg_num_in_line2 = Var(m, 1:1:num_lines)
244     userVar11, pos_num_in_line2, neg_num_in_line2 = Var(m, 1:1:num_lines)
245     userVar12, pos_num_in_line2, neg_num_in_line2 = Var(m, 1:1:num_lines)
246
247     addConstraint(m, pos_num_in_line2 == 1)
248     addConstraint(m, pos_num_in_line2 == 2)
249     addConstraint(m, pos_num_in_line2 == 3)
250
251     # The defenders must be on PTC
252     addConstraint(m, sum(defenders[i], skaters_lineup[i], 1:1:num_skaters, j, 1:num_teams) == sum(defenders[i], skaters_lineup[i], 1:1:num_skaters))
253
254     # Overlap Constraint
255     addConstraint(m, constri[1:1:(size(lines)[1]), 1:1:(size(lines)[2]), i, 1:num_skaters, j, 1:num_skaters] == sum(lines[num_skaters], 1:1:goalies_lineup[i], j, 1:num_goalies) - num_overlap)
256
257
258     # Status optional
259     skaters_lineup_copy = Array{Int64}(0)
260     for i in 1:size(skaters_lineup, 1)
261         if getvalue(skaters_lineup[i]) == 0.0 getvalue(skaters_lineup[i]) == 1.0
262             skaters_lineup_copy = vcat(skaters_lineup_copy, fill(0, 1))
263         else
264             skaters_lineup_copy = vcat(skaters_lineup_copy, fill(1, 1))
265         end
266     end
267
268     for i in 1:size(goalies_lineup, 1)
269         if getvalue(goalies_lineup[i]) == 0.0 getvalue(goalies_lineup[i]) == 1.0
270             skaters_lineup_copy = vcat(skaters_lineup_copy, fill(0, 1))
271         else
272             skaters_lineup_copy = vcat(skaters_lineup_copy, fill(1, 1))
273         end
274     end
275
276     status = solve(m)
```

<http://arxiv.org/pdf/1604.01455v1.pdf>

Performance Time

< 30 Minutes



Solver

MIP and Statistics: Inference for the Chilean Earthquake

The 2010 Chilean Earthquake



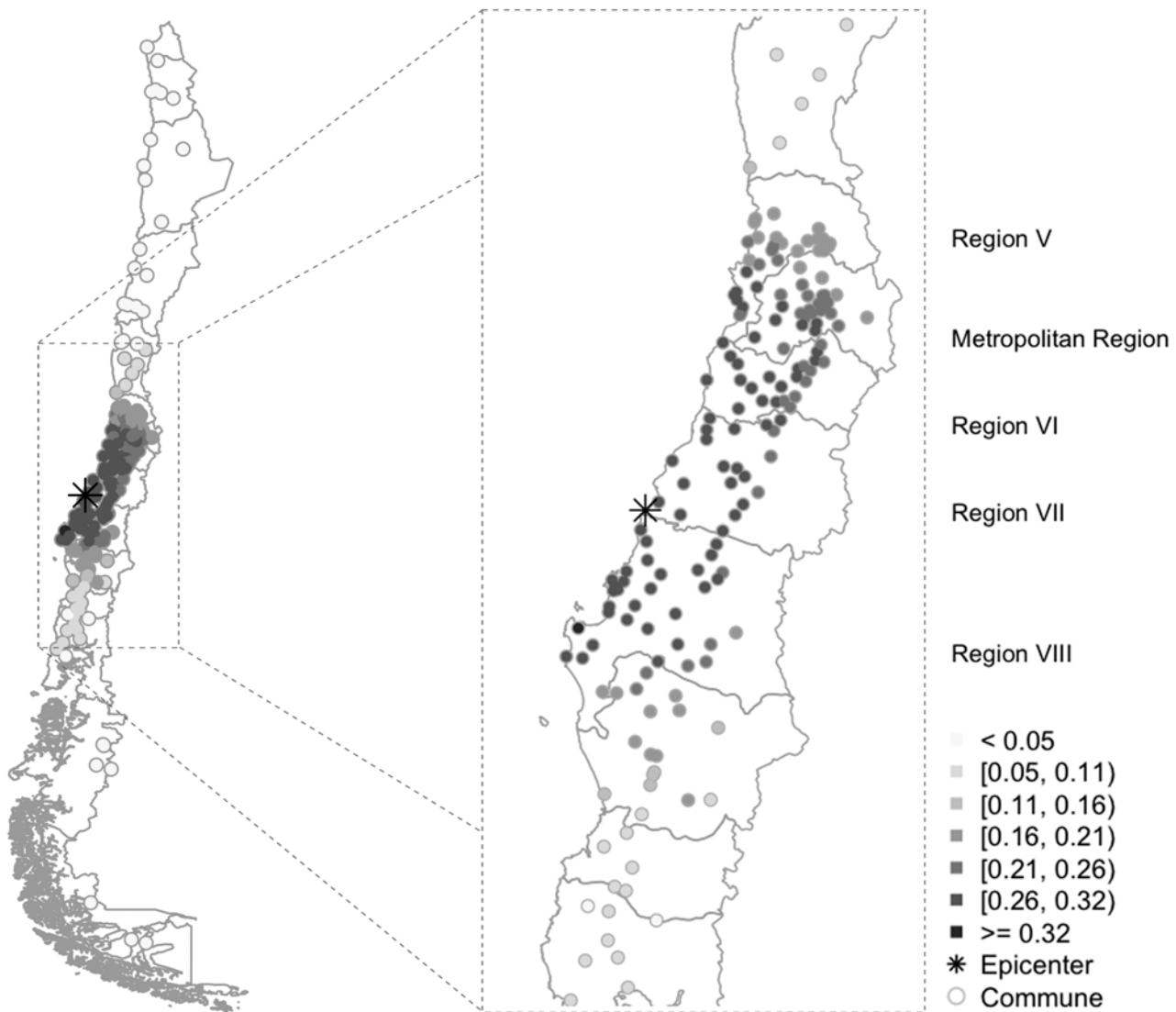
6th Strongest in Recorded History (8.8)



Impact on Educational Achievement? PSU = SAT

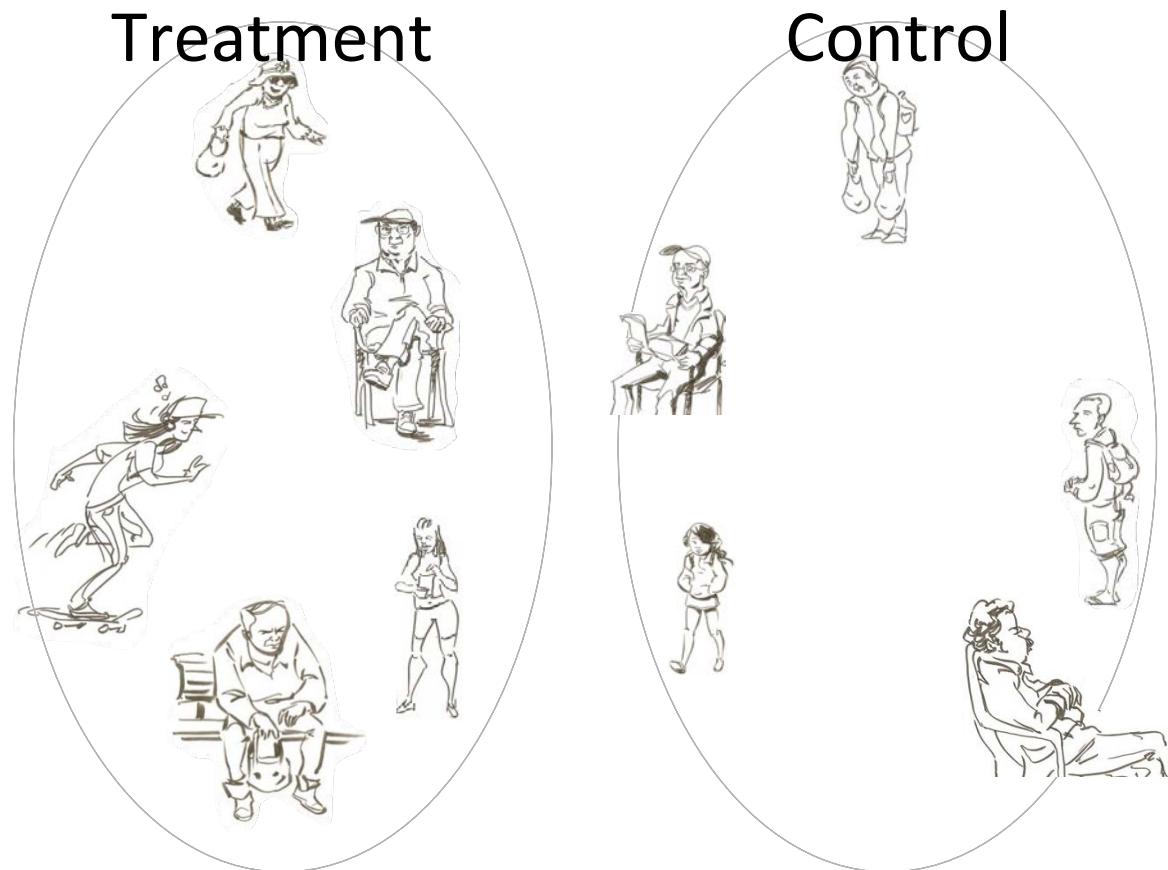


Earthquake Intensity + Great Demographic Info



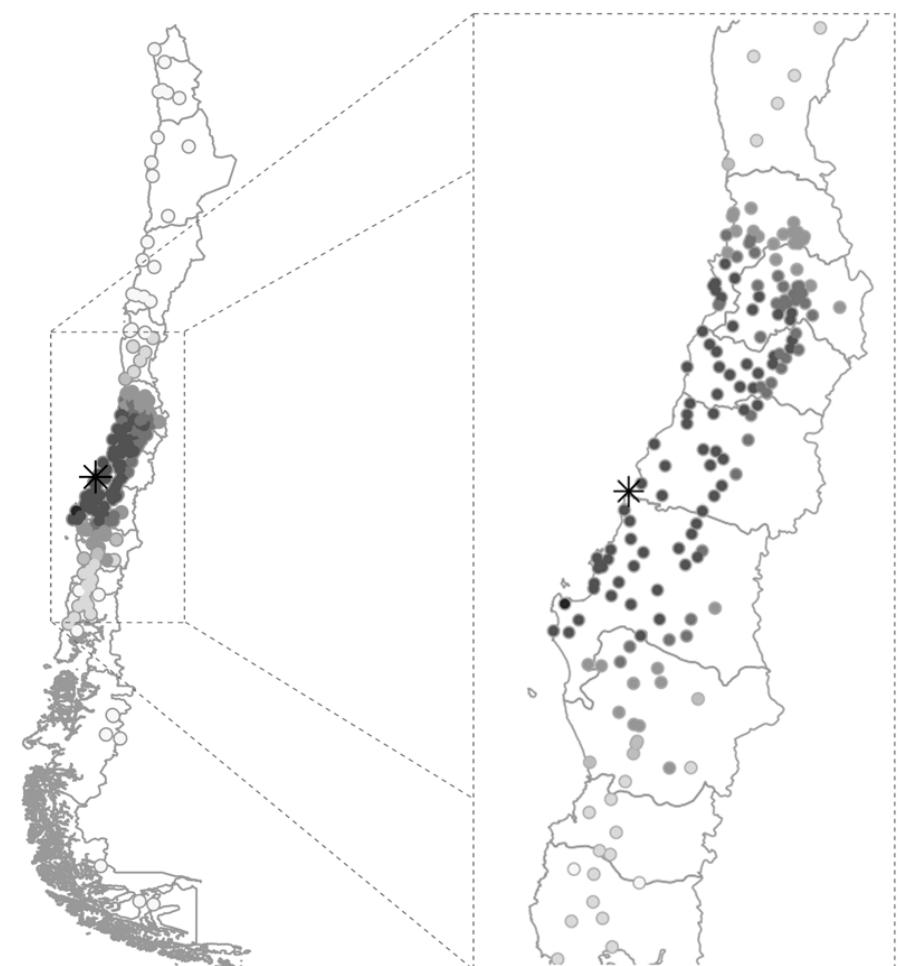
Randomized experiment

- Treatment / control have similar characteristics (covariates).



Covariate Balance Important for Inference

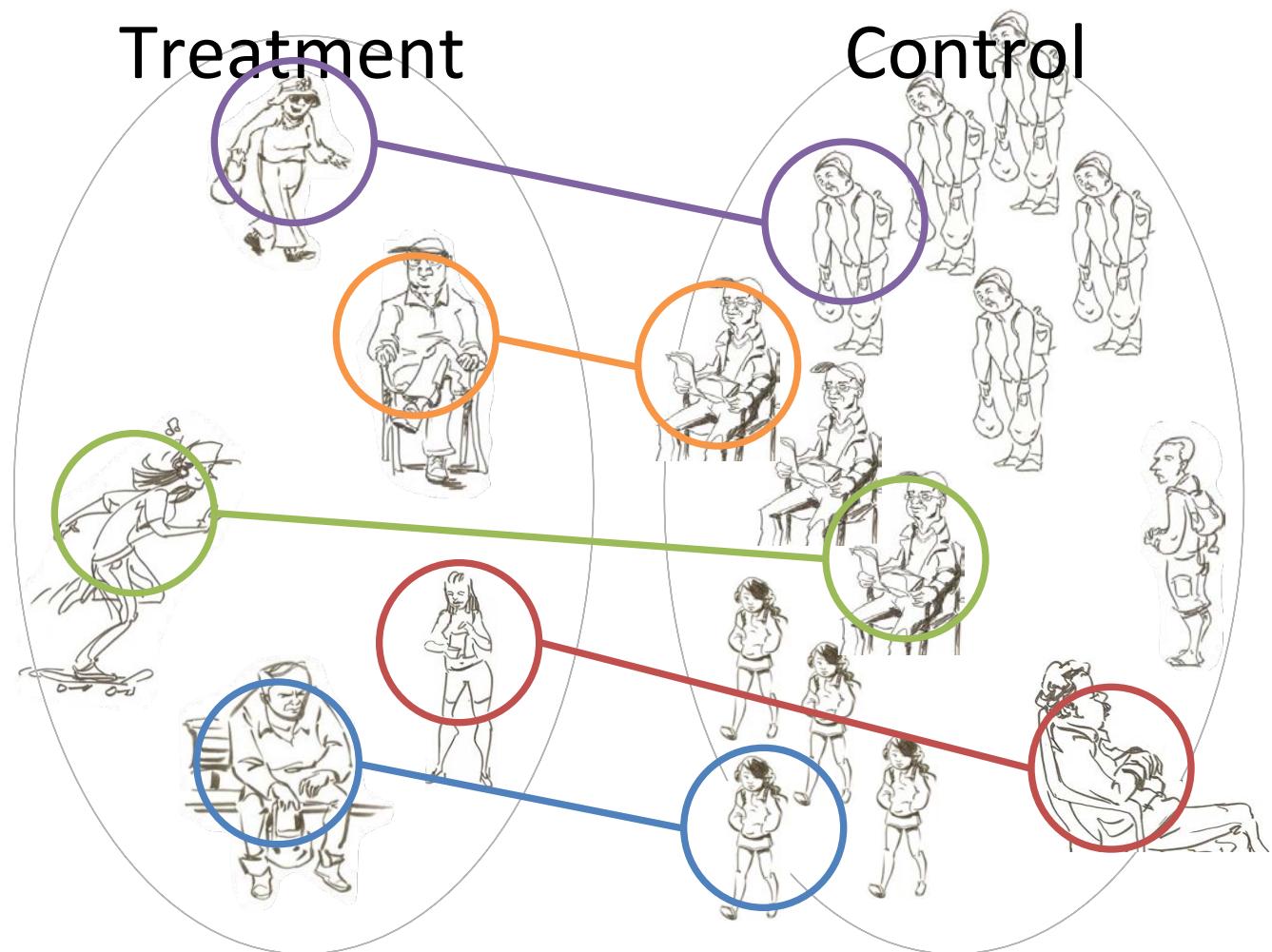
Covariate	Dose	
	1	2
Gender		
Male	462	462
Female	538	538
School SES		
Low	75	75
Mid-low	327	327
Medium	294	294
Mid-high	189	189
High	115	115
Mother's education		
Primary	335	335
Secondary	426	426
Technical	114	114
College	114	114
Missing	11	11
⋮		



Observational Study: e.g. After Earthquake

- Treatment / control can have different characteristics.

Solution
=
Matching?



Matching

Treated Units: $\mathcal{T} = \{t_1, \dots, t_T\}$

Control Units: $\mathcal{C} = \{c_1, \dots, c_C\}$

Observed Covariates: $\mathcal{P} = \{p_1, \dots, p_P\}$

Covariate Values: $\mathbf{x}^t = (x_p^t)_{p \in \mathcal{P}}, \quad t \in \mathcal{T}$

$\mathbf{x}^c = (x_p^c)_{p \in \mathcal{P}}, \quad c \in \mathcal{C}$

Nearest Neighbor Matching

$$\underset{\mathbf{m}}{\text{minimize}} \quad \sum_{t \in \mathcal{T}} \sum_{c \in \mathcal{C}} \delta_{t,c} m_{t,c}$$

$$\text{subject to} \quad \sum_{c \in \mathcal{C}} m_{t,c} = 1, \quad t \in \mathcal{T}$$

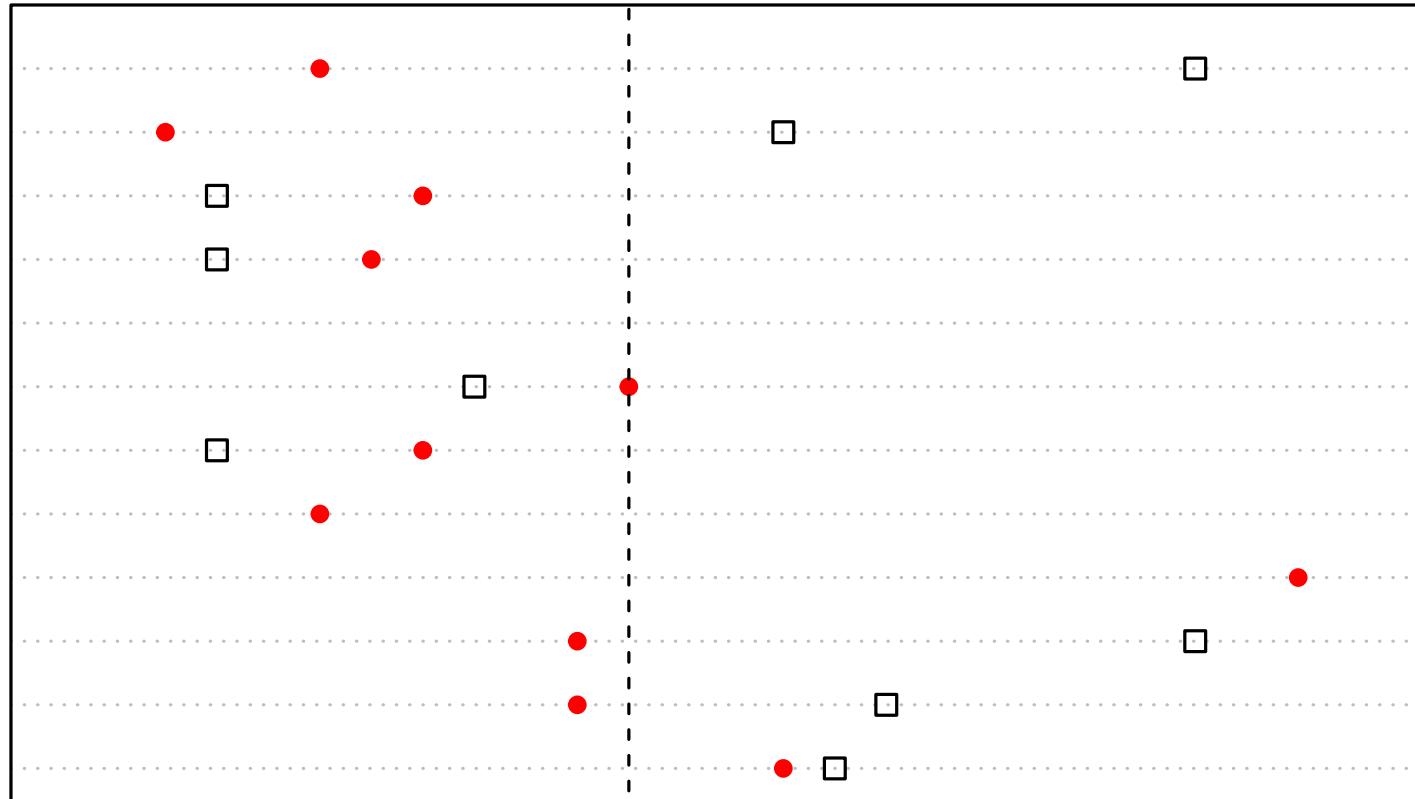
$$\sum_{t \in \mathcal{T}} m_{t,c} \leq 1, \quad c \in \mathcal{C}$$

$$0 \leq m_{t,c} \leq 1 \quad \underline{m_{t,c}} \in \{0, 1\}, \quad t \in \mathcal{T}, c \in \mathcal{C}$$

- e.g. $\delta_{t,c} = \|\mathbf{x}^t - \mathbf{x}^c\|_2$
- Easy to solve

Balance Before After Matching

SIMCE school (decile)
SIMCE student (decile)
GPA ranking (decile)
Attendance (decile)
Rural school
Catholic school
High SES school
Mid-High SES school
Mid SES school
Mid-Low SES school
Public School
Voucher School



Maximum Cardinality Matching

$$\begin{aligned} \max & \quad \sum_{t \in \mathcal{T}} \sum_{c \in \mathcal{C}} m_{t,c} \\ s.t. & \end{aligned}$$

$$\mathcal{K}(p) = \{\mathbf{x}_p^c\}_{c \in \mathcal{P}} \cup \{\mathbf{x}_p^t\}_{t \in \mathcal{T}}$$

$$\mathcal{C}_{p,k} = \{c \in \mathcal{C} : \mathbf{x}_p^c = k\}$$

$$\mathcal{T}_{p,k} = \{t \in \mathcal{T} : \mathbf{x}_p^t = k\}$$

$$\sum_{t \in \mathcal{T}} m_{t,c} \leq 1, \quad \forall c \in \mathcal{C}$$

$$\sum_{c \in \mathcal{C}} m_{t,c} \leq 1, \quad \forall t \in \mathcal{T}$$

$$\sum_{t \in \mathcal{T}_{p,k}} \sum_{c \notin \mathcal{C}_{p,k}} m_{t,c} = \sum_{t \notin \mathcal{T}_{p,k}} \sum_{c \in \mathcal{C}_{p,k}} m_{t,c} \quad \forall p \in \mathcal{P}, k \in \mathcal{K}(p)$$

$$m_{t,c} \in \{0, 1\} \quad \forall t \in \mathcal{T}, \quad c \in \mathcal{C}.$$

- Very hard to solve (and very hard to understand!)

Advanced Maximum Cardinality Matching

$$\max \sum_{t \in \mathcal{T}} x_t$$

s.t.

$$\sum_{t \in \mathcal{T}} x_t = \sum_{c \in \mathcal{C}} y_c,$$

$$\sum_{t \in \mathcal{T}_{p,k}} x_t = \sum_{c \in \mathcal{C}_{p,k}} y_c, \quad \forall p \in \mathcal{P}, \quad k \in \mathcal{K}(p)$$

$$x_t \in \{0, 1\} \quad \forall t \in \mathcal{T}$$

$$y_c \in \{0, 1\} \quad \forall c \in \mathcal{C}.$$

$$\mathcal{K}(p) = \{\mathbf{x}_p^c\}_{c \in \mathcal{P}} \cup \{\mathbf{x}_p^t\}_{t \in \mathcal{T}}$$

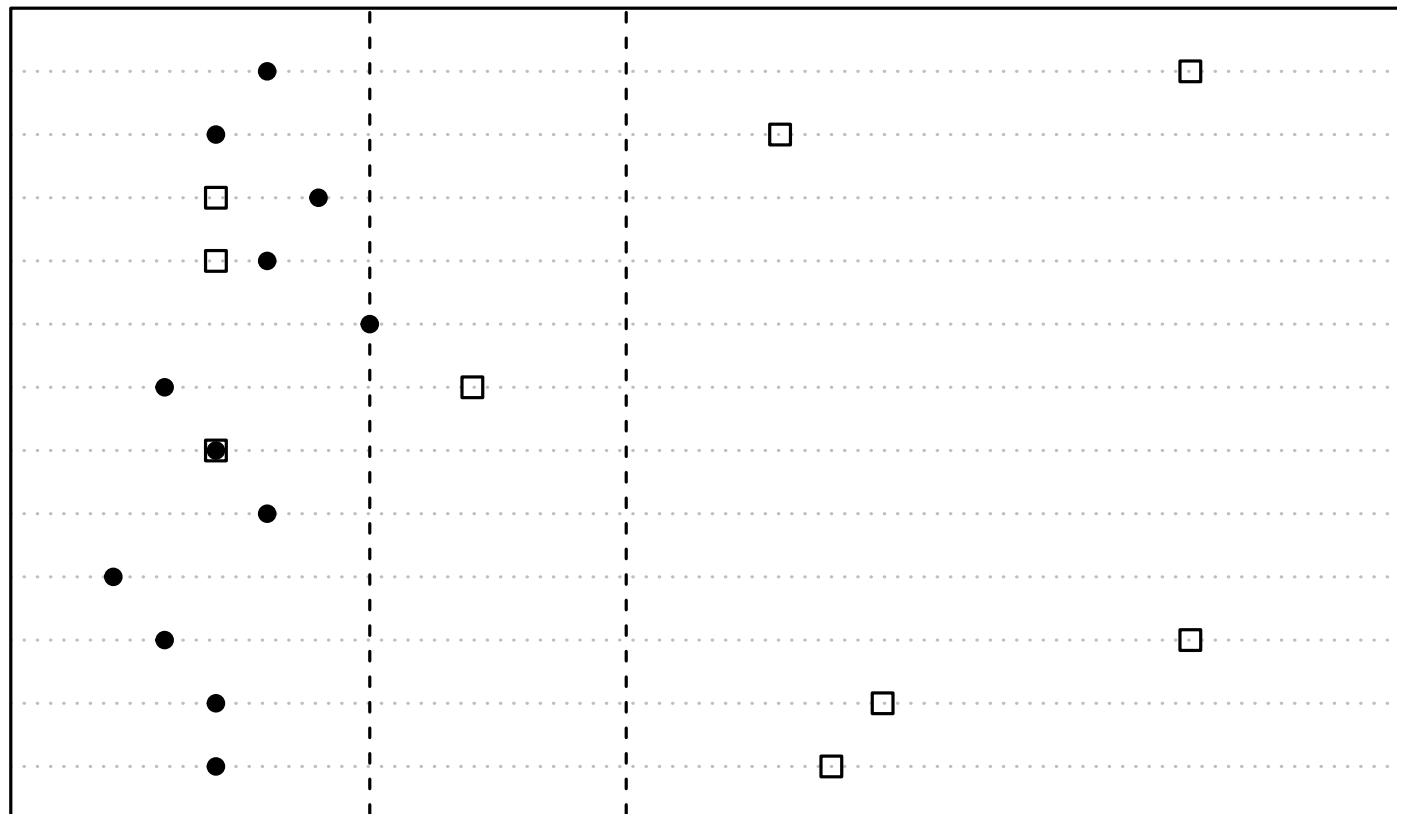
$$\mathcal{C}_{p,k} = \{c \in \mathcal{C} : \mathbf{x}_p^c = k\}$$

$$\mathcal{T}_{p,k} = \{t \in \mathcal{T} : \mathbf{x}_p^t = k\}$$

- Matching without matching variables
- Easy to solve: Small, but inherits matching properties

Balance Before After Cardinality Matching

SIMCE school (decile)
SIMCE student (decile)
GPA ranking (decile)
Attendance (decile)
Rural school
Catholic school
High SES school
Mid-High SES school
Mid SES school
Mid-Low SES school
Public School
Voucher School

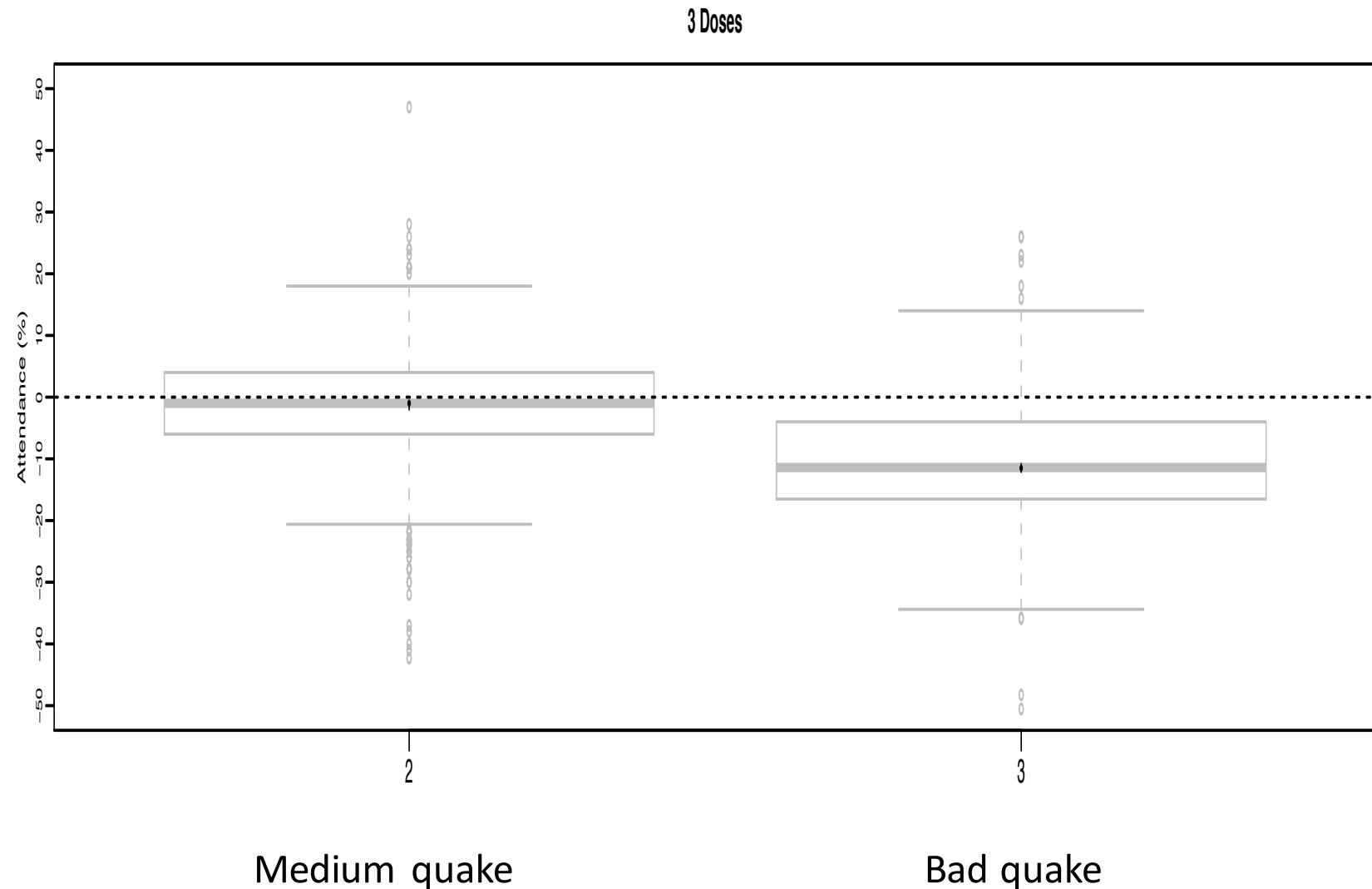


Can Also do Multiple Doses

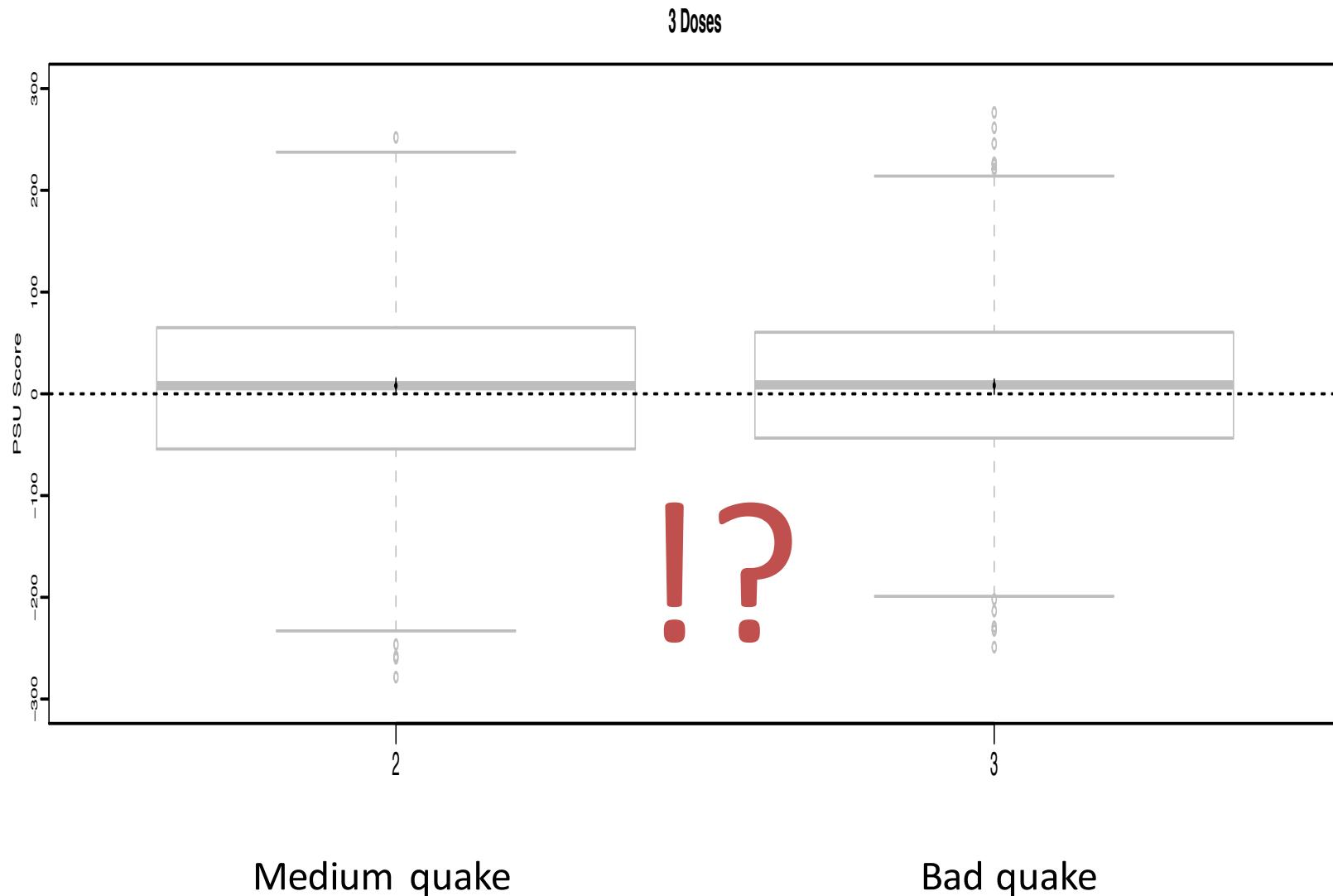
- Dose
 - 1. No quake
 - 2. Medium quake
 - 3. Bad quake

Covariate	Dose		
	1	2	3
Gender			
Male	462	462	462
Female	538	538	538
School SES			
Low	75	75	75
Mid-low	327	327	327
Medium	294	294	294
Mid-high	189	189	189
High	115	115	115
Mother's education			
Primary	335	335	335
Secondary	426	426	426
Technical	114	114	114
College	114	114	114
Missing	11	11	11
	:		

Relative (To no Quake) Attendance Impact



Relative (To no Quake) PSU Score Impact



MIP and Marketing: Chewbacca or BB-8?

Adaptive Preference Questionnaires



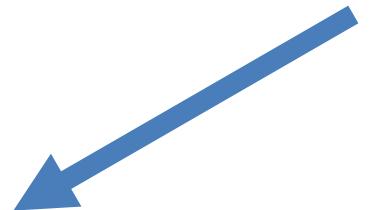
Feature	SX530	RX100
Zoom	50x	3.6x
Prize	\$249.99	\$399.99
Weight	15.68 ounces	7.5 ounces
Prefer	<input checked="" type="checkbox"/>	<input type="checkbox"/>



Feature	TG-4	G9
Waterproof	Yes	No
Prize	\$249.99	\$399.99
Weight	7.36 lb	7.5 lb
Prefer	<input type="checkbox"/>	<input checked="" type="checkbox"/>



Feature	TG-4	Galaxy 2
Waterproof	Yes	No
Prize	\$249.99	\$399.99
Viewfinder	Electronic	Optical
Prefer	<input checked="" type="checkbox"/>	<input type="checkbox"/>



We recommend:



Choice-based Conjoint Analysis (CBCA)



Feature	Chewbacca	BB-8
Wookiee	Yes	No
Droid	No	Yes
Blaster	Yes	No
I would buy toy	<input checked="" type="checkbox"/>	<input type="checkbox"/>

$$\begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} = x^2$$

Product Profile

x^1

x^2

Preference Model and Geometric Interpretation

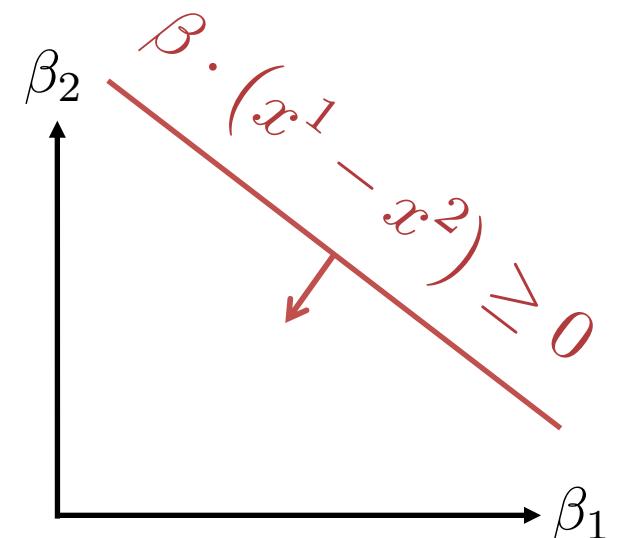
- Utilities for 2 products, d features, logit model

$$U_1 = \beta \cdot x^1 + \epsilon_1 = \sum_{i=1}^d \beta_i x_i^1 + \epsilon_1$$
$$U_2 = \beta \cdot x^2 + \epsilon_2 = \sum_{i=1}^d \beta_i x_i^2 + \epsilon_2$$

part-worths ↑
product profile ↑ noise (gumbel)

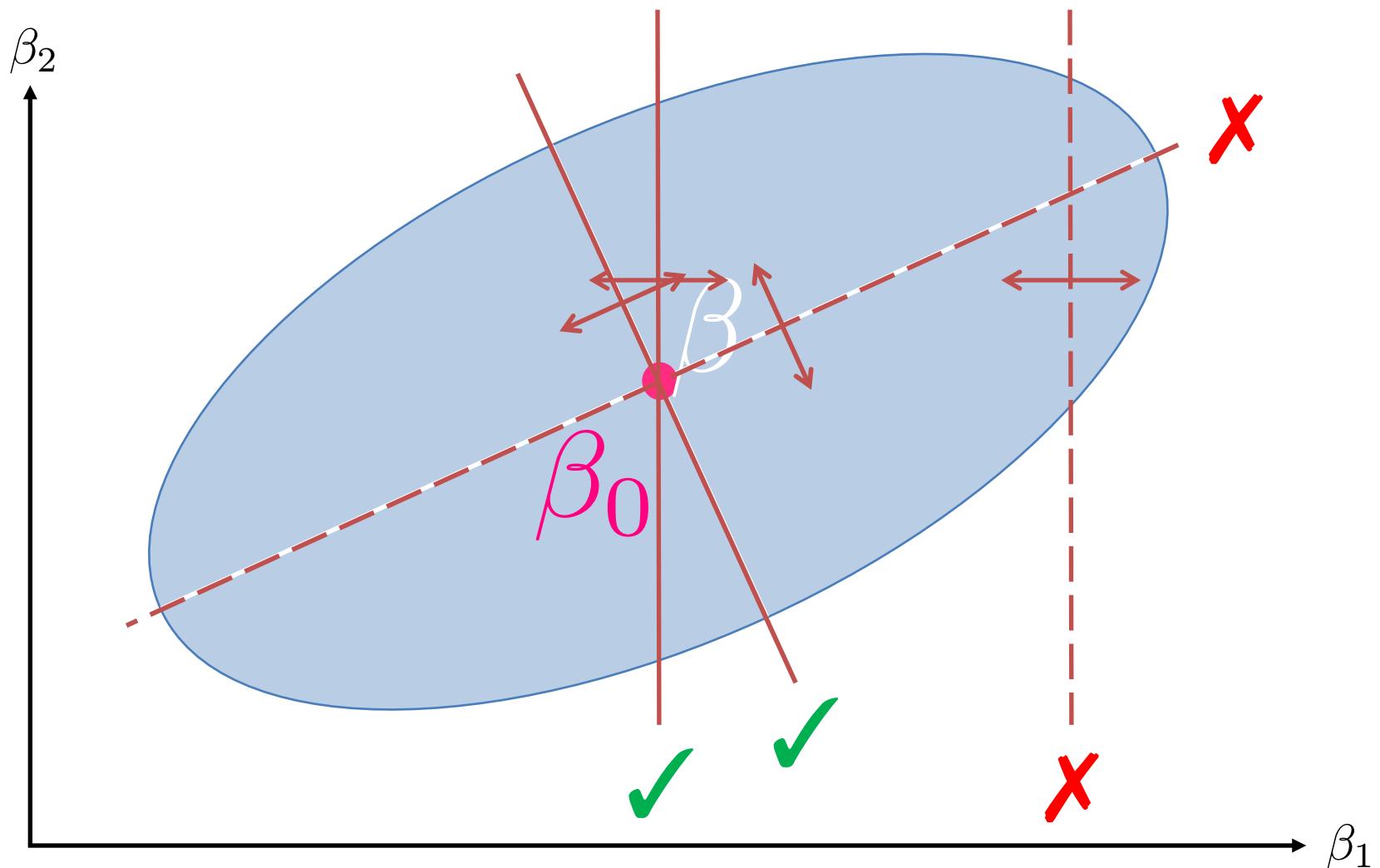
- Utility maximizing customer
 - Geometric interpretation of preference for product 1 **without error**

$$x^1 \succeq x^2 \Leftrightarrow U_1 \geq U_2$$



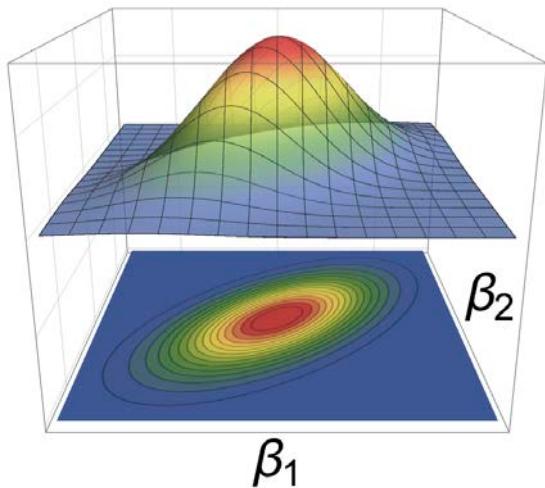
Next Question = Minimize (Expected) Volume

Good Estimator? for β ? ~~Ein Punkt für die bessere Schätzweise~~

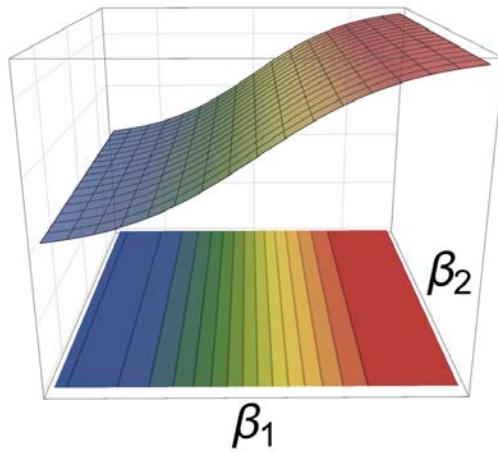


With Error = Volume of Ellipsoid $f(x^1, x^2)$

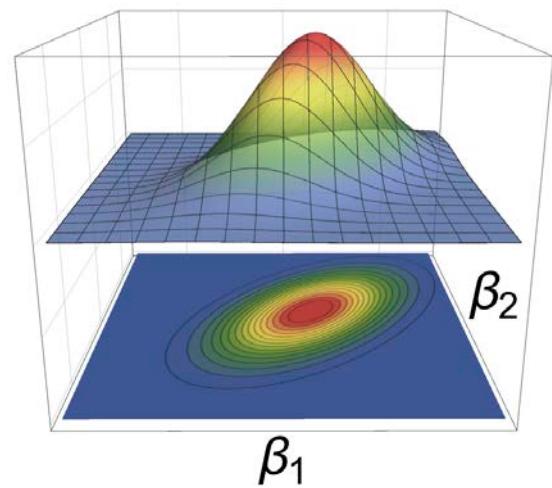
Prior distribution



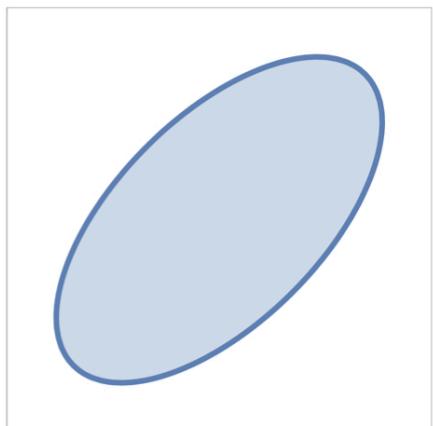
Answer likelihood



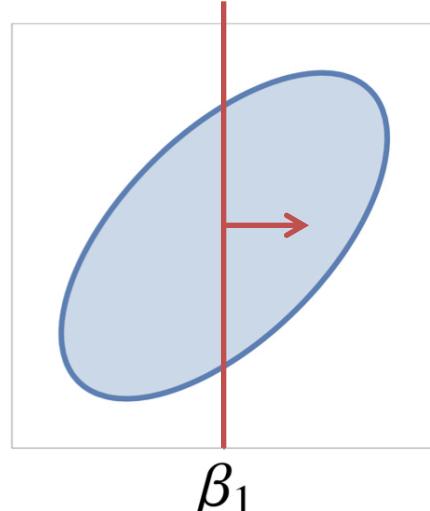
Posterior distribution



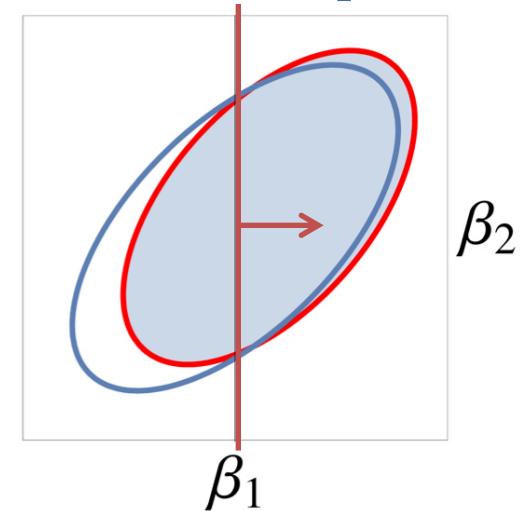
Prior ellipsoid



Question/Answer



Posterior ellipsoid

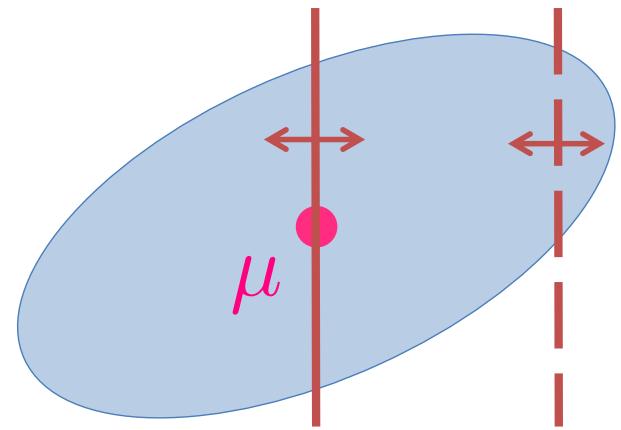


Rules of Thumb Still Good For Ellipsoid Volume

$$(\beta - \mu)' \cdot \Sigma^{-1} \cdot (\beta - \mu) \leq r$$

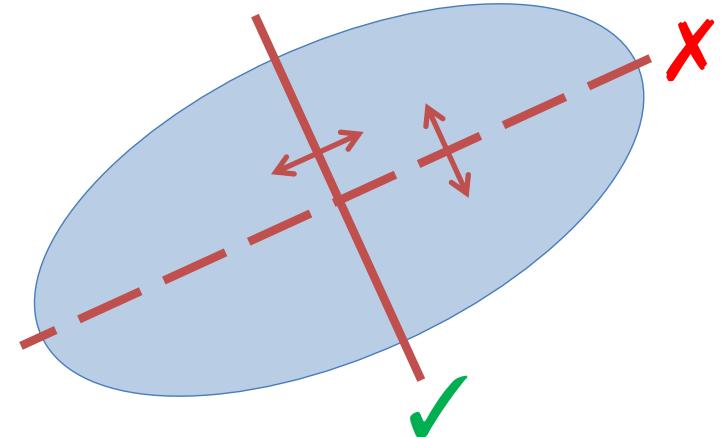
- Choice balance:
 - Minimize **distance** to center

$$\mu \cdot (x^1 - x^2)$$



- Postchoice symmetry:
 - Maximize **variance** of question

$$(x^1 - x^2)' \cdot \sum \cdot (x^1 - x^2)$$

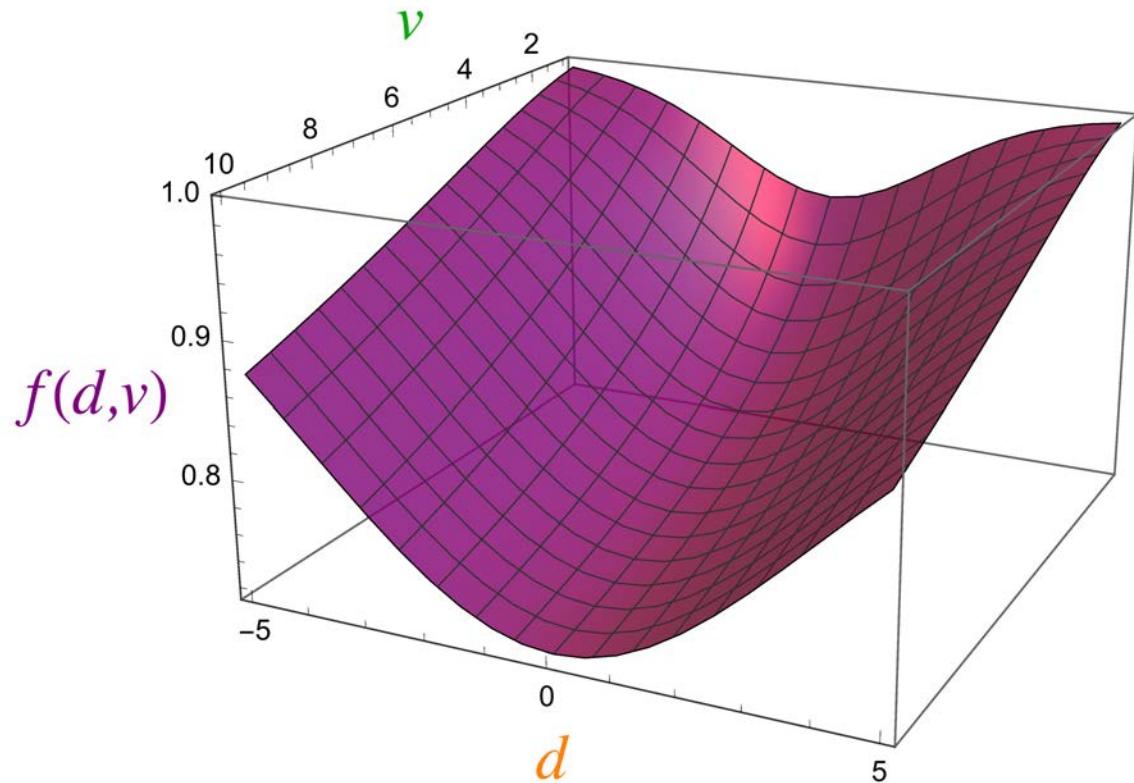


“Simple” Formula for Expected Volume

- Expected Volume = Non-convex function $f(d, v)$ of

distance: $d := \mu \cdot (x^1 - x^2)$

variance: $v := (x^1 - x^2)' \cdot \sum \cdot (x^1 - x^2)$



Can evaluate $f(d, v)$
with 1-dim integral 😊

Optimization Model

min

$$f(d, v)$$

X

s.t.

$$\mu \cdot (x^1 - x^2) = d \quad \checkmark$$

$$(x^1 - x^2)' \cdot \sum \cdot (x^1 - x^2) = v \quad \text{X}$$

$$A^1 x^1 + A^2 x^2 \leq b \quad \checkmark$$

Formulation trick:

$$\text{linearize } x_i^k \cdot x_j^l \quad x^1 \neq x^2 \quad \text{X}$$

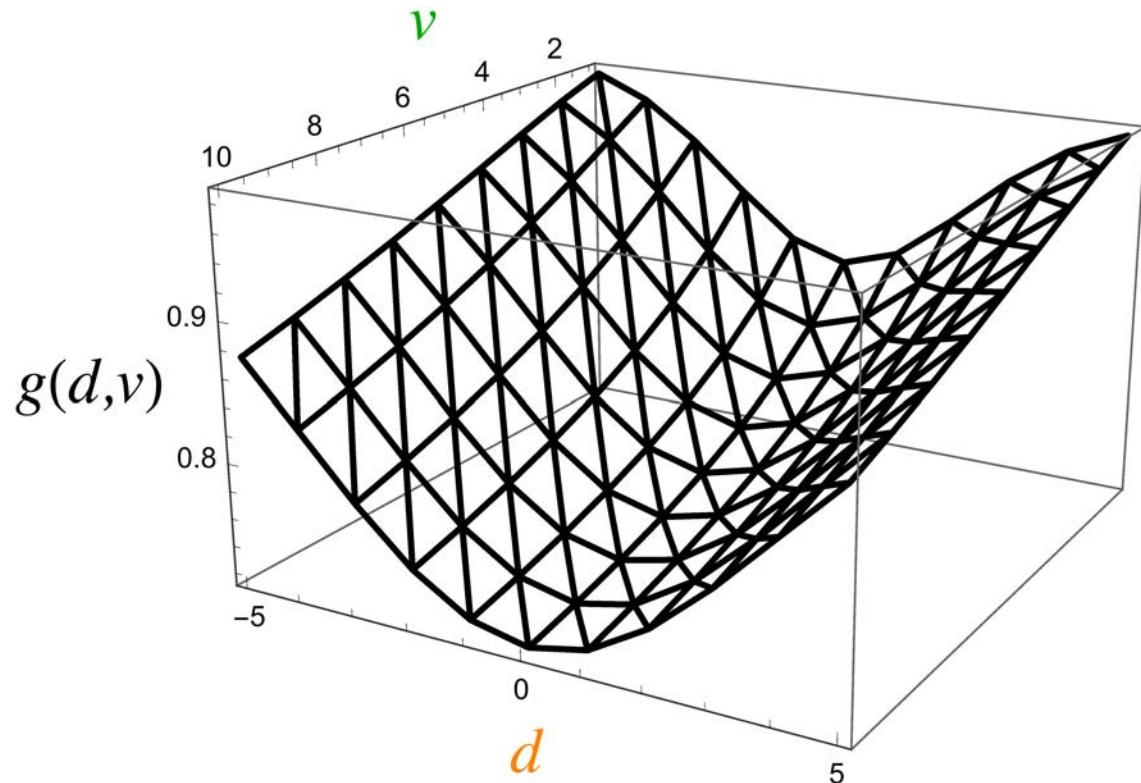
$$x^1, x^2 \in \{0, 1\}^n$$

Technique 2: Piecewise Linear Functions

- D-efficiency = Non-convex function $f(d, v)$ of

distance: $d := \mu \cdot (x^1 - x^2)$

variance: $v := (x^1 - x^2)' \cdot \sum \cdot (x^1 - x^2)$



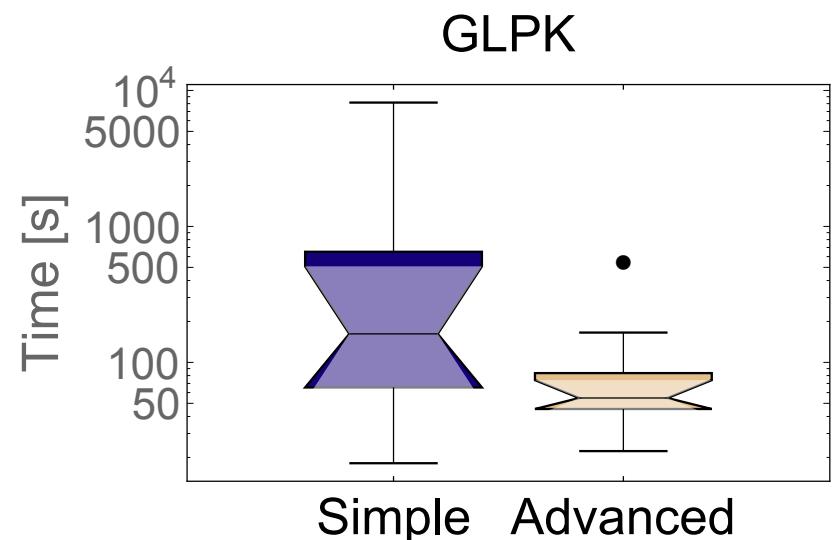
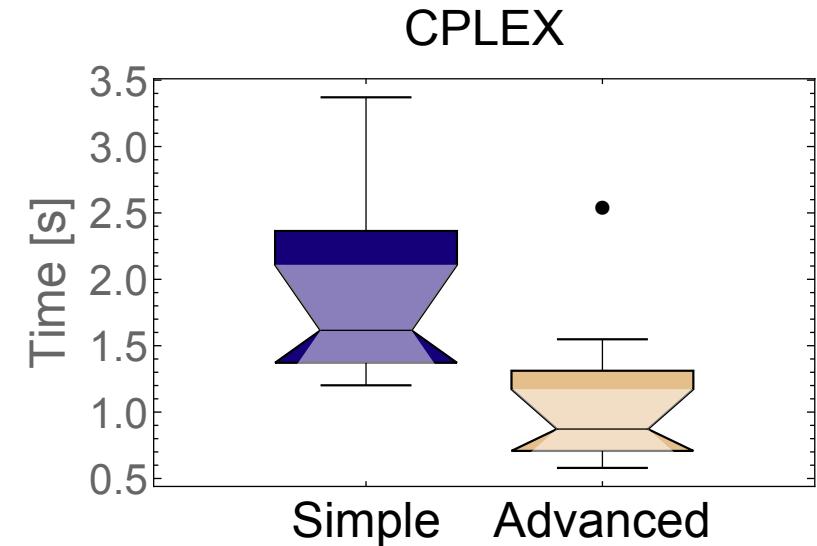
Can evaluate $f(d, v)$
with 1-dim integral 😊

Piecewise Linear
Interpolation

MIP formulation

Computational Performance

- Advanced formulations provide an computational advantage
- Advantage is significantly more important for free solvers
- State of the art commercial solvers can be significantly better than free solvers
- Still, free is free!



Summary and Main Messages

- Always choose Chewbacca!
- How to YOU use MIP / Optimization / OR / Analytics?
 - Study for the 2nd midterm!
 - Use JuMP and Julia Opt.
 - How about grad school down the river?
 - Masters of Business Analytics / OR
 - Ph.D. in Operations Research



How Hard is MIP?

How hard is MIP: Traveling Salesman Problem ?

The left side of the image shows the **AMERICAN Scientist** magazine cover. The title "Cyber-Insecurity" is prominently displayed in yellow. Below it, a subtitle reads: "The latest digital threats call for a smarter, stronger response." The right side of the image shows a Google Map of North America and the Caribbean. A blue line represents a travel route, starting in the western US, going north through Canada, then south through the Great Lakes region and down the eastern coast of the US and into the Caribbean.

Paradoxes, Contradictions, and the Limits of Science

Many research results define boundaries of what cannot be known, predicted, or described. Classifying these limitations shows us the structure of science and reason.

Noson S. Yanofsky

“A computer would have to check all these possible routes to find the shortest one.”

MIP = Avoid Enumeration

- Number of tours for 49 cities = $48!/2 \approx 10^{60}$
- Fastest supercomputer $\approx 10^{17}$ flops
- Assuming one floating point operation per tour:
 $> 10^{35}$ years $\approx 10^{25}$ times the age of the universe!
- How long does it take on an iphone?
 - Less than a second!
 - 4 iterations of **cutting plane** method!
 - Dantzig, Fulkerson and Johnson 1954 did it by hand!
 - For more info see tutorial in ConcordeTSP app
 - **Cutting planes** are the key for effectively solving (**even NP-hard**) MIP problems in practice.

50+ Years of MIP = Significant Solver Speedups

- Algorithmic Improvements (Machine Independent):
 - **CPLEX** v1.2 (1991) – v11 (2007): 29,000x speedup
 - Gurobi v1 (2009) – v6.5 (2015): 48.7x speedup
 - Commercial, but free for academic use
- (Reasonably) effective free / open source solvers:
 - GLPK, **COIN-OR (CBC)** and SCIP (only for non-commercial)
- Easy to use, fast and versatile modeling languages
 - Julia based JuMP modelling language
 - <http://julialang.org>
 - <http://www.juliaopt.org>

Technique 1: Binary Quadratic $x^1, x^2 \in \{0, 1\}^n$

$$(x^1 - x^2)' \cdot \sum \cdot (x^1 - x^2) = v$$

$$X_{i,j}^l = x_i^l \cdot x_j^l \quad (l \in \{1, 2\}, \quad i, j \in \{1, \dots, n\}) :$$

$$X_{i,j}^l \leq x_i^l, \quad X_{i,j}^l \leq x_j^l, \quad X_{i,j}^l \geq x_i^l + x_j^l - 1, \quad X_{i,j}^l \geq 0$$

$$W_{i,j} = x_i^1 \cdot x_j^2 :$$

$$W_{i,j} \leq x_i^1, \quad W_{i,j} \leq x_j^2, \quad W_{i,j} \geq x_i^1 + x_j^2 - 1, \quad W_{i,j} \geq 0$$

$$\sum_{i,j=1}^n (X_{i,j}^1 + X_{i,j}^2 - W_{i,j} - W_{j,i}) \sum_{i,j} = v$$

Technique 1: Binary Quadratic $x^1, x^2 \in \{0, 1\}^n$

$$x^1 \neq x^2 \iff \|x^1 - x^2\|_2^2 \geq 1$$

$$X_{i,j}^l = x_i^l \cdot x_j^l \quad (l \in \{1, 2\}, \quad i, j \in \{1, \dots, n\}) :$$

$$X_{i,j}^l \leq x_i^l, \quad X_{i,j}^l \leq x_j^l, \quad X_{i,j}^l \geq x_i^l + x_j^l - 1, \quad X_{i,j}^l \geq 0$$

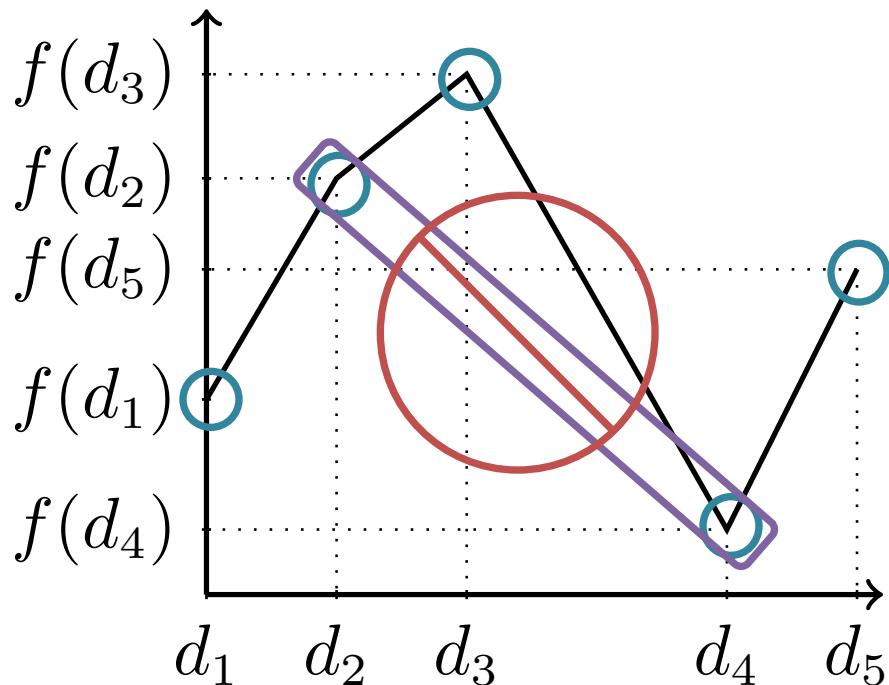
$$W_{i,j} = x_i^1 \cdot x_j^2 :$$

$$W_{i,j} \leq x_i^1, \quad W_{i,j} \leq x_j^2, \quad W_{i,j} \geq x_i^1 + x_j^2 - 1, \quad W_{i,j} \geq 0$$

$$\sum_{i,j=1}^n (X_{i,j}^1 + X_{i,j}^2 - W_{i,j} - W_{j,i}) \geq 1$$

Simple Formulation for Univariate Functions

$$z = f(x)$$



Size = $O(\# \text{ of segments})$

Non-Ideal: Fractional Extreme Points

$$\binom{x}{z} = \sum_{j=1}^5 \binom{d_j}{f(d_j)} \lambda_j$$

$$1 = \sum_{j=1}^5 \lambda_j, \quad \lambda_j \geq 0$$

$$y \in \{0, 1\}^4, \quad \sum_{i=1}^4 y_i = 1$$

$$0 \leq \lambda_1 \leq y_1$$

$$0 \leq \lambda_2 \leq y_1 + y_2$$

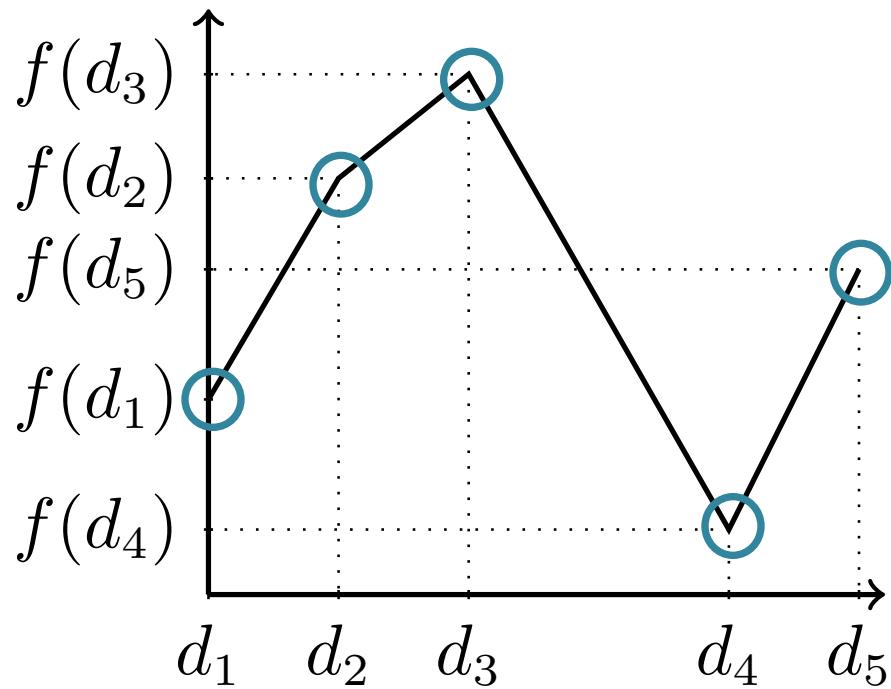
$$0 \leq \lambda_3 \leq y_2 + y_3$$

$$0 \leq \lambda_4 \leq y_3 + y_4$$

$$0 \leq \lambda_5 \leq y_4$$

Advanced Formulation for Univariate Functions

$$z = f(x)$$



$$\begin{pmatrix} x \\ z \end{pmatrix} = \sum_{j=1}^5 \begin{pmatrix} d_j \\ f(d_j) \end{pmatrix} \lambda_j$$

$$1 = \sum_{j=1}^5 \lambda_j, \quad \lambda_j \geq 0$$

$$y \in \{0, 1\}^2$$

$$0 \leq \lambda_1 + \lambda_5 \leq 1 - y_1$$

$$0 \leq \lambda_3 \leq y_1$$

$$0 \leq \lambda_4 + \lambda_5 \leq 1 - y_2$$

$$0 \leq \lambda_1 + \lambda_2 \leq y_2$$

Size = $O(\log_2 \# \text{ of segments})$

Ideal: Integral Extreme Points